Numerical algorithms for the treatment of parametric multiobjective optimization problems and applications

Von der Fakultät für Elektrotechnik, Informatik und Mathematik
der Universität Paderborn
zur Erlangung des akademischen Grades

eines Doktor der Naturwissenschaften
– Dr. rer. nat. –

genehmigte Dissertation

von
Katrin Witting

Paderborn 2012

Datum der Einreichung: 14.12.2011
Tag der mündlichen Prüfung: 22.02.2012

Gutachter: Prof. Dr. Michael Dellnitz
Universität Paderborn

Prof. Dr. Kathrin Klamroth
Bergische Universität Wuppertal

Abstract

Nowadays, mathematical optimization plays an important role in many applications. Often not only one objective is desired to be optimized but several aims have to be considered at the same time. For example, in manufacturing not only costs should be minimized but at the same time the product should be of high quality. The development of theoretic and algorithmic principles for the mathematical description of these problems is the concern of *multiobjective optimization*. In the present thesis, different multiobjective optimization problems from mechanical engineering are studied. Motivated by the example of the operating point assignment of a linear drive, the focus of this thesis lies on the study of *parametric multiobjective optimization problems*. Firstly, a new approach is presented which allows to solve time-dependent multiobjective optimization problems by use of numerical path following algorithms. Subsequently, the computation of solutions which change as little as possible under variations of the external parameter is another central topic of this thesis. For the numerical approximation of these so-called *robust Pareto points*, in this work two new approaches are presented. The first approach is based on the classical calculus of variations while the second one makes use of numerical path following methods. Finally, geometrical properties of the solution set of non-convex, parametric multiobjective optimization problems are studied and connections to bifurcation theory are pointed out.

Zusammenfassung

Heutzutage spielt die mathematische Optimierung in vielen Anwendungen eine wesentliche Rolle. Häufig ergibt sich der Wunsch, nicht nur ein einziges Ziel sondern mehrere Ziele gleichzeitig zu optimieren. So sollten beispielsweise bei der Fertigung eines Produktes nicht nur die Kosten minimiert werden, sondern gleichzeitig sollte auch ein qualitativ hochwertiges Produkt entstehen. Die Entwicklung von theoretischen und algorithmischen Grundlagen für die mathematische Beschreibung und Lösung solcher Probleme bietet die *Mehrzieloptimierung*. In der vorliegenden Arbeit werden verschiedene Mehrzieloptimierungsprobleme aus dem ingenieurwissenschaftlichen Bereich untersucht. Motiviert durch das Beispiel der Arbeitspunktsteuerung eines Linearmotors liegt der Fokus dieser Arbeit auf der Studie *parameterabhängiger Mehrzieloptimierungsprobleme*. Es wird zum einen ein neuer Ansatz vorgestellt, der mittels Verwendung von Algorithmen zur numerischen Pfadverfolgung die Lösung zeitabhängiger Mehrzieloptimierungsprobleme erlaubt. Zum anderen ist die Bestimmung von Lösungen, die sich gegenüber Schwankungen eines externen Parameters möglichst wenig verändern, ein weiteres zentrales Thema dieser Arbeit. Für die numerische Approximation dieser sogenannten *robusten Paretopunkte* werden in der vorliegenden Arbeit zwei neuartige Ansätze präsentiert, die zum einen auf der Variationsrechnung und zum anderen auf numerischer Pfadverfolgung basieren. Abschließend werden geometrische Eigenschaften der Lösungsmenge nicht-konvexer, parametrischer Mehrzieloptimierungsprobleme untersucht und Zusammenhänge zur Verzweigungstheorie aufgezeigt.

Acknowledgments

First of all, I wish to express my gratitude to my parents Dagmar and Wilfried Baptist. Many, many thanks for your love, patience and support in all aspects of my life. Just as much I would like to thank my beloved husband Sven who acted as a loving motivator (and a perfect contact person for any computer problems). With love I thank our son Benjamin who makes me happy every day through his cheerful nature.

I am greatly indebted to Prof. Dr. Michael Dellnitz for giving me the chance to write this thesis under his supervision. I thank him for suggesting the problem and for many stimulating conversations. Very fruitful discussions have also taken place with my colleagues. Many thanks to Prof. Dr. Oliver Schütze who once introduced me to the topic of multiobjective optimization and was my first contact person at the beginning. I also wish to express my special thanks to Dr. Mirko Hessel–von Molo who seems to know everything. Thank you, Mirko, for discussing many, many scientific (and other) questions. Especially, two products of this type of discussion were the construction of Examples 4.24 and 4.26, and the geometrical interpretation of Condition (4.24) on page 96. I would also like to thank Jun.-Prof. Dr. Sina Ober-Blöbaum who spent her time to explain how to formulate the discretized Euler-Lagrange equations, and this way supported me in formulating the robustness concept described in Paragraph 4.4.2. Further, I thank my other colleagues and ex-colleagues Tanja Bürger, Dr. Roberto Castelli, Dr. Alessandro Dell'Aere, Dr. Laura Di Gregorio, Helene Derksen-Riesen, Kathrin Flaßkamp, Sebastian Hage-Packhäuser, Christian Horenkamp, Prof. Dr. Oliver Junge, Marianne Kalle, Dr. Stefan Klus, Dr. Arvind Krishnamurthy, Anna-Lena Meyer, Jun.-Prof. Dr. Kathrin Padberg-Gehle, Pierpaolo Pergola, Michael Petry, Dr. Marcus Post, Prof. Dr. Robert Preis, Marcel Schwalb, Stefan Sertl, Simon Sonntag, Robert Timmermann, Bianca Thiere, Dr. Wang Fang, and Anna Zanzottera for the good colleagueship, friendship and support in not only scientific questions.

This work was developed in the course of the Collaborative Research Center 614 – Self-Optimizing Concepts and Structures in Mechanical Engineering – University of Paderborn, and funded by the Deutsche Forschungsgemeinschaft. Thus, my work in large part was interdisciplinary. I wish to thank the Chair of Power Electronics and Electrical Drives (especially Bernd Schulz, Tobias Schneider, Christoph Romaus, Dr.-Ing. Andreas Pottharst, Dr.-Ing. Rongyuan Li and Dr.-Ing. Tobias Knoke), the Chair of Control Engineering and Mechatronics (especially Jens Geisler, Dr.-Ing. Oliver Oberschelp, Henner Vöcking, Peter Reinold and Martin Krüger) and the Department

of System and Circuit Technology (especially Dr.-Ing. Matthias Blesken) for their cooperation, discussions and motivation through technical applications. I am very grateful to the student research assistents that worked within our research project over the last years. Namely, I want to thank Albert Seifried and Dominik Steenken. Thanks, Albert, for the execution of numerous computations, for documenting existing code and for supporting me with your knowledge about the creation of pretty Matlab figures. Thanks, Dominik, for developing a parallelized version of our multiobjective optimization software which speeds up most of the computations significantly, and for maintaining the cooperation with Matthias Blesken and hereby producing many interesting results within your diploma thesis (which partly are mentioned in my thesis as well).

A special thanks goes to all my friends, primarily Michael Mair for proofreading the manuscript. Also, at this point I would like to thank Marc Pfetsch who put the LaTeX sources for setting his thesis into the web. It was very helpful for me to use these settings as a basis for my own LaTeX style.

Contents

List of Acronyms

CRC	collaborative research center
HES	hybrid energy storage system
h. o. t.	higher order terms
iff	if and only if
MOP	multiobjective optimization problem
NLO	nonlinear optimization problem
$NLO^{=}$	equality-constrained nonlinear optimization problem
ParMOP	parametric multiobjective optimization problem
$uPOP$	unconstrained parametric optimization problem
$uNLO$	unconstrained nonlinear optimization problem
VC	problem of variational calculus, variational problem
VC_{2p}	variational problem with fixed end points
VC_{2p}-c	variational problem with fixed end points with equality constraints
VC_{ff}-c	variational problem with free end points with equality constraints
wlog	without loss of generality

List of Symbols

$\mathbb{1}$	identity matrix
$\|x\| = \|x\|_2$	Euclidean norm of a vector $x \in \mathbb{R}^n$
$< \cdot, \cdot >$	Euclidean scalar product
α	weight vector
A^T	transpose of the matrix $A \in \mathbb{R}^{m \times n}$
C^m	m-times continuously differentiable functions
$\delta_j f(p)$	partial derivative of $f : \mathbb{R}^n \to \mathbb{R}$ with respect to the j'th variable at the point p
$\frac{\partial f}{\partial x_i}(p)$	partial derivative of $f : \mathbb{R}^n \to \mathbb{R}$ with respect to the variable x_i at the point p
im A	image of the matrix $A \in \mathbb{R}^{m \times n}$, i. e. $\{Ax \| x \in \mathbb{R}^n\}$
ker A	kernel of the matrix $A \in \mathbb{R}^{m \times n}$, i. e. $\{x \in \mathbb{R}^n \| Ax = 0\}$
T	time
t, or $(t_1^2, \dots t_k^2)$ respectively	weight vector
P_λ	λ-dependent Pareto set
PF_λ	λ-dependent Pareto front
$\mathbb{R}_+$	Set of all positive real numbers
$\mathbb{R}_{+,0}$	Set of all real numbers that are ≥ 0
$\mathbb{R}^d_{incl} \subseteq \mathbb{R}^k$	$\{x \in \mathbb{R}^k : x_{d+1} = \dots = x_k = 0\}$ (canonical inclusion)
S_λ	λ-dependent set of substationary points
$\overline{B}$	closure of a set B
$V^\perp$	orthogonal complement of the vector space V

Chapter 1

Introduction

In almost any application optimization plays an important role. In a variety of these not only one objective but several ones are desired to be optimal at the same time. For instance, in manufacturing *cost* has to be minimized, but at the same time also *quality* is desired to be maximal – at least to a certain degree. The development of theory and algorithms for the determination of solutions that are as good as possible with respect to all objectives is the task of *multiobjective optimization*. Mathematically, a multiobjective optimization problem is given as

$$\min_x \{F(x) : x \in S \subseteq \mathbb{R}^n\},$$

where F is defined as the vector of the objective functions $f_1, \ldots, f_k$, $k \geq 2$, which each map from $\mathbb{R}^n$ to $\mathbb{R}$, and S denotes the feasible region. The example mentioned above already illustrates that the several objectives typically contradict each other and thus do not have identical optima. Consequently, the solution of a multiobjective optimization problem is given by the set of optimal compromises of the objectives, the so-called *Pareto set*. In the case of minimization problems the Pareto set is given by the set of solutions in which the value of any objective function can only be decreased at the cost of increasing another one.

To obtain solutions that lie within the Pareto set many algorithms have been developed. There are essentially two different types: algorithms that allow for the computation of only one or a few Pareto points, and algorithms that approximate the entire Pareto set. In the first case often a priori information, such as a specific weighting of the objectives or some kind of ordering, is required. Examples for those methods are the 'weighted sums method', the 'ε-constraint-method' and the 'lexicographic ordering' (see [69], [24]). Over the last years algorithms that are able to approximate the entire Pareto set (restricted to a moderate number k of objectives) have been developed (see e. g. [14, 85, 15, 32, 12, 112]). For the computations of Pareto sets in the examples given in this thesis set-oriented, numerical methods which are implemented in the software package GAIO[1] have been used (see [92]),

[1] Global Analysis of Invariant Objects, see www.math.upb.de/~agdellnitz

and Section 3.2.3 for more details).

For the development of modern, innovative mechatronic systems, the concept of *self-optimization* has been introduced (cf. [1]). A technical system is self-optimizing whenever three steps are repeated iteratively during operation: the analyis of the current situation (Step 1), the determination of the system's objectives (Step 2) and the adaptation of the system's behavior (Step 3). Thus, self-optimization goes beyond the classical adaptive control in which the system's objectives are not changing during runtime. Multiobjective optimization plays an important role in the self-optimization process. One possibility to realize Step 3 – for a technical system which can be described by a mathematical model – is to choose different solutions during operation time from a previously approximated Pareto set. One technical application that serves as an example to demonstrate self-optimization is the *RailCab system*. The RailCab is a novel linear motor driven railway system including the latest technologies and innovative prototypes, developed by the project RailCab ("Neue Bahntechnik Paderborn" [73]). Many subsystems contained in the RailCab provide great potential for self-optimization. In this thesis, in particular three different subsystems of the RailCab have been studied in close collaboration with the engineers developing these systems: the hybrid energy storage system, the guidance module and the linear drive.

The *hybrid energy storage system* is intended to guarantee the availability of energy without any interruption. It consists of batteries which are well-suited for long term storage and double layer capacitors which perfectly fit the purpose of short term storage. The mathematical challenge is computing an operation strategy that distributes the power flows to the respective storage devices in an optimal way during a drive of the vehicle. In this thesis the simultaneous minimization of the normalized energy losses of the hybrid energy storage system and maximization of the power reserve has been studied as a multiobjective optimization problem in close cooperation with the Chair of Power Electronics and Electrical Drives, University of Paderborn. Making use of a special numerical, image set oriented predictor-corrector method for multiobjective optimization Pareto optimal operation strategies for a given total power profile have been approximated. In Section 3.4.2 the results of the multiobjective optimization of the hybrid energy storage system are given for an example of a track section of one minute.

The *guidance module* allows to actively control the lateral displacement of the RailCab vehicle within the rails. Thus, the RailCab can be steered along

arbitrary reference trajectories within the maximum distance between the flanges and the rail-heads, the so-called clearance. How to choose these reference trajectories has been considered in this thesis in close cooperation with the Chair of Control Engineering and Mechatronics, University of Paderborn (see Section 3.4.3). To obtain good trajectories four different objectives have been taken into account: maximization of the passengers comfort measured through the structure acceleration, minimization of the deviation from the track centerline, minimization of the average energy consumption of the hydraulic actuators, and maximization of the steering angle reserve. As a start, an optimal control problem of the form

$$\begin{aligned} &\min_{x\in\mathbb{R}^n, u\in\mathbb{R}^m} J(x,u) \\ &\text{s. t. } \dot{x} = f(x,u) \end{aligned}$$

is formulated in which the differential equation is given by a linear model for the lateral dynamics of the RailCab vehicle, and the functional J is vector-valued. Linear systems $\dot{x} = f(x,u)$ are differentially flat, which means that the inputs and controls can be computed directly from the outputs. Thus, the trajectory optimization of the guidance module can be reformulated into a multiobjective optimization problem in which only the outputs are optimized. By this approach we have shown how the work of Murray et al on trajectory optimization for differentially flat systems (see [72, 104, 70, 31]) can be extended to the multiobjective case. In Section 3.4.3 the results of the multiobjective optimization of the reference trajectories for an exemplary track section are presented.

The RailCab vehicle is propelled by a doubly-fed *linear drive* which consists of a primary motor part between the rails of the track and a secondary part on the vehicle. The power transfer from the primary to the secondary motor part depends on the operating point of the linear motor which is given through the currents and frequencies in both motor parts. The operating points are desired to maximize both the degree of efficiency of the linear motor and the power factor. In this work, this multiobjective optimization problem has been addressed in close cooperation with the Chair of Power Electronics and Electrical Drives, University of Paderborn. The amount of power that has to be transferred into the secondary motor part depends on the maneuver the vehicle is intended to perform. Thus, the objective functions depend on the planned maneuver which consists of the planned course of velocities, thrusts and demanded powers. These dependencies can be reduced to time-dependency, i. e. the objective functions depend on one external parameter. In a first study presented in Section 3.4.4 the approximation of

Pareto sets for discretized points in time has been studied. A certain new *decision heuristic* has been developed in this work that allows to choose one Pareto point from the entire Pareto set depending on the actual situation. Here, two different parameters that are not considered in the optimization process come into play: the temperature of the secondary motor part and the load value of a battery which is installed at the motor to guarantee constant power supply even if the connection to the primary motor part is disrupted. Also, the shape of the Pareto front is taken into account in the decision heuristic such that no points in very flat or steep regions are chosen (under the assumption that each objective is scaled adequately). To sum up, the decision heuristic allows an automated, situation-dependent choice of Pareto optimal solutions during operation time, and hereby is an important element of the self-optimization of the linear motor.

In order to generate a satisfactory approximation of operating points for a drive of the linear motor, the computation of Pareto sets would be necessary for very many discretization points in time, approximately every one or two milliseconds. This computation cannot be realized during runtime as the computational effort is too high. An a priori approximation of the Pareto sets, its storage and an application of the decision heuristic during runtime on the stored data is not preferable because arbitrarily many different maneuvers have to be considered.

Motivated hereby, in this thesis a new mathematical concept has been developed which allows the treatment of *time-dependent multiobjective optimization problems*. This concept allows the computation of Pareto optimal solutions over time and can in the case of the linear drive be realized during operation time. Mathematically speaking, a one-parametric multiobjective optimization problem has to be solved which can be formulated as

$$\min_x \{F(x,\lambda) : x \in \mathbb{R}^n, \lambda \in [\lambda_{\text{start}}, \lambda_{\text{end}}] \subseteq \mathbb{R}\}$$

with $F : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^k$, $F(x,\lambda) = (f_1(x,\lambda), \ldots, f_k(x,\lambda))$.

In the case of the linear drive, the external parameter λ is given by time. The Pareto set of a parametric multiobjective optimization problem obviously depends on the external parameter λ. A necessary condition for Pareto optimality is given by the famous Kuhn-Tucker equations (see [63]) which indicate that in each Pareto optimal point the gradients of the objective functions are linearly dependent. In the case of parametric multiobjective optimization problems this system of equations also depends on the external parameter λ

and is given by

$$H_{\mathrm{KT}}(x, \alpha, \lambda) = \sum_{i=1}^{k} \alpha_i \nabla_x f_i(x, \lambda) = 0,$$

where $\sum_{i=1}^{k} \alpha_i = 1$ and $\alpha_i \geq 0 \; \forall i = 1, \ldots, k$.

Solutions x which satisfy the Kuhn-Tucker equations are called *substationary points*. One of the main ideas presented in this thesis is to apply *numerical path following* methods (cf. [22, 2, 75]) to the parameter-dependent Kuhn-Tucker necessary condition. Numerical path following methods allow the approximation of λ-dependent solution paths for systems of equations with one more unknowns than equations. The solution paths are obtained by iteratively repeating predictor and corrector steps. Starting from a known point $(x^\star, \lambda^\star)$ on the solution curve in the predictor step a point (x_p, λ_p) with $\lambda_p = \lambda^\star + h$ is computed, where h is an adequate step size. This point serves as an initial guess for the numerical solution of the system of equations with a fixed value $\lambda = \lambda_p$ which is performed in the so-called corrector step.

The Kuhn-Tucker equations do not only depend on the optimization parameter x and the external parameter λ but also on the weight vector α. Different weight vectors lead to the computation of different substationary points. Thus, the main challenge in applying numerical path following methods to the Kuhn-Tucker equations lies in the treatment of the weight vector α. This vector characterizes the direction in which a solution path is "steered" through the λ-dependent set of substationary points. Consequently, additionally to the Kuhn-Tucker equations some more equations have to be derived which fix the direction of the solution path. How to define such a direction through additional equations depends on the application under consideration.

For time-dependent multiobjective optimization problems like in the case of the linear drive it turns out to be a good idea to consider paths with fixed weights over certain time periods. Here, first an initial Pareto set is approximated. During operation time, a suitable point within this set is chosen by the decision heuristic and the corresponding weight vector obtained by solving the Kuhn-Tucker equations is set fixed. By use of classical predictor-corrector path following methods a solution path over time is computed and at the respective points in time the corresponding solutions are adjusted to the motor. A recomputation of the entire Pareto set is only performed if variables included in the decision heuristic change drastically, or if the optimization setup changes due to other technical reasons. Performing these steps during operation of the linear drive the system becomes self-optimizing. More detailed information about this application is given in Section 4.3.2.

Besides time, the parameter λ can describe the influence of any other param-

eter to the objective functions. Imagine for instance the design process of a car which is desired to be optimal with respect to comfort (first objective) and safety (second objective). On the road, the car will be caught in a crossfire of influences, like side wind, wet roads, or changes in temperature. The only information one might have is that these influences can be estimated to lie within a certain interval, probably given by the weather forecast. For the designer of this car subsets of Pareto points are especially desirable, for which the Pareto points or their images under F change as little as possible while the parameter varies. This fictitious example already inspires the development of special **robustness concepts for parametric multiobjective optimization problems** which is the main matter of this thesis. The study of the robustness of solutions is a relatively young area of research in multiobjective optimization, especially in evolutionary multiobjective optimization (cf. [16], [29] for example). Following [10], multiobjective optimization problems can be formulated as

$$\min_{x',x''} F(x', x'' + \varepsilon, \lambda)$$

where $x' \in \mathbb{R}^{n_1}$ are deterministic optimization variables, $x'' + \varepsilon$ describes some kind of stochastic optimization variables with $x'' \in \mathbb{R}^{n_2}$, $n_1 + n_2 = n$, and $\lambda \in \mathbb{R}^p$ denotes external parameters. Different kinds of robustness for multiobjective optimization problems are studied in the literature (see Section 4.4.1 for an overview). This work restricts itself to the case of one real, external parameter λ. None of the concepts developed in the literature includes the condition that the solutions should stay Pareto optimal w. r. t. the perturbed parameter values for the original objective functions. In this thesis, two new robustness concepts have been developed which measure how much a Pareto optimal solution (or, a solution on the Pareto front) has to change when demanding Pareto optimality within a certain interval $[\lambda_{\text{start}}, \lambda_{\text{end}}]$ of the external parameter. Loosely speaking, the more robust a Pareto point for a fixed external parameter value is the less it changes to other Pareto points for different values of the external parameter within a given interval. The same considerations can be transferred to the study of robustness on Pareto fronts. In both cases, the techniques used to obtain robust solutions are completely different from the ones used in the literature. The first concept is based on the calculus of variations whereas in the second concept again numerical path following methods are employed for a certain system of equations.

More precisely, in the first concept the variational integral

$$\min_{x,\alpha} \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} \|x'(\lambda)\|_2 \, \mathrm{d}\lambda$$
$$\text{s. t. } H_{\text{KT}}(x(\lambda), \alpha(\lambda), \lambda) = 0$$

with the boundary conditions that $x(\lambda_{\text{start}})$ and $x(\lambda_{\text{end}})$ are solutions of the Kuhn-Tucker equations for the respective parameter values, is studied. This means that the length of the curve of substationary points on the parameter-dependent solution set of the Kuhn-Tucker equations is minimized. A curve $x(\lambda)$ which solves this variational problem is called a λ-*robust substationary point w. r. t. Concept I.*

For the computation of solutions of this constrained variational problem the multiplier rule (cf. [9]) is used and the Euler-Lagrange equations are derived. They are given as a system of differential algebraic equations that provide a necessary condition for optimality of the variational integral. To allow numerical computations of λ-robust substationary points the variational integral is discretized following the lines of [68] and the discrete Euler-Lagrange equations are derived. One ends up with a nonlinear system of equations which can be solved with Newton's method, for example. In Section 4.4.2 the derivation of the discrete Euler-Lagrange equations for constrained variational problems is described in detail and numerical examples are given.

A second robustness concept is developed in Section 4.4.2. In contrast to the variational approach of the first concept, here only local information is taken into account: solution curves are computed that stay at the same point constantly as long as this point is feasible under the variation of the external parameter λ. If a point on the curve is not feasible for an increased value of λ, then the solution path consists of those points that vary as little as necessary compared to the previously computed point. The robustness of a Pareto point against variations of the external parameter can be measured through the length of these paths. In order to realize the numerical approximation of solution paths, again numerical path following methods can come into play. In order to force the path into the direction in which the Pareto point changes as little as possible for increasing values of λ, the Kuhn-Tucker system has to be expanded by certain equations.

To attack this problem, first the Pareto sets for two different values λ_0 and λ_1 of the external parameter have been considered. A constrained nonlinear optimization problem is formulated in which the distance of any point on the set of substationary points for λ_1 to the Pareto set for λ_0 is minimized. By using a classical necessary condition for optimality of this problem a system of equations is obtained that characterizes the solution on the set of

substationary points for $\lambda = \lambda_1$ that is located as near as possible to the Pareto set for λ_0 in x-space.

When generating a path of solutions that step by step changes as little as possible in x-space the same situation occurs repeatedly. A predictor-corrector method is used to compute the solution path. In the predictor step the value of λ is increased by an adequate step size h. In the corrector step the system of equations consisting of the Kuhn-Tucker equations and the distance minimization equations mentioned above is solved for the predicted value of λ. For the predictor-corrector method it is required that the Jacobian matrix of the system of equations is regular. In this thesis, the structure of the Jacobian matrix for the new system of equations has been studied. It is proven that the Jacobian is indeed regular if a second order sufficient condition for the underlying minimization problem is satisfied.

The new concept is illustrated for several numerical examples. Finally, in Section 4.4.3 an application of the second robustness concept in the design of integrated circuits is given. Here, the robustness of Pareto points with respect to changes in temperature and supply voltage has been computed.

In Section 4.4.2 the two robustness concept developed within this thesis are compared. To make this possible the variational problem of Concept I is reformulated for fixed starting points on the initial Pareto set. It it shown that the length of solution paths with respect to Concept I is less or equal to the length of paths obtained with Concept II.

In general, the computation of paths with respect to Concept II is less computationally costly than by use of Concept I. However, the choice of the 'right' concept for the study of robustness of a technical application depends highly on the application. The first concept delivers a substationary point which changes as little as possible over an entire interval the external parameter is assumed to vary in. On the contrary, by use of the second concept solution paths are obtained that locally, from one computed point to another, vary as little as possible.

Motivated by the fact that in the case of nonconvex objective functions it is not possible to compute all Pareto optimal solutions by the weighted sums method, in this thesis also the occurrence of *dents* in Pareto fronts has been studied. In [13], Das and Dennis give a trigonometric argument why – in the case of two objectives – the weighted sums method cannot be used to compute points in the *nonconvex part*, as they call the subset of the Pareto front that contains no global optima of the weighted sum of the objectives for every weight vector. It is an interesting question to find out if a Pareto front contains nonconvex parts. If one assumes that the Pareto front is connected,

then every nonconvex part contains a region in which the Pareto front 'curves inside the feasible region'. This part of the Pareto front will be called a *dent*. In Section 3.3 a formal definition of a dent is given. It will be shown that at the border of a dent (seen as a subset of the Pareto front) typically the Hessian of the weighted sum of the objectives is singular. These points will be called *dent border points* and the corresponding preimages on the Pareto set will be called *dent border preimages*.

In the case of parametric multiobjective optimization problems naturally the question comes up, how dents evolve under the variation of the external parameter. This question is addressed in Section 4.5. Making use of results from bifurcation theory it is proven that under certain assumptions dent border preimages are turning points of the Kuhn-Tucker equations

$$H_{\mathrm{KT}}^{\alpha^\star}(x,\lambda) = \sum_{i=1}^{k} \alpha_i^\star \nabla_x f_i(x,\lambda) = 0,$$

where $\alpha^\star$ ist the weight vector corresponding to the dent border preimage. Several examples of parametric multiobjective optimization problems in which the Pareto front contains dents will be given at the end of Section 4.5.

The outline of this thesis is as follows: Chapter 2 contains a brief summary of the theoretical background needed within this thesis. Here, also the notation is introduced. Chapter 3 deals with the theoretical background and algorithmic approaches for multiobjective optimization. In this chapter, three different engineering applications are presented that have been considered extensively in the context of this thesis. Chapter 3 also contains the new ideas for the computation of dents in Pareto fronts.

Chapter 4 is the central chapter of this thesis. Here, especially the two new robustness concepts are derived and a technical application is given. Also, the design of a new algorithm that allows to compute Pareto optimal solutions for a linear drive during operation time is presented. Furthermore, in Chapter 4 the study of dents in parametric multiobjective optimization problems is addressed. The thesis concludes with a short summary of the results and a discussion of possible future directions and applications.

Chapter 2

Theoretical Background

In this chapter the theoretical background needed for theory and applications developed in this thesis is summed up. After a short survey on basic definitions and notations this work is based on, short introductions into the mathematical fields *nonlinear optimization*, the classical *calculus of variations* and *parametric optimization* are given. Some of the technical applications considered in this thesis are so-called *self-optimizing systems*. The definition of self-optimization is given in the last section of this chapter.

2.1 Preliminaries and Notation

As differentiable maps play an important role in this work, in the following the basic understanding of differentiability, together with the notation used in this work, is summed up. Everything introduced in this section is elementary and can be found in the books [52, 61] and many, many others.

Definition 2.1 (Differentiability) *A map $f : U \to \mathbb{R}^m$ is called* differentiable *in a point $p \in U \subseteq \mathbb{R}^n$, if there exists a linear map $L : \mathbb{R}^n \to \mathbb{R}^m$ such that the residuum $R(h)$ given through*

$$f(p+h) = f(p) + L(h) + R(h)$$

satisfies the condition

$$\lim_{h \to 0} \frac{R(h)}{\|h\|} = 0$$

in a certain neighborhood of $0 \in \mathbb{R}^n$. The linear map L is called the differential *of f in p and is denoted by $df(p)$.*

The differential $df(p)$ can be represented through a matrix which is called the *Jacobian* (matrix) of f in p, denoted by $f'(p)$. It is given by the matrix of all partial derivatives of f,

$$f'(p) = \frac{\partial f}{\partial x}(p) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(p) & \dots & \frac{\partial f_1}{\partial x_n}(p) \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1}(p) & \dots & \frac{\partial f_m}{\partial x_n}(p) \end{pmatrix},$$

where

$$\frac{\partial f_i}{\partial x_j}(p) = \lim_{h \to 0} \frac{1}{h} \cdot (f_i(x_1, \ldots, x_j + h, \ldots, x_n) - f_i(x_1, \ldots, x_n))$$

for $h \in \mathbb{R} \setminus \{0\}$.

If $m = 1$, i. e. the function f maps from $\mathbb{R}^n \to \mathbb{R}$, then the Jacobian matrix reduces to one row. We denote the the transpose of this row vector as the *gradient*

$$\nabla f = \nabla_x f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}.$$

If f depends on several variables one wants to distinguish, for example $f : \mathbb{R}^{n_a + n_b} \to \mathbb{R}$ with $f(a, b) = y$, then we denote the *gradient with respect to* a as

$$\nabla_a f = \begin{pmatrix} \frac{\partial f}{\partial a_1} \\ \vdots \\ \frac{\partial f}{\partial a_{n_a}} \end{pmatrix}$$

and the *gradient with respect to* b as

$$\nabla_b f = \begin{pmatrix} \frac{\partial f}{\partial b_1} \\ \vdots \\ \frac{\partial f}{\partial b_{n_b}} \end{pmatrix}.$$

The matrix representation of the *differential of second order* (which is defined as a symmetric, bilinear map) of $f : \mathbb{R}^n \to \mathbb{R}$ in a point p is given by the *Hessian matrix* $f''(p)$, which is defined as

$$f''(p) = \frac{\partial^2 f}{\partial x^2}(p) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(p) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(p) \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(p) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}(p) \end{pmatrix}.$$

A map $\gamma = (\gamma_1, \ldots, \gamma_m)^T : I \to \mathbb{R}^m$ of an interval $I \subseteq \mathbb{R}$, i. e. a *parametrized curve in* $\mathbb{R}^m$, is differentiable in a point $T_0 \in I$ if and only if each of its components is differentiable in T_0,

$$\frac{\partial \gamma}{\partial T}(T_0) = \dot{\gamma}(T_0) = (\dot{\gamma_1}(T_0), \ldots, \dot{\gamma_m}(T_0))^T.$$

2.2 Nonlinear Optimization

Optimization plays an important role in various applications in almost any research area. In this section, optimization problems are considered whose objective function is nonlinear and possibly subject to equality constraints. Here, only the mathematical problem definition and some necessary and sufficient conditions for optimality that will be needed in the context of this thesis are summed up. Comprehensive introductions into the field of nonlinear optimization, both theoretically and algorithmically oriented, can be found in numerous textbooks like [67, 57, 54, 95].

2.2.1 Problem Formulation

In general, a constrained *optimization problem* is given by

$$\min_x \{f(x) : x \in S \subseteq \mathbb{R}^n\}, \tag{NLO}$$

where the feasible set S is given as

$$S = \{x \in \mathbb{R}^n : h(x) = 0,\, g(x) \leq 0\}$$

with $h : \mathbb{R}^n \to \mathbb{R}^m$, $m \leq n$ and $g : \mathbb{R}^n \to \mathbb{R}^q$ and an objective function $f : S \to \mathbb{R}$. Thus, solutions $x^\star \in S$ are to be determined that satisfy

$$f(x^\star) \leq f(x)\ \forall x \in S. \tag{2.1}$$

In this case, $x^\star$ is called a *(global) minimizer* of f (in S) and $f(x^\star)$ is called the *(global) minimum*. If there exists a neighborhood U of $x^\star$ such that equation 2.1 is only valid for all $x \in S \cap U$, then $x^\star$ is called a *local minimizer* of f (in S) and $f(x^\star)$ is called the *local minimum*. If the inequality in equation (2.1) is strict, i. e. $f(x^\star) < f(x)\ \forall x \in S$ or $x \in U \cap S$ respectively, then $x^\star$ is called a *strict (global) minimizer* or a *strict local minimizer* respectively. Minima are said to be *isolated* if they are unique within a certain neighborhood.

If no constraints have to be satisfied, which means $S = \mathbb{R}^n$, then the optimization problem (NLO) is called *unconstrained*. An optimization problem is called *nonlinear* if the objective function (and possibly also the constraints) are nonlinear.

Example 2.2 Consider the nonlinear objective function $f : \mathbb{R} \to \mathbb{R}$,

$$f(x) = -\frac{1}{40}x^5 + \frac{7}{8}x^4 - \frac{59}{6}x^3 + 47x^2 - 96x + 65,$$

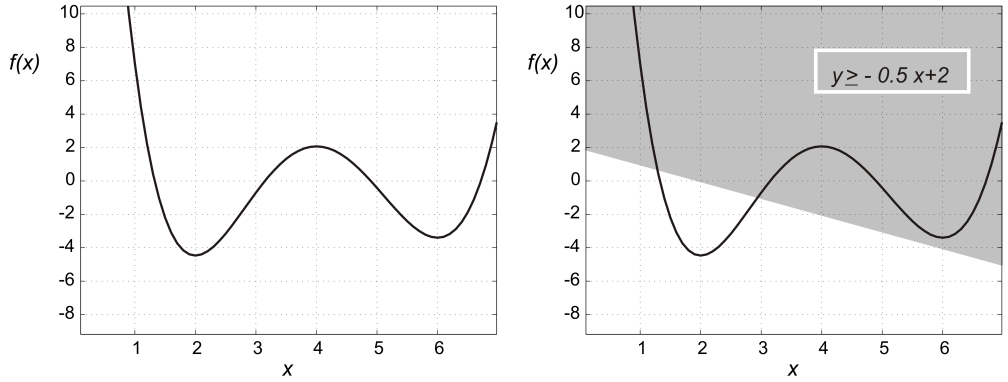

Figure 2.1: Example for an unconstrained and a constrained nonlinear optimization problem

which is wanted to be minimized for $x \in [0, 7]$. The graph of f for $x \in [0, 7]$ is plotted in Figure 2.1. As one can observe, the function $f(x)$ attains its *global minimum* in the point $x_1 = 2$ and a *local minimum* at $x_2 = 6$.

Now, the same objective function with an additional constraint is considered. The optimal solution is desired to lie in the upper semiplane of the straight line G given by $G(x) = -\frac{1}{2}x + 2$. This means that the inequality condition $-f(x) - x + 2 \leq 0$ has to be true. As one can easily see in the right plot of Figure 2.1, in this case $x_2 = 6$ is the global minimizer.

In an analogous way, maximization problems can be defined. As it is possible to convert any maximization problem into a minimization problem, within this work only minimization problems are considered.

2.2.2 Optimality Conditions

Solutions of optimization problems can under certain assumptions be characterized by different necessary and/or sufficient conditions, which are summarized in the following.

Optimality Conditions for Unconstrained Optimization Problems

Consider the *unconstrained optimization problem* , given by

$$\min_x \{f(x) : x \in \mathbb{R}^n\}. \qquad (uNLO)$$

Theorem 2.3 (First-order necessary condition [37]) *Let $X \subseteq \mathbb{R}^n$ open and $f : X \to \mathbb{R}$ a continuously differentiable function. If $x^\star \in X$ is a local*

minimum of f, then

$$\nabla f(x^\star) = 0.$$

$x^\star$ is called a stationary point *of f.*

Proof: A proof of this theorem can for example be found in [37]. □

By use of this first-order necessary condition, the problem of finding local solutions of $(uNLO)$ can be traced back to solving a system of nonlinear equations. The solution set of this system of equations contains not only local minimizers but possibly also local maximizers and saddle points (cf. Definition 2.5).

A sufficient condition for generating local minimizers can be formulated by taking the definiteness of the Hessian matrix of the objective function into account. In general, a matrix $A \in \mathbb{R}^{n\times n}$ is called *positive* or *negative definite*, if $x^T Ax > 0$ or $x^T Ax < 0$, respectively, for all $x \in \mathbb{R}^n$ with $x \neq 0$. If the evaluation of $x^T Ax$ for $x \in \mathbb{R}^n$ leads to both positive and negative values, then the matrix A is called *indefinite*. It is well-known that a matrix A is positive (negative) definite if and only if all its eigenvalues, i. e. values λ with $Ax = \lambda x$ for a vector $x \in \mathbb{R}^n$ are positive (negative). The matrix is indefinite if and only if both eigenvalues > 0 and < 0 exist.

Theorem 2.4 (Second-order sufficient condition [37]) *Let $X \subseteq \mathbb{R}^n$ open and $f : X \to \mathbb{R}$ a twice continuously differentiable function. If*

- *$\nabla f(x^\star) = 0$ and*
- *the Hessian of f, $f''(x^\star)$, is positive definite,*

then $x^\star$ is a strict local minimizer of f (on X).

Proof: For the proof of this theorem the reader is referred to [37]. □

If the Hessian matrix is indefinite, then saddle points occur.

Definition 2.5 (saddle point) *A point $\bar{x} \in \mathbb{R}^n$ is called a* saddle point *of the function $f : \mathbb{R}^n \to \mathbb{R}$ if $\nabla f(\bar{x}) = 0$ and the Hessian $f''(\bar{x})$ is indefinite.*

Example 2.6 For Example 2.2 the necessary condition looks as follows:

$$\nabla f(x) = f'(x) = -\frac{1}{8}x^4 + \frac{7}{2}x^3 - \frac{59}{2}x^2 + 94x - 96 = 0.$$

Polynomial long division with $x_1 = 2$, $x_2 = 6$ and $x_3 = 4$ yields

$$f'(x) = (x-2)(x-6)(x-4)(-\frac{1}{8}x + 2) = 0.$$

Thus, the stationary points are given by $x_1 = 2$, $x_2 = 6$, $x_3 = 4$ and $x_4 = 16$, which is outside the interval regarded here. One can easily check that $f''(2) = 14$, $f''(4) = -6$ and $f''(6) = 10$. Thus, $x_1 = 2$ and $x_2 = 6$ are – as already observed in Figure 2.1 – the two local minimizers. A comparison of the objective values shows that $x_1 = 2$ is the global optimizer.

Including inequality and/or equality conditions into an optimization problem makes the necessary and sufficient conditions for optimality a bit more complicated. A combination of gradient and constraints has to become zero and also some regularity conditions have to be satisfied.

Optimimality Conditions for Equality-constrained Optimization Problems

In this paragraph, the *equality-constrained optimization problem*

$$\begin{aligned} &\min_x f(x) \\ &\text{subject to } h(x) = 0 \end{aligned} \qquad (NLO^{=})$$

with functions $f : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}^m$ is studied. The following theorem gives a necessary condition for optimality.

Theorem 2.7 (First-order necessary condition [67]) *Let $x^\star$ be a local minimum point of f subject to the constraints $h(x) = 0$. Assume further that $x^\star$ is a regular point of these constraints (i. e. the gradient vectors $\nabla h_1(x^\star)$, ..., $\nabla h_m(x^\star)$ are linearly independent). Then there exist Lagrangian multipliers $(\mu_1, \ldots, \mu_m)^T = \mu \in \mathbb{R}^m$ such that*

$$\nabla f(x^\star) - \sum_{i=1}^{m} \mu_i \nabla h_i(x^\star) = 0.$$

Proof: The proof of this theorem can be found in [67], for example. □

A sufficient condition for optimality can be given by taking the second derivative of the *Lagrangian function*, which is defined as

$$L(x) = f(x) - \mu^T h(x),$$

into account, where μ is the vector of Lagrangian multipliers (cf. Theorem 2.7).

Theorem 2.8 (Second-order sufficiency condition [67]) *Suppose there is a point $x^\star \in \mathbb{R}^n$ satisfying $h(x^\star) = 0$, and a $\mu = (\mu_1, \ldots, \mu_m)^T \in \mathbb{R}^m$ such that*

$$\nabla f(x^\star) - \sum_{i=1}^{m} \mu_i \nabla h_i(x^\star) = 0.$$

Suppose also that the matrix $L''(x^\star) = f''(x^\star) - \sum_{i=1}^{m} \mu_i h_i''(x^\star)$ is positive definite on $M = \{y \in \mathbb{R}^n : (\nabla h_1(x^\star) \cdots \nabla h_m(x^\star))^T y = 0\}$, that is, for $0 \neq y \in M$ there holds $y^T L''(x^\star) y > 0$.
Then $x^\star$ is a strict local minimum of f subject to $h(x) = 0$.

Example 2.9 Consider a paraboloid described by the objective function $f : \mathbb{R}^2 \to \mathbb{R}$,

$$f(x_1, x_2) = x_1^2 + x_2^2.$$

Assume that this function has to be minimized under the constraint that the solution lies on the intersection of the function values with a plane in $\mathbb{R}^3$ which is defined by

$$\{(x_1, x_2, x_3) | x_3 + x_1 - 2 = 0\}.$$

Figure 2.2 shows the paraboloid and the plane. The optimization problem can be posed as

$$\min_x f(x_1, x_2)$$
$$\text{such that } h(x_1, x_2) = x_1^2 + x_2^2 + x_1 - 2 = 0.$$

Solving the equality condition for x_1 yields $x_1^{\{1,2\}} = \pm\frac{1}{2}\sqrt{9 - 4x_2^2} - \frac{1}{2}$. Solving the necessary condition

$$\begin{aligned} \nabla f(x_1, x_2) - \mu_1 \nabla h(x_1, x_2) &= 0 \\ \Leftrightarrow \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} - \mu_1 \begin{pmatrix} 2x_1 + 1 \\ 2x_2 \end{pmatrix} &= 0 \end{aligned}$$

(cf. Theorem 2.7) leads to

$$x_2 = 0 \text{ and } x_1 = \frac{\mu_1}{2(1 - \mu_1)}.$$

As an assumption for the sufficient condition (cf. Theorem 2.8) both conditions have to be true. Solving

$$\frac{\mu_1}{2(1 - \mu_1)} = \pm\frac{1}{2}\sqrt{9 - 4 \cdot 0} - \frac{1}{2}$$

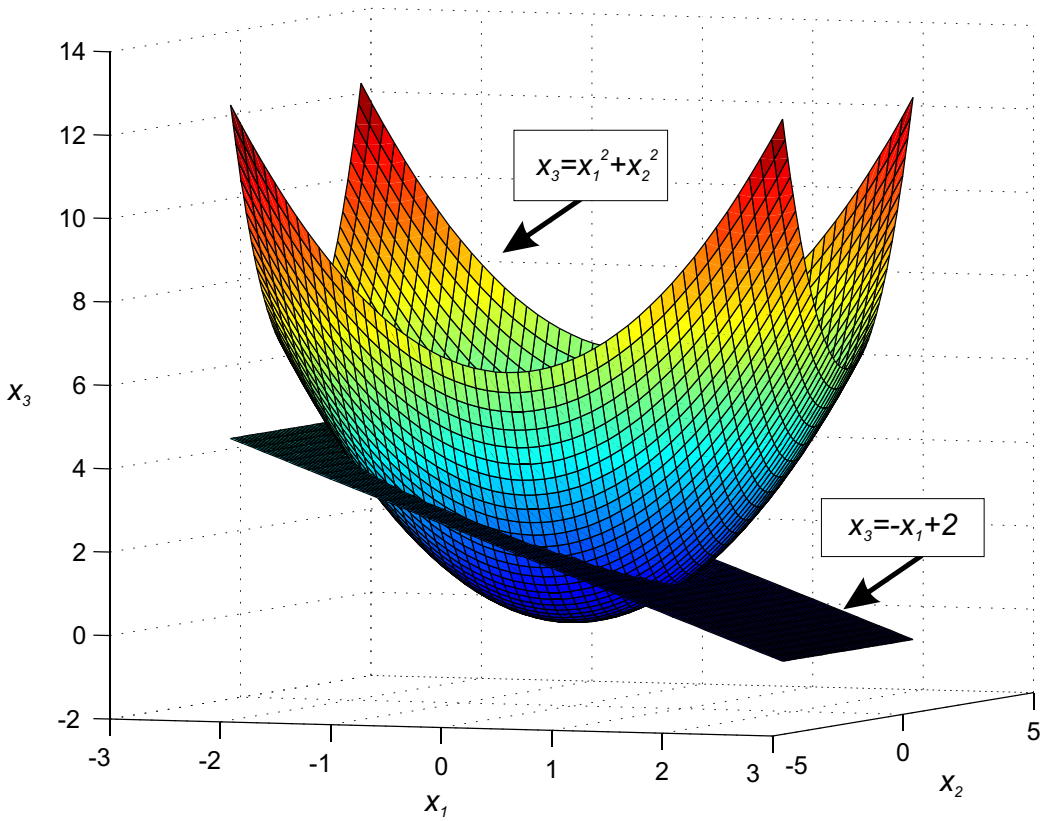

Figure 2.2: In Example 2.9 a paraboloid is desired to be minimized on a plane

leads to two possible solution points p_1 and p_2 given by

$$\begin{aligned} p_1 &= (x_1^{\{1\}}, x_2^{\{1\}}, \mu_1^{\{1\}}) = (1, 0, \frac{2}{3}) \\ p_2 &= (x_1^{\{2\}}, x_2^{\{2\}}, \mu_1^{\{2\}}) = (-2, 0, \frac{4}{3}). \end{aligned}$$

The Hessian of the Lagrangian function is given by

$$\begin{aligned} L''(x_1, x_2) &= f''(x_1, x_2) - \mu_1 h''(x_1, x_2) \\ &= \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} - \mu_1 \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \\ &= \begin{pmatrix} 2(1-\mu_1) & 0 \\ 0 & 2(1-\mu_1) \end{pmatrix} \end{aligned}$$

Obviously, this matrix is positive definite for all $\mu_1 < 1$ (independently of x_1 and x_2) and thus also in the point p_1. By use of Theorem 2.8 it follows that the point $(x_1^{\{2\}}, x_2^{\{2\}}) = (1, 0)$ is a strict local minimizer.

2.3 Calculus of Variations

In contrast to nonlinear optimization where the objective to be optimized is given as a function $f : \mathbb{R}^n \to \mathbb{R}$, the calculus of variations is concerned with the optimization of a functional. The task is to find a continuously differentiable curve $\gamma : [\lambda_{\text{start}}, \lambda_{\text{end}}] \to \mathbb{R}^n$, $\gamma(\lambda) = (\gamma_1(\lambda), \ldots, \gamma_n(\lambda))^T$ that minimizes a variational integral

$$\mathcal{J}(\gamma) = \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} C(\gamma(\lambda), \gamma'(\lambda), \lambda)\, \mathrm{d}\lambda \tag{VP}$$

for a given sufficiently often differentiable function C that may depend on the curve $\gamma(\lambda)$, its derivative $\gamma'(\lambda) = \frac{\mathrm{d}}{\mathrm{d}\lambda}\gamma(\lambda)$, and on the parameter λ itself. Additionally, boundary conditions may be included.

The basics summed up in the following can be found in numerous textbooks and papers, for example [9], [41], [34] and [79]. In this thesis only functionals depending on a one-dimensional real parameter λ are considered. Also, only integrals that depend on $\gamma(\lambda)$ and $\gamma'(\lambda)$ are studied within this work – a generalization to the case that C depends on derivatives of higher order of γ with respect to λ can for example be found in [9].

The main motivation for the development of the calculus of variations was given by Johann Bernoulli in 1696 by posing the 'Brachistochrone problem': Given two points P_0 and P_1 in a vertical plane, a curve for a moving point M is searched that begins in the point P_0 with an initial velocity v_0 and ends up in the point P_1 in virtue of its own gravity in the shortest possible time (cf. [34]). Even much earlier, in the middle of the eighth century before Christ, the so-called 'Dido's Problem' occured: Princess Dido had to flee to the African coast and asked the local chief for a piece of land as large as she could envelop with a cowhide. The chief agreed and Dido cleverly cut the cowhide into little stripes and in this way got a much larger piece of land as the chief had in mind. Following the legend of the foundation of Carthage, this coastal strip (the so-called 'Byrsa') was the nucleus of Carthage. Bernoulli called the mathematical problem behind it an 'isoperimetric problem' (cf. [34]). The theoretical framework that allows to compute solutions for both kinds of problems (and also several others) mainly goes back to Leonhard Euler and Joseph Louis Lagrange.

Most of the classical problems like the two examples mentioned in the last paragraph have in common that curves connecting two fixed end points are to be computed. In this case, the problem can be formulated as follows:

$$\min_{\gamma} \mathcal{J}(\gamma) = \min_{\gamma} \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} C(\gamma(\lambda), \gamma'(\lambda), \lambda)\, \mathrm{d}\lambda \qquad (\mathrm{VC}_{2p})$$
$$\gamma(\lambda_{\text{start}}) = A,\ \gamma(\lambda_{\text{end}}) = B$$

with two specified vectors $A, B \in \mathbb{R}^n$.

A minimizing curve $\gamma^\star$ for (VC_{2p}) with $\gamma^\star(\lambda_{\text{start}}) = A$ and $\gamma^\star(\lambda_{\text{end}}) = B$ has to satisfy the property that

$$\mathcal{J}(\gamma^\star) \leq \mathcal{J}(\gamma)$$

for all continuous curves $\gamma : [\lambda_{\text{start}}, \lambda_{\text{end}}] \to \mathbb{R}^n$ with $\gamma(\lambda_{\text{start}}) = A$ and $\gamma(\lambda_{\text{end}}) = B$.

Following [79] it is assumed that these curves are at least twice continuously differentiable for the following derivation of a necessary condition for optimality of (VC_{2p}). However, the same results can also be proven by the weaker assumption that $\gamma(\lambda)$ is only once continuously differentiable by the use of the 'Lemma of Du Bois-Reymond' (cf. e. g. [77]).

Let $\eta(\lambda) = (\eta_1(\lambda), \ldots, \eta_n(\lambda))^T$ be a twice continuously differentiable curve with $\eta(\lambda_{\text{start}}) = \eta(\lambda_{\text{end}}) = 0$ and assume that $\gamma(\lambda)$ is a minimizing curve of (VC_{2p}). $\delta\gamma = \eta$ is called a *variation* of γ. Let $\varepsilon \in \mathbb{R}$ and consider the function $v : \mathbb{R} \to \mathbb{R}$ defined as

$$v(\varepsilon) \;=\; \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} C\left(\gamma(\lambda) + \varepsilon\eta(\lambda), \gamma'(\lambda) + \varepsilon\eta'(\lambda), \lambda\right)\, \mathrm{d}\lambda \qquad (2.2)$$

Note, that in this case the variations η_i of the components γ_i are assumed to be independent of each other. Because of the minimality of $\gamma(\lambda)$ it follows that

$$v(\varepsilon) \geq v(0)\ \forall \varepsilon \in \mathbb{R}.$$

Thus, v attains a minimum in the point $\varepsilon = 0$ and

$$v'(0) = 0. \qquad (2.3)$$

$v'(0)$ is called the *first variation* of $\mathcal{J}$ and is denoted by $\delta\mathcal{J}$.

In the following, for simplicity, whenever C occurs the evaluation at the point $(\gamma_1(\lambda), \ldots, \gamma_n(\lambda), \gamma_1'(\lambda), \ldots, \gamma_n'(\lambda), \lambda)$ is meant. Additionally, in the following computations the partial derivative with respect to the j'th variable is denoted by ∂_j.

$$\begin{aligned}\frac{\mathrm{d}\,v}{\mathrm{d}\,\varepsilon}(0) &= \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} \left(\sum_{j=1}^{n} \partial_j C \cdot \frac{\mathrm{d}\,(\gamma_j + \varepsilon\eta_j)}{\mathrm{d}\,\varepsilon} + \right. \\ &\quad + \sum_{k=n+1}^{2n} \partial_k C \cdot \frac{\mathrm{d}\,(\gamma'_{k-n} + \varepsilon\eta'_{k-n})}{\mathrm{d}\,\varepsilon} + \\ &\quad \left. + \partial_{2n+1} C \cdot \frac{\mathrm{d}\,\lambda}{\mathrm{d}\,\varepsilon} \right) \mathrm{d}\lambda \Big|_{\varepsilon=0} \\ &= \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} \frac{\partial}{\partial\gamma} C \cdot \eta + \frac{\partial}{\partial\gamma'} C \cdot \eta' \,\mathrm{d}\lambda = 0. \end{aligned} \tag{2.4}$$

Integration by parts of the second summand yields

$$\int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} \frac{\partial}{\partial\gamma'} C \cdot \eta' \,\mathrm{d}\lambda = \left[\frac{\partial}{\partial\gamma'} C \cdot \eta \right]_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} - \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} \frac{\mathrm{d}}{\mathrm{d}\lambda} \left(\frac{\partial}{\partial\gamma'} C \right) \cdot \eta \,\mathrm{d}\lambda. \tag{2.5}$$

Insertion into equation (2.4) and the condition $\eta(\lambda_{\text{start}}) = \eta(\lambda_{\text{end}}) = 0$ now leads to

$$\int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} \left(\frac{\partial}{\partial\gamma} C - \frac{\mathrm{d}}{\mathrm{d}\lambda} \left(\frac{\partial}{\partial\gamma'} C \right) \right) \cdot \eta \,\mathrm{d}\lambda = 0.$$

As these equations have to be true for every arbitrary set of functions η_i, $i = 1, \ldots, n$, the integral becomes zero only if

$$\frac{\partial}{\partial\gamma} C - \frac{\mathrm{d}}{\mathrm{d}\lambda} \left(\frac{\partial}{\partial\gamma'} C \right) = 0.$$

The formal proof of this so-called 'Fundamental lemma of the calculus of variations' can for example be found in [41].

In short (and by multiplication with -1), it has been shown that

$$\frac{\mathrm{d}}{\mathrm{d}\lambda} \left(\frac{\partial C}{\partial\gamma'} \right) - \frac{\partial C}{\partial\gamma} = 0 \tag{2.6}$$

is a necessary condition for optimality of (VC_{2p}). These equations are called the *Euler-Lagrange equations*.

A simple example for a variational problem is given in the following:

Example 2.10 Which is the shortest curve connecting the two points $(1, 1)$ and $(2, 2)$ in the x_1-x_2-plane in $\mathbb{R}^2$ (cf. Figure 2.3)?

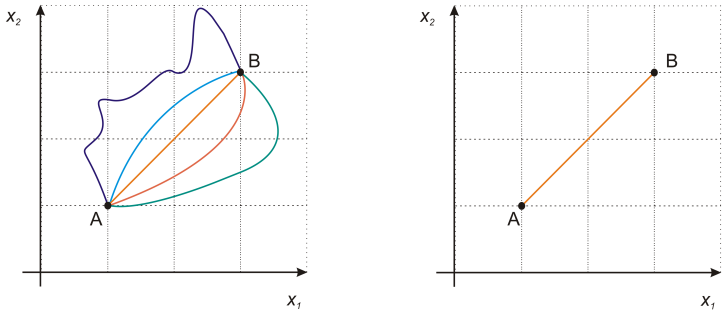

Figure 2.3: Some possible curves connecting the points $(1, 1)$ and $(2, 2)$ in $\mathbb{R}^2$ (on the left) and the connecting curve of minimal length (on the right)

One way to attack this problem is to consider the variational integral

$$\min_x \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} \frac{1}{2} \|x'(\lambda)\|^2 \, \mathrm{d}\lambda$$

with $x(\lambda) = \begin{pmatrix} x_1(\lambda) \\ x_2(\lambda) \end{pmatrix}$ and the boundary conditions

$$\begin{aligned} \lambda_{\text{start}} &= 0, x(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ \lambda_{\text{end}} &= 1, x(1) = \begin{pmatrix} 2 \\ 2 \end{pmatrix}. \end{aligned}$$

The Euler-Lagrange equations for this example are given as

$$\frac{\mathrm{d}}{\mathrm{d}\lambda} \begin{pmatrix} \frac{\partial F}{\partial x'} \end{pmatrix} - \frac{\partial F}{\partial x} = \frac{\mathrm{d}}{\mathrm{d}\lambda} \begin{pmatrix} x_1'(t) \\ x_2'(t) \end{pmatrix} - 0 = \begin{pmatrix} x_1''(\lambda) \\ x_2''(\lambda) \end{pmatrix} \stackrel{!}{=} 0.$$

A solution for this ordinary differential equation obviously is given by

$$\begin{pmatrix} x_1(\lambda) \\ x_2(\lambda) \end{pmatrix} = \begin{pmatrix} c_1\lambda + d_1 \\ c_2\lambda + d_2 \end{pmatrix}$$

with constants $c_{1,2}, d_{1,2} \in \mathbb{R}$.
Application of the boundary conditions yields $c_1 = c_2 = 1$ and $d_1 = d_2 = 1$. Thus

$$\begin{pmatrix} x_1(\lambda) \\ x_2(\lambda) \end{pmatrix} = \begin{pmatrix} \lambda + 1 \\ \lambda + 1 \end{pmatrix},$$

which – as expected – is a straight line connecting the two points.

In practice, often constraints have to be satisfied by the solution of a variational problem. In this work, the case of $m < n$ equality constraints given by $\varphi : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ with $\varphi(\gamma(\lambda), \lambda) = (\varphi_1(\gamma(\lambda), \lambda), \ldots, \varphi_m(\gamma(\lambda), \lambda))^T = 0$ occurs (so-called *holonomic constraints*, cf. [41]). In this case, the variational problem can be formulated as follows:

$$\begin{aligned} \min_{\gamma} \mathcal{J}(\gamma) = \min_{\gamma} \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} C(\gamma(\lambda), \gamma'(\lambda), \lambda)\, \mathrm{d}\lambda \\ \text{subject to } \varphi(\gamma(\lambda), \lambda) = 0 \\ \gamma(\lambda_{\text{start}}) = A,\ \gamma(\lambda_{\text{end}}) = B \end{aligned} \tag{VC$_{2p}$-c}$$

with two specified vectors $A, B \in \mathbb{R}^n$. A necessary condition for optimality is given in the following theorem, called the *multiplier rule*, which goes back to Euler and Lagrange:

Theorem 2.11 (Multiplier rule [41]) *If a twice continuously differentiable curve γ is a solution of VC$_{2p}$-c, then there exist functions $\mu_1, \ldots, \mu_m$, each mapping from $[\lambda_{\text{start}}, \lambda_{\text{end}}]$ to $\mathbb{R}$, such that γ is a solution of the unconstrained variational problem*

$$\begin{aligned} \min_{\gamma,\mu} \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} L(\gamma(\lambda), \gamma'(\lambda), \mu(\lambda), \lambda)\, \mathrm{d}\lambda \\ \gamma(\lambda_{\text{start}}) = A,\ \gamma(\lambda_{\text{end}}) = B \end{aligned} \tag{2.7}$$

with the Lagrangian L *defined as*

$$L(\gamma(\lambda), \gamma'(\lambda), \mu(\lambda), \lambda) = C(\gamma(\lambda), \gamma'(\lambda), \lambda) + \sum_{i=1}^{m} \mu_i(\lambda) \cdot \varphi_i(\gamma(\lambda), \lambda).$$

By this theorem, using the same derivation as before for the Lagrangian L w. r. t. γ and μ the Euler-Lagrange equations for fixed boundaries read as

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\left(\frac{\partial C}{\partial \gamma'}\right) - \frac{\partial C}{\partial \gamma} - \left(\mu(\lambda)^T \cdot \frac{\partial}{\partial \gamma}\varphi\left(\gamma(\lambda), \lambda\right)\right)^T = 0 \tag{2.8}$$

$$\varphi(\gamma(\lambda), \lambda) = 0, \tag{2.9}$$

where equation (2.9) stems from taking variations with respect to $\mu(\lambda) = (\mu_1, \ldots, \mu_m)^T(\lambda)$.

So far, the curves γ have been considered only between two fixed points A and B. If these boundary values of the curve γ are not fixed but rather constrained by

$$\varphi(\gamma(\lambda_{\text{start}}), \lambda_{\text{start}}) = \varphi(\gamma(\lambda_{\text{end}}), \lambda_{\text{end}}) = 0,$$

as it will be the case in the robustness formulation presented in Chapter 4, the Euler-Lagrange equations have to be enlarged by the so-called *transversality conditions*. They are derived in the following way:

Since $\varphi(\gamma(\lambda), \lambda) = 0$ has to be satisfied at the boundaries for all possible varied curves, the variations η_i for each component of γ are not independent any more. To derive conditions for admissible variations, let $\varepsilon \in \mathbb{R}$ and let the function $\bar{v} : \mathbb{R} \to \mathbb{R}^m$ be defined as

$$\bar{v}(\varepsilon) := \varphi(\gamma(\lambda) + \varepsilon\eta(\lambda), \lambda),$$

where for all ε only such variations are considered for which the constraints are satisfied at the boundaries, i. e. $\bar{v}(\varepsilon) = 0$ for all ε with $\lambda = \lambda_{\text{start}}$ and $\lambda = \lambda_{\text{end}}$. Therefore, also the derivative w. r. t. ε has to be zero what for $\varepsilon = 0$ results in

$$0 = \bar{v}'(0) = \frac{d}{d\varepsilon}\varphi(\gamma(\lambda) + \varepsilon\eta(\lambda), \lambda)\Big|_{\varepsilon=0} = \frac{\partial}{\partial\gamma}\varphi(\gamma(\lambda), \lambda) \cdot \eta(\lambda)$$

for $\lambda = \lambda_{\text{start}}$ and $\lambda = \lambda_{\text{end}}$. This directly imposes conditions for the variations $\eta(\lambda)$ at the boundaries as thus a variation $\eta(\lambda)$ is feasible if and only if

$$\eta(\lambda) \in \ker\left(\frac{\partial}{\partial\gamma}\varphi(\gamma(\lambda), \lambda)\right) \quad \text{for} \quad \lambda = \lambda_{\text{start}} \quad \text{and} \quad \lambda = \lambda_{\text{end}}. \tag{2.10}$$

Furthermore, from integration by parts (cp. Equation (2.5)), the condition

$$\left[\frac{\partial}{\partial\gamma'}C \cdot \eta\right]_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} = 0$$

has to be satisfied, which means that

$$\frac{\partial}{\partial\gamma'}C \perp \eta \quad \text{for} \quad \lambda = \lambda_{\text{start}} \quad \text{and} \quad \lambda = \lambda_{\text{end}}$$

for all feasible variations $\eta(\lambda)$. Together with Equation (2.10) this results in

$$\frac{\partial}{\partial\gamma'}C \perp \ker\left(\frac{\partial}{\partial\gamma}\varphi\right) \quad \text{for} \quad \lambda = \lambda_{\text{start}} \quad \text{and} \quad \lambda = \lambda_{\text{end}}. \tag{2.11}$$

Equation (2.11) says that at the boundaries the partial derivative of C w. r. t. γ' has to be transversal to the tangent space of the hyperplane which is implicitly defined by $\varphi(\gamma(\lambda_{\text{start}}), \lambda_{\text{start}}) = 0$ and $\varphi(\gamma(\lambda_{\text{end}}), \lambda_{\text{end}}) = 0$, respectively. Together, Equations (2.8)-(2.9) and (2.11) form a boundary value problem that has to be solved numerically.

2.4 Parametric Optimization

Optimization problems whose solutions may change under the influence of some external parameters are called *parametric optimization problems*. Parameter-dependencies occur in many applications. For example, optimization problems may depend on temperature, wind velocity or other physical values. Also time-dependent optimization problems are very important, especially in engineering.
Furthermore, the development of methods for parametric optimization problems can be very helpful for finding solutions of complicated nonlinear optimization problems: In this case one tries to reduce the solution of the original optimization problem to another optimization problem whose solution is already known or easy to compute. These two problems then are combined with the help of a parameter $\lambda \in \mathbb{R}$. Denote the original objective function by $f(x)$ and the objective function of the already solved problem by $g(x)$. Then the parameter-dependent problem $\min_x (1 - \lambda) \cdot g(x) + \lambda \cdot f(x)$ has to be solved, varying λ from 0 to 1. Thus one starts with the already known solution for g and ends up with the desired minimum of f. Approaches of this kind are commonly known as *homotopy methods*.

In the literature, parametric optimization is also adressed as *parametric programming*. There are various publications on this topic like for example [44, 5, 46, 60, 2] and [102].

In this work only one-parametric, unconstrained optimization problems are considered. This means that the objective function depends on an additional parameter $\lambda \in \mathbb{R}$. Additionally, it is assumed that the objective function is at least twice continuously differentiable. In the following two sections, some theoretical basics are summed up and a short introduction into numerical path following methods that allow to compute solutions of parametric optimization problems is given.

2.4.1 Theoretical Basics for Unconstrained One-parametric Optimization Problems

Consider a twice continuously differentiable function $f : \mathbb{R}^n \times \Lambda \to \mathbb{R}$ with $\Lambda = [\lambda_{\text{start}}, \lambda_{\text{end}}] \subset \mathbb{R}$. In an unconstrained *parametric optimization problem* a solution curve $x(\lambda)$ of

$$\min_x \{f(x, \lambda) : x \in \mathbb{R}^n\} \qquad (uPOP)$$

is to be found for $\lambda \in \Lambda$. It is assumed that the initial solution $x(\lambda_{\text{start}})$ is known or can be computed easily.

As the solution curve $x(\lambda)$ has to be optimal in each point $\lambda \in \Lambda$, a necessary condition for the optimality of $x(\lambda)$ is given by

$$H(x, \lambda) = \frac{\partial}{\partial x} f(x, \lambda) = 0 \; \forall \lambda \in \Lambda$$

(cf. Theorem 2.3). Here, H is a nonlinear system of equations with n equations and $n + 1$ (or, in the multi-parametric case $n + p$) variables. The existence of solutions to this kind of problems within a neighborhood of an already known solution is guaranteed by the implicit function theorem given as follows:

Theorem 2.12 (Implicit Function Theorem [52, 27]) *Let $H : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ be a k-times continuously differentiable mapping. Suppose that the point $(\bar{x}, \bar{y}) \in \mathbb{R}^n \times \mathbb{R}^p$ satisfies $H(\bar{x}, \bar{y}) = 0$ and that the Jacobian with respect to x, $\frac{\partial F}{\partial x}(\bar{x}, \bar{y})$, is regular.*

Then there exists a neighborhood of $\bar{y}$, $U(\bar{y}) \subseteq \mathbb{R}^p$ and a unique, k-times continuously differentiable function $c : U(\bar{y}) \to \mathbb{R}^n$ with

$$H(c(y), y) = 0 \text{ for all } y \in U(\bar{y})$$

and $c(\bar{y}) = \bar{x}$. In this case, the function c is said to be implicitly defined *by the equation $H(c(y), y) = 0$.*

Proof: A proof of this theorem is for example given in [52]. □

Applying the implicit function theorem to the necessary condition that $\frac{\partial}{\partial x} f(x, \lambda) = 0 \; \forall \lambda \in \Lambda$, the following theorem results directly:

Theorem 2.13 (Sensitivity analysis for an unconstrained parametric optimization problem [27]) *Let $f(x, \lambda)$, $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ be a twice continuously differentiable function in x. Let $\lambda^\star \in \mathbb{R}^p$. Assume that $x^\star$ is a strict local*

minimizer of the unconstrained optimization problem $\min_x f(x, \lambda^\star)$*, i. e.* $x^\star$ *satisfies the second order sufficient condition (see Theorem 2.4). Let* $\frac{\partial f}{\partial x}(x, \lambda)$ *be once continuously differentiable w. r. t.* λ *within a neighborhood of* $(x^\star, \lambda^\star)$*.*

Then, for λ *near* $\lambda^\star$*, there exists a unique, once continuously differentiable curve* $x(\lambda)$ *with* $x(\lambda^\star) = x^\star$ *which satisfies the second-order sufficient condition for* $f(x, \lambda)$*. Hence,* $x(\lambda)$ *is an isolated, locally unique local minimum of* $f(x, \lambda)$*.*

Remark 2.14 In [27] also a more general formulation of Theorem 2.13 is given that guarantees the local existence of solutions to parametric optimization problems with equality and inequality constraints.

Example 2.15 Consider the objective function $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$

$$f(x, \lambda) = \lambda(x - 2\lambda)^2 + (1 - \lambda)(-(x - 2\lambda)^2) + 4\lambda$$

Let the parameter λ be varied from 0 to 1. By $f(x, 0)$ a parabola that attains its maximum at $x = 0$ (apex $(0, 0)$) is described. Due to the variation of λ towards $\lambda = 1$ the parabola changes into another parabola that attains its minimum at $x = 2$ (apex $(2, 4)$). To compute a solution curve $x(\lambda)$ for $\lambda \in [0, 1]$ the necessary condition

$$\begin{aligned} \frac{\partial}{\partial x} f(x, \lambda) &= 0 && (2.12) \\ \Leftrightarrow (x - 2\lambda)(4\lambda - 2) &= 0 && (2.13) \end{aligned}$$

is to be studied. One can observe that the change from a maximum to a minimum occurs at $\lambda = 0.5$. In this point the Hessian of f given by

$$\frac{\partial^2 f}{\partial x^2}(x, \lambda) = 4\lambda - 2$$

becomes singular (here $= 0$). Thus, for $\lambda = 0.5$ the sufficient condition of second order is not satisfied and Theorem 2.13 does not guarantee the existence of a solution. As one can easily see, zeros of (2.12) are given by $x(\lambda) = 2\lambda$. For $\lambda \in]0.5, 1]$ the Hessian is positive definite (here > 0) and thus the unique existence of a minimal solution of (2.12) is guaranteed by Theorem 2.13. For $\lambda \in [0, 0.5[$ the Hessian is negative definite (here simply < 0) which means that f attains a maximum. Both theorems 2.4 and 2.13 can also be formulated for maximization problems in the same way. It would result that $x(\lambda) = 2\lambda$ is a unique maximal solution in this interval. In Figure 2.4 the graph of f is plotted for different values of λ. In the same plot, the path of λ-dependent solutions of Equation (2.12) is visualized (black line).

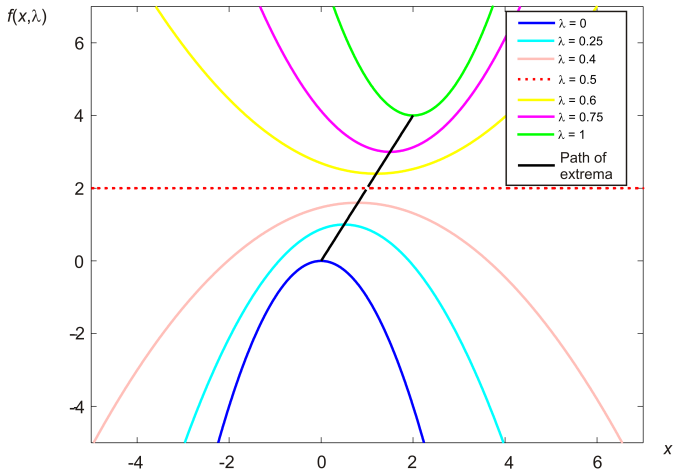

Figure 2.4: The graph of the function f defined in Example 2.15 for different values of the parameter λ

Remark 2.16 The main theoretical challenge occurs when singularities occur and the implicit function theorem cannot be applied. In [44] necessary conditions for constrained parametric optimization problems are studied and, for the one-parametric case, the generic singularities are described. Numerical path following methods (see next section) are used to compute curves of local minimizers.

2.4.2 Numerical Path Following

The computation of solutions of parametric optimization problems can be reduced to the solution of parameter-dependent systems of equations as described in the last section. This kind of equations can be solved by *numerical path following methods* (also called *continuation methods*) as developed in [2, 22] and [75] for example. Following the lines of [2], this section contains a short outline how numerical path following methods work in principle.

Consider a smooth function $H : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$ whose set of zeros is to be found. This means, a system of equations of the form

$$H(y, \lambda) = 0 \tag{2.14}$$

has to be solved, where $y \in D \subseteq \mathbb{R}^N$, D open. The purpose of a numerical path following method is to generate a series of points (y_i, λ_i) $i = 0, 1, 2, \ldots$ that satisfies $H(y, \lambda) = 0$.

Let $u_0 = (y_0, \lambda_0)$ be a solution of (2.14), i. e. $H(u_0) = 0$. Suppose that the Jacobian $H'(u_0)$ has full rank. In this case the solution set $H^{-1}(0)$ can locally be parametrized by one parameter s in the neighbourhood of u_0. Thus, one obtains a solution curve $c(s)$ with

$$c(0) = u_0 \quad \text{and} \quad H(c(s)) = 0.$$

Differentiating this last equality we obtain

$$H'(c(s))c'(s) = 0.$$

Therefore $c'(s)$ spans the one-dimensional kernel of $H'(c(s))$. By choosing s to be the arclength it follows that $\|c'(s)\| = 1$.
Moreover one can show that $c(s)$ is the solution of the initial value problem

$$\begin{aligned} \dot{u} &= t(H'(u)) \\ u(0) &= u_0. \end{aligned} \tag{2.15}$$

Here $t(A)$ denotes for a matrix $A \in \mathbb{R}^{N\times(N+1)}$ with $rank(A) = N$ the unique vector t satisfying the following conditions:

$$At = 0, \quad \|t\| = 1 \quad \text{and} \quad \det \begin{pmatrix} A \\ t^T \end{pmatrix} > 0.$$

The last condition allows us to fix the orientation of the tangent vector.

A numerical path following method generates a series of points in the following way: Let u_0 be an initial point satisfying $H(u_0) = 0$. Then u_{i+1} is generated inductively in two steps, a *predictor* and a *corrector step* (cf. also Figure 2.5):

Predictor Step: Solve (2.15) numerically, e.g. by the explicit Euler method:

$$v_{i+1} = u_i + h \cdot t(H'(u_i)),$$

where $h > 0$ is a certain steplength.

Corrector Step: Here one uses the fact that $c(s)$ solves $H(c(s)) = 0$. Define w_{i+1} to be that point on the curve c which is nearest to v_{i+1}, i.e. one has to solve

$$\min\{\|v_{i+1} - w\| : H(w) = 0\}.$$

The solution of this problem – this is typically determined by Newton's method – defines the new point u_{i+1}.

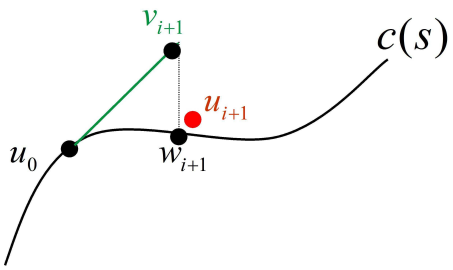

Figure 2.5: Illustration of possible predictor and corrector steps in numerical path following methods

Remark 2.17 The assumption that the Jacobian has full rank induces that possible branch points of solution curves cannot be computed. For the study of problems with singular Jacobians and also an analysis of the step size control for different possible predictors the reader is referred to [2].

Remark 2.18 There are several software packages available which contain codes for numerical path following. A very popular one is AUTO2000[1] (see [23]). In fact, this is also the software we used for part of the computations within the numerical examples presented in Chapter 4, Section 4.3.

2.5 Self-Optimization

For the development of modern, innovative mechatronic systems, the concept of *self-optimization* has been introduced[2] over the last few years. This concept is formulated in [1] as follows:

"Self-optimization describes the ability of a technical system to endogenously adapt its objective regarding changing influences and thus an autonomous adaption of the system's behavior in accordance with the objectives. The behavior adaption may be implemented by changing the parameters or the structure of the system. Thus, self-optimization goes considerably beyond the familiar rule-based and adaptive control strategies; self-optimization facilitates systems with inherent 'intelligence' that are able to take action and react autonomously and flexibly to changing operating conditions."

Self-optimization can be described as a process that consists of the fol-

[1] AUTO2000 is a software package developed by E. Doedel et al. that allows the continuation and bifurcation analysis of ordinary differential equations and parameter-dependent systems of equations.

[2] by the project CRC 614, see www.sfb614.de

lowing three steps (cf. also [1]):

Step 1: Analysis of the current situation

Step 2: Determination of the system's objectives

Step 3: Adaptation of the system's behavior

More precisely, in the first step the state of the system and observations of the environment are analyzed. It is evaluated, whether the current objectives of the system are appropriate for the current situation. If this is not the case, the new objectives are determined in the second step. This can be performed through the *selection* of an alternative from a given finite set of possible objectives, through an *adaptation* of the current objectives by changing the priorities, or even through the *generation* of new objectives independently of the existing ones. The resulting *objective adaptation* leads directly to the third step, in which the *behavior adaptation* is performed such that the new objectives are optimized.

For further information about self-optimizing systems, the reader is referred to [33, 1, 19]. In Chapters 3 and 4 several applications will be described for which the concept of self-optimization has been realized.

Chapter 3

Multiobjective Optimization

In industry and business a variety of applications occur, where several objective functions are desired to be optimized at the same time. For instance, in manufactoring *cost* has to be minimized, but at the same time also *quality* is wanted to be maximal. This example already illustrates that the several objectives typically contradict each other and therefore do not have identical optima. Thus, a set of optimal compromises has to be determined. This set is called the *Pareto optimal set* or just *Pareto set*, named after the Italian engineer, sociologist and economist Vilfredo Pareto (1848 - 1923). In his 'Manual of Political Economy' [76], Pareto developed a theory which includes the study of tastes of men and the value of goods for individuals. He coined the term of *ophelimity*, which is the pleasure that certain quantities of a thing afford an individual. In this way, Pareto optimality means that members of a collectivity enjoy maximal ophelimity in a certain position. This especially means that in a Pareto optimum no individuum can increase its ophelimity without decreasing the ophelimity of another individuum.

The mathematical field of *multiobjective optimization* deals – besides the theoretical study of such problems – with the computation of Pareto optimal solutions.

In this chapter, first of all a formal definition of Pareto optimality and a necessary condition are given. After that, geometrical properties of Pareto sets are reflected and a summary of the most important methods for solving multiobjective optimization problems with continuous variables and objective functions is given. Among these are set-oriented methods which have been applied to several multiobjective optimization problems in engineering applications also described in this chapter.

Having generated the entire Pareto set for a given multiobjective optimization problem, often a decision which solution to choose for the underlying technical system has to be made. This leads to the topic of *Decision Making* on which will be reported in direct connection to the applications considered in the last section of this chapter.

3.1 Problem Formulation and Pareto Optimality

A (constrained) *multiobjective optimization problem* (MOP) is given by

$$\min_x \{F(x) : x \in S \subseteq \mathbb{R}^n\}, \tag{MOP}$$

where F is defined as the vector of the objective functions $f_1, \ldots, f_k$, $k \geq 2$, which each map from $\mathbb{R}^n$ to $\mathbb{R}$, i. e.

$$F : \mathbb{R}^n \to \mathbb{R}^k,\ F(x) = (f_1(x), \ldots, f_k(x)).$$

The feasible set S is given as

$$S = \{x \in \mathbb{R}^n : h(x) = 0,\ g(x) \leq 0\}$$

with equality constraints $h : \mathbb{R}^n \to \mathbb{R}^m$, $m \leq n$ and inequality constraints $g : \mathbb{R}^n \to \mathbb{R}^q$. The MOP is called *unconstrained MOP*, if $S = \mathbb{R}^n$. In all the following considerations it is assumed that $F = (f_1, \ldots, f_k)$ consists of at least continuous objective functions.

It has to be explained what is meant by 'min' in the problem (MOP), as a vector-valued function has to be minimized. The following definition which introduces an appropriate partial order on $\mathbb{R}^k$ allows comparisons of vectors (cf. [20]).

Definition 3.1 *Let u, v be two vectors in $\mathbb{R}^k$. Then the vector u is* less than *v if*

$$u_i < v_i \quad \textit{for all } i \in \{1, \ldots, k\},$$

denoted by $u <_p v$. In an analogous way, the relation $\leq_p$ is defined. The vector u is said to dominate *the vector v if*

$$u \leq_p v \textit{ and } u_i < v_i \textit{ for at least one } i \in \{1, \ldots, k\}.$$

Using the relation $\leq_p$ it can now be defined what is meant by a solution of (MOP).

Definition 3.2 *A point $x^\star \in \mathbb{R}^n$ is called* globally Pareto optimal *for (MOP) (or a* global Pareto point *of (MOP)) if there exists no $x \in S \subseteq \mathbb{R}^n$ with*

$$F(x) \leq_p F(x^\star) \textit{ and } f_j(x) < f_j(x^\star) \textit{ for at least one } j \in \{1, \ldots, k\}.$$

If this property is only valid inside a neighborhood $U(x^\star) \subset S \subseteq \mathbb{R}^n$, then $x^\star$ is called locally Pareto optimal *(or a* local Pareto point*).*

The set of all Pareto points is the Pareto set*. Following [3], the set of the function values of all Pareto points is called the* Pareto front*.*

In the literature, one can find several different names for Pareto optimal solutions. Examples are 'efficient solutions' [97, 25], 'noninferior solutions' [3], 'nondominated points' [59], 'vector minimum points' [6], and 'admissible points' [45]. Especially the image of a Pareto optimal solution often is denoted as an *efficient point*.

An equivalent definition of Pareto optimality that is graphically more intuitive, as illustrated in Figure 3.1, can be found e. g. in [25]:

Definition 3.3 *A point $x^\star$ is Pareto optimal, if there exists no $F(x) \in F(S) \setminus \{F(x^\star)\}$ such that*

$$F(x) \in F(x^\star) - \mathbb{R}^k_{+,0}.$$

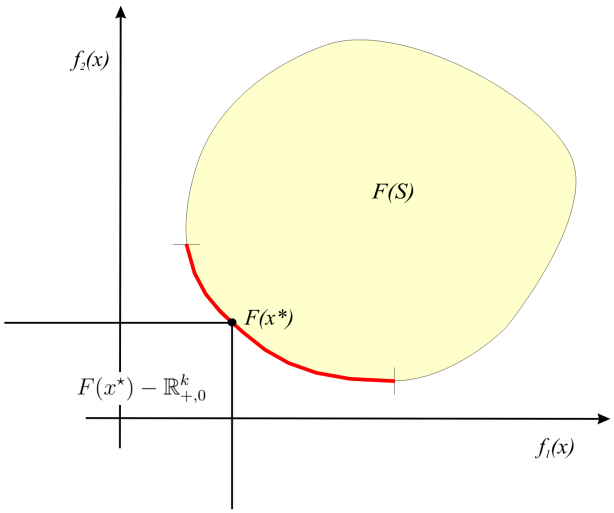

Figure 3.1: Pareto points only intersect in the point $F(x^\star)$ with the region $F(x^\star) - \mathbb{R}^k_{+,0}$

In the literature, many definitions related to Pareto optimality are considered. For example an expansion of the Pareto optimal set to the set that evolves from omitting the condition that $f_j(x) < f_j(x^\star)$ for at least one $j \in \{1, \ldots, k\}$ (cf. Definition 3.2) is denoted as the *weakly Pareto optimal set* (cf. [69]). On the contrary, the *properly Pareto optimal set* is a restriction of the Pareto set to the subset which does not contain points with unbounded trade-offs, i. e. unbounded ratios of change, between the objectives. The main idea is that only those points are properly Pareto optimal for which a decrease of one objective value is only possible at the expense of some reasonable increments in the other objective values. There are several definitions which

reflect this intuitive idea (cf. [69]). One definition was given by Geoffrin in 1968:

Definition 3.4 (Geoffrin, 1968 [40]) *A point $x^\star \in S$ is called* properly Pareto optimal *if it is Pareto optimal and if, additionally, there exists some $M \in \mathbb{R}$ such that for each objective function f_i and each $x \in S$ with $f_i(x) < f_i(x^\star)$ there is at least one f_j such that*

$$f_j(x^\star) < f_j(x) \text{ and } \frac{f_i(x^\star) - f_i(x)}{f_j(x) - f_j(x^\star)} \leq M.$$

In this definition, only the existence of the number M is required. As for the decision maker Pareto points with a predefined bound for the trade-offs may be interesting, this definition has been modified in such a way that the value M is set explicitly:

Definition 3.5 (Shukla, Dutta, Deb, 2005 [93]) *Given a positive number $M > 0$ a point $x^\star \in S$ is called* M-properly Pareto optimal *if it is Pareto optimal and if for all i and $x \in S$ satisfying $f_i(x) < f_i(x^\star)$, there exists at least one index j such that*

$$f_j(x^\star) < f_j(x) \text{ and } \frac{f_i(x^\star) - f_i(x)}{f_j(x) - f_j(x^\star)} \leq M.$$

For two objectives (i. e. $k = 2$) this definition means that any secant on the Pareto front has a slope between $-M$ and $-\frac{1}{M}$.

Example 3.6 Consider the following (very simple) multiobjective optimization problem with two objectives f_1 and f_2 and one variable $x \in \mathbb{R}$:

$$\begin{aligned} \min_{x \in \mathbb{R}} F(x) &= (f_1(x), f_2(x)) \\ \text{with } f_1(x) &= (x-1)^2 \\ \text{and } f_2(x) &= (x+1)^2 \end{aligned}$$

Obviously, the Pareto set is given by all points $x^\star \in [-1, 1]$. All points except the individual optima of f_1 and f_2 are properly Pareto optimal. Setting the value M given in Definition 3.5 to 2, the set of M-properly Pareto optimal points reduces to the inverval $[-\frac{1}{3}, \frac{1}{3}]$. Figure 3.2 illustrates this example.

The following classical result which goes back to Kuhn and Tucker [63] provides a necessary condition for Pareto optimality. The version of the theorem written down here can be found in [53], which itself is a reformulated version of the one given in [43].

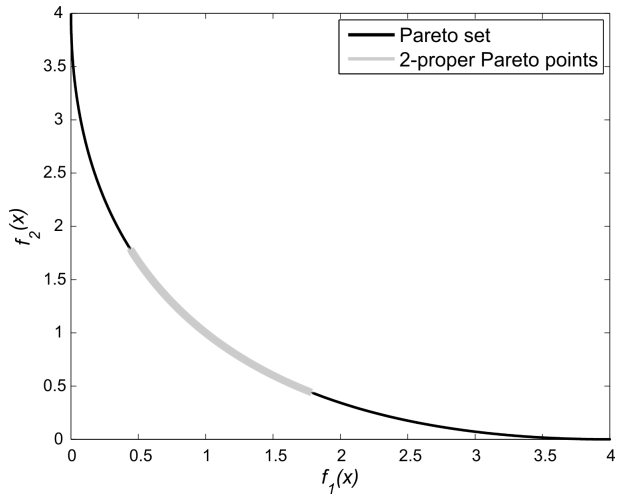

Figure 3.2: Pareto front and its restriction to 2-proper Pareto points for Example 3.6

Theorem 3.7 (Kuhn and Tucker, 1951 [63]) *Let $x^\star$ be a Pareto optimal solution of (MOP). It is assumed that $\nabla h_i(x^\star), i = 1, \ldots, m$ and $\nabla g_j(x^\star)$ for $j \in \{J : g_J(x^\star) = 0\}$ (the active constraints) are linearly independent. Then there exist vectors $\alpha \in \mathbb{R}^k$ with $\alpha_i \geq 0$ for $i = 1, \ldots, k$ and $\sum_{i=1}^k \alpha_i = 1$, $\gamma \in \mathbb{R}^m$ and $\delta \in \mathbb{R}^q$ with $\delta_j \geq 0$ for $j = 1, \ldots, q$ such that*

$$\begin{aligned} \sum_{i=1}^{k} \alpha_i \nabla f_i(x^\star) + \sum_{j=1}^{m} \gamma_j \nabla h_j(x^\star) + \sum_{l=1}^{q} \delta_l \nabla g_l(x^\star) = 0 \\ \delta_j \cdot g_j(x^\star) = 0, \ \forall j = 1, \ldots, q. \end{aligned} \tag{3.1}$$

Following [69], points $x^\star \in \mathbb{R}^n$ that satisfy the Kuhn-Tucker condition (3.1) are called *substationary points*. Given a Pareto point $x^\star$ the vector of multipliers α is called the *weight vector corresponding to $x^\star$*.

Obviously the condition in the Kuhn-Tucker theorem is not sufficient for Pareto optimality in general. In the case of convex[1] objective functions, convex inequality constraints and affine[2] equality constraints, it is proven that for $\alpha > 0$ the Kuhn-Tucker conditions are also sufficient [43]. However, numerical methods often make use of this criterion.

[1] A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex, if $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$ for all $x, y \in \mathbb{R}^n$, $0 \leq \lambda \leq 1$, see e. g. [106].

[2] A function $f : \mathbb{R}^n \to \mathbb{R}$ is affine, if $f(\lambda x + (1-\lambda)y) = \lambda f(x) + (1-\lambda) f(y)$ for all $x, y \in \mathbb{R}^n$, $0 \leq \lambda \leq 1$, see e. g. [106].

3.2 Methods for the Computation of Pareto Points

In this section, first of all a short overview on methods for the computation of Pareto optima is given. Then the 'weighted sums method' and some recently developed set-oriented techniques for the computation of the entire Pareto set will be explained in more detail, as they play an important role in parts of this thesis.

3.2.1 Overview

There are many different approaches for solving multiobjective optimization problems (cf. [69, 3] for example). They range from methods for the computation of single Pareto optimal points to interactive methods to methods for the approximation of the entire global Pareto set.

In the case of the computation of one single Pareto optimum, often a priori information, e. g. a special weighting of the objective functions or an ordering of the objectives is required. Examples for methods of this kind are

- the 'weighted sums method' [36, 111] (which is described in more detail in Section 3.2.2),
- the 'ε-constraint method' [48], in which part of the objective functions are treated as constraints, which have to be greater than or equal to some values ε_j specified in advance,
- different types of reference point methods, in which a distance of some desired reference point to the Pareto front is minimized.

Over the last few years, many algorithms for the computation of the entire Pareto set have been developed. Examples are the 'Normal Boundary Intersection' [14], which especially works well in convex problems, a stochastically motivated algorithm [85] and a technique based on numerical path following [53].

A broad class of algorithms for the computation of entire global Pareto sets is based on evolutionary algorithms. The books of Deb [15, 62], and Coello Coello [12] give an overview on multiobjective evolutionary algorithms (MOEAs).

The use of set-oriented techniques is a different approach, which is very well-suited for multiobjective optimization (cf. [92]). Here, on the one hand the permanent subdivision and selection in parameter space allows the numerical approximation of Pareto sets (so-called *subdivision techniques*). On the other hand, based on given solutions, algorithms which recover the Pareto

set or Pareto front by evaluating points in the neighborhood, have been developed (so-called *recovering techniques*). The subdivision and recovering techniques will be explained in detail in 3.2.3. These methods have been used for the computation of Pareto sets in the applications described in this work.

3.2.2 The Weighted Sums Method

The weighted sums method, also called the 'weighting method', goes back to Gass and Saaty [36] and Zadeh [111]. It is a very popular approach which makes use of the intuitive idea of converting the multiobjective optimization problem into a single objective one. For this, the objective functions are summed up, each multiplied with an individual weight. More precisely, k weights α_i are chosen such that $\alpha_i \geq 0$ for $i = 1, \ldots, k$ and $\sum_{i=1}^{k} \alpha_i = 1$ and the problem

$$\begin{aligned} &\min_x g_\alpha(x) \\ &\text{s. t. } x \in S \subseteq \mathbb{R}^n, \end{aligned} \tag{3.2}$$

with $g_\alpha(x) = \sum_{i=1}^{k} \alpha_i f_i(x)$ is considered.

Varying the weights α_i, different Pareto points can be obtained by solving (3.2) – in the case of convex objective functions even all Pareto points can be computed in this way. The reason for this is that the shape of the Pareto front is also convex in such a situation. Moreover, the optimization of the weighted sums results in different points on the Pareto front for different weight vectors.

In contrast to this, for nonconvex objective functions the Pareto front can contain nonconvex parts as illustrated in Figure 3.3 on the right. Here, the nonconvex part is defined to be a subset of the Pareto front that contains no global optima of the weighted sum of the objectives for every weight vector. Pareto points, which are mapped into the nonconvex part of the Pareto front, can have the same weight vector as other Pareto points whose weighted sum has a smaller value (cf. Figure 3.4), as they are only local minima or saddle points of $g_\alpha(x)$. In [13], Das and Dennis give a trigonometric argument, why – in the case of two objectives – the weighted sums method cannot be used to compute points in the nonconvex part.

It is an interesting question to find out if a Pareto front contains nonconvex parts. If one assumes that the Pareto front is connected, then any nonconvex part contains a region in which the Pareto front 'curves inside the image of the feasible region'. This part of the Pareto front will be called

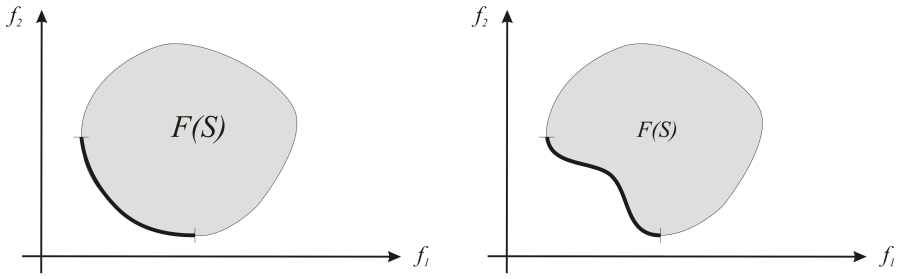

Figure 3.3: Typical shape of a Pareto front for a convex problem (left figure) and possible shape in a nonconvex problem (right figure)

a 'dent'. In Section 3.3 the formal definition of a dent is given and an approach which allows the numerical computation of dents in Pareto fronts is presented.

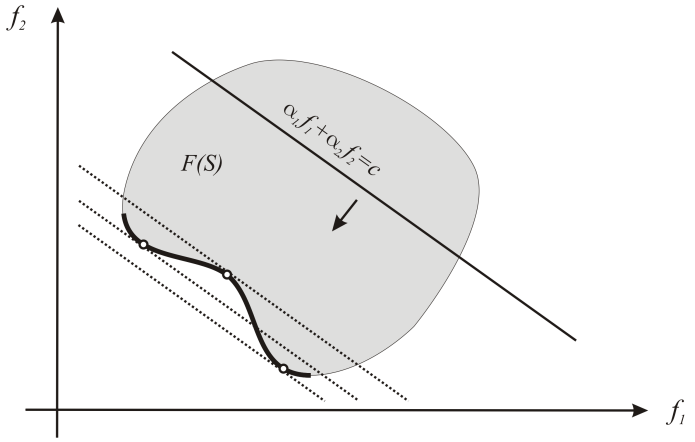

Figure 3.4: Schematic illustration of the weighted sums approach

3.2.3 Set-oriented Methods

For the computation of the entire Pareto set, recently some set-oriented methods have been developed (cf. [20, 87, 88, 90], and, for an overview, [92]), which can be categorized into two main classes: the *subdivision* and the *recovering* techniques. The subdivision techniques are of global nature and can even deal with multiobjective optimization problems in which no

analytical derivatives are available. The drawback is that the number of optimization variables which can be treated efficiently is restricted to moderate dimensions. The recovering techniques are of local nature but also work well for higher-dimensional problems (especially in the case of smooth objective functions).

The principal approach of the subdivision techniques can be characterized as follows: Based on a covering of the feasible parameter space (a box), the Pareto set is approximated numerically through a successive refinement and selection of boxes in parameter space. Figure 3.5 visualizes this approach.

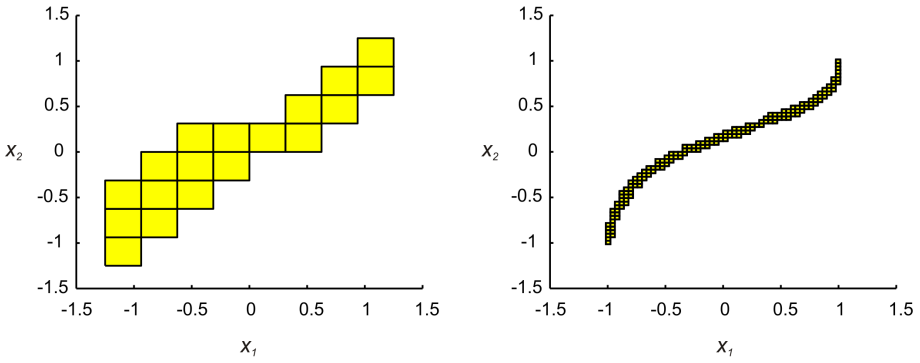

Figure 3.5: Approximation of the Pareto set for the MOP "$\min_x F = (f_1, f_2)^T : \mathbb{R}^2 \to \mathbb{R}^2$" with $f_1(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^4$ and $f_2(x_1, x_2) = (x_1 + 1)^2 + (x_2 + 1)^2$ using the subdivision techniques with 10 (on the left) or 16 (on the right) subdivision steps respectively

After subdividing the boxes into two smaller boxes in each subdivision step, test points are generated in each box. One or a few steps of a descent direction for the MOP are applied to each test point and only those boxes are kept in which at least one of the iterated points has stayed. Many different methods for the choice of test points and descent directions have been implemented. More details about the algorithms, which are implemented in the software tool GAIO[3], and a proof of convergence can be found in [20, 88, 90] and [87].

If one or a few Pareto optima are already known, then the recovering techniques (cf. [87, 89]) allow the computation of the entire, connected components of the Pareto set that contain these starting points. For this, little boxes around the starting points are defined, and other Pareto optimal points are generated by studying neighboring boxes. Here, one uses the fact that – under certain smoothness and regularity assumptions – Pareto sets are locally $(k-1)$-dimensional manifolds (cf. [53]). Thus, it is guaranteed that

[3]Global Analysis of Invariant Objects, see www.math.upb.de/∼agdellnitz

in the neighborhood of Pareto optima other Pareto optimal points can be found. To decide which neighboring boxes may contain Pareto optima, test points are generated and iterated in the same way as in the above described subdivision techniques. Figure 3.6 visualizes the principal functioning of the recovering techniques.

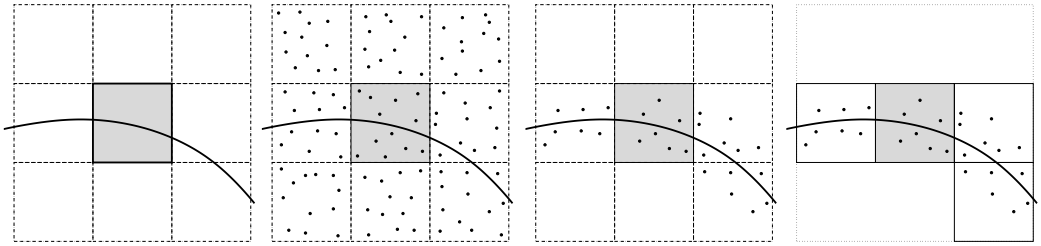

Figure 3.6: Principal functioning of the recovering techniques, here using test points for the generation of new Pareto points in neighboring boxes (the black curve illustrates the unknown Pareto set) [96]

For high-dimensional problems (i. e. MOPs with many optimization variables), a special image set oriented recovering algorithm has been developed in [18]. For a given initial, Pareto optimal solution a box on the Pareto front is generated around the image of this solution. Step by step, all neighboring boxes are inserted that contain points on the Pareto front. The insertion of boxes is based on the idea to create vectors of desired values for the objectives, so-called *targets* T, in the neighborhood of the given boxes. Then the following distance minimization problem is solved:

$$\min_x \|F(x) - T\|.$$

Using this algorithm, the entire Pareto set can be covered for unconstrained multiobjective optimization problems with convex objective functions, as – under certain smoothness assumptions – Pareto sets are parts of $(k-1)$-dimensional manifolds [53] and, in the convex case, they are connected and the map from the Pareto set to the Pareto front is bijective [69]. In Figure 3.7, a schematic representation of the image set oriented recovering technique is given.

3.3 Dents in Pareto Fronts

In Figure 3.3 it has already been illustrated that the Pareto front may curve inside the image of the feasible region in the case of nonconvex objective functions. As already mentioned in Section 3.2.2 Pareto points whose images lie in such a dent cannot be computed by using the weighted sums method. The

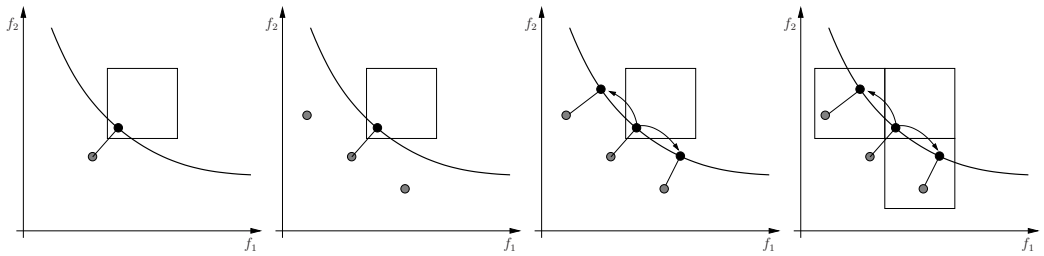

Figure 3.7: Principal functioning of the image set oriented recovering technique (the black curve is the unknown Pareto front)

reason is that two or more Pareto points satisfy the Kuhn-Tucker equations with the same weight vector α while the weighted sum $\sum_{i=1}^{k} \alpha_i f_i(x)$ is not minimal for all these solutions. The following definition gives a mathematical description of a dent.

Definition 3.8 (Dent point, Dent preimage) *Let $P \subseteq S$ be the Pareto set of a multiobjective optimization problem $\min_{x \in S} F(x)$ with $F : \mathbb{R}^n \to \mathbb{R}^k$, $F(x) = (f_1(x), \ldots, f_k(x))^T$ and f_i at least twice continuously differentiable $\forall\, i = 1, \ldots, k$. Let*

$$g_\alpha(x) = \sum_{i=1}^{k} \alpha_i f_i(x),$$

$g_\alpha : \mathbb{R}^n \to \mathbb{R}$, and $\alpha_i \in [0, 1]$, $\sum_{i=1}^{k} \alpha_i = 1$.

A point $x^\star \in P$ is called a dent preimage *if it is a saddle point of g_α. The corresponding point $y^\star = F(x^\star)$ on the Pareto front is called a* dent point*.*

Definition 3.9 (Dent, dent border, complete dent) *Let $P \subseteq S$ be the Pareto set of an at least twice continuously differentiable multiobjective optimization problem $\min_{x \in S} F : \mathbb{R}^n \to \mathbb{R}^k$ and let $PF = F(P)$ be the Pareto front. Let $y^\star \in PF$ be a dent point. Then, the connected component of dent points which includes $y^\star$ is called a* dent corresponding to $y^\star$*, denoted by $D_{y^\star}$:*

$$D_{y^\star} = \{y \in PF | \exists \delta \geq 0 \text{ and } \exists c : [0, \delta] \to PF, c \text{ continuous, with } c(0) = y^\star, \\ c(\delta) = y, \text{ and } c(s) \text{ is a dent point } \forall\, s \in [0, \delta]\}\,.$$

A dent $D_{y^\star}$ is called complete*, if*

$$\partial PF \cap \overline{D_{y^\star}} = \emptyset,$$

where ∂PF is the boundary of the Pareto front PF as a subset of $\partial F(S)$ (with the induced topology from $\mathbb{R}^k$).

The boundary $\partial D_{y^\star}$ of a complete dent $D_{y^\star}$ (seen as a subset of PF) is called dent border *and a boundary point $y_b \in \partial D_{y^\star}$ is called a* dent border point. *A point $x_b \in P$ with $F(x_b) = y_b$ is called a* dent border preimage *of y_b.*

Remark 3.10 In [53] dents have been studied from a differential geometric point of view. In this book it has been shown that – under certain geometrical assumptions on the multiobjective optimization problem – saddle points of g_α occur if and only if the corresponding point on the Pareto front has at least one negative principal curvature.

In [53] it has already been considered what happens during the transition from a minimizer x_1 of g_{α_1} to a saddle point x_2 of g_{α_2} on a connected Pareto front i. e. during the transition of non-dent preimages to dent preimages (α_1 and α_2 denote the weight vectors corresponding to x_1 and x_2, respectively):
Assume that the non-dent preimage x_1 can be connected to the dent preimage x_2 by a continuous curve $\gamma : [0,1] \to P \subseteq S$ with $\gamma(\tau) = (x(\tau), \alpha(\tau))$, $\gamma(0) = (x_1, \alpha_1)$ and $\gamma(1) = (x_2, \alpha_2)$. To each curve point $\gamma(\tau)$ the n-tuple of eigenvalues of $\frac{\partial^2}{\partial x^2} g_\alpha(x)$, denoted by $(e_1(\tau), \ldots, e_n(\tau))^T$, is assigned, where α again is the corresponding weight vector to x. This leads to another continuous curve $\tilde{\gamma} : \tau \mapsto (e_1(\tau), \ldots, e_n(\tau))^T$ corresponding to γ. As x_1 is a minimizer of g_{α_1}, $\tilde{\gamma}(0) > 0$. In the saddle point x_2 of g_{α_2}, there exists an index $i \in \{1, \ldots, n\}$ such that $e_i(1) < 0$. Because of the continuity of $\tilde{\gamma}$ there must exist $\bar{\tau} \in [0,1]$ with $e_i(\bar{\tau}) = 0$.

To sum up, in dent border points a zero-eigenvalue of the Hessian of g_α occurs.

Definition 3.11 (Simple dent border point/preimage) *Let $P \subseteq S$ be the Pareto set of an at least twice continuously differentiable multiobjective optimization problem $F : \mathbb{R}^n \to \mathbb{R}^k$ and let $PF = F(P)$ be the Pareto front. Let $y_b^\star \in PF$ be a dent border point and let $x_b^\star \in P$ be a dent border preimage of $y_b^\star$.*

Then, $y_b^\star$ is called a simple dent border point *if the zero eigenvalue of the Hessian $g''_\alpha(x_b^\star)$ is simple. In this case, $x_b^\star$ is called a* simple dent border preimage.

Example 3.12 (Computation of dents)
Consider the two objectives

$$f_1(x, \lambda) = \frac{1}{2}(\sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} + x_1 - x_2) + \lambda \cdot e^{-(x_1 - x_2)^2}$$
$$f_2(x, \lambda) = \frac{1}{2}(\sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} - x_1 + x_2) + \lambda \cdot e^{-(x_1 - x_2)^2}$$

with a fixed value $\lambda = 0.6$ and $x = (x_1, x_2)$. Then, the Pareto set can be approximated e. g. by use of the set-oriented techniques described in Section 3.2.3.

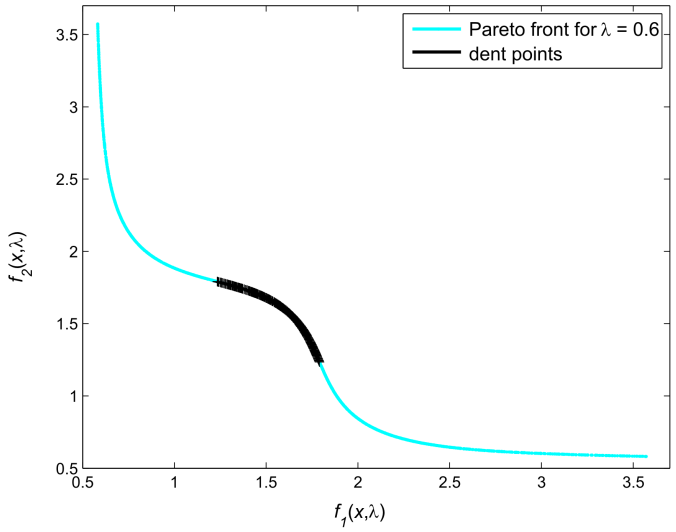

Figure 3.8: Pareto front and dent points (black) for Example 3.12

The algorithm returns a set of boxes that covers the Pareto set. Within these boxes, a number of test points is evaluated, the best of which are in the following considered as the Pareto set. By solving the Kuhn-Tucker equations of $F = (f_1, f_2)$ for each of these points, the corresponding weight vectors $\alpha \in \mathbb{R}^k$ can be computed. Then, the eigenvalues of the Hessians of the weighted sums of the objectives are determined. All points x, in which both eigenvalues > 0 and eigenvalues < 0 exist, are dent preimages. The Pareto front and the resulting dent points for this example are visualized in Figure 3.8.

3.4 Treatment of Multiobjective Optimization Problems in several Technical Applications

In close cooperation with engineers working within the CRC 614 "Self-optimizing systems and structures in mechanical engineering" [4] we have considered several technical applications in which multiple objectives occur that are desired to be optimized at the same time. The applications presented in this

[4] www.sfb614.de

work are subsystems of the *RailCab* vehicle which is a novel linear motor driven railway system including the latest technologies and innovative prototypes, developed by the project RailCab ("Neue Bahntechnik Paderborn", cf. [73]). In particular, we have considered a hybrid energy storage system, the guidance module and the doubly-fed linear drive. The new aspect of our study of these applications is the numerical approximation of entire Pareto sets. Based on the resulting global picture on the solution set, the engineers were now able to choose suitable Pareto optimal solutions. In the case of the linear drive we even developed a situation-dependent heuristic which allows the automatic choice of suitable Pareto optimal solutions during operation time. In the case of the guidance module certain reference trajectories are desired to be optimized with respect to several objectives. This would directly lead to an optimal control problem with multiple objectives. As the guidance module is modeled as a differentially flat system, we were able to reformulate the optimal control problem into a nonlinear multiobjective optimization problem. This type of reformulation has been introduced by Murray et al (cf. [72, 104, 70, 31]) for the case of one single objective function. In this work it is shown that the same idea works for the multiobjective case.

In the following, first a short introduction into the functionality of the RailCab system is given. Afterwards, more detailed information about the three subsystems considered in this work and our new results for the multiobjective optimization of these subsystems are presented.

3.4.1 The RailCab System

The RailCab system is a novel rail-bound personal rapid transit (PRP), developed by the project RailCab ("Neue Bahntechnik Paderborn") [73].

The idea is that the RailCab system acts like a "taxi cab on rails" which on the one hand saves energy by using common track sections in a convoy and on the other hand is very comfortable for the passengers, as the desired destination is directly approached. In particular, thus it is not necessary to change trains and the timetable is not relevant. The concept is also very well-suited for light cargo transport.

RailCab vehicles travel autonomously and independently of other vehicles on the common, existing rails, equipped with additional motor elements of the linear drive between the rails and novel passive switches, which allow the RailCabs to steer actively into the desired direction. The steering is performed by the so-called *guidance module* for the front and rear axle. This module can actively control the lateral displacement of the RailCab vehicle on the rails. Within the clearance between the flanges and the rail-heads, the

Figure 3.9: Picture of the RailCab test vehicles (source: Collaborative Research Center 614 (SFB614))

RailCab vehicle can be moved freely. This is a big advantage over traditional railway systems, in which an uncontrolled periodic movement and at higher speed even a striking of the rail-heads against the flanges occurs.

The RailCab is propelled by a doubly-fed linear drive that consists of a primary motor part between the rails of the track and a secondary part on the vehicle. Energy is transferred from the track to the vehicle by use of a magnetic field and the thrust is generated on the secondary motor part (see e. g. [110, 80]).

In order to guarantee the availability of energy without any interruption the RailCab is equipped with an onboard energy storage. At present, the energy storage consists of batteries. In the future, a hybrid energy storage system consisting of batteries and double layer capacitors will be installed. So far, a test plant for the new hybrid energy storage system has been built at the Chair of Power Electronics and Electrical Drives, University of Paderborn, Germany.

Furthermore, the RailCab vehicle contains an *active suspension module* which gives the vehicle the possibility to improve the passenger comfort [51].

Within the RailCab project, two RailCab vehicles and a test track in the scale of 1:2.5 have been built next to the campus of the University of Paderborn. Figure 3.9 shows a picture of the test vehicles. The test track is an oval of about 530 m track length which contains an artificial hill with an altitude difference of about 2.5 m and gradients up to 5.3% and a novel passive switch. More detailed information about the RailCab system can for

example be found in [1].

3.4.2 Multiobjective Optimization of the Hybrid Energy Storage System[5]

The hybrid energy storage system (HES) contains two different types of energy storages: batteries and double layer capacitors (DLC). Batteries are well-suited for long term storage as they have a high energy density. The drawback is that batteries have a poor power density and a small number of full load cycles. On the contrary, double layer capacitors perfectly fit for short term storage as they can be cycled many times without degradation and offer a high power densitiy but only a low energy density. By the combination of the two energy storages the individual advantages are utilized.

Figure 3.10 shows a picture of the test rig for the hybrid energy storage system that has been built at the Chair of Power Electronics and Electrical Drives, University of Paderborn, Germany.

From the mathematical point of view, the challenge in this application is to find an operation strategy that distributes the power flows to the respective storage devices in an optimal way during a drive of the vehicle. Here, several aims come into play. In the study described in [83] two objectives are considered: the minimization of the normalized energy losses e_{losses} of the HES and the maximization of the power reserve (expressed by the minimization of the function p_{res}).

To facilitate the calculation of the operating strategy the planned drive of the vehicle is divided into several sections of one to five minutes. A multiobjective optimization with respect to the above-mentioned objectives is executed for each of these sections. For technical reasons, the sum of the powers of both energy storages and the losses is equal to the total on-board power of the vehicle which is given as a profile for the planned drive. Thus, only the battery current has to be optimized over time. The resulting profiles of the battery power, the state of charge of the battery, the DLC power and the state of charge of the DLC can be computed directly from the optimized profile for the battery current. The optimization variables stem from a discretization of values for the battery current over time. The discretization points are chosen at those points in time where either the total power changes significantly or a certain energy has been produced. The final trajectory is set together by piecewise constant values of the battery currents.

In Figure 3.11 one possible profile for the total power of the HES is given. In this profile, the total power changes drastically at six different points in

[5] in cooperation with the Chair of Power Electronics and Electrical Drives, University of Paderborn, Germany, cf. [66, 83]

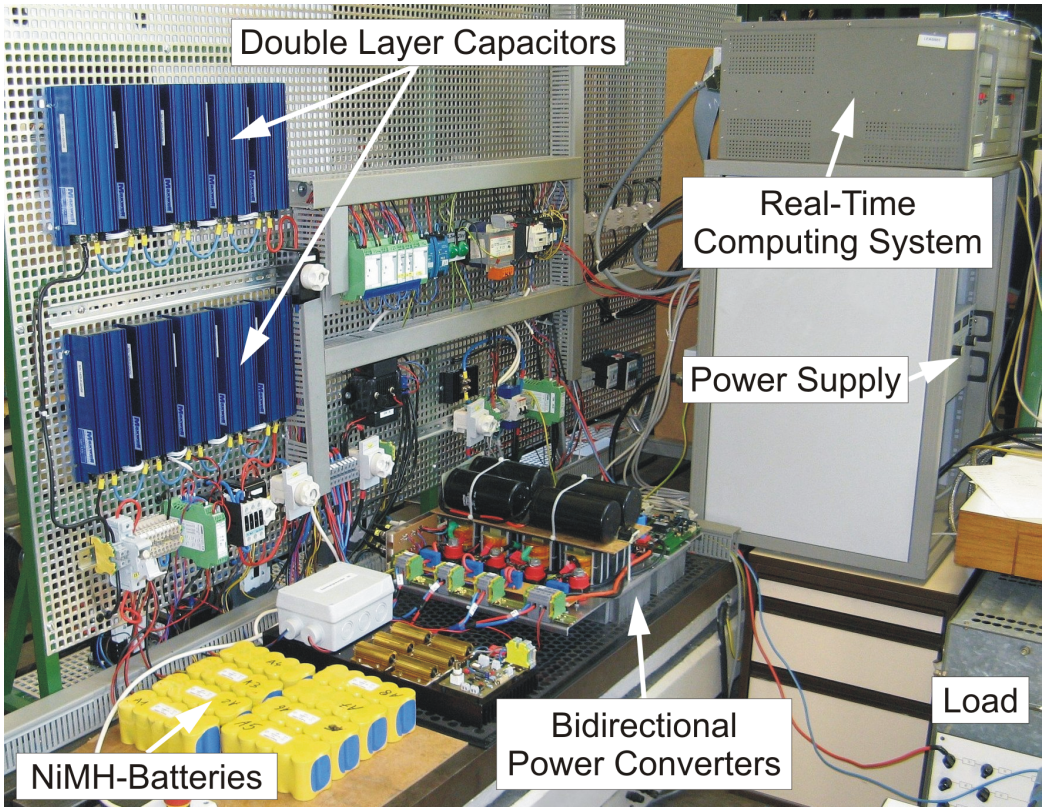

Figure 3.10: Photograph of the test rig of the hybrid energy storage system [83]

time $T_0, \cdots, T_5$ (cf. Figure 3.11). The values of the battery current at these points in time are optimized with respect to the minimization of e_{losses} and p_{res}. The battery and the DLC are supposed to have a medium state of charge at the beginning ($T = T_0 = 0$) in this example. Figure 3.12 shows an approximation of the Pareto front. It has been computed by use of a special image set oriented predictor-corrector method for the generation of Pareto points. This algorithm uses basic ideas of the image set oriented method and has been especially tailored to multiobjective optimization problems with two objectives in which no analytic derivatives of the objectives are computable and evenly spread solutions on the Pareto front are desired (cf. [83] for more details). Each point within this Pareto front is related to an optimized operating strategy. The three marked points (cf. Figure 3.12) have been chosen to demonstrate the results. The corresponding trajectories of the battery current are plotted in Figure 3.13. Figure 3.14 gives the resulting states of charge of both the battery and the DLC. One can observe that in Point 1, where the energy losses are minimal, the values of the battery current are near zero and the state of charge of the battery is almost constant. As

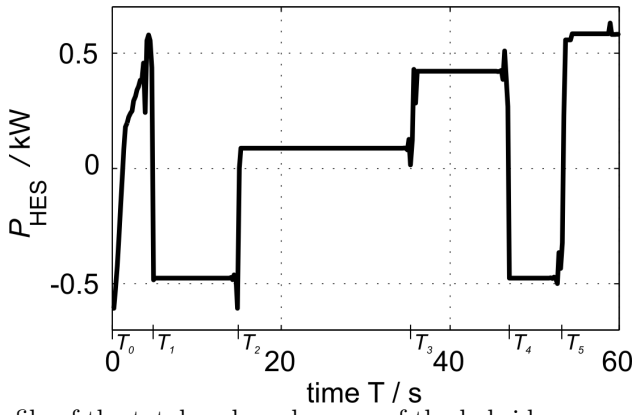

Figure 3.11: Profile of the total on-board power of the hybrid energy storage system for a section of one minute

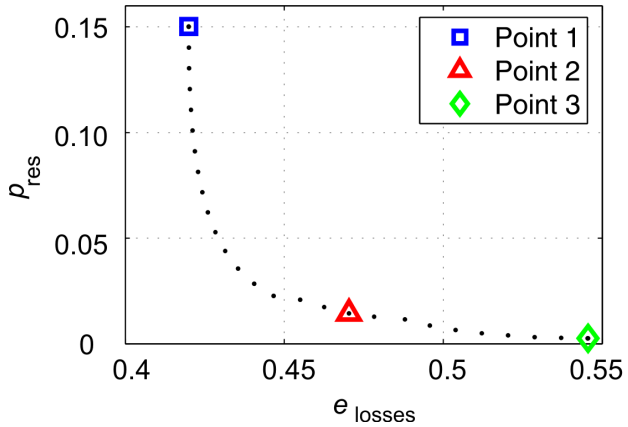

Figure 3.12: Approximation of the Pareto front for the HES assuming the total power profile given in Figure 3.11

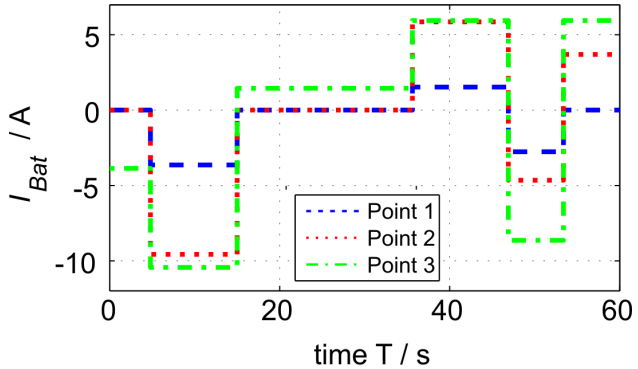

Figure 3.13: Resulting trajectories of the battery current I_{Bat} based on the total power profile given in Figure 3.11

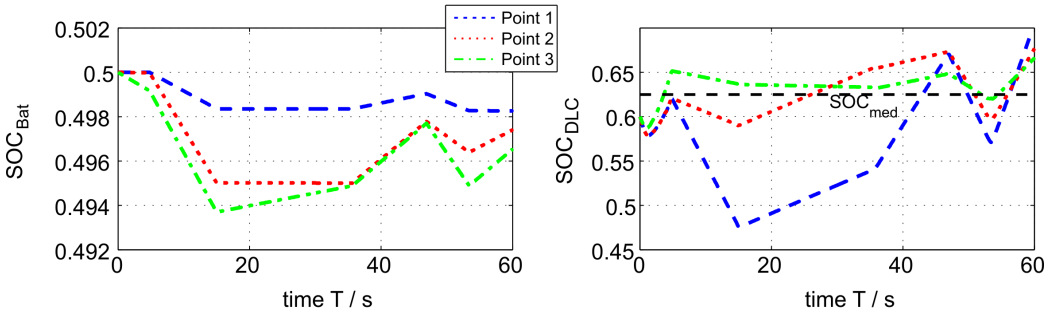

Figure 3.14: Results for the state of charge of the battery (on the left) and the state of charge of the DLC (on the right) based on the total power profile given in Figure 3.11

a consequence, the state of charge of the DLC fluctuates much more in this point. On the contrary, in Point 3 in which p_{res} is minimal and thus the power reserve is maximal, the state of charge of the battery changes more significantly and the state of charge of the DLC is varied only within a small range around the medium value SOC_{med}. Point 2 is one example for an optimal compromise.

3.4.3 Self-optimization of the Guidance Module[6]

As mentioned above, the guidance module allows to actively control the lateral displacement of the RailCab vehicle in the rails. Within a given clearance, the RailCab can be moved freely. This is very important, because track laying does not result in ideally straight rails and flange strikes, i. e. bumpings of the rail-heads against the flanges, have to be avoided because they cause noise as well as wear on the wheels and rails. Figure 3.15 shows a typical rail and a sketch of the clearance, which is the maximum distance between the flanges and the rail-heads.

Within the clearance, the RailCab can be steered along arbitrary reference trajectories. The challenge was to compute (Pareto-) optimal trajectories that meet several aims:

1. minimize the deviation of the vehicle from the track centerline, i. e. maximize “safety”

2. maximize the passenger comfort

3. minimize the average energy consumption of the hydraulic actuators

[6] in cooperation with the Chair of Control Engineering and Mechatronics, University of Paderborn, Germany, cf. [38, 39]

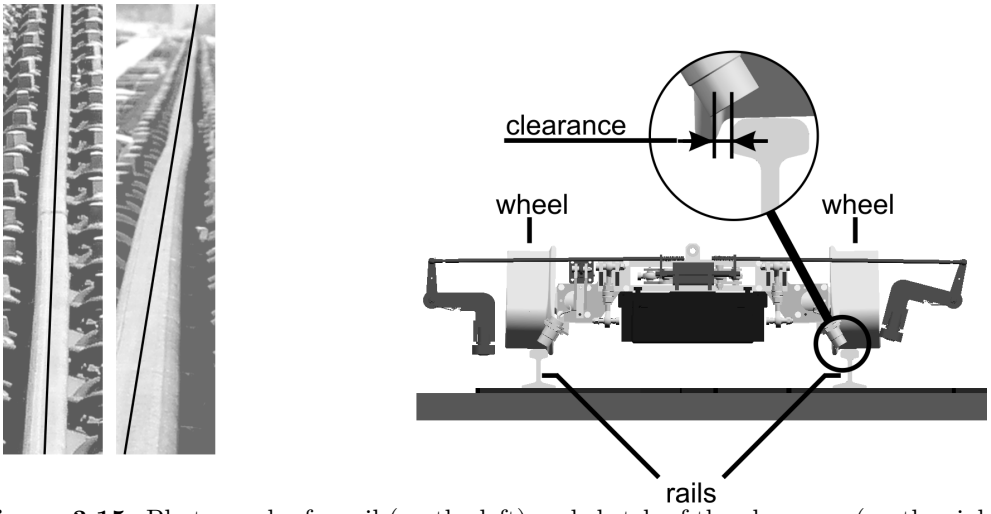

Figure 3.15: Photograph of a rail (on the left) and sketch of the clearance (on the right) [38]

4. maximize the steering angle reserve

Based on a linear model of fourth order for the lateral dynamics of the RailCab vehicle (see e. g. [38] for more details), an optimal control problem with the above mentioned objectives is formulated. In this model, the controlled outputs are (differentially) flat, i. e. the inputs and states can be represented as a function of these flat outputs and a finite number of their derivatives w. r. t. time (cf. [31]).

Definition 3.13 ((Differential) Flatness [31, 72]) *A system*

$$\dot{x} = f(x, u)$$

with states $x \in \mathbb{R}^n$ *and controls* $u \in \mathbb{R}^m$ *is called* differentially flat *or just* flat *if there exists a fictitious output* $y \in \mathbb{R}^m$ *with*

$$\begin{aligned} y &= h(x, u, \dot{u}, \ddot{u}, \ldots, u^{(p)}) \\ &\quad \textit{such that} \\ x &= \alpha(y, \dot{y}, \ddot{y}, \ldots, y^{(q)}) \\ u &= \beta(y, \dot{y}, \ddot{y}, \ldots, y^{(q)}). \end{aligned} \tag{3.3}$$

Here, h, α *and* β *denote real-analytic functions*[7] *and* $p, q \in \mathbb{N}$. y *is called a* flat output.

[7] Functions that locally can be developed into a Taylor series are called real-analytic functions (see e. g. [61]).

Differentially flat system can especially be utilized in trajectory optimization (cf. [104]). The big advantage is that in this case the trajectories can be optimized in the space of the outputs y and afterwards, the corresponding inputs and states can be computed.

Thus, an optimal control problem of the form

$$\begin{aligned} &\min_{x\in\mathbb{R}^n, u\in\mathbb{R}^m} J(x,u) \\ &\text{s. t. } \dot{x} = f(x,u) \end{aligned} \tag{3.4}$$

with a flat system $\dot{x} = f(x,u)$ can be transformed into an optimization problem of the form

$$\min_{y\in\mathbb{R}^m} f(y), \tag{3.5}$$

where y denotes the flat output (cf. e. g. [104, 78] and [70]).

This concept can be easily extended to the case of several objectives as required in the trajectory optimization of the guidance module. Then, the optimal control problem is transformed into a conventional multiobjective optimization problem of the form

$$\min_{y\in\mathbb{R}^m} F(y), \tag{3.6}$$

where F maps from $\mathbb{R}^m$ to $\mathbb{R}^k$ and y denotes the flat output.

In the case of the guidance module, it is assumed that the measured position of the rails (the track centerline), is known a priori. The desired reference trajectories of length s_h for both the front and the rear axle are approximated by piecewise polynomials, so-called splines. Many different types of splines can be found in the literature – most common are cubic splines and B-splines. In [70] B-splines have been used for the trajectory optimization. In the case of the optimization of the guidance module, the use of B-splines has no additional benefits and thus cubic splines which have a more intuitive parametrization have been applied. Using cubic splines, the trajectory is described by a finite number n_p of parameters y_j, which are interpolation values at given knot points x_j (track positions) and which have to be optimized. Additional boundary conditions are defined for the cubic spline that keep the transition between two sequenced trajectory parts continuously differentiable. The computational advantage that might arise from using B-splines is negliblible in the scenario considered here, as only 25 collocation points have to be evaluated. A detailed description of cubic splines can be found in [82]. More details about the special setting of the spline for the application considered (choice of knot points etc.) are given in [39].

For the computation of Pareto optimal trajectories, the image set oriented recovering algorithm described in Section 3.2.3 has been used. The fact that the RailCab has to stay within the clearance leads to a hard constraint in the multiobjective optimization problem given by

$$|y - y_{tr}| < r_{cl},$$

where r_{cl} is half the clearance, y_{tr} the track centerline positioned in the middle of the clearance and y the current position of the axle center. This constraint can be easily included by setting the center of the initial boxes (defined in preimage space) to y_{tr} and the radius to $y_{tr} + r_{cl}$ in each knot point.

In a first study described in [38] the trajectories of the guidance module have been optimized with respect to the first three objectives given on page 51. The exact formulas for these objectives are given in [38]. Figure 3.16 shows an approximation of the Pareto front. To each point within this Pareto front corresponds a Pareto optimal trajectory on which the RailCab vehicle can be steered.

Figure 3.17 gives two projections of the Pareto front. In this plot it becomes clear that energy and comfort are correlated. An improvement of safety causes a deterioration in both comfort and energy. This is due to the fact that a smooth, comfortable trajectory also requires less steering and thus less energy consumption of the hydraulic actuators.

Two points within the Pareto front have been chosen (marked by a circle and a square) to demonstrate the results. The circle is an example for a more safe trajectory and the square for a more comfortable one. In Figure 3.18 the corresponding trajectories and the trajectories which stem from single objective optimizations for each of the three objectives are given. (Here, only the optimized interpolation values at the knot points, connected with lines, are plotted.) The black line is the (shifted) track centerline and the gray lines describe the clearance around it.

One can observe that the trajectory which is more safe (line with circles) stays close to the centerline whereas the more comfortable trajectory (line with squares) "cuts the corners" and is smoother. As expected, the energy optimal (dashed) and comfort optimal (dash-dot) trajectories lie close together.

In [39] the fourth objective mentioned on page 51, the steering angle reserve, has been additionally considered. The exact formula for this objective is given in [39]. Figure 3.19 shows two projections of the three-dimensional Pareto front which has also been approximated by the image set oriented recovering algorithm.

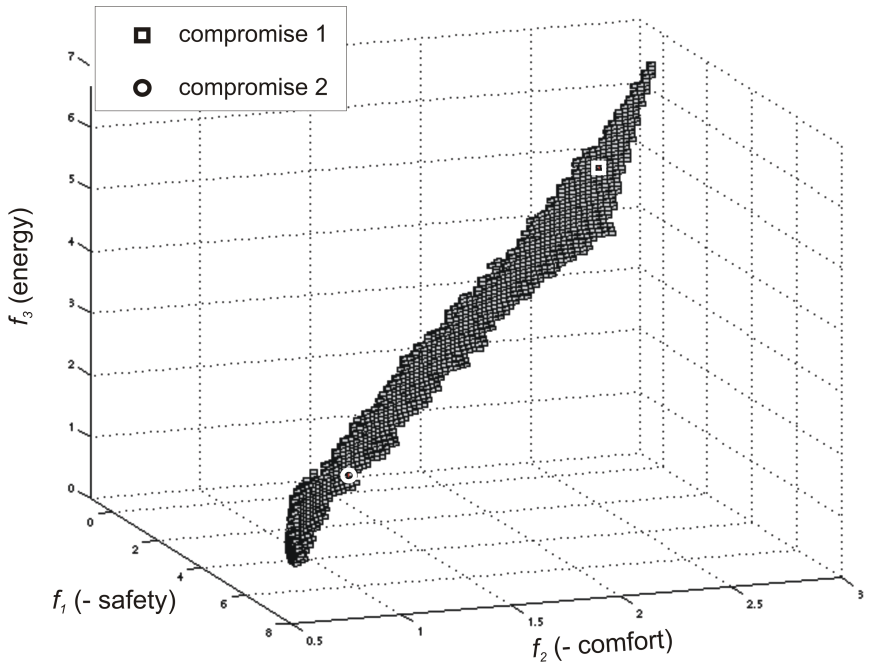

Figure 3.16: Numerically approximated Pareto front for the multiobjective optimization of the guidance module

The guidance module becomes a self-optimizing system by continuously running through the following three steps (cf. Section 2.5):

Step 1: Analysis of the current situation:
The values of the velocity and power allocation of the RailCab vehicle and control errors are measured. Track irregularities have been estimated offline in advance and are selected from a database.

Step 2: Determination of the system's objectives:
The Pareto set has a priori been approximated offline by use of the image set oriented recovering algorithm described in Section 3.2.3. To save disk space, only the points of the Pareto front and the weights that would lead to the same Pareto point when optimizing a weighted sum of the objectives are stored in a database (the objectives are convex). Within this set, a point that fits for the current situation is chosen

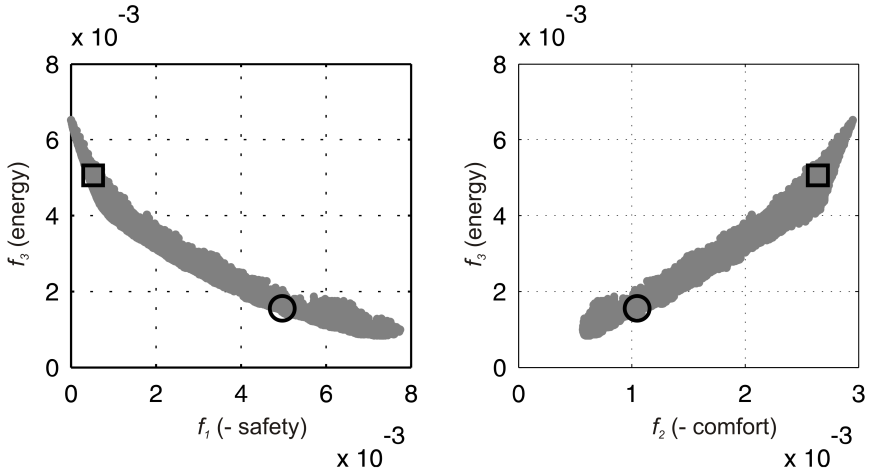

Figure 3.17: Two projections of the Pareto front for the multiobjective optimization of the guidance module

by requiring that the power allocation is adequate and the number of flange strikes is minimal.

Step 3: Adaptation of the system's behavior:
During runtime, the trajectory is optimized with respect to the weighted sum of the objectives using the weight determined in Step 2. Thus, the Pareto optimal trajectory can be reconstructed during runtime by using for example the methodology described in [104].

3.4.4 Multiobjective Optimization of the Linear Drive[8]

As mentioned above, the RailCab contains a doubly-fed linear drive that consists of a primary motor part between the rails and a secondary motor part on the vehicle. If both components of the motor are energized, then a magnetic traveling wave emerges in the motor. This magnetic field generates the thrust (cf. [49]). Figure 3.20 visualizes the working principle of the doubly-fed linear drive.

During operation time, power is transferred from the primary to the secondary motor part. The absolute value of this transferred power depends on the operating point of the linear motor given through the currents and frequencies in both motor parts. In [50] a mathematical model of the doubly-fed

[8] in cooperation with the Chair of Power Electronics and Electrical Drives, University of Paderborn, Germany, cf. [81, 108, 86]

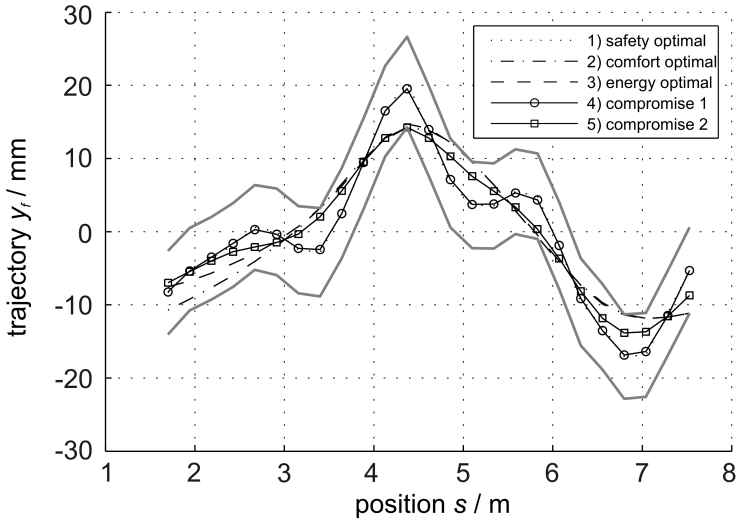

Figure 3.18: Examples of Pareto optimal trajectories for the RailCab vehicle

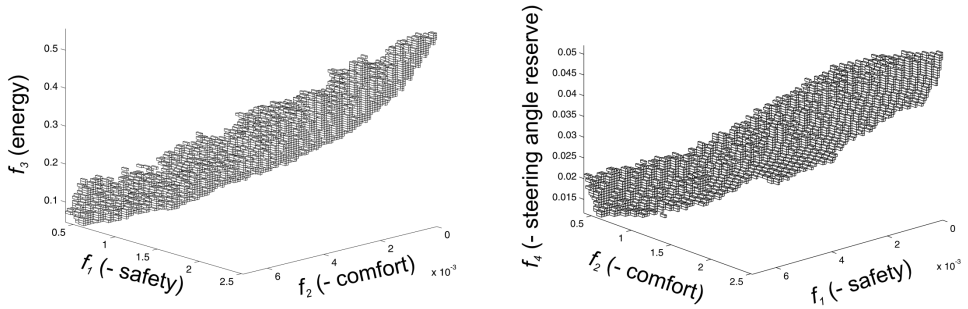

Figure 3.19: Two projections of the approximated Paretofront for the guidance module with four objectives

linear drive has been developed. For a given target position of the RailCab, the reference values of the current position $x_M^\star$ and the speed $v_M^\star$ are assigned by a jerk limited profile generator. This generated profile is also called a *maneuver*. The reference thrust $F_M^\star$ is determined as well as the demanded electrical power $P_B^\star$ that has to be transferred into the vehicle. Taking into account these two values and the measured speed v_M of the RailCab, reference values for the currents and the frequencies of both motor parts can be determined by the operating point assignment.

The operating point is desired to meet two objectives at the same time (cf.

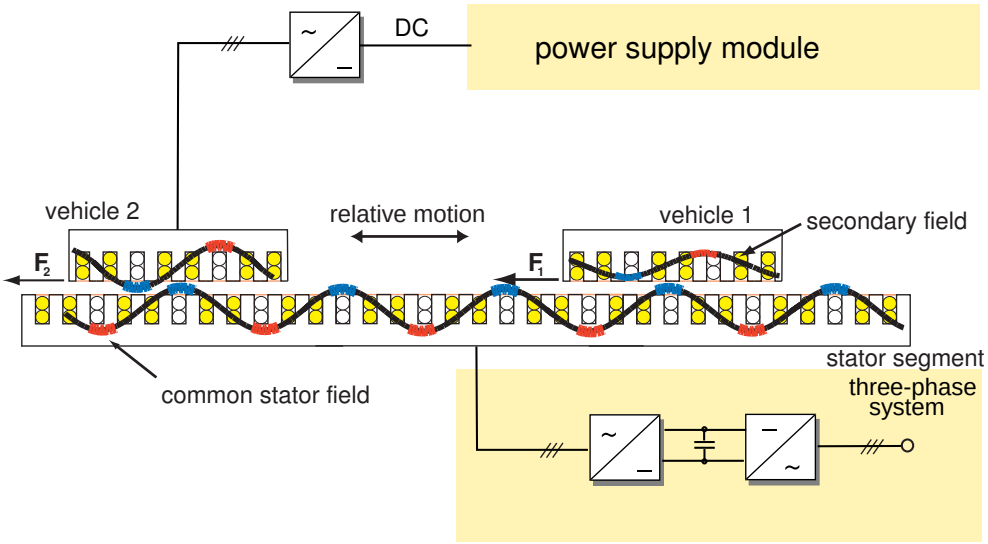

Figure 3.20: Working principle of the doubly-fed linear drive [49]

[81]). On the one hand, the degree of efficiency η_{LM} has to be maximized and on the other hand the power factor η_{SN} that takes the ratio of real output power to the apparent power of the linear motor into account, has to be maximized, too. This leads to the multiobjective optimization problem

$$\min_{x} \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} = \min_{i_{\mathrm{Sd}}} \begin{pmatrix} -\eta_{\mathrm{LM}}(i_{\mathrm{Sd}}) \\ -\eta_{\mathrm{SN}}(i_{\mathrm{Sd}}) \end{pmatrix}$$

in which the current $i_{\mathrm{Sd}} \in \mathbb{R}$ is the optimization variable. These objectives are both convex.

By use of set-oriented multiobjective optimization techniques (see Section 3.2.3), an approximation of the Pareto optimal operating points was obtained. This optimization is performed for discretized values of the maneuver and for each point in time a Pareto set has been computed. One of the resulting Pareto fronts is plotted in Figure 3.23 on the left.

In more detail, here a simulated drive of the RailCab around the test track has been considered (cf. [81]). Figure 3.21 shows the velocity profile for the RailCab. Additionally, in this figure assumed values for the state of charge of the battery and the temperature of the secondary motor part are plotted. Based on these values, the decision which point within the Pareto set, i. e. which value for the current i_{Sd} is adjusted to the motor, is made.

More precisely, a so-called *decision heuristic* that realized the choice of one Pareto point has been developed. The first step of this decision heuristic is based on the following technical properties:

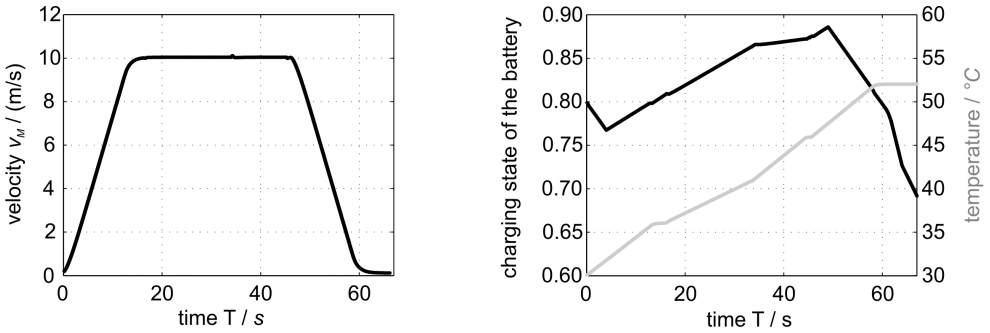

Figure 3.21: Profiles of the velocity of the RailCab, the state of charge of the battery and the temperature of the primary motor part which form the basis for the optimization

- If the temperature in the secondary motor part rises, then an optimization of the power factor should be preferred.
- If the energy storage of the RailCab is discharging and the state of charge of the battery is low, then the optimization of the degree of efficiency becomes more important (the reason is that in this case the reactive power oscillation between the secondary motor part and the battery, the so-called flicker-effect, generates an extra load to the battery).
- If the battery is being charged, then only the temperature θ affects the choice of the Pareto point.

These properties are put into formulas in such a way that the upper point on the Pareto front, which corresponds to the maximum with respect to η_{SN}, is chosen, if the state of charge and the temperature are high and the lower point on the Pareto front, which corresponds to the maximum of η_{LM}, is chosen if both the load value and the temperature are low. In between, points are chosen by a linear interpolation between the individual optima depending on the load and temperature values.

Figure 3.22 depicts how the decision heuristic works in this specific example in more detail: First, the extremal points PP_1 (the minimum of f_2) and PP_2 (the minimum of f_1) of the Pareto set are determined. For this specific linear motor scenario, PP_1 is selected if the state of charge is at least 90% and the temperature is at least 80° C. Conversely, PP_2 is selected if the state of charge is at most 20% and the temperature is at most 20° C. In the case of parameter values between these extremal points a Pareto point is chosen by using linear interpolation.
Consider for example the state of charge. Let q_{cur} be the current state of charge, $q_{\mathrm{min}} = 0.2$ the lowest possible value for q_{cur} and $q_{\mathrm{range}} = 0.7$ the

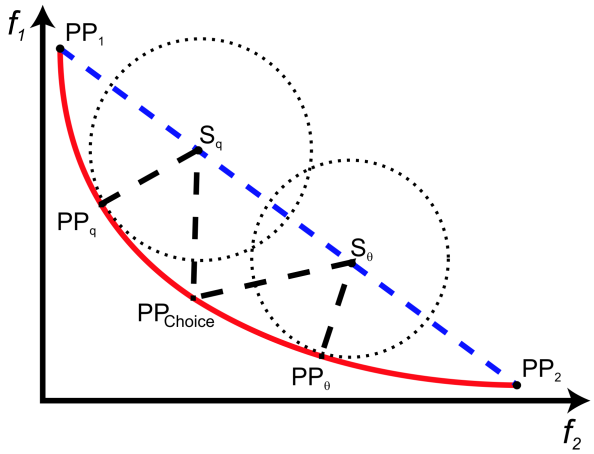

Figure 3.22: Working principle of the first step of the decision heuristic for the operating point assignment of a doubly-fed linear motor [86]

greatest amount by which q_{cur} can vary. Then q_{scale} and T_q are defined as

$$q_{scale} = \frac{q_{\text{cur}} - q_{\text{min}}}{q_{\text{range}}}$$

and

$$S_q = q_{scale} \cdot PP_2 + (1 - q_{scale}) \cdot PP_1.$$

The procedure for S_θ works analogously with $\theta_{\text{min}} = 20°$ C and $\theta_{\text{max}} = 80°$ C. Since the temperature and the state of charge can vary independently, linear interpolation only works for each parameter individually. The optimal Pareto point PP from the Pareto front with respect to the decision variables is chosen by minimizing the following formula

$$w \cdot (S_q - PP) + (1 - w) \cdot (T_\theta - PP)$$

with some weight $w \in [0, 1]$ which in this application has been set to 0.5.

After this decision based on external parameters, the decision heuristic always contains a second step in which the curvature of the Pareto front in the chosen point is studied. If the Pareto front is "flat" or "steep" around the chosen point, then a slight movement to another point on the Pareto point causes great benefits in one objective and only little losses in the other one. Thus, in this second step, the tangent to the Pareto front is approximated numerically, and the solution is possibly moved to another point on the Pareto front in which the tangent has a (predefined) minimal or maximal slope M. Using the definition of M-properly Pareto optimal points (see Definition 3.5 on page 36) one can say that in the second step that M-proper Pareto point

is chosen whose objective value is as near as possible to the objective value of the point determined in the first step. In Figure 3.23 on the right this second step is visualized.

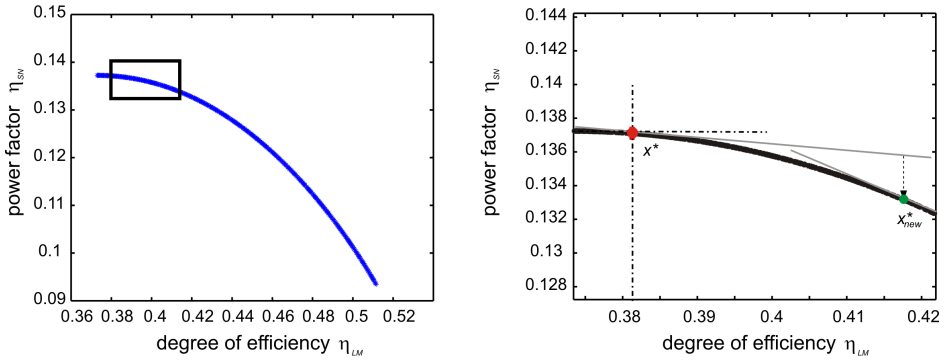

Figure 3.23: Results of the multiobjective optimization for the linear drive: On the left, the Pareto front corresponding to $t = 20s$ is plotted. In the right plot, the black box from the left is zoomed in more detail and a sketch of the working principle of the decision heuristic is given.

The values for the charging state of the battery and the temperature of the secondary motor part are not consistent with measured values which change very slowly over time. To demonstrate the working principle of the decision heuristic, here values that change considerably have been chosen.

Figure 3.24 shows the results of the multiobjective optimization for a round trip of the RailCab on the test track with the given profiles of velocity (and also some given profiles of the force and power), states of charge of the battery and temperatures of the secondary motor part. In the left plot, the maximum possible values of η_{LM} and η_{SN} which result from a single objective optimization are drawn over time. As the objectives are conflicting, it is not possible that both values are reached at the same time. The dashed lines are the values of η_{SN} (lower dashed line) and η_{LM} (upper dashed line) that are achieved with the multiobjective optimization together with the decision heuristic as described above. The right plot of Figure 3.24 shows the corresponding values of the optimization variable i_{Sd} that lies as expected between the values of i_{Sd} for the individual maxima of the objectives.

In this figure, one can also observe the effect of the decision heuristic: for example in the time interval from 50 to 60 seconds, where the values of the charging state of the battery and the temperature of the secondary motor part become higher and higher, Pareto points are chosen that more and more move towards the maximum of η_{SN}.

The computation of Pareto sets for many discretized points of time, as

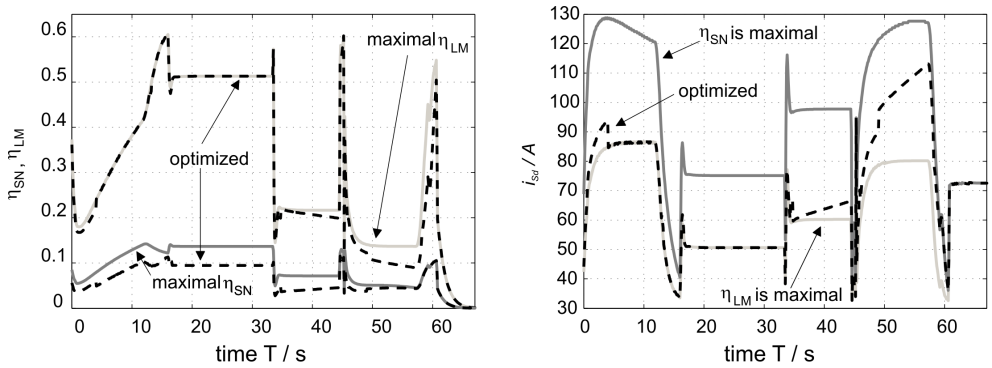

Figure 3.24: Results of the multiobjective optimization for the linear drive illustrated between the maxima of each individually optimized objective. On the left, the values of the objectives are plotted whereas on the right, the corresponding values of the optimization variable, the current i_{Sd}, are shown.

performed in the operating point assignment here, is computationally costly and cannot be executed during operation time. To allow a multiobjective optimization during operation time, the same problem has been considered as a time-dependent multiobjective optimization problem. For this kind of problem a numerical algorithm has been used that allows the efficient computation of Pareto optimal paths over time. This algorithm and its application to the doubly-fed linear drive are described in Section 4.3.

Chapter 4

Parametric Multiobjective Optimization

In this chapter, multiobjective optimization problems that additionally depend on an external parameter $\lambda \in \mathbb{R}$ are studied. First, in Section 4.1 the problem is stated formally, the Kuhn-Tucker theorem is reformulated for this context and a motivating example is given.

Parametric multiobjective optimization problems occur in many real world applications.

On the one hand, time-dependencies are included in many applications. These often can be modeled by use of such an external parameter λ. By the combination of techniques from multiobjective optimization and numerical path following methods, Pareto optimal paths over time can be computed. Numerical path following (cf. Section 2.4.2) allows to compute the solutions of parameter-dependent systems of equations. Multiobjective optimization and numerical path following can be linked together by means of the Kuhn-Tucker necessary condition (3.1) (see page 37) which characterizes Pareto optimal points as solutions of a certain system of equations. This approach will be explained in more detail in Section 4.3. How this new concept has been applied to the optimization of the operating point assignment for a doubly-fed linear drive is described in Section 4.3.2. Part of the contents of this section have already been published in [108] and [86].

On the other hand, the external parameter λ may describe the influence of any other parameter to the objective functions. Imagine for instance the design process of a car which is desired to be optimal with respect to comfort (first objective) and safety (second objective). On the road, the car will be caught in a cross-fire of influences, like side wind, wet roads, or changes in temperature. The only information one has is that these influences can be estimated to lie in a certain interval, probably given by the weather forecast. Mathematically, this leads in the simplest case to the parametric multiobjective optimization problem $\min_x F(x, \lambda)$ with $F : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^k$ – thus the solution (the Pareto set) also depends on the parameter λ. For the designer of a car those Pareto points are desirable, where the Pareto points or their images under F change as little as possible while the parameter varies.

After a short summary of the state-of-the-art of robustness concepts in multiobjective optimization in Section 4.4.1, two new concepts that describe how to compute these robust Pareto points are introduced in Section 4.4.2.

In Section 3.3 dents in Pareto fronts have been studied. The consideration of dents is extended to parametric multiobjective optimization problems in Section 4.5 of this chapter. Here, the connection between results from bifurcation theory and the computation of points in which dents originate or vanish under the variation of the external parameter λ is pointed out.

4.1 Problem setting

An unconstrained (one-) parametric multiobjective optimization problem is given as

$$\min_x \{F(x,\lambda) : x \in \mathbb{R}^n, \lambda \in [\lambda_{\text{start}}, \lambda_{\text{end}}] \subseteq \mathbb{R}\}, \tag{ParMOP}$$

where F is defined as the vector of the objective functions, i. e.

$$F : \mathbb{R}^n \times [\lambda_{\text{start}}, \lambda_{\text{end}}] \to \mathbb{R}^k,\ F(x,\lambda) = (f_1(x,\lambda), \ldots, f_k(x,\lambda)).$$

The solution set of (ParMOP) is a λ-dependent family of Pareto sets. Every point in this family satisfies the necessary condition of Kuhn and Tucker with respect to the x-variables. As (ParMOP) is an unconstrained multiobjective optimization problem, Theorem 3.7 reduces to the fact that there exist multipliers $\alpha_1, \ldots, \alpha_k \in \mathbb{R}_{+,0}$ such that

$$H_{\text{KT}}(x,\alpha,\lambda) = \begin{pmatrix} \sum_{i=1}^k \alpha_i \nabla_x f_i(x,\lambda) \\ \sum_{i=1}^k \alpha_i - 1 \end{pmatrix} = 0, \tag{4.1}$$

where (x,λ) is a solution of (ParMOP).

For the numerical solution of (4.1), the non-negativity requirement for α_i might present a problem. This problem is avoided if one uses an equivalent[1] formulation saying that there exist multipliers $t_1, \ldots, t_k \in \mathbb{R}$ such that

$$H_{\text{KT}}(x,t,\lambda) = \begin{pmatrix} \sum_{i=1}^k t_i^2 \nabla_x f_i(x,\lambda) \\ \sum_{i=1}^k t_i^2 - 1 \end{pmatrix} = 0, \tag{4.2}$$

where (x,λ) is a solution of (ParMOP). In the following, both α_i and t_i^2 will be used synonymously.

[1] Equivalence is meant here in the sense that the same set of substationary points (x,λ) is described.

Definition 4.1 *If $x \in \mathbb{R}^n$ satisfies the Kuhn-Tucker condition (4.2) for a specific value of λ, then it is called – as in the non-parametric case – a* substationary point. *The set of all substationary points for the respective value of λ is denoted by S_λ.*

Example 4.2 Two objective functions $f_{1,2} : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ defined as

$$\begin{aligned} f_1(x,\lambda) &= \lambda((x_1-2)^2+(x_2-2)^2)+(1-\lambda)((x_1-2)^4+(x_2-2)^8) \\ f_2(x,\lambda) &= (x_1+2\lambda)^2+(x_2+2\lambda)^2. \end{aligned}$$

in which the parameter λ is varies from 0 to 1 are considered. To provide insight into the behavior of the Pareto sets for different values of λ in this problem, numerical approximations of the Pareto sets for some specific values of λ are shown in Figure 4.1 on the left. The right plot shows an approximation of the entire Pareto sets for $\lambda \in [0,1]$ in x_1-x_2-λ-space.

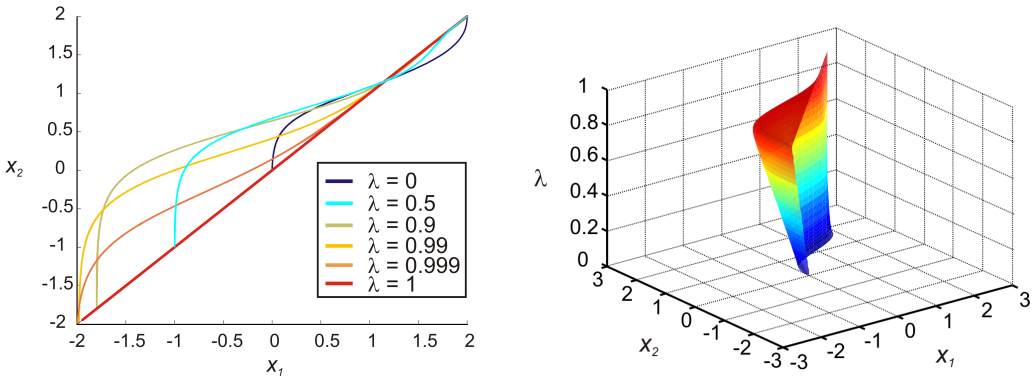

Figure 4.1: Pareto sets for some specified values of λ (on the left) and numerical approximation of the entire, λ-dependent Pareto set in x_1-x_2-λ-space (on the right) for Example 4.2.

4.2 Numerical Path Following and Multiobjective Optimization

In Chapter 2, Section 2.4.2, numerical path following methods have been introduced. These methods allow to compute solutions to a parameter-dependent system of equations

$$H(y,\lambda) = 0.$$

In the case of multiobjective optimization such a system of equations is given by the necessary condition of Kuhn and Tucker (cf. Equation (4.2)). It can be used to analyze the behavior of parameter-dependent sets of substationary

points with respect to variations of the parameter λ. The Kuhn-Tucker equations consist of $n+1$ equations in $n+k+1$ unknowns (given by $x \in \mathbb{R}^n$, $t \in [0,1]^k$, $\lambda \in \mathbb{R}$). Numerical path following methods can be applied to systems of equations which contain only one more unknowns (given through λ) than equations. Thus, additionally to the Kuhn-Tucker equations some more equations have to be derived. These equations define the direction of the path between the λ-dependent sets of substationary points.

In Figure 4.2 the principal idea of computing paths between λ-dependent sets of substationary points is visualized.

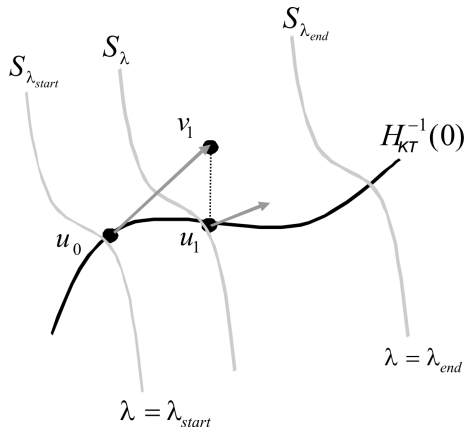

Figure 4.2: Application of numerical path following methods to parameter-dependent sets of substationary points.

How to choose the direction of the path depends on the application one considers. In the following, additional equations for time-dependent applications and for paths that allow the study of robustness against λ-variations are described. The initial solutions, i. e. starting points for $\lambda = \lambda_{\text{start}}$, stem from a numerical approximation of the Pareto set for $\lambda = \lambda_{\text{start}}$.

Numerical path following methods have already been used in the context of multiobjective optimization (see e. g. [53] and [26]), but only for the generation of Pareto sets itself and not for varying external parameters in parametric multiobjective optimization problems.

4.3 Time-dependent Multiobjective Optimization Problems

In this section, the parameter λ is identified with the time T, i. e. the Pareto set of the multiobjective optimization problems considered here changes over time. In engineering problems it is often required that Pareto optimal solutions over time can be computed very efficiently, preferably during runtime.

As numerical path following methods allow the fast generation of solutions under the variation of the parameter, in this case time, they perfectly fit for the solution of these problems. How the solution paths are computed is described in more detail in Section 4.3.1. In Section 4.3.2 the application of the new concept for the optimization of the operating point assignment of a doubly-fed linear motor is described. The contents of this section can also be found in [108].

4.3.1 Computation of Solution Paths

As mentioned above, the parameter-dependent Kuhn-Tucker system (4.2) consists of $n+1$ equations and $n+k+1$ variables. If the weights $\alpha_i = t_i^2$ are fixed to values $\bar{t}_i^2$, then the Kuhn-Tucker system reduces to n equations and $n+1$ variables. The values $\bar{t}_i$ can be computed for every starting point x on the initial Pareto set by solving the Kuhn-Tucker equations for $\lambda = \lambda_{\text{start}}$. The system $H(y,\lambda) = H(x,\bar{t},\lambda)$ is then simply given by the Kuhn-Tucker equations with fixed values of t. It is assumed that the Jacobian of $H_{\text{KT}}(x,t,\lambda)$ w. r. t. (x,t,λ) is regular. By this, one can apply the numerical path following methods as described in Section 2.4.2 directly and follow the curve which is generated when varying $\lambda \in [\lambda_{\text{start}}, \lambda_{\text{end}}]$.

The result is a series of points that lie on the solution curve starting in $x_{\text{start}} \in S_{\lambda_{\text{start}}}$ and approximately ending in a point $x_{\text{end}} \in S_{\lambda_{\text{end}}}$. For every starting point in the initial Pareto set for $\lambda = \lambda_{\text{start}}$ one obtains such a path, i. e. a series

$$\{u_0 = (x_{\text{start}}, \bar{t}, \lambda_{\text{start}}), u_1, \cdots, u_N = (x_{\text{end}}, \bar{t}, \lambda_{\text{end}})\},$$

where N denotes the corresponding number of predictor steps in the path following algorithm, i. e. the solution curve conists of $N+1$ points.

Example 4.3 Consider again (as in Example 4.2) the two objective functions $f_{1,2} : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ defined as

$$\begin{aligned} f_1(x,\lambda) &= \lambda((x_1-2)^2 + (x_2-2)^2) + (1-\lambda)((x_1-2)^4 + (x_2-2)^8) \\ f_2(x,\lambda) &= (x_1+2\lambda)^2 + (x_2+2\lambda)^2. \end{aligned}$$

in which the parameter λ is varied from 0 to 1.

An approximation of the entire Pareto set for $\lambda = 0$ has been computed using the set-oriented subdivision technique described in Section 3.2.3. By inserting every point x contained in this Pareto set into the Kuhn-Tucker system and setting $\lambda = 0$, the corresponding $\alpha = \bar{\alpha}$ was computed by simply solving these linear equations. For every starting point x in the computed

Pareto set α is fixed to the computed value and paths for each starting point are computed by use of the software package AUTO2000 [23]. In Figure 4.3 some of the resulting paths are illustrated.

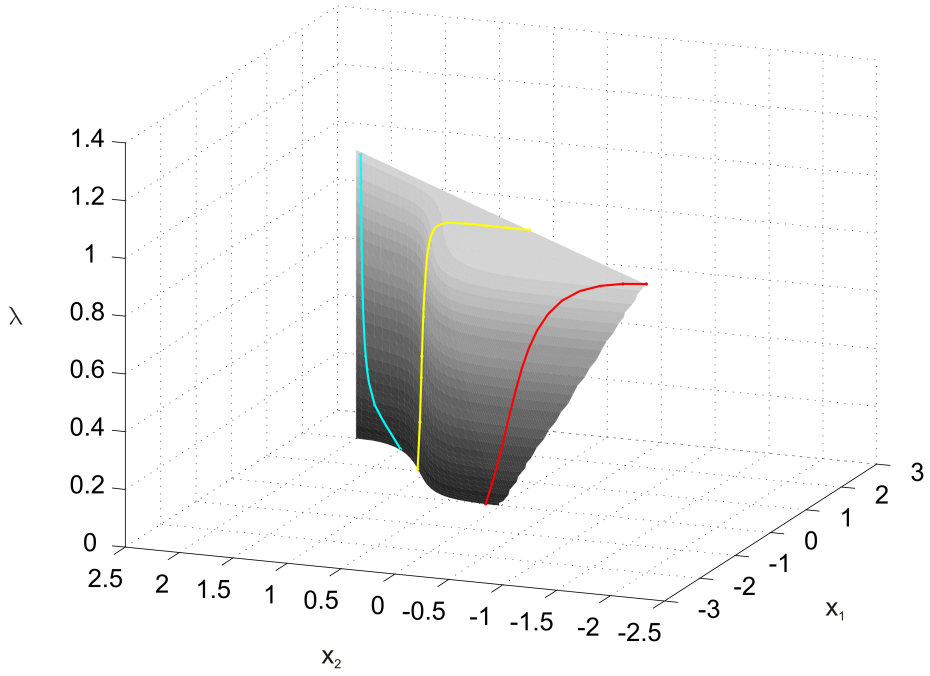

Figure 4.3: Three examples for paths with a fixed weight vector α in x_1-x_2-λ-space, visualized on the λ-dependent Pareto set, for Example 4.3.

In the case of time-dependent multiobjective optimization problems in which solutions are required to be generated during runtime (online) it is not possible to compute entire Pareto sets at each point in time (or a fine discretization of time). By keeping the weighting of the objectives constant for certain intervals within which the solution is adapted to the current point in time by the numerical path following method, solutions to the time-dependent multiobjective optimization problem can be obtained efficiently. The following three steps reflect this idea:

> **Step 1:** Compute the entire Pareto set for the given problem at an initial point in time. Based on a certain decision heuristic (i. e. a calculation rule for the choice of one point from the Pareto set that is based on external variables which change situationally) the most preferred adjustment for the system at this initial point in time is calculated.

Step 2: The corresponding ratio of the objectives (weight vector α) is set fixed. Using path following techniques, the solution curve corresponding to this α is computed over time.

Step 3: After a certain time, the entire actual Pareto set is recomputed, the decision heuristic is applied again, and the adjustment of the system is updated. Then proceed with Step 2.

Figure 4.4 sketches Step 1 and Step 2 in objective space.

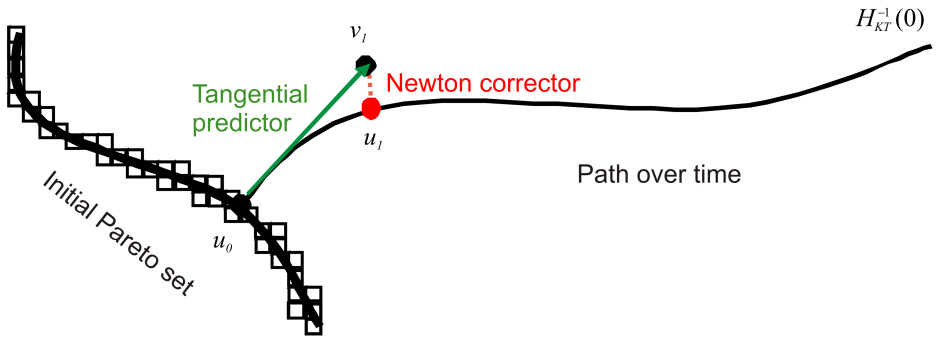

Figure 4.4: Visualization of Step 1 and Step 2 in the proposed treatment of time-dependent multiobjective optimization problems

There are different ways how to decide that the new entire Pareto set for the current point in time has to be computed in Step 3. The first trivial idea is to compute the Pareto set at fixed points in time, e. g. every minute (assuming that the dimension of the problem is low enough). This computation can be performed in parallel to the computation of the path of solutions so that the assignment of the optimal solutions never has to be interrupted. The computation of the entire Pareto set can also be started anew in reaction to external influences. As an example in the case of the doubly-fed linear motor (cf. Section 4.3.2) some precomputed velocity profiles are contained in the optimization. If these profiles cannot be realized any more, e. g. because the vehicle has to brake unexpectedly, then a recomputation of the entire Pareto set and the corresponding path (based on the new profiles) is necessary.
Including such criteria, the system performs the self-optimizing scheme described in Section 2.5.

4.3.2 Application: Online Optimization of the Operating Point Assignment of a Linear Motor[2]

In Section 3.4.4 the multiobjective optimization of the operating point assignment of a doubly-fed linear motor has been described. Whereas in that section a simulated drive on the test track for the RailCab has been studied, in this section the scenarios considered are driven on a test bed of the linear motor. Figure 4.5 shows a picture of the test bed.

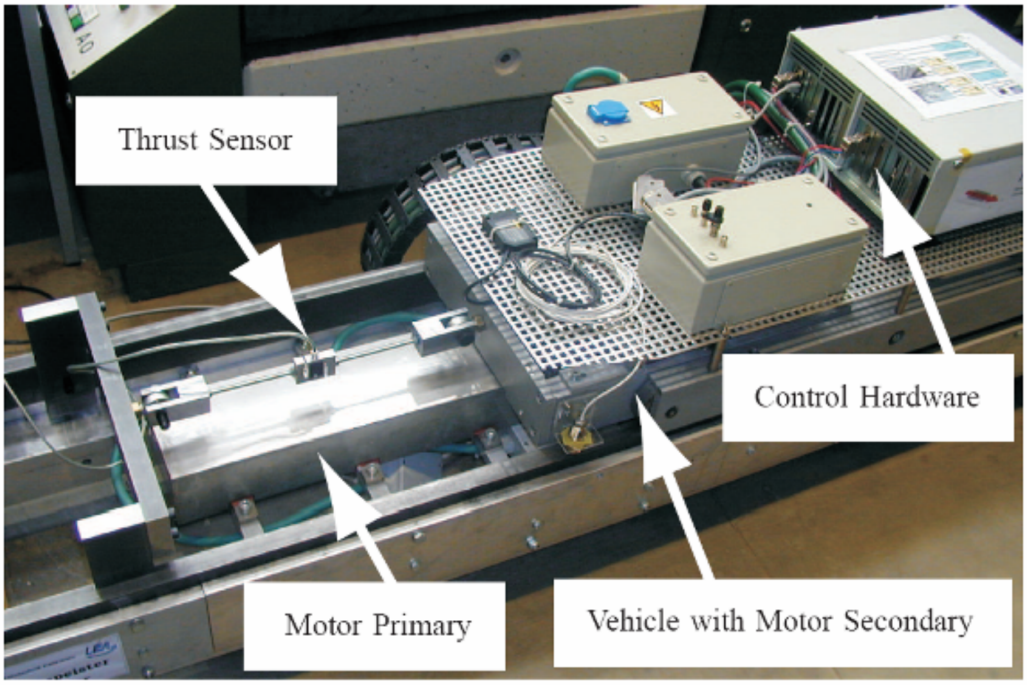

Figure 4.5: Photograph of the linear motor test bed of the Chair of Power Electronic and Electrical Drives, University of Paderborn [86]

In this application, the optimization results depend on the current velocity, thrust and power of the vehicle which all can be represented as depending on time. In the approach described in Section 3.4.4 Pareto sets for discretized points in time have been approximated. This cannot be realized during runtime as the computational effort is too high. By use of the algorithm for the treatment of time-dependent multiobjective optimization algorithms described in the last section, Pareto optimal paths of solutions can be generated 'online', i. e. during runtime. For this, the three steps given on page 68 have been employed and adapted for this special application.

[2] in cooperation with the Chair of Power Electronics and Electrical Drives, University of Paderborn, Germany, cf. [108, 86]

Step 1: Computation of Pareto sets and decision heuristic

As already described in Section 3.4.4 the Pareto set is approximated by using a set-oriented subdivision technique contained in the software tool GAIO (cf. Section 3.2.3). After the computation of the entire Pareto set for an initial point in time, e. g. $T = T_{\text{start}} = 0$, it is sensible to choose one point in this set as the most preferred adjustment of the system. The decision heuristic for the application considered here has also already been described in Section 3.4.4.

Step 2: Numerical path following

As the objective functions in the case considered are convex all Pareto optimal points are solutions of the Kuhn-Tucker equations. Thus, given a Pareto optimal point that has been chosen in Step 1, the corresponding weight vector α can be computed by solving the Kuhn-Tucker system for the fixed point in time. Then α is set fixed to the computed value and a standard predictor-corrector method with a tangential predictor and a Newton corrector (cf. Section 2.4.2) is applied. The resulting solution curve is stored in a buffer for the reference values and fed to the controller of the linear motor at the corresponding positions of the vehicle.

Step 3: Recomputation of the Pareto set

In the case of the doubly-fed linear motor two different scenarios have been derived that require a recomputation of the Pareto set:

1. motor parameters that are modeled as fixed values have changed, i. e. for example the air gap has changed, or
2. the decision variables, i. e. the state of charge of the battery and/or the temperature of the secondary motor part, have changed drastically, or,
3. the absolute value of the thrust is (almost) zero.

The repeated evaluation of Step 1 to Step 3 mirrors the idea of self-optimization (cf. Section 2.5): The *analysis of the current situation* is performed through the evaluation of the profiles for the maneuver, the values of the decision variables and the motor parameters for certain points in time. The *determination of the system's objectives* is realized by the computation of the Pareto set and the application of the decision heuristic, and the *behavior of the system is adapted* by assigning the optimized values of the currents resulting from the numerical path following to the motor during runtime. The optimization algorithm is running on a PC that communicates with the

linear motor test bed via LAN.

The implementation of the algorithm is supported by the concept of the Operator-Controller-Module (OCM) (see Figure 4.6 for the OCM of the linear motor and [33] for a more detailed description of the general OCM-concept).

At the lowest level the controllers perform the current- and speed-control under hard real-time conditions. The reflective operator links the hard real-time of the controller with the soft real-time of the optimization algorithm in the cognitive operator. It buffers the calculated path from the cognitive operator and feeds reference values to the controller. It also supports possible emergency response in cases like unexpected change of these requirements or changing parameters in the linear drive. Until a new optimization is performed and a new path is available, the current optimization value is replaced by a sub-optimal value from a look-up table. In this case, a new strategy for following selected operating points influenced by changing conditions or parameters is required. This is realized through the application of the decision heuristic to the new Pareto set and the subsequent numerical path following process.

The main advantage of this very fast algorithm is that it prevents the time-consuming calculation of a new Pareto set in every time step and allows to use a multiobjective optimization approach for this complex application.

The results for one of the tested scenarios are described in the following. A time-interval of 6 seconds is considered in which the motor accelerates up to 2 m/s and then brakes down, cf. left plot of Figure 4.7. In the same figure on the right the profile for the desired thrust $F_M^\star$ is plotted over time. While the vehicle accelerates, the thrust is positive whereas it becomes negative when the vehicle brakes. If the absolute value of the thrust is small, then the required transferred power (which in this study has been set to a constant value of ≈ 111 W) cannot be assigned. Thus, no optimization is reasonable in these points. In such a case the Pareto set and the path has to be recomputed at a certain point after this zero.

As mentioned before, the temperature of the secondary motor part and the charging state of the batteries depend on the operating point. Here, it is assumed that both values can be measured during the drive.

Figure 4.8 and Figure 4.9 show the results of the time-dependent multiobjective optimization of the operating point assignment. In Figure 4.8 the Pareto optimal values of the current are plotted over time. Here, the vertical grey lines are the Pareto sets that have been computed at $T = 0.3$ s and at $T = 3.3$ s. One point on these sets has been chosen by use of the decision heuristic taking into account the measured values of the state of charge of the

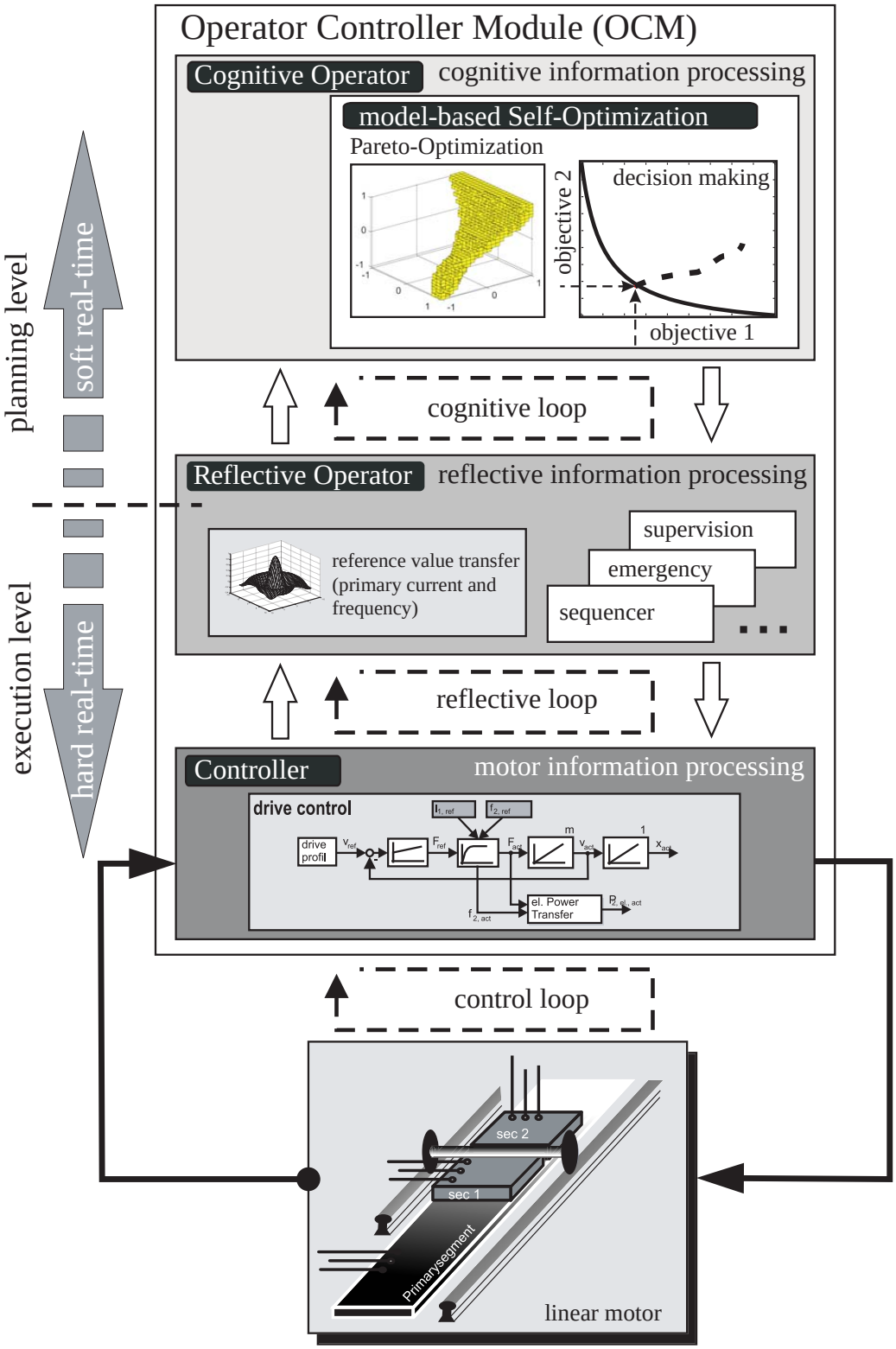

Figure 4.6: Operator-Controller-Module of the operating point assignment of the doubly-fed linear motor [8]

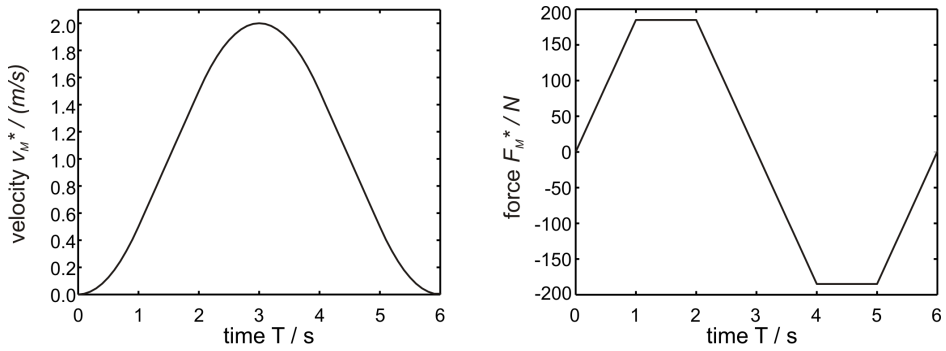

Figure 4.7: Profiles for the velocity and force of the vehicle in the scenario considered

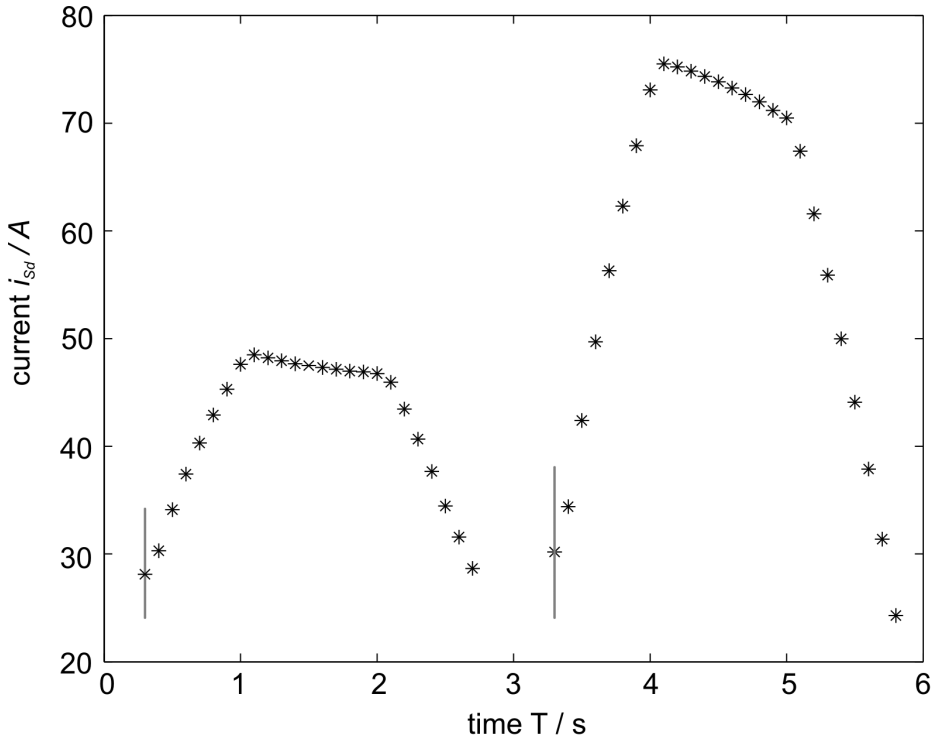

Figure 4.8: Pareto optimal paths for a maneuver of the linear motor

battery and the temperature of the secondary motor part. The black stars are the results of the numerical path following of this solution. The recomputation of the entire Pareto set for $T = 3.3$ s has been necessary because the thrust is near zero in the time interval $(2.7, 3.3)$. Figure 4.9 visualizes the same paths in objective space, i. e. the values of the objective functions are plotted versus time.

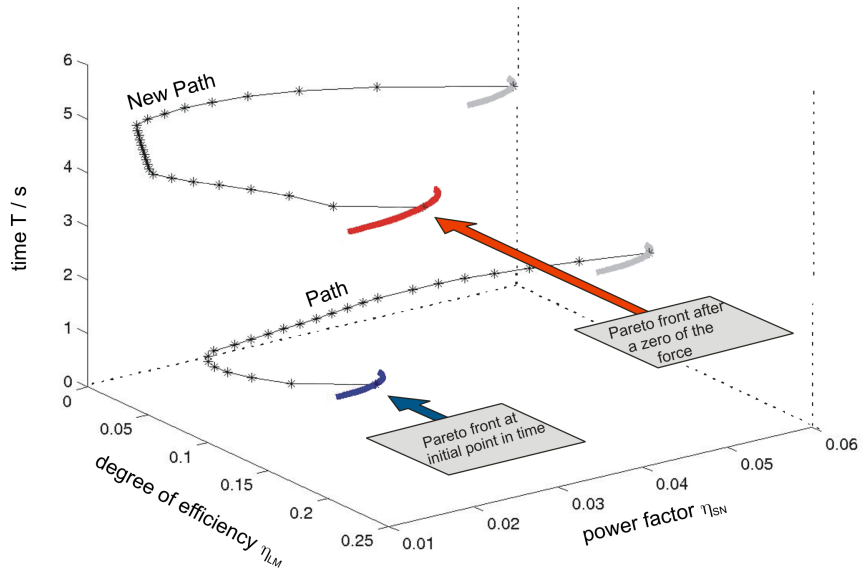

Figure 4.9: Pareto fronts and Pareto optimal paths in objective space for a maneuver of the linear motor

In another study of the doubly-fed linear drive two different objective functions (minimization of total losses, maximization of transferred power) and their changes over time have been considered. For more details, the reader is referred to [86].

4.4 Robust Pareto Points

In this section, parametric multiobjective optimization problems are considered that are desired to be robust with respect to changes of the external parameter. After an overview on existing robustness concepts in multiobjective optimization (not restricted to the parameter-dependent case) two new concepts for robustness will be introduced.

The first one is based on the idea of finding curves of minimum length on the set of substationary points. The mathematical formulation of this task is based on the calculus of variations (cf. Section 2.3).

For the second robustness concept solution paths are computed that stepwise vary as little as possible between the Pareto sets for discretized parameter values. This approach is less computationally costly and very intuitive. Here, again techniques from multiobjective optimization and numerical path

following have been combined. One special challenge in the context of this robustness concept was the derivation of equations that steer a path on the shortest possible, feasible way from one Pareto set to another. Based on these new equations combined with the Kuhn-Tucker equations specific predictor-corrector methods have been applied and related regularity conditions have been proven which are also described in Section 4.4.2.

The last part of Section 4.4.2 is devoted to the comparison of the two new robustness concepts. To make the concepts comparable the variational problem arising in Concept I is reformulated for the case of fixed starting points on a given initial Pareto set. It will turn out that the length of solution paths w. r. t. Concept I is less or equal to the length of paths obtained with Concept II. If a λ-robust Pareto point exists whose corresponding path has length zero, then it is a solution to both robustness concepts.

Finally, in Section 4.4.3 an application in the design of integrated circuits is given. Here, the robustness of Pareto points with respect to changes in temperature and supply voltage has been computed making use of Concept II.

4.4.1 Existing Robustness Definitions in Multiobjective Optimization

In multiobjective optimization, the study of robustness of solutions is a relatively young area of research (cf. [16], [29] for example), especially in evolutionary multiobjective optimization. Different kinds of robustness are studied in the literature mainly motivated by engineering applications. On the one hand uncertainty in the optimization variables can occur. Practically, this can mean that the parameters for a Pareto optimal design of some system cannot be manufactured with the desired accuracy or can only be modeled inaccurately. On the other hand, external (also called environmental) parameters may influence the behavior of the system under consideration. In practice, external parameters could characterize the influence on the system under consideration through changing temperatures, wind velocity, air pressure, etc. To put it in a nutshell, the multiobjective optimization problem can be formulated depending on different kinds of parameters (cf. [10]):

$$\min_{x',x''} F(x', x'' + \varepsilon, \lambda) \tag{4.3}$$

where $x' \in \mathbb{R}^{n_1}$ are deterministic optimization variables, $x'' + \varepsilon$ describes uncertain optimization variables with $x'' \in \mathbb{R}^{n_2}$, $n_1 + n_2 = n$, and $\lambda \in \mathbb{R}^p$ denotes external parameters.

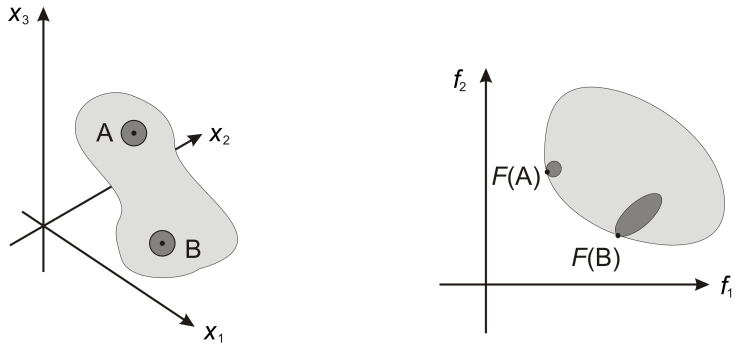

Figure 4.10: Point A is more robust than point B with respect to perturbations of $x = (x_1, x_2, x_3)$ [17]

In [17] two robustness definitions are given. They both treat the case of uncertain optimization parameters only. The first one is based on using mean effective objective functions instead of the original ones. Here, robustness is understood in the sense that perturbations of the optimization parameters lead to small changes in the objective values as illustrated in Figure 4.10.

Definition 4.4 (Multiobjective Robust Solution of Type I, Deb 2006 [17]) *Consider a multiobjective optimization problem*

$$\min_{x \in S} F(x),\ F(x) = (f_1(x), \ldots, f_k(x))\,.$$

A point $x^\star$ is called a multiobjective robust solution of type I *if it is a global, feasible Pareto optimal solution of the multiobjective optimization problem*

$$\min_{x \in S} F_{\text{eff}}(x) \text{ with } F_{\text{eff}}(x) = \left(f_1^{\text{eff}}(x), \ldots, f_k^{\text{eff}}(x)\right)$$
$$\text{and } f_j^{\text{eff}}(x) = \frac{1}{|U_\delta(x)|} \int_{s \in U_\delta(x)} f_j(s)\, ds,$$

where $U_\delta(x) \subseteq S$ is a δ-neighborhood of x and S denotes the feasible region.

In this definition, the neighborhood to be studied in parameter space is set fixed through the parameter δ.

Remark 4.5 The definition of multiobjective robust solution of type I has been inspired by a robustness concept for single objective optimization described in [11] and [103], for example. The main idea of this single objective optimization robustness concept is, that minima which are sharp peaks of

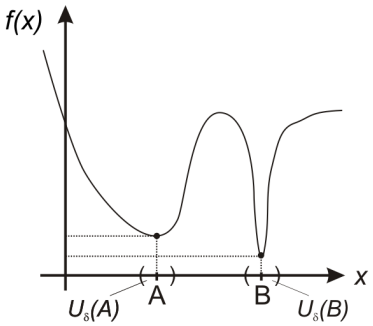

Figure 4.11: Illustration of the fact that a local minimum point (A) can be more robust with respect to perturbations of x than the global minimum point (B) in single objective optimization problems (cf. [17])

the objective functions are not robust (cf. Figure 4.11 for an intuitive understanding). To avoid the computation of these peaks, a mean effective objective function is minimized instead of the original objective. This mean objective function $f^{\text{eff}}(x)$, as given in Definition 4.4, flattens sharp peaks by taking the average values of the original objective function within certain neighborhoods.

The second definition for robustness given in [17] (so-called robust solutions of type II, cf. Definition 4.6) incorporates properties of the solution in objective space. Here, the normalized difference between the objective values of the perturbed function and the original function values is desired to be less than a user-defined value. This inequality is included as a constraint to the original multiobjective optimization problem in order to compute robust solutions of type II.

Definition 4.6 (Multiobjective Robust Solution of Type II, Deb 2006 [17])
Consider a multiobjective optimization problem

$$\min_{x \in S} F(x),\ F(x) = (f_1(x), \ldots, f_k(x))\,.$$

A point $x^\star$ is called a multiobjective robust solution of type II *if it is a global, feasible Pareto optimal solution of the multiobjective optimization problem*

$$\min_{x \in S} F(x)$$
$$\textit{subject to } \frac{\|F^p(x) - F(x)\|}{\|F(x)\|} \leq \eta$$

where the limiting parameter η can be chosen as an arbitrary constant value, $F^p(x)$ denotes the function values of the perturbed point (e. g. $F^p(x) = F_{\text{eff}}(x)$ can be used) and $\|.\|$ is a suitable vector norm.

In [17] possible scenarios of the correspondence between the original Pareto optimal solution (minimization of F) and Pareto optima of the mean effective objective function F_{eff} is discussed. Moreover, the computation of robust solutions with respect to these two robustness types for several test problems has been considered in the same paper.

In [35] different fitness functions for multiobjective evolutionary algorithms are given that lead to robust solutions. The authors distinguish between *expectation measures* and *variance measures*. Expectation measures estimate the value of the objective function assuming a certain probability distribution of disturbances of the optimization parameters within a neighborhood. The main idea of variance measures is to formulate new objectives that describe the mean and the variance of the original objective functions.

In [47] objective functions are studied that may depend on all kinds of parameters mentioned in (4.3). Here, similar to Definition 4.6, a region around the point on the Pareto front to be analyzed is set fixed in objective space. The region in the space of the uncertain and external parameters that leads to these objective values is determined, the so-called *sensitivity region*. Here, the parameters are assumed to vary within known intervals. The larger the sensitity region the more robust is the corresponding Pareto optimal solution. As the sensitivity region may be assymetric, in [47] the *worst case sensitivity region* is introduced which is given by a ball whose radius equals the shortest distance to the boundary of the sensitivity region. Figure 4.12 visualizes the principal idea of this robustness concept. For the computation of robust Pareto sets, in [47] the condition that the radius of the worst case sensitivity region is bigger than a user-defined value is handled as a constraint to the original multiobjective optimization problem. In [65] this concept is extended to the computation of the *objective sensitivity region*. Among others things, in that work the effect of parameter variation within an interval on the objective values is analyzed. Here, the ratio of the objective sensitivity region to the user-defined acceptable objective variation is used for the definition of a robustness measure, the so-called *objective robustness index*.

In [109] robustness with respect to changes in external parameters is studied. It is assumed that the parameters change in a stochastic manner. In this work, probabilistic robustness objectives are defined. In order to obtain

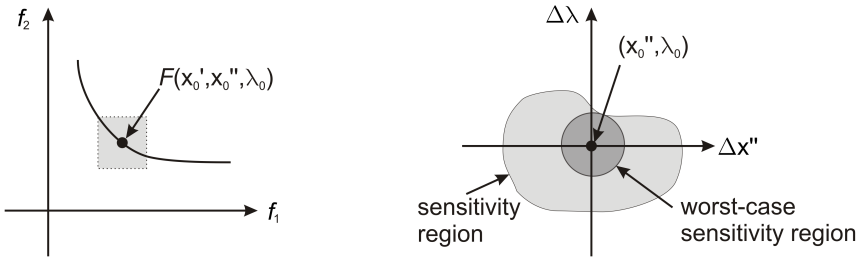

Figure 4.12: Visualization of the sensitivity region and the worst case sensitivity region as defined in [47]

so-called *multiobjective probabilistic acceptable solutions* the original multiobjective optimization problem is reformulated by use of some *violation probability* of each objective. In this case, the violation probability is given by the integral over the product of an indicator function that gives back if the function values are acceptable or not, and the probability density function over the expected external parameter space.

A lot of papers investigate multiobjective optimization problems with fuzzy data. Here, the possible ranges of uncertain parameters are modelled by fuzzy sets (cf. for example [56, 30]). A comparison of fuzzy approaches to stochastic approaches is given in [94].

A different approach is to include robustness directly into the definition of dominance (cf. Definition 3.1). In order to obtain a Pareto set that is robust either with respect to uncertainty of the optimization variables or noisy objective functions different concepts like *probabilistic dominance* [100], *ε-dominance* [91] and many others (e. g. in [55, 28, 99, 84]) have been developed.

4.4.2 New Concepts for Robustness against Variations of an External Parameter

In the following subsections, two new concepts are given which allow the computation of substationary points that are robust with respect to changes of an external parameter in one-parametric multiobjective optimization problems. In contrast to the approaches for robustness that already have been developed in multiobjective optimization (as summarized in the last section), the change of the Pareto sets themselves with respect to different values of the external parameter is studied. Here, the objective functions stay the same as in the original problem and no additional constraints are added.

In Example 4.2 one can observe that the Pareto sets for different values of the parameter λ intersect in some points and differ significantly in other regions. In this section, algorithms and the theoretical background are developed that allow to compute how 'robust' a Pareto point is with respect to changes of the parameter λ within a given interval $[\lambda_{\text{start}}, \lambda_{\text{end}}]$. Loosely speaking, the more robust a Pareto point for a fixed external parameter value is the less it changes to other Pareto points for different values of the external parameter within a given interval. The same considerations can be transferred to the study of robustness on Pareto fronts. In both cases, the techniques used to obtain robust solutions are completely different from the ones used in the approaches described above. The first concept, which we have published in [107], is based on the calculus of variations whereas in the second concept numerical path following methods are employed for a certain system of equations (see also [21]).

Concept I: Robustness formulated via the classical calculus of variations

The classical calculus of variations allows, in the case of at least twice continuously differentiable objective functions, the computation of a substationary point that varies as little as possible with respect to changes of the external parameter λ. To be more precise, the aim is to determine a continuously differentiable curve $\gamma(\lambda) = (x(\lambda), t(\lambda))^T$ of minimal length from an arbitrary starting point on the set of substationary points $S_{\lambda_{\text{start}}}$ to an arbitrary end point on $S_{\lambda_{\text{end}}}$ that lies within the λ-dependent set of substationary points. This problem can be formulated as a variational problem by

$$\begin{aligned} &\min_{\gamma} \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} \|x'(\lambda)\|_2 \, \mathrm{d}\lambda \\ &\text{s. t. } H_{\text{KT}}(\gamma(\lambda), \lambda) = 0 \end{aligned} \tag{4.4}$$

with the boundary conditions $x(\lambda_{\text{start}}) \in S_{\lambda_{\text{start}}}$ and $x(\lambda_{\text{end}}) \in S_{\lambda_{\text{end}}}$. Here, the functional in (4.4) corresponds to the length of the curve $x(\lambda)$ that has to be minimized, and $H_{\text{KT}}(\gamma(\lambda), \lambda)$ describes the Kuhn-Tucker equations as given in Equation 4.2. Note, that due to the equality constraint $H_{KT}\left(x(\lambda), t(\lambda), \lambda\right) = 0$ the path is enforced to lie on the set of substationary points, i. e. $x(\lambda) \in S_\lambda \; \forall \lambda \in [\lambda_{\text{start}}, \lambda_{\text{end}}]$.

For the following considerations it is assumed that the curve $x(\lambda)$ is regular, i. e. $\|x'(\lambda)\| \neq 0$ for all $\lambda \in [\lambda_{\text{start}}, \lambda_{\text{end}}]$.

Definition 4.7 (Robustness w. r. t. Concept I)
Let $\gamma^\star(\lambda) = (x^\star(\lambda), t^\star(\lambda))^T$ *be a solution of Problem (4.4) with the free boundary conditions as stated above. Then* $x^\star(\lambda)$ *is called a* λ-robust solution path of Concept I with respect to the interval $[\lambda_{\text{start}}, \lambda_{\text{end}}]$. *The point* $x^\star(\lambda_{\text{start}})$ *is called a* λ-robust substationary point of Concept I w. r. t. the interval $[\lambda_{\text{start}}, \lambda_{\text{end}}]$. *If the objective functions are convex and* $t^\star(\lambda_{\text{start}}) \neq 0$, *i. e. the substationary point* $x^\star(\lambda_{\text{start}})$ *is indeed a Pareto point, then* $x^\star(\lambda_{\text{start}})$ *is called a* λ-robust Pareto point of Concept I with respect to the interval $[\lambda_{\text{start}}, \lambda_{\text{end}}]$.

Remark 4.8 The intuition behind Definition 4.7 is to study the influence on a Pareto set when increasing the value of the external parameter. Instead, using the same methodology as described in the following, one could consider variations of the external parameter λ in both directions. Starting with the Pareto set for a fixed value $\lambda = \lambda^\star$ one would consider the change of $\lambda \in [\lambda^\star - \varepsilon, \lambda^\star + \varepsilon]$ for a fixed value $\varepsilon \in \mathbb{R}$. In this case, λ-robust substationary points with respect to the interval $[\lambda^\star - \varepsilon, \lambda^\star + \varepsilon]$ would be defined as solutions of the variational problem

$$\begin{aligned} &\min_{\gamma} \int_{\lambda^\star - \varepsilon}^{\lambda^\star + \varepsilon} \|x'(\lambda)\|_2 \, \mathrm{d}\lambda \\ &\text{s. t. } H_{KT}\left(x(\lambda), t(\lambda), \lambda\right) = 0. \end{aligned}$$

The multiplier rule (Theorem 2.11) states that a necessary condition for optimiality of this problem is given by the Euler-Lagrange equations for the functional

$$\mathcal{L}(\gamma(\lambda), \gamma'(\lambda), \mu(\lambda), \lambda) = \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} \|x'(\lambda)\|_2 + (\mu(\lambda))^T \cdot H_{\text{KT}}(\gamma(\lambda), \lambda) \mathrm{d}\lambda, \quad (4.5)$$

where $\mu : [\lambda_{\text{start}}, \lambda_{\text{end}}] \to \mathbb{R}^{n+1}$, $\mu(\lambda) = (\mu_1(\lambda), \cdots, \mu_{n+1}(\lambda))^T$ (cf. Chapter 2, Section 2.3). Defining

$$L(\gamma(\lambda), \gamma'(\lambda), \mu(\lambda), \lambda) = \|x'(\lambda)\|_2 + (\mu(\lambda))^T \cdot H_{\text{KT}}(\gamma(\lambda), \lambda), \quad (4.6)$$

the Euler-Lagrange equations for (4.4) are given by

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\lambda}\left(\frac{\partial L}{\partial \gamma'}\right) - \left(\frac{\partial L}{\partial \gamma}\right) &= 0 \quad (4.7) \\ H_{\text{KT}}(\gamma(\lambda), \lambda) &= 0 \quad (4.8) \end{aligned}$$

(the Kuhn-Tucker equations (4.8) stem from the derivative of L with respect to μ). The single terms of this equation are given by

$$\begin{aligned}
\left(\frac{\partial L}{\partial \gamma'}\right) &= \frac{\partial}{\partial \gamma'}\left(\|x'(\lambda)\|_2 + \mu(\lambda)^T \cdot H_{\mathrm{KT}}(\gamma(\lambda),\lambda)\right) \\
&= \frac{1}{\|x'(\lambda)\|_2} \cdot (x_1'(\lambda), \cdots, x_n'(\lambda), 0) \\
\frac{\mathrm{d}}{\mathrm{d}\lambda}\left(\frac{\partial L}{\partial \gamma'}\right) &= \begin{pmatrix} \frac{x_1''(\lambda)}{\|x'(\lambda)\|_2} - \frac{x_1'(\lambda)}{\|x'(\lambda)\|_2^3} \cdot \left(\sum_{i=1}^n x_i'(\lambda) \cdot x_i''(\lambda)\right) \\ \vdots \\ \frac{x_n''(\lambda)}{\|x'(\lambda)\|_2} - \frac{x_n'(\lambda)}{\|x'(\lambda)\|_2^3} \cdot \left(\sum_{i=1}^n x_i'(\lambda) \cdot x_i''(\lambda)\right) \end{pmatrix}^T
\end{aligned}$$

and

$$\frac{\partial L}{\partial \gamma} = \begin{pmatrix} (\mu_1(\lambda), \cdots, \mu_{n+1}(\lambda)) \cdot \begin{pmatrix} \frac{\partial}{\partial x_1} H_1(x(\lambda), t(\lambda), \lambda) \\ \vdots \\ \frac{\partial}{\partial x_1} H_n(x(\lambda), t(\lambda), \lambda) \\ 0 \end{pmatrix} \\ \vdots \\ (\mu_1(\lambda), \cdots, \mu_{n+1}(\lambda)) \cdot \begin{pmatrix} \frac{\partial}{\partial x_n} H_1(x(\lambda), t(\lambda), \lambda) \\ \vdots \\ \frac{\partial}{\partial x_n} H_n(x(\lambda), t(\lambda), \lambda) \\ 0 \end{pmatrix} \\ (\mu_1(\lambda), \cdots, \mu_{n+1}(\lambda)) \cdot \begin{pmatrix} \frac{\partial}{\partial t_1} H_1(x(\lambda), t(\lambda), \lambda) \\ \vdots \\ \frac{\partial}{\partial t_1} H_{n+1}(x(\lambda), t(\lambda), \lambda) \end{pmatrix} \\ \vdots \\ (\mu_1(\lambda), \cdots, \mu_{n+1}(\lambda)) \cdot \begin{pmatrix} \frac{\partial}{\partial t_k} H_1(x(\lambda), t(\lambda), \lambda) \\ \vdots \\ \frac{\partial}{\partial t_k} H_{n+1}(x(\lambda), t(\lambda), \lambda) \end{pmatrix} \end{pmatrix}^T.$$

Altogether, this leads to the Euler-Lagrange equations

$$\frac{x''(\lambda)}{\|x'(\lambda)\|_2} - \frac{x'(\lambda)}{\|x'(\lambda)\|_2^3}\left(\sum_{i=1}^n x_i'(\lambda) \cdot x_i''(\lambda)\right) - \left(\mu(\lambda)^T \cdot \frac{\partial}{\partial x}\begin{pmatrix} H_{\mathrm{KT}}^1 \\ \vdots \\ H_{\mathrm{KT}}^n \end{pmatrix}(\gamma(\lambda),\lambda)\right)^T = 0 \tag{4.9}$$

$$\left(-\mu(\lambda)^T \cdot \frac{\partial}{\partial t}\begin{pmatrix} H_{\mathrm{KT}}^1 \\ \vdots \\ H_{\mathrm{KT}}^{n+1} \end{pmatrix}(\gamma(\lambda),\lambda)\right)^T = 0 \tag{4.10}$$

$$\begin{pmatrix} H_{\mathrm{KT}}^1 \\ \vdots \\ H_{\mathrm{KT}}^n \\ H_{\mathrm{KT}}^{n+1} \end{pmatrix}(\gamma(\lambda),\lambda) = \begin{pmatrix} \sum_{i=1}^k (t_i(\lambda))^2 \nabla_x f_i(x,\lambda) \\ \sum_{i=1}^k (t_i(\lambda))^2 - 1 \end{pmatrix} = 0. \tag{4.11}$$

The free boundary conditions (cf. Section 2.3) lead to the additional conditions

$$\begin{aligned} \frac{\partial L}{\partial x'}\Big|_{\lambda=\lambda_{\text{start}}} &\perp \ker\left(\frac{\partial}{\partial x} H_{\text{KT}}(\gamma(\lambda_{\text{start}}), \lambda_{\text{start}})\right) \\ \Leftrightarrow \frac{1}{\|x'(\lambda_{\text{start}})\|_2} \cdot x'(\lambda_{\text{start}}) &\perp \ker\left(\frac{\partial}{\partial x} H_{\text{KT}}(\gamma(\lambda_{\text{start}}), \lambda_{\text{start}})\right) \end{aligned} \tag{4.12}$$

and

$$\begin{aligned} \frac{\partial L}{\partial x'}\Big|_{\lambda=\lambda_{\text{end}}} &\perp \ker\left(\frac{\partial}{\partial x} H_{\text{KT}}(\gamma(\lambda_{\text{end}}), \lambda_{\text{end}})\right) \\ \Leftrightarrow \frac{1}{\|x'(\lambda_{\text{end}})\|_2} \cdot x'(\lambda_{\text{end}}) &\perp \ker\left(\frac{\partial}{\partial x} H_{\text{KT}}(\gamma(\lambda_{\text{end}}), \lambda_{\text{end}})\right). \end{aligned} \tag{4.13}$$

In order to compute a λ-robust solution numerically, the variational problem is discretized following the idea of the construction of variational integrators [68]. Rather than discretizing the system of equations which consists of the Euler-Lagrange equations and optionally additional constraints, the variational integral is discretized directly and a discrete version of the calculus of variations is applied to derive the set of equations known as the *discrete Euler-Lagrange equations* and discretized constraints. In the following, we consider a discretization of the problem

$$\min_{\gamma,\mu} \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} L\left(\gamma(\lambda), \gamma'(\lambda), \mu(\lambda), \lambda\right) \, \mathrm{d}\lambda \tag{4.14}$$

with free boundary conditions that satisfy the constraints

$$\varphi(x(\lambda_{\text{start}}), \lambda_{\text{start}}) = \varphi(x(\lambda_{\text{end}}), \lambda_{\text{end}}) = 0.$$

Note that, as before, L is the Lagrangian given by

$$L\left(\gamma(\lambda), \gamma'(\lambda), \mu(\lambda), \lambda\right) = C\left(\gamma(\lambda), \gamma'(\lambda), \lambda\right) + \mu(\lambda)^T \cdot \varphi\left(\gamma(\lambda), \lambda\right).$$

To this aim the parameter λ is assumed to grow with a fixed step size $h \in \mathbb{R}$ and we define

$$\lambda_d = \{\lambda_{\text{start}} + jh\}_{j=0}^{N}$$

with $\lambda_N = \lambda_{\text{end}}$. The hitherto continuous curves $\gamma(\lambda)$ and $\mu(\lambda)$ are now replaced by discrete curves

$$\begin{aligned} &\gamma_d : \lambda_d \to \mathbb{R}^n, \;\; \gamma_d = \{\gamma_j\}_{j=0}^{N}, \text{ where } \gamma_j = \gamma_d(\lambda_j), \\ &\mu_d : \lambda_d \to \mathbb{R}^m, \; \mu_d = \{\mu_j\}_{j=0}^{N}, \text{ where } \mu_j = \mu_d(\lambda_j). \end{aligned}$$

Based on these discrete objects, the integral of the Lagrangian along a short interval is approximated by a discrete Lagrangian L_d as

$$\int_{\lambda_j}^{\lambda_{j+1}} L(\gamma(\lambda), \gamma'(\lambda), \mu(\lambda), \lambda)\, d\lambda \approx L_d(\gamma_j, \mu_j, \lambda_j, \gamma_{j+1}, \mu_{j+1}, \lambda_{j+1}) \tag{4.15}$$

which depends on two neighboring points (γ_j, μ_j) and $(\gamma_{j+1}, \mu_{j+1})$ and on two external parameter values λ_j and λ_{j+1}. Finally, the discrete variational integral is defined as

$$\mathcal{L}_d(\gamma_d, \mu_d) = \sum_{j=0}^{N-1} L_d(\gamma_j, \mu_j, \lambda_j, \gamma_{j+1}, \mu_{j+1}, \lambda_{j+1}).$$

Figure 4.13 visualizes the approximation of the integral by this finite sum.

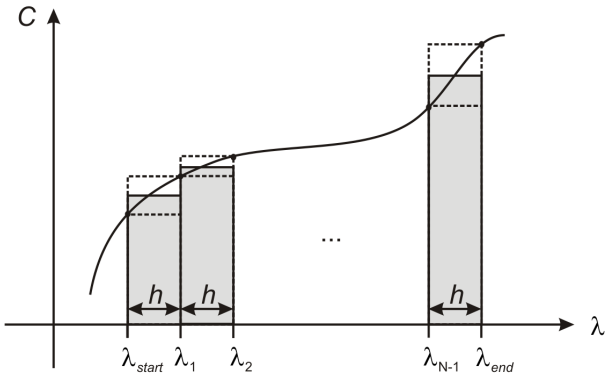

Figure 4.13: Discretization of an integral by a finite sum as used in (4.16)

Different quadrature rules can be chosen for the approximation in (4.15) (cf. [68]). For a second order approximation, typically the difference quotient

$$\frac{\gamma_{j+1} - \gamma_j}{h}$$

is used for the approximation of derivatives of γ at each discretization point. Furthermore, following [68, 64], a midpoint evaluation for the function C and a trapezoidal rule for the constraints lead to the discrete variational integral

$$\begin{aligned} \mathcal{L}_d(\gamma_d, \mu_d) &= \sum_{j=0}^{N-1} L_d(\gamma_j, \mu_j, \lambda_j, \gamma_{j+1}, \mu_{j+1}, \lambda_{j+1}) \\ &= \sum_{j=0}^{N-1} h \cdot \left(C\left(\frac{\gamma_j + \gamma_{j+1}}{2}, \frac{\gamma_{j+1} - \gamma_j}{h}, \frac{\lambda_j + \lambda_{j+1}}{2} \right) + \right. \\ &\qquad \left. \frac{1}{2}\left(\mu_j^T \cdot \varphi(\gamma_j, \lambda_j)\right) + \frac{1}{2}\left(\mu_{j+1}^T \cdot \varphi(\gamma_{j+1}, \lambda_{j+1})\right) \right). \end{aligned} \tag{4.16}$$

In the following, the first index corresponds to the dimension and the second index corresponds to the discretization point on the respective discretized curves. The points on the discretized curves are given as

$$\gamma_j = \begin{pmatrix} \gamma_{1,j} \\ \vdots \\ \gamma_{n,j} \end{pmatrix}, \quad \mu_j = \begin{pmatrix} \mu_{1,j} \\ \vdots \\ \mu_{m,j} \end{pmatrix}$$

for $j = 0, \ldots, N$.
Let

$$\eta_d : \lambda_d \to \mathbb{R}^{n+m}, \ \eta_d = \{\eta_j\}_{j=0}^N, \text{ with } \eta_j = \eta_d(\lambda_j),$$

be the discretization of a twice continuously differentiable curve.
Assume that

$$z_d = \begin{pmatrix} \gamma_d \\ \mu_d \end{pmatrix} \text{ with } \ z_j = z_d(\lambda_j), \ j = 0, \ldots, N$$

is the discretization of a minimizing curve of the variational problem (4.14). In analogue to the continuous case, $\delta z_d = \eta_d$ is called a variation of z_d. More detailed, we have

$$\eta_j = \begin{pmatrix} \eta_{1,j} \\ \vdots \\ \eta_{n+m,j} \end{pmatrix} = \begin{pmatrix} \delta\gamma_j \\ \delta\mu_j \end{pmatrix} = \begin{pmatrix} \delta\gamma_{1,j} \\ \vdots \\ \delta\gamma_{n,j} \\ \delta\mu_{1,j} \\ \vdots \\ \delta\mu_{m,j} \end{pmatrix}$$

for $j = 0, \ldots, N$.

Let $\varepsilon \in \mathbb{R}$. Similar to the derivation of the Euler-Lagrange equations for the continuous case [41], the first variation δ of $\mathcal{L}_d$ is defined as $v'(0)$, where $v : \mathbb{R} \to \mathbb{R}$,

$$v(\varepsilon) = \sum_{j=0}^{N-1} L_d \left(z_j + \varepsilon\eta_j, \lambda_j, z_{j+1} + \varepsilon\eta_{j+1}, \lambda_{j+1} \right).$$

The vanishing of the first variation of $\mathcal{L}_d(\gamma_d, \mu_d)$ leads to the following computations:

$$\delta \left(\sum_{j=0}^{N-1} h \cdot \Big(C \left(\frac{\gamma_j + \gamma_{j+1}}{2}, \frac{\gamma_{j+1} - \gamma_j}{h}, \frac{\lambda_j + \lambda_{j+1}}{2} \right) + \right.$$
$$\left. \frac{1}{2} \left(\mu_j^T \varphi(\gamma_j, \lambda_j) \right) + \frac{1}{2} \left(\mu_{j+1}^T \varphi(x_{j+1}, \lambda_{j+1}) \right) \Big) \right) = 0.$$

It follows that

$$
\begin{aligned}
&\sum_{j=0}^{N-1} h \cdot \Bigg(\frac{\partial}{\partial \gamma_j} \left(C\left(\frac{\gamma_j + \gamma_{j+1}}{2}, \frac{\gamma_{j+1} - \gamma_j}{h}, \frac{\lambda_j + \lambda_{j+1}}{2} \right) + \frac{1}{2} \mu_j^T \cdot \varphi(\gamma_j, \lambda_j) \right) \delta\gamma_j + \\
&\qquad \frac{\partial}{\partial \gamma_{j+1}} \left(C\left(\frac{\gamma_j + \gamma_{j+1}}{2}, \frac{\gamma_{j+1} - \gamma_j}{h}, \frac{\lambda_j + \lambda_{j+1}}{2} \right) + \right. \\
&\qquad\qquad \left. \frac{1}{2} \mu_{j+1}^T \cdot \varphi(\gamma_{j+1}, \lambda_{j+1}) \right) \delta\gamma_{j+1} + \\
&\qquad \frac{\partial}{\partial \mu_j} \left(\frac{1}{2} \mu_j^T \cdot \varphi(\gamma_j, \lambda_j) \right) \delta\mu_j + \frac{\partial}{\partial \mu_{j+1}} \left(\mu_{j+1}^T \cdot \varphi(\gamma_{j+1}, \lambda_{j+1}) \right) \delta\mu_{j+1} \Bigg) = 0 \\
\Leftrightarrow\ & h \cdot \Bigg(\frac{\partial}{\partial \gamma_0} \left(C\left(\frac{\gamma_0 + \gamma_1}{2}, \frac{\gamma_1 - \gamma_0}{h}, \frac{\lambda_0 + \lambda_1}{2} \right) + \frac{1}{2} \mu_0 \cdot \varphi(\gamma_0, \lambda_{\text{start}}) \right) \delta\gamma_0 + \\
&\quad \frac{\partial}{\partial \gamma_N} \left(C\left(\frac{\gamma_{N-1} + \gamma_N}{2}, \frac{\gamma_N - \gamma_{N-1}}{h}, \frac{\lambda_{N-1} + \lambda_N}{2} \right) + \frac{1}{2} \mu_N^T \cdot \varphi(\gamma_N, \lambda_{\text{end}}) \right) \delta\gamma_N + \\
&\quad \sum_{j=1}^{N-1} \frac{\partial}{\partial \gamma_j} \Big(C\left(\frac{\gamma_j + \gamma_{j+1}}{2}, \frac{\gamma_{j+1} - \gamma_j}{h}, \frac{\lambda_j + \lambda_{j+1}}{2} \right) + \\
&\quad C\left(\frac{\gamma_{j-1} + \gamma_j}{2}, \frac{\gamma_j - \gamma_{j-1}}{h}, \frac{\lambda_{j-1} + \lambda_j}{2} \right) + \mu_j^T \cdot \varphi(\gamma_j, \lambda_j) \Big) \delta\gamma_j + \\
&\quad \frac{\partial}{\partial \mu_0} \left(\frac{1}{2} \mu_0^T \cdot \varphi(\gamma_0, \lambda_{\text{start}}) \right) \delta\mu_0 + \\
&\quad \frac{\partial}{\partial \mu_N} \left(\frac{1}{2} \mu_N^T \cdot \varphi(\gamma_N, \lambda_{\text{end}}) \right) \delta\mu_N + \\
&\quad \sum_{j=1}^{N-1} \frac{\partial}{\partial \mu_j} \left(\mu_j^T \cdot \varphi(\gamma_j, \lambda_j) \right) \delta\mu_j \Bigg) = 0.
\end{aligned}
$$

As the variations $\delta\gamma_j$ and $\delta\mu_j$ are arbitrary for $j = 0, \ldots, N$ each term multiplied with them has to equal zero. It follows that for all $j = 1, \ldots, N-1$

$$
\begin{aligned}
h \cdot \frac{\partial}{\partial \gamma_j} \Bigg(C\left(\frac{\gamma_j + \gamma_{j+1}}{2}, \frac{\gamma_{j+1} - \gamma_j}{h}, \frac{\lambda_j + \lambda_{j+1}}{2} \right) + & \\
C\left(\frac{\gamma_{j-1} + \gamma_j}{2}, \frac{\gamma_j - \gamma_{j-1}}{h}, \frac{\lambda_{j-1} + \lambda_j}{2} \right) + \mu_j^T \varphi(\gamma_j, \lambda_j) \Bigg) & = 0 \\
\varphi(\gamma_j, \lambda_j) & = 0.
\end{aligned}
$$

At the free ends, the variations $\delta\gamma_0$, $\delta\gamma_N$, $\delta\mu_0$ and $\delta\mu_N$ can be arbitrarily chosen as well and thus it turns out that the following terms have to vanish:

$$
\begin{aligned}
h \cdot \frac{\partial}{\partial \gamma_0} \left(C \left(\frac{\gamma_0 + \gamma_1}{2}, \frac{\gamma_1 - \gamma_0}{h}, \frac{\lambda_0 + \lambda_1}{2} \right) \right) + \frac{1}{2} h \cdot \left(\mu_0^T \frac{\partial}{\partial \gamma_0} \varphi(\gamma_0, \lambda_{\text{start}}) \right)^T &= 0 \\
\varphi(\gamma_0, \lambda_{\text{start}}) &= 0 \\
h \cdot \frac{\partial}{\partial \gamma_N} \left(C \left(\frac{\gamma_{N-1} + \gamma_N}{2}, \frac{\gamma_N - \gamma_{N-1}}{h}, \frac{\lambda_{N-1} + \lambda_N}{2} \right) \right) + \frac{1}{2} h \cdot \left(\mu_N^T \frac{\partial}{\partial \gamma_N} \varphi(\gamma_N, \lambda_{\text{end}}) \right)^T &= 0 \\
\varphi(\gamma_N, \lambda_{\text{end}}) &= 0.
\end{aligned}
$$

To put it in a nutshell, one ends up with the system of equations

$$
\begin{aligned}
&\forall\, j = 0, \ldots, N: \\
&\qquad \varphi(\gamma_j, \lambda_j) = 0 \\
&\forall\, j = 1, \ldots, N-1: \\
&\qquad h \cdot \frac{\partial}{\partial \gamma_j} \left(C \left(\frac{\gamma_j + \gamma_{j+1}}{2}, \frac{\gamma_{j+1} - \gamma_j}{h}, \frac{\lambda_j + \lambda_{j+1}}{2} \right) + \right. \\
&\qquad\qquad \left. C \left(\frac{\gamma_{j-1} + \gamma_j}{2}, \frac{\gamma_j - \gamma_{j-1}}{h}, \frac{\lambda_{j-1} + \lambda_j}{2} \right) + \mu_j^T \varphi(\gamma_j, \lambda_j) \right) = 0 \\
&\qquad h \cdot \frac{\partial}{\partial \gamma_0} \left(C \left(\frac{\gamma_0 + \gamma_1}{2}, \frac{\gamma_1 - \gamma_0}{h}, \frac{\lambda_0 + \lambda_1}{2} \right) \right) + \frac{1}{2} h \cdot \left(\mu_0^T \frac{\partial}{\partial \gamma_0} \varphi(\gamma_0, \lambda_{\text{start}}) \right)^T = 0 \\
&\qquad h \cdot \frac{\partial}{\partial \gamma_N} \left(C \left(\frac{\gamma_{N-1} + \gamma_N}{2}, \frac{\gamma_N - \gamma_{N-1}}{h}, \frac{\lambda_{N-1} + \lambda_N}{2} \right) \right) + \frac{1}{2} h \cdot \left(\mu_N^T \frac{\partial}{\partial \gamma_N} \varphi(\gamma_N, \lambda_{\text{end}}) \right)^T = 0 \\
&\qquad \varphi(\gamma_N, \lambda_{\text{end}}) = 0.
\end{aligned}
$$

The solution of this system of equations provides a numerical approximation of the solution of the boundary value problem given by the Euler-Lagrange equations (2.8)-(2.9) and the transversality condition (2.11) (see pages 23/24). This can in particular be seen if we consider the special case of computing λ-robust substationary points w. r. t. Concept I, where the functions

$$
C\left(\gamma(\lambda), \gamma'(\lambda), \lambda\right) = \|x'(\lambda)\|_2 \quad \text{and} \quad \varphi(\gamma(\lambda), \lambda) = H_{\text{KT}}(\gamma(\lambda), \lambda)
$$

are considered (remember that $\gamma(\lambda) = (x(\lambda), t(\lambda))^T$). For this special case, one ends up with the system of equations

$$
\begin{aligned}
&\forall\, j = 0, \ldots, N: \\
&\qquad \left(\mu_j^T \cdot \frac{\partial}{\partial t_j} H_{\text{KT}}(x_j, t_j, \lambda_j) \right)^T = 0 \\
&\forall\, j = 0, \ldots, N: \\
&\qquad H_{\text{KT}}(x_j, t_j, \lambda_j) = 0 \\
&\forall\, j = 1, \ldots, N-1: \\
&\frac{1}{h} \cdot \left(\frac{x_{j+1} - x_j}{\|x_{j+1} - x_j\|} - \frac{x_j - x_{j-1}}{\|x_j - x_{j-1}\|} \right) - \left(\mu_j^T \cdot \frac{\partial}{\partial x_j} H_{\text{KT}}(x_j, t_j, \lambda_j) \right)^T = 0 \\
&\qquad \frac{1}{h} \cdot \frac{x_1 - x_0}{\|x_1 - x_0\|} - \frac{1}{2} \left(\mu_0^T \cdot \frac{\partial}{\partial x_0} H_{\text{KT}}(x_0, t_0, \lambda_{\text{start}}) \right)^T = 0 \\
&\qquad \frac{1}{h} \cdot \frac{x_N - x_{N-1}}{\|x_N - x_{N-1}\|} + \frac{1}{2} \left(\mu_N^T \cdot \frac{\partial}{\partial x_N} H_{\text{KT}}(x_N, t_N, \lambda_{\text{end}}) \right)^T = 0.
\end{aligned}
$$

The solution of this system of equations approximates the solution of the Euler-Lagrange equations (4.9)-(4.13). This is obvious for the first two equations. To understand why the first part of the third equation also gives an approximation of the first two terms in equation (4.9) one has to remember that the term stems from the λ-derivative of $\frac{x'(\lambda)}{\|x'(\lambda)\|_2}$. In the third equation, this derivative is approximated by a difference quotient. The last two equations approximate the transversality conditions (4.12) and (4.13) since for a matrix A it holds $(\ker(A))^\perp = \operatorname{im}(A^T)$).

Remark 4.9 By solving this system of equations only one (local) solution will be obtained. Which solution is found, highly depends on the initial values of the parameters. If the same variational problem is formulated with fixed starting point $(x_0, \lambda_{\text{start}})$ on the initial Pareto set, the function value of the variational integral $\int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} \|x'(\lambda)\|_2 \, \mathrm{d}\lambda$ can be seen as a measure for robustness of the starting point $(x_0, \lambda_{\text{start}})$.

Example 4.10 Consider the parametric multiobjective optimization problem $\min_x F : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2,\ F = (f_1, f_2)$,

$$\begin{aligned} f_1(x, \lambda) &= (x_1 - \sin(\lambda))^2 + (x_2 - \cos(\lambda))^2 \\ f_2(x, \lambda) &= (x_1 - 2\sin(\lambda))^2 + (x_2 - 2\cos(\lambda))^2 \end{aligned}$$

in which the parameter λ is varied from 0 to 2π.

The Pareto set of these objectives is given by a straight line of length one that λ-dependently circles around the origin with a distance of one to two. The discretization of the Euler-Lagrange equations as described above and the solution of the resulting nonlinear system of equations with the MATLAB routine *fsolve* (cf. [101]) leads to the results given in Figures 4.14 and 4.15. Here, $N = 32$ discretization points have been used. As one could expect, the solution curve of smallest length on the set of substationary points lies on the 'inner circle'.

Hitherto it has been assumed that $\|x'(\lambda)\| \neq 0$ holds. This assumption can be cancelled if one considers the minimization of *energy* instead of *length* of the curve $x(\lambda)$. Here, energy is defined by the variational integral

$$\int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} \frac{1}{2} \|x'(\lambda)\|_2^2 \, \mathrm{d}\lambda. \tag{4.17}$$

As the Kuhn-Tucker constraints of the variational problem (4.4) depend on λ explicitly, the minimization of energy instead of length does not lead to the same optimal solution (cf. [74]).

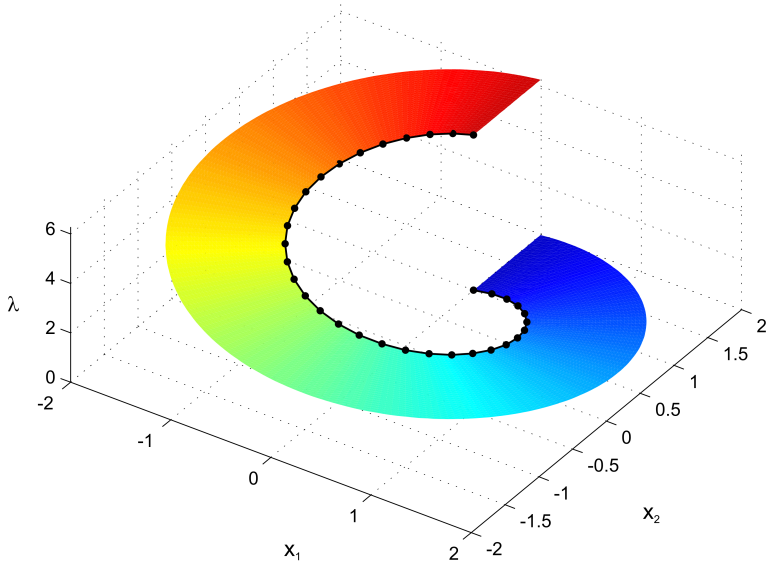

Figure 4.14: Visualization of the λ-robust solution path (black) and λ-dependent Pareto set

However, if a solution curve of length zero exists, i. e. there exists a point in which the set of substationary points does not change in the x-space under the variation of λ, this solution will be both obtained by minimizing length and energy. Thus, if a λ-robust substationary point that stays constant under the variation of λ in x-space is to be computed, the following variational problem can be considered instead of (4.4):

$$\begin{aligned} \min_{\gamma} \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} \frac{1}{2} \|x'(\lambda)\|_2^2 \, \mathrm{d}\lambda \\ \text{s. t. } H_{\text{KT}}(\gamma(\lambda), \lambda) = 0 \end{aligned} \tag{4.18}$$

with the boundary conditions $x(\lambda_{\text{start}}) \in S_{\lambda_{\text{start}}}$ and $x(\lambda_{\text{end}}) \in S_{\lambda_{\text{end}}}$.
In analogue to the computations for (4.4) this leads to the Euler-Lagrange equations

$$\begin{pmatrix} x''(\lambda) \\ 0 \end{pmatrix} - \left(\mu(\lambda)^T \cdot \frac{\partial}{\partial(x,t)} \begin{pmatrix} H_{\text{KT}}^1 \\ \vdots \\ H_{\text{KT}}^{n+1} \end{pmatrix} (x(\lambda), t(\lambda), \lambda) \right)^T = 0 \tag{4.19}$$

$$\begin{pmatrix} H_{\text{KT}}^1 \\ \vdots \\ H_{\text{KT}}^{n} \\ H_{\text{KT}}^{n+1} \end{pmatrix} (x(\lambda), t(\lambda), \lambda) = \begin{pmatrix} \sum_{i=1}^k (t_i(\lambda))^2 \nabla_x f_i(x, \lambda) \\ \sum_{i=1}^k (t_i(\lambda))^2 - 1 \end{pmatrix} = 0 \tag{4.20}$$

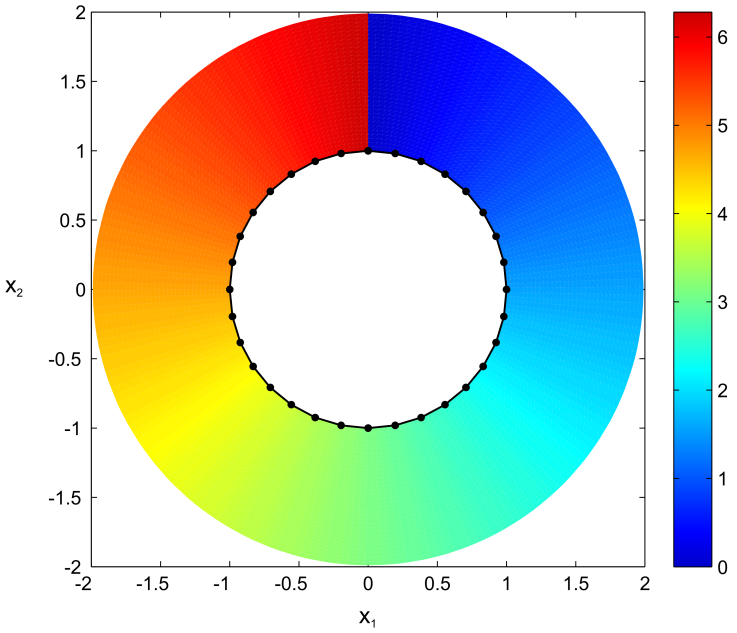

Figure 4.15: Projection of the solution visualized in Figure 4.14 onto the x_1-x_2-plane

with boundary conditions

$$x'(\lambda_{\text{start}}) \quad \perp \quad \ker\left(\frac{\partial}{\partial x} H_{\text{KT}}(\gamma(\lambda_{\text{start}}), \lambda_{\text{start}})\right) \tag{4.21}$$

$$x'(\lambda_{\text{end}}) \quad \perp \quad \ker\left(\frac{\partial}{\partial x} H_{\text{KT}}(\gamma(\lambda_{\text{end}}), \lambda_{\text{end}})\right), \tag{4.22}$$

and the discretized Euler-Lagrange equations

$$\begin{aligned}
H_{\text{KT}}(x_j, t_j, \lambda_j) &= 0 \ \forall\, j = 0, \ldots, N \\
\left(\mu_j^T \cdot \frac{\partial}{\partial t_j} H_{\text{KT}}(x_j, t_j, \lambda_j)\right)^T &= 0 \ \forall\, j = 0, \ldots, N \\
\left(\frac{x_{j+1} - 2x_j + x_{j-1}}{h^2}\right) - \left(\mu_j^T \cdot \frac{\partial}{\partial x_j} H_{\text{KT}}(x_j, t_j, \lambda_j)\right)^T &= 0 \ \forall\, j = 1, \ldots, N-1 \\
\left(\frac{x_1 - x_0}{h^2}\right) - \frac{1}{2}\left(\mu_0^T \cdot \frac{\partial}{\partial x_0} H_{\text{KT}}(x_0, t_0, \lambda_{\text{start}})\right)^T &= 0 \\
-\left(\frac{x_N - x_{N-1}}{h^2}\right) - \frac{1}{2}\left(\mu_N^T \cdot \frac{\partial}{\partial x_N} H_{\text{KT}}(x_N, t_N, \lambda_{\text{end}})\right)^T &= 0
\end{aligned}$$

Example 4.11 Consider again the parametric multiobjective optimization problem $\min_x F : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2,\ F = (f_1, f_2)$,

$$\begin{aligned} f_1(x,\lambda) &= \lambda\left((x_1-2)^2+(x_2-2)^2\right)+(1-\lambda)\left((x_1-2)^4+(x_2-2)^8\right) \\ f_2(x,\lambda) &= (x_1+2\lambda)^2+(x_2+2\lambda)^2 \end{aligned}$$

in which each objective function depends on the parameter $\lambda \in [0,1]$, i. e. $\lambda_{\text{start}} = 0$ and $\lambda_{\text{end}} = 1$.

To find out if one Pareto point exists for which all λ-dependent Pareto sets intersect, Problem (4.18) has been considered. The discretized Euler-Lagrange equations for energy minimization of this example have been solved with the MATLAB routine *fsolve* (cf. [101]) using $N = 20$ discretization points. The left plot in Figure 4.16 shows the solution path (green with black dots) of the λ-robust Pareto point (red) in the x_1-x_2-λ-space. In this plot, also some Pareto sets (gray curves) for different values of λ are plotted. In the right plot of Figure 4.16 a projection of the same data to the x_1-x_2-plane is shown – one can observe that the λ-robust Pareto point (red) indeed does not change under the variation of λ.

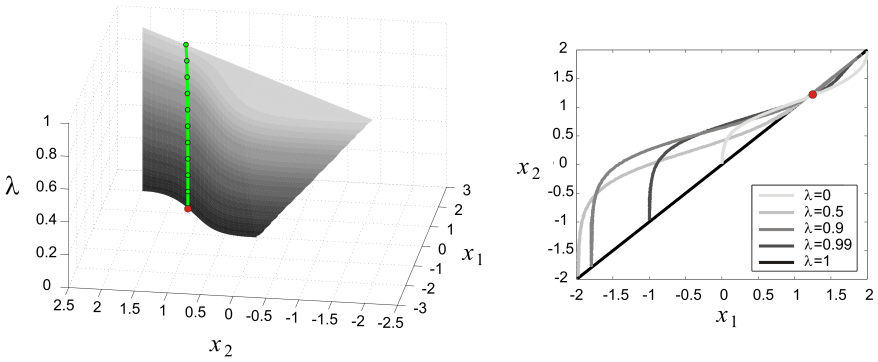

Figure 4.16: λ-robust solution path and some Pareto sets (on the left) and the projection onto the x_1-x_2-plane (on the right)

Concept II: Locally robust Pareto points

In contrast to the robustness concept proposed in the last section now a local strategy is considered. The following results have already been published in a like manner in [21]. Again, it is assumed that the parameter λ is varied within a given interval $[\lambda_{\text{start}}, \lambda_{\text{end}}] \subset \mathbb{R}$. For each point of a discrete covering of the initial Pareto set, λ-dependent solution paths are approximated that satisfy the Kuhn-Tucker equations and do locally change as little as possible

under the variation of λ. Thus, as long as a starting point is feasible for increasing values of λ the solution path is given by this constant value. If a point on the path is not feasible for an increased value of λ, then the solution path consists of those points that vary as little as necessary compared to the previously computed point. The robustness of a Pareto point against variations of the external parameter can be measured through the length of these paths.

In order to realize the numerical approximation of these solution paths, again numerical path following methods can come into play. In order to force the path into that direction in which the Pareto point changes as little as possible for increasing values of λ the Kuhn-Tucker system has to be expanded by certain equations.

Approaching this problem, first only a given point $y_0 \in S_{\lambda_0}$ and a set of substationary points S_{λ_1} are considered. The aim is to compute a point on S_{λ_1} which has minimal distance to y_0. This leads to a minimization problem with equality constraints as described in Section 2.2 (pages 16 ff.).

The following proposition gives a necessary condition for a point on S_{λ_1} to have minimal distance to the point y_0.

Proposition 4.12 *Let λ_0,$\lambda_1 \in [\lambda_{\text{start}}, \lambda_{\text{end}}]$ be two nonequal fixed values of the continuation parameter λ. Let $y_0 \in S_{\lambda_0}$ be fixed, let $x \in S_{\lambda_1}$ and $F = (f_1, \ldots, f_k)^T$. Let*

$$\nabla_{(x,t)} H^1_{\text{KT}}(x, t, \lambda_1), \ldots, \nabla_{(x,t)} H^{n+1}_{\text{KT}}(x, t, \lambda_1)$$

be linearly independent. If

a) x has the minimal Euclidean distance to y_0 or

b) $F(x, \lambda_1)$ has the minimal Euclidean distance to $F(y_0, \lambda_0)$,

then there exist multipliers $t \in \mathbb{R}^k$ and $(\mu_1, \ldots, \mu_{n+1})^T = \mu \in \mathbb{R}^{n+1}$, such that

$$\begin{aligned} \begin{pmatrix} H^1_{\text{KT}} \\ \vdots \\ H^n_{\text{KT}} \\ H^{n+1}_{\text{KT}} \end{pmatrix}(x, t, \lambda_1) = & \begin{pmatrix} \sum_{i=1}^k t_i^2 \nabla_x f_i(x, \lambda_1) \\ \sum_{i=1}^k t_i^2 - 1 \end{pmatrix} = 0 \\ \begin{pmatrix} H^1_L \\ \vdots \\ H^{n+k}_L \end{pmatrix}(x, t, \mu, \lambda_1) = & \begin{pmatrix} d(x) \\ 0 \end{pmatrix} - \left(\mu^T \cdot \tfrac{\partial}{\partial(x,t)} \begin{pmatrix} H^1_{\text{KT}} \\ \vdots \\ H^{n+1}_{\text{KT}} \end{pmatrix}(x, t, \lambda_1) \right)^T = 0, \end{aligned} \tag{4.23}$$

where – in the corresponding cases – $d : \mathbb{R}^n \to \mathbb{R}^n$ is given as

a) $d(x) = d(x; y_0, \lambda_0, \lambda_1) = 2(x - y_0)$ *or*

b) $d(x) = d(x; y_0, \lambda_0, \lambda_1) = 2\left(\sum_{i=1}^{k}(f_i(x, \lambda_1) - f_i(y_0, \lambda_0))\nabla_x f_i(x, \lambda_1)\right)$.

Here, d is called a **decision function**.

Proof:

a) A point x with minimal Euclidean distance to y_0 is given by the solution of the optimization problem

$$\begin{aligned} &\min_x \|x - y_0\|_2^2 \\ &\text{subject to } x \in S_{\lambda_1}. \end{aligned}$$

The set S_{λ_1} is described by the n Kuhn-Tucker equations, and thus this minimization problem leads to an optimization problem with equality constraints:

$$\begin{aligned} &\min_x \|x - y_0\|_2^2 \\ &\text{subject to } \begin{pmatrix} H_{\mathrm{KT}}^1 \\ \vdots \\ H_{\mathrm{KT}}^{n+1} \end{pmatrix}(x, t, \lambda_1) = \begin{pmatrix} \sum_{i=1}^{k} t_i^2 \nabla_x f_i(x, \lambda_1) \\ \sum_{i=1}^{k} t_i^2 - 1 \end{pmatrix} = 0. \end{aligned}$$

If the point (x, t) is an extremum, then there exist Lagrangian multipliers $\mu_1, \cdots, \mu_{n+1} \in \mathbb{R}$, so that

$$\begin{aligned} &\nabla_{(x,t)}\|x - y_0\|_2^2 - \mu_1 \nabla_{(x,t)} H_{\mathrm{KT}}^1(x, t, \lambda_1) - \cdots \\ &\qquad \cdots - \mu_{n+1}\nabla_{(x,t)} H_{\mathrm{KT}}^{n+1}(x, t, \lambda_1) &&= 0 \\ \Leftrightarrow \quad &\begin{pmatrix} 2(x - y_0) \\ 0 \end{pmatrix} - \mu_1 \nabla_{(x,t)} H_{\mathrm{KT}}^1(x, t, \lambda_1) - \cdots \\ &\qquad \cdots - \mu_{n+1}\nabla_{(x,t)} H_{\mathrm{KT}}^{n+1}(x, t, \lambda_1) &&= 0 \end{aligned}$$

(by use of Theorem 2.7). Combining these $n + k$ equations with the equality constraints one obtains a system of equations as stated in the proposition.

b) The same idea works in objective space. Here, the norm of the difference of the function values of x and y_0 has to be minimized:

$$\begin{aligned} &\min_x \left\| \begin{pmatrix} f_1(x, \lambda_1) \\ \vdots \\ f_k(x, \lambda_1) \end{pmatrix} - \begin{pmatrix} f_1(y_0, \lambda_0) \\ \vdots \\ f_k(y_0, \lambda_0) \end{pmatrix} \right\|_2^2 \\ &\qquad \text{subject to } x \in S_{\lambda_1}. \end{aligned}$$

The gradient w. r. t. x of the objective function of this minimization problem is given as

$$\nabla_x \left(\left\| \begin{pmatrix} f_1(x,\lambda_1) \\ \vdots \\ f_k(x,\lambda_1) \end{pmatrix} - \begin{pmatrix} f_1(y_0,\lambda_0) \\ \vdots \\ f_k(y_0,\lambda_0) \end{pmatrix} \right\|_2^2 \right) =$$

$$= \begin{pmatrix} 2(f_1(x,\lambda_1) - f_1(y_0,\lambda_0))\frac{\partial f_1(x,\lambda_1)}{\partial x_1} + \cdots + 2(f_k(x,\lambda_1) - f_k(y_0,\lambda_0))\frac{\partial f_k(x,\lambda_1)}{\partial x_1} \\ \vdots \\ 2(f_1(x,\lambda_1) - f_1(y_0,\lambda_0))\frac{\partial f_1(x,\lambda_1)}{\partial x_n} + \cdots + 2(f_k(x,\lambda_1) - f_k(y_0,\lambda_0))\frac{\partial f_k(x,\lambda_1)}{\partial x_n} \end{pmatrix}$$

$$= 2\left(\sum_{i=1}^k (f_i(x,\lambda_1) - f_i(y_0,\lambda_0))\nabla_x f_i(x,\lambda_1)\right)$$

Additionally,

$$\nabla_t \left(\left\| \begin{pmatrix} f_1(x,\lambda_1) \\ \vdots \\ f_k(x,\lambda_1) \end{pmatrix} - \begin{pmatrix} f_1(y_0,\lambda_0) \\ \vdots \\ f_k(y_0,\lambda_0) \end{pmatrix} \right\|_2^2 \right) = 0.$$

Combining this with the Lagrangian multipliers which stem from the First-Order Necessary Condition given in Theorem 2.7, the desired result follows. □

Solutions of the system of equations given in (4.23) are isolated if the corresponding Jacobian matrix is regular. This is in particular important for the numerical computations described below, because the full rank of the Jacobian is needed as an assumption for the numerical path following used therein. In the following, it is proven that this Jacobian matrix is indeed regular if the Second-order Sufficiency Condition (cf. Theorem 2.8) for the underlying mimization problem described in the proof of Proposition 4.12 is satisfied. In Proposition 4.13 first the condition for the regularity of the Jacobian J is reformulated in a geometrical way and afterwards this condition is derived from the Second-order Sufficiency Condition in Corrolary 4.14.

Proposition 4.13 *Let*

$$X = \begin{pmatrix} \sum_{i=1}^k t_i^2 \frac{\partial^2}{\partial x^2} f_i & 2t_1\nabla_x f_1 & \cdots & 2t_k\nabla_x f_k \\ 0 & 2t_1 & \cdots & 2t_k \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+k)}$$

and

$$Y = \begin{pmatrix} \frac{\partial}{\partial x}(d - m_x) & -(\frac{\partial}{\partial x} m_t)^T \\ -\frac{\partial}{\partial x} m_t & -\frac{\partial}{\partial t}(m_t) \end{pmatrix} \in \mathbb{R}^{(n+k)\times(n+k)}$$

with

$$\begin{aligned} m_t &= (\mu_1 \nabla_t H^1_{\mathrm{KT}} + \ldots + \mu_{n+1} \nabla_t H^{n+1}_{\mathrm{KT}}), \\ m_x &= (\mu_1 \nabla_x H^1_{\mathrm{KT}} + \ldots + \mu_n \nabla_x H^n_{\mathrm{KT}}). \end{aligned}$$

Let $t^\star$ and $\mu^\star$ denote the corresponding multipliers from the Kuhn-Tucker- and the Lagrangian equations.

The Jacobian J of the left hand side in (4.23) is regular in the point $(x^\star, t^\star, \mu^\star; y_0, \lambda_0, \lambda_1)$ if and only if

$$\mathrm{im} Y \Big|_{\ker X} \cap (\ker X)^\perp = \{0\}. \tag{4.24}$$

Proof: First of all, the structure of the Jacobian of the system of equations as stated in Proposition 4.23 is examined.

Claim: The Jacobian is given as $J = \begin{pmatrix} X & 0 \\ Y & -X^T \end{pmatrix} \in \mathbb{R}^{(2n+k+1)\times(2n+k+1)}$.

Proof: The Jacobian matrix consists of the following entries:

$$\begin{aligned} J &= \begin{pmatrix} \frac{\partial H^{1,\ldots,n}_{\mathrm{KT}}}{\partial x} & \frac{\partial H^{1,\ldots,n}_{\mathrm{KT}}}{\partial t} & \frac{\partial H^{1,\ldots,n}_{\mathrm{KT}}}{\partial \mu} \\ \frac{\partial H^{n+1}_{\mathrm{KT}}}{\partial x} & \frac{\partial H^{n+1}_{\mathrm{KT}}}{\partial t} & \frac{\partial H^{n+1}_{\mathrm{KT}}}{\partial \mu} \\ \frac{\partial H^{1,\ldots,n}_{L}}{\partial x} & \frac{\partial H^{1,\ldots,n}_{L}}{\partial t} & \frac{\partial H^{1,\ldots,n}_{L}}{\partial \mu} \\ \frac{\partial H^{n+1,\ldots,n+k}_{L}}{\partial x} & \frac{\partial H^{n+1,\ldots,n+k}_{L}}{\partial t} & \frac{\partial H^{n+1,\ldots,n+k}_{L}}{\partial \mu} \end{pmatrix} \\ &= \begin{pmatrix} K & C & 0 \\ 0 & v & 0 \\ B & H & G \\ A & L & F \end{pmatrix} \in \mathbb{R}^{(2n+k+1)\times(2n+k+1)} \end{aligned}$$

where

$$
\begin{aligned}
K &= \sum_{i=1}^{k} t_i^2 \frac{\partial^2}{\partial x^2} f_i \in \mathbb{R}^{n \times n} \\
B &= \frac{\partial}{\partial x}(d - m_x) \in \mathbb{R}^{n \times n} \\
&\quad \text{with } m_x = (\mu_1 \nabla_x H_{\mathrm{KT}}^1 + \ldots + \mu_n \nabla_x H_{\mathrm{KT}}^n) \\
A &= -\frac{\partial}{\partial x} m_t \in \mathbb{R}^{k \times n} \\
&\quad \text{with } m_t = (\mu_1 \nabla_t H_{\mathrm{KT}}^1 + \ldots + \mu_{n+1} \nabla_t H_{\mathrm{KT}}^{n+1}) \\
C &= \begin{pmatrix} 2t_1 \nabla_x f_1 & \cdots & 2t_k \nabla_x f_k \end{pmatrix} \in \mathbb{R}^{n \times k} \\
H &= -\frac{\partial}{\partial t} m_x \in \mathbb{R}^{n \times k} \\
L &= -\frac{\partial}{\partial t} m_t \in \mathbb{R}^{k \times k} \\
G &= \begin{pmatrix} -\nabla_x H_{\mathrm{KT}}^1 & \cdots & -\nabla_x H_{\mathrm{KT}}^n & 0 \end{pmatrix} \in \mathbb{R}^{n \times n+1} \\
F &= \begin{pmatrix} -\nabla_t H_{\mathrm{KT}}^1 & \cdots & -\nabla_t H_{\mathrm{KT}}^{n+1} \end{pmatrix} \in \mathbb{R}^{k \times n+1} \\
v &= \begin{pmatrix} 2t_1 & \cdots & 2t_k \end{pmatrix} \in \mathbb{R}^k.
\end{aligned}
$$

One can easily see that

$$
\begin{aligned}
H &= -\frac{\partial}{\partial t} m_x \\
&= \frac{\partial}{\partial t} \left(-\mu_1 \nabla_x H_{\mathrm{KT}}^1 - \ldots - \mu_{n+1} \nabla_x H_{\mathrm{KT}}^{n+1} \right) \\
&= \frac{\partial}{\partial t} \left(\nabla_x (-\mu_1 H_{\mathrm{KT}}^1 - \ldots - \mu_n H_{\mathrm{KT}}^{n+1} \right) \\
&= \left(\tfrac{\partial}{\partial x} (\nabla_t (-\mu_1 H_{\mathrm{KT}}^1 - \ldots - \mu_n H_{\mathrm{KT}}^{n+1})) \right)^T \\
&= A^T.
\end{aligned}
$$

Setting $\tilde{K} = \begin{pmatrix} K \\ 0 \end{pmatrix} \in \mathbb{R}^{(n+1)\times n}$ it follows that

$$\begin{aligned} G &= \begin{pmatrix} -\nabla_x H^1_{\mathrm{KT}} & \cdots & -\nabla_x H^n_{\mathrm{KT}} & 0 \end{pmatrix} \\ &= \begin{pmatrix} -\frac{\partial H^1_{\mathrm{KT}}}{\partial x_1} & \cdots & -\frac{\partial H^n_{\mathrm{KT}}}{\partial x_1} & 0 \\ \vdots & & \vdots & \vdots \\ -\frac{\partial H^1_{\mathrm{KT}}}{\partial x_n} & \cdots & -\frac{\partial H^n_{\mathrm{KT}}}{\partial x_n} & 0 \end{pmatrix} \\ &= \begin{pmatrix} -\sum_{i=1}^k t_i^2 \frac{\partial^2}{\partial x^2} f_i & 0 \end{pmatrix} \\ &= -\tilde{K}^T. \end{aligned}$$

Defining $\tilde{C} = \begin{pmatrix} C \\ v \end{pmatrix}$ one gets

$$\begin{aligned} \tilde{C} &= \begin{pmatrix} 2t_1 \nabla_x f_1 & \cdots & 2t_k \nabla_x f_k \\ 2t_1 & \cdots & 2t_k \end{pmatrix} \\ &= \begin{pmatrix} \frac{\partial H^1_{\mathrm{KT}}}{\partial t_1} & \cdots & \frac{\partial H^1_{\mathrm{KT}}}{\partial t_k} \\ \vdots & & \vdots \\ \frac{\partial H^n_{\mathrm{KT}}}{\partial t_1} & \cdots & \frac{\partial H^n_{\mathrm{KT}}}{\partial t_k} \\ \frac{\partial H^{n+1}_{\mathrm{KT}}}{\partial t_1} & \cdots & \frac{\partial H^{n+1}_{\mathrm{KT}}}{\partial t_k} \end{pmatrix} \\ &= \begin{pmatrix} (\nabla_t H^1_{\mathrm{KT}})^T \\ \vdots \\ (\nabla_t H^{n+1}_{\mathrm{KT}})^T \end{pmatrix} \\ &= \begin{pmatrix} \nabla_t H^1_{\mathrm{KT}} & \cdots & \nabla_t H^{n+1}_{\mathrm{KT}} \end{pmatrix}^T \\ &= -F^T. \end{aligned}$$

Thus, including this structure the Jacobian is given as

$$J = \begin{pmatrix} \tilde{K} & \tilde{C} & 0 \\ B & A^T & -\tilde{K}^T \\ A & L & -\tilde{C}^T \end{pmatrix} = \begin{pmatrix} X & 0 \\ Y & -X^T \end{pmatrix} \in \mathbb{R}^{(2n+k+1)\times(2n+k+1)}$$

with $X = \begin{pmatrix} \tilde{K} & \tilde{C} \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+k)}$ and $Y = \begin{pmatrix} B & A^T \\ A & L \end{pmatrix} \in \mathbb{R}^{(n+k)\times(n+k)}$. The entries of these matrices are exactly as assumed in the proposition and thus the claim is proven.

The Jacobian J is regular if and only if for vectors $a \in \mathbb{R}^{n+k}$ and $b \in \mathbb{R}^{n+1}$

$$\begin{pmatrix} X & 0 \\ Y & -X^T \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = 0 \Leftrightarrow \begin{pmatrix} a \\ b \end{pmatrix} = 0, \tag{4.25}$$

i. e. if the system of equations

$$\begin{aligned} Xa &= 0 \\ Ya - X^T b &= 0 \end{aligned}$$

implies $a = 0$ and $b = 0$. Therefore, J is regular if for $0 \neq a \in \ker X$ and any $0 \neq b \in \mathbb{R}^{n+1}$, $Ya \neq X^T b$, that is, $Ya \notin \mathrm{im} X^T \; \forall 0 \neq a \in \ker X$, which means

$$\mathrm{im} Y\Big|_{\ker X} \cap \mathrm{im} X^T = \{0\}.$$

As $\mathrm{im}\,(X^T) = (\ker X)^\perp$ (cf. e. g. [98]), this is equivalent to

$$\mathrm{im} Y\Big|_{\ker X} \cap (\ker X)^\perp = \{0\}.$$

□

Corollary 4.14 *Let $D : \mathbb{R}^n \to \mathbb{R}$ be a function with $\nabla_x D = d$. Assume there is a point $x^\star \in \mathbb{R}^n$ that satisfies the Second-order Sufficiency Condition given in Theorem 2.8 for the problem*

$$\begin{aligned} &\min_x D(x; y_0, \lambda_0, \lambda_1) \\ &\text{subject to } x \in S_{\lambda_1}. \end{aligned} \tag{4.26}$$

Let $t^\star$ and $\mu^\star$ denote the corresponding multipliers from the Kuhn-Tucker- and the Lagrangian equations.

Then the Jacobian of the left hand side in (4.23) is regular in the point $(x^\star, t^\star, \mu^\star; y_0, \lambda_0, \lambda_1)$.

Proof: Solutions of (4.23) satisfy the equality constraints of the optimization problem and also have the property that the gradient of the Lagrangian function equals zero. Thus, the important assumption in Theorem 2.8 is the positivity property of the Hessian matrix $L''(x^\star, t^\star)$. Observe that, as the Lagrangian function is given as

$$L(x, t; \mu, y_0, \lambda_0, \lambda_1) = D(x; y_0, \lambda_0, \lambda_1) - \mu^T \cdot \begin{pmatrix} H^1_{\mathrm{KT}} \\ \vdots \\ H^{n+1}_{\mathrm{KT}} \end{pmatrix} (x, t; \lambda_1),$$

it holds

$$L''(x^\star, t^\star; \mu^\star) = Y(x^\star, t^\star)\Big|_{\mu=\mu^\star}.$$

Furthermore, the matrix X is the Jacobian of the equality constraints in problem 4.26 and thus, the set M defined in Theorem 2.8 equals $\ker X$.

In the following, it is proven by contradiction that, if Y is positive definite on $\ker X$, then $\mathrm{im} Y\Big|_{\ker X} \cap (\ker X)^\perp = \{0\}$:

Let $0 \neq s \in \mathrm{im} Y\Big|_{\ker X} \cap (\ker X)^\perp$. Then there exists

$$0 \neq a \in \ker X \text{ with } Ya = s \text{ and } s \in (\ker X)^\perp$$

and thus $a^T Y a = a^T s = 0$, as a and s are orthogonal. This is a contradiction to the assumption that Y is positive definite.

In particular, it now follows from Proposition 4.13 that the Jacobian J is regular if the Second-order Sufficiency Condition is satisfied for the underlying optimization problem. □

Condition (4.24) can be interpreted geometrically. It indicates that the set of points satisfying the Kuhn-Tucker equations may not approach a level set of the function D with more than first order. As D gives the distance between S_{λ_1} and y_0 this especially means that S_{λ_1} is not allowed to be (partly) circular around y_0.

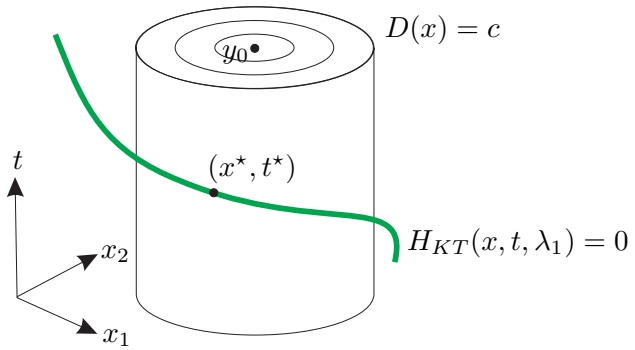

Figure 4.17: Condition (4.24) is not satisfied in such a case

More precisely, consider a point $(x^\star, t^\star)$ on the set ST_{λ_1}, where

$$ST_{\lambda_1} = \{(x, t) \in \mathbb{R}^n \times \mathbb{R}^k \mid H_{\mathrm{KT}}(x, t, \lambda_1) = 0\},$$

and let $\mu^\star$ be the corresponding Lagrangian multiplier from Theorem 2.7. In the following, the restriction of the Lagrangian L to the set ST_{λ_1} and a

fixed value $\mu = \mu^\star$ will be considered defining $\tilde{L} : ST_{\lambda_1} \to \mathbb{R}$ by $\tilde{L}(x,t) = L(x,t;\mu^\star)$. As on ST_{λ_1} the function H_{KT} is identically zero, $\tilde{L}$ coincides with $D\Big|_{ST_{\lambda_1}}$. For the sake of brevity, in the following computation $u = (x,t)$ and $u^\star = (x^\star, t^\star)$ are used. The Taylor expansion of $\tilde{L}$ around the point $(x^\star, t^\star)$ yields

$$\begin{aligned}\tilde{L}(u) &= L(u^\star;\mu^\star) + \nabla_u L(u^\star;\mu^\star)(u-u^\star) + \cdots \\ &\quad + \frac{1}{2}(u-u^\star)^T L''(u^\star;\mu^\star)(u-u^\star) + \text{h. o. t.} \\ &\overset{\text{Thm. 2.7}}{=} L(u^\star;\mu^\star) + \frac{1}{2}(u-u^\star)^T Y(u^\star)\Big|_{\mu=\mu^\star}(u-u^\star) + \text{h. o. t.}\end{aligned}$$

From this it follows that condition (4.24) is satisfied if the Taylor expansion of $\tilde{L}$ has a positive definite term of second order. As D equals $\tilde{L}$ on ST_{λ_1}, this is not the case in particular if ST_{λ_1} approaches a level set of D with more than first order, i. e. if the curvature of ST_{λ_1} in $(x^\star, t^\star)$ is equal to the curvature of the level set of D. Figure 4.17 visualizes this forbidden case.

Algorithmic implementation

The aim is to detect those points $x \in S_{\lambda_{\text{start}}}$ that vary hardly under the variation of the parameter λ – in preimage space or in objective space respectively. Therefore, certain paths that contain substationary points for different values of λ varying from λ_{start} to λ_{end} are computed. These paths are calculated in such a way that the distance from the previously computed point to the next point – which has to lie on the set of substationary points for the new λ-value – is minimal in every step. In this case, the situation described in Proposition 4.12 arises in every step of a certain path following method which is going to be described in the following.

As explained in Section 2.4.2 numerical path following can be performed by predictor-corrector methods. The previously described predictor-corrector methods are adapted for this special context: Starting on the Pareto set for a special value of λ, first a solution for a higher value of λ is predicted with a fixed vector α and then the solution for the new fixed λ is corrected to the one with minimal distance to the preceding Pareto set or set of substationary points respectively. These steps are repeated iteratively until λ equals or exceeds the desired value λ_{end}.

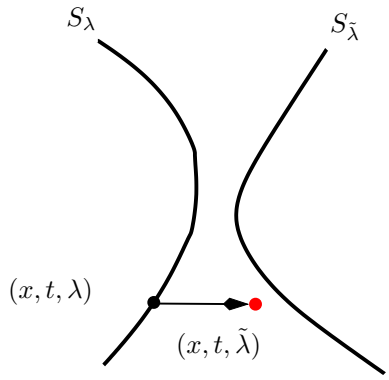

Figure 4.18: Schematic illustration of the predictor step

Generation of starting points:

First of all, an approximation of the Pareto set for $\lambda = \lambda_{\text{start}}$ (e. g. using the algorithms described in Section 3.2.3) is computed. This yields a set of points which are in particular substationary points. For every point x_0 within this set the corresponding vector t_0 (or α_0 respectively) can be computed by solving the Kuhn-Tucker equations for $x = x_0$, $\lambda = \lambda_{\text{start}}$.

Initially, $\mu_0 = 0$, $(x_{\text{old}}, \lambda_{\text{old}}) = (x_0, \lambda_{\text{start}})$ and $u = (x_0, t_0, \lambda_{\text{start}}, \mu_0)$ are set.

Predictor step:

In this step, the value of λ for fixed values of (x, t, μ) (or (x, α, μ) respectively) is simply increased, i. e.

$$u_{\text{Pred}} = u + \begin{pmatrix} 0 \\ 0 \\ h \\ 0 \end{pmatrix} = \begin{pmatrix} x \\ t \\ \tilde{\lambda} \\ \mu \end{pmatrix}$$

where h is an adequately controlled stepsize.

Corrector step:

Now $\lambda = \tilde{\lambda}$ is set fixed and x, t and μ are varied until the point with

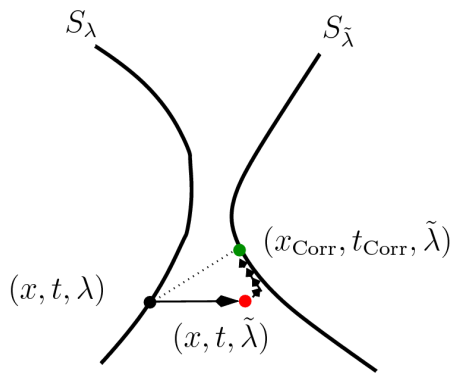

Figure 4.19: Schematic illustration of the corrector step

minimal distance to the preceding set S_λ is determined. Therefore, the system

$$H_{\text{Corr}}(x,t,\mu) = \begin{pmatrix} \sum_{i=1}^{k} t_i^2 \nabla_x f_i(x,\tilde{\lambda}) \\ \sum_{i=1}^{k} t_i^2 - 1 \\ \begin{pmatrix} d(x; x_{\text{old}}, \lambda_{\text{old}}, \tilde{\lambda}) \\ 0 \end{pmatrix} - \mu_1 \nabla_{(x,t)} H^1_{\text{KT}}(x,t,\tilde{\lambda}) - \cdots \\ \cdots - \mu_n \nabla_{(x,t)} H^n_{\text{KT}}(x,t,\tilde{\lambda}) - \mu_{n+1} \begin{pmatrix} 0 \\ 2t_1 \\ \vdots \\ 2t_k \end{pmatrix} \end{pmatrix} = 0 \tag{4.27}$$

is considered and solved numerically with Newton's method by using the predictor as an initial guess.

Let $(x_{\text{Corr}}, t_{\text{Corr}}, \mu_{\text{Corr}})$ be a solution of $H_{\text{Corr}}(x,t,\mu) = 0$ and set $(x_{\text{old}}, \lambda_{\text{old}}) = (x_{\text{Corr}}, \tilde{\lambda})$ and $u = (x_{\text{Corr}}, t_{\text{Corr}}, \tilde{\lambda}, \mu_{\text{Corr}})$.

The predictor and corrector steps are repeated until $\tilde{\lambda} \geq \lambda_u$ is reached.

This algorithm comes up with different paths for different starting points $u_0 = (x_0, t_0, \lambda_{\text{start}}, \mu_0)$. To measure the length of every path, the Euclidean distances of the points x_i on that path are simply summed up in case a). That means, if the path consists of $N+1$ points, its length is computed by

$$\text{Path-II-Length}(x_0) = \|x_1 - x_0\| + \|x_2 - x_1\| + \ldots + \|x_N - x_{N-1}\| \tag{4.28}$$

for every starting point x_0 on the Pareto set for $\lambda = \lambda_{\text{start}}$. In case b), i. e. paths of minimal length in objective space, the Euclidean distances of the

objective values for the computed paths are summed up. More precisely, again assuming that $N+1$ points on this path have been computed,

$$\text{Path-II-Length}(F(x_0)) = \\ \|F(x_1,\lambda_1) - F(x_0,\lambda_{\text{start}})\| + \ldots + \|F(x_N,\lambda_N) - F(x_{N-1},\lambda_{N-1})\|.$$

The aim is to determine the points in the Pareto set for $\lambda = \lambda_{\text{start}}$ that hardly vary under the variation of λ. With the following definition it is obvious how to compute these points.

Definition 4.15 (Robustness w. r. t. Concept II (cf. [21])) *Starting points x_0 whose corresponding paths have minimal Path-II-Length in preimage or objective space are called* λ-robust Pareto points w. r. t. Concept II *in preimage or objective space respectively.*

Example 4.16 Consider again the parametric multiobjective optimization problem $\min_x F : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2,\ F = (f_1, f_2)$,

$$\begin{aligned} f_1(x,\lambda) &= \lambda\left((x_1-2)^2 + (x_2-2)^2\right) + (1-\lambda)\left((x_1-2)^4 + (x_2-2)^8\right) \\ f_2(x,\lambda) &= (x_1+2\lambda)^2 + (x_2+2\lambda)^2 \end{aligned}$$

for which each objective function depends on the parameter $\lambda \in [0,1]$, i. e. $\lambda_{\text{start}} = 0$ and $\lambda_{\text{end}} = 1$. The Kuhn-Tucker equations for this multiobjective optimization problem are given as

$$\begin{aligned} t_1^2\left(2\lambda(x_1-2) + 4(1-\lambda)(x_1-2)^3\right) + 2t_2^2\,(x_1+2\lambda) &= 0 \\ t_1^2\left(2\lambda(x_2-2) + 8(1-\lambda)(x_2-2)^7\right) + 2t_2^2\,(x_2+2\lambda) &= 0 \\ t_1^2 + t_2^2 - 1 &= 0 \end{aligned} \tag{4.29}$$

a) *Robust Pareto points in parameter space:*

Expanding the Kuhn-Tucker systems as stated in Proposition 4.12 results in the following system of nonlinear equations for the corrector step:

$$\begin{aligned} t_1^2(2\tilde\lambda(x_1-2) + 4(1-\tilde\lambda)(x_1-2)^3) + 2t_2^2(x_1+2\tilde\lambda) &= 0 \\ t_1^2(2\tilde\lambda(x_2-2) + 8(1-\tilde\lambda)(x_2-2)^7) + 2t_2^2(x_2+2\tilde\lambda) &= 0 \\ t_1^2 + t_2^2 - 1 &= 0 \\ 2(x_1 - x_1^{old}) - \mu_1(t_1^2(2\tilde\lambda + 12(1-\tilde\lambda)(x_1-2)^2) + 2t_2^2) &= 0 \\ 2(x_2 - x_2^{old}) - \mu_2(t_1^2(2\tilde\lambda + 56(1-\tilde\lambda)(x_2-2)^6) + 2t_2^2) &= 0 \\ -\mu_1(2\tilde\lambda(x_1-2) + 4(1-\tilde\lambda)(x_1-2)^3 - (x_1+2\tilde\lambda))\ldots & \\ \ldots - \mu_2(2\tilde\lambda(x_2-2) + 8(1-\tilde\lambda)(x_2-2)^7 - (x_2+2\tilde\lambda)) &= 0 \end{aligned}$$

Making use of the numerical path following methods described in Section 2.4.2 points on the solution curves varying $\lambda \in [0,1]$ for every starting point on the Pareto set for $\lambda = 0$ were computed. Some examples of these paths are shown in Figure 4.20 on the left. Having generated these paths, the length of each path is computed by simply summing up the squared Euclidean distances as mentioned above. The result is shown on the right hand side of Figure 4.20. The length of each path is plotted versus the ith starting point, where i traverses the set of Pareto points for $\lambda = 0$. In this case the starting points are ordered in the following way: they wander from below (point $x_1 = x_2 = 0$) towards the upper points of the Pareto set ($x_1 = x_2 = 2$). One can observe that the lengths of the paths decrease up to the 112th starting point. This point is in fact (an approximation of) the one for which the Pareto sets for every $\lambda \in [0,1]$ coincide, i. e. one of the demanded robust Pareto points. As one can see, the point $x_1 = x_2 = 2$ has length zero, too, so it can also be detected as a robust Pareto point although the Jacobian is not regular in this point. It can be determined in this special case because $x_1 = x_2 = 2$ is a solution for every value of $\lambda \in [0,1]$. Therefore, in our path following algorithm only predictor steps are performed wherein the Jacobian is not required.

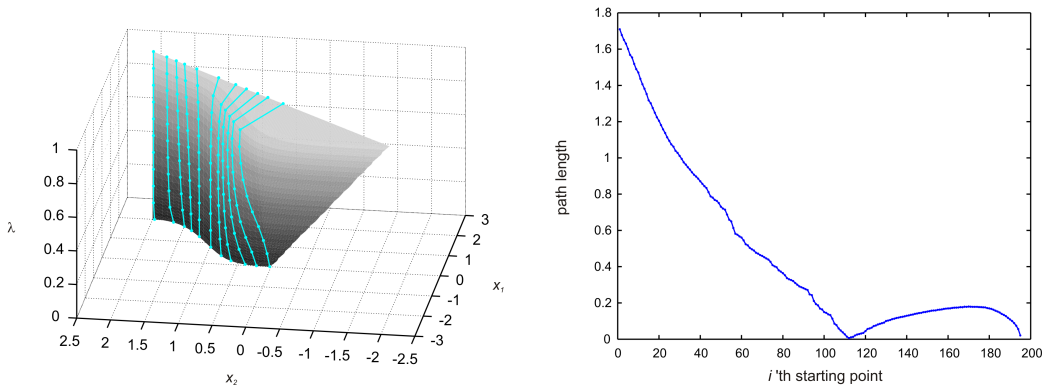

Figure 4.20: Some examples for the computed paths in parameter space (on the left) and the length of each computed path (on the right)

b) *Robust Pareto points in objective space:*

As stated in the proposition the same ideas are working when considering the distances of the images of substationary points. This ansatz

does in general not result in the same robust Pareto points as in parameter space, because the function values do not only depend on the points in parameter space itself but also on the values of λ. For example, in the previous case $x_s \approx (1.1621, 1.1621)$ is a robust Pareto point in parameter space, but the function value for $\lambda = 0$ which is $(f_1, f_2)(x_s, 0) \approx (0.7359, 2.7010)$ is far away from the one for $\lambda = 1$ which is $(f_1, f_2)(x_s, 1) = (1.4042, 19.9978)$.

For the computation of robust Pareto points in objective space, the following equations have to be considered in addition to the Kuhn-Tucker system in the corrector step:

$$\begin{aligned}
2(f_1(x,\tilde{\lambda}) - f_1^{old}(x,\tilde{\lambda}))\frac{\partial f_1}{\partial x_1} + 2(f_2(x,\tilde{\lambda}) - f_2^{old}(x,\tilde{\lambda}))\frac{\partial f_2}{\partial x_1} + \cdots \\
-\mu_1(t_1^2(2\tilde{\lambda} + 12(1-\tilde{\lambda})(x-2)2) + 2t_2^2) = 0 \\
2(f_1(x,\tilde{\lambda}) - f_1^{old}(x,\tilde{\lambda}))\frac{\partial f_1}{\partial x_2} + 2(f_2(x,\tilde{\lambda}) - f_2^{old}(x,\tilde{\lambda}))\frac{\partial f_2}{\partial x_2} + \cdots \\
-\mu_2(t_1^2(2\tilde{\lambda} + 56(1-\tilde{\lambda})(x_2-2)6) + 2t_2^2) = 0 \\
-2\mu_1 t_1(2\tilde{\lambda}(x_1-2) + 4(1-\tilde{\lambda})(x_1-2)^3) \\
-2\mu_2 t_1(2\tilde{\lambda}(x_2-2) + \cdots \\
+8(1-\tilde{\lambda})(x_2-2)^7) - 2\mu_3 t_1 = 0 \\
-4\mu_1 t_2(x_1 + 2\tilde{\lambda}) - 4\mu_2 t_2(x_2 + 2\tilde{\lambda}) - 2\mu_3 t_2 = 0
\end{aligned}$$

Some examples for resulting paths in objective space are shown in Figure 4.21. Comparing the distances in objective space, one finds that the left blue path is the desired path of robust Pareto points. The corresponding starting point is $x \approx (2, 2)$. So this point varies as little as possible both in objective and preimage space.

Figure 4.21 also shows the image of paths which have been optimized with respect to the variation in preimage space for the same starting points. As one can easily see these paths are not at all the same.

Example 4.17 Consider the multiobjective optimization problem

$$\min_x F(x),\ F : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}^3,\ F = (f_1, f_2, f_3)$$

with the objective functions

$$\begin{aligned}
f_1(x,\lambda) &= (x-4)^2 + (y-4)^2 + (z-4+\lambda)^2 \\
f_2(x,\lambda) &= (x+4)^2 + (y+4)^2 + (z+4+\lambda)^2 \\
f_3(x,\lambda) &= (x+4)^2 + (y-4)^2 + (z+4+\lambda)^2.
\end{aligned}$$

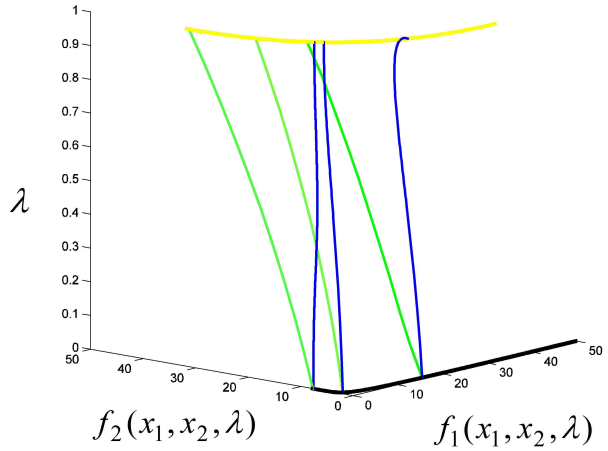

Figure 4.21: Examples of paths with minimum lengths in objective space (blue) and preimage space (green) between the Pareto sets for $\lambda = 0$ (black) and $\lambda = 1$ (yellow), visualized both in (f_1, f_2, λ)-space

The Pareto set is given by triangles in $\mathbb{R}^3$ which move in the z-direction when varying the parameter λ. The triangles lie in the plane E given by

$$E : \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \\ 4+\lambda \end{pmatrix} + t_1 \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + t_2 \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

for $t_1, t_2 \in \mathbb{R}$. The vector $v = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$ is orthogonal to this plane.

Figure 4.22 shows some examples for paths computed with the previously described algorithm. As one can observe these paths in fact follow the shortest possible distance between the Pareto sets for $\lambda = -1$ and $\lambda = 1$. For example, the black path exactly describes a translation by v of the starting point on S_{-1} to S_1. At the boundary of the Pareto set, where an orthogonal translation is not feasible, one can observe that indeed the shortest path within the Pareto sets for fixed values of λ is computed.

Comparison of Concept I and II

This section deals with the comparison of the two new concepts for robustness defined above. To make the two approaches comparable, first of all the variational approach is formulated in such a way that solutions from a fixed starting point on the Pareto set for $\lambda = \lambda_{\text{start}}$ to an arbitrary end point on

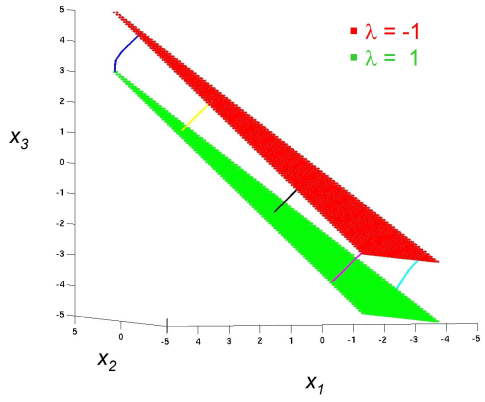

Figure 4.22: Some examples for computed paths between the Pareto sets for $\lambda = -1$ and $\lambda = 1$ in the 3d-example

$S_{\lambda_{\text{end}}}$ can be computed which lie within the λ-dependent set of substationary points. Thus, the starting point that so far has been free in Concept I has to be fixed instead. This leads to the variational integral

$$\begin{aligned} \min_{\gamma} \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} \|x'(\lambda)\|_2 \mathrm{d}\lambda \\ \text{s. t. } H_{\text{KT}}(x(\lambda), t(\lambda), \lambda) = 0 \end{aligned} \tag{4.30}$$

with the boundary conditions $x(\lambda_{\text{start}}) = x_0$, $t(\lambda_{\text{start}}) = t_0$ and the free end condition $x(\lambda_{\text{end}}) \in S_{\lambda_{\text{end}}}$.

The same Euler-Lagrange equations as before result with the only difference that one only obtains equations from the transversality condition for one free end. Therefore, in this case Equation (4.12) does not apply. To sum up, in the case of a fixed starting point the Euler-Lagragrange equations are given as:

$$\frac{x''(\lambda)}{\|x'(\lambda)\|_2} - \frac{x'(\lambda)}{\|x'(\lambda)\|_2^3} \left(\sum_{i=1}^{n} x_i'(\lambda) \cdot x_i''(\lambda) \right) - \left(\mu(\lambda)^T \cdot \frac{\partial}{\partial x} \begin{pmatrix} H_{\text{KT}}^1 \\ \vdots \\ H_{\text{KT}}^n \end{pmatrix} (\gamma(\lambda), \lambda) \right)^T = 0 \tag{4.31}$$

$$- \left(\mu(\lambda)^T \cdot \frac{\partial}{\partial t} \begin{pmatrix} H_{\text{KT}}^1 \\ \vdots \\ H_{\text{KT}}^{n+1} \end{pmatrix} (\gamma(\lambda), \lambda) \right)^T = 0 \tag{4.32}$$

$$\begin{pmatrix} H_{\text{KT}}^1 \\ \vdots \\ H_{\text{KT}}^n \\ H_{\text{KT}}^{n+1} \end{pmatrix} (\gamma(\lambda), \lambda) = \begin{pmatrix} \sum_{i=1}^{k} (t_i(\lambda))^2 \nabla_x f_i(x, \lambda) \\ \sum_{i=1}^{k} (t_i(\lambda))^2 - 1 \end{pmatrix} = 0, \tag{4.33}$$

and the end condition

$$\frac{1}{\|x'(\lambda_{\text{end}})\|_2} \cdot x'(\lambda_{\text{end}}) \quad \perp \quad \ker\left(\frac{\partial}{\partial x} H_{\text{KT}}(\gamma(\lambda_{\text{end}}), \lambda_{\text{end}})\right). \tag{4.34}$$

For numerical computations we again use the discretization of the variational problem, which in this case leads to the system of equations

$$\begin{aligned}
\frac{1}{h}\left(\frac{x_j - x_{j-1}}{\|x_j - x_{j-1}\|} - \frac{x_{j+1} - x_j}{\|x_{j+1} - x_j\|}\right) + \left(\mu_j^T \frac{\partial}{\partial x_j} H_{\text{KT}}(x_j, t_j, \lambda_j)\right)^T &= 0 \quad (4.35)\\
\left(\mu_j^T \cdot \frac{\partial}{\partial t_j} H_{\text{KT}}(x_j, t_j, \lambda_j)\right)^T &= 0 \quad (4.36)\\
H_{\text{KT}}(x_j, t_j, \lambda_j) &= 0 \quad (4.37)
\end{aligned}$$

for all $j = 1, \ldots, N-1$, and

$$\begin{aligned}
\frac{1}{h} \cdot \left(-\frac{x_N - x_{N-1}}{\|x_N - x_{N-1}\|}\right) + \frac{1}{2}\left(\mu_N^T \frac{\partial}{\partial x_N} H_{\text{KT}}(x_N, t_N, \lambda_{\text{end}})\right)^T &= 0 \quad (4.38)\\
\left(\mu_N^T \cdot \frac{\partial}{\partial t_N} H_{\text{KT}}(x_N, t_N, \lambda_{\text{end}})\right)^T &= 0 \quad (4.39)\\
H_{\text{KT}}(x_N, t_N, \lambda_{\text{end}}) &= 0. \quad (4.40)
\end{aligned}$$

Let $p_I = (x_0^\star, \ldots, x_N^\star)$ with $x_0^\star = x_0$ be a solution path[3] which – together with the corresponding multipliers – solves Equations (4.35) – (4.40). The length of this solution path can be computed as follows:

$$\mathit{Path}\text{-I-}\mathit{length}(x_0^\star) = \|x_1^\star - x_0^\star\| + \|x_2^\star - x_1^\star\| + \ldots + \|x_N^\star - x_{N-1}^\star\|$$

In the following example, solution paths of the discretized Euler-Lagrange equations (4.35) – (4.40) are compared with the solution paths from the numerical path following approach for determining λ-robust Pareto points (Concept II). To allow a fair comparison, the external parameter λ is discretized with the same fixed step size both in the discretization of the variational integral (4.30) and in the predictor step of the numerical path following algorithm (see page 102). We consider λ-robust Pareto points in preimage space, i. e. the decision function for Concept II is given as $d(x; x_{\text{old}}, \lambda_{\text{old}}, \lambda_{\text{old}} + h) = 2(x - x_{\text{old}})$.

[3] In the results presented here, the MATLAB solver fsolve [101] has been used to solve Equations (4.35) – (4.40) numerically.

Example 4.18 Consider again (as in Example) the parametric multiobjective optimization problem $\min_x F : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2,\ F = (f_1, f_2)$,

$$\begin{aligned} f_1(x,\lambda) &= (x_1 - \sin(\lambda))^2 + (x_2 - \cos(\lambda))^2 \\ f_2(x,\lambda) &= (x_1 - 2\sin(\lambda))^2 + (x_2 - 2\cos(\lambda))^2 \end{aligned}$$

in which the parameter λ is varied from 0 to 2π.

In the following, λ-robust solutions w. r. t. Concept I (using the discretized variational problem with fixed starting points) and w. r. t. Concept II (making use of the numerical path following method) will be compared for this example. To illustrate the difference between the two approaches, five starting points at the initial Pareto set for $\lambda = 0$ have been considered. Figure 4.23 visualizes the results for robust solution paths w. r. t. both concepts. One can observe that the computations that are based on the starting point

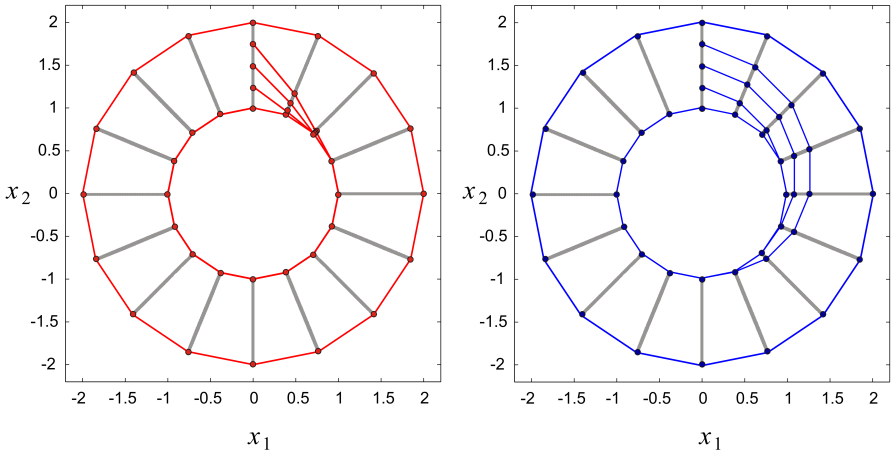

Figure 4.23: Results of the computation of λ-robust solution paths using the variational approach (on the left) and the numerical path following algorithm (on the right) for Example 4.18

$(0, 2)$ lead to the 'outer circle' which is the global maximum of both the variational problem and each subproblem occurring in the numerical path following method. As the concepts both only make use of necessary conditions, these maximal solutions are also feasible results. Moreover it can be seen that all other solution paths spiral towards the inner circle for both concepts. The paths computed by the Concept I (plot on the left) approach the inner circle faster than the paths generated with Concept II (plot on the right). To illustrate this effect, in Figure 4.24 the two solution paths presented in Figure 4.23 that both correspond to the same starting point $x_0^\star = (0, 1.75)^T$ are plotted. On the right of Figure 4.24 the lengths of all

the paths that have been visualized in Figure 4.23 are compared. Here one also can observe that the minimization of the respective variational integral indeed leads to shorter paths than the numerical path following approach.

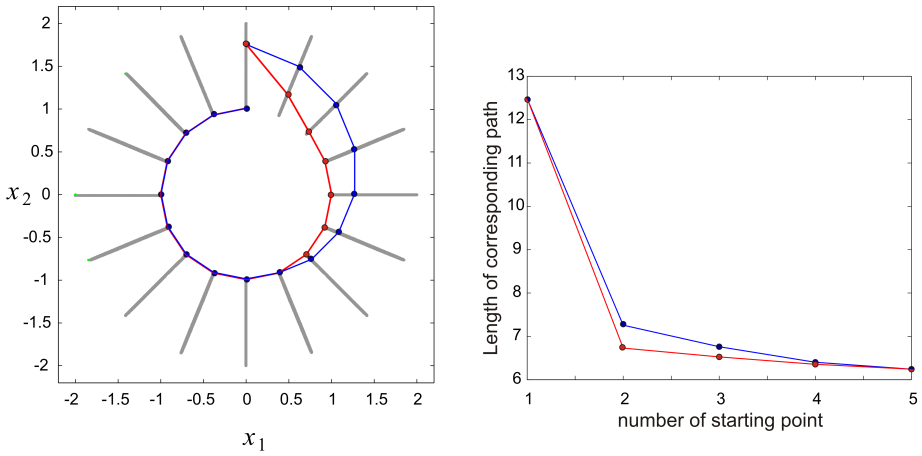

Figure 4.24: Comparison of the two concepts for one fixed starting point, and comparison of the lengths of the four exemplary paths that were considered in Example 4.18

The result that the paths w. r. t. Concept I are shorter than those w. r. t. Concept II in Example 4.18 is not amazing. If one neglects the numerical error that occurs from the discretization of the variational integral (4.30) and assumes that the computed solution indeed minimizes the variational integral, one can argue that the path w. r. t. Concept I is the shortest possible path connecting the fixed starting point with a free end point at $S_{\lambda_{\text{end}}}$. On the contrary, paths w. r. t. Concept II are computed by a local, step-wise greedy strategy. Thus, it can be expected that

$$\text{Path-I-Length}(x) \leq \text{Path-II-Length}(x)$$

for all $x \in P_{\text{discrete}}$, where $P_{\text{discrete}} = \{x_i^{start}\}_{i=1,\ldots,M}$ is a set of M starting points within the Pareto set for $\lambda = \lambda_{\text{start}}$.

As mentioned on page 89, the computation of constant solution paths is only possible by minimizing energy instead of length of the curve $x(\lambda)$. In this case, the consideration of fixed starting points leads to the variational integral

$$\begin{aligned} &\min_{\gamma} \int_{\lambda_{\text{start}}}^{\lambda_{\text{end}}} \frac{1}{2} \|x'(\lambda)\|_2^2 \, \mathrm{d}\lambda \\ &\text{s. t. } H_{\text{KT}}(\gamma(\lambda), \lambda) = 0 \end{aligned} \tag{4.41}$$

with the boundary conditions $x(\lambda_{\text{start}}) = x_0$, $t(\lambda_{\text{start}}) = t_0$ and the free end condition $x(\lambda_{\text{end}}) \in S_{\lambda_{\text{end}}}$.
If so, the discretization of the Euler-Lagrange equations is given as

$$\left(\mu_j^T \cdot \frac{\partial}{\partial t_j} H_{\text{KT}}(x_j, t_j, \lambda_j)\right)^T = 0 \ \forall\, j = 1, \ldots, N \tag{4.42}$$

$$H_{\text{KT}}(x_j, t_j, \lambda_j) = 0 \ \forall\, j = 1, \ldots, N \tag{4.43}$$

$$\left(\frac{x_{j+1} - 2x_j + x_{j-1}}{h^2}\right) - \left(\mu_j^T \cdot \frac{\partial}{\partial x_j} H_{\text{KT}}(x_j, t_j, \lambda_j)\right)^T = 0 \ \forall\, j = 1, \ldots, N\text{–}1 \tag{4.44}$$

$$-\left(\frac{x_N - x_{N-1}}{h^2}\right) - \frac{1}{2}\left(\mu_N^T \cdot \frac{\partial}{\partial x_N} H_{\text{KT}}(x_N, t_N, \lambda_{\text{end}})\right)^T = 0. \tag{4.45}$$

In the following, it is again assumed that $P_{\text{discrete}} = \{x_i^{start}\}_{i=1,\ldots,M}$ is a set of M starting points within the Pareto set for $\lambda = \lambda_{\text{start}}$, and that the step sizes in the computations w. r. t. Concept I and Concept II are chosen in the same way. As before, for Concept II the preimage space decision function is considered.

Assume that $x_0^\star \in P_{\text{discrete}}$. Assume further that for $x_j^\star \in \mathbb{R}^n$ there exist multipliers $t_j^\star \in \mathbb{R}^k$ and $\mu_j^\star \in \mathbb{R}^{n+1}$ such that equations (4.42) – (4.45) are true for each discretization point λ_j.

Let the length of the corresponding solution path $p_{\text{I}_{\text{energy}}} = (x_0^\star, x_1^\star, \ldots, x_N^\star)$ be defined as

$$\mathit{Path}\text{-I}_{\text{energy}}\text{-}\mathit{Length}(x_0^\star) = \|x_1^\star - x_0^\star\| + \|x_2^\star - x_1^\star\| + \ldots + \|x_N^\star - x_{N-1}^\star\|$$

(in analogue to the definition of the length of a λ-robust solution path w. r. t. Concept II in (4.28) on page 103).

Proposition 4.19 *Let $x_0^\star \in P_{\text{discrete}}$. Then,*

$$\mathit{Path}\text{-}I_{\text{energy}}\text{-}\mathit{Length}(x_0^\star) = 0 \ \Leftrightarrow \mathit{Path}\text{-}\mathit{II}\text{-}\mathit{Length}(x_0^\star) = 0.$$

Proof:

"⇒" Let $p_{\text{I}_{\text{energy}}} = (x_0^\star, x_1^\star, \ldots, x_N^\star)$ be a solution path that – together with the corresponding multipliers – solves equations (4.42) – (4.45) and has Path-I_{energy}-Length$(x_0^\star) = 0$.

$$\begin{aligned} \text{Path-I}_{\text{energy}}\text{-Length}(x_0^\star) &= 0 \\ \Leftrightarrow \|x_1^\star - x_0^\star\| + \|x_2^\star - x_1^\star\| + \ldots + \|x_N^\star - x_{N-1}^\star\| &= 0 \\ \Leftrightarrow x_0^\star = x_1^\star = \ldots &= x_N^\star \end{aligned}$$

Thus, any path which has Path-I_{energy}-Length zero is a path of constant values, $p_{\text{I}_{\text{energy}}} = (x_0^\star, x_0^\star, \ldots, x_0^\star)$.

By (4.43), there exist weight vectors $t_j^\star$, $j = 1, \ldots, N$ such that $x_0^\star$ solves the Kuhn-Tucker equations at each discretization point λ_j. Thus, after each predictor step of the numerical path following used in Concept II the Kuhn-Tucker equations are still satisfied. Moreover, the Euclidean distance between two successive points on the path $p_{\mathrm{I}_{\mathrm{energy}}}$ equals zero, and thus is minimal. Thus, no corrector steps of the numerical path following used in Concept II are performed.

$\Rightarrow$ The predictor-corrector algorithm used in Concept II (cf. page 102) generates the same path $p_{\mathrm{II}} = (x_0^\star, x_0^\star, \ldots, x_0^\star)$.

$\Rightarrow$ *Path*-II-*Length*$(x_0^\star) = 0$.

"$\Leftarrow$" Let $p_{\mathrm{II}} = (x_0^\star, x_1^\star, \ldots, x_N^\star)$ be a path that has been generated by the numerical path following algorithm of Concept II and assume that *Path*-II-*Length*$(x_0^\star) = 0$. In this case,

$$\begin{aligned} & \text{Path-II-Length}(x_0^\star) && = 0 \\ \Leftrightarrow\ & \|x_1^\star - x_0^\star\| + \|x_2^\star - x_1^\star\| + \ldots + \|x_N^\star - x_{N-1}^\star\| && = 0 \\ & \Leftrightarrow x_0^\star = x_1^\star = \ldots && = x_N^\star, \end{aligned}$$

and thus $p_{\mathrm{II}} = (x_0^\star, x_0^\star, \ldots, x_0^\star)$.
As p_{II} is a solution path w. r. t. Concept II, the necessary conditions (4.23) (cf. Proposition 4.12) are satisfied in each point on the path. As in this special case $d(x^\star) = 2(x^\star - x^\star) = 0$, equation (4.23) says that there exist multipliers $t_j^\star \in \mathbb{R}^k$ and $\mu_j^\star \in \mathbb{R}^{n+1}$, such that

$$-\left(\mu_j^\star\right)^T \cdot \frac{\partial}{\partial(x_j^\star, t_j^\star)} H_{KT}(x_j^\star, t_j^\star, \lambda_j) = 0 \ \forall j = 0, \ldots, N,$$

and that additionally the Kuhn-Tucker equations are satisfied in each point of p_{II}.

$\Rightarrow$ Equations (4.42) and (4.43) are satisfied. As $\frac{x^\star - 2x^\star + x^\star}{h} = 0$ and $\frac{x^\star - x^\star}{h^2} = 0$, Equations (4.44) und (4.45) are also satisfied.

$\Rightarrow$ p_{II} is a solution path which – together with the corresponding multipliers – solves the discretized Euler-Lagrange equations and has Path-$\mathrm{I}_{\mathrm{energy}}$-Length$(x^\star) = 0$.

□

4.4.3 Application: Robust Pareto Points in the Design of Integrated Circuits[4]

Integrated circuits are composed of a couple of standard cells that are multiply combined and repeated. Thus, one expects that an optimization of these standard cells causes a remarkable improvement of the entire integrated circuit. In [7] three different standard cells were considered: *inverters, NAND-gates and NOR-gates* (both in 65nm and 90nm technology).

These standard cells are a combination of the most basic elements, the *transistors*, that deliver the desired functionality. In Figure 4.25 the principal functioning of a metal-oxide semiconductor field-effect transistor (MOSFET) is sketched. One distinguishes between pMOS and nMOS transistors. In CMOS (complementary metal oxide semiconductor) technology, transistors of these two types are combined to reach the desired functionality (see [4] for an extensive introduction). The task of an *inverter* is to invert the incoming

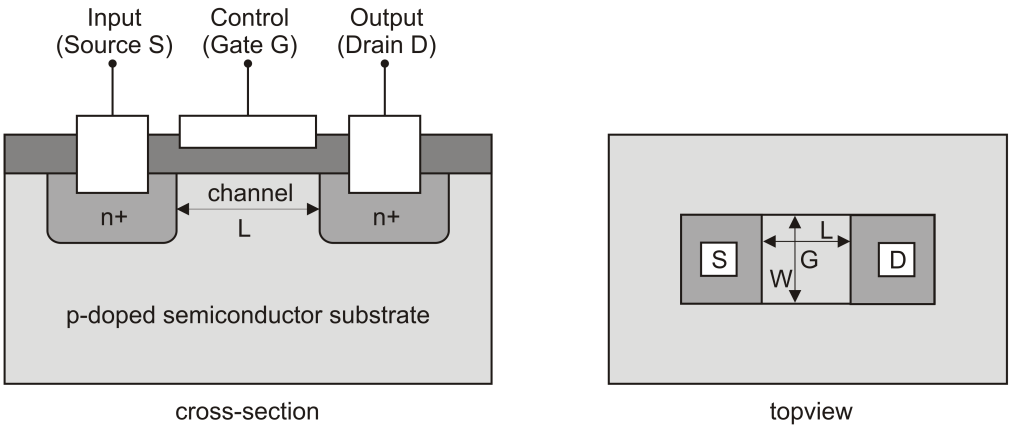

Figure 4.25: Schematic illustration of a metal-oxide semiconductor field-effect transistor (MOSFET) (cp. [4])

signal (1 is converted into 0 and vice versa). This can be reached by the combination of two transistors. The CMOS *NOR-gate* is composed of four transistors as shown in Figure 4.26. Its output is the negated value of the boolean function OR of two inputs. Similarly, the CMOS *NAND-gate* delivers the negated output of the boolean function AND of two inputs.

In [7, 96] it has been studied how to choose the width W of each transistor contained in inverters, NAND- and NOR-gates in order to guarantee a good

[4] in cooperation with the Department of System and Circuit Technology, Prof. Dr.-Ing. U. Rückert, Paderborn, especially Matthias Blesken, and in cooperation with Dominik Steenken, cf. [96, 7]

performance of these standard cells. Here, the length L is assumed to be as small as technically realizable.

Three different objectives are taken into account:

- maximization of the *noise margin* NM, defined as the maximal possible deviation of the input signal from the nominal value such that the input signal is still interpreted correctly,
- minimization of the *propagation delay* t_{pd}, defined as the time it takes that a change in the input signal is reflected in the output,
- and minimization of the *dynamic energy consumption* $E_{in} + E_{dyn}$ which sums up any energy losses that occur when the input signal changes.

These objective functions not only depend on the optimization variables given by the widths of the transistors but also on several process parameters (like doping) and runtime parameters (like temperature and supply voltage). In [96] different robustness concepts have been studied and applied to the case of varying temperatures and supply voltages. In this thesis only the results for the 65nm CMOS NOR-gate are summarized.

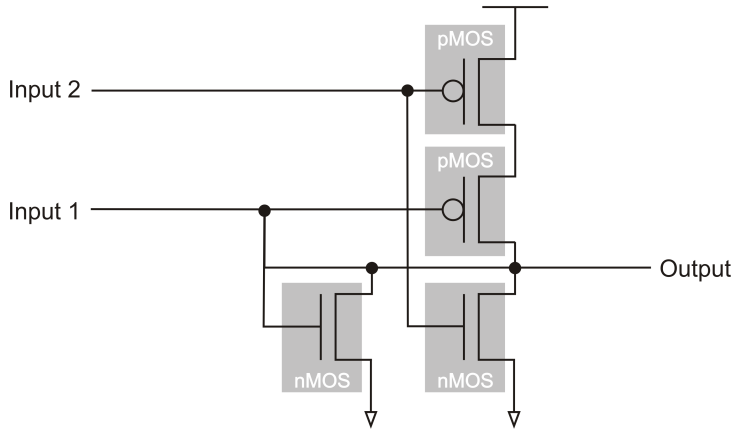

Figure 4.26: Net list of a CMOS NOR-gate (cp. [4])

A NOR-gate consists of two nMOS and two pMOS transistors (see Figure 4.26). In order to make the results comparable to values proposed by a commercial standard cell library, the widths of transistors of the same kind are supposed to be the same (in [96] also a multiobjective optimization was performed that allows different width of the pMOS transistors). Thus, two different widths denoted by W_p (width of each pMOS transistor) and W_n (width of each nMOS transistor) have to be optimized. The evaluation of

the three objective functions mentioned above is very time-consuming as it is performed using the circuit simulator HSPICE[5]. In order to reduce the computing time, in [96] the set-oriented subdivision and recovering techniques for the approximation of the Pareto set (cf. Section 3.2.3) have been parallelized. A compute server has been developed that allows the distribution of function evaluations to different clients. In Figure 4.27 the Pareto set of the NOR-gate with respect to the three objectives mentioned above is plotted both in parameter and objective space. Also in this figure the values proposed by the standard cell library are given both in parameter and objective space (black crosses). As one can observe they lie outside the Pareto front. Thus, the performance of this gate (and also several other gates as reported in [7, 96]) can be significantly improved by the multiobjective optimization approach.

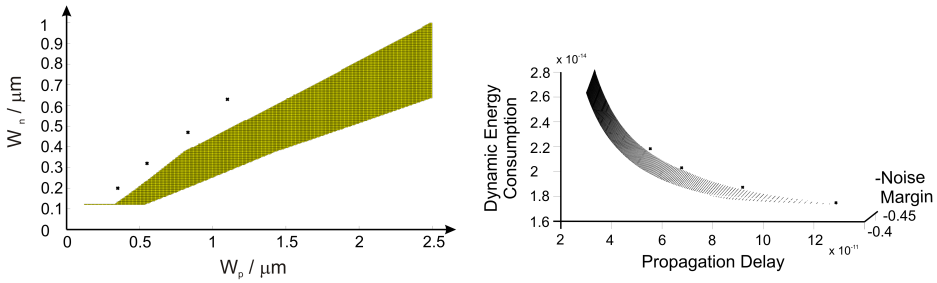

Figure 4.27: Results of the multiobjective optimization of the NOR-gate: the Pareto set (on the left) and the Pareto front (on the right) and the reference values from the standard cell library (black crosses) [96]

A study of the robustness of the Pareto optimal solutions with respect to changes in the external parameters temperature and supply voltage can help to support the decision which point to choose from the Pareto set for transistor fabrication. The numerical path following method for computing paths of mimimal length in preimage space described in Section 4.4.2 has been applied to this problem. In order to make the computation possible the three objective functions have been approximated by smoothing splines (cf. [96] for more details). In Figures 4.28 and 4.29 the results are visualized. Here, dark blue regions refer to Pareto points with small path lengths. The technical interpretation of the results is actually still an object of research.

[5]HSPICE is a commercial software tool for circuit simulation developed by Synopsys

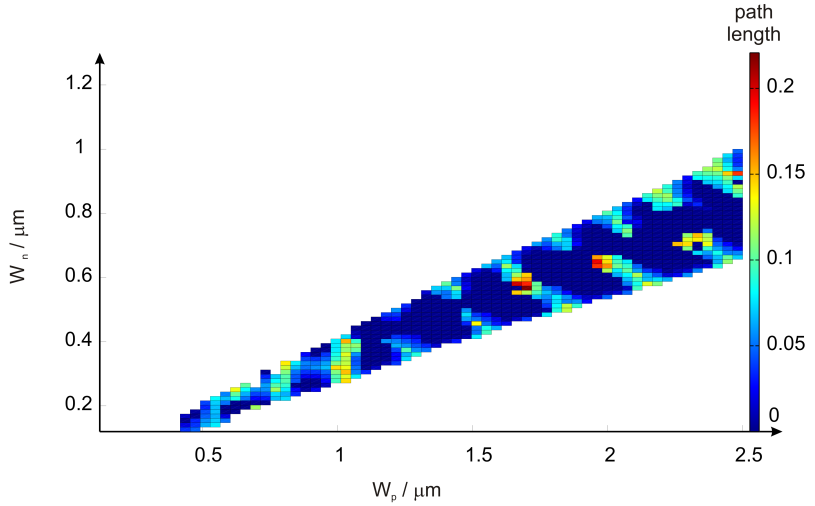

Figure 4.28: Visualization of the degree of robustness with respect to temperature changes from 10° C to 75° C [96]

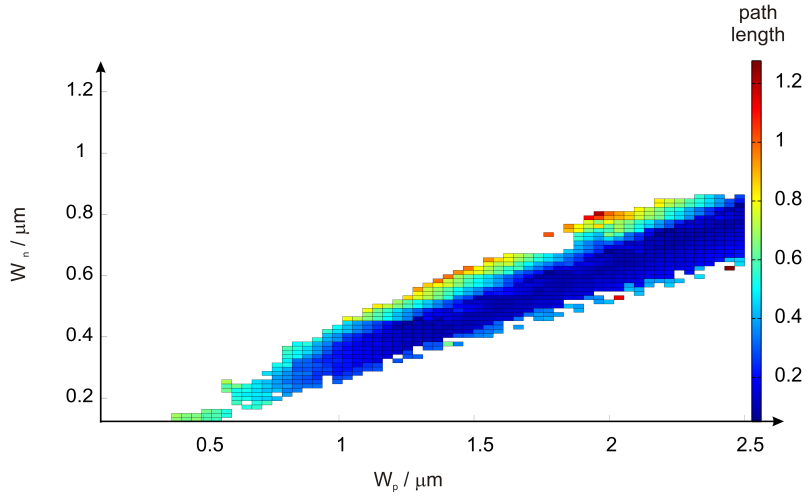

Figure 4.29: Visualization of the degree of robustness with respect to changes in the supply voltage within the inverval from 0.6 V to 1.8 V [96]

4.5 Origin of Dents in Parameter-dependent Pareto Fronts

In Section 3.3 dents in Pareto fronts have been defined motivated by the fact that these points cannot be computed by the weighted sums method. Also, dent border points have been defined. When considering parametric multiobjective optimization problems, naturally the question arises, how dents

and especially dent border points evolve. The Kuhn-Tucker equations (4.2) provide a necessary condition for Pareto optimality. Within this section this parametric system of equations will be analyzed in order to obtain results about the local behavior of the parameter-dependent Pareto fronts.

Solutions of parametric systems of equations have already been widely studied in bifurcation theory. The first paragraph of this section deals with the study of properties of dent border points. After summarizing some basics from bifurcation theory it will be proven that under certain assumptions dent border preimages are turning points of the Kuhn-Tucker equations. In the second part of this section, several new numerical examples for parametric multiobjective optimization problems in which dents occur are given.

Within this section it is assumed that the objective functions are at least twice continuously differentiable. Only points $x \in P_\lambda$ are considered for which the corresponding weight vector α is an element of $(0,1)^k$. Define $H^\alpha_{\mathrm{KT}} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ by

$$H^\alpha_{\mathrm{KT}}(x,\lambda) = \sum_{i=1}^{k} \alpha_i \nabla_x f_i(x,\lambda).$$

4.5.1 Properties of Dent Border Points

First, it will be shown that dent border preimages can be characterized as certain turning points of the Kuhn-Tucker equations. It has been shown in Section 3.3 that in a dent border point a zero eigenvalue of the Hessian of g_α occurs, where $g_\alpha(x,\lambda) = \sum_{i=1}^k \alpha_i f_i(x,\lambda)$. The Hessian of g_α equals $\frac{\partial}{\partial x} H^\alpha_{\mathrm{KT}}$, as $H^\alpha_{\mathrm{KT}}(x,\lambda) = \nabla_x g_\alpha(x,\lambda)$. Thus, the implicit function theorem is not applicable to the Kuhn-Tucker equations (with respect to x) in a dent border point. The study of the behavior of solutions in such cases is the main topic of bifurcation theory. Whenever the Jacobian with respect to x of a parametric system of equations is singular, the structure of the solution set may change. One possibility is that the system of equations has no solution before the singularity occurs, and two solutions afterwards (here, "before" and "afterwards" have to be understood in terms of the values of λ). In this case, the solution curve "turns" at the point $(x^\star, \lambda^\star)$, where the Jacobian with respect to x is singular. More formally, such a *turning point* – which sometimes is also called *saddle-node bifurcation* or *fold* in the literature – is defined as follows:

Definition 4.20 (Turning point (see [71])) *Consider the solutions of a nonlinear system of equations $H(x,\lambda) = 0$, where $H : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$. Assume that $(x^\star, \lambda^\star)$ is such a solution which satisfies*

(i) there exists $\phi^\star \in \mathbb{R}^N \backslash \{0\}$ *with* $\ker\left(\frac{\partial}{\partial x} H(x^\star, \lambda^\star)\right) = \operatorname{span}\{\phi^\star\}$,

(ii) $\frac{\partial}{\partial \lambda} H(x^\star, \lambda^\star) \notin \operatorname{im} \frac{\partial}{\partial x} H(x^\star, \lambda^\star)$.

Then, $(x^\star, \lambda^\star)$ *is called a* turning point.

Let $\psi^\star$ *be a left eigenvector of the zero eigenvalue of* $\frac{\partial}{\partial x} H(x^\star, \lambda^\star)$, *i. e.*

$$\psi^\star \frac{\partial}{\partial x} H(x^\star, \lambda^\star) = 0.$$

If in addition to (i) and (ii) $\psi^\star \left(\frac{\partial^2}{\partial x^2} H(x^\star, \lambda^\star)\phi^\star\phi^\star\right) \neq 0$, *then the point* $(x^\star, \lambda^\star)$ *is called a* simple turning point.

In Figure 4.30 an example of an equation whose solution curve includes a turning point is sketched. As one can observe, the solution curve turns in the point $(0, 0)$.

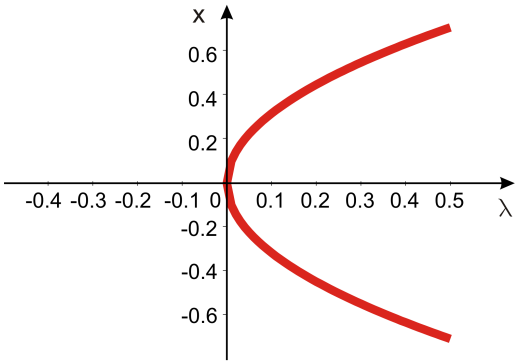

Figure 4.30: For the equation $H(x, \lambda) = x^2 - \lambda = 0$, $H : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, a turning point occurs in the point $(0, 0)$

Proposition 4.21 *Let* $P_\lambda \subseteq S_\lambda$ *be the Pareto set of a parametric multiobjective optimization problem* $\min F : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^k$ *with* $F(x, \lambda) = (f_1(x, \lambda), \ldots, f_k(x, \lambda))^T$. *Let* $x^\star \in P_{\lambda^\star}$ *be a simple dent border preimage. Let* $\alpha^\star$ *denote the weight vector corresponding to* $x^\star$ *and assume that the Jacobian* $H_{\mathrm{KT}}^{\alpha^\star\,\prime}(x, \lambda)$ *has full rank.*

Then, $(x^\star, \lambda^\star)$ *is a turning point of* $H_{\mathrm{KT}}^{\alpha^\star}(x, \lambda)$ *with respect to* λ.

Proof: It has been shown in [53] (see also Section 3.3) that dent border preimages $x^\star$ are solutions of the Kuhn-Tucker equations $H_{\mathrm{KT}}^{\alpha^\star}(x^\star, \lambda^\star) = 0$ in which $\frac{\partial^2}{\partial x^2} g_{\alpha^\star}(x^\star, \lambda^\star)$ is singular. Thus, there exists an eigenvector $\phi^\star$ of $\frac{\partial^2}{\partial x^2} g_{\alpha^\star}(x^\star, \lambda^\star)$ with

$$\left(\frac{\partial^2}{\partial x^2} g_{\alpha^\star}(x^\star, \lambda^\star) \right) \phi^\star = 0. \tag{4.46}$$

From the assumption that the dent border preimage is simple, i. e. exactly one eigenvalue of $\frac{\partial^2}{\partial x^2} g_{\alpha^\star}(x^\star, \lambda^\star)$ equals zero (cf. Definition 3.11), it directly follows that

$$\dim \ker \left(\frac{\partial^2}{\partial x^2} g_{\alpha^\star}(x^\star, \lambda^\star) \right) = 1. \tag{4.47}$$

As $H_{\mathrm{KT}}^{\alpha}(x, \lambda) = \nabla_x g_\alpha(x, \lambda)$, and thus

$$\frac{\partial}{\partial x} H_{\mathrm{KT}}^{\alpha}(x, \lambda) = \frac{\partial}{\partial x} \left(\nabla_x g_\alpha(x, \lambda) \right) = \frac{\partial^2}{\partial x^2} g_\alpha(x, \lambda),$$

(4.46) is equivalent to

$$\frac{\partial}{\partial x} H_{\mathrm{KT}}^{\alpha^\star}(x^\star, \lambda^\star) \varphi^\star = 0$$

and (4.47) is the same as

$$\dim \ker \left(\frac{\partial}{\partial x} H_{\mathrm{KT}}^{\alpha^\star}(x^\star, \lambda^\star) \right) = 1.$$

Thus, property (i) of Definition 4.20 is proven for $H(x, \lambda) = H_{\mathrm{KT}}^{\alpha^\star}(x, \lambda)$.

Property (ii) of Definition 4.20 directly follows from the assumption that the Jacobian ${H_{\mathrm{KT}}^{\alpha^\star}}'(x, \lambda)$ has full rank, i. e. rank n: if the vector $\frac{\partial}{\partial \lambda} H_{\mathrm{KT}}^{\alpha^\star}(x^\star, \lambda^\star)$ were in the image of the $n \times n$-matrix $\frac{\partial}{\partial x} H_{\mathrm{KT}}^{\alpha^\star}(x^\star, \lambda^\star)$, then the rank of the Jacobian ${H_{\mathrm{KT}}^{\alpha^\star}}'(x, \lambda)$ were $n - 1$, in contradiction to the assumption. Thus, it has to be true that $\left(\frac{\partial}{\partial \lambda} H_{\mathrm{KT}}^{\alpha^\star}(x^\star, \lambda^\star) \right) \notin \operatorname{im} \frac{\partial}{\partial x} H_{\mathrm{KT}}^{\alpha^\star}(x^\star, \lambda^\star)$.

To sum up, both properties given in Definition 4.20 are satisfied, and thus $(x^\star, \lambda^\star)$ is a turning point of $H_{\mathrm{KT}}^{\alpha^\star}(x, \lambda)$ with respect to λ. □

Remark 4.22 Using the notation of the proof of Proposition 4.21 one observes that the matrix $\frac{\partial}{\partial x} H_{\mathrm{KT}}^{\alpha^\star}$ is symmetric, as it is the Hessian of the weighted sums function g_α. Thus, $\psi^\star = (\phi^\star)^T$ is a left eigenvector of $\frac{\partial}{\partial x} H_{\mathrm{KT}}^{\alpha^\star}(x^\star, \lambda^\star)$. It follows that, if additionally to (i) and (ii) of Definition 4.20

$$(\phi^\star)^T \left(\frac{\partial^2}{\partial x^2} H_{\mathrm{KT}}^{\alpha^\star}(x^\star, \lambda^\star) \right) \phi^\star \phi^\star \neq 0,$$

then $(x^\star, \lambda^\star)$ is a simple turning point of $H_{\mathrm{KT}}^{\alpha^\star}(x, \lambda)$ with respect to λ.

To sum up, a dent border preimage can be obtained by solving the system of equations

$$\begin{aligned} H_{\mathrm{KT}}^{\alpha^\star}(x, \lambda) &= 0 \\ \frac{\partial}{\partial x} H_{\mathrm{KT}}^{\alpha^\star}(x, \lambda) \cdot \phi &= 0 \\ l^T \phi - 1 &= 0 \end{aligned} \tag{4.48}$$

with an arbitrary but fixed vector $l \in \mathbb{R}^n$ which satisfies $l^T \phi^\star \neq 0$ and has non-zero entries, and $x, \phi \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$. In the literature, this system of equations is also called the *extended system* of $H_{\mathrm{KT}}^{\alpha^\star}(x, \lambda)$ (cf. [71]).

Remark 4.23 A family of dent border preimages can be obtained by solving

$$\begin{aligned} \nabla_x g_\alpha(x, \lambda) &= 0 \\ \sum_{i=1}^{k} \alpha_i - 1 &= 0 \\ \frac{\partial^2}{\partial x^2} g_\alpha(x, \lambda) \cdot \phi &= 0 \\ l^T \phi - 1 &= 0 \end{aligned} \tag{4.49}$$

with an arbitrary fixed vector $l \in \mathbb{R}^n$ which satisfies $l^T \phi^\star \neq 0$ and has non-zero entries, $x, \phi \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$ and $\alpha \in \mathbb{R}^k$ with $\alpha_i > 0 \; \forall i = 1, \ldots, k$.

4.5.2 Examples for Parametric Multiobjective Optimization Problems with Dents

In the following, several new examples for parametric multiobjective optimization problems are presented. They all have in common that the corresponding Pareto fronts contain dents for specific values of the external parameter λ. Moreover, under the variation of λ, dents originate or vanish (cf. Examples 4.24 and 4.26), or dents double or merge (cf. Example 4.25).

Example 4.24 Consider the two objectives

$$\begin{aligned} f_1(x_1, x_2, \lambda) &= \frac{1}{2}(\sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} + x_1 - x_2) + \lambda \cdot e^{-(x_1 - x_2)^2} \\ f_2(x_1, x_2, \lambda) &= \frac{1}{2}(\sqrt{1 + (x_1 + x_2)^2} + \sqrt{1 + (x_1 - x_2)^2} - x_1 + x_2) + \lambda \cdot e^{-(x_1 - x_2)^2}. \end{aligned}$$

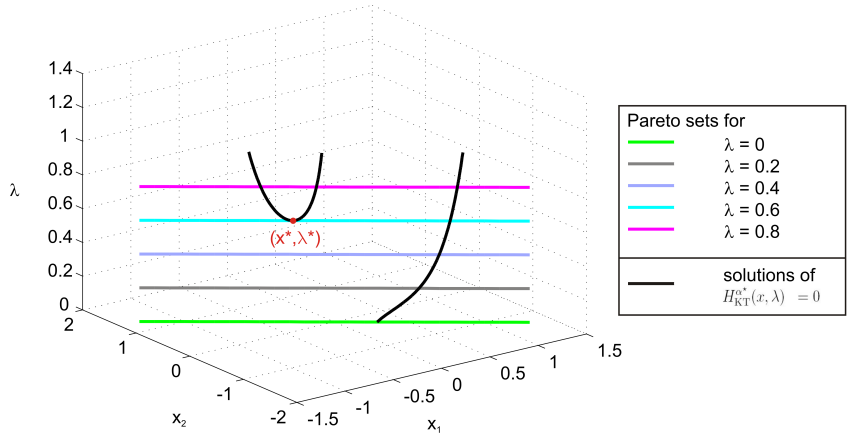

Figure 4.31: Visualization of some Pareto sets and the solution curve of $H_{\mathrm{KT}}^{\alpha^\star}(x,\lambda) = 0$ (black) for the dent border preimage $x^\star$ (red) for Example 4.24

In Figure 4.31 the Pareto sets of these two objectives for $(x_1, x_2) \in [-1.3, 1.3]^2$ are plotted for different values of the external parameter λ: for $\lambda = 0$ (green), $\lambda = 0.2$ (gray), $\lambda = 0.4$ (light blue), $\lambda = 0.6$ (cyan), and for $\lambda = 0.8$ (magenta).

In the same figure, an example for a λ-dependent solution path of the Kuhn-Tucker equations with a fixed weight vector $\alpha^\star \approx (0.25, 0.75)$ which corresponds to the dent border preimage $(x_1^\star, x_2^\star) \approx (-0.28, 0.28)$ of the dent border point $(y_1^\star, y_2^\star) \approx (1.23, 1.79)$, located on the Pareto front for $\lambda^\star = 0.6$, is visualized for $\lambda \in [0, 1]$ (black paths). The paths have been computed with the help of the software package AUTO2000 [23]. As one can observe, $y^\star$ is indeed a simple turning point of $H_{\mathrm{KT}}^{\alpha^\star}(x, \lambda)$ with respect to λ. Figure 4.32 shows the same results in objective space.

In Figure 4.33 the entire curve of dent border points in objective space is plotted. To compute this λ-dependent solution curve (red), again the software package AUTO2000 has been used. One can observe that the point $(y_1^\star, y_2^\star, \lambda^\star) = (1.25, 1.25, 0.25)$ (with the corresponding weight vector $\alpha^\star = (0.5, 0.5)$), marked by a black dot, is specific: in this point a dent originates, i. e. for $\lambda < \lambda^\star$ the Pareto front contains no dent whereas for $\lambda > \lambda^\star$ a dent is contained in the Pareto front.

In Figure 4.34 the solutions of $H_{\mathrm{KT}}^{\alpha^\star}(x, \lambda) = 0$ with $\alpha^\star = (0.5, 0.5)$, i. e. the λ-dependent path containing the specific point in which a non-dent point changes into a dent point, are visualized for $\lambda \in [0, 1]$. One can observe that

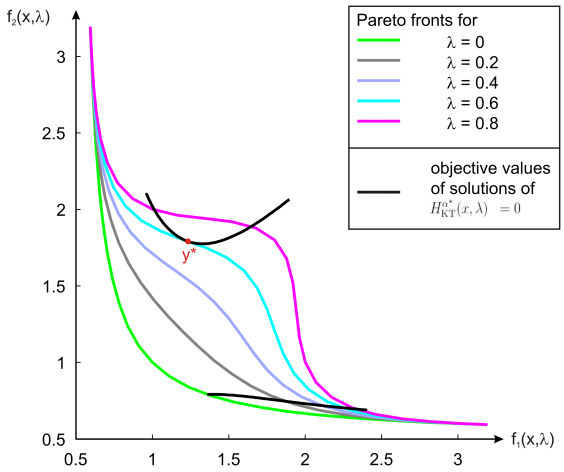

Figure 4.32: Visualization of some Pareto fronts and the image of the solution curve of $H_{\mathrm{KT}}^{\alpha^\star}(x,\lambda) = 0$ (black) for a dent border point $y^\star$ (red) for Example 4.24

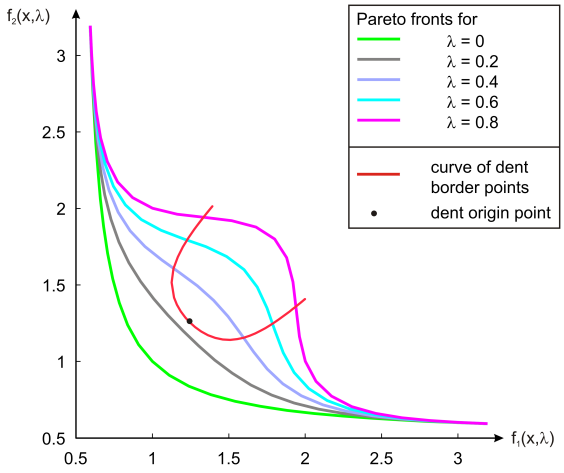

Figure 4.33: Some Pareto fronts, the curve of dent border points (red) and the point in which the dent originates (black dot) for Example 4.24

a pitchfork bifurcation[6] occurs in this point. Figure 4.35 shows the same results in objective space.

[6]The definition and statements about properties of pitchfork bifurcations can be found in [42] and [105], for example.

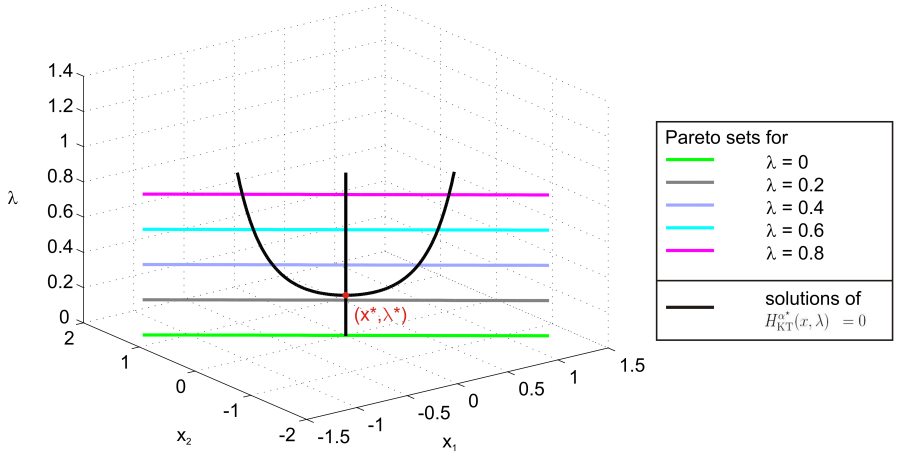

Figure 4.34: Visualization of some Pareto sets and the λ-dependent solution curve of $H_{\mathrm{KT}}^{\alpha^\star}(x,\lambda) = 0$ with $\alpha^\star = (0.5, 0.5)$ (black) for Example 4.24

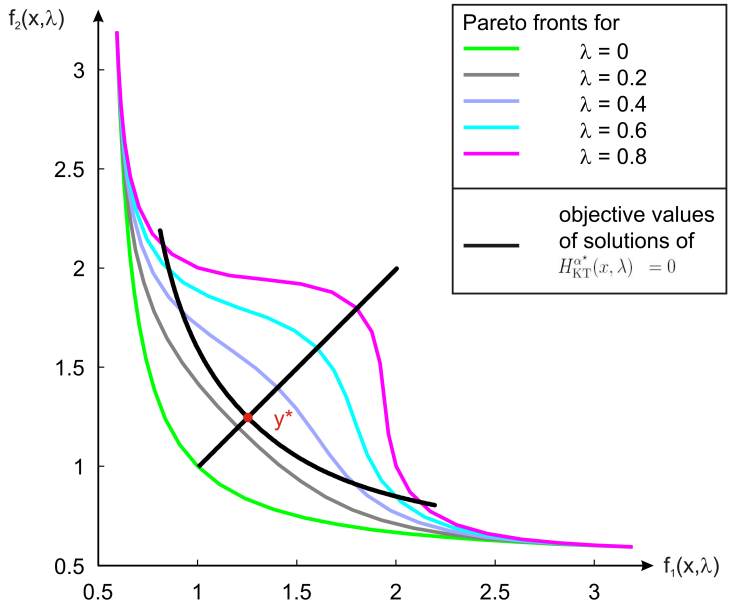

Figure 4.35: Visualization of some Pareto fronts and the image of the solution curve of $H_{\mathrm{KT}}^{\alpha^\star}(x,\lambda) = 0$ with $\alpha^\star = (0.5, 0.5)$ (black) for Example 4.24

Example 4.25 Consider the two objectives

$$\begin{aligned} f_1(x_1, x_2, \lambda) &= \sqrt{1+x_1^2} + \sqrt{1+x_2^2} + e^{-(x_2-\lambda)^2} + e^{-(x_2+\lambda)^2} - x_2 \\ f_2(x_1, x_2, \lambda) &= \sqrt{1+x_1^2} + \sqrt{1+x_2^2} + e^{-(x_2-\lambda)^2} + e^{-(x_2+\lambda)^2} + x_2. \end{aligned}$$

In Figure 4.36 the Pareto fronts which result from the minimization of these two objectives are plotted for different values of λ. As one can observe the Pareto front contains one dent for $\lambda = 0.4$, for example. Under the variation of λ it changes into two dents (cp. for instance $\lambda = 0.8$). In between, there is a specific point in which the dent splits up into two dents, which is given by $(x_1, x_2, \lambda) \approx (0, 0, 0.5716)$ with the corresponding weight vector $\alpha^\star = (0.5, 0.5)$. In Figure 4.37 the solutions of the Kuhn-Tucker equations for the fixed weight vector $\alpha^\star = (0.5, 0.5)$ are sketched. One can observe that in the point where the dent splits up into two dents a pitchfork bifurcation occurs.

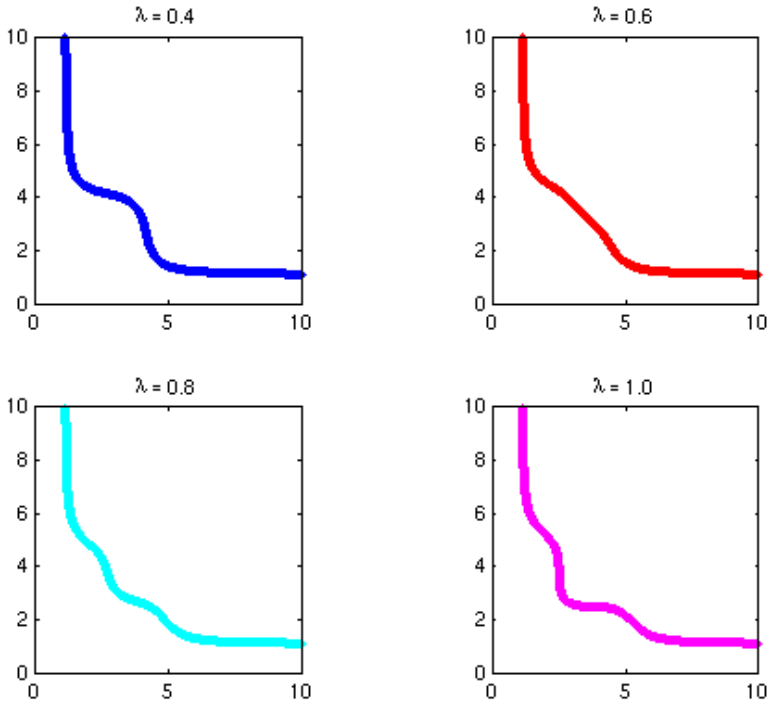

Figure 4.36: Pareto fronts for the multiobjective optimization problem given in Example 4.25 for different values of λ

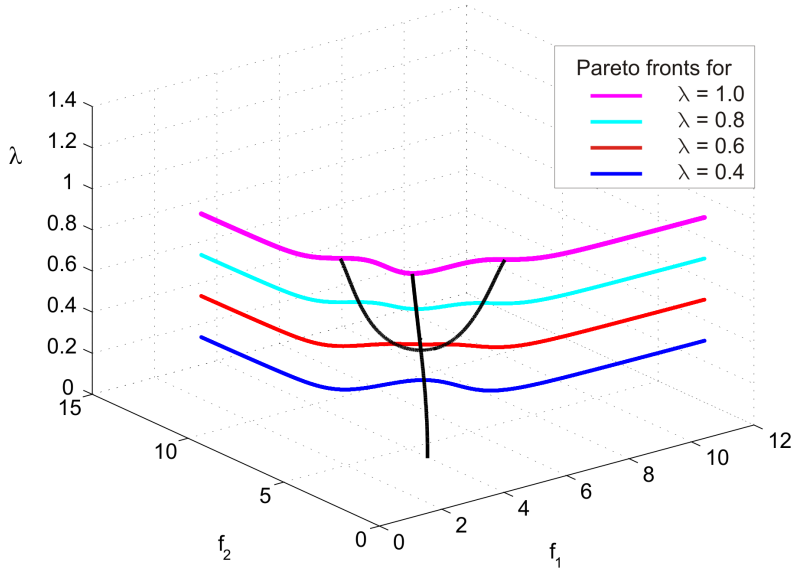

Figure 4.37: Pareto fronts and solutions of the Kuhn-Tucker equations for $\alpha^\star = (0.5, 0.5)$ in (f_1, f_2, λ)-space for Example 4.25

Example 4.26 Consider the three objectives

$$\begin{aligned} f_1(x_1,x_2,x_3,\lambda) &= \sqrt{1+x_1^2}+\sqrt{1+x_2^2}+\sqrt{1+x_3^2}+\lambda\cdot e^{-(x_2^2+x_3^2)}+\sqrt{2}x_2 \\ f_2(x_1,x_2,x_3,\lambda) &= \sqrt{1+x_1^2}+\sqrt{1+x_2^2}+\sqrt{1+x_3^2}+\lambda\cdot e^{-(x_2^2+x_3^2)}-\frac{\sqrt{2}}{2}x_2+\sqrt{\frac{3}{2}}x_3 \\ f_3(x_1,x_2,x_3,\lambda) &= \sqrt{1+x_1^2}+\sqrt{1+x_2^2}+\sqrt{1+x_3^2}+\lambda\cdot e^{-(x_2^2+x_3^2)}-\frac{\sqrt{2}}{2}x_2-\sqrt{\frac{3}{2}}x_3. \end{aligned}$$

Figure 4.38 shows the Pareto fronts which result when minimizing these three objectives for $x \in [-2, 2]^3$ for different values of λ. One can observe that under the variation of λ a dent originates.

Remark 4.27 It has been observed in Examples 4.24 and 4.25 that pitchfork bifurcations occur in those points where – under the variation of λ – a dent originates in the Pareto front PF_λ, or a dent splits up into two dents, respectively. Pitchfork bifurcations typically occur if the system of equations $H(x, \lambda) = 0$, in our case $H_{\mathrm{KT}}^{\alpha^\star}(x, \lambda) = 0$, includes a symmetry of the form

$$H(Sx, \lambda) = SH(x, \lambda), \tag{Z2}$$

where S is a suitable symmetry matrix with $S \neq \mathbb{1}$ and $S^2 = \mathbb{1}$ (cf. [105]).

In the examples mentioned above, indeed symmetries occur. The Kuhn-Tucker equations of the objective functions given in Example 4.24 have a

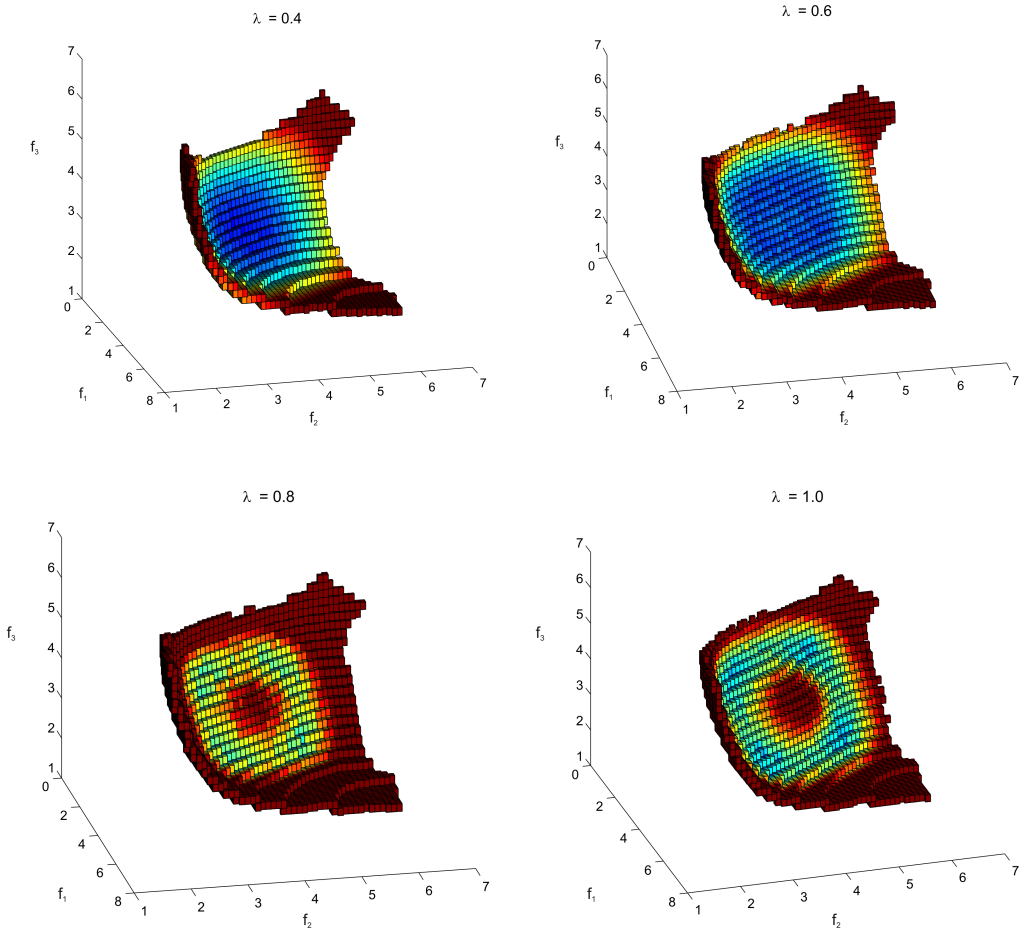

Figure 4.38: Pareto fronts for the multiobjective optimization problem given in Example 4.26 for different values of the parameter λ

$\mathbb{Z}_2$-symmetry for $\alpha^\star = (0.5, 0.5)$. In this case, possible symmetry matrices are given as

$$S_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \text{ and } S_2 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}.$$

The Kuhn-Tucker equations of the objective functions given in Example 4.25 satisfy the symmetry condition (Z2) with

$$S_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \text{ if } \alpha^\star = (0.5, 0.5), \text{ and}$$

$$S_2 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \text{ independent of the weight vector } \alpha.$$

The Kuhn-Tucker equations for the objective functions given in Example 4.26 also satisfy the symmetry condition (Z2) with

$$S = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

for arbitrary weight vectors $\alpha^\star$.

Chapter 5

Conclusion and Outlook

In this thesis, four main areas have been addressed:

1. **Robustness concepts for parametric multiobjective optimization problems**

   The main issue of this thesis was the study of parametric optimization problems. Two new robustness concepts have been developed that allow to find out how Pareto optimal points evolve under the variation of an external, real parameter. In contrast to hitherto existing approaches, it is requested that the solutions should stay Pareto optimal for the original objective functions w. r. t. the perturbed parameter values. Especially for technical systems in which the parameter configuration can be adjusted during operation time and the external parameter can be measured, the new robustness concepts offer big advantages. For instance, assume that there exists a point on a Pareto front which stays constant under variations of the external parameter. This robust Pareto point can be computed by the robustness concepts proposed in this thesis. During operation time it would be possible to switch parameter-dependently between solutions of the computed path such that the values of the objective functions stay constant.

   In the first concept presented in this thesis, robustness is defined by use of the classical calculus of variations. A variational integral is minimized which computes the length of the solution curve under the constraint that the curve lives on the parameter-dependent solution set of the Kuhn-Tucker equations. To allow numerical computations, a discrete version of the Euler-Lagrange equations, which give a necessary condition for optimality of variational integrals, is derived. By this approach a nonlinear system of equations is obtained whose solution gives a discrete approximation of the λ-robust solution curve.

   The second concept is of local nature. Here, the length of a solution path is not required to be minimal over the entire interval the external parameter is assumed to live in. Instead, the path consists of points

that locally change as little as possible in x-space under the variation of the external parameter λ. Alternatively, solution paths that change as little as possible in objective space can be described by a similar approach. To compute these paths, numerical path following methods are used. They allow to approximate solution curves of nonlinear systems of equations that depend on one external parameter. The Kuhn-Tucker equations do not only depend on the optimization parameters and the external parameter but also on the weight vector. The main challenge in applying numerical path following methods to the Kuhn-Tucker equations lies in the treatment of this weight vector. It characterizes the direction into which a solution path moves under the variation of the external parameter. In this thesis, additional equations that define suitable directions have been derived. In order to attack this problem, firstly two fixed Pareto sets P_{λ_0} and P_{λ_1} have been considered. A constrained nonlinear optimization problem has been formulated whose solution is a point on P_{λ_1} which has minimal Euclidean distance to a fixed Pareto point on P_{λ_0}. Analogously, a constrained nonlinear optimization problem has been formulated which minimizes the distance between the Pareto fronts. In order to obtain numerical results, again a necessary condition is used which also leads to a nonlinear system of equations.
For the numerical approximation of an entire solution path a predictor-corrector method is used. Here, the new equations that define the path direction are used in combination with the Kuhn-Tucker equations in each corrector step. It is required for the predictor-corrector method that the Jacobian matrix of this combined system of equations is regular. In this work, the structure of this special Jacobian has been studied. It has been proven that the Jacobian is indeed regular if a standard assumption for the underlying minimization problem is satisfied. The new robustness concept has been illustrated for several numerical examples and a technical application arising in the design of integrated circuits.

To make both new robustness concepts comparable, the variational problem of Concept I has been reformulated for fixed starting points on the initial Pareto set. A comparison of both concepts shows that typically the length of solution paths w. r. t. Concept I is less or equal to the length of paths obtained with Concept II. It has been shown that if there exists a substationary point $x^\star$ which does not change under the variation of the external parameter λ, $\lambda \in [\lambda_{\text{start}}, \lambda_{\text{end}}] \subset \mathbb{R}$ (i. e. the solution path $x(\lambda)$ is constant), this path is a solution with respect

to both the variational approach using energy minimization and the numerical path following approach for the determination of λ-robust substationary points. However, in general it highly depends on the application under consideration which concept to choose for the study of robustness of Pareto optimal solutions.

There are various engineering applications including dependencies on external parameters in which robustness with respect to parameter changes in a multiobjective sense is required. Within the Collaborative Research Centre 614 "Self-optimizing concepts and structures in mechanical engineering" especially one application has turned out to be of great interest, for which robust solutions will be studied in the future. It concerns an X-by-wire vehicle called "Chamäleon".[1] In this vehicle any of the four wheels can be steered individually. As the vehicle is not very heavy, the mass of the driver has a noticeable influence on the behavior of the system. The distribution of the forces acting on the wheels has to be optimized with respect to several objectives. In addition, it is desired to obtain solutions that are robust with respect to changes of the drivers' masses. Another value that influences the system is the friction coefficient. It is worth investigating robustness against this parameter, too. In both cases, a parametric multiobjective optimization problem arises that depends on additional equality constraints which stem from the fact that the sum of forces and momenta has to equal a constant value. Inspired by this example, in the future the robustness techniques developed in this thesis will also be extended to the case of constrained parametric multiobjective optimization problems.

2. **Time-dependent multiobjective optimization problems**

Motivated by the multiobjective optimization of a linear drive, the special case of time-dependent multiobjective optimization problems has also been studied. In order to follow solutions over time again numerical path following methods have been used. The optimization procedure proposed in this thesis consists of several recurring steps. First, the initial Pareto set is computed, one solution is chosen and the corresponding weight vector is set fixed. The decision which solution to choose is based on the current situation, i. e. for example on pa-

[1] This vehicle is developed at the Chair of Control Engineering and Mechatronics, University of Paderborn

rameters which are not modeled as optimization parameters but have an influence on the properties of the technical system. Secondly, a solution path over time consisting of substationary points is computed by solving the Kuhn-Tucker equations for this fixed weight vector with a classical predictor-corrector method. Depending on the situation, during operation time it is decided if the computation of a new, entire Pareto set is necessary, and the steps are repeated.

This new approach developed within this thesis allows to use a multiobjective optimization approach for the generation of solution trajectories during operation time (for appropriate applications). It has been applied to the operating point assignment of a linear drive and was implemented and tested at the test plant. For future considerations it is interesting to study the influence of additional external parameters like a changing air gap on the time-dependent solution trajectories. It is worth investigating whether the robustness concepts mentioned above can be extended to the treatment of solution trajectories.

3. **Dents in Pareto fronts**

In this thesis also the occurrence of *dents* in Pareto fronts has been studied. A formal definition of a dent has been given. Points at the border of a (complete) dent have a significant property. In these points a zero eigenvalue of the Hessian of the weighted sum of the objectives occurs. Thus, dent border points are solutions of a certain system of equations. Given a sufficiently smooth multiobjective optimization problem it is possible to find out if dent border points, and thus also possibly dents, occur in the Pareto front by solving this system of equations. Consequently, information about the geometry of the Pareto front can be obtained without computing the entire Pareto set. This information can for example serve as a criterion for the choice of the algorithm one wants to use for solving the multiobjective optimization problem.

Based on theoretical results from bifurcation theory, parameter-dependencies in multiobjective optimization problems have been studied in this thesis. It has been proven that dent border points are turning points of the Kuhn-Tucker equations with a fixed weight vector corresponding to the dent border point. Several examples for parametric multiobjective optimization problems have been constructed in which dents occur. In the future, the question will be addressed what happens if a dent originates or vanishes under the variation of the external

parameter. The examples given at the end of Section 4.5 lead to the conjecture that in this case pitchfork bifurcations of the Kuhn-Tucker equations occur. However, the theoretical analysis of this statement will be addressed in future work.

4. Applications from mechatronics including multiobjective optimization problems

In the third chapter of this thesis, in addition to the introduction into multiobjective optimization also several engineering applications are presented which have been studied in cooperation with the respective chairs. These applications are related to the RailCab system which serves as a demonstrator for the self-optimizing concepts developed within the CRC 614. Besides the basic study of a model for the operating point assignment of the linear drive which was the motivation for developing the concepts for time-dependent multiobjective optimization described above, two more applications have been considered.

The first one concerns a multiobjective optimization of the hybrid energy storage. From the mathematical point of view, the challenge in this application was to compute an operation strategy that distributes the power flows to the respective storage devices in an optimal way during a drive of the vehicle. In the approach presented in this work the time interval under consideration has been discretized, and the battery currents at the corresponding points were optimized with respect to two objectives. Alternatively, it is possible to attack the same problem with optimal control algorithms like DMOC[2]. The advantage would be that a Pareto optimal profile for the battery current can be computed also over longer time intervals with a fine discretization. The disadvantage is that only single Pareto optimal trajectories can be computed by optimizing a weighted sum of the objectives. In the future, the combination of DMOC with concepts for multiobjective optimization will be investigated.

The second application dealt with the generation of Pareto optimal reference trajectories for the guidance of the RailCab, i. e. an optimal control problem occurred. The mathematical model of the guidance module considered in this work was differentially flat, which means that the inputs and states of the system can be represented as a function of the outputs and a finite number of their derivatives w. r. t. time.

[2]Discrete Mechanics and Optimal Control, see [58]

It has been shown in this work how the multiobjective optimal control problem can be transformed into a classical multiobjective optimization problem which can be solved with existing algorithms. Furthermore, it has been described how self-optimization of the guidance module can be performed during operation time.

So far, several subsystems of the RailCab system have been studied independently from each other. Within the total system these subsystems partly depend on the same parameters. For example, the energy supply provided by the hybrid energy storage system is important for the operating point assignment of the linear drive (how much energy has to be transferred into the hybrid energy storage system to enable the system to drive?) and the guidance module (how much energy may the hydraulic actuators use?). Thus, at present a model that includes the connections between the subsystems within the RailCab is investigated within the CRC 614. Based on this model, it will be of interest to integrate the concepts for the treatment of multiobjective optimization problems with parameter-dependencies into a hierarchical framework.

To sum up, this thesis mainly contributes to the field of parametric multiobjective optimization by defining new robustness concepts, by studying bifurcation phenomena in parametric Pareto fronts, and by providing an algorithm that allows the computation of time-dependent Pareto optimal solutions possibly even during operation time. Moreover, several engineering applications in which multiobjective optimization problems arise have been presented.

Bibliography

[1] P. Adelt, J. Donoth, J. Gausemeier, J. Geisler, S. Henkler, S. Kahl, B. Klöpper, A. Krupp, E. Münch, S. Oberthür, C. Paiz, M. Porrmann, R. Radkowski, C. Romaus, A. Schmidt, B. Schulz, H. Vöcking, U. Witkowski, K. Witting, and O. Znamenshchykov, *Selbstoptimierende Systeme des Maschinenbaus - Definitionen, Anwendungen, Konzepte*, HNI-Schriftenreihe 234, Bonifatius GmbH, 2009. [2, 30, 31, 48]

[2] E. L. Allgower and K. Georg, *Numerical Continuation Methods*, Springer, 1990. [5, 25, 28, 30]

[3] Arakawa, Billaut, Coello Coello, Ehrgott, Gandibleux, Göpfert, Jones, Kuk, Miettinen, Nakayama, Romero, Sakawa, Steuer, T'Kindt, Tamiz, Tanino, and Yun, *Multiple Criteria Optimization. State of the Art Annotated Bibliographic Surveys*, International Series in Operations Research and Management Science 52, Kluwer Academic Publishers, Boston, 2002. [34, 35, 38]

[4] R. J. Baker, *CMOS: Circuit Design, Layout, and Simulation*, Wiley-Interscience, second ed., 2008. [114, 115]

[5] B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer, *Non-Linear Parametric Optimization*, Birkhäuser, 1983. [25]

[6] G. Bigi and M. Castellani, *Uniqueness of KKT multipliers in multiobjective optimization*, Applied Mathematics Letters **17**, no. 11 (2004), pp. 1285–1290. [35]

[7] M. Blesken, U. Rückert, D. Steenken, K. Witting, and M. Dellnitz, *Multiobjective Optimization for Transistor Sizing of CMOS Logic Standard Cells Using Set-Oriented Numerical Techniques*, in Proceedings of the 27th Norchip Conference, 2009. [114, 116]

[8] J. Böcker, B. Schulz, T. Knoke, and N. Fröhleke, *Self-optimization as a framework for advanced control systems*, in Int. Electronics Conference (IECON), Paris, Frankreich, 2006. [73]

[9] O. Bolza, *Vorlesungen über Variationsrechnung*, Chelsea Publishing Company, New York, USA, 1909. [7, 19]

[10] J. BRANKE, *Robustness and reliability in EMO.* Talk at Dagstuhl Seminar 09041, Dagstuhl, Germany, 2009. [6, 76]

[11] J. BRANKE, *Creating robust solutions by means of evolutionary algorithms*, in Parallel Problem Solving from Nature, LNCS 1498, Springer-Verlag, 1998, pp. 119–128. [77]

[12] C. A. COELLO COELLO, G. LAMONT, AND D. V. VELDHUIZEN, *Evolutionary Algorithms for Solving Multi-Objective Optimization Problems*, Springer, Berlin, Germany, second ed., 2007. [1, 38]

[13] I. DAS AND J. DENNIS, *A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems*, Structural Optimization **14**, no. 1 (1997), pp. 63–69. [8, 39]

[14] I. DAS AND J. E. DENNIS, *Normal boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems*, SIAM J. Optim. **8** (1998), pp. 631–657. [1, 38]

[15] K. DEB, *Multi-Objective Optimization using Evolutionary Algorithms*, Wiley-Interscience Series in Systems and Optimization, John Wiley & Sons, Chichester, England, 2001. [1, 38]

[16] K. DEB, S. GRECO, K. MIETTINEN, AND E. ZITZLER, *09041 summary – hybrid and robust approaches to multiobjective optimization*, Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2009, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany. [6, 76]

[17] K. DEB AND H. GUPTA, *Introducing robustness in multi-objective optimization*, Evolutionary Computation **4**, no. 14 (2006), pp. 463–494. [77, 78, 79]

[18] A. DELL'AERE, *Multi-objective optimization in self-optimizing systems*, in Proceedings of the IEEE 32nd Annual Conference on Industrial Electronics (IECON), 2006, pp. 4755 – 4760. [42]

[19] A. DELL'AERE, M. HIRSCH, B. KLÖPPER, M. KOESTER, A. KRUPP, M. KRÜGER, T. MÜLLER, S. OBERTHÜR, S. POOK, C. PRIESTERJAHN, C. ROMAUS, A. SCHMIDT, C. SONDERMANN-WÖLKE, M. TICHY, H. VÖCKING, AND D. ZIMMER, *Verlässlichkeit selbstoptimierender Systeme - Potenziale nutzen und Risiken vermeiden*, HNI-Schriftenreihe 235, Bonifatius GmbH, 2009. [31]

[20] M. DELLNITZ, O. SCHÜTZE, AND T. HESTERMEYER, *Covering Pareto sets by multilevel subdivision techniques*, Journal of Optimization Theory and Application **124 (1)** (2005), pp. 113–136. [34, 40, 41]

[21] M. DELLNITZ AND K. WITTING, *Computation of robust Pareto points*, Int. Journal of Computing Science and Mathematics (IJCSM) **2 (3)** (2009), pp. 243–266. [81, 92, 104]

[22] P. DEUFLHARD AND A. HOHMANN, *Numerical Analysis in Modern Scientific Computing*, Springer, New York, NY, USA, 2003. [5, 28]

[23] E. J. DOEDEL, A. R. CHAMPNEYS, R. C. PAFFENROTH, T. F. FAIRGRIEVE, Y. A. KUZNETSOV, B. E. OLDEMAN, B. SANDSTEDE, AND X.-J. WANG, *AUTO2000: Continuation and bifurcation software for ordinary differential equations (with homcont)*, tech. report, California Institute of Technology, Pasadena, California USA, 2000. [30, 68, 122]

[24] M. EHRGOTT, *Multicriteria Optimization*, Lecture Notes in Economics and Mathematical Systems 491, Springer-Verlag, Berlin, 2000. [1]

[25] M. EHRGOTT, *Multicriteria Optimization*, Springer-Verlag, Berlin, second ed., 2005. [35]

[26] R. ENKHBAT, J. GUDDAT, AND A. CHINCHULUUN, *Parametric multiobjective optimization*, Springer Optimization and Its Applications **17** (2008), pp. 529–538. [66]

[27] A. FIACCO, *Sensitivity, Stability and Parametric Analysis*, Elsevier Science Ltd, 1983. [26, 27]

[28] J. E. FIELDSEND AND R. M. EVERSON, *Multi-objective optimisation in the presence of uncertainty*, in Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC05), 2005, pp. 476–483. [80]

[29] J. FIGUEIRA, M. GEIGER, S. GRECO, J. JAHN, K. KLAMROTH, M. INUIGUCHI, V. MOUSSEAU, S. SAYIN, R. SLOWINSKI, M. M. WIECEK, AND K. WITTING, *09041 working group on mcdm for robust multiobjective optimization (1st round)*, in Hybrid and Robust Approaches to Multiobjective Optimization, K. Deb, S. Greco, K. Miettinen, and E. Zitzler, eds., no. 09041 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2009, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany. [6, 76]

[30] J. FIGUEIRA, S. GRECO, AND M. EHRGOTT (EDS.), *Multiple Criteria Decision Analysis: State of the Art Surveys*, International Series in Operations Research & Management Science 78, Springer Science + Business Media B. V., New York, 2005. [80]

[31] M. FLIESS, J. LÉVINE, P. MARTIN, AND P. ROUCHON, *Flatness and defect of non-linear systems: Introductory theory and examples*, International Journal of Control **61**, no. 6 (1995), pp. 1327–1361. [3, 46, 52]

[32] C. M. FONSECA, P. J. FLEMING, E. ZITZLER, K. DEB, AND L. THIELE, *Evolutionary Multi-Criterion Optimization. Second International Conference EMO 2003*, Springer, 2003. [1]

[33] U. FRANK, J. GAUSEMEIER, H. GIESE, F. KLEIN, O. OBERSCHELP, A. SCHMIDT, B. SCHULZ, H. VÖCKING, AND K. WITTING, *Selbstoptimierende Systeme des Maschinenbaus - Definitionen und Konzepte*, HNI-Schriftenreihe 155, Bonifatius GmbH, 2004. [31, 72]

[34] P. FUNK, *Variationsrechnung und ihre Anwendung in Physik und Technik*, Springer-Verlag, second ed., 1970. [19]

[35] A. GASPAR-CUNHA AND J. A. COVAS, *Robustness in multi-objective optimization using evolutionary algorithms*, Journal of Computational Optimization and Applications **39**, no. 1 (2008), pp. 75–96. [79]

[36] S. GASS AND T. SAATY, *The computational algorithm for the parametric objective function*, Naval Research Logistics Quarterly 2 (1955), pp. 39–45. [38, 39]

[37] C. GEIGER AND C. KANZOW, *Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben*, Springer, Berlin, 1999. [14, 15]

[38] J. GEISLER, K. WITTING, A. TRÄCHTLER, AND M. DELLNITZ, *Multiobjective optimization of control trajectories for the guidance of a rail-bound vehicle*, in 17th IFAC World Congress, July 6-11, 2008, Seoul, Korea, 2008. [51, 52, 54]

[39] J. GEISLER, K. WITTING, A. TRÄCHTLER, AND M. DELLNITZ, *Self-optimization of the guidance module of a rail-bound vehicle*, in 7th International Heinz Nixdorf Symposium 'Self-optimizing Mechatronic Systems: Design the Future', February 20-21, 2008, Paderborn, J. Gausemeier, F. Rammig, and W. Schäfer, eds., Paderborn, 2008, HNI-Verlagsschriftenreihe, pp. 85–100. [51, 53, 54]

[40] A. M. GEOFFRION, *Proper efficiency and the theory of vector maximization*, Journal of Mathematical Analysis and Applications **22 (3)** (1968), pp. 618–630. [36]

[41] M. GIAQUINTA AND S. HILDEBRANDT, *Calculus of Variations I*, Springer-Verlag, 1996. [19, 21, 23, 86]

[42] M. GOLUBITSKY AND D. G. SCHAEFFER, *Singularities and groups in bifurcation theory. Vol. I*, Applied Mathematical Sciences 51, Springer-Verlag, New York, 1985. [123]

[43] A. GÖPFERT AND R. NEHSE, *Vektoroptimierung*, Teubner Verlagsgesellschaft Leipzig, 1990. [36, 37]

[44] J. Guddat, F. Guerra Vasquez, and H. T. Jongen, *Parametric Optimization: Singularities, Pathfollowing and Jumps*, John Wiley & Sons Ltd, 1990. [25, 28]

[45] J. Guddat, F. Guerra Vasquez, K. Tammer, and K. Wendler, *Multiobjective and Stochastic Optimization Based on Parametric Optimization*, Akademie-Verlag, Berlin, 1985. [35]

[46] J. Guddat, F. G. Vasquez, K. Tammer, and K. Wendler, *Multiobjective and Stochastic Optimization Based on Parametric Optimization*, Mathematical Research 26, Akademie-Verlag Berlin, 1985. [25]

[47] S. Gunawan and S. Azarm, *Multi-objective robust optimization using a sensitivity region concept*, Structural and Multidisciplinary Optimization **29**, no. 1 (2005), pp. 50–60. [79, 80]

[48] Y. Y. Haimes, L. S. Lasdon, and D. A. Wismer, *On a bicriterion formulation of the problems of integrated system identification and system optimization*, IEEE Transactions on Systems, Man, and Cybernetics **1** (1971), pp. 296–297. [38]

[49] C. Henke, N. Fröhleke, and J. Böcker, *Advanced convoy control strategy for autonomously driven railway vehicles*, in Proceedings of the IEEE Conf. on Intelligent Transportation Systems, Toronto, Kanada, September 2006. [56, 58]

[50] M. Henke, *Antrieb mit doppeltgespeistem Linearmotor für ein spurgeführtes Bahnfahrzeug*, PhD thesis, University of Paderborn, Germany, 2003. [56]

[51] T. Hestermeyer, P. Schlautmann, and C. Ettingshausen, *Active suspension system for railway vehicles – system design and kinematics*, in 2nd IFAC - Conference on Mechatronic Systems, Berkeley, California, USA, 2002. [47]

[52] H. Heuser, *Analysis 2*, Teubner Verlag, fourth ed., 1988. [11, 26]

[53] C. Hillermeier, *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*, Birkhäuser, 2001. [36, 38, 41, 42, 44, 66, 120]

[54] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, Springer, 1996. [13]

[55] E. Hughes, *Evolutionary multi-objective ranking with uncertainty and noise*, in EMO '01: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, London, UK, 2001, Springer-Verlag, pp. 329–343. [80]

[56] M. INUIGUCHI, *Robust optimization under softness in a fuzzy linear programming problem*, International Journal of Approximate Reasoning **18**, no. 1-2 (1998), pp. 21 – 34. [80]

[57] F. JARRE AND J. STOER, *Optimierung*, Springer, Berlin, 2004. [13]

[58] O. JUNGE, J. E. MARSDEN, AND S. OBER-BLÖBAUM, *Discrete mechanics and optimal control*, Proc. IFAC Conf., Prague, June 2005. [133]

[59] B. KIM, E. GEL, J. FOWLER, W. CARLYLE, AND J. WALLENIUS, *Evaluation of nondominated solution sets for k-objective optimization problems: An exact method and approximations*, European Journal of Operational Research **173 (2)** (2006), pp. 565–582. [35]

[60] D. KLATTE AND B. KUMMER, *Nonsmooth Equations in Optimization: Regularity, Calculus, Methods and Applications (Nonconvex Optimization and Its Applications)*, Kluwer Academic Publishers, 2002. [25]

[61] K. KÖNIGSBERGER, *Analysis 2*, Springer-Verlag, second ed., 1997. [11, 52]

[62] J. KNOWLES, D. CORNE, AND K. DEB, *Multiobjective Problem Solving from Nature: From Concepts to Applications (Natural Computing Series)*, Springer, 2008. [38]

[63] H. KUHN AND A. TUCKER, *Nonlinear programming*, Proc. 2nd Berkeley Symp. Math. Statist. Probability, (J. Neumann, ed.) (1951), pp. 481–492. [4, 36, 37]

[64] S. LEYENDECKER, J. E. MARSDEN, AND M. ORTIZ, *Variational integrators for constrained dynamical systems*, J. Applied Mathematics and Mechanics **88**, no. 9 (2008), pp. 677–708. [85]

[65] M. LI, *Robust Optimization and Sensitivity Analysis with Multi-objective Genetic Algorithms: Single- and Multi-Disziplinary Applications*, PhD thesis, 2007. [79]

[66] R. LI, A. POTTHARST, K. WITTING, O. ZNAMENSHCHYKOV, J. BÖCKER, N. FRÖHLEKE, R. FELDMANN, AND M. DELLNITZ, *Design and implementation of a hybrid energy supply system for railway vehicles*, in Proc. of APEC2005, IEEE Applied Power Electronics Conference 2005, Austin, Texas, USA, 2005, pp. 474–480. [48]

[67] D. G. LUENBERGER, *Linear and Nonlinear Programming*, Addison Wesley Publishing Company, second ed., 1984. [13, 16, 17]

[68] J. E. MARSDEN AND M. WEST, *Discrete mechanics and variational integrators*, Acta Numerica , no. 10 (2001), pp. 357–514. [7, 84, 85]

[69] K. MIETTINEN, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, 1999. [1, 35, 36, 37, 38, 42]

[70] M. MILAM, K. MUSHAMBI, AND R. MURRAY, *A new computational approach to real-time trajectory generation for constrained mechanical systems*, in Proceedings of the 39th IEEE Conference on Decision and Control, 2000, pp. 845–551. [3, 46, 53]

[71] G. MOORE AND A. SPENCE, *The calculation of turning points of nonlinear equations*, SIAM Journal of Numerical Analysis **17**, no. 4 (1980), pp. 567–576. [118, 121]

[72] R. MURRAY, M. RATHINAM, AND W. SLUIS, *Differential flatness of mechanical control systems: A catalog of prototype systems*, in Proceedings of the 1995 ASME International Congress and Exposition, San Francisco, 1995. [3, 46, 52]

[73] "'NEUE BAHNTECHNIK PADERBORN"' – PROJECT, *http://www.railcab.de*. Web-Page. [2, 46]

[74] S. NISHIKAWA, *Variational problems in geometry*, American Mathematical Society, 2002. [89]

[75] J. ORTEGA AND W. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, Inc., London, 1970. [5, 28]

[76] V. PARETO, *Manual of Political Economy*, The MacMillan Press, 1971 (original edition in French in 1927). [33]

[77] L. A. PARS, *An Introduction to the Calculus of Variations*, Heinemann, London, 1962. [20]

[78] N. PETIT, M. MILAM, AND R. MURRAY, *Inversion based constrained trajectory optimization*, in 5th IFAC Symposium on Nonlinear Control Systems, 2001. [53]

[79] E. R. PINCH, *Optimal Control and the Calculus of Variations*, Oxford University Press, 1993. [19, 20]

[80] A. POTTHARST, *Energieversorgung und Leittechnik einer Anlage mit Linearmotor getriebenen Bahnfahrzeugen*, PhD thesis, University of Paderborn, Germany, 2006. [47]

[81] A. POTTHARST, K. BAPTIST, O. SCHÜTZE, J. BÖCKER, N. FRÖHLECKE, AND M. DELLNITZ, *Operating point assignment of a linear motor driven vehicle using multiobjective optimization methods*, in Proc. of the 11th International Conference EPE-PEMC 2004, Riga, Latvia, 09/2004. [56, 58]

[82] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, New York, NY, USA, 1992. [53]

[83] C. Romaus, J. Böcker, K. Witting, A. Seifried, and O. Znamenshchykov, *Optimal energy management for a hybrid energy storage system combining batteries and double layer capacitors*, in Proceedings of the IEEE Energy Conversion Congress and Exposition (ECCE), 2009. [48, 49]

[84] R. Roy, Y. Azene, D. Farrugia, C. Onisa, and J. Mehnen, *Evolutionary multi-objective design optimisation with real life uncertainty and constraints*, CIRP Annals - Manufacturing Technology **58**, no. 1 (2009), pp. 169 – 172. [80]

[85] S. Schäffler, R. Schultz, and K. Weinzierl, *A stochastic method for the solution of unconstrained vector optimization problems*, J. Opt. Th. Appl. **114 (1)** (2002), pp. 209–222. [1, 38]

[86] T. Schneider, B. Schulz, C. Henke, K. Witting, D. Steenken, and J. Böcker, *Energy transfer via linear doubly-fed motor in different operating modes*, in Proceedings of the International Electric Machines and Drives Conference, May 3-6, 2009, Miami, Florida, USA. [56, 60, 63, 70, 75]

[87] O. Schütze, *Set Oriented Methods for Global Optimization*, PhD thesis, University of Paderborn, Germany, 2004. [40, 41]

[88] O. Schütze, *A new data structure for the nondominance problem in multi-objective optimization*, in Evolutionary Multi-Criterion Optimization (EMO 03), C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, eds., Springer 2, 2003, pp. 509–518. [40, 41]

[89] O. Schütze, A. Dell'Aere, and M. Dellnitz, *On continuation methods for the numerical treatment of multi-objective optimization problems*, in Practical Approaches to Multi-Objective Optimization, J. Branke, K. Deb, K. Miettinen, and Steuer, R. E. (eds.), eds., no. 04461 in Dagstuhl Seminar Proceedings, Dagstuhl, 2005, Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany. [41]

[90] O. Schütze, S. Mostaghim, M. Dellnitz, and J. Teich, *Covering Pareto sets by multilevel evolutionary subdivision techniques*, in Evolutionary Multi-Criterion Optimization, C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, eds., Lecture Notes in Computer Science, 2003, pp. 118–132. [40, 41]

[91] O. Schütze, M. Vasile, and C. A. Coello Coello, *Approximate solutions in space mission design*, Lecture Notes in Computer Science **5199** (2008), pp. 805–814. [80]

[92] O. SCHÜTZE, K. WITTING, S. OBER-BLÖBAUM, AND M. DELLNITZ, *Set oriented methods for the numerical treatment of multiobjective optimization problems*. To appear in EVOLVE – A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation, Emilia Tantar et al., eds., Springer, 2011. [1, 38, 40]

[93] P. K. SHUKLA, J. DUTTA, AND K. DEB, *On properly pareto optimal solutions*, in Practical Approaches to Multi-Objective Optimization, J. Branke, K. Deb, K. Miettinen, and R. E. Steuer, eds., no. 04461 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2005. http://drops.dagstuhl.de/opus/volltexte/2005/240. [36]

[94] R. SLOWINSKI AND J. TEGHEM (EDS.), *Stochastic versus Fuzzy approaches to Multiobjective Mathematical Programming under uncertainty*, Kluwer Academic, Dordrecht, 1990. [80]

[95] P. SPELLUCCI, *Numerische Verfahren der nichtlinearen Optimierung*, Birkhäuser Verlag, 1993. [13]

[96] D. STEENKEN, *Multiobjective optimization and sensitivity evaluation of IC base components using set-oriented methods*, 2009. Diploma thesis, University of Paderborn, Germany. [42, 114, 115, 116, 117]

[97] R. E. STEUER, *Multiple Criteria Optimization: Theory, Computation, and Application*, John Wiley & Sons, Inc., 1986. [35]

[98] G. STRANG, *Introduction to linear algebra*, Wellesley-Cambridge Press, 1993. [99]

[99] A. SÜLFLOW, N. DRECHSLER, AND R. DRECHSLER, *Robust multi-objective optimization in high dimensional spaces*, in Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, 2006, pp. 715–726. [80]

[100] J. TEICH, *Pareto-front exploration with uncertain objectives*, in EMO '01: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, London, UK, 2001, Springer-Verlag, pp. 314–328. [80]

[101] THE MATHWORKS COMPANY, *http://www.mathworks.com/access/helpdesk/help/toolbox/optim/ug/fsolve.html*. Web-Page. [89, 92, 109]

[102] C. TIAHRT AND A. POORE, *A bifurcation analysis of the nonlinear parametric programming problem*, Mathematical Programming **47** (1990), pp. 117–141. [25]

[103] S. TSUTSUI AND A. GOSH, *Genetic algorithms with a robust solution searching scheme*, IEEE Transactions on Evolutionary Computation **1**, no. 3 (1997), pp. 201–219. [77]

[104] M. J. VAN NIEUWSTADT AND R. M. MURRAY, *Real-time trajectory generation for differentially flat systems*, International Journal of Robust and Nonlinear Control **8 (11)** (1998), pp. 995–1020. [3, 46, 53, 56]

[105] B. WERNER AND A. SPENCE, *The computation of symmetry-breaking bifurcation points*, SIAM Journal on Numerical Analysis **21**, no. 2 (1984), pp. 388–399. [123, 126]

[106] D. WERNER, *Funktionalanalysis*, Springer-Verlag, 2005. [37]

[107] K. WITTING, S. OBER-BLÖBAUM, AND M. DELLNITZ, *A new definition for robustness in parametric multiobjective optimization problems based on the calculus of variations* (2011). Submitted to the Journal of Global Optimization, Special Issue on Multiobjective Optimization. [81]

[108] K. WITTING, B. SCHULZ, M. DELLNITZ, J. BÖCKER, AND N. FRÖHLEKE, *A new approach for online multiobjective optimization of mechatronic systems*, International Journal on Software Tools for Technology Transfer STTT **10 (3)** (2008), pp. 223–231. [56, 63, 67, 70]

[109] Y. XUE, D. LI, W. SHAN, AND C. WANG, *Multi-objective robust optimization using probabilistic indices*, 2007. Third International Conference on Natural Computation (ICNC 2007). [79]

[110] B. YANG, M. HENKE, AND H. GROTSTOLLEN, *Pitch analysis and control design for the linear motor of a railway carriage*, 2001. IEEE Industrial Applications Society Annual Meeting (IAS), Chicago, USA, pp. 2360-2365. [47]

[111] L. ZADEH, *Optimality and non-scalar-valued performance criteria*, IEEE Transactions on Automatic Control **8** (1963), pp. 59–60. [38, 39]

[112] E. ZITZLER, M. LAUMANNS, AND L. THIELE, *SPEA2: Improving the strength of pareto evolutionary algorithms for multiobjective optimization*, in Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001), 2002, pp. 95–100. [1]

Index