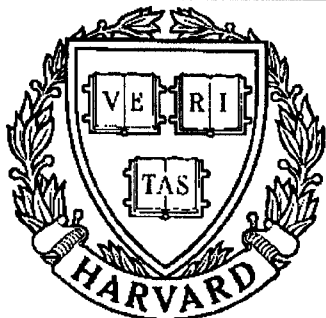


THESIS REPORT

Ph.D.



S Y S T E M S
R E S E A R C H
C E N T E R



*Supported by the
National Science Foundation
Engineering Research Center
Program (NSFD CD 8803012),
the University of Maryland,
Harvard University,
and Industry*

Fast, Globally Convergent Optimization Algorithms, with Application to Engineering System Design

*by J. Zhou
Advisor: A.L. Tits*

**FAST, GLOBALLY CONVERGENT OPTIMIZATION
ALGORITHMS, WITH APPLICATION TO
ENGINEERING SYSTEM DESIGN**

by
Jian Zhou

Dissertation submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
1992

Advisory Committee:

Professor André L. Tits, Chairman/Advisor
Professor William S. Levine
Professor John S. Baras
Associate Professor Eyad Abed
Associate Professor Evangelos Zafiriou

© Copyright by
Jian Zhou

DEDICATION

To my wife : Li Shi
and
to my parents in China

Acknowledgements

It would be impossible for me to make any significant progress in my research and in the preparation of my dissertation without Dr. André L. Tits' systematic and enlightening guidance, his generous time and patience for invaluable discussions, and his continuous encouragement that helps me to build up my confidence in academic research. What I have benefited from being a student of Dr. Tits' will definitely benefit my future career. It is beyond words to express my gratitude to him, yet I cannot help but say "Thank you for everything, André!"

I am grateful to Dr. Eliane R. Panier for a great many valuable discussions.

I sincerely appreciate the time that every committee member spent in the process of my defense.

I am indebted to Academia Sinica that supported me financially in the beginning of my Ph.D. study, to the leaders and colleagues at the Institute of Optics and Electronics of Academia Sinica, who have cared for my study at the Institute and abroad.

Thanks go to the Systems Research Center that provides excellent research environment and facilities, and the professional service of the staff.

My research has been supported by NSF grant DMC-88-15996, by NSF's Engineering Research Centers Program grant NSF-D-CDR-88-03012 and by a grant from the Westinghouse Corporation.

I would like to thank my lovely wife Li Shi who has constantly been behind me through thick and thin.

ABSTRACT

Title of Dissertation: FAST, GLOBALLY CONVERGENT OPTIMIZATION ALGORITHMS, WITH APPLICATIONS TO ENGINEERING SYSTEMS DESIGN

Jian Zhou, Doctor of Philosophy, 1991

Dissertation directed by: André L. Tits, Professor
Electrical Engineering Department

Complex engineering system design usually involves multiple objective specifications. Tradeoffs have to be made among these specifications, possibly under constraints, as they oftentimes compete with one another. It has been realized that these problems can be faithfully translated into (inequality) constrained minimax problems, or a sequence of such problems and then the original design problems can be solved with the support of numerical optimization techniques. This dissertation develops a number of optimization algorithms for the solution of these problems, with emphasis on many distinctive features of engineering systems that existing optimization algorithms have not fully exploited or accounted for. Efforts are made to maintain “desirable” analytic properties that existing popular optimization algorithms enjoy.

Algorithms are developed in a very general setup, yet specialized to indicate the direct connections with different branches of engineering applications. The adaptation of advanced optimization techniques, such as feasible sequential quadratic programming (FSQP) and nonmonotone line search (NLS), to the solution of engineering problems are carefully addressed. Efficiency of these algorithms has been given high priority in the development, since it is usually very time-consuming to evaluate specifications in engineering applications.

In the presence of functional specifications, *i.e.*, specifications that are to be satisfied over an interval of a free parameter such as time or frequency (shaping of time or frequency response, for instance), an efficient algorithm is proposed that significantly outperforms the existing algorithms in the same context. This algorithm

has been successfully applied for the solution of many engineering problems in the context of optimization-based system design.

These new algorithms are satisfactorily characterized by conventional notions of optimization such as global convergence and rate of local convergence. Interesting and meaningful results on the rate of local convergence are obtained with the aid of nonmonotone line search. The performance of these algorithms represented by extensive numerical experiments measures up to the theoretic analysis.

A number of feedback system design problems are provided to illustrate the efficiency and applicability of these algorithms.

Table of Contents

List of Tables	vii
List of Figures	viii
Notations	ix
1 Introduction	1
1.1 Optimization problems arising in engineering design	1
1.2 State of the art of optimization techniques	4
1.3 Objectives	6
2 Some Basic Concepts of Optimization	9
2.1 Classical SQP scheme	9
2.2 Variable metric methods	10
2.3 First order necessary conditions of optimality	11
2.4 KKT point, stationary point and Lagrange multiplier vector	11
2.5 Strict complementary slackness	12
2.6 Second order sufficiency conditions of optimality	12
2.7 Global convergence	13
2.8 Armijo line search	13
2.9 Rate of convergence	13
2.9.1 Linear rate of convergence	14
2.9.2 One-step superlinear rate of convergence	14
2.9.3 Two-step superlinear rate of convergence	14
2.10 Maratos effect	15
3 NLS for Unconstrained Minimax Problems	17
3.1 Introduction	17
3.2 Algorithm	19
3.3 Convergence analysis	22
3.3.1 Global convergence	23
3.3.2 Superlinear convergence	26
3.4 Numerical experiments	32

3.5	Alternative line search rules	33
4	NLS for Inequality Constrained Minimax Problems	37
4.1	Introduction	37
4.2	Algorithm	41
4.3	Convergence results	45
4.3.1	Global convergence	45
4.3.2	Superlinear convergence	46
4.4	Numerical experiments	47
5	Nonmonotone Line Search Revisited	51
5.1	Motivations	51
5.2	One-step superlinear convergence	53
5.2.1	Feasible iterates algorithm	53
5.2.2	Penalty-type algorithm	60
5.3	Numerical experiments	62
6	Finely Discretized Problems from SIP	67
6.1	Introduction	67
6.2	Algorithm	69
6.3	Convergence Analysis	73
6.3.1	Global convergence	74
6.3.2	Local convergence	84
6.4	Extension to constrained SIP problems	88
6.5	Implementation issues	91
6.6	Numerical experiments	92
7	Examples of Engineering Design	95
7.1	Controller design using Q -parameterization	95
7.2	Controller design using state feedback	104
7.3	PID controller design	108
8	Concluding Remarks	111
	Appendix A. Some Proofs	115
A.1	Proofs for Chapter 3	115
A.2	Proofs for Chapter 4	117

A.3	Proofs for Chapter 5	124
A.4	Proofs for Chapter 6	125
Appendix B. Auxiliary Materials for Design Examples		129
B.1	Scripts for the design using Q -parameterization	130
B.1.1	Without constraints on u_1	130
B.1.2	With constraints on u_1	133
B.2	Scripts for the state feedback design	137
B.3	Scripts for the PID control design	140
Bibliography		143

List of Tables

3.1	Numerical Results for Minimax Problems	35
4.1	Numerical Results for Constrained Minimax Problems	49
4.2	Numerical Results for Constrained Non-Minimax Problems	50
5.1	Numerical Results for (Constrained) Minimax Problems	64
5.2	Numerical Results of Penalty-Type Algorithm	65
6.1	Numerical Performance of Algorithm 6.2	93
7.1	Comparison of Algorithm 6.2 and a First Order Algorithm (a)	102
7.2	Comparison of Algorithm 6.2 and a First Order Algorithm (b)	106
7.3	Comparison of Algorithm 6.2 and a First Order Algorithm (c)	110

List of Figures

1.1	Predictive Control Using Neural Net Dynamic Model	1
1.2	Block Diagram of a Feedback System	3
1.3	Shaping the Step Response of a Dynamic System	4
1.4	Graph of a <i>max</i> Function for $p = 2$ and $x \in \mathbb{R}$	5
7.1	A Feedback System	95
7.2	Unit Step Response of the Plant	96
7.3	Desired Bound for Unit Step Response of Closed Loop System	97
7.4	Initial Response of Closed Loop System	99
7.5	Final Response via Algorithm 6.2 (a)	99
7.6	Final Response via Algorithm 6.2 (b)	100
7.7	Final Response via a First Order Algorithm	100
7.8	Step Response with the Reduced Order Controller (a)	101
7.9	Step Response with the Reduced Order Controller (b)	101
7.10	Unconstrained Control Signal to Plant	103
7.11	Desired Bound of the Step Response with Constrained Control . . .	104
7.12	Step Response with Constrained Control Signal to Plant	105
7.13	Constrained Control Signal to Plant	105
7.14	State Feedback System	106
7.15	Initial Response of State Feedback System	107
7.16	Final Response of State Feedback System	107
7.17	Optimal Nyquist Plot of Return Difference	109
7.18	Bode Plot of the Open Loop System	109
7.19	Final Response of PID Controlled Closed Loop System	110

Notation

Symbol	Meaning
$\mathbb{R}$:	one-dimensional Euclidean space.
$\mathbb{R}^n$:	n -dimensional Euclidean space.
$\mathbb{R}^{n \times m}$:	$n \times m$ -dimensional Euclidean space.
$\mathbb{N}$:	set of natural numbers, <i>i.e.</i> , $\mathbb{N} = \{0, 1, 2, \dots\}$.
$K \subset \mathbb{N}$:	subset of $\mathbb{N}$.
$\emptyset$:	empty set.
Ω :	compact interval of $\mathbb{R}$.
Ω^q :	discretized subset of Ω .
Ω_k :	subset of Ω^q .
$\omega \in \Omega$:	an element of Ω .
Σ :	summation.
$\forall$:	for all.
$x \in \mathbb{R}^n$:	one-column and n -row vector in $\mathbb{R}^n$.
x_s :	x is subscripted by s with s either the index of the s th component of x or the index of iterations.
$H \in \mathbb{R}^{n \times m}$:	n -row and m -column matrix in $\mathbb{R}^{n \times m}$.
H^T :	transpose of some matrix $H \in \mathbb{R}^{n \times m}$, resulting in an m -row and n -column matrix.
$\frac{\partial f(x,y)}{\partial x_i}$:	first order partial derivative of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with respect to $x_i \in \mathbb{R}$.

$\frac{\partial^2 f(x,y)}{\partial^2 x_i}$: second order partial derivatives of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with respect to $x_i \in \mathbb{R}$.

$\nabla_x f(x,y)$: gradient of f with respect to $x \in \mathbb{R}^n$, resulting in a one-row and n -column vector, *i.e.*, $\nabla_x^T f(x,y) = \frac{\partial f(x,y)}{\partial x}$.

$\langle x, y \rangle$: inner product of two vectors x and y both in $\mathbb{R}^n$, defined by

$$\langle x, y \rangle = \sum_{i=1}^n x_i \times y_i$$

with x_i and y_i the i th component of x and y respectively.

$\|x\|$: Euclidean norm of the vector $x \in \mathbb{R}^n$, defined as $\sqrt{\langle x, x \rangle}$.

$|p|$: absolute value for $p \in \mathbb{R}$ or cardinality for p indexed by $K \subset \mathbb{N}$.

$x_k \xrightarrow{K} y$: the sequence $\{x_k\}$ converges to y on K .

$A \setminus B$: subset of A , containing all elements of A except for those in B .

Chapter 1

Introduction

1.1 Optimization problems arising in engineering design

Consider the following control system that involves neural net dynamic model. The model $M(x)$ is parameterized by a set x of parameters. One of the tasks is to train the neural net model, namely, to identify the parameters x so that the resulting model can faithfully characterize a nonlinear plant and then an optimal control law can be designed using the model. A commonly used criterion is to minimize the error between the predictive output of the model and the desired output. Certain constraints are imposed, for instance, to maintain a sensible training region of the neural net model

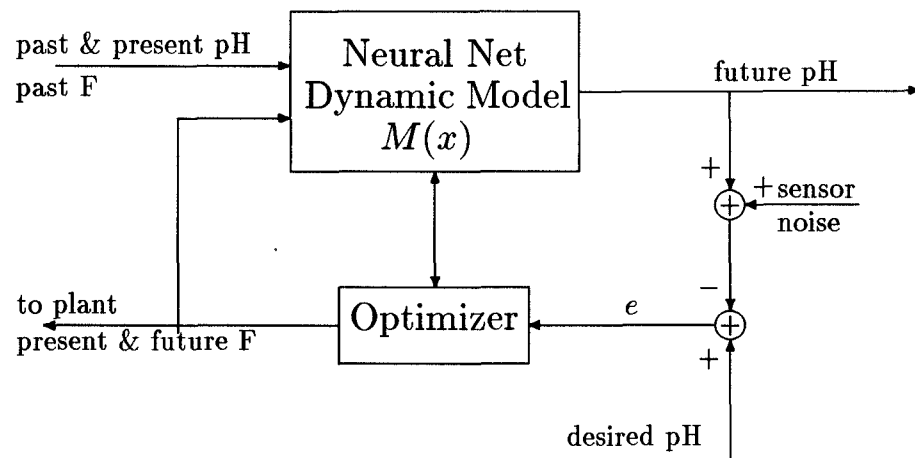


Figure 1.1: Predictive Control Using Neural Net Dynamic Model

so that the predictive output can be trusted. And, at each sampling time, the output of the optimizer should yield the corresponding value of each parameter in x and the required control signal. Therefore, this problem can be formulated as the following “off-line” optimization problem between sampling instants (thus, explicitly independent of time):

$$\begin{aligned} \min_{F,x} \quad & \|e(F, x)\|^2 \\ \text{s.t.} \quad & g(F, x) \leq 0. \end{aligned}$$

Other criteria could also be specified in order to achieve more sophisticated performance requirements of the neural net modeled system. In general, complex engineering system design usually faces the challenge that multiple specifications must be achieved simultaneously, possibly with constraints. Tradeoffs have to be made oftentimes among these specifications since they are likely to conflict with one another. It would be impossible, except for extremely simple cases, to achieve any tradeoffs in a certain optimal sense without the support of numerical optimization. As a matter of fact, numerical optimization has been applied successfully to many fields of engineering designs (see, *e.g.*, [9,23,24,53,67,71]). These problems can often be faithfully formulated, especially in the context of interactive optimization-based engineering systems design, as inequality constrained minimax optimization problems of the following form (see, *e.g.*, [57,81,82])

$$\begin{aligned} (P) \quad & \min_{x \in R^n} \quad \max_{i=1,\dots,p} f_i(x) \\ \text{s.t.} \quad & g_j(x) \leq 0, \quad j = 1, \dots, m. \end{aligned}$$

In many cases, inclusion of equality constraints can be avoided or effectively eliminated. Use of the *max*-type objective function in (P) is one of the useful ways that enable one to exploit the compromises among different specifications, provided that, among other aspects, the various specifications are well balanced and are made dimension free (see [57] for effective approaches to achieving these). To exploit the compromises meaningfully, some (“hard”) inequality constraints must be satisfied through the optimization process. Hard constraints could include the stability of a dynamic system, the requirement of the parameter regions out of which some specifications are not well-defined (for instance, the training region of the neural net model), *etc.* Clearly, it is advantageous, or even crucial in some situations, that, once a feasible point (point satisfying all constraints) has been obtained, all the subsequent iterates be feasible. Algorithms that generate feasible iterates are of particular interest in engineering applications.

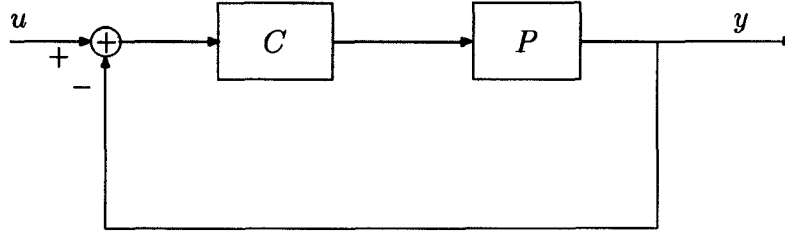


Figure 1.2: Block Diagram of a Feedback System

Many engineering design problems are also characterized by the feature that some specifications (either cost functions or constraints or both) are required to be satisfied over intervals of one or more independent parameters (time, frequency, *etc.*). A simplest such example would be to shape the step-response of the feedback system in Figure 1.2; where P represents the model of a plant to be controlled, C is the controller to be designed and u is the desired value of the output y . By assuming (tentatively) the structure of the controller C and letting $x \in \mathbb{R}^n$ denote the design parameters (the coefficients of numerator and denominator of the transfer function $C(s)$, for instance), the unit step response $y(x, t)$ of the system is desired to be within the envelope in Figure 1.3, shaped by an upper bound $\bar{y}(t)$ and a lower bound $\underline{y}(t)$. If we define

$$f_1(x, t) = y(x, t) - \bar{y}(t) \quad \text{and} \quad f_2(x, t) = -y(x, t) + \underline{y}(t)$$

and assume that the time interval of interest is $[0, T]$ for some $T > 0$, the shaping problem can then be formulated as the optimization problem

$$\min_{x \in \mathbb{R}^n} \max_{i=1,2} \max_{t \in [0, T]} \{f_i(x, t)\}.$$

Constraints, such as gain and phase margin (in frequency domain), can be easily accounted for. Other examples include optimal filtering, robust controller design, and worst case circuit design (see, *e.g.*, [23,36,45,62,67,69,73]). Such problems can be formulated as the general optimization problem

$$(SIP) \quad \begin{aligned} & \min_{x \in \mathbb{R}^n} \max_{i=1, \dots, p} \max_{\omega \in \Omega} f_i(x, \omega) \\ & \text{s.t.} \quad g_j(x, \omega) \leq 0, \quad j = 1, \dots, m; \quad \forall \omega \in \Omega \end{aligned}$$

with Ω compact. (Use of ω and Ω is to follow conventions.) Different independent parameters (t in Figure 1.3) can be involved in the specifications. It can be observed

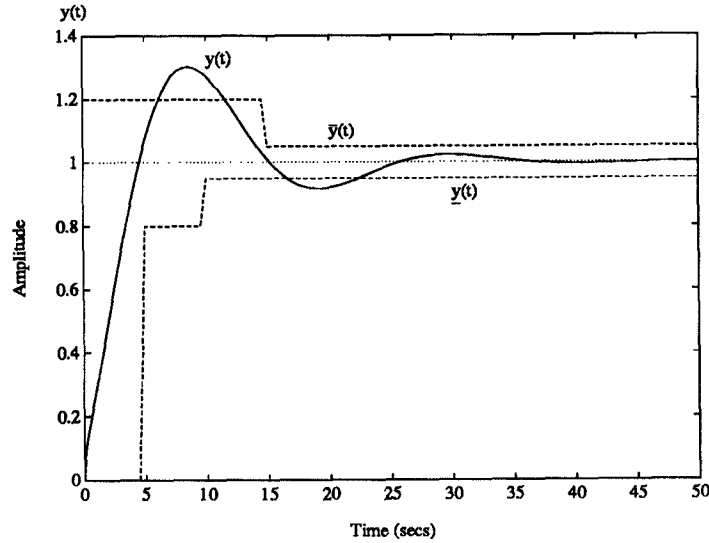


Figure 1.3: Shaping the Step Response of a Dynamic System

that, if Ω contains only a finite number of points, (SIP) and (P) are essentially the same. (SIP) usually becomes much more challenging than (P) if Ω is a general compact subset of $\mathbb{R}$. The complexity of both algorithm and analysis is drastically more involved. Many existing optimization algorithms solve problem (SIP) by actually solving a sequence of problems in the form of (P) , and the sequence of the solution of the corresponding (P) would eventually have limit points that solve (SIP) (see, e.g., [25,29,36,37,54,62,69]).

Exploitation of optimization algorithms that efficiently solve (P) and (SIP) are of the primary concern of this dissertation. As the nature of numerical algorithms may vary drastically when the nature of specifications varies, we assume throughout this dissertation that $f_i(\cdot)$, $i = 1, \dots, p$, and $g_j(\cdot)$, $j = 1, \dots, m$, are smooth, and $f_i(\cdot, \cdot)$, $i = 1, \dots, p$, and $g_j(\cdot, \cdot)$, $j = 1, \dots, m$, are continuous and are smooth in their first argument. These assumptions will be made precise in subsequent chapters.

1.2 State of the art of optimization techniques

One of the key tasks that challenge optimization experts in solving engineering problems is to solve the problem (P) with the consideration of many distinctive features such as feasibility and nondifferentiability of the *max*-type objective function, and to do so efficiently. Difficulties can be identified easily. The *max* function in the problem

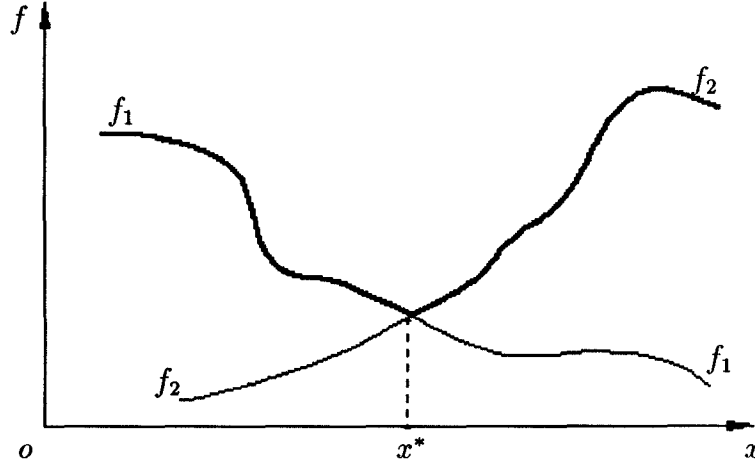


Figure 1.4: Graph of a *max* Function for $p = 2$ and $x \in \mathbb{R}$

(P) is not differentiable at any point where more than one objective function assume the same value, as can be seen at x^* in Figure 1.4. This, for one thing, leads to the difficulty that the conventional stopping criterion for smooth function cannot be used, since the gradient at the optimal point x^* does not vanish in the current situation. The feasible region may be highly nonlinear in the presence of nonlinear constraints, thus hard to reach. While (P) can be readily translated, by introducing an auxiliary variable $x_{n+1} \in \mathbb{R}$, into the constrained smooth minimization problem

$$(P_m) \quad \begin{array}{ll} \min_{x \in \mathbb{R}^n, x_{n+1} \in \mathbb{R}} & x_{n+1} \\ \text{s.t.} & f_i(x) - x_{n+1} \leq 0, \quad i = 1, \dots, p \\ & g_j(x) \leq 0, \quad j = 1, \dots, m \end{array}$$

and (P_m) can then be solved by invoking standard algorithms for constrained smooth optimization, its particular structure has been exploited to obtain more efficient algorithms (see, *e.g.*, [17,34,35,50,74] for unconstrained minimax problems and see, *e.g.*, [43,65,66] for constrained minimax problems). The issue of feasibility has been studied for many decades. However, most existing feasible direction methods enjoy very slow rate of convergence (see, *e.g.*, [43,66,93]). Algorithms that achieve both feasibility and fast local convergence recently were proposed [6,64], but they did not account for nondifferentiable objective functions such as the *max*-type. The only algorithm that solves (P) with many desirable features was given in [65]. However, its efficiency can be argued in the context of engineering applications. It would be valuable to

adapt advanced techniques, such as a more rational choice of the feasible descent direction proposed in [64] and the idea of nonmonotone decrease of the objective function proposed in [6], to the solution of (P) in order to improve the efficiency of the optimization process in the engineering design.

The presence of ω in (SIP) introduces the difficulty of an infinite number of cost and constraint functions. Exact evaluation of $\max_{\omega \in \Omega} \{f_i(x, \omega)\}$ for given x is very time-consuming. Many methods have been proposed for the solution of this problem. Globally convergent algorithms (algorithms generating “desirable” limiting values of design parameter from any starting point) are mostly based on the approximation to $\max_{\omega \in \Omega} \{f_i(x, \omega)\}$ by means of discretization. They do not usually achieve fast local convergence (convergence in the neighborhood of a solution) (see, *e.g.*, [29,37,62,68, 69]). Algorithms that enjoy fast local convergence, on the other hand, are mostly based on the characterization of maximizers of each function $f_i(x, \cdot)$ over Ω for given x in the neighborhood of a local solution of (SIP) . Under mild assumptions, the set of such maximizers contains a “small” number of points. The original problem is reduced to a problem at these maximizers and existing techniques, such as Newton’s method for unconstrained problem and Sequential Quadratic Programming (SQP) for constrained case, for achieving fast convergence can be applied to the reduced problem. However, global convergence of algorithms based on such reduction is not guaranteed in general (see, *e.g.*, [38,40,72]). To induce global convergence, typically, a globally convergent algorithm is used to steer the iterates to the neighborhood of a desirable point (see, *e.g.*, [32,72]). Exceptions can be found in, *e.g.*, [86] where both global and fast local convergence were achieved by a single algorithm, but the proposed algorithm is potentially expensive in terms of function evaluations, since exact evaluation of $\max_{\omega \in \Omega} \{f_i(x, \omega)\}$ at every trial point is required during a line search process which is used to induce global convergence. New methodologies have to be investigated to meet the demanding need of engineering applications.

1.3 Objectives

There are basically two issues in using optimization techniques to solve engineering problems. One is the need for powerful optimization algorithms in an engineering context. The other is the need for friendly design environment that eases the interaction between problem specifications via a designer and the optimization process. Much progress has been made to meet the second need. In the past decade, several

optimization-based design tools have emerged (see, *e.g.*, [7,23,47,71]). With the aid of advanced computer technology, these tools have played central role in various complex engineering designs. However, their applications to a wider range of engineering problems and their efficiency in solving real-world problems have been hindered by currently existing algorithms that usually do not account for distinctive features of engineering problems, as mentioned in previous section. And most of the underlying optimization algorithms also exhibit slow convergence in the neighborhood of a solution. New optimization algorithms have to be designed in order for these tools to assist engineering designs more adequately and more efficiently. This dissertation is devoted to the solution of problems (P) and (SIP) , along that direction. A number of algorithms are proposed for these two general problems and for problems which are derived from them that will reveal more direct connections, in our opinion, to special engineering fields.

The dissertation is organized as follows. Chapter 2 reviews some basic concepts of optimization that are repeatedly used in subsequent chapters. Chapter 3 deals with unconstrained minimax problems, *i.e.*, (P) with $m = 0$. Chapter 4 is devoted to the solution of problem (P) with feasible iterates. Chapter 5 re-examines the nonmonotone line search scheme used in Chapter 3 and Chapter 4, under slightly stronger assumptions. The solution of (SIP) by means of discretization is pursued in Chapter 6. Examples of feedback control systems design applications are provided in Chapter 7. A few final remarks are given in Chapter 8 with suggestions for future research. Two appendixes are provided to contain auxiliary materials.

The style for the presentation of optimization algorithms follows conventions in the optimization community: an algorithm is presented first and then convergence properties are analyzed which are followed by numerical results.

Chapter 2

Some Basic Concepts of Optimization

To facilitate expositions in subsequent chapters and to provide the minimum knowledge of optimization that is required for the reader to fully understand the material to be presented later, a number of basic concepts in constrained optimization are reviewed in this chapter.

Consider problem (P) . Given $x \in \mathbb{R}^n$, for some $\mu \in \mathbb{R}^p$ and $\lambda \in \mathbb{R}^m$, define the Lagrange function for this problem by

$$L(x, \mu, \lambda) = \sum_{i=1}^p \mu_i f_i(x) + \sum_{j=1}^m \lambda_j g_j(x). \quad (2.1)$$

An algorithm in our context is referred to as an iterative process of computation that yields a sequence of iterates $\{x_k\}$ with $k = 0, 1, 2, \dots$, based on certain rules.

2.1 Classical SQP scheme

The classical Sequential Quadratic Programming (SQP) was originally developed for problem (P) with $p = 1$. Specifically, starting from a point x_k at iteration k , an increment $d_k^0 \in \mathbb{R}^n$ can be defined for some symmetric matrix $H_k \in \mathbb{R}^{n \times n}$ by the minimum norm solution of the inequality constrained quadratic program $QP(x_k, H_k)$

$$\begin{aligned} \min_{d^0 \in \mathbb{R}^n} \quad & \frac{1}{2} \langle d^0, H_k d^0 \rangle + \langle \nabla f(x_k), d^0 \rangle \\ \text{s.t.} \quad & g_j(x_k) + \langle \nabla g_j(x_k), d^0 \rangle \leq 0, \quad j = 1, \dots, m \end{aligned}$$

and a new iterate is obtained by setting

$$x_{k+1} = x_k + d_k^0. \quad (2.2)$$

The matrices H_k approximate, in a certain manner, the second order derivative $\nabla_{xx}^2 L(x, \mu, \lambda)$, *i.e.*, the Hessian matrix of the Lagrange function with respect to x , since it is impossible to compute the true Hessian due to the involvement of unknown (μ and) λ in (2.1).

A variant of $QP(x_k, H_k)$ is to consider equality constrained quadratic program where only a subset of the linear constraints in $QP(x_k, H_k)$ is included as equality constraints (see, *e.g.*, [14,15,56]). However, this scheme does not prove potentially useful in our context and will not be touched upon in subsequent chapters.

2.2 Variable metric methods

One of the popular and efficient methods that have been used to obtain suitable H_k in the SQP scheme is the so-called variable metric methods. The idea is to accumulate the second order information based on available first order information. The required value of (μ and) λ could be approximated by the value of (μ_k and) λ_k associated with the inequality constraints from the solution of $QP(x_k, H_k)$ or by other updating approaches (see, *e.g.*, [56,80]). The most successful updating formula for H_k is the BFGS update (see, *e.g.*, [48]), which can be essentially described as follows. Given x_k , x_{k+1} and H_k , define the variable increment by

$$\zeta_k = x_{k+1} - x_k$$

and the gradient increment of the Lagrange function by

$$\eta_k = \nabla_x L(x_{k+1}, \mu_k, \lambda_k) - \nabla_x L(x_k, \mu_k, \lambda_k). \quad (2.3)$$

The new H_{k+1} is then obtained by

$$H_{k+1} = H_k - \frac{H_k \zeta_k (H_k \zeta_k)^T}{\langle \zeta_k, H_k \zeta_k \rangle} + \frac{\eta_k \eta_k^T}{\langle \eta_k, \zeta_k \rangle}, \quad (2.4)$$

where H_0 is a symmetric matrix (and thus so is H_k for all k). This formula has the nice feature that, if H_k is positive definite (the smallest eigenvalue of H_k is positive) and if $\langle \eta_k, \zeta_k \rangle > 0$, then H_{k+1} remains positive definite. The positive definiteness is desirable in the SQP scheme because, *e.g.*, the solution d_k^0 of $QP(x_k, H_k)$ is then uniquely defined.

However, it is impossible in general to achieve $\langle \eta_k, \zeta_k \rangle > 0$ in constrained optimization. Powell [75,76] suggested a simple modification as follows. η_k used in (2.4) is replaced by $\theta_k \eta_k + (1 - \theta_k) H_k \zeta_k$ with

$$\theta_k = \begin{cases} 1 & \text{if } \langle \eta_k, \zeta_k \rangle \geq 0.2 \langle \zeta_k, H_k \zeta_k \rangle \\ \frac{0.8 \langle \zeta_k, H_k \zeta_k \rangle}{\langle \zeta_k, H_k \zeta_k \rangle - \langle \eta_k, \zeta_k \rangle} & \text{otherwise.} \end{cases}$$

With this choice, it can be readily checked that H_{k+1} is always positive definite if H_k is. In practice, it has been observed that, at least close to a “desirable” point, θ_k is mostly 1. The resulting formula is often referred to as “BFGS with Powell’s modification”.

2.3 First order necessary conditions of optimality

$x^* \in \mathbb{R}^n$ is said to satisfy the first order necessary conditions of optimality for (P) if there exist vectors $\mu^* \in \mathbb{R}^p$ and $\lambda^* \in \mathbb{R}^m$ such that

$$\left. \begin{aligned} \nabla_x L(x^*, \mu^*, \lambda^*) &= 0 \\ \mu_i^* &= 0 \quad \forall i \text{ s.t. } f_i(x^*) < f(x^*) \\ \sum_{i=1}^p \mu_i^* &= 1 \quad \text{and} \quad \mu^* \geq 0 \\ \lambda_j^* g_j(x^*) &= 0, \quad j = 1, \dots, m \\ \lambda_j^* &\geq 0 \quad \text{and} \quad g_j(x^*) \leq 0, \quad j = 1, \dots, m. \end{aligned} \right\} \quad (2.5)$$

This definition is a generalization of the conventional definition for either a single objective function ($p = 1$) or unconstrained minimax problems ($m = 0$).

2.4 KKT point, stationary point and Lagrange multiplier vector

Assume x^* satisfies the first order necessary conditions of optimality. Then, following conventions, x^* is said to be a KKT point if $p = 1$ and is called a stationary point if $m = 0$. If $p > 1$ and $m > 0$, we will call x^* a generalized KKT (GKKT) point (there is no consistent term for this case in the literature). The term “optimal point” or “desirable point” will be used in this dissertation to refer to any one of these three kinds of point. μ^* and λ^* are both called Lagrange multiplier vector associated with x^* .

2.5 Strict complementary slackness

Given $x \in \mathbb{R}^n$, define

$$I(x) = \{i \in \{1, \dots, p\} : f_i(x) = f(x)\}$$

and

$$J(x) = \{j \in \{1, \dots, m\} : g_j(x) = 0\}.$$

$I(x)$ is the index set of active objective functions and $J(x)$ is the index set of active constraints.

The strict complementary slackness at an optimal point x^* refers to the condition that $\mu_i^* > 0$, $\forall i \in I(x^*)$ and $\lambda_j^* > 0$, $\forall j \in J(x^*)$.

2.6 Second order sufficiency conditions of optimality

The second order sufficiency conditions of optimality for (P) are satisfied at x^* if the first order necessary conditions of optimality are satisfied at x^* and if

$$\langle d, \nabla_{xx}^2 L(x^*, \mu^*, \lambda^*) d \rangle > 0, \quad \forall d \in \mathcal{S},$$

where

$$\mathcal{S} = \mathcal{S}_1 \cap \mathcal{S}_2$$

with

$$\begin{aligned} \mathcal{S}_1 = \{d : \langle \nabla f_i(x^*), d \rangle = \langle \nabla f_j(x^*), d \rangle, \\ \forall i, j \in I(x^*) \text{ s.t. } \mu_i^* > 0 \text{ and } \mu_j^* > 0\} \end{aligned}$$

and

$$\mathcal{S}_2 = \{d : \langle \nabla g_j(x^*), d \rangle = 0, \quad \forall j \in J(x^*) \text{ s.t. } \lambda_j^* > 0\}.$$

In almost all situations, this second order conditions are used in conjunction with the strict complementary slackness. Thus, both $\mathcal{S}_1$ and $\mathcal{S}_2$ can be simplified by

$$\mathcal{S}_1 = \{d : \langle \nabla f_i(x^*) - \nabla f_j(x^*), d \rangle = 0, \quad \forall i, j \in I(x^*)\}$$

and

$$\mathcal{S}_2 = \{d : \langle \nabla g_j(x^*), d \rangle = 0, \quad \forall j \in J(x^*)\}.$$

2.7 Global convergence

An algorithm that generates a sequence of $\{x_k\}$ is said to be globally convergent if, starting from any initial point x_0 , every accumulation point x^* of this sequence satisfies the first order necessary conditions of optimality.

Precautions have to be taken by the reader who is not familiar with optimization theory: such an x^* need not even be a local minimizer in general without further conditions.

It is worth noting that the classic SQP scheme with iterates updated by (2.2) is not guaranteed to have global convergence. There are many globalizing schemes that modify the updating of iterates so that global convergence is ensured. One of them is described in the following.

2.8 Armijo line search

To simplify expositions, we assume that $m = 0$ in the context of the classical SQP scheme. It can be checked that, if H_k is positive definite, d_k^0 obtained from $QP(x_k, H_k)$ is a descent direction for $f(\cdot)$ at x_k (i.e., $\langle \nabla f(x_k), d_k^0 \rangle < 0$, namely, $x_k + d_k^0$ decreases $f(\cdot)$ at x_k to the first order). Given $\alpha, \beta \in (0, 1)$, under mild assumptions, a steplength $t_k \in (0, 1]$, the largest number in the sequence $\{1, \beta, \beta^2, \dots\}$, can always be found so that

$$f(x_k + t_k d_k^0) \leq f(x_k) + \alpha t_k \langle \nabla f(x_k), d_k^0 \rangle. \quad (2.6)$$

The process for finding such a t_k is referred to as the Armijo line search (see, e.g., [48]). The new iterate is then defined by

$$x_{k+1} = x_k + t_k d_k^0.$$

The inequality (2.6) is occasionally referred to as Armijo criterion [2].

2.9 Rate of convergence

This concept is concerned with local convergence of an algorithm, i.e., the convergence of $\{x_k\}$ to an optimal point from its neighborhood. Only three such definitions are given below that are most related to the material in subsequent chapters. Interested readers are referred to [58] for classical theory on this topic and to, e.g., [4, 42, 76] and references therein for most recent developments. Assume that, at each iteration k , an

increment d_k^0 is available (for instance, obtained as the solution of $QP(x_k, H_k)$) and a new iterate x_{k+1} is constructed according to certain rules.

2.9.1 Linear rate of convergence

An algorithm is said to have $(q-)$ linear rate of convergence if

$$\lim_{n \rightarrow \infty} \sup_{k \geq n} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} < 1.$$

Linear rate is generally considered as a slow rate of convergence.

2.9.2 One-step superlinear rate of convergence

An algorithm is said to have one-step $(q-)$ superlinear rate of convergence if

$$x_{k+1} = x_k + d_k^0$$

for k large enough and

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0. \quad (2.7)$$

Algorithms enjoying one-step $(q-)$ superlinear rate of convergence are very desirable because, after each iteration close to a solution, the new iterate is significantly closer to the solution. In practice, $\|d_k^0\|$ is a good indication of how close x_k is to x^* since (2.7) implies

$$\lim_{k \rightarrow \infty} \frac{\|d_k^0\|}{\|x_k - x^*\|} = 1.$$

In other words, x_k is a good approximation to x^* whenever $\|d_k^0\|$ is small.

2.9.3 Two-step superlinear rate of convergence

An algorithm is said to have two-step $(q-)$ superlinear rate of convergence if

$$x_{k+1} = x_k + d_k^0 \quad (2.8)$$

for k large enough and

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+2} - x^*\|}{\|x_k - x^*\|} = 0. \quad (2.9)$$

In other words, after every two iterations, the new iterate is significantly closer to a local solution. Similarly to one-step superlinear rate of convergence, $\|d_k^0 + d_{k+1}^0\|$ is a good indication of how close x_k is to x^* since (2.8) and (2.9) imply

$$\lim_{k \rightarrow \infty} \frac{\|d_k^0 + d_{k+1}^0\|}{\|x_k - x^*\|} = 1,$$

i.e., x_k is a good approximation to x^* whenever $\|d_k^0 + d_{k+1}^0\|$ is small.

Both one-step and two-step rate of convergence are considered fast rates of convergence.

2.10 Maratos effect

In unconstrained smooth optimization, SQP method reduces to Newton like method. Algorithms based on the SQP scheme have fast convergence if H_k are appropriately updated and iterates are updated by (2.2). It can be proved that, if $\alpha < 0.5$ in the Armijo line search, $t_k = 1$ for k large enough and the new iterate is eventually obtained by (2.2). Therefore, the globalization via the Armijo line search has no effect on the local rate of convergence.

Unfortunately, this is not the case in general in constrained optimization. Essentially, $t_k = 1$ may be rejected by the Armijo line search criterion. This phenomenon may take place even if arbitrarily close to a solution. The truncation of stepsize in the line search is referred to as the Maratos effect (it was first observed by Maratos [51]). Effectively, any algorithms claimed to have fast convergence must avoid the Maratos effect.

Chapter 3

NLS for Unconstrained Minimax Problems

This chapter considers minimax problems without constraints, namely, problems of the form

$$(P_1) \quad \min_{x \in R^n} \max_{i=1, \dots, p} f_i(x).$$

Clearly, (P_1) is a special case of (P) and any algorithm that solves (P) should be able to solve (P_1) . However, (P_1) deserves particular treatment essentially for three reasons: (i) historically, it has been an interesting problem that many researchers have actively studied; (ii) practically, there is a large class of engineering problems that falls into this framework directly; and (iii) usually the resulting algorithm is much simpler. We choose to present the algorithm for (P_1) first due to its simplicity and ease of understanding. The extension of the proposed algorithm to linearly constrained minimax problems, *i.e.*, all g_j 's in (P) are affine, is straightforward and is briefly discussed in the end of this chapter.

3.1 Introduction

Classical treatment of this problem can be found in [19]. From the point of view of engineering applications, this framework has been used to solve problems in worst case circuit analysis [50], in statistical estimation [28], in data analysis of mechanical dimensioning and tolerancing [24] and in many other fields.

Several authors have proposed, among other approaches (e.g., [17,70,74]), extensions of the popular sequential quadratic programming (SQP) scheme to the minimax

framework (e.g., [16,34,35,55,59,87]). For $x \in \mathbb{R}^n$, define

$$f(x) = \max_{i=1,\dots,p} f_i(x). \quad (3.1)$$

Global convergence is usually insured by means of a line search, forcing a decrease of f at each iteration. Typically, under mild assumptions, these algorithms exhibit a local superlinear (or two-step superlinear) rate of convergence provided the step size is not truncated by the line search when a solution is approached. As explained in Chapter 1, the *max* type objective function is nondifferentiable at any point where two or more individual functions (so-called active functions) achieve the max value. This objective function introduces the difficulty, in addition to that mentioned in Chapter 1, that, as first observed by Maratos [51] in the context of constrained optimization, the full step does not yield a decrease of f in general, essentially due to the fact that step of one along the search direction ensures those active functions are agreed with only to the first order. Thus, the line search may prevent superlinear convergence to take place (the Maratos-like effect). Two types of techniques have been used, in the context of constrained smooth optimization, to circumvent this difficulty. In the watchdog technique [12], the full step of one is occasionally accepted even if the line search criterion may be violated; subsequent backtracking is performed if no significant decrease of the objective function has been made after a given number of iterations. In the “bending” techniques [27,52], backtracking is avoided by performing a search along an arc whose construction requires evaluation of individual functions at an auxiliary point; along this arc, those active functions are agreed with to the second order and the full step of one is accepted close to a solution. These two techniques can be easily extended to the minimax framework, as pointed out by Womersley and Fletcher [87] and by Conn and Li [17]. However, possible backtracking means waste of function and gradient evaluations during early iterations and function values at additional points are required by the bending technique at all iterations close to a solution. Neither of these two approaches is desirable in the context of engineering applications, because function evaluations may be very costly.

A few years ago, in the context of Newton method for smooth unconstrained optimization, Grippo, Lampariello and Lucidi [31] proposed a “nonmonotone” line search according to which the objective function is not forced to decrease at every iteration but merely every M iterations, where M is a freely selected positive integer. They showed that with such a line search global convergence is still guaranteed, and they pointed out that, as the full Newton step can then be taken earlier, convergence

may often be sped up. Their numerical tests were indeed very promising. Recently, it was shown that making use of a suitable extension of this scheme to smooth constrained optimization, in the framework of SQP with penalty function-based line search, has the additional advantage of automatically allowing a full step to be taken locally and thus avoiding the Maratos effect [63].

Many of the schemes that have been proposed for the solution of minimax problems can be viewed as follows. First (P) is replaced by the equivalent smooth constrained problem

$$(P') \quad \begin{array}{ll} \min_{x \in \mathbb{R}^n, x_{n+1} \in \mathbb{R}} & x_{n+1} \\ \text{s.t.} & f_i(x) \leq x_{n+1} \quad i = 1, \dots, p, \end{array}$$

and application of a constrained optimization algorithm to this problem is considered. The resulting iteration is then refined to exploit the structure of the problem. In particular, in the case of sequential quadratic programming, refinements include (i) line search on f rather than on a penalty function, (ii) constraints made tight at the end of each iteration, and (iii) estimation of a Hessian of size $n \times n$ instead of $(n+1) \times (n+1)$. The question thus arises here of whether similar refinements on the nonmonotone line search scheme of [63] are viable. Specifically, (i) does a nonmonotone line search in the “max” function f still enforce global convergence? (ii) does such a line search prevent the Maratos effect? It turns out that the answer to both questions are positive. In addition, apparently even more than in the smooth constrained case, nonmonotone line search in the minimax case leads to significantly improved results on numerical tests. In this chapter, a nonmonotone line search based algorithm is described and analyzed and numerical experiments on commonly used test problems are presented.

3.2 Algorithm

Our algorithm can be viewed as dealing with (P_1) in the same spirit as in [34,35] and [65]. Specifically, at iteration k , an SQP direction d_k^0 is first computed as the solution of the quadratic problem $QP_1(x_k, H_k)$ defined for $x_k \in \mathbb{R}^n$ and $H_k \in \mathbb{R}^{n \times n}$ symmetric positive definite by

$$\min_{d^0 \in \mathbb{R}^n} \quad \frac{1}{2} \langle d^0, H_k d^0 \rangle + f'(x_k, d^0)$$

where

$$f'(x_k, d^0) = \max_{i=1, \dots, p} \{f_i(x_k) + \langle \nabla f_i(x_k), d^0 \rangle\} - f(x_k), \quad (3.2)$$

a first order approximation to $f(x_k + d^0) - f(x_k)$ at x_k in direction d^0 . It is well known that, under suitable assumptions, the iteration obtained by setting

$$x_{k+1} = x_k + d_k^0 \quad (3.3)$$

converges superlinearly to a local optimal solution. It turns out (Theorem 3.10 below) that, given any $\alpha > 0$, this iteration satisfies

$$f(x_{k+1}) \leq f(x_{k-2}) - \alpha \langle d_k^0, H_k d_k^0 \rangle \quad (3.4)$$

for k large enough. This suggests that no Maratos effect would arise if global convergence was enforced by means of a line search criterion requiring that the stepsize t_k satisfy

$$f(x_k + t_k d_k^0) \leq \max_{\ell=0,1,2} \{f(x_{k-\ell})\} - \alpha t_k \langle d_k^0, H_k d_k^0 \rangle \quad (3.5)$$

where the “max” insures that a positive step will always be accepted in early iterations. Since

$$f'(x_k, d_k^0) = -\langle d_k^0, H_k d_k^0 \rangle + \sum_{i=1}^p \mu_{k,i} \{f_i(x_k) - f(x_k)\} \leq -\langle d_k^0, H_k d_k^0 \rangle, \quad (3.6)$$

it follows that (3.5) is less stringent than the Armijo type line search

$$f(x_k + t_k d_k^0) \leq \max_{\ell=0,1,2} f(x_{k-\ell}) + \alpha t_k f'(x_k, d_k^0).$$

As f is not required to decrease at each iteration, such line search is referred to as a nonmonotone line search. It is known to induce global convergence when f is smooth [31]; we show below (Theorem 3.4) that it still does here. In view of (3.4), the nonmonotone line search criterion would preserve the superlinear convergence provided the “undamped” iteration (3.3) has been used for the last two iterations. To this end, following [6,63], we propose to initialize this procedure,¹ whenever $t_k = 1$ does not satisfy (3.5), by performing an arc search based on a correction $\tilde{d}_k$ so that a stepsize t_k is determined to satisfy

$$f(x_k + t_k d_k^0 + t_k^2 \tilde{d}_k) \leq \max_{\ell=0,1,2} f(x_{k-\ell}) - \alpha t_k \langle d_k^0, H_k d_k^0 \rangle. \quad (3.7)$$

$\tilde{d}_k$ will be chosen in such a way as to guarantee that (i) $t_k = 1$ is accepted in (3.7) for k large enough, where $\alpha \in (0, \frac{1}{2})$; and (ii) $d_k^0 + \tilde{d}_k$ converges to d_k^0 in order to

¹This is not required if $p = 1$.

preserve the properties of the SQP direction. Such $\tilde{d}_k$ can be chosen, for instance, as the solution $\tilde{d}$ of the quadratic program $\widetilde{QP}_1(x_k, d_k^0, H_k)$ given by

$$\min_{\tilde{d} \in \mathbb{R}^n} \frac{1}{2} \langle (d_k^0 + \tilde{d}), H_k(d_k^0 + \tilde{d}) \rangle + \tilde{f}'(x_k + d_k^0, x_k, \tilde{d}) \quad (3.8)$$

if $\|\tilde{d}\| \leq \|d_k^0\|$, and zero otherwise. In (3.8),

$$\tilde{f}'(x_k + d_k^0, x_k, \tilde{d}) = \max_{i=1, \dots, p} \{f_i(x_k + d_k^0) + \langle \nabla f_i(x_k), \tilde{d} \rangle\} - f(x_k + d_k^0).$$

It is shown below (Proposition 3.5) that $\tilde{d}_k$ obtained from (3.8) is always suitable for k large enough.

Algorithm 3.1.

Parameters. $\alpha \in (0, \frac{1}{2})$, $\beta \in (0, 1)$.

Data. $x_0 \in \mathbb{R}^n$, $H_0 \in \mathbb{R}^{n \times n}$ and $H_0 = H_0^T > 0$.

Step 0. Initialization. Set $k = 0$ and $x_{-2} = x_{-1} = x_0$.

Step 1. Computation of search direction and stepsize.

i. Compute d_k^0 by solving $QP_1(x_k, H_k)$. If $\|d_k^0\| = 0$, stop.

ii. If

$$f(x_k + d_k^0) \leq \max_{\ell=0,1,2} \{f(x_{k-\ell})\} - \alpha \langle d_k^0, H_k d_k^0 \rangle, \quad (3.9)$$

set $t_k = 1$, $\tilde{d}_k = 0$ and go to *Step 2*.

iii. Compute $\tilde{d}_k$ by solving $\widetilde{QP}_1(x_k, d_k^0, H_k)$.

If $\|\tilde{d}_k\| > \|d_k^0\|$, set $\tilde{d}_k = 0$.

iv. Compute t_k , the first number t in the sequence $\{1, \beta, \beta^2, \dots\}$ satisfying

$$f(x_k + t d_k^0 + t^2 \tilde{d}_k) \leq \max_{\ell=0,1,2} \{f(x_{k-\ell})\} - \alpha t \langle d_k^0, H_k d_k^0 \rangle. \quad (3.10)$$

Step 2. Updates.

Set

$$x_{k+1} = x_k + t_k d_k^0 + t_k^2 \tilde{d}_k.$$

Compute a new symmetric positive definite approximation H_{k+1} to the Hessian of the Lagrange function. Increase k by 1. Go back to *Step 1*.

□

Remark 3.1. Without *Step 1 ii*, the algorithm is a simple combination of Han's method (except that t_k is determined differently) and a second order correction to obtain superlinear convergence.

3.3 Convergence analysis

Given $x \in \mathbb{R}^n$, the index set of active functions at x is defined by

$$I(x) = \{i \in \{1, \dots, p\} : f_i(x) = f(x)\}.$$

The following standard assumptions are made throughout this dissertation.

A3.1. The functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, p$, are continuously differentiable.

A3.2. For any $x_0 \in \mathbb{R}^n$, the set $L = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is compact.

A3.3. There exist $\sigma_1, \sigma_2 > 0$ such that the matrices H_k satisfy

$$\sigma_1 \|x\|^2 \leq \langle x, H_k x \rangle \leq \sigma_2 \|x\|^2, \quad \forall x \in \mathbb{R}^n, \quad \forall k.$$

For problem (P_1) , the Lagrangian is defined by

$$L(x, \mu) = \sum_{i=1}^p \mu_i f_i(x).$$

As was defined in Chapter 2, a point $x^* \in X$ is stationary for (P_1) if there exist $\mu_i^* \geq 0, i = 1, \dots, p$, such that

$$\left. \begin{aligned} \nabla_x L(x^*, \mu^*) &= 0 \quad \& \quad \sum_{i=1}^p \mu_i^* = 1 \\ \mu_i^* &= 0 \quad \forall i \quad \text{s.t.} \quad f_i(x^*) < f(x^*). \end{aligned} \right\} \quad (3.11)$$

Clearly, any local solution of (P_1) is stationary. The first order necessary conditions of optimality for $QP_1(x_k, H_k)$ can be expressed as follows. If d_k^0 solves $QP_1(x_k, H_k)$, there exist $\mu_{k,i} \geq 0, i = 1, \dots, p$, such that

$$\left. \begin{aligned} H_k d_k^0 + \nabla_x L(x_k, \mu_k) &= 0 \quad \& \quad \sum_{i=1}^p \mu_{k,i} = 1 \\ \mu_{k,i} &= 0 \quad \forall i \quad \text{s.t.} \quad f_i(x_k) + \langle \nabla f_i(x_k), d_k^0 \rangle - f(x_k) < f'(x_k, d_k^0). \end{aligned} \right\} \quad (3.12)$$

3.3.1 Global convergence

In view of A3.1, A3.2 and A3.3, $QP_1(x_k, H_k)$ and $\widetilde{QP}_1(x_k, d_k^0, H_k)$ have unique and bounded (as k goes to ∞) solutions d_k^0 and $\tilde{d}_k$ respectively. The following lemma shows that d_k^0 is a direction of descent for f at x_k (see, *e.g.*, [35]). Its proof is given in Appendix A.

Lemma 3.1. The directional derivative $Df(x_k, d_k^0)$ of f at x_k along d_k^0 satisfies

$$Df(x_k, d_k^0) \leq -\langle d_k^0, H_k d_k^0 \rangle \quad \forall k,$$

and d_k^0 is zero if and only if x_k is a stationary point. $\square$

In view of A3.3, of the continuity of $f(x)$, and of the boundedness of d_k^0 and $\tilde{d}_k$, it follows from Lemma 3.1 that d_k^0 is a descent direction for f at x_k and a nonzero t_k can be always found in *step 1 iv* of the algorithm, thus the algorithm is well defined. Therefore, unless the algorithm stops at *Step 1 i* at a stationary point, it constructs an infinite sequence $\{x_k\}$. In the sequel, we assume the latter.

The following property, which holds even though monotone decrease of f is not enforced, is a key to global convergence. Although the underlying ideas of the proof are analogous to those used by Grippo *et al.* in the smooth unconstrained case [31], the details of the extension to the present situation are nontrivial.

Lemma 3.2. The sequence $\{x_k\}$ is bounded and the sequences $\{t_k d_k^0\}$ and $\{x_{k+1} - x_k\}$ both converge to zero.

Proof. Clearly, $f(x_k) \leq f(x_0)$ for all k . Thus the boundedness of $\{x_k\}$ follows from A3.2. Now, for k given, let $\ell(k)$ be an index such that

$$f(x_{\ell(k)}) = \max_{\ell=0,1,2} \{f(x_{k-\ell})\} = \max_{\ell=k-2, k-1, k} \{f(x_\ell)\}.$$

We first show that, for some $f^* \in R$,

$$f(x_{\ell(k)}) \rightarrow f^* \quad \text{as } k \rightarrow \infty. \quad (3.13)$$

For this, note that, in view of the definition of $\ell(k)$,

$$\begin{aligned} f(x_{\ell(k+1)}) &= \max_{\ell=k-1, k, k+1} \{f(x_\ell)\} \\ &\leq \max_{\ell=k-2, \dots, k+1} \{f(x_\ell)\} \\ &= \max\{f(x_{\ell(k)}), f(x_{k+1})\} \\ &= f(x_{\ell(k)}) \end{aligned}$$

since, in view of the construction of x_{k+1} in the algorithm, $f(x_{k+1}) \leq f(x_{\ell(k)})$. Thus $f(x_{\ell(k)})$ is nonincreasing. Since $x_k \in \Omega$ for all k , (3.13) then follows from A3.1 and A3.2.

Second we show that, for any integer j , the following implications hold:

$$f(x_{\ell(k)-j}) \rightarrow f^* \text{ as } k \rightarrow \infty \Rightarrow t_{\ell(k)-(j+1)} d_{\ell(k)-(j+1)}^0 \rightarrow 0 \text{ as } k \rightarrow \infty \quad (3.14)$$

and

$$f(x_{\ell(k)-j}) \rightarrow f^* \text{ as } k \rightarrow \infty \Rightarrow x_{\ell(k)-j} - x_{\ell(k)-(j+1)} \rightarrow 0 \text{ as } k \rightarrow \infty. \quad (3.15)$$

(Throughout the remainder of this proof, k is taken large enough for the indexes to make sense.) Indeed from the construction of x_{k+1} and in view of A3.3, we have

$$\begin{aligned} f(x_{\ell(k)-j}) &\leq f(x_{\ell(k)-(j+1)}) - \alpha t_{\ell(k)-(j+1)} \langle d_{\ell(k)-(j+1)}^0, H_{\ell(k)-(j+1)} d_{\ell(k)-(j+1)}^0 \rangle \\ &\leq f(x_{\ell(k)-(j+1)}) - \alpha \sigma_1 t_{\ell(k)-(j+1)} \|d_{\ell(k)-(j+1)}^0\|^2. \end{aligned}$$

In view of (3.13), the left hand side of (3.14) implies

$$f^* \leq f^* - \lim_{k \rightarrow \infty} \alpha \sigma_1 t_{\ell(k)-(j+1)} \|d_{\ell(k)-(j+1)}^0\|^2.$$

Thus

$$t_{\ell(k)-(j+1)} \|d_{\ell(k)-(j+1)}^0\|^2 \rightarrow 0 \text{ as } k \rightarrow \infty.$$

Since t_k is bounded, (3.14) follows. Since $\|\tilde{d}_k\| \leq \|d_k^0\|$ and $|t_k| \leq 1 \forall k$,

$$\|x_{\ell(k)-j} - x_{\ell(k)-(j+1)}\| \leq 2 t_{\ell(k)-(j+1)} \|d_{\ell(k)-(j+1)}^0\|$$

and (3.15) also follows.

Third we show by induction on j that, if j is any nonnegative integer,

$$f(x_{\ell(k)-j}) \rightarrow f^* \text{ as } k \rightarrow \infty. \quad (3.16)$$

In view of (3.13), (3.16) holds for $j = 0$. Suppose it holds for some $\hat{j}$. Then, from (3.15),

$$x_{\ell(k)-\hat{j}} - x_{\ell(k)-(\hat{j}+1)} \rightarrow 0 \text{ as } k \rightarrow \infty.$$

Since $\{x_k\}$ is bounded, continuity of f and the induction hypothesis imply

$$f(x_{\ell(k)-(\hat{j}+1)}) \rightarrow f^* \text{ as } k \rightarrow \infty$$

and this completes the proof of (3.16).

The proof of the lemma can now be readily completed. Indeed, (3.14), (3.15) and (3.16) imply that, for any nonnegative integer j ,

$$t_{\ell(k)-(j+1)} d_{\ell(k)-(j+1)}^0 \rightarrow 0 \quad \text{as } k \rightarrow \infty \quad (3.17)$$

$$x_{\ell(k)-j} - x_{\ell(k)-(j+1)} \rightarrow 0 \quad \text{as } k \rightarrow \infty. \quad (3.18)$$

From the fact (see definition of $\ell(k)$) that, for all k ,

$$\ell(k) - 1 \leq (k - 1) \leq \ell(k - 1) + 2$$

and

$$\ell(k) \leq k,$$

it follows that

$$\ell(k - 1) \leq \ell(k) \leq k \leq \ell(k - 1) + 3 \quad (3.19)$$

and thus the three subsequences (3.17) (resp. (3.18)) corresponding to $j = 0, 1, 2$ cover the entire sequence $\{t_k d_k^0\}$ (resp. $\{x_{k+1} - x_k\}$), so that

$$t_k d_k^0 \rightarrow 0 \quad \text{as } k \rightarrow \infty$$

$$x_{k+1} - x_k \rightarrow 0 \quad \text{as } k \rightarrow \infty.$$

□

The following result is extracted from above proof for later use.

Lemma 3.3. The sequence $\{f(x_k)\}$ converges.

Proof. It follows directly from (3.16) and (3.19) and A3.2. □

Theorem 3.4. Let x^* be an accumulation point of the sequence generated by the algorithm and let $K \subset \mathbb{N}$ be such that $\{x_k\}$ converges to x^* on K . Then, x^* is a stationary point of (P) and the sequence $\{d_k^0\}$ converges to zero on K .

Proof. We first show that $\{d_k^0\}$ converges to zero on K . Proceeding by contradiction, we assume there exists an infinite subset $K' \subset K$ such that $\inf_{k \in K'} \|d_k^0\| > 0$, i.e., there exists $\underline{d} > 0$ such that $\|d_k^0\| \geq \underline{d}, \forall k \in K'$. We show that there exists $\underline{t} > 0$ independent of k such that (3.9) or (3.10) is always satisfied for some $t_k \geq \underline{t}$ for all $k \in K'$. Expanding f_i at x_k gives

$$f_i(x_k + t d_k^0 + t^2 \tilde{d}_k) = f_i(x_k) + \langle \nabla f_i(x_k), t d_k^0 + t^2 \tilde{d}_k \rangle + o(t d_k^0 + t^2 \tilde{d}_k).$$

Thus, in view of (3.12) and the boundedness of d_k^0 and $\tilde{d}_k$, we have, for $t \in [0, 1]$ and $i = 1, \dots, p$,

$$\begin{aligned}
f_i(x_k + td_k^0 + t^2\tilde{d}_k) &= f_i(x_k) + t\langle \nabla f_i(x_k), d_k^0 \rangle + o(t) \\
&\leq f_i(x_k) + t \left\{ \max_{i=1, \dots, p} \{f_i(x_k) + \langle \nabla f_i(x_k), d_k^0 \rangle\} - f_i(x_k) \right\} + o(t) \\
&= (1-t)f_i(x_k) + t \sum_{i=1}^p \mu_{k,i} f_i(x_k) + t \sum_{i=1}^p \mu_{k,i} \langle \nabla f_i(x_k), d_k^0 \rangle + o(t) \\
&= (1-t)f_i(x_k) + t \sum_{i=1}^p \mu_{k,i} f_i(x_k) - t\langle d_k^0, H_k d_k^0 \rangle + o(t) \\
&\leq f(x_k) - t\langle d_k^0, H_k d_k^0 \rangle + o(t) \\
&\leq f(x_k) - \alpha t \langle d_k^0, H_k d_k^0 \rangle - t(1-\alpha)\langle d_k^0, H_k d_k^0 \rangle + o(t).
\end{aligned}$$

In view of A3.3 and the contradiction assumption, it follows that

$$f_i(x_k + td_k^0 + t^2\tilde{d}_k) \leq f(x_k) - \alpha t \langle d_k^0, H_k d_k^0 \rangle - t(1-\alpha)\sigma_1 \underline{d}^2 + o(t).$$

Therefore, since $\alpha < 1$, there exist $\underline{t}_i > 0$ independent of k such that, for all $t \in [0, \underline{t}_i]$,

$$f_i(x_k + td_k^0 + t^2\tilde{d}_k) \leq f(x_k) - \alpha t \langle d_k^0, H_k d_k^0 \rangle, \quad i = 1, \dots, p.$$

If we choose $\underline{t} = \beta \min_{i=1, \dots, p} \underline{t}_i$, then, for all $k \in K'$ at which a stepsize t_k is obtained via a line search, $t_k \geq \underline{t}$. Therefore, $\{t_k d_k^0\}$ is uniformly bounded from below on K' by $\underline{t} \underline{d}$, a contradiction to Lemma 3.2. Thus, $\{d_k^0\}$ converges to zero on K .

Now, since $\{\mu_k\}$ is in the compact set

$$U = \{\mu \in \mathbb{R}^p : \sum_{i=1}^p \mu_i = 1 \text{ \& } \mu_i \geq 0 \text{ } \forall i = 1, \dots, p\},$$

there exist $K' \subset K$ and $\mu^* \in U$ such that $\{\mu_k\}$ converges to μ^* on K' . Taking limit of (3.12) on K' , in view of A3.3, it follows that $\{x_k, d_k^0, \mu_k\}$ converges on K' to $(x^*, 0, \mu^*)$ and (x^*, μ^*) satisfies (3.11). Therefore, x^* is stationary. $\square$

3.3.2 Superlinear convergence

Assumption A3.1 is replaced by the following stronger one.

A3.1'. The functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, p$, are three times continuously differentiable.

Let x^* be an accumulation point of $\{x_k\}$ and let $\mu_i^*, i = 1, \dots, p$, be the corresponding Lagrangian multipliers. The following assumptions are used in the analysis of local convergence.

A3.4. At x^* , any scalars $\mu_i, i \in I(x^*)$ satisfying

$$\sum_{i \in I(x^*)} \mu_i \nabla f_i(x^*) = 0 \quad \& \quad \sum_{i \in I(x^*)} \mu_i = 0$$

must all be zero.

A3.5. The second order sufficiency conditions with strict complementary slackness are satisfied at x^* , i.e., $\mu_i^* > 0 \forall i \in I(x^*)$ and

$$\langle h, \nabla_{xx}^2 L(x^*, \mu^*) h \rangle > 0 \quad \forall h \in S^*, h \neq 0,$$

with

$$S^* = \{h : \langle h, \nabla f_i(x^*) \rangle = \langle h, \nabla f_j(x^*) \rangle \quad \forall i, j \in I(x^*)\}.$$

Proposition 3.5. (i) The entire sequence $\{x_k\}$ converges to x^* and the entire sequence $\{d_k^0\}$ converges to zero; (ii) the multiplier vector μ_k associated with the solution d_k^0 of $QP_1(x_k, H_k)$ converges to μ^* and, for k large enough,

$$\{i : \mu_{k,i} > 0\} = I(x^*); \tag{3.20}$$

(iii)

$$\|\tilde{d}_k\| = O(\|d_k^0\|^2). \tag{3.21}$$

Proof. The argument for (i) and (ii) is standard and thus is left out. For (iii), it can be shown in view of (i) and (ii) and the stated assumptions that, for k large enough, it holds that

$$\{i : \tilde{\mu}_{k,i} > 0\} = I(x^*) \tag{3.22}$$

with $(\tilde{d}_k, \tilde{\mu}_k)$ the solution of $\widetilde{QP}_1(x_k, d_k^0, H_k)$.

In view of (3.20) and (3.12), the unique solution (d_k^0, μ_k) of $QP_1(x_k, H_k)$ is also the unique solution of the linear system in (d, μ)

$$\left. \begin{aligned} H_k d + \sum_{i \in I(x^*)} \mu_i \nabla f_i(x_k) &= 0 \\ \sum_{i \in I(x^*)} \mu_i &= 1, \quad \mu_i = 0 \quad \forall i \notin I(x^*) \\ f_i(x_k) + \langle \nabla f_i(x_k), d \rangle &= f_j(x_k) + \langle \nabla f_j(x_k), d \rangle, \quad \forall i, j \in I(x^*). \end{aligned} \right\} \tag{3.23}$$

Similarly, $(\tilde{d}_k, \tilde{\mu}_k)$ is also the unique solution of the linear system in $(\tilde{d}, \tilde{\mu})$

$$\left. \begin{aligned} H_k(d_k^0 + \tilde{d}) + \sum_{i \in I(x^*)} \tilde{\mu}_i \nabla f_i(x_k) &= 0 \\ \sum_{i \in I(x^*)} \tilde{\mu}_i &= 1, \quad \tilde{\mu}_i = 0 \quad \forall i \notin I(x^*) \\ f_i(x_k + d_k^0) + \langle \nabla f_i(x_k), \tilde{d} \rangle &= f_j(x_k + d_k^0) + \langle \nabla f_j(x_k), \tilde{d} \rangle, \\ &\quad \forall i, j \in I(x^*). \end{aligned} \right\} \quad (3.24)$$

By expanding $f_i(x_k + d_k^0)$, $i \in I(x^*)$, to second order around x_k , (3.24) is equivalent to the linear system

$$\left. \begin{aligned} H_k(d_k^0 + \tilde{d}) + \sum_{i \in I(x^*)} \tilde{\mu}_i \nabla f_i(x_k) &= 0 \\ \sum_{i \in I(x^*)} \tilde{\mu}_i &= 1, \quad \tilde{\mu}_i = 0 \quad \forall i \notin I(x^*) \\ f_i(x_k) + \langle \nabla f_i(x_k), d_k^0 + \tilde{d} \rangle + O(\|d_k^0\|^2) &= \\ f_j(x_k) + \langle \nabla f_j(x_k), d_k^0 + \tilde{d} \rangle + O(\|d_k^0\|^2), &\quad \forall i, j \in I(x^*). \end{aligned} \right\} \quad (3.25)$$

The only difference between (3.23) and (3.25) viewed as systems of equations in unknown (d, μ) and $(d_k^0 + \tilde{d}, \tilde{\mu})$ respectively is a perturbation of order $O(\|d_k^0\|^2)$. In view of A3.4, both systems are nonsingular for k large enough and claim (iii) then follows from the implicit function theorem. $\square$

Now, without loss of generality, assume that $I(x^*) = \{1, \dots, r\}$ for some r and define, for any $j \in I(x^*)$, $\bar{f}_j(x) = [f_i(x) - f_j(x) : \forall i \in I(x^*) \setminus \{j\}]^T \in \mathbb{R}^{r-1}$.

A3.6. H_k approximates the Hessian of the Lagrangian at x^* in the sense that

$$\lim_{k \rightarrow \infty} \frac{\|P_k \{H_k - \nabla_{xx}^2 L(x^*, \mu^*)\} P_k d_k^0\|}{\|d_k^0\|} = 0, \quad (3.26)$$

where the matrices P_k are defined by

$$P_k = I - R_k(R_k^T R_k)^{-1} R_k^T$$

with $R_k = \frac{\partial \bar{f}_j}{\partial x}(x_k)$ (in view of A3.4, $R_k^T R_k$ is invertible for k large enough).

Remark 3.2. Note that elementary column operations on R_k do not affect P_k . Thus, P_k is unchanged if R_k is replaced by $\frac{\partial \bar{f}_j}{\partial x}(x_k)$ for any $j \in I(x^*)$.

Assumption A3.6 has been observed to often hold, *e.g.*, under some conditions, when H_k is updated using Powell's modification of the BFGS formula (see [75]). In the presence of the strong properties stated in Proposition 3.5, it ensures that the iteration is close enough to the SQP iteration that a full step is eventually accepted by the line search, namely,

Proposition 3.6. For k large enough, $t_k = 1$.

The following lemma, proven in Appendix A, is used to facilitate the proof of Proposition 3.6.

Lemma 3.7. The SQP direction d_k^0 admits the following decomposition

$$d_k^0 = P_k d_k^0 + d'_k \quad (3.27)$$

where $d'_k = R_k(R_k^T R_k)^{-1} \bar{f}_1(x_k)$. Also, there exists $c_1 > 0$ such that, for k large enough,

$$\|\bar{f}_j(x_k)\| \geq c_1 \|d'_k\|, \quad j = 1, \dots, r. \quad (3.28)$$

Furthermore, there exists $c_2 > 0$ such that, for k large enough and for all $j_k \in I(x^*)$ such that $f_{j_k}(x_k) = f(x_k)$,

$$\langle \bar{f}_{j_k}(x_k), \bar{\mu}_k \rangle \leq -c_2 \|d'_k\| \quad (3.29)$$

where $\bar{\mu}_k = [\mu_{k,i} : \forall i \in I(x^*) \setminus \{j_k\}]^T \in \mathbb{R}^{r-1}$, with components in the same order as those of $\bar{f}_{j_k}(x_k)$. $\square$

Proof of Proposition 3.6. Throughout the proof, the phrase “for k large enough” is implicit. We show that

$$f(x_k + d_k^0 + \tilde{d}_k^0) \leq f(x_k) - \alpha \langle d_k^0, H_k d_k^0 \rangle, \quad (3.30)$$

which clearly implies the claim. In view of (3.22), it follows that

$$f_i(x_k + d_k^0) + \langle \nabla f_i(x_k), \tilde{d}_k \rangle = f_j(x_k + d_k^0) + \langle \nabla f_j(x_k), \tilde{d}_k \rangle \quad \forall i, j \in I(x^*).$$

This implies, in view of A3.1' and (3.21), that

$$f_i(x_k + d_k^0 + \tilde{d}_k) = f_j(x_k + d_k^0 + \tilde{d}_k) + O(\|d_k^0\|^3) \quad \forall i, j \in I(x^*),$$

which in turn implies

$$f(x_k + d_k^0 + \tilde{d}_k) = f_i(x_k + d_k^0 + \tilde{d}_k) + O(\|d_k^0\|^3) \quad \forall i \in I(x^*).$$

Multiplying both sides of this equation by the corresponding $\mu_{k,i}$ and summing up over all $i \in I(x^*)$ yield

$$\begin{aligned} f(x_k + d_k^0 + \tilde{d}_k) &= \sum_{i \in I(x^*)} \mu_{k,i} f_i(x_k + d_k^0 + \tilde{d}_k) + O(\|d_k^0\|^3) \\ &= L(x_k + d_k^0 + \tilde{d}_k, \mu_k) + O(\|d_k^0\|^3). \end{aligned}$$

Expanding $L(x_k + d_k^0 + \tilde{d}_k, \mu_k)$ around x_k gives

$$\begin{aligned} f(x_k + d_k^0 + \tilde{d}_k) &= L(x_k, \mu_k) + \langle \nabla_x L(x_k, \mu_k), d_k^0 + \tilde{d}_k \rangle \\ &\quad + \frac{1}{2} \langle d_k^0, \nabla_{xx}^2 L(x_k, \mu_k) d_k^0 \rangle + O(\|d_k^0\|^3). \end{aligned} \quad (3.31)$$

Since, for any $j \in I(x^*)$,

$$\begin{aligned} L(x_k, \mu_k) &= \sum_{i \in I(x^*)} \mu_{k,i} f_i(x_k) = f_j(x_k) + \sum_{i \in I(x^*)} \mu_{k,i} \{f_i(x_k) - f_j(x_k)\} \\ &= f_j(x_k) + \langle \bar{f}_j(x_k), \bar{\mu}_k \rangle \end{aligned}$$

and, in view of (3.12) and (3.21),

$$\langle \nabla_x L(x_k, \mu_k), d_k^0 + \tilde{d}_k \rangle = -\langle d_k^0, H_k d_k^0 \rangle + O(\|d_k^0\|^3),$$

(3.31) becomes

$$\begin{aligned} f(x_k + d_k^0 + \tilde{d}_k) &= f_{j_k}(x_k) + \langle \bar{f}_{j_k}(x_k), \bar{\mu}_k \rangle - \langle d_k^0, H_k d_k^0 \rangle \\ &\quad + \frac{1}{2} \langle d_k^0, \nabla_{xx}^2 L(x_k, \mu_k) d_k^0 \rangle + O(\|d_k^0\|^3), \end{aligned}$$

with j_k such that $f_{j_k}(x_k) = f(x_k)$. It follows that, in view of Lemma 3.7,

$$\begin{aligned} f(x_k + d_k^0 + \tilde{d}_k) &\leq f(x_k) - c_2 \|d_k^0\| - \frac{1}{2} \langle d_k^0, H_k d_k^0 \rangle \\ &\quad + \frac{1}{2} \langle d_k^0, \{\nabla_{xx}^2 L(x_k, \mu_k) - H_k\} d_k^0 \rangle + O(\|d_k^0\|^3) \\ &= f(x_k) - \alpha \langle d_k^0, H_k d_k^0 \rangle - c_2 \|d_k^0\| - \left(\frac{1}{2} - \alpha\right) \langle d_k^0, H_k d_k^0 \rangle \\ &\quad + \frac{1}{2} \langle d_k^0, P_k \{\nabla_{xx}^2 L(x_k, \mu_k) - H_k\} P_k d_k^0 \rangle \\ &\quad + o(\|d_k^0\|) + O(\|d_k^0\|^3). \end{aligned}$$

Therefore, (3.30) follows in view of (3.26) and A3.3, since $\alpha \in (0, \frac{1}{2})$ and $c_2 > 0$. $\square$

Next, because the correction $\tilde{d}_k$ is small (see (3.21)), A3.6 together with Proposition 3.6 implies two-step superlinear convergence in the present context, as it does when the unperturbed SQP iteration is used (see, e.g., [76, 79]).

Theorem 3.8. Under the stated assumptions, the convergence rate of the sequence $\{x_k\}$ is two-step superlinear, i.e.,

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+2} - x^*\|}{\|x_k - x^*\|} = 0.$$

Proof. The proof is a direct simplification of that of Theorem 4.5 in Chapter 4.

$\square$

Finally, and most importantly, as mentioned in the introduction, two-step superlinear convergence implies that, for k large enough,

$$f(x_k + d_k^0) \leq f(x_{k-2}) - \alpha \langle d_k^0, H_k d_k^0 \rangle$$

and thus *Step 1 iii* of the algorithm is eventually bypassed. The proof of this results involves the following lemma which is established in [27] for nonsmooth optimization. The proof modified for current case is given in Appendix A.

Lemma 3.9. There exists $c_3 > 0$ such that, for all x close to x^* ,

$$f(x) - f(x^*) \geq c_3 \|x - x^*\|^2.$$

□

Theorem 3.10. For k large enough, $x_k + d_k^0$ is always accepted and *Step 1 iii* (computation of $\tilde{d}_k$) is not performed.

Proof. As suggested above, we show that, for k large enough,

$$f_i(x_k + d_k^0) \leq f(x_{k-2}) - \alpha \langle d_k^0, H_k d_k^0 \rangle \quad \forall i = 1, \dots, p. \quad (3.32)$$

In view of the continuity of f and of Proposition 3.5(i), it follows that, for k large enough,

$$f(x_k + d_k^0) = \max_{i \in I(x^*)} \{f_i(x_k + d_k^0)\}. \quad (3.33)$$

Therefore, it suffices to prove (3.32) for all $i \in I(x^*)$. Let $i, j \in I(x^*)$. Expanding $f_i(x_k + d_k^0)$ around x^* gives, in view of A3.1' and (3.11),

$$\begin{aligned} f_i(x_k + d_k^0) &= f_i(x^*) + \langle \nabla f_i(x^*), x_k + d_k^0 - x^* \rangle + O(\|x_k + d_k^0 - x^*\|^2) \end{aligned} \quad (3.34)$$

$$\begin{aligned} &= f_i(x^*) - \sum_{\ell \in I(x^*)} \mu_\ell^* \langle \nabla f_\ell(x^*) - \nabla f_i(x^*), x_k + d_k^0 - x^* \rangle \\ &\quad + O(\|x_k + d_k^0 - x^*\|^2). \end{aligned} \quad (3.35)$$

Since $f_i(x^*) = f_j(x^*)$, (3.34) implies

$$\begin{aligned} f_j(x_k + d_k^0) - f_i(x_k + d_k^0) &= \\ &\langle \nabla f_i(x^*) - \nabla f_j(x^*), x_k + d_k^0 - x^* \rangle + O(\|x_k + d_k^0 - x^*\|^2). \end{aligned} \quad (3.36)$$

On the other hand, expanding $f_i(x_k + d_k^0)$ around x_k gives

$$f_i(x_k + d_k^0) = f_i(x_k) + \langle \nabla f_i(x_k), d_k^0 \rangle + O(\|d_k^0\|^2)$$

which implies, in view of (3.12) and A3.1',

$$f_j(x_k + d_k^0) - f_i(x_k + d_k^0) = O(\|d_k^0\|^2) \quad \forall i, j \in I(x^*). \quad (3.37)$$

By substituting (3.37) and (3.36) in (3.35), we obtain

$$f_i(x_k + d_k^0) = f_i(x^*) + O(\|d_k^0\|^2) + O(\|x_k + d_k^0 - x^*\|^2) \quad \forall i \in I(x^*).$$

Therefore, in view of (3.33), A3.3, Lemma 3.9 and Theorem 3.8, the above expression implies

$$\begin{aligned} f(x_k + d_k^0) &= f(x^*) + O(\|d_k^0\|^2) + O(\|x_k + d_k^0 - x^*\|^2) \\ &= f(x^*) - \alpha \langle d_k^0, H_k d_k^0 \rangle + O(\|d_k^0\|^2) + O(\|x_k + d_k^0 - x^*\|^2) \\ &\leq f(x_{k-2}) - \alpha \langle d_k^0, H_k d_k^0 \rangle - c_3 \|x_{k-2} - x^*\|^2 \\ &\quad + O(\|d_k^0\|^2) + O(\|x_k + d_k^0 - x^*\|^2) \\ &= f(x_{k-2}) - \alpha \langle d_k^0, H_k d_k^0 \rangle - c_3 \|x_{k-2} - x^*\|^2 + o(\|x_{k-2} - x^*\|^2). \end{aligned} \quad (3.38)$$

Therefore (3.32) holds. $\square$

We have established satisfactory convergence properties of Algorithm 3.1. The following section provides limited numerical results that measures up to these analytic results.

3.4 Numerical experiments

As can be seen easily, the presence of linear constraints does not increase the complexity of the algorithm and a set of linearly constrained minimax problems has been included in our test. Also, for some problems, the *max* function takes the form $\max_{i=1,\dots,p} |f_i(x)|$. Our algorithm is modified so as to take this into account. In the implementation, $\alpha = 0.1$, $\beta = 0.5$, and H_k is updated by means of the BFGS formula with Powell's modification [75], with $H_0 = I$ the identity matrix.

Results obtained on selected minimax problems are summarized in Table 3.1. In the table, **PROB** indicates the test problem and **CD** indicates the code names. **NF** stands for the number of objective functions, **NCMF** for the number of evaluations of the max function, **NCFI** for the total number of individual function evaluations ($\leq \text{NCMF} \times \text{NF}$), **IT** for the number of iterations, **OBJMAX** for the (absolute) max value of the objective functions, **KKT** for the norm of the KKT vector (the gradient of the Lagrangian at the final iterate) and **EPS** stands for the stopping criterion as explained

below. An asterisk * is placed if the corresponding information is not available. All computations were performed on a SUN 4/SPARC station 1. Gradients were computed by finite differences (for the i th component, the perturbation parameter was $2 \times 10^{-8} \max\{1, |x_{k,i}|\}$). Problems BARD, DAV2, F&R, HETT, and WA are from [84]; CB2, CB3, R-S, WONG and COLV are from [13, Examples 5.1-5]; M1 to M8 are from [49, Examples 1-8]. Some of these test problems allow one to freely select the number of variables; problems WA-6 and WA-20 correspond to 6 and 20 variables, respectively, and M8-10, M8-30 and M8-50 to 10, 30 and 50 variables respectively. Problems BARD down to WONG are unconstrained and M1 down to M8 are linearly constrained minimax problems. In Table 3.1, the performance of Algorithm 3.1 (denoted in the table by NL) is compared with that of the same algorithm with an Armijo line search (AL), *i.e.*, without *Step 1 ii* and $\ell \equiv 0$ in (3.10) in Algorithm 3.1, and with that of algorithms proposed in [17] (CL) and [49] (MS). To make such comparison meaningful, we attempted to best approximate the stopping rule used in each of the references. Thus (i) for problems BARD down to WONG, execution was terminated when $\|d_k^0\|$ was smaller than the corresponding value in the EPS column, and (ii) for problems M1 down to M8-50, execution was terminated when $\|d_k^0\|$ was smaller than $\|x_k\|$ times the corresponding value in the EPS column. As pointed out by Madsen and Schjær-Jacobsen, all their problems cited here except M2 satisfy Haar's condition.

The following observations can be made. First, NL performs much better than AL in terms of the number NCMF of function evaluations. Second, it compares well with other algorithms. WA20 is peculiar since from iteration 20 on, the 14 significant digits printed out by our code do not change. On problems M1-M8 for which the Haar condition holds, the performance of NL appears to be comparable to that of the algorithm of [49].

3.5 Alternative line search rules

It is not hard to show that all the theoretical results would hold if line search (3.10) were replaced by monotone line search. Yet, in conjunction with (3.9) which is essential, it is natural to use the former. (Note that Grippo *et al.* used it merely to perform larger steps.) It is easy to check that if $\ell = 0, 1, 2$ were replaced by $\ell = 0, \dots, M$ for some arbitrary positive integer $M \geq 2$, Lemma 3.2 would still be true and global convergence would still be guaranteed; $M \geq 2$ is needed for Theorem 3.10 to hold.

($M \equiv 0$ corresponds to a monotone line search as used in [16,34,35,55]; as discussed in the introduction, Theorem 3.10 would not hold in this case.)

A line search requiring a decrease of f by an amount proportional to $-\langle d_k^0, H_k d_k^0 \rangle$ was first used in [35] for minimax problems and Han argued there that a larger step would be allowed than if $f'(x_k, d_k^0)$ defined by (3.2) were used. In our context, if $f'(x_k, d_k^0)$ were used in the line search (see [6]), a similar analysis could be carried out with the difference that the “max” in the line search is over last four iterates ($\ell = 0, 1, 2, 3$) instead of three in order for Theorem 3.10 to hold. This can be easily checked as follows.

In view of (3.37) and Theorem 3.8, it follows that

$$\begin{aligned} f'(x_k, d_k^0) &= -\langle d_k^0, H_k d_k^0 \rangle + \sum_{i=1}^p \mu_{k,i} \{f_i(x_k) - f(x_k)\} \\ &= O(\|d_k^0\|^2) + O(\|d_{k-1}\|^2) \\ &= O(\|x_k - x^*\|^2) + O(\|x_{k-1} - x^*\|^2) \\ &= o(\|x_{k-3} - x^*\|^2) + o(\|x_{k-3} - x^*\|^2). \end{aligned}$$

The claim follows if $\alpha f'(x_k, d_k^0)$ is added to and subtracted from the right-hand side of (3.38). This claim applies to all algorithms in subsequent chapters.

PROB	CD	NF	NCF	NCFI	IT	OBJMAX	KKT	EPS
BARD	CL	15	10		*	*	*	.50E-05
	AL		15	168	8	.508163265E-01	.63E-10	.50E-05
	NL		7	105	7	.508168686E-01	.42E-05	.50E-05
CB2	CL	3	11		*	*	*	.50E-05
	AL		11	30	6	.195222453E+01	.10E-06	.50E-05
	NL		6	18	6	.195222453E+01	.82E-06	.50E-05
CB3	CL	3	6		*	*	*	.50E-05
	AL		5	15	3	.200000000E+01	.75E-06	.50E-05
	NL		5	15	5	.200000000E+01	.94E-09	.50E-05
COLV	CL	6	49		*	*	*	.50E-05
	AL		21	240	21	.323486790E+02	.29E-05	.50E-05
	NL		17	102	17	.323486790E+02	.14E-05	.50E-05
DAV2	CL	20	20		*	*	*	.50E-05
	AL		20	342	10	.115706440E+03	.59E-06	.50E-05
	NL		11	220	10	.115706440E+03	.93E-06	.50E-05
F&R	CL	2	11		*	*	*	.50E-05
	AL		17	32	9	.494895210E+01	.24E-06	.50E-05
	NL		10	20	10	.494895210E+01	.21E-06	.50E-05
HETT	CL	5	11		*	*	*	.50E-05
	AL		19	125	10	.245935695E-02	.28E-05	.50E-05
	NL		11	75	10	.245939485E-02	.19E-05	.50E-05
R-S	CL	4	12		*	*	*	.50E-05
	AL		22	71	9	-.440000000E+02	.13E-05	.50E-05
	NL		16	68	10	-.440000000E+02	.99E-07	.50E-05
WA6	CL	31	24		*	*	*	.50E-05
	AL		23	623	12	.127170954E-01	.14E-05	.50E-05
	NL		14	433	13	.127170913E-01	.31E-08	.50E-05
WA20	CL	31	22		*	*	*	.50E-05
	AL		106	1953	42	.138908355E-07	.35E-06	.50E-05
	NL		45	1023	43	.141191856E-07	.17E-06	.50E-05
WONG	CL	5			*	*	*	.50E-05
	AL		67	182	20	.680630057E+03	.12E-05	.50E-05
	NL		49	171	26	.680630057E+03	.42E-05	.50E-05
M1	MS	3	*		8	*	*	.10E-11
	AL		9	24	5	-.389659516E+00	.35E-16	.10E-11
	NL		6	18	6	-.389659516E+00	.89E-10	.10E-11
M2	MS	3	*		*	*	*	.10E-11
	AL		21	25	11	-.330357143E+00	.13E-10	.10E-11
	NL		19	21	18	-.330357143E+00	.81E-10	.10E-11
M4	MS	3	*		8	*	*	.10E-11
	AL		11	29	6	-.448910786E+00	.90E-16	.10E-11
	NL		8	24	8	-.448910786E+00	.90E-16	.10E-11
M5	MS	3	*		8	*	*	.10E-11
	AL		13	30	7	-.100000000E+01	.16E-16	.10E-11
	NL		8	24	8	-.100000000E+01	.35E-13	.10E-11
M6	MS	163	8		*	.113105 E+00	*	.10E-11
	AL		11	1084	6	.113104635E+00	.20E-10	.10E-11
	NL		8	1141	8	.113104727E+00	.72E-15	.10E-11
M8-10	MS	18	18		*	*	*	.10E-11
	AL		19	291	10	.381173963E+00	.99E-12	.10E-11
	NL		14	252	14	.381173963E+00	.22E-15	.10E-11
M8-30	MS	58	17		*	*	*	.10E-11
	AL		30	1373	15	.547620496E+00	.21E-15	.10E-11
	NL		20	1102	18	.547620496E+00	.21E-10	.10E-11
M8-50	MS	98	18		*	*	*	.10E-11
	AL		39	2930	20	.579276202E+00	.20E-15	.10E-11
	NL		21	2084	21	.579276202E+00	.22E-13	.10E-11

Table 3.1: Numerical Results for Minimax Problems

Chapter 4

NLS for Inequality Constrained Minimax Problems

To the best of our knowledge, the algorithm given in [65] was the first one that solves (P) with feasible iterates and with superlinear rate of convergence. However, the scheme for maintaining feasibility in the context of Armijo-type line search was significantly improved in [64], and the assumption of linear independence of active functions given in [65] needs clarifying since it is vital for establishing convergence properties. In this chapter, an algorithm for the solution of problem (P) in the context of nonmonotone line search is presented. The basic ideas behind the algorithm are not novel: the algorithm for a single objective function, *i.e.*, $p = 1$, has been completely analyzed in [6] and, for $p > 1$, suitable modifications on the algorithm has been proposed in the same paper without analytical results. The purpose of this chapter is to clarify some confusing issues in the literature and to foreshadow the presentation of subsequent chapters by introducing the framework of Feasible Sequential Quadratic Programming (FSQP) that has been thoroughly studied in, *e.g.*, [61,64]. Convergence properties are presented and their proofs are given in Appendix A. Most material of this chapter is taken from [64] and [6]. An efficient implementation of the algorithm is also discussed with numerical results.

4.1 Introduction

As mentioned in Chapter 1, one of the fundamental issues in the solution of (P) is to maintain feasibility of inequality constraints throughout the optimization process in the context of engineering applications. Feasible direction methods were proposed

and analyzed by several authors (see, *e.g.*, [43,66,93,94]). Classic first order feasible direction methods suffer from slow convergence. Recently, it was shown that the popular, superlinearly convergent Sequential Quadratic Programming (SQP) iteration could be adapted so as to generate feasible iterates [61,64]. The efficiency of the proposed feasible SQP (FSQP) algorithm was significantly improved in [6] by means of nonmonotone line search scheme. All these results were established for the case that the objective function is smooth. Their extension to minimax type objective function was suggested in [6]. In the following, we integrate all these ingredients (minimax function, feasibility, nonmonotone line search, *etc.*) into one algorithm.

Let f be defined by (3.1) and let X denote the feasible region, *i.e.*,

$$X = \{x \in \mathbb{R}^n : g_j(x) \leq 0 \quad \forall j = 0, \dots, m\}$$

with the following assumption.

A4.1. $g_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 0, \dots, m$, are continuously differentiable.

Ideally, one would expect that, whenever x_k is feasible, a line search enforcing global convergence was such that the two conditions

$$x_{k+1} \in X \tag{4.1}$$

$$f(x_{k+1}) \leq f(x_k) \tag{4.2}$$

could be achieved at each iteration. The classic SQP iteration consists of first computing a search iteration d_k^0 , at each iteration k for $x_k \in X$, by solving the quadratic program $QP_2(x_k, H_k)$

$$\begin{aligned} \min_{d^0 \in \mathbb{R}^n} \quad & \frac{1}{2} \langle d^0, H_k d^0 \rangle + f'(x_k, d^0) \\ \text{s.t.} \quad & g_j(x_k) + \langle \nabla g_j(x_k), d^0 \rangle \leq 0 \quad j = 1, \dots, m, \end{aligned}$$

with $H_k \in \mathbb{R}^{n \times n}$ a suitable symmetric positive definite estimate to the Hessian of the Lagrangian and $f'(\cdot, \cdot)$ defined in (3.2). It can be readily checked that d_k^0 is a descent direction for f at x_k . However, d_k^0 may not be feasible at x_k since $QP_2(x_k, H_k)$ allows $\langle \nabla g_j(x_k), d_k^0 \rangle = 0$ for some $g_j(x_k) = 0$. Also, even when d_k^0 is feasible, a line search that ensures both (4.1) and (4.2) may prevent a unit step from being accepted in the neighborhood of a solution, which is an ultimate requirement for superlinear convergence. Two simple mechanisms were proposed in [64] to circumvent these difficulties.

Specifically, the SQP direction d_k^0 is first replaced by the following convex combination

$$d_k = (1 - \rho_k) d_k^0 + \rho_k d_k^1 \tag{4.3}$$

for some (essentially arbitrary) feasible descent direction d_k^1 . For instance, d_k^1 can be chosen as the minimizer for the quadratic program

$$\begin{aligned} \min_{d^1 \in R^n, \xi \in R} \quad & \eta \langle d_k^0 - d^1, d_k^0 - d^1 \rangle + \xi \\ \text{s.t.} \quad & f'(x_k, d^1) \leq \xi \\ & g_j(x_k) + \langle \nabla g_j(x_k), d^1 \rangle \leq \xi \quad j = 1, \dots, m, \end{aligned}$$

with $\eta > 0$ a constant that ensures that the resulting direction d_k^1 is not too far away from d_k^0 . To ensure that d_k inherits nice convergence properties of d_k^0 , ρ_k has to go to zero fast enough. In order to make sure that both (4.1) and (4.2) are satisfied with step of one close to a solution, d_k is then suitably bent, *i.e.*, a search along an arc $x_k + td_k + t^2 \tilde{d}_k$ is performed. $\tilde{d}_k$ is selected in such a way that, (i) in a neighborhood of a solution x^* , $x_{k+1} (= x_k + d_k + \tilde{d}_k)$ satisfies both (4.1) and (4.2), (ii) $d_k + \tilde{d}_k$ converges to d_k , so as to preserve fast convergence properties of d_k^0 . To take advantage of the available second order information H_k , $\tilde{d}_k$ can be obtained as the solution of the quadratic program $\widetilde{QP}_2(x_k, d_k, H_k)$

$$\begin{aligned} \min_{\tilde{d} \in R^n} \quad & \frac{1}{2} \langle (d_k + \tilde{d}), H_k(d_k + \tilde{d}) \rangle + f'(x_k + d_k, x_k, \tilde{d}) \\ \text{s.t.} \quad & g_j(x_k + d_k) + \langle \nabla g_j(x_k), \tilde{d} \rangle \leq -\|d_k\|^\tau \quad j = 1, \dots, m, \end{aligned}$$

and set to zero if $\tilde{d}$ too large, where

$$f'(x_k + d_k, x_k, \tilde{d}) = \begin{cases} f'(x_k, d_k + \tilde{d}) & \text{if } p = 1 \\ f'(x_k + d_k, \tilde{d}) & \text{if } p > 1. \end{cases}$$

In $\widetilde{QP}_2(x_k, d_k, H_k)$, $\tau \in (2, 3)$ is required that ensures the correction is significant enough (τ on the smaller side, assuming $\|d_k\| < 1$ for k large enough) to restore the feasibility, yet $x_k + d_k + \tilde{d}_k$ is close enough to the boundaries of the active constraints to achieve descent of f (τ on the larger side).

Under mild assumptions, the outlined algorithm exhibits very satisfactory convergence properties. Note that three quadratic programs are solved at each iteration and constraint evaluation at auxiliary point $x_k + d_k$ is also required for the construction of $\tilde{d}_k$. In the context of engineering applications, oftentimes function evaluations are far more computationally demanding than solution of a quadratic program. Therefore, a major criticism may go to the request of $\tilde{d}_k$.

It turns out that, for given $\alpha > 0$, if a sequence of iterates $\{x_k\}$ is generated by the basic SQP iteration for (P) , *i.e.*, $x_{k+1} = x_k + d_k^0$, then, if x_0 is sufficiently close

to a strong local minimizer x^* for (P) and the entire sequence $\{x_k\}$ happens to be feasible,

$$f(x_{k+1}) \leq f(x_{k-2}) - \alpha \langle d_k^0, H_k d_k^0 \rangle \quad (4.4)$$

will hold for k large enough. Thus, a line search rule requiring that the condition

$$f(x_k + t d_k^0) \leq \max_{\ell=0,1,2} \{f(x_{k-\ell})\} - \alpha t \langle d_k^0, H_k d_k^0 \rangle$$

be satisfied would eventually *always* accept the full step of one, provided it does so *two times in a row*. Initialization of this process is by making use of Mayne-Polak's correction [52]. Again, locally around x^* , evaluation of the constraints at auxiliary points would not be necessary. Global convergence could be ensured by means of a nonmonotone line search. However, a major difficulty remains: the full step of one will likely be rejected due to infeasibility.

To ensure the feasibility of a full step of one close to a solution, consider a "local" direction d_k^ℓ such that

$$d_k^\ell = d_k^0 + O(\|d_k^0\|^2) \quad (4.5)$$

satisfying

$$g_j(x_k) + \langle \nabla g_j(x_k), d_k^\ell \rangle \leq -C \|d_k^0\|^2, \quad j = 1, \dots, m, \quad (4.6)$$

for given $C > 0$. It is easy to show that an inequality analogous to (4.4), namely

$$f(x_k + d_k^\ell) \leq f(x_{k-2}) - \alpha \langle d_k^0, H_k d_k^0 \rangle \quad (4.7)$$

still holds if the sequence $\{x_k\}$ is constructed via the iteration $x_{k+1} = x_k + d_k^\ell$. Under mild assumptions, when x_k is close to x^* , d_k^0 is small and, if the gradients of the active constraints at x^* are linearly independent, it is always possible to construct such d_k^ℓ . If (4.6) holds, the sequence $\{x_k\}$ constructed by the iteration $x_{k+1} = x_k + d_k^\ell$ will satisfy

$$\begin{aligned} g_j(x_{k+1}) &= g_j(x_k) + \langle \nabla g_j(x_k), d_k^\ell \rangle + \frac{1}{2} \langle d_k^\ell, \frac{\partial^2 g_j}{\partial x^2}(x_k + \xi_{j,k} d_k^\ell) d_k^\ell \rangle \\ &\leq -C \|d_k^0\|^2 + \frac{1}{2} \|d_k^\ell\|^2 \left\| \frac{\partial^2 g_j}{\partial x^2}(x_k + \xi_{j,k} d_k^\ell) \right\|, \quad j = 1, \dots, m \end{aligned}$$

for some $\xi_{j,k} \in [0, 1]$. And if (4.5) holds, for k large enough we will have $g_j(x_{k+1}) \leq 0$, $j = 1, \dots, m$, provided $2C$ is strictly larger than the largest among all eigenvalues of the Hessians $\frac{\partial^2 g_j}{\partial x^2}(x^*)$, $j = 1, \dots, m$. While these Hessians are obviously unknown, one could attempt to adaptively obtain a suitably large value of C , by increasing C

whenever the step of one along d_k^ℓ is not feasible. It will be shown below that this can indeed be achieved.

A final difficulty stems from the fact that, away from x^* , d_k^ℓ may not be a descent direction for f . And a descent property of a search direction is crucial for ensuring global convergence. Indeed, conditions (4.5) and (4.6) and the descent requirement (4.2) are usually incompatible. This can be addressed by resorting, whenever the full step of one along d_k^ℓ is not accepted for the descent criterion, to a search along the arc,

$$x(t) = x_k + td_k^g + t^2\tilde{d}_k$$

with d_k^g a “global” feasible descent direction and $\tilde{d}_k$ a second-order correction of the Mayne-Polak type. Following the idea of (4.3), both d_k^ℓ and d_k^g are as convex combination of d_k^0 and a direction $d^1(x_k)$ obtained via a certain map $d^1(\cdot)$. d_k^g should also comply with a similar rule to (4.5). In [64], $d^1(x)$ is essentially any feasible descent direction at x and thus vanishes at any Karush-Kuhn-Tucker (KKT) point. In this context however, to ensure that (4.6) can be achieved close to x^* , we will require that $d^1(x)$ be a (nonzero) direction of strict feasibility even at KKT points. Clearly then, one cannot any more require that $d^1(x)$ be a direction of descent for f . We are ready to present the FSQP algorithm for the solution of (P) .

4.2 Algorithm

In addition to A3.1 and A4.1, the following assumptions are required.

A4.2. For any $x_0 \in X$, the set $L = \{x \in X : f(x) \leq f(x_0)\}$ is compact.

A4.3. $\forall x \in X$, $\{\nabla g_j(x), j \in J(x)\}$ are linearly independent, where

$$J(x) = \{j \in \{1, \dots, m\} : g_j(x) = 0\}$$

is the index set of active constraints.

The index set of active objective functions is also defined by

$$I(x) = \{i \in \{1, \dots, p\} : f_i(x) = f(x)\}.$$

Remark 4.1. The assumption given in [65, page 68] states that “For any $x \in X$, the sets $\{\nabla_x f_i(x), i \in I(x)\}$ and $\{\nabla_x g_j(x), j \in J(x)\}$ are individually made up of linearly independent vectors,” which is misleading. In particular, for $J(x) = \emptyset$, the linear independence of the first set should not be required for establishing global convergence as demonstrated in Chapter 3. See also Remark 4.4.

As indicated above, Algorithm 4.1 below makes use of two directions, d_k^ℓ and d_k^g , both of which are convex combinations of d_k^0 and of a *feasible* direction $d^1(x_k)$ obtained via a map $d^1(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$. This map is required to be continuous¹ and to yield for every $x \in X$ (including KKT points) a direction $d^1(x)$ satisfying

$$g_j(x) + \langle \nabla g_j(x), d^1(x) \rangle < 0, \quad j = 1, \dots, m. \quad (4.8)$$

In view of assumption A4.2, such a direction can, for example, be obtained as the solution of the quadratic program $QP_2^1(x)$

$$\min_{d \in \mathbb{R}^n} \quad \eta \|d\|^2 + \max_{j=1, \dots, m} \{g_j(x) + \langle \nabla g_j(x), d \rangle\},$$

where $\eta > 0$ is a constant.

Further clarification is in order concerning the specific construction of d_k^ℓ and d_k^g . The “local” direction d_k^ℓ is constructed based on a constant C_k , corresponding to C in (4.6), which is iteratively adapted as suggested in the introduction. Essentially, it is increased if $\|d_k^0\|$ is reasonably small (indicating that x^* is nearby) but $x_k + d_k^\ell$ is not feasible. If $x_k + d_k^\ell$ is feasible, C_k is kept to its previous value and if $\|d_k^0\|$ is large, it is reset to some small value. Next, it is easily checked that if C_k remains bounded as $k \rightarrow \infty$ (which is true under mild assumptions) and d_k^ℓ is constructed according to

$$d_k^\ell = (1 - \rho_k^\ell) d_k^0 + \rho_k^\ell d^1(x_k),$$

with $\rho_k^\ell \in [0, 1]$ as small as possible subject to satisfaction of (4.6), the requirement (4.5) will be satisfied. Away from x^* however it may be impossible to satisfy (4.6) with $\rho_k^\ell \in [0, 1]$. A suitable choice in such case would be $\rho_k^\ell = 1$. In *Step 1 ii* in Algorithm 4.1 below, ρ_k^ℓ is constructed essentially according to these rules, with the additional feature that ρ_k^ℓ (and thus d_k^ℓ) is forced to go to zero whenever d_k^0 does, to preserve global convergence even if C_k tends to grow without bound. For the “global” direction d_k^g , the requirement is that it should be a feasible descent direction, and that

$$d_k^g = d_k^0 + O(\|d_k^0\|^2).$$

This can be achieved by selecting

$$d_k^g = (1 - \rho_k^g) d_k^0 + \rho_k^g d^1(x_k)$$

with $\rho_k^g \in [0, \rho_k^\ell]$, as large as possible subject to the condition

$$f'(x_k, d_k^g) \leq \theta f'(x_k, d_k^0),$$

¹For simplicity of exposition; the results still hold with milder assumptions

where $\theta \in (0, 1)$ is a fixed parameter. The second-order correction $\tilde{d}_k$ is defined by the solution of $\widetilde{QP}_2(x_k, d_k^g, H_k)$ and is set to zero if the solution is too large.

Algorithm 4.1.

Parameters. $\alpha \in (0, \frac{1}{2})$, $\beta \in (0, 1)$, $\theta \in (0, 1)$, $\gamma \in (2, 3)$, $\eta > 0$, $\underline{C} > 0$, $\underline{d} > 0$.

Data. $x_0 \in X$, $H_0 \in \mathbb{R}^{n \times n}$ and $H_0 = H_0^T$, $C_0 = \underline{C}$.

Step 0. Initialization. Set $k = 0$. Set $x_{-2} = x_{-1} = x_0$.

Step 1. Computation of a new iterate.

i. Compute d_k^0 by solving $QP_2(x_k, H_k)$. If $d_k^0 = 0$ stop.

ii. Compute d_k^1 by solving $QP_2^1(x_k)$, let

$$\varsigma_k = \min\{C_k \|d_k^0\|^2, \|d_k^0\|\}$$

and define values $\rho_{k,j}$ for $j = 1, \dots, m$ by $\rho_{k,j}$ equal to zero if

$$g_j(x_k) + \langle \nabla g_j(x_k), d_k^0 \rangle \leq -\varsigma_k$$

or equal the maximum ρ in $[0, 1]$ such that

$$g_j(x_k) + \langle \nabla g_j(x_k), (1 - \rho)d_k^0 + \rho d_k^1 \rangle \geq -\varsigma_k$$

otherwise. Finally, let $\rho_k^\ell = \max_{j=1, \dots, m} \{\rho_{k,j}\}$.

iii. Obtain a “local” direction

$$d_k^\ell = (1 - \rho_k^\ell)d_k^0 + \rho_k^\ell d_k^1.$$

iv. If

$$f(x_k + d_k^\ell) \leq \max_{\ell=0,1,2} \{f(x_{k-\ell})\} - \alpha \langle d_k^0, H_k d_k^0 \rangle \quad (4.9)$$

and

$$g_j(x_k + d_k^\ell) \leq 0, \quad j = 1, \dots, m, \quad (4.10)$$

set $t_k = 1$, $x_{k+1} = x_k + d_k^\ell$ and go to *Step 2*. Otherwise, go to *Step 1 v*.

v. Obtain a “global” direction

$$d_k^g = (1 - \rho_k^g) d_k^0 + \rho_k^g d_k^1,$$

where ρ_k^g is the largest number in $[0, \rho_k^\ell]$ such that

$$f'(x_k, d_k^g) \leq \theta f'(x_k, d_k^0). \quad (4.11)$$

vi. Compute $\tilde{d}_k$ by solving $\widetilde{QP}_2(x_k, d_k^g, H_k)$. If there is no solution or if $\|\tilde{d}_k\| > \|d_k^0\|$, set $\tilde{d}_k = 0$.

vii. Compute t_k , the first number t in the sequence $\{1, \beta, \beta^2, \dots\}$ satisfying

$$f(x_k + t d_k^g + t^2 \tilde{d}_k) \leq \max_{\ell=0,1,2} \{f(x_{k-\ell})\} - \alpha t \theta \langle d_k^0, H_k d_k^0 \rangle, \quad (4.12)$$

$$g_j(x_k + t d_k^g + t^2 \tilde{d}_k) \leq 0, \quad j = 1, \dots, m$$

and set $x_{k+1} = x_k + t_k d_k^g + t_k^2 \tilde{d}_k$.

Step 2. Updates.

i. Compute a new symmetric positive definite approximation H_{k+1} to the Hessian of the Lagrangian.

ii. If $\|d_k^0\| > \underline{d}$, set $C_k = \underline{C}$. Otherwise, if $g_j(x_k + d_k^\ell) \leq 0$, $j = 1, \dots, m$, set $C_{k+1} = C_k$. Otherwise, set $C_{k+1} = 2C_k$.

iii. Increase k by 1 and go back to *Step 1*.

□

Remark 4.2. In (4.12), $-\theta \langle d_k^0, H_k d_k^0 \rangle$ is used instead of $f'(x_k, d_k^g)$ to be consistent with (4.9). It can be checked that (cf. (3.6))

$$f'(x_k, d_k^g) \leq \theta f'(x_k, d_k^0) \leq -\theta \langle d_k^0, H_k d_k^0 \rangle.$$

Thus, (4.12) is less stringent than if $f'(x_k, d_k^g)$ were used.

Remark 4.3. Note that while Algorithm 4.1 involves possible auxiliary constraint evaluations at $x_k + d_k^\ell$ and $x_k + d_k^g$ (*Step 1 iv* and *Step 1 vi*), it can be easily modified so as to require at most one auxiliary constraint evaluation per iteration (either at $x_k + d_k^\ell$ or at $x_k + d_k^g$). See §4.4 for implementation.

Remark 4.4. The updating of C_k given in the algorithm is the simplest possible formula. One could seek out more sophisticated approaches to obtain more accurate estimate C_k^e of C_k . For instance, if

$$g_j(x_k + d_k^\ell) > 0 \quad \text{for some } j,$$

then, for $\xi_k \in [0, 1]$,

$$\begin{aligned} g_j(x_k + d_k^\ell) &= g_j(x_k) + \langle \nabla g_j(x_k), d_k^\ell \rangle + \frac{1}{2} \langle d_k^\ell, \nabla^2 g_j(x_k + \xi_k d_k^\ell) d_k^\ell \rangle \\ &\leq g_j(x_k) + \langle \nabla g_j(x_k), d_k^\ell \rangle + \frac{1}{2} \|d_k^\ell\|^2 \|\nabla^2 g_j(x_k + \xi_k d_k^\ell)\|. \end{aligned}$$

Thus, a possible estimate of C_k would be

$$C_k^e \geq \max_{j: g_j(x_k + d_k^\ell) > 0} \frac{g_j(x_k + d_k^\ell) - g_j(x_k) - \langle \nabla g_j(x_k), d_k^\ell \rangle}{\|d_k^\ell\|^2}.$$

This estimate could be properly incorporated into the algorithm to accelerate the increase of C_k whenever required.

4.3 Convergence results

As mentioned in the introduction, Algorithm 4.1 is essentially an integration of algorithms in [6,64] with extension to minimax problems. Therefore, we will not present a complete analysis of its convergence properties. Instead, only major results are provided. Their proofs are given in Appendix A.

4.3.1 Global convergence

The first order necessary conditions of optimality for $QP_2(x_k, H_k)$ can be expressed as follows. If d_k^0 solves $QP_2(x_k, H_k)$, there exist a vector $\mu_k \in \mathbb{R}^p$ and a vector $\lambda_k \in \mathbb{R}^m$ such that

$$\left. \begin{aligned} H_k d_k^0 + \nabla_x L(x_k, \mu_k, \lambda_k) &= 0 \\ \mu_{k,i} &= 0 \quad \forall \quad i \quad \text{s.t.} \quad f_i(x_k) + \langle \nabla f_i(x_k), d_k^0 \rangle < f(x_k, d_k^0) \\ \sum_{i=1}^p \mu_{k,i} &= 1 \quad \text{and} \quad \mu_k \geq 0 \\ \lambda_{k,j} \{g_j(x_k) + \langle \nabla g_j(x_k), d_k^0 \rangle\} &= 0, \quad j = 1, \dots, m \\ \lambda_{k,j} &\geq 0 \quad \text{and} \quad g_j(x_k) + \langle \nabla g_j(x_k), d_k^0 \rangle \leq 0, \quad j = 1, \dots, m. \end{aligned} \right\} \quad (4.13)$$

Under assumptions A3.1, A3.2, A3.3, A4.1 and A4.2, the following two results can be established.

Lemma 4.1. The sequence $\{x_k\}$ is bounded and the sequences $\{t_k d_k^0\}$ and $\{\|x_{k+1} - x_k\|\}$ both converge to zero. $\square$

Theorem 4.2. Let x^* be an accumulation point of the sequence $\{x_k\}$ and let $K \subset \mathbb{N}$ be such that $\{x_k\}$ converges to x^* on K . Then, x^* is a GKKT point of (P) and the subsequence $\{d_k^0\}$ converges to zero on K . $\square$

4.3.2 Superlinear convergence

In order to prove superlinear convergence, we make more regularity assumptions on the functions involved. Assumption A4.1 is replaced by

A4.1'. The functions $g_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 1, \dots, m$, are three times continuously differentiable.

A4.4. The matrix $R(x) = [\frac{\partial \bar{f}_1}{\partial x}(x), \nabla G(x)] \in \mathbb{R}^{n \times (|I(x)|-1+|J(x)|)}$ is full rank, where $\nabla G(x) = [\nabla g_j(x) : j \in J(x)] \in \mathbb{R}^{n \times |J(x)|}$ and $\bar{f}_1 = [f_i(x) - f_1(x) : \forall i \in I(x) \setminus \{1\}]^T \in \mathbb{R}^{n \times (I(x)-1)}$ (cf. comments in front of A3.6 on page 28).

Remark 4.5. As local convergence is concerned, the assumption given in [65] is again misleading (see Remark 4.1). In particular, it is incorrect when $J(x^*) = \emptyset$.

Let x^* be an accumulation point of the sequence generated by the algorithm (a GKKT point of (P) in view of Theorem 4.2). The Lagrange multiplier vectors at x^* are denoted by μ^* and λ^* respectively for f_i 's and g_j 's.

A4.5. At x^* , the second order sufficiency conditions with strict complementary slackness are satisfied (see Chapter 2).

Lemma 4.3. (i) The sequence $\{x_k\}$ converges. (ii) There exists $\bar{C} > 0$ such that $C_k \leq \bar{C}$, $\forall k$. $\square$

Assumption A3.6 now has to be extended to the following one which is used to show that the step of one will be eventually accepted.

A4.6. The sequence $\{H_k\}$ converges in the active subspace to the Hessian of the Lagrangian at x^* in the sense that

$$\lim_{k \rightarrow \infty} \frac{\|P_k \{H_k - \nabla_{xx}^2 L(x^*, \mu^*, \lambda^*)\} P_k d_k^0\|}{\|d_k^0\|} = 0 \quad (4.14)$$

where the matrices P_k are defined by

$$P_k = I - R_k(R_k^T R_k)^{-1} R_k^T$$

with $R_k = R(x_k)$. (Note that, in view of the independence of the gradients of the active constraints at x^* (A4.4), the matrices $R_k^T R_k$ are invertible for k large enough.)

The following set of results can be now established.

Proposition 4.4. For k large enough,

- (i). the multipliers $\{\mu_k^0\}$ and $\{\lambda_k^0\}$ associated with the solution d_k^0 of $QP_2(x_k, H_k)$ converge to μ^* and λ^* respectively, and

$$\mu_{k,i}^0 = 0 \quad \forall i \notin I(x^*), \quad \lambda_{k,j}^0 = 0 \quad \forall j \notin J(x^*);$$

- (ii).

$$f_i(x_k) + \langle \nabla f_i(x_k), d_k^0 \rangle - f(x_k) = f'(x_k, d_k^0) \quad \forall i \in I(x^*)$$

$$g_j(x_k) + \langle \nabla g_j(x_k), d_k^0 \rangle = 0, \quad \forall j \in J(x^*);$$

- (iii).

$$d_k^\ell = d_k^0 + O(\|d_k^0\|^2), \quad d_k^g = d_k^0 + O(\|d_k^0\|^2); \quad (4.15)$$

- (iv).

$$\tilde{d}_k = O(\|d_k^0\|^2)$$

$$t_k = 1.$$

□

Finally, the convergence rate properties of SQP-type methods are preserved and auxiliary constraint evaluations are performed in the early iterations only.

Theorem 4.5. Under the stated assumptions, the convergence is two-step superlinear, *i.e.*,

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+2} - x^*\|}{\|x_k - x^*\|} = 0.$$

□

The key result concerning the solution of (P) in conjunction with nonmonotone line search is given below.

Theorem 4.6. For k large enough, d_k^ℓ is always used and a correction $\tilde{d}_k$ is not computed. □

4.4 Numerical experiments

An efficient implementation of the algorithm described in this chapter together with the algorithm in previous chapter has been completed (FSQP Version 2.4 [92]). The following minor modifications with respect to the algorithm as described in §4.2 were found to be beneficial and were implemented:

(i) *Step 1 v* is performed before *Steps 1 iii* and *Step 1 iv* and, in *Step 1 iii*, ρ^g is used instead of ρ_k^ℓ if the step of one was not accepted at the previous iteration or if $\rho_k^\ell > 0.5$ (this reduces the number of auxiliary constraint evaluations, in the spirit of the suggestion made in Remark 4.3). Also, if $m = 0$ (in particular, in the case of unconstrained minimax problems), *Steps 1 ii, iii* and *v* are skipped and d_k^ℓ and d_k^g are set to d_k^0 .

(ii) In the computation of $\tilde{d}$, $\|d_k^g\|^\gamma$ is replaced by

$$\min\{0.01\|d_k^g\|, \|d_k^g\|^\gamma\}$$

to prevent $\tilde{d}$ from being too large in the early iterations.

(iii) In *Step 2*, if $\|d_k^0\| > \underline{d}$, C_{k+1} is set to $\max\{0.5C_k, \underline{C}\}$ to prevent too rapid a decrease of C_k . If $\|d_k^0\| \geq \underline{d}$ and $g_j(x_k + d_k^\ell) > 0$ for some $j \in \{1, \dots, m\}$, C_{k+1} is set to $10C_k$ instead of $2C_k$.

The FSQP code includes special provisions for efficient handling of affine constraints and it also accepts affine *equality* constraints.

Results on two sets of experiments are presented in Tables 4.1 and 4.2. Notations are consistent with those in Table 3.1. In addition, **NNL** stands for the number of nonlinear constraints, **NCGI** for the number of individual nonlinear constraint evaluations and **OBJECTIVE** stands for the objective function value at the final iterate. Algorithm 4.1 (NL) is compared to the algorithm analyzed in [64] (AL) (with suitable modification to take care of the *max* function) which is the best available “feasible iterate” algorithm to our knowledge.² All computations were performed on a SUN 4/SPARC station 1. For problems in Table 4.2, gradients were computed analytically; for problems in Table 4.1, they were computed by finite differences (for the i th component, the perturbation parameter was $10^{-7} \max\{1, |x_{k,i}|\}$).

Unable to find nonlinearly constrained minimax test problems in the literature, we constructed problems P43M through P117M from problems 43, 84, 113 and 117 in [39] by removing certain constraints and including instead additional objectives of the form $f_i(x) = f(x) + \alpha_i g_j(x)$ where the α_i ’s are positive scalars and $g_j(x) \leq 0$. Specifically, P43M is constructed from problem 43 by taking out the first two constraints and including two corresponding objectives with $\alpha_i = 15$ for both; P84M similarly corresponds to problem 84 without constraints 5 and 6 but with two corresponding additional objectives, with $\alpha_i = 20$ for both; for P113M the first three linear constraints from problem 113 were turned into objectives, with $\alpha_i = 10$ for all three;

²FSQP Version 2.4 [92] gives the user the option to select either AL or NL.

for P117M, the first two nonlinear constraints were turned into objectives, again with $\alpha_i = 10$ for both. The performance of FSQP on these problems are summarized in Table 4.1. The algorithm was terminated if $\|d_k^0\| \leq 0.5 \times 10^{-5}$.

PROB	CD	NF	NNL	NCMF	NCFI	NCGI	IT	OBJMAX	KKT
P43M	AL	3	1	27	80	43	15	-.440000000E+02	.30E-06
	NL			20	63	25	16	-.440000000E+02	.39E-05
P84M	AL	3	4	7	17	20	4	-.528033513E+07	.0
	NL			3	9	12	3	-.528033513E+07	.37E-03
P113M	AL	4	5	25	108	127	14	.243062091E+02	.31E-05
	NL			21	84	105	14	.243062091E+02	.31E-05
P117M	AL	3	3	48	124	144	21	.323486790E+02	.46E-05
	NL			19	57	54	17	.323486790E+02	.26E-04

Table 4.1: Numerical Results for Constrained Minimax Problems

Table 4.2 contains results of (non-minimax) test problems from [39]. The algorithm was terminated when the Euclidean norm of the gradient of the Lagrangian is smaller than the corresponding value under the column of EPS.

In Table 4.1, algorithm NL performs much better than AL. In Table 4.2, except for P67 and P85, the number of nonlinear constraint evaluations is lower in NL, often dramatically so. For P67 and P85, t_k is one for all k in NL, but more iterations are required than in AL.

The suggestion in Remark 4.4 was also tested by simply replacing $C_{k+1} = 10C_k$ with $C_{k+1} = \max\{10C_k, 0.05C_k^e\}$ in (iii) on page 48. The results were not visibly different. (See also §5.3 for another case.)

PROB	CD	NNL	NCFI	NCGI	IT	OBJECTIVE	KKT	EPS
P12	AL	1	7	14	7	-.3000000000E+02	.72E-06	.10E-05
	NL		7	12	7	-.3000000000E+02	.79E-06	.10E-05
P29	AL	1	11	20	10	-.226274170E+02	.41E-05	.10E-04
	NL		12	16	12	-.226274170E+02	.63E-05	.10E-04
P30	AL	1	13	25	13	.1000000000E+01	.26E-07	.10E-06
	NL		14	14	14	.1000000000E+01	.43E-07	.10E-06
P31	AL	1	10	21	8	.6000000000E+01	.34E-06	.10E-04
	NL		10	18	10	.6000000000E+01	.50E-06	.10E-04
P32	AL	1	3	5	3	.1000000000E+01	.15E-14	.10E-07
	NL		3	4	3	.1000000000E+01	.64E-16	.10E-07
P33	AL	2	4	11	4	-.4000000000E+01	.13E-11	.10E-07
	NL		5	10	5	-.4000000000E+01	.47E-11	.10E-07
P34	AL	2	7	28	7	-.834032443E+00	.19E-08	.10E-07
	NL		9	24	9	-.834032445E+00	.38E-11	.10E-07
P43	AL	3	11	51	9	-.4400000000E+02	.12E-05	.10E-04
	NL		12	49	12	-.4400000000E+02	.16E-06	.10E-04
P44	AL	0	6		6	-.1500000000E+02	.0	.10E-07
	NL		6		6	-.1500000000E+02	.0	.10E-07
P51	AL	0	8		6	.505655658E-15	.46E-06	.10E-05
	NL		9		8	.505655658E-15	.34E-08	.10E-05
P57	AL	1	5	7	3	.306463061E-01	.29E-05	.10E-04
	NL		5	7	3	.306463061E-01	.28E-05	.10E-04
P66	AL	2	8	30	8	.518163274E+00	.50E-09	.10E-07
	NL		9	24	9	.518163274E+00	.14E-08	.10E-07
P67	AL	14	21	305	21	-.116211927E+02	.52E-05	.10E-04
	NL		61	854	61	-.116211927E+02	.12E-05	.10E-04
P70	AL	1	31	35	29	.940197325E-02	.13E-07	.10E-06
	NL		31	31	31	.940197325E-02	.19E-07	.10E-06
P76	AL	0	6		6	-.468181818E+01	.34E-04	.10E-03
	NL		6		6	-.468181818E+01	.34E-04	.10E-03
P84	AL	6	4	30	4	-.528033513E+07	.85E-09	.10E-07
	NL		4	29	4	-.528033513E+07	.38E-09	.10E-07
P85	AL	38	37	1458	34	-.239953357E+01	.43E-03	.10E-02
	NL		80	3040	80	-.237335088E+01	.81E-03	.10E-02
P86	AL	0	8		6	-.323486790E+02	.22E-08	.10E-05
	NL		7		6	-.323486790E+02	.53E-06	.10E-05
P93	AL	2	15	58	12	.135075968E+03	.37E-03	.10E-02
	NL		15	36	15	.135075964E+03	.24E-04	.10E-02
P100	AL	4	23	114	16	.680630057E+03	.62E-06	.10E-03
	NL		20	102	17	.680630057E+03	.49E-04	.10E-03
P110	AL	0	9		8	-.457784697E+02	.50E-06	.10E-05
	NL		9		8	-.457784697E+02	.50E-06	.10E-05
P113	AL	5	12	108	12	.243063768E+02	.81E-03	.10E-02
	NL		12	100	12	.243064357E+02	.83E-03	.10E-02
P117	AL	5	20	219	19	.323486790E+02	.58E-04	.10E-03
	NL		18	93	17	.323486790E+02	.34E-04	.10E-03
P118	AL	0	19		19	.664820450E+03	.13E-14	.10E-07
	NL		19		19	.664820450E+03	.17E-14	.10E-07

Table 4.2: Numerical Results for Constrained Non-Minimax Problems

Chapter 5

Nonmonotone Line Search Revisited

Nonmonotone line search has been successfully used in constrained optimization to improve the efficiency of algorithms. Global convergence is maintained and two-step superlinear convergence is preserved (see [6,63] and previous chapters). In this chapter, it is re-examined in the direction to investigate the possibility of avoiding computation of $\tilde{d}$ altogether.

5.1 Motivations

The basic idea behind the nonmonotone line search in conjunction with two-step superlinear convergence is that, if $\{x_k\}$ converges two-step superlinearly, then, close to a strong local minimizer x^* , $x_k + d_k^\ell$ is feasible and satisfies

$$f(x_k + d_k^\ell) \leq \max_{\ell=0,1,2} f(x_{k-\ell}) - \alpha \langle d_k^0, H_k d_k^0 \rangle. \quad (5.1)$$

The following arc search

$$f(x_k + t d_k^g + t^2 \tilde{d}_k) \leq \max_{\ell=0,1,2} f(x_{k-\ell}) - \alpha t \theta \langle d_k^0, H_k d_k^0 \rangle$$

is performed to initialize the superlinear convergence. Thus, for x_k close to a strong local minimizer, it holds that

$$f(x_k + d_k^g + \tilde{d}_k) \leq \max_{\ell=0,1,2} f(x_{k-\ell}) - \alpha \theta \langle d_k^0, H_k d_k^0 \rangle. \quad (5.2)$$

Then, the superlinear convergence is preserved if eventually (5.2) is replaced by (5.1) (cf. (4.12)). With $x_{k+1} = x_k + d_k^\ell$ and (5.1) in effect, the nonmonotone decrease

criterion avoids the computation of $\tilde{d}_k$ and maintains the superlinear convergence of $\{x_k\}$. Specifically, it is shown that, under mild assumptions, the condition (*cf.* A3.6 and A4.6)

$$\lim_{k \rightarrow \infty} \frac{\|P_k\{H_k - \nabla_{xx}^2 L(x^*, \lambda^*)\}P_k d_k^0\|}{\|d_k^0\|} = 0 \quad (5.3)$$

implies

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+2} - x^*\|}{\|x_k - x^*\|} = 0, \quad (5.4)$$

and, close to a strong minimizer x^* , $x_{k+1} = x_k + d_k^\ell$ and $\tilde{d}_k$ is not computed. The computation of $\tilde{d}_k$ in early iterations involves function evaluation at auxiliary points and the solution of a quadratic program. This is true even if feasible iterates are not required [63]. While the solution of an additional quadratic program may be within reasonable amount of time, the function evaluation at auxiliary points could be prohibitively expensive in the context of engineering applications.

The question now is: could the superlinear convergence be achieved without the need of the second order correction $\tilde{d}_k$ from the very beginning. Convergence properties established in previous chapters can hardly be strengthened under the same assumptions. Intuitively, the answer could be affirmative only under stronger assumptions.

Optimization theory provides us another important result on the rate of local convergence. Under mild assumptions, the stronger condition (note the difference with (5.3))

$$\lim_{k \rightarrow \infty} \frac{\|P_k\{H_k - \nabla_{xx}^2 L(x^*, \lambda^*)\}d_k^0\|}{\|d_k^0\|} = 0 \quad (5.5)$$

is equivalent to (see, *e.g.*, [77])

$$\lim_{k \rightarrow \infty} \frac{\|x_k + d_k^0 - x^*\|}{\|x_k - x^*\|} = 0. \quad (5.6)$$

This is also established in [4] under stronger conditions on both H_k and x_k . Therefore, one-step superlinear convergence of $\{x_k\}$ can be achieved if either (5.5) or (5.6) can be satisfied, provided the full step of one is accepted close to a solution. It is impossible in general, even under these assumptions, to accept a full step in constrained optimization along the SQP direction d_k^0 or some variants (*e.g.*, d_k , d_k^q and d_k^ℓ used in previous chapter) without violating feasibility or increasing the objective. This is again the so-called Maratos effect. However, as will be shown below (Theorem 5.8), step of one can be taken close to a strong minimizer under the same condition if the

monotone decrease criterion of the objective function is replaced by a nonmonotone decrease criterion similar to the one described in Chapter 4, with no need of the second order correction. Thus, the computation of $\tilde{d}_k$ can be discarded from the very beginning of optimization process, resulting in complete avoidance of an additional quadratic program and function evaluations at auxiliary point.

In practice, it is observed that condition (5.3) often holds. And counter-examples have been constructed to illustrate that condition (5.5) need not hold (see, *e.g.*, [11,88]). However, if the Hessian matrix of the Lagrange function is positive definite (not just on the subspace of active constraints) at a local solution, popular updating formulas, such as BFGS and DFT, satisfy condition (5.5) (see [4]). It has been shown (see, *e.g.*, [10]) that, if the Lagrange function is augmented by a certain term with sufficiently positive quadrature, the positive definiteness of the augmented Lagrange function can be achieved. Active research has been undertaken to construct suitable augmentations, which is beyond the purpose of this dissertation. Here we report some results by making use of (5.5) in conjunction with nonmonotone line search.

5.2 One-step superlinear convergence

We now present a complete analysis for the algorithm generating feasible iterates. Extension to algorithms of penalty type is briefly discussed; the analysis is analogous and is thus left out.

5.2.1 Feasible iterates algorithm

In this case, both global and local directions are required as argued in the introduction of Chapter 4. Thus, we assume a sequence $\{x_k\}$ is generated by setting

$$x_{k+1} = x_k + t_k d_k$$

where $d_k = d_k^g$ or $d_k = d_k^\ell$, and t_k is such that, for some $M > 0$,

$$f(x_k + t_k d_k^g) < f(x_{k-M})$$

$$g_j(x_k + t_k d_k^g) \leq 0, \quad j = 1, \dots, m.$$

We are interested in the question that whether it is possible, by choosing proper M , to achieve

$$f(x_k + d_k^\ell) < f(x_{k-M})$$

$$g_j(x_k + d_k^\ell) \leq 0, \quad j = 1, \dots, m$$

for k large enough.

To avoid redundant presentation, Algorithm 5.1 below is stated in terms of Algorithm 4.1. For the time being, M can be any integer larger than 0, because we know that it may not be possible to accept the step of one with $M = 0$.

Algorithm 5.1.

Perform Algorithm 4.1, with the following exceptions:

- (i). $\tilde{d}_k \equiv 0 \forall k$;
- (ii). (4.9) is replaced by

$$f(x_k + d_k^\ell) \leq \max_{\ell=0,1,\dots,M} \{f(x_{k-\ell})\} - \alpha \langle d_{k-M}^0, H_{k-M} d_{k-M}^0 \rangle. \quad (5.7)$$

□

Remark 5.1. Note that $\langle d_{k-M}^0, H_{k-M} d_{k-M}^0 \rangle$ is used in (5.7) instead of $\langle d_k^0, H_k d_k^0 \rangle$ as in (4.9). It turns out that, if the latter were used, our main results (Theorem 5.8 below) need not hold. $\langle d_{k-M}^0, H_{k-M} d_{k-M}^0 \rangle$ can however be replaced by

$$\min_{\ell=0,1,\dots,M} \langle d_{k-\ell}^0, H_{k-\ell} d_{k-\ell}^0 \rangle.$$

In the rest of this chapter, we assume $p = 1$. Thus, $f(x) = f_1(x)$ is smooth and $f'(x, d) = \langle \nabla f(x), d \rangle$. There is no conceptual difficulty to extend our analysis to the case $p > 1$. The corresponding Lagrange function is given by

$$L(x, \lambda) = f(x) + \sum_{j=1}^m \lambda_j g_j(x).$$

We assume that all relevant assumptions made in previous chapters hold. It can be shown that global convergence of Algorithm 4.1 will not be destroyed for any fixed M (actually, M can be modified finitely many times). Therefore, all results from Proposition 4.1 up to Proposition 4.4(iii) can be similarly established, since they are independent of $\tilde{d}_k$.

Let x^* be any accumulation point of the sequence $\{x_k\}$ and λ^* be the associated Lagrange multiplier vector.

we now explicitly assume condition (5.5) holds, *i.e.*,

A5.1. The sequence $\{H_k\}$ converges to the Hessian of the Lagrangian at x^* in the sense that

$$\lim_{k \rightarrow \infty} \frac{\|P_k\{H_k - \nabla_{xx}^2 L(x^*, \lambda^*)\}d_k^0\|}{\|d_k^0\|} = 0$$

where the matrices P_k are defined by

$$P_k = I - R_k(R_k^T R_k)^{-1} R_k^T$$

with $R_k = [\nabla g_j(x_k) : j \in J(x^*)] \in \mathbb{R}^{n \times |J(x^*)|}$ (remember that $p = 1$).

In the literature, (5.6) is sometimes assumed (see, *e.g.*, [5,12]) instead of (5.5), thus avoiding the issue of how to update H_k . We understand now that they are equivalent [4,77]. Assuming (5.5) is more implementation-oriented because, in practice, suitable methods have to be sought out to construct $\{H_k\}$.

The following lemma shows that the line search is always well defined and thus so is the algorithm.

Lemma 5.1. There exists $\underline{t} \in (0, 1)$ such that

$$\inf_k t_k \geq \underline{t}.$$

Proof. Without loss of generality, we assume $t_k < 1$. In view of A3.1' and A4.1' and Proposition 4.4(iii), for $t \in [0, 1]$, there exist $c_1 > 0$ and $c_2 > 0$ such that

$$f(x_k + td_k^g) \leq f(x_k) + t\langle \nabla f(x_k), d_k^g \rangle + c_1 t^2 \|d_k^0\|^2. \quad (5.8)$$

and

$$g_j(x_k + td_k^g) \leq g_j(x_k) + t\langle \nabla g_j(x_k), d_k^g \rangle + c_2 t^2 \|d_k^0\|^2, \quad j = 1, \dots, m. \quad (5.9)$$

Thus, in view of (4.13), (5.8), (4.11) and A3.3, it holds that

$$\begin{aligned} f(x_k + td_k^g) &\leq f(x_k) + t\langle \nabla f(x_k), d_k^g \rangle + c_1 t^2 \|d_k^0\|^2 \\ &= f(x_k) + \alpha t\langle \nabla f(x_k), d_k^g \rangle + t(1 - \alpha)\langle \nabla f(x_k), d_k^g \rangle + c_1 t^2 \|d_k^0\|^2 \\ &\leq f(x_k) + \alpha t\langle \nabla f(x_k), d_k^g \rangle + t(1 - \alpha)\theta\langle \nabla f(x_k), d_k^0 \rangle + c_1 t^2 \|d_k^0\|^2 \\ &\leq f(x_k) + \alpha t\langle \nabla f(x_k), d_k^g \rangle - t(1 - \alpha)\theta\langle d_k^0, H_k d_k^0 \rangle + c_1 t^2 \|d_k^0\|^2 \\ &\leq f(x_k) + \alpha t\langle \nabla f(x_k), d_k^g \rangle - t(1 - \alpha)\theta\sigma_1 \|d_k^0\|^2 + c_1 t^2 \|d_k^0\|^2 \\ &= f(x_k) + \alpha t\langle \nabla f(x_k), d_k^g \rangle + t\|d_k^0\|^2 \underbrace{\{(\alpha - 1)\theta\sigma_1 + c_1 t\}}_{< 0 \text{ for } t \text{ small}}. \end{aligned}$$

Letting $\hat{t} = \frac{(1-\alpha)\theta\sigma_1}{c_1}$, it follows that, for all $t \in [0, \hat{t}]$,

$$f(x_k + td_k^g) \leq f(x_k) + \alpha t \langle \nabla f(x_k), d_k^g \rangle. \quad (5.10)$$

As constraints are concerned, in view of Proposition 4.4(i)—(iii), it suffices to consider all active constraints. For any j such that $\lambda_{k,j} > 0$,

$$\begin{aligned} g_j(x_k + td_k^g) &\leq g_j(x_k) + t \langle \nabla g_j(x_k), d_k^g \rangle + c_2 t^2 \|d_k^0\|^2 \\ &= g_j(x_k) + t \langle \nabla g_j(x_k), (1 - \rho_k^g) d_k^0 + \rho_k^g d_k^1 \rangle + c_2 t^2 \|d_k^0\|^2 \\ &= (1 - t) g_j(x_k) + t(1 - \rho_k^g) \{g_j(x_k) + \langle \nabla g_j(x_k), d_k^0 \rangle\} \\ &\quad + t \rho_k^g \{g_j(x_k) + \langle \nabla g_j(x_k), d_k^1 \rangle\} + c_2 t^2 \|d_k^0\|^2. \end{aligned} \quad (5.11)$$

In view of (4.13),

$$g_j(x_k) + \langle \nabla g_j(x_k), d_k^0 \rangle = 0, \quad \forall j : \lambda_{k,j} > 0$$

and, in view of the property of d^1 at x^* (cf. (4.8))

$$g_j(x^*) + \langle \nabla g_j(x^*), d^1 \rangle < 0, \quad \forall j = 1, \dots, m,$$

there exists $\epsilon^* > 0$ such that, for all k ,

$$g_j(x_k) + \langle \nabla g_j(x_k), d_k^1 \rangle \leq -\epsilon^*, \quad \forall j = 1, \dots, m.$$

In addition, it can be derived from Proposition 4.4(i) and (iii) and $\|d^{1*}\| > 0$ that there exists $c_3 > 0$ such that

$$\rho_k^g \leq c_3 \|d_k^0\|^2.$$

Thus, (5.11) becomes

$$\begin{aligned} g_j(x_k + td_k^g) &\leq (1 - t) g_j(x_k) - t \epsilon^* \|d_k^0\|^2 + c_2 t^2 \|d_k^0\|^2 \\ &= (1 - t) g_j(x_k) - t \|d_k^0\|^2 (\epsilon^* - c_2 t), \quad \forall j : \lambda_{k,j} > 0. \end{aligned}$$

Hence, taking $\tilde{t} = \frac{\epsilon^*}{c_2}$, because of the feasibility of x_k , for all $t \in [0, \tilde{t}]$,

$$g_j(x_k + td_k^g) \leq 0 \quad \forall j = 1, \dots, m.$$

By choosing $\bar{t} = \beta \min\{\hat{t}, \tilde{t}\}$, it follows that $t_k \geq \bar{t}$ and

$$f(x_k + t_k d_k^g) \leq f(x_k) + \alpha t_k \langle \nabla f(x_k), d_k^g \rangle$$

$$g_j(x_k + t_k d_k^g) \leq 0 \quad \forall j = 1, \dots, m.$$

The lemma is proven. $\square$

Lemma 5.2. Let d_k be either d_k^0 or d_k^g or d_k^ℓ . Then, it holds that

$$\lim_{k \rightarrow \infty} \frac{\|x_k + d_k - x^*\|}{\|x_k - x^*\|} = 0. \quad (5.12)$$

Moreover, for k large enough, (5.12) implies

$$\|d_k\| \leq \frac{1 - \frac{\underline{t}}{2}}{1 - \underline{t}} \|x_k - x^*\|, \quad (5.13)$$

where $\underline{t}$ is as in Lemma 5.1.

Proof. For $d_k = d_k^0$, (5.12) follows directly from assumption A5.1. Thus, it follows that

$$\|d_k^0\| = O(\|x_k - x^*\|). \quad (5.14)$$

Now, for $d_k = d_k^\ell$ or $d_k = d_k^g$, in view of Proposition 4.4(iii),

$$\begin{aligned} \|x_k + d_k - x^*\| &= \|x_k + d_k^0 + O(\|d_k^0\|^2) - x^*\| \\ &\leq \|x_k + d_k^0 - x^*\| + O(\|d_k^0\|^2) \end{aligned}$$

and the claim follows in view of (5.12) with $d_k = d_k^0$ and of (5.14).

Next, from the triangle inequality

$$\|x_k + d_k - x^*\| \geq \left| \|x_k - x^*\| - \|d_k\| \right|,$$

it follows that, in view of (5.12),

$$\lim_{k \rightarrow \infty} \frac{\|d_k\|}{\|x_k - x^*\|} = 1.$$

Thus, given $\epsilon > 0$, there exists N such that, for all $k \geq N$,

$$\frac{\|d_k\|}{\|x_k - x^*\|} \leq 1 + \epsilon.$$

(5.13) follows if we take $\epsilon = \frac{\frac{\underline{t}}{2}}{1 - \underline{t}}$. $\square$

The following result is the key to the establishment of our main theorem. It would not hold in general if merely (5.3) were assumed.

Proposition 5.3. The sequence $\{x_k\}$ converges linearly, *i.e.*, there exists $c_4 \in [0, 1)$ such that, for k large enough,

$$\|x_{k+1} - x^*\| \leq c_4 \|x_k - x^*\|.$$

Proof. Assume k is large enough. Since $x_{k+1} = x_k + t_k d_k$ with $d_k = d_k^\ell$ or $d_k = d_k^g$, it follows that, in view of (5.13) and Lemma 5.1,

$$\begin{aligned} \|x_{k+1} - x^*\| &= \|x_k + d_k + (t_k - 1)d_k - x^*\| \\ &\leq \|x_k + d_k - x^*\| + (1 - t_k)\|d_k\| \\ &\leq o(\|x_k - x^*\|) + (1 - \underline{t})\|d_k\| \\ &\leq o(\|x_k - x^*\|) + (1 - \frac{t}{2})\|x_k - x^*\| \\ &\leq (1 - \frac{t}{4})\|x_k - x^*\|. \end{aligned}$$

Taking $c_4 = 1 - \frac{t}{4}$ proves the lemma. $\square$

We note that the linear convergence of $\{x_k\}$ is actually assumed in [4] to establish the equivalence of (5.5) and (5.6). It turns out this is unnecessary as shown in [77].

The following lemma is the same as Lemma A.3 in Appendix A.

Lemma 5.4. There exists $c_5 > 0$ such that, for all x close to x^* ,

$$f(x) \geq f(x^*) + c_5\|x - x^*\|^2.$$

$\square$

A suitable modification of Lemma 3.8 in [12] gives the following result, which is proven in Appendix A.

Lemma 5.5. There exists $c_6 \in (0, \frac{1}{2})$ such that, for k large enough,

$$f(x_k) - f(x^*) + \alpha \langle \nabla f(x_k), d_k^0 \rangle \geq c_6 \{f(x_k) - f(x^*)\} > 0. \quad (5.15)$$

$\square$

Lemma 5.6. There exists $c_7 > 0$ such that

$$f(x_k + d_k^\ell) \leq f(x^*) + c_7\|x_k - x^*\|^2.$$

Proof. In view of Proposition 4.4(i)—(iii) and the optimality conditions (2.5), we obtain

$$f(x_k + d_k^\ell) = f(x^*) - \sum_{j \in J(x^*)} \langle \nabla g_j(x^*), x_k + d_k^\ell - x^* \rangle + O(\|x_k + d_k^\ell - x^*\|^2)$$

and

$$\begin{aligned} g_j(x_k + d_k^\ell) &= O(\|d_k^0\|^2) \\ &= g_j(x^*) + \langle \nabla g_j(x^*), x_k + d_k^\ell - x^* \rangle + O(\|x_k + d_k^\ell - x^*\|^2). \end{aligned}$$

The lemma then follows in view of Lemma 5.2 and $g_j(x^*) = 0 \forall j \in J(x^*)$. $\square$

The following lemma shows that, indeed, the nonmonotone decrease criterion of the objective function can be achieved with a unit step.

Lemma 5.7. There exists $M > 0$ such that, for k large enough,

$$f(x_k + d_k^\ell) \leq f(x_{k-M}) - \alpha \langle d_{k-M}^0, H_{k-M} d_{k-M}^0 \rangle.$$

Proof. From Lemmas 5.3—5.6, we have

$$\begin{aligned} f(x_k + d_k^\ell) &\leq f(x^*) + c_7 \|x_k - x^*\|^2 \\ &\leq f(x^*) + c_7 c_4 \|x_{k-1} - x^*\|^2 \\ &\leq f(x^*) + c_7 c_4^M \|x_{k-M} - x^*\|^2 \\ &\leq f(x^*) + \frac{c_7 c_4^M}{c_5} \{f(x_{k-M}) - f(x^*)\} \\ &\leq f(x^*) + \frac{c_7 c_4^M}{c_5 c_6} \{f(x_{k-M}) - f(x^*) + \alpha \langle \nabla f(x_{k-M}), d_{k-M}^0 \rangle\}. \end{aligned}$$

Since $c_4 < 1$, there exists M large enough such that

$$\frac{c_7 c_4^M}{c_5 c_6} \leq 1.$$

The claim follows. $\square$

Theorem 5.8. There exists $M > 0$ such that the sequence $\{x_k\}$ generated by Algorithm 5.1 converges with one-step superlinear rate of convergence.

Proof. A contradiction based on Lemma 4.3(ii) proves, for k large enough,

$$g_j(x_k + d_k^\ell) \leq 0, \quad j = 1, \dots, m.$$

In view of Lemma 5.1 and Lemma 5.7, the theorem follows. $\square$

This theorem says that, given the problem, there exists $M > 0$ such that, if assumption A5.1 is satisfied, one-step superlinear convergence can be achieved without the initialization of a second order correction. Unfortunately, M is usually unknown for a given problem. It would be quite interesting to investigate whether it is possible to devise some schemes that adaptively obtain an estimate of the right M . For instance, in view of Proposition 5.3 and Lemma 5.7, we could try to detect whether the linear convergence of $\{x_k\}$ occurs and, if it does, M could be increased slowly. As long as M is modified finitely many times, global convergence is preserved. Our numerical experiments indicate that $M = 1$ is often suitable.

5.2.2 Penalty-type algorithm

It is straightforward to show that results presented in §5.2.1 apply to the case that the objective function is a max-type function, as was done in Chapter 4 for two-step superlinear convergence. In the following, we explicitly state extensions to algorithms based on exact penalty function.

In the context of exact penalty function approaches where feasible iterates is not of primary concern, an exact penalty function $p(x, r)$ for (P) (again, assuming $p = 1$) can be defined by

$$p(x, r) = f(x) + r \sum_{j=1}^m \max\{0, g_j(x)\} \quad (5.16)$$

with $r > 0$ the penalty parameter. Nonlinear equality constraints

$$h_\ell(x) = 0, \quad \ell = 1, \dots, s$$

can now be easily included in the form

$$p(x, r) = f(x) + r \sum_{j=1}^m \max\{0, g_j(x)\} + r \sum_{\ell=1}^s |h_\ell(x)|.$$

To simplify expositions, we omit the equality constraints. The first order approximation to the directional derivative of $p(x, r)$ along a certain direction d is defined by

$$\begin{aligned} p'(x, d) &= \langle \nabla f(x), d \rangle + r \sum_{j=1}^m \max\{0, g_j(x) + \langle \nabla g_j(x), d \rangle\} \\ &\quad - r \sum_{j=1}^m \max\{0, g_j(x)\}. \end{aligned}$$

Since $p(x, r)$ can be readily rewritten in a form of *max*-type function, the framework developed in Chapter 3 applies with proper modifications. In particular, an algorithm based on the SQP scheme and a nonmonotone line search can be simply stated as follows, which is a simplification of the one given in [63]. The quadratic program $QP_1(x_k, H_k)$ for computing d_k^0 remains the same except to replace f' by p' .

Algorithm 5.2.

Parameters. $\alpha \in (0, \frac{1}{2})$, $\beta \in (0, 1)$, $M > 0$.

Data. $x_0 \in \mathbb{R}^n$, $H_0 \in \mathbb{R}^{n \times n}$, and $H_0 = H_0^T$.

Step 0. Initialization. Set $k = 0$. Set $x_{-M} = x_{-M+1} = \cdots = x_0$.

Step 1. Computation of a new iterate.

- i. Compute d_k^0 by solving $QP_1(x_k, H_k)$. If $d_k^0 = 0$, stop.
- ii. Compute t_k , the first number t in the sequence $\{1, \beta, \beta^2, \dots\}$ satisfying

$$p(x_k + td_k^0, r) \leq \max_{\ell=0,1,\dots,M} \{p(x_{k-\ell}, r)\} - \alpha t \langle d_{k-M}^0, H_{k-M} d_{k-M}^0 \rangle. \quad (5.17)$$

Set $x_{k+1} = x_k + t_k d_k^0$.

Step 2. Updates. Compute a new symmetric positive definite approximation H_{k+1} to the Hessian of the Lagrangian. Increase k by 1 and go back to *Step 1*.

□

It has been shown that, if $r > r^*$ for some r^* , a minimizer x^* of $p(\cdot, r)$ is also a minimizer of (P) and algorithms based on this idea have been successfully designed (see, e.g., [33,75]). In the above algorithm, we assumed such an r is suitably chosen. (See, e.g., [52,75] for iterative approaches for obtaining such r .)

Based on the analysis in §5.2, a step of one ($t_k = 1$) will be eventually accepted in *Step 1 ii*, i.e.,

Theorem 5.9. There exists $M > 0$ such that, for k large enough, the sequence $\{x_k\}$ generated by Algorithm 5.2 is obtained by setting

$$x_{k+1} = x_k + d_k^0$$

and it converges one-step superlinearly. □

This result might be very significant because it has been known that the SQP direction d_k^0 alone cannot achieve superlinear convergence in general. In practice, we could simply turn off any provisions, such as watchdog or second order correction, that are used to avoid the Maratos effect in existing codes such as VF03AD [75] and FSQP [92]. Furthermore, smooth exact penalty functions can be avoided, which are usually very complicated and expensive to evaluate in the line search (see, e.g., [3, 20,78]). The nonmonotone decrease of the penalty function does not hurt in any sense, since it does not relate to the increase or decrease of the objective function

even in their original setup. Our results strongly encourage more active research for approaches that ensure A5.1.

In a similar manner to what was done in Chapter 3, results here apply to (affine constrained) minimax problems. The resulting algorithm should take the exact form of Algorithm 5.2, with exceptions that the penalty function is replaced by the *max* function.

5.3 Numerical experiments

Test on both algorithms is performed. The performance of Algorithm 5.1 is very sensitive to the estimate of C_k when nonlinear constraints are present. The updating rule suggested in Remark 4.4 together with (iii) on page 48 seems to improve the numerical performance of the algorithm on the few nonlinearly constrained minimax problems tested in Chapter 4 (*cf.* Table 4.1). However, we failed to obtain uniformly better results for non-minimax nonlinear constrained problems. More sophisticated schemes for updating C_k are still under investigation.

Results on minimax problems used in Chapter 3 and Chapter 4 are summarized in Table 5.1 for both $M = 1$ and $M = 5$. The algorithm is terminated, using the same stopping criteria as in previous chapters. We observe that $M = 5$ always yields better performance than $M = 1$, though step of one are still accepted close to solution for all problems, except for problem HETT which rejected step of one when $M = 1$. Compared with results in previous chapters, significant improvement is achieved for the first three nonlinearly constrained minimax problems, while similar performances are obtained for all other problems.

To implement Algorithm 5.2, VF03AD [1] was modified to take advantage of the updating scheme for the penalty parameter. The watchdog technique in VF03AD is turned off, leaving everything else unchanged. A practical problem associated with $QP_1(x_k, H_k)$ in this context is that $QP_1(x_k, H_k)$ need not be feasible for all k . If the infeasibility happens, the claim that (*cf.* (3.6))

$$p'(x, d) \leq -\langle d, Hd \rangle$$

may no longer hold. To avoid this difficulty, we replace $-\langle d_{k-M}^\ell, H_{k-M} d_{k-M}^\ell \rangle$ by

$$\max\{-\langle d_{k-M}^\ell, H_{k-M} d_{k-M}^\ell \rangle, p'(x_{k-M}, d_{k-M})\}$$

in Algorithm 5.2. We only tested Algorithm 5.2 with $M = 1$. Step of one is accepted close to the solution for all problems. The results are summarized in Table 5.2. ACC

indicates the termination requirement by VF03AD. Both VF03AD and Algorithm 5.2 are terminated by the same value. The performances for both algorithms are very comparable, though the former has the advantage to avoid possible backtracking associated with the watchdog technique. And Algorithm 5.2 does perform much better for P67 and P100 than VF03AD.

Note that, in all these tests, the matrices H_k were obtained by BFGS update with Powell's modification, which does not guarantee assumption A5.1. We believe the algorithmic behavior of these new algorithms are bound to be improved by updating schemes that ensures A5.1.

PROB	NNL	M	NF	NCFI	NCGI	IT	OBJMAX	KKT
BARD	0	5	15	105		7	.508168686E-01	.70E-02
		1		105		7	.508168686E-01	.70E-02
CB2	0	5	3	18		6	.195222451E+01	.80E-03
		1		18		6	.195222451E+01	.80E-03
CB3	0	5	3	15		5	.200000000E+01	.19E-03
		1		15		5	.200000000E+01	.19E-03
COLV	0	5	6	102		17	.323486790E+02	.28E-03
		1		102		17	.323486790E+02	.28E-03
DAVD2	0	5	20	220		11	.115706440E+03	.51E-04
		1		223		11	.115706440E+03	.16E-04
F-R	0	5	2	20		10	.494895210E+01	.57E-05
		1		20		10	.494895210E+01	.57E-05
HETT	0	5	5	80		14	.245935691E-02	.78E-05
		1		3814		412	.245936158E-02	.37E-04
R-S	0	5	4	44		10	-.440000000E+02	.21E-02
		1		44		10	-.440000000E+02	.21E-02
WATS-6	0	5	31	403		13	.127170944E-01	.38E-04
		1		403		13	.127170944E-01	.38E-04
WATS-20	0	5	31	1023		33	.897632047E-07	.13E-05
		1		1023		32	.898278737E-07	.13E-05
WONG	0	5	5	126		21	.680630057E+03	.93E-03
		1		212		30	.680630057E+03	.54E-03
MAD1	0	5	3	18		6	-.389659516E+00	.48E-10
		1		18		6	-.389659516E+00	.48E-10
MAD2	0	5	3	18		5	-.330357143E+00	.37E-07
		1		43		5	-.330357145E+00	.37E-07
MAD4	0	5	3	24		8	-.448910525E+00	.38E-16
		1		24		8	-.448910525E+00	.38E-16
MAD5	0	5	3	24		8	-.100000000E+01	.94E-14
		1		24		8	-.100000000E+01	.94E-14
MAD6	0	5	163	1141		7	.113104727E+00	.20E-10
		1		1141		7	.113104727E+00	.20E-10
MAD8-10	0	5	18	235		13	.381173963E+00	.26E-11
		1		250		14	.381173963E+00	.20E-13
MAD8-30	0	5	58	1102		18	.547620496E+00	.23E-14
		1		1103		18	.5e7920496E+00	.69E-12
MAD8-50	0	5	98	2156		22	.579276202E+00	.13E-16
		1		2256		23	.579276202E+00	.40E-16
P43M	1	5	3	36	24	11	-.440000000E+02	.20E-03
		1		36	24	11	-.440000000E+02	.20E-03
P84M	4	5	3	9	12	3	-.528033513E+07	.26E+03
		1		9	12	3	-.528033511E+07	.76E-05
P113M	5	5	4	60	91	15	.243062091E+02	.23E-03
		1		61	91	15	.243062091E+02	.11E-02
P117M	3	5	3	56	60	18	.323486790E+02	.31E-04
		1		56	60	18	.323486790E+02	.31E-04

Table 5.1: Numerical Results for (Constrained) Minimax Problems

PROB	CD	NNL	M	NCFI	NCGI	IT	OBJECTIVE	ACC
P12	VF03AD	1		10	10	8	-.300000000E+02	.1E-05
	ALG05.2		5	9	9	8	-.300000000E+02	.1E-05
	ALG05.2		1	9	9	8	-.300000000E+02	.1E-05
P29	VF03AD	1		14	14	12	-.226274209E+02	.1E-04
	ALG05.2		5	13	13	11	-.226274200E+02	.1E-04
	ALG05.2		1	13	13	11	-.226274200E+02	.1E-04
P30	VF03AD	1		14	14	13	.100000000E+01	.1E-06
	ALG05.2		5	14	14	13	.100000000E+01	.1E-06
	ALG05.2		1	14	14	13	.100000000E+01	.1E-06
P31	VF03AD	1		8	8	5	.600000000E+01	.1E-04
	ALG05.2		5	9	9	6	.600000000E+01	.1E-04
	ALG05.2		1	9	9	6	.600000000E+01	.1E-04
P32	VF03AD	1		3	3	2	.100000000E+01	.1E-07
	ALG05.2		5	3	3	2	.100000000E+01	.1E-07
	ALG05.2		1	3	3	2	.100000000E+01	.1E-07
P33	VF03AD	2		5	5	4	-.400000000E+01	.1E-07
	ALG05.2		5	5	5	4	-.400000000E+01	.1E-07
	ALG05.2		1	5	5	4	-.400000000E+01	.1E-07
P34	VF03AD	2		8	8	7	-.834032445E+00	.1E-07
	ALG05.2		5	8	8	7	-.834032445E+00	.1E-07
	ALG05.2		1	8	8	7	-.834032445E+00	.1E-07
P43	VF03AD	3		14	14	11	-.440000000E+02	.1E-04
	ALG05.2		5	13	13	11	-.440000000E+02	.1E-04
	ALG05.2		1	13	13	11	-.440000000E+02	.1E-04
P57	VF03AD	1		4	4	1	.306463060E-01	.1E-04
	ALG05.2		5	5	5	2	.306463061E-01	.1E-04
	ALG05.2		1	5	5	2	.306463061E-01	.1E-04
P66	VF03AD	2		7	7	6	.518163273E+00	.1E-07
	ALG05.2		5	7	7	6	.518163274E+00	.1E-07
	ALG05.2		1	7	7	6	.518163274E+00	.1E-07
P67	VF03AD	2		33	33	24	-.120892092E+03	.1E-04
	ALG05.2		5	29	29	26	-.120892094E+03	.1E-04
	ALG05.2		1	29	29	26	-.120892094E+03	.1E-04
P70	VF03AD	0		32		29	.940282430E-02	.1E-06
	ALG05.2		5	32		29	.940282430E-02	.1E-06
	ALG05.2		1	32		29	.940282430E-02	.1E-06
P76	VF03AD	0		5		4	-.468181818E+01	.1E-03
	ALG05.2		5	5		4	-.468181818E+01	.1E-03
	ALG05.2		1	5		4	-.468181818E+01	.1E-03
P84	VF03AD	6		3	3	2	-.528033513E+07	.1E-01
	ALG05.2		5	3	3	2	-.528033513E+07	.1E-01
	ALG05.2		1	3	3	2	-.528033513E+07	.1E-01
P85	VF03AD	0		35	35	34	-.240083622E+01	.1E-04
	ALG05.2		5	35	35	34	-.240083622E+01	.1E-04
	ALG05.2		1	35	35	34	-.240083622E+01	.1E-04
P93	VF03AD	2		13	13	10	.135075905E+03	.1E-02
	ALG05.2		5	13	13	10	.135075916E+03	.1E-02
	ALG05.2		1	13	13	10	.135075916E+03	.1E-02
P100	VF03AD	4		23	23	15	.680630057E+03	.1E-03
	ALG05.2		5	19	19	12	.680630052E+03	.1E-03
	ALG05.2		1	19	19	12	.680630052E+03	.1E-03
P110	VF03AD	0		9		6	-.457784697E+02	.1E-07
	ALG05.2		5	8		5	-.457784697E+02	.1E-07
	ALG05.2		1	8		5	-.457784697E+02	.1E-07
P113	VF03AD	5		14	14	11	.243063763E+02	.1E-02
	ALG05.2		5	15	15	12	.243063051E+02	.1E-02
	ALG05.2		1	15	15	12	.243063051E+02	.1E-02
P117	VF03AD	5		15	15	15	.323486842E+02	.1E-03
	ALG05.2		5	15	15	14	.323486842E+02	.1E-03
	ALG05.2		1	15	15	14	.323486842E+02	.1E-03

Table 5.2: Numerical Results of Penalty-Type Algorithm

Chapter 6

Finely Discretized Problems from SIP

In this chapter, we first consider the solution of problem (SIP) without constraints, for the clarity of expositions of the main idea. Then, extensions of the proposed scheme are made for the solution of (SIP) in its general constrained form.

6.1 Introduction

Specifically, we consider the following optimization problem

$$(SIP_1) \quad \min_{x \in \mathbb{R}^n} f_\Omega(x)$$

with

$$f_\Omega(x) = \max_{i=1,\dots,p} \max_{\omega \in \Omega} f_i(x, \omega); \quad (6.1)$$

where Ω is a compact interval of real line and $f_i : \mathbb{R}^n \times \Omega \rightarrow \mathbb{R}$, $i = 1, \dots, p$, are continuous and are continuously differentiable with respect to the first argument for each $\omega \in \Omega$. We take $\Omega = [0, 1]$ and $p = 1$ in the sequel. There is no conceptual difficulty in extending our algorithm and analysis to the case of more general Ω and $p > 1$. For notational clarity, we denote $\phi(x, \omega) = f_1(x, \omega)$. The difficulties in solving (SIP_1) stem mostly from the facts that (i) accurate evaluation of $f_\Omega(x)$, at each x , requires a potentially time consuming global maximization over ω , and (ii) $f_\Omega(x)$ is nondifferentiable in general. Many methods have been proposed to circumvent these difficulties. Globally convergent algorithms are mostly based on the approximation to f_Ω by means of discretization. They do not usually achieve fast local convergence (see, *e.g.*, [29,37,62,68,69]). Algorithms that enjoy fast local convergence, on the

other hand, are mostly based on the characterization of maximizers of $\phi(x, \cdot)$ over Ω in the neighborhood of a local solution of (SIP_1) . Under mild assumptions, the set of such maximizers contains a “small” number of points. The solution of the original problem can then be reduced to the solution of a problem characterized at these maximizers. Application of Newton method, or more precisely the classical SQP method to the reduced problem brings about the fast rate of convergence. However, as is for the conventional Newton method, global convergence is not guaranteed in general (see, *e.g.*, [38,40,72]). To induce global convergence, typically, a globally convergent algorithm is used to steer the iterates to the neighborhood of a desirable point (see, *e.g.*, [32,72]). Unfortunately, there is a lack of generally applicable satisfactory switching rules. Exceptions can be found in, *e.g.*, [86] where both global and fast local convergence were achieved by a single algorithm, but the proposed algorithm is potentially expensive in terms of function evaluations, since evaluation of f_Ω at every trial point is required during a line search process which is used to induce global convergence.

In the rest of this chapter, we investigate the possibility of accelerating the rate of convergence of a class of globally convergent algorithms based on discretization. The purpose of discretization in such algorithms is to suitably approximate the set of the maximizers over Ω (see [37] for linear problems and [62,69] for general nonlinear problems). The proposed schemes are based on an adaptively refined discretization of the interval $[0, 1]$. Specifically, a discrete set can be defined, *e.g.*, by

$$\Omega^q = \{0, \frac{1}{q}, \frac{2}{q}, \dots, \frac{q-1}{q}, 1\} \quad (6.2)$$

with q a positive integer and the following problem

$$(SIP'_1) \quad \min_{x \in R^n} f_{\Omega^q}(x)$$

can be solved. The original problem can then be solved by the solution of a sequence of such discrete problem with a progressively refined discretization (larger q). In [69] and [62], essentially (SIP'_1) is solved by means of first order (*i.e.*, slow) methods. In [69], the construction of the search direction at iteration k is based on the gradient $\nabla_x \phi(x_k, \omega)$ at *all* “ ϵ -maximizing” values of $\omega \in \Omega$. In [62], it is shown that a smaller subset of all the ϵ -maximizing values can be used by suitably detecting “critical” values of ω and “remembering” them from iteration to iteration. This scheme is efficient because, under mild assumptions, there are very few points of Ω^q that are effective at a solution x^* (*i.e.*, the set of the maximizers of $\phi(x^*, \cdot)$ over Ω^q contains a

“small” number of points). Thus, computational efficiency is significantly improved (for instance, the dimension of the problem for constructing a search direction can be much smaller and gradients are not required at every grid point) especially when q is very large. However, at each level of discretization (*i.e.*, for each fixed q), these two algorithms exhibit at best a linear rate of convergence.

Our question is : can we adapt the classical SQP scheme to the solution of the discretized problem (SIP'_1), especially when q is large? The key issue is how to construct a quadratic sub-program of “small” size which still generates suitable search directions. In next section, the basic framework of SQP-type algorithm is used in conjunction with the memory scheme proposed in [62]. The convergence analysis is presented in Section 6.3. Extension to constrained SIP problems is made in Section 4 and implementation issues are briefly discussed in Section 5.

6.2 Algorithm

Clearly, techniques such as that described in Chapter 3 for minimax problems may not be efficient if Ω^q contains a large number of points. Also, it may be advantageous to exploit the regularity properties of $\phi(x, \cdot)$ as a function of ω . Given $\epsilon > 0$, we define for $x \in \mathbb{R}^n$ the set of ϵ -active global maximizers by

$$\Omega_\epsilon^f(x) = \{\omega \in \Omega^q : \phi(x, \omega) \geq f_{\Omega^q}(x) - \epsilon\}.$$

In [62], a sequence of subsets $\{\Omega_k^f\}$ of Ω^q is constructed and, at each iteration k , a first order direction d_k^0 is obtained using Ω_k^f . In general, Ω_k^f is much smaller than $\Omega_\epsilon^f(x_k)$ and thus the computation of d_k^0 based on Ω_k^f , instead of on $\Omega_\epsilon^f(x_k)$ as in [69], is apparently more efficient. After a new iterate x_{k+1} is obtained at the end of iteration k , Ω_{k+1}^f is constructed by including (i) all ω 's that globally maximize $(x_{k+1}, \cdot)$ over Ω^q , *i.e.*, all $\omega \in \Omega_0^f(x_{k+1})$; (ii) all ω 's that affected direction d_k^0 ; and (iii) all ω 's that restricted the step length during the line search at iteration k . This updating scheme is adapted in this chapter to the classical SQP scheme. Specifically, based on Ω_k^f at each iteration k , an SQP direction d_k^0 is computed as the solution of the following quadratic problem $QP_3(x_k, H_k, \Omega_k^f)$ defined for $x_k \in \mathbb{R}^n$ and $H_k \in \mathbb{R}^{n \times n}$ symmetric positive definite by

$$\min_{d^0 \in \mathbb{R}^n} \frac{1}{2} \langle d^0, H_k d^0 \rangle + f'_{\Omega_k^f}(x_k, d^0)$$

where

$$f'_{\Omega_k^f}(x_k, d^0) = f_{\Omega_k^f}(x_k, d^0) - f_{\Omega^q}(x_k)$$

is a first order approximation to $f_{\Omega^q}(x_k + d_k^0) - f_{\Omega^q}(x_k)$, with

$$f_{\Omega_k^f}(x_k, d^0) = \max_{\omega \in \Omega_k^f} \{ \phi(x_k, \omega) + \langle \nabla_x \phi(x_k, \omega), d^0 \rangle \}.$$

The first order necessary conditions of optimality for $QP_3(x_k, H_k, \Omega_k^f)$ can be expressed as follows. If d_k^0 solves $QP_3(x_k, H_k, \Omega_k^f)$, there exist $\mu_{k,\omega}$ for all $\omega \in \Omega_k^f$ such that

$$\left. \begin{aligned} H_k d_k^0 + \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \nabla_x \phi(x_k, \omega) &= 0 \\ \mu_{k,\omega} &\geq 0 \quad \forall \omega \in \Omega_k^f \quad \text{and} \quad \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} = 1 \\ \mu_{k,\omega} &= 0 \quad \forall \omega \in \Omega_k^f \quad \text{s.t.} \quad \phi(x_k, \omega) + \langle \nabla_x \phi(x_k, \omega), d_k^0 \rangle < f(x_k, d_k^0). \end{aligned} \right\} \quad (6.3)$$

Thus,

$$\begin{aligned} f_{\Omega_k^f}'(x_k, d^0) &= \max_{\omega \in \Omega_k^f} \{ \phi(x_k, \omega) + \langle \nabla_x \phi(x_k, \omega), d^0 \rangle \} - f_{\Omega^q}(x_k) \\ &= \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \{ \phi(x_k, \omega) + \langle \nabla_x \phi(x_k, \omega), d^0 \rangle \} - f_{\Omega^q}(x_k). \end{aligned}$$

Define the set of “binding” points of Ω^q by

$$\Omega_a^f(x_k) = \{ \omega \in \Omega_k^f : \mu_{k,\omega} > 0 \}. \quad (6.4)$$

Clearly, $\Omega_a^f(x_k)$ is the set containing all ω 's that affected d_k^0 . It is easy to check that d_k^0 is a descent direction for f_{Ω^q} at x_k (Lemma 6.2 below). Thus, an Armijo-type line search can be devised to induce global convergence, namely, for some $\delta_k < 0$, a step t_k , the largest number t in the sequence $\{1, \beta, \beta^2, \dots\}$ satisfying

$$f_{\Omega^q}(x_k + t d_k^0) \leq f_{\Omega^q}(x_k) + \alpha t \delta_k, \quad (6.5)$$

is sought, where $\alpha, \beta \in (0, 1)$ are given. If $t_k < 1$, we can define $\bar{x}_{k+1} = x_k + \frac{t_k}{\beta} d_k^0$. In such case, the set of ω 's that restricted the step length can be expressed by $\Omega_0^f(\bar{x}_{k+1})$, i.e.,

$$\Omega_0^f(\bar{x}_{k+1}) = \{ \omega \in \Omega^q : \phi(\bar{x}_{k+1}, \omega) = f_{\Omega^q}(\bar{x}_{k+1}) \}.$$

Thus, $\Omega_0^f(\bar{x}_{k+1})$ essentially consists of those $\omega \in \Omega^q \setminus \Omega_k^f$ that appear to be critical. With these notations, the aforementioned rule for updating Ω_k^f can be written as

$$\Omega_{k+1}^f = \Omega_0^f(x_{k+1}) \cup \Omega_a(x_k) \cup \Omega_0^f(\bar{x}_{k+1}).$$

Yet, an extremely short step may be required at a specific iteration k due to the fact that some *almost* maximizers are not included in Ω_k^f . Our strategy for updating Ω_k^f ensures that this effect may happen only during early iterations. If it does happen, the idea of null step in nonsmooth optimization (see, *e.g.*, [44,46]) or a nonmonotone line search (see, *e.g.*, [6,31,63] and previous chapters) can be adapted to circumvent this difficulty. We observe however that the null step strategy may not be adequate if evaluation of function and gradient is expensive, as it involves a line search that potentially requires such evaluation at auxiliary points.

As we just mentioned, d_k^0 may not be adequate at a specific iteration in the sense that a very small stepsize is necessary in order to satisfy a line search criterion. If this happens, we would hope that, if Ω_{k+1}^f is properly updated, d_{k+1}^0 would be significantly different from, and usually better than, d_k^0 . However, this need not be the case if H_{k+1} is different from H_k . This gives rise to an additional difficulty to enforce global convergence. H_k may change from iteration to iteration. We note that, even in [62] where only one subsequence is required to converge to a stationary point, the proof there for convergence with refined discretization may no longer hold if $H_{k+1} \neq H_k$ for k large enough. And, if we are interested in the solution of (SIP'_1) on its own, it is often desirable that every convergent subsequence converge to an “optimal” point. This suggests that, at some points, H_k should not be updated. On the other hand, resetting H_k may jeopardize our original intention to speed up the rate of local convergence by suitably constructing H_k . From the analysis for ordinary minimax problems in Chapter 3, we know that, normally, $\{d_k^0\}$ goes to zero and $\{t_k\}$ is bounded away from zero. This is the case in the current context as we can prove that, eventually, all critical points of Ω^g are included in Ω_k^f (Proposition 6.18 below). A possible criterion for resetting H_k would be

$$t_k < \|d_k^0\|.$$

Therefore, H_k will be updated normally close to a solution to retain fast local rate of convergence. But this is not completely suitable since unexpected resetting may take place, in particular, when $\|d_k^0\| \geq 1$. To get around this effect, we prescribe a small number $\underline{\epsilon} \in (0, 1)$ so that we set $H_{k+1} = H_k$ if

$$t_k < \min\{\underline{\epsilon}, \|d_k^0\|\}.$$

Since a search for t_k starts with $t = 1$ in (6.5), taking $\underline{\epsilon} \ll 1$ is beneficial in practice to avoid resetting H_k when t_k is not too small. We will prove that this criterion is

indeed adequate for both establishing global convergence and for maintaining fast rate of local convergence.

Finally, as for ordinary minimax problems in Chapter 3, the line search may truncate the unit step even arbitrarily close to a solution and thus a Maratos-like effect is possible. A step of one is essential for achieving a superlinear rate of convergence. This difficulty is circumvented by a second order correction $\tilde{d}_k$, as in Chapter 3. $\tilde{d}_k$ has to be suitably chosen to guarantee that $d_k^0 + \tilde{d}_k$ converges to d_k^0 in order to preserve the properties of Newton direction d_k^0 . Such a $\tilde{d}_k$ can be chosen, for instance, as the solution of the quadratic program $\widetilde{QP}_3(x_k, d_k^0, H_k, \Omega_k^f)$ defined by

$$\min_{\tilde{d} \in \mathbb{R}^n} \quad \frac{1}{2} \langle d_k^0 + \tilde{d}, H_k(d_k^0 + \tilde{d}) \rangle + \tilde{f}'_{\Omega_k^f}(x_k + d_k^0, x_k, \tilde{d})$$

(but $\tilde{d}_k$ is discarded if too large), where

$$\tilde{f}'_{\Omega_k^f}(x_k + d_k^0, x_k, \tilde{d}) = f_{\Omega_k^f}(x_k + d_k^0, x_k, \tilde{d}) - f_{\Omega^q}(x_k + d_k^0)$$

with

$$f_{\Omega_k^f}(x_k + d_k^0, x_k, \tilde{d}) = \begin{cases} f_{\Omega_k^f}(x_k, d_k^0 + \tilde{d}_k) & \text{if } \Omega_k^f \text{ is a singleton} \\ f_{\Omega_k^f}(x_k + d_k^0, \tilde{d}_k) & \text{otherwise.} \end{cases}$$

The subscript Ω_k^f or Ω^q will be dropped if clear from the context. It was shown in Chapter 3 for ordinary minimax problems that computation of $\tilde{d}_k$ could be eventually avoided if we were content with

$$f(x_k + d_k^0) < f(x_{k-2})$$

instead of a monotone decrease. This remains true (Theorem 6.21) in the current situation. Therefore, under mild assumptions, the Maratos effect can be avoided efficiently and two-step superlinear convergence is maintained. Based on these ideas, the following algorithm is proposed for the solution of (SIP'_1) .

Algorithm 6.1.

Parameters. $\alpha \in (0, \frac{1}{2})$, $\beta \in (0, 1)$, $0 < \epsilon \ll 1$.

Data. $x_0 \in \mathbb{R}^n$, $H_0 \in \mathbb{R}^{n \times n}$ and $H_0 = H_0^T > 0$.

Step 0. Initialization. Set $k = 0$, $x_{-2} = x_{-1} = x_0$ and $\Omega_0^f = \Omega_0^f(x_0)$.

Step 1. Computation of search direction and step length.

(i). Compute d_k^0 by solving $QP_3(x_k, H_k, \Omega_k^f)$. If $\|d_k^0\| = 0$, stop.

(ii). If

$$f_{\Omega^q}(x_k + d_k^0) \leq \max_{\ell=0,1,2} \{f_{\Omega^q}(x_{k-\ell})\} - \frac{\alpha}{2} \langle d_k^0, H_k d_k^0 \rangle. \quad (6.6)$$

set $t_k = 1$, $\tilde{d}_k = 0$ and go to *Step 2*.

(iii). Compute $\tilde{d}_k$ by solving $\widetilde{QP}_3(x_k, d_k^0, H_k, \Omega_k^f)$.

If $\|\tilde{d}_k\| > \|d_k^0\|$, set $\tilde{d}_k = 0$.

(iv). Compute t_k , the first number t in the sequence $\{1, \beta, \beta^2, \dots\}$ satisfying

$$f_{\Omega^q}(x_k + t d_k^0 + t^2 \tilde{d}_k) \leq \max_{\ell=0,1,2} \{f_{\Omega^q}(x_{k-\ell})\} - \frac{\alpha}{2} t \theta \langle d_k^0, H_k d_k^0 \rangle. \quad (6.7)$$

Step 2. Updates. Set $x_{k+1} = x_k + t_k d_k^0 + t_k^2 \tilde{d}_k$. Define $\bar{t}_k = \min\{1, \frac{t_k}{\beta}\}$ and set $\bar{x}_{k+1} = x_k + \bar{t}_k d_k^0 + \bar{t}_k^2 \tilde{d}_k$. Set

$$\Omega_{k+1}^f = \begin{cases} \Omega_0^f(x_{k+1}) \cup \Omega_a^f(x_k) & \text{if } t_k = 1 \\ \Omega_0^f(x_{k+1}) \cup \Omega_a^f(x_k) \cup \Omega_0^f(\bar{x}_{k+1}) & \text{if } t_k < 1. \end{cases}$$

If $t_k < \min\{\underline{\epsilon}, \|d_k^0\|\}$, set $H_{k+1} = H_k$; otherwise, compute a new approximation H_{k+1} to the Hessian of the Lagrangian of (SIP'_1) . Set $k = k + 1$. Go back to *Step 1*.

□

Remark 6.1. As mentioned in the introduction, the purpose of a nonmonotone line search in this algorithm is two fold: to avoid too small a step length during early iterations and, more importantly, to avoid the Maratos effect.

Remark 6.2. We used $-\frac{1}{2} \langle d_k^0, H_k d_k^0 \rangle$ instead of $-\langle d_k^0, H_k d_k^0 \rangle$ in the line search. It is not clear to us whether it is possible to generalize our analysis of global convergence if the latter is used.

6.3 Convergence Analysis

Although (SIP'_1) takes the same form of the minimax problem (P_1) in Chapter 3, the convergence properties established there cannot be directly transposed to the present situation since, at each iteration, only a subset of the discretized set Ω^q is employed in Algorithm 6.1 to construct a search direction, and the discrete set Ω_k is changing from iteration to iteration. In this section, we prove that, under mild assumptions, satisfactory convergence properties hold for Algorithm 6.1.

6.3.1 Global convergence

The global convergence of Algorithm 6.1 is analyzed first.

A6.1. The functions $\phi : \mathbb{R}^n \times [0, 1] \rightarrow \mathbb{R}$ is continuous and, for every $\omega \in [0, 1]$, $\phi(\cdot, \omega)$ is continuously differentiable.¹

A6.2. For any $x_0 \in \mathbb{R}^n$, the level set $L = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is compact (cf. A3.2).

We also assume that A3.3 holds, i.e., there exist $\sigma_1, \sigma_2 > 0$ such that

$$\sigma_1 \|d\|^2 \leq \langle d, H_k d \rangle \leq \sigma_2 \|d\|^2 \quad \forall d \in \mathbb{R}^n, \forall k.$$

Recall that a point $x^* \in \mathbb{R}^n$ is stationary for (SIP'_1) if there exist μ_ω^* at x^* for all $\omega \in \Omega^q$ such that

$$\left. \begin{aligned} \sum_{\omega \in \Omega^q} \mu_\omega^* \nabla_x \phi(x^*, \omega) &= 0 \\ \mu_\omega^* &\geq 0 \quad \forall \omega \in \Omega^q \quad \text{and} \quad \sum_{\omega \in \Omega^q} \mu_\omega^* = 1 \\ \mu_\omega^* &= 0 \quad \forall \omega \in \Omega^q \quad \text{s.t.} \quad \phi(x^*, \omega) < f(x^*). \end{aligned} \right\} \quad (6.8)$$

Lemma 6.1. d_k^0 and $\tilde{d}_k$ are unique KKT points of quadratic problems $QP_3(x_k, H_k, \Omega_k^f)$ and $\widetilde{QP}_3(x_k, d_k^0, H_k, \Omega_k^f)$ respectively.

Proof. In view of A3.3, $QP_3(x_k, H_k, \Omega_k^f)$ is strongly convex and thus has a KKT point. That $\tilde{d}_k$ is the unique KKT point of $\widetilde{QP}_3(x_k, d_k^0, H_k, \Omega_k^f)$ follows similarly. $\square$

The following lemma shows that if $d_k^0 \neq 0$ it is a descent direction for f_{Ω^q} at x_k , even though not all points of Ω^q are used in constructing d_k^0 . We define

$$\delta_k = -\frac{1}{2} \langle d_k^0, H_k d_k^0 \rangle$$

in the rest of analysis to simplify notations.

Lemma 6.2. If d_k^0 nonzero solves $QP_3(x_k, H_k, \Omega_k^f)$, then there exists $\bar{t}_k > 0$ such that, for all $t \in [0, \bar{t}_k]$,

$$f_{\Omega^q}(x_k + t d_k^0 + t^2 \tilde{d}_k) \leq f_{\Omega^q}(x_k) - \frac{\alpha}{2} t \langle d_k^0, H_k d_k^0 \rangle.$$

Proof. It is clear that, in view of (6.3) and A3.3,

$$f'_{\Omega_k^f}(x_k, d_k^0) \leq \delta_k < 0.$$

¹The joint continuity of $\phi(\cdot, \cdot)$ is required actually only when the discretization is adaptively refined. And the continuity of $\phi(x, \cdot)$ can be replaced by piecewise continuity.

Thus, it can be shown by standard argument that there exists $\hat{t}_k \in (0, 1]$ such that, for all $t \in [0, \hat{t}_k]$,

$$\max_{\omega \in \Omega_k^f} \phi(x_k + td_k^0 + t^2 \tilde{d}_k, \omega) \leq f_{\Omega^q}(x_k) + t\alpha f'_{\Omega_k^f}(x_k, d_k^0) \leq f_{\Omega^q}(x_k) + t\alpha\delta_k.$$

On the other hand, $\phi(x_k, \omega) < f_{\Omega^q}(x_k)$ for all $\omega \notin \Omega_k^f$. In view of our continuity assumptions, this implies there exists $\tilde{t}_k \in (0, 1]$ such that, for all $t \in [0, \tilde{t}_k]$,

$$\max_{\omega \notin \Omega_k^f} \phi(x_k + td_k^0 + t^2 \tilde{d}_k, \omega) \leq f_{\Omega^q}(x_k) + t\alpha\delta_k.$$

By choosing $\bar{t}_k = \min\{\hat{t}_k, \tilde{t}_k\}$, our claim is proven. $\square$

Lemma 6.3. The sequences $\{x_k\}$, $\{d_k^0\}$ and $\{\tilde{d}_k\}$ are bounded.

Proof. The line search rules ensure that $f(x_k) \leq f(x_0)$ for all k . The boundedness of $\{x_k\}$ follows from A6.2. Since, in view of (6.3),

$$d_k^0 = -H_k^{-1} \left\{ \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \nabla_x \phi(x_k, \omega) \right\}$$

and $\{\mu_k\}$ is bounded for all k , the boundedness of $\{d_k^0\}$ follows from A3.3, A6.1, and the boundedness of $\{x_k\}$. The boundedness of $\{\tilde{d}_k\}$ follows similarly. $\square$

The following result is important for establishing global convergence, because it enables one to study the optimality of any limiting point of $\{x_k\}$ by studying instead the properties of the sequence $\{d_k^0\}$.

Lemma 6.4. For any k , d_k^0 is zero if and only if x_k is stationary. Furthermore, given $K \subset \mathbb{N}$, $\{d_k^0\}$ converges to zero on K if and only if all accumulation points of $\{x_k\}$ on K are stationary.

Proof. For any k , we associate a $\mu_{k,\omega}$ of zero value for each $\omega \in \Omega^q \setminus \Omega_k^f$. In view of Lemma 6.1, d_k^0 is the unique KKT point that satisfies (6.3). Thus, $d_k^0 = 0$ if and only if (6.8) holds. Therefore, $d_k^0 = 0$ if and only if x_k is stationary.

Now, assume that all accumulation points of $\{x_k\}$ on K are stationary. If $\{d_k^0\}$ does not converge to zero on K , there would exist $K' \subset K$ such that $\inf_{k \in K'} \|d_k^0\| > 0$. In view of the boundedness of $\{\mu_k\}$, A3.3 and Lemma 6.3, there exists $K'' \subset K'$ such that $\{x_k\}$, $\{d_k^0\}$ and $\{H_k\}$ converge on K'' to some x^* , d^{0*} and H^* respectively, and $\{\mu_{k,\omega}\}$ converges to some μ_ω^* on K'' for $\omega \in \Omega_k^f$ and Ω_k^f is a constant set on K'' . Thus, by taking limit of (6.3) on K'' , we have

$$\sum_{\omega \in \Omega_k^f} \mu_\omega^* \nabla_x \phi(x^*, \omega) = -H^* d^{0*} \neq 0$$

and, in view of the uniqueness of stationary point for $QP_3(x^*, H^*, \Omega_*^f)$, x^* is not stationary, a contradiction.

Next, assume that $\{d_k^0\}$ converges to zero on K . Let $K' \subset K$ be such that $\{x_k\}$ converges to some $\hat{x}$ on K' . Similarly to the argument used above, there exists $K'' \subset K'$ such that Ω_k^f is a constant set and $\{\mu_{k,\omega}\}$ converges to some μ_ω^* for $\omega \in \Omega_k^f$, both on K'' . Taking limit of (6.3) yields that $\{x_k\}$ converges on K'' to $\hat{x}$ and $\hat{x}$ is stationary. $\square$

Lemma 6.5. The sequence $\{f(x_k)\}$ converges.

Proof. See proof of Lemma 3.3. $\square$

In view of Lemma 6.5, we now assume that an infinite sequence $\{x_k\}$ is generated by Algorithm 6.1. The following lemma can be proven by an argument similar to that used in the second part of the proof for Lemma 3.2.

Lemma 6.6. The sequences $\{t_k d_k^0\}$ and $\{x_{k+1} - x_k\}$ both converge to zero. $\square$

More notations are defined to simplify later expositions:

$$\begin{aligned} \pi_k(\omega) &= f_{\Omega^q}(x_k) - \phi(x_k, \omega) \\ \gamma_k &= \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \{f_{\Omega^q}(x_k) - \phi(x_k, \omega)\} \\ \gamma_{k+} &= \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \{f_{\Omega^q}(x_{k+1}) - \phi(x_{k+1}, \omega)\} \\ g_k(\omega) &= H_k^{-\frac{1}{2}} \nabla_x \phi(x_k, \omega) \\ p_k &= \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} H_k^{-\frac{1}{2}} \nabla_x \phi(x_k, \omega) \\ p_{k+} &= \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} H_{k+1}^{-\frac{1}{2}} \nabla_x \phi(x_{k+1}, \omega). \end{aligned}$$

The following lemma is instrumental in establishing global convergence.

Lemma 6.7. It holds that

$$\lim_{k \rightarrow \infty} |\gamma_k - \gamma_{k+}| = 0 \tag{6.9}$$

and, for any $K \subset \mathbb{N}$, the implication

$$\lim_{\substack{k \in K \\ k \rightarrow \infty}} \|H_{k+1} - H_k\| = 0 \quad \implies \quad \lim_{\substack{k \in K \\ k \rightarrow \infty}} \|p_k - p_{k+}\| = 0 \tag{6.10}$$

holds.

Proof. It is clear that

$$\begin{aligned}
|\gamma_k - \gamma_{k+1}| &= \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \{|f(x_k) - f(x_{k+1}) + \phi(x_{k+1}, \omega) - \phi(x_k, \omega)|\} \\
&\leq \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \{|f(x_k) - f(x_{k+1})|\} + \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} |\phi(x_{k+1}, \omega) - \phi(x_k, \omega)| \\
&\leq \max_{\omega \in \Omega_k^f} \{|f(x_k) - f(x_{k+1})| + |\phi(x_{k+1}, \omega) - \phi(x_k, \omega)|\}.
\end{aligned}$$

Thus, (6.9) follows from A6.1, Lemma 6.3, Lemma 6.4 and Lemma 6.6. Also,

$$\begin{aligned}
\|p_k - p_{k+1}\| &\leq \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \|H_k^{\frac{1}{2}}\| \cdot \|\nabla_x \phi(x_k, \omega) - \nabla_x \phi(x_{k+1}, \omega)\| \\
&\quad + \|H_k^{\frac{1}{2}} - H_{k+1}^{\frac{1}{2}}\| \cdot \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \|\nabla_x \phi(x_{k+1}, \omega)\| \\
&\leq \max_{\omega \in \Omega_k^f} \|H_k^{\frac{1}{2}}\| \cdot \|\nabla_x \phi(x_k, \omega) - \nabla_x \phi(x_{k+1}, \omega)\| \\
&\quad + \|H_k^{\frac{1}{2}} - H_{k+1}^{\frac{1}{2}}\| \cdot \max_{\omega \in \Omega_k^f} \|\nabla_x \phi(x_{k+1}, \omega)\|
\end{aligned}$$

and then (6.10) follows similarly. $\square$

The dual of $QP_3(x_k, H_k, \Omega_k^f)$, denoted by $\overline{QP}_3(x_k, H_k, \Omega_k^f)$ and given below, will facilitate our analysis.

Lemma 6.8. The quadratic program $\overline{QP}_3(x_k, H_k, \Omega_k^f)$ at x_k is given by

$$\begin{aligned}
\max_{\mu \in R^{|\Omega_k^f|}} & - \left\{ \frac{1}{2} \left\| \sum_{\omega \in \Omega_k^f} \mu_{\omega} H_k^{-\frac{1}{2}} \nabla_x \phi(x_k, \omega) \right\|^2 \right. \\
& \quad \left. + \sum_{\omega \in \Omega_k^f} \mu_{\omega} \{f_{\Omega^g}(x_k) - \phi(x_k, \omega)\} \right\} \\
\text{s.t.} \quad & \mu_{\omega} \geq 0 \quad \forall \omega \in \Omega_k^f \quad \text{and} \quad \sum_{\omega \in \Omega_k^f} \mu_{\omega} = 1,
\end{aligned}$$

where μ_{ω} is an element of μ for each $\omega \in \Omega_k^f$.

Proof. It is proven in Appendix A. $\square$

If we denote the optimal value of $QP_3(x_k, H_k, \Omega_k^f)$ by v_k , namely,

$$v_k = \frac{1}{2} \langle d_k^0, H_k d_k^0 \rangle + f'_{\Omega_k^f}(x_k, d_k^0) \quad (6.11)$$

and the optimal value of $\overline{QP}_3(x_k, H_k, \Omega_k^f)$ by $\bar{v}_k$, namely,

$$\bar{v}_k = - \left\{ \frac{1}{2} \left\| \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} H_k^{-\frac{1}{2}} \nabla_x \phi(x_k, \omega) \right\|^2 \right.$$

$$\begin{aligned}
& + \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \{f_{\Omega^q}(x_k) - \phi(x_k, \omega)\} \Big\} \\
& = - \left\{ \frac{1}{2} \|p_k\|^2 + \gamma_k \right\}, \tag{6.12}
\end{aligned}$$

then, by duality,

$$\bar{v}_k = v_k. \tag{6.13}$$

Also, in view of the optimality conditions (6.3),

$$\begin{aligned}
f'_{\Omega_k^f}(x_k, d_k^0) &= \max_{\omega \in \Omega_k^f} \{ \phi(x_k, \omega) + \langle \nabla_x \phi(x_k, \omega), d_k^0 \rangle \} - f_{\Omega^q}(x_k) \\
&= \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \{ \phi(x_k, \omega) + \langle \nabla_x \phi(x_k, \omega), d_k^0 \rangle \} - f_{\Omega^q}(x_k) \\
&= \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \langle \nabla_x \phi(x_k, \omega), d_k^0 \rangle + \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \phi(x_k, \omega) - f_{\Omega^q}(x_k) \\
&= -\langle d_k^0, H_k d_k^0 \rangle - \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \{ f_{\Omega^q}(x_k) - \phi(x_k, \omega) \} \\
&\leq -\langle d_k^0, H_k d_k^0 \rangle.
\end{aligned}$$

Thus, in view of (6.11),

$$v_k = \frac{1}{2} \langle d_k^0, H_k d_k^0 \rangle + f'_{\Omega_k^f}(x_k, d_k^0) \leq -\frac{1}{2} \langle d_k^0, H_k d_k^0 \rangle = \delta_k. \tag{6.14}$$

(6.13) and (6.14) will be used later on.

Lemma 6.9. Let Ω' be such that $\Omega_k^f \subset \Omega' \subset \Omega^q$ and let v'_k be the optimal value of $QP_3(x_k, H_k, \Omega')$, then,

$$v'_k \geq v_k.$$

Proof. It follows directly from $\Omega_k^f \subset \Omega'$. $\square$

Next lemma shows that the optimality property of $\{d_k^0\}$ is equivalent to that of $\{v_k\}$ in a certain sense.

Lemma 6.10. Let $K \subset \mathbb{N}$. Then, $\{d_k^0\}$ converges to zero on K if and only if $\{v_k\}$ converges to zero on K .

Proof. In view of (6.12) and (6.13),

$$-v_k = -\frac{1}{2} \langle d_k^0, H_k d_k^0 \rangle - f'_{\Omega_k^f}(x_k, d_k^0) \tag{6.15}$$

$$= \frac{1}{2} \|H_k^{\frac{1}{2}} d_k^0\|^2 + \gamma_k \geq \frac{1}{2} \|H_k^{\frac{1}{2}} d_k^0\|^2. \tag{6.16}$$

Therefore, the “if” part follows directly from (6.16) and A3.3. Since $f_{\Omega_k^f}(x_k) = f_{\Omega^q}(x_k)$ for all k , it holds that

$$\lim_{d \rightarrow 0} f'_{\Omega_k^f}(x_k, d) = 0.$$

Thus, the “only if” part follows from (6.15) and from the convergence of $\{d_k^0\}$ to zero on K' . $\square$

The establishment of the global convergence of the algorithm is by means of contradiction. If x_k tends to stay away from a stationary point, we first show that a significantly different direction can be obtained by taking into account more points from Ω^q that appear to be critical. In such situations, the optimal value $-v_k > 0$ will be decreased at x_{k+1} . If this occurs on a subsequence and in consecutively many iterations starting from any point of the subsequence, v_k would become arbitrarily close to zero on the same subsequence. In view of Lemma 6.10, $\{d_k^0\}$ would also become arbitrarily close to zero on the same subsequence. Therefore, in view of Lemma 6.3, $\{x_k\}$ must converge to a stationary point. The contradiction argument is based on a number of lemmas.

In view of Lemma 6.5 and Lemma 6.6, if some accumulation point is not stationary, $\|d_k^0\|$ is bounded away from zero and t_k becomes very small on the corresponding subsequence. In the construction of Ω_{k+1}^f , it is ensured that, in such case, new points will repeatedly be added to Ω_{k+1}^f . This is established in the next lemma.

Lemma 6.11. Let $K \subset \mathbb{N}$ be such that $\{d_k^0\}$ converges on K and $\inf_{k \in K} \|d_k^0\| > 0$. Then, for $k \in K$, k large enough,

$$\Omega_{k+1} \setminus \Omega_k \neq \emptyset. \quad (6.17)$$

Proof. Since $\{d_k\}$ does not converge to zero on K , a similar argument to that in the proof of Theorem 3.3 can be used to show that there is a $\tilde{t} > 0$, independent of k , such that the line search is satisfied for all $t \in [0, \tilde{t}]$ at every point of Ω_k^f . Therefore, any point ω at which the step length is driven to approaching zero must be outside of Ω_k^f for $k \in K$, k large enough. (6.17) then follows from the construction of Ω_{k+1}^f . $\square$

Now, in view of Lemma 6.11, since $\{t_k\}$ converges to zero on K , $\Omega_0^f(\bar{x}_{k+1})$ must contain points that are not in Ω_k^f for $k \in K$, k large enough. The following lemma helps to show that, with $\Omega_0^f(\bar{x}_{k+1})$ included in Ω_{k+1}^f , the optimal value v_{k+1} will become significantly closer to zero than v_k (Lemma 6.14). Thus, the algorithm is ensured not to get stuck at non-stationary point. The proof of this lemma is similar to that of Proposition 5.1 in [60] and is given in Appendix A.

Lemma 6.12. Let $\underline{\delta} < 0$ and suppose that, for some $K \subset \mathbb{N}$, the sequence $\{t_k\}$ converges to zero on K and $-\langle d_k^0, H_k d_k^0 \rangle < \underline{\delta}$ on K . Then, for $k \in K$, k large enough and for all $\bar{\omega} \in \Omega_0^f(\bar{x}_{k+1})$,

$$\phi(x_{k+1}, \bar{\omega}) + \langle \nabla_x \phi(x_{k+1}, \bar{\omega}), d_k^0 \rangle - \max_{\ell=0,1,2} \{f_{\Omega^\ell}(x_{k-\ell})\} \geq -\alpha \langle d_k^0, H_k d_k^0 \rangle. \quad (6.18)$$

□

The smoothness of $\phi(\cdot, \omega)$ over the compact set L and the boundedness of H_k ensure the boundedness of related quantities, as shown below.

Lemma 6.13. There exists a constant $C \geq 1$ such that, for all k ,

$$C \geq \max\{\|p_k\|, \|p_{k+}\|, \|\bar{g}_{k+1}\|, |\gamma_k|\}.$$

Proof. This follows from A3.3, A6.1—A6.3 and the boundedness of μ_k . □

By making use of Lemma 6.12, the following lemma shows that the optimal value v_{k+1} is closer to zero than v_k if t_k is close to zero and d_k is bounded away from zero. Its proof is inspired by that of Lemma 3.4.11 in [44].

Lemma 6.14. Let C be as in Lemma 6.13. Let K and $\underline{d} > 0$ be such that $\{d_k^0\}$ converges on K and $\inf_{k \in K} \|d_k^0\| > \underline{d}$. Then, for $k \in K$ large enough, it holds that

$$-v_{k+1} \leq -v_k - \frac{(1-2\alpha)^2}{8C^2} v_k^2 + \epsilon_k \quad (6.19)$$

for some sequence $\{\epsilon_k\}$ with $\lim_{k \rightarrow \infty} \epsilon_k = 0$.

Proof. Let $s_k = -v_k$ to simplify notations. In the proof, the phrase “ $k \in K$ large enough” is implicit. Let $\bar{\omega}_k \in \Omega_0^f(\bar{x}_{k+1})$ and define $\Omega' = \Omega_a^f(x_k) \cup \{\bar{\omega}_k\}$. In view of their constructions, $\Omega' \subset \Omega_{k+1}^f$. Let $-s'_{k+1}$ ($= \bar{v}'_{k+1}$) denote the optimal value of $\overline{QP}_3(x_{k+1}, H_{k+1}, \Omega')$. In view of Lemma 6.7 and relation (6.13), $\bar{v}'_{k+1} \leq \bar{v}_{k+1}$, or equivalently, $s_{k+1} \leq s'_{k+1}$. Thus, it suffices to prove (6.19) with s_{k+1} ($= -v_{k+1}$) replaced by s'_{k+1} ($= -v'_{k+1}$).

We define the following quadratic function in ν

$$\begin{aligned} Q(\nu) &= \frac{1}{2} \left\| (1-\nu) \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} H_{k+1}^{-\frac{1}{2}} \nabla_x \phi(x_{k+1}, \omega) + \nu H_{k+1}^{-\frac{1}{2}} \nabla_x \phi(x_{k+1}, \bar{\omega}_k) \right\|^2 \\ &\quad + (1-\nu) \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \{f(x_{k+1}) - \phi(x_{k+1}, \omega)\} + \nu \{f(x_{k+1}) - \phi(x_{k+1}, \bar{\omega}_k)\} \\ &= \frac{1}{2} \left\| (1-\nu) p_{k+} + \nu g_{k+1}(\bar{\omega}_k) \right\|^2 + (1-\nu) \gamma_{k+} + \nu \pi_{k+1}(\bar{\omega}_k) \end{aligned}$$

and consider the problem of finding a ν^* such that

$$Q(\nu^*) = \min_{\nu \in [0,1]} Q(\nu).$$

Since $\mu_{k,\omega} = 0$ for all $\omega \in \Omega_k^f \setminus \Omega_a^f(x_k)$, the summation in $Q(\nu)$ can be equivalently performed over $\Omega_a^f(x_k)$. Now, in the quadratic program $\overline{QP}_3(x_{k+1}, H_{k+1}, \Omega')$, if we choose

$$\mu_{\bar{\omega}_k} = \nu^*$$

and

$$\mu_\omega = (1 - \nu^*)\mu_{k,\omega}, \quad \forall \omega \in \Omega_a^f(x_k),$$

the constraints are all satisfied. This implies $\bar{v}'_{k+1} \geq -Q(\nu^*)$, namely,

$$s'_{k+1} \leq Q(\nu^*). \quad (6.20)$$

We next show that (6.19) can be satisfied with s_{k+1} ($= -v_{k+1}$) replaced by $Q(\nu^*)$.

Expanding the quadratic term of $Q(\nu)$ yields

$$\begin{aligned} Q(\nu) &= \frac{1}{2}\|p_{k+}\|^2 + \frac{\nu^2}{2}\|p_{k+} - g_{k+1}(\bar{\omega}_k)\|^2 \\ &\quad + \nu\{\langle g_{k+1}(\bar{\omega}_k), p_{k+} \rangle - \|p_{k+}\|^2\} \\ &\quad + (1 - \nu)\gamma_{k+} + \nu\pi_{k+1}(\bar{\omega}_k). \end{aligned} \quad (6.21)$$

Since, in view of Lemma 6.6, $\{t_k\}$ converges to zero on K , it follows that $t_k < \min\{\underline{\epsilon}, \|d_k^0\|\}$, for $k \in K$. Thus, from *Step 2* in Algorithm 6.1, $H_{k+1} = H_k$ for $k \in K$. Also, in view of A3.3 and $\underline{d} > 0$, there exists $\underline{\delta} < 0$ such that $\delta_k < \underline{\delta}$ for all $k \in K$. And, from the optimality conditions (6.3),

$$p_k = -H_k^{\frac{1}{2}} d_k^0.$$

In view of the fact that $f_{\Omega^q}(x_{k+1}) \leq \max_{\ell=0,1,2} \{f_{\Omega^q}(x_{k-\ell})\}$ and (6.13) and (6.14), we now obtain from (6.18) since $\bar{\omega}_k \in \Omega_0(\bar{x}_{k+1})$, for $k \in K$,

$$\begin{aligned} \langle g_{k+1}(\bar{\omega}_k), p_k \rangle &= -\langle \nabla_x \phi(x_{k+1}, \bar{\omega}_k), d_k^0 \rangle \\ &\leq \phi(x_{k+1}, \bar{\omega}_k) - \max_{\ell=0,1,2} \{f_{\Omega^q}(x_{k-\ell})\} - 2\alpha\delta_k \\ &\leq \phi(x_{k+1}, \bar{\omega}_k) - f_{\Omega^q}(x_{k+1}) - 2\alpha\delta_k \\ &= -\pi_{k+1}(\bar{\omega}_k) - 2\alpha\delta_k \\ &\leq -\pi_{k+1}(\bar{\omega}_k) - 2\alpha v_k = -\pi_{k+1}(\bar{\omega}_k) + 2\alpha s_k. \end{aligned}$$

Hence,

$$\begin{aligned} \langle g_{k+1}(\bar{\omega}_k), p_{k+} \rangle &= \langle g_{k+1}(\bar{\omega}_k), p_k \rangle + \langle g_{k+1}(\bar{\omega}_k), p_{k+} - p_k \rangle \\ &\leq -\pi_{k+1}(\bar{\omega}_k) + 2\alpha s_k + \langle g_{k+1}(\bar{\omega}_k), p_{k+} - p_k \rangle \\ &= -\pi_{k+1}(\bar{\omega}_k) + 2\alpha s_k + O(\|p_{k+} - p_k\|). \end{aligned}$$

Also,

$$\begin{aligned}
\|p_{k+}\|^2 &= \|p_k + p_{k+} - p_k\|^2 \\
&= \|p_k\|^2 + \|p_{k+} - p_k\|^2 + 2\langle p_k, p_{k+} - p_k \rangle \\
&= \|p_k\|^2 + O(\|p_{k+} - p_k\|)
\end{aligned}$$

and, in view of Lemma 6.13,

$$\|p_{k+} - g_{k+1}(\bar{\omega}_k)\|^2 \leq 4C^2.$$

Substituting all these into (6.21) yields, using (6.12),

$$\begin{aligned}
Q(\nu) &\leq \frac{1}{2}\|p_k\|^2 + 2C^2\nu^2 + \nu(2\alpha s_k - \|p_k\|^2) + (1-\nu)\gamma_{k+} \\
&\quad + O(\|p_{k+} - p_k\|) \\
&= \frac{1}{2}\|p_k\|^2 + \gamma_k + 2C^2\nu^2 + \nu(2\alpha s_k - \|p_k\|^2 - \gamma_k) \\
&\quad + (1-\nu)(\gamma_{k+} - \gamma_k) + O(\|p_{k+} - p_k\|) \\
&= s_k + 2C^2\nu^2 + \nu(2\alpha s_k - \frac{1}{2}\nu\|p_k\|^2 - \gamma_k) - \frac{1}{2}\|p_k\|^2 \\
&\quad + O(|\gamma_{k+} - \gamma_k|) + O(\|p_{k+} - p_k\|) \\
&\leq s_k + 2C^2\nu^2 + \nu(2\alpha - 1)s_k \\
&\quad + O(|\gamma_{k+} - \gamma_k|) + O(\|p_{k+} - p_k\|).
\end{aligned}$$

Now, we define

$$\epsilon_k = O(|\gamma_{k+} - \gamma_k|) + O(\|p_{k+} - p_k\|) \quad (6.22)$$

and

$$q_k(\nu) = 2C^2\nu^2 + \nu(2\alpha - 1)s_k.$$

In view of $H_{k+1} = H_k$ for $k \in K'$ and Lemma 6.7, $\lim_{k \rightarrow \infty} \epsilon_k = 0$. And the minimum of $q_k(\nu)$ is achieved with $\bar{\nu}_k = \frac{(1-2\alpha)}{4C^2}s_k$. It follows that

$$Q(\bar{\nu}_k) \leq s_k + q_k(\bar{\nu}_k) + \epsilon_k = s_k - \frac{(1-2\alpha)^2}{8C^2}s_k^2 + \epsilon_k.$$

In view of (6.12) and Lemma 6.13,

$$0 \leq s_k \leq \frac{1}{2}C^2 + C \leq 2C^2 \quad (6.23)$$

and thus $0 < \bar{\nu}_k < 1$ since $\alpha \in (0, \frac{1}{2})$. Therefore, (6.19) follows from the relation

$$s_{k+1} \leq s'_{k+1} \leq Q(\nu^*) \leq Q(\bar{\nu}_k).$$

□

Using the following lemma, it can be shown that, if assumptions of Lemma 6.14 are satisfied in consecutively many iterations starting at any x_k , the optimal value $-v_k > 0$ will be significantly decreased.

Lemma 6.15. Let $\eta > 0$ and $\rho \in (0, 1)$. Let $\{z_k\}$ be the sequence of real numbers satisfying

$$0 \leq z_{k+1} \leq z_k - \eta z_k^2 + \tau_k \text{ and } \eta z_k < 1, \quad \forall k \in \mathbb{N} \quad (6.24)$$

for some $\tau_k > 0$. If $\tau_k \leq \rho \eta z_k^2$ for all k , then $\{z_k\}$ converges to zero.

Proof. From (6.24), it holds that, for all $k \in \mathbb{N}$,

$$\begin{aligned} z_k - z_{k+1} &\geq \eta z_k^2 - \tau_k \\ &\geq \eta z_k^2 - \rho \eta z_k^2 = \eta(1 - \rho) z_k^2. \end{aligned}$$

Summing up both sides gives

$$z_0 - z_{k+1} \geq \eta(1 - \rho) \sum_{i=0}^k z_i^2.$$

Since

$$z_0 - z_{k+1} \leq z_0 < \frac{1}{\eta} < \infty,$$

it holds that

$$\sum_{i=0}^k z_i^2 < \infty, \quad \forall k$$

and the claim follows. □

If we set $\eta = \frac{(1-2\alpha)^2}{8C^2}$ and $z_k = s_k = -v_k$ in above lemma, the condition $\eta s_k < 1$ is automatically satisfied in view of (6.23) and Lemma 6.13.

The following lemma shows that the same convergence properties on some subsequence can be carried over to any subsequences that are finitely many iterations away from the original subsequence.

Lemma 6.16. Suppose that there exists $K \subset \mathbb{N}$ such that $\{x_k\}$ converges to x^* on K and suppose that, for some $K' \subset K$, $\{d_k^0\}$ converges on K' and $\inf_{k \in K'} \|d_k^0\| > 0$. Then, for any finite integer N ,

- (i). $\{x_{k+N}\}_{k \in K}$ converges to x^* ;
- (ii). $\inf_{k \in K'} \|d_{k+N}^0\| > 0$ and $\inf_{k \in K'} |v_{k+N}| > 0$;
- (iii). $\{t_{k+N}\}_{k \in K'}$ converges to zero;

(iv). there exists k_0 such that $H_{k+N} = H_k$ for $k \in K'$, $k \geq k_0$.

Proof. (i) follows directly from Lemma 6.6. In view of Lemma 6.5 and $\inf_{k \in K'} \|d_k^0\| > 0$, x^* is not stationary. Thus, in view of (i) and Lemma 6.5, $\inf_{k \in K'} \|d_{k+N}^0\| > 0$. (ii) follows from Lemma 6.10. (iii) follows from Lemma 6.6 and (ii). In view of (ii) and (iii), there exists k_0 such that $t_{k+N} \leq \{\underline{\epsilon}, \|d_{k+N}^0\|\}$ for $k \in K'$, $k \geq k_0$. Therefore, (iv) follows. $\square$

We are now ready to establish the global convergence of Algorithm 6.1.

Theorem 6.17. Let $\{x_k\}$ be the sequence generated by Algorithm 6.1. Then, every accumulation point of the sequence is stationary.

Proof. Let x^* and K be such that $\{x_k\}$ converges to x^* on K . Proceeding by contradiction, we assume x^* is not stationary. In view of Lemma 6.5, there exists $K' \subset K$ such that $\inf_{k \in K'} \|d_k^0\| > 0$. Let $s_k = -v_k$. In view of Lemma 6.3, there exists $K'' \subset K'$ such that $\{d_k^0\}$ converges on K'' . This in turn implies that $\inf_{k \in K''} s_k > 0$ in view of Lemma 6.10. Thus, there exists $\epsilon_1 > 0$ and N_1 such that

$$s_k > \epsilon_1 \quad \forall k > N_1, k \in K''. \quad (6.25)$$

On the other hand, in view of Lemma 6.16, assumptions of Lemma 6.12 and Lemma 6.14 are satisfied on K'' and on any finitely many consecutive iterations starting from any $k \in K''$. Now, pick up any $k_1 \in K''$ such that $k_1 > N_1$ and, in Lemma 6.15, let $\tau_k = \epsilon_k$, which is defined in (6.22), and let $z_k = s_k$. Since $\lim_{k \rightarrow \infty} \epsilon_k = 0$ and $\inf_{k \in K''} s_k > 0$, in view of Lemma 6.10, there exists $\rho \in (0, 1)$ such that $\tau_k \leq \rho \eta z_k^2$ for $k \in K''$, k large enough and for any finitely many consecutive iterations. By choosing $\epsilon = \frac{1}{2}\epsilon_1$, in view of Lemma 6.14 and Lemma 6.15, there exists N_ϵ such that, for any N_2 such that $N_\epsilon + N_2 + k_1 > k_2$ with any $k_2 \in K''$ such that $k_2 > k_1 + N_\epsilon$,

$$s_{k_1+i} < \epsilon \quad \forall i \in [N_\epsilon, N_\epsilon + N_2].$$

In particular,

$$s_{k_2} < \epsilon = \frac{1}{2}\epsilon_1$$

which contradicts (6.25). Therefore, $\{d_k^0\}$ converges to zero on K . The claim then follows from Lemma 6.5. $\square$

6.3.2 Local convergence

Next, we investigate the rate of local convergence of $\{x_k\}$ generated by Algorithm 6.1. Assumption A6.1 is replaced by

A6.1'. The function $\phi : \mathbb{R}^n \times [0, 1] \rightarrow \mathbb{R}$ is continuous and $\phi(\cdot, \omega)$, for every $\omega \in [0, 1]$, is three times continuously differentiable.

Let x^* be an accumulation point of $\{x_k\}$ and, for all $\omega \in \Omega^q$, let μ_ω^* be the corresponding multipliers.

A6.3. At x^* , any scalars μ_ω , $\omega \in \Omega_0^f(x^*)$, satisfying

$$\sum_{\omega \in \Omega_0^f(x^*)} \mu_\omega \nabla_x \phi(x^*, \omega) = 0 \quad \text{and} \quad \sum_{\omega \in \Omega_0^f(x^*)} \mu_\omega = 0$$

must be all zero.

A6.4. The second order sufficiency conditions with strict complementary slackness are satisfied at x^* (cf. A3.5 and Chapter 2).

Proposition 6.18. (i) The entire sequence $\{x_k\}$ converges to x^* and the entire sequence $\{d_k^0\}$ converges to zero. (ii) For k large enough, the discrete set Ω_k^f contains exactly all critical points, i.e.,

$$\Omega_k^f = \Omega_0^f(x^*).$$

(iii) The multiplier vector μ_k associated with the solution d_k^0 of $QP_3(x_k, H_k, \Omega_k^f)$ converges to the corresponding components of μ^* and, for k large enough,

$$\{\omega \in \Omega_k^f : \mu_{k,\omega} > 0\} = \Omega_0^f(x^*). \quad (6.26)$$

(iv).

$$\|\tilde{d}_k^0\| = O(\|d_k^0\|^2). \quad (6.27)$$

Proof. A6.3 and A6.4 imply that x^* is an isolated minimizer. Thus, the convergence of $\{x_k\}$ can be proven by standard argument. The convergence of $\{d_k\}$ to zero follows directly from Lemma 6.5.

To prove (ii), we first show, for all k large enough,

$$\Omega_k^f \supset \Omega_0^f(x^*).$$

Proceeding by contradiction, we assume that there exist $K \subset \mathbb{N}$ and some $\bar{\omega} \in \Omega_0^f(x^*)$ such that $\bar{\omega} \notin \Omega_k^f$ for all $k \in K$. And, in view of (i), there exists $K' \subset K$ such that Ω_k^f is a constant set on K' . By associating a multiplier $\mu_{k,\bar{\omega}}$ of zero value with $\bar{\omega}$ for $k \in K'$, we have that $\mu_{k,\bar{\omega}} = \mu_{\bar{\omega}}^* = 0$ for all $k \in K'$. However, $\phi(x^*, \bar{\omega}) = f(x^*)$. In view of the optimality of x^* , the strict complementary slackness assumption (cf.

A6.4) is violated. Therefore, any $\bar{\omega} \in \Omega_0^f(x^*)$ is in Ω_k^f for k large enough. On the other hand, in view of continuity of ϕ and f and (i), for k large enough,

$$\Omega_0^f(x_k) \subset \Omega_0^f(x^*)$$

and, in view of (i),

$$\Omega_a(x_{k-1}) \subset \Omega_0^f(x^*).$$

Also, for $t_{k-1} < 1$,

$$\Omega_0^f(\bar{x}_k) = \{\omega : \phi(\bar{x}_k, \omega) = f(\bar{x}_k) > f(x_{k-1}) + \alpha \bar{t}_{k-1} \delta_{k-1}\}.$$

Thus, in view of (i) and Lemma 6.4 and for k large enough,

$$\Omega_0^f(\bar{x}_k) \subset \Omega_0^f(x^*).$$

By construction,

$$\Omega_k^f \subset \Omega_0^f(x^*)$$

and (ii) is proven.

The rest of the proposition therefore follows from Proposition 3.4 in view of (i) and (ii). $\square$

In view of Proposition 6.18, we can carry over all the results of local convergence in Chapter 3, provided the Hessian matrix is not reset for k large enough.

Lemma 6.19. For k large enough, $t_k > \min\{\underline{\epsilon}, \|d_k^0\|\}$.

Proof. We show that $\{t_k\}$ is uniformly bounded away from zero. In view of A6.1' and Proposition 6.18(iv), for $t \in [0, 1]$, there exist $C_1 > 0$ and $C_2 > 0$ such that, for all $\omega \in \Omega^q$,

$$\phi(x_k + td_k^0 + t^2 \tilde{d}_k, \omega) \leq \phi(x_k, \omega) + t \langle \nabla_x \phi(x_k, \omega), d_k^0 \rangle + C_1 t^2 \|d_k^0\|^2 \quad (6.28)$$

and

$$\phi(x_k + td_k^0 + t^2 \tilde{d}_k, \omega) \leq \phi(x_k, \omega) + C_2 t \|d_k^0\|. \quad (6.29)$$

Now, for any $\omega \in \Omega_k^f$, in view of (6.3) and (6.28) and A3.3,

$$\begin{aligned} & \phi(x_k + td_k^0 + t^2 \tilde{d}_k, \omega) \\ & \leq (1-t)\phi(x_k, \omega) + t\{\phi(x_k, \omega) + \langle \nabla_x \phi(x_k, \omega), d_k^0 \rangle\} + C_1 t^2 \|d_k^0\|^2 \\ & \leq (1-t)\phi(x_k, \omega) + t \max_{\omega \in \Omega_k^f} \{\phi(x_k, \omega) + \langle \nabla_x \phi(x_k, \omega), d_k^0 \rangle\} + C_1 t^2 \|d_k^0\|^2 \end{aligned}$$

$$\begin{aligned}
&= (1-t)\phi(x_k, \omega) + t \sum_{\omega \in \Omega_k^f} \mu_{k,\omega} \{ \phi(x_k, \omega) + \langle \nabla_x \phi(x_k, \omega), d_k^0 \rangle \} + C_1 t^2 \|d_k^0\|^2 \\
&\leq f(x_k) - t \langle d_k^0, H_k d_k^0 \rangle + C_1 t^2 \|d_k^0\|^2 \\
&= f(x_k) + \alpha t \delta_k + t \left(\frac{\alpha}{2} - 1 \right) \langle d_k^0, H_k d_k^0 \rangle + C_1 t^2 \|d_k^0\|^2 \\
&\leq f(x_k) + \alpha t \delta_k + t \left(\frac{\alpha}{2} - 1 \right) \sigma_1 \|d_k^0\|^2 + C_1 t^2 \|d_k^0\|^2 \\
&= f(x_k) + \alpha t \delta_k + t \|d_k^0\|^2 \underbrace{\left\{ \left(\frac{\alpha}{2} - 1 \right) \sigma_1 + C_1 t \right\}}_{< 0 \text{ for } t \text{ small}}.
\end{aligned}$$

Thus, with $\hat{t} = \frac{(1-\frac{\alpha}{2})\sigma_1}{C_1}$, for all $t \in [0, \hat{t}]$,

$$\phi(x_k + t d_k^0 + t^2 \tilde{d}_k, \omega) \leq f(x_k) + \alpha t \delta_k. \quad (6.30)$$

On the other hand, since $\phi(x^*, \omega) < f(x^*)$ for all $\omega \in \Omega^q \setminus \Omega_k^f$, there exists $\epsilon^* > 0$ such that

$$\phi(x^*, \omega) \leq f(x^*) - \epsilon^*, \quad \forall \omega \in \Omega^q \setminus \Omega_k^f.$$

It follows that, for k large enough,

$$\phi(x_k, \omega) \leq f(x_k) - \frac{\epsilon^*}{2}, \quad \forall \omega \in \Omega^q \setminus \Omega_k^f. \quad (6.31)$$

We obtain, from (6.29) and (6.31), for all $\omega \in \Omega^q \setminus \Omega_k^f$,

$$\begin{aligned}
\phi(x_k + t d_k^0 + t^2 \tilde{d}_k, \omega) &\leq \phi(x_k, \omega) + C_2 t \|d_k^0\| \\
&\leq f(x_k) - \frac{\epsilon^*}{2} + C_2 t \|d_k^0\| \\
&= f(x_k) + \alpha t \delta_k - \alpha t \delta_k - \frac{\epsilon^*}{2} + C_2 t \|d_k^0\| \\
&\leq f(x_k) + \alpha t \delta_k - \frac{\alpha}{2} t \sigma_1 \|d_k^0\|^2 - \frac{\epsilon^*}{2} + C_2 t \|d_k^0\|.
\end{aligned}$$

Thus, there exists $\tilde{t} > 0$, independent of k , such that, for all $t \in [0, \tilde{t}]$,

$$\phi(x_k + t d_k^0 + t^2 \tilde{d}_k, \omega) \leq f(x_k) + \alpha t \delta_k, \quad \forall \omega \in \Omega^q \setminus \Omega_k^f.$$

By taking $\bar{t} = \beta \min\{\hat{t}, \tilde{t}\}$, it follows that, for k large enough, $t_k \geq \bar{t}$ and

$$\begin{aligned}
f(x_k + t_k d_k^0 + t_k^2 \tilde{d}_k) &\leq f(x_k) + \alpha t_k \delta_k \\
&\leq \max_{\ell=0,1,2} \{f(x_{k-\ell})\} + \alpha t_k \delta_k.
\end{aligned}$$

The lemma is proven in view of the convergence of $\{d_k^0\}$ to zero. $\square$

Therefore, the resetting scheme for H_k does not impair the rate of local convergence. Without loss of generality, we assume that $\Omega_0^f(x^*) = \{\omega_1, \dots, \omega_s\}$ for some s and define, for any $\omega_j \in \Omega_0^f(x^*)$, $\bar{f}_j(x) = [\phi(x, \omega_i) - \phi(x, \omega_j) : \forall \omega_i \in \Omega_0^f(x^*) \setminus \{\omega_j\}]^T$.

A6.5. H_k approximates the Hessian of the Lagrangian in the sense that

$$\lim_{k \rightarrow \infty} \frac{\|P_k \{H_k - \nabla_{xx}^2 L(x^*, \mu^*)\} P_k d_k^0\|}{\|d_k^0\|} = 0, \quad (6.32)$$

where the matrices P_k are defined by

$$P_k = I - R_k(R_k^T R_k)^{-1} R_k^T$$

with $R_k = \frac{\partial \bar{f}_1^T}{\partial x}(x_k)$ (in view of A6.3, $R_k^T R_k$ is invertible for k large enough).

It is clear now, in view of Proposition 6.18, local results in Chapter 3 hold for Algorithm 6.1.

Proposition 6.20. Under the stated assumptions, t_k is one for k large enough and the convergence rate is two-step superlinear, *i.e.*,

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+2} - x^*\|}{\|x_k - x^*\|} = 0.$$

□.

The result concerning the use of nonmonotone line search is given below.

Theorem 6.21. For k large enough, $x_k + d_k^0$ is always accepted and *Step 1(iii)* (computation of $\tilde{d}_k$) is not performed. □

6.4 Extension to constrained SIP problems

Technical details are given in this section for the extension of the scheme of updating Ω_k^f to the solution of the constrained (SIP) problem. We assume $m = 1$ to simplify expositions. Specifically, for the problem

$$(SIP_2) \quad \min_{x \in R^n} f_\Omega(x) \quad \text{s.t.} \quad g(x, \omega) \leq 0, \quad \forall \omega \in \Omega = [0, 1],$$

we derive the corresponding discrete problem

$$(SIP_2') \quad \min_{x \in R^n} f_{\Omega^q}(x) \quad \text{s.t.} \quad g(x, \omega) \leq 0, \quad \forall \omega \in \Omega^q,$$

where Ω^q is defined by (6.2).

We observe that, if feasible iterates are not required, a penalty function similar to (5.16) can be devised and the solution of (SIP_2') can be transferred to the solution

of the corresponding unconstrained problem. In this case, since the penalty function takes exactly the form of (SIP'_1) , Algorithm 6.1 can be applied with corresponding modifications, provided a scheme is properly incorporated to update the penalty parameter. In what follows, we present extensions to the solution of (SIP'_2) with feasible iterates. As can be expected, the framework of Algorithm 4.1 will be borrowed over.

A6.6. $g : \mathbb{R}^n \times [0, 1] \rightarrow \mathbb{R}$ is continuous and $g(\cdot, \omega)$, for every $\omega \in [0, 1]$, is continuously differentiable.

Define the feasible set by

$$X = \{x \in \mathbb{R} : g(x, \omega) \leq 0, \forall \omega \in \Omega^g\}$$

and

$$g_{\Omega^g}^+(x) = \max\{0; g(x, \omega) \forall \omega \in [0, 1]\}.$$

The ϵ -active set of global maximizers of the constraint is then defined by

$$\Omega_\epsilon^g(x) = \{\omega \in \Omega^g : g(x, \omega) \geq g_{\Omega^g}^+(x) - \epsilon\}.$$

A6.7. For every $x \in X$, the vectors of gradients $\nabla_x g(x, \omega)$, for all $\omega \in \Omega_0^g(x)$, are linearly independent.

A sequence $\{\Omega_k^g\}$ is constructed in a similar way to the construction of $\{\Omega_k^f\}$. Given Ω_k^f and Ω_k^g , the corresponding quadratic sub-problem $QP_4(x_k, H_k, \Omega_k^f, \Omega_k^g)$ for determining d_k^0 is defined by

$$\begin{aligned} \min_{d^0 \in \mathbb{R}^n} \quad & \frac{1}{2} \langle d^0, H_k d^0 \rangle + f'_{\Omega_k^f}(x_k, d^0) \\ \text{s.t.} \quad & g(x_k, \omega) + \langle \nabla_x g(x_k, \omega), d^0 \rangle \leq 0, \quad \forall \omega \in \Omega_k^g. \end{aligned}$$

The set of “binding” points of constraints is defined by

$$\Omega_a^g(x_k) = \{\omega \in \Omega_k^g : \lambda_{k,\omega} > 0\}$$

with λ_k the multiplier vector associated with constraints in quadratic problem $QP_4(x_k, H_k, \Omega_k^f, \Omega_k^g)$. The quadratic sub-problems for determining d_k^1 , a feasible descent direction, and for determining the second order correction $\tilde{d}_k$ can be defined accordingly based on Ω_k^f and Ω_k^g . Now, the algorithm can be presented.

Algorithm 6.2.

Parameters. $\alpha \in (0, \frac{1}{2})$, $\beta \in (0, 1)$, $\theta \in (0, 1)$, $\gamma \in (2, 3)$, $\underline{C} > 0$, $\underline{d} > 0$, $0 < \underline{\epsilon} \ll 1$.

Data. $x_0 \in X$, $H_0 \in \mathbb{R}^{n \times n}$ and $H_0 = H_0^T > 0$, $C_0 = \underline{C}$.

Step 0. Initialization. Set $k = 0$, $x_{-2} = x_{-1} = x_0$. Set $\Omega_0^f = \Omega_0^f(x_0)$ and $\Omega_0^g = \Omega_0^g(x_0)$.

Step 1. Computation of a new iterate.

- i. Compute d_k^0 by solving $QP_4(x_k, H_k, \Omega_k^f, \Omega_k^g)$. If $d_k^0 = 0$, stop.
- ii. Compute $d_k^1 = d^1(x_k, \Omega_k^g)$, let $\varsigma_k = \min\{C_k \|d_k^0\|^2, \|d_k^0\|\}$ and define values $\rho_{k,\omega}$ for $\omega \in \Omega_k^g$ by $\rho_{k,\omega}$ equal to zero if

$$g(x_k, \omega) + \langle \nabla_x g(x_k, \omega), d_k^0 \rangle \leq -\varsigma_k$$

or equal the maximum ρ in $[0, 1]$ such that

$$g(x_k, \omega) + \langle \nabla_x g(x_k, \omega), (1 - \rho)d_k^0 + \rho d_k^1 \rangle \geq -\varsigma_k$$

otherwise. Finally, let $\rho_k^\ell = \max_{\omega \in \Omega_k^g} \{\rho_{k,\omega}\}$.

- iii. Obtain a “local” direction

$$d_k^\ell = (1 - \rho_k^\ell)d_k^0 + \rho_k^\ell d_k^1.$$

- iv. If

$$f(x_k + d_k^\ell) \leq \max_{\ell=0,1,2} \{f(x_{k-\ell})\} - \frac{\alpha}{2} \langle d_k^0, H_k d_k^0 \rangle \quad (6.33)$$

and

$$g(x_k + d_k^\ell, \omega) \leq 0, \quad \omega \in \Omega^g,$$

set $t_k = 1$, $x_{k+1} = x_k + d_k^\ell$ and go to *Step 2*. Otherwise, go to *Step 1 v*.

- v. Obtain a “global” direction

$$d_k^g = (1 - \rho_k^g)d_k^0 + \rho_k^g d_k^1,$$

where ρ_k^g is the largest number in $[0, \rho_k^\ell]$ such that

$$f'(x_k, d_k^g) \leq \theta f'(x_k, d_k^0).$$

- vi. Compute $\tilde{d}_k$ by solving $\widetilde{QP}_4(x_k, d_k^g, H_k, \Omega_k^f, \Omega_k^g)$. If there is no solution or if $\|\tilde{d}_k\| > \|d_k^0\|$, set $\tilde{d}_k = 0$.

- vii. Compute t_k , the first number t in the sequence $\{1, \beta, \beta^2, \dots\}$ satisfying

$$f_{\Omega^q}(x_k + td_k^g + t^2 \tilde{d}_k) \leq \max_{\ell=0,1,2} \{f_{\Omega^q}(x_{k-\ell})\} - \frac{\alpha}{2} t \theta \langle d_k^0, H_k d_k^0 \rangle, \quad (6.34)$$

$$g(x_k + td_k^g + t^2 \tilde{d}_k, \omega) \leq 0, \quad \forall \omega \in \Omega^q$$

and set $x_{k+1} = x_k + t_k d_k^g + t_k^2 \tilde{d}_k$.

Step 2. Updates.

- i. Define $\bar{t}_k = \min\{1, \frac{t_k}{\beta}\}$ and set $\bar{x}_{k+1} = x_k + \bar{t}_k d_k^g + \bar{t}_k^2 \tilde{d}_k$. Set

$$\Omega_{k+1}^f = \begin{cases} \Omega_0^f(x_{k+1}) \cup \Omega_a^f(x_k) & \text{if } t_k = 1 \\ \Omega_0^f(x_{k+1}) \cup \Omega_a^f(x_k) \cup \Omega_0^f(\bar{x}_{k+1}) & \text{if } t_k < 1, \end{cases}$$

$$\Omega_{k+1}^g = \begin{cases} \Omega_0^g(x_{k+1}) \cup \Omega_a^g(x_k) & \text{if } t_k = 1 \\ \Omega_0^g(x_{k+1}) \cup \Omega_a^g(x_k) \cup \Omega_0^g(\bar{x}_{k+1}) & \text{if } t_k < 1. \end{cases}$$

- ii. If $t_k < \min\{\underline{\epsilon}, \|d_k^0\|\}$, set $H_{k+1} = H_k$; otherwise, compute a new approximation H_{k+1} to the Hessian of the Lagrangian of (SIP'_2) .
- iii. If $\|d_k^0\| > \underline{d}$, set $C_k = \underline{C}$. Otherwise, if $g(x_k + d_k^l, \omega) \leq 0, \quad \forall \omega \in \Omega^q$, set $C_{k+1} = C_k$. Otherwise, set $C_{k+1} = 2C_k$.
- iv. Increase k by 1 and go back to *Step 1*.

□

Remark 6.3. If $-\frac{1}{2}\theta \langle d_k^0, H_k d_k^0 \rangle$ in (6.34) were replaced by $f'(x_k, d_k^g)$, it is not clear to us how to generalize our analysis of global convergence for Algorithm 6.1 to Algorithm 6.2.

6.5 Implementation issues

The updating rule for Ω_k^f (and Ω_k^g), as presented in previous sections, accumulates minimum information that is required to ensure global convergence. In practice, more points of interest can be added to Ω_k^f , especially in early iterations. This may help identify critical points more quickly and thus to improve the overall performance of the algorithm. Certainly, careful consideration has to be given to select additional points, as it may as well increase the amount of computation significantly. Following

the idea in [30,62], in a slightly different context, we add a set of ϵ -active left local maximizers, whenever they are not in Ω_k^f , to Ω_k^f at each x_k .

A left local maximizer ($\ell\ell m$) of ϕ over Ω^q at x is a point satisfying the two inequalities, or whichever is well-defined (remember that ϕ is defined on $\mathbb{R}^n \times [0, 1]$),

$$\phi(x, \omega) > \phi(x, \omega - \frac{1}{q})$$

$$\phi(x, \omega) \geq \phi(x, \omega + \frac{1}{q}).$$

Given $\epsilon > 0$, the set of ϵ -active left local maximizers is given by

$$\Omega_\epsilon^{f\ell\ell m}(x) = \{\omega \in \Omega_\epsilon^f(x) : \omega \text{ is a left local maximizer of } \phi \text{ over } \Omega^q \text{ at } x\}.$$

Similarly, we can define $\Omega_\epsilon^{g\ell\ell m}(x)$ for constraints. In the implementation, $\epsilon = \max\{1, 0.1|f_{\Omega^q}(x_k)|\}$ for each k .

The update of $\{H_k\}$ is important for achieving fast rate of convergence. Currently, the BFGS formula with Powell's modification is used as in ordinary situation, with the specification that the value of multipliers at points not in the discrete sets Ω_k^f and Ω_k^g is all zero.

6.6 Numerical experiments

Algorithm 6.2, which covers Algorithm 6.1, has been implemented in CONSOL-OPTCAD [22], an optimization-based interactive engineering systems design tool, except that the nonmonotone line search is not yet included.

We only found a small number of SIP problems in the literature and, due to the insufficient reported information, comparisons of our algorithms with existing algorithms are difficult. Thus, Table 6.1 only contains the numerical performance of our algorithm on problems from [85] with the same problem numbers. In the table, symbols are consistent with those used before. In addition, **NVAR** stands for the number of variables, **MESHP** for the number of points in Ω^q with uniform discretization, **NCFG** for the number of evaluations of both objective and constraints and **NAV** for the average number of such evaluations per iteration (CONSOL-OPTCAD compute both values simultaneously). **NACT** indicates the number of points in Ω_k^f or Ω_k^g at the final iterate. The algorithm is terminated either if $\|d_k^0\| \leq 2 \times 10^{-8}$ or if the objective value is smaller than the known optimal value and does not seem to improve significantly. Since we have not discussed the issue of generating a feasible initial point, each problem is tested with a hand-picked feasible point. Specifically, $x_0 = (1, 2)^T$ for **EXAMPL2**,

$x_0 = (-100, 1, 1)^T$ for **EXAMPL3**, $x_0 = (5, \dots, 5)^T$ for **EXAMPL4**, $x_0 = (1, 0.5, 0)^T$ for **EXAMPL5** and $x_0 = (0.5, -2)^T$ for **EXAMPL6**. For **EXAMPL6**, the constraint is not well scaled and the algorithm took a large number of function evaluation in the first iteration to satisfy our line search criteria. This also suggests that the initial sets Ω_0^f and Ω_0^g should be chosen in a more sophisticated fashion. The data in the parenthesis correspond to the numbers after the first iteration.

It can be observed that the number of **NACT** is much smaller than that of **MESHP**. Normally, it holds that $\text{NACT} \leq \text{NVAR} + 3$ if Ω_k^f and Ω_k^g are constructed by formulas given in our algorithms.

In Chapter 7, a number of typical engineering design problems are solved by means of SIP optimization using **CONSOL-OPTCAD**. There we demonstrate a basic procedure that a designer would follow to solve a specific design problem.

PROB	NVAR	MESHP	NCFG	IT	NAV	NACT	OBJECTIVE
EXAMPL2	2	100	9	4	2.0	4	2.6180360
		500	9	4	2.0	3	2.6180360
EXAMPL3	3	100	41	15	2.7	5	5.3346903
		500	28	11	2.5	4	5.3346903
EXAMPL4	3	100	28	12	2.3	6	0.6490311
		500	30	13	2.2	5	0.6490311
EXAMPL4	6	100	41	16	2.5	9	0.6168058
		500	48	17	2.8	8	0.6168325
EXAMPL4	8	100	63	20	3.1	9	0.6163822
		500	67	22	3.0	8	0.6163822
EXAMPL5	3	100	10	4	2.3	7	4.3011579
		500	10	4	2.3	5	4.3011838
EXAMPL6	2	100	53(19)	7	7.4(3.2)	4	97.159034
		500	57(22)	7	8.0(3.7)	3	97.159034

Table 6.1: Numerical Performance of Algorithm 6.2

Chapter 7

Examples of Engineering Design

This chapter provides some engineering design examples. Problem specifications and the resulting designs need not be the best in the particular engineering context. The basic purpose is to illustrate the procedures to solve engineering problems by optimization technique and to demonstrate the efficiency of our algorithm. Examples tested in previous chapters already include many problems that originated from engineering applications. Here, we are mainly interested in demonstrating the performance of Algorithm 6.2 without the nonmonotone line search (the latter is still in the process of implementation). Necessary procedures are explained in order to formulate the optimization problem corresponding to each design requirement. All these designs are carried out by CONSOL-OPTCAD. Simulations, *i.e.*, evaluation of specifications, are performed by MATLAB. Detailed and ready-to-use scripts are given in Appendix B.

7.1 Controller design using Q -parameterization

Consider first designing a controller $C(s)$ in the feedback system in Figure 7.1. $P(s)$

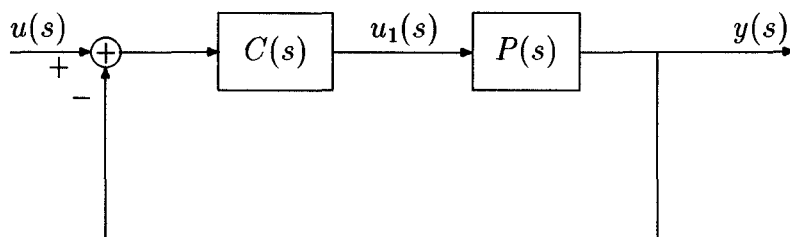


Figure 7.1: A Feedback System

is the model of a flexible robot arm taken from [47] and is described by the following

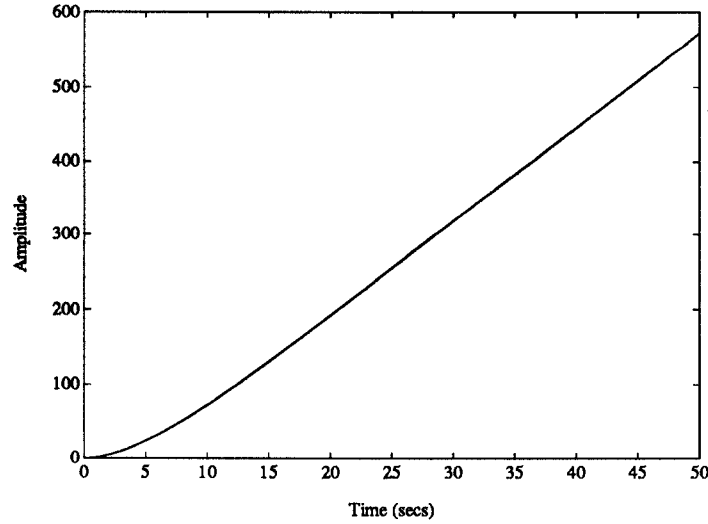


Figure 7.2: Unit Step Response of the Plant

transfer function

$$\begin{aligned}
 P(s) &= \frac{-4.906(s + 3.5687)(s - 8.4488)}{s(s + 0.20003)(s^2 + 0.35434s + 139.53)} \\
 &= \frac{k(s + z_1)(s + z_2)}{s(s + p_1)(s^2 + p_2s + p_3)}.
 \end{aligned} \tag{7.1}$$

This model is non-minimum phase and unstable. Its step response is given in Figure 7.2. Our goal is to obtain a controller $C(s)$ such that the closed loop system is stable and the unit step response lies within the envelop defined by the upper bound ($\bar{y}(t)$) and the lower bound ($\underline{y}(t)$) as shown in Figure 7.3. In particular, the steady-state error is required to be less than 1%. Clearly, the frequency domain input-output relation of the closed loop system in Figure 7.1 is given by

$$y(s) = P(s)C(s)(1 + P(s)C(s))^{-1}u(s) = P(s)Q(s)u(s)$$

with

$$Q(s) = C(s)(1 + P(s)C(s))^{-1}.$$

The Q -parameterization technique (see, *e.g.*, [90,91]) is used to first find a stabilizing controller. For this particular problem, it can be shown (see, *e.g.*, [47, Theorem 2.1]) that, if $Q(s)$ is proper and admits the following decompositions

$$Q(s) = s \cdot Q_1(s) \quad \& \quad 1 + P(s)Q(s) = s \cdot G(s) \tag{7.2}$$

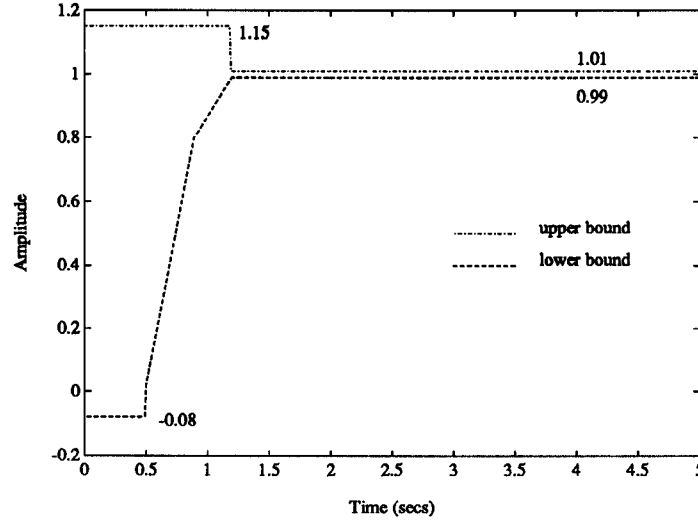


Figure 7.3: Desired Bound for Unit Step Response of Closed Loop System

where $Q_1(s)$ and $G(s)$ are both Hurwitz stable, $Q(s)$ stabilizes the closed loop system. In such case, there is a one-to-one correspondence between $C(s)$ and $Q(s)$, given by

$$Q(s) = C(s)(1 + P(s)C(s))^{-1} \quad \& \quad C(s) = Q(s)(1 - P(s)Q(s))^{-1},$$

and $C(s)$ is also proper and stabilizing. Thus, the design of $C(s)$ can be fulfilled by the design of $Q(s)$. Let $Q_1(s)$ be given by

$$Q_1(s) = \frac{b_0 s^4 + b_1 s^3 + b_2 s^2 + b_3 s + b_4}{(s^2 + a_1 s + a_2)(s^2 + a_3 s + a_4)(s + a_5)}.$$

It is easy to check that, if $a_i > 0$, $i = 1, \dots, 5$, and $b_4 = \frac{a_2 a_4 a_5 p_1 p_3}{k z_1 z_2}$, $Q(s)$ is proper and admits the decompositions given by (7.2), and thus stabilizes the closed loop system. The advantage of working with $Q(s)$ instead of $C(s)$ is apparent because the stability of the closed loop system can be easily maintained at every trial point in the optimization process. We can replace $a_i > 0$, $i = 1, 3, 5$, by $a_i \geq c_1$ for a prescribed $c_1 > 0$ to obtain better stability region. The next step is to choose proper values of all the parameters in $Q_1(s)$ so that the desired step response can be achieved. If we let $x = [a_1, \dots, a_5, b_0, \dots, b_3]^T$ be the vector of design parameters and let b_4 be replaced in terms of x , the corresponding optimization problem can be formulated as

$$\begin{aligned} \min_{x \in R^9} \quad & \max_{i=1,2} \max_{t \in [0,5]} \{f_i(x, t)\} \\ \text{s.t.} \quad & c_1 - x_i \leq 0, \quad i = 1, 3, 5 \\ & c_2 - x_i \leq 0, \quad i = 2, 4; \end{aligned}$$

where

$$f_1(x, t) = y(x, t) - \bar{y}(t), \quad f_2(x, t) = -y(x, t) + \underline{y}(t). \quad (7.3)$$

The interval $[0, 5]$ is discretized uniformly. Gradients are all approximated by finite differences. By choosing $c_1 = c_2 = 0.1$ and initial value $x_0 = [5, \dots, 5]^T$, the step response at x_0 is given in Figure 7.4. With $|\Omega^a| = 500$, after 32 iterations of Algorithm 6.2, the step response, shown in Figure 7.5, gets into the envelop defined in Figure 7.3. To look at the detail of the response in early time, we redraw this response in Figure 7.6. $Q(s)$ is given by

$$Q(s) = \frac{0.889819s^5 + 1.329009s^4 + 129.790109s^3 + 132.827781s^2 + 21.181594s}{(s^2 + 5.745373s + 11.063579)(s^2 + 4.806459s + 30.555433)(s + 0.797351)}$$

and the corresponding controller $C(s)$ is given by

$$C(s) = \frac{e_0s^7 + e_1s^6 + e_2s^5 + e_3s^4 + e_4s^3 + e_5s^2 + e_6s + e_7}{d_0s^7 + d_1s^6 + d_2s^5 + d_3s^4 + d_4s^3 + d_5s^2 + d_6s + d_7}$$

where

e_0	=	0.889809	d_0	=	1.000000
e_1	=	1.822293	d_1	=	11.903552
e_2	=	254.745088	d_2	=	227.905350
e_3	=	415.145148	d_3	=	1946.287628
e_4	=	18250.723496	d_4	=	12155.736558
e_5	=	22177.083377	d_5	=	42618.438207
e_6	=	6664.217132	d_6	=	34810.843979
e_7	=	591.182217	d_7	=	4990.074412.

The order of this controller is very high. The Control System Toolbox in MATLAB was used to perform a model reduction and to obtain a third order controller whose transfer function is given by

$$C_r^0(s) = \frac{0.889810s^3 + 0.534202s^2 + 131.293781s^1 + 25.898322}{s^3 + 10.881180s^2 + 74.410016s^1 + 235.805772}.$$

The step response of the closed loop system with the reduced order controller is given in Figure 7.8. There are slight violations of the envelope boundary. Running one iteration of optimization gives the controller transfer function

$$C_r^1(s) = \frac{1.088396s^3 + 0.539987s^2 + 131.283576s^1 + 25.896367}{0.939428s^3 + 10.901813s^2 + 74.419146s^1 + 235.807238}.$$

The corresponding step response is given in Figure 7.9.

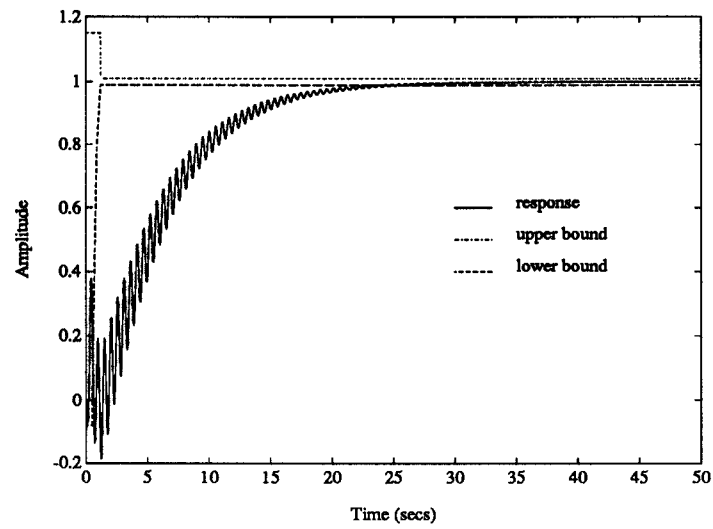


Figure 7.4: Initial Response of Closed Loop System

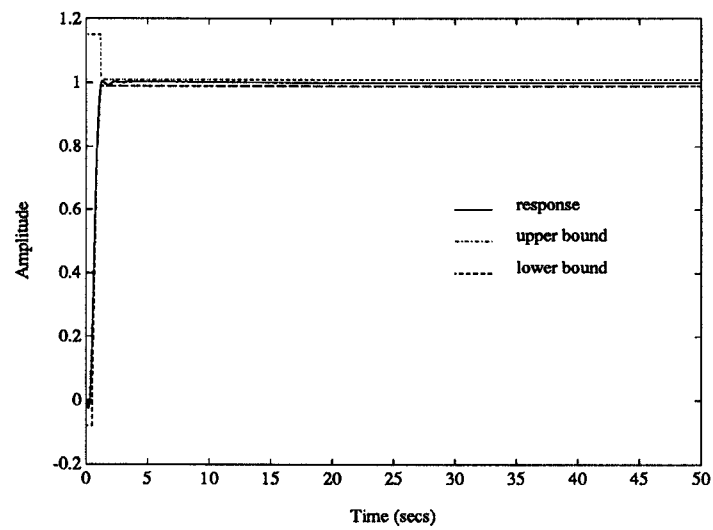


Figure 7.5: Final Response via Algorithm 6.2 (a)

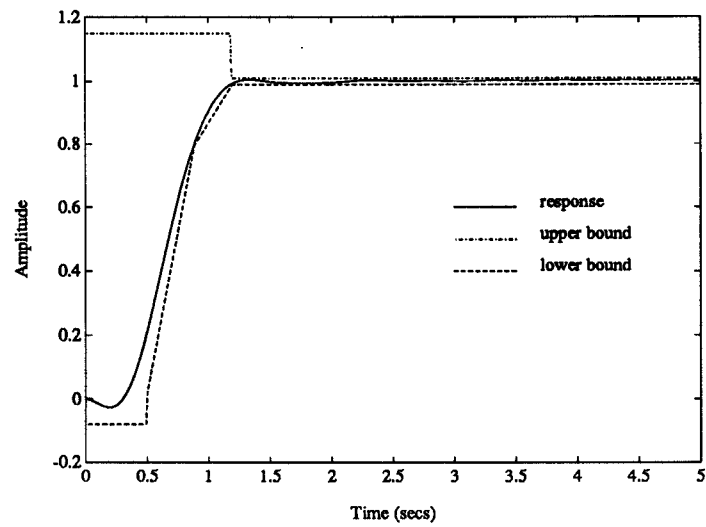


Figure 7.6: Final Response via Algorithm 6.2 (b)

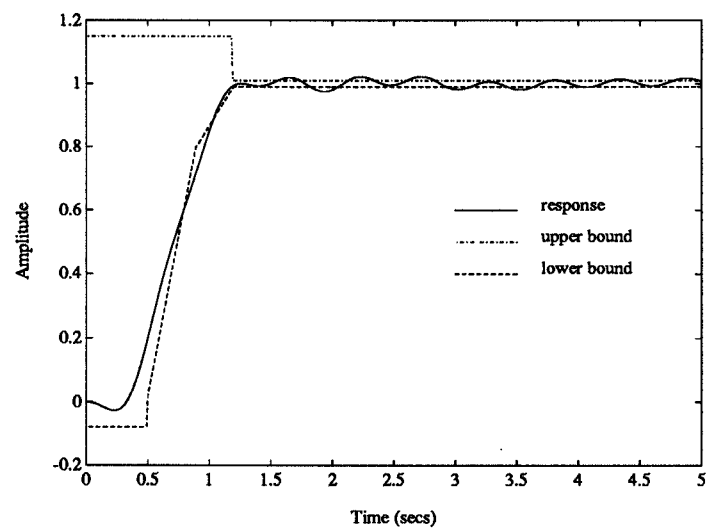
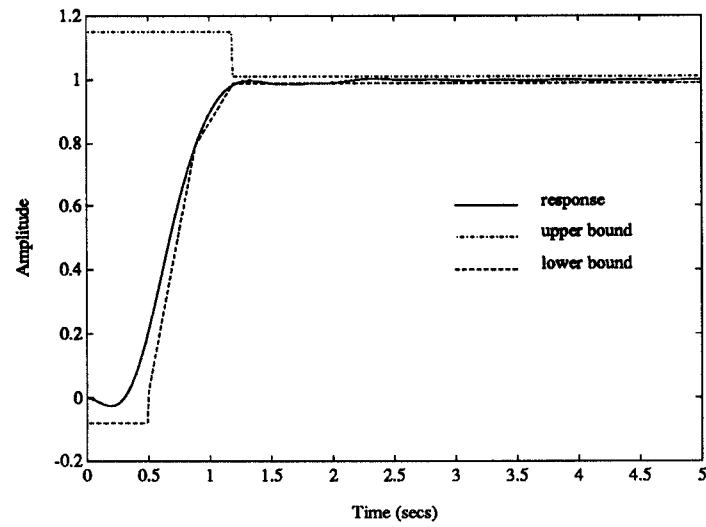
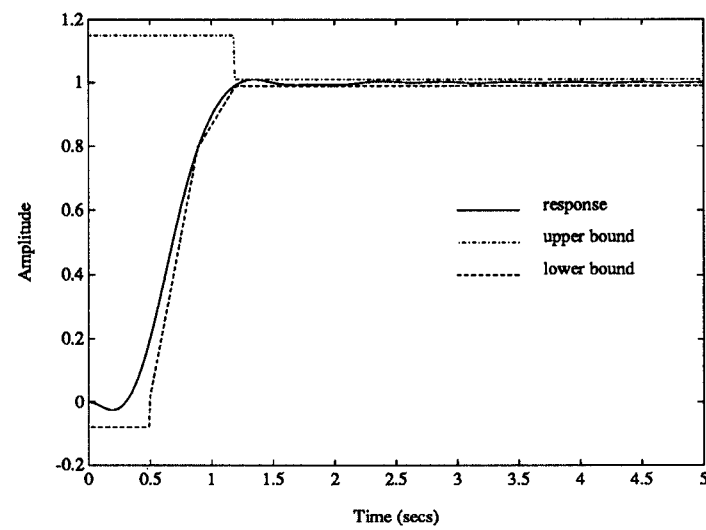


Figure 7.7: Final Response via a First Order Algorithm

**Figure 7.8:** Step Response with the Reduced Order Controller (a)**Figure 7.9:** Step Response with the Reduced Order Controller (b)

So far, we have not been concerned with the size of the control signal $u_1(t)$ to the plant. In practice, constraints usually have to be exercised since a plant may not be able to accept any large signal in magnitude. This is dealt with in the end of this section.

It is interesting to test the efficiency of Algorithm 6.2 when Ω^q contains different number of points and when compared with first order algorithms. An old version of CONSOL-OPTCAD is used to solve the same problem. The first order algorithm used in the CONSOL-OPTCAD was of essentially an ϵ -active type as mentioned in the introduction of this chapter. To make the comparison more meaningful, we terminate both algorithms whenever the response curve falls into the desired envelope ($f \leq 0$) if they do not reach another stationary point, or if the algorithms appear to take forever time to reach the envelope. With $|\Omega^q| = 500$, the step response obtained after 179 iterations of the first order algorithm (the optimization process found a local solution) is given in Figure 7.7. The comparison is summarized in Table 7.1.

CD	MESHP	TIME	NCMF	NAG	NACT	IT	NAV	OBJMAX	S-Err
ALGO6.2	50	140	271	612	13	65	4.2	-0.033	< 1%
	500	69	123	297	11	32	3.8	-0.025	< 1%
	1000	179	289	697	21	70	4.1	-0.022	< 1%
ALGO1st	50	514	933	1746	36	194	4.8	-0.001	< 1%
	500	463	922	1611	58	179	5.2	0.015	> 1%
	1000	456	844	1350	152	150	5.6	0.022	> 3%

Table 7.1: Comparison of Algorithm 6.2 and a First Order Algorithm (a)

In the table, results with different discretization are also reported. **ALGO6.2** stands for Algorithm 6.2, **ALGO1st** for the first order algorithm in the old CONSOL-OPTCAD, **MESHP** for the total mesh points of Ω^q of uniform discretization, **TIME** for the execution time in seconds, **NCMF** for the total number of simulations called upon during the line search, **NACT** for the number of points in $[0, 5]$ that is used for the computation of a search direction at the point of interest, **IT** stands for the number of iterations each algorithm is executed and **S-Err** for the steady-state error. Total number of simulations used to obtain gradients for each algorithm is reported under column **NAG**. **TPO** stands for the percentage of **NCMF** over $(\text{NCMF} + \text{NAG})$. Thus, $\text{TIME} \times \text{TPO}$ roughly reflects the time of optimization process with gradients available. **NAV** stands for the average number of simulations per iteration, excluding **NAG**.

Note that, while **NAV** of Algorithm 6.2 is not much smaller than that of the first

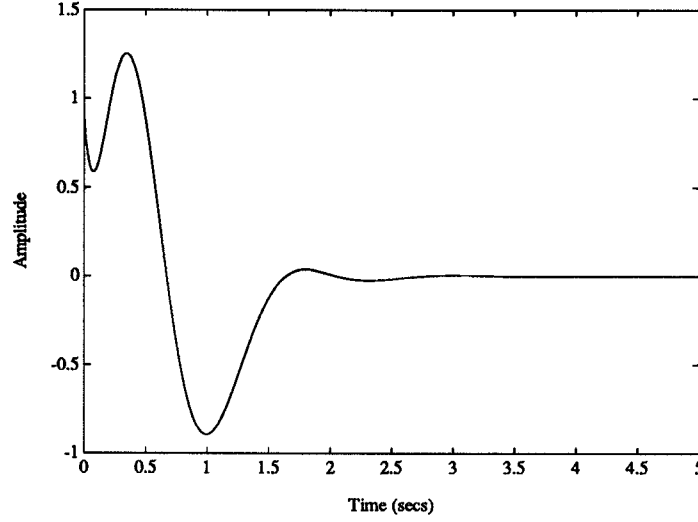


Figure 7.10: Unconstrained Control Signal to Plant

order algorithm, the former takes much smaller number of iterations and thus saves a large number of overall simulations. The dimension of Ω_k^f (NACT) is very small. For the first order algorithm, the resulting response is poor and the steady-state errors are large.

One of the important issues in control system design is the physical limit of the size of the actuator signals (see, *e.g.*, [8]). The actuator signal could be due to a command response or due to disturbances. In our case, if we are only interested in the command response, the limit can be easily imposed upon the magnitude of $u_1(x, t)$ in Figure 7.1 in the form of

$$|u_1(x, t)| \leq u_b.$$

The control signal $u_1(x, t)$ with $C_r^1(s)$ is given in Figure 7.10.

As an example, taking $|\Omega^q| = 500$, $u_b = 0.5$, and letting x be the variable vector consisting of the coefficients of numerator and denominator of a third order controller $C(s)$ with initial values equal to those of $C_r^1(s)$, the corresponding constrained optimization problem can now be formulated as

$$\begin{aligned} \min_x \quad & \max_{i=1,2} \max_{t \in [0,5]} \{f_i(x, t)\} \\ \text{s.t.} \quad & -0.5 \leq u_1(x, t) \leq 0.5, \quad \forall t \in [0, 5] \end{aligned}$$

where f_i 's are defined in (7.3). With a constrained control signal to the plant, the envelope boundary defined in Figure 7.3 is modified and given in Figure 7.11 to allow

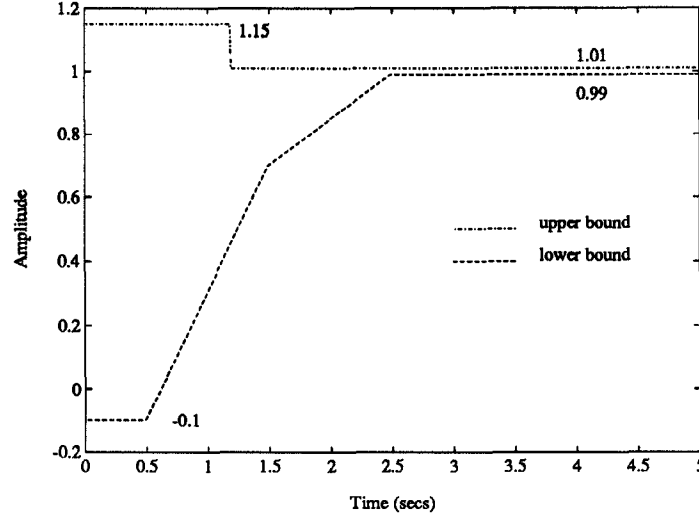


Figure 7.11: Desired Bound of the Step Response with Constrained Control

a slower step response. After 52 iterations of Algorithm 6.2, the step response is given in Figure 7.12 with the control signal $u_1(x_{52}, t)$ in Figure 7.13. The final controller is given by

$$C(s) = \frac{0.489062s^3 + 7.147021s^2 + 78.379718s^1 + 16.647966}{s^3 + 39.203683s^2 + 129.875744s^1 + 244.985352}.$$

7.2 Controller design using state feedback

Now, given the same plant $P(s)$ in (7.1), we design a state feedback control system so that the same step response shape is obtained, assuming that states are all available for simplicity. More precisely, we want to find the constant gain K in the feedback system in Figure 7.14. (A, B, C, D) is the state-space representation of the plant $P(s)$ in the form

$$\begin{aligned}\dot{x} &= Ax + Bu_1 \\ y &= Cx + Du_1.\end{aligned}$$

The optimization setup should be the same as in previous section, except that, in this case, the constant vector K becomes the design variable vector, *i.e.*, $x^T = K = [k_1, \dots, k_4]$. From the viewpoint of optimization, we could choose any initial

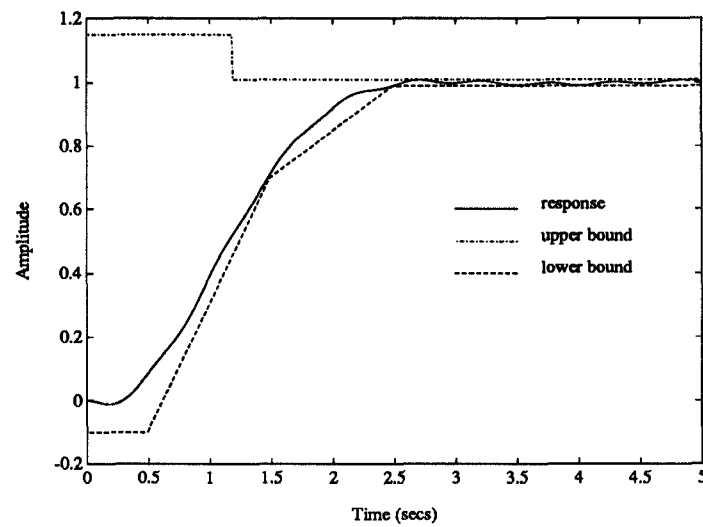


Figure 7.12: Step Response with Constrained Control Signal to Plant

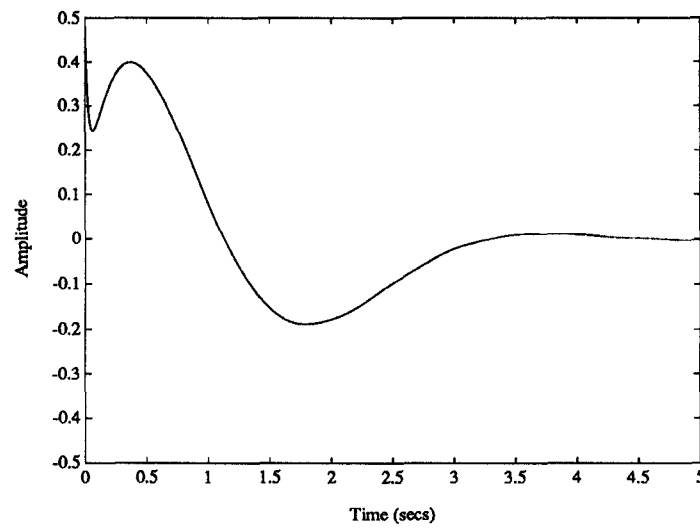
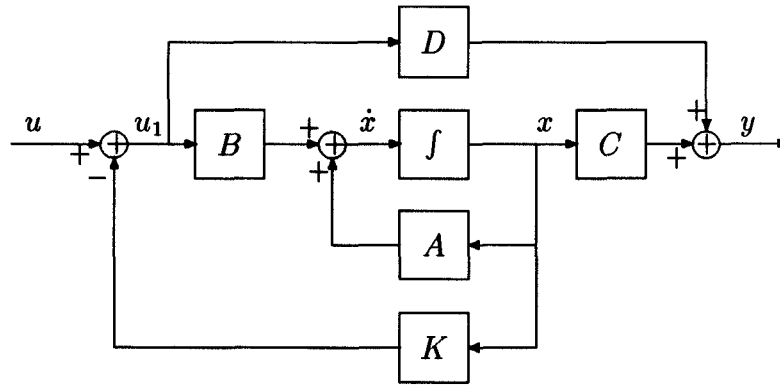


Figure 7.13: Constrained Control Signal to Plant

**Figure 7.14:** State Feedback System

point to start with. However, if we start with x_0 such that the closed loop system is unstable, unexpected numerical difficulty might appear in the simulation. Using the *LQG* technique, a stabilizing initial value $x_0 = [0.743, 0.188, 4.6306, 1.000]^T$ is found. At this point, the step response of the closed loop system is given in Figure 7.15. It is far away from our desired response. With $|\Omega^q| = 500$, after 27 iterations of Algorithm 6.2, we obtained $x_{27} = [9.944575, -68.006352, 211.537837, 356.357727]^T$ and the step response fits into the envelope defined in Figure 7.3. The results for different discretization points and the comparison with the first order algorithm are all summarized in Table 7.2. The final response looks similar to Figure 7.16 for all cases and for both algorithms. Stopping criterion is the same as for the first example. Observe that, while the iteration numbers are roughly of the same order in this

CD	MESHP	TIME	NCMF	NAG	TPO	NACT	IT	NAV	OBJMAX	S-Err
ALGO6.2	50	24	112	152	0.42	9	48	2.3	-0.002	< 1%
	500	21	74	124	0.37	12	31	2.4	-0.001	< 1%
	1000	40	105	184	0.53	13	46	2.3	-0.004	< 1%
ALGO1st	50	12	119	44	0.73	5	11	10.8	-0.005	< 1%
	500	56	303	72	0.81	28	18	16.8	-0.002	< 1%
	1000	69	195	44	0.56	91	11	17.7	1.245	< 1%

Table 7.2: Comparison of Algorithm 6.2 and a First Order Algorithm (b)

example, the NAV number (close to ideal) of Algorithm 6.2 is much smaller than that of the first order algorithm. The increase of size of the problem for finding a search direction is drastic for the first order algorithm, when discretization becomes finer.

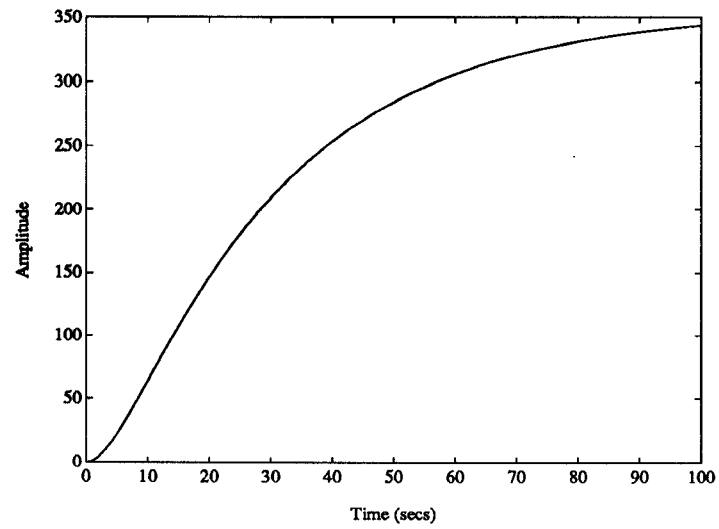


Figure 7.15: Initial Response of State Feedback System

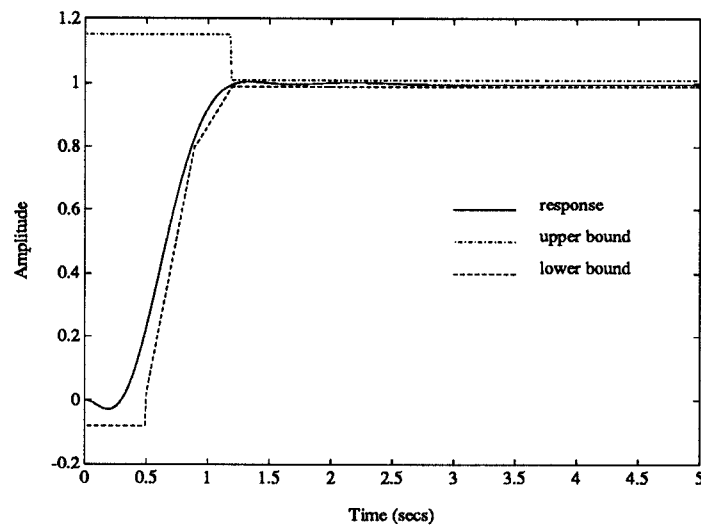


Figure 7.16: Final Response of State Feedback System

7.3 PID controller design

This example is taken from [69] and has been used, *e.g.*, in [29,30,54]. Given the system in Figure 7.1 with

$$P(s) = \frac{1}{(s+3)(s^2+2s+2)},$$

the requirement is to design a PID controller

$$C(s) = x_1 + \frac{x_2}{s} + x_3s,$$

with $x = [x_1, x_2, x_3]^T$ the design variable vector, so that the closed-loop system has sufficient stability margin and minimizes the error $e(x, t)$ in the mean-square sense in the zero-state response to a unit step input. Thus, the objective function is given by

$$\begin{aligned} f(x) &= \int_0^\infty e^2(x, t) dt = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} E(x, s) E(x, -s) ds \\ &= \frac{x_2(122 + 17x_1 + 6x_3 - 5x_2 + x_1x_3) + 180x_3 - 36x_1 + 1224}{x_2(408 + 56x_1 - 50x_2 + 60x_3 + 10x_1x_3 - 2x_1^2)}, \end{aligned}$$

using Parseval's theorem by assuming the closed loop system is stable. $E(x, s)$ is the Laplace transformation of $e(x, t)$. The stability margin requirement is formulated, in terms of return difference $T(s) = 1 + P(s)C(s)$, as an inequality constraint

$$g(x, \omega) = \text{Im}T(j\omega) - 3.33(\text{Re}T(j\omega))^2 + 1.0 \leq 0.$$

This constraint corresponds to the modified Nyquist plot in Figure 7.17. Simple bound constraints are placed on x with

$$0 \leq x_1 \leq 100, \quad 0.1 \leq x_2 \leq 100, \quad 0 \leq x_3 \leq 100.$$

With $|\Omega^q| = 300$, after 18 iterations of Algorithm 6.2, the optimal parameters are given by $x_{18} = [16.836310, 40.957968, 34.597778]^T$. The bode plot of the open loop system is given in Figure 7.18 and the step response of the closed loop system is given in Figure 7.19. Other results are summarized in Table 7.3. Since we know the optimal value of the objective function is about 1.75, both algorithms are stopped when the actual objective value becomes smaller than this value. **NCFG** in the table stands for the number of evaluations of objective and constraint, as the values of both objective and constraint are obtained simultaneously in CONSOL-OPTCAD. We observe again that the **NAV** number of Algorithm 6.2 is much smaller. The **NACT** number of the first order algorithm is all zero, that is in fact the reason that causes many simulations in a line search since the search direction does not take into account any points in Ω^q , thus a bad search direction.

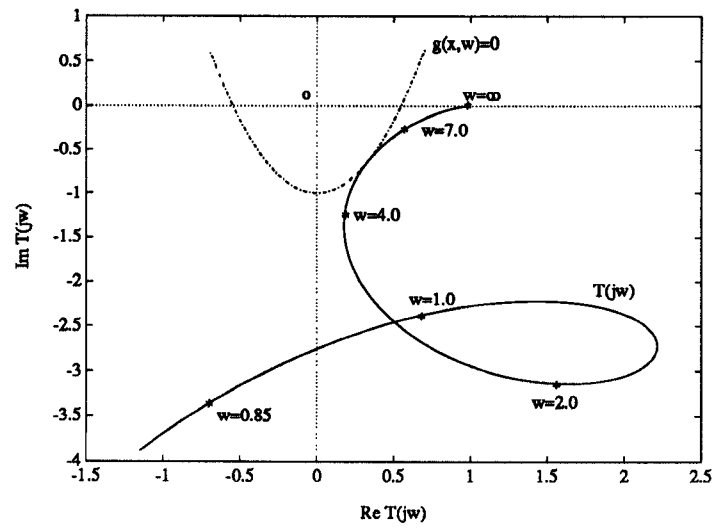


Figure 7.17: Optimal Nyquist Plot of Return Difference

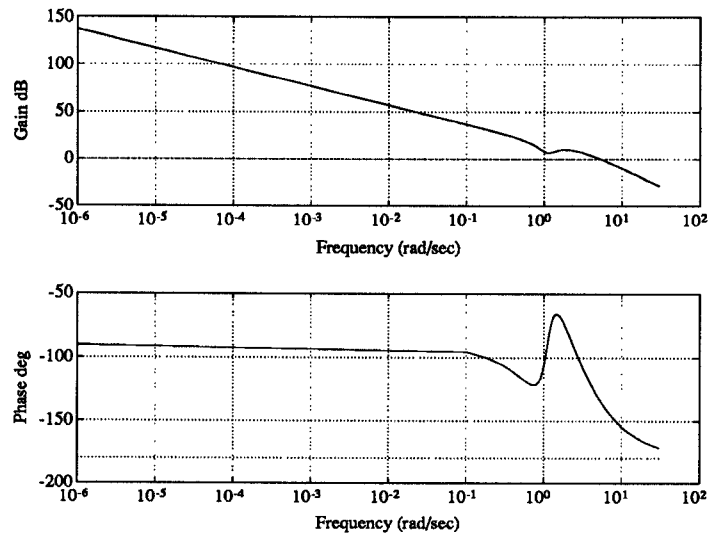


Figure 7.18: Bode Plot of the Open Loop System

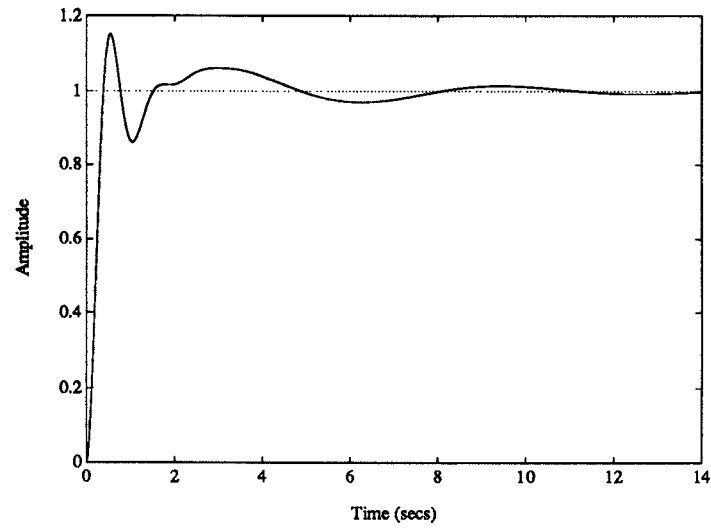


Figure 7.19: Final Response of PID Controlled Closed Loop System

CD	MESHP	TIME	NCFG	NAG	TPO	NACT	IT	NAV	OBJECTIVE
ALGO6.2	60	8	62	63	0.50	5	21	3.0	0.17459
	300	10	59	63	0.48	6	21	2.8	0.17459
	1000	14	62	63	0.50	12	21	3.0	0.17459
ALGO1st	60	10	108	42	0.72	0	14	7.7	0.17444
	300	13	112	42	0.73	0	14	8.0	0.17444
	1000	21	121	45	0.73	0	15	8.1	0.17346

Table 7.3: Comparison of Algorithm 6.2 and a First Order Algorithm (c)

Chapter 8

Concluding Remarks

A number of optimization algorithms have been developed in this dissertation. They are all justified by rigorous theoretic analysis and measured up by extensive numerical tests. They have found successful applications in engineering designs such as predictive control using neural net dynamic models [21], batch process optimal profile determination [89] and optimal capacitor design [83]. We hope that these algorithms, embedded in advanced optimization-based systems design tools such as CONSOL-OPTCAD, will continue to play significant roles in the computer-aided design in various engineering branches.

New computer technologies have made, and will continue to make, it possible for many advanced mathematical approaches to be transplanted into engineering application. Yet, without careful account of special requirements from many different engineering fields, we will easily run into limitations. Our research on engineering application-oriented optimization techniques shows that engineering designs do stimulate challenging mathematical problems that existing mathematical methodologies, however advanced and powerful they might be, may not be adequate, at least in their original framework, for achieving many distinctive features that are unique in engineering designs. Undoubtedly, this will remain true for many years to come. We have made some progress in this context and we will continue to meet the challenge from the real world.

There are many issues to be resolved, both theoretically and experimentally. We end this dissertation by indicating a few research topics, still along the line of the applications of optimization techniques in engineering designs.

1. One of the major tasks in the near future is to include nonlinear equality con-

straints in all algorithms developed in this dissertations. While we argued that equality constraints can be eliminated in generally in engineering applications, provisions that take them into account will usually facilitate the setup of engineering design problems in a suitable form for optimization. The challenge is to maintain all desirable properties, such as feasibility of inequality constraints,

2. From Table 7.1, for instance, we can see that almost 70% of simulations are called up for obtaining approximations to gradients by finite differences. Each simulation may mean minutes or hours in the context of engineering applications. This would exhibit prohibitive limitations on the application of advanced optimization-based design tools to the solution of the real world problem, especially when there are a large number of design parameters. Application of automatic differentiation technique (see, *e.g.*, [41]) should be exploited in this context.
3. Currently, the discrete sets Ω_k^f and Ω_k^g are determined automatically by the computer. In the context of interactive computer-aided design, it is desirable to incorporate the designer's knowledge to improve the construction of these discrete sets, as the designer may have good knowledge of some critical points at optimal solution. This provision would help to accumulate the critical points especially during early stage of the optimization process. CONSOL-OPTCAD should provide such provision in the near future, probably with the aid of visualization of specifications in question.
4. In the context of finely discretized optimization problems, usually a single simulation returns values at each discrete point of the corresponding specifications. Assuming simulation is so expensive that any computation on the optimization part is negligible, the computation of gradients at all discrete points may be as cheap as the computation at a small number of points. Definitely, the use of gradients at all gradients usually will yield a better search direction, especially in early iterations of each run of the algorithm. One interesting experiment is to transpose the idea of adaptively constructing the set of critical points into the quadratic programming. In other words, gradients at all discrete points are used in setting up the quadratic program that defines the search direction. Then, the quadratic program, possibly of huge number of constraints, is solved by solving a sequence of quadratic problem of smaller size, each being constructed using the same idea as Ω_k^f was constructed. The final search direction would be more

adequate for a line search, which in turn may imply fewer simulations to call to meet a line search criterion.

Appendix A

Some Proofs

This appendix provides some proofs of properties presented in previous chapters.

A.1 Proofs for Chapter 3

Proof of Lemma 3.1. Because of assumption A3.1 and the boundedness of d_k^0 , the first order expansion of f_i around x_k along direction d_k^0 yields, for $t > 0$,

$$f_i(x_k + td_k^0) = f_i(x_k) + t\langle \nabla f_i(x_k), d_k^0 \rangle + o(t). \quad (\text{A.1})$$

In view of (3.12),

$$\begin{aligned} \langle \nabla f_i(x_k), d_k^0 \rangle &\leq \max_i \{f_i(x_k) + \langle \nabla f_i(x_k), d_k^0 \rangle\} - f_i(x_k) \\ &= \sum_i \mu_{k,i} f_i(x_k) + \sum_i \mu_{k,i} \langle \nabla f_i(x_k), d_k^0 \rangle - f_i(x_k) \\ &\leq f(x_k) - f_i(x_k) + \langle d_k^0, \nabla_x L(x_k, \mu_k) \rangle \\ &= f(x_k) - f_i(x_k) - \langle d_k^0, H_k d_k^0 \rangle. \end{aligned}$$

Without loss of generality, we assume $t \leq 1$. Therefore, (A.1) becomes

$$f_i(x_k + td_k^0) \leq f(x_k) - t\langle d_k^0, H_k d_k^0 \rangle + o(t), \quad \forall i = 1, \dots, p.$$

Taking a max over i on the left hand side of above inequality and taking $t \rightarrow 0^+$, the first claim is proved. Comparing (3.11) and (3.12) and in view of assumption A3.3, the second claim results easily. $\square$

Proof of Lemma 3.7. In view of Proposition 3.5(ii), d_k^0 that solves $QP_1(x_k, H_k)$ satisfies, for k large enough, the following set of linear equations (since $I(x^*) = \{1, \dots, r\}$)

$$f_i(x_k) + \langle \nabla f_i(x_k), d_k^0 \rangle = f_1(x_k) + \langle \nabla f_1(x_k), d_k^0 \rangle, \quad i = 2, \dots, r$$

which implies

$$R_k^T d_k^0 = \bar{f}_1(x_k).$$

Since, from the definition of P_k in A3.6, we have

$$P_k d_k^0 = d_k^0 - R_k(R_k^T R_k)^{-1} R_k^T d_k^0,$$

(3.27) follows. Since, from assumptions A3.1, A3.2 and A3.6, R_k and $(R_k^T R_k)^{-1}$ are bounded for large k , in view of Remark 3.2, (3.28) follows directly from the definition of d_k' .

Now for any $j_k \in I(x^*)$ such that $f_{j_k}(x_k) = f(x_k)$, $\bar{f}_{j_k}(x_k) \leq 0$. Also Proposition 3.5(ii) implies there exists $\underline{\mu} > 0$ such that, for k large enough,

$$\min_{j \in I(x^*)} \mu_{k,j} \geq \underline{\mu}.$$

Therefore, in view of (3.28), we have

$$\begin{aligned} \langle \bar{f}_{j_k}(x_k), \bar{\mu}_k \rangle &\leq -\underline{\mu} \|\bar{f}_{j_k}(x_k)\| \\ &\leq -\underline{\mu} c_1 \|d_k'\| \end{aligned}$$

and (3.29) follows. $\square$

Proof of Lemma 3.9. Given $x \in \mathbb{R}^n$, define the augmented Lagrange function

$$L^+(x, \mu) = L(x, \mu) + r \sum_{i \in I(x^*)} (f(x) - f_i(x))^2$$

for a constant $r > 0$. Similarly to the argument for Theorem 1 in [26], assumption A3.5 implies that there exists an r such that, for k large enough,

$$\nabla_{xx}^2 L^+(x^*, \mu^*) > 0.$$

Therefore, it can be shown that there exists $c_3 > 0$ such that, for k large enough,

$$L^+(x, \mu^*) \geq L(x^*, \mu^*) + c_3 \|x - x^*\|^2 = f(x^*) + c_3 \|x - x^*\|^2. \quad (\text{A.2})$$

In addition, for k large enough,

$$\begin{aligned} f(x) - L^+(x, \mu^*) &= f(x) - \sum_{i=1}^p \mu_i^* \{f_i(x) - r \sum_{i \in I(x^*)} (f(x) - f_i(x))^2\} \\ &= \sum_{i=1}^p \{\mu_i^* - r(f(x) - f_i(x))\} \{f(x) - f_i(x)\}. \end{aligned}$$

Thus, $f(x) - L^+(x, \mu^*) \geq 0$ in view of Lemma 3.3 and the claim follows in view of (A.2). $\square$

A.2 Proofs for Chapter 4

Proof of Lemma 4.1. The argument is essentially the same as that used in the proof of Lemma 3.2, by taking into account the facts that (i) d_k^1 is bounded in view of the continuity of $d^1(\cdot)$ and x_k is in a compact set; and (ii), since ς_k is bounded by $\|d_k^0\|$, in view of the constraints in $QP_2(x_k, H_k)$, of the properties of $d^1(\cdot)$ and of the definition of ρ_k^ℓ and ρ_k^g it follows that ρ_k^ℓ and ρ_k^g converge to zero whenever d_k^0 does. Therefore, $t_k d_k^\ell$ and $t_k d_k^g$ converge to zero and, since whenever it is defined $\tilde{d}_k$ satisfies $\|\tilde{d}_k\| \leq \|d_k^0\|$, we obtain $\|x_{k+1} - x_k\| \rightarrow 0$. $\square$

Proof of Theorem 4.2. Let x^* be the accumulation point such that $\{x_k\}$ converges to x^* on K . In view of A3.3, there exists $K' \subset K$ such that $\{H_k\}$ converges on K' to some symmetric positive definite matrix H^* . In that case, in view of the work in [18] (see also [33, Lemma 3.2]), the sequence $\{d_k^0\}$ converges to a vector d^{0*} on K' . In order to conclude, we show that $d^{0*} = 0$, so that, the feasible point x^* (limit of feasible iterates) is a GKKT point for (P) . We now suppose that $d^{0*} \neq 0$ so that $\exists \underline{d} > 0$ s.t. $\|d_k^0\| \geq \underline{d}$, $k \in K'$, and we show that, in that case, there exists $\underline{t} > 0$ such that, for all $k \in K'$ at which a stepsize is computed in *Step 1 vii*, $t_k \geq \underline{t}$, contradicting Lemma 4.1. In view of A3.3,

$$f'(x_k, d_k^0) \leq -\sigma_1 \underline{d}^2$$

and, from the definition of d_k^g in *Step 1 v*,

$$f'(x_k, d_k^g) \leq -\theta \sigma_1 \underline{d}^2. \quad (\text{A.3})$$

Also, in view of the continuity of $d^1(\cdot)$ and (4.8) and the definition of d_k^g in *Step 1 v*, there exists $\underline{\rho} > 0$ such that, for $k \in K'$ large enough it holds

$$g_j(x_k) + \langle \nabla g_j(x_k), d_k^g \rangle \leq -\underline{\rho}. \quad (\text{A.4})$$

Showing that, on K' , t_k is bounded away from zero by a positive number can be done similarly to the proof of Proposition 3.2 in [61] using (A.3), (A.4) and the fact that, in view of the continuity of $d^1(\cdot)$ and the definition of d_k^g and $\tilde{d}_k$ in *Step 1 v* and *vi*, $\|d_k^g\|$ and $\|\tilde{d}_k\|$ are bounded on K . $\square$

Proof of Lemma 4.3. The proof of the convergence of $\{x_k\}$ is similar to the one of Proposition 3.4 in [63]. In view of (i), the properties of $d^1(\cdot)$ and the definition of d_k^ℓ in *Step 1 iii*, $\exists M > 0$ such that

$$\|d_k^\ell\| \leq M \|d_k^0\| \quad \forall k. \quad (\text{A.5})$$

Let

$$N = \frac{1}{2} \max_{j \in \{1, \dots, m\}, t \in (0,1), k \in \mathbb{N}} \left\{ \left\| \frac{\partial^2 g_j}{\partial x^2}(x_k + td_k^\ell) \right\| \right\}$$

and let $C = MN$. In view of the regularity of the functions g_j 's (assumption A4.1') and the boundedness of $\{x_k\}$ and $\{d_k^\ell\}$, this quantity is well defined. For any $j \in \{1, \dots, m\}$, we have

$$g_j(x_k + d_k^\ell) \leq g_j(x_k) + \langle \nabla g_j(x_k), d_k^\ell \rangle + N \|d_k^\ell\|^2.$$

In view of the definition of d_k^ℓ in *Step 1 iii*, the properties of $d^1(\cdot)$ and the fact that $\{d_k^0\}$ converges to zero, for k large enough we get

$$g_j(x_k + d_k^\ell) \leq -\min\{C, C_k\} \|d_k^0\|^2 + N \|d_k^\ell\|^2.$$

Therefore, from (A.5)

$$g_j(x_k + d_k^\ell) \leq -\min\{C, C_k\} \|d_k^0\|^2 + MN \|d_k^0\|^2.$$

This completes the proof since, if C_k keeps increasing, for k big enough $C_k \geq C$ and, from the definition of C we obtain

$$g_j(x_k + d_k^\ell) \leq 0$$

and, in view of *Step 2* in the algorithm, C_k would remain constant, a contradiction. $\square$

Proof of Proposition 4.4. The proof of the first two statements is very similar to that of Proposition 4.2 in [61]. (iii) follows from Theorem 4.2, Lemma 4.3, the definition of d_k^ℓ and d_k^g in *Step 1 iii* and v and the properties of $d^1(\cdot)$. The first relation of (iv) can be proven similarly to what is done in the proof of Proposition 3.6 [63]. Making use of (i)—(iii) and Lemma 3.7, it can be shown that

$$f(x_k + d_k^g + \tilde{d}_k) \leq \max_{\ell=0,1,2} \{f(x_{k-\ell})\} + \alpha \langle \nabla f(x_k), d_k^g \rangle.$$

Finally, expanding $g_j(x_k + d_k^g + \tilde{d}_k)$ about $x_k + d_k^g$, using the properties of $\tilde{d}_k$ and the fact that $\gamma \in (2, 3)$, it can be shown that, for k large enough,

$$g_j(x_k + d_k^g + \tilde{d}_k) \leq 0, \quad j = 1, \dots, m.$$

Thus, step of one is accepted. $\square$

Next, we provide the proof for Theorem 4.5. It is a modification of that for Theorem 2 in [77]. We first establish some auxiliary properties.

In view of Proposition 4.4(i), it suffices to consider all active objective functions and active constraints for k large enough, *i.e.*, it is equivalent to considering the problem (since $I(x^*) = \{1, \dots, r\}$; see comment in front of assumption A3.6)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f_1(x) \\ \text{s.t.} \quad & f_i(x) = f_1(x) \quad \forall i \in I(x^*) \setminus \{1\} \\ & g_j(x) = 0 \quad \forall j \in J(x^*), \end{aligned}$$

where f_1 could be replaced by f_ℓ for any $\ell \in I(x^*)$ in view of Proposition 4.4(i) and Remark 3.2. Furthermore, (x^*, μ^*, λ^*) solves the system in (x, μ, λ)

$$\begin{cases} \sum_{i \in I(x^*)} \mu_i \nabla f_i(x) + \sum_{j \in J(x^*)} \lambda_j \nabla g_j(x) = 0 \\ f_i(x) = f_1(x) & i \in I(x^*) \\ g_j(x) = 0 & j \in J(x^*). \end{cases} \quad (\text{A.6})$$

Define the Jacobian matrix of (A.6) by

$$J_{oc}(x, \mu, \lambda) = \begin{pmatrix} \nabla_{xx}^2 L(x, \mu, \lambda) & R(x) \\ R^T(x) & 0 \end{pmatrix} \quad (\text{A.7})$$

and its variant

$$J_{oc}^*(x) = \begin{pmatrix} \nabla_{xx}^2 L(x^*, \mu^*, \lambda^*) & R(x) \\ R^T(x) & 0 \end{pmatrix} = \begin{pmatrix} G^* & R(x) \\ R^T(x) & 0 \end{pmatrix}, \quad (\text{A.8})$$

where $R(x)$ was defined in assumption A4.4. Now, we can apply Newton's method iteratively to system (A.6) and let $(\underline{d}_k, \Delta \underline{\mu}_k, \Delta \underline{\lambda}_k)$ solves the corresponding system linearized at some point $(x_k, \underline{\mu}_k, \underline{\lambda}_k)$

$$J_{oc}(x_k, \underline{\mu}_k, \underline{\lambda}_k) \begin{pmatrix} \underline{d}_k \\ \Delta \underline{\mu}_k \\ \Delta \underline{\lambda}_k \end{pmatrix} = \begin{pmatrix} -\nabla_x L(x_k, \underline{\mu}_k, \underline{\lambda}_k) \\ -\bar{f}_1(x_k) \\ -g(x_k) \end{pmatrix}, \quad (\text{A.9})$$

where $\bar{f}_1$ was defined immediately before assumption A3.6 and $g(x) = [g_j(x) : j \in J(x^*)]^T$. If we define $\underline{\psi}_k = (\underline{\mu}_k^T, \underline{\lambda}_k^T)^T$, $\Delta \underline{\psi}_k = (\Delta \underline{\mu}_k^T, \Delta \underline{\lambda}_k^T)^T$ and $\bar{g}(x) = [\bar{f}_1^T(x), g^T(x)]^T$, the above system can be written as

$$J_{oc}(x_k, \underline{\psi}_k) \begin{pmatrix} \underline{d}_k \\ \Delta \underline{\psi}_k \end{pmatrix} = \begin{pmatrix} -\nabla f_1(x_k) - R_k \underline{\psi}_k \\ -\bar{g}(x_k) \end{pmatrix}$$

which is equivalent to

$$J_{oc}(x_k, \underline{\psi}_k) \begin{pmatrix} \underline{d}_k \\ \underline{\psi}_k + \Delta \underline{\psi}_k \end{pmatrix} = \begin{pmatrix} -\nabla f_1(x_k) \\ -\bar{g}(x_k) \end{pmatrix}. \quad (\text{A.10})$$

Since $J_{oc}^*(x^*) = J_{oc}(x^*, \mu^*, \lambda^*)$ is nonsingular in view of assumption A4.4, $J_{oc}(x_k, \underline{\psi}_k)$ and $J_{oc}^*(x_k)$ are both nonsingular when $(x_k, \underline{\psi}_k)$ is close enough to (x^*, ψ^*) with $\psi = (\mu^T, \lambda^T)^T$. Thus, the solution of the following system

$$J_{oc}^*(x_k) \begin{pmatrix} \underline{d}_k \\ \underline{\psi}_k + \Delta \underline{\psi}_k \end{pmatrix} = \begin{pmatrix} -\nabla f_1(x_k) \\ -\bar{g}(x_k) \end{pmatrix} \quad (\text{A.11})$$

is also well defined. Clearly, $\underline{d}_k$ is independent of $\underline{\psi}_k$.

Lemma A.1. Let $(\underline{d}_k, \underline{\psi}_k + \Delta \underline{\psi}_k)$ be the solution of system (A.11). Suppose $\{(x_k, \underline{\psi}_k)\}$ converges to (x^*, ψ^*) . Then,

$$\lim_{k \rightarrow \infty} \frac{\|x_k + \underline{d}_k - x^*\|}{\|x_k - x^*\|} = 0. \quad (\text{A.12})$$

Proof. It is easy to show by standard argument for Newton's method that $\{(x_k, \underline{\psi}_k)\}$ converges superlinearly, i.e.,

$$\lim_{k \rightarrow \infty} \frac{\|x_k + \underline{d}_k - x^*\| + \|\underline{\psi}_k + \Delta \underline{\psi}_k - \psi^*\|}{\|x_k - x^*\| + \|\underline{\psi}_k - \psi^*\|} = 0. \quad (\text{A.13})$$

However, from (A.11), $\underline{d}_k$ is independent of $\underline{\psi}_k$. This implies we can take $\underline{\psi}_k = \psi^*$ in (A.11) and thus in (A.13). Therefore, the claim follows. $\square$

Lemma A.2. Let $\{x_k\}$ be the sequence generated by Algorithm 4.1. Then,

$$\|d_k^0\| = O(\|x_k - x^*\|) \quad (\text{A.14})$$

and

$$\|x_k - x^*\| = O(\|x_{k-1} - x^*\|). \quad (\text{A.15})$$

Proof. From the optimality conditions (4.13) and Proposition 4.4(i) and (ii), $(d_k^0, \psi_k) = (d_k^0, \mu_k, \lambda_k)$ satisfies

$$\begin{pmatrix} H_k & R_k \\ R_k^T & 0 \end{pmatrix} \begin{pmatrix} d_k^0 \\ \psi_k \end{pmatrix} = \begin{pmatrix} -\nabla f_1(x_k) \\ -\bar{g}(x_k) \end{pmatrix}. \quad (\text{A.16})$$

In Lemma A.1, we can take the same x_k and, since $\underline{d}_k$ is independent of $\underline{\psi}_k$, we can also take $\underline{\psi}_k = \psi_k$. Thus, assumptions of Lemma A.1 are satisfied because $\{x_k\}$ and $\{\psi_k\}$ both converge. Subtracting (A.11) from (A.16) yields

$$J_{oc}^*(x_k) \begin{pmatrix} \underline{d}_k - d_k^0 \\ \Delta \underline{\psi}_k \end{pmatrix} = \begin{pmatrix} (H_k - G^*)d_k^0 \\ 0 \end{pmatrix}. \quad (\text{A.17})$$

Adding to and subtracting from the first term of the right hand side of (A.17) by a term $R_k(R_k^T R_k)^{-1} R_k^T d_k^0$ and re-arranging the system give

$$J_{oc}^*(x_k) \begin{pmatrix} \underline{d}_k - d_k^0 \\ \underline{\psi}_k + \Delta \underline{\psi}_k - \psi_k - (R_k^T R_k)^{-1} R_k^T d_k^0 \end{pmatrix} = \begin{pmatrix} P_k(H_k - G^*)d_k^0 \\ 0 \end{pmatrix}. \quad (\text{A.18})$$

It follows that

$$\|\underline{d}_k - d_k^0\| = O(\|P_k(H_k - G^*)d_k^0\|) \quad (\text{A.19})$$

in view of the nonsingularity of $J_{oc}^*(x_k)$ for k large enough. On the other hand, (4.14) implies

$$\begin{aligned} \|P_k(H_k - G^*)d_k^0\| &\leq \|P_k(B_k - G^*)R_k(R_k^T R_k)^{-1} R_k^T d_k^0\| + o(\|d_k^0\|) \\ &= O(\|\bar{g}(x_k)\|) + o(\|d_k^0\|) \end{aligned} \quad (\text{A.20})$$

$$= O(\|x_k - x^*\|) + o(\|d_k^0\|) \quad (\text{A.21})$$

where (A.20) follows because, in view of optimality conditions (4.13),

$$\bar{g}(x_k) + R_k^T d_k^0 = 0 \quad (\text{A.22})$$

and (A.21) follows from the fact that

$$\bar{g}(x^*) = 0. \quad (\text{A.23})$$

From (A.19) and (A.21), we have

$$\begin{aligned} \|d_k^0\| - \|\underline{d}_k\| &\leq \|\underline{d}_k - d_k^0\| = O(\|P_k(H_k - G^*)d_k^0\|) \\ &= O(\|x_k - x^*\|) + o(\|d_k^0\|) \end{aligned}$$

and, from (A.12), we have

$$\|\underline{d}_k\| = O(\|x_k - x^*\|). \quad (\text{A.24})$$

Thus,

$$\|d_k^0\| = O(\|x_k - x^*\|) + o(\|d_k^0\|)$$

which implies (A.14).

Now, let d_k be either d_k^ℓ or $d_k^g + \tilde{d}_k$. In view of Proposition 4.4(iii) and (iv),

$$\begin{aligned} \|x_k - x^*\| &= \|x_{k-1} - x^* + d_{k-1}\| \\ &\leq \|x_{k-1} - x^*\| + \|d_{k-1}\| \\ &= \|x_{k-1} - x^*\| + O(\|d_{k-1}^0\|) + O(\|d_{k-1}^0\|^2). \end{aligned}$$

(A.15) then follows from (A.14) and the fact that $\{d_k^0\}$ converges to zero. $\square$

Proof of Theorem 4.5. Let $d_k = d_k^\ell$ or $d_k = d_k^g + \tilde{d}_k$. In view of (A.22) at $k-1$ and Proposition 4.4,

$$\bar{g}(x_k) = \bar{g}(x_{k-1} + d_{k-1}) = O(\|d_{k-1}^0\|^2). \quad (\text{A.25})$$

And thus, in view of (A.19), (A.20) and (A.25), it follows that

$$\begin{aligned} \|\underline{d}_k - d_k\| &= \|\underline{d}_k - d_k^0\| + O(\|d_k^0\|^2) \\ &= O(\|\bar{g}(x_k)\|) + o(\|d_k^0\|) \\ &= O(\|d_{k-1}^0\|^2) + o(\|d_k^0\|) \\ &= o(\|d_{k-1}^0\|) + o(\|d_k^0\|). \end{aligned}$$

Therefore, in view of (A.12) and Lemma A.2,

$$\begin{aligned} \|x_k + d_k - x^*\| &\leq \|x_k + \underline{d}_k - x^*\| + \|\underline{d}_k - d_k\| \\ &= o(\|x_k - x^*\|) + o(\|d_{k-1}^0\|) + o(\|d_k^0\|) \\ &= o(\|x_{k-1} - x^*\|). \end{aligned}$$

The two-step superlinear convergence follows. $\square$

To prove Theorem 4.6, we need the following result which is a simple extension of Lemma 3.8 in [12].

Lemma A.3. There exists $c > 0$ such that, for all x close to x^* ,

$$f(x) - f(x^*) \geq c\|x_k - x^*\|^2.$$

Proof of Theorem 4.6. It is a combination of the proof for Theorem 3.10 in Chapter 3 and that for Theorem 3.8 in [6]. We first show that, for k large enough,

$$f_i(x_k + d_k^\ell) + \alpha \langle d_k^0, H_k d_k^0 \rangle \leq f(x_{k-2}) \quad \forall i = 1, \dots, p. \quad (\text{A.26})$$

In view of Proposition 4.4, it suffices to prove (A.26) for all $i \in I(x^*)$. Let $i, s \in I(x^*)$. Expanding $f_i(x_k + d_k^\ell)$ to first order about x^* and making use of the optimality conditions (2.5) at x^* give

$$\begin{aligned}
 f_i(x_k + d_k^\ell) &= f_i(x^*) + \langle \nabla f_i(x^*), x_k + d_k^\ell - x^* \rangle + O(\|x_k + d_k^\ell - x^*\|^2) \\
 &\quad - \langle \nabla_x L(x^*, \mu^*, \lambda^*), x_k + d_k^\ell - x^* \rangle \\
 &= f_i(x^*) - \sum_{s \in I(x^*)} \mu_s^* \langle \nabla f_s(x^*) - \nabla f_i(x^*), x_k + d_k^\ell - x^* \rangle \\
 &\quad - \sum_{j \in J(x^*)} \lambda_j^* \langle \nabla g_j(x^*), x_k + d_k^\ell - x^* \rangle \\
 &\quad + O(\|x_k + d_k^\ell - x^*\|^2).
 \end{aligned} \tag{A.27}$$

Making use of (3.36) and (3.37) and observing that $g_j(x^*) = 0$ for all $j \in J(x^*)$, expanding $g_j(x_k + d_k)$ about x^* for all $j \in J(x^*)$ and substituting in (A.27) we obtain

$$\begin{aligned}
 f_i(x_k + d_k^\ell) &= f_i(x^*) - \sum_{j \in J(x^*)} \lambda_j^* g_j(x_k + d_k^\ell) \\
 &\quad + O(\|d_k^\ell\|^2) + O(\|x_k + d_k^\ell - x^*\|^2).
 \end{aligned} \tag{A.28}$$

On the other hand, in view of Proposition 4.4(i), the optimality conditions (4.13) implies

$$g_j(x_k + d_k^0) = O(\|d_k^0\|^2) \quad \forall j \in J(x^*).$$

In view of Proposition 4.4(iii), it thus follows from (A.28) that

$$f(x_k + d_k^\ell) - \alpha \langle d_k^0, H_k d_k^0 \rangle = f(x^*) + O(\|x_k + d_k^\ell - x^*\|^2) + O(\|d_k^0\|^2)$$

since, in view of Proposition 4.4(i), the λ_k 's are bounded. It then follows from Theorem 4.5 and (A.14) that

$$f(x_k + d_k^\ell) - \alpha \langle d_k^0, H_k d_k^0 \rangle = f(x^*) + o(\|x_{k-2} - x^*\|^2).$$

(A.26) then follows in view of Lemma A.3. The theorem follows from (A.26) and the fact that, in view of Proposition 4.4(iv) and Lemma 4.3, for k large enough,

$$g_j(x_k + d_k^\ell) \leq 0, \quad j = 1, \dots, m.$$

□

A.3 Proofs for Chapter 5

Proof of Lemma 5.5. $f(x_k) - f(x^*) > 0$ follows from the convergence of $\{x_k\}$ and Lemma 5.4. In view of (4.13) and from the definition of $L(x_k, \lambda^*)$, we obtain

$$\begin{aligned} \langle \nabla f(x_k), d_k^0 \rangle &= \langle \nabla f(x_k), d_k^0 \rangle + \sum_{j=1}^m \lambda_{k,j} \{g_j(x_k) + \langle \nabla g_j(x_k), d_k^0 \rangle\} \\ &= L(x_k, \lambda^*) - f(x_k) + \langle \nabla_x L(x_k, \lambda^*), d_k^0 \rangle \\ &= L(x_k, \lambda^*) - f(x_k) + \langle \nabla_x L(x_k, \lambda^*), x^* - x_k \rangle \end{aligned} \quad (\text{A.29})$$

$$+ \langle \nabla_x L(x_k, \lambda^*), x_k + d_k^0 - x^* \rangle \quad (\text{A.30})$$

$$= L(x_k, \lambda^*) - f(x_k) + \langle \nabla_x L(x_k, \lambda^*), x^* - x_k \rangle + o(\|x_k + d_k^0 - x^*\|^2). \quad (\text{A.31})$$

Since

$$\begin{aligned} \nabla_x L(x^*, \lambda^*) &= 0 \\ &= \nabla_x L(x_k, \lambda^*) + \langle \nabla_{xx}^2 L(x^*, \lambda^*), x^* - x_k \rangle \\ &\quad + O(\|x^* - x_k\|^2), \end{aligned}$$

it follows that

$$\begin{aligned} L(x^*, \lambda^*) &= L(x_k, \lambda^*) + \langle \nabla_x L(x_k, \lambda^*), x^* - x_k \rangle \\ &\quad + \frac{1}{2} \langle x^* - x_k, \nabla_{xx}^2 L(x^*, \lambda^*)(x^* - x_k) \rangle + o(\|x^* - x_k\|^2) \\ &= L(x_k, \lambda^*) + \frac{1}{2} \langle \nabla_x L(x_k, \lambda^*), x^* - x_k \rangle \\ &\quad + o(\|x^* - x_k\|^2). \end{aligned}$$

Now, (A.31) becomes

$$\begin{aligned} \langle \nabla f(x_k), d_k^0 \rangle &= L(x_k, \lambda^*) - f(x_k) + 2\{L(x^*, \lambda^*) - L(x_k, \lambda^*)\} \\ &\quad + o(\|x_k + d_k^0 - x^*\|^2) \\ &= 2L(x^*, \lambda^*) - f(x_k) - L(x_k, \lambda^*) \\ &\quad + o(\|x_k + d_k^0 - x^*\|^2). \end{aligned}$$

Therefore, the left hand side of (5.15) can be given by

$$\begin{aligned} f(x_k) - f(x^*) + \alpha \langle \nabla f(x_k), d_k^0 \rangle &= f(x_k) - f(x^*) - \alpha f(x_k) + 2\alpha f(x^*) - \alpha L(x^*, \lambda^*) \\ &\quad + o(\|x_k - x^*\|^2) \\ &= (1 - 2\alpha)\{f(x_k) - f(x^*)\} + \alpha\{f(x_k) - L(x^*, \lambda^*)\} \\ &\quad + o(\|x_k - x^*\|^2). \end{aligned}$$

Since $f(x_k) - L(x_k, \lambda^*) \geq 0$, we obtain, in view of Lemma 5.3,

$$\begin{aligned}
& f(x_k) - f(x^*) + \alpha \langle \nabla f(x_k), d_k^0 \rangle \\
& \geq (1 - 2\alpha) \{f(x_k) - f(x^*)\} + o(\|x_k - x^*\|^2) \\
& = \frac{1}{2}(1 - 2\alpha) \{f(x_k) - f(x^*)\} + \frac{1}{2}(1 - 2\alpha) \{f(x_k) - f(x^*)\} \\
& \quad + o(\|x_k - x^*\|^2) \\
& \geq \frac{1}{2}(1 - 2\alpha) \{f(x_k) - f(x^*)\} + \frac{1}{2}(1 - 2\alpha) c_5 \|x_k - x^*\|^2 \\
& \quad + o(\|x_k - x^*\|^2) \\
& \geq c_6 \{f(x_k) - f(x^*)\},
\end{aligned}$$

with $c_6 = \frac{1}{2}(1 - 2\alpha)$. □

A.4 Proofs for Chapter 6

Proof of Lemma 6.9. As previously, let $\delta_k = -\frac{1}{2} \langle d_k^0, H_k d_k^0 \rangle$. We suppose by contradiction the claim is not true, i.e., there exists $K' \subset K$ such that, for all $k \in K'$ there exists $\bar{\omega}_k \in \Omega_0^f(\bar{x}_{k+1})$, such that

$$\phi(x_{k+1}, \bar{\omega}_k) + \langle \nabla_x \phi(x_{k+1}, \bar{\omega}_k), d_k^0 \rangle - \max_{\ell=0,1,2} \{f_{\Omega^q}(x_{k-\ell})\} < 2\alpha\delta_k. \quad (\text{A.32})$$

Without loss of generality, we may assume $t_k < 1$ for all $k \in K'$. In view of Lemma 6.3 and A3.3, δ_k is bounded. Therefore, there exists a subset $K'' \subset K'$ such that the sequences $\{x_k\}$, $\{d_k^0\}$, $\{\delta_k\}$ and $\{\bar{\omega}_k\}$ converge on K'' respectively to x^* , d^* , δ^* and ω^* , and $\delta^* \leq \underline{\delta} < 0$. In view of Lemma 6.6, $\{\bar{x}_{k+1}\}$, $\{x_{k-1}\}$ and $\{x_{k-2}\}$ all converge to x^* on K'' , and, in view of Lemma 6.5 and Lemma 6.6, $\{t_k \delta_k\}$ converges to zero. Furthermore, since $\omega_k \in \Omega_0^f(x_{k+1})$ for all $k \in K''$, it follows from continuity of ϕ and f (A6.1) that $\omega^* \in \Omega_0^f(x^*)$. Thus, taking limit of (A.32) on K'' yields

$$\langle \nabla_x \phi(x^*, \omega^*), d^* \rangle \leq 2\alpha\delta^*. \quad (\text{A.33})$$

On the other hand, since $\bar{\omega}_k \in \Omega_0^f(\bar{x}_{k+1})$ and $t_k < 1$, it follows from the line search rule that

$$\phi(x_k + \frac{t_k}{\beta} d_k^0 + (\frac{t_k}{\beta})^2 \tilde{d}_k, \bar{\omega}_k) > \max_{\ell=0,1,2} \{f_{\Omega^q}(x_{k-\ell})\} + \alpha \frac{t_k}{\beta} \delta_k \geq \phi(x_k, \bar{\omega}_k) + \alpha \frac{t_k}{\beta} \delta_k.$$

Thus,

$$\frac{\phi(x_k + \frac{t_k}{\beta} d_k^0 + (\frac{t_k}{\beta})^2 \tilde{d}_k, \bar{\omega}_k) - \phi(x_k, \bar{\omega}_k)}{\frac{t_k}{\beta}} > \alpha \delta_k. \quad (\text{A.34})$$

Taking limit of (A.34) as $k \rightarrow \infty$ on K'' yields

$$\langle \nabla_x \phi(x^*, \omega^*), d^* \rangle \geq \alpha \delta^*,$$

which contradicts (A.33). The claim is proven. $\square$

Proof of Lemma 6.12. Define

$$L_\omega(d^0) = \frac{1}{2} \langle d^0, H_k d^0 \rangle + \phi(x_k, \omega) + \langle \nabla_x \phi(x_k, \omega), d^0 \rangle,$$

$$L(d^0, \mu) = \sum_{\omega \in \Omega_k^f} \mu_\omega \left\{ \frac{1}{2} \langle d^0, H_k d^0 \rangle + \phi(x_k, \omega) + \langle \nabla_x \phi(x_k, \omega), d^0 \rangle \right\}$$

and

$$U = \{ \mu \in \mathbb{R}^{|\Omega_k^f|} : \sum_{\omega \in \Omega_k^f} \mu_\omega = 1 \ \& \ \mu_\omega \geq 0 \ \forall \omega \in \Omega_k^f \}.$$

With these notations, it follows that

$$\begin{aligned} & \frac{1}{2} \langle d^0, H_k d^0 \rangle + f'_{\Omega_k^f}(x_k, d^0) \\ &= \frac{1}{2} \langle d^0, H_k d^0 \rangle + \max_{\omega \in \Omega_k^f} \{ \phi(x_k, \omega) + \langle \nabla_x \phi(x_k, \omega), d^0 \rangle \} \\ &= \max_{\omega \in \Omega_k^f} L_\omega(d^0). \end{aligned}$$

Given $d^0 \in \mathbb{R}^n$, it is easy to see that

$$\begin{aligned} \max_{\omega \in \Omega_k^f} L_\omega(d^0) &= \max_{\mu \in U} \sum_{\omega \in \Omega_k^f} \mu_\omega L_\omega(d^0) \\ &= \max_{\mu \in U} L(d^0, \mu). \end{aligned}$$

However, $L(d^0, \mu)$ is convex in d^0 and is concave in μ . It follows that

$$\min_{d^0 \in \mathbb{R}^n} \max_{\mu \in U} L(d^0, \mu) = \max_{\mu \in U} \min_{d^0 \in \mathbb{R}^n} L(d^0, \mu).$$

Given $\mu \in U$, in view of assumption A3.3, it follows from $\nabla_{d^0} L(d^0, \mu) = 0$ that

$$d^0 = -H_k^{-1} \left\{ \sum_{\omega \in \Omega_k^f} \mu_\omega \nabla_x \phi(x_k, \omega) \right\}$$

is the unique solution of the problem

$$\min_{d^0 \in \mathbb{R}^n} L(d^0, \mu).$$

Also, from $\langle \nabla_{d^0} L(d^0, \mu), d^0 \rangle = 0$, it holds that

$$\sum_{\omega \in \Omega_k^f} \mu_\omega \langle \nabla_x \phi(x_k, \omega), d^0 \rangle = -\langle d^0, H_k d^0 \rangle.$$

Therefore,

$$\begin{aligned} \min_{d^0 \in R^n} L(d^0, \mu) &= -\frac{1}{2} \langle d^0, H_k d^0 \rangle + \sum_{\omega \in \Omega_k^f} \mu_\omega \{ \phi(x_k, \omega) - f_{\Omega^q}(x_k) \} \\ &= -\frac{1}{2} \|H^{\frac{1}{2}} d^0\|^2 + \sum_{\omega \in \Omega_k^f} \mu_\omega \{ \phi(x_k, \omega) - f_{\Omega^q}(x_k) \} \end{aligned}$$

and the claim follows. □

Appendix B

Auxiliary Materials for Design Examples

Detailed scripts are provided for those readers who are interested in using CONSOL-OPTCAD together with MATLAB. We are not to provide the whole rationale behind CONSOL-OPTCAD. Instead, only the minimal necessary information is supplied that helps to understand the formulation of the optimization problems in the form acceptable to CONSOL-OPTCAD.

Given a specification $f(x, t)$, such as the step response $y(x, t)$ in previous chapter, CONSOL-OPTCAD translates it to the scaled specification

$$f_s(x, t) = \frac{f(x, t) - \text{good_curve}}{\text{bad_curve} - \text{good_curve}}$$

where `good_curve` is a guess on the possibly optimal value of the specification, such as $\bar{y}(t)$ used before, and `bad_curve` is something worse than $\bar{y}(t)$. Thus, the objective function $f_1(x, t) = f_s(x, t)$ in (7.3) is obtained by setting $f(x, t) = y(x, t)$ and

$$\text{good_curve} = \bar{y}(t) \ \& \ \text{bad_curve} = \text{good_curve} + 1,$$

and $f_2(x, t) = f_s(x, t)$ is obtained by setting $f(x, t) = y(x, t)$ and

$$\text{good_curve} = \underline{y}(t) \ \& \ \text{bad_curve} = \text{good_curve} - 1.$$

Similarly, for a non-functional specification $f(x)$, CONSOL-OPTCAD translates it to the scaled function

$$f_s(x) = \frac{f(x) - \text{good_value}}{\text{bad_value} - \text{good_value}}.$$

It can be checked from the optimality conditions for each optimization problems that, as long as the difference ($\text{bad_value} - \text{good_value}$) is the same for all specifications, the original optimization problem is unchanged. For advanced use of $\text{good/bad_curve/values}$, see [22] for details.

For constraints, at least in our cases, $\text{good_curves/values}$ are always zero and bad_curve/values are something negative. All the user-supplied information is input to CONSOL-OPTCAD via a problem description file (PDF). Since MATLAB is chosen to evaluate the various specifications, necessary initialization is performed in the optional `init.m` file which is automatically checked up by CONSOL-OPTCAD. To perform a specified simulation, the file `simu.m` has to be written for each PDF file. `simu.m` communicates with both CONSOL-OPTCAD and MATLAB.

B.1 Scripts for the design using Q -parameterization

B.1.1 Without constraints on u_1

With the choice that $\text{bad_curve} - \text{good_curve} = 1$, both Algorithm 6.2 and the first order algorithm stopped at a local solution without getting into the specified enveloped. Thus, we modified the value of bad_curves so as to meet our need.

```

/*=====
    PDF file without constraints on u_1
    =====*/
design_parameter a1  init=5 min=0.1
design_parameter a2  init=5 min=0.1
design_parameter a3  init=5 min=0.1
design_parameter a4  init=5 min=0.1
design_parameter a5  init=5 min=0.1
design_parameter b0  init=5
design_parameter b1  init=5
design_parameter b2  init=5
design_parameter b3  init=5
INIT_TIME  = 0
FINAL_TIME = 5
DELTA_TIME = 0.01

/*===== description of f_1(x,t) =====*/

```

```

functional_objective "step upp"
  for t from INIT_TIME to FINAL_TIME by DELTA_TIME
  minimize {
    import DELTA_TIME;
    int i;
    double getout();
    i = t/DELTA_TIME + 1.5;
    return getout("resp", i);
  }
  good_curve = {if (t < 1) return 1.15;
                else if (t>=1) return 1.01;}
  bad_curve = {if (t < 1) return 1.17;
               else if (t>=1) return 1.02;}

/*===== description of f_2(x,t) =====*/
functional_objective "step low"
  for t from INIT_TIME to FINAL_TIME by DELTA_TIME
  maximize {
    import DELTA_TIME;
    int i;
    double getout();
    i = t/DELTA_TIME + 1.5;
    return getout("resp", i);
  }
  good_curve = {if (t < 0.5) return -0.08;
                else if (t>=0.5 && t<0.9)
                  return 2.e0*(t-0.5);
                else if (t>=0.9 && t<1.2)
                  return 0.63*t+0.23;
                else return 0.99;}
  bad_curve = {if (t < 0.5) return -0.1;
               else if (t>=0.5 && t<0.9)
                  return 2.e0*(t-0.5)-0.05;
               else if (t>=0.9 && t<1.2)
                  return 0.63*t+0.23-0.05;

```

```

        else return 0.98;}
/*===== end of PDF =====*/

%=====
%   init.m : initialization for MATLAB and CONSOL
%=====
k=-4.906;
z1=8.5687;
z2=-8.4488;
p1=0.20003;
p2=0.35434;
p3=139.53;
nump=[k k*(z1+z2) k*z1*z2];
denp=[1 p1+p2 p1*p2+p3 p1*p3 0];
%===== time interval and step size for MATLAB =====
DELTA_TIME_M = 0.01;
FINAL_TIME = 5;
FINAL_TIME_i = FINAL_TIME/DELTA_TIME_M + 1;
t = 0:DELTA_TIME_M:FINAL_TIME;
%===== end of init.m =====

%=====
%   simu.m for simulation via MATLAB
%=====
numq=[b0 b1 b2 b3 a2*a4*a5*p1*p3/(k1*k2*k3) 0];
denq=[1 a3+a1+a5 (a3+a1)*a5+a4+a1*a3+a2 ...
      a5*(a4+a1*a3+a2)+a1*a4+a2*a3 ...
      (a1*a4+a2*a3)*a5+a2*a4 a2*a4*a5];
[num,den]=series(nump,denp,numq,denq);
resp=step(num,den,t);
save simu resp numq denq;
%===== end of simu.m =====

%=====
%   graph.m to draw step response and control signal

```

```

%=====
t=0:0.01:5;
for i=1:120,yu1(i)=1.15;end;
for i=120:501,yu1(i)=1.01;end;
for i=1:50,yl1(i)=-0.08;end;
for i=51:90,yl1(i)=2*(0.01*i-0.5);end;
for i=91:120,yl1(i)=0.63*0.01*i+0.23;end;
for i=121:501,yl1(i)=0.99;end;
t1=3:0.1:3.5;
for i=1:1:6; slt(i)=0.6; end;
for i=1:1:6; ldt(i)=0.5; end;
for i=1:1:6; dot(i)=0.4; end;
plot(t,yu1,'-.',t,yl1,'--',t,resp,t1,dot,'--',...
      t1,ldt,'-.',t1,slt,'-');
text(3.7,0.59,'response');
text(3.7,0.49,'upper bound');
text(3.7,0.39,'lower bound');
%u1=step(numq,denq);
%plot(t,u1); %draw control signal u_1
xlabel('Time (secs)');
ylabel('Amplitude');
%bode(numg,deng);
%===== end of graph.m =====

```

B.1.2 With constraints on u_1

```

/*=====
    PDF file with constraints on u_1
    =====*/

design_parameter a1  init=10.881180
design_parameter a2  init=74.410016
design_parameter a3  init=235.805772
design_parameter b0  init=0.889810
design_parameter b1  init=0.534202
design_parameter b2  init=131.293781
design_parameter b3  init=25.898322

```

```

initialization {
    extern double pdelta;
    pdelta=1.e-5;
    return;
}
INIT_TIME = 0
FINAL_TIME = 5
DELTA_TIME = 0.01

functional_objective "step upp"
for t from INIT_TIME to FINAL_TIME by DELTA_TIME
    minimize {
        import DELTA_TIME;
        int i;
        double getout();
        i = t/DELTA_TIME + 1.5;
        return getout("resp", i);
    }
    good_curve = {if (t < 1) return 1.15;
                  else if (t>=1) return 1.01;}
    bad_curve = {if (t < 1) return 2.15;
                  else if (t>=1) return 2.01;}

functional_objective "step low"
for t from INIT_TIME to FINAL_TIME by DELTA_TIME
    maximize {
        import DELTA_TIME;
        int i;
        double getout();
        i = t/DELTA_TIME + 1.5;
        return getout("resp", i);
    }
    good_curve = {if (t < 0.5) return -0.10;
                  else if (t>=0.5 && t<1.5)
                      return 0.8e0*(t-0.5)-0.1;

```

```

        else if (t>=1.5 && t<2.5)
            return 0.29*(t-1.5)+0.7;
        else
            return 0.99;}
bad_curve = {if (t < 0.5) return -1.10;
            else if (t>=0.5 && t<1.5)
                return 0.8e0*(t-0.5)-1.1;
            else if (t>=1.5 && t<2.5)
                return 0.29*(t-1.5)-0.3;
            else
                return -0.01;}

/*===== set up the constraint u_1 <= 0.5 =====*/
functional_constraint "control p" hard
for t from INIT_TIME to FINAL_TIME by DELTA_TIME
{
    import DELTA_TIME;
    int i;
    double getout();
    i = t/DELTA_TIME + 1.5;
    return getout("u1", i);
}
<=good_curve={return 0.5;}
bad_curve={return 1.5;}

/*===== set up the constraint u_1 >= -0.5 =====*/
functional_constraint "control n" hard
for t from INIT_TIME to FINAL_TIME by DELTA_TIME
{
    import DELTA_TIME;
    int i;
    double getout();
    i = t/DELTA_TIME + 1.5;
    return getout("u1", i);
}
>=good_curve={return -0.5;}
bad_curve={return -1.5;}

```

```
/*===== end of PDF file =====*/
```

```
%=====
%  simu.m for simulation via MATLAB
%=====
numc=[b0 b1 b2 b3];
denc=[1.0 a1 a2 a3];
[numg,deng]=series(nump,denp,numc,denc);
[num,den]=feedback(numg,deng,1,1);
[numq,denq]=feedback(numc,denc,nump,denp);
resp=step(num,den,t);
u1=step(numq,denq,t);
save simu resp u1;
%===== end of simu.m =====

%=====
%  graph.m:draw step response and control signal u_1
%=====
t=0:0.01:5;
for i=1:120,yu1(i)=1.15;end;
for i=120:501,yu1(i)=1.01;end;
for i=1:50,yl1(i)=-0.1;end;
for i=51:150,yl1(i)=0.8*(0.01*i-0.5)-0.1;end;
for i=151:250,yl1(i)=0.29*(0.01*i-1.5)+0.7;end;
for i=251:501,yl1(i)=0.99;end;
t1=3:0.1:3.5;
for i=1:1:6; slt(i)=0.6; end;
for i=1:1:6; ldt(i)=0.5; end;
for i=1:1:6; dot(i)=0.4; end;
plot(t,yu1,'-.',t,yl1,'--',t,resp,t1,dot,'--',...
      t1,ldt,'-.',t1,slt,'-');
text(3.7,0.59,'response');
text(3.7,0.49,'upper bound');
text(3.7,0.39,'lower bound');
%plot(t,u1) %draw control signal u_1
```

```

xlabel('Time (secs)');
ylabel('Amplitude');
%===== end of graph.m =====

```

B.2 Scripts for the state feedback design

```

/*=====
    PDF file for the state feedback design
    [k1 k2 k3 k4]=K0 which is obtained in init.m
    =====*/

design_parameter k1  init=0.743
design_parameter k2  init=0.188
design_parameter k3  init=4.6406
design_parameter k4  init=1.
INIT_TIME  = 0
FINAL_TIME = 5
DELTA_TIME = 0.01
functional_objective "step upp"
for t from INIT_TIME to FINAL_TIME by DELTA_TIME
    minimize {
        import DELTA_TIME;
        int i;
        double getout();
        i = t/DELTA_TIME + 1.5;
        return getout("resp", i);
    }
    good_curve = {if (t < 1) return 1.15;
                  else if (t>=1) return 1.01;}
    bad_curve =  {if (t < 1) return 2.15;
                  else if (t>=1) return 2.01;}
functional_objective "step low"
for t from INIT_TIME to FINAL_TIME by DELTA_TIME
    maximize {
        import DELTA_TIME;
        int i;

```

```

    double getout();
    i = t/DELTA_TIME + 1.5;
    return getout("resp", i);
}

good_curve = {if (t < 0.5) return -0.08;
              else if (t>=0.5 && t<0.9)
                  return 2.e0*(t-0.5);
              else if (t>=0.9 && t<1.2)
                  return 0.63*t+0.23;
              else return 0.99;};

bad_curve = {if (t < 0.5) return -1.08;
             else if (t>=0.5 && t<0.9)
                 return 2.e0*(t-0.5)-1.0;
             else if (t>=0.9 && t<1.2)
                 return 0.63*t+0.23-1.0;
             else return -0.01;};

/*===== end of PDF file =====*/

%=====
%  init.m
%=====

k=-4.906;
z1=8.5687;
z2=-8.4488;
p1=0.20003;
p2=0.35434;
p3=139.53;
nump=[k k*(z2+z3) k*z2*z3];
denp=[1 p1+p2 p1*p2+p3 p1*p3 0];
[A,B,C,D]=tf2ss(nump,denp);
%===== time interval and step size for MATLAB =====
DELTA_TIME_M = 0.01;
FINAL_TIME = 5;
FINAL_TIME_i = FINAL_TIME/DELTA_TIME_M + 1;
t = 0:DELTA_TIME_M:FINAL_TIME;

```

```

%= find initial stablizing state feedback using LQG =
q=[1 1 1 1];
r=1;
Q=diag(q);
K0=lqr(A,B,Q,r);
%===== end of the init.m file =====

%=====
%   simu.m
%=====
K=[k1 k2 k3 k4];
AA=A-B*K;
resp=step(AA,B,C,D,1,t);
save simu resp AA B C D;
%===== end of the simu.m file =====

%=====
%   graph.m:draw the step response
%=====
t=0:0.01:5;
y=mstep(AA,B,C,D,1,t);
for i=1:1:120 yu1(i)=1.15;end
for i=120:1:501 yu1(i)=1.01;end
for i=1:1:50 yl1(i)=-0.08;end
for i=51:1:90 yl1(i)=2*(0.01*i-0.5);end
for i=91:1:120 yl1(i)=0.63*0.01*i+0.23;end
for i=121:1:501 yl1(i)=0.99;end
t1=3:0.1:3.5;
for i=1:1:6 slt(i)=0.6; end
for i=1:1:6 ldt(i)=0.5; end
for i=1:1:6 dot(i)=0.4; end
plot(t,yu1,'-.',t,yl1,'--',t,y,t1,dot,'--',...
      t1,ldt,'-.',t1,slt,'-');
text(3.7,0.59,'response');
text(3.7,0.49,'upper bound');

```

```

text(3.7,0.39,'lower bound');
xlabel('Time (secs)');
ylabel('Amplitude');
/*===== end of the graph.m file =====*/

```

B.3 Scripts for the PID control design

There is no need of MATLAB for this problem. We use it to facilitate our data collection in making the tests reports, since the interface for CONSOL-OPTCAD and MATLAB reports each call of the simulator (MATLAB).

```

/*=====
PDF for PID controller design
=====*/
design_parameter x1 init = 1  min = 0    max = 100
design_parameter x2 init = 1  min = 0.1  max = 100
design_parameter x3 init = 1  min = 0    max = 100
initialization {
    extern double pdelta;
    pdelta=1.e-03;
    return;
}

/*===== description of the objective =====*/
objective "mmsr"
minimize
{
    import x1, x2, x3;
    double f,g;
    f=x2*(122+17*x1+6*x3-5*x2+x1*x3)+180*x3-36*x1+1224;
    g=(x2*(408+56*x1-50*x2+60*x3+10*x1*x3-2*x1*x1);
    return f/g;
}
good_value = 0.1746
bad_value  = 1.1746

```

```

/*===== description of the constraint =====*/
functional_constraint "mmsr" hard
    for w from 0.000001 to 30.000001 by 1
    {
        double getout();
        int i;
        i=w/1+1.5;
        return getout("result",i);
    }
    <= good_curve = { return 0.0; }
    bad_curve = { return 1.0; }
/*===== end of PDF file =====*/

%=====
% simu.m
%=====
for i=1:1:31
    w=0.000001+(i-1);
    den = w^7+9*w^5+4*w^3+36*w;
    ImT = (x1-5*x3)*w^4+(6*x3+5*x2-8*x1)*w*w-6*x2;
    ImT = ImT/den;
    ReT = -x3*w^5+(x2+8*x3-5*x1)*w^3+(6*x1-8*x2)*w;
    ReT = 1+ReT/den;
    result(i)=ImT-3.33*ReT*ReT+1;end
    save simu result;
%===== end of simu.m =====

%=====
% graph.m: draw bode plot
%=====
w=0.000001:0.1:30;
nump=[0 0 0 1];
denp=[1 5 8 6];
numc=[x3 x1 x2];
denc=[0 1 0];

```

```
[numg,deng]=series(nump,denp,numc,denc);  
%bode(numg,deng,w);  
[num,den]=feedback(numg,deng,1,1);  
step(num,den);  
%===== end of graph.m =====
```

Bibliography

- 1 Harwell Subroutine Library, Library Reference Manual, Harwell, England, 1985.
- 2 L. Armijo, Minimization of Functions Having Continuous Partial Derivatives, *Pacific Journal of Mathematics* 16 (1966), 1–3.
- 3 P. T. Boggs and J. W. Tolle, A Strategy for Global Convergence in a Sequential Quadratic Programming Algorithm, *SIAM J. Numer. Anal.* 26 (1989), 600–623.
- 4 P. T. Boggs, J. W. Tolle and P. Wang, On the Local Convergence of Quasi-Newton Methods for Constrained Optimization, *SIAM J. Control Optim.* 20 (1982), 161–171.
- 5 J. F. Bonnans, Asymptotic Admissibility of the Unit Stepsize in Exact Penalty Methods, *J. Optim. Theory Appl.* 27 (1989), 631–641.
- 6 J. F. Bonnans, E. R. Panier, A. L. Tits and J. L. Zhou, Avoiding the Maratos Effect by Means of a Nonmonotone Line Search. II. Inequality Constrained Problems – Feasible Iterates, *SIAM J. Numer. Anal.* 29 (August 1992), to appear.
- 7 S. P. Boyd, V. Balakrishnan, C. H. Barratt, N. M. Khraishi, X. Li, D. G. Meyer and S. A. Norman, A New CAD Method and Associated Architectures for Linear Controllers, *IEEE Trans. Automat. Control* AC-33 (1988), 268–283.
- 8 S. P. Boyd and C. H. Barratt, *Linear Controller Design: Limits of Performance*, Prentice-Hall, Englewood Cliffs, NJ, 1991.
- 9 R. K. Brayton, S. W. Director and G. D. Hachtel, A New Algorithm for Statistical Circuit Design Based on Quasi-Newton Methods and Function Splitting, *IEEE Trans. Acoust. Speech Signal Process.* CAS-26 (September 1979), 784–794.
- 10 R. H. Byrd, R. A. Tapia and Y. Zhang, An SQP Augmented Lagrangian BFGS Algorithm for Constrained Optimization, Department of Mathematical Sciences, Technical Report 89-4, Rice University, Houston, Texas, November, 1989 .

- 11 R. H. Byrd, An Example of Irregular Convergence in Some Constrained Methods That Use the Projected Hessian, *Math. Programming* 32 (1985), 232–237.
- 12 R. M. Chamberlain, M. J. D. Powell, C. Lemarechal and H. C. Pedersen, The Watchdog Technique for Forcing Convergence in Algorithms for Constrained Optimization, *Math. Programming Stud.* 16 (1982), 1–17.
- 13 C. Charalambous and A. R. Conn, An Efficient Method to Solve the Minimax Problem Directly, *SIAM J. Numer. Anal.* 15 (1978), 162–187.
- 14 T. F. Coleman and A. R. Conn, Nonlinear Programming Via an Exact Penalty Function: Asymptotic Analysis, *Math. Programming* 24 (1982), 123–136.
- 15 T. F. Coleman and A. R. Conn, Nonlinear Programming Via an Exact Penalty Function: Global Analysis, *Math. Programming* 24 (1982), 137–161.
- 16 A. R. Conn, An Efficient Second Order Method to Solve the (Constrained) Minimax Problem, Research Report CORR 79-5, Department of Combinatorics and Optimization, University of Waterloo, Ontario, 1979.
- 17 A. R. Conn and Y. Li, An Efficient Algorithm for Nonlinear Minimax Problems, Research Report CS-88-41, University of Waterloo, Waterloo, Ontario, N2L 3G1 Canada, November, 1989 .
- 18 J. M. Daniel, Stability of the Solution of Definite Quadratic Programs, *Math. Programming* 5 (1970), 41–53.
- 19 V. F. Dem'yanov and V. N. Malozenov, *Introduction to Minimax*, Halsted Press (John Wiley and Sons), 1974.
- 20 G. DiPillo and L. Grippo, A Continuously Differentiable Exact Penalty Function for Nonlinear Programming with Inequality Constraints, *SIAM J. Control Optim.* 23 (1985), 72–84.
- 21 J. S. Donat, N. Bhat, T. J. McAvoy and J. Zhou, Predictive Control Using Neural Net Dynamic Models and Optimization, IMACS Congress, Dublin, Ireland, July, 1991.
- 22 M. K. H. Fan, A. L. Tits, J. L. Zhou, L. S. Wang and J. Koninckx, CONSOL-OPTCAD User's Manual (Version 1.1, Released 9/1990), Technical Report TR-87-212r3, Systems Research Center, University of Maryland, College Park, Maryland 20742, 1991.

- 23 M. K. H. Fan, L. S. Wang, J. Koninckx and A. L. Tits, CONSOLE User's Manual, Technical Report TR-87-212, Systems Research Center, University of Maryland, College Park, Maryland 20742, 1987.
- 24 S. C. Feng and T. H. Hopp, A Review of Current Geometric Tolerancing Theories and Inspection Data Analysis Algorithms, Technical Report, NISTIR 4509, National Institute of Standards and Technology, U.S. Department of Commerce, February 1991.
- 25 A. V. Fiacco and K. O. Kortanek, *Semi-Infinite Programming and Applications*, Lecture Notes in Economic and Mathematical Systems), No. 215, Springer-Verlag, New York-Heidelberg-Berlin, 1983.
- 26 R. Fletcher, An Ideal Penalty Function for Constrained Optimization, *J. Inst. Maths Applies* 15 (1975), 319-342.
- 27 R. Fletcher, Second Order Corrections for Non-differentiable Optimization, in *Numerical Analysis, Dundee, 1981, Lecture Notes in Mathematics* 912 (G. A. Watson, Ed.), Springer-Verlag, 1981, pp. 85-114.
- 28 R. Gonin and A. Money, *Nonlinear LP-Norm Estimation*, Marcel Dekker, Inc., New York, 1989.
- 29 C. Gonzaga and E. Polak, On Constraint Dropping Schemes and Optimality Functions for a Class of Outer Approximations Algorithms, *SIAM J. Control Optim.* 17 (1979), 477-493.
- 30 C. Gonzaga, E. Polak and R. Trahan, An Improved Algorithm for Optimization Problems with Functional Inequality Constraints, *IEEE Trans. Automat. Control* AC-25 (1980), 49-54.
- 31 L. Grippo, F. Lampariello and S. Lucidi, A Nonmonotone Line Search Technique for Newton's Method, *SIAM J. Numer. Anal.* 23 (1986), 707-716.
- 32 S. A. Gustafson, A Three-Phase Algorithm for Semi-Infinite Programs, in *Semi-Infinite Programming and Applications*, Lecture Notes in Economics and Mathematical Systems, No. 215 (A. V. Fiacco and K. O. Kortanek, Eds.), Springer-Verlag, New York-Heidelberg-Berlin, 1983, pp. 138-157.
- 33 S. P. Han, A Globally Convergent Method for Nonlinear Programming, *J. Optim. Theory Appl.* 22 (1977), 297-309.
- 34 S. P. Han, Superlinear Convergence of a Minimax Method, TR78-336, Department of Computer Science, Cornell University, 1978.

- 35 S. P. Han, Variable Metric Methods for Minimizing a Class of Nondifferentiable Functions, *Math. Programming* 20 (1981), 1–13.
- 36 R. Hettich, *Semi-Infinite Programming*, Lecture Notes in Control and Information Sciences), No. 15, Springer Verlag, 1979.
- 37 R. Hettich, An Implementation of a Discretization Method for Semi-Infinite Programming, *Math. Programming* 34 (1986), 354–361.
- 38 R. Hettich and W. van Honstede, On Quadratically Convergent Methods for Semi-Infinite Programming, in *Semi-Infinite Programming*, Lecture Notes in Control and Information Sciences, No. 15 (R. Hettich, Ed.), Springer Verlag, 1979, pp. 97–111.
- 39 W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems (187), Springer Verlag, 1981.
- 40 W. van Honstede, An Approximation Method for Semi-Infinite Problems, in *Semi-Infinite Programming*, Lecture Notes in Control and Information Sciences, No. 15 (R. Hettich, Ed.), Springer Verlag, 1979, pp. 126–136.
- 41 J. E. Horwedel, B. A. Worley, E. M. Oblow, F. G. Pin and R. Q. Wright, GRESS Version 0.0 User's manual, 1989.
- 42 J. E. Dennis Jr. and J. J. More, A Characterization of Superlinear Convergence and Its Application to Quasi-Newton Methods, *Math. Comp.* 28 (1974), 549–560.
- 43 K. C. Kiwiel, A Phase I — Phase II Method for Inequality Constrained Minimax Problems, *Control and Cybernetics* 12 (1983), 55–75.
- 44 K. C. Kiwiel, *Methods of Descent in Nondifferentiable Optimization*, Lecture Notes in Mathematics), No. 1133, Springer-Verlag, Berlin, Heidelberg, New-York, Tokyo, 1985.
- 45 T. P. Lee, W. T. Nye and A. L. Tits, The Design of Digital Filters Using Interactive Optimization, *IEEE Trans. Circuits and Syst.* CAS-30 (1983), 821–824.
- 46 C. Lemaréchal, Nondifferentiable Optimization, in *Optimization*, Handbooks in Operations Research and Management Science (G. Nemhauser, A. Rinnooy-Kan and M. Todd, Eds.), North Holland, 1989.
- 47 C. -A. Lin and S. -H. Chen, Optimization-Based Control Systems Design, in *Proceedings of the 29th IEEE Conference on Decision and Control*, December 1990, pp. 2462–2468.

- 48 D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, Mass., 1973.
- 49 K. Madsen and H. Schjær-Jacobsen, Linearly Constrained Minimax Optimization, *Math. Programming* 14 (1978), 208–223.
- 50 K. Madsen and H. Schjaer-Jacobsen, Algorithms for Worst Case Tolerance Optimization, *IEEE Trans. Circuits and Syst. CAS-26* (1979).
- 51 N. Maratos, *Exact Penalty Function Algorithms for Finite Dimensional and Optimization Problems*, Ph.D. Thesis, Imperial College of Science and Technology, London, U.K., 1978.
- 52 D. Q. Mayne and E. Polak, A Superlinearly Convergent Algorithm for Constrained Optimization Problems, *Math. Programming Stud.* 16 (1982), 45–61.
- 53 D. Q. Mayne, E. Polak and A. Voreadis, A Cut-Map Algorithm for Design Problems with Parameter Tolerances, *IEEE Trans. Acoust. Speech Signal Process. CAS-29* (January 1982), 35–45.
- 54 D. Q. Mayne, E. Polak and R. Trahan, An Outer Approximations Algorithm for Computer-Aided Design Problems, *J. Optim. Theory Appl.* 28 (1979), 331–351.
- 55 W. Murray and M. L. Overton, A Projected Lagrangian Algorithm for Nonlinear Minimax Optimization, *SIAM J. Sci. Stat. Comput.* 1 (September, 1980).
- 56 J. Nocedal and M. L. Overton, Projected Hessian Updating Algorithms for Nonlinearly Constrained Optimization, *SIAM J. Numer. Anal.* 22 (October, 1985).
- 57 W. T. Nye and A. L. Tits, An Application-Oriented, Optimization-Based Methodology for Interactive Design of Engineering Systems, *Internat. J. Control* 43 (1986), 1693–1721.
- 58 J. Ortega and W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
- 59 M. L. Overton, Algorithms for Nonlinear ℓ_1 and ℓ_∞ Fitting, in *Nonlinear Optimization 1981* (M. J. D. Powell, Ed.), Academic Press, London, 1982, pp. 213–221.
- 60 E. R. Panier and A. L. Tits, Globally Convergent Algorithms for Semi-Infinite Optimization Problems Arising in Engineering Design, Technical Report TR-87-28, Systems Research Center, University of Maryland, College Park, MD 20742, 1987.

- 61 E. R. Panier and A. L. Tits, A Superlinearly Convergent Feasible Method for the Solution of Inequality Constrained Optimization Problems, *SIAM J. Control Optim.* 25 (1987), 934–950.
- 62 E. R. Panier and A. L. Tits, A Globally Convergent Algorithm with Adaptively Refined Discretization for Semi-Infinite Optimization Problems Arising in Engineering Design, *IEEE Trans. Automat. Control* AC-34 (1989).
- 63 E. R. Panier and A. L. Tits, Avoiding the Maratos Effect by Means of a Non-monotone Line Search. I. General Constrained Problems, *SIAM J. Numer. Anal.* 28 (1991), 1183–1195.
- 64 E. R. Panier and A. L. Tits, On Combining Feasibility, Descent and Superlinear Convergence in Inequality Constrained Optimization, *Math. Programming*, 1992, to appear.
- 65 E. R. Panier and A. L. Tits, A Superlinearly Convergent Method of Feasible Directions for Optimization Problems Arising in the Design of Engineering Systems, in *Proc. of the Seventh International Conference on Analysis and Optimization of Systems — Antibes, June 25-27, 1986*, Lecture Notes in Control and Information Sciences, No. 83 (A. Bensoussan and J. L. Lions, Eds.), Springer Verlag, Berlin-Heidelberg-New York-Tokyo, June 1986, pp. 65–73.
- 66 E. Polak, *Computational Methods in Optimization*, Academic Press, New York, N.Y., 1971.
- 67 E. Polak, On the Mathematical Foundations of Nondifferentiable Optimization, *SIAM Rev.* 29 (1987), 21–89.
- 68 E. Polak and L. He, Rate Preserving Discretization Strategies for Semi-Infinite Programming and Optimal Control, in *Proceedings of the 29th IEEE Conference on Decision and Control*, December 1990, pp. 2444–2449.
- 69 E. Polak and D. Q. Mayne, An Algorithm for Optimization Problems with Functional Inequality Constraints, *IEEE Trans. Automat. Control* 21 (1976), 184–193.
- 70 E. Polak, D. Q. Mayne and J. E. Higgins, A Superlinearly Convergent Algorithm for Min-max Problems, *Proceedings of the 28th IEEE Conference on Decision and Control*, December 1989.

- 71 E. Polak, P. Siegel, T. Wu, W. T. Nye and D. Q. Mayne, DELIGHT.MIMO: An Interactive, Optimization-Based Multivariable Control System Design Package, in *Computer-Aided Control Systems Engineering* (M. Jamshidi and C. J. Herget, Eds.), Elsevier Science Publishers B.V. (North-Holland), Amsterdam, New York, Oxford, 1985, pp. 139–147.
- 72 E. Polak and A. L. Tits, A Recursive Quadratic Programming Algorithm for Semi-Infinite Optimization Problems, *Appl. Math. Optim.* 8 (1982), 325–349.
- 73 E. Polak and Y. Wardi, Nondifferentiable Optimization Algorithm for the Design of Control Systems Subject to Singular Value Inequalities, *Automatica—J. IFAC* 18 (1982), 267–283.
- 74 R. A. Polyak, Smooth Optimization Methods for Minimax Problems, *SIAM J. Control Optim.* 26 (1988), 1274–1286.
- 75 M. J. D. Powell, A Fast Algorithm for Nonlinearly Constrained Optimization Calculations, in *Numerical Analysis, Dundee, 1977, Lecture Notes in Mathematics 630* (G. A. Watson, Ed.), Springer-Verlag, 1978, pp. 144–157.
- 76 M. J. D. Powell, The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations, in *Nonlinear Programming 3* (O. L. Mangasarian, R. R. Meyer and S. M. Robinson, Eds.), Academic Press, New York, 1978, pp. 27–63.
- 77 M. J. D. Powell, Variable Metric Methods for Constrained Optimization, in *Mathematical Programming, The State of the Art, Bonn 1982* (A. Bachem, M. Grötschel and B. Korte, Eds.), Springer-Verlag, New York–Heidelberg–Berlin, 1983, pp. 288–311.
- 78 K. Schittkowski, The Nonlinear Programming Method of Wilson, Han, and Powell with an Augmented Lagrangian Type Line Search Function. Part 1: Convergence Analysis, *Numerische Mathematik* 38 (1981), 83–114.
- 79 J. Stoer, The Convergence of Sequential Quadratic Programming Methods for Solving Nonlinear Programs, in *Recent Advances in Communication and Control Theory* (R. E. Kalman, G. I. Marchuk, A. E. Ruberti and A. J. Viterbi, Eds.), Optimization Software, Inc., New York, N.Y., 1987, pp. 412–421.
- 80 R. A. Tapia, Diagonalized Multiplier Methods and Quasi-Newton Methods for Constrained Optimization, *J. Optim. Theory Appl.* 22 (1977), 135–194.

- 81 A. L. Tits, M. K. H. Fan and E. R. Panier, Aspects of Optimization-Based CADCS, 4th IFAC Symposium on Computer-Aided Design in Control Systems, Beijing, China, August 23-25, 1988.
- 82 A. L. Tits and J. Zhou, Fast Feasible Direction Methods, With Engineering Applications, in *Proceedings of the 1991 European Control Conference*, Hermès, Paris, Grenoble, 1991, pp. 194–199.
- 83 L. Q. Vu, Optimal Capacitor Design, Technical Report, Aerospace Engineering, Mechanics and Engineering Science, University of Florida, Gainesville, FL 32611, 1991.
- 84 G. A. Watson, The Minimax Solution of an Overdetermined System of Non-linear Equations, *J. Inst. Math. Appl.* 23 (1979), 167–180.
- 85 G. A. Watson, Globally Convergent Methods for Semi-Infinite Programming, *BIT* 21 (1981), 362–373.
- 86 G. A. Watson, Lagrangian Methods for Semi-Infinite Programming Problems, in *Infinite Programming*, Lecture Notes in Economics and Mathematical Systems, No. 259 (E. J. Anderson and A. B. Philpott, Eds.), Springer-Verlag, New York–Heidelberg–Berlin, 1984, pp. 90–107.
- 87 R. S. Womersley and R. Fletcher, An Algorithm for Composite Nonsmooth Optimization Problems, *J. Optim. Theory Appl.* 48 (1986), 493–523.
- 88 Y. X. Yuan, An Only 2-Step Q-Superlinear Convergence Example for Some Algorithms that use Reduced Hessian Approximations, *Math. Programming* 32 (1985), 224–231.
- 89 E. Zafriou and J. M. Zhu, Iterative Batch-to-Batch Optimal Profile Determination for Batch and Semi-Batch Processes, Systems Research Center, University of Maryland, College Park, MD 20742, 1991, in preparation.
- 90 G. Zames, Feedback and Optimal Sensitivity: Model Reference Transformation, Multiplicative Seminorms, and Approximate Inverse, *IEEE Trans. Automat. Control* AC-26 (1981), 301–320.
- 91 G. Zames and B. A. Francis, Feedback, Minimax Sensitivity, and Optimal Robustness, *IEEE Trans. Automat. Control* AC-28 (1983), 585–601.

- 92 J. L. Zhou and A. L. Tits, User's Guide for FSQP Version 2.4 – A Fortran Code for Solving Optimization Problems, Possibly Minimax, with General Inequality Constraints and Linear Equality Constraints, Generating Feasible Iterates, SRC TR-90-60r1c, Systems Research Center, University of Maryland, College Park, MD 20742, 1991.
- 93 G. Zoutendijk, *Methods of Feasible Directions*, Elsevier Scientific Publishing Company, Amsterdam, The Netherlands, 1960.
- 94 S. I. Zukhovitskii, R. A. Polyak and M. E. Primak, An Algorithm for the Solution of Convex Programming Problems, *DAN USSR* 153 (1963), 991–1000.

