CAMBRIDGE

国外信息科学与技术优秀图书系列　电子学与通信技术

信号处理与通信中的凸优化理论

Convex Optimization in Signal Processing and Communications

（英文版）

〔西班牙〕 Daniel P. Palomar
〔以色列〕 Yonina C. Eldar
等 著

科学出版社

国外信息科学与技术优秀图书系列

信号处理与通信中的凸优化理论

（英文版）

Convex Optimization in Signal Processing and Communications

〔西班牙〕 Daniel P. Palomar

〔以色列〕 Yonina C. Eldar 等 著

科学出版社

北 京

图字：01-2012-6185

内 容 简 介

凸优化理论是信号处理领域具有重要应用价值的理论分析工具，最近二十年一大批的信号处理问题都基于凸优化理论取得了突破进展。本书以通信与信号处理中的经典与前沿问题为脉络，深入浅出地介绍了各类凸优化分析的建模方法与基本理论。内容包括图模型理论、基于梯度的信号重建算法、半定松弛(SDP)算法、基于 SDP 的雷达信号设计、图像处理中的盲信源分离、现代抽样理论，特别是宽带移动通信中的 MIMO 信号检测、认知无线电中的波束成形理论、分布式多目标优化理论与博弈论等。

本书可作为电子与通信工程等相关领域科研人员、工程技术人员的参考书，也可供相关专业高年级本科生、研究生阅读。

图书在版编目(CIP)数据

信号处理与通信中的凸优化理论= Convex Optimization in Signal Processing and Communications : 英文 / (西) 帕洛马 (Palomar, D.P.) 等著. —北京 : 科学出版社, 2012
(国外信息科学与技术优秀图书系列)
ISBN 978-7-03-035430-3
Ⅰ. ①信… Ⅱ. ①帕… Ⅲ. ①通信系统-信号处理-凸分析-研究-英文 Ⅳ. ①TN911.72
中国版本图书馆 CIP 数据核字(2012)第 203888 号

责任编辑：张 濮 王 哲 / 责任印制：张 倩 / 封面设计：刘可红

科学出版社出版
北京东黄城根北街16号
邮政编码：100717
http://www.sciencep.com
双青印刷厂 印刷

科学出版社发行 各地新华书店经销
*
2013 年 1 月第 一 版 开本：B5(720×1000)
2013 年 1 月第一次印刷 印张：32
字数：640 000
定价：118.00 元
（如有印装质量问题，我社负责调换）

Contributors

Sergio Barbarossa
University of Rome – La Sapienza
Italy

Amir Beck
Technion – Israel Institute
of Technology
Haifa
Israel

Stephen Boyd
Stanford University
California
USA

Tsung-Han Chan
Tsing Hua University
Hsinchu
Taiwan

Tsung-Hui Chang
Tsing Hua University
Hsinchu
Taiwan

Chong-Yung Chi
Tsing Hua University
Hsinchu
Taiwan

Joachim Dahl
Anybody Technology A/S
Denmark

Yonina C. Eldar
Technion – Israel Institute of Technology
Haifa
Israel

Amr El-Keyi
Alexandia University
Egypt

Francisco Facchinei
University of Rome – La Sapienza
Rome
Italy

Alex B. Gershman
Darmstadt University of Technology
Darmstadt
Germany

Yongwei Huang
Hong Kong University of Science
and Technology
Hong Kong

Thia Kirubarajan
McMaster University
Hamilton, Ontario
Canada

Zhi-Quan Luo
University of Minnesota
Minneapolis
USA

Wing-Kin Ma
Chinese University of Hong Kong
Hong Kong

Antonio De Maio
Università degli Studi di Napoli – Federico II
Naples
Italy

Jacob Mattingley
Stanford University
California
USA

Tomer Michaeli
Technion – Israel Institute of Technology
Haifa
Israel

Angelia Nedić
University of Illinois at Urbana-Champaign
Illinois
USA

Asuman Ozdaglar
Massachusetts Institute of Technology
Boston, Massachusetts
USA

Daniel P. Palomar
Hong Kong University of Science and Technology
Hong Kong

Jong-Shi Pang
University of Illinois at Urbana-Champaign
Illinois
USA

Michael Rübsamen
Darmstadt University of Technology
Darmstadt
Germany

Gesualdo Scutari
Hong Kong University of Science and Technology
Hong Kong

Anthony Man-Cho So
Chinese University of Hong Kong
Hong Kong

Jitkomut Songsiri
University of California
Los Angeles, California
USA

Marc Teboulle
Tel-Aviv University
Tel-Aviv
Israel

Lieven Vandenberghe
University of California
Los Angeles, California
USA

Yue Wang
Virginia Polytechnic Institute and State University
Arlington
USA

Yinyu Ye
Stanford University
California
USA

Shuzhong Zhang
Chinese University of Hong Kong
Hong Kong

Preface

The past two decades have witnessed the onset of a surge of research in optimization. This includes theoretical aspects, as well as algorithmic developments such as generalizations of interior-point methods to a rich class of convex-optimization problems. The development of general-purpose software tools together with insight generated by the underlying theory have substantially enlarged the set of engineering-design problems that can be reliably solved in an efficient manner. The engineering community has greatly benefited from these recent advances to the point where convex optimization has now emerged as a major signal-processing technique. On the other hand, innovative applications of convex optimization in signal processing combined with the need for robust and efficient methods that can operate in real time have motivated the optimization community to develop additional needed results and methods. The combined efforts in both the optimization and signal-processing communities have led to technical breakthroughs in a wide variety of topics due to the use of convex optimization. This includes solutions to numerous problems previously considered intractable; recognizing and solving convex-optimization problems that arise in applications of interest; utilizing the theory of convex optimization to characterize and gain insight into the optimal-solution structure and to derive performance bounds; formulating convex relaxations of difficult problems; and developing general purpose or application-driven specific algorithms, including those that enable large-scale optimization by exploiting the problem structure.

This book aims at providing the reader with a series of tutorials on a wide variety of convex-optimization applications in signal processing and communications, written by worldwide leading experts, and contributing to the diffusion of these new developments within the signal-processing community. The goal is to introduce convex optimization to a broad signal-processing community, provide insights into how convex optimization can be used in a variety of different contexts, and showcase some notable successes. The topics included are automatic code generation for real-time solvers, graphical models for autoregressive processes, gradient-based algorithms for signal-recovery applications, semidefinite programming (SDP) relaxation with worst-case approximation performance, radar waveform design via SDP, blind non-negative source separation for image processing, modern sampling theory, robust broadband beamforming techniques, distributed multiagent optimization for networked systems, cognitive radio systems via game theory, and the variational-inequality approach for Nash-equilibrium solutions.

There are excellent textbooks that introduce nonlinear and convex optimization, providing the reader with all the basics on convex analysis, reformulation of optimization problems, algorithms, and a number of insightful engineering applications. This book is targeted at advanced graduate students, or advanced researchers that are already familiar with the basics of convex optimization. It can be used as a textbook for an advanced graduate course emphasizing applications, or as a complement to an introductory textbook that provides up-to-date applications in engineering. It can also be used for self-study to become acquainted with the state of-the-art in a wide variety of engineering topics.

This book contains 12 diverse chapters written by recognized leading experts worldwide, covering a large variety of topics. Due to the diverse nature of the book chapters, it is not possible to organize the book into thematic areas and each chapter should be treated independently of the others. A brief account of each chapter is given next.

In Chapter 1, Mattingley and Boyd elaborate on the concept of convex optimization in real-time embedded systems and automatic code generation. As opposed to generic solvers that work for general classes of problems, in real-time embedded optimization the same optimization problem is solved many times, with different data, often with a hard real-time deadline. Within this setup, the authors propose an automatic code-generation system that can then be compiled to yield an extremely efficient custom solver for the problem family.

In Chapter 2, Beck and Teboulle provide a unified view of gradient-based algorithms for possibly nonconvex and non-differentiable problems, with applications to signal recovery. They start by rederiving the gradient method from several different perspectives and suggest a modification that overcomes the slow convergence of the algorithm. They then apply the developed framework to different image-processing problems such as ℓ_1-based regularization, TV-based denoising, and TV-based deblurring, as well as communication applications like source localization.

In Chapter 3, Songsiri, Dahl, and Vandenberghe consider graphical models for autoregressive processes. They take a parametric approach for maximum-likelihood and maximum-entropy estimation of autoregressive models with conditional independence constraints, which translates into a sparsity pattern on the inverse of the spectral-density matrix. These constraints turn out to be nonconvex. To treat them, the authors propose a relaxation which in some cases is an exact reformulation of the original problem. The proposed methodology allows the selection of graphical models by fitting autoregressive processes to different topologies and is illustrated in different applications.

The following three chapters deal with optimization problems closely related to SDP and relaxation techniques.

In Chapter 4, Luo and Chang consider the SDP relaxation for several classes of quadratic-optimization problems such as separable quadratically constrained quadratic programs (QCQPs) and fractional QCQPs, with applications in communications and signal processing. They identify cases for which the relaxation is tight as well as classes of quadratic-optimization problems whose relaxation provides a guaranteed, finite worst-case approximation performance. Numerical simulations are carried out to assess the efficacy of the SDP-relaxation approach.

In Chapter 5, So and Ye perform a probabilistic analysis of SDP relaxations. They consider the problem of maximum-likelihood detection for multiple-input–multiple-output systems via SDP relaxation plus a randomization rounding procedure and study its loss in performance. In particular, the authors derive an approximation guarantee based on SDP weak-duality and concentration inequalities for the largest singular value of the channel matrix. For example, for MPSK constellations, the relaxed SDP detector is shown to yield a constant factor approximation to the ML detector in the low signal-to-noise ratio region.

In Chapter 6, Huang, De Maio, and Zhang treat the problem of radar design based on convex optimization. The design problem is formulated as a nonconvex QCQP. Using matrix rank-1 decompositions they show that nonetheless strong duality holds for the nonconvex QCQP radar code-design problem. Therefore, it can be solved in polynomial time by SDP relaxation. This allows the design of optimal coded waveforms in the presence of colored Gaussian disturbance that maximize the detection performance under a control both on the region of achievable values for the Doppler-estimation accuracy and on the similarity with a given radar code.

The next three chapters consider very different problems, namely, blind source separation, modern sampling theory, and robust broadband beamforming.

In Chapter 7, Ma, Chan, Chi, and Wang consider blind non-negative source separation with applications in imaging. They approach the problem from a convex-analysis perspective using convex-geometry concepts. It turns out that solving the blind separation problem boils down to finding the extreme points of a polyhedral set, which can be efficiently solved by a series of linear programs. The method is based on a deterministic property of the sources called local dominance which is satisfied in many applications with sparse or high-contrast images. A robust method is then developed to relax the assumption. A number of numerical simulations show the effectiveness of the method in practice.

In Chapter 8, Michaeli and Eldar provide a modern perspective on sampling theory from an optimization point of view. Traditionally, sampling theories have addressed the problem of perfect reconstruction of a given class of signals from their samples. During the last two decades, it has been recognized that these theories can be viewed in a broader sense of projections onto appropriate subspaces. The authors introduce a complementary viewpoint on sampling based on optimization theory. They provide extensions and generalizations of known sampling algorithms by constructing optimization problems that take into account the goodness of fit of the recovery to the samples as well as any other prior information on the signal. A variety of formulations are considered including aspects such as noiseless/noisy samples, different signal priors, and different least-squares/minimax objectives.

In Chapter 9, Rübsamen, El-Keyi, Gershman, and Kirubarajan develop several worst-case broadband beamforming techniques with improved robustness against array manifold errors. The methods show a robustness matched to the presumed amount of uncertainty, each of them offering a different trade-off in terms of interference suppression capability, robustness against signal self-nulling, and computational complexity.

The authors obtain convex second-order cone programming and SDP reformulations of the proposed beamformer designs which lead to efficient implementation.

The last three chapters deal with optimization of systems with multiple nodes. Chapter 10 takes an optimization approach with cooperative agents, whereas Chapters 11 and 12 follow a game-theoretic perspective with noncooperative nodes.

In Chapter 10, Nedic and Ozdaglar study the problem of distributed optimization and control of multiagent networked systems. Within this setup, a network of agents has to cooperatively optimize in a distributed way a global-objective function, which is a combination of local-objective functions, subject to local and possibly global constraints. The authors present both classical results as well as recent advances on design and analysis of distributed-optimization algorithms, with recent applications. Two main approaches are considered depending on whether the global objective is separable or not; in the former case, the classical Lagrange dual decompositions can be employed, whereas in the latter case consensus algorithms are the fundamental building block. Practical issues associated with the implementation of the optimization algorithms over networked systems are also considered such as delays, asynchronism, and quantization effects in the network implementation.

In Chapter 11, Scutari, Palomar, and Barbarossa apply the framework of game theory to different communication systems, namely, ad-hoc networks and cognitive radio systems. Game theory describes and analyzes scenarios with interactive decisions among different players, with possibly conflicting goals, and is very suitable for multiuser systems where users compete for the resources. For some problem formulations, however, game theory may fall short, and it is then necessary to use the more general framework of variational-inequality (VI) theory. The authors show how many resource-allocation problems in ad-hoc networks and in the emerging field of cognitive radio networks fit naturally either in the game-theoretical paradigm or in the more general theory of VI (further elaborated in the following chapter). This allows the study of existence/uniqueness of Nash-equilibrium points as well as the design of practical algorithms with provable converge to an equilibrium.

In Chapter 12, Facchinei and Pang present a comprehensive mathematical treatment of the Nash-equilibrium problem based on the variational-inequality and complementarity approach. They develop new results on existence of equilibria based on degree theory, global uniqueness, local-sensitivity analysis to data variation, and iterative algorithms with convergence conditions. The results are then illustrated with an application in communication systems.

Contents

1 Automatic code generation for real-time convex optimization

Jacob Mattingley and Stephen Boyd

This chapter concerns the use of convex optimization in real-time embedded systems, in areas such as signal processing, automatic control, real-time estimation, real-time resource allocation and decision making, and fast automated trading. By "embedded" we mean that the optimization algorithm is part of a larger, fully automated system, that executes automatically with newly arriving data or changing conditions, and without any human intervention or action. By "real-time" we mean that the optimization algorithm executes much faster than a typical or generic method with a human in the loop, in times measured in milliseconds or microseconds for small and medium size problems, and (a few) seconds for larger problems. In real-time embedded convex optimization the same optimization problem is solved many times, with different data, often with a hard real-time deadline. In this chapter we propose an automatic code generation system for real-time embedded convex optimization. Such a system scans a description of the problem family, and performs much of the analysis and optimization of the algorithm, such as choosing variable orderings used with sparse factorizations and determining storage structures, at code generation time. Compiling the generated source code yields an extremely efficient custom solver for the problem family. We describe a preliminary implementation, built on the Python-based modeling framework CVXMOD, and give some timing results for several examples.

1.1 Introduction

1.1.1 Advisory optimization

Mathematical optimization is traditionally thought of as an aid to human decision making. For example, a tool for portfolio optimization *suggests* a portfolio to a human decision maker, who possibly carries out the proposed trades. Optimization is also used in many aspects of engineering design; in most cases, an engineer is in the decision loop, continually reviewing the proposed designs and changing parameters in the problem specification, if needed.

When optimization is used in an advisory role, the solution algorithms do not need to be especially fast; an acceptable time might be a few seconds (for example, when analyzing scenarios with a spreadsheet), or even tens of minutes or hours for very large

Convex Optimization in Signal Processing and Communication, eds. Daniel P. Palomar and Yonina C. Eldar.
Published by Cambridge University Press © Cambridge University Press, 2010.

problems (e.g., engineering design synthesis, or scheduling). Some unreliability in the solution methods can be tolerated, since the human decision maker will review the proposed solutions, and hopefully catch problems.

Much effort has gone into the development of optimization algorithms for these settings. For adequate performance, they must detect and exploit a generic problem structure not known (to the algorithm) until the particular problem instance is solved. A good generic *linear programming* (LP) solver, for example, can solve, on human-based time scales, large problems in digital circuit design, supply chain management, filter design, or automatic control. Such solvers are often coupled with optimization modeling languages, which allow the user to efficiently describe optimization problems in a high level format. This permits the user to rapidly see the effect of new terms or constraints.

This is all based on the conceptual model of a human in the loop, with most previous and current solver development effort focusing on scaling to *large* problem instances. Not much effort, by contrast, goes into developing algorithms that solve small- or medium-sized problems on fast (millisecond or microsecond) time scales, and with great reliability.

1.1.2 Embedded optimization

In this chapter we focus on embedded optimization, where solving optimization problems is part of a wider, automated algorithm. Here the optimization is deeply embedded in the application, and no human is in the loop. In the introduction to the book *Convex Optimization* [1], Boyd and Vandenberghe state:

> A relatively recent phenomenon opens the possibility of many other applications for mathematical optimization. With the proliferation of computers embedded in products, we have seen a rapid growth in *embedded optimization*. In these embedded applications, optimization is used to automatically make real-time choices, and even carry out the associated actions, with no (or little) human intervention or oversight. In some application areas, this blending of traditional automatic control systems and embedded optimization is well under way; in others, it is just starting. Embedded real-time optimization raises some new challenges: in particular, it requires solution methods that are extremely reliable, and solve problems in a predictable amount of time (and memory).

In real-time embedded optimization, different instances of the same small- or medium-size problem must be solved extremely quickly, for example, on millisecond or microsecond time scales; in many cases the result must be obtained before a strict real-time deadline. This is in direct contrast to generic algorithms, which take a variable amount of time, and exit only when a certain precision has been achieved.

An early example of this kind of embedded optimization, though not on the time scales that we envision, is *model predictive control* (MPC), a form of feedback control system. Traditional (but still widely used) control schemes have relatively simple control policies, requiring only a few basic operations like matrix-vector multiplies and lookup table searches at each time step [2, 3]. This allows traditional control policies to be executed rapidly, with strict time constraints and high reliability. While the control policies themselves are simple, great effort is expended in developing and tuning (i.e., choosing parameters in) them. By contrast, with MPC, at each step the control action is determined by solving an optimization problem, typically a (convex) *quadratic program* (QP). It was

first deployed in the late 1980s in the chemical process industry, where the hard real-time deadlines were in the order of 15 minutes to an hour per optimization problem [4]. Since then, we have seen huge computer processing power increases, as well as substantial advances in algorithms, which allow MPC to be carried out on the same fast time scales as many conventional control methods [5, 6]. Still, MPC is generally not considered by most control engineers, even though there is much evidence that MPC provides better control performance than conventional algorithms, especially when the control inputs are constrained.

Another example of embedded optimization is program or algorithmic trading, in which computers initiate stock trades without human intervention. While it is hard to find out what is used in practice due to trade secrets, we can assume that at least some of these algorithms involve the repeated solution of linear or quadratic programs, on short, if not sub-second, time scales. The trading algorithms that run on faster time scales are presumably just like those used in automatic control; in other words, simple and quickly executable. As with traditional automatic control, huge design effort is expended to develop and tune the algorithms.

In signal processing, an algorithm is used to extract some desired signal or information from a received noisy or corrupted signal. In *off-line signal processing*, the entire noisy signal is available, and while faster processing is better, there are no hard real-time deadlines. This is the case, for example, in the restoration of audio from wax cylinder recordings, image enhancement, or geophysics inversion problems, where optimization is already widely used. In *on-line* or *real-time signal processing*, the data signal samples arrive continuously, typically at regular time intervals, and the results must be computed within some fixed time (typically, a fixed number of samples). In these applications, the algorithms in use, like those in traditional control, are still relatively simple [7].

Another relevant field is communications. Here a noise-corrupted signal is received, and a decision as to which bit string was transmitted (i.e., the decoding) must be made within some fixed (and often small) period of time. Typical algorithms are simple, and hence fast. Recent theoretical studies suggest that decoding methods based on convex optimization can deliver improved performance [8–11], but the standard methods for these problems are too slow for most practical applications. One approach has been the development of custom solvers for communications decoding, which can execute far faster than generic methods [12].

We also envisage real-time optimization being used in statistics and machine learning. At the moment, most statistical analysis has a human in the loop. But we are starting to see some real-time applications, e.g., spam filtering, web search, and automatic fault detection. Optimization techniques, such as support vector machines (SVMs), are heavily used in such applications, but much like in traditional control design, the optimization problems are solved on long time scales to produce a set of model parameters or weights. These parameters are then used in the real-time algorithm, which typically involves not much more than computing a weighted sum of features, and so can be done quickly. We can imagine applications where the weights are updated rapidly, using some real-time, optimization-based method. Another setting in which an optimization problem might be solved on a fast time scale is real-time statistical inference, in which estimates of the

probabilities of unknown variables are formed soon after new information (in the form of some known variables) arrives.

Finally, we note that the ideas behind real-time embedded optimization could also be useful in more conventional situations with no real-time deadlines. The ability to extremely rapidly solve problem instances from a specific problem family gives us the ability to solve large numbers of similar problem instances quickly. Some example uses of this are listed below.

- *Trade-off analysis.* An engineer formulating a design problem as an optimization problem solves a large number of instances of the problem, while varying the constraints, to obtain a sampling of the optimal trade-off surface. This provides useful design guidelines.
- *Global optimization.* A combinatorial optimization problem is solved using branch-and-bound or a similar global optimization method. Such methods require the solution of a large number of problem instances from a (typically convex, often LP) problem family. Being able to solve each instance very quickly makes it possible to solve the overall problem much faster.
- *Monte Carlo performance analysis.* With Monte Carlo simulation, we can find the distribution of minimum cost of an optimization problem that depends on some random parameters. These parameters (e.g., prices of some resources or demands for products) are random with some given distribution, but will be known before the optimization is carried out. To find the distribution of optimized costs, we use Monte Carlo: we generate a large number of samples of the price vector (say), and for each one we carry out optimization to find the minimal cost. Here, too, we end up solving a large number of instances of a given problem family.

1.1.3 Convex optimization

Convex optimization has many advantages over general nonlinear optimization, such as the existence of efficient algorithms that can reliably find a globally optimal solution. A less appreciated advantage is that algorithms for specific convex optimization problem families can be highly robust and reliable; unlike many general purpose optimization algorithms, they do not have parameters that must be manually tuned for particular problem instances. Convex optimization problems are, therefore, ideally suited to real-time embedded applications, because they can be reliably solved.

A large number of problems arising in application areas like signal processing, control, finance, statistics and machine learning, and network operation can be cast (exactly, or with reasonable approximations) as convex problems. In many other problems, convex optimization can provide a good heuristic for approximate solution of the problem [13, 14].

In any case, much of what we say in this chapter carries over to local optimization methods for nonconvex problems, although without the global optimality guarantee, and with some loss in reliability. Even simple methods of extending the methods of convex optimization can work very well in pratice. For example, we can use a basic interior-point