
# THÈSE

En vue de l'obtention du

## DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

**Délivré par :**
Université Toulouse III - Paul Sabatier (UPS)

**Spécialité :**
Mathématiques appliquées

---

**Présentée et soutenue par :**

ĐÀO Ngọc Minh

le 12 septembre 2014

# Nonsmooth optimization techniques with applications in automatic control and contact mechanics

# Techniques d'optimisation non lisse avec des applications en automatique et en mécanique des contacts

# Kỹ thuật tối ưu không trơn với áp dụng trong điều khiển tự động và cơ học tiếp xúc

---

**École doctorale :**
Mathématiques, Informatique et Télécommunications de Toulouse (MITT) - ED 475

**Unité de recherche :**
Institut de Mathématiques de Toulouse (IMT) - UMR CNRS 5219

**Directeur de thèse :**

| | |
|---|---|
| NOLL Dominikus, | Université Toulouse III - Paul Sabatier |

**Rapporteurs :**

| | |
|---|---|
| ADLY Samir, | Université de Limoges |
| ZASADZINSKI Michel, | Université de Lorraine |

**Jury :**

| | |
|---|---|
| ADLY Samir, | Université de Limoges |
| APKARIAN Pierre, | ONERA - Centre de Toulouse |
| NOLL Dominikus, | Université Toulouse III - Paul Sabatier |
| ZASADZINSKI Michel, | Université de Lorraine |

*For my dear family!*

*Cho gia đình thân yêu của tôi!*

## Summary

Nonsmooth optimization is an active branch of modern nonlinear programming, where objective and constraints are continuous but not necessarily differentiable functions. Generalized subgradients are available as a substitute for the missing derivative information, and are used within the framework of descent algorithms to approximate local optimal solutions. Under practically realistic hypotheses we prove convergence certificates to local optima or critical points from an arbitrary starting point.

In this thesis we develop especially nonsmooth optimization techniques of bundle type, where the challenge is to prove convergence certificates without convexity hypotheses. Satisfactory results are obtained for two important classes of nonsmooth functions in applications, lower- and upper-$C^1$ functions.

Our methods are applied to design problems in control system theory and in unilateral contact mechanics and in particular, in destructive mechanical testing for delamination of composite materials. We show how these fields lead to typical nonsmooth optimization problems, and we develop bundle algorithms suited to address these problems successfully.

**Keywords.** Nonconvex and nonsmooth optimization · bundle method · Hankel norm · optimal control · eigenstructure assignment · delamination problem.

# Tóm tắt

Tối ưu không trơn là một lĩnh vực năng động của quy hoạch phi tuyến hiện đại, trong đó các hàm mục tiêu và ràng buộc liên tục nhưng không nhất thiết khả vi. Để thay thế cho những thông tin đạo hàm còn thiếu, dưới gradient suy rộng đã xuất hiện và được sử dụng trong khuôn khổ các thuật toán giảm nhằm xấp xỉ các nghiệm tối ưu địa phương. Với những giả thiết thực tế trong vận dụng, chúng tôi chứng minh sự hội tụ của thuật toán đến các điểm tối ưu địa phương hoặc tới hạn từ một điểm khởi tạo bất kì.

Trong luận án này, chúng tôi tập trung phát triển những kỹ thuật tối ưu không trơn dạng bó với yêu cầu đặt ra là chứng minh sự hội tụ không sử dụng tính lồi. Những kết quả thỏa dụng đạt được cho hai lớp hàm không trơn quan trọng trong ứng dụng, đó là các hàm $C^1$-dưới và $C^1$-trên.

Các phương pháp của chúng tôi được áp dụng cho những bài toán thiết kế trong lý thuyết hệ thống điều khiển và cơ học tiếp xúc một phía, đặc biệt là trong thử nghiệm cơ học phá hủy cho sự tách lớp vật liệu composite. Chúng tôi chuyển các vấn đề này về những bài toán tối ưu không trơn điển hình rồi phát triển những thuật toán bó phù hợp để giải quyết chúng một cách hiệu quả.

**Từ khóa.** Tối ưu không trơn không lồi · thuật toán bó · chuẩn Hankel · điều khiển tối ưu · gán cấu trúc riêng · bài toán tách lớp.

# Résumé

L'optimisation non lisse est une branche active de programmation non linéaire moderne, où l'objectif et les contraintes sont des fonctions continues mais pas nécessairement différentiables. Les sous-gradients généralisés sont disponibles comme un substitut à l'information dérivée manquante, et sont utilisés dans le cadre des algorithmes de descente pour se rapprocher des solutions optimales locales. Sous des hypothèses réalistes en pratique, nous prouvons des certificats de convergence vers les points optimums locaux ou critiques à partir d'un point de départ arbitraire.

Dans cette thèse, nous développons plus particulièrement des techniques d'optimisation non lisse de type faisceaux, où le défi consiste à prouver des certificats de convergence sans hypothèse de convexité. Des résultats satisfaisants sont obtenus pour les deux classes importantes de fonctions non lisses dans des applications, fonctions $C^1$-inférieurement et $C^1$-supérieurement.

Nos méthodes sont appliquées à des problèmes de design dans la théorie du système de contrôle et dans la mécanique de contact unilatéral et en particulier, dans les essais mécaniques destructifs pour la délaminage des matériaux composites. Nous montrons comment ces domaines conduisent à des problèmes d'optimisation non lisse typiques, et nous développons des algorithmes de faisceaux appropriés pour traiter ces problèmes avec succès.

**Mots-clés.** Optimisation non lisse et non convexe · méthode de faisceaux · norme de Hankel · contrôle optimal · placement de structure propre · problème de délaminage.

# Acknowledgments

I would like to express my deeply gratitude to my advisor Dominikus Noll for his guidance, support and patience over the years. I feel fortunate to have had the opportunity to work with him and thankful for all the doors he opened for me.

It is a pleasure for me to thank to my two referees Samir Adly and Michel Zasadzinski for their detailed review and comments. My sincere thanks are due to Pierre Apkarian for his friendly discussions and fruitful collaborative work. I am also honoured that he agreed to be a member of the committee.

I wish to acknowledge my other co-authors Joachim Gwinner and Nina Ovcharova on works that have contributed to the thesis. These contributions have been and continue to be crucial.

This work was financially supported by the Vietnamese Government through the 322 project, to which I am grateful. I also would like to give thanks to my colleagues in Hanoi National University of Education who have supported me in my work.

Many thanks to all the members at the Institute of Mathematics of Toulouse for giving me the opportunity to learn and study in an academic environment. I specially appreciate the help from Stanislas, Fabien, Marion, Rémi, Mathieu, Anne-Charline, Amira, and Elissar.

I wish to thank all my Vietnamese friends in Toulouse, especially, Mrs. Châu, Mr. Thanh–Mrs. Céline, Mr. Zũng, Minh–Hà, Tùng, Mạnh, Giang, Bình, Phong, Chinh, Minh, Hùng–Yến, Sơn, Trang, Tuấn–Lan, An–Mai Anh, Long–Hoa, Hòa–Nhi for their attention and help.

I am greatly indebted to all my teachers who have helped me directly and indirectly to develop my knowledge and understanding.

My deepest gratitude and love belong to my whole family for their endless love and unconditional support. The most special thanks goes to my friend, colleague and wife Liên for loving me, listenning me, encouraging me, standing by me and sharing with me through the ups and downs of life. This thesis is a dedication to my little son Paul. He is my biggest source of inspiration and motivation.

# Contents

# List of Figures

**IV Robust eigenstructure clustering by nonsmooth optimization**



**V Nonconvex bundle method with application to a delamination problem**

# List of Tables

# Introduction

Optimization is a key technique in various fields of science and engineering such as mathematics [4], mechanics [15], physics [6], economics [16], optimal control [13], computational chemistry and biology [5]. Most optimization problems in real-life applications do not have explicit solutions and numerical optimization techniques have to be developed to approximate local optimal solutions numerically. We expect such an iterative procedure to converge to a local solution when started at an arbitrary initial guess.

Mathematically, a general optimization problem involves minimizing a function, possibly subject to constraints imposed on the variables of the function. It may be formulated as

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in C \end{array} \tag{1}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ is continuous, and the constraint set (also called the feasible set) $C$ is closed in $\mathbb{R}^n$. Notice that maximization problems can be transformed to minimization problems by reversing the sign of the objective function. If $C = \mathbb{R}^n$ then (1) is called an unconstrained optimization problem. Otherwise, (1) is called a constrained optimization problem, where the constraint set $C$ could for instance be given by linear and nonlinear inequalities, such as $Ax \leqslant b$, $h(x) \leqslant 0$ with $A, b$ given matrix and vector, and $h : \mathbb{R}^n \to \mathbb{R}$ a nonlinear function. Here boundary constraints are included in linear inequalities and equality constraints may be regarded as inequalities.

In this work we are particularly interested in nonsmooth optimization, where the objective function or the constraints are no longer differentiable, but have weaker properties like local Lipschitz continuity. This allows to replace the missing derivative information by generalized subgradients in the sense of Clarke, and to use these elements in a descent algorithm. Following [3, Theorem 1], a necessary condition for $x$ to be a solution of (1) is that

$$0 \in \partial f(x) + N_C(x), \tag{2}$$

where $\partial f(x)$ denotes the Clarke subdifferential of $f$ at $x$, and $N_C(x)$ stands for the (generalized) normal cone to $C$ at $x$. For unconstrained optimization problems, the

optimality condition (2) is reduced to $0 \in \partial f(x)$. It is reasonable to seek for points $x^*$ satisfying (2), called critical points. The purpose of numerical methods is therefore to approximate the solution of problem (1) by generating a sequence $x^j$ of estimates converging to a critical point $x^*$ in a suitable sense. Starting with an initial guess for the solution, numerical methods for solving problem (1) usually provide a search direction and a step size at each iteration in order to move the approximate point from the current position $x^j$ to a new position $x^{j+1}$. Basically, these methods can be classified in two main groups, namely, subgradient methods [18, 1] and bundle methods [12, 9, 13]. While the first ones require only one arbitrary subgradient of the objective function at each iteration, the latter ones approximate the whole subdifferential and involve a quadratic subproblem for finding search directions and step sizes.

At the current time, bundle methods and their variations are known to be among the most efficient optimization methods for nonsmooth problems. Initially proposed by Lemaréchal [11] and Wolfe [20], these methods accumulate subgradients from past iterations into a bundle in order to perform a quadratic tangent program based on the stored information for generating a trial step which is then a serious step if the function value is improved or a null step otherwise. Subsequently, based on the classical cutting plane methods due to Cheney and Goldstein [2] and to Kelley [8], Kiwiel [9] introduced an approach to the bundle methods which builds a convex piecewise linear approximation of the objective function using the linearizations engendered by subgradients. In his work, Kiwiel also used subgradient selection and aggregation techniques to restrict the number of cumulated subgradients. Nevertheless, cutting plane methods [2, 8] and their inherited bundle methods [9, 10, 21, 7] both use cutting planes to form the lower approximation of the objective function, and this is only guaranteed in the convex case.

Expanding on the nonconvex case, Mifflin [14] presented a bundle method using the so-called downshift technique to solve the nonsmooth optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) \leqslant 0 \end{array}$$

where the functions $f$ and $h$ are real-valued locally Lipschitz but not necessarily convex on $\mathbb{R}^n$. Developing this problem in the direction of adding linear constraints, Mäkelä and Neittaanmäki [13] proposed a proximal bundle method dealing with the constraints due to the improvement function

$$F(y, x) = \max\{f(y) - f(x), h(y)\}.$$

As these approaches rely on line search techniques, they only provide weak convergence certificates where at best one of the accumulation points of the sequence $x^j$ of serious iterates is critical.

In this thesis we strive at better certificates in the sense that *every* accumulation point $x^*$ of the sequence $x^j$ is critical. To achieve this, we use a nonconvex bundle technique in tandem with proximity control as a backtracking mechanism We consider a more general constrained optimization problem of the form

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) \leqslant 0 \\ & x \in \mathbf{C} \end{array} \tag{3}$$

where the functions $f$ and $h$ are real-valued locally Lipschitz but not necessarily smooth or convex on $\mathbb{R}^n$, and the set $\mathbf{C}$ is closed convex in $\mathbb{R}^n$. Note that this formulation also covers the case of multiple constraints $h_i(x) \leqslant 0$, $i = 1, \ldots, m$ by simply taking $h(x)$ as the pointwise maximum of the $h_i(x)$. Typically, additional linear constraints can be included in $\mathbf{C}$ due to the convexity of their solution set. To solve this problem, we suggest a nonconvex bundle method using downshifted tangents and a proximity control management, which gives a strong convergence certificate for both nonsmoothness classes of lower-$C^1$ and upper-$C^1$ types in the sense of [19, 17].

Our methods are applied to design problems in control system theory and in unilateral contact mechanics and in particular, in destructive mechanical testing for delamination of composite materials. We show how these fields lead to typical nonsmooth optimization problems, and we develop bundle algorithms suited to address these problems successfully.

The rest of the thesis contains five chapters that correspond to the following five contributions.

I. M. N. Dao, *Bundle method for nonconvex nonsmooth constrained optimization.*
   We develop a nonconvex bundle method based on the downshift mechanism and a proximity control management technique to solve nonconvex nonsmooth constrained optimization problems. The global convergence of the algorithm in the sense of subsequences is proved for both classes of lower-$C^1$ and upper-$C^1$ functions.

II. M. N. Dao and D. Noll, *Minimizing memory effects of a system.*
   Given a stable linear time-invariant system with tunable parameters, we present a method to tune these parameters in such a way that undesirable responses of the system to past excitations, known as system ringing, are avoided or reduced. This problem is addressed by minimizing the Hankel norm of the system, which quantifies the influence of past inputs on future outputs. We indicate by way of examples that minimizing the Hankel norm has a wide scope for possible applications. We show that the Hankel norm minimization program may be cast as an eigenvalue optimization problem, which we solve by a nonsmooth bundle algorithm with a local convergence certificate. Numerical experiments are used to demonstrate the efficiency of our approach.

III. M. N. Dao and D. Noll, *Simultaneous plant and controller optimization based on nonsmooth techniques.*
   We present an approach to simultaneous design optimization of a plant and its controller. This is based on a bundling technique for solving nonsmooth optimization problems under nonlinear and linear constraints. In the absence of convexity, a substitute for the convex cutting plane mechanism is proposed. The method is illustrated on a problem of steady flow in a graph and in robust feedback control design of a mass-spring-damper system.

IV. M. N. Dao, D. Noll, and P. Apkarian, *Robust eigenstructure clustering by nonsmooth optimization.*
   We extend classical eigenstructure assignment to more realistic problems

where additional performance and robustness specifications arise. Our aim is to combine time-domain constraints, as reflected by pole location and eigenvector structure, with frequency-domain objectives such as the $H_2$, $H_\infty$ or Hankel norms. Using pole clustering, we allow poles to move in polydisks of prescribed size around their nominal values, driven by optimization. Eigenelements, that is poles and eigenvectors, are allowed to move simultaneously and serve as decision variables in a specialized nonsmooth optimization technique. Two aerospace applications illustrate the power of the new method.

V. M. N. Dao, J. Gwinner, D. Noll, and N. Ovcharova, *Nonconvex bundle method with application to a delamination problem.*
Delamination is a typical failure mode of composite materials caused by weak bonding. It arises when a crack initiates and propagates under a destructive loading. Given the physical law characterizing the properties of the interlayer adhesive between the bonded bodies, we consider the problem of computing the propagation of the crack front and the stress field along the contact boundary. This leads to a hemivariational inequality, which after discretization by finite elements we solve by a nonconvex bundle method, where upper-$C^1$ criteria have to be minimized. As this is in contrast with other classes of mechanical problems with non-monotone friction laws and in other applied fields, where criteria are typically lower-$C^1$, we propose a bundle method suited for both types of nonsmoothness. We prove its global convergence in the sense of subsequences and test it on a typical delamination problem of material sciences.

## References


1. D. P. Bertsekas, *Nonlinear programming*, Athena Scientific, Belmont, 1999.
2. E. W. Cheney and A. A. Goldstein, *Newton's method for convex programming and Tchebycheff approximation*, Numer. Math. **1** (1959), 253–268.
3. F. H. Clarke, *A new approach to Lagrange multipliers*, Math. Oper. Res. **1** (1976), no. 2, 165–174.
4. ———, *Optimization and nonsmooth analysis*, Canad. Math. Soc. Ser. Monogr. Adv. Texts, John Wiley & Sons, Inc., New York, 1983.
5. C. A. Floudas and P. M. Pardalos, *Optimization in computational chemistry and molecular biology. Local and global approaches*, Nonconvex Optim. Appl., vol. 40, Kluwer Academic Publishers, Dordrecht, 2000.
6. A. K. Hartmann and H. Rieger, *Optimization algorithms in physics*, Wiley-VCH Verlag Berlin GmbH, Berlin, 2002.
7. J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms. II. Advanced theory and bundle methods*, Grundlehren Math. Wiss., vol. 306, Springer-Verlag, Berlin, 1993.
8. J. E. Kelley, *The cutting-plane method for solving convex programs*, J. SIAM **8** (1960), no. 4, 703–712.
9. K. C. Kiwiel, *Methods of descent for nondifferentiable optimization*, Lecture Notes in Math., vol. 1133, Springer-Verlag, Berlin, 1985.
10. ———, *Proximity control in bundle methods for convex nondifferentiable minimization*, Math. Programming, Ser. A **46** (1990), no. 1, 105–122.
11. C. Lemaréchal, *An extension of Davidon methods to non differentiable problems*, Nondifferentiable Optimization (M. L. Balinski and P. Wolfe, eds.), Math. Programming Stud., vol. 3, North-Holland Publishing Co., Amsterdam, 1975, pp. 95–109.

12. _____, *Nondifferentiable optimization*, Nonlinear Optimization (Proc. Internat. Summer School, Univ. Bergamo, Bergamo, 1979), Birkhäuser, Boston, 1980, pp. 149–199.
13. M. M. Mäkelä and P. Neittaanmäki, *Nonsmooth optimization: Analysis and algorithms with applications to optimal control*, World Scientific Publishing Co., Singapore, 1992.
14. R. Mifflin, *A modification and extension of Lemarechal's algorithm for nonsmooth minimization*, Nondifferential and Variational Techniques in Optimization (D. C. Sorensen and R. J.-B. Wets, eds.), Math. Programming Stud., vol. 17, North-Holland Publishing Co., Amsterdam, 1982, pp. 77–90.
15. J.-J. Moreau, P. D. Panagiotopoulos, and G. Strang, *Topics in nonsmooth mechanics*, Birkhäuser Verlag, Basel, 1988.
16. J. Outrata, M. Kočvara, and J. Zowe, *Nonsmooth approach to optimization problems with equilibrium constraints. Theory, applications and numerical results*, Nonconvex Optim. Appl., vol. 28, Kluwer Academic Publishers, Dordrecht, 1998.
17. R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, Springer-Verlag, Berlin, 1998.
18. N. Z. Shor, *Minimization methods for non-differentiable functions*, Springer Ser. Comput. Math., vol. 3, Springer-Verlag, Berlin, 1985, Translated from the Russian by K. C. Kiwiel and A. Ruszczyński.
19. J. E. Spingarn, *Submonotone subdifferentials of Lipschitz functions*, Trans. Amer. Math. Soc. **264** (1981), no. 1, 77–89.
20. P. Wolfe, *A method of conjugate subgradients for minimizing nondifferentiable functions*, Nondifferentiable Optimization (M. L. Balinski and P. Wolfe, eds.), Math. Programming Stud., vol. 3, North-Holland Publishing Co., Amsterdam, 1975, pp. 145–173.
21. J. Zowe, *Nondifferentiable optimization*, Computational Mathematical Programming (K. Schittkowski, ed.), NATO Adv. Sci. Inst. Ser. F Comput. Systems Sci., vol. 15, Springer, Berlin, 1985, pp. 323–356.

# I

## Bundle method for nonconvex nonsmooth constrained optimization *

Minh Ngoc Dao

**Abstract.** The paper develops a nonconvex bundle method based on the downshift mechanism and a proximity control management technique to solve nonconvex nonsmooth constrained optimization problems. We prove its global convergence in the sense of subsequences for both classes of lower-$C^1$ and upper-$C^1$ functions.

**Keywords.** Nonsmooth optimization · constrained optimization · bundle method · lower-$C^1$ function· upper-$C^1$ function.

### 1. Introduction

Nonsmooth optimization problems appear frequently in practical applications such as economics, mechanics, and control theory. There are several methods for solving nonsmooth optimization problems, and they can be divided into two main groups: subgradient methods and bundle methods. We want to mention the latter ones because of their proven efficiency in solving relevant problems. Bundle methods were first introduced by Lemaréchal [12] and have been developed over the years based on subsequent works of Kiwiel [9], Lemaréchal, Nemirovskii, and Nesterov [13]. The main idea of bundle methods is to estimate the Clarke subdifferential [3] of the objective function by accumulating subgradients from past iterations into a bundle, and then to generate a trial step by a quadratic tangent program using information stored in the bundle.

Extending Lemarechal's algorithm to the nonconvex case, Mifflin [16] gives a bundle algorithm using the so-called downshift mechanism for the nonsmooth minimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) \leqslant 0 \end{array}$$

*Paper submitted for publication.

where $f$ and $h$ are real-valued locally Lipschitz but potentially nonconvex functions on $\mathbb{R}^n$. Subsequently, Mäkelä and Neittaanmäki [14] present a proximal bundle method for the above problem adding linear constraints. This method uses the improvement function

$$F(y,x) = \max\{f(y) - f(x), h(y)\}$$

for the handling of the constraints. While these works use a line search procedure which admits only weak convergence certificates in the sense that at least one of the accumulation points of the sequence of serious iterates is critical, we are interested in using a proximity control mechanism along with a suitable backtracking strategy. This brings to stronger convergence certificates, where every accumulation point of the sequence of serious iterates is critical. Recently, Gabarrou, Alazard and Noll [7] showed a strong convergence for the case where $f$ and $h$ are lower-$C^1$ functions in the sense of [23, 22]. However, a convergence proof for upper-$C^1$ functions still remains open.

In present framework we consider a more general constrained optimization problem of the form

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) \leqslant 0 \\ & x \in \mathbf{C} \end{array} \tag{1}$$

where $f$ and $h$ are real-valued locally Lipschitz but potentially nonsmooth and nonconvex functions, and where $\mathbf{C}$ is a closed convex set of $\mathbb{R}^n$. For solving this problem, we propose a nonconvex bundle method based on downshifted tangents and a proximity control management mechanism, in which a strong convergence certificate is valid for both classes of lower-$C^1$ and upper-$C^1$ functions.

The motivation of this paper rises from the fact that many application problems are addressed by minimizing lower-$C^1$ functions. For instance, some problems in the context of automatic control are quite successfully solved in [19, 17, 18, 7, 5] by applying bundling techniques to lower-$C^1$ functions. In particular, the problem of maximizing the memory of a system [5] can be reformulated as minimizing upper-$C^1$ functions.

The rest of the paper is organized as follows. Sections 2–5 present elements of the proximity control algorithm. In section 6 we introduce a theoretical tool in the convergence proof which is referred to as the upper envelope model. Some preparatory information on semismooth, lower-$C^1$ and upper-$C^1$ functions is given in section 7. The central sections 8, 9 prove global convergence of the algorithm.

## 2. Progress function

Following an idea of Polak in [21, Section 2.2.2], to solve problem (1) we use the progress function

$$F(y,x) = \max\{f(y) - f(x) - \mu h(x)_+, h(y) - h(x)_+\},$$

with $\mu > 0$ a fixed parameter and $h(x)_+ = \max\{h(x), 0\}$. Here $x$ represents the current iterate, and $y$ is the next iterate or a candidate for the next iterate.

Let $\partial f(x)$ denote the Clarke subdifferential of $f$ at $x$. For functions of two variables, the notation $\partial_1$ stands for the Clarke subdifferential with respect to the first variable. We first remark that $F(x,x)=0$. Moreover, $F(\cdot,x)$ is also locally Lipschitz, and by [4, Proposition 2.3.12] (see also [2, Proposition 9]),

$$\begin{cases} \partial_1 F(x,x) = \partial f(x) & \text{if } h(x)<0,\\ \partial_1 F(x,x) \subset \operatorname{conv}\{\partial f(x)\cup \partial h(x)\} & \text{if } h(x)=0,\\ \partial_1 F(x,x) = \partial h(x) & \text{if } h(x)>0, \end{cases} \tag{2}$$

where conv signifies convex hull, and where equality holds if $f$ and $h$ are regular at $y$ in the sense of Clarke [3]. Recall that the indicator function of a convex set $\mathbf{C}\subset\mathbb{R}^n$ defined by

$$i_{\mathbf{C}}(x)=\begin{cases}0 & \text{if } x\in\mathbf{C},\\ \infty & \text{otherwise,}\end{cases}$$

we have $i_{\mathbf{C}}(\cdot)$ is a convex function, and $\partial i_{\mathbf{C}}(x)$ is the normal cone to $\mathbf{C}$ at $x$,

$$N_{\mathbf{C}}(x)=\{g\in\mathbb{R}^n : g^\top(y-x)\leqslant 0 \text{ for all } y\in\mathbf{C}\},$$

if $x\in\mathbf{C}$, and the empty set otherwise. It is worth to notice that if $\mathbf{C}$ is a polyhedral set having the form

$$\mathbf{C}=\{x\in\mathbb{R}^n : a_i^\top x\leqslant b_i, i=1,\dots,m\},$$

where $a_i$ and $b_i$ are respectively given vectors and scalars, then

$$\partial i_{\mathbf{C}}(x)=N_{\mathbf{C}}(x)=\{\lambda_1 a_1+\cdots+\lambda_m a_m : \lambda_i\geqslant 0, \lambda_i=0 \text{ if } a_i^\top x<b_i\}$$

for all $x\in\mathbf{C}$ (see [22, Theorem 6.46]). Motivated by [1, Lemma 5.1] and [2, Theorem 1], we now establish the following result.

**Lemma 2.1.** *Let $f$ and $h$ be locally Lipschitz functions, then the following statements hold.*

(i) *If $x^*$ is a local minimum of problem (1), it is also a local minimum of $F(\cdot,x^*)$ in $\mathbf{C}$, and then $0\in\partial_1 F(x^*,x^*)+\partial i_{\mathbf{C}}(x^*)$. Furthermore, if $x^*$ is a F. John critical point of (1) then $0\in\partial_1 F(x^*,x^*)+\partial i_{\mathbf{C}}(x^*)$ in the case where $f$ and $h$ are regular at $x^*$.*

(ii) *Conversely, if $0\in\partial_1 F(x^*,x^*)+\partial i_{\mathbf{C}}(x^*)$ for some $x^*\in\mathbf{C}$ then only one of the following situations occurs.*
   (a) *$h(x^*)>0$, in which case $x^*$ is a critical point of $h$ in $\mathbf{C}$, called a critical point of constraint violation.*
   (b) *$h(x^*)\leqslant 0$, in which case $x^*$ is a F. John critical point of (1). In addition, we have either $h(x^*)=0$ and $0\in\partial h(x^*)+\partial i_{\mathbf{C}}(x^*)$, or $x^*$ is a Karush-Kuhn-Tucker point of (1).*

*Proof.* (i) Let $x^*$ be a local minimum of problem (1), then $h(x^*)\leqslant 0, x\in\mathbf{C}$, which gives $h(x^*)_+=0$, and so

$$F(y,x^*)=\max\{f(y)-f(x^*),h(y)\}.$$

Moreover, there exists a neighborhood $U$ of $x^*$ such that $f(y)\geqslant f(x^*)$ for all $y\in U\cap\mathbf{C}$ satisfying $h(y)\leqslant 0$. We will show that $F(y,x^*)\geqslant F(x^*,x^*)$ for all $y\in U\cap\mathbf{C}$. Indeed, if $h(y)>0$ then

$$F(y,x^*)\geqslant h(y)>0=F(x^*,x^*).$$

If $h(y) \leqslant 0$ then $f(y) \geqslant f(x^*)$, and therefore

$$F(y, x^*) \geqslant f(y) - f(x^*) \geqslant 0 = F(x^*, x^*).$$

This means that $x^*$ is a local minimum of $F(\cdot, x^*)$ in $\mathbf{C}$, which implies $0 \in \partial_1 F(x^*, x^*) + \partial i_{\mathbf{C}}(x^*)$.

Now assume that $x^*$ is a F. John critical point of (1), i.e., there exist constants $\lambda_0, \lambda_1$ such that

$$\begin{aligned} &0 \in \lambda_0 \partial f(x^*) + \lambda_1 \partial h(x^*) + \partial i_{\mathbf{C}}(x^*), \\ &\lambda_0 \geqslant 0, \lambda_1 \geqslant 0,\ \lambda_0 + \lambda_1 = 1, \\ &\lambda_1 h(x^*) = 0. \end{aligned}$$

Then if $h(x^*) < 0$, we have $\lambda_1 = 0$, $\lambda_0 = 1$, and by using (2), $\partial_1 F(x^*, x^*) = \partial f(x^*)$, which implies $0 \in \partial_1 F(x^*, x^*) + \partial i_{\mathbf{C}}(x^*)$. In the case where $f$ and $h$ are regular at $x^*$, if $h(x^*) = 0$ then $\partial_1 F(x^*, x^*) = \operatorname{conv}\{\partial f(x^*) \cup \partial h(x^*)\}$, and thus

$$0 \in \lambda_0 \partial f(x^*) + \lambda_1 \partial h(x^*) + \partial i_{\mathbf{C}}(x^*) \subset \partial_1 F(x^*, x^*) + \partial i_{\mathbf{C}}(x^*).$$

(ii) Suppose that $0 \in \partial_1 F(x^*, x^*) + \partial i_{\mathbf{C}}(x^*)$ for some $x^* \in \mathbf{C}$. Then by (2), there exist constants $\lambda_0, \lambda_1$ such that

$$\begin{aligned} &0 \in \lambda_0 \partial f(x^*) + \lambda_1 \partial h(x^*) + \partial i_{\mathbf{C}}(x^*), \\ &\lambda_0 \geqslant 0, \lambda_1 \geqslant 0,\ \lambda_0 + \lambda_1 = 1. \end{aligned}$$

If $h(x^*) > 0$ then $\partial_1 F(x^*, x^*) = \partial h(x^*)$, and so $0 \in \partial h(x^*) + \partial i_{\mathbf{C}}(x^*)$, that is, $x^*$ is a critical point of $h$ in $\mathbf{C}$.

If $h(x^*) < 0$ then $\partial_1 F(x^*, x^*) = \partial f(x^*)$, which gives $\lambda_1 = 0$, and therefore $x^*$ is a Karush-Kuhn-Tucker point and also a F. John critical point of (1).

In the case of $h(x^*) = 0$, we see immediately that $x^*$ is a F. John critical point of (1). If $x^*$ fails to be a Karush-Kuhn-Tucker point then $\lambda_0 = 0$ and we get $0 \in \partial h(x^*) + \partial i_{\mathbf{C}}(x^*)$. The lemma is proved completely. □

## 3. Tangent program and acceptance test

In accordance with Lemma 2.1, it is reasonable to seek for points $x^*$ satisfying $0 \in \partial_1 F(x^*, x^*) + \partial i_{\mathbf{C}}(x^*)$. We present our nonconvex bundle method for finding solutions of problem (1), which generates a sequence $x^j$ of estimates converging to a solution $x^*$ in the sense of subsequence.

Denote the current iterate of the outer loop by $x$, or $x^j$ if the outer loop counter $j$ is used. When a new iterate of the outer loop is found, it will be denoted by $x^+$, or $x^{j+1}$. At the current iterate $x$, we build first-order working models $\phi_k(\cdot, x)$, which approximates $F(\cdot, x)$ in a neighborhood of $x$. Those are updated iteratively during the inner loop, and have to satisfy the following properties at all times $k$:

- $\phi_k(\cdot, x)$ is convex;
- $\phi_k(x, x) = F(x, x) = 0$ and $\partial_1 \phi_k(x, x) \subset \partial_1 F(x, x)$.

The latter is ensured when the so-called exactness plane $m_0(\cdot, x) = g(x)^\top(\cdot - x)$ with $g(x) \in \partial_1 F(x, x)$ is an affine minorant of $\phi_k(\cdot, x)$. Note that due to (2) we can choose $g(x) \in \partial f(x)$ if $h(x) \leqslant 0$, and $g(x) \in \partial h(x)$ if $h(x) > 0$.

Once the first-order working model $\phi_k(\cdot, x)$ has been decided on, we define an associated second-order working model

$$\Phi_k(\cdot, x) = \phi_k(\cdot, x) + \frac{1}{2}(\cdot - x)^\top Q(x)(\cdot - x),$$

where $Q(x)$ is a symmetric matrix depending only on the current iterate $x$. Now we find a new trial step $y^k$ via the tangent program

$$\begin{array}{ll} \text{minimize} & \Phi_k(y, x) + \frac{\tau_k}{2}\|y - x\|^2 \\ \text{subject to} & y \in \mathbf{C} \end{array} \tag{3}$$

where $\tau_k > 0$ is called the proximity control parameter. Note that this program is strictly convex and has a unique solution as soon as we assure $Q(x) + \tau_k I \succ 0$.

In the sequel, we write $\partial_1(\phi(y, x) + i_{\mathbf{C}}(y))$ for the Clarke subdifferential of $\phi(y, x) + i_{\mathbf{C}}(y)$ with respect to the first variable at $y$. Let us note that $\partial_1(\phi(y, x) + i_{\mathbf{C}}(y)) \subset \partial_1\phi(y, x) + \partial i_{\mathbf{C}}(y)$, and that equality need not hold. The necessary optimality condition for tangent program (3) gives

$$0 \in \partial_1(\phi_k(y^k, x) + i_{\mathbf{C}}(y^k)) + (Q(x) + \tau_k I)(y^k - x).$$

Therefore, if $y^k = x$ then $0 \in \partial_1\phi_k(x, x) + \partial i_{\mathbf{C}}(x)$, and so $0 \in \partial_1 F(x, x) + \partial i_{\mathbf{C}}(x)$ due to the fact that $\partial_1\phi_k(x, x) \subset \partial_1 F(x, x)$. The consequence of this argument is that once $0 \notin \partial_1 F(x, x) + \partial i_{\mathbf{C}}(x)$, the trial step $y^k$ will always bring something new. From this time forth we suppose that $0 \notin \partial_1 F(x, x) + \partial i_{\mathbf{C}}(x)$. Then $y^k \neq x$ is the solution of the tangent program, so $\Phi_k(y^k, x) + \frac{\tau_k}{2}\|y^k - x\|^2 \leqslant \Phi_k(x, x)$, which gives $\Phi_k(y^k, x) < \Phi_k(x, x) = 0$. In other words, there is always a progress predicted by the working model $\Phi_k(\cdot, x)$, unless $x$ is already a critical point of (1) in the sense that $0 \in \partial_1 F(x, x) + \partial i_{\mathbf{C}}(x)$.

Following standard terminology, $y^k$ is called a serious step if it is accepted as the new iterate, and a null step otherwise. In order to decide whether $y^k$ is accepted or not, we compute the test quotient

$$\rho_k = \frac{F(y^k, x)}{\Phi_k(y^k, x)},$$

which measures the agreement between $F(\cdot, x)$ and $\Phi_k(\cdot, x)$ at $y^k$. If the current model $\Phi_k$ represents $F$ precisely at $y^k$, it is awaited that $\rho_k \approx 1$. Fixing a constant $0 < \gamma < 1$, we accept the trial step $y^k$ already as the new serious step $x^+$ if $\rho_k \geqslant \gamma$. Here the inner loop ends. Otherwise $y^k$ is rejected and the inner loop continues.

*Remark* 3.1. If the current iterate $x$ is feasible then the serious step $x^+$ is strictly feasible and $f(x^+) < f(x)$. Indeed, we have $F(x^+, x) = \max\{f(x^+) - f(x), h(x^+)\}$ due to the feasibility of $x$. Assume that the serious step $x^+$ is accepted at inner loop counter $k$, which means $x^+ = y^k \in \mathbf{C}$ with $\rho_k \geqslant \gamma > 0$. Since $x^+ = y^k \neq x$ is the optimal solution of (3), $\Phi_k(x^+, x) + \frac{\tau_k}{2}\|x^+ - x\|^2 \leqslant \Phi_k(x, x) = 0$, and so $\Phi_k(x^+, x) < 0$. This combined with $\rho_k > 0$ gives $F(x^+, x) < 0$, which implies that $f(x^+) < f(x)$ and $h(x^+) < 0$.

## 4. Working model update

Suppose that $y^k$ is a null step, we will improve the next model $\phi_{k+1}(\cdot, x)$. Notice that the exactness plane is always kept in first-order working models. To make $\phi_{k+1}(\cdot, x)$ better than $\phi_k(\cdot, x)$, we need two more elements, referred to as cutting and aggregate planes. Let us first look at the cutting plane generation.

The cutting plane $m_k(\cdot, x)$ is a basic element in bundle methods which cuts away the unsuccessful trial step $y^k$. The idea is to construct $m_k(\cdot, x)$ in the way that $y^k$ is no longer solution of the new tangent program as soon as $m_k(\cdot, x)$ is an affine minorant of $\phi_{k+1}(\cdot, x)$. For each subgradient $g_k \in \partial_1 F(y^k, x)$, the tangent $t_k(\cdot) = F(y^k, x) + g_k^\top(\cdot - y^k)$ to $F(\cdot, x)$ at $y^k$ is used as a cutting plane in the case where $F(\cdot, x)$ is convex. Without convexity, tangent planes may be useless, and a substitute has to be found. We exploit a mechanism first described in [16], which consists in shifting the tangent down until it becomes useful for $\phi_{k+1}(\cdot, x)$. Fixing a parameter $c > 0$ once and for all, we define the downshift as $s_k = [t_k(x) + c\|y^k - x\|^2]_+$, and introduce the cutting plane

$$m_k(\cdot, x) = t_k(\cdot) - s_k = a_k + g_k^\top(\cdot - x),$$

with $a_k = \min\{t_k(x), -c\|y^k - x\|^2\} \leqslant -c\|y^k - x\|^2 < 0$ by the fact that $y^k \neq x$.

*Remark* 4.1. Let $\phi_{k+1}(\cdot, x) = \max\{m_i(\cdot, x) : i = 0, \ldots, k\}$, then $\phi_{k+1}(\cdot, x)$ is convex, and $\phi_{k+1}(x, x) = F(x, x) = 0$, $\partial_1 \phi_{k+1}(x, x) \subset \partial_1 F(x, x)$. Indeed, since $\phi_{k+1}(\cdot, x)$ is a maximum of affine planes, and $m_i(x, x) = a_i < 0 = m_0(x, x)$ for $i \geqslant 1$, we get convexity of $\phi_{k+1}(\cdot, x)$, and also $\phi_{k+1}(x, x) = 0$, $\partial_1 \phi_{k+1}(x, x) = \partial_1 m_0(x, x) = \{g(x)\} \subset \partial_1 F(x, x)$.

Next we see that the optimality condition for (3) can be written as

$$(Q(x) + \tau_k I)(x - y^k) = g_k^* + h_k^*, \text{ for } g_k^* \in \partial_1 \phi_k(y^k, x), h_k^* \in \partial i_{\mathbf{C}}(y^k). \tag{4}$$

If $\phi_k(\cdot, x) = \max\{m_i(\cdot, x) : i = 0, \ldots, r\}$ then there exist $\lambda_0, \ldots, \lambda_r$ are non-negative and sum up to 1 such that

$$g_k^* = \sum_{i=0}^{r} \lambda_i g_i, \ \phi_k(y^k, x) = \sum_{i=0}^{r} \lambda_i m_i(y^k, x).$$

We call $g_k^*$ the aggregate subgradient as traditional, and build the aggregate plane

$$m_k^*(\cdot, x) = a_k^* + g_k^{*\top}(\cdot - x)$$

with $a_k^* = \sum_{i=0}^{r} \lambda_i a_i = \phi_k(y^k, x) + g_k^{*\top}(x - y^k)$. Then $\phi_k(y^k, x) = m_k^*(y^k, x) \leqslant \phi_{k+1}(y^k, x)$ if we require that $m_k^*(\cdot, x)$ is an affine minorant of $\phi_{k+1}(\cdot, x)$. To avoid overflow, when generating the new working model $\phi_{k+1}(\cdot, x)$, we may replace all older planes corresponding to $\lambda_i > 0$ by the aggregate plane. This construction follows the original lines as proposed in [9]. It does not change the conclusion of Remark 4.1, nor the definition of aggregate planes.

*Remark* 4.2. Typically, the new working model $\phi_{k+1}(\cdot, x)$ can be given by

$$\phi_{k+1}(\cdot, x) = \max\{m_0(\cdot, x), m_k(\cdot, x), m_k^*(\cdot, x)\},$$

which satisfies the required properties of a first-order working model.

As we pass from $x$ to a new serious step $x^+$, the planes $m(\cdot, x) = a + g^\top(\cdot - x)$ from previous serious steps may become useless at $x^+$ since we have no guarantee that $m(x^+, x) \leqslant F(x^+, x^+) = 0$. But we can recycle the old planes by using again the downshift mechanism as

$$m(\cdot, x^+) = m(\cdot, x) - s^+, \; s^+ = [m(x^+, x) + c\|x^+ - x\|^2]_+.$$

For more details, we refer to [17].

## 5. Proximity control management

The management of the proximity control parameter $\tau_k$ is a major difference between the convex and nonconvex bundle methods. In the convex case, the proximity control can remain unchanged during the inner loop. In the absence of convexity, the parameter $\tau_k$ has to follow certain basic rules to assure convergence of the algorithm. The central rule which we have to respect is that during the inner loop, the parameter may only increase infinitely often due to the strong discrepancy between the current working model $\phi_k$ and the best possible working model. Assuming the trial step $y^k$ is a null step, as a means to decide when to increase $\tau_k$ or not, we compute the secondary test

$$\tilde{\rho}_k = \frac{M_k(y^k, x)}{\Phi_k(y^k, x)},$$

where $M_k(\cdot, x) = \max\{m_0(\cdot, x), m_k(\cdot, x)\} + \frac{1}{2}(\cdot - x)^\top Q(x)(\cdot - x)$ with $m_0(\cdot, x)$ the exactness plane at the current iterate $x$, and $m_k(\cdot, x)$ the cutting plane at $x$ and $y^k$. If $\tilde{\rho}_k \approx 1$ which indicates that little to no progress is achieved by adding the cutting plane, the proximity control must be increased to force smaller steps. In the case where $\tilde{\rho}_k$ is too far from 1, we hope that the situation will be improved without having to increase the proximity control. Fixing parameters $\tilde{\gamma}$ and $\theta$ with $0 < \gamma < \tilde{\gamma} < 1 < \theta < \infty$, we make the following decision

$$\tau_{k+1} = \begin{cases} \tau_k & \text{if } \tilde{\rho}_k < \tilde{\gamma}, \\ \theta\tau_k & \text{if } \tilde{\rho}_k \geqslant \tilde{\gamma}. \end{cases}$$

Let us next consider the management of the proximity parameter between serious steps $x \to x^+$, respectively, $x^j \to x^{j+1}$. To do this we use a memory element $\tau_j^\sharp$, which is computed as soon as a serious step is made. Suppose that the serious step $x^{j+1}$ is achieved at inner loop counter $k_j$, that is $x^{j+1} = y^{k_j}$ with $\rho_{k_j} \geqslant \gamma$. We consider the test

$$\rho_{k_j} = \frac{F(y^{k_j}, x^j)}{\Phi_{k_j}(y^{k_j}, x^j)} \overset{?}{\geqslant} \Gamma,$$

where $0 < \gamma < \Gamma < 1$ is fixed throughout. If $\rho_{k_j} < \Gamma$ then we memorize the last parameter used, that means $\tau_{j+1}^\sharp = \tau_{k_j}$. On the other hand, if $\rho_{k_j} \geqslant \Gamma$ then we may trust the model and store $\tau_{j+1}^\sharp = \theta^{-1}\tau_{k_j} < \tau_{k_j}$. At the first inner loop of the $j$th outer loop, the memory element $\tau_j^\sharp$ serves to initialize $\tau_1 = \max\{\tau_j^\sharp, -\lambda_{\min}(Q_j) + \kappa\}$ or $\tau_1 = T > q + \kappa$ with $\lambda_{\min}(\cdot)$ the minimum eigenvalue of a symmetric matrix, and $0 < \kappa \ll 1$ fixed, which assures always that $Q_j + \tau_k I \succ 0$ during the $j$th outer loop.

Figure 1 shows a flowchart of the algorithm, while the detailed statement is presented as Algorithm 1.

**Algorithm 1.** Proximity control algorithm with downshifted tangents

**Parameters:** $0 < \gamma < \tilde{\gamma} < 1, 0 < \gamma < \Gamma < 1, 1 < \theta < \infty, 0 < \kappa \ll 1, 0 < q < \infty$, $q + \kappa < T < \infty$.

▷ **Step 1** (`Outer loop initialization`). Choose initial feasible guess $x^1$, fix memory control parameter $\tau_1^\sharp$, and put outer loop counter $j = 1$.

◇ **Step 2** (`Stopping test`). At outer loop counter $j$, stop the algorithm if $0 \in \partial_1 F(x^j, x^j) + \partial i_{\mathbf{C}}(x^j)$. Otherwise, take a symmetric matrix $Q_j$ respecting $-qI \preceq Q_j \preceq qI$, and goto inner loop.

▷ **Step 3** (`Inner loop initialization`). Put inner loop counter $k = 1$ and initialize control parameter $\tau_1 = \max\{\tau_j^\sharp, -\lambda_{\min}(Q_j) + \kappa\}$, where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of a symmetric matrix. Reset $\tau_1 = T$ if $\tau_1 > T$, and choose initial working model $\phi_1(\cdot, x^j)$ using the exactness plane $m_0(\cdot, x^j)$ and possibly recycling some planes from previous loop.

▷ **Step 4** (`Tangent program`). At inner loop counter $k$, let

$$\Phi_k(\cdot, x^j) = \phi_k(\cdot, x^j) + \frac{1}{2}(\cdot - x^j)^\top Q_j(\cdot - x^j)$$

and find solution $y^k$ (`trial step`) of the tangent program

$$\begin{array}{ll} \text{minimize} & \Phi_k(y, x) + \frac{\tau_k}{2}\|y - x\|^2 \\ \text{subject to} & y \in \mathbf{C}. \end{array}$$

◇ **Step 5** (`Acceptance test`). Compute the quotient

$$\rho_k = \frac{F(y^k, x^j)}{\Phi_k(y^k, x^j)}.$$

If $\rho_k \geqslant \gamma$ (`serious step`), put $x^{j+1} = y^k$, compute new memory element

$$\tau_{j+1}^\sharp = \begin{cases} \tau_k & \text{if } \rho_k < \Gamma, \\ \theta^{-1}\tau_k & \text{if } \rho_k \geqslant \Gamma, \end{cases}$$

and quit inner loop. Increase outer loop counter $j$ and loop back to step 2. If $\rho_k < \gamma$ (`null step`), continue inner loop with step 6.

▷ **Step 6** (`Working model update`). Generate a cutting plane $m_k(\cdot, x^j)$ at null step $y^k$ and counter $k$ using downshifted tangents. Compute aggregate plane $m_k^*(\cdot, x^j)$ at $y^k$, and then build a new working model $\phi_{k+1}(\cdot, x^j)$ by adding the new cutting plane, keeping the exactness plane and using aggregation to avoid overflow.

◇ **Step 7** (`Proximity control management`). Compute secondary control parameter

$$\tilde{\rho}_k = \frac{M_k(y^k, x^j)}{\Phi_k(y^k, x^j)},$$

with $M_k(\cdot, x^j) = \max\{m_0(\cdot, x^j), m_k(\cdot, x^j)\} + \frac{1}{2}(\cdot - x^j)^\top Q_j(\cdot - x^j)$, and then put

$$\tau_{k+1} = \begin{cases} \tau_k & \text{if } \tilde{\rho}_k < \tilde{\gamma}, \\ \theta\tau_k & \text{if } \tilde{\rho}_k \geqslant \tilde{\gamma}. \end{cases}$$

Increase inner loop counter $k$ and loop back to step 4.

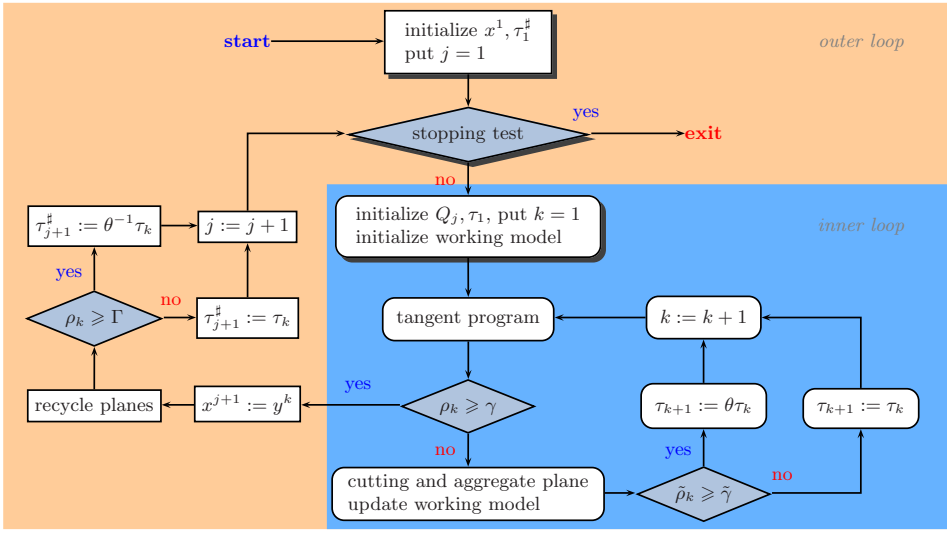



FIGURE 1. Flowchart of proximity control algorithm. Inner loop is shown in the lower right box

## 6. Upper envelope model

To analyse the convergence of the algorithm, we adapt a notion from [17, 18] for the progress function $F$. At the current iterate $x$ of the outer loop, the upper envelope model is defined as

$$\phi^\uparrow(y, x) = \sup\{m_{y^+,g}(y, x) : y^+ \in B(x, M), g \in \partial_1 F(y^+, x)\},$$

where $B(x, M)$ is a fixed ball large enough to contain all possible trial steps during the inner loop, and where $m_{y^+,g}(\cdot, x)$ is the cutting plane at serious iterate $x$ and trial step $y^+$ with subgradient $g \in \partial_1 F(y^+, x)$. We see immediately that $\phi^\uparrow(\cdot, x)$ is well-defined due to boundedness of $B(x, M)$ and boundedness of all possible trial steps during the inner loop which will be proved without using the notion $\phi^\uparrow$ in Lemma 8.1(i) and Lemma 8.2(i). Furthermore, we have the following result.

**Lemma 6.1.** *Let $f$ and $h$ be locally Lipschitz functions, then the following statements hold.*

(i) *$\phi^\uparrow(\cdot, x)$ is a convex function and $\phi_k(\cdot, x) \leqslant \phi^\uparrow(\cdot, x)$ for all counters $k$.*
(ii) *$\phi^\uparrow(x, x) = 0$ and $\partial_1\phi^\uparrow(x, x) = \partial_1 F(x, x)$.*
(iii) *$\phi^\uparrow$ is jointly upper semi-continuous.*

*Proof.* (i) The first statement is followed from the definition of $\phi^\uparrow(\cdot, x)$ and the construction of $\phi_k(\cdot, x)$.

(ii) By construction, $m_{y^+,g}(x, x) \leqslant 0$ and $m_{x,g}(x, x) = 0$, which implies $\phi^\uparrow(x, x) = 0$.

We now take an arbitrary $\bar{g} \in \partial_1\phi^\uparrow(x, x)$ and the tangent plane $\bar{m}(\cdot, x) = \bar{g}^\top(\cdot - x)$ to the graph of $\phi^\uparrow(\cdot, x)$ at $x$ associated with $\bar{g}$. Since $\phi^\uparrow(\cdot, x)$ is a convex function, $\bar{m}(\cdot, x) \leqslant \phi^\uparrow(\cdot, x)$. Fixing a vector $v \in \mathbb{R}^n$, for each $t > 0$, by definition of $\phi^\uparrow(\cdot, x)$, there exists a cutting plane at trial step $y_t$ with subgradient $g_t \in \partial_1 F(y_t, x)$ such that $\phi^\uparrow(x + tv, x) \leqslant m_{y_t,g_t}(x + tv, x) + t^2$. Note that $m_{y_t,g_t}(\cdot, x)$ can be represented

as $m_{y_t,g_t}(\cdot,x) = m_{y_t,g_t}(x,x) + g_t^\top(\cdot - x)$ and $m_{y_t,g_t}(x,x) \leqslant -c\|y_t - x\|^2$. This gives

$$t\bar{g}^\top v = \bar{m}(x+tv,x) \leqslant \phi^\uparrow(x+tv,x) \leqslant m_{y_t,g_t}(x+tv,x) + t^2 \leqslant -c\|y_t - x\|^2 + tg_t^\top v + t^2.$$

Let $t \to 0^+$, we get $y_t \to x$. By passing to a subsequence and using the upper semi-continuity of the Clarke subdifferential, we may assume that $g_t \to g$ for some $g \in \partial_1 F(x,x)$. In addition, the above estimate also gives $\bar{g}^\top v \leqslant g_t^\top v + t$ for all $t > 0$, which infers $\bar{g}^\top v \leqslant g^\top v$, and so

$$\bar{g}^\top v \leqslant \max\{g^\top v : g \in \partial_1 F(x,x)\}.$$

The expression on the right is the Clarke directional derivative of $F(\cdot,x)$ at $x$ in direction $v$. Since this relation holds true for every $v \in \mathbb{R}^n$, $\bar{g} \in \partial_1 F(x,x)$. Hence, $\partial_1\phi^\uparrow(x,x) \subset \partial_1 F(x,x)$.

It only remain to show $\partial_1 F(x,x) \subset \partial_1\phi^\uparrow(x,x)$. In order to do this, we consider the limit set

$$\underset{\to}{\partial}_1 F(x,x) = \{\lim_{k\to\infty} \nabla_1 F(y^k,x) : y^k \to x,\ F(\cdot,x) \text{ is differentiable at } y^k\}.$$

Here $\nabla_1 F(y^k,x)$ denote the subgradient of $F(\cdot,x)$ at $y^k$ in the case where $F(\cdot,x)$ is differentiable at $y^k$. We use the symbol $\underset{\to}{\partial}$ for the limit set, following Hiriart-Urruty [8]. By [2, Proposition 5] (see also [4, Theorem 2.5.1]), $\partial_1 F(x,x) = \operatorname{conv}(\underset{\to}{\partial}_1 F(x,x))$. We will prove that $\underset{\to}{\partial}_1 F(x,x) \subset \partial_1\phi^\uparrow(x,x)$. Indeed, take $g \in \underset{\to}{\partial}_1 F(x,x)$, there exist $y^k \to x$ and $g_k = \nabla_1 F(y^k,x) \in \partial_1 F(y^k,x)$ such that $g_k \to g$. Let $m_k(\cdot,x)$ be the cutting plane drawn at $y^k$ with subgradient $g_k$ then $m_k(y,x) \leqslant \phi^\uparrow(y,x)$ for all $y \in \mathbb{R}^n$ and

$$m_k(\cdot,x) = a_k + g_k^\top(\cdot - x), \quad a_k = \min\{t_k(x), -c\|y^k - x\|^2\},$$

where $t_k(x) = F(y^k,x) + g_k^\top(x - y^k)$. From $y^k \to x$, $g_k \to g$ and $F(x,x) = 0$, it follows that $a_k \to 0$, and so $m_k(y,x) \to g^\top(y - x)$, which implies $g^\top(y - x) \leqslant \phi^\uparrow(y,x)$ for all $y$. This together with $\phi^\uparrow(x,x) = 0$ gives $g \in \partial_1\phi^\uparrow(x,x)$. We obtain $\underset{\to}{\partial}_1 F(x,x) \subset \partial_1\phi^\uparrow(x,x)$ and then $\partial_1 F(x,x) = \operatorname{conv}(\underset{\to}{\partial}_1 F(x,x)) \subset \partial_1\phi^\uparrow(x,x)$ due to the convexity of $\partial_1\phi^\uparrow(x,x)$.

(iii) Let $(y^j,x^j) \to (y,x)$, we have to prove that $\limsup \phi^\uparrow(y^j,x^j) \leqslant \phi^\uparrow(y,x)$. Pick a sequence $\varepsilon_j \to 0^+$, by the definition of $\phi^\uparrow$, there exist cutting planes $m_{z^j,g_j}(\cdot,x^j) = t_{z^j}(\cdot) - s_j$ at serious iterate $x^j$, drawn at $z^j \in B(x^j,M)$ with $g_j \in \partial_1 F(z^j,x^j)$ such that

$$\phi^\uparrow(y^j,x^j) \leqslant m_{z^j,g_j}(y^j,x^j) + \varepsilon_j,$$

where $t_{z^j}(\cdot) = F(z^j,x^j) + g_j^\top(\cdot - z^j)$ and $s_j = [t_{z^j}(x^j) + c\|z^j - x^j\|^2]_+$. Since $x^j \to x$ and $z^j \in B(x^j,M)$, the sequence $z^j$ is bounded. Passing to a subsequence, we may assume without loss that $z^j \to z \in B(x,M)$ and $g_j \to g \in \partial_1 F(z,x)$ by the upper semi-continuity of the Clarke subdifferential. This gives $t_{z^j}(\cdot) \to t_z(\cdot) = F(z,x) + g^\top(\cdot - z)$, and so $s_j \to s = [t_z(x) + c\|z - x\|^2]_+$. It follows that

$$m_{z^j,g_j}(\cdot,x^j) = t_{z^j}(\cdot) - s_j \to t_z(\cdot) - s = m_{z,g}(\cdot,x)$$

as $i \to \infty$, and then also

$$m_{z^j,g_j}(y^j,x^j) = t_{z^j}(y^j) - s_j \to t_z(y) - s = m_{z,g}(y,x),$$

where uniformity comes from boundedness of the $g_j$. Therefore,

$$\limsup_{j\to\infty} \phi^{\uparrow}(y^j, x^j) \leqslant m_{z,g}(y, x) \leqslant \phi^{\uparrow}(y, x).$$

□

## 7. Lower-$C^1$ and upper-$C^1$ functions

According to Mifflin [15], a function $f : \mathbb{R}^n \to \mathbb{R}$ is *semismooth* at $x \in \mathbb{R}^n$ if $f$ is Lipschitz on a ball about $x$, and for $d \in \mathbb{R}^n$, $\{t_k\} \subset \mathbb{R}_+$, $\{\theta_k\} \subset \mathbb{R}^n$, $\{g_k\} \subset \mathbb{R}^n$ satisfying $t_k \downarrow 0$, $\theta_k/t_k \to 0 \in \mathbb{R}^n$, $g_k \in \partial f(x+t_kd+\theta_k)$, the sequence $g_k^\top d$ has exactly one accumulation point. The following lemma can be seen as a generalization of [15, Lemma 2].

**Lemma 7.1.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ Lipschitz near $x$ is semismooth at $x$ if and only if for any $\{d_k\} \subset \mathbb{R}^n$, $\{t_k\} \subset \mathbb{R}_+$, $\{g_k\} \subset \mathbb{R}^n$ satisfying $d_k \to d \in \mathbb{R}^n$, $t_k \downarrow 0$, $g_k \in \partial f(x + t_kd_k)$, we have*

$$\lim_{k\to\infty} g_k^\top d_k = f'(x; d).$$

*Proof.* Assume that $f$ is semismooth at $x$. Taking $s_k \downarrow 0$, by Lebourg's mean value theorem established in [10, Theorem 2.1] and proved in [11, Theorem 1.7], there exist $t_k^* \in (0, s_k)$ and $g_k^* \in \partial f(x + t_k^*d_k)$ such that

$$f(x + s_kd_k) - f(x) = g_k^{*\top} s_kd_k.$$

Then $t_k^* \downarrow 0$, $x + t_k^*d_k \to x$, and by [22, Theorem 9.13], the sequence $g_k^*$ is bounded, which gives $g_k^{*\top}(d_k - d) \to 0$. Observing that $g_k^* \in \partial f(x + t_k^*d + \theta_k)$ with $\theta_k = t_k^*(d_k - d)$, $\theta_k/t_k^* = d_k - d \to 0$, due to semismoothness of $f$, the sequence $g_k^{*\top}d$ has exactly one accumulation point, and so does $g_k^{*\top}d_k = g_k^{*\top}d + g_k^{*\top}(d_k - d)$. On the other hand,

$$g_k^{*\top}d_k = \frac{f(x + s_kd_k) - f(x)}{s_k} = \frac{f(x + s_kd) - f(x)}{s_k} + \frac{f(x + s_kd_k) - f(x + s_kd)}{s_k}.$$

The second term tends to 0 as $k \to \infty$ since $f$ is Lipschitz near $x$ and $d_k \to d$. This implies that $\lim_{k\to\infty} g_k^{*\top}d_k = f'(x; d)$. Now for any sequence $t_k \downarrow 0$, $g_k \in \partial f(x+t_kd_k)$, then $g_k \in \partial f(x + t_kd + \theta_k)$ with $\theta_k = t_k(d_k - d)$, $\theta_k/t_k = d_k - d \to 0$. By merging sequences $\{t_k\}$ and $\{t_k^*\}$, $\{g_k\}$ and $\{g_k^*\}$ and using again semismoothness of $f$, we must have $\lim_{k\to\infty} g_k^\top d_k = \lim_{k\to\infty} g_k^{*\top}d_k = f'(x; d)$. Conversely, writing $t_kd + \theta_k = t_k\,(d + \theta_k/t_k)$ with $d_k = d + \theta_k/t_k \to d$, we complete the proof of the lemma. □

**Corollary 7.2.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be semismooth at $x \in \mathbb{R}^n$. Then for any $y^k \to x$, $g_k \in \partial f(y^k)$ and for $\varepsilon > 0$,*

$$g_k^\top(x - y^k) \leqslant f(x) - f(y^k) + \varepsilon\|x - y^k\|$$

*for infinitely many $k$.*

*Proof.* Let $y^k \to x$ and $g_k \in \partial f(y^k)$. Passing to a subsequence, we may assume without loss of generality that $d_k = \frac{y^k - x}{\|y^k - x\|} \to d$ as $k \to \infty$. Set $t_k = \|y^k - x\|$, then

$y^k = x + t_k d_k$ and by Lemma 7.1,

$$\lim_{k\to\infty} \frac{g_k^\top (y^k - x)}{\|y^k - x\|} = f'(x; d).$$

We have also

$$\frac{f(y^k) - f(x)}{\|y^k - x\|} = \frac{f(x + t_k d_k) - f(x)}{t_k} = \frac{f(x + t_k d) - f(x)}{t_k} + \frac{f(x + t_k d_k) - f(x + t_k d)}{t_k}$$

converges to $f'(x; d)$ as $k \to \infty$ due to Lipschitzness of $f$ near $x$. Hence,

$$\frac{g_k^\top (x - y^k)}{\|x - y^k\|} - \frac{f(x) - f(y^k)}{\|x - y^k\|} \to 0$$

as $k \to \infty$, which completes the proof. □

We recall here the notion of lower-$C^1$ and upper-$C^1$ functions introduced in [23] and [22]. A function $f : \mathbb{R}^n \to \mathbb{R}$ is *lower-$C^1$* at (or around) $x_0 \in \mathbb{R}^n$, if there are a compact set $S$, a neighborhood $U$ of $x_0$, and a jointly continuous function $g : U \times S \to \mathbb{R}$ whose partial derivative with respect to the first variable is also jointly continuous, such that

$$f(x) = \max_{s\in S} g(x, s)$$

for all $x \in U$. The function $f$ is *upper-$C^1$* at $x_0$ if $-f$ is lower-$C^1$ at $x_0$. For the following, we collect some facts on lower-$C^1$ and upper-$C^1$ functions.

*Remark* 7.3. According to Proposition 2.4 and Theorem 3.9 in [23], if $f$ is lower-$C^1$ at $x_0$ then $f$ is regular and semismooth at $x_0$, but the converse need not be true.

**Lemma 7.4.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz. For all $x_0 \in \mathbb{R}^n$, the following statements are equivalent.*

(i) *$f$ is lower-$C^1$ at $x_0$.*
(ii) *$\partial f$ is strictly submonotone at $x_0$ in the sense that*

$$\liminf_{\substack{x\neq y\\ x,y\to x_0}} \frac{(g_x - g_y)^\top (x - y)}{\|x - y\|} \geqslant 0,$$

*whenever $g_x \in \partial f(x), g_y \in \partial f(y)$.*
(iii) *For every $\varepsilon > 0$ and $x, y$ close enough to $x_0$,*

$$g_y^\top (x - y) \leqslant f(x) - f(y) + \varepsilon\|x - y\|,$$

*whenever $g_y \in \partial f(y)$.*

*Proof.* The equivalence of (i) and (ii) is already established in [23, Theorem 3.9]. We will show that (ii) and (iii) are equivalent.

⋆ (ii) ⇒ (iii). For any distinct $x, y$, by Lebourg's mean value theorem, there exist $\lambda \in (0, 1)$ and $g_z \in \partial f(z)$ with $z = \lambda x + (1 - \lambda)y$ such that $f(x) - f(y) = g_z^\top (x - y)$. Take arbitrary $g_y \in \partial f(y)$ and note that $z - y = \lambda(x - y)$, we can write

$$f(x) - f(y) = g_y^\top (x - y) + (g_z - g_y)^\top (x - y) = g_y^\top (x - y) + \frac{(g_z - g_y)^\top (z - y)}{\|z - y\|}\|x - y\|.$$

Assume that $\partial f$ is strictly submonotone. Then, for fixed $x_0 \in \mathbb{R}^n$ and $\varepsilon > 0$, there exists $\delta > 0$ such that for any distinct $z, y \in B(x_0, \delta)$,

$$\frac{(g_z - g_y)^\top (z - y)}{\|z - y\|} \geqslant -\varepsilon.$$

Now for every $x, y \in B(x_0, \delta)$, $x \neq y$, we also have $z, y \in B(x_0, \delta)$, $z \neq y$, and thus (iii) holds due to the above expression and estimate.

$\star$ (iii) $\Rightarrow$ (ii). Let $x_0 \in \mathbb{R}^n$ and $\varepsilon > 0$ be fixed. If (iii) holds true, we can pick $x, y$ in a neighborhood of $x_0$ such that

$$g_y^\top (x - y) \leqslant f(x) - f(y) + \frac{\varepsilon}{2}\|x - y\|,$$

and also

$$g_x^\top (y - x) \leqslant f(y) - f(x) + \frac{\varepsilon}{2}\|y - x\|.$$

After adding these inequalities, reversing the sign and taking limit inferior, we get (ii). □

By applying Lemma 7.4 to function $-f$, we obtain immediately the following

**Corollary 7.5.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz. Then $f$ is upper-$C^1$ at $x_0 \in \mathbb{R}^n$ if and only if for every $\varepsilon > 0$ and $x, y$ close enough to $x_0$,*

$$g_x^\top (x - y) \leqslant f(x) - f(y) + \varepsilon\|x - y\|,$$

*whenever $g_x \in \partial f(x)$.*

## 8. Analysis of the inner loop

In this section we show that the inner loop terminates with a serious iterate after a finite number of steps. The current iterate $x$ is fixed, and so is $Q := Q(x)$. Assume that the inner loop at serious iterate $x$ turns infinitely, then either $\tau_k$ is increased infinitely often, or $\tau_k$ is frozen from some counter $k_0$ onwards. These two scenarios will be analyzed in Lemmas 8.1 and 8.2. Denote by $\mathcal{F}$ the feasible set of problem (1), i.e., $\mathcal{F} = \{x \in \mathbf{C} : h(x) \leqslant 0\}$, we have the following results.

**Lemma 8.1.** *Let $f$ and $h$ be locally Lipschitz on $\mathbb{R}^n$ such that at every point of $\mathcal{F}$, $f$ is semismooth or upper-$C^1$, and $h$ is semismooth. Suppose that the inner loop at serious iterate $x$ produces an infinite sequence of null step $y^k$ and the proximity control parameter is increased infinitely often. Then the following statements hold.*

(i) $y^k \to x$ *and* $\Phi_k(y^k, x) \to F(x, x) = 0$ *as* $k \to \infty$.
(ii) $0 \in \partial_1 F(x, x) + \partial i_{\mathbf{C}}(x)$.

*Proof.* (i) We see that the proximity parameter $\tau_k$ is never decreased in the inner loop, which combines with the assumption on $\tau_k$ implies that $\tau_k \to \infty$. Since $y^k$ is the optimal solution of the tangent program (3),

$$\tau_k(x - y^k) \in \partial_1(\Phi_k(y^k, x) + i_{\mathbf{C}}(y^k)).$$

Using the subgradient inequality and noting that $\Phi_k(x, x) = 0, x \in \mathbf{C}, y^k \in \mathbf{C}$, we get

$$\tau_k\|x - y^k\|^2 \leqslant \Phi_k(x, x) + i_{\mathbf{C}}(x) - \Phi_k(y^k, x) - i_{\mathbf{C}}(y^k) = -\Phi_k(y^k, x),$$

which implies

$$0 \leqslant \frac{\tau_k}{2}\|x-y^k\|^2 \leqslant -\phi_k(y^k,x) - \frac{1}{2}(x-y^k)^\top(Q+\tau_k I)(x-y^k)$$
$$\leqslant -\phi_k(y^k,x) \leqslant \|g(x)\|\|x-y^k\|.$$

Here we recall that $Q+\tau_k I \succ 0$ and $m_0(\cdot,x) \leqslant \phi_k(\cdot,x)$ with $m_0(\cdot,x) = g(x)^\top(\cdot - x)$ the exactness plane at $x$. It thus follows $\tau_k\|x-y^k\| \leqslant 2\|g(x)\|$. This gives $y^k \to x$ since $\tau_k \to \infty$. Using again the above estimate, we have $\phi_k(y^k,x) \to 0$, and so $\Phi_k(y^k,x) \to 0$.

(ii) Let

$$\mathbf{g}_k^* := (Q+\tau_k I)(x-y^k) \in \partial_1(\phi_k(y^k,x) + i_{\mathbf{C}}(y^k)),$$

then the sequence $\mathbf{g}_k^*$ is bounded since $\|\mathbf{g}_k^*\|$ is proportional to $\tau_k\|x-y^k\| \leqslant 2\|g(x)\|$ for $k$ large enough. Passing to a subsequence if necessary, we may assume without loss that $\mathbf{g}_k^* \to \mathbf{g}^*$ for some $\mathbf{g}^*$. We claim that $\mathbf{g}^* \in \partial_1 F(x,x) + \partial i_{\mathbf{C}}(x)$. For all $y \in \mathbb{R}^n$, the subgradient inequality gives

$$\mathbf{g}_k^{*\top}(y-y^k) \leqslant \phi_k(y,x) + i_{\mathbf{C}}(y) - \phi_k(y^k,x) - i_{\mathbf{C}}(y^k)$$
$$\leqslant \phi^\uparrow(y,x) - \phi_k(y^k,x) + i_{\mathbf{C}}(y),$$

due to Lemma 6.1 and the fact that $i_{\mathbf{C}}(y^k) = 0$. Passing to the limit in the above estimate and using the results in part (i), we get

$$\mathbf{g}^{*\top}(y-x) \leqslant \phi^\uparrow(y,x) + i_{\mathbf{C}}(y)$$

for all $y \in \mathbb{R}^n$. This together with $\phi^\uparrow(x,x) = 0$ and $i_{\mathbf{C}}(x) = 0$ gives $\mathbf{g}^* \in \partial_1(\phi^\uparrow(x,x) + i_{\mathbf{C}}(x))$. Using again Lemma 6.1, it implies that $\mathbf{g}^* \in \partial_1 F(x,x) + \partial i_{\mathbf{C}}(x)$.

We now prove $\mathbf{g}^* = 0$. Since the inner loop at serious iterate $x$ turns infinitely, $\rho_k < \gamma$ for all $k$. Moreover, the proximity parameter $\tau_k$ is increased infinitely, so there is an infinity of counters $k$ where $\tilde{\rho}_k \geqslant \tilde{\gamma}$. Therefore

$$\tilde{\gamma} - \gamma < \tilde{\rho}_k - \rho_k = \frac{F(y^k,x) - M_k(y^k,x)}{-\Phi_k(y^k,x)}. \tag{5}$$

It has already been shown in part (i) that $-\Phi_k(y^k,x) \geqslant \tau_k\|x-y^k\|^2$. Fixing $0 < \delta < 1$ and using $\tau_k \to \infty$ we have

$$\|\mathbf{g}_k^*\| \leqslant (1+\delta)\tau_k\|x-y^k\|,$$

and then

$$-\Phi_k(y^k,x) \geqslant \frac{1}{1+\delta}\|\mathbf{g}_k^*\|\|x-y^k\| \tag{6}$$

for $k$ large enough. Next we estimate the difference $F(y^k,x) - M_k(y^k,x)$. By construction,

$$M_k(y^k,x) \geqslant m_k(y^k,x) + \frac{1}{2}(y^k-x)^\top Q(y^k-x)$$

with $m_k(\cdot,x) = t_k(\cdot) - [t_k(x) + c\|y^k-x\|^2]_+$, where $t_k(\cdot) = F(y^k,x) + g_k^\top(\cdot - y^k)$ and $g_k \in \partial_1 F(y^k,x)$. This gives

$$F(y^k,x) - M_k(y^k,x) \leqslant [t_k(x) + c\|y^k-x\|^2]_+ - \frac{1}{2}(y^k-x)^\top Q(y^k-x).$$

Observing that the algorithm assures the feasibility of $x$, we first consider the case when $f$ and $h$ are semismooth at $x$. Then $F(\cdot,x)$ is semismooth at $x$ due to [15,

Theorem 6]. For each $\varepsilon > 0$, using $y^k \to x$ and Corollary 7.2, and passing to a subsequence, we find $k(\varepsilon)$ such that for $k \geqslant k(\varepsilon)$,

$$g_k^\top(x - y^k) \leqslant F(x, x) - F(y^k, x) + \varepsilon\|x - y^k\|,$$

which implies

$$t_k(x) = F(y^k, x) + g_k^\top(x - y^k) \leqslant \varepsilon\|x - y^k\|,$$

and then for $k$ large enough,

$$F(y^k, x) - M_k(y^k, x) \leqslant (1 + \delta)\varepsilon\|x - y^k\|. \tag{7}$$

In the case where $f$ is upper-$C^1$ and $h$ is semismooth at $x$, notice that

$$F(y^k, x) = \max\{f(y^k) - f(x), h(y^k)\}.$$

If $f(y^k) - f(x) < h(y^k)$ then $F(y^k, x) = h(y^k)$, $\partial_1 F(y^k, x) = \partial h(y^k)$, and so $t_k(x) = h(y^k) + g_k^\top(x - y^k)$ with $g_k \in \partial h(y^k)$. Using again Corollary 7.2, for $k$ large enough,

$$t_k(x) \leqslant h(x) + \varepsilon\|x - y^k\| \leqslant \varepsilon\|x - y^k\|,$$

which yields (7). On the other hand, noting that the exactness plane $m_0(\cdot, x) = g(x)^\top(\cdot - x)$ is based on $g(x) \in \partial f(x)$ since $h(x) \leqslant 0$, and then applying Corollary 7.5, we get

$$m_0(y^k, x) = g(x)^\top(y^k - x) \geqslant -f(x) + f(y^k) - \varepsilon\|x - y^k\|$$

for $k$ large enough. Now if $f(y^k) - f(x) \geqslant h(y^k)$ then $F(y^k, x) = f(y^k) - f(x)$, and (7) holds true due to the fact that $M_k(y^k, x) \geqslant m_0(y^k, x) + \frac{1}{2}(y^k - x)^\top Q(y^k - x)$.

From (5), (6) and (7), we obtain

$$\|\mathbf{g}_k^*\| \leqslant \frac{(1 + \delta)^2}{\tilde{\gamma} - \gamma}\varepsilon$$

for $k$ large enough. This holds for all $\varepsilon > 0$, so $\mathbf{g}^* = 0$, and the lemma is proved. □

**Lemma 8.2.** *Let $f$ and $h$ be locally Lipschitz functions. Suppose that the inner loop at serious iterate $x$ produces an infinite sequence of null step $y^k$ and the proximity control parameter is increased finitely often. Then the following statements hold.*

(i) $y^k \to x$ *and* $\Phi_k(y^k, x) \to F(x, x) = 0$ *as* $k \to \infty$.
(ii) $0 \in \partial_1 F(x, x) + \partial i_{\mathbf{C}}(x)$.

*Proof.* (i) Since the control parameter $\tau_k$ is increased finitely often, it remains unchanged from counter $k_0$ onwards, i.e., $\tau_k = \tau_{k_0} := \tau$ for all $k \geqslant k_0$. This means that $\rho_k < \gamma$ and $\tilde{\rho}_k < \tilde{\gamma}_k$ for all $k \geqslant k_0$. We consider the objective function of tangent program (3) for $k \geqslant k_0$,

$$\Psi_k(y, x) = \Phi_k(y, x) + \frac{\tau}{2}\|y - x\|^2 = \phi_k(y, x) + \frac{1}{2}\|y - x\|^2_{Q+\tau I},$$

where $\|\cdot\|_{Q+\tau I}$ denote the Euclidean norm derived from the positive definite matrix $Q + \tau I$. Then

$$\Psi_{k+1}(y, x) = \phi_{k+1}(y, x) + \frac{1}{2}\|y - x\|^2_{Q+\tau I}.$$

It follows from the construction of $\phi_{k+1}(\cdot,x)$ that $\phi_{k+1}(y,x) \geqslant m_k^*(y,x)$ with $m_k^*(\cdot,x)$ the aggregate plane at null step $y^k$. For $y \in \mathbf{C}$, we have

$$\begin{aligned} m_k^*(y,x) &= \phi_k(y^k,x) + g_k^{*\top}(y-y^k) \\ &= \phi_k(y^k,x) + [(Q+\tau I)(x-y^k)]^\top(y-y^k) - h_k^{*\top}(y-y^k) \\ &\geqslant \phi_k(y^k,x) + (x-y^k)^\top(Q+\tau I)(y-y^k), \end{aligned}$$

by using (4) and noting that $h_k^{*\top}(y-y^k) \leqslant i_{\mathbf{C}}(y) - i_{\mathbf{C}}(y^k) = 0$ due to the subgradient inequality. In addition,

$$\begin{aligned} \|y-x\|_{Q+\tau I}^2 &= \|(y^k-x)+(y-y^k)\|_{Q+\tau I}^2 \\ &= \|y^k-x\|_{Q+\tau I}^2 + \|y-y^k\|_{Q+\tau I}^2 - 2(x-y^k)^\top(Q+\tau I)(y-y^k), \end{aligned}$$

using the fact that $(x-y)^\top(Q+\tau I)(y-y^k) = (y-y^k)^\top(Q+\tau I)(x-y)$. Hence, for $y \in \mathbf{C}$,

$$\Psi_{k+1}(y,x) \geqslant \phi_k(y^k,x) + \frac{1}{2}\|y^k-x\|_{Q+\tau I}^2 + \frac{1}{2}\|y-y^k\|_{Q+\tau I}^2 = \Psi_k(y^k,x) + \frac{1}{2}\|y-y^k\|_{Q+\tau I}^2.$$

Substituting $y = y^{k+1}$ and remarking that $y^{k+1}$ is the minimizer of $\Psi_{k+1}(y,x)$, we have

$$\Psi_k(y^k,x) + \frac{1}{2}\|y^{k+1}-y^k\|_{Q+\tau I}^2 \leqslant \Psi_{k+1}(y^{k+1},x) \leqslant \Psi_{k+1}(x,x) = \Phi_{k+1}(x,x) = 0.$$

This shows that the sequence $\Psi_k(y^k,x)$ is monotonically increasing and bounded above by 0, so $\Psi_k(y^k,x) \to \Psi^*$ as $k \to \infty$ for some $\Psi^* \leqslant 0$. Letting $k \to \infty$ in the above inequality, we obtain $\frac{1}{2}\|y^{k+1}-y^k\|_{Q+\tau I}^2 \to 0$, which implies

$$\|y^{k+1}-y^k\| \to 0 \text{ as } k \to \infty. \tag{8}$$

On the other hand, proceeding as in the proof of Lemma 8.1, we have

$$\tau\|x-y^k\| \leqslant 2\|g(x)\|, \quad k \geqslant k_0,$$

which proves that the sequence of trial steps $y^k$ is bounded. By combining with (8),

$$\|y^{k+1}-x\|_{Q+\tau I}^2 - \|y^k-x\|_{Q+\tau I}^2 = (y^k-y^{k+1})^\top(Q+\tau I)[(y^{k+1}-x)+(y^k-x)] \to 0 \text{ as } k \to \infty.$$

Recalling that $\phi_k(y,x) = \Psi_k(y,x) - \frac{1}{2}\|y-x\|_{Q+\tau I}^2$ and using the above convergence results, we get

$$\begin{aligned} &\phi_{k+1}(y^{k+1},x) - \phi_k(y^k,x) = \\ &\qquad \Psi_{k+1}(y^{k+1},x) - \Psi_k(y^k,x) - \frac{1}{2}\left(\|y^{k+1}-x\|_{Q+\tau I}^2 - \|y^k-x\|_{Q+\tau I}^2\right) \end{aligned} \tag{9}$$

converges to 0 as $k \to \infty$.

We now claim that $\phi_{k+1}(y^k,x) - \phi_k(y^k,x) \to 0$, and then also $\Phi_{k+1}(y^k,x) - \Phi_k(y^k,x) \to 0$ as $k \to \infty$. By the construction of the model $\phi_{k+1}(\cdot,x)$, there exists a cutting plane $m_{i_k}(\cdot,x) = a_{i_k} + g_{i_k}^\top(\cdot - x)$ at null step $y^{i_k}$, $i_k \in \{1,\dots,k\}$, with $g_{i_k} \in \partial_1 F(y^{i_k},x)$ such that $\phi_{k+1}(y^k,x) = m_{i_k}(y^k,x)$. Then

$$\phi_{k+1}(y^k,x) = m_{i_k}(y,x) - g_{i_k}^\top(y-y^k) \leqslant \phi_{k+1}(y,x) - g_{i_k}^\top(y-y^k)$$

for all $y$. Therefore,

$$\begin{aligned} 0 &\leqslant \phi_{k+1}(y^k, x) - \phi_k(y^k, x) \\ &\leqslant \phi_{k+1}(y^{k+1}, x) - \phi_k(y^k, x) + \|g_{i_k}\|\|y^{k+1} - y^k\| \end{aligned}$$

and this term converges to 0 due to (8), (9) and boundedness of $g_{i_k}$. Here boundedness of the $g_{i_k} \in \partial_1 F(y^{i_k}, x)$ follows from boundedness of the subdifferential of $F(\cdot, x)$ on the bounded set of trial steps $y^k$ (cf. [22, Theorem 9.13]). We obtain $\phi_{k+1}(y^k, x) - \phi_k(y^k, x) \to 0$, and so

$$\Phi_{k+1}(y^k, x) - \Phi_k(y^k, x) \to 0 \text{ as } k \to \infty. \tag{10}$$

We next show that $\Phi_k(y^k, x) \to F(x, x) = 0$, of course also $\phi_k(y^k, x) \to 0$, and then $y^k \to x$ as $k \to \infty$. Assume this is not the case, then $\eta := \limsup_{k\to\infty} \Phi_k(y^k, x) < 0$. Choose $\varepsilon > 0$ such that $0 < \varepsilon < -(1 - \tilde{\gamma})\eta$. Thanks to (10), there exists $k_1 \geqslant k_0$ such that

$$\Phi_{k+1}(y^k, x) \leqslant \Phi_k(y^k, x) + \varepsilon$$

for all $k \geqslant k_1$. Since $\tilde{\rho}_k < \tilde{\gamma}$ for all $k \geqslant k_1 \geqslant k_0$ and $\Phi_k(y^k, x) \leqslant \Phi_k(x, x) = 0$,

$$\tilde{\gamma}\Phi_k(y^k, x) \leqslant M_k(y^k, x) \leqslant \Phi_{k+1}(y^k, x) \leqslant \Phi_k(y^k, x) + \varepsilon,$$

using $M_k(\cdot, x) \leqslant \Phi_{k+1}(\cdot, x)$ by construction. Passing to the limit, we get $\tilde{\gamma}\eta \leqslant \eta + \varepsilon$, which contradicts the choice of $\varepsilon$. That gives $\eta = 0$, as claimed.

By the definitions of $\Phi_k$ and $y^k$ we have

$$\Phi_k(y^k, x) + \frac{\tau}{2}\|y^k - x\|^2 = \Psi_k(y^k, x) \leqslant \Psi_k(x, x) = \Phi_k(x, x) = 0.$$

This together with $\Phi_k(y^k, x) \to F(x, x) = 0$ gives $y^k \to x$ as $k \to \infty$.

(ii) We observe that by the necessary optimality condition for (3) and the subgradient inequality,

$$\begin{aligned} (x - y^k)^\top (Q + \tau I)(y - y^k) &\leqslant \phi_k(y, x) + i_{\mathbf{C}}(y) - \phi_k(y^k, x) - i_{\mathbf{C}}(y^k) \\ &\leqslant \phi^\uparrow(y, x) + i_{\mathbf{C}}(y) - \phi_k(y^k, x) - i_{\mathbf{C}}(y^k) \end{aligned}$$

for all $y$. Passing to the limit and noting that $\phi^\uparrow(x, x) = \phi(x, x) = 0$, $i_{\mathbf{C}}(y^k) = i_{\mathbf{C}}(x) = 0$, we obtain

$$0 \leqslant \phi^\uparrow(y, x) + i_{\mathbf{C}}(y) - \phi^\uparrow(x, x) - i_{\mathbf{C}}(x),$$

which implies $0 \in \partial_1(\phi^\uparrow(x, x) + i_{\mathbf{C}}(x))$, and since $\partial_1\phi^\uparrow(x, x) = \partial_1 F(x, x)$, we are done. □

We end this section with the following conclusion.

**Proposition 8.3.** *Let $f$ and $h$ be locally Lipschitz on $\mathbb{R}^n$ such that at every point of $\mathcal{F}$, $f$ is semismooth or upper-$C^1$, and $h$ is semismooth. Then the inner loop finds a serious iterate after a finite number of trial steps.*

*Proof.* Suppose that the inner loop at serious iterate $x$ turns infinitely. Then, as proved in Lemmas 8.1 and 8.2, we must have $0 \in \partial_1 F(x, x) + \partial i_{\mathbf{C}}(x)$. This contradicts the fact that the inner loop is only entered when $0 \notin \partial_1 F(x, x) + \partial i_{\mathbf{C}}(x)$. □

## 9. Convergence of the outer loop

We show in this section a strong convergence of our algorithm under the assumption that at every point of the feasible set $\mathcal{F}$, $f$ is lower-$C^1$ or upper-$C^1$, and $h$ is lower-$C^1$. By Proposition 8.3 and Remark 7.3, this assumption on $f$ and $h$ assures that the inner loop always terminates finitely.

**Theorem 9.1.** *Assume $f$ and $h$ in problem (1) are locally Lipschitz on $\mathbb{R}^n$ such that at every point of the feasible set $\mathcal{F}$, $f$ is lower-$C^1$ or upper-$C^1$, and $h$ is lower-$C^1$. Let $\{x \in \mathcal{F} : f(x) < f(x^1)\}$ be bounded, and let $x^j$ be the sequence of serious iterates generated by Algorithm 1. Then $x^j$ is a sequence of feasible points for (1), and one of the following two statements holds.*

(i) *The sequence $x^j$ ends finitely at a F. John critical point $x^{j^*}$ of (1). In the case $j^* > 1$, $x^{j^*}$ is even a Karush-Kuhn-Tucker point.*
(ii) *The sequence $x^j$ is bounded infinite, and every accumulation point $x^*$ is a F. John critical point of (1). In other words, $x^*$ is either a critical point of constraint violation, or a Karush-Kuhn-Tucker point.*

We see immediately that feasibility of sequence $x^j$ follows from feasibility of $x^1$ and Remark 3.1. If the sequence $x^j$ is finite, then the first statement of the theorem holds due to the stopping test of Algorithm 1 and Lemma 2.1. In the sequel, we focus on the case where the sequence $x^j$ is infinite, and suppose that in the $j$th outer loop, the serious step is accepted at inner loop counter $k_j$, that is, $x^{j+1} = y^{k_j}$. At the $j$th outer loop and the $k$th inner loop, we denote more precisely the proximity control parameter as $\tau_k^j$, and write $\tau_{k_j}$ for $\tau_{k_j}^j$. We also write $Q_j := Q(x^j)$ for the matrix of the second-order model, which depends on the serious iterates $x^j$.

**Lemma 9.2.** *Let $f$ and $h$ be locally Lipschitz functions such that $\{x \in \mathcal{F} : f(x) < f(x^1)\}$ is bounded. Then the sequence of serious iterates $x^j$ is bounded. In addition, $F(x^{j+1}, x^j) \to 0$, $\tau_{k_j}\|x^j - x^{j+1}\|^2 \to 0$ and $\|x^j - x^{j+1}\|^2_{Q_j+\tau_{k_j}I} \to 0$ as $j \to \infty$.*

*Proof.* Following Remark 3.1, feasibility of $x^1$ gives $f(x^{j+1}) < f(x^j)$ and $h(x^{j+1}) < 0$ for all $j$. Thus, $x^j$ is feasible for all $j$, and sequence $f(x^j)$ is decreased. This yields $\{x^j : j = 1, 2, \ldots\} \subset \{x \in \mathcal{F} : f(x) < f(x^1)\}$, and so the sequence $x^j$ is bounded.

Now for every accumulation point $x^*$ of the sequence $x^j$, the local Lipschitz continuity of $f$ implies that $f(x^*)$ is an accumulation point of the sequence $f(x^j)$, and then $f(x^j) \to f(x^*)$ due to the monotone sequence theorem. Therefore,

$$\liminf_{j\to\infty} F(x^{j+1}, x^j) \geqslant \lim_{j\to\infty} (f(x^{j+1}) - f(x^j)) = 0.$$

This together with $F(x^{j+1}, x^j) \leqslant 0$ gives $F(x^{j+1}, x^j) \to 0$ as $j \to \infty$.

Since $x^{j+1} = y^{k_j}$ is the optimal solution of tangent program (3),

$$(Q_j + \tau_{k_j} I)(x^j - x^{j+1}) \in \partial_1(\phi_{k_j}(x^{j+1}, x^j) + i_{\mathbf{C}}(x^{j+1})).$$

Using the subgradient inequality, we obtain

$$\begin{aligned}(x^j - x^{j+1})^\top (Q_j + \tau_{k_j} I)(x^j - x^{j+1}) &\leqslant \phi_{k_j}(x^j, x^j) + i_{\mathbf{C}}(x^j) - \phi_{k_j}(x^{j+1}, x^j) - i_{\mathbf{C}}(x^{j+1}) \\ &= -\phi_{k_j}(x^{j+1}, x^j) \\ &= -\Phi_{k_j}(x^{j+1}, x^j) + \frac{1}{2}(x^j - x^{j+1})^\top Q_j (x^j - x^{j+1}).\end{aligned}$$

By noting that $Q_j + \tau_{k_j} I \succ 0$, this implies

$$\frac{1}{2}\|x^j - x^{j+1}\|^2_{Q_j + \tau_{k_j} I} + \frac{1}{2}\tau_{k_j}\|x^j - x^{j+1}\|^2 \leqslant -\Phi_{k_j}(x^{j+1}, x^j).$$

Moreover, $-\gamma\Phi_{k_j}(x^{j+1}, x^j) \leqslant -F(x^{j+1}, x^j)$ due to the acceptance test and the fact that $\Phi_{k_j}(x^{j+1}, x^j) \leqslant 0$. Hence,

$$\frac{1}{2}\|x^j - x^{j+1}\|^2_{Q_j + \tau_{k_j} I} + \frac{1}{2}\tau_{k_j}\|x^j - x^{j+1}\|^2 \leqslant -\frac{1}{\gamma}F(x^{j+1}, x^j).$$

Combining with $F(x^{j+1}, x^j) \to 0$, we complete the proof. □

**Lemma 9.3.** *Let $f$ and $h$ be locally Lipschitz functions such that $\{x \in \mathcal{F} : f(x) < f(x^1)\}$ is bounded. Suppose there exists an infinite subset $J \subset \mathbb{N}$ such that $x^j \to x^*$, $j \in J$. Let $\mathbf{g}_j^* = (Q_j + \tau_{k_j} I)(x^j - x^{j+1})$ be the aggregate subgradient belonging to $x^{j+1}$ in the $j$th outer loop. Then if the sequence $(\mathbf{g}_j^*)_{j \in J}$ has a subsequence which converges to 0 we have that $0 \in \partial_1 F(x^*, x^*) + \partial i_{\mathbf{C}}(x^*)$.*

*Proof.* Assume that there exists an infinite subset $J'$ of $J$ such that $\mathbf{g}_j^* \to 0$, $j \in J'$. Since $\mathbf{g}_j^* \in \partial_1(\phi_{k_j}(x^{j+1}, x^j) + i_{\mathbf{C}}(x^{j+1}))$, for any $y \in \mathbb{R}^n$, the subgradient inequality gives

$$\begin{aligned}\mathbf{g}_j^{*\top}(y - x^{j+1}) &\leqslant \phi_{k_j}(y, x^j) + i_{\mathbf{C}}(y) - \phi_{k_j}(x^{j+1}, x^j) - i_{\mathbf{C}}(x^{j+1}) \\ &= \phi_{k_j}(y, x^j) - \Phi_{k_j}(x^{j+1}, x^j) + \frac{1}{2}(x^{j+1} - x^j)^\top Q_j (x^{j+1} - x^j) + i_{\mathbf{C}}(y) \\ &\leqslant \phi_{k_j}(y, x^j) - \Phi_{k_j}(x^{j+1}, x^j) + \frac{1}{2}\|x^j - x^{j+1}\|^2_{Q_j + \tau_{k_j} I} + i_{\mathbf{C}}(y) \\ &\leqslant \phi^\uparrow(y, x^j) - \frac{1}{\gamma}F(x^{j+1}, x^j) + \frac{1}{2}\|x^j - x^{j+1}\|^2_{Q_j + \tau_{k_j} I} + i_{\mathbf{C}}(y).\end{aligned}$$

Here the last estimate is obtained by Lemma 6.1 and the acceptance test of the algorithm. By passing to the limit and using the hypothesis $\mathbf{g}_j^* \to 0$ and the results from Lemmas 6.1(iii) and 9.2, we get

$$0 \leqslant \phi^\uparrow(y, x^*) + i_{\mathbf{C}}(y).$$

It follows that $0 \in \partial_1(\phi^\uparrow(x^*, x^*) + i_{\mathbf{C}}(x^*))$ since $\phi^\uparrow(x^*, x^*) = 0$ and $i_{\mathbf{C}}(x^*) = 0$. Together with $\partial_1 \phi^\uparrow(x^*, x^*) = \partial_1 F(x^*, x^*)$, this ends the proof of the lemma. □

**Lemma 9.4.** *Under the hypotheses of Lemma 9.3, if $\|\mathbf{g}_j^*\| \geqslant \zeta$ for some $\zeta > 0$ and every $j \in J$ then the following statements hold.*

(i) *$\tau_{k_j} \to \infty$ as $j \in J$, $j \to \infty$.*
(ii) *There exists an infinite subset $J^+$ of $J$ such that the $\tau$-parameter was increased at least once during the $j$th outer loop for all $j \in J^+$. Suppose this happened for the last time at stage $r_j$ for some $r_j$. Then $x^j - y^{r_j} \to 0$ and $\phi_{r_j}(y^{r_j}, x^j) \to 0$ as $j \in J^+$, $j \to \infty$.*

(iii) *If at every point of $\mathcal{F}$, $f$ is lower-$C^1$ or upper-$C^1$, and $h$ is lower-$C^1$, then* $0 \in \partial_1 F(x^*, x^*) + \partial i_{\mathbf{C}}(x^*)$.

*Proof.* (i) Suppose on the contrary that the sequence $(\tau_{k_j})_{j\in J}$ has a bounded subsequence, then by passing to a subsequence, we may assume without loss of generality that $(\tau_{k_j})_{j\in J}$ is bounded. By combining with boundedness of the $Q_j$ and boundedness of the serious steps $x^j$ shown in Lemma 9.2, there exists an infinite subset $J'$ of $J$ such that $\tau_{k_j} \to \bar{\tau}$, $Q_j \to \bar{Q}$ and $x^j - x^{j+1} \to \Delta x$ as $j \in J', j \to \infty$. It follows that $\mathbf{g}_j^* \to (\bar{Q}+\bar{\tau}I)\Delta x$ with $\|(\bar{Q}+\bar{\tau}I)\Delta x\| \geqslant \zeta > 0$ and $\mathbf{g}_j^{*\top}(x^j - x^{j+1}) \to \Delta x^\top(\bar{Q}+\bar{\tau}I)\Delta x$. According to Lemma 9.2, $\mathbf{g}_j^{*\top}(x^j - x^{j+1}) = \|x^j - x^{j+1}\|^2_{Q_j+\tau_{k_j}I} \to 0$, which implies $\Delta x^\top(\bar{Q} + \bar{\tau}I)\Delta x = 0$. Since $\bar{Q} + \bar{\tau}I$ is positive semidefinite symmetric, we deduce $(\bar{Q} + \bar{\tau}I)\Delta x = 0$, that contradicts $\|(\bar{Q} + \bar{\tau}I)\Delta x\| \geqslant \zeta > 0$. Hence, $\tau_{k_j} \to \infty$ as $j \to \infty$.

(ii) For each outer loop counter $j \in J$, either $\tau_{k_j} > \tau_1^j$ or $\tau_{k_j} = \tau_1^j$ with $\tau_1^j \leqslant T < \infty$ by the algorithm. But $\tau_{k_j} \to \infty$ as $j \to \infty$, $j \in J$, set $J^- = \{j \in J : \tau_{k_j} = \tau_1^j\}$ therefore must be finite, which implies the infinity of set $J^+ = \{j \in J : \tau_{k_j} > \tau_1^j\}$. Suppose that for each $j \in J^+$, the $\tau$-parameter was increased for the last time at counter $r_j$, then $r_j \in \{1, \ldots, k_j - 1\}$ since at inner loop counter $k_j$ the serious step is accepted. That is

$$\tau_{k_j} = \tau_{k_j-1} = \cdots = \tau_{r_j+1} = \theta\tau_{r_j}.$$

Conforming to the update proximity control parameter of the algorithm, the increase at stage $r_j$ is due to the fact that

$$\rho_{r_j} < \gamma \text{ and } \tilde{\rho}_{r_j} \geqslant \tilde{\gamma}. \tag{11}$$

Noting that $\tau_{r_j} = \theta^{-1}\tau_{k_j} \to \infty$ ($j \in J^+$) and $y^{r_j}$ is the optimal solution of tangent program (3), we have

$$\tau_{r_j}(x^j - y^{r_j}) \in \partial_1(\Phi_{r_j}(y^{r_j}, x^j) + i_{\mathbf{C}}(y^{r_j})).$$

By the subgradient inequality and the fact that $\Phi_{r_j}(x^j, x^j) = 0, i_{\mathbf{C}}(x^j) = i_{\mathbf{C}}(y^{r_j}) = 0$,

$$\tau_{r_j}\|x^j - y^{r_j}\|^2 \leqslant -\Phi_{r_j}(y^{r_j}, x^j). \tag{12}$$

It follows that

$$\begin{aligned} 0 \leqslant \frac{\tau_{r_j}}{2}\|x^j - y^{r_j}\|^2 &\leqslant -\phi_{r_j}(y^{r_j}, x^j) - \frac{1}{2}(x^j - y^{r_j})^\top(Q_j + \tau_{r_j}I)(x^j - y^{r_j}) \\ &\leqslant -\phi_{r_j}(y^{r_j}, x^j) \leqslant \|g(x^j)\|\|x^j - y^{r_j}\|, \end{aligned}$$

where $m_0(\cdot, x^j) = g(x^j)^\top(\cdot - x^j)$ is the exactness plane at $x^j$. This implies $\tau_{r_j}\|x^j - y^{r_j}\| \leqslant 2\|g(x^j)\|$. Remark that the sequence $g(x^j)$ is bounded due to [22, Theorem 9.13], and then $x^j - y^{r_j} \to 0$ since $\tau_{r_j} \to \infty$. The term $-\phi_{r_j}(y^{r_j}, x^j)$ therefore is squeezed in between two convergent terms with the same limit 0, which gives $\phi_{r_j}(y^{r_j}, x^j) \to 0$.

(iii) We now consider

$$\tilde{\mathbf{g}}_j := \left(Q_j + \tau_{r_j}I\right)(x^j - y^{r_j}) \in \partial_1(\phi_{r_j}(y^{r_j}, x^j) + i_{\mathbf{C}}(y^{r_j})),$$

then as $\tau_{r_j} \to \infty$ and the $Q_j$ are bounded, $\|\tilde{\mathbf{g}}_j\|$ behaves asymptotically like constant times $\tau_{r_j}\|x^j - y^{r_j}\| \leqslant 2\|g(x^j)\|$, which implies boundedness of the sequence $\tilde{\mathbf{g}}_j$.

Therefore, possibly passing to a subsequence, we have $\tilde{\mathbf{g}}_j \to \tilde{\mathbf{g}}$ for some $\tilde{\mathbf{g}}$. By using the subgradient inequality and Lemma 6.1, and noting that $i_{\mathbf{C}}(y^{r_j}) = 0$,

$$\begin{aligned}\tilde{\mathbf{g}}_j^\top(y - y^{r_j}) &\leqslant \phi_{r_j}(y, x^j) + i_{\mathbf{C}}(y) - \phi_{r_j}(y^{r_j}, x^j) - i_{\mathbf{C}}(y^{r_j}) \\ &\leqslant \phi^\uparrow(y, x^j) + i_{\mathbf{C}}(y) - \phi_{r_j}(y^{r_j}, x^j).\end{aligned}$$

for all $y \in \mathbb{R}^n$. Passing to the limit and using the results in part (ii), we obtain

$$\tilde{\mathbf{g}}^\top(y - x^*) \leqslant \phi^\uparrow(y, x^*) + i_{\mathbf{C}}(y),$$

which implies $\tilde{\mathbf{g}} \in \partial_1(\phi^\uparrow(x^*, x^*) + i_{\mathbf{C}}(x^*))$ since $\phi^\uparrow(x^*, x^*) = 0$ and $i_{\mathbf{C}}(x^*) = 0$. By Lemma 6.1, we deduce that $\tilde{\mathbf{g}} \in \partial_1 F(x^*, x^*) + \partial i_{\mathbf{C}}(x^*)$.

Let us next show $\tilde{\mathbf{g}} = 0$. Fix $0 < \delta < 1$, it follows from $\tau_{r_j} \to \infty$ that for $j$ large enough,

$$\|\tilde{\mathbf{g}}_j\| \leqslant (1 + \delta)\tau_{r_j}\|x^j - y^{r_j}\|,$$

which combined with (12) gives

$$\|\tilde{\mathbf{g}}_j\| \leqslant (1 + \delta)\frac{-\Phi_{r_j}(y^{r_j}, x^j)}{\|x^j - y^{r_j}\|}. \tag{13}$$

On the other hand, from (11) we have

$$\tilde{\gamma} - \gamma < \tilde{\rho}_{r_j} - \rho_{r_j} = \frac{F(y^{r_j}, x^j) - M_{r_j}(y^{r_j}, x^j)}{-\Phi_{r_j}(y^{r_j}, x^j)}. \tag{14}$$

Remarking that

$$M_{r_j}(\cdot, x^j) \geqslant m_{r_j}(\cdot, x^j) + \frac{1}{2}(\cdot - x^j)^\top Q_j(\cdot - x^j),$$

where $m_{r_j}(\cdot, x^j) = t_{r_j}(\cdot) - [t_{r_j}(x^j) + c\|y^{r_j} - x^j\|^2]_+$, and $t_{r_j}(\cdot) = F(y^{r_j}, x^j) + g_{r_j}^\top(\cdot - y^{r_j})$ with $g_{r_j} \in \partial_1 F(y^{r_j}, x^j)$, we get

$$F(y^{r_j}, x^j) - M_{r_j}(y^{r_j}, x^j) \leqslant [t_{r_j}(x^j) + c\|y^{r_j} - x^j\|^2]_+ - \frac{1}{2}(y^{r_j} - x^j)^\top Q_j(y^{r_j} - x^j).$$

For $\varepsilon > 0$ fixed, we distinguish the following two cases.

*Case I.* The both functions $f$ and $h$ are lower-$C^1$ at $x^*$, so is $F(\cdot, x^j)$. By the assumption that $x^j \to x^*$ and the fact that $x^j - y^{r_j} \to 0$ proved in part (ii), thanks to Lemma 7.4, there exists $j(\varepsilon)$ such that

$$g_{r_j}^\top(x^j - y^{r_j}) \leqslant F(x^j, x^j) - F(y^{r_j}, x^j) + \varepsilon\|x^j - y^{r_j}\|$$

for every $j \geqslant j(\varepsilon)$. This implies

$$t_{r_j}(x^j) = F(y^{r_j}, x^j) + g_{r_j}^\top(x^j - y^{r_j}) \leqslant \varepsilon\|x^j - y^{r_j}\|,$$

and thus for $j$ large enough,

$$F(y^{r_j}, x^j) - M_{r_j}(y^{r_j}, x^j) \leqslant (1 + \delta)\varepsilon\|x^j - y^{r_j}\|. \tag{15}$$

*Case II.* The function $f$ is upper-$C^1$ and the function $h$ is lower-$C^1$ at $x^*$. By the feasibility of $x^j$, if $f(y^{r_j}) - f(x^j) < h(y^{r_j})$ then

$$F(y^{r_j}, x^j) = \max\{f(y^{r_j}) - f(x^j), h(y^{r_j})\} = h(y^{r_j}), \partial_1 F(y^{r_j}, x^j) = \partial h(y^{r_j}),$$

and therefore the tangent $t_{r_j}(\cdot) = h(y^{r_j}) + g_{r_j}^\top(\cdot - y^{r_j})$ with $g_{r_j} \in \partial h(y^{r_j})$. The estimate (15) holds based on the inequality

$$t_{r_j}(x^j) \leqslant h(x^j) + \varepsilon\|x^j - y^{r_j}\| \leqslant \varepsilon\|x^j - y^{r_j}\|, \text{ for } j \text{ large enough,}$$

using Lemma 7.4. Conversely, if $f(y^{r_j}) - f(x^j) \geqslant h(y^{r_j})$ then $F(y^{r_j}, x^j) = f(y^{r_j}) - f(x^j)$, and by recalling the exactness plane $m_0(\cdot, x^j) = g(x^j)^\top(\cdot - x^j)$ with $g(x^j) \in \partial f(x^j)$, we have

$$M_{r_j}(y^{r_j}, x^j) - \frac{1}{2}(y^{r_j} - x^j)^\top Q_j(y^{r_j} - x^j) \geqslant m_0(y^{r_j}, x^j) \geqslant -f(x^j) + f(y^{r_j}) - \varepsilon\|x^j - y^{r_j}\|$$

due to Corollary 7.5. This gives (15).

Now it follows from (13), (14) and (15) that

$$\|\tilde{\mathbf{g}}_j\| \leqslant \frac{(1+\delta)^2}{\tilde{\gamma} - \gamma}\varepsilon$$

for $j$ large enough. Since $\varepsilon > 0$ is arbitrary, we conclude that $\tilde{\mathbf{g}} = 0$, meaning $0 \in \partial_1 F(x^*, x^*) + \partial i_{\mathbf{C}}(x^*)$. □

*Proof of Theorem 9.1.* As discussed just after the statement of the theorem, the sequence $x^j$ consists of feasible points for (1) and verifies statement (i) when it is finite. Suppose that the sequence $x^j$ is infinite, then it is bounded by Lemma 9.2. Let $x^*$ be an accumulation point of the sequence $x^j$, we have $h(x^*) \leqslant 0, x^* \in \mathbf{C}$ due to feasibility of $x^j$ for all $j$, continuity of $h(\cdot)$ and closed convexity of $\mathbf{C}$. It follows from Lemmas 9.3 and 9.4 that $0 \in \partial_1 F(x^*, x^*) + \partial i_{\mathbf{C}}(x^*)$. This together with Lemma 2.1 gives the last statement of the theorem. □

In practice, a challenge is the lack of convexity, by which it is difficult to guarantee convergence to a single critical point. Some satisfactory results can nevertheless be obtained from the following corollaries.

**Corollary 9.5.** *Under the hypotheses of Theorem 9.1, for every $\varepsilon > 0$ there exists an index $j_0(\varepsilon) \in \mathbb{N}$ such that every $j \geqslant j_0(\varepsilon)$, $x^j$ is within $\varepsilon$-distance of the set*

$$L = \{x^* \in \mathbf{C} : 0 \in \partial_1 F(x^*, x^*) + \partial i_{\mathbf{C}}(x^*)\}.$$

*Proof.* Suppose there exists $\varepsilon > 0$ and an infinite subsequence $x^j$, $j \in J$, such that $\|x^j - x^*\| > \varepsilon$ for all $j \in J$ and all $x^* \in L$. Since the sequence $x^j$, $j \in J$, is bounded, it has an accumulation point $x^*$, and by Theorem 9.1, $x^* \in L$. That is a contradiction. □

**Corollary 9.6.** *Under the hypotheses of Theorem 9.1, if the set $L$ in Corollary 9.5 is totally disconnected* [6, Definition 9.4.1]*, then the sequence $x^j$ converges to a single point $x^* \in \mathbf{C}$ with $0 \in \partial_1 F(x^*, x^*) + \partial i_{\mathbf{C}}(x^*)$.*

*Proof.* Recall that $\|x^j - x^{j+1}\|^2_{Q_j + \tau_{k_j} I} \to 0$ as $j \to \infty$ due to Lemma 9.2. In each outer loop counter $j$, since $T > q + \kappa \geqslant -\lambda_{\min}(Q_j) + \kappa$, so

$$\tau_{k_j} \geqslant \tau_1 \geqslant -\lambda_{\min}(Q_j) + \kappa,$$

and therefore $\lambda_{\min}(Q_j + \tau_{k_j} I) \geqslant \kappa$, which implies that

$$\|x^j - x^{j+1}\|^2_{Q_j + \tau_{k_j} I} \geqslant \kappa\|x^j - x^{j+1}\|^2.$$

It follows that $\|x^j - x^{j+1}\|^2 \to 0$, and also $x^j - x^{j+1} \to 0$ as $j \to \infty$. By Ostrowski's theorem [20, Theorem 26.1], the set $K$ of accumulation points of the sequence $x^j$ is either singleton or a compact continuum. Theorem 9.1 gives $K \subset L$, and so $K$ must be singleton thanks to the hypothesis of $L$. □

In the case where subgradients are inexact, working with the approximate subdifferential

$$\partial^\varepsilon f(x) = \partial f(x) + \varepsilon B,$$

where $\partial$ is the exact Clarke subdifferential, and $B$ the unit ball in some fixed Euclidean norm, we have the following

**Corollary 9.7.** *Let $f$ and $h$ in problem (1) be locally Lipschitz on $\mathbb{R}^n$ such that at every point of $\mathcal{F}$, $f$ is lower-$C^1$ or upper-$C^1$, and $h$ is lower-$C^1$. Suppose that $\{x \in \mathcal{F} : f(x) < f(x^1)\}$ is bounded, and subgradients are drawn from $\partial_1^\varepsilon F(y,x)$, whereas function values are exact. Then the sequence of serious iterates $x^j$ is a bounded sequence of feasible points for (1), and every accumulation point $x^*$ of the $x^j$ satisfies $h(x^*) \leqslant 0, x^* \in \mathbf{C}$ and $0 \in \partial_1^{\tilde\varepsilon} F(x^*,x^*) + \partial^{\tilde\varepsilon} i_{\mathbf{C}}(x^*)$, where $\tilde\varepsilon = (1+(\tilde\gamma-\gamma)^{-1})\varepsilon$.*

*Proof.* Noting that in this case $\partial_1\phi^\uparrow(x,x) = \partial_1^\varepsilon F(x,x)$, we proceed as in proof of Theorem 9.1, and have just to replace (7) and (15) by the following estimates for every $\varepsilon' > 0$,

$$F(y^k,x) - M_k(y^k,x) \leqslant (1+\delta)(\varepsilon'+\varepsilon)\|x-y^k\| \text{ for } k \text{ large enough,}$$
$$F(y^{r_j},x^j) - M_{r_j}(y^{r_j},x^j) \leqslant (1+\delta)(\varepsilon'+\varepsilon)\|x^j-y^{r_j}\| \text{ for } j \text{ large enough.}$$

For a detailed proof in the case of unconstrained optimization, we refer to [18]. □

## 10. Conclusion

We have presented a nonconvex bundle method using downshifted tangents and the management of proximity control, which is adapted for nonconvex nonsmooth constrained optimization problems with lower-$C^1$ and upper-$C^1$ functions. A global convergence of the algorithm was proved in the sense that every accumulation point of the sequence of serious iterates is critical. Some satisfactory convergence results for practical purpose have been given as corollaries.

## Acknowledgements

The author thanks Professor Dominikus Noll for many useful discussions, and acknowledges the anonymous reviewers for valuable comments.

## References

1. P. Apkarian, D. Noll, and A. Rondepierre, *Mixed $H_2/H_\infty$ control via nonsmooth optimization*, SIAM J. Control Optim. **47** (2008), no. 3, 1516–1546.
2. F. H. Clarke, *A new approach to Lagrange multipliers*, Math. Oper. Res. **1** (1976), no. 2, 165–174.
3. ———, *Generalized gradients of Lipschitz functionals*, Adv. in Math. **40** (1981), no. 1, 52–67.
4. ———, *Optimization and nonsmooth analysis*, Canad. Math. Soc. Ser. Monogr. Adv. Texts, John Wiley & Sons, Inc., New York, 1983.

5. M. N. Dao and D. Noll, *Minimizing memory effects of a system*, Math. Control Signals Syst. (2014), doi: 10.1007/s00498-014-0135-9.
6. K. R. Davidson and A. P. Donsig, *Real analysis with real applications*, Prentice Hall, New Jersey, 2002.
7. M. Gabarrou, D. Alazard, and D. Noll, *Design of a flight control architecture using a non-convex bundle method*, Math. Control Signals Syst. **25** (2013), no. 2, 257–290.
8. J.-B. Hiriart-Urruty, *Mean value theorems in nonsmooth analysis*, Numer. Funct. Anal. Optim. **2** (1980), no. 1, 1–30.
9. K. C. Kiwiel, *An aggregate subgradient method for nonsmooth convex minimization*, Math. Programming **27** (1983), no. 3, 320–341.
10. G. Lebourg, *Valeur moyenne pour gradient généralisé (French)*, C. R. Acad. Sci. Paris Sér. A-B **281** (1975), no. 19, Ai, A795–A797.
11. ———, *Generic differentiability of Lipschitzian functions*, Trans. Amer. Math. Soc. **256** (1979), 125–144.
12. C. Lemaréchal, *Bundle methods in nonsmooth optimization*, Nonsmooth Optimization (Proc. IIASA Workshop, Laxenburg, 1977) (C. Lemaréchal and R. Mifflin, eds.), IIASA Proc. Ser., vol. 3, Pergamon, Oxford-Elmsford, 1978, pp. 79–102.
13. C. Lemaréchal, A. Nemirovskii, and Y. Nesterov, *New variants of bundle methods*, Math. Programming, Ser. B **69** (1995), no. 1, 111–147, Nondifferentiable and large-scale optimization (Geneva, 1992).
14. M. M. Mäkelä and P. Neittaanmäki, *Nonsmooth optimization: Analysis and algorithms with applications to optimal control*, World Scientific Publishing Co., Singapore, 1992.
15. R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control Optimization **15** (1977), no. 6, 959–972.
16. ———, *A modification and extension of Lemarechal's algorithm for nonsmooth minimization*, Nondifferential and Variational Techniques in Optimization (D. C. Sorensen and R. J.-B. Wets, eds.), Math. Programming Stud., vol. 17, North-Holland Publishing Co., Amsterdam, 1982, pp. 77–90.
17. D. Noll, *Cutting plane oracles to minimize non-smooth non-convex functions*, Set-Valued Var. Anal. **18** (2010), no. 3-4, 531–568.
18. ———, *Bundle method for non-convex minimization with inexact subgradients and function values*, Computational and Analytical Mathematics (D. H. Bailey et al., ed.), Springer Proc. Math. Stat., vol. 50, Springer, New York, 2013, pp. 555–592.
19. D. Noll, O. Prot, and A. Rondepierre, *A proximity control algorithm to minimize nonsmooth and nonconvex functions*, Pac. J. Optim. **4** (2008), no. 3, 571–604.
20. A. M. Ostrowski, *Solutions of equations in Euclidean and Banach spaces*, Pure and Applied Mathematics, vol. 9, Academic Press, New York-London, 1973.
21. E. Polak, *Optimization: Algorithms and consistent approximations*, Appl. Math. Sci., vol. 224, Springer-Verlag, New York, 1997.
22. R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, Springer-Verlag, Berlin, 1998.
23. J. E. Spingarn, *Submonotone subdifferentials of Lipschitz functions*, Trans. Amer. Math. Soc. **264** (1981), no. 1, 77–89.

# II

## Minimizing memory effects of a system *

Minh Ngoc Dao and Dominikus Noll

**Abstract.** Given a stable linear time-invariant system with tunable parameters, we present a method to tune these parameters in such a way that undesirable responses of the system to past excitations, known as system ringing, are avoided or reduced. This problem is addressed by minimizing the Hankel norm of the system, which quantifies the influence of past inputs on future outputs. We indicate by way of examples that minimizing the Hankel norm has a wide scope for possible applications. We show that the Hankel norm minimization program may be cast as an eigenvalue optimization problem, which we solve by a nonsmooth bundle algorithm with a local convergence certificate. Numerical experiments are used to demonstrate the efficiency of our approach.

**Keywords.** System ringing · system memory · Hankel norm · system reduction · controller design · system with tunable parameters.

### 1. Introduction

Ringing generally designates undesired responses of a system to past excitations. In electronic systems, ringing arises under various forms of noise, such as gate ringing in converters, undesired oscillations in digital controllers, or input ring back in clock signals. In mechanical systems, ringing effects, when combined with resonance, may accelerate breakdown. In audio systems, ringing may cause echoes to occur before transients.

In more abstract terms, ringing may be understood as a tendency of the system to store energy, which is retrieved later to produce undesired effects. One way to quantify this capacity uses the Hankel norm of a system, which measures the effects of past inputs on future outputs.

This paper focuses on the problem of minimizing system ringing by casting it as a Hankel norm minimization program. This leads to an eigenvalue optimization

*Paper published in Math. Control Signals Syst., doi: 10.1007/s00498-014-0135-9. Conference version published in Proc. Asian Control Conf. (ASCC), Istanbul, June 2013.

problem, for which we propose a nonsmooth bundle algorithm which assures convergence to a critical point from an arbitrary starting point. We demonstrate that a variety of problems such as Hankel synthesis, maximizing the memory of a system, and control of flow in a graph, can be interpreted as Hankel norm minimization programs and solved efficiently using the proposed algorithm.

There is a considerable body of literature dedicated to Hankel norm system reduction, the original contribution being [12]. Our present approach is complementary to this classical line, as we focus on Hankel norm optimization problems which cannot be solved by linear algebra techniques. This makes our method closer in spirit to $H_2$- or $H_\infty$-controller or filter design [26].

The structure of the paper is as follows. After presenting the problem in abstract form in Sect. 2, we show in Sect. 3 how it can be cast as a nonconvex eigenvalue optimization program. Section 4 describes how Clarke subgradients of a Hankel norm objective can be computed. In Sect. 5 we extend the Hankel norm to systems with direct transmission in a physically meaningful way. Sections 6, 7 present typical applications for the purpose of motivation of the Hankel minimization problem. Section 8 discusses a proximal bundle algorithm used to solve the Hankel norm minimization program. We propose a smooth relaxation of the Hankel norm in Sect. 9. Experiments with typical applications are given in Sect. 10.

### Notation

Terminology in nonsmooth optimization is covered by [8], system theory by [26]. Following the latter reference, given a transfer matrix function $G(s) = C(sI - A)^{-1}B + D$, we use the standard notations

$$G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \text{ or } G = (A, B, C, D)$$

to indicate that

$$G : \left\{ \begin{array}{ll} \dot{x} & = Ax + Bw \\ z & = Cx + Dw \end{array} \right.$$

is a state-space realization of $z(s) = G(s)w(s)$. Similar notations apply to discrete time systems.

We shall work in the set of rectangular matrices with the corresponding scalar product $\langle M, N\rangle = \mathrm{Tr}(M^\top N) = \mathrm{Tr}(N^\top M)$, where $M^\top$ and $\mathrm{Tr}(M)$ are transpose and trace of a matrix. For symmetric matrices, $M \succ 0$ means positive definite, $M \succeq 0$ positive semidefinite.

## 2. Hankel norm minimization

Consider a linear time-invariant system

$$G : \left\{ \begin{array}{ll} \dot{x} & = Ax + Bw \\ z & = Cx \end{array} \right.$$

with state $x \in \mathbb{R}^{n_x}$, input $w \in \mathbb{R}^m$, and output $z \in \mathbb{R}^p$. Suppose $G$ is internally stable in the sense that all eigenvalues of $A$ have negative real part. If we think of $w(t)$ as an excitation at the input which acts over the time period $0 \leqslant t \leqslant T$

with dynamics started at $x(0) = 0$, then the ring of the system after the excitation has stopped at time $T$ is $z(t)$ for $t > T$. If signals are measured in the energy norm, this leads to the definition of the Hankel norm of an internally stable system $G = (A, B, C)$ with input $w$ and output $z = Gw$ as

$$\|G\|_H = \sup_{T>0} \left\{ \left( \int_T^\infty z^\top z \, \mathrm{d}t \right)^{1/2} : \int_0^T w^\top w \, \mathrm{d}t \leqslant 1, w(t) = 0 \text{ for } t > T \right\}.$$

For the discrete time case, the Hankel norm of an internally stable system

$$G : \left\{ \begin{array}{ll} x(t+1) & = Ax(t) + Bw(t) \\ z(t) & = Cx(t) \end{array} \right.$$

is given by

$$\|G\|_H = \sup_{T>0} \left\{ \left( \sum_{t=T}^\infty z(t)^\top z(t) \right)^{1/2} : \sum_{t=0}^T w(t)^\top w(t) \leqslant 1, w(t) = 0 \text{ for } t > T \right\},$$

where now internally stable means that all eigenvalues of $A$ have magnitude $< 1$, and where it is again understood that $z = Gw$. A formula which works in both cases is

$$\|G\|_H = \sup_{T>0} \left\{ \|z\|_{2,[T,\infty)} : \|w\|_{2,[0,T]} \leqslant 1, w \in L^2[0,T], w(t) = 0, t > T \right\}. \tag{1}$$

Note that the system $G$ in the above definition has no direct transmission $D$. This accounts for the fact, proved in Lemma 5.1 in Sect. 5, that $D$ causes no memory effects, and is therefore not seen by the Hankel norm (1). In consequence, on the space of systems $G = (A, B, C, D)$ with direct transmission, $\|\cdot\|_H$ is only a semi-norm and not a norm.

By definition, the Hankel norm can be interpreted as a measure of the effects of past inputs, that is, the memory of the system, on the states and future outputs. Here, we are interested in systems $G(\mathbf{x})$ with tunable parameters $\mathbf{x} \in \mathbb{R}^n$, where the matrices $A(\mathbf{x}), B(\mathbf{x}), C(\mathbf{x})$ depend smoothly on a design parameter $\mathbf{x}$ varying in $\mathbb{R}^n$ or in some constrained subset of $\mathbb{R}^n$. Our goal is to tune $\mathbf{x}$ such that system ringing is avoided or reduced while internal stability of the system is guaranteed. This leads to the following Hankel norm minimization program

$$\begin{array}{ll} \text{minimize} & \|G(\mathbf{x})\|_H \\ \text{subject to} & G(\mathbf{x}) \text{ internally stable} \\ & \mathbf{x} \in \mathbb{R}^n. \end{array} \tag{2}$$

We will discuss various instances, where program (2) may be of interest. Then, we present a nonsmooth optimization method based on techniques from eigenvalue optimization to solve (2), and discuss a smooth relaxation motivated by a result of Nesterov in [15].

## 3. Representation of the Hankel norm

A representation of the Hankel norm $\|\cdot\|_H$ amenable to computations is obtained through the observability and controllability Gramians, defined in [26, Section 3.8]. Based on the results in [12, Section 2.3], see also [26, Theorem 8.1], we have the following

**Lemma 3.1.** *Let $G = (A, B, C)$ be an internally stable linear time-invariant system with input $w$ and output $z$, and let $\Gamma_G : L^2(-\infty, 0] \longrightarrow L^2[0, \infty)$ be the Hankel operator associated with $G$, defined by*

$$(\Gamma_G w)(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bw(\tau)\mathrm{d}\tau,\ t \geqslant 0.$$

*Then, the following definitions are equivalent:*

(i) $\|G\|_H = \sup_{T>0} \left\{ \|z\|_{2,[T,\infty)} : \|w\|_{2,[0,T]} \leqslant 1, w \in L^2[0,T], w(t) = 0, t > T \right\}$.
(ii) $\|G\|_H = \|\Gamma_G\| = \sup \left\{ \|\Gamma_G w\|_{2,[0,\infty)} : \|w\|_{2,(-\infty,0]} \leqslant 1, w \in L^2(-\infty, 0] \right\}$.
(iii) $\|G\|_H = \sqrt{\lambda_1(XY)}$, *where $\lambda_1$ denotes the maximum eigenvalue of a matrix, and $X, Y$ are the controllability and observability Gramians of the system.*

*Proof.* We assume $x(-\infty) = 0$ for the Hankel operator $\Gamma_G$ and obtain

$$z(t) = \int_{-\infty}^{t} Ce^{A(t-\tau)}Bw(\tau)\mathrm{d}\tau.$$

If we now focus on input signals $w_-$ that live for times $t \leqslant 0$ and vanish for $t > 0$, then the output restricted to $t \geqslant 0$ is

$$z_+(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bw_-(\tau)\mathrm{d}\tau = \Gamma_G w_-,\ t \geqslant 0.$$

Assuming $x(0) = 0$ in (i), it now follows from the time-invariance that

$$\sup_{\substack{T>0 \\ 0\neq w\in L^2[0,T] \\ w(t)=0,\, t>T}} \frac{\|z\|_{2,[T,\infty)}}{\|w\|_{2,[0,T]}} = \sup_{\substack{T>0 \\ 0\neq w\in L^2[-T,0] \\ w(t)=0,\, t>0}} \frac{\|z\|_{2,[0,\infty)}}{\|w\|_{2,[-T,0]}} = \sup_{\substack{0\neq w\in L^2(-\infty,0] \\ w(t)=0,\, t>0}} \frac{\|z\|_{2,[0,\infty)}}{\|w\|_{2,(-\infty,0]}}$$

$$= \sup_{0\neq w_-\in L^2(-\infty,0]} \frac{\|z_+\|_{2,[0,\infty)}}{\|w_-\|_{2,(-\infty,0]}} = \|\Gamma_G\|.$$

This gives the equivalence of (i) and (ii). Next, we have

$$\begin{aligned}
\langle w, \Gamma_G^* z\rangle_{L^2(-\infty,0]} &= \langle \Gamma_G w, z\rangle_{L^2[0,\infty)} \\
&= \int_0^\infty \left( \int_{-\infty}^0 w(\tau)^\top B^\top e^{A^\top(t-\tau)} C^\top \mathrm{d}\tau \right) z(t)\mathrm{d}t \\
&= \int_{-\infty}^0 w(\tau)^\top \left( \int_0^\infty B^\top e^{A^\top(t-\tau)} C^\top z(t)\mathrm{d}t \right) \mathrm{d}\tau,
\end{aligned}$$

which implies

$$(\Gamma_G^* z)(\tau) = \int_0^\infty B^\top e^{A^\top(t-\tau)} C^\top z(t)\mathrm{d}t,\ \tau \leqslant 0.$$

Note that the operator norm of $\Gamma_G$ is equal to its maximum singular value. Therefore, to complete the proof, we show that $\sigma_i^2(\Gamma_G) = \lambda_i(XY)$, where $\sigma_i(\cdot)$ and $\lambda_i(\cdot)$ denote, respectively, the $i$th singular value and $i$th eigenvalue of an operator or matrix. Suppose $\sigma$ is a nonzero singular value of $\Gamma_G$, and $w$ is an eigenvector corresponding to the eigenvalue $\sigma^2$ of $\Gamma_G^*\Gamma_G$, i.e., $\Gamma_G^*\Gamma_G w = \sigma^2 w$. Setting $z(t) = (\Gamma_G w)(t) = Ce^{At}x_0$ with $x_0 = \int_{-\infty}^0 e^{-A\tau}Bw(\tau)\mathrm{d}\tau$, and noting by [26, Lemma 3.18] that

$$X = \int_0^\infty e^{At}BB^\top e^{A^\top t}\mathrm{d}t,\ Y = \int_0^\infty e^{A^\top t}C^\top Ce^{At}\mathrm{d}t,$$

we have

$$\sigma^2 w = \Gamma_G^* z = B^\top e^{-A^\top \tau} \int_0^\infty e^{A^\top t} C^\top z(t) \mathrm{d}t$$
$$= B^\top e^{-A^\top \tau} \int_0^\infty e^{A^\top t} C^\top C e^{At} x_0 \mathrm{d}t = B^\top e^{-A^\top \tau} Y x_0.$$

It follows that

$$\sigma^2 x_0 = \int_{-\infty}^0 e^{-A\tau} B \sigma^2 w(\tau) \mathrm{d}\tau = \int_{-\infty}^0 e^{-A\tau} B B^\top e^{-A^\top \tau} Y x_0 \mathrm{d}\tau = XY x_0.$$

Moreover, $x_0 \neq 0$ since otherwise $\sigma^2 w = 0$, which is impossible. Thus, $\sigma^2$ is an eigenvalue of $XY$. Conversely, if $\sigma^2 \neq 0$ is an eigenvalue and $x_0 \neq 0$ is a corresponding eigenvector of $XY$, i.e., $XYx_0 = \sigma^2 x_0$, then by setting $w = B^\top e^{-A\tau} Y x_0$ we obtain $w \neq 0$ and $\Gamma_G^* \Gamma_G w = \sigma^2 w$. Hence, $\sigma_i^2(\Gamma_G) = \lambda_i(XY)$, and so

$$\|\Gamma_G\| = \sigma_1(\Gamma_G) = \sqrt{\lambda_1(XY)}.$$

The lemma is proved. □

Lemma 3.1 shows that the Hankel norm can be considered as a measure of controllability and observability of the system, and that it does not depend on the state-space representation of the system. It is now clear that problem (2) may be cast as an eigenvalue optimization program. In the sequel, we examine how this problem can be solved algorithmically.

## 4. Subgradients of the Hankel norm

In this section, we compute Clarke subgradients [8, Section 2.1] of the nonconvex composite function $f(\mathbf{x}) = \|G(\mathbf{x})\|_H^2$. This is a fundamental tool for our optimization method.

Let $G(\mathbf{x})$ be a linear time-invariant system with state-space realization $(A(\mathbf{x}), B(\mathbf{x}), C(\mathbf{x}))$ depending smoothly on a design parameter $\mathbf{x} \in \mathbb{R}^n$. Let $X(\mathbf{x}), Y(\mathbf{x})$ be the controllability and observability Gramians. Suppose the maximum eigenvalue $\lambda_1(Z(\mathbf{x}))$ of the matrix $Z(\mathbf{x}) = X(\mathbf{x})^{\frac{1}{2}} Y(\mathbf{x}) X(\mathbf{x})^{\frac{1}{2}}$ has multiplicity $r(\mathbf{x})$, and let $R = R(\mathbf{x})$ be a matrix whose columns form an orthonormal basis of the eigenspace associated with $\lambda_1(Z(\mathbf{x}))$. For any matrix function $M(\mathbf{x})$, put $M_k(\mathbf{x}) = \frac{\partial M(\mathbf{x})}{\partial \mathbf{x}_k}$ and write $M_k^{\frac{1}{2}}$ for $(M^{\frac{1}{2}})_k$, $k = 1, \ldots, n$. We have the following

**Proposition 4.1.** *The function $f(\mathbf{x}) = \|G(\mathbf{x})\|_H^2$ is well defined and locally Lipschitz on the set $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : A(\mathbf{x})$ stable$\}$. In addition, for every $\mathbf{x}$ in the set $\mathcal{S}_0 = \{\mathbf{x} \in \mathcal{S} : (A(\mathbf{x}), B(\mathbf{x}))$ controllable$\}$ the Clarke subgradients of $f$ at $\mathbf{x}$ have the form*

$$g_U = \begin{bmatrix} \mathrm{Tr}(UR^\top Z_1(\mathbf{x})R) & \ldots & \mathrm{Tr}(UR^\top Z_n(\mathbf{x})R) \end{bmatrix}^\top, \tag{3}$$

*where $U$ is symmetric of size $r \times r$, $U \succeq 0$, $\mathrm{Tr}(U) = 1$, and where the partial derivatives $Z_k(\mathbf{x}), k = 1, \ldots, n$ are given by*

$$Z_k(\mathbf{x}) = X_k^{\frac{1}{2}}(\mathbf{x}) Y X^{\frac{1}{2}} + X^{\frac{1}{2}} Y_k(\mathbf{x}) X^{\frac{1}{2}} + X^{\frac{1}{2}} Y X_k^{\frac{1}{2}}(\mathbf{x}). \tag{4}$$

*Here, $X_k(\mathbf{x})$, $Y_k(\mathbf{x})$ and $X_k^{\frac{1}{2}}(\mathbf{x})$ are the solutions of the following Lyapunov equations*

$$AX_k(\mathbf{x}) + X_k(\mathbf{x})A^\top = -A_k(\mathbf{x})X - XA_k(\mathbf{x})^\top - B_k(\mathbf{x})B^\top - BB_k(\mathbf{x})^\top, \tag{5}$$

$$A^\top Y_k(\mathbf{x}) + Y_k(\mathbf{x})A = -A_k(\mathbf{x})^\top Y - YA_k(\mathbf{x}) - C_k(\mathbf{x})^\top C - C^\top C_k(\mathbf{x}), \tag{6}$$

$$X^{\frac{1}{2}}X_k^{\frac{1}{2}}(\mathbf{x}) + X_k^{\frac{1}{2}}(\mathbf{x})X^{\frac{1}{2}} = X_k(\mathbf{x}). \tag{7}$$

*Proof.* (1) By Lemma 3.1,

$$f(\mathbf{x}) = \|G(\mathbf{x})\|_H^2 = \lambda_1(X(\mathbf{x})Y(\mathbf{x})),$$

where the Gramians $X(\mathbf{x})$ and $Y(\mathbf{x})$ depend on the tunable parameters $\mathbf{x}$ and are the solutions of the Lyapunov equations

$$A(\mathbf{x})X + XA(\mathbf{x})^\top + B(\mathbf{x})B(\mathbf{x})^\top = 0, \tag{8}$$

$$A(\mathbf{x})^\top Y + YA(\mathbf{x}) + C(\mathbf{x})^\top C(\mathbf{x}) = 0. \tag{9}$$

Note that despite the symmetry of $X$ and $Y$ the product $XY$ is not necessarily symmetric, but stability of $A(\mathbf{x})$ guarantees $X \succeq 0$, $Y \succeq 0$ in (8), (9), so that we can write

$$\lambda_1(XY) = \lambda_1(X^{\frac{1}{2}}YX^{\frac{1}{2}}) = \lambda_1(Y^{\frac{1}{2}}XY^{\frac{1}{2}}),$$

which brings us back to the realm of eigenvalue theory of symmetric matrices. By positive semidefiniteness of $X(\mathbf{x})$ and $Y(\mathbf{x})$, the function $f$ is now well defined on $\mathcal{S}$.

(2) Let us next prove that $f$ is locally Lipschitz on $\mathcal{S}$. Using the Kronecker product [3], Eq. (8) can be written as

$$(I \otimes A(\mathbf{x}) + A(\mathbf{x}) \otimes I)\mathrm{vec}(X(\mathbf{x})) = -\mathrm{vec}(B(\mathbf{x})B(\mathbf{x})^\top),$$

where $I$ is a conformable identity matrix, and where $\mathrm{vec}(\cdot)$ vectorizes a matrix by stacking its columns in order. Since $A(\mathbf{x})$ is smooth in $\mathbf{x}$ and $M(\mathbf{x}) = (I \otimes A(\mathbf{x}) + A(\mathbf{x}) \otimes I)$ is invertible by the stability of $A(\mathbf{x})$, $M(\mathbf{x})^{-1}$ is also smooth in $\mathbf{x}$, and since $B(\mathbf{x})$ depends smoothly on $\mathbf{x}$, then so does $\mathrm{vec}(X(\mathbf{x})) = -M(\mathbf{x})^{-1}\mathrm{vec}\left(B(\mathbf{x})B(\mathbf{x})^\top\right)$. A similar argument shows smooth dependence of $Y(\mathbf{x})$ on $\mathbf{x}$. This can also be justified based on the explicit formulas

$$X(\mathbf{x}) = \int_0^\infty e^{A(\mathbf{x})t}B(\mathbf{x})B(\mathbf{x})^\top e^{A(\mathbf{x})^\top t}\mathrm{d}t, Y(\mathbf{x}) = \int_0^\infty e^{A(\mathbf{x})^\top t}C(\mathbf{x})^\top C(\mathbf{x})e^{A(\mathbf{x})t}\mathrm{d}t$$

(see e.g., [26, Lemmas 2.7 and 3.18]), where uniform convergence of these integrals on any bounded set of $\mathbf{x}$ gives differentiability in $\mathbf{x}$. We infer that the coefficients of the characteristic polynomial of $X(\mathbf{x})Y(\mathbf{x})$ also depend smoothly on $\mathbf{x}$. Since this characteristic polynomial is hyperbolic, that is, has only real roots, we may invoke the multi-parameter version of Bronstein's theorem [6], for which an elegant proof is given in [19, Theorem 2], to conclude that $f(\mathbf{x}) = \lambda_1(X(\mathbf{x})Y(\mathbf{x}))$ is locally Lipschitz on $\mathcal{S}$.

(3) Let us finally establish formula (3) for the subdifferential $\partial f(\mathbf{x})$ at points $\mathbf{x} \in \mathcal{S}_0$. By the above argument, $f(\mathbf{x}) = \lambda_1(Z(\mathbf{x}))$. Observe that controllability of $(A(\mathbf{x}), B(\mathbf{x}))$ implies that $X(\mathbf{x})$ is positive definite [26, Theorem 3.1], and since the operator $X \to X^{\frac{1}{2}}$ is smooth on the set of matrices $X \succ 0$, the chain rule gives smoothness of $\mathbf{x} \to X^{\frac{1}{2}}(\mathbf{x})$, and so of $Z(\mathbf{x}) = X^{\frac{1}{2}} Y X^{\frac{1}{2}}$, on $\mathcal{S}_0$.

Applying [18, Theorem 3], the Clarke subgradients of $f$ at $\mathbf{x}$ are of the form $g_U = \begin{bmatrix} g_1 & \dots & g_n \end{bmatrix}^\top$, where

$$g_k = \left\langle U, R^\top Z_k(\mathbf{x}) R \right\rangle = \mathrm{Tr}(U R^\top Z_k(\mathbf{x}) R)$$

for $U$ symmetric of size $r \times r$, $U \succeq 0$, $\mathrm{Tr}(U) = 1$. It now remains to calculate $Z_k(\mathbf{x}), k = 1, \dots, n$. We first have (4) by the definition of $Z$. Taking derivatives with respect to $\mathbf{x}$ on both sides of (8)–(9), we get (5)–(6), and then also $X_k(\mathbf{x})$, $Y_k(\mathbf{x})$. Finally, to compute $X_k^{\frac{1}{2}}(\mathbf{x})$, we use (7), which is obtained by differentiating $X^{\frac{1}{2}} X^{\frac{1}{2}} = X$. Altogether, we obtain Clarke subgradients of $f$ at each $\mathbf{x}$ due to (3)–(9).

□

*Remark* 1. Formula (3) also holds if controllability of $(A(\mathbf{x}), B(\mathbf{x}))$ is replaced by observability of $(A(\mathbf{x}), C(\mathbf{x}))$ (cf. [26, Definition 3.4]). Here, we work with $Z = Y^{\frac{1}{2}} X Y^{\frac{1}{2}}$ instead.

*Remark* 2. In the discrete time case, the Gramians $X(\mathbf{x})$ and $Y(\mathbf{x})$ are the solutions of the discrete Lyapunov equations

$$A(\mathbf{x}) X A(\mathbf{x})^\top - X + B(\mathbf{x}) B(\mathbf{x})^\top = 0,$$
$$A(\mathbf{x})^\top Y A(\mathbf{x}) - Y + C(\mathbf{x})^\top C(\mathbf{x}) = 0,$$

so that $X_k(\mathbf{x})$ and $Y_k(\mathbf{x})$ are solutions, respectively, of the following equations

$$A X_k(\mathbf{x}) A^\top - X_k(\mathbf{x}) = -A_k(\mathbf{x}) X A^\top - A X A_k(\mathbf{x})^\top - B_k(\mathbf{x}) B^\top - B B_k(\mathbf{x})^\top,$$
$$A^\top Y_k(\mathbf{x}) A - Y_k(\mathbf{x}) = -A_k(\mathbf{x})^\top Y A - A^\top Y A_k(\mathbf{x}) - C_k(\mathbf{x})^\top C - C^\top C_k(\mathbf{x}).$$

*Remark* 3. Subgradients of $f$ at $\mathbf{x} \in \mathcal{S} \setminus \mathcal{S}_0$ are no longer represented by (3), since the solution of (7) need not exist, as only $X^{\frac{1}{2}} \succeq 0$ is guaranteed. Nonetheless, by Clarke subdifferentiability at points $\mathbf{x} \in \mathcal{S} \setminus \mathcal{S}_0$ proved above, we can be sure that for every sequence $\mathbf{x}_k \in \mathcal{S}_0$ converging to $\mathbf{x} \in \mathcal{S} \setminus \mathcal{S}_0$ and $g_k \in \partial f(\mathbf{x}_k)$ computed via (3), the $g_k$ stay bounded and each of their accumulation points $g$ is an element of $\partial f(\mathbf{x})$. This guarantees stability of our numerical procedure even when iterates get close to the set $\mathcal{S} \setminus \mathcal{S}_0$.

*Remark* 4. Practical parametrizations $G(\mathbf{x})$ use elementary computable operations, which can be expressed in mathematical terms by assuming that $A(\mathbf{x}), B(\mathbf{x}), C(\mathbf{x})$ are smooth *definable* functions of $\mathbf{x}$ in the sense of [25, Chap. 1, Sect. 5.3]. In that case, one can say a little more about the behavior of $f$ at points $\mathbf{x} \in \mathcal{S}$. Namely, it then follows from [21, Theorem 4.12] that for every smooth definable curve $\mathbf{x}(t) \in \mathcal{S}$ the eigenvalues $\lambda_i(t) = \lambda_i(X(\mathbf{x}(t)) Y(\mathbf{x}(t)))$ are smooth functions of $t$, so that $f(\mathbf{x}(t))$ is a finite maximum of smooth functions of $t$. On $\mathcal{S}_0$ this property is a consequence of symmetric eigenvalue theory, which is true without the definability hypothesis.

Note that this does not mean that $f$ is a finite maximum of smooth functions of $\mathbf{x} \in \mathbb{R}^n$, but it nonetheless indicates a favorable structure.

## 5. An extension of the Hankel norm

Lemma 3.1 shows why the Hankel norm is only a semi-norm on the space of internally stable systems $G$. It does not see a direct transmission $D$ from $w$ to $z$, as the latter does not create memory transmitted from the past to the future. This rises the question how to assess a direct transmission block in the context of (1) or (2). Namely, in some applications, attributing no cost to a block $D(\mathbf{x})$ which is free to vary with the tunable parameters $\mathbf{x}$ bears the risk that optimization favors a solution with a high energy direct transmission.

It is well known that $\|G\|_H \leqslant \|G\|_\infty$ in the case $D = 0$ (See e.g., [5, Sect. 5.5]), and this may guide us to define an extension. Note first that

**Lemma 5.1.** $\|(A, B, C)\|_H \leqslant \|(A, B, C, D)\|_\infty$ *for every internally stable system* $G = (A, B, C, D)$.

*Proof.* Let $G^0 = (A, B, C)$ be the system where the direct transmission is ignored. Consider an input $w$ with $w(t) = 0$ for $t > T$, and let $z^0 = G^0 w$, $z = Gw$. Then, $z(t) = z^0(t)$ for $t > T$, because the direct transmission creates no memory, and since $w(t) = 0$ for $t > T$, its influence on the output ends at $T$. Combining this with $\|w\|_{2,[0,T]} = \|w\|_2$ and $\|z\|_{2,[T,\infty)} \leqslant \|z\|_2$, we obtain

$$\begin{aligned}\|(A, B, C)\|_H &= \sup_{\substack{T>0 \\ 0\neq w\in L^2[0,T] \\ w(t)=0,\, t>T}} \frac{\|z\|_{2,[T,\infty)}}{\|w\|_{2,[0,T]}} \leqslant \sup_{\substack{T>0 \\ 0\neq w\in L^2[0,T] \\ w(t)=0,\, t>T}} \frac{\|z\|_2}{\|w\|_2} \\ &\leqslant \sup_{w\neq 0} \frac{\|z\|_2}{\|w\|_2} = \|(A, B, C, D)\|_\infty.\end{aligned}$$

□

This suggests the following extension of Hankel norm $\|\cdot\|_H$ to systems $G = (A, B, C, D)$ with direct transmission $D$.

**Definition 5.2.** Let $G = (A, B, C, D)$ be an internally stable linear time-invariant system. Then,

$$\|G\|_H = \max\{\|(A, B, C)\|_H, \sigma_1(D)\} \tag{10}$$

is called the extended Hankel norm of the system. Here, $\sigma_1$ denotes the maximum singular value of a matrix. □

This definition agrees with the usual Hankel norm for a system without direct transmission, and also preserves the inequality $\|G\|_H \leqslant \|G\|_\infty$, since the term $\sigma_1(D)$ is part of the maximum $\|G\|_\infty = \max_\omega \sigma_1(G(j\omega))$ at $\omega = \infty$.

As the proof of Lemma 5.1 shows, a direct transmission does not change the value of $\|\cdot\|_H$ defined according to (1). In the sequel, we therefore adopt the convention that in the case $D \neq 0$, $\|(A, B, C)\|_H$ is the usual Hankel norm, where the direct transmission is ignored, while $\|(A, B, C, D)\|_H$ is the extended Hankel norm.

An advantage of (10) is that the new function is still a maximum eigenvalue function. Namely, stability of $G$ implies positive semidefiniteness of the Gramians $X$ and $Y$, and so

$$\|G\|_H^2 = \max\left\{\lambda_1(X^{\frac{1}{2}}YX^{\frac{1}{2}}), \lambda_1(D^\top D)\right\} = \lambda_1 \begin{bmatrix} X^{\frac{1}{2}}YX^{\frac{1}{2}} & 0 \\ 0 & D^\top D \end{bmatrix}. \tag{11}$$

Proceeding as in the proof of Proposition 4.1, we get immediately the following

**Corollary 5.3.** *Let $G(\mathbf{x})$ be a linear time-invariant system depending smoothly on $\mathbf{x} \in \mathcal{S}$ with $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : A(\mathbf{x}) \text{ stable}\}$. Suppose the maximum eigenvalue $\lambda_1(\mathcal{Z}(\mathbf{x}))$ of the matrix*

$$\mathcal{Z}(\mathbf{x}) = \begin{bmatrix} X(\mathbf{x})^{\frac{1}{2}}Y(\mathbf{x})X(\mathbf{x})^{\frac{1}{2}} & 0 \\ 0 & D(\mathbf{x})^\top D(\mathbf{x}) \end{bmatrix}$$

*has multiplicity $r = r(\mathbf{x})$, and $R = R(\mathbf{x})$ is a matrix whose columns form an orthonormal basis of the eigenspace associated with $\lambda_1(\mathcal{Z}(\mathbf{x}))$. With the notations of Proposition 4.1, the function $f(\mathbf{x}) = \|G(\mathbf{x})\|_H^2$ is locally Lipschitz on $\mathcal{S}$ and its Clarke subgradients on $\mathcal{S}_0 = \{\mathbf{x} \in \mathcal{S} : (A(\mathbf{x}), B(\mathbf{x})) \text{ controllable}\}$ have the form*

$$g_U = \begin{bmatrix} \operatorname{Tr}(UR^\top \mathcal{Z}_1(\mathbf{x})R) & \dots & \operatorname{Tr}(UR^\top \mathcal{Z}_n(\mathbf{x})R) \end{bmatrix}^\top,$$

*for $U$ symmetric of size $r \times r$, $U \succeq 0$, $\operatorname{Tr}(U) = 1$, where the partial derivatives $\mathcal{Z}_k(\mathbf{x}), k = 1, \dots, n$ are given by*

$$\mathcal{Z}_k(\mathbf{x}) = \begin{bmatrix} Z_k(\mathbf{x}) & 0 \\ 0 & D_k(\mathbf{x})^\top D(\mathbf{x}) + D(\mathbf{x})^\top D_k(\mathbf{x}) \end{bmatrix}$$

*and the $Z_k(\mathbf{x})$ are defined in Proposition 4.1.* □

To justify the use of (10) rigorously, we consider the extended Hankel norm minimization program (2) based on (10), and compare it to the following constraint program

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) = \|(A(\mathbf{x}), B(\mathbf{x}), C(\mathbf{x}))\|_H \\ \text{subject to} & h(\mathbf{x}) = \sigma_1\left(D(\mathbf{x})\right) \leqslant \eta. \end{array} \tag{12}$$

For the following, recall from [13] that $\mathbf{x}^* \in \mathbb{R}^n$ is called a Fritz John critical point of the constraint program $\min\{f(\mathbf{x}) : h(\mathbf{x}) \leqslant \eta\}$ if there exist multipliers $\lambda_0^* \geqslant 0$, $\lambda_1^* \geqslant 0$, *not both zero*, such that

$$0 \in \lambda_0^* \partial f(\mathbf{x}^*) + \lambda_1^* \partial h(\mathbf{x}^*), \quad h(\mathbf{x}^*) \leqslant \eta, \quad \lambda_1^*\left(h(\mathbf{x}^*) - \eta\right) = 0.$$

If in addition $\lambda_0^* > 0$, then $\mathbf{x}^*$ is called a Karush–Kuhn–Tucker point. Remember that every local minimum $\mathbf{x}^*$ of the constraint program is automatically a Fritz John critical point, while it will in general only be a Karush–Kuhn–Tucker point if an additional constraint qualification is satisfied [13, Chapter 7]. For later on, we call $\mathbf{x}^*$ a critical point of constraint violation if $0 \in \partial h(\mathbf{x}^*)$ and $h(\mathbf{x}^*) > \eta$.

With these preparations, we have the following

**Proposition 5.4.** *Let $\mathbf{x}^*$ be a critical point of the extended Hankel norm minimization program (2) with (10). Then, $\mathbf{x}^*$ is a Fritz John critical point of program (12) for a suitable choice of $\eta$. More precisely, $\mathbf{x}^*$ is either a Karush–Kuhn–Tucker point of (12), or a critical point of $h(\mathbf{x}) = \sigma_1(D(\mathbf{x}))$ alone.*

*Proof.* Note that $\|G(\mathbf{x})\|_H = \max\{f(\mathbf{x}), h(\mathbf{x})\}$. Now, if $\mathbf{x}^*$ is a critical point of $\|G(\mathbf{x})\|_H$, then we have three possibilities, $f(\mathbf{x}^*) > h(\mathbf{x}^*)$, $f(\mathbf{x}^*) = h(\mathbf{x}^*)$, or $f(\mathbf{x}^*) < h(\mathbf{x}^*)$. In the first case, $\mathbf{x}^*$ is a critical point of $f$ alone, hence also a Karush–Kuhn–Tucker point of (12). The third case corresponds to a critical point of $h$ alone. In the case of equality, the situation is more complex. There exist multipliers $\lambda_0^* \geqslant 0$, $\lambda_1^* \geqslant 0$, not both zero, such that $0 \in \lambda_0^* \partial f(\mathbf{x}^*) + \lambda_1^* \partial h(\mathbf{x}^*)$. If $\lambda_0^* = 0$ then $\lambda_1^* \neq 0$ and $0 \in \partial h(\mathbf{x}^*)$, so $\mathbf{x}^*$ is a critical point of $h$. In case $\lambda_0^* \neq 0$, we have $0 \in \partial f(\mathbf{x}^*) + (\lambda_1^*/\lambda_0^*)\partial h(\mathbf{x}^*)$. This is the first part of the Karush–Kuhn–Tucker conditions. If we put $\eta = f(\mathbf{x}^*)$, then we also get the second half. That completes the argument. □

*Remark* 5. Suppose we solve program $\min\{f(\mathbf{x}) : h(\mathbf{x}) \leqslant \eta\}$ starting at an infeasible point $h(\mathbf{x}^1) > \eta$, then we will usually try to minimize $h$ alone to find a feasible iterate. Suppose a descent method used to minimize $h$ runs into a local minimum $\mathbf{x}^*$ of $h$ satisfying $h(\mathbf{x}^*) > \eta$. Such a *local minimum of constraint violation* indicates a failure, since nothing better will be found in a neighborhood of $\mathbf{x}^*$ due to local optimality, so that the search for a feasible point has to be stared anew elsewhere; cf. [20, Section 2.2] for this theme complex.

By Proposition 5.4 we can now interpret minimization of the extended Hankel norm (2) with (10) as a trade-off between minimizing the memory effects of $(A(\mathbf{x}), B(\mathbf{x}), C(\mathbf{x}))$, subject to a constraint $\sigma_1(D(\mathbf{x})) \leqslant \eta$, or dually, as of minimizing $\sigma_1(D(\mathbf{x}))$ subject to a constraint on the memory effects of $G(\mathbf{x})$. Since $f(\mathbf{x})$ is a valid measure of the memory or ringing effects of $G(\mathbf{x})$, such an interpretation is physically meaningful.

We conclude this section by showing that the Hankel norm is amenable to optimization techniques, as this will be needed later. According to Spingarn [24] a function $f : U \to \mathbb{R}$, where $U$ is an open set in $\mathbb{R}^n$, is *lower-$C^1$* on $U$, if for each $\mathbf{x}_0 \in U$, there are a compact space $K$, a neighborhood $V$ of $\mathbf{x}_0$, and a jointly continuous function $F : V \times K \to \mathbb{R}$ whose partial derivative $D_{\mathbf{x}}F$ with respect to $\mathbf{x}$ exists and is jointly continuous, such that $f(\mathbf{x}) = \max_{\mathbf{z}\in K} F(\mathbf{x}, \mathbf{z})$ for all $\mathbf{x} \in V$.

**Proposition 5.5.** *Let $G(\mathbf{x}) = (A(\mathbf{x}), B(\mathbf{x}), C(\mathbf{x}), D(\mathbf{x}))$ be a linear time-invariant system depending smoothly on the set $\mathcal{S}_0$ of all $\mathbf{x} \in \mathbb{R}^n$ such that $A(\mathbf{x})$ is stable and $(A(\mathbf{x}), B(\mathbf{x}))$ is controllable or $(A(\mathbf{x}), C(\mathbf{x}))$ is observable. Then, $f(\mathbf{x}) = \|G(\mathbf{x})\|_H^2$ is lower-$C^1$ on $\mathcal{S}_0$.*

*Proof.* For each $\mathbf{x} \in \mathcal{S}_0$, according to (11) and using the Rayleigh quotient,

$$f(\mathbf{x}) = \lambda_1(\mathcal{Z}(\mathbf{x})) = \max_{\|\mathbf{z}\|=1} \mathbf{z}^\top \mathcal{Z}(\mathbf{x})\mathbf{z},$$

where $\mathcal{Z}$ is symmetric and depends smoothly on $\mathbf{x}$. Set $K = \{\mathbf{z} \in \mathbb{R}^m : \|\mathbf{z}\| = 1\}$ and $F(\mathbf{x}, \mathbf{z}) = \mathbf{z}^\top \mathcal{Z}(\mathbf{x})\mathbf{z}$, then $K$ is compact, $f(\mathbf{x}) = \max_{\mathbf{z}\in K} F(\mathbf{x}, \mathbf{z})$, and both $F$ and its partial derivatives $F_{\mathbf{x}}$ are jointly continuous on $\mathcal{S}_0 \times K$ and smooth in $\mathbf{x}$. Therefore, $f$ is lower-$C^1$ on $\mathcal{S}_0$. □

## 6. Hankel synthesis

The first application of program (2) we consider is output feedback controller synthesis, where performance is assessed by the Hankel norm. Consider a linear

time-invariant plant in standard form

$$P(s): \begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix}, \tag{13}$$

where $x \in \mathbb{R}^{n_x}$ is the state, $u \in \mathbb{R}^{m_2}$ the control, $w \in \mathbb{R}^{m_1}$ the vector of exogenous inputs, $y \in \mathbb{R}^{p_2}$ the measurements, and $z \in \mathbb{R}^{p_1}$ the controlled or performance vector,

$$P(s) := \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} (sI - A)^{-1} \begin{bmatrix} B_1 & B_2 \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}.$$

Without loss of generality, it is assumed that $D_{22} = 0$. Let $u(s) = K(s)y(s)$ be an output feedback controller for the open-loop plant (13), with

$$K: \begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \begin{bmatrix} x_K \\ y \end{bmatrix},$$

where $x_K \in \mathbb{R}^k$ is the state of $K$. The closed-loop transfer function of the performance channel $w \to z$ is obtained as

$$T_{w\to z}(K, s) = P_{11}(s) + P_{12}(s)K(s)(I - P_{22}(s)K(s))^{-1}P_{21}(s).$$

Our aim is to find an optimal controller $K$ which stabilizes the system in closed-loop such that $\|T_{w\to z}(K)\|_H$ is minimized among all stabilizing $K$. By substituting $u = Ky$ into (13), the state-space representation of the closed-loop performance channel $w \to z$ is

$$T_{w\to z}(K): \begin{bmatrix} \dot{\xi} \\ z \end{bmatrix} = \begin{bmatrix} \mathcal{A}(K) & \mathcal{B}(K) \\ \mathcal{C}(K) & \mathcal{D}(K) \end{bmatrix} \begin{bmatrix} \xi \\ w \end{bmatrix},$$

where $\xi = (x, x_K)$ and

$$\mathcal{A}(K) = \begin{bmatrix} A + B_2 D_K C_2 & B_2 C_K \\ B_K C_2 & A_K \end{bmatrix}, \quad \mathcal{B}(K) = \begin{bmatrix} B_1 + B_2 D_K D_{21} \\ B_K D_{21} \end{bmatrix},$$
$$\mathcal{C}(K) = \begin{bmatrix} C_1 + D_{12} D_K C_2 & D_{12} C_K \end{bmatrix}, \quad \mathcal{D}(K) = D_{11} + D_{12} D_K D_{21}.$$

This problem is now a specific instance of (2), where in agreement with our general theme we try to minimize the memory of a specific channel $w \to z$ within the plant $P$. If we allow structured control laws $K(\mathbf{x})$ in the sense of [1], then we obtain the following optimization program

$$\begin{array}{ll} \text{minimize} & \|T_{w\to z}(K)\|_H \\ \text{subject to} & K \text{ stabilizes (13) internally} \\ & K = K(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n. \end{array} \tag{14}$$

*Example* 1. Typical examples of structured controllers are, for instance, PIDs or observer-based controllers, which in state-space have the form

$$K_{\text{pid}}(\mathbf{x}) = \left[\begin{array}{cc|c} 0 & 0 & r_i \\ 0 & -\tau & r_d \\ \hline 1 & 1 & d_K \end{array}\right], \quad K_{\text{obs}}(\mathbf{x}) = \left[\begin{array}{c|c} A + B_2 K_c + K_f C_2 & -K_f \\ \hline K_c & 0 \end{array}\right].$$

For a PID, the tunable parameters are $\mathbf{x} = (r_i, r_d, d_K, \tau)$, while for observer-based controllers $K_{\text{obs}}(\mathbf{x})$ the vector $\mathbf{x}$ gathers the elements of $K_c, K_f$. Other examples are decentralized, fixed reduced order controllers, and more generally, control architectures combining basic building blocks such as PIDs with filters, feed-forward blocks, and much else (see [1]).

*Remark* 6. The norm in program (14) is the usual Hankel norm (1) if $\mathcal{D}(K) = 0$, which is the case e.g., under standard assumption as in $H_2$-synthesis, where $D_{11} = 0$ and either $D_{21} = 0$ or $D_{12} = 0$ or $K$ strictly proper. In contrast, if $\mathcal{D}(K) \neq 0$, then we should use the extended Hankel norm (10), or likewise, the constraint program (12), to control the direct transmission. It is also possible to neglect the direct transmission term $\mathcal{D}(K)$ and optimize the semi-norm $\|(\mathcal{A}(K), \mathcal{B}(K), \mathcal{C}(K))\|_H$. We then exercise caution by monitoring the term $\sigma_1(\mathcal{D}(K))$ during optimization to check whether a large direct transmission gain $\sigma_1(\mathcal{D}(K))$ is favored. If that is the case, switching to the extended Hankel norm becomes mandatory.

In the sequel of this section, we discuss two particular cases of the Hankel synthesis problem (14).

**6.1. System reduction.** System reduction is the most widely known application of the Hankel norm minimization problem. Given a stable system

$$G : \begin{cases} \dot{x} &= Ax + Bw \\ z &= Cx + Dw \end{cases}$$

of order $n_x$, we wish to find a stable system

$$G_k : \begin{cases} \dot{x} &= A_k x + B_k w \\ z &= C_k x + Dw \end{cases}$$

of reduced order $k < n_x$ with input–output behavior as close as possible to the original system $G$. If the model matching error $e = (G - G_k)w$ is measured in the Hankel norm, then the program

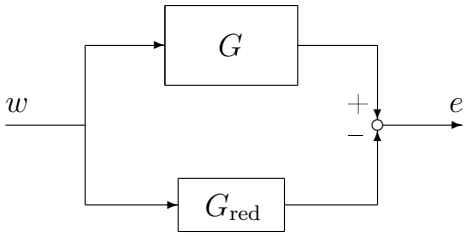
$$\begin{array}{ll} \text{minimize} & \|G - G_k(\mathbf{x})\|_H \\ \text{subject to} & G - G_k(\mathbf{x}) \text{ internally stable} \\ & \mathbf{x} = (A_k, B_k, C_k) \end{array} \tag{15}$$

is a particular case of (14), where we define plant and controller as

$$P : \left[\begin{array}{c|cc} A & B & 0 \\ \hline C & D & -I \\ 0 & I & 0 \end{array}\right], \qquad K : \left[\begin{array}{c|c} A_k & B_k \\ \hline C_k & D \end{array}\right], \tag{16}$$

the tunable parameters $\mathbf{x}$ being the elements of $A_k$, $B_k$ and $C_k$.

(17)

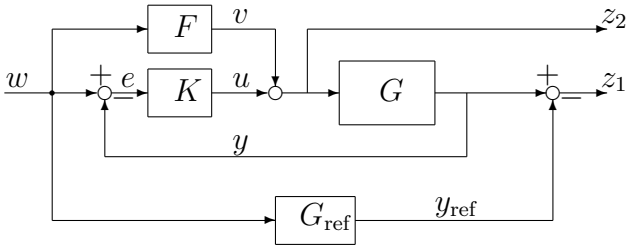



Due to the seminal work of Glover [12], program (15) has an explicit solution based on linear algebra, at least when no additional structural constraints on the matrices $A_k$, $B_k$, $C_k$ are imposed. This allows us to implement a blind testing of Algorithm 1 in Sect. 8, which is applied to (15), considered as a particular case of (14) using (16). The value obtained by Algorithm 1 is then compared to the theoretical value obtained by an explicit Hankel system reduction.

**6.2. Maximizing the memory of a system.** Within the present framework, it is also possible to maximize the memory effects of a system $G$ via feedback if a reference system $G_{\text{ref}}$ with desirable memory properties is used. In other words, while minimizing $\|G(\mathbf{x})\|_H$ leads to a system which is the least biased, we now bias $G(\mathbf{x})$ as much as possible by bringing it as close as possible to $G_{\text{ref}}$, and we achieve this by making $G(\mathbf{x}) - G_{\text{ref}}$ as less biased as possible.

*Example* 2. As a motivating example, we consider a 2-DOF synthesis scheme of the following form

(18)




where the decentralized controller structure was chosen to challenge our method in a typical situation in practice.

Assuming that $G_{\text{ref}}$ has desirable memory features which do not lead to ringing, the idea is to tune the parameters in feed-forward filter $F$ and controller $K$ in such a way that $G$ in closed-loop follows $G_{\text{ref}}$, independently of the input $w$. That is, the undesirable part of the memory of $G$, which contributes to the mismatch $z_1 = y - y_{\text{ref}}$, is reduced by minimizing $\|T_{w\to z_1}(F,K)\|_H$. It may be beneficial to arrange this by adding a constraint $\|z_2\|_2 \leqslant \eta_2$ or $\|z_2\|_\infty \leqslant \eta_\infty$, where $z_2 = u + v$, to avoid exceedingly large controller actions. This problem can be cast as a particular case of program (14) if the following plant and decentralized controller structures are used

$$P: \left[\begin{array}{cc|ccc} A & 0 & 0 & B & B \\ 0 & A_{\text{ref}} & B_{\text{ref}} & 0 & 0 \\ \hline C & -C_{\text{ref}} & -D_{\text{ref}} & D & D \\ 0 & 0 & 0 & I & I \\ -C & 0 & I & -D & -D \\ 0 & 0 & I & 0 & 0 \end{array}\right], \qquad K: \left[\begin{array}{cc|cc} A_F & 0 & B_F & 0 \\ 0 & A_K & 0 & B_K \\ \hline C_F & 0 & D_F & 0 \\ 0 & C_K & 0 & D_K \end{array}\right].$$

Notice that

$$F: \left\{\begin{array}{ll} \dot{x}_F &= A_F x_F + B_F w \\ v &= C_F x_F + D_F w \end{array}\right., \qquad K: \left\{\begin{array}{ll} \dot{x}_K &= A_K x_K + B_K e \\ u &= C_K x_K + D_K e \end{array}\right.$$

can be further structured if we wish. In our experiment, we will use this example with $F$ a reduced-order filter, and $K$ a PID.

## 7. Control of flow in a graph

We consider the flow in a directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{A})$ with interior nodes, sources and sinks, $\mathscr{V} = \mathscr{V}_{\text{stay}} \cup \mathscr{V}_{\text{in}} \cup \mathscr{V}_{\text{out}}$, and not excluding self-arcs. For nodes $i, j \in \mathscr{V}$ connected by an arc $(i,j) \in \mathscr{A}$ the transition probability $i \to j$ quantifies the tendency of flow going from node $i$ towards node $j$. As an example consider for instance a large fairground with separated entrances and exits, with itineraries represented by the graph. By acting on the transition probabilities between nodes connected by

arcs, we expect to guide the crowd in such a way that a steady flow is assured, and a safe evacuation is possible.

Assume that an individual at interior node $j \in \mathscr{V}_{\text{stay}}$ decides with probability $a_{jj'} \geqslant 0$ to proceed to a neighboring node $j' \in \mathscr{V}_{\text{stay}}$, where neighboring means $(j,j') \in \mathscr{A}$, or with probability $a_{jk} \geqslant 0$ to move to a neighboring exit node $k \in \mathscr{V}_{\text{out}}$, where $(j,k) \in \mathscr{A}$. The case $(j,j) \in \mathscr{A}$ of deciding to stay at stand $j \in \mathscr{V}_{\text{stay}}$ is not excluded. Similarly, an individual entering at $i \in \mathscr{V}_{\text{in}}$ proceeds to a neighboring interior node $j \in \mathscr{V}_{\text{stay}}$ with probability $b_{ij} \geqslant 0$, where $(i,j) \in \mathscr{A}$. We suppose for simplicity that there is no direct transmission from entrances to exits. Then,

$$\sum_{j' \in \mathscr{V}_{\text{stay}} : (j,j') \in \mathscr{A}} a_{jj'} + \sum_{k \in \mathscr{V}_{\text{out}} : (j,k) \in \mathscr{A}} a_{jk} = 1, \tag{19}$$

for every $j \in \mathscr{V}_{\text{stay}}$, and

$$\sum_{j \in \mathscr{V}_{\text{stay}} : (i,j) \in \mathscr{A}} b_{ij} = 1 \tag{20}$$

for every $i \in \mathscr{V}_{\text{in}}$. Let $x_j(t)$ denote the number of people present at interior node $j \in \mathscr{V}_{\text{stay}}$ and time $t$, and $w_i(t)$ the number of people entering the fairground through entry $i \in \mathscr{V}_{\text{in}}$ at time $t$. Then, the number of people present at interior node $j \in \mathscr{V}_{\text{stay}}$ and time $t+1$ is

$$x_j(t+1) = \sum_{j' \in \mathscr{V}_{\text{stay}} : (j',j) \in \mathscr{A}} a_{j'j} x_{j'}(t) + \sum_{i \in \mathscr{V}_{\text{in}} : (i,j) \in \mathscr{A}} b_{ij} w_i(t),$$

while the number of people leaving the fairground at time $t$ through exit $k \in \mathscr{V}_{\text{out}}$ is $\sum_{j \in \mathscr{V}_{\text{stay}} : (j,k) \in \mathscr{A}} a_{jk} x_j(t)$. To assess the evacuation pattern, we quantify the total number of people still inside the fairground at time $t$ via the weighted sum

$$z(t) = \sum_{j \in \mathscr{V}_{\text{stay}}} c_j x_j(t),$$

where $c_j > 0$ are fixed weights, and where $c_j = 1$ would correspond to simply counting the number of people inside the fairground. We let $\mathbf{x}$ regroup the parameters $a_{jj'}, a_{jk}, b_{ij}$, so that the discrete linear time-invariant system has the form $G(\mathbf{x}) = (A(\mathbf{x}), B(\mathbf{x}), C)$, where $C$ is the row vector of $c_j$'s.

Let us now consider an evacuation scenario, where at time $T$ the inflow $w(t)$ through the entrance gates is stopped by closing the gates, and the time until the fairground is evacuated is assessed by measuring the evacuation pattern $z(t)$, $t > T$. This corresponds to computing the Hankel norm $\|G(\mathbf{x})\|_H$, which identifies the worst case evacuation scenario. Minimizing $\|z\|_{2,[T,\infty)} / \|w\|_{2,(0,T]}$ may then be understood as enhancing overall safety of the network by orienting the crowd in such a way that the worst case evacuation time is minimized. This leads to the optimization program

$$\begin{array}{ll} \text{minimize} & \|G(\mathbf{x})\|_H \\ \text{subject to} & G(\mathbf{x}) \text{ internally stable} \\ & a_{jj'} \geqslant 0, a_{jk} \geqslant 0, b_{ij} \geqslant 0, (19), (20) \end{array} \tag{21}$$

which is a discrete version of (2) including linear constraints. Notice that these linear constraints are readily added in our algorithmic approach. In an extended model, one might consider measuring the number of people $y$ at some selected nodes $i \in$

$\mathscr{V}_{\text{stay}} \cup \mathscr{V}_{\text{out}}$, and use this to react via feedback $u = Ky$ at the entry gates. This leads to a problem where controller and parts of the plant are optimized simultaneously. Other variants include cases, where some of the probabilities $a_{jj'}$, $b_{ij}$ are imposed and cannot be modified by the designer.

## 8. Proximal bundle algorithm

In this section, we present our main algorithm to solve programs (2) and (12). Let us consider an abstract constrained optimization program of the form

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & h(\mathbf{x}) \leqslant 0 \end{array} \tag{22}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the decision variable, and $f$ and $h$ are locally Lipschitz but potentially nonsmooth and nonconvex functions, representing objective and constraints. To find solutions of the constraint program (22), using an idea inspired by Polak [20, Section 2.2.2], we introduce the progress function

$$F(\mathbf{y}, \mathbf{x}) = \max\{f(\mathbf{y}) - f(\mathbf{x}) - \nu h(\mathbf{x})_+, h(\mathbf{y}) - h(\mathbf{x})_+\},$$

where $h(\mathbf{x})_+ = \max\{h(\mathbf{x}), 0\}$, and $\nu > 0$ is some fixed parameter (with $\nu = 1$ a typical value). One can think of $\mathbf{x}$ as the current iterate, and $\mathbf{y}$ as the next iterate or as a candidate to become the next iterate. We need to collect a few facts about $F$. Note first that $F(\mathbf{x}, \mathbf{x}) = 0$. For the subdifferential, we have the useful

**Lemma 8.1.** *Suppose $f$ and $h$ are lower-$C^1$ functions. Then, the Clarke subdifferential of the progress function $F$ with respect to the first variable is obtained as*

$$\partial_1 F(\mathbf{x}, \mathbf{x}) = \begin{cases} \partial f(\mathbf{x}) & \text{if } h(\mathbf{x}) < 0, \\ \operatorname{conv}\{\partial f(\mathbf{x}) \cup \partial h(\mathbf{x})\} & \text{if } h(\mathbf{x}) = 0, \\ \partial h(\mathbf{x}) & \text{if } h(\mathbf{x}) > 0. \end{cases}$$

*Proof.* Applying the formula for the Clarke subdifferential of a maximum [8, Proposition 2.3.12] we readily get $\partial_1 F(\mathbf{x}, \mathbf{x}) = \partial f(\mathbf{x})$ if $h(\mathbf{x}) < 0$, $\partial_1 F(\mathbf{x}, \mathbf{x}) \subset \operatorname{conv}\{\partial f(\mathbf{x}) \cup \partial h(\mathbf{x})\}$ if $h(\mathbf{x}) = 0$, and $\partial_1 F(\mathbf{x}, \mathbf{x}) = \partial h(\mathbf{x})$ if $h(\mathbf{x}) > 0$. But since $f$ and $g$ are lower-$C^1$, according to [24, Proposition 2.4, Theorem 3.9], they are Clarke regular, so we have equality in the second case $h(\mathbf{x}) = 0$. □

**Lemma 8.2.** *Suppose $\mathbf{x}^*$ is a local minimum of program (22), then it is also a local minimum of $F(\cdot, \mathbf{x}^*)$, and $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)$. Conversely, if $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)$ then $\mathbf{x}^*$ is either a Karush–Kuhn–Tucker point of (22), or a critical point of constraint violation.*

*Proof.* Since $\mathbf{x}^*$ is a local minimum of (22), we have feasibility $h(\mathbf{x}^*) \leqslant 0$, and so $h(\mathbf{x}^*)_+ = 0$, which implies $F(\mathbf{y}, \mathbf{x}^*) = \max\{f(\mathbf{y}) - f(\mathbf{x}^*), h(\mathbf{y})\}$. Now, there exists a neighborhood $U$ of $\mathbf{x}^*$ such that $f(\mathbf{y}) \geqslant f(\mathbf{x}^*)$ for every $\mathbf{y} \in U$ with $h(\mathbf{y}) \leqslant 0$. We argue that $F(\mathbf{y}, \mathbf{x}^*) \geqslant F(\mathbf{x}^*, \mathbf{x}^*)$ for every $\mathbf{y} \in U$. Namely, if $h(\mathbf{y}) > 0$, then $F(\mathbf{y}, \mathbf{x}^*) \geqslant h(\mathbf{y}) > 0 = F(\mathbf{x}^*, \mathbf{x}^*)$. On the other hand, if $h(\mathbf{y}) \leqslant 0$, then $\mathbf{y}$ is feasible, and we have $f(\mathbf{y}) \geqslant f(\mathbf{x}^*)$ by what was said before. But then $F(\mathbf{y}, \mathbf{x}^*) \geqslant f(\mathbf{y}) - f(\mathbf{x}^*) \geqslant 0 = F(\mathbf{x}^*, \mathbf{x}^*)$. This proves $\mathbf{x}^*$ is a local minimum of $F(\cdot, \mathbf{x}^*)$, and so $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)$.

Next, suppose $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)$, then by Lemma 8.1, there exist non-negative constants $\lambda_0^*, \lambda_1^*$ summing up to 1 such that $0 \in \lambda_0^* \partial f(\mathbf{x}^*) + \lambda_1^* \partial h(\mathbf{x}^*)$. If $h(\mathbf{x}^*) > 0$, we have $\partial_1 F(\mathbf{x}^*, \mathbf{x}^*) = \partial h(\mathbf{x}^*)$, and then $0 \in \partial h(\mathbf{x}^*)$, meaning that $\mathbf{x}^*$ is a critical point of $h$. If $h(\mathbf{x}^*) < 0$ then $\partial_1 F(\mathbf{x}^*, \mathbf{x}^*) = \partial f(\mathbf{x}^*)$, so $\lambda_1^* = 0$ and $\mathbf{x}^*$ is a Karush–Kuhn–Tucker point of (22). Assume that $h(\mathbf{x}^*) = 0$ but $\mathbf{x}^*$ fails to meet the Karush–Kuhn–Tucker conditions, we then obtain $\lambda_0^* = 0$ and $0 \in \partial h(\mathbf{x}^*)$. This completes the proof of the lemma. □

The consequence of this argument is that we should seek points $\mathbf{x}^*$ with $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)$. We now present our method for computing solutions of program (22), which is based on this rationale. It generates a sequence $\mathbf{x}^j$ of estimates which converges to a solution $\mathbf{x}^*$ in the sense of subsequences. At the current iterate $\mathbf{x}$, the inner loop of the algorithm constructs first-order working models $\phi_k(\cdot, \mathbf{x})$ and the corresponding second-order working models

$$\Phi_k(\mathbf{y}, \mathbf{x}) = \phi_k(\mathbf{y}, \mathbf{x}) + \frac{1}{2}(\mathbf{y} - \mathbf{x})^\top Q(\mathbf{x})(\mathbf{y} - \mathbf{x}),$$

updated with counter $k$. The $\Phi_k(\cdot, \mathbf{x})$ are approximations of $F(\cdot, \mathbf{x})$ around $\mathbf{x}$, where $Q(\mathbf{x})$ is symmetric, depends only on the current iterate $\mathbf{x}$, and may reflect second-order information of $F$ around $\mathbf{x}$. The first-order working model $\phi_k(\cdot, \mathbf{x})$ has to satisfy $\phi_k(\mathbf{x}, \mathbf{x}) = F(\mathbf{x}, \mathbf{x}) = 0$ and $\partial_1 \phi_k(\mathbf{x}, \mathbf{x}) \subset \partial_1 F(\mathbf{x}, \mathbf{x})$ at all instants $k$. This is guaranteed when $m_e(\cdot, \mathbf{x}) = g(\mathbf{x})^\top(\cdot - \mathbf{x})$ with $g(\mathbf{x}) \in \partial_1 F(\mathbf{x}, \mathbf{x})$ is an affine minorant of $\phi_k(\cdot, \mathbf{x})$ at all times $k$. We refer to $m_e(\cdot, \mathbf{x})$ as the exactness plane at $\mathbf{x}$.

For a given working model, we solve the tangent program

$$\min_{\mathbf{y} \in \mathbb{R}^n} \Phi_k(\mathbf{y}, \mathbf{x}) + \frac{\tau_k}{2}\|\mathbf{y} - \mathbf{x}\|^2,$$

with the so-called proximity control parameter $\tau_k > 0$. We require $Q(\mathbf{x}) + \tau_k I \succ 0$, which assures that the tangent program is strictly convex and has a unique solution $\mathbf{y}^k$, called the trial step. According to standard terminology, $\mathbf{y}^k$ is called a serious step if it is accepted as the new iterate $\mathbf{y}^k = \mathbf{x}^+$, and a null step otherwise. Suppose $\mathbf{y}^k$ is a null step, then we will have to make sure that the next working model $\phi_{k+1}(\cdot, \mathbf{x})$ improves over $\phi_k(\cdot, \mathbf{x})$. This is achieved by adding cutting and aggregate planes. Let us first look at aggregation. The optimality condition for the tangent program implies

$$g_k^* := (Q(\mathbf{x}) + \tau_k I)(\mathbf{x} - \mathbf{y}^k) \in \partial_1 \phi_k(\mathbf{y}^k, \mathbf{x}).$$

We call $m_k^*(\cdot, \mathbf{x}) = \phi_k(\mathbf{y}^k, \mathbf{x}) + g_k^{*\top}(\cdot - \mathbf{y}^k) = a_k^* + g_k^{*\top}(\cdot - \mathbf{x})$ with $a_k^* = \phi_k(\mathbf{y}^k, \mathbf{x}) + g_k^{*\top}(\mathbf{x} - \mathbf{y}^k)$ the aggregate plane. By assuring that $m_k^*(\cdot, \mathbf{x})$ is an affine minorant of $\phi_{k+1}(\cdot, \mathbf{x})$, we have $\phi_{k+1}(\mathbf{y}^k, \mathbf{x}) \geqslant m_k^*(\mathbf{y}^k, \mathbf{x}) = \phi_k(\mathbf{y}^k, \mathbf{x})$.

A central element in bundle methods is the cutting plane whose role is to cut away the unsuccessful trial step $\mathbf{y}^k$. For each subgradient $g_k \in \partial_1 F(\mathbf{y}^k, \mathbf{x})$, the affine function $t_k(\cdot) = F(\mathbf{y}^k, \mathbf{x}) + g_k^\top(\cdot - \mathbf{y}^k)$ is a tangent to $F(\cdot, \mathbf{x})$ at $\mathbf{y}^k$. Without convexity, we cannot use $t_k(\cdot)$ directly as a cutting plane. Instead, we use a technique first analyzed in [14], which shifts the tangent down. Fixing a parameter $c > 0$, we define the cutting plane as

$$m_k(\cdot, \mathbf{x}) = t_k(\cdot) - s = a_k + g_k^\top(\cdot - \mathbf{x}), \tag{23}$$

where $a_k = \min\{t_k(\mathbf{x}), -c\|\mathbf{y}^k - \mathbf{x}\|^2\}$, and where $s = [t_k(\mathbf{x}) + c\|\mathbf{y}^k - \mathbf{x}\|^2]_+$ is the downshift. The detailed statement is described as Algorithm 1, while a flowchart of the algorithm is shown in Fig. 1. For more details we refer to [17, Section 3], [16, Section 4] for unconstrained optimization case, and [2, Section 5], [11, Section 3] for the constrained case.

**Algorithm 1.** Proximal bundle algorithm with downshifted tangents

**Parameters:** $0 < \gamma < \widetilde{\gamma} < \Gamma < 1, 0 < \delta \ll 1, 0 < q < T \leqslant \infty$.

▷ **Step 1** (`Initialize outer loop`)**.** Choose initial feasible guess $\mathbf{x}^1$, fix memory control parameter $\tau_1^\sharp$, and put outer loop counter $j = 1$.

◇ **Step 2** (`Stopping test`)**.** At outer loop counter $j$, stop if $0 \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$. Otherwise, take a symmetric matrix $Q_j$ respecting $-qI \preceq Q_j \preceq qI$, and goto inner loop.

▷ **Step 3** (`Initialize inner loop`)**.** Put inner loop counter $k = 1$ and initialize control parameter $\tau_1 = \max\{\tau_j^\sharp, -\lambda_{\min}(Q_j) + \delta\}$, where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of a symmetric matrix. Choose initial working model $\phi_1(\cdot, \mathbf{x}^j) = g(\mathbf{x}^j)^\top(\cdot - \mathbf{x}^j)$ with $g(\mathbf{x}^j) \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$.

▷ **Step 4** (`Tangent program`)**.** At inner loop counter $k$, let $\Phi_k(\mathbf{y}, \mathbf{x}^j) = \phi_k(\mathbf{y}, \mathbf{x}^j) + \frac{1}{2}(\mathbf{y} - \mathbf{x}^j)^\top Q_j(\mathbf{y} - \mathbf{x}^j)$ and find solution $\mathbf{y}^k$ (`trial step`) of the tangent program

$$\min_{\mathbf{y}\in\mathbb{R}^n} \Phi_k(\mathbf{y}, \mathbf{x}^j) + \frac{\tau_k}{2}\|\mathbf{y} - \mathbf{x}^j\|^2.$$

◇ **Step 5** (`Acceptance test`)**.** Compute the quotient

$$\rho_k = \frac{F(\mathbf{y}^k, \mathbf{x}^j)}{\Phi_k(\mathbf{y}^k, \mathbf{x}^j)}.$$

If $\rho_k \geqslant \gamma$ (`serious step`), put $\mathbf{x}^{j+1} = \mathbf{y}^k$ and update memory element $\tau_{j+1}^\sharp$ as $\tau_k$ if $\rho_k < \Gamma$, and $\frac{1}{2}\tau_k$ otherwise. Reset $\tau_{j+1}^\sharp = T$ if $\tau_{j+1}^\sharp > T$, increase outer loop counter $j$ and loop back to step 2. If $\rho_k < \gamma$ (`null step`), continue inner loop with step 6.

▷ **Step 6** (`Update working model`)**.** Generate a cutting plane $m_k(\cdot, \mathbf{x}^j)$ at null step $\mathbf{y}^k$ and counter $k$ using downshifted tangents. Compute aggregate plane $m_k^*(\cdot, \mathbf{x}^j)$ at $\mathbf{y}^k$, and then build new working model $\phi_{k+1}(\cdot, \mathbf{x}^j)$ by adding the new cutting plane, keeping the exactness plane and using aggregation to avoid overflow.

◇ **Step 7** (`Update control parameter`)**.** Compute secondary control parameter

$$\widetilde{\rho}_k = \frac{M_k(\mathbf{y}^k, \mathbf{x}^j)}{\Phi_k(\mathbf{y}^k, \mathbf{x}^j)},$$

with $M_k(\mathbf{y}, \mathbf{x}^j) = m_k(\mathbf{y}, \mathbf{x}^j) + \frac{1}{2}(\mathbf{y} - \mathbf{x}^j)^\top Q_j(\mathbf{y} - \mathbf{x}^j)$. If $\widetilde{\rho}_k < \widetilde{\gamma}$ then keep $\tau_{k+1} = \tau_k$, otherwise step up $\tau_{k+1} = 2\tau_k$. Increase inner loop counter $k$ and loop back to step 4.

Next, we establish the following result on the convergence of Algorithm 1.

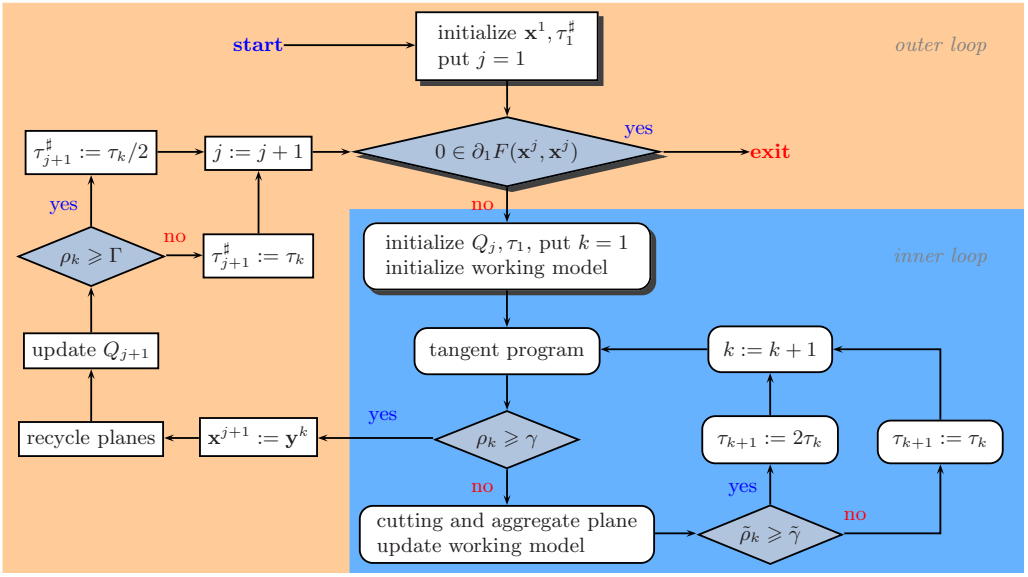



FIGURE 1. Flowchart of proximal bundle algorithm. Inner loop is shown in the *lower right box*

**Theorem 8.3.** *Suppose that $f$ and $h$ in (22) are lower-$C^1$ functions, and let $\{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) \leqslant f(\mathbf{x}^1)\}$ be bounded. Then, every accumulation point $\mathbf{x}^*$ of the sequence of serious iterates $\mathbf{x}^j$ generated by Algorithm 1 satisfies $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)$. In other words, $\mathbf{x}^*$ is either a critical point of constraint violation, or a Karush–Kuhn–Tucker point of (22).*

*Proof.* We will adapt the proof of Theorem 6.6 and Corollary 6.7 in [17] to our needs. For that let us recall a notion from [17, Definitions 2.1 and 6.1], which we apply here to the progress function $F$. We call $\phi : \mathbb{R}^n \times \mathcal{S} \to \mathbb{R}$ a strict first-order model of $F$ on the set $\mathcal{S} \subset \mathbb{R}^n$ if for every $\mathbf{x} \in \mathcal{S}$ the function $\phi(\cdot, \mathbf{x})$ is convex and the following axioms hold:

($M_1$) $\phi(\mathbf{x}, \mathbf{x}) = F(\mathbf{x}, \mathbf{x}) = 0$ and $\partial_1 \phi(\mathbf{x}, \mathbf{x}) \subset \partial_1 F(\mathbf{x}, \mathbf{x})$.

($\widehat{M}_2$) If $\mathbf{y}_j \to \mathbf{x}$ and $\mathbf{x}_j \to \mathbf{x}$ then there exists $\varepsilon_j \to 0^+$ such that $F(\mathbf{y}_j, \mathbf{x}_j) - \phi(\mathbf{y}_j, \mathbf{x}_j) \leqslant \varepsilon_j \|\mathbf{y}_j - \mathbf{x}_j\|$.

($M_3$) $\phi$ is jointly upper semicontinuous on $\mathbb{R}^n \times \mathcal{S}$, i.e., if $(\mathbf{y}_j, \mathbf{x}_j) \to (\mathbf{y}, \mathbf{x})$ then $\limsup_{j\to\infty} \phi(\mathbf{y}_j, \mathbf{x}_j) \leqslant \phi(\mathbf{y}, \mathbf{x})$.

Representing the cutting plane in (23) as $m_{\mathbf{y}^+}(\cdot, \mathbf{x}) = a + g^\top(\cdot - \mathbf{x})$ with $g \in \partial_1 F(\mathbf{y}^+, \mathbf{x})$ and $a = \min\{t_{\mathbf{y}^+}(\mathbf{x}), -c\|\mathbf{y}^+ - \mathbf{x}\|^2\}$, $t_{\mathbf{y}^+}(\cdot) = F(\mathbf{y}^+, \mathbf{x}) + g^\top(\cdot - \mathbf{y}^+)$, we define

$$\phi(\mathbf{y}, \mathbf{x}) = \sup\{m_{\mathbf{y}^+}(\mathbf{y}, \mathbf{x}) : \mathbf{y}^+ \in B(\mathbf{x}, r)\},$$

where $B(\mathbf{x}, r)$ is a fixed ball large enough to contain all possible trial steps, and where the supremum is over all possible cases of $m_{\mathbf{y}^+}(\cdot, \mathbf{x})$. It then follows that $\phi$ is a strict model of $F$ in the sense of the above definition. This can be shown as in [16, Lemmas 7–9]. Axiom ($\widehat{M}_2$) relies on the fact that $F(\cdot, \mathbf{x})$ is lower-$C^1$ by the assumptions on $f$ and $h$. Furthermore, the construction of $\phi$ and $\phi_k$ also guarantees that the working models $\phi_k$ are lower approximations of $\phi$ satisfying

$\phi_k(\mathbf{x},\mathbf{x}) = \phi(\mathbf{x},\mathbf{x}) = F(\mathbf{x},\mathbf{x}) = 0$, $\partial_1\phi_k(\mathbf{x},\mathbf{x}) \subset \partial_1\phi(\mathbf{x},\mathbf{x})$ and $\phi_k(\cdot,\mathbf{x}) \leqslant \phi(\cdot,\mathbf{x})$. The difference with [17] is that here the cutting planes $m_k(\cdot,\mathbf{x})$ are not directly tangents of $\phi$, but we shall argue that the essential link between $\phi_k$ and $\phi$ rests the same.

The proof now follows essentially [17, Theorem 6.6, Corollary 6.7], which assures that every accumulation point $\mathbf{x}^*$ of the iterates $\mathbf{x}^j$ satisfies $0 \in \partial_1 F(\mathbf{x}^*,\mathbf{x}^*)$. Note that $f(\mathbf{x}^j)$ and $f(\mathbf{y}^k)$ used in [17] have to be replaced by $F(\mathbf{x}^j,\mathbf{x}^j) = 0$ and $F(\mathbf{y}^k,\mathbf{x})$. The fact that $\Phi(\mathbf{y}^{k+1})$ in the definition of $\widetilde{\rho}_k$ in [17] is changed to $M_k(\mathbf{y}^k,\mathbf{x})$ can be treated using the property that if $\mathbf{y}_j \to \mathbf{x}$ and $\mathbf{x}_j \to \mathbf{x}$ then there exists $\varepsilon_j \to 0^+$ such that $F(\mathbf{y}_j,\mathbf{x}_j) - m_{\mathbf{y}_j}(\mathbf{y}_j,\mathbf{x}_j) \leqslant \varepsilon_j\|\mathbf{y}_j - \mathbf{x}_j\|$, as follows from [16, Lemma 8], using again crucially that $F(\cdot,\mathbf{x})$ is lower-$C^1$. The equality $\phi_{k+1}(\mathbf{y}^{k+1},\mathbf{x}) = \phi(\mathbf{y}^{k+1},\mathbf{x})$ used in the proof of [17, Lemma 4.2] is now replaced by $\phi_{k+1}(\mathbf{y}^k,\mathbf{x}) \geqslant m_k(\mathbf{y}^k,\mathbf{x})$. Finally, Lemma 8.2 completes the last statement of the theorem. □

## 9. A smooth relaxation of the Hankel norm

Here, we introduce a smooth relaxation of the Hankel norm based on a result of Nesterov in [15]. He provides a fine analysis of the convex bundle method in situations where the objective $f(\mathbf{x})$ has the specific structure of a max-function, including the case of a convex maximum eigenvalue function. These findings indicate that for a given precision, such programs may be solved with lower algorithmic complexity using smooth relaxations. While these results are *a priori* limited to the convex case, it may be interesting to apply Nesterov's idea as a heuristic in the nonconvex situation. This leads to the following

**Proposition 9.1.** *Let $\mathcal{Z}$ be a symmetric matrix of order $m$ depending smoothly on a parameter $\mathbf{x} \in \mathbb{R}^n$ with eigenvalues $\lambda_1(\mathcal{Z}) \geqslant \cdots \geqslant \lambda_m(\mathcal{Z})$. Then, for a tolerance parameter $\mu > 0$, the function*

$$f_\mu(\mathbf{x}) = \mu \ln\left(\sum_{i=1}^{m} e^{\lambda_i(\mathcal{Z}(\mathbf{x}))/\mu}\right) \tag{24}$$

*is a uniform smooth approximation of the nonsmooth function $f(\mathbf{x}) = \lambda_1(\mathcal{Z}(\mathbf{x}))$ in the sense that $f_\mu(\mathbf{x})$ converges uniformly to $f(\mathbf{x})$ as $\mu \to 0$.*

*Proof.* Following [15, Section 4], $f_\mu$ is smooth in $\mathcal{Z}$ and

$$\nabla f_\mu(\mathcal{Z}) = \left(\sum_{i=1}^{m} e^{\lambda_i(\mathcal{Z})/\mu}\right)^{-1} \sum_{i=1}^{m} e^{\lambda_i(\mathcal{Z})/\mu} q_i q_i^\top,$$

where $q_i$ is the $i$th column of the orthogonal matrix $Q(\mathcal{Z})$ from the eigendecomposition of the symmetric matrix $\mathcal{Z} = Q(\mathcal{Z})D(\mathcal{Z})Q(\mathcal{Z})^\top$. This implies that $f_\mu$ is smooth at $\mathbf{x}$ with the gradient given by

$$\nabla f_\mu(\mathbf{x}) = \begin{bmatrix}\operatorname{Tr}(\nabla f_\mu(\mathcal{Z}(\mathbf{x}))^\top \mathcal{Z}_1(\mathbf{x})) & \dots & \operatorname{Tr}(\nabla f_\mu(\mathcal{Z}(\mathbf{x}))^\top \mathcal{Z}_m(\mathbf{x}))\end{bmatrix}^\top.$$

On the other hand, we have the estimate

$$f(\mathbf{x}) \leqslant f_\mu(\mathbf{x}) \leqslant f(\mathbf{x}) + \mu \ln m,$$

which says that $f_\mu(\mathbf{x})$ is a uniform approximation of the function $f(\mathbf{x})$. □

Now, we can try to solve problem (2) and (12) on replacing the function $f(\mathbf{x}) = \lambda_1(\mathcal{Z}(\mathbf{x}))$ by its smooth approximation $f_\mu(\mathbf{x})$ in (24). Due to the estimate in the above proof, to find an $\varepsilon$-solution $\bar{\mathbf{x}}$ of problem (2) and (12), we have to find an $\frac{\varepsilon}{2}$-solution of the smooth problem

$$\min\{f_\mu(\mathbf{x}) : h(\mathbf{x}) \leqslant 0\} \tag{25}$$

with $\mu = \frac{\varepsilon}{2 \ln m}$. Here, we use a local solution of (25) to initialize the nonsmooth Algorithm 1. The smooth problem (25) can be solved using standard NLP software.

## 10. Numerical experiments

In this section, we apply our approach to a variety of problems. Let us start by commenting on practical ways to implement the stopping test $0 \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$ in step 2 of the algorithm. In practice, this is delegated to the inner loop. If the inner loop at $\mathbf{x}^j$ finds a new feasible serious iterate $\mathbf{x}^{j+1}$ satisfying

$$\frac{|f(\mathbf{x}^{j+1}) - f(\mathbf{x}^j)|}{1 + |f(\mathbf{x}^j)|} < \mathrm{tol}_1, \tag{26}$$

then we accept $\mathbf{x}^{j+1}$ as optimal. This corresponds to stopping the algorithm in step 2 of the $(j+1)$st outer loop. In our experiments, we have used $\mathrm{tol}_1 = 10^{-8}$.

On the other hand, if the inner loop has difficulties finding a serious step and provides three unsuccessful trial steps satisfying

$$\frac{\|\mathbf{x}^j - \mathbf{y}^k\|}{1 + \|\mathbf{x}^j\|} < \mathrm{tol}_2, \tag{27}$$

then we interpret this in the sense that $\mathbf{x}^j$ is already optimal. This corresponds to stopping the algorithm in step 2 of the $j$th outer loop. Here, we have used $\mathrm{tol}_2 = 10^{-7}$. Theoretically, both tests are based on the observation that $0 \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$ if and only if $\mathbf{y}^k = \mathbf{x}^j$ is solution of the tangent program in the trial step generation (see [11] for theoretical results).

In general, our stopping strategy is similar to recommendations in smooth optimization, see e.g., [10, Chapter 7], where the goal is to obtain scale independent choices of the tolerances $\mathrm{tol}_1$ and $\mathrm{tol}_2$. Nonetheless, one has to accept that a nonsmooth algorithm converges very slowly at the final stages, which makes stopping a delicate task.

Before applying Algorithm 1 to solve examples of (2), note that internal stability is not a constraint in the usual sense of mathematical programming since the set $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : G(\mathbf{x}) \text{ internally stable}\}$ is open. The stability of the system can be formulated as a constraint $\alpha(A(\mathbf{x})) \leqslant -\varepsilon$ using the spectral abscissa $\alpha(A) = \max\{\mathrm{Re}(\lambda) : \lambda \text{ eigenvalue of } A\}$ in the continuous time case, and as $\rho(A(\mathbf{x})) \leqslant 1-\varepsilon$ using the spectral radius $\rho(A) = \max\{|\lambda| : \lambda \text{ eigenvalue of } A\}$ in the discrete time case, for $\varepsilon > 0$ some small threshold. Theoretical properties of the spectral abscissa and the spectral radius have been studied in [7]. In general, before optimization can start, one has, indeed, to find a stabilizing $\mathbf{x}$. Using the method in [4], this can be achieved by an initial phase where $\alpha(A(\mathbf{x}))$ is minimized until an iterate $\mathbf{x}^1$ with $\alpha(A(\mathbf{x}^1)) \leqslant -\varepsilon$ is found.

**10.1. Hankel feedback synthesis.** We introduce an application of program (14) to a classical 1-DOF control system design, using an example from [5, Section 2.4]. The open-loop system $G$, exogenous input $w$ and regulated output $z$, are given by

$$G = \frac{10-s}{s^2(10+s)}, \quad w = \begin{bmatrix} d \\ n_y \\ r \end{bmatrix}, \quad z = \begin{bmatrix} y_p \\ u \end{bmatrix}.$$

The corresponding plant is

$$P : \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right],$$

where

$$\begin{aligned} A &= \begin{bmatrix} -10 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} & B_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & B_2 &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ C_1 &= \begin{bmatrix} 0 & -1 & 10 \\ 0 & 0 & 0 \end{bmatrix} & & & D_{12} &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ C_2 &= \begin{bmatrix} 0 & 1 & -10 \end{bmatrix} & D_{21} &= \begin{bmatrix} 0 & -1 & 1 \end{bmatrix}. \end{aligned}$$

Inspired by a manually tuned controller

$$K_b = \frac{219.6s^2 + 1973.95s + 724.5}{s^3 + 19.15s^2 + 105.83s + 965.95},$$

proposed in [5, Section 2.4], we compute the optimal Hankel controller $K_H$ with the same proposed structure and compare it to $K_b$ and also to the optimal $H_\infty$-controller $K_\infty$ of that same structure

$$K(\mathbf{x}) = \frac{as^2+bs+c}{s^3+ms^2+ns+p} = \left[\begin{array}{ccc|c} -m & -n & -p & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline a & b & c & 0 \end{array}\right],$$

where $\mathbf{x} = [m, n, p, a, b, c]^\top$ regroups the unknown tunable parameters. Using the Matlab function `hinfstruct` based on [1], we obtain

$$K_\infty = \frac{7941.9s^2 + 13028.4s + 3611.6}{s^3 + 3206.2s^2 + 12528.3s + 11078.3}.$$

The interest in this example is also to show that parametrizations $\mathbf{x}$ may arise naturally in the frequency domain. Note also that the closed-loop has no direct transmission term since $D_{11} = 0$ and $K$ is strictly proper. To compute $K_H$, we solve (14) with the standard Hankel norm (1) and start Algorithm 1 at an initial stabilizing controller

$$\mathbf{x}^1 = [2.1460, 12.7448, 7.4208, 1.2271, 1.8013, 0.3517]^\top$$

with $f(\mathbf{x}^1) = 455.2874$, using the stability constraint $h(\mathbf{x}) = \alpha(A(\mathbf{x})) + \varepsilon \leqslant 0$ with a typical value $\varepsilon = 10^{-8}$. The stopping tests were (26) and (27). The algorithm came to a halt due to (26) and returned the optimal solution

$$\mathbf{x}^* = [77.0614, 255.2324, 74.6195, 188.0709, 133.9333, 22.2401]^\top$$

with $f(\mathbf{x}^*) = 10.8419$, meaning $\|T_{w\to z}(P, K_H)\|_H = 3.2927$ and

$$K_H := K(\mathbf{x}^*) = \frac{77.0614s^2 + 255.2324s + 74.6195}{s^3 + 188.0709s^2 + 133.9333s + 22.2401}.$$

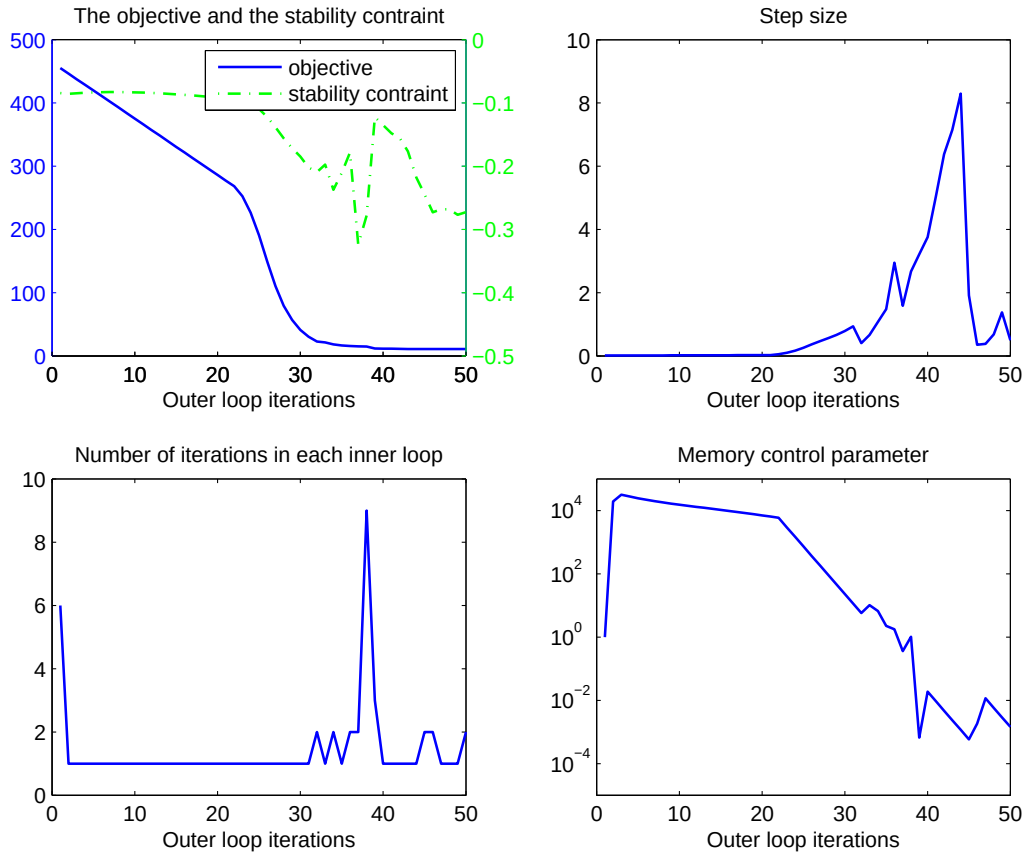



FIGURE 2. Hankel feedback synthesis. Bearing of the algorithm. *Top left* shows $j \mapsto f(\mathbf{x}^j)$ and $j \mapsto \alpha(A(\mathbf{x}^j)) + 10^{-8}$. *Top right* shows $j \mapsto \|\mathbf{x}^j - \mathbf{x}^{j+1}\|$. *Lower left* shows $j \mapsto k_j$, *lower right* shows $j \mapsto \tau_j^\sharp$, the evolution of the memory control parameter at serious steps

The algorithm needed 50 serious iterates with 2.3 s CPU to reach the local minimum $K_H$. Bearing of the algorithm is shown in Fig. 2. The improvement of $\|T_{w\to z}(P, K_H)\|_H = 3.2927$ over $\|T_{w\to z}(P, K_\infty)\|_H = 3.3265$ is moderate, while the improvement over $\|T_{w\to z}(P, K_b)\|_H = 109.52$ is plain. Step responses and magnitude plots of the controllers $K_b$, $K_H$ and $K_\infty$ are shown in Fig. 3. Posterior testing displays ringing effects caused by various input signals $w$, including $w =$ unit step, white noise and sinc, shown in Fig. 4. As can be seen e.g., in Fig. 4, middle image, for a truncated white noise function $w_T = w\chi_{[0,T]}$, with $T = 3$, comparison of the responses $z_H = T_{w\to z}(K_H)w_T$ and $z_\infty = T_{w\to z}(K_\infty)w_T$, while confirming optimality $\|z_\infty\|_\infty = 0.5413 < \|z_H\|_\infty = 0.5498$, reveals that the bulk of energy in $z_\infty$ has a wider spread over time, and $\|z_H\|_{2,[T,\infty)} = 1.1626 < \|z_\infty\|_{2,[T,\infty)} = 1.1878$ corroborating that the memory effects in $K_H$ are reduced by the use of program (14).

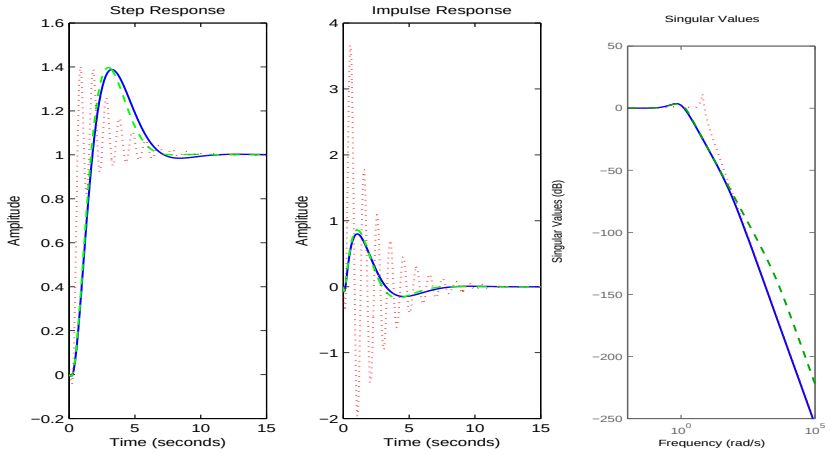



FIGURE 3. Hankel feedback synthesis. Step responses (*left*), impulse responses (*middle*), magnitude plot (*right*) for controllers $K_b$ (*dotted*), $K_\infty$ (*dashed*), and $K_H$ (*solid*)

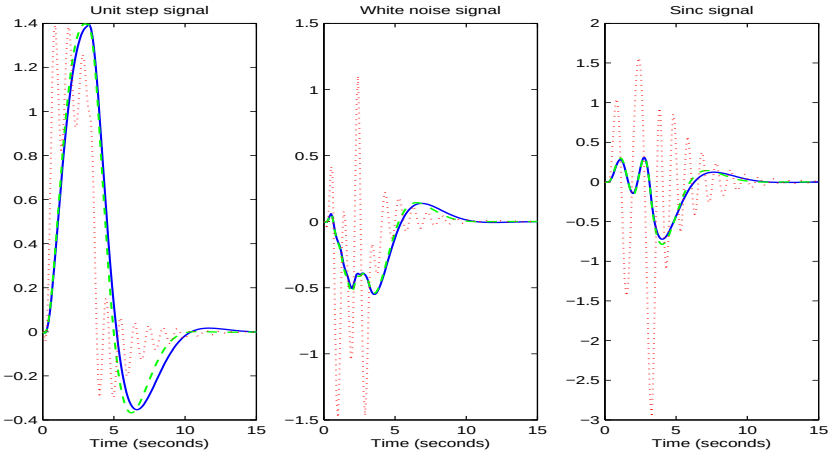



FIGURE 4. Hankel feedback synthesis. Ringing for controllers $K_b$ (*dotted*), $K_\infty$ (*dashed*), and $K_H$ (*solid*). Inputs: Unit step signal (*left*), white noise signal (*middle*), sinc signal (*right*)

**10.2. Hankel system reduction.** In this section, we solve program (15) with the usual Hankel norm, where our tests use the 15th order Rolls-Royce Spey gas turbine engine model described in [23, Chapter 11], with data available for download on I. Postlethwaites's homepage as `aero0.mat`. The goal of this study is to use the theoretical values to perform a blind testing of our algorithm. For $k = 1, 2, \ldots, 14$, using Algorithm 1, we computed Hankel reduced-order systems $G_k$ of order $k$, and compared the achieved objective $f(\mathbf{x}^*) = \|G - G_k(\mathbf{x}^*)\|_H$ of (15) with the theoretically known optimal Hankel norm approximation errors $\|G - G_k\|_H = \sigma_{k+1}$, the $(k+1)$st Hankel singular value of $G$. As can be seen in columns 2 and 3 of Table 1, this error is within the limits of numerical precision.

In each run, the algorithm was started from a random initial guess, and no information as to the specific structure of problem (15) was provided. On average, the algorithm needed about 103 serious steps to reach the optimal objective function

TABLE 1. Hankel system reduction. Comparison of optimal values $\|G - G_k(\mathbf{x}^*)\|_H$ with theoretical values $\sigma_{k+1}$

| $k$ | $\sigma_{k+1}$ | $\|G - G_{\text{red}}\|_H$ | No of iterations | Time |
|---|---|---|---|---|
| 1 | 4.046418 | 4.046418 | 26 | 3.5 |
| 2 | 2.754623 | 2.754624 | 71 | 21.0 |
| 3 | 1.763527 | 1.763529 | 124 | 47.3 |
| 4 | 1.296531 | 1.299542 | 151 | 101.5 |
| 5 | 0.629640 | 0.629640 | 88 | 118.0 |
| 6 | 0.166886 | 0.166887 | 183 | 197.3 |
| 7 | 0.093407 | 0.093408 | 93 | 185.8 |
| 8 | 0.022193 | 0.022201 | 76 | 132.4 |
| 9 | 0.015669 | 0.015675 | 162 | 203.7 |
| 10 | 0.013621 | 0.013624 | 175 | 191.3 |
| 11 | 0.003997 | 0.003997 | 140 | 380.0 |
| 12 | 0.001179 | 0.001179 | 57 | 488.4 |
| 13 | 0.000324 | 0.000324 | 24 | 224.2 |
| 14 | 0.000033 | 0.000033 | 68 | 372.5 |

value within a tolerance of $< 10^{-10}$. See Table 1 for number of iterations and running times in seconds.

*Remark* 7. The results show no clear relation between running times and the order of the reduced system, as one might have expected. This is due to the fact that local optimization techniques depend very sensibly on the initial guess, which in this comparison was chosen randomly.

*Remark* 8. In [9], we have used the same example to give a comparison between Hankel system reduction and $H_\infty$-system reduction, which is compared to the $H_\infty$-bound (see [12]).

**10.3. Maximizing the memory of a system.** We use here an illustrative example for (18), where $G$ and $G_{\text{ref}}$ are defined as

$$G(s) = \frac{1}{s-1}, \quad G_{\text{ref}} = \frac{11.11}{s^2 + 6s + 11.11}.$$

The filter $F$ is chosen of order 2,

$$F(s) = \frac{as^2 + bs + c}{s^2 + ds + e} = \left[\begin{array}{cc|c} -d & -e & 1 \\ 1 & 0 & 0 \\ \hline b - ad & c - ae & a \end{array}\right],$$

which leads to 5 tunable parameters, whereas $K$ is a PID

$$K(s) = k_p + \frac{k_i}{s} + \frac{k_d s}{T_f s + 1} = \left[\begin{array}{cc|c} 0 & 0 & k_i \\ 0 & -\frac{1}{T_f} & -\frac{k_d}{T_f^2} \\ \hline 1 & 1 & k_p + \frac{k_d}{T_f} \end{array}\right],$$

adding another 4 unknowns. We have added a low-pass filter $W_1(s) = \frac{0.25s+0.6}{s+0.006}$ to the output $z_1$ to asses the tracking error $y - y_{\text{ref}}$ in low-frequency, and a high-pass filter $W_2(s) = \frac{s}{s+0.001}$ on the control output $z_2$ to reduce high-frequency components of the control signal $u + v$.

Due to the choice of the performance channel $w \to z = (W_1 z_1, W_2 z_2)$, the closed-loop has a non-vanishing direct transmission term. We therefore solve problem (14) for the setup (18) using the extended Hankel program (2) with (10), and also using the constraint program (12). Running Algorithm 1 from the same starting point, these two methods give Hankel controllers $(F_{eH}, K_{eH})$ and $(F_{cH}, K_{cH})$ with

$$F_{eH}(s) = \frac{-3.4778s^2-13.9996s-0.0546}{s^2+1.9202s+0.0001},\ K_{eH}(s) = 6.3078 + \frac{3.6689}{s} - \frac{1.0924}{0.4739s+1},$$
$$F_{cH}(s) = \frac{-3.6552s^2-13.6987s-0.0522}{s^2+1.9588s+0.0001},\ K_{cH}(s) = 6.1959 + \frac{3.8435}{s} - \frac{0.7121}{0.3644s+1},$$

where we used the constraint $\sigma_1(D) \leqslant \eta$ with $\eta = 1$. For comparison, we also synthesized the usual Hankel norm controller, where the direct transmission is ignored, and the $H_\infty$-controller, both with the same architecture:

$$F_H(s) = \frac{-2.2376s^2-1.9738s-2.4161}{s^2+0.9054s+0.9836},\ K_H(s) = 2.4482 + \frac{0.7883}{s} + \frac{0.8023}{0.7817s+1},$$
$$F_\infty(s) = \frac{-9.9366s^2-1.5077s-0.0349}{s^2+0.9969s+0.0273},\ K_\infty(s) = 11.5131 + \frac{0.2673}{s} - \frac{0.5507}{1.0117s+1}.$$

Figure 5 compares step responses $y$ and step reference responses $y_{\text{ref}}$ for these controllers. The evolution of the optimization method for the three Hankel controllers can be traced in Fig. 6. The achieved Hankel norms are

$$\|T_{w\to z}(F_{eH}, K_{eH})\|_H = 0.8767 < \|T_{w\to z}(F_{cH}, K_{cH})\|_H = 0.8862$$
$$< \|T_{w\to z}(F_H, K_H)\|_H = 1.0160 < \|T_{w\to z}(F_\infty, K_\infty)\|_H = 1.0277.$$

This example is again interesting in so far as the parametrization of $F$ and $K$ arises naturally in the frequency domain.

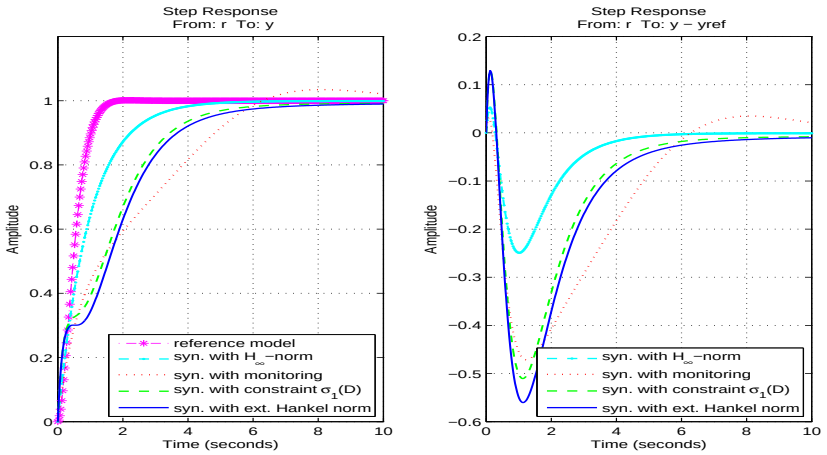



FIGURE 5. Maximizing memory. Comparison between step responses $y$ and $y_{\text{ref}}$ for $H_\infty$-controller and Hankel controllers computed by programs (2) with monitoring (*dotted*), (12) (*dashed*) and (2) with (10) (*solid*)

**10.4. Control of flow in a graph.** Here, we give an application of program (21). Let $\mathscr{V}_{\text{stay}} = \{1, 2, \ldots, n_x\}$, $\mathscr{V}_{\text{in}} = \{1, 2, \ldots, m\}$, $\mathscr{V}_{\text{out}} = \{1, 2, \ldots, p\}$. Let $\mathbf{x}$ regroup the unknown tunable parameters $a_{jj'}, b_{ij}$ and set $A(\mathbf{x}) = [a_{jj'}]_{n_x \times n_x}^\top$, $B(\mathbf{x}) = [b_{ij}]_{m \times n_x}^\top$, $C = [c_1, \ldots, c_{n_x}]$, where $a_{jj'} = 0$ if $(j, j') \notin \mathscr{A}$, $b_{ij} = 0$ if $(i, j) \notin \mathscr{A}$. We

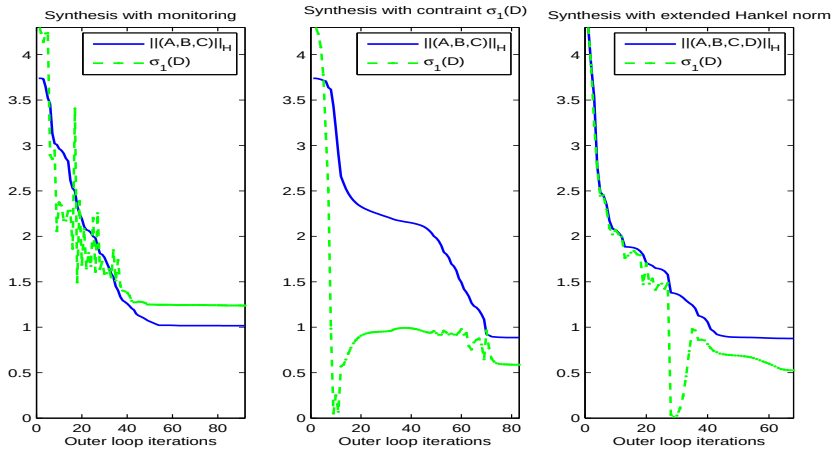


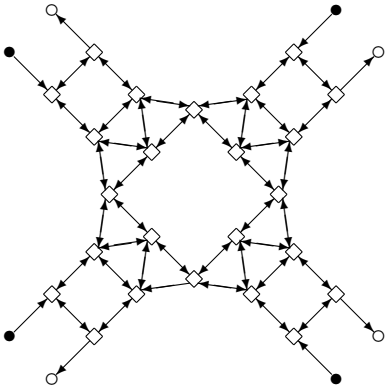

FIGURE 6. Maximizing memory. Comparison between standard Hankel program (2) with monitoring (*left*), constraint program (12) (*middle*), and extended Hankel program (2) with (10) (*right*). While (2) with (10) and (12) give comparable results, minimization of $\|(A, B, C)\|_H$ alone (*left*) gives a large direct transmission

have a discrete linear time-invariant system

$$G(\mathbf{x}) : \left\{ \begin{array}{ll} x(t+1) & = A(\mathbf{x})x(t) + B(\mathbf{x})w(t) \\ z(t) & = Cx(t). \end{array} \right.$$

Remark that the linear constraint conditions in (21) can be transferred to the form $A_{\text{eq}}\mathbf{x} = b_{\text{eq}}, \mathbf{x} \geqslant 0$, which are added in each trial step generation of Algorithm 1.

We now take the following graph $\mathscr{G} = (\mathscr{V}, \mathscr{A})$ with $n_x = 24, m = 4$ and $p = 4$.



Let $z(t)$ be the total number of people on the fairground, which corresponds to the weights $c_1 = \cdots = c_{n_x} = 1$. We start Algorithm 1 at the uniform distribution $\mathbf{x}^1$, where $f(\mathbf{x}^1) = 714.8634$, and $\|G(\mathbf{x}^1)\|_H = 26.7369$. After 2469 serious iterates with 8768 s CPU, our algorithm returns the optimal $\mathbf{x}^*$ with $f(\mathbf{x}^*) = 8.6056$, meaning $\|G(\mathbf{x}^*)\|_H = 2.9335$. For comparison, with the Matlab function `fmincon` started at $\mathbf{x}^1$, we obtain $\mathbf{x}^\dagger$ with $f(\mathbf{x}^\dagger) = 12.5994 > f(\mathbf{x}^*) = 8.6056$. However, if we take $\mathbf{x}^\dagger$ as initial for Algorithm 1, the result is $f(\mathbf{x}^*) = 8.6056$, meaning $\|G(\mathbf{x}^*)\|_H = 2.9335$, which is achieved very fast (29 serious iterates, 87 s CPU).

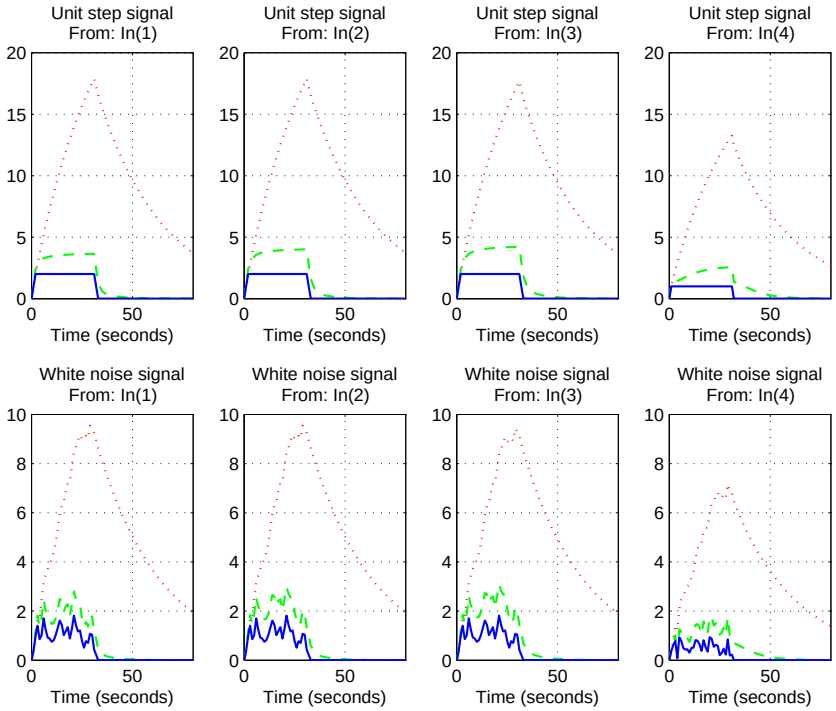
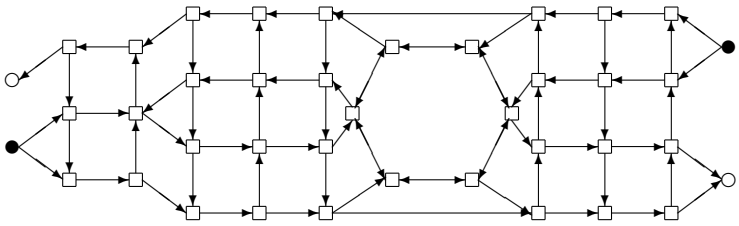



FIGURE 7. Ringing effects of three systems $G(\mathbf{x}^1)$ (*dotted*), $G(\mathbf{x}^\dagger)$ (*dashed*) and $G(\mathbf{x}^*)$ (*solid*) for the first graph. Input: Unit step signal (*top*) and white noise signal (*bottom*)



We next consider an example using the second graph with $n_x = 36, m = 2$ and $p = 2$. Let $z(t)$ quantify the number of people on the fairground, where the 6 central nodes are counted twice. In this example, we will directly compare our nonsmooth method to the heuristic in Sect. 9. Optimization starts again at the uniform distribution $\mathbf{x}^1$. Minimizing smooth function $f_\mu(\mathbf{x})$ in (24) with initial $\mathbf{x}^1$ leads to $\mathbf{x}^\dagger$, where $f(\mathbf{x}^\dagger) = 21.7291, \|G(\mathbf{x}^\dagger)\|_H = 4.6614$, while $f(\mathbf{x}^1) = 578.6875, \|G(\mathbf{x}^1)\|_H = 24.0559$. We now use $\mathbf{x}^\dagger$ to initialize the nonsmooth Algorithm 1. After 44 serious steps with 168 s CPU, our algorithm returns the optimal $\mathbf{x}^*$ with $f(\mathbf{x}^*) = 14.8353$, meaning $\|G(\mathbf{x}^*)\|_H = 3.8517$.

For the two displayed graphs, Figs. 7 and 8 compare ringing effects in unit step and white noise responses truncated at $T = 30$ for the three systems $G(\mathbf{x}^1), G(\mathbf{x}^\dagger)$ and $G(\mathbf{x}^*)$. We can see that ringing for $G(\mathbf{x}^\dagger)$ and $G(\mathbf{x}^*)$ is substantially reduced.

Tables 2 and 3 show a simulated study, where we compare the effects of the transition probability distributions $\mathbf{x}^1, \mathbf{x}^\dagger, \mathbf{x}^*$ by recording the evacuation of people from the fairground. We simulate crowd entering through the gates $1, \ldots, 4$ for different scenarios $w$. We then close the entrance gates at time $T = 15$, when in the

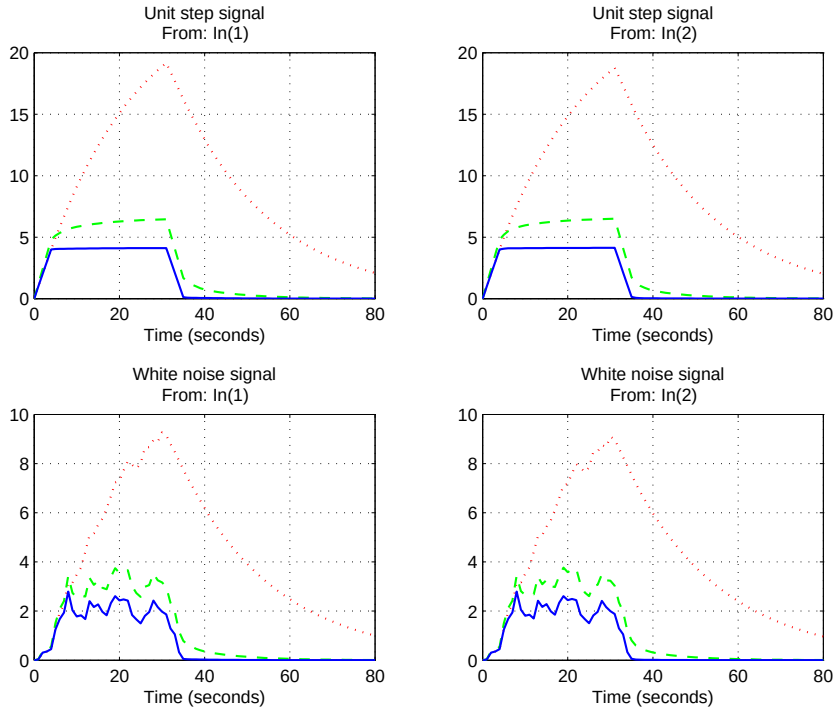



FIGURE 8. Ringing effects of three systems $G(\mathbf{x}^1)$ (*dotted*), $G(\mathbf{x}^\dagger)$ (*dashed*) and $G(\mathbf{x}^*)$ (*solid*) for the second graph. Input: Unit step signal (*top*) and white noise signal (*bottom*)

TABLE 2. First graph, three distributions $\mathbf{x}^1$, $\mathbf{x}^\dagger$, $\mathbf{x}^*$. Times when 90% of crowd in fairground has been evacuated

| Input signal | People | $z^1(T)$ | $G(\mathbf{x}^1)$ | $z^\dagger(T)$ | $G(\mathbf{x}^\dagger)$ | $z^*(T)$ | $G(\mathbf{x}^*)$ |
|---|---|---|---|---|---|---|---|
| | Entering | Remain | Evac. time | Remain | Evac. time | Remain | Evac. time |
| $[w_1; w_2; w_3; 0]$ | 6994 | 4680 | 78 | 1478 | 18 | 1141 | 17 |
| $[w_1; w_2; 0; w_3]$ | 6994 | 4375 | 75 | 1293 | 18 | 941 | 17 |
| $[w_1; 0; w_2; w_3]$ | 6994 | 4367 | 75 | 1306 | 18 | 941 | 17 |
| $[0; w_1; w_2; w_3]$ | 6994 | 4367 | 75 | 1374 | 18 | 941 | 17 |

Entry gates are closed at $T = 15$

TABLE 3. Second graph, three distributions. Times when 90% of crowd in the fairground has been evacuated

| Input signal | People | $z^1(T)$ | $G(\mathbf{x}^1)$ | $z^\dagger(T)$ | $G(\mathbf{x}^\dagger)$ | $z^*(T)$ | $G(\mathbf{x}^*)$ |
|---|---|---|---|---|---|---|---|
| | Entering | Remain | Evac. time | Remain | Evac. time | Remain | Evac. time |
| $[w_1; w_2]$ | 4994 | 3794 | 63 | 1530 | 20 | 1216 | 19 |
| $[w_1; w_3]$ | 5200 | 3901 | 63 | 1546 | 20 | 1227 | 19 |
| $[w_2; w_3]$ | 3794 | 2704 | 63 | 1034 | 20 | 804 | 20 |

Entry gates are closed at $T = 15$

first study 6994 people have entered the ground, and record the time which passes until 90% of the crowd has been evacuated. In our tests $w_1$ is a step signal, $w_2$ is

a sine wave, and $w_3$ is a square wave. A similar approach is chosen in the second graph.

Column $z^1(T)$ gives the number of people still present on the fairground at time $T$ when distribution $\mathbf{x}^1$ is used, and column $G(\mathbf{x}^1)$ gives the time which then elapses until this crowd is reduced below 10% of the total number 6994. Columns 5–8 are analogous. As compared to $\mathbf{x}^1$, the optimal strategy $\mathbf{x}^*$ reduces the evacuation time to close to 1/5 in the first graph, and to close to 1/3 in the second graph.

## 11. Conclusion

We have proposed a new methodology to reduce unwanted ringing effects in a tunable linear time-invariant system. The problem was addressed by minimizing the Hankel norm of the system, a problem which leads to an eigenvalue optimization program for the associated Hankel operator. A proximal bundle algorithm was presented to solve a variety of test problems successfully, and a smooth heuristic, based on work of Nesterov [15], was added and used to initialize the algorithm with a favorable initial seed.

## Acknowledgements

The authors acknowledge helpful discussions with Dr. Armin Rainer (University of Vienna).

## References

1. P. Apkarian and D. Noll, *Nonsmooth $H_\infty$ synthesis*, IEEE Trans. Automat. Control **51** (2006), no. 1, 71–86.
2. P. Apkarian, D. Noll, and A. Rondepierre, *Mixed $H_2/H_\infty$ control via nonsmooth optimization*, SIAM J. Control Optim. **47** (2008), no. 3, 1516–1546.
3. R. Bellman, *Kronecker products and the second method of Lyapunov*, Math Nachr **20** (1959), 17–19.
4. V. Bompart, P. Apkarian, and D. Noll, *Non-smooth techniques for stabilizing linear systems*, Proc. American Control Conf. (New York), July 2007, pp. 1245–1250.
5. S. Boyd and C. Barratt, *Linear controller design: Limits of performance*, Prentice Hall, New York, 1991.
6. M. D. Bronshtein, *Smoothness of roots of polynomials depending on parameters*, Sibirsk. Mat. Zh. **20** (1979), no. 3, 493–501, 690, English transl. in Siberian Math. J. 20 (1980), 347-352.
7. J. V. Burke and M. L. Overton, *Differential properties of the spectral abscissa and the spectral radius for analytic matrix-valued mappings*, Nonlinear Anal. **23** (1994), no. 4, 467–488.
8. F. H. Clarke, *Optimization and nonsmooth analysis*, Canad. Math. Soc. Ser. Monogr. Adv. Texts, John Wiley & Sons, Inc., New York, 1983.
9. M. N. Dao and D. Noll, *Minimizing the memory of a system*, Proc. Asian Control Conf. (Istanbul), June 2013.
10. J. E. Dennis, Jr. and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Prentice Hall, New Jersey, 1983.
11. M. Gabarrou, D. Alazard, and D. Noll, *Design of a flight control architecture using a non-convex bundle method*, Math. Control Signals Syst. **25** (2013), no. 2, 257–290.
12. K. Glover, *All optimal Hankel-norm approximations of linear multivariable systems and their $L^\infty$-error bounds*, Internat. J. Control **39** (1984), no. 6, 1115–1193.
13. O. L. Mangasarian, *Nonlinear programming*, McGraw-Hill Book Co., New York-London-Sydney, 1969.

14. R. Mifflin, *A modification and extension of Lemarechal's algorithm for nonsmooth minimization*, Nondifferential and Variational Techniques in Optimization (D. C. Sorensen and R. J.-B. Wets, eds.), Math. Programming Stud., vol. 17, North-Holland Publishing Co., Amsterdam, 1982, pp. 77–90.
15. Y. Nesterov, *Smoothing technique and its applications in semidefinite optimization*, Math. Program., Ser. A **110** (2007), no. 2, 245–259.
16. D. Noll, *Cutting plane oracles to minimize non-smooth non-convex functions*, Set-Valued Var. Anal. **18** (2010), no. 3-4, 531–568.
17. D. Noll, O. Prot, and A. Rondepierre, *A proximity control algorithm to minimize nonsmooth and nonconvex functions*, Pac. J. Optim. **4** (2008), no. 3, 571–604.
18. M. L. Overton, *Large-scale optimization of eigenvalues*, SIAM J. Optim. **2** (1992), no. 1, 88–120.
19. A. Parusiński and A. Rainer, *A new proof of Bronshtein's theorem*, (2014), arXiv:1309.2150v2.
20. E. Polak, *Optimization: Algorithms and consistent approximations*, Appl. Math. Sci., vol. 124, Springer-Verlag, New York, 1997.
21. A. Rainer, *Smooth roots of hyperbolic polynomials with definable coefficients*, Israel J. Math. **184** (2011), 157–182.
22. R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, Springer-Verlag, Berlin, 1998.
23. S. Skogestad and I. Postlethwaite, *Multivariable feedback control: Analysis and design*, John Wiley & Sons, Chichester, 2005.
24. J. E. Spingarn, *Submonotone subdifferentials of Lipschitz functions*, Trans. Amer. Math. Soc. **264** (1981), no. 1, 77–89.
25. L. van den Dries, *Tame topology and o-minimal structures*, London Math. Soc. Lecture Note Ser., vol. 248, Cambridge University Press, Cambridge, 1998.
26. K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control*, Prentice Hall, New Jersey, 1996.

# III

## Simultaneous plant and controller optimization based on nonsmooth techniques *

Minh Ngoc Dao and Dominikus Noll

**Abstract.** We present an approach to simultaneous design optimization of a plant and its controller. This is based on a bundling technique for solving nonsmooth optimization problems under nonlinear and linear constraints. In the absence of convexity, a substitute for the convex cutting plane mechanism is proposed. The method is illustrated on a problem of steady flow in a graph and in robust feedback control design of a mass-spring-damper system.

**Keywords.** Robust control · Hankel norm · system with tunable parameters · nonlinear optimization · steady flow.

### 1. Introduction

In modern control system, desirable closed-loop characteristics include stability, speed, accuracy, and robustness and depend on both structural and control specifications. Traditionally, structural design with its drive elements precedes and is disconnected from controller synthesis, which may result in a sub-optimal system. In contrast, optimizing plant structure and controller simultaneously may lead to a truly optimal solution. We therefore propose design methods which allow to optimize various elements such as system structure, actuators, sensors, and the controller simultaneously.

Here we focus on simultaneous optimization of certain plant and controller parameters to achieve the best performance for a closed-loop system with constraints. This leads to a complex nonlinear optimization problem involving nonsmooth and nonconvex objectives and constraints. Suitable optimization methods are discussed to address such problems.

*Paper published in Lecture Notes in Engineering and Computer Science: Proc. World Congress Eng. Comp. Sci. (WCECS), vol. II, San Francisco, 2013, pp. 855–861.

Consider a stable LTI state-space control system

$$G : \begin{cases} \delta x = Ax + Bu \\ y = Cx + Du \end{cases}$$

where $\delta x$ represents $\dot{x}(t)$ for continuous-time systems and $x(t+1)$ for discrete-time systems, and where $x \in \mathbb{R}^{n_x}$ is the state vector, $u \in \mathbb{R}^m$ the control input vector, and $y \in \mathbb{R}^p$ the output vector. Our interest is the case in which system $G$ is placed in a control system containing actuators, sensors and a feedback controller $K$, and matrices $A, B, C, D$ and controller $K$ depend smoothly on a design parameter $\mathbf{x}$ varying in $\mathbb{R}^n$ or in some constrained subset of $\mathbb{R}^n$. Denoting by $T_{w\to z}(\mathbf{x})$ the closed-loop performance channel $w \to z$, this brings to the optimization program

$$\begin{array}{ll} \text{minimize} & \|T_{w\to z}(\mathbf{x})\| \\ \text{subject to} & \mathbf{x} \in \mathbb{R}^n, \\ & K = K(\mathbf{x}) \text{ assures closed-loop stability} \end{array} \tag{1}$$

Here standard choices of $\|\cdot\|$ include the $H_\infty$-norm $\|\cdot\|_\infty$, the $H_2$-norm $\|\cdot\|_2$, or the Hankel norm $\|\cdot\|_H$ which is discussed in more detail in the sections 3 and 6. Solving (1) leads to nonsmooth optimization problems.

## 2. A proximity control algorithm

Bundle methods are currently among the most effective approachs to solve nonsmooth optimization problems. In these methods, subgradients from past iterations are accumulated in a bundle, and a trial step is obtained by a quadratic tangent program based on information stored in the bundle. In the absence of convexity, tangent planes can no longer be used as cutting planes, and a substitute has to be found. A sophisticated management of the proximity control mechanism is also required to obtain a satisfactory convergence theory. We will show in which way these elements can be assembled into a successful algorithm.

For the purpose of solving the problem (1), we present here a nonsmooth algorithm for general constrained optimization programs of the form

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & c(\mathbf{x}) \leqslant 0 \\ & A\mathbf{x} \leqslant b \end{array} \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the decision variable, and $f$ and $c$ are potentially nonsmooth and nonconvex, and where the linear constraints are gathered in $A\mathbf{x} \leqslant b$ and handled directly.

Expanding on an idea in [15, Section 2.2.2], we use a progress function at the current iterate $\mathbf{x}$,

$$F(\cdot, \mathbf{x}) = \max\{f(\cdot) - f(\mathbf{x}) - \nu c(\mathbf{x})_+, c(\cdot) - c(\mathbf{x})_+\},$$

where $c(\mathbf{x})_+ = \max\{c(\mathbf{x}), 0\}$, and $\nu > 0$ is a fixed parameter. It is easy to see that $F(\mathbf{x}, \mathbf{x}) = 0$, where either the left branch $f(\cdot) - f(\mathbf{x}) - \nu c(\mathbf{x})_+$ or the right branch $c(\cdot) - c(\mathbf{x})_+$ in the expression of $F(\cdot, \mathbf{x})$ is active at $\mathbf{x}$, i.e., attains the maximum, depending on whether $\mathbf{x}$ is feasible for the non-linear constraint or not.

Setting $P = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leqslant b\}$, it follows from [16, Theorem 6.46] that the normal cone to $P$ at $\mathbf{x}$ is given by

$$N_P(\mathbf{x}) = \{A^\top \eta : \eta \geqslant 0, \eta^\top (A\mathbf{x} - b) = 0\}.$$

We remark therefore that if $\mathbf{x}^*$ is a local minimum of program (2), it is also a local minimum of $F(\cdot, \mathbf{x}^*)$ on $P$, and then $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*) + A^\top \eta^*$ for some multiplier $\eta^* \geqslant 0$ with $\eta^{*\top}(A\mathbf{x}^* - b) = 0$. The symbol $\partial_1$ here stands for the Clarke subdifferential with respect to the first variable. Indeed, if $\mathbf{x}^*$ is a local minimum of (2) then $\mathbf{x}^* \in P, c(\mathbf{x}^*) \leqslant 0$, and so for $\mathbf{y}$ in a neighborhood of $\mathbf{x}^*$ in $P$ we have $F(\mathbf{y}, \mathbf{x}^*) = \max\{f(\mathbf{y}) - f(\mathbf{x}^*), c(\mathbf{y})\} \geqslant f(\mathbf{y}) - f(\mathbf{x}^*) \geqslant 0 = F(\mathbf{x}^*, \mathbf{x}^*)$ if $\mathbf{y}$ is feasible, and $F(\mathbf{y}, \mathbf{x}^*) \geqslant c(\mathbf{y}) > 0$ otherwise. This implies that $\mathbf{x}^*$ is a local minimum of $F(\cdot, \mathbf{x}^*)$ on $P$, and therefore $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*) + N_P(\mathbf{x}^*)$. We now present the following algorithm for computing solutions of program (2).

Convergence theory of Algorithm 1 is discussed in [7, 10] and based on these results, we can prove the following theorem.

**Theorem 2.1.** *Suppose $f$ and $c$ in program (2) are lower-$C^1$ functions such that the following conditions hold:*

(a) *$f$ is weakly coercive on constraint set $\Omega = \{\mathbf{x} \in \mathbb{R}^n : c(\mathbf{x}) \leqslant 0, A\mathbf{x} \leqslant b\}$, i.e., if $\mathbf{x}^j \in \Omega$, $\|\mathbf{x}^j\| \to \infty$, then $f(\mathbf{x}^j)$ is not monotonically decreasing.*

(b) *$c$ is weakly coercive on $P$, i.e., if $\mathbf{x}^j \in P$, $\|\mathbf{x}^j\| \to \infty$, then $c(\mathbf{x}^j)$ is not monotonically decreasing.*

*Then the sequence of serious iterates $\mathbf{x}^j \in P$ generated by Algorithm 1 is bounded, and every accumulation point $\mathbf{x}^*$ of the $\mathbf{x}^j$ satisfies $\mathbf{x}^* \in P$ and $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*) + A^\top \eta^*$ for some multiplier $\eta^* \geqslant 0$ with $\eta^{*\top}(A\mathbf{x}^* - b) = 0$.* □

An immediate consequence of Theorem 2.1 is the following

**Corollary 2.2.** *Under the hypotheses of the theorem, every accumulation point of the sequence of serious iterates generated by Algorithm 1 is either a critical point of constraint violation, or a Karush-Kuhn-Tucker point of program (2).*

*Proof.* Suppose $\mathbf{x}^*$ is an accumulation point of the sequence of serious iterates generated by Algorithm 1. According to Theorem 2.1 we have $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*) + N_P(\mathbf{x}^*)$. By using [4, Proposition 9] (see also [5, Proposition 2.3.12]), there exist constants $\lambda_0, \lambda_1$ such that

$$0 \in \lambda_0 \partial f(\mathbf{x}^*) + \lambda_1 \partial c(\mathbf{x}^*) + N_P(\mathbf{x}^*),$$
$$\lambda_0 \geqslant 0, \lambda_1 \geqslant 0, \ \lambda_0 + \lambda_1 = 1.$$

If $c(\mathbf{x}^*) > 0$ then $\partial_1 F(\mathbf{x}^*, \mathbf{x}^*) = \partial c(\mathbf{x}^*)$, and therefore $0 \in \partial c(\mathbf{x}^*) + N_P(\mathbf{x}^*)$, which means that $\mathbf{x}^*$ is a critical point of constraint violation. In the case of $c(\mathbf{x}^*) \leqslant 0$, if $\mathbf{x}^*$ fails to be a Karush-Kuhn-Tucker point of (2), then $\lambda_0$ must equal 0, and so $0 \in \partial c(\mathbf{x}^*) + N_P(\mathbf{x}^*)$. We obtain that $\mathbf{x}^*$ is either a critical point of constraint violation, or a Karush-Kuhn-Tucker point of program (2). □

In the absence of convexity, proving convergence to a single Karush-Kuhn-Tucker point is generally out of reach, but the following result gives nonetheless a satisfactory answer for stopping of the algorithm.

**Algorithm 1.** Proximity control with downshift

**Parameters:** $0 < \gamma < \widetilde{\gamma} < 1, 0 < \gamma < \Gamma < 1, 0 < q < \infty$, $0 < c < \infty$, $q < T < \infty$.

▷ **Step 1** (`Initialize outer loop`). Choose initial iterate $\mathbf{x}^1$ with $A\mathbf{x}^1 \leqslant b$ and matrix $Q_1 = Q_1^\top$ with $-qI \preceq Q_1 \preceq qI$. Initialize memory element $\tau_1^\sharp$ such that $Q_1 + \tau_1^\sharp I \succ 0$. Put $j = 1$.

◇ **Step 2** (`Stopping test`). At outer loop counter $j$, stop the algorithm if $0 \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j) + A^\top \eta^j$, for a multiplier $\eta^j \geqslant 0$ with $\eta^{j\top}(A\mathbf{x}^j - b) = 0$. Otherwise, goto inner loop.

▷ **Step 3** (`Initialize inner loop`). Put inner loop counter $k = 1$ and initialize $\tau_1 = \tau_j^\sharp$. Build initial working model

$$F_1(\cdot, \mathbf{x}^j) = g_{0j}^\top(\cdot - \mathbf{x}^j) + \tfrac{1}{2}(\cdot - \mathbf{x}^j)^\top Q_j(\cdot - \mathbf{x}^j),$$

where $g_{0j} \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$.

▷ **Step 4** (`Trial step generation`). At inner loop counter $k$ find solution $\mathbf{y}^k$ of the tangent program

$$\begin{array}{ll} \text{minimize} & F_k(\mathbf{y}, \mathbf{x}^j) + \frac{\tau_k}{2}\|\mathbf{y} - \mathbf{x}^j\|^2 \\ \text{subject to} & A\mathbf{y} \leqslant b, \mathbf{y} \in \mathbb{R}^n. \end{array}$$

◇ **Step 5** (`Acceptance test`). If

$$\rho_k = \frac{F(\mathbf{y}^k, \mathbf{x}^j)}{F_k(\mathbf{y}^k, \mathbf{x}^j)} \geqslant \gamma,$$

put $\mathbf{x}^{j+1} = \mathbf{y}^k$ (serious step), quit inner loop and goto step 8. Otherwise (null step), continue inner loop with step 6.

▷ **Step 6** (`Update working model`). Generate a cutting plane $m_k(\cdot, \mathbf{x}^j) = a_k + g_k^\top(\cdot - \mathbf{x}^j)$ at null step $\mathbf{y}^k$ and counter $k$ using downshifted tangents. Compute aggregate plane $m_k^*(\cdot, \mathbf{x}^j) = a_k^* + g_k^{*\top}(\cdot - \mathbf{x}^j)$ at $\mathbf{y}^k$, and then build new working model $F_{k+1}(\cdot, \mathbf{x}^j)$.

◇ **Step 7** (`Update proximity control parameter`). Compute secondary control parameter

$$\widetilde{\rho}_k = \frac{F_{k+1}(\mathbf{y}^k, \mathbf{x}^j)}{F_k(\mathbf{y}^k, \mathbf{x}^j)}$$

and put

$$\tau_{k+1} = \begin{cases} \tau_k & \text{if } \widetilde{\rho}_k < \widetilde{\gamma}, \\ 2\tau_k & \text{if } \widetilde{\rho}_k \geqslant \widetilde{\gamma}. \end{cases}$$

Increase inner loop counter $k$ and loop back to step 4.

◇ **Step 8** (`Update` $Q_j$ `and memory element`). Update $Q_j \to Q_{j+1}$ respecting $Q_{j+1} = Q_{j+1}^\top$ and $-qI \preceq Q_{j+1} \preceq qI$. Then store new memory element

$$\tau_{j+1}^\sharp = \begin{cases} \tau_k & \text{if } \rho_k < \Gamma, \\ \frac{1}{2}\tau_k & \text{if } \rho_k \geqslant \Gamma. \end{cases}$$

Increase $\tau_{j+1}^\sharp$ if necessary to ensure $Q_{j+1} + \tau_{j+1}^\sharp I \succ 0$. If $\tau_{j+1}^\sharp > T$ then re-set $\tau_{j+1}^\sharp = T$. Increase outer loop counter $j$ and loop back to step 2.

**Corollary 2.3.** *Under the hypotheses of the theorem, for every $\varepsilon > 0$ there exists an index $j_0(\varepsilon) \in \mathbb{N}$ such that for every $j \geqslant j_0(\varepsilon)$, $\mathbf{x}^j$ is within $\varepsilon$-distance of the set $L$ of critical points $\mathbf{x}^*$ in the sense of the theorem.*

*Proof.* By the fact that our algorithm assures always $\mathbf{x}^j - \mathbf{x}^{j+1} \to 0$ and Ostrowski's theorem [13, Theorem 26.1], the set of limit point $L$ of the sequence $\mathbf{x}^j$ is either singleton or a compact continuum. Our construction then assures convergence of $\mathbf{x}^j$ to the limiting set $L$ in the sense of the Hausdorff distance. See [11] for the details. □

## 3. Hankel norm

Given a stable LTI system

$$G : \begin{cases} \dot{x} = Ax + Bw \\ z = Cx \end{cases}$$

with state $x \in \mathbb{R}^{n_x}$, input $w \in \mathbb{R}^m$, and output $z \in \mathbb{R}^p$, if we think of $w(t)$ as an excitation at the input which acts over the time period $0 \leqslant t \leqslant T$, then the ring of the system after the excitation has stopped at time $T$ is $z(t)$ for $t > T$. If signals are measured in the energy norm, this leads to that the Hankel norm of the system $G$ is defined as

$$\|G\|_H = \sup_{T>0} \left\{ \left( \int_T^\infty z(t)^\top z(t)\, dt \right)^{1/2} : \int_0^T w(t)^\top w(t)\, dt \leqslant 1, w(t) = 0 \text{ for } t > T, z = Gw \right\}.$$

For the discrete-time case, the Hankel norm of $G$ is given by

$$\|G\|_H = \sup_{T>0} \left\{ \left( \sum_{t=T}^\infty z(t)^\top z(t) \right)^{1/2} : \sum_{t=0}^T w(t)^\top w(t) \leqslant 1, w(t) = 0 \text{ for } t > T, z = Gw \right\}.$$

The Hankel norm can be understood as measuring the tendency of a system to store energy, which is later retrieved to produce undesired noise effects known as system ring. Minimizing the Hankel norm therefore reduces the ringing in the system. It is worth to note that in both continuous-time and discrete-time cases we have the following

**Proposition 3.1.** *If $X$ and $Y$ are the controllability and observability Gramians of the stable system $G$, then*

$$\|G\|_H = \sqrt{\lambda_1(XY)},$$

*where $\lambda_1$ denotes the maximum eigenvalue of a symmetric or Hermitian matrix.*

*Proof.* See [6] and also [8, Section 2.3]. □

## 4. Steady flow in a graph

Here we consider the problem of steady flow in a directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{A})$ with sources, sinks, and interior nodes, $\mathscr{V} = \mathscr{V}_{\text{stay}} \cup \mathscr{V}_{\text{in}} \cup \mathscr{V}_{\text{out}}$, and not excluding self-arcs. For nodes $i, j \in \mathscr{V}$ connected by an arc $(i, j) \in \mathscr{A}$ the transition probability $i \to j$ quantifies the tendency of flow going from node $i$ towards node $j$. As an example we may for instance consider a large fairground with separated entrances and exits, where itineraries between stands, entrances and exits are represented by the graph. By acting on the transition probabilities between nodes connected by arcs, we expect to guide the crowd in such a way that a steady flow is assured, and a safe evacuation is possible in case of an emergency.

Assume that an individual at interior node $j \in \mathscr{V}_{\text{stay}}$ decides with probability $a_{jj'} \geqslant 0$ to proceed to a neighboring node $j' \in \mathscr{V}_{\text{stay}}$, where neighboring means $(j, j') \in \mathscr{A}$, or with probability $a_{jk} \geqslant 0$ to proceed to a neighboring exit node $k \in \mathscr{V}_{\text{out}}$, where $(j, k) \in \mathscr{A}$. The case $(j, j) \in \mathscr{A}$ of deciding to stay at stand $j \in \mathscr{V}_{\text{stay}}$ is not excluded. Similarly, an individual entering at $i \in \mathscr{V}_{\text{in}}$ proceeds to a neighboring interior node $j \in \mathscr{V}_{\text{stay}}$ with probability $b_{ij} \geqslant 0$, where $(i, j) \in \mathscr{A}$. We assume for simplicity that there is no direct transmission from entrances to exits. Then

$$\sum_{j' \in \mathscr{V}_{\text{stay}} : (j,j') \in \mathscr{A}} a_{jj'} + \sum_{k \in \mathscr{V}_{\text{out}} : (j,k) \in \mathscr{A}} a_{jk} = 1, \tag{3}$$

for every $j \in \mathscr{V}_{\text{stay}}$, and

$$\sum_{j \in \mathscr{V}_{\text{stay}} : (i,j) \in \mathscr{A}} b_{ij} = 1 \tag{4}$$

for every $i \in \mathscr{V}_{\text{in}}$. Let $x_j(t)$ denote the number of people present at interior node $j \in \mathscr{V}_{\text{stay}}$ and time $t$, and $w_i(t)$ the number of people entering the fairground through entry $i \in \mathscr{V}_{\text{in}}$ at time $t$. Then the number of people present at interior node $j \in \mathscr{V}_{\text{stay}}$ and time $t + 1$ is

$$x_j(t+1) = \sum_{j' \in \mathscr{V}_{\text{stay}} : (j',j) \in \mathscr{A}} a_{j'j} x_{j'}(t) + \sum_{i \in \mathscr{V}_{\text{in}} : (i,j) \in \mathscr{A}} b_{ij} w_i(t).$$

We quantify the total number of individuals still inside the fairground via the weighted sum

$$z(t) = \sum_{j \in \mathscr{V}_{\text{stay}}} c_j x_j(t)$$

at time $t$, where $c_j > 0$ are fixed weights. We assess the performance of the network by using the $L_2$-norm to quantify input and output flows $w, z$. This attributes a high cost to a strong concentration of people at a single spot. Take $\mathbf{x}$ to regroup the parameters $a_{jj'}, a_{jk}, b_{ij}$, the discrete LTI system above has the form $G(\mathbf{x}) = (A(\mathbf{x}), B(\mathbf{x}), C, 0)$, where $C$ is the row vector of $c_j$'s. The Hankel norm $\|G(\mathbf{x})\|_H$ may then be interpreted as computing the worst-case of all scenarios where the inflow $w$ is stopped at some time $T$, and the outflow is measured via the pattern $z(t)$, $t \geqslant T$, with which the fairground is emptied. Minimizing $\|z\|_{2,[T,\infty)} / \|w\|_{2,(0,T]}$ may then be understood as enhancing overall safety of the network. It leads to the

optimization program

$$\begin{array}{lll} & \text{minimize} & \|G(\mathbf{x})\|_H \\ (5) & \text{subject to} & G(\mathbf{x}) \text{ internally stable} \\ & & a_{jj'} \geqslant 0, a_{jk} \geqslant 0, b_{ij} \geqslant 0, (3), (4) \end{array}$$

which is a version of (1).

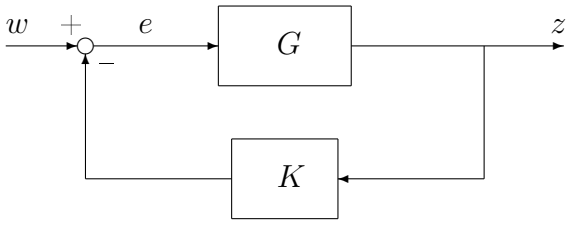



FIGURE 1. Control architecture in the fairground.

In an extended model one might consider measuring the number of people at some selected nodes $j \in \mathscr{V}_{\text{stay}} \cup \mathscr{V}_{\text{out}}$, and use this to react via a feedback controller at the entry gates as in Figure 1. With this controller, we can regulate the number of people in the fairground. More accurately, the feedback controller $K = K(\kappa)$ includes admission rates $\kappa_i$ at entry gate $i$, and the number of people entering may be restricted based on the total weighted number of people inside the fairground. Letting $T_{w\to z}(\mathbf{x}, \kappa)$ denote the closed-loop transfer function of the performance channel mapping $w$ into $z$, this leads to the following problem where controller and parts of the plant are optimized simultaneously.

$$\begin{array}{lll} & \text{minimize} & \|T_{w\to z}(\mathbf{x}, \kappa)\|_H \\ (6) & \text{subject to} & K = K(\kappa) \text{ assures closed-loop stability,} \\ & & a_{jj'} \geqslant 0, a_{jk} \geqslant 0, b_{ij} \geqslant 0, \kappa_i \geqslant 0, (3), (4) \end{array}$$

## 5. Robust control of a mass-spring-damper system

In this section we discuss a 1DOF mass-spring-damper system with mass $m$, spring stiffness $k$ and damping coefficient $c$. The values can be in any consistent system of units, for example, in SI units, $m$ in kilograms, $k$ in newtons per meter, and $c$ in newton-seconds per meter or kilograms per second. The system is of second order, since it has a mass which can contain both kinetic and potential energy. The force $F$ is considered as input $u$, and the displacement $p$ of the mass from the equilibrium position is considered as output $y$ of this system. By Hooke's law, the force exerted by the spring is

$$F_s = -kp.$$

Let $v$ be the velocity of the mass, then the damping force $F_d$ is expressed as

$$F_d = -cv = -c\frac{dp}{dt} = -c\dot{p}$$

due to d'Alembert's principle. Using Newton's second law, we have

$$F + F_s + F_d = m\frac{d^2p}{dt^2} = m\ddot{p},$$

which gives

$$m\ddot{p} + c\dot{p} + kp = u.$$

A possible selection of state variables is the displacement $p$ and the velocity $v$. The linear model of the mass-spring-damper is then described by

$$G : \begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \text{ and } C = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

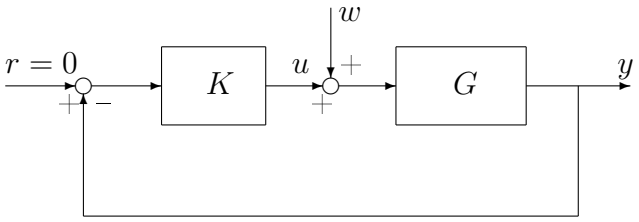



FIGURE 2. Structure of mass-spring-damper control system.

The design objective for the mass-spring-damper system with a disturbance is to find an output feedback control law $u = Ky$ which stabilizes the closed-loop system while minimizing worst-case energy of output $z = [y\ u]^\top$ in order to avoid the disturbance input $w$ to affect the system. In realistic systems, the physical parameters $k$ and $c$ are not known exactly but can be enclosed in intervals. Assuming the controller is parameterized as $K(\kappa)$, taking $\mathbf{x}$ to regroup the tunable parameters $k, c$ and $\kappa$, and denoting by $T_{w\to z}(\mathbf{x})$ the closed-loop performance channel $w \to z$, this leads to the optimization problem

$$\begin{array}{ll} \text{minimize} & \|T_{w\to z}(\mathbf{x})\| \\ \text{subject to} & \mathbf{x} = (k, c, \kappa) \in \mathbb{R}^n, \\ & K = K(\kappa) \text{ assures closed-loop stability,} \\ & k \text{ and } c \text{ are in some intervals} \end{array} \tag{7}$$

where choices of $\|\cdot\|$ include the $H_\infty$-norm $\|\cdot\|_\infty$ or the Hankel norm $\|\cdot\|_H$.

## 6. Clarke subdifferential of the Hankel norm

In order to apply nonlinear and nonsmooth optimization techniques to programs of the form (5), (6) and (7) it is necessary to provide derivative information of the objective function

$$f(\mathbf{x}) = \|G(\mathbf{x})\|_H^2 = \lambda_1(X(\mathbf{x})Y(\mathbf{x})),$$

where $X(\mathbf{x})$ and $Y(\mathbf{x})$ are the controllability and observability Gramians. In the discrete-time case, $X(\mathbf{x})$ and $Y(\mathbf{x})$ can be obtained from the Lyapunov equations

$$A(\mathbf{x})XA^\top(\mathbf{x}) - X + B(\mathbf{x})B^\top(\mathbf{x}) = 0, \tag{8}$$

$$A^\top(\mathbf{x})YA(\mathbf{x}) - Y + C^\top(\mathbf{x})C(\mathbf{x}) = 0. \tag{9}$$

Remark that despite the symmetry of $X$ and $Y$ the product $XY$ need not be symmetric, but stability of $A(\mathbf{x})$ guarantees $X \succeq 0$, $Y \succeq 0$ in (8), (9), so that we can write

$$\lambda_1(XY) = \lambda_1(X^{\frac{1}{2}}YX^{\frac{1}{2}}) = \lambda_1(Y^{\frac{1}{2}}XY^{\frac{1}{2}}),$$

which brings us back in the realm of eigenvalue theory of symmetric matrices.

Recalling the definition of the spectral radius of a matrix

$$\rho(M) = \max\{|\lambda| : \lambda \text{ eigenvalue of } M\},$$

we can address programs (5) and (6) in the following program

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) := \|G(\mathbf{x})\|_H^2 \\ \text{subject to} & c(\mathbf{x}) := \rho(A(\mathbf{x})) - 1 + \varepsilon \leqslant 0 \end{array} \tag{10}$$

for some fixed small $\varepsilon > 0$. Notice that $f = \|\cdot\|_H^2 \circ G(\cdot)$ is a composite function of a semi-norm and a smooth mapping $\mathbf{x} \mapsto G(\mathbf{x})$, which implies that it is lower-$C^2$, and therefore also lower-$C^1$ in the sense of [16, Definition 10.29]. Theoretical properties of the spectral radius $c(\mathbf{x})$, used in the constraint, have been studied in [3]. We also have $X(\mathbf{x}) \succeq 0$ and $Y(\mathbf{x}) \succeq 0$ on the feasible set $\mathcal{C} = \{\mathbf{x} : c(\mathbf{x}) \leqslant 0\}$, so that $f$ is well-defined and locally Lipschitz on $\mathcal{C}$.

Let $\mathbb{M}_{n,m}$ be the space of $n \times m$ matrices, equipped with the corresponding scalar product $\langle X, Y\rangle = \mathrm{Tr}(X^\top Y)$, where $X^\top$ and $\mathrm{Tr}(X)$ are respectively the transpose and the trace of matrix $X$. We denote by $\mathbb{B}_m$ the set of $m \times m$ symmetric positive semidefinite matrices with trace 1. Set $Z := X^{\frac{1}{2}}YX^{\frac{1}{2}}$ and pick $Q$ to be a matrix whose columns form an orthonormal basis of the $\nu$-dimensional eigenspace associated with $\lambda_1(Z)$. By [14, Theorem 3], the Clarke subdifferential of $f$ at $\mathbf{x}$ consists of all subgradients $g_U$ of the form

$$g_U = (\mathrm{Tr}(Z_1(\mathbf{x})^\top QUQ^\top), \ldots, \mathrm{Tr}(Z_n(\mathbf{x})^\top QUQ^\top))^\top,$$

where $U \in \mathbb{B}_\nu$, and where $M_i(\mathbf{x}) := \frac{\partial M(\mathbf{x})}{\partial \mathbf{x}_i}, i = 1, \ldots, n$ for any matrix $M(\mathbf{x})$.We next have

$$Z_i(\mathbf{x}) = \chi_i(\mathbf{x})YX^{\frac{1}{2}} + X^{\frac{1}{2}}Y_i(\mathbf{x})X^{\frac{1}{2}} + X^{\frac{1}{2}}Y\chi_i(\mathbf{x}), \tag{11}$$

where $\chi_i(\mathbf{x}) := \frac{\partial X^{\frac{1}{2}}(\mathbf{x})}{\partial \mathbf{x}_i}$. It follows from (8) and (9) that

$$\begin{aligned} &A(\mathbf{x})X_i(\mathbf{x})A^\top(\mathbf{x}) - X_i(\mathbf{x}) = -A_i(\mathbf{x})XA^\top(\mathbf{x}) \\ &- A(\mathbf{x})X[A_i(\mathbf{x})]^\top - B_i(\mathbf{x})B^\top(\mathbf{x}) - B(\mathbf{x})[B_i(\mathbf{x})]^\top, \end{aligned} \tag{12}$$

$$\begin{aligned} &A^\top(\mathbf{x})Y_i(\mathbf{x})A(\mathbf{x}) - Y_i(\mathbf{x}) = -[A_i(\mathbf{x})]^\top YA(\mathbf{x}) \\ &- A^\top(\mathbf{x})YA_i(\mathbf{x}) - [C_i(\mathbf{x})]^\top C(\mathbf{x}) - C^\top(\mathbf{x})C_i(\mathbf{x}). \end{aligned} \tag{13}$$

Since $X^{\frac{1}{2}}X^{\frac{1}{2}} = X$,

$$X^{\frac{1}{2}}\chi_i(\mathbf{x}) + \chi_i(\mathbf{x})X^{\frac{1}{2}} = X_i(\mathbf{x}). \tag{14}$$

Altogether, we obtain Algorithm 2 to compute elements of the subdifferential of $f(\mathbf{x})$.

**Algorithm 2.** Computing subgradients

**Input:** $\mathbf{x} \in \mathbb{R}^n$. **Output:** $g \in \partial f(\mathbf{x})$.

▷ **Step 1.** Compute $A_i(\mathbf{x}) = \frac{\partial A(\mathbf{x})}{\partial \mathbf{x}_i}, B_i(\mathbf{x}) = \frac{\partial B(\mathbf{x})}{\partial \mathbf{x}_i}$, $C_i(\mathbf{x}) = \frac{\partial C(\mathbf{x})}{\partial \mathbf{x}_i}$, $i = 1, \ldots, n$ and $X, Y$ solutions of (8), (9), respectively.

▷ **Step 2.** Compute $X^{\frac{1}{2}}$ and $Z = X^{\frac{1}{2}} Y X^{\frac{1}{2}}$.

▷ **Step 3.** For $i = 1, \ldots, n$ compute $X_i(\mathbf{x})$ and $Y_i(\mathbf{x})$ solutions of (12) and (13), respectively.

▷ **Step 4.** For $i = 1, \ldots, n$ compute $\chi_i(\mathbf{x})$ solution of (14) and $Z_i(\mathbf{x})$ using (11).

▷ **Step 5.** Determine a matrix $Q$ whose columns form an orthonormal basis of the $\nu$-dimensional eigenspace associated with $\lambda_1(Z)$.

▷ **Step 6.** Pick $U \in \mathbb{B}_\nu$, and return

$$(\mathrm{Tr}(Z_1(\mathbf{x})^\top QUQ^\top), \ldots, \mathrm{Tr}(Z_n(\mathbf{x})^\top QUQ^\top))^\top,$$

a subgradient of $f$ at $\mathbf{x}$.

*Remark* 1. In the continuous-time case, the Gramians $X(\mathbf{x})$ and $Y(\mathbf{x})$ can be obtained from the continuous Lyapunov equations

$$A(\mathbf{x})X + XA^\top(\mathbf{x}) + B(\mathbf{x})B^\top(\mathbf{x}) = 0, \tag{15}$$

$$A^\top(\mathbf{x})Y + YA(\mathbf{x}) + C^\top(\mathbf{x})C(\mathbf{x}) = 0, \tag{16}$$

Therefore, $X_i(\mathbf{x})$ and $Y_i(\mathbf{x})$ are solutions respectively of the following equations

$$\begin{aligned} A(\mathbf{x})X_i(\mathbf{x}) + X_i(\mathbf{x})A^\top(\mathbf{x}) &= -A_i(\mathbf{x})X - X[A_i(\mathbf{x})]^\top \\ &\quad - B_i(\mathbf{x})B^\top(\mathbf{x}) - B(\mathbf{x})[B_i(\mathbf{x})]^\top, \end{aligned} \tag{17}$$

$$\begin{aligned} A^\top(\mathbf{x})Y_i(\mathbf{x}) + Y_i(\mathbf{x})A(\mathbf{x}) &= -[A_i(\mathbf{x})]^\top Y - YA_i(\mathbf{x}) \\ &\quad - [C_i(\mathbf{x})]^\top C(x) - C^\top(\mathbf{x})C_i(\mathbf{x}). \end{aligned} \tag{18}$$

In addition, let us note that for this case, the stability constraint in program (10) is $c(\mathbf{x}) = \alpha(A(\mathbf{x})) + \varepsilon \leqslant 0$, where $\alpha(\cdot)$ denotes the spectral abscissa of a square matrix, i.e., the maximum of the real parts of its eigenvalues. □

We now introduce a smooth relaxation of Hankel norm. It is based on a result established by Y. Nesterov in [9], which gives a fine analysis of the convex bundle method in situations where the objective $f(\mathbf{x})$ has the specific structure of a max-function, including the case of a convex maximum eigenvalue function. These findings indicate that for a given precision, such programs may be solved with lower algorithmic complexity using smooth relaxations. While these results are *a priori* limited to the convex case, it may be interesting to apply this idea as a heuristic in the nonconvex situation. More precisely, we can try to solve problem (10), (2) by replacing the function $f(\mathbf{x}) = \lambda_1(Z(\mathbf{x}))$ by its smooth approximation

$$f_\mu(\mathbf{x}) := \mu \ln \left( \sum_{i=1}^{n_x} e^{\lambda_i(Z(\mathbf{x}))/\mu} \right), \tag{19}$$

where $\mu > 0$ is a tolerance parameter, $n_x$ the order of matrix $Z$, and where $\lambda_i$ denotes the $i$th eigenvalue of a symmetric or Hermitian matrix. Then

$$\nabla f_\mu(Z) = \left(\sum_{i=1}^{n_x} e^{\lambda_i(Z)/\mu}\right)^{-1} \sum_{i=1}^{n_x} e^{\lambda_i(Z)/\mu} q_i(Z) q_i(Z)^\top,$$

with $q_i(Z)$ the $i$th column of the orthogonal matrix $Q(Z)$ from the eigendecomposition of symmetric matrix $Z = Q(Z)D(Z)Q(Z)^\top$. This yields

$$\nabla f_\mu(\mathbf{x}) = (\mathrm{Tr}(Z_1(\mathbf{x})^\top \nabla f_\mu(Z)), \ldots, \mathrm{Tr}(Z_n(\mathbf{x})^\top \nabla f_\mu(Z)))^\top.$$

Let us note that

$$f(\mathbf{x}) \leqslant f_\mu(\mathbf{x}) \leqslant f(\mathbf{x}) + \mu \ln n_x.$$

Therefore, to find an $\varepsilon$-solution of problem (2), we have to find an $\frac{\varepsilon}{2}$-solution of the smooth problem

$$\begin{array}{ll} \text{minimize} & f_\mu(\mathbf{x}) \\ \text{subject to} & c(\mathbf{x}) \leqslant 0 \\ & A\mathbf{x} \leqslant b \end{array} \tag{20}$$

with $\mu = \frac{\varepsilon}{2 \ln n_x}$. This smoothed problem can be solved using standard NLP software. We have initialized the nonsmooth Algorithm 1 with the solution of problem (20).

## 7. Numerical experiments

**7.1. Steady flow in a graph.** We give an illustration of programs (5) and (6). Let $\mathscr{V}_{\text{stay}} = \{1, 2, \ldots, n_x\}$, $\mathscr{V}_{\text{in}} = \{1, 2, \ldots, m\}$ and $\mathscr{V}_{\text{out}} = \{1, 2, \ldots, p\}$. Taking $\mathbf{x}$ to regroup the unknown tunable parameters $a_{jj'}, b_{ij}$ and setting $A(\mathbf{x}) = [a_{jj'}]_{n_x \times n_x}^\top$, $B(\mathbf{x}) = [b_{ij}]_{m \times n_x}^\top$, $C = [c_1, \ldots, c_{n_x}]$, where $a_{jj'} = 0$ if $(j, j') \notin \mathscr{A}$, $b_{ij} = 0$ if $(i, j) \notin \mathscr{A}$, we have a discrete LTI system

$$G(\mathbf{x}) : \begin{cases} x(t+1) = A(\mathbf{x})x(t) + B(\mathbf{x})w(t) \\ z(t) \qquad\quad = Cx(t). \end{cases}$$

Note that the linear constraint conditions in (5) as well as (6) can be transferred to the form

$$\begin{cases} A_{\text{eq}}\mathbf{x} = b_{\text{eq}}, \\ \mathbf{x} \geqslant 0. \end{cases}$$

We now take the graph $\mathscr{G} = (\mathscr{V}, \mathscr{A})$ with $n_x = 36$, $m = 2$ and $p = 2$ as in Figure 3. Let $z(t)$ be the total number of individuals inside the fairground with

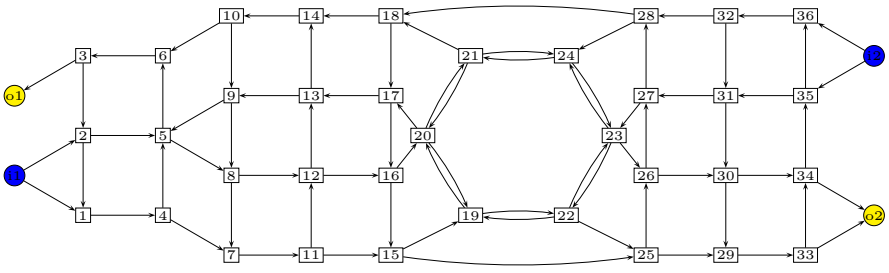



FIGURE 3. Model of the fairground

doubled weights at 6 nodes in the center that form a hexagon as compared to the other nodes. We start with the case without controller and initialize at the uniform distribution $\mathbf{x}^1$, where $f(\mathbf{x}^1) = 528.7672$ and $\|G(\mathbf{x}^1)\|_H = 22.9949$. In order to save time, we use the minimizer of the relaxation $f_\mu(\mathbf{x})$ in (19) with initial $\mathbf{x}^1$ to initialize the nonsmooth Algorithm 1. Our algorithm then returns the optimal $\mathbf{x}^\dagger$ with $f(\mathbf{x}^\dagger) = 16.5817$, meaning $\|G(\mathbf{x}^\dagger)\|_H = 4.0721$.

In the case with controller $K = K(\kappa)$, $\kappa = [\kappa_1 \ \dots \ \kappa_m]^\top$, as shown in Figure 1, we have

$$T_{w\to z}(\mathbf{x},\kappa): \begin{cases} x(t+1) = A(\mathbf{x})x(t) + B(\mathbf{x})e(t) \\ z(t) \qquad\quad = Cx(t). \end{cases}$$

Here $e(t) = w(t) - Kz(t) = w(t) - KCx(t)$, which gives

$$T_{w\to z}(\mathbf{x},\kappa) = \left[\begin{array}{c|c} A(\mathbf{x}) - B(\mathbf{x})K(\kappa)C & B(\mathbf{x}) \\ \hline C & 0 \end{array}\right].$$

Initializing at $(\mathbf{x},\kappa) = (\mathbf{x}^1, 0)$ with remarking that $T_{w\to z}(\mathbf{x},0) = G(\mathbf{x})$ and proceeding as in the previous case, we obtain the optimal $(\mathbf{x}^*,\kappa^*)$ with $f(\mathbf{x}^*,\kappa^*) = 2.0001$, meaning $\|T_{w\to z}(\mathbf{x}^*,\kappa^*)\|_H = 1.4142$. Step responses and ringing effects in unit step and white noise responses truncated at $T = 30$ for the three systems $G(\mathbf{x}^1) = T_{w\to z}(\mathbf{x}^1,0)$, $G(\mathbf{x}^\dagger)$ and $T_{w\to z}(\mathbf{x}^*,\kappa^*)$ are compared in Figure 4 and Figure 5.

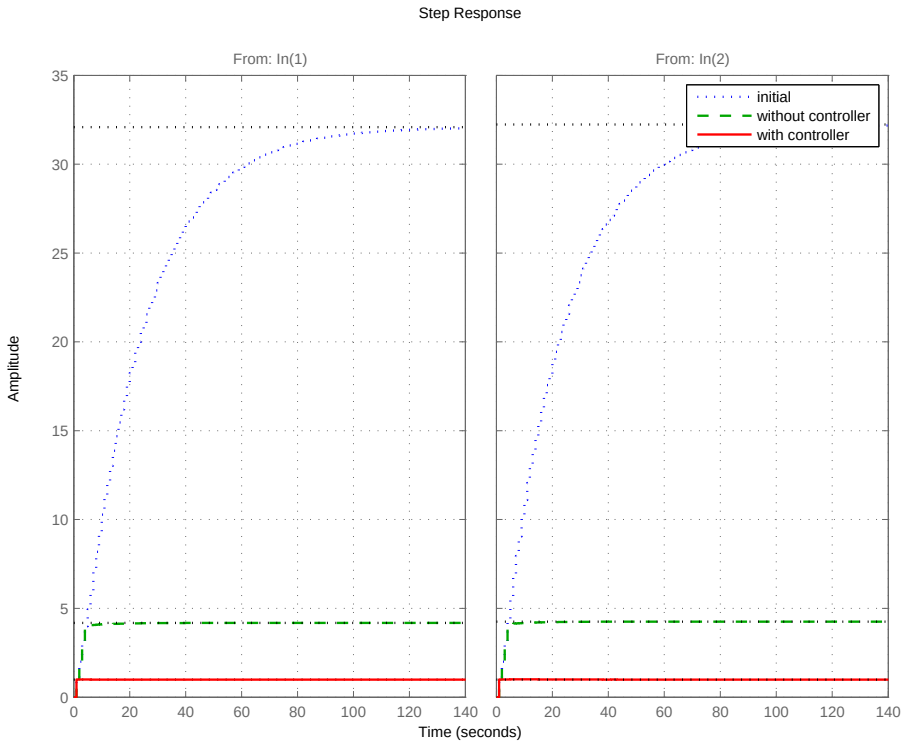



FIGURE 4. Experiment 1. Step responses of three systems $G(\mathbf{x}^1)$ (dotted), $G(\mathbf{x}^\dagger)$ (dashed) and $T_{w\to z}(\mathbf{x}^*,\kappa^*)$ (solid)

**7.2. Robust control of a mass-spring-damper system.** Here we apply Algorithm 1 to solve problem (7), where the mass-spring-damper plant with a disturbance

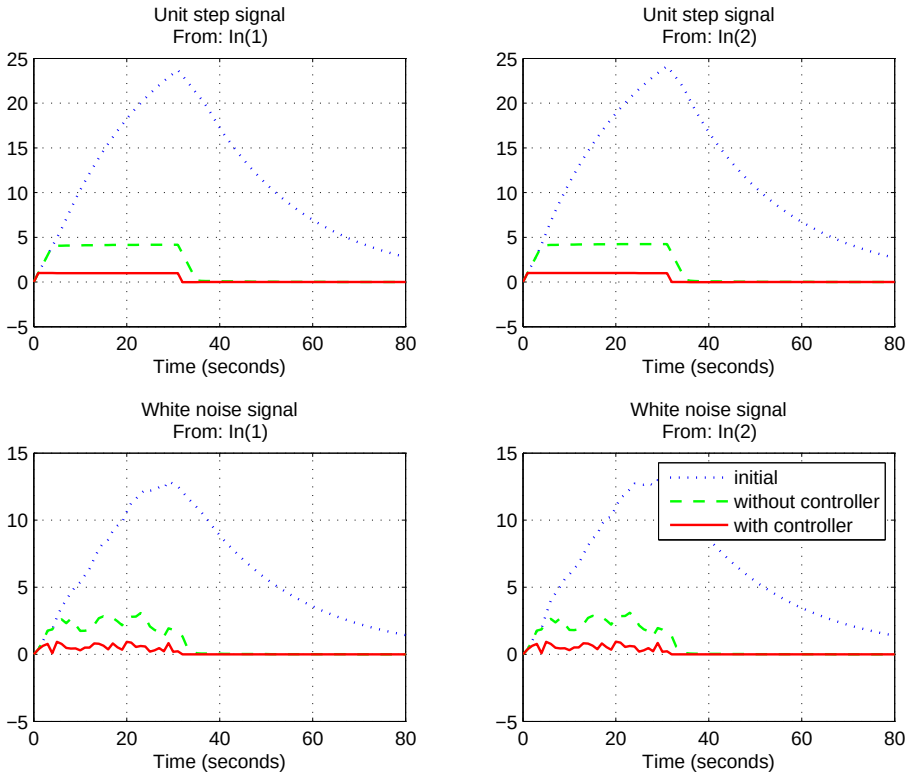



FIGURE 5. Experiment 1. Ringing effects of three systems $G(\mathbf{x}^1)$ (dotted), $G(\mathbf{x}^\dagger)$ (dashed) and $T_{w\to z}(\mathbf{x}^*, \kappa^*)$ (solid). Input: Unit step signal (top) and white noise signal (bottom)

is given by

$$P: \begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \left[\begin{array}{c|cc} A & B_1 & B \\ \hline C_1 & 0 & D_{12} \\ C & 0 & 0 \end{array}\right] \begin{bmatrix} x \\ w \\ u \end{bmatrix},$$

with

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix}, \quad B_1 = B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}$$
$$C_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad D_{12} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{and } C = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

The controller $K$ is chosen of order 2, namely

$$K(\kappa) = \frac{\kappa_1 s^2 + \kappa_2 s + \kappa_3}{s^2 + \kappa_4 s + \kappa_5}$$
$$= \left[\begin{array}{cc|c} -\kappa_4 & \kappa_5 & 1 \\ 1 & 0 & 0 \\ \hline \kappa_2 - \kappa_1\kappa_4 & \kappa_3 - \kappa_1\kappa_5 & \kappa_1 \end{array}\right] := \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array}\right].$$

Then, the closed-loop transfer function of the performance channel channel $w \to z$ has the state-space representation

$$T_{w\to z}(\mathbf{x}): \begin{bmatrix} \dot{\xi} \\ z \end{bmatrix} = \begin{bmatrix} A(\mathbf{x}) & B(\mathbf{x}) \\ C(\mathbf{x}) & 0 \end{bmatrix} \begin{bmatrix} \xi \\ w \end{bmatrix},$$

where $\xi = [x\ x_K]^\top$, $x_K$ the state of $K$, and where

$$A(\mathbf{x}) = \begin{bmatrix} A + BD_KC & BC_K \\ B_KC & A_K \end{bmatrix},$$
$$B(\mathbf{x}) = \begin{bmatrix} B_1 + BD_KD_{21} \\ B_KD_{21} \end{bmatrix},$$
$$C(\mathbf{x}) = \begin{bmatrix} C_1 + D_{12}D_KC & D_{12}C_K \end{bmatrix}.$$

Assume that mass $m = 4$, and spring stiffness $k$ and damping coefficient $c$ belong to the intervals $[4, 12]$ and $[0.5, 1.5]$, respectively. Using the Matlab function `hinfstruct` based on [1], we optimized $H_\infty$-norm and obtained $k = 12, c = 1$ and

$$K_\infty = \frac{-6.0927s^2 - 0.3981s - 5.1816}{s^2 + 19.0834s + 1.1708}.$$

In the Hankel norm synthesis case, our Algorithm 1 returned $k = 12, c = 1.5$ and

$$K_H = \frac{-6.1975s^2 - 2.1828s - 4.2523}{s^2 + 19.3261s + 3.9198}.$$

Figure 6 compares step responses and white noise responses in two synthesis cases. Bearing of the algorithm is shown in Figure 7.

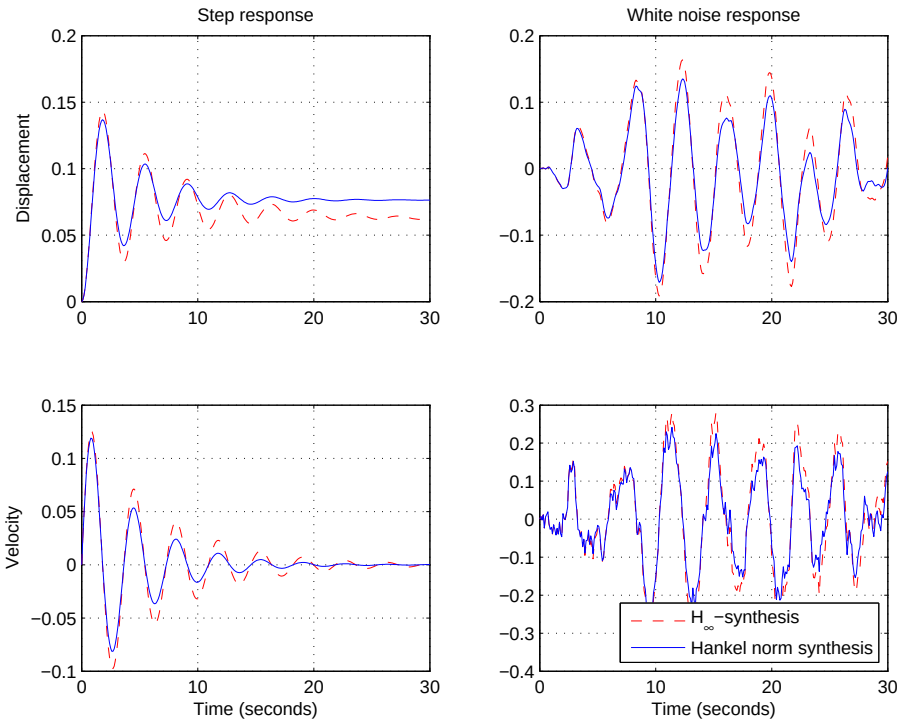



FIGURE 6. Experiment 2. Step responses (left) and white noise responses (right) in two synthesis cases

## 8. Conclusion

We have shown that it is possible to optimize plant and controller simultaneously if the idea of a structured control law introduced in [1] is applied. Our approach was

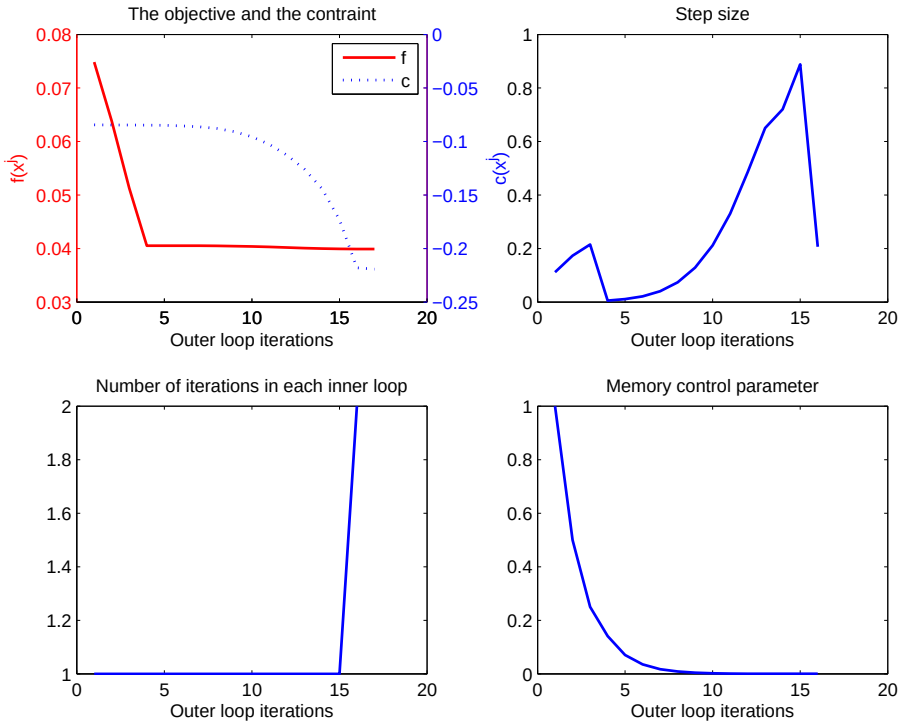



FIGURE 7. Experiment 2. Bearing of the algorithm

illustrated for Hankel norm synthesis as well as for $H_\infty$-synthesis, and for a continuous and a discrete system. Due to inherent nonsmoothness of the cost functions, nonsmooth optimization was applied, and in particular, a nonconvex bundle method was presented. For eigenvalue optimization, as required for Hankel norm synthesis, a relaxation developed by Nesterov for the convex case was successfully used as a heuristic in the nonconvex case to initialize the bundle method.

## References


1. P. Apkarian and D. Noll, *Nonsmooth $H_\infty$ synthesis*, IEEE Trans. Automat. Control **51** (2006), no. 1, 71–86.
2. _____, *Nonsmooth optimization for multidisk $H_\infty$ synthesis*, Eur. J. Control **12** (2006), no. 3, 229–244.
3. J. V. Burke and M. L. Overton, *Differential properties of the spectral abscissa and the spectral radius for analytic matrix-valued mappings*, Nonlinear Anal. **23** (1994), no. 4, 467–488.
4. F. H. Clarke, *Generalized gradients of Lipschitz functionals*, Adv. in Math. **40** (1981), no. 1, 52–67.
5. _____, *Optimization and nonsmooth analysis*, Canad. Math. Soc. Ser. Monogr. Adv. Texts, John Wiley & Sons, Inc., New York, 1983.
6. M. N. Dao and D. Noll, *Minimizing the memory of a system*, Proc. Asian Control Conf. (Istanbul), June 2013.
7. M. Gabarrou, D. Alazard, and D. Noll, *Design of a flight control architecture using a non-convex bundle method*, Math. Control Signals Syst. **25** (2013), no. 2, 257–290.
8. K. Glover, *All optimal Hankel-norm approximations of linear multivariable systems and their $L^\infty$-error bounds*, Internat. J. Control **39** (1984), no. 6, 1115–1193.
9. Y. Nesterov, *Smoothing technique and its applications in semidefinite optimization*, Math. Program., Ser. A **110** (2007), no. 2, 245–259.

10. D. Noll, *Cutting plane oracles to minimize non-smooth non-convex functions*, Set-Valued Var. Anal. **18** (2010), no. 3-4, 531–568.
11. ———, *Convergence of non-smooth descent methods using the Kurdyka-Łojasiewicz inequality*, J. Optim. Theory Appl. **160** (2014), no. 2, 553–572.
12. D. Noll, O. Prot, and A. Rondepierre, *A proximity control algorithm to minimize nonsmooth and nonconvex functions*, Pac. J. Optim. **4** (2008), no. 3, 571–604.
13. A. M. Ostrowski, *Solutions of equations in Euclidean and Banach spaces*, Pure and Applied Mathematics, vol. 9, Academic Press, New York-London, 1973.
14. M. L. Overton, *Large-scale optimization of eigenvalues*, SIAM J. Optim. **2** (1992), no. 1, 88–120.
15. E. Polak, *Optimization: Algorithms and consistent approximations*, Appl. Math. Sci., vol. 124, Springer-Verlag, New York, 1997.
16. R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, Springer-Verlag, Berlin, 1998.

# IV

## Robust eigenstructure clustering by nonsmooth optimization*

Minh Ngoc Dao, Dominikus Noll, and Pierre Apkarian

**Abstract.** We extend classical eigenstructure assignment to more realistic problems where additional performance and robustness specifications arise. Our aim is to combine time-domain constraints, as reflected by pole location and eigenvector structure, with frequency-domain objectives such as the $H_2$, $H_\infty$ or Hankel norms. Using pole clustering, we allow poles to move in polydisks of prescribed size around their nominal values, driven by optimization. Eigenelements, that is poles and eigenvectors, are allowed to move simultaneously and serve as decision variables in a specialized nonsmooth optimization technique. Two aerospace applications illustrate the power of the new method.

**Keywords.** Structured feedback control · eigenstructure assignment · modal shaping · nonsmooth optimization · frequency-domain · robust design

### 1. Introduction

Since its introduction by Wonham [30] and Moore [17], eigenstructure assignment has developed into a powerful controller design tool in the aerospace sector and in other high technology fields. Eigenstructure assignment aims at shaping the responses of the closed-loop system to certain input signals by way of two mechanisms. The placement of closed-loop modes to stabilize and achieve satisfactory transients, and eigenvector structure to decouple responses to specific initial conditions. In this paper we are concerned with the design of output feedback control laws, where only partial eigenstructure assignment or pole placement can be expected. In that case the standard approach to first selecting a partial set of closed-loop modes $\lambda_1, \ldots, \lambda_p$, and then using the remaining degrees of freedom to shape the corresponding closed-loop eigenvectors $(v_i, w_i)$, is prone to failure to stabilize the system, as the remaining closed-loop modes cannot be influenced directly.

*Paper submitted for publication. Conference version published in Proc. Internat. Conf. Informatics in Control, Automation and Robotics (ICINCO), Reykjavík, July 2013, pp. 307–314.

As a remedy we propose to assign the eigenelements $(\lambda_i, v_i, w_i)$ *simultaneously*. We allow eigenelements $(\lambda_i, v_i, w_i)$ to move in the neighborhood of their nominal values $(\lambda_i^0, v_i^0, w_i^0)$ in such a way that closed-loop stability and performance can be further improved. The price for this gain of flexibility is that eigenelement assignment can no longer be achieved by linear algebra methods alone. Instead, a combination of nonlinear optimization and linear algebra is required.

Over the years there have already been attempts to enhance eigenspace control using off-the-shelf optimization. An early approach is Sobel and Shapiro [26], where hand-tuning of eigenvalues was shown to improve stability margins of the controlled system. In [27] the same authors elaborate on this idea and suggest a first-order gradient method. In [1, 18], a sequential quadratic programming (SQP) technique with finite-difference gradients was used to improve $\mu$ robustness indicators, with eigenvalues and some eigenvectors as decision variables. In [22], Patton and Liu make full use of the freedom offered by eigenstructure assignment to improve the frequency-domain sensitivities functions $S$ and $KS$. They use a genetic algorithm in tandem with gradient-based techniques. The same idea is applied to a variety of problems in their monograph [14]. In the same vein, reference [13] exploits the Nelder-Mead direct search method to optimize assignable eigenvalues and eigenvectors, while safeguarding stability of unassigned eigenvalues via constraints. In [15], eigenstructure assignment with dynamic compensators and linear programming (LP) or quadratic programming (QP) are used to achieve stability and performance for an entire family of plants. Merits of these approaches have been demonstrated in numerous applications. See [14] and references therein.

In this work, we suggest a novel approach to eigenstructure assignment based on a nonsmooth optimization technique, which has the following features:

- Unassigned poles are constrained to be stable, which secures stability of the closed-loop system.
- Additional performance or robustness requirements such as $H_2$ or $H_\infty$ are handled rigorously by accounting for their nonsmoothness.

Nonsmoothness arises due to the spectral abscissa, and via $H_\infty$-norm or Hankel norm based requirements, but also when max-function of differentiable functions such as the $H_2$-norm are built. The key observation is that disregarding nonsmoothness is a serious source of numerical trouble. Avoiding this pitfall is a central motivation of this work. Our investigation leads to a theoretically justified nonsmooth method with local convergence certificate, which has good performance in practical applications. The focus of this paper is on control aspects. A thorough convergence analysis of the proposed algorithm is given in [21, 19, 8, 6] for the interested readers.

The structure of the paper is as follows. Section 2 recalls the basics of eigenstructure assignment using static output feedback and its variation as pole clustering, where poles are allowed to move in small polydisks around their nominal values. Section 3 extends the pole clustering problem to a variety of performance or robustness criteria and gives a pseudo-code of our algorithmic approach to those problems. Overdetermined and underdetermined eigenproblems are discussed in Section 4. Section 5 shows how subgradients are computed for typical design requirements. Our nonsmooth solver, along with its convergence properties, is presented in Section 6.

Sections 7 and 8 illustrate our approach. We design a launcher and an aircraft control system, two cases where poles and eigenvector structure play an important role.

## 2. Partial eigenstructure assignment

Consider a linear time-invariant system described by the equations

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{1}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$. Given a self-conjugate set $\Lambda = \{\lambda_1, \ldots, \lambda_p\} \subset \mathbb{C}^-$, partial pole placement consists in computing a static output feedback control law $u = Ky$ for (1) such that $\lambda_1, \ldots, \lambda_p$ become eigenvalues of the closed-loop system

$$\dot{x} = (A + BKC)x.$$

As is well-known [17], solving the set of linear equations

$$\left[\begin{array}{c|c} A - \lambda_i I_n & B \end{array}\right] \left[\begin{array}{c} v_i \\ \hline w_i \end{array}\right] = 0,$$

with $v_i \in \mathbb{C}^n$, $w_i \in \mathbb{C}^m$, $i = 1, \ldots, p$ leads to a (static) control law

$$K = [w_1, \ldots, w_p] \left(C\,[v_1, \ldots, v_p]\right)^{-1} \in \mathbb{R}^{m \times p} \tag{2}$$

with the desired closed-loop modes, provided the $v_i$ are chosen in such a way that the $p \times p$ matrix $C\,[v_1, \ldots, v_p]$ is invertible, i.e., if $\operatorname{span}\{v_1, \ldots, v_p\} \cap \ker(C) = \{0\}$. Note that the outlined technique is readily specialized to state-feedback $C = I$ and extended to nonzero feedthrough $D \neq 0$ and to dynamic compensators through a preliminary augmentation of the plant [28].

In the case $m > 1$, it is possible to achieve more. One may then additionally *shape* the $v_i$, or $w_i$, e.g. by arranging $v_{ij} = 0$ or $w_{ik} = 0$ for certain $j, k$. Formally this can be expressed by linear equations

$$\left[\begin{array}{c|c} A - \lambda_i I_n & B \\ \hline M_i & N_i \end{array}\right] \left[\begin{array}{c} v_i \\ \hline w_i \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline r_i \end{array}\right], \tag{3}$$

with suitable $M_i \in \mathbb{C}^{m_i \times n}$, $N_i \in \mathbb{C}^{m_i \times m}$, $r_i \in \mathbb{C}^{m_i}$, $m_i \geqslant 0$, $i = 1, \ldots, p$, leaving at least one degree of freedom in each triplet $(\lambda_i, v_i, w_i) \in \mathbb{C}^{1+n+m}$. This is usually referred to as *partial eigenstructure assignment*. Typical choices of $M_i, N_i, r_i$ can be found in our experimental Sections 7 and 8.

The traditional approach to eigenstructure assignment consists in first choosing the set $\Lambda \subset \mathbb{C}^-$, then introducing the desired structural constraints on the eigenvectors $v_i, w_i$ via the matrices $M_i, N_i$ and the vector $r_i$, using the remaining degrees of freedom, and then computing $v_i, w_i$ accordingly. Unfortunately, fixing the $\lambda_i$ may be too restrictive, because partial eigenvalue placement does not guarantee stability in closed-loop, so that some post-processing based on trial-and-error is often required. Greater flexibility in the design is achieved by moving $(\lambda_i, v_i, w_i)$ simultaneously.

What we have in mind is to interpret the eigenstructure equations (3) as mathematical programming constraints and then optimize closed-loop stability subject

to these constraints. With the definition $\alpha(A) := \max\{\operatorname{Re}\lambda : \lambda \text{ eigenvalue of } A\}$ of the spectral abscissa, this leads us to an optimization program of the form

$$
\begin{array}{ll}
\text{minimize} & \alpha(A+BKC) \\
\text{subject to} & \left[\begin{array}{c|c} A-\lambda_i I_n & B \\ \hline M_i & N_i \end{array}\right] \left[\begin{array}{c} v_i \\ \hline w_i \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline r_i \end{array}\right] \text{ for } i = 1,\dots,p \\
& |\operatorname{Re}\lambda_i - \operatorname{Re}\lambda_i^0| \leqslant \delta_i, |\operatorname{Im}\lambda_i - \operatorname{Im}\lambda_i^0| \leqslant \delta_i, i = 1,\dots,p \\
& K = W(CV)^{-1} \text{ as in (2).}
\end{array} \tag{4}
$$

Here the $\lambda_i^0 \in \mathbb{C}^-$ are nominal closed-loop poles, and the $\delta_i$ are tolerances which allow the poles to move around their nominal values. As soon as $K$ with $\alpha(A+BKC) < 0$ is reached, the optimization of (4) can be stopped with an internally stabilizing solution of the partial eigenstructure assignment procedure.

## 3. Including performance criteria

While (4) is a natural approach to optimize closed-loop stability in partial eigenstructure assignment, it seems even more attractive to include also closed-loop performance or robustness criteria into the set-up. Given a linear time-invariant plant $P$ in standard form

$$
P: \quad \begin{cases} \dot{x} = & Ax & + & B_1 w & + & Bu \\ z = & C_1 x & + & D_{11} w & + & D_{12} u \\ y = & Cx & + & D_{21} w & & \end{cases} \tag{5}
$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ the vector of control inputs, $w \in \mathbb{R}^{m_1}$ the vector of exogenous inputs, $y \in \mathbb{R}^p$ the vector of measurements and $z \in \mathbb{R}^{p_1}$ the controlled or performance vector, let $u = Ky$ be a static output feedback control law for (5). Then the closed-loop performance channel $w \to z$ has the state-space representation

$$
T_{w\to z}(K): \quad \begin{cases} \dot{x} &= & (A+BKC)x & + & (B_1+BKD_{21})w \\ z &= & (C_1+D_{12}KC)x & + & (D_{11}+D_{12}KD_{21})w. \end{cases}
$$

Note the slight abuse of notation in (5) because the state-space data of $P$ may include filters, weightings or other dynamic elements that are not present in (1). We assume the distinction will be clear from the context.

Given a self-conjugate eigenvalue set $\Lambda^0 = \{\lambda_1^0, \dots, \lambda_p^0\} \subset \mathbb{C}^-$ and tolerances $\delta_i$, we now consider the following extension of (4):

$$
\begin{array}{ll}
\text{minimize} & \|T_{w\to z}(K)\| \\
\text{subject to} & \left[\begin{array}{c|c} A-\lambda_i I_n & B \\ \hline M_i & N_i \end{array}\right] \left[\begin{array}{c} v_i \\ \hline w_i \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline r_i \end{array}\right] \text{ for } i = 1,\dots,p \\
& |\operatorname{Re}\lambda_i - \operatorname{Re}\lambda_i^0| \leqslant \delta_i, |\operatorname{Im}\lambda_i - \operatorname{Im}\lambda_i^0| \leqslant \delta_i, i = 1,\dots,p \\
& K = K(\lambda, v, w) \text{ as in (2)}
\end{array} \tag{6}
$$

where $\lambda_i^0$ are nominal closed-loop pole positions, and (3) again conveys additional structural constraints on $v, w$. As compared to (4), the cost function $\|T_{w\to z}(K)\|$ in (6) may now be used to enhance stability and to achieve additional performance or robustness specifications of the design.

Standard choices of $\|\cdot\|$ include the $H_\infty$-norm $\|\cdot\|_\infty$, the $H_2$-norm $\|\cdot\|_2$, or the Hankel norm $\|\cdot\|_H$. One generally expects that $\|T_{w\to z}(K)\| < \infty$ implies closed-loop stability, but should this fail, it is possible to add a stability constraint $c(\lambda, v, w) = \alpha(A+BKC)+\varepsilon \leqslant 0$ to the cast (6), where $\varepsilon > 0$ is some small threshold. Altogether we propose the following

**Algorithm 1.** Optimized partial eigenstructure assignment

**Input:** Nominal modal set $\Lambda^0 = \{\lambda_1^0, \ldots, \lambda_p^0\}$ with distinct $\lambda_i^0$.
**Output:** Optimal modal set $\Lambda = \{\lambda_1, \ldots, \lambda_p\}$, $v_i, w_i$, $K^*$.

▷ **Step 1** (`Nominal assignment`)**.** Perform standard eigenstructure assignment based on $\Lambda^0$ and structural constraints $M_i, N_i, r_i$. Obtain nominal eigenvectors $v_i^0$, $w_i^0$, $i = 1, \ldots, p$. Assure that $C[v_1^0, \ldots, v_p^0]$ is invertible and obtain nominal $K^0 = W^0(CV^0)^{-1}$.

◇ **Step 2** (`Stability and performance`)**.** If $K^0$ assures closed-loop stability and good performance $\|T_{w\to z}(K^0)\|$, stop the algorithm. Otherwise, goto step 3.

▷ **Step 3** (`Tolerances`)**.** Allow tolerances $|\operatorname{Re}\lambda_i - \operatorname{Re}\lambda_i^0| \leqslant \delta_i$, $|\operatorname{Im}\lambda_i - \operatorname{Im}\lambda_i^0| \leqslant \delta_i$, $i = 1, \ldots, p$.

▷ **Step 4** (`Parametric clustering`)**.** Solve the optimization program (6) using a nonsmooth descent algorithm with $(\lambda^0, v^0, w^0)$ as initial seed.

▷ **Step 5** (`Synthesis`)**.** Return optimal $\Lambda = \{\lambda_1, \ldots, \lambda_p\}$, $v, w$, and $K^*$.

## 4. Structure of eigenproblems

In this section we discuss practical ways to deal with the general nonlinear constraint (3) in (6). We assume that $(A, B)$ is controllable, which is equivalent to $[A - \lambda I_n \; B]$ having full row rank $n$ for all $\lambda$ in $\mathbb{C}$ (see, e.g., [31, Theorem 3.1]). To deal with (3), we observe that the $m_i$'s can be distinct and the possibility $m_i = 0$ is not excluded. We now distinguish two cases.

The first case is when $m_i \geqslant m$. Here pole assignment is ensured by pre-solving for $v_i$ in (3). We get

$$v_i = (\lambda_i I - A)^{-1} B w_i \,.$$

In this case eigenvector decoupling is only possible in the least-square sense by minimizing the Euclidean norm of $M_i v_i + N_i w_i - r_i$. Upon defining the transfer function $F_i(\lambda) := M_i(\lambda I - A)^{-1}B + N_i$, and assuming for simplicity that $F_i(\lambda)$ has full-column rank for $\lambda$ in the neighborhood of the nominal $\lambda^0$, we have

$$w_i = F_i(\lambda_i)^\dagger r_i \,,$$

where $F_i(\lambda_i)^\dagger$ denotes the Moore-Penrose inverse or left-inverse of $F_i$ at $\lambda_i$. Altogether we have derived the expression

$$\left[\frac{v_i}{w_i}\right] = \left[\frac{(\lambda_i I - A)^{-1}B}{I}\right] F_i(\lambda_i)^\dagger r_i \,. \tag{7}$$

Vectors $v_i$ and $w_i$ are now defined explicitly as functions of $\lambda_i$. It follows that a parametrization of the control law (2) in the sense of structured synthesis introduced

in [2] has been obtained. Tunable variables in this parametrization are the desired assignable eigenvalues $\Lambda = \{\lambda_1, \ldots, \lambda_p\}$.

The rationale in this first case is as follows. We want to gain some flexibility in the assignment by allowing $\lambda_i$ to move in a neighborhood of the nominal $\lambda_i^0$. Now if the $(v_i^0, w_i^0)$ are computed from (7) for the nominal value $\lambda_i^0$, the $(v_i, w_i)$, depending continuously on $\lambda_i$ via (7), will move in a neighborhood of the nominal $(v_i^0, w_i^0)$, so that optimization may decrease the cost function and thereby enhance stability and performance. The outlined approach therefore generalizes eigenstructure assignment with approximate decoupling as discussed in [28].

If $F_i(\lambda)$ is not guaranteed to have full-column rank in the neighborhood $\lambda^0$, the cast in (6) could be modified as follows:

$$
\begin{array}{lll}
& \text{minimize} & \max\left\{ \|T_{w\to z}(K)\|, \mu \max\limits_{i=1,\ldots,p} \|F_i(\lambda_i) w_i - r_i\|_2 \right\} \\
(8) & \text{subject to} & |\operatorname{Re}\lambda_i - \operatorname{Re}\lambda_i^0| \leqslant \delta_i,\ |\operatorname{Im}\lambda_i - \operatorname{Im}\lambda_i^0| \leqslant \delta_i,\ i = 1, \ldots, p. \\
& & K = [w_1, \ldots, w_p]\,(C\,[v_1, \ldots, v_p])^{-1} \\
& & K \text{ closed-loop stabilizing}
\end{array}
$$

where $\mu$ is a penalty parameter used to weigh the relative importance of robustness and performance as expressed through $\|T_{w\to z}(K)\|$ against eigenvector shaping. Here the objective becomes a max-function which is truly nonsmooth and thus requires special handling.

The second case is when $m_i < m$. Here we partition

$$
B = [B_i\ Q_i],\ N_i = [P_i\ R_i],\ w_i = \begin{bmatrix} u_i \\ t_i \end{bmatrix},
$$

such that $B_i, P_i$ have $m_i$ columns and $u_i \in \mathbb{C}^{m_i}$. Then (3) becomes

$$
\left[\begin{array}{c|c} A - \lambda_i I_n & B_i \\ \hline M_i & P_i \end{array}\right] \left[\begin{array}{c} v_i \\ \hline u_i \end{array}\right] = \left[\begin{array}{c} 0 \\ r_i \end{array}\right] - \left[\begin{array}{c} Q_i \\ R_i \end{array}\right] t_i.
$$

Assuming that the matrix

$$
\mathbf{A}_i(\lambda_i) = \left[\begin{array}{c|c} A - \lambda_i I_n & B_i \\ \hline M_i & P_i \end{array}\right] \in \mathbb{C}^{(n+m_i)\times(n+m_i)}
$$

is invertible in a neighborhood of the nominal $\lambda_i^0$, we get the parametrization

$$
v_i = v_i(\lambda_i, t_i),\ u_i = u_i(\lambda_i, t_i),
$$

which in explicit form is

$$
\left[\begin{array}{c} v_i \\ \hline u_i \end{array}\right] = \mathbf{A}_i(\lambda_i)^{-1} \left[\begin{array}{c} -Q_i t_i \\ r_i - R_i t_i \end{array}\right]. \tag{9}
$$

The idea is now the same as in the first case. Allow $\lambda_i$ to move around their nominal values $\lambda_i^0$, and $t_i \in \mathbb{C}^{m-m_i}$ around the nominal $t_i^0$. That also allows the dependent variables $v_i, u_i$ to move in a neighborhood of their nominal values $v_i^0, u_i^0$, and optimization uses this to enhance stability and robustness. In this second case we have enough degrees of freedom to achieve true decoupling of some of the channels by satisfying $M_i v_i + N_i w_i = r_i$ exactly.

In order to apply nonlinear and nonsmooth optimization techniques to programs of the form (6) it is necessary to provide derivative information at acceptable cost. As we shall see, this may be implemented by simple linear algebra techniques. We have the following propositions with proofs given in the Appendix.

**Proposition 4.1** (Over-specified eigenstructure)**.** *Let* $K = W(CV)^{-1}$ *with* $W = [w_1 \dots w_p]$ *and* $V = [v_1 \dots v_p]$*. If* $m_i \geqslant m$ *then*

$$\frac{\partial K}{\partial \lambda_i} = \left[0 \cdots \frac{\partial w_i}{\partial \lambda_i} - KC\frac{\partial v_i}{\partial \lambda_i} \cdots 0\right](CV)^{-1}, \tag{10}$$

*where* $v_i$*,* $w_i$ *are given in (7) and*

$$\left[\begin{array}{c} \dfrac{\partial v_i}{\partial \lambda_i} \\ \hline \dfrac{\partial w_i}{\partial \lambda_i} \end{array}\right] = \left[\begin{array}{c} (\lambda_i I - A)^{-1} B F_i(\lambda_i)^\dagger M_i - I \\ \hline F_i(\lambda_i)^\dagger M_i \end{array}\right](\lambda_i I - A)^{-2} B F_i(\lambda_i)^\dagger r_i\,. \tag{11}$$

*Proof.* See Appendix. □

**Proposition 4.2** (Under-specified eigenstructure)**.** *Let* $K = W(CV)^{-1}$ *with* $W = [w_1 \dots w_p]$ *and* $V = [v_1 \dots v_p]$*. Suppose* $m_i < m$*, partitioning* $w_i = \begin{bmatrix} u_i \\ t_i \end{bmatrix}$ *with* $u_i \in \mathbb{C}^{m_i}$ *and* $t_i = [t_{1i}, \dots, t_{(m-m_i)i}]^\top \in \mathbb{C}^{m-m_i}$*, then*

$$\begin{aligned} \frac{\partial K}{\partial \lambda_i} &= \left[0 \cdots \left| \begin{smallmatrix} \frac{\partial u_i}{\partial \lambda_i} \\ 0 \end{smallmatrix} \right| - KC\frac{\partial v_i}{\partial \lambda_i} \cdots 0\right](CV)^{-1}, \\ \frac{\partial K}{\partial t_{ki}} &= \left[0 \cdots \left| \begin{smallmatrix} \frac{\partial u_i}{\partial t_{ki}} \\ e_{ki} \end{smallmatrix} \right| - KC\frac{\partial v_i}{\partial t_{ki}} \cdots 0\right](CV)^{-1}, \end{aligned} \tag{12}$$

*where* $e_{ki} \in \mathbb{R}^{m-m_i}$ *is the vector all of whose components are zero, except the* $k$*th component which is one, and*

$$\left[\begin{array}{c|c} \dfrac{\partial v_i}{\partial \lambda_i} & \dfrac{\partial v_i}{\partial t_{ki}} \\ \hline \dfrac{\partial u_i}{\partial \lambda_i} & \dfrac{\partial u_i}{\partial t_{ki}} \end{array}\right] = \left[\begin{array}{cc} I_n & 0_{n\times m_i} \\ \hline 0_{m_i \times n} & I_{m_i} \end{array}\right] \mathbf{A}_i(\lambda_i)^{-1} \left[\begin{array}{c|c} v_i & \\ 0 & -s_{ik} \end{array}\right], \tag{13}$$

*with* $s_{ik}$ *the* $k$*th column of* $\begin{bmatrix} Q_i \\ R_i \end{bmatrix}$*.*

*Proof.* See Appendix. □

*Remark* 1. As derivatives have to be evaluated repeatedly in minimization programs, it is desirable to pre-calculate as many elements as possible in (10) and (12). This is what we discuss next. Substantial speed-up can be achieved in the under-specified case $m_i < m$ since $\mathbf{A}_i(\lambda_i)$ is a reduced rank modification of a constant matrix, that is, not depending on $\lambda_i$. We therefore pre-compute

$$\begin{bmatrix} A & B_i \\ M_i & P_i \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{P}_{11}^i & \mathbf{P}_{12}^i \\ \mathbf{P}_{21}^i & \mathbf{P}_{22}^i \end{bmatrix},$$

where $\mathbf{P}^i_{11}$ and $\mathbf{P}^i_{22}$ are of size $n \times n$ and $m_i \times m_i$, respectively. Using the Sherman-Woodbury-Morrison formula [11] for

$$\mathbf{A}_i(\lambda_i) = \begin{bmatrix} A & B_i \\ M_i & P_i \end{bmatrix} + \begin{bmatrix} -I \\ 0 \end{bmatrix} (\lambda_i I) \begin{bmatrix} I & 0 \end{bmatrix}$$

gives

$$\mathbf{A}_i(\lambda_i)^{-1} = \begin{bmatrix} (I_n - \lambda_i \mathbf{P}^i_{11})^{-1}\mathbf{P}^i_{11} & (I_n - \lambda_i \mathbf{P}^i_{11})^{-1}\mathbf{P}^i_{12} \\ \mathbf{P}^i_{21}(I_n - \lambda_i \mathbf{P}^i_{11})^{-1} & \mathbf{P}^i_{22} + \mathbf{P}^i_{21}\lambda_i(I_n - \lambda_i \mathbf{P}^i_{11})^{-1}\mathbf{P}^i_{12} \end{bmatrix}.$$

As a consequence, there is only need to compute the inverse of the smaller matrix $(I_n - \lambda_i \mathbf{P}^i_{11})$ to get the entries in (13).

*Remark* 2. Our algorithm can be extended to include nonlinear constraints on $v_i$. We just add those to program (6). Note also that the algorithm will return the standard nominal modal set $\lambda^0 = \{\lambda^0_1, \ldots, \lambda^0_p\}$ if we choose $\delta_i = 0$, $i = 1, \ldots, p$, so we present a genuine extension of the traditional assignment procedure. □

## 5. System norms and their subdifferential in closed-loop

To solve program (6) algorithmically, we have to compute function values and subgradients of the cost function $f(\mathbf{x}) := \|T_{w\to z}(K(\mathbf{x}))\|^2$, where $\|\cdot\|$ is the $H_\infty$-norm $\|\cdot\|_\infty$, the $H_2$-norm $\|\cdot\|_2$ or the Hankel norm $\|\cdot\|_H$, and where $\mathbf{x}$ represents the decision variables. Here $\mathbf{x}$ regroups $\lambda_i$ if $m_i \geqslant m$, and $(\lambda_i, t_i)$ if $m_i < m$, $i = 1, \ldots, p$. The gradients given in (10), respectively (12), are generally complex gradients. Algorithmic implementation requires passing from complex to real gradients. This is done using Wirtinger formulas [12, Section 2.3]. For a complex variable $z$, we have that

$$\begin{array}{rcccl} \partial K/\partial \operatorname{Re} z & = & \partial K/\partial z + \partial K/\partial \bar{z} & = & 2\operatorname{Re}(\partial K/\partial z), \\ \partial K/\partial \operatorname{Im} z & = & \jmath(\partial K/\partial z - \partial K/\partial \bar{z}) & = & -2\operatorname{Im}(\partial K/\partial z). \end{array}$$

For simplicity of the notation, it is assumed from now on that $\mathbf{x}$ is a real $q$-dimensional vector regrouping real and imaginary parts of all free parameters $(\lambda_i, t_i)$. Partial derivatives with respect to $\mathbf{x}$ will be denoted $K_i(\mathbf{x}) := \partial K(\mathbf{x})/\partial \mathbf{x}_i$ in the sequel of the paper. In consequence it now remains to compute Clarke subgradients of $\|T_{w\to z}(K)\|^2$ with respect to $K$. By the generalized chain rule [5], this requires subgradients of the norm in question, and the derivative of the transfer function $T_{w\to z}(K)$ with respect to $K$.

Concerning the closed-loop, and to prepare the following, by setting

$$\begin{array}{ll} A_{cl} = A + BKC, & B_{cl} = B_1 + BKD_{21}, \\ C_{cl} = C_1 + D_{12}KC, & D_{cl} = D_{11} + D_{12}KD_{21}, \end{array}$$

the controllability Gramian $X$ and the observability Gramian $Y$ can be obtained from the Lyapunov equations [31]

$$A_{cl}X + XA_{cl}^\top + B_{cl}B_{cl}^\top = 0, \tag{14}$$

$$A_{cl}^\top Y + YA_{cl} + C_{cl}^\top C_{cl} = 0. \tag{15}$$

**5.1. The $H_\infty$-norm.** Consider a stable LTI system

$$G : \left\{ \begin{array}{l} \dot{x} = Ax + Bw \\ z = Cx + Dw \end{array} \right.$$

with state $x \in \mathbb{R}^n$, input $w \in \mathbb{R}^m$, and output $z \in \mathbb{R}^p$. It is well-known that the $H_\infty$-norm of $G$ is defined as

$$\|G\|_\infty = \sup_{\omega \in \mathbb{R}} \sigma_{\max}(G(j\omega)) = \sup_{\omega \in \mathbb{R}} \sqrt{\lambda_{\max}(G(j\omega)^H G(j\omega))},$$

where $\sigma_{\max}$ denotes the maximum singular value of a matrix, and $\lambda_{\max}$ denotes the maximum eigenvalue of a matrix. We now replace $G$ by $T_{w\to z}(K)$ and rewrite

$$f(K) = \|T_{w\to z}(K)\|_\infty^2 = \sup_{\omega \in \mathbb{R}} f(K, \omega),$$

with $f(K,\omega) := \lambda_{\max}\left(T_{w\to z}(K, \jmath\omega)^H T_{w\to z}(K, \jmath\omega)\right)$. Using the notation

$$\begin{bmatrix} T_{w\to z}(K,s) & G_{12}(K,s) \\ G_{21}(K,s) & * \end{bmatrix} = \begin{bmatrix} C_{cl} \\ C \end{bmatrix} (sI - A_{cl})^{-1} \begin{bmatrix} B_{cl} & B \end{bmatrix} + \begin{bmatrix} D_{cl} & D_{12} \\ D_{21} & * \end{bmatrix},$$

and following [4, Lemma 1], closed-loop stability implies that either $f(K) = f(K,\omega)$ for all $\omega$ or $f(K) = f(K,\omega)$ for a finite number of active frequencies $\omega_1, \ldots, \omega_q$. From [2, Section IV] we now obtain the Clarke subgradients of $f$ at $K$ as

$$\Phi_U = 2 \sum_{k=1}^{q} \mathrm{Re}\left(G_{21}(K, \jmath\omega_k) T_{w\to z}(K, \jmath\omega_k)^H R_k U_k R_k^H G_{12}(K, \omega_k)\right)^\top,$$

where $R_k$ is a matrix whose columns form an orthonormal basis of the eigenspace of dimension $r_k \in \mathbb{N}$ associated with $\lambda_{\max}\left(T_{w\to z}(K, \jmath\omega_k)^H T_{w\to z}(K, \jmath\omega_k)\right)$, and where $U_k \in \mathbb{S}_{r_k}$, $U_k \succeq 0$, $\sum_{k=1}^q \mathrm{Tr}(U_k) = 1$. The symbol $\mathbb{S}_m$ stands for the space of $m \times m$ symmetric or Hermitian matrices, and $\mathrm{Tr}(M)$ denotes the trace of $M$. By the application of the chain rule in [5], we deduce that the Clarke subdifferential of $f$ at $\mathbf{x}$ is the set

$$\partial f(\mathbf{x}) = \left\{ \left(\mathrm{Tr}(K_1(\mathbf{x})^\top \Phi_U), \ldots, \mathrm{Tr}(K_q(\mathbf{x})^\top \Phi_U)\right)^\top : \Phi_U \in \partial f(K) \right\}.$$

**5.2. The $H_2$-norm.** The $H_2$-norm of a system $G$ of the form

$$G : \left\{ \begin{array}{l} \dot{x} = Ax + Bw \\ z = Cx \end{array} \right. \tag{16}$$

is defined as

$$\|G\|_2 = \left( \frac{1}{2\pi} \int_{-\infty}^{+\infty} \mathrm{Tr}(G(j\omega)^H G(j\omega)) d\omega \right)^{1/2}.$$

Suppose $D_{cl}$ does not explicitly depend on $K$, which is e.g. the case for $D_{12} = 0$ or $D_{21} = 0$. Then it is reasonable to assess the closed-loop system via the $H_2$-norm of $(A_{cl}, B_{cl}, C_{cl}, 0)$. We have

$$f(K) = \|T_{w\to z}(K)\|_2^2 = \mathrm{Tr}(B_{cl}^\top Y B_{cl}) = \mathrm{Tr}(C_{cl} X C_{cl}^\top).$$

Using (14) and (15), it follows from [25, Theorem 3.2] that $f$ is differentiable at each closed-loop stabilizing $K$, and

$$\nabla f(K) = 2\left(B^\top Y + D_{12}^\top C_{cl}\right) X C^\top + 2 B^\top Y B_{cl} D_{21}^\top.$$

Therefore,

$$\nabla f(\mathbf{x}) = \left(\mathrm{Tr}(K_1(\mathbf{x})^\top \nabla f(K)), \ldots, \mathrm{Tr}(K_q(\mathbf{x})^\top \nabla f(K))\right)^\top$$

for all $\mathbf{x}$ for which $K(\mathbf{x})$ is closed-loop stabilizing.

**5.3. The Hankel norm.** For a stable system $G$ of the form (16), we think of $w(t)$ as an excitation at the input which acts over the time period $0 \leqslant t \leqslant T$. Then the ring of the system $G$ after the excitation has stopped at time $T$ is $z(t)$ for $t > T$. If signals are measured in the energy norm, this leads to the Hankel norm of $G$ defined as

$$\|G\|_H = \sup_{T>0} \left\{ \left( \int_T^\infty z^\top z dt \right)^{1/2} : z = Gw, \int_0^T w^\top w dt \leqslant 1, w(t) = 0 \text{ for } t > T \right\}.$$

The Hankel norm [9, 6] can be understood as measuring the tendency of a system to store energy, which is later retrieved to produce undesired noise effects known as system ring. Minimizing the Hankel norm $\|T_{w\to z}(K)\|_H$ therefore reduces ringing in the closed-loop channel $w \to z$.

If we assume as above that $D_{cl}$ does not explicitly depend on $K$, it is reasonable to assess the channel $w \to z$ via the objective

$$f(K) = \|T_{w\to z}(K)\|_H^2 = \lambda_{\max}(XY),$$

where $X$ and $Y$ are the closed-loop Gramians (14) and (15); see also [6, Lemma 1]. Due to positive semidefiniteness of $B_{cl}B_{cl}^\top$ and $C_{cl}^\top C_{cl}$, closed-loop stability assures positive semidefiniteness of $X$ and $Y$ in (14) and (15). Therefore, although the product XY need not be symmetric, we have

$$\lambda_{\max}(XY) = \lambda_{\max}(X^{\frac{1}{2}} Y X^{\frac{1}{2}}) = \lambda_{\max}(Y^{\frac{1}{2}} X Y^{\frac{1}{2}}),$$

which brings us back to the realm of eigenvalue theory for symmetric matrices. Let $Z := X^{\frac{1}{2}} Y X^{\frac{1}{2}}$ and take $R$ to be a matrix whose columns form an orthonormal basis of the eigenspace of $Z$ of dimension $r \in \mathbb{N}$ associated with $\lambda_{\max}(Z)$. We write $M_i(\mathbf{x}) := \partial M(\mathbf{x})/\partial \mathbf{x}_i$ as before, and $M_i^{\frac{1}{2}}$ short for $(M^{\frac{1}{2}})_i$, $i = 1, \ldots, q$. Then according to [6, Proposition 1], the Clarke subdifferential of $f$ at $\mathbf{x}$ is

$$\partial f(\mathbf{x}) = \{(\mathrm{Tr}(RUR^\top Z_1(\mathbf{x})), \ldots, \mathrm{Tr}(RUR^\top Z_q(\mathbf{x})))^\top : U \in \mathbb{S}_r, U \succeq 0, \mathrm{Tr}(U) = 1\},$$

with

$$Z_i(\mathbf{x}) = X_i^{\frac{1}{2}}(\mathbf{x}) Y X^{\frac{1}{2}} + X^{\frac{1}{2}} Y_i(\mathbf{x}) X^{\frac{1}{2}} + X^{\frac{1}{2}} Y X_i^{\frac{1}{2}}(\mathbf{x}). \tag{17}$$

Here $X_i(\mathbf{x}), Y_i(\mathbf{x})$ and $X_i^{\frac{1}{2}}(\mathbf{x})$ are the solutions of the following Lyapunov equations

$$\begin{aligned} A_{cl}X_i(\mathbf{x}) + X_i(\mathbf{x})A_{cl}^\top = &-BK_i(\mathbf{x})CX - X(BK_i(\mathbf{x})C)^\top \\ &- BK_i(\mathbf{x})D_{21}B_{cl}^\top - B_{cl}(BK_i(\mathbf{x})D_{21})^\top, \end{aligned} \tag{18}$$

$$\begin{aligned} A_{cl}^\top Y_i(\mathbf{x}) + Y_i(\mathbf{x})A_{cl} = &-(BK_i(\mathbf{x})C)^\top Y - YBK_i(\mathbf{x})C \\ &- (D_{12}K_i(\mathbf{x})C)^\top C_{cl} - C_{cl}^\top D_{12}K_i(\mathbf{x})C, \end{aligned} \tag{19}$$

$$X^{\frac{1}{2}} X_i^{\frac{1}{2}}(\mathbf{x}) + X_i^{\frac{1}{2}}(\mathbf{x}) X^{\frac{1}{2}} = X_i(\mathbf{x}). \tag{20}$$

## 6. Nonsmooth solver

Step 4 of our main Algorithm 1 requires a subroutine to solve (6). Here we use a nonsmooth descent algorithm, presented as Algorithm 2, which we now discuss briefly. To extend the scope, we consider a constrained optimization programs of the more abstract form

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & h(\mathbf{x}) \leqslant 0 \\ & A\mathbf{x} \leqslant b \end{array} \tag{21}$$

where $\mathbf{x} \in \mathbb{R}^q$ is the decision variable, and $f$ and $h$ are potentially nonsmooth and nonconvex. This covers program (6), where $f(\mathbf{x}) = \|T_{w\to z}\,(K(\mathbf{x}))\,\|^2$ for one of the norms discussed in Section 5, while the constraint $h(\mathbf{x}) \leqslant 0$ represents (3) after eliminating $v, w$ via (7), respectively, (9). The polydisk constraints $|\operatorname{Re}\lambda_i - \operatorname{Re}\lambda_i^0| \leqslant \delta_i$, $|\operatorname{Im}\lambda_i - \operatorname{Im}\lambda_i^0| \leqslant \delta_i$ in (6) can easily be converted to the form $A\mathbf{x} \leqslant b$. According to the cases discussed in Section 4, the decision variable $\mathbf{x}$ regroups either the $\lambda_i$, or the $(\lambda_i, t_i)$ as in (9). The cast (8) is also covered by (21).

To solve (21) we use a progress function at the current iterate $\mathbf{x}$,

$$F(\cdot, \mathbf{x}) = \max\{f(\cdot) - f(\mathbf{x}) - \nu h(\mathbf{x})_+, h(\cdot) - h(\mathbf{x})_+\},$$

for some fixed parameter $\nu > 0$, which is successively minimized subject to the linear constraints. Antecedents of this idea can for instance be found in Polak [23, Section 2.2.2] in the smooth case, or Polak and Wardi [24] in a nonsmooth setting, and in our own contributions [3, 8, 6], where more details and convergence proofs can be found.

Convergence theory of Algorithm 2 is discussed in [8, 6]. The following result is slightly more general than the main convergence theorem in [8] or [6], but can be obtained based essentially on the same convergence analysis:

**Theorem 6.1.** *Suppose $f$ and $h$ in program (21) are lower-$C^1$ functions in the sense of [29] such that the following conditions hold:*

- (i) *$f$ is weakly coercive on the constraint set $\Omega = \{\mathbf{x} \in \mathbb{R}^q : h(\mathbf{x}) \leqslant 0, A\mathbf{x} \leqslant b\}$, i.e., if $\mathbf{x}^j \in \Omega$ and $\|\mathbf{x}^j\| \to \infty$, then $f(\mathbf{x}^j)$ is not monotonically decreasing.*
- (ii) *$h$ is weakly coercive on $P = \{\mathbf{x} \in \mathbb{R}^q : A\mathbf{x} \leqslant b\}$, i.e., if $\mathbf{x}^j \in P$ and $\|\mathbf{x}^j\| \to \infty$, then $h(\mathbf{x}^j)$ is not monotonically decreasing.*

*Then the sequence of serious iterates $\mathbf{x}^j \in P$ generated by Algorithm 2 is bounded, and every accumulation point $\mathbf{x}^*$ of the $\mathbf{x}^j$ satisfies $\mathbf{x}^* \in P$ and $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*) + A^\top \eta^*$ for some multiplier $\eta^* \geqslant 0$ with $\eta^{*\top}(A\mathbf{x}^* - b) = 0$. In other words, $\mathbf{x}^*$ is either a critical point of constraint violation, or a Karush-Kuhn-Tucker point of program (21).* □

Note that the functions $f, h$ used in (6) are indeed lower-$C^1$ functions, see [8, 6], so our convergence theory applies. Convergence for even larger classes of nonsmooth functions is discussed in [21, 19]. For additional insight into this type of nonconvex bundle method see [3, 21, 19, 20].

**Algorithm 2.** Nonsmooth optimization subroutine

**Parameters:** $0 < \gamma < \widetilde{\gamma} < 1, 0 < \gamma < \Gamma < 1, 0 < q < \infty$, $q < T < \infty$.

▷ **Step 1** (`Initialize outer loop`). Choose initial iterate $\mathbf{x}^1$ with $A\mathbf{x}^1 \leqslant b$ and matrix $Q_1 = Q_1^\top$ with $-qI \preceq Q_1 \preceq qI$. Initialize memory control parameter $\tau_1^\sharp > 0$ such that $Q_1 + \tau_1^\sharp I \succ 0$. Put outer loop counter $j = 1$.

◇ **Step 2** (`Stopping test`). At outer loop counter $j$, stop if $\mathbf{x}^j$ is a KKT-point or a critical point of constraint violation. Otherwise, goto inner loop.

▷ **Step 3** (`Initialize inner loop`). Put inner loop counter $k = 1$ and initialize $\tau_1 = \tau_j^\sharp$. Build initial working model

$$\Phi_1(\cdot, \mathbf{x}^j) = g_{0j}^\top(\cdot - \mathbf{x}^j) + \tfrac{1}{2}(\cdot - \mathbf{x}^j)^\top Q_j(\cdot - \mathbf{x}^j),$$

where $g_{0j} \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$.

▷ **Step 4** (`Trial step generation`). At inner loop counter $k$ find solution $\mathbf{y}^k$ of the tangent program

$$\begin{array}{ll} \text{minimize} & \Phi_k(\mathbf{y}, \mathbf{x}^j) + \frac{\tau_k}{2}\|\mathbf{y} - \mathbf{x}^j\|^2 \\ \text{subject to} & A\mathbf{y} \leqslant b, \mathbf{y} \in \mathbb{R}^n. \end{array}$$

◇ **Step 5** (`Acceptance test`). If

$$\rho_k = \frac{F(\mathbf{y}^k, \mathbf{x}^j)}{\Phi_k(\mathbf{y}^k, \mathbf{x}^j)} \geqslant \gamma,$$

put $\mathbf{x}^{j+1} = \mathbf{y}^k$ (serious step), quit inner loop and goto step 8. Otherwise (null step), continue inner loop with step 6.

▷ **Step 6** (`Update working model`). Generate a cutting plane $m_k(\cdot, \mathbf{x}^j) = a_k + g_k^\top(\cdot - \mathbf{x}^j)$ at null step $\mathbf{y}^k$ and counter $k$ using downshifted tangents. Compute aggregate plane $m_k^*(\cdot, \mathbf{x}^j) = a_k^* + g_k^{*\top}(\cdot - \mathbf{x}^j)$ at $\mathbf{y}^k$, and then build new working model $\Phi_{k+1}(\cdot, \mathbf{x}^j)$ by including cutting plane and aggregate plane.

◇ **Step 7** (`Update proximity control parameter`). Compute secondary control parameter

$$\widetilde{\rho}_k = \frac{\Phi_{k+1}(\mathbf{y}^k, \mathbf{x}^j)}{\Phi_k(\mathbf{y}^k, \mathbf{x}^j)}$$

and put

$$\tau_{k+1} = \begin{cases} \tau_k & \text{if } \widetilde{\rho}_k < \widetilde{\gamma}, \\ 2\tau_k & \text{if } \widetilde{\rho}_k \geqslant \widetilde{\gamma}. \end{cases}$$

Increase inner loop counter $k$ and loop back to step 4.

◇ **Step 8** (`Update` $Q_j$ `and memory element`). Update matrix $Q_j \to Q_{j+1}$ respecting $Q_{j+1} = Q_{j+1}^\top$ and $-qI \preceq Q_{j+1} \preceq qI$. Then store new memory element

$$\tau_{j+1}^\sharp = \begin{cases} \tau_k & \text{if } \rho_k < \Gamma, \\ \frac{1}{2}\tau_k & \text{if } \rho_k \geqslant \Gamma. \end{cases}$$

Increase $\tau_{j+1}^\sharp$ if necessary to ensure $Q_{j+1} + \tau_{j+1}^\sharp I \succ 0$. If $\tau_{j+1}^\sharp > T$ then re-set $\tau_{j+1}^\sharp = T$. Increase outer loop counter $j$ and loop back to step 2.

## 7. Control of a launcher in atmospheric flight

We consider attitude control of a satellite launcher in atmospheric flight. The linear model

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

is specified as

$$A = \begin{bmatrix} Z_w & Z_q + U_0 & Z_\theta & Z_v & 0 & Z_\psi & Z_p & Z_\phi \\ M_w & M_q & 0 & 0 & M_r & 0 & M_p & 0 \\ 0 & T_q & 0 & 0 & T_r & 0 & 0 & 0 \\ Y_w & 0 & Y_\theta & Y_v & Y_r & Y_\psi & Y_p & Y_\phi \\ 0 & N_q & 0 & N_v & N_r & 0 & N_p & 0 \\ 0 & P_q & 0 & 0 & P_r & 0 & 0 & 0 \\ 0 & L_q & 0 & 0 & L_r & 0 & L_p & 0 \\ 0 & F_q & 0 & 0 & F_r & 0 & 1 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} Z_{\beta z} & M_{\beta z} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Y_{\beta y} & N_{\beta y} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & L_{\beta r} & 0 \end{bmatrix}^\top.$$

The states and controls are defined in Tables 1 and 2, while the vector of measurements is $y = [q\ \theta\ r\ \psi\ p\ \phi]^\top \in \mathbb{R}^6$. The model has been obtained from

TABLE 1. States definitions

| name | meaning |
|---|---|
| $w$ | vertical velocity (m/s) |
| $q$ | pitch rate (deg/s) |
| $\theta$ | pitch angle (deg) |
| $v$ | lateral velocity (m/s) |
| $r$ | yaw rate (deg/s) |
| $\psi$ | yaw angle (deg) |
| $p$ | roll rate (deg/s) |
| $\phi$ | roll angle (deg) |

TABLE 2. Controls definitions

| name | meaning |
|---|---|
| $\beta_z$ | deflection of pitch nozzle actuator (deg) |
| $\beta_y$ | deflection of yaw nozzle actuator (deg) |
| $\beta_r$ | deflection of roll nozzle actuator (deg) |

linearization of the nonlinear equations [16] about a steady state flight point

$$\begin{array}{lll} U_0 = 88.11 \text{ m/s}, & v_0 = 0.678 \text{ m/s}, & w_0 = -1.965 \text{ m/s}, \\ p_0 = -0.0006 \text{ rad/s}, & q_0 = 0.0026 \text{ rad/s}, & r_0 = 0.0046 \text{ rad/s}, \\ \theta_0 = 8.38°, & \psi_0 = 3.48°, & \phi_0 = 11.99°, \end{array}$$

the procedure being explained in [10]. Numerical data in $A$, $B$ are gathered in Table 3.

TABLE 3. Numerical coefficients at steady state flight point

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $Z_w$ | -0.0162 | $M_w$ | 0.0022 | $Y_w$ | -6e-4 | $N_q$ | 5e-4 |
| $Z_q$ | 87.9 - 88.11 | $M_q$ | 0.0148 | $Y_\theta$ | -2.11 | $N_v$ | $-M_w$ |
| $Z_\theta$ | -9.48 | $M_r$ | -0.0005 | $Y_v$ | $Z_w$ | $N_r$ | 0.0151 |
| $Z_v$ | 0.0006 | $M_p$ | 0.0042 | $Y_r$ | -87.9 | $N_p$ | -0.0024 |
| $Z_\psi$ | -2.013 | $T_q$ | 0.98 | $Y_\psi$ | 9.47 | $P_q$ | 0.2078 |
| $Z_p$ | -0.687 | $T_r$ | -0.2084 | $Y_p$ | -1.965 | $P_r$ | 0.9782 |
| $Z_\phi$ | 0.399 | $L_q$ | 0 | $Y_\phi$ | 1.3272 | $F_q$ | 0.0704 |
| $L_r$ | 0 | $L_p$ | -0.0289 | $L_{\beta r}$ | 25.89 | $F_r$ | -0.015 |
| $Z_{\beta z}$ | 10.87 | $M_{\beta z}$ | 4.08 | $Y_{\beta y}$ | -10.87 | $N_{\beta y}$ | 4.08 |

**7.1. Control law specifications.** The control law specifications include

- Decoupling of the 3 axes $(\theta, q)$, $(\psi, r)$, and $(\phi, p)$.
- Well-damped responses to set-points in $\theta$, $\psi$, and $\phi$, the selector outputs.
- Settling times around 2.5 seconds.

We use a set-point tracking control architecture with MIMO PI feedback as shown in Figure 1. Tunable matrix gains are therefore $K_P$ and $K_I$.

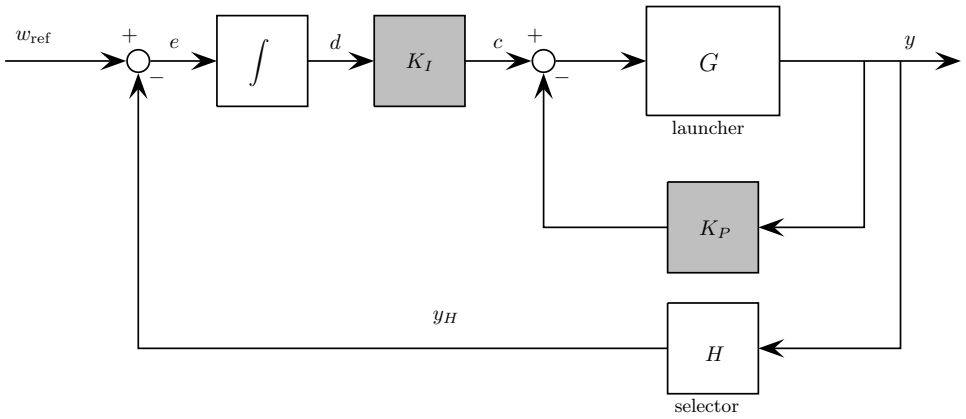



FIGURE 1. Launcher control architecture with MIMO PI-controller

Tracking performance is incorporated into program (6) by minimizing the tracking error transfer function $T_{w_{\text{ref}} \to e}(K)$.

Pole placement with integral action is easily formulated using the augmented state-space matrices

$$A_a = \begin{bmatrix} A & 0 \\ -HC & 0 \end{bmatrix}, B_a = \begin{bmatrix} B \\ 0 \end{bmatrix}, C_a = \begin{bmatrix} C & 0 \\ 0 & I_3 \end{bmatrix}.$$

The control law is structured conformably upon defining

$$W = [w_1 \dots w_9], \; V = [v_1 \dots v_9], \qquad [A_a - \lambda_i I_{11} | B_a] \begin{bmatrix} v_i \\ w_i \end{bmatrix} = 0.$$

$$K_a = W(C_a V)^{-1}, \quad K_a = [-K_P \;\; K_I]\,.$$

**7.2. Study 1.** In a first study we compare traditional and optimized partial pole placement without shaping of eigenvectors. We start by choosing reference values $\xi, \omega$ to achieve appropriate second-order system responses. We have chosen the desired damping $\xi = \frac{\sqrt{2}}{2}$, and natural frequencies

$$\omega_1 = 2.1, \omega_2 = 2.2, \omega_3 = 1.8,$$

which leads to the nominal modal set $\Lambda = \{\lambda_1^0, \ldots, \lambda_9^0\}$, with

$$\begin{aligned}
&\lambda_1^0 = -\omega_1 \left(\xi + \jmath\sqrt{1-\xi^2}\right), \lambda_2^0 = -\omega_1 \left(\xi - \jmath\sqrt{1-\xi^2}\right),\\
&\lambda_3^0 = -\omega_2 \left(\xi + \jmath\sqrt{1-\xi^2}\right), \lambda_4^0 = -\omega_2 \left(\xi - \jmath\sqrt{1-\xi^2}\right),\\
&\lambda_5^0 = -\omega_3 \left(\xi + \jmath\sqrt{1-\xi^2}\right), \lambda_6^0 = -\omega_3 \left(\xi - \jmath\sqrt{1-\xi^2}\right),\\
&\lambda_7^0 = -3.5,\ \lambda_8^0 = -4,\ \lambda_9^0 = -4.5.
\end{aligned}$$

Classical pole placement now leads to the initial controller $K^0$ in Algorithm 1. To find the optimal controller $K^*$, we follow Algorithm 1 and minimize the tracking error $w_{\text{ref}} \to e$ subject to the pole placement constraint in (6) via Algorithm 2, which returns the optimal controller $K^*$.

TABLE 4. Launcher study 1. Cost for initial $K^0$ and optimal $K^*$ controllers

| | Hankel | $H_\infty$ | $H_2$ |
|---|---|---|---|
| $K^0$ | 66.7208 | 2.3714 | 45.3537 |
| $K^*$ | 0.7135 | 1.4058 | 3.0845 |

We have run program (6) with three different norms, the Hankel norm, the $H_\infty$-, and the $H_2$-norm. The improvements in the cost function can be seen in Table 4. The wandering of the poles during optimization shown in Figure 3 corresponds to the case of the Hankel norm. Figure 2 shows that decoupling is substantially improved in all three cases. Note the sluggish responses for the initial controller are due to unassigned modes of classical eigenstructure assignment. This is in contrast with the proposed approach in which modes that are left unspecified are indirectly assigned to achieve additional performance requirements.

**7.3. Study 2.** In our second study we compare standard and optimized eigenstructure assignment. We achieve preliminary decoupling of the modes by choosing structural constraints on eigenvectors $v_i$. These constraints comply with decoupling requirements of the launcher motion. The eigenvectors $v_1$ and its complex conjugate $v_2$ are chosen to have zero entries in the rows corresponding to $\psi$ and $\phi$. The eigenvector $v_3$ and its conjugate $v_4$ have entries 0 relative to $\theta$ and $\phi$. The eigenvectors $v_5$ and its conjugate $v_6$ have zero entries in the rows associated with $\theta$ and $\psi$. For the real modes, the eigenvectors are chosen as

$$\begin{aligned}
v_7 &= [* \;\; * \;\; 1 \;\; * \;\; * \; 0 \;\; * \;\; 0 \;\; * \;\; * \;\; *]^\top,\\
v_8 &= [* \;\; * \;\; 0 \;\; * \;\; * \; 1 \;\; * \;\; 0 \;\; * \;\; * \;\; *]^\top,\\
v_9 &= [* \;\; * \;\; 0 \;\; * \;\; * \; 0 \;\; * \;\; 1 \;\; * \;\; * \;\; *]^\top.
\end{aligned}$$

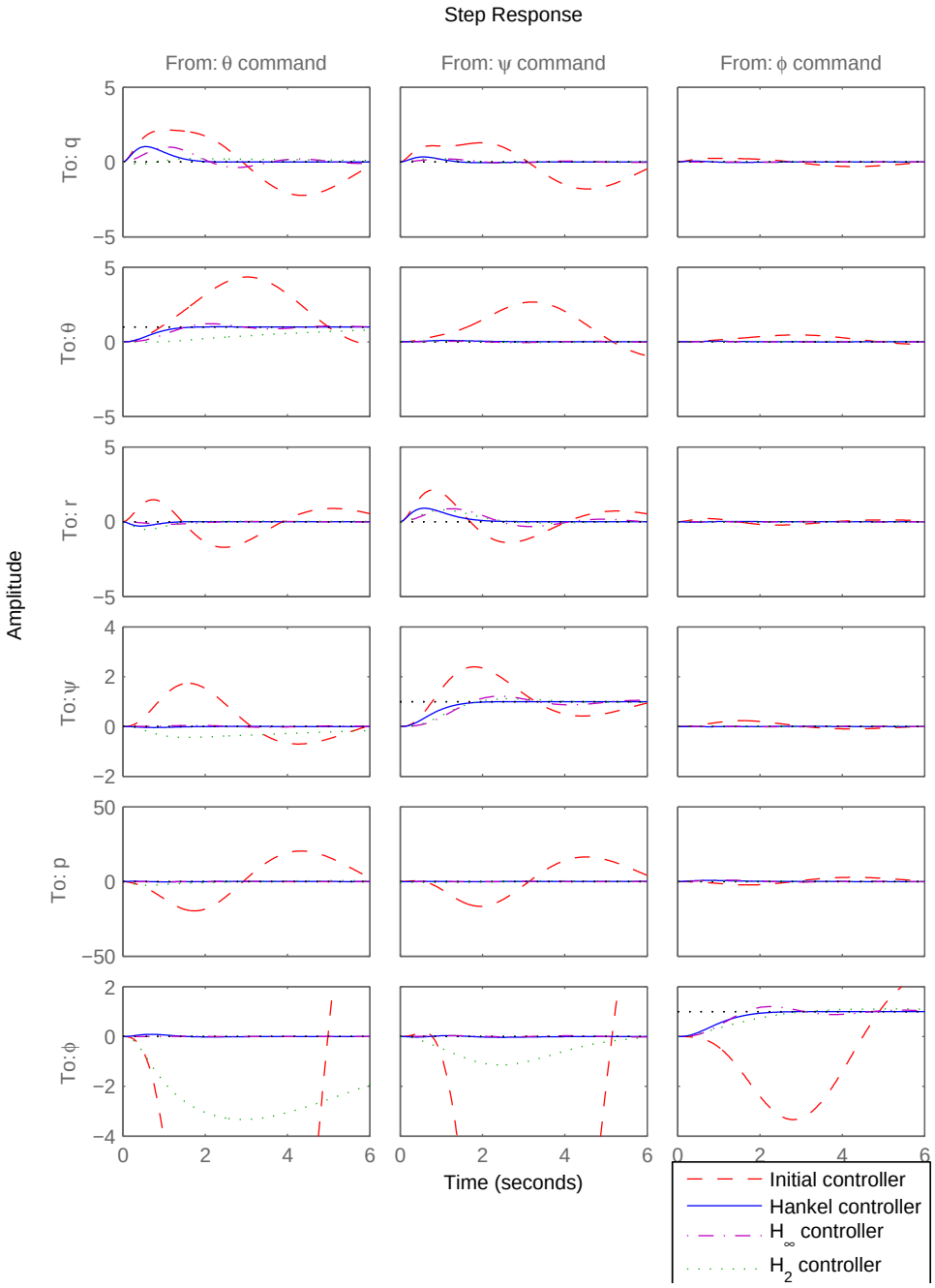



FIGURE 2. Control of a launcher, study 1. Initial and final controllers obtained respectively by standard and optimized eigenstructure assignment in the case where eigenvectors are not structured ($m_i = 0$). Decoupling is improved for each norm

These structural constraints define the matrices $M_i, N_i$ of (3) in each case. We have again tested the Hankel, $H_\infty$ and $H_2$-norms in the objective $f$ of (6).

The optimal controller $K^*$ computed by Algorithm 1 for the Hankel norm gives the value $\|T(P_{\text{perf}}, K^*)\|_H = 0.7360$, while the initial controller $K^0$ leads to $\|T(P_{\text{perf}}, K^0)\|_H = 0.7787$. Similar improvements are obtained for the other norms. The step responses are shown in Figure 4.

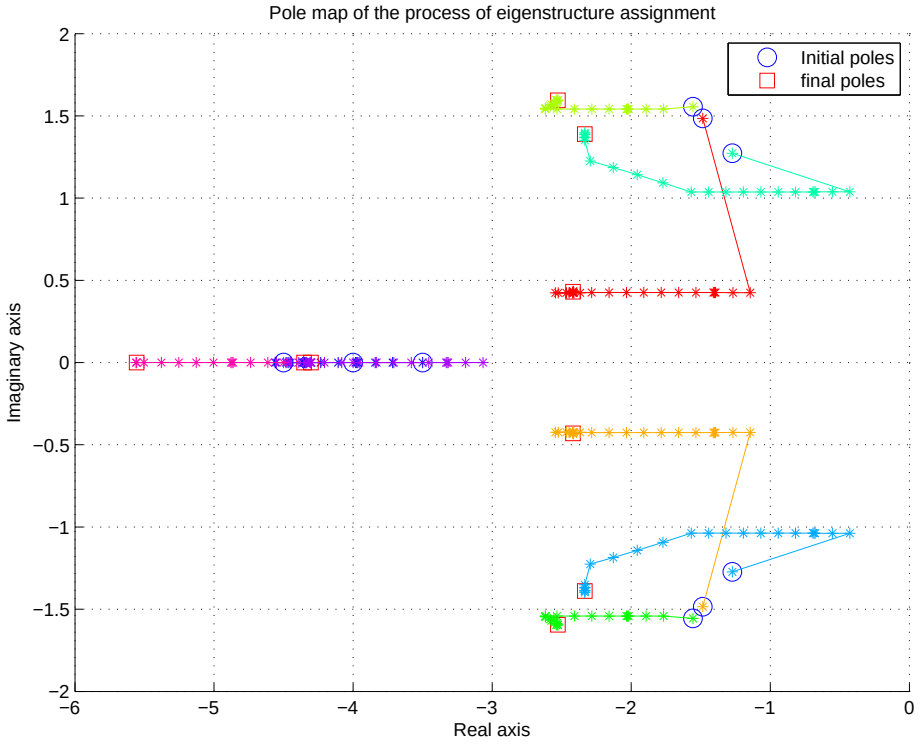



FIGURE 3. Control of launcher, study 1. Itineraries of closed-loop poles in optimized eigenstructure assignment based on the Hankel program (6)

In conclusion, the launcher application shows that decoupling can be significantly enhanced through optimization even without shaping of the $v_i$ (study 1) if the performance channel $T_{w_{\text{ref}} \to e}$ is used within optimization program (6). The second study shows that even when 0's are assigned to specific $v_{ik}$'s, the use of optimization is still useful, as it significantly enhances decoupling as demonstrated by simulation.

## 8. Application to autopilot design for a civil aircraft

In this section, we consider the longitudinal dynamics for the robust civil aircraft model (RCAM) at a nominal condition with the aircraft in its standard configuration: aircraft air speed of 80 m/s, aircraft altitude of 305 m (1000 ft), aircraft mass of 120 tons, aircraft centre of gravity at 23% horizontal MAC and 0% vertical MAC, flight path angle of 0° (level) and still air (no wind effects). The linear longitudinal model is given by

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

where states are described in Table 5, the input vector is $u = [\delta_t \ \delta_{th}]^\top$ with $\delta_t$ the tailplane deflection and $\delta_{th}$ the throttle position. The vector of measurements is $y = [q \ n_z \ w_V \ z \ V_c]^\top$, where $n_z$ is vertical acceleration, $w_V$ vertical velocity, and $V_c$

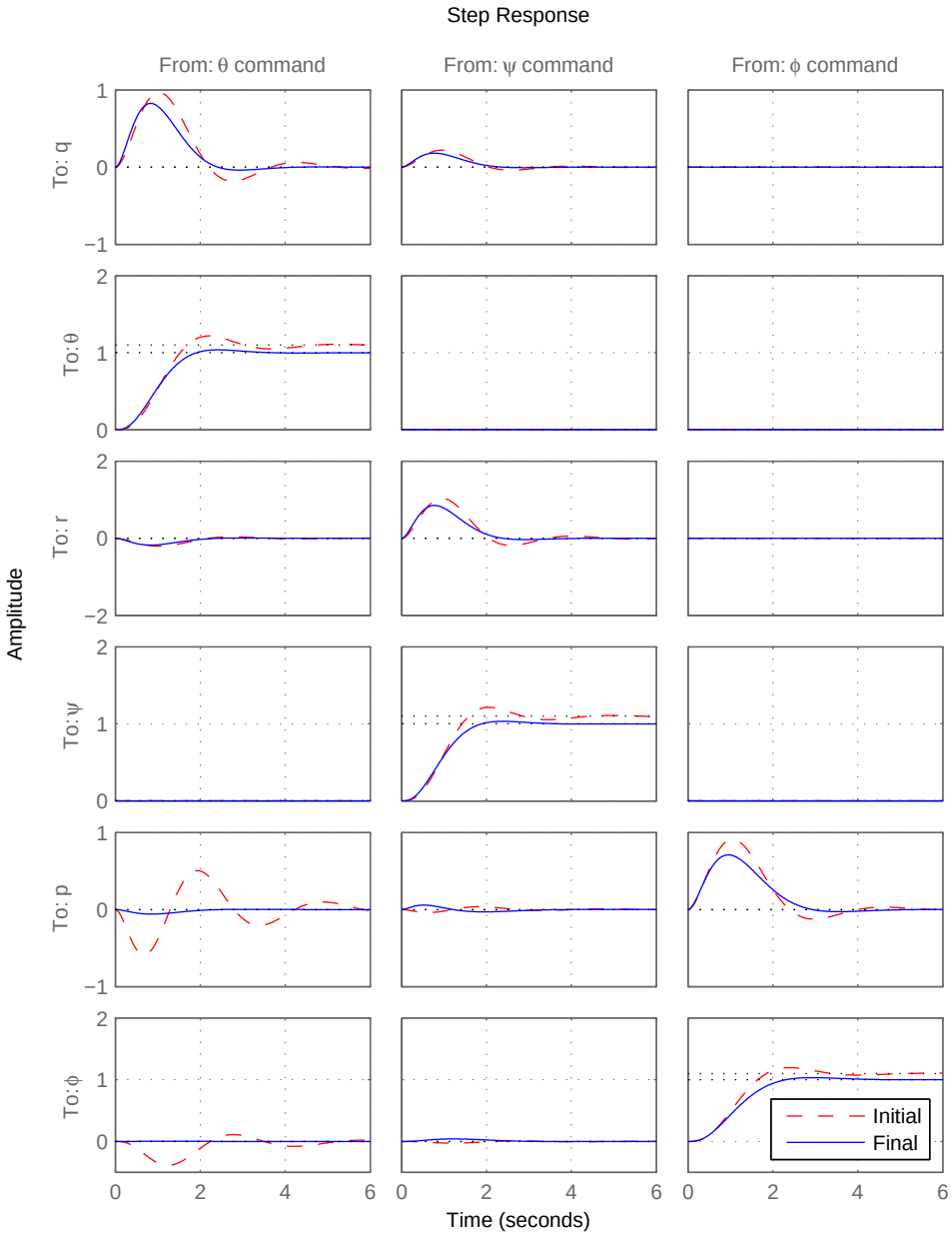



FIGURE 4. Control of launcher, study 2. Initial and final controller obtained respectively by standard and optimized eigenstructure assignment based on Hankel program with $m_i = m$ or $m_i = m - 1$

the air speed. Data borrowed from [7] are given as

$$A = \begin{bmatrix} -0.9825 & 0 & -0.0007 & -0.0161 & 0 & -2.4379 & 0.5825 \\ 1 & 0 & 0 & 0 & & 0 & 0 \\ -2.1927 & -9.7758 & -0.0325 & 0.0743 & 0 & 0.1836 & 19.6200 \\ 77.3571 & -0.7674 & -0.2265 & -0.6683 & 0 & -6.4785 & 0 \\ 0 & -79.8667 & -0.0283 & 0.9996 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -6.6667 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -6.6667 \end{bmatrix},$$

TABLE 5. States of the longitudinal model

| name | meaning |
|---|---|
| $q$ | pitch rate |
| $\theta$ | pitch angle |
| $u_B$ | forward speed |
| $w_B$ | upwards velocity |
| $z$ | altitude |
| $x_t$ | the state corresponding to the first order tailplane model |
| $x_{th}$ | the state corresponding to the first order engine model |

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 6.6667 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6.6667 \end{bmatrix}^\top,$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.2661 & 0 & -0.0231 & -0.0681 & 0 & -0.6604 & 0 \\ 0 & -79.8667 & -0.0283 & 0.9996 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0.9996 & 0.0290 & 0 & 0 & 0 \end{bmatrix}.$$

The controller structure of the longitudinal autopilot with tunable gains $K_I, K_P$ is similar to the launcher structure given in Figure 1. The output is now $y = [q\ n_z\ w_V\ z]^T$, and the selector produces $y_H = [z\ V_c]^T$. We next design a closed-loop controller such that altitude is decoupled from air speed command and conversely. This leads to decoupling altitude and altitude-tracking modes from forward speed $u_B$, and decoupling of the air speed track mode from the upwards velocity $w_B$. Other modes are also decoupled from some states to reduce the mutual influence of the aircraft variables. Accordingly, we take the nominal modes as follows:

$$\begin{aligned} &\lambda^0_{1,2} = -0.8 \pm \jmath 0.8, \\ &\lambda^0_{3,4} = -0.15 \pm \jmath 0.15, \\ &\lambda^0_5 = -0.3,\ \lambda^0_6 = -0.4,\ \lambda^0_7 = -0.5. \end{aligned}$$

The corresponding desired eigenvectors are shaped as

$$\begin{aligned} v_{1,2} &= [* \ * \ 0 \ * \ * \ * \ * \ * \ *]^\top, \\ v_{3,4} &= [* \ * \ * \ 0 \ * \ * \ * \ * \ *]^\top, \\ v_5 &= [* \ * \ 0 \ * \ * \ * \ * \ * \ *]^\top, \\ v_6 &= [* \ * \ * \ 0 \ * \ * \ * \ * \ *]^\top, \\ v_7 &= [* \ * \ 0 \ * \ * \ * \ * \ * \ *]^\top, \end{aligned}$$

which defines the data $M_i, N_i$ and $r_i$ in (3). The optimal controller $K^*$ computed by Algorithm 1 gives $\|T(P_{\text{perf}}, K^*)\|_H = 0.6270$, while the initial controller $K^0$ obtained by standard assignment had $\|T(P_{\text{perf}}, K^0)\|_H = 1.5041$. The closed-loop eigenvalues

returned by the algorithm are

$$\lambda_{1,2} = -0.8 \pm \jmath 0.8,$$
$$\lambda_{3,4} = -0.35 \pm \jmath 0.05,$$
$$\lambda_5 = -0.3,\ \lambda_6 = -0.05,\ \lambda_7 = -0.37,$$

which shows that some of the poles took indeed the opportunity to wander away from their nominal values once they were allowed to do so. Step responses are compared in Figure 5. The interpretation of the results is that optimization is useful to further enhance decoupling even when eigenvectors are already shaped.

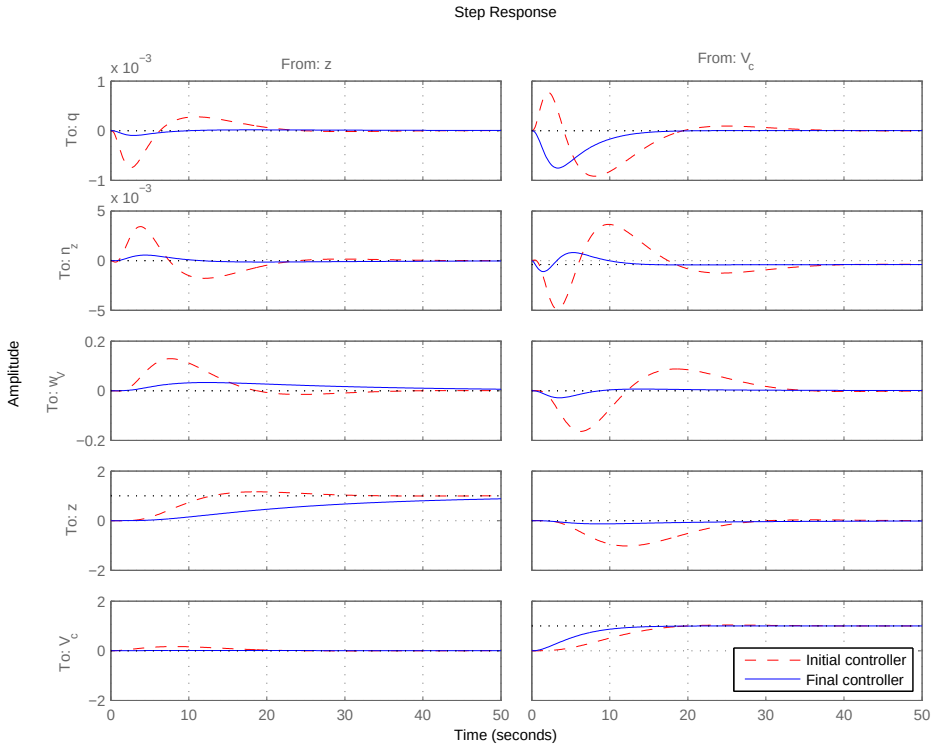



FIGURE 5. Aircraft attitude control. Responses to a step command in altitude and in air speed. Optimal controller computed by optimized eigenstructure assignment ($m_i = 1$, $m = 2$) reduces coupling

## 9. Conclusion

We have presented a new approach to partial eigenstructure assignment in output feedback control in which the eigenelements $(\lambda, v, w)$ are allowed to move *simultaneously* in a neighborhood of their nominal values $(\lambda^0, v^0, w^0)$ obtained by standard partial assignment. The flexibility gained in allowing this is apparent on two fronts. First, stability of unassigned modes is guaranteed, leading to an internally stable closed-loop system. Secondly, criteria such as $H_\infty$, $H_2$ and Hankel norms can be incorporated into our formulation to improve performance and/or robustness of the controlled system. The efficiency of the new approach was demonstrated on two aerospace applications, control of a launcher in atmospheric flight, and attitude control of a civil aircraft.

## Appendix

*Proof of Propositions 4.1 and 4.2.* Let us start by discussing the case $m_i \geqslant m$. Derivatives of $w_i$ with respect to $\lambda_i$ can be derived from the normal equations $F_i(\lambda_i)^H F_i(\lambda_i) w_i = F_i(\lambda_i)^H r_i$ or directly from the expression of $w_i$ in (7). Assuming $F_i(\lambda_i)$ is full-column rank, we rewrite $F_i(\lambda_i)^\dagger = (F_i(\lambda_i)^H F_i(\lambda_i))^{-1} F_i(\lambda_i)^H$. The partial derivative of $w_i$ with respect to $\lambda_i$ is then readily derived by exploiting the facts that for an invertible matrix $\mathbf{M}$ depending smoothly on a parameter $t$, the derivative of its inverse is obtained as

$$\frac{\partial \mathbf{M}^{-1}}{\partial t} = -\mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial t} \mathbf{M}^{-1} .$$

Also, $\partial F_i(\lambda_i)^H / \partial \lambda_i$ is identically zero since $\partial \lambda_i^H / \partial \lambda_i = 0$. This gives

$$\begin{aligned} \frac{\partial w_i}{\partial \lambda_i} &= -(F_i(\lambda_i)^H F_i(\lambda_i))^{-1} F_i(\lambda_i)^H \frac{\partial F_i(\lambda_i)}{\partial \lambda_i} (F_i(\lambda_i)^H F_i(\lambda_i))^{-1} F_i(\lambda_i)^H r_i \\ &= F_i(\lambda_i)^\dagger M_i(\lambda_i I - A)^{-2} B F_i(\lambda_i)^\dagger r_i \, . \end{aligned}$$

The derivative of $v_i$ is obtained in much the same way using the upper part of (7). Finally, collecting the results for $w_i$ and $v_i$ leads to expression (11).

This allows us now to express the terms $\partial K / \partial \lambda_i$ where $K = W(CV)^{-1}$. Using again the derivative of a matrix inverse, we have that

$$\frac{\partial (CV)^{-1}}{\partial \lambda_i} = -(CV)^{-1} C \left[ 0 \cdots 0 \; \frac{\partial v_i}{\partial \lambda_i} \; 0 \cdots 0 \right] (CV)^{-1}.$$

Combining with

$$\frac{\partial W}{\partial \lambda_i} = \left[ 0 \cdots 0 \; \frac{\partial w_i}{\partial \lambda_i} \; 0 \cdots 0 \right]$$

yields (10).

Next consider the under-specified case $m_i < m$. We have that (9) yields (13) analogously to the over-specified case. Finally, formulas for $\partial K / \partial \lambda_i$ and $\partial K / \partial t_{ki}$ in (12) are obtained from the fact that $K = W(CV)^{-1}$ with $V = [v_1 \dots v_p]$ and

$$W = \left[ w_1 \cdots \left| \begin{matrix} u_i \\ t_i \end{matrix} \right| \cdots w_p \right] .$$

□

## References


1. P. Apkarian, *Structured stability robustness improvement by eigenspace techniques: a hybrid methodology*, AIAA Guid., Nav. and Control Conf. **12** (1989), no. 2, 162–168.
2. P. Apkarian and D. Noll, *Nonsmooth $H_\infty$ synthesis*, IEEE Trans. Automat. Control **51** (2006), no. 1, 71–86.
3. P. Apkarian, D. Noll, and A. Rondepierre, *Mixed $H_2/H_\infty$ control via nonsmooth optimization*, SIAM J. Control Optim. **47** (2008), no. 3, 1516–1546.
4. V. Bompart, D. Noll, and P. Apkarian, *Second-order nonsmooth optimization for $H_\infty$ synthesis*, Numer. Math. **107** (2007), no. 3, 433–454.
5. F. H. Clarke, *Generalized gradients of Lipschitz functionals*, Adv. in Math. **40** (1981), no. 1, 52–67.
6. M. N. Dao and D. Noll, *Minimizing memory effects of a system*, Math. Control Signals Syst. (2014), doi: 10.1007/s00498-014-0135-9.

7. Flight Mechanics Action Group 08, *Robust flight. Control design challenge problem formulation and manual: the research civil aircraft model (RCAM)*, Technical Report GARTEUR/TP-088-3, Department of Electrical Engineering, Linkoping University, June 15, 1995, Version 1.
8. M. Gabarrou, D. Alazard, and D. Noll, *Design of a flight control architecture using a non-convex bundle method*, Math. Control Signals Syst. **25** (2013), no. 2, 257–290.
9. K. Glover, *All optimal Hankel-norm approximations of linear multivariable systems and their $L^\infty$-error bounds*, Internat. J. Control **39** (1984), no. 6, 1115–1193.
10. A. L. Greensite, *Elements of modern control theory*, Spartan Books, New York, 1970.
11. N. J. Higham, *Accuracy and stability of numerical algorithms*, second ed., Society for Industrial and Applied Mathematics, Philadelphia, 2002.
12. A. Hjørungnes, *Complex-valued matrix derivatives. With applications in signal processing and communications*, Cambridge University Press, Cambridge, 2011.
13. N. Kshatriya, U. D. Annakkage, F. M. Hughes, and A. M. Gole, *Optimized partial eigenstructure assignment-based design of a combined PSS and active damping controller for a DFIG*, IEEE Trans. Power Syst. **25** (2010), no. 2, 866–876.
14. G. P. Liu and R. J. Patton, *Eigenstructure assignment for control system design*, John Wiley & Sons, Inc., Chichester, 1998.
15. J.-F. Magni, *A toolbox for robust modal control design (RMCT)*, Proc. IEEE Internat. Symposium on Computer-Aided Control System Design (Anchorage), September 2000, pp. 202–207.
16. D. McLean, *Automatic flight control systems*, Prentice Hall, London, 1990.
17. B. C. Moore, *On the flexibility offered by state feedback in multivariable systems beyond closed loop eigenvalue assignment*, IEEE Trans. Automat. Control **AC-21** (1976), 689–692.
18. J. C. Morris, P. Apkarian, and J. C. Doyle, *Synthesizing robust mode shapes with $\mu$ and implicit model following*, Proc. IEEE Conf. on Control Applications (Dayton), September 1992, pp. 1018–1023.
19. D. Noll, *Cutting plane oracles to minimize non-smooth non-convex functions*, Set-Valued Var. Anal. **18** (2010), no. 3-4, 531–568.
20. ———, *Convergence of non-smooth descent methods using the Kurdyka-Łojasiewicz inequality*, J. Optim. Theory Appl. **160** (2014), no. 2, 553–572.
21. D. Noll, O. Prot, and A. Rondepierre, *A proximity control algorithm to minimize nonsmooth and nonconvex functions*, Pac. J. Optim. **4** (2008), no. 3, 571–604.
22. R. J. Patton and G. P. Liu, *Robust control design via eigenstructure assignment, genetic algorithms and gradient-based optimisation*, IEE P-Contr. Theor. Ap. **141** (1994), no. 3, 202–208.
23. E. Polak, *Optimization: Algorithms and consistent approximations*, Appl. Math. Sci., vol. 124, Springer-Verlag, New York, 1997.
24. E. Polak and Y. Wardi, *Nondifferentiable optimization algorithm for designing control systems having singular value inequalities*, Automatica–J. IFAC **18** (1982), no. 3, 267–283.
25. T. Rautert and E. W Sachs, *Computational design of optimal output feedback controllers*, SIAM J. Optim. **7** (1997), no. 3, 837–852.
26. K. M. Sobel and E. Y. Shapiro, *A robust pitch pointing control law*, Proc. IEEE Conf. on Decision and Control (San Antonio), December 1983, pp. 1088–1093.
27. ———, *Flight control examples of robust eigenstructure assignment*, Proc. IEEE Conf. on Decision and Control (Los Angeles), December 1987, pp. 1290–1291.
28. K. N. Sobel, E. Y. Shapiro, and A. N. Andry, *Eigenstructure assignment*, Internat. J. Control **59** (1994), no. 1, 13–37.
29. J. E. Spingarn, *Submonotone subdifferentials of Lipschitz functions*, Trans. Amer. Math. Soc. **264** (1981), no. 1, 77–89.
30. W. M. Wonham, *On pole assignment in multi-input controllable linear systems*, IEEE Trans. Automatic Control **AC-12** (1967), 660–665.
31. K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control*, Prentice Hall, New Jersey, 1996.

# V

## Nonconvex bundle method with application to a delamination problem *

M. N. Dao, J. Gwinner, D. Noll, and N. Ovcharova

**Abstract.** Delamination is a typical failure mode of composite materials caused by weak bonding. It arises when a crack initiates and propagates under a destructive loading. Given the physical law characterizing the properties of the interlayer adhesive between the bonded bodies, we consider the problem of computing the propagation of the crack front and the stress field along the contact boundary. This leads to a hemivariational inequality, which after discretization by finite elements we solve by a nonconvex bundle method, where upper-$C^1$ criteria have to be minimized. As this is in contrast with other classes of mechanical problems with non-monotone friction laws and in other applied fields, where criteria are typically lower-$C^1$, we propose a bundle method suited for both types of nonsmoothness. We prove its global convergence in the sense of subsequences and test it on a typical delamination problem of material sciences.

**Keywords.** Composite material · delamination · crack front propagation · hemivariational inequality · Clarke directional derivative · nonconvex bundle method · lower- and upper-$C^1$ function · convergence.

### 1. Introduction

We develop a bundle technique to solve nonconvex variational problems arising in contact mechanics and in other applied fields. We are specifically interested in the delamination of composite structures with an adhesive bonding under destructive loading, a failure mode which is studied in the material sciences. When the properties of the interlayer adhesive between the bonded bodies are given in the form of a physical law relating the normal component of the stress vector to the relative displacement between the upper and lower boundaries at the crack tip, the challenge is to compute the displacement and stress fields in order to assess the reactive destructive forces along the contact boundary, as the latter are difficult to measure in situ. This leads to minimization of an energy functional, where a specific form

*Paper submitted for publication.

of nonsmoothness arises in the boundary integral at the contact boundary. After discretization via piecewise linear finite elements using the trapezoidal quadrature rule, this leads to a finite-dimensional nonsmooth optimization problem of the form

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax \leqslant b \end{array} \tag{1}$$

where $f$ is locally Lipschitz and neither smooth nor convex. Depending on the nature of the frictional forces, the criterion $f$ may be upper-$C^1$ or lower-$C^1$, see e.g. Figure 1. As these two classes of nonsmooth functions behave substantially differently when minimized, we are forced to expand on existing bundle strategies and develop an algorithm general enough to encompass both types of nonsmoothness. We prove its convergence to a critical point in the sense of subsequences, and show that it provides satisfactory numerical results in a simulation of the double cantilever beam test [35], one of the most popular destructive tests to qualify structural adhesive joints.

The difficulty in nonconvex bundling is to provide a suitable cutting plane oracle which replaces the no longer available convex tangent plane. One of the oldest oracles, discussed already in Mifflin [21], and used in the bundle codes of Lemaréchal and Sagastizábal [16, 17], or the BT-codes of Zowe [36, 33], uses the method of *downshifted tangents*. While these authors use linesearch with Armijo and Wolfe type conditions, which allows only weak convergence certificates in the sense that *some* accumulation point of the sequence of serious iterates is critical, we favor proximity control in tandem with a suitable backtracking strategy. This leads to stronger convergence certificates, where *every* accumulation point of the sequence of serious iterates is critical. For instance, in [24, 26, 7] a strong certificate for downshifted tangents with proximity control was proved within the class of lower-$C^1$ functions, but its validity for upper-$C^1$ criteria remained open. An oracle for upper-$C^1$ functions with a rigorous convergence theory can be based on the *model approach* of [24, 26, 25], but the latter is not compatible with the downshift oracle.

To have two strings to one bow is unsatisfactory, as one could hardly expect practitioners to select their strategy according to such a distinction, which might not be easy to make in practice. In this work we will resolve this impasse and present a cutting plane oracle based on downshifted tangents, which leads to a bundle method with strong convergence certificate for both types of nonsmoothness. In its principal components our method agrees with existing strategies for downshifted tangents, like [16, 36, 19, 20], and could therefore be considered as a justification of this technique for a wide class of applications. Differences with existing methods occur in the management of the proximity control parameter, which in our approach has to respect certain rules to assure convergence to a critical point, without impeding good practical performance.

The structure of the paper is as follows. Section 2 gives some preparatory information on lower- and upper-$C^1$ functions. Section 4 presents the algorithm and comments on its ingredients. Theoretical tools needed to prove convergence are presented and employed in sections 3 and 5. Section 6 gives the main convergence result, while section 7 discusses practical aspects of the algorithm. In section 8, we discuss the delamination problem, which we solve numerically using our bundle algorithm.

Numerical results for contact problem with adhesion based on the bundle-Newton method of L. Lukšan and J. Vlček [18] can be found e.g. in the book of Haslinger et al. [12], in [19, 20], and in the more recent [4, 23]. Mathematical analysis and numerical results for quasistatic delamination problems can be found in [15, 31].

## 2. Lower- and upper-$C^1$ functions

Following Spingarn [34], a locally Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}$ is lower-$C^1$ at $x_0$, if there exists a compact Hausdorff space $K$, a neighborhood $U$ of $x_0$, and a mapping $F : U \times K \to \mathbb{R}$ such that both $F$ and $D_x F$ are jointly continuous and

$$f(x) = \max\{F(x, y) : y \in K\} \tag{2}$$

is satisfied for $x \in U$. The function $f$ is upper-$C^1$ at $x_0$ if $-f$ is lower-$C^1$ at $x_0$.

In a minimization problem (1), we expect lower- and upper-$C^1$ functions to behave completely differently. Minimizing a lower-$C^1$ function ought to lead to real difficulties, as on descending we move *into* the zone of nonsmoothness, which for lower-$C^1$ goes downward. In contrast, upper-$C^1$ functions are generally expected to be well-behaved, as intuitively on descending we move *away* from the nonsmoothness, which here goes upward. The present application shows that this argument is too simplistic. Minimization of upper-$C^1$ functions leads to real difficulties, which we explain subsequently. In delamination for composite materials we encounter objective functions of the form

$$f(x) = f_s(x) + \int_0^1 \min_{i \in I} f_i(x, t)\, dt, \tag{3}$$

where $f_s$ gathers the smooth part, while the integral term, due to the minimum, is responsible for the nonsmoothness.

**Lemma 2.1.** *Suppose $f_s$ is of class $C^1$ and the $f_i$ are jointly of class $C^1$. Then the function* (3) *is upper-$C^1$ and can be represented in the form*

$$f(x) = f_s(x) + \min_{\sigma \in \Sigma} \int_0^1 f_{\sigma(t)}(x, t)\, dt, \tag{4}$$

*where $\Sigma$ is the set of all measurable mappings $\sigma : [0, 1] \to I$.*

*Proof.* Let us first prove (4). For $\sigma \in \Sigma$ and fixed $x \in \mathbb{R}^n$ the function $t \mapsto f_{\sigma(t)}(x, t)$ is measurable, and since $\min_{i \in I} f_i(x, t) \leqslant f_{\sigma(t)}(x, t) \leqslant \max_{i \in I} f_i(x, t)$, it is also integrable. Hence $F(x, \sigma) = f_s(x) + \int_0^1 f_{\sigma(t)}(x, t)\, dt$ is well-defined, and clearly $F(x, \sigma) \geqslant f(x)$, so we have $\inf_{\sigma \in \Sigma} F(x, \sigma) \geqslant f(x)$.

To prove the reverse estimate, fix $x \in \mathbb{R}^n$ and consider the closed-valued multifunction $\Phi : [0, 1] \to 2^I$ defined by $\Phi(t) = \{i \in I : f_i(x, t) = \min_{i' \in I} f_{i'}(x, t)\}$. Since the $f_i(x, \cdot)$ are measurable and $I$ is finite, $\Phi$ is a measurable multifunction. Choose a measurable selection $\sigma$, that is, $\sigma \in \Sigma$ satisfying $\sigma(t) \in \Phi(t)$ for every $t \in [0, 1]$. Then clearly $F(x, \sigma) = f(x)$. This proves (4).

Let us now show that $f$ is upper-$C^1$. We consider $\varphi(x,t) = \min_{i\in I} f_i(x,t)$. In view of [34] $\varphi(\cdot,t)$ is upper-$C^1$ and its Clarke subdifferential $\partial\varphi(\cdot,t)$ is strictly supermonotone uniformly over $t \in [0,1]$. By Theorem 2 in [5], $\varphi(\cdot,t)$ is approximately concave uniformly over $t \in [0,1]$. Integration with respect to $t \in [0,1]$ then yields an approximately concave function with respect to $x$, which by the equivalences in [5] and [34] is upper-$C^1$. □

Note that the minimum (4) is semi-infinite even though $I$ is finite. Minimization of (3) cannot be converted into a NLP, as would be possible in the min-max case. The representation (4) highlights the difficulty in minimizing (3). Minimizing a minimum has a disjunctive character, and due to the large size of $\Sigma$ this could lead to a combinatorial situation with intrinsic difficulty.

## 3. The model concept

The model of a nonsmooth function was introduced in [26] and is a key element in understanding the bundle concept.

**Definition 3.1** (Compare [26])**.** A function $\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is called a model of the locally Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}$ on the set $\Omega \subset \mathbb{R}^n$ if the following axioms are satisfied:

- $(M_1)$ For every $x \in \Omega$ the function $\phi(\cdot,x) : \mathbb{R}^n \to \mathbb{R}$ is convex, $\phi(x,x) = f(x)$ and $\partial_1\phi(x,x) \subset \partial f(x)$.
- $(M_2)$ For every $x \in \Omega$ and every $\varepsilon > 0$ there exists $\delta > 0$ such that $f(y) \leqslant \phi(y,x) + \varepsilon\|y-x\|$ for every $y \in B(x,\delta)$.
- $(M_3)$ The function $\phi$ is jointly upper semicontinuous, i.e., $(y_j,x_j) \to (y,x)$ on $\mathbb{R}^n \times \Omega$ implies $\limsup_{j\to\infty} \phi(y_j,x_j) \leqslant \phi(y,x)$. □

We recall that every locally Lipschitz function $f$ has the so-called *standard model*

$$\phi^\sharp(y,x) = f(x) + f^0(x, y-x),$$

where $f^0(x,d)$ is the Clarke directional derivative of $f$ at $x$ in direction $d$. The same function $f$ may in general have several models $\phi$, and following [24, 25], the standard $\phi^\sharp$ is the smallest one. Every model $\phi$ gives rise to a bundle strategy. The question is then whether this bundle strategy is successful. This depends on the following property of $\phi$.

**Definition 3.2.** A model $\phi$ of $f$ on $\Omega$ is said to be strict at $x_0 \in \Omega$ if axiom $(M_2)$ is replaced by the stronger

- $(\widehat{M}_2)$ For every $\varepsilon > 0$ there exists $\delta > 0$ such that $f(y) \leqslant \phi(y,x) + \varepsilon\|y-x\|$ for all $x, y \in B(x_0,\delta)$.

We say that $\phi$ is a strict model on $\Omega$, if it is strict at every $x_0 \in \Omega$. □

*Remark* 1. We may write axiom $(M_2)$ in the form $f(y) \leqslant \phi(y,x_0) + o(\|y-x_0\|)$ for $y \to x_0$, and $(\widehat{M}_2)$ as $f(y) \leqslant \phi(y,x) + o(\|y-x\|)$ for $x, y \to x_0$. Except for the fact that these concepts are one-sided, this is precisely the difference between differentiability and strict differentiability. Hence the nomenclature.

**Lemma 3.3** (Compare [24, 25]). *Suppose $f$ is upper-$C^1$. Then its standard model $\phi^\sharp$ is strict, and hence every model $\phi$ of $f$ is strict.* □

*Remark* 2. For convex $f$ the standard model $\phi^\sharp$ is in general not strict, but $f$ may be used as its own model $\phi(\cdot, x) = f$. For nonconvex $f$, a wide range of applications is covered by composite functions $f = g \circ F$ with $g$ convex and $F$ differentiable. Here the so-called natural model $\phi(y, x) = g(F(x) + F'(x)(y - x))$ can be used, because it is strict as soon as $F$ is class $C^1$. This includes lower-$C^2$ functions in the sense of [30], lower-$C^{1,\alpha}$ functions in the sense of [6], or amenable functions in the sense of [29], which allow representations of the form $f = g \circ F$ with $F$ of class $C^{1,1}$.

We conclude with the remark that lower-$C^1$ functions also admit strict models, even though in that case the construction is more delicate. The strict model in that case cannot be exploited algorithmically, and for lower-$C^1$ functions we prefer the oracle concept, which will be discussed in section 5.

## 4. Elements of the algorithm

In this section we briefly explain the main features of the algorithm. This concerns building the working model, computing the solution of the tangent program, checking acceptance, updating the working model after null steps, and the management of the proximity control parameter.

**4.1. Working model.** At the current serious iterate $x$ the inner loop of the algorithm at counter $k$ computes an approximation $\phi_k(\cdot, x)$ of $f$ in a neighborhood of $x$, called a first-order working model. The working model is a polyhedral convex function of the form

$$\phi_k(\cdot, x) = \max_{(a,g)\in\mathcal{G}_k} a + g^\top(\cdot - x), \tag{5}$$

where $\mathcal{G}_k$ is a finite set of affine functions $y \mapsto a + g^\top(y - x)$ satisfying $a \leqslant f(x)$, referred to as *planes*. The set $\mathcal{G}_k$ is updated during the inner loop $k$. At each step $k$ the following rules have to be respected when updating $\mathcal{G}_k$ into $\mathcal{G}_{k+1}$:

- ($R_1$) One or several cutting planes at the null step $y^k$, generated by an abstract cutting plane oracle, are added to $\mathcal{G}_{k+1}$.
- ($R_2$) The so-called aggregate plane $(a^*, g^*)$, which consists of convex combinations of elements of $\mathcal{G}_k$, is added to $\mathcal{G}_{k+1}$.
- ($R_3$) Some older planes in $\mathcal{G}_k$, which become obsolete through the addition of the aggregate plane, are discarded and not kept in $\mathcal{G}_{k+1}$.
- ($R_4$) Every $\mathcal{G}_k$ contains at least one so-called exactness plane $(a_0, g_0)$, where exactness plane means $a_0 = f(x)$, $g_0 \in \partial f(x)$. This assures $\phi_k(x, x) = f(x)$, hence the name.
- ($R_5$) We have to make sure that each working model $\phi_k$ satisfies $\partial_1\phi_k(x, x) \subset \partial f(x)$.

Once the first-order working model $\phi_k(\cdot, x)$ has been built, the second-order working model $\Phi_k(\cdot, x)$ is of the form

$$\Phi_k(\cdot, x) = \phi_k(\cdot, x) + \tfrac{1}{2}(\cdot - x)^\top Q(x)(\cdot - x), \tag{6}$$

where $Q(x) = Q(x)^\top$ is a possibly indefinite symmetric matrix, depending only on the current serious iterate $x$, and fixed during the inner loop $k$. The second-order term includes curvature information on $f$, if available.

**4.2. Tangent program and acceptance test.** Once the second-order working model (6) is formed and the proximity control parameter $\tau_{k-1} \to \tau_k$ is updated, we solve the tangent program

$$\begin{array}{ll} \text{minimize} & \Phi_k(y,x) + \frac{\tau_k}{2}\|y-x\|^2 \\ \text{subject to} & Ay \leqslant b \end{array} \tag{7}$$

Here the proximity control parameter $\tau_k$ satisfies $Q + \tau_k I \succ 0$, which assures that (7) is strictly convex and has a unique solution, $y^k$, called the *trial step*. The trial step is a candidate to become the new serious iterate $x^+$. In order to decide whether $y^k$ is acceptable, we compute the test

$$\rho_k = \frac{f(x) - f(y^k)}{f(x) - \Phi_k(y^k, x)} \overset{?}{\geqslant} \gamma, \tag{8}$$

where $0 < \gamma < 1$ is some fixed parameter. If $\rho_k \geqslant \gamma$, then $x^+ = y^k$ is accepted and called a *serious step*. In this case the inner loop ends successfully. On the other hand, if $\rho_k < \gamma$, then $y^k$ is rejected and called a *null step*. In this case the inner loop $k$ continues. This means we will update working model $\Phi_k(\cdot, x) \to \Phi_{k+1}(\cdot, x)$, adjust the proximity control parameter $\tau_k \to \tau_{k+1}$, and solve (7) again.

Note that the test (8) corresponds to the usual Armijo descent condition used in linesearches, or to the standard acceptance test in trust region methods.

**4.3. Updating the working model via aggregation.** Suppose the trial step $y^k$ fails the acceptance test (8) and is declared a null step. Then the inner loop has to continue, and we have to improve the working model at the next sweep in order to perform better. Since the second-order part of the working model $\frac{1}{2}(\cdot - x)^\top Q(x)(\cdot - x)$ remains invariant, we will update the first-order part only.

Concerning rule $(R_2)$, by the necessary optimality condition for (7), there exists a multiplier $\eta^*$ such that

$$0 \in \partial_1 \Phi_k(y^k, x) + \tau_k(y^k - x) + A^\top \eta^*,$$

or what is the same,

$$(Q(x) + \tau_k I)(y^k - x) - A^\top \eta^* \in \partial_1 \phi_k(y^k, x).$$

Since $\phi_k(\cdot, x)$ is by construction a maximum of affine planes, we use the standard description of the convex subdifferential of a max-function. Writing $\mathcal{G}_k = \{(a_0, g_0), \ldots, (a_p, g_p)\}$ for $p = \operatorname{card}(\mathcal{G}_k) + 1$, we find non-negative multipliers $\lambda_0, \ldots, \lambda_p$ summing up to 1 such that

$$(Q(x) + \tau_k I)(y^k - x) - A^\top \eta^* = \sum_{i=0}^{p} \lambda_i g_i,$$

and in addition, $a_i + g_i^\top(y^k - x) = \phi_k(y^k, x)$ for all $i \in \{0, \ldots, p\}$ with $\lambda_i > 0$. We say that those planes which are active at $y^k$ are *called by the aggregate plane*. In the

above rule $(R_3)$ we allow those to be removed from $\mathcal{G}_k$. We now define the aggregate plane as:

$$a_k^* = \sum_{i=0}^{p} \lambda_i a_i, \quad g_k^* = \sum_{i=0}^{p} \lambda_i g_i.$$

Note that by construction the aggregate plane $m_k^*(\cdot, x) = a_k^* + g_k^{*\top}(\cdot - x)$ at null step $y^k$ satisfies $m_k^*(y^k, x) = a^* + g^{*\top}(y^k - x) = \phi_k(y^k, x)$. This construction is standard and follows the original idea in Kiwiel [14]. It assures in particular that $\Phi_{k+1}(y^k, x) \geqslant m_k^*(y^k, x) + \frac{1}{2}(y^k - x)^\top Q(x)(y^k - x) = \Phi_k(y^k, x)$.

**4.4. Updating the working model by cutting planes and exactness planes.** The crucial improvement in the first-order working model is in adding a cutting plane which cuts away the unsuccessful trial step $y^k$ according to rule $(R_1)$. We shall denote the cutting plane as $m_k(\cdot, x) = a_k + g^\top(\cdot - x)$. The only requirement for the time being is that $a_k \leqslant f(x)$, as this assures $\phi_{k+1}(x, x) \leqslant f(x)$. Since we also maintain at least one exactness plane of the form $m_0(\cdot, x) = f(x) + g_0^\top(\cdot - x)$ with $g_0 \in \partial f(x)$, we assure $\phi_{k+1}(x, x) = \Phi_{k+1}(x, x) = f(x)$. Later we will also have to check the validity of $(R_5)$.

It is possible to integrate so-called *anticipated cutting planes* in the new working model $\mathcal{G}_{k+1}$. Here anticipated designates all planes which are not based on the rules exactness, aggregation, cutting planes. Naturally, adding such planes can not be allowed in an arbitrary way, because axioms $(R_1) - (R_5)$ have to be respected.

*Remark* 3. It may be beneficial to choose a new exactness plane $m_0(\cdot, x) = f(x) + g^\top(\cdot - x)$ after each null step $y$, namely the one which satisfies $m_0(y, x) = f^0(x, y - x)$. If $x$ is a point of differentiability of $f$, then all these exactness planes are identical anyway, so no extra work occurs. On the other hand, computing $g \in \partial f(x)$ such that $g^\top(y - x) = f^0(x, y - x)$ is usually cheap. Consider for instance eigenvalue optimization, where $f(x) = \lambda_1(F(x))$, $x \in \mathbb{R}^n$, $F : \mathbb{R}^n \to \mathbb{S}^m$, and $\lambda_1 : \mathbb{S}^m \to \mathbb{R}$ is the maximum eigenvalue function of $\mathbb{S}^m$. Then $f^0(x, d) = \lambda_1'(X, D) = \lambda_1(Q^\top D Q)$, where $X = F(x)$, $D = F'(x)d$, and where $Q$ is a $t \times m$ matrix whose columns form an orthogonal basis of the maximum eigenspace of $X$ of dimension $t$ [3]. Then $G = QQ^\top \in \partial\lambda_1(X)$ attains $\lambda_1'(X, D)$, hence $g = F'(x)^* QQ^\top$ attains $f'(x, d)$. Since usually $t \ll m$, the computation of $g$ is cheap.

**4.5. Management of proximity control.** The central novelty of the bundle methods developed in [24, 26, 1] is the discovery that in the absence of convexity the proximity control parameter $\tau$ has to follow certain basic rules to assure convergence of the sequence $x^j$ of serious iterates. This is in contrast with convex bundle methods, where $\tau$ could in principle be frozen once and for all. More precisely, suppose $\phi_k(\cdot, x)$ has failed and produced only a null step $y^k$. Having built the new model $\phi_{k+1}(\cdot, x)$, we compute the secondary test

$$\widetilde{\rho}_k = \frac{f(x) - \Phi_{k+1}(y^k, x)}{f(x) - \Phi_k(y^k, x)} \overset{?}{\geqslant} \widetilde{\gamma}, \tag{9}$$

where $0 < \gamma < \widetilde{\gamma} < 1$ is fixed. Our decision is

$$\tau_{k+1} = \begin{cases} 2\tau_k & \text{if } \widetilde{\rho}_k \geqslant \widetilde{\gamma} \\ \tau_k & \text{if } \widetilde{\rho}_k < \widetilde{\gamma} \end{cases} \tag{10}$$

The rationale of (9) is to decide whether improving the model by adding planes will suffice, or shorter steps have to be forced by increasing $\tau$.

The denominator in (9) gives the model predicted progress $f(x) - \phi_k(y^k, x) = \phi_k(x, x) - \phi_k(y^k, x) > 0$ at $y^k$. On the other hand, the numerator $f(x) - \phi_{k+1}(y^k, x)$ gives the progress over $x$ we would achieve at $y^k$, had we already known the cutting planes drawn at $y^k$. Due to aggregation we know that $\phi_{k+1}(y^k, x) \geqslant \phi_k(y^k, x)$, so that $\widetilde{\rho}_k \leqslant 1$, but values $\widetilde{\rho}_k \approx 1$ indicate that little to no progress is achieved by adding the cutting plane. In this case we decide that the $\tau$-parameter must be increased to force smaller steps, because that reinforces the agreement between $f$ and $\phi_{k+1}(\cdot, x)$.

In the test (10) we replace $\widetilde{\rho}_k \approx 1$ by $\widetilde{\rho}_k \geqslant \widetilde{\gamma}$ for some fixed $0 < \gamma < \widetilde{\gamma} < 1$. If $\widetilde{\rho}_k < \widetilde{\gamma}$, then the quotient if far from 1 and we decide that adding planes has still the potential to improve the situation. In that event we do not increase $\tau$.

Let us next consider the management of $\tau$ in the outer loop. Since $\tau$ can only increase or stay fixed in the inner loop, we allow $\tau$ to decrease between serious steps $x \to x^+$, respectively, $x^j \to x^{j+1}$. This is achieved by the test

$$\rho_{k_j} = \frac{f(x^j) - f(x^{j+1})}{f(x^j) - \Phi_{k_j}(x^{j+1}, x^j)} \overset{?}{\geqslant} \Gamma, \tag{11}$$

where $0 < \gamma \leqslant \Gamma < 1$ is fixed. In other words, if at acceptance we have not only $\rho_{k_j} \geqslant \gamma$, but even $\rho_{k_j} \geqslant \Gamma$, then we decrease $\tau$ at the beginning of the next inner loop $j + 1$, because we may trust the model. On the other hand, if $\gamma \leqslant \rho_{k_j} < \Gamma$ at acceptance, then we memorize the last $\tau$-parameter used, that is $\tau_{k_j}$ at the end of the $j$th inner loop.

*Remark* 4. We should compare our management of the proximity control parameter $\tau$ with other strategies in the literature. For instance Mäkelä *et al.* [19] consider a very different management of $\tau$, which is motivated by the convex case.

**4.6. Statement of the algorithm.** We are now ready to give our formal statement of Algorithm 1 (See next page).

## 5. Nonconvex cutting plane oracles

In the convex cutting plane method [32, 13] unsuccessful trial steps $y^k$ are cut away by adding a tangent plane to $f$ at $y^k$ into the model. Due to convexity, the cutting plane is below $f$ and can therefore be used to construct an approximation (5) of $f$. For nonconvex $f$, cutting planes are more difficult to construct, but several ideas have been discussed. We mention [11, 21]. In [24] we have proposed an axiomatic approach, which has the advantage that it covers the applications we are aware of, and allows a convenient convergence theory. Here we use this axiomatic approach in the convergence proof.

**Definition 5.1** (Compare [24])**.** Let $f$ be locally Lipschitz. A cutting plane oracle for $f$ on the set $\Omega$ is an operator $\mathscr{O}$ which, with every pair $(x, y)$, $x$ a serious iterate in $\Omega$, $y \in \mathbb{R}^n$ a null step, associates an affine function $m_y(\cdot, x) = a + g^\top(\cdot - x)$, called the cutting plane at null step $y$ for serious iterate $x$, so that the following axioms are satisfied:

**Algorithm 1.** Proximity control algorithm for (1)

**Parameters:** $0 < \gamma < \Gamma < 1$, $\gamma < \widetilde{\gamma} < 1$, $0 < q < \infty$, $q < T < \infty$.

▷ **Step 1** (`Initialize outer loop`). Choose initial guess $x^1$ with $Ax^1 \leqslant b$ and an initial matrix $Q_1 = Q_1^\top$ with $-qI \preceq Q_1 \preceq qI$. Fix memory control parameter $\tau_1^\sharp$ such that $Q_1 + \tau_1^\sharp I \succ 0$. Put $j = 1$.

◇ **Step 2** (`Stopping test`). At outer loop counter $j$, stop if $0 \in \partial f(x^j) + A^\top \eta^*$ for some multiplier $\eta^* \geqslant 0$. Otherwise goto inner loop.

▷ **Step 3** (`Initialize inner loop`). Put inner loop counter $k = 1$ and initialize $\tau$-parameter using the memory element, i.e., $\tau_1 = \tau_j^\sharp$. Choose initial convex working model $\phi_1(\cdot, x^j)$, possibly recycling some planes from previous sweep $j-1$, and let $\Phi_1(\cdot, x^j) = \phi_1(\cdot, x^j) + \frac{1}{2}(\cdot - x^j)^\top Q_j(\cdot - x^j)$.

▷ **Step 4** (`Trial step generation`). At inner loop counter $k$ solve tangent program

$$\min_{Ay \leqslant b} \Phi_k(y, x^j) + \tfrac{\tau_k}{2}\|y - x^j\|^2.$$

The solution is the new trial step $y^k$.

◇ **Step 5** (`Acceptance test`). Check whether

$$\rho_k = \frac{f(x^j) - f(y^k)}{f(x^j) - \Phi_k(y^k, x^j)} \geqslant \gamma.$$

If this is the case put $x^{j+1} = y^k$ (serious step), quit inner loop and goto step 8. If this is not the case (null step) continue inner loop with step 6.

▷ **Step 6** (`Update working model`). Build new convex working model $\phi_{k+1}(\cdot, x^j)$ based on null step $y^k$ by adding an exactness plane $m_k^\sharp(\cdot, x^j)$ satisfying $m_k^\sharp(y^k, x^j) = f^0(x^j, y^k - x^j)$, a downshifted tangent $m_k^\downarrow(\cdot, x^j)$, and the aggregate plane $m_k^*(\cdot, x^j)$. Apply rule $(R_3)$ to avoid overflow. Build $\Phi_{k+1}(\cdot, x^j)$, and goto step 7.

◇ **Step 7** (`Update proximity parameter`). Compute

$$\widetilde{\rho}_k = \frac{f(x^j) - \Phi_{k+1}(y^k, x^j)}{f(x^j) - \Phi_k(y^k, x^j)}.$$

Put

$$\tau_{k+1} = \begin{cases} \tau_k, & \text{if } \widetilde{\rho}_k < \widetilde{\gamma} \qquad \text{(bad)} \\ 2\tau_k, & \text{if } \widetilde{\rho}_k \geqslant \widetilde{\gamma} \qquad \text{(too bad)} \end{cases}$$

Then increase counter $k$ and continue inner loop with step 4.

◇ **Step 8** (`Update` $Q_j$ `and memory element`). Update matrix $Q_j \to Q_{j+1}$, respecting $Q_{j+1} = Q_{j+1}^\top$ and $-qI \preceq Q_{j+1} \preceq qI$. Then store new memory element

$$\tau_{j+1}^\sharp = \begin{cases} \tau_k, & \text{if } \gamma \leqslant \rho_k < \Gamma \qquad \text{(not bad)} \\ \frac{1}{2}\tau_k, & \text{if } \rho_k \geqslant \Gamma \qquad \text{(good)} \end{cases}$$

Increase $\tau_{j+1}^\sharp$ if necessary to ensure $Q_{j+1} + \tau_{j+1}^\sharp I \succ 0$.

◇ **Step 9** (`Large multiplier safeguard rule`). If $\tau_{j+1}^\sharp > T$ then re-set $\tau_{j+1}^\sharp = T$. Increase outer loop counter $j$ by 1 and loop back to step 2.

($O_1$) For $y = x$ we have $a = f(x)$ and $g \in \partial f(x)$.
($O_2$) Let $y_j \to x$. Then there exist $\varepsilon_j \to 0^+$ such that $f(y_j) \leqslant m_{y_j}(y_j, x) + \varepsilon_j \|y_j - x\|$.
($O_3$) Let $x_j \to x$ and $y_j, y_j^+ \to y$. Then there exists $z \in \mathbb{R}^n$ such that $\limsup_{j\to\infty} m_{y_j^+}(y_j, x_j) \leqslant m_z(y, x)$. □

As we shall see, these axioms are aligned with the model axioms $(M_1) - (M_3)$. Not unexpectedly, there is also a strict version of $(O_2)$.

**Definition 5.2.** A cutting plane oracle $\mathscr{O}$ for $f$ is called strict at $x_0$ if the following strict version of $(O_2)$ is satisfied:

($\widehat{O}_2$) Suppose $y_j, x_j \to x$. Then there exist $\varepsilon_j \to 0^+$ such that $f(y_j) \leqslant m_{y_j}(y_j, x_j) + \varepsilon_j \|y_j - x_j\|$. □

We now discuss two versions of the oracle which are of special interest for our applications.

*Example* 5.1 (Model-based oracle). Suppose $\phi$ is a model of $f$. Then we can generate a cutting plane for serious iterate $x$ and trial step $y$ by taking $g \in \partial_1 \phi(y, x)$ and putting

$$m_y(\cdot, x) = \phi(y, x) + g^\top(\cdot - y) = \phi(y, x) + g^\top(x - y) + g^\top(\cdot - x).$$

Oracles generated by a model $\phi$ in this way will be denoted $\mathscr{O}_\phi$. Note that $\mathscr{O}_\phi$ coincides with the standard oracle if $f$ is convex and $\phi(\cdot, x) = f$, i.e., if the convex $f$ is chosen as its own model. In more general cases, the simple idea of this oracle is that in the absence of convexity, where tangents to $f$ at $y$ are not useful, we simply take tangents of $\phi(\cdot, x)$ at $y$. Note that the model-based oracle $\mathscr{O}_\phi$ is strict as soon as the model $\phi$ is strict. □

*Example* 5.2 (Standard oracle). A special case of the model-based oracle is obtained by choosing the standard model $\phi^\sharp$. Due to its significance for our present work we call this the standard oracle. The standard cutting plane for serious step $x$ and null step $y$ is $m_y^\sharp(\cdot, x) = f(x) + g^\top(\cdot - x)$, where the Clarke subgradient $g \in \partial f(x)$ is one of those that satisfy $g^\top(y - x) = f^0(x, y - x)$. The standard oracle is strict iff $\phi^\sharp$ is strict. As was observed before, this is for instance the case when $f$ is upper-$C^1$. Note a specificity of the standard oracle: every standard cutting plane $m_y^\sharp(\cdot, x)$ is also an exactness plane at $x$. □

*Example* 5.3 (Downshifted tangents). Probably the oldest oracle used for nonconvex functions are downshifted tangents, which we define as follows. For serious iterate $x$ and null step $y$ let $t(\cdot) = f(y) + g^\top(\cdot - y)$ be a tangent of $f$ at $y$. That is, $g \in \partial f(y)$. Then we shift $t(\cdot)$ down until it becomes useful for the model (5). Fixing a parameter $c > 0$, this is organized as follows: We define the cutting plane as $m_y^\downarrow(\cdot, x) = t(\cdot) - s$, where the downshift $s \geqslant 0$ satisfies

$$s = [t(x) - f(x) + c\|y - x\|^2]_+.$$

In other words, $m_y^\downarrow(\cdot, x) = a + g^\top(\cdot - x)$, where $a = \min\{t(x), f(x) - c\|y - x\|^2\}$. Note that this procedure aways satisfies axioms $(O_1)$ and $(O_3)$, whereas axioms $(O_2)$, respectively, $(\widehat{O}_2)$, are satisfied if $f$ is lower-$C^1$ at $x_0$. In other words, see [24], for $f$ lower-$C^1$ this is an oracle, which is automatically strict. □

Motivated by the previous examples, we now define an oracle which works for both lower-$C^1$ and upper-$C^1$.

*Example* 5.4 (Modified downshift). Let $x$ be the current serious iterate, $y$ a null step in the inner loop belonging to $x$. Then we form the downshifted tangent $m_y^{\downarrow}(\cdot, x) := t(\cdot) - s$, that is, the cutting plane we would get from the downshift oracle, and we form the standard oracle plane $m_y^{\sharp}(\cdot, x) = f(x) + g^{\top}(\cdot - x)$, where the Clarke subgradient $g$ satisfies $f^0(x, y - x) = g^{\top}(y - x)$. Then we define

$$m_y(\cdot, x) = \begin{cases} m_y^{\downarrow}(\cdot, x) & \text{if } m_y^{\downarrow}(y, x) \geqslant m_y^{\sharp}(y, x) \\ m_y^{\sharp}(\cdot, x) & \text{else} \end{cases}$$

In other words, among the two candidate cutting planes $m_y^{\downarrow}(\cdot, x)$ and $m_y^{\sharp}(\cdot, x)$, we take the one which has the larger value at the null step $y$.

Note that this is the oracle we use in our algorithm. Theorem 6.1 clarifies when this oracle is strict. □

Given an operator $\mathscr{O}$ which with every pair $(x, y)$ of serious step $x$ and null step $y$ associates a cutting plane $m_y(\cdot, x) = a + g^{\top}(\cdot - x)$, we fix a constant $M > 0$ and define what we call the upper envelope function of the oracle

$$\phi^{\uparrow}(\cdot, x) = \sup\{m_y(\cdot, x) : \|y - x\| \leqslant M\}.$$

The crucial property of $\phi^{\uparrow}$ is the following

**Lemma 5.3.** *Suppose $\mathscr{O} : (x, y) \mapsto m_y(\cdot, x)$ is a cutting plane oracle satisfying axioms $(O_1) - (O_3)$. Then $\phi^{\uparrow}$ is a model of $f$. Moreover, if the oracle satisfies $(\widehat{O}_2)$, then $\phi^{\uparrow}$ is strict.* □

The proof can be found in [24]. We refer to $\phi^{\uparrow}$ as the upper envelope model associated with the oracle $\mathscr{O}$. Since in turn every model $\phi$ gives rise to a model-based oracle, $\mathscr{O}_{\phi}$, it follows that having a strict oracle and having a strict model are equivalent properties of $f$. Note, however, that the model $\phi^{\uparrow}$ is in general not practically useful. It is a theoretical tool in the convergence proof.

*Remark* 5. If we start with a model $\phi$, then build $\mathscr{O}_{\phi}$, and go back to $\phi^{\uparrow}$, we get back to $\phi$, at least locally.

On the other hand, going from an oracle $\mathscr{O}$ to its envelope model $\phi^{\uparrow}$, and then back to the model based oracle $\mathscr{O}_{\phi^{\uparrow}}$ does *not* necessarily lead back to the oracle $\mathscr{O}$.

We are now in the position to check axiom $(R_5)$.

**Corollary 5.4.** *All working models $\phi_k$ constructed in our algorithm satisfy $\partial_1 \phi_k(x, x) \subset \partial f(x)$.* □

## 6. Main convergence result

In this section we state and prove the main result of this work and give several consequences.

**Theorem 6.1.** *Let $f$ be locally Lipschitz and suppose for every $x \in \mathbb{R}^n$, $f$ is either lower-$C^1$ or upper-$C^1$ at $x$. Let $x^1$ be such that $Ax^1 \leqslant b$ and $\{x \in \mathbb{R}^n : f(x) \leqslant f(x^1), Ax \leqslant b\}$ is bounded. Then every accumulation point $x^*$ of the sequence $x^j$ of serious iterates generated by Algorithm 1 is a KKT point of (1).*

*Proof.* The result will follow from [24, Theorem 1] as soon as we show that downshifted tangents as modified in Example 5.4 and used in the algorithm is a strict cutting plane oracle in the sense of definition 5.2. The remainder of the proof is to verify this.

1) Let us denote cutting planes arising from the standard model $\phi^\sharp$ by $m_y^\sharp(\cdot, x)$, cutting planes obtained by downshift as $m_y^\downarrow(\cdot, x) = t(\cdot) - s$, and the true cutting plane of the oracle as $m_y(\cdot, x)$. Then as we know $m_y(\cdot, x) = m_y^\downarrow(\cdot, x)$ if $m_y^\downarrow(y, x) \geqslant m_y^\sharp(y, x)$, and otherwise $m_y(\cdot, x) = m_y^\sharp(\cdot, x)$. We have to check $(O_1)$, $(\widehat{O}_2)$, $(O_3)$.

2) The validity of $(O_1)$ is clear, as both oracles provide Clarke tangent planes to $f$ at $x$ for $y = x$.

3) Let us now check $(O_3)$. Consider $x_j \to x$, and $y_j, y_j^+ \to y$. Here $y_j^+$ is a null step at serious step $x_j$. Passing to a subsequence, we may distinguish case I, where $m_{y_j^+}(\cdot, x_j) = m_{y_j^+}^\sharp(\cdot, x_j)$ for every $j$, and case II, where $m_{y_j^+}(\cdot, x_j) = m_{y_j^+}^\downarrow(\cdot, x_j)$ for every $j$.

Consider case I first. Let $m_{y_j^+}^\sharp(y_j, x_j) = f(x_j) + g_j^\top(y_j - x_j)$, where $g_j \in \partial f(x_j)$ satisfies $f^0(x_j, y_j^+ - x_j) = g_j^\top(y_j^+ - x_j)$. Passing to yet another subsequence, we may assume $g_j \to g$, and upper semi-continuity of the Clarke subdifferential gives $g \in \partial f(x)$. Therefore $m_{y_j^+}(y_j, x_j) = f(x_j) + g_j^\top(y_j - x_j) \to f(x) + g^\top(y - x) \leqslant m_y^\sharp(y, x) \leqslant m_y(y, x)$. So here $(O_3)$ is satisfied with $z = y$.

Newt consider case II. Here we have $m_{y_j^+}(y_j, x_j) = t_{g_j}(y_j) - s_j$, where $t_{g_j}(\cdot)$ is a tangent to $f$ at $y_j^+$ with subgradient $g_j \in \partial f(y_j^+)$, and $s_j$ is the corresponding downshift

$$s_j = \left[t_{g_j}(x_j) - f(x_j) + c\|y_j^+ - x_j\|^2\right]_+ .$$

Passing to a subsequence, we may assume $g_j \to g$, and by upper semi-continuity of $\partial f$ we have $g \in \partial f(y)$. Therefore $s_j \to [t_g(x) - f(x) + c\|y - x\|^2]_+ =: s$, where uniform convergence $t_{g_j}(y_j) \to t_g(y)$ occurs due to the boundedness of $\partial f$. But now we see that $s$ is the downshift for the pair $(x, y)$ when $g \in \partial f(y)$ is used. Hence $m_{y_j^+}(y_j, x_j) \to m_y^\downarrow(y, x)$, and since $m_y^\downarrow(y, x) \leqslant m_y(y, x)$, we are done. So again the $z$ in $(O_3)$ equals $y$ here.

4) Let us finally check axiom $(\widehat{O}_2)$. Let $x_j, y_j \to x$ be given. We first consider the case when $f$ is upper-$C^1$ at $x$. We have to find $\varepsilon_j \to 0^+$ such that $f(y_j) \leqslant m_{y_j}(y_j, x_j) + \varepsilon_j\|y_j - x_j\|$ as $j \to \infty$, and by the definition of the oracle, it clearly suffices to show $f(y_j) \leqslant m_{y_j}^\sharp(y_j, x_j) + \varepsilon_j\|y_j - x_j\|$. By Spingarn [34], or Daniilidis and Georgiev [5], $-f$, which is lower-$C^1$ at $x$, has the following property: For every $\varepsilon > 0$ there exists $\delta > 0$ such that for all $0 < t < 1$ and $y, z \in B(x, \delta)$,

$$f(y) \leqslant f(z) + t^{-1}\left(f(z + t(y - z)) - f(z)\right) + \varepsilon(1 - t)\|z - y\|.$$

Taking the limit superior $t \to 0^+$ implies

$$f(y) \leqslant f(z) + f'(z, y-z) + \varepsilon\|y-z\| \leqslant f(z) + f^0(z, y-z) + \varepsilon\|y-z\|.$$

Choosing $z = x_j$, $y = y_j$, $\delta_j = \|y_j - z_j\| \to 0$, we can find $\varepsilon_j \to 0^+$ such that $f(y_j) \leqslant f(x_j) + f^0(x_j, y_j - x_j) + \varepsilon_j\|y_j - x_j\|$, hence $f(y_j) \leqslant m^\sharp_{y_j}(y_j, x_j) + \varepsilon_j\|y_j - x_j\|$ by the definition of $m^\sharp_{y_j}(\cdot, x_j)$. That settles the upper-$C^1$ case.

Now consider the case where $f$ is lower-$C^1$ at $x$. We have to find $\varepsilon_j \to 0^+$ such that $f(y_j) \leqslant m_{y_j}(y_j, x_j) + \varepsilon_j\|y_j - x_j\|$ as $j \to \infty$, and it suffices to show $f(y_j) \leqslant m^\downarrow_{y_j}(y_j, x_j) + \varepsilon_j\|y_j - x_j\|$. Since $m^\downarrow_{y_j}(y_j, x_j) \geqslant f(y_j) - s_j$, where $s_j$ is the downshift $s_j = [t(x_j) - f(x_j) + c\|y_j - x_j\|^2]_+$, and $t(\cdot) = f(y_j) + g_j^\top(\cdot - y_j)$ for some $g_j \in \partial f(y_j)$, it suffices to exhibit $\varepsilon_j \to 0^+$ such that $f(y_j) \leqslant f(y_j) - s_j + \varepsilon_j\|y_j - x_j\|$, or what is the same, $s_j \leqslant \varepsilon_j\|y_j - x_j\|$. For that it suffices to arrange $[t(x_j) - f(x_j)]_+ \leqslant \varepsilon_j\|y_j - x_j\|$, because once this is verified, we get $s_j \leqslant [t(x_j) - f(x_j)]_+ + c\|y_j - x_j\|^2 \leqslant (\varepsilon_j + c\|y_j - x_j\|)\|y_j - x_j\| =: \widetilde{\varepsilon}_j\|y_j - x_j\|$. Note again that by [34, 5] $f$ has the following property at $x$: For every $\varepsilon > 0$ there exists $\delta > 0$ such that $f(tz + (1-t)y) \leqslant tf(z) + (1-t)f(y) + \varepsilon t(1-t)\|z - y\|$ for all $y, z \in B(x, \delta)$. Dividing by $t > 0$ and passing to the limit $t \to 0^+$ gives $f^0(y, z - y) \leqslant f(z) - f(y) + \varepsilon\|y - z\|$, using the fact that $f$ is locally Lipschitz. But for every $g \in \partial f(y)$, $g^\top(z - y) \leqslant f^0(y, z - y)$. Using $\|y_j - x_j\| =: \delta_j \to 0$ and taking $y = y_j$, $z = x_j$, this allows us to find $\varepsilon_j \to 0^+$ such that $g_j^\top(x_j - y_j) \leqslant f(x_j) - f(y_j) + \varepsilon_j\|y_j - x_j\|$. Substituting this above gives $t(x_j) - f(x_j) = f(y_j) - f(x_j) + g_j^\top(x_j - y_j) \leqslant \varepsilon_j\|y_j - x_j\|$ as desired. That settles the lower-$C^1$ case. □

## 7. Practical aspects of the algorithm

In this section we discuss several technical aspects of the algorithm, which are important for its performance.

**7.1. Stopping.** The stopping test in step 2 of the algorithm is stated in this form for the sake of the convergence proof. In practice we delegate stopping to the inner loop using the following two-stage procedure.

If the inner loop at serious iterate $x^j$ finds the new serious step $x^{j+1}$ such that

$$\frac{\|x^{j+1} - x^j\|}{1 + \|x^j\|} < \mathrm{tol}_1, \quad \frac{|f(x^{j+1}) - f(x^j)|}{1 + |f(x^j)|} < \mathrm{tol}_2,$$

then we decide that $x^{j+1}$ is optimal. In consequence, the $(j+1)$st inner loop will not be executed. On the other hand, if the inner loop has difficulties terminating and produces five consecutive null steps $y^k$ where

$$\frac{\|y^k - x^j\|}{1 + \|x^j\|} < \mathrm{tol}_1, \quad \frac{|f(y^k) - f(x^j)|}{1 + |f(x^j)|} < \mathrm{tol}_2,$$

or if a maximum number $k_{\max}$ of allowed steps in the inner loop is reached, then we decide that $x^j$ is optimal. In our experiments we use $\mathrm{tol}_1 = 10^{-5}$, $\mathrm{tol}_2 = 10^{-5}$, and $k_{\max} = 50$.

**7.2. Recycling of planes.** At the beginning of a new inner loop at serious step $x^{j+1}$, we do not want to start building the working model $\phi_1(\cdot, x^{j+1})$ from scratch. It is more efficient to recycle some of the planes $(a, g) \in \mathcal{G}_{k_j}$ in the latest working model $\phi_{k_j}(\cdot, x^j)$. In the convex cutting plane method, this is self-understood, as cutting planes are affine minorants of $f$, and can at leisure stay on in the sets $\mathcal{G}$ at all times $j, k$. Without convexity, we need the following recycling procedure:

Given a plane $m(\cdot, x^j) = a + g^\top(\cdot - x^j)$ in the latest set $\mathcal{G}_{k_j}$, we form the new downshifted plane

$$m(\cdot, x^{j+1}) = m(\cdot, x^j) - s,$$

where the downshift is organized as

$$s = \left[m(x^{j+1}, x^j) - f(x^{j+1}) + c\|x^{j+1} - x^j\|^2\right]_+ .$$

In other words, we treat $m(\cdot, x^j)$ like a tangent to $f$ at null step $x^j$ with respect to the serious step $x^{j+1}$ in the downshift oracle. We put

$$m(\cdot, x^{j+1}) = a + g^\top(\cdot - x^j) - s = a - s + g^\top(x^{j+1} - x^j) + g^\top(\cdot - x^{j+1}),$$

and we accomodate $(a - s + g^\top(x^{j+1} - x^j), g) \in \mathcal{G}_1$ at the beginning of the $(j + 1)$st inner loop. In the modified version we only keep a plane of this type in $\mathcal{G}_1$ after comparing it to the exactness plane $m_0(\cdot, x^{j+1}) = f(x^{j+1}) + g^\top(\cdot - x^{j+1})$, $g \in \partial f(x^{j+1})$, which satisfies $g^\top(x^j - x^{j+1}) = f^0(x^{j+1}, x^j - x^{j+1})$. Indeed, when $m(x^j, x^{j+1}) \geqslant m_0(x^j, x^{j+1})$, then we keep the downshifted plane, otherwise we add $m_0(\cdot, x^{j+1})$ as additional exactness plane.

## 8. The delamination benchmark problem

The interface behavior of laminated composite materials is modeled by a non-monotone multi-valued function $\partial j$, characteristic of the interlayer adhesive placed at the contact boundary $\Gamma_c$. In more precise terms, $\partial j$ is the physical law which holds between the normal component $-S_n(s)|\Gamma_c$ of the stress vector and the relative displacement $u_2(s)|\Gamma_c$, or *jump*, between the upper and lower boundaries. A typical law $\partial j$ for an interlayer adhesive is shown in Figure 1 (left). In the material sciences, the knowledge of $\partial j$ is crucial for the understanding of the basic failure modes of the composite material.

The adhesive law $\partial j$ is usually determined experimentally using the double cantilever beam test [35] or other destructive testing methods. The result of a typical experiment is shown schematically in Figure 3 from [35], where three probes with different levels of contamination have been exposed. While the intact material shows stable propagation of the crack front (dashed curve), the 10% contaminated specimen shows a typical zig-zag profile (bold solid curve), indicating unstable crack front propagation. Indeed, when reaching the critical load $P = 140$N, the crack starts to propagate. Since by the growth of the crack-elongation, the compliance of the structure increases, the crack propagation slows down and the crack is "caught", i.e., stops at $u_2 = 0.25$mm and the load $P$ in the structure drops from $P = 140$N to $P = 40$N. Thereafter, due to the continuously increased load, the crack starts again to propagate until reaching another critical load level at $P = 90N$ and $u_2 = 5$mm. This phenomenon occurs five to six times, as seen in Figure 3.

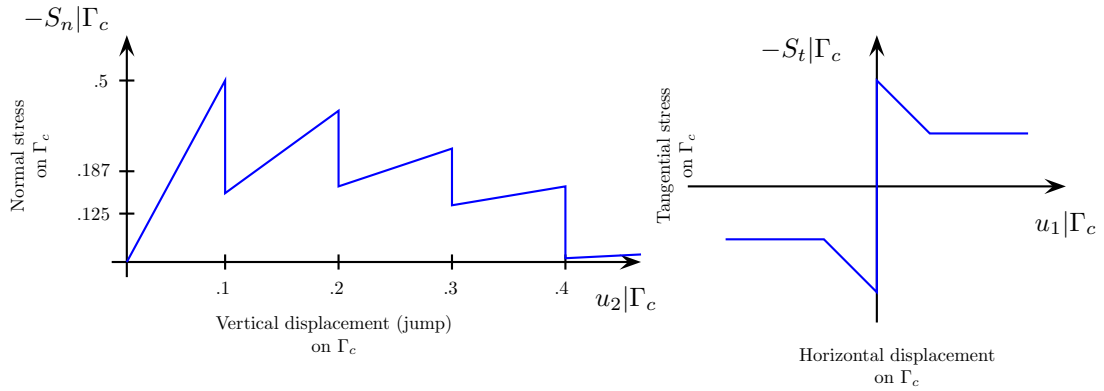



FIGURE 1. Left image shows non-monotone delamination law $\partial j$, leading to an upper-$C^1$ objective. Right image shows non-monotone friction law, leading to a lower-$C^1$ objective

The 50% contaminated specimens (dotted curve) shows micro-cracks that appear at a finer level and are not visible in the Figure 3. The lower level of the adhesive energy, which is represented by the area below the load-displacement curve, indicates now that this specimen is of minor resistance.

Even though the displacement $u_2$ in Figure 3 can only be measured at the crack tip, in order to proceed one now *stipulates* the law $\partial j$ all along $s \in \Gamma_c$ by assuming that the normal stresses $S_n(s)|\Gamma_c$ follow the measured behavior

$$-S_n(s) \in \partial j(s, u_2(s)),\ s \in \Gamma_c. \tag{12}$$

Under this hypothesis one now solves the variational inequality for the unknown displacement field $\mathbf{u} = (u_1, u_2)$, and then validates (12). Note that $S_n(s)|\Gamma_c$ is the *truly* relevant information, as it indicates the action of the destructive forces along $\Gamma_c$, explaining eventual failure of the composite. In current practice in the material sciences, this information cannot be assessed by direct measurement, and is therefore estimated by heuristic formulae [35]. Our approach could be interpreted as one such estimation technique based on mathematical modeling.

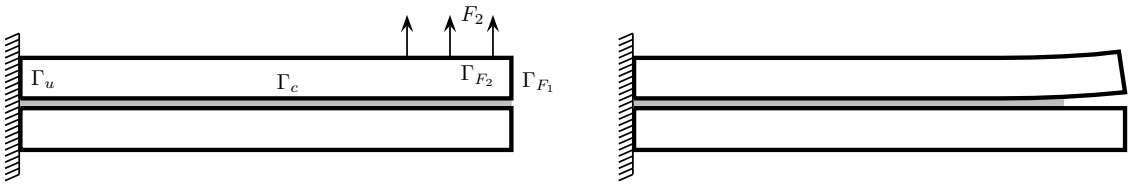



FIGURE 2. Schematic view of cantilever beam testing. Under applied traction force $F_2$ the crack front propagates to the left. In program (16) traction force $F_2$ and crack front length are given, while the corresponding displacement $u$ and reactive forces $-S_n|\Gamma_c$ along the contact boundary $\Gamma_c$ have to be computed

**8.1. Delamination study.** Within the framework of plane linear elasticity we consider a symmetric laminated structure with an interlayer adhesive under loading (see Fig. 2). Because of the symmetry of the structure, it suffices to consider only

the upper half of the specimen, represented by $\Omega \subset \mathbb{R}^2$. The Lipschitz boundary $\Gamma$ of $\Omega$ consists of four disjoint parts $\Gamma_u$, $\Gamma_c$, $\Gamma_{F_1}$ and $\Gamma_{F_2}$. The body is fixed on $\Gamma_u$, i.e.,

$$u_i = 0 \text{ on } \Gamma_u, \quad i = 1, 2.$$

On $\Gamma_{F_1}$ the traction forces $\mathbf{F}$ are constant and given as

$$\mathbf{F} = (0, F_2) \quad \text{on } \Gamma_{F_1}.$$

The part $\Gamma_{F_2}$ is load-free. We adopt standard notation from linear elasticity and introduce the bilinear form of linear elasticity

$$a(\mathbf{u}, \mathbf{v}) = \int_\Omega \varepsilon(\mathbf{u}) \,:\, \sigma(\mathbf{v})\, dx, \tag{13}$$

where $\mathbf{u} = (u_1, u_2)$ is the displacement vector, $\varepsilon(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ the linearized strain tensor, and $\sigma(\mathbf{v}) = \mathbf{C} : \varepsilon(\mathbf{v})$ the stress tensor. Here, $\mathbf{C}$ is the elasticity tensor with symmetric positive $L^\infty$ coefficients. The bilinear form is symmetric and due to the first Korn inequality, coercive. The linear form $\langle \mathbf{g}, \cdot \rangle$ is defined by

$$\langle \mathbf{g}, \mathbf{v} \rangle = F_2 \int_{\Gamma_{F_1}} v_2 \, ds.$$

On the contact boundary $\Gamma_c$ we have the unilateral constraint

$$u_2 \geqslant 0 \quad \text{a.e. on } \Gamma_c$$

and we apply the non-monotone multi-valued adhesive law

$$-S_n(s) \in \partial j(s, u_2(s)) \quad \text{for a.a. } s \in \Gamma_c. \tag{14}$$

Here $S_n = \sigma_{ij} n_j n_i$, where $\mathbf{n} = (n_1, n_2)$ is the outward unit normal vector to $\Gamma_c$.

A typical non-monotone law $\partial j(s, \cdot)$ for delamination, describing the behavior of the adhesive, is shown in Fig. 1. This law is derived from a nonconvex and a nonsmooth locally Lipschitz super-potential $j$ expressed in terms of a minimum function. In particular, $j(s, \cdot)$ is a minimum of four convex quadratic and one linear function.

We also assume that tangental traction can be neglected on $\Gamma_c$, i.e., $S_t(s) = 0$. The weak formulation of the delamination problem is then given by the following hemivariational inequality: Find $\mathbf{u} \in K$ such that

$$a(\mathbf{u}, \mathbf{v} - \mathbf{u}) + \int_{\Gamma_c} j^0(s, u_2(s); v_2(s) - u_2(s))\, ds \geqslant \langle \mathbf{g}, \mathbf{v} - \mathbf{u} \rangle \quad \forall\, \mathbf{v} \in K, \tag{15}$$

where $j^0(s, u; d)$ is the Clarke directional derivative of $j(s, \cdot)$ at $u$ in direction $d$, $K$ is the nonempty, closed convex set of all admissible displacements defined by

$$K = \{\mathbf{v} \in V \,:\, v_2 \geqslant 0 \ \text{ on } \ \Gamma_c\},$$

contained in the function space

$$V = \{\mathbf{v} \in H^1(\Omega; \mathbb{R}^2) \ :\ \mathbf{v} = 0 \ \text{ on } \ \Gamma_u\}.$$

The potential energy of the problem is

$$\Pi(\mathbf{v}) = \frac{1}{2} a(\mathbf{v}, \mathbf{v}) + J(\mathbf{v}) - \langle \mathbf{g}, \mathbf{v} \rangle,$$

where $J : V \to \mathbb{R}$ defined by

$$J(\mathbf{v}) = \int_{\Gamma_c} j(s, v_2(s))\, ds$$

is the term responsible for the nonsmoothness. Using the potential energy, the hemivariational inequality (15) can be transformed to the following nonsmooth, nonconvex constrained optimization problem of the form (1)

$$\begin{array}{ll} \text{minimize} & \Pi(\mathbf{u}) \\ \text{subject to} & \mathbf{u} \in K \end{array} \tag{16}$$

where the objective is upper-$C^1$, because the super-potential $j(s, \cdot)$ is a minimum. In particular, we have an objective of the form (3), where the smooth part $f_s$ comprises $\frac{1}{2}a(\mathbf{v}, \mathbf{v}) - \langle \mathbf{g}, \mathbf{v} \rangle$, while the nonsmooth part $J(\mathbf{v}) = \int_{\Gamma_c} j(s, v_2(s))\, ds$ has the form (3) with a finite index set $I$ once the boundary integral is suitably parametrized.

According to the existence theory in [22], problem (16) has at least one Clarke critical point $\mathbf{u}^*$ satisfying the necessary optimality condition

$$0 \in \partial\, \Pi(\mathbf{u}^*) + N_K(\mathbf{u}^*),$$

where $N_K(\mathbf{u})$ is the normal cone to $K$ at $\mathbf{u}$, and vice versa, by a result in [20] every critical point of $\Pi$ on $K$ is a solution of (15) (see also [19]).

**8.2. Discrete problem.** We consider a regular triangulation $\{\mathcal{T}_h\}$ of $\Omega$, where we first divide $\Omega$ into small squares of size $h$ and then each square by its diagonal into two triangles. To approximate $V$ and $K$ we use a piecewise linear finite element approximation and set

$$V_h = \{v_h \in C(\overline{\Omega}; \mathbb{R}^2) \,:\, v_{h|_T} \in (\mathbf{P}_1)^2,\, \forall\, T \in \mathcal{T}_h,\, v_{h|_{\Gamma_u}} = 0\},$$

$$K_h = \{v_h \in V_h : v_{h2}(s_\nu) \geqslant 0 \quad \forall\, s_\nu \in \overline{\Gamma}_c \backslash \overline{\Gamma}_u\}.$$

Similar to low order finite element approximations of nonsmooth convex contact problems [8, 9], we use the trapezoidal quadrature rule to approximate the functional $J$ by

$$J_h(v_h) = \frac{1}{2} \sum_{s_\nu \in \overline{\Gamma}_c \backslash \overline{\Gamma}_u} |s_\nu s_{\nu+1}| \big[ j(s_\nu, v_{h2}(s_\nu)) + j(s_{\nu+1}, v_{h2}(s_{\nu+1})) \big], \tag{17}$$

where we are summing over the nodes $s_\nu$ on the contact boundary $\overline{\Gamma}_c \backslash \overline{\Gamma}_u$, with $s_{\nu+1}$ being the neighbor of node $s_\nu$ on $\Gamma_c$ in the sense of integration. This can be regrouped as

$$J_h(v_h) = \sum_{s_\nu \in \overline{\Gamma}_c \backslash \overline{\Gamma}_u} c_\nu j(s_\nu, v_{h2}(s_\nu)) = \sum_{s_\nu \in \overline{\Gamma}_c \backslash \overline{\Gamma}_u} c_\nu \min_{i \in I} j_i(s_\nu, v_{h2}(s_\nu))$$

with appropriate weights $c_\nu > 0$. Here, $I$ is the set of zig-zags in the graph of $\partial j$.

The bundle algorithm is applied to minimize the discrete functional

$$\Pi_h(v_h) = \frac{1}{2} a(v_h, v_h) + J_h(v_h) - \langle g, v_h \rangle \quad \text{on} \quad K_h. \tag{18}$$

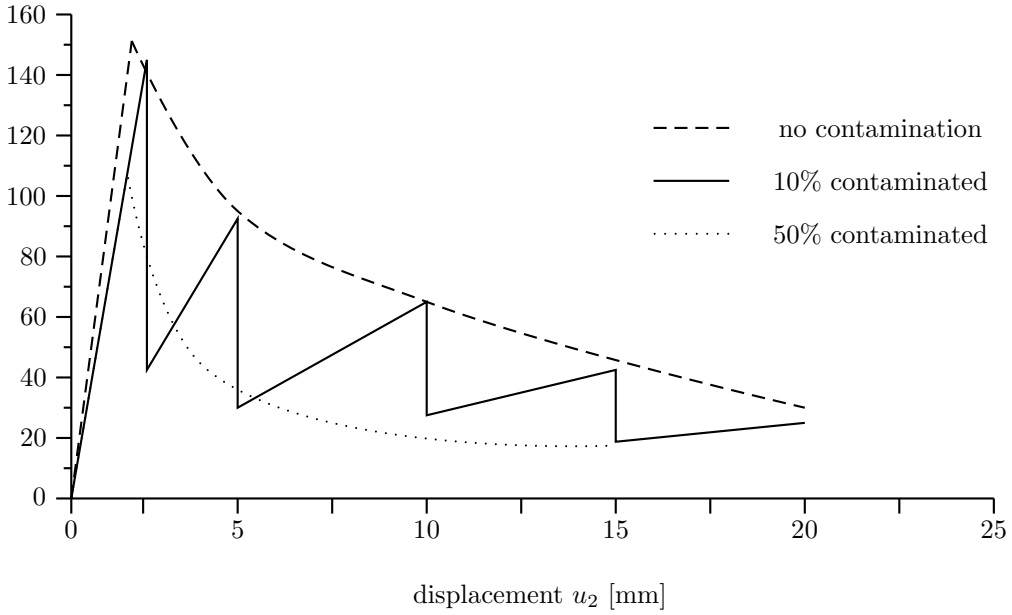



FIGURE 3. Load-displacement curve determined by double cantilever beam test. Dashed curve shows stable behavior for material without contamination. The 10% contaminated specimen (bold solid curve) shows unstable crack growth. After initial linear growth, when the critical load $P = 140$N is reached, the crack starts to propagate. But then the propagation speed slows down, since by the crack the compliance of the specimen increases, and the crack is "caught" at $u_2 = 0.25$mm. The load $P$ drops from $P = 140$N to $P = 40$N. Then, by the constantly applied traction force, there is a linear growth of the load $P$ from $P = 40$N to the critical load $P = 90$N, where the crack propagates again and stops at $u_2 = 5$mm, with the load now reduced to $P = 30$N. The 50% contaminated specimen exhibits micro-cracks not visible at the chosen scale.

Introducing an index set $N$ for the nodes $s_\nu$ on the contact boundary $\overline{\Gamma}_c$, we may pull out the minimum from under the sum, which leads to the expression

$$\Pi_h(v_h) = \frac{1}{2}a(v_h, v_h) + \min_{i(\cdot)\in I^N} \sum_{\nu\in N} c_\nu j_{i(\nu)}(s_\nu, v_{h2}(s_\nu)) - \langle g, v_h\rangle.$$

This is the discrete version of (4), where $\frac{1}{2}a(v_h, v_h) - \langle g, v_h\rangle$ is the smooth term $f_s$, and $J_h$ the nonsmooth part.

While computation of Clarke subgradients is straightforward here, we still have to explain how the matrix $Q = Q(v)$ in the second-order working model (6) is chosen. Discretizing the quadratic form of linear elasticity as $a(v_h, v_h) = v_h^\top \mathbf{A} v_h$ with the symmetric stiffness matrix $\mathbf{A}$, and observing that $\langle g, v_h\rangle = \mathbf{g}^\top v_h$ is linear, we choose $Q(v) = \mathbf{A} + \sum_{\nu\in N} \nabla^2 j_{i(\nu)}(s_\nu, v_h(s_\nu))$, where $i(\nu) \in I$ is one of those indices, where the minimum $\min_{i\in I} j_i(s_\nu, v_{h2}(s_\nu))$ is attained.

For convergence of the lowest-order finite element approximation used here we refer to the results in [27]. Higher-order approximations with no limitation in the

polynomial degree, which lead to nonconforming approximation of unilateral constraints, have only recently been analyzed for monotone contact problems, see [10].

**8.3. Numerical results.** We present numerical results obtained in a delamination simulation with modulus of elasticity $E = 210$ GPa and Poisson ratio $\nu = 0.3$ corresponding to a steel specimen. In all examples we use the benchmark model of [2] with geometrical characteristics $(0, 100) \times (0, 10)$ in [mm] and thickness 5mm. We apply our bundle method to (16) and compare the results to those obtained by the regularization technique in [27, 28]. All computations use piecewise linear functions and the discretization $40 \times 4$ corresponding to $h = 0.25$cm. In this case, the number of the unknowns in the discrete problem (18) is 80.

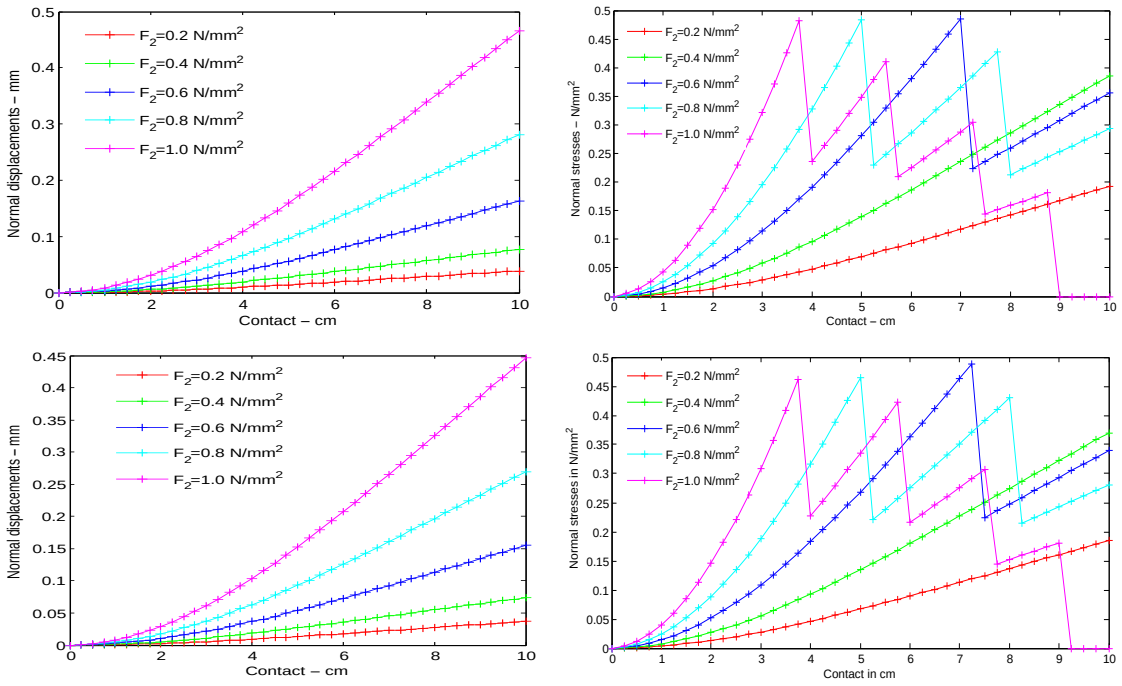



FIGURE 4. Upper: regularization method of [27, 28]. Lower: optimization method. Left image shows vertical displacement $u_2$ for 5 different values of $F_2$. Right image shows vertical component of reactive force along contact boundary for same 5 scenarios

TABLE 1. Regularization. Vertical displacement [mm] at 4 intermediate points for same 5 scenarios

| $F_2[N/mm^2]$ | $u_2(x_1)$ | $u_2(x_2)$ | $u_2(x_3)$ | $u_2(x_4)$ |
|---|---|---|---|---|
| 0.2 | 4.154500e-06 | 1.394500e-05 | 2.601700e-05 | 3.858700e-05 |
| 0.4 | 8.308100e-06 | 2.788800e-05 | 5.202800e-05 | 7.716600e-05 |
| 0.6 | 1.633200e-05 | 5.622700e-05 | 1.080000e-04 | 1.640000e-04 |
| 0.8 | 2.792500e-05 | 9.663100e-05 | 1.860000e-04 | 2.810000e-04 |
| 1.0 | 4.600600e-05 | 1.590000e-04 | 3.080000e-04 | 4.660000e-04 |

TABLE 2. Optimization. Vertical displacement [mm] at four intermediate points for same 5 scenarios

| $F_2[N/mm^2]$ | $u_2(x_1)$ | $u_2(x_2)$ | $u_2(x_3)$ | $u_2(x_4)$ |
|---|---|---|---|---|
| 0.2 | 4.022500e-06 | 1.345400e-05 | 2.499300e-05 | 3.691900e-05 |
| 0.4 | 8.069300e-06 | 2.698800e-05 | 5.013300e-05 | 7.404900e-05 |
| 0.6 | 1.564800e-05 | 5.373900e-05 | 1.030000e-04 | 1.550000e-04 |
| 0.8 | 2.691300e-05 | 9.297200e-05 | 1.790000e-04 | 2.700000e-04 |
| 1.0 | 4.414000e-05 | 1.530000e-04 | 2.940000e-04 | 4.470000e-04 |

TABLE 3. Regularization. Horizontal displacement [mm] at four intermediate points for same 5 scenarios

| $F_2[N/mm^2]$ | $u_2(x_1)$ | $u_2(x_2)$ | $u_2(x_3)$ | $u_2(x_4)$ |
|---|---|---|---|---|
| 0.2 | 1.481900e-06 | 2.251300e-06 | 2.474400e-06 | 2.499500e-06 |
| 0.4 | 2.963600e-06 | 4.502200e-06 | 4.948300e-06 | 4.998500e-06 |
| 0.6 | 5.918500e-06 | 9.400600e-06 | 1.077100e-05 | 1.097500e-05 |
| 0.8 | 1.015200e-05 | 1.625600e-05 | 1.866400e-05 | 1.904000e-05 |
| 1.0 | 1.674400e-05 | 2.690100e-05 | 3.100500e-05 | 3.167000e-05 |

TABLE 4. Optimization. Horizontal displacement [mm] at four intermediate points for same 5 scenarios

| $F_2[N/mm^2]$ | $u_2(x_1)$ | $u_2(x_2)$ | $u_2(x_3)$ | $u_2(x_4)$ |
|---|---|---|---|---|
| 0.2 | 1.432200e-06 | 2.161500e-06 | 2.356100e-06 | 2.368400e-06 |
| 0.4 | 2.872700e-06 | 4.335000e-06 | 4.724700e-06 | 4.748800e-06 |
| 0.6 | 5.663400e-06 | 8.957000e-06 | 1.023200e-05 | 1.041100e-05 |
| 0.8 | 9.777300e-06 | 1.561000e-05 | 1.787700e-05 | 1.822600e-05 |
| 1.0 | 1.606400e-05 | 2.578000e-05 | 2.970700e-05 | 3.034700e-05 |

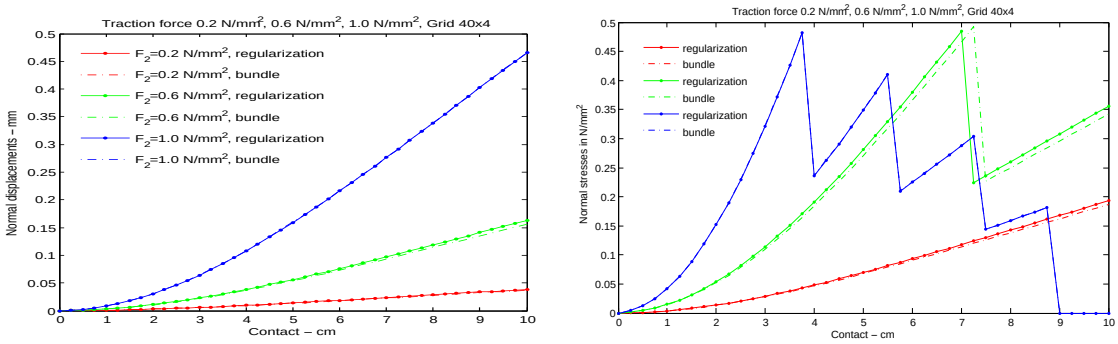



FIGURE 5. Comparison of regularization (bold solid curves) and optimization (dashed) for 3 different values of $F_2$. Left vertical displacement, right reactive force

TABLE 5. Comparison of optimal valued obtained by regularization and optimization

| $F_2[N/m^2]$ | $\Pi_{h\mathrm{reg}}[Nm]$ | $\Pi_{h\mathrm{opt}}$ [Nm] |
|---|---|---|
| 200000 | -1.32894 | -1.29271 |
| 400000 | -2.35224 | -2.30025 |
| 600000 | -3.83972 | -3.74609 |
| 800000 | -5.08164 | -5.05389 |
| 1000000 | -5.66771 | -5.66770 |

## 9. Conclusion

We have presented a bundle method based on the mechanism of downshifted tangents which is suited to optimize upper- and lower-$C^1$ functions. Our method allows to integrate second-order information, if available, and gives a convergence certificate in the sense of subsequences. Every accumulation point of the sequence of serious iterates with an arbitrary starting point is critical. We have successfully applied our method to a delamination problem arising in the material sciences, where upper-$C^1$ functions have to be minimized. Results obtained by optimization were compared to results obtained by the regularization technique of [27, 28], and both methods are in good agreement.

## Acknowledgments

The authors thank H.-J. Gudladt for many useful discussions. The authors were partially supported by Bayerisch-Französisches Hochschulzentrum (BFHZ).

## References

1. P. Apkarian, D. Noll, and O. Prot, *A trust region spectral bundle method for nonconvex eigenvalue optimization*, SIAM J. Optim. **19** (2008), no. 1, 281–306.
2. C. C. Baniotopoulos, J. Haslinger, and Z. Morávková, *Mathematical modeling of delamination and nonmonotone friction problems by hemivariational inequalities*, Appl. Math. **50** (2005), no. 1, 1–25.
3. J. Cullum, W. E. Donath, and P. Wolfe, *The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices*, Nondifferentiable Optimization (M. L. Balinski and P. Wolfe, eds.), Math. Programming Stud., vol. 3, North-Holland Publishing Co., Amsterdam, 1975, pp. 35–55.
4. J. Czepiel, *Proximal bundle method for a simplified unilateral adhesion contact problem of elasticity*, Schedae Informaticae **20** (2011), 115–136.
5. A. Daniilidis and P. Georgiev, *Approximate convexity and submonotonicity*, J. Math Anal. Appl. **291** (2004), no. 1, 292–301.
6. A. Daniilidis and J. Malick, *Filling the gap between lower-$C^1$ and lower-$C^2$ functions*, J. Convex Anal. **12** (2005), no. 2, 315–329.
7. M. Gabarrou, D. Alazard, and D. Noll, *Design of a flight control architecture using a nonconvex bundle method*, Math. Control Signals Syst. **25** (2013), no. 2, 257–290.
8. R. Glowinski, *Numerical methods for nonlinear variational problems*, Springer Ser. Comput. Phys., Springer-Verlag, New York, 1984.
9. J. Gwinner, *Finite-element convergence for contact problems in plane linear elastostatics*, Quart. Appl. Math. **50** (1992), no. 1, 11–25.

10. ———, *hp-FEM convergence for unilateral contact problems with Tresca friction in plane linear elastostatics*, J. Comput. Appl. Math. **254** (2013), 175–184.
11. W. Hare and C. Sagastizábal, *Computing proximal points of nonconvex functions*, Math. Program., Ser. B **116** (2009), no. 1-2, 221–258.
12. J. Haslinger, M. Miettinen, and P. D. Panagiotopoulos, *Finite element method for hemivariational inequalities. Theory, methods and applications*, Nonconvex Optim. Appl., vol. 35, Kluwer Academic Publishers, Dordrecht, 1999.
13. J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms, vol. I. Fundamentals, vol. II. Advanced theory and bundle methods*, Grundlehren Math. Wiss., vol. 305-306, Springer-Verlag, Berlin, 1993.
14. K. C. Kiwiel, *An aggregate subgradient method for nonsmooth convex minimization*, Math. Programming **27** (1983), no. 3, 320–341.
15. M. Kočvara, A. Mielke, and T. Roubíček, *A rate-independent approach to the delamination problem*, Math. Mech. Solids **11** (2006), no. 4, 423–447.
16. C. Lemaréchal, *Bundle methods in nonsmooth optimization*, Nonsmooth Optimization (Proc. IIASA Workshop, Laxenburg, 1977) (C. Lemaréchal and R. Mifflin, eds.), IIASA Proc. Ser., vol. 3, Pergamon, Oxford-Elmsford, 1978, pp. 79–102.
17. C. Lemaréchal and C. Sagastizábal, *Variable metric bundle methods: from conceptual to implementable forms*, Math. Programming, Ser. B **76** (1997), no. 3, 393–410.
18. L. Lukšan and J. Vlček, *A bundle-Newton method for nonsmooth unconstrained minimization*, Math. Programming, Ser. A **83** (1998), no. 3, 373–391.
19. M. M. Mäkelä, M. Miettinen, L. Lukšan, and J. Vlček, *Comparing nonsmooth nonconvex bundle methods in solving hemivariational inequalities*, J. Global Optim. **14** (1999), no. 2, 117–135.
20. M. Miettinen, M. M. Mäkelä, and J. Haslinger, *On numerical solution of hemivariational inequalities by nonsmooth optimization methods*, J. Global Optim. **6** (1995), no. 4, 401–425.
21. R. Mifflin, *A modification and extension of Lemarechal's algorithm for nonsmooth minimization*, Nondifferential and Variational Techniques in Optimization (D. C. Sorensen and R. J.-B. Wets, eds.), Math. Programming Stud., vol. 17, North-Holland Publishing Co., Amsterdam, 1982, pp. 77–90.
22. Z. Naniewicz and P. D. Panagiotopoulos, *Mathematical theory of hemivariational inequalities and applications*, Monogr. Textbooks Pure Appl. Math., vol. 118, Marcel Dekker, Inc., New York, 1995.
23. L. Nesemann and E. P. Stephan, *Numerical solution of an adhesion problem with FEM and BEM*, Appl. Numer. Math. **62** (2012), no. 5, 606–619.
24. D. Noll, *Cutting plane oracles to minimize non-smooth non-convex functions*, Set-Valued Var. Anal. **18** (2010), no. 3-4, 531–568.
25. ———, *Convergence of non-smooth descent methods using the Kurdyka-Łojasiewicz inequality*, J. Optim. Theory Appl. **160** (2014), no. 2, 553–572.
26. D. Noll, O. Prot, and A. Rondepierre, *A proximity control algorithm to minimize nonsmooth and nonconvex functions*, Pac. J. Optim. **4** (2008), no. 3, 571–604.
27. N. Ovcharova, *Regularization methods and finite element approximation of hemivariational inequalities with applications to nonmonotone contact problems*, Cuvillier Verlag, Göttingen, 2012, PhD Thesis, Universität der Bundeswehr München.
28. N. Ovcharova and J. Gwinner, *A study of regularization techniques of nondifferentiable optimization in view of application to hemivariational inequalities*, J. Optim. Theory Appl. **162** (2014), no. 3, 754–778.
29. R. A. Poliquin and R. T. Rockafellar, *Prox-regular functions in variational analysis*, Trans. Amer. Math. Soc. **348** (1996), no. 5, 1805–1838.
30. R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, Springer-Verlag, Berlin, 1998.
31. T. Roubíček, V. Mantic, and C. G. Panagiotopoulos, *A quasistatic mixed-mode delamination model*, Discrete Contin. Dyn. Syst. Ser. S **6** (2013), no. 2, 591–610.
32. A. Ruszczyński, *Nonlinear optimization*, Princeton University Press, Princeton, 2006.
33. H. Schramm and J. Zowe, *A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results*, SIAM J. Optim. **2** (1992), no. 1, 121–152.

34. J. E. Spingarn, *Submonotone subdifferentials of Lipschitz functions*, Trans. Amer. Math. Soc. **264** (1981), no. 1, 77–89.
35. M. Wetzel, J. Holtmannspötter, H.-J. Gudladt, and J. V. Czarnecki, *Sensitivity of double cantilever beam test to surface contamination and surface pretreatment*, Int. J. Adhes. Adhes. **46** (2013), 114–121.
36. J. Zowe, *The BT-algorithm for minimizing a nonsmooth functional subject to linear constraints*, Nonsmooth Optimization and Related Topics (Proc. Int. School Math., Erice, 1988) (F. H. Clarke, V. F. Dem'yanov, and F. Giannessi, eds.), Ettore Majorana Internat. Sci. Ser. Phys. Sci., vol. 43, Plenum Press, New York, 1989, pp. 459–480.

## Résumé

L'optimisation non lisse est une branche active de programmation non linéaire moderne, où l'objectif et les contraintes sont des fonctions continues mais pas nécessairement différentiables. Les sous-gradients généralisés sont disponibles comme un substitut à l'information dérivée manquante, et sont utilisés dans le cadre des algorithmes de descente pour se rapprocher des solutions optimales locales. Sous des hypothèses réalistes en pratique, nous prouvons des certificats de convergence vers les points optimums locaux ou critiques à partir d'un point de départ arbitraire.

Dans cette thèse, nous développons plus particulièrement des techniques d'optimisation non lisse de type faisceaux, où le défi consiste à prouver des certificats de convergence sans hypothèse de convexité. Des résultats satisfaisants sont obtenus pour les deux classes importantes de fonctions non lisses dans des applications, fonctions $C^1$-inférieurement et $C^1$-supérieurement.

Nos méthodes sont appliquées à des problèmes de design dans la théorie du système de contrôle et dans la mécanique de contact unilatéral et en particulier, dans les essais mécaniques destructifs pour la délaminage des matériaux composites. Nous montrons comment ces domaines conduisent à des problèmes d'optimisation non lisse typiques, et nous développons des algorithmes de faisceaux appropriés pour traiter ces problèmes avec succès.



**Mots-clés.** Optimisation non lisse et non convexe · méthode de faisceaux · norme de Hankel · contrôle optimal · placement de structure propre · problème de délaminage.



## Tóm tắt

Tối ưu không trơn là một lĩnh vực năng động của quy hoạch phi tuyến hiện đại, trong đó các hàm mục tiêu và ràng buộc liên tục nhưng không nhất thiết khả vi. Để thay thế cho những thông tin đạo hàm còn thiếu, dưới gradient suy rộng đã xuất hiện và được sử dụng trong khuôn khổ các thuật toán giảm nhằm xấp xỉ các nghiệm tối ưu địa phương. Với những giả thiết thực tế trong vận dụng, chúng tôi chứng minh sự hội tụ của thuật toán đến các điểm tối ưu địa phương hoặc tới hạn từ một điểm khởi tạo bất kì.

Trong luận án này, chúng tôi tập trung phát triển những kỹ thuật tối ưu không trơn dạng bó với yêu cầu đặt ra là chứng minh sự hội tụ không sử dụng tính lồi. Những kết quả thỏa dụng đạt được cho hai lớp hàm không trơn quan trọng trong ứng dụng, đó là các hàm $C^1$-dưới và $C^1$-trên.

Các phương pháp của chúng tôi được áp dụng cho những bài toán thiết kế trong lý thuyết hệ thống điều khiển và cơ học tiếp xúc một phía, đặc biệt là trong thử nghiệm cơ học phá hủy cho sự tách lớp vật liệu composite. Chúng tôi chuyển các vấn đề này về những bài toán tối ưu không trơn điển hình rồi phát triển những thuật toán bó phù hợp để giải quyết chúng một cách hiệu quả.



**Từ khóa.** Tối ưu không trơn không lồi · thuật toán bó · chuẩn Hankel · điều khiển tối ưu · gán cấu trúc riêng · bài toán tách lớp.



## Summary

Nonsmooth optimization is an active branch of modern nonlinear programming, where objective and constraints are continuous but not necessarily differentiable functions. Generalized subgradients are available as a substitute for the missing derivative information, and are used within the framework of descent algorithms to approximate local optimal solutions. Under practically realistic hypotheses we prove convergence certificates to local optima or critical points from an arbitrary starting point.

In this thesis we develop especially nonsmooth optimization techniques of bundle type, where the challenge is to prove convergence certificates without convexity hypotheses. Satisfactory results are obtained for two important classes of nonsmooth functions in applications, lower- and upper-$C^1$ functions.

Our methods are applied to design problems in control system theory and in unilateral contact mechanics and in particular, in destructive mechanical testing for delamination of composite materials. We show how these fields lead to typical nonsmooth optimization problems, and we develop bundle algorithms suited to address these problems successfully.



**Keywords.** Nonconvex and nonsmooth optimization · bundle method · Hankel norm · optimal control · eigenstructure assignment · delamination problem.