ROBUST EIGENSTRUCTURE CLUSTERING BY NONSMOOTH OPTIMIZATION

Minh Ngoc Dao, Dominikus Noll, Pierre Apkarian

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_∞ 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 [1] and Moore [2], 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.

As a remedy we propose to assign the eigenelements (λ_i, v_i, w_i) *simultaneously*. We allow eigenelements (λ_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 [3], where hand-tuning of eigenvalues was shown to improve stability margins of the controlled system. In [4] the same authors elaborate on this idea and suggest a first-order gradient method. In [5, 6], a sequential quadratic programming (SQP) technique with finite-difference gradients was used to improve μ robustness indicators, with eigenvalues and some eigenvectors as decision variables. In [7], 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

[*]Institut de Mathématiques, Université de Toulouse, France.
[†]Department of Mathematics and Informatics, Hanoi National University of Education, Vietnam.
[‡]Control System Department, ONERA, Toulouse, France.

idea is applied to a variety of problems in their monograph [8]. In the same vein, reference [9] exploits the Nelder-Mead direct search method to optimize assignable eigenvalues and eigenvectors, while safeguarding stability of unassigned eigenvalues via constraints. In [10], 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 [8] 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_∞ are handled rigorously by accounting for their nonsmoothness.

Nonsmoothness arises due to the spectral abscissa, and via H_∞-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 [11, 12, 13, 14] 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 [2], 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 [15].

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 λ_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 (λ_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

$$\begin{array}{ll} \text{minimize} & \alpha(A + BKC) \\ \text{subject to} & \left[\begin{array}{c|c} A - \lambda_i I_n & B \\ \hline M_i & N_i \end{array}\right] \left[\begin{array}{c} v_i \\ \hline w_i \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline r_i \end{array}\right] \text{ for } i = 1, \ldots, p \\ & |\operatorname{Re}\lambda_i - \operatorname{Re}\lambda_i^0| \leqslant \delta_i, |\operatorname{Im}\lambda_i - \operatorname{Im}\lambda_i^0| \leqslant \delta_i, i = 1, \ldots, p \\ & K = W(CV)^{-1} \text{ as in (2).} \end{array} \tag{4}$$

Here the $\lambda_i^0 \in \mathbb{C}^-$ are nominal closed-loop poles, and the δ_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 $\Lambda^0 = \{\lambda_1^0, \ldots, \lambda_p^0\} \subset \mathbb{C}^-$ and tolerances δ_i, we now consider the following extension of (4):

$$
\begin{array}{ll}
\text{minimize} & \|T_{w\to z}(K)\| \\
\text{subject to} & \left[\begin{array}{c|c} A-\lambda_i I_n & B \\ \hline M_i & N_i \end{array}\right] \left[\begin{array}{c} v_i \\ \hline w_i \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline r_i \end{array}\right] \text{ for } i = 1, \ldots, p \\
& |\operatorname{Re}\lambda_i - \operatorname{Re}\lambda_i^0| \leqslant \delta_i, |\operatorname{Im}\lambda_i - \operatorname{Im}\lambda_i^0| \leqslant \delta_i, i = 1, \ldots, p \\
& K = K(\lambda, v, w) \text{ as in (2)}
\end{array}
\tag{6}
$$

where λ_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_∞-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 λ_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 Λ^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 (λ^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 λ in $\mathbb{C}$ (see, e.g., [16, 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 λ in the neighborhood of the nominal λ^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 λ_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 λ_i. It follows that a parametrization of the control law (2) in the sense of structured synthesis introduced in [17] 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 λ_i to move in a neighborhood of the nominal λ_i^0. Now if the (v_i^0, w_i^0) are computed from (7) for the nominal value λ_i^0, the (v_i, w_i), depending continuously on λ_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 [15].

If $F_i(\lambda)$ is not guaranteed to have full-column rank in the neighborhood λ^0, the cast in (6) could be modified as follows:

$$\begin{array}{ll} \text{minimize} & \max\left\{\|T_{w\to z}(K)\|, \mu \max\limits_{i=1,\ldots,p} \|F_i(\lambda_i)w_i - r_i\|_2\right\} \\ \text{subject to} & |\operatorname{Re}\lambda_i - \operatorname{Re}\lambda_i^0| \leqslant \delta_i,\ |\operatorname{Im}\lambda_i - \operatorname{Im}\lambda_i^0| \leqslant \delta_i,\ i = 1, \ldots, p. \\ & K = [w_1, \ldots, w_p]\,(C\,[v_1, \ldots, v_p])^{-1} \\ & K \text{ closed-loop stabilizing} \end{array} \tag{8}$$

where μ is a penalty parameter used to weigh the relative importance of robustness or 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

$$\left[\begin{array}{c|c} A - \lambda_i I_n & B_i \\ \hline M_i & P_i \end{array}\right] \left[\frac{v_i}{u_i}\right] = \left[\begin{array}{c} 0 \\ \hline r_i \end{array}\right] - \left[\begin{array}{c} Q_i \\ \hline R_i \end{array}\right] t_i.$$

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 λ_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

$$\left[\frac{v_i}{u_i}\right] = \mathbf{A}_i(\lambda_i)^{-1} \left[\begin{array}{c} -Q_i t_i \\ \hline r_i - R_i t_i \end{array}\right]. \tag{9}$$

The idea is now the same as in the first case. Allow λ_i to move around their nominal values λ_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 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

$$\left[\begin{array}{c} \frac{\partial v_i}{\partial \lambda_i} \\ \hline \frac{\partial w_i}{\partial \lambda_i} \end{array}\right] = \left[\begin{array}{c} (\lambda_i I - A)^{-1} B F_i(\lambda_i)^\dagger M_i - I \\ \hline F_i(\lambda_i)^\dagger M_i \end{array}\right](\lambda_i I - A)^{-2} B F_i(\lambda_i)^\dagger r_i\,. \tag{11}$$

Proof. See Appendix. □

Proposition 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*

$$\begin{aligned} \frac{\partial K}{\partial \lambda_i} &= \left[0 \cdots \left| \begin{array}{c} \frac{\partial u_i}{\partial \lambda_i} \\ 0 \end{array} \right| - KC\frac{\partial v_i}{\partial \lambda_i} \cdots 0\right](CV)^{-1}, \\ \frac{\partial K}{\partial t_{ki}} &= \left[0 \cdots \left| \begin{array}{c} \frac{\partial u_i}{\partial t_{ki}} \\ e_{ki} \end{array} \right| - KC\frac{\partial v_i}{\partial t_{ki}} \cdots 0\right](CV)^{-1}, \end{aligned} \tag{12}$$

where $e_{ki} \in \mathbb{R}^{m-m_i}$ is the vector all of whose components are zero, except the kth component which is one, and

$$\left[\begin{array}{c|c} \frac{\partial v_i}{\partial \lambda_i} & \frac{\partial v_i}{\partial t_{ki}} \\ \hline \frac{\partial u_i}{\partial \lambda_i} & \frac{\partial u_i}{\partial t_{ki}} \end{array}\right] = \left[\begin{array}{cc} I_n & 0_{n\times m_i} \\ \hline 0_{m_i \times n} & I_{m_i} \end{array}\right] \mathbf{A}_i(\lambda_i)^{-1} \left[\begin{array}{c|c} v_i & \\ 0 & -s_{ik} \end{array}\right], \tag{13}$$

with s_{ik} the kth column of $\left[\begin{array}{c} Q_i \\ R_i \end{array}\right]$.

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 λ_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 $\mathbf{P}_{11}^i$ and $\mathbf{P}_{22}^i$ are of size $n \times n$ and $m_i \times m_i$, respectively. Using the Sherman-Woodbury-Morrison formula [18] 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

$$\mathbf{A}_i(\lambda_i)^{-1} = \begin{bmatrix} (I_n - \lambda_i \mathbf{P}_{11}^i)^{-1}\mathbf{P}_{11}^i & (I_n - \lambda_i \mathbf{P}_{11}^i)^{-1}\mathbf{P}_{12}^i \\ \mathbf{P}_{21}^i(I_n - \lambda_i \mathbf{P}_{11}^i)^{-1} & \mathbf{P}_{22}^i + \mathbf{P}_{21}^i \lambda_i (I_n - \lambda_i \mathbf{P}_{11}^i)^{-1}\mathbf{P}_{12}^i \end{bmatrix}.$$

As a consequence, there is only need to compute the inverse of the smaller matrix $(I_n - \lambda_i \mathbf{P}_{11}^i)$ 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 $\lambda^0 = \{\lambda_1^0, \ldots, \lambda_p^0\}$ 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_∞-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 λ_i if $m_i \geqslant m$, and (λ_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 [19, 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 (λ_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 [20], 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 [16]

$$A_{cl}X + XA_{cl}^\top + B_{cl}B_{cl}^\top = 0, \tag{14}$$

$$A_{cl}^\top Y + YA_{cl} + C_{cl}^\top C_{cl} = 0. \tag{15}$$

5.1. The H_∞-norm.

Consider a stable LTI system

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

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_∞-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 [21, Lemma 1], closed-loop stability implies that either $f(K) = f(K,\omega)$ for all ω or $f(K) = f(K,\omega)$ for a finite number of active frequencies $\omega_1,\ldots,\omega_q$. From [17, 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 [20], 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

$$G : \begin{cases} \dot{x} = Ax + Bw \\ z = Cx \end{cases} \tag{16}$$

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 [22, Theorem 3.2] that f is differentiable at each closed-loop stabilizing K, and

$$\nabla f(K) = 2\left(B^\top Y + D_{12}^\top C_{cl}\right) X C^\top + 2B^\top Y B_{cl} D_{21}^\top.$$

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 [23, 14] 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 [14, 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

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

which brings us back to the realm of eigenvalue theory for symmetric matrices. Let $Z := X^{\frac{1}{2}}YX^{\frac{1}{2}}$ 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 [14, 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

$$Z_i(\mathbf{x}) = X_i^{\frac{1}{2}}(\mathbf{x})YX^{\frac{1}{2}} + X^{\frac{1}{2}}Y_i(\mathbf{x})X^{\frac{1}{2}} + X^{\frac{1}{2}}YX_i^{\frac{1}{2}}(\mathbf{x}). \tag{17}$$

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 $h(\mathbf{x}) \leqslant 0$ could represent the stability constraint $\alpha(A + BKC) + \epsilon \leq 0$. 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 λ_i with (7), or (λ_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 [24, Section 2.2.2] in the smooth case, or Polak and Wardi [25] in a nonsmooth setting, and in our own contributions [26, 13, 14], where more details and convergence proofs can be found.

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.

Convergence theory of Algorithm 2 is discussed in [13, 14]. The following result is slightly more general than the main convergence theorem in [13] or [14], but can be obtained based essentially on the same convergence analysis:

Theorem 1. *Suppose f and h in program (21) are lower-C^1 functions in the sense of [27] 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 [13, 14], so our convergence theory applies. Convergence for even larger classes of nonsmooth functions is discussed in [11, 12]. For additional insight into this type of nonconvex bundle method see [26, 11, 12, 28].

7. Control of a launcher in atmospheric flight

We consider attitude control of a satellite launcher in atmospheric flight. The linear model

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}$$

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 linearization of the nonlinear equations [29] about a steady state flight point

$$\begin{array}{lll} U_0 = 88.11 \text{ m/s}, & v_0 = 0.678 \text{ m/s}, & w_0 = -1.965 \text{ m/s}, \\ p_0 = -0.0006 \text{ rad/s}, & q_0 = 0.0026 \text{ rad/s}, & r_0 = 0.0046 \text{ rad/s}, \\ \theta_0 = 8.38^\circ, & \psi_0 = 3.48^\circ, & \phi_0 = 11.99^\circ, \end{array}$$

the procedure being explained in [30]. Numerical data in A, B are gathered in Table 3.

TABLE 1. States definitions

name	meaning
w	vertical velocity (m/s)
q	pitch rate (deg/s)
θ	pitch angle (deg)
v	lateral velocity (m/s)
r	yaw rate (deg/s)
ψ	yaw angle (deg)
p	roll rate (deg/s)
ϕ	roll angle (deg)

TABLE 2. Controls definitions

name	meaning
β_z	deflection of pitch nozzle actuator (deg)
β_y	deflection of yaw nozzle actuator (deg)
β_r	deflection of roll nozzle actuator (deg)

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_θ	-2.11	N_v	$-M_w$
Z_θ	-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_ψ	-2.013	T_q	0.98	Y_ψ	9.47	P_q	0.2078
Z_p	-0.687	T_r	-0.2084	Y_p	-1.965	P_r	0.9782
Z_ϕ	0.399	L_q	0	Y_ϕ	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 (θ, q), (ψ, r), and (ϕ, p).
- Well-damped responses to set-points in θ, ψ, and ϕ, 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.

Tracking performance is incorporated into program (6) by minimizing the tracking error transfer function $T_{w_{\rm ref}\to e}(K)$. Pole placement with integral action is easily formulated using the augmented state-space matrices

$$A_a = \begin{bmatrix} A & 0 \\ -HC & 0 \end{bmatrix}, B_a = \begin{bmatrix} B \\ 0 \end{bmatrix}, C_a = \begin{bmatrix} C & 0 \\ 0 & I_3 \end{bmatrix},$$

where

$$H = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

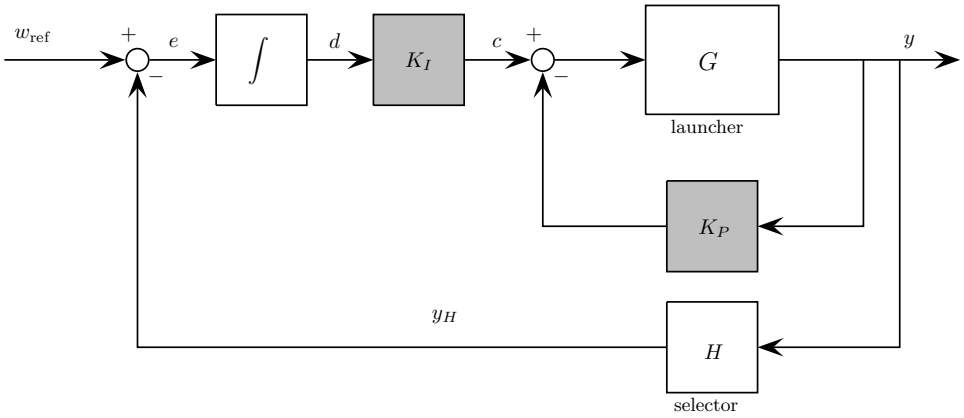

FIGURE 1. Launcher control architecture with MIMO PI-controller

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 ξ, ω 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 $\Lambda = \{\lambda_1^0, \dots, \lambda_9^0\}$, with

$$\lambda_1^0 = -\omega_1 \left(\xi + \jmath\sqrt{1-\xi^2}\right), \lambda_2^0 = -\omega_1 \left(\xi - \jmath\sqrt{1-\xi^2}\right),$$
$$\lambda_3^0 = -\omega_2 \left(\xi + \jmath\sqrt{1-\xi^2}\right), \lambda_4^0 = -\omega_2 \left(\xi - \jmath\sqrt{1-\xi^2}\right),$$
$$\lambda_5^0 = -\omega_3 \left(\xi + \jmath\sqrt{1-\xi^2}\right), \lambda_6^0 = -\omega_3 \left(\xi - \jmath\sqrt{1-\xi^2}\right),$$
$$\lambda_7^0 = -3.5,\ \lambda_8^0 = -4,\ \lambda_9^0 = -4.5.$$

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_∞	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_∞-, 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

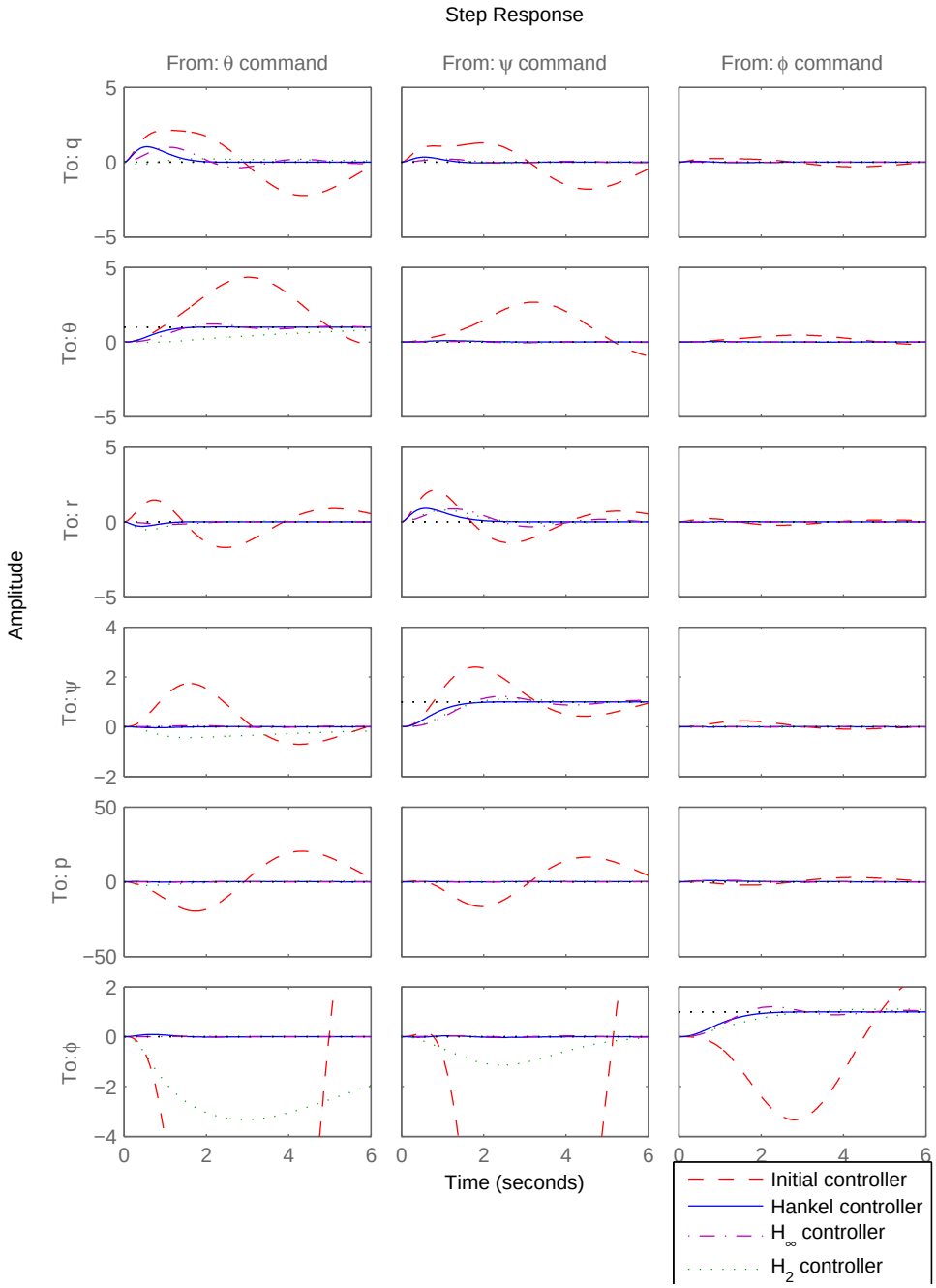

FIGURE 2. Control of a launcher, study 1. Initial and final controllers obtained respectively by standard and optimized eigenstructure assignment in the case where eigenvectors are not structured ($m_i = 0$). Decoupling is improved for each norm

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 ψ and ϕ. The eigenvector v_3 and its conjugate

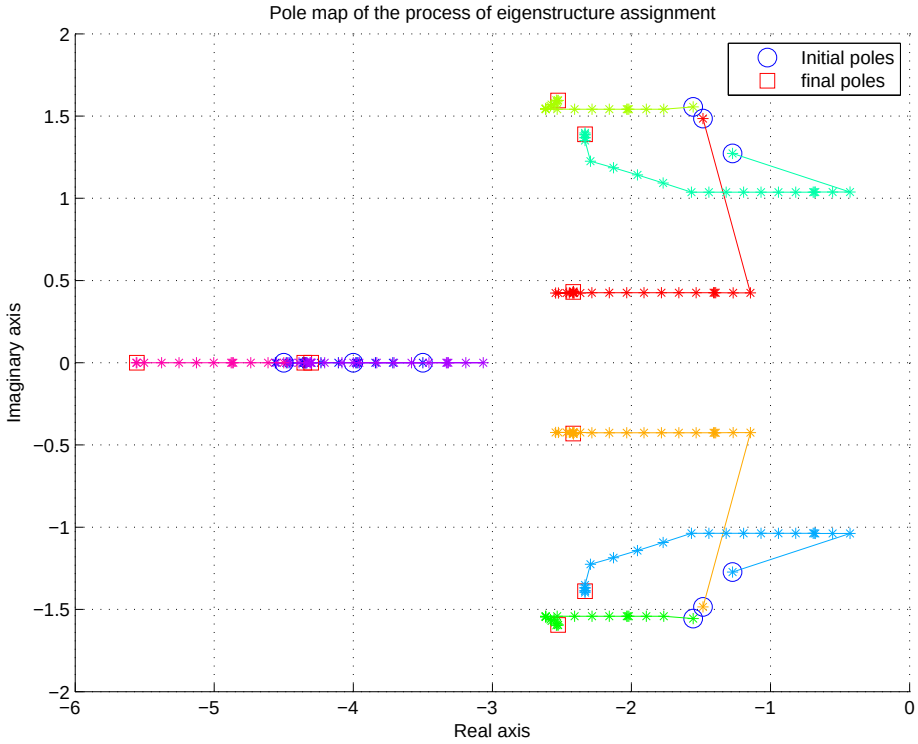

FIGURE 3. Control of launcher, study 1. Itineraries of closed-loop poles in optimized eigenstructure assignment based on the Hankel program (6)

v_4 have entries 0 relative to θ and ϕ. The eigenvectors v_5 and its conjugate v_6 have zero entries in the rows associated with θ and ψ. For the real modes, the eigenvectors are chosen as

$$\begin{aligned} v_7 &= [* \ * \ 1 \ * \ * 0 \ * \ 0 \ * \ * *]^\top, \\ v_8 &= [* \ * \ 0 \ * \ * 1 \ * \ 0 \ * \ * *]^\top, \\ v_9 &= [* \ * \ 0 \ * \ * 0 \ * \ 1 \ * \ * *]^\top. \end{aligned}$$

These structural constraints define the matrices M_i, N_i of (3) in each case. We have again tested the Hankel, H_∞ 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.

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°

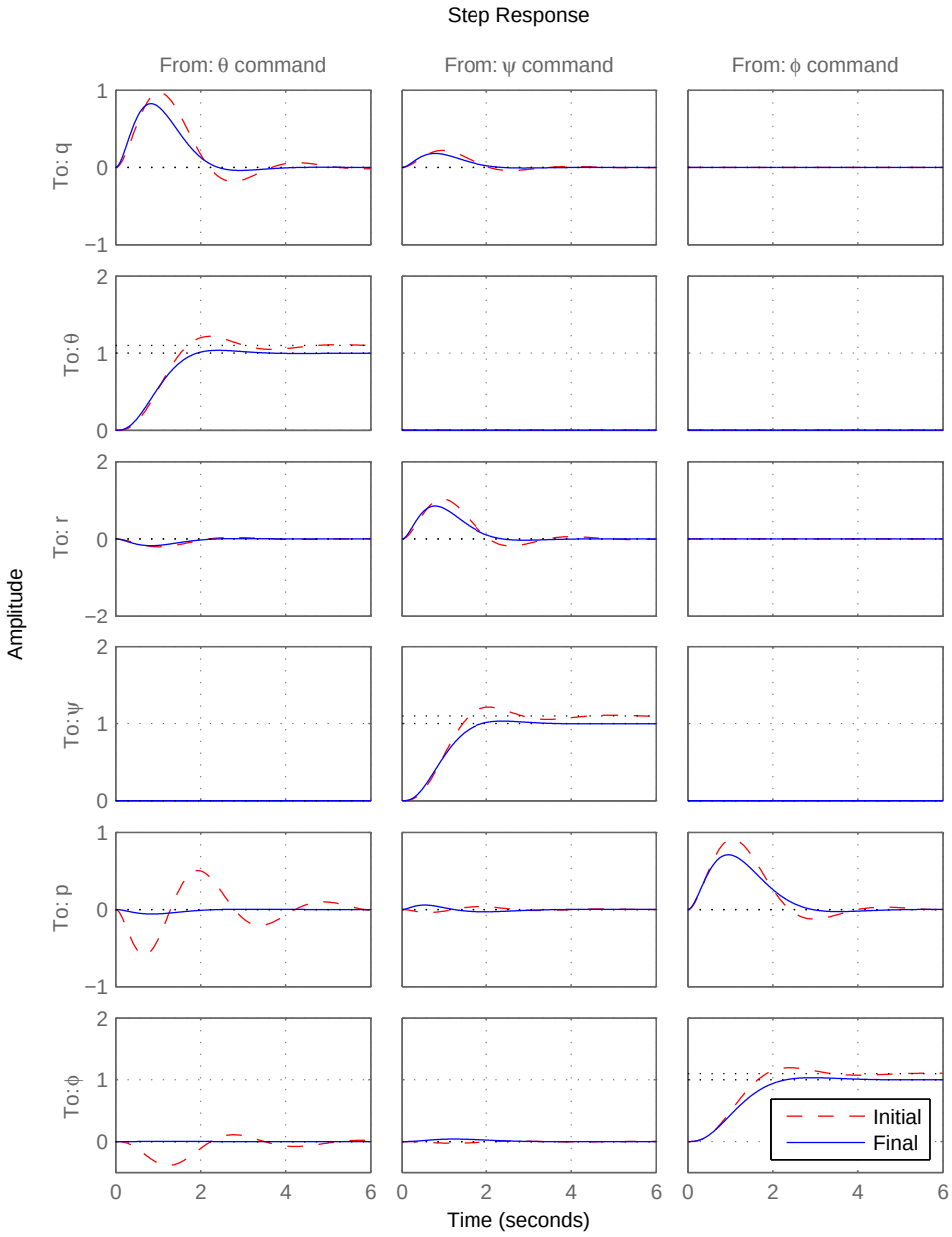

FIGURE 4. Control of launcher, study 2. Initial and final controller obtained respectively by standard and optimized eigenstructure assignment based on Hankel program with $m_i = m$ or $m_i = m - 1$

(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 δ_t the tailplane deflection and δ_{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 the air

TABLE 5. States of the longitudinal model

name	meaning
q	pitch rate
θ	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

speed. Data borrowed from [31] are given as

$$A = \begin{bmatrix} -0.9825 & 0 & -0.0007 & -0.0161 & 0 & -2.4379 & 0.5825 \\ 1 & 0 & 0 & 0 & & 0 & 0 \\ -2.1927 & -9.7758 & -0.0325 & 0.0743 & 0 & 0.1836 & 19.6200 \\ 77.3571 & -0.7674 & -0.2265 & -0.6683 & 0 & -6.4785 & 0 \\ 0 & -79.8667 & -0.0283 & 0.9996 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -6.6667 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -6.6667 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 6.6667 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6.6667 \end{bmatrix}^\top,$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.2661 & 0 & -0.0231 & -0.0681 & 0 & -0.6604 & 0 \\ 0 & -79.8667 & -0.0283 & 0.9996 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0.9996 & 0.0290 & 0 & 0 & 0 \end{bmatrix}.$$

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 $\|T(P_{\rm perf}, K^*)\|_H = 0.6270$, while the initial controller K^0 obtained by standard assignment had $\|T(P_{\rm perf}, K^0)\|_H = 1.5041$. The closed-loop eigenvalues returned by the algorithm are

$$\begin{aligned}
&\lambda_{1,2} = -0.8 \pm \jmath 0.8,\\
&\lambda_{3,4} = -0.35 \pm \jmath 0.05,\\
&\lambda_5 = -0.3,\ \lambda_6 = -0.05,\ \lambda_7 = -0.37,
\end{aligned}$$

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.

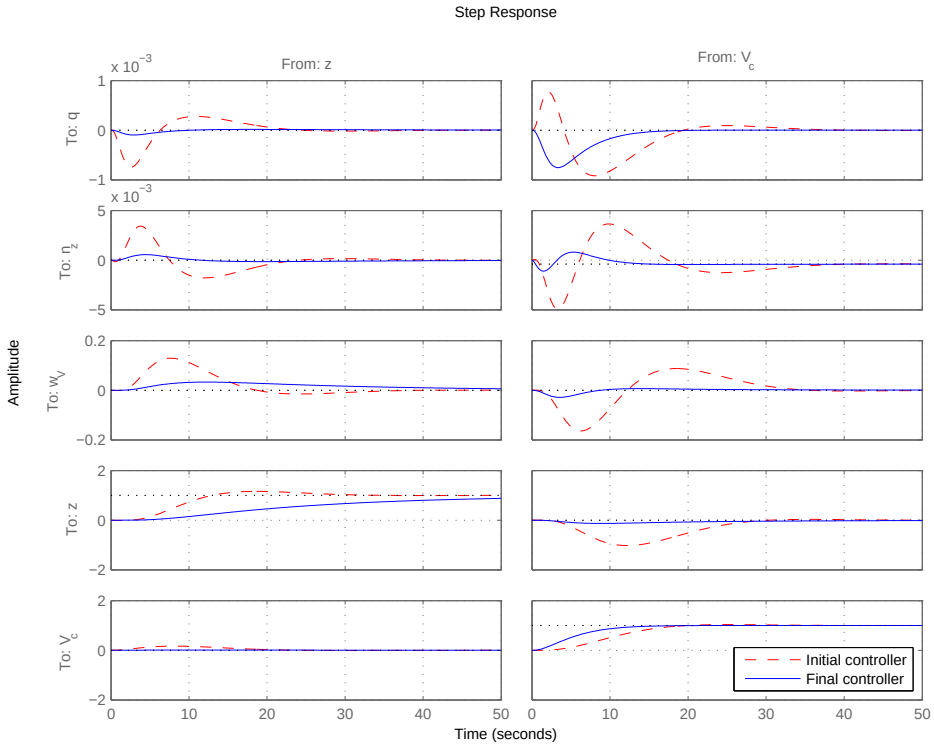

FIGURE 5. Aircraft attitude control. Responses to a step command in altitude and in air speed. Optimal controller computed by optimized eigenstructure assignment ($m_i = 1$, $m = 2$) reduces coupling

9. Conclusion

We have presented a new approach to partial eigenstructure assignment in output feedback control in which the eigenelements (λ, v, w) are allowed to move *simultaneously* in a neighborhood of their nominal values (λ^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_∞, 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 1 and 2. Let us start by discussing the case $m_i \geqslant m$. Derivatives of w_i with respect to λ_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 λ_i is then readily derived by exploiting 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

$$\frac{\partial (CV)^{-1}}{\partial \lambda_i} = -(CV)^{-1} C \left[0 \cdots 0 \; \frac{\partial v_i}{\partial \lambda_i} \; 0 \cdots 0 \right] (CV)^{-1}.$$

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 $V = [v_1 \ldots v_p]$ and

$$W = \left[w_1 \cdots \left| \begin{matrix} u_i \\ t_i \end{matrix} \right| \cdots w_p \right] .$$

□

References

[1] W. M. Wonham, "On pole assignment in multi-input controllable linear systems," *IEEE Trans. Automatic Control*, vol. AC-12, pp. 660–665, 1967.

[2] B. C. Moore, "On the flexibility offered by state feedback in multivariable systems beyond closed loop eigenvalue assignment," *IEEE Trans. Automat. Control*, vol. AC-21, pp. 689–692, 1976.

[3] K. M. Sobel and E. Y. Shapiro, "A robust pitch pointing control law," in *Proc. IEEE Conf. on Decision and Control*, San Antonio, December 1983, pp. 1088–1093.

[4] ——, "Flight control examples of robust eigenstructure assignment," in *Proc. IEEE Conf. on Decision and Control*, Los Angeles, December 1987, pp. 1290–1291.

[5] P. Apkarian, "Structured stability robustness improvement by eigenspace techniques: a hybrid methodology," *AIAA Guid., Nav. and Control Conf.*, vol. 12, no. 2, pp. 162–168, February 1989.

[6] J. C. Morris, P. Apkarian, and J. C. Doyle, "Synthesizing robust mode shapes with μ and implicit model following," in *Proc. IEEE Conf. on Control Applications*, Dayton, September 1992, pp. 1018–1023.

[7] R. J. Patton and G. P. Liu, "Robust control design via eigenstructure assignment, genetic algorithms and gradient-based optimisation," *IEE P-Contr. Theor. Ap.*, vol. 141, no. 3, pp. 202–208, May 1994.

[8] G. P. Liu and R. J. Patton, *Eigenstructure Assignment for Control System Design.* Chichester: John Wiley & Sons, Inc., 1998.

[9] 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.*, vol. 25, no. 2, pp. 866–876, May 2010.

[10] J.-F. Magni, "A toolbox for robust modal control design (RMCT)," in *Proc. IEEE Internat. Symposium on Computer-Aided Control System Design*, Anchorage, September 2000, pp. 202–207.

[11] D. Noll, O. Prot, and A. Rondepierre, "A proximity control algorithm to minimize nonsmooth and nonconvex functions," *Pac. J. Optim.*, vol. 4, no. 3, pp. 571–604, 2008.

[12] D. Noll, "Cutting plane oracles to minimize non-smooth non-convex functions," *Set-Valued Var. Anal.*, vol. 18, no. 3-4, pp. 531–568, 2010.

[13] M. Gabarrou, D. Alazard, and D. Noll, "Design of a flight control architecture using a non-convex bundle method," *Math. Control Signals Syst.*, vol. 25, no. 2, pp. 257–290, 2013.

[14] M. N. Dao and D. Noll, "Minimizing memory effects of a system," *Math. Control Signals Syst.*, 2014, doi: 10.1007/s00498-014-0135-9.

[15] K. N. Sobel, E. Y. Shapiro, and A. N. Andry, "Eigenstructure assignment," *Internat. J. Control*, vol. 59, no. 1, pp. 13–37, 1994.

[16] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. New Jersey: Prentice Hall, 1996.

[17] P. Apkarian and D. Noll, "Nonsmooth H_∞ synthesis," *IEEE Trans. Automat. Control*, vol. 51, no. 1, pp. 71–86, 2006.

[18] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2002.

[19] A. Hjørungnes, *Complex-valued matrix derivatives. With applications in signal processing and communications*. Cambridge: Cambridge University Press, 2011.

[20] F. H. Clarke, "Generalized gradients of Lipschitz functionals," *Adv. in Math.*, vol. 40, no. 1, pp. 52–67, 1981.

[21] V. Bompart, D. Noll, and P. Apkarian, "Second-order nonsmooth optimization for H_∞ synthesis," *Numer. Math.*, vol. 107, no. 3, pp. 433–454, 2007.

[22] T. Rautert and E. W. Sachs, "Computational design of optimal output feedback controllers," *SIAM J. Optim.*, vol. 7, no. 3, pp. 837–852, 1997.

[23] K. Glover, "All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds," *Internat. J. Control*, vol. 39, no. 6, pp. 1115–1193, 1984.

[24] E. Polak, *Optimization: Algorithms and Consistent Approximations*, ser. Appl. Math. Sci. New York: Springer-Verlag, 1997, vol. 124.

[25] E. Polak and Y. Wardi, "Nondifferentiable optimization algorithm for designing control systems having singular value inequalities," *Automatica–J. IFAC*, vol. 18, no. 3, pp. 267–283, 1982.

[26] P. Apkarian, D. Noll, and A. Rondepierre, "Mixed H_2/H_∞ control via nonsmooth optimization," *SIAM J. Control Optim.*, vol. 47, no. 3, pp. 1516–1546, 2008.

[27] J. E. Spingarn, "Submonotone subdifferentials of Lipschitz functions," *Trans. Amer. Math. Soc.*, vol. 264, no. 1, pp. 77–89, 1981.

[28] D. Noll, "Convergence of non-smooth descent methods using the Kurdyka-Łojasiewicz inequality," *J. Optim. Theory Appl.*, vol. 160, no. 2, pp. 553–572, 2014.

[29] D. McLean, *Automatic Flight Control Systems*. London: Prentice Hall, 1990.

[30] A. L. Greensite, *Elements of Modern Control Theory*. New York: Spartan Books, 1970.

[31] Flight Mechanics Action Group 08, "Robust flight. Control design challenge problem formulation and manual: the research civil aircraft model (RCAM)," Department of Electrical Engineering, Linkoping University, Technical Report GARTEUR/TP-088-3, June 15, 1995, version 1.