Optimized Eigenstructure Assignment

Ngoc Minh Dao[1,2], Dominikus Noll[1] and Pierre Apkarian[3]
[1]*Institut de Mathématiques, Université de Toulouse, Toulouse, France*
[2]*Department of Mathematics and Informatics, Hanoi National University of Education, Hanoi, Vietnam*
[3]*ONERA, Department of System Dynamics, Toulouse, France*

Keywords: Eigenstructure Assignment, Output Feedback Control, Nonlinear Optimization, Hankel Norm.

Abstract: This paper considers the problem of eigenstructure assignment for output feedback control. We introduce a new method for partial eigenstructure assignment, which allows to place the eigenelements (λ_i, v_i, w_i) simultaneously. This is possible by a combination of linear algebra and nonlinear optimization techniques. The advantage of the new approach is illustrated through the control of a launcher in atmospheric flight.

1 INTRODUCTION

Eigenstructure assignment has been shown to be a powerful controller design tool in the aerospace sector and in other high technology fields. This approach aims at shaping the responses of the closed-loop system to certain input signals by way of three mechanisms. The placement of closed-loop modes in order to arrange satisfactory decay rates, the choice of suitable eigenvectors to shape specific responses, and the possibility to decide to what extent initial conditions contribute to these responses.

In this paper we focus on the design of output feedback control laws, where only partial eigenstructure assignment or pole placement can be expected. Here the standard approach to first selecting a partial set of closed-loop modes and then using the remaining degrees of freedom to shape the corresponding closed-loop eigenvectors, may fail to stabilize the system in closed-loop, as the remaining closed-loop modes cannot be influenced directly.

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, 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 $v_i \in \mathbb{R}^n$, $w_i \in \mathbb{R}^m$, $i = 1, \ldots, p$ leads to a control law

$$K = [w_1, \ldots, w_p]\,(C[v_1, \ldots, v_p])^{-1} \tag{2}$$

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 $\mathrm{span}\{v_1, \ldots, v_p\} \cap \ker(C) = \{0\}$.

In case $m > 1$ it is possible to achieve more. One may then shape the v_i, respectively w_i, e.g. by arranging $v_{ij} = 0$ or $w_{ik} = 0$ for certain j, k. This can be expressed by linear equations

$$\left[\begin{array}{c|c} A - \lambda_i I_n & B \\ \hline M_i & N_i \end{array}\right] \left[\begin{array}{c} v_i \\ \hline w_i \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline r_i \end{array}\right], \tag{3}$$

where $M_i \in \mathbb{R}^{m_i \times n}$, $N_i \in \mathbb{R}^{m_i \times m}$, $r_i \in \mathbb{R}^{m_i}$, $m_i \geqslant 0$, $i = 1, \ldots, p$, leaving at least one degree of freedom. This is referred to as partial eigenstructure assignment.

The traditional approach in eigenstructure assignment consists in choosing the set $\Lambda \subset \mathbb{C}^-$, and then adding the desired structural constraints on the eigenvectors v_i, w_i, using the remaining degrees of freedom. However, fixing the λ_i may be too restrictive for the second step, because we should not forget that partial eigenvalue placement does not guarantee stability in closed-loop, so that some post-processing may be required, which often leads to unsatisfactory trial-and-error. Greater flexibility in the design could be

Minh Dao N., Noll D. and Apkarian P..
Optimized Eigenstructure Assignment.
DOI: 10.5220/0004594903070314
In *Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2013), pages 307-314
ISBN: 978-989-8565-70-9

achieved by moving (λ_i, v_i, w_i) simultaneously. This may in particular be achieved by optimization if (3) is used as a constraint, under which closed-loop stability or performance are improved.

2 PROBLEM STATEMENT

We discuss the problem of partial eigenstructure assignment for a static feedback output controller. Consider a linear time-invariant plant P in standard form

$$P: \begin{array}{rcccccc} \dot{x} = & Ax & + & B_1 w & + & Bu \\ z = & C_1 x & + & D_{11} w & + & D_{12} u \\ y = & Cx & + & D_{21} w & + & D_{22} u \end{array}$$

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. Assuming without loss that $D_{22} = 0$, let $u = Ky$ be a static output feedback control law for the open-loop plant P, and let $T_{wz}(K)$ denote the closed-loop performance channel $w \to z$. Then $T_{wz}(K)$ has the state-space representation

$$\begin{array}{lll} \dot{x} & = (A + BKC)x & + (B_1 + BKD_{21})w \\ z & = (C_1 + D_{12}KC)x & + (D_{11} + D_{12}KD_{21})w. \end{array}$$

Given a self-conjugate eigenvalues set $\Lambda^0 = \{\lambda_1^0, \ldots, \lambda_p^0\}$ and tolerances δ_i, we consider the optimization program

$$\begin{array}{ll} \text{minimize} & \|T_{wz}(K)\| \\ \text{subject to} & \left[\begin{array}{c|c} A - \lambda_i I_n & B \\ \hline M_i & N_i \end{array}\right] \left[\begin{array}{c} v_i \\ \hline w_i \end{array}\right] = \begin{bmatrix} 0 \\ r_i \end{bmatrix} \\ & |\lambda_i - \lambda_i^0| \leqslant \delta_i, i = 1, \ldots, p \\ & K = K(\lambda, v, w) \text{ as in (2)} \end{array} \qquad (4)$$

where λ_i^0 are nominal closed-loop pole positions and (3) as above conveys additional structural constraints on v_i, w_i. This is now a parametrization of the control law (2) in the sense of structured synthesis introduced in (Apkarian and Noll, 2006). The cost function $\|T_{wz}(K)\|$ in (4) 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 $\|\cdot\|_H$, to which special attention will be given here. One generally expects that $\|T_{wz}(K)\| < \infty$ 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 (4), where $\varepsilon > 0$ is some small threshold, and where we recall the definition of the spectral abscissa of a matrix

$$\alpha(M) = \max\{\mathrm{Re}(\lambda) : \lambda \text{ eigenvalue of } M\}.$$

Altogether, we now establish the following algorithm for partial eigenstructure assignment.

3 HANKEL NORM AND ITS CLARKE SUBGRADIENTS IN CLOSED-LOOP

Consider a stable LTI system

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

with state $x \in \mathbb{R}^n$, 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 $t \leqslant T$, 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 the definition of the Hankel norm of the system G:

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

Algorithm 1: Optimized 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, and K^*.

1: **Nominal assignment**. Perform standard eigenstructure assignment based on Λ^0 and structural constraints M_i, N_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 $K^0 = \left[w_1^0, \ldots, w_p^0\right] \left(C\left[v_1^0, \ldots, v_p^0\right]\right)^{-1}$.

2: **Stability and performance**. If K^0 assures closed-loop stability and performance $\|T_{wz}(K^0)\|$, stop the algorithm. Otherwise, goto step 3.

3: **Tolerances**. Allow tolerances $|\lambda_i - \lambda_i^0| \leqslant \delta_i$, $i = 1, \ldots, p$.

4: **Parametric assignment**. Solve the optimization program

$$\begin{array}{ll} \min & \|T_{wz}(K)\| \\ \text{s.t.} & \left[\begin{array}{c|c} A - \lambda_i I_n & B \\ \hline M_i & N_i \end{array}\right] \left[\begin{array}{c} v_i \\ \hline w_i \end{array}\right] = \begin{bmatrix} 0 \\ r_i \end{bmatrix} \\ & |\lambda_i - \lambda_i^0| \leqslant \delta_i, i = 1, \ldots, p. \\ & K = [w_1, \ldots, w_p]\,(C[v_1, \ldots, v_p])^{-1} \\ & K \text{ closed-loop stabilizing} \end{array} \qquad (5)$$

using (λ^0, v^0, w^0) as initial seed.

5: **Synthesis**. Return optimal $\Lambda = \{\lambda_1, \ldots, \lambda_p\}$, v_i, w_i, and K^*.

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 $\|T_{wz}(K)\|_H$ therefore reduces the ringing in the system.

In order to solve program (5) we will have to compute function values and subgradients of the function $f(\mathbf{x}) = \|T_{wz}(K(\mathbf{x}))\|_H^2$, where $\mathbf{x}$ represents the tunable parameters $\mathbf{x} = (\lambda, v, w)$. Introducing the notation

$$A_c = A + BKC,\ B_c = B_1 + BKD_{21},$$
$$C_c = C_1 + D_{12}KC,\ D_c = D_{11} + D_{12}KD_{21}$$

for the closed-loop, and assuming for the time being that $D_c = D_{11}$ does not explicitly depend on K, a hypothesis which can be arranged e.g. by the standard assumption that $D_{12} = 0$ or $D_{21} = 0$, we have

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

where λ_1 denotes the maximum eigenvalue of a symmetric or Hermitian matrix, and $X(\mathbf{x})$ and $Y(\mathbf{x})$ are the controllability and observability Gramians that can be obtained from the Lyapunov equations

$$A_cX + XA_c^\top + B_cB_c^\top = 0, \quad (6)$$
$$A_c^\top Y + YA_c + C_c^\top C_c = 0. \quad (7)$$

Notice that despite the symmetry of X and Y the product XY needs not be symmetric, but stability of A_c in closed-loop guarantees $X \succ 0$, $Y \succ 0$ in (6), (7), 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.

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 $\mathbb{S}_m$ the space of $m \times m$ symmetric matrices and define

$$\mathbb{B}_m := \{X \in \mathbb{S}_m : X \succeq 0, \mathrm{Tr}(X) = 1\}.$$

Setting $Z := X^{\frac{1}{2}}YX^{\frac{1}{2}}, Z_i(\mathbf{x}) := \partial Z(\mathbf{x})/\partial \mathbf{x}_i$ and taking Q to be a matrix whose columns form an orthonormal basis of the eigenspace of dimension ν associated with $\lambda_1(Z)$, then according to (Overton, 1992, Theorem 3), the Clarke subdifferential of f at $\mathbf{x}$ consists of all subgradients g_U of the form

$$g_U = (\mathrm{Tr}(Z_1(\mathbf{x})^\top QUQ^\top), \ldots, \mathrm{Tr}(Z_{n_\mathbf{x}}(\mathbf{x})^\top QUQ^\top))^\top,$$

where $U \in \mathbb{B}_\nu$, and $n_\mathbf{x}$ is the number of coordinates of $\mathbf{x}$. On the other hand, denoting by D_KF the derivate of F with respect to K, we have

$$Z_i(\mathbf{x}) = D_KZ(\mathbf{x})K_i(\mathbf{x})$$
$$= \varphi_iYX^{\frac{1}{2}} + X^{\frac{1}{2}}\psi_iX^{\frac{1}{2}} + X^{\frac{1}{2}}Y\varphi_i, \quad (8)$$

where $K_i(\mathbf{x}) := \partial K(\mathbf{x})/\partial \mathbf{x}_i$, $\varphi_i := D_KX^{\frac{1}{2}}K_i(\mathbf{x})$, $\psi_i := D_KYK_i(\mathbf{x})$. From (6) and (7), and on putting $\phi_i := D_KXK_i(\mathbf{x})$, we obtain

$$A_c\phi_i + \phi_iA_c^\top = -BK_i(\mathbf{x})CX - X(BK_i(\mathbf{x})C)^\top$$
$$- BK_i(\mathbf{x})D_{21}B_c^\top - B_c(BK_i(\mathbf{x})D_{21})^\top, \quad (9)$$
$$A_c^\top\psi_i + \psi_iA_c = -(BK_i(\mathbf{x})C)^\top Y - YBK_i(\mathbf{x})C$$
$$- (D_{12}K_i(\mathbf{x})C)^\top C_c - C_c^\top D_{12}K_i(\mathbf{x})C, \quad (10)$$

by using the fact that $D_KA_cK_i(\mathbf{x}) = BK_i(\mathbf{x})C$, $D_KB_cK_i(\mathbf{x}) = BK_i(\mathbf{x})D_{21}$, $D_KC_cK_i(\mathbf{x}) = D_{21}K_i(\mathbf{x})C$. Since $X^{\frac{1}{2}}X^{\frac{1}{2}} = X$,

$$X^{\frac{1}{2}}\varphi_i + \varphi_iX^{\frac{1}{2}} = \phi_i. \quad (11)$$

Altogether, we have the following Algorithm 2 to compute subgradients of f at $\mathbf{x}$.

Algorithm 2: Computing subgradients.

Input: $\mathbf{x} \in \mathbb{R}^{n_\mathbf{x}}$. **Output:** $g \in \partial f(\mathbf{x})$.

1: Compute $K_i(\mathbf{x}) = \partial K(\mathbf{x})/\partial \mathbf{x}_i, i = 1, \ldots, n_\mathbf{x}$ and X, Y solutions of (6), (7), respectively.
2: Compute $X^{\frac{1}{2}}$ and $Z = X^{\frac{1}{2}}YX^{\frac{1}{2}}$.
3: For $i = 1, \ldots, n_\mathbf{x}$ compute ϕ_i and ψ_i solutions of (9) and (10), respectively.
4: For $i = 1, \ldots, n_\mathbf{x}$ compute φ_i solution of (11) and $Z_i(\mathbf{x})$ using (8).
5: Determine a matrix Q whose columns form an orthonormal basis of the eigenspace of dimension ν associated with $\lambda_1(Z)$.
6: Pick $U \in \mathbb{B}_\nu$, and return

$$(\mathrm{Tr}(Z_1(\mathbf{x})^\top QUQ^\top), \ldots, \mathrm{Tr}(Z_{n_\mathbf{x}}(\mathbf{x})^\top QUQ^\top))^\top,$$

a subgradient of f at $\mathbf{x}$.

4 PROXIMITY CONTROL ALGORITHM FOR NON-SMOOTH FUNCTIONS

We describe here our algorithm to solve program (5). More generally, we consider an abstract constrained optimization program of the form

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & c(\mathbf{x}) \leqslant 0 \end{array} \quad (12)$$

where $\mathbf{x} \in \mathbb{R}^{n_\mathbf{x}}$ is the decision variable, and f and c are locally Lipschitz but potentially non-smooth and non-convex functions. Expanding on an idea in (Polak, 1997, 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 (12) or not. If $\mathbf{x}$ is infeasible, meaning $c(\mathbf{x}) > 0$, then the right hand term in the expression of $F(\cdot,\mathbf{x})$ is active at $\mathbf{x}$, whereas the left hand term equals $-\nu c(\mathbf{x}) < 0$ at $\mathbf{x}$. Reducing $F(\cdot,\mathbf{x})$ below its value 0 at the current $\mathbf{x}$ therefore reduces constraint violation. If $\mathbf{x}$ is feasible, meaning $c(\mathbf{x}) \leqslant 0$, then the left hand term in $F(\cdot,\mathbf{x})$ becomes dominant, so reducing $F(\cdot,\mathbf{x})$ below its current value 0 at $\mathbf{x}$ now reduces f, while maintaining feasibility, and where the true optimization of f takes place.

Observe that if $\mathbf{x}^*$ is a local minimum of program (12), it is also a local minimum of $F(\cdot,\mathbf{x}^*)$, and then $0 \in \partial_1 F(\mathbf{x}^*,\mathbf{x}^*)$. The symbol ∂_1 here stands for the Clarke subdifferential with respect to the first variable. Indeed, if $\mathbf{x}^*$ is a local minimum of (12) then $c(\mathbf{x}^*) \leqslant 0$, and so for $\mathbf{y}$ in a neighborhood of $\mathbf{x}^*$ we have

$$\begin{aligned} F(\mathbf{y},\mathbf{x}^*) &= \max\{f(\mathbf{y}) - f(\mathbf{x}^*), c(\mathbf{y})\} \\ &\geqslant f(\mathbf{y}) - f(\mathbf{x}^*) \geqslant 0 = F(\mathbf{x}^*,\mathbf{x}^*). \end{aligned}$$

This implies that $\mathbf{x}^*$ is a local minimum of $F(\cdot,\mathbf{x}^*)$, and therefore $0 \in \partial_1 F(\mathbf{x}^*,\mathbf{x}^*)$. We now present Algorithm 3 for computing solutions of program (5).

Convergence theory of iterative algorithm 3 is discussed in (Gabarrou et al., 2013; Noll, 2010) and based on these results, we can prove the following theorem.

Theorem 1. *Assume that functions f and c in program (12) are lower-C^1 and satisfy the following conditions:*

(i) f is weakly coercive on the constraint set $\Omega = \{\mathbf{x} \in \mathbb{R}^{n_\mathbf{x}} : c(\mathbf{x}) \leqslant 0\}$ in the sense that if $\mathbf{x}^j \in \Omega$ and $\|\mathbf{x}^j\| \to \infty$, then $f(\mathbf{x}^j)$ is not monotonically decreasing.

(ii) c is weakly coercive in the sense that if $\|\mathbf{x}^j\| \to \infty$, then $c(\mathbf{x}^j)$ is not monotonically decreasing.

Then the sequence $\mathbf{x}^j$ of serious iterates generated by Algorithm 3 is bounded, and every accumulation point $\mathbf{x}^*$ of the $\mathbf{x}^j$ satisfies $0 \in \partial_1 F(\mathbf{x}^*,\mathbf{x}^*)$. □

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 (Rockafellar and Wets, 1998, Definition 10.29). Theoretical properties of the spectral abscissa $c(\mathbf{x})$, used in the constraint, have been studied in (Burke and Overton, 1994). Lower C^2-functions cover the preponderant part of non-smooth functions encountered in applications. Convergence

Algorithm 3: Proximity control with downshift.

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

1: **Initialize outer loop.** Choose initial iterate $\mathbf{x}^1$ and matrix $Q_1 = Q_1^\top$ with $-qI \preceq Q_1 \preceq qI$. Initialize memory control parameter $\tau_1^\sharp$ such that $Q_1 + \tau_1^\sharp I \succ 0$. Put $j = 1$.

2: **Stopping test.** At outer loop counter j, stop if $0 \in \partial_1 F(\mathbf{x}^j,\mathbf{x}^j)$. Otherwise, goto inner loop.

3: **Initialize inner loop.** Put inner loop counter $k = 1$, initialize $\tau_1 = \tau_j^\sharp$, and build initial working model $F_1(\cdot,\mathbf{x}^j)$ using matrix Q_j.

4: **Trial step generation.** Compute

$$\mathbf{y}^k = \operatorname{argmin} F_k(\mathbf{y},\mathbf{x}^j) + \frac{\tau_k}{2}\|\mathbf{y} - \mathbf{x}^j\|^2.$$

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.

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 $F_{k+1}(\cdot,\mathbf{x}^j)$.

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.

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 $Q_{j+1} + \tau_{j+1}^\sharp I \succ 0$. Increase outer loop counter j and loop back to step 2.

theory for even larger classes of non-smooth functions can be found in (Noll, 2010; Noll et al., 2008).

Corollary 1. *Under the hypotheses of the theorem, every accumulation point of the sequence of serious iterates generated by Algorithm 3 is either a critical point of constraint violation, or a Karush-Kuhn-Tucker point of program (12).*

Proof. Suppose $\mathbf{x}^*$ is an accumulation point of the sequence of serious iterates generated by Algorithm 3. Then $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)$ due to Theorem 1. By using (Clarke, 1981, Proposition 9) (see also (Clarke, 1983, Proposition 2.3.12)), there exist constants λ_0, λ_1 such that

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

If $c(\mathbf{x}^*) > 0$ then $\partial_1 F(\mathbf{x}^*, \mathbf{x}^*) = \partial c(\mathbf{x}^*)$, and therefore $0 \in \partial c(\mathbf{x}^*)$, that is, $\mathbf{x}^*$ is a critical point of constraint violation. In the case of $c(\mathbf{x}^*) \leqslant 0$, if $\mathbf{x}^*$ is not a Karush-Kuhn-Tucker point of (12), then we must have $\lambda_0 = 0$, and so $0 \in \partial c(\mathbf{x}^*)$. We deduce that $\mathbf{x}^*$ is either a critical point of constraint violation, or a Karush-Kuhn-Tucker point of program (12). □

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.

Corollary 2. *Under the hypotheses of the theorem, for every $\varepsilon > 0$ there exists an index $j_0(\varepsilon) \in \mathbb{N}$ such that every $j \geqslant j_0(\varepsilon)$, $\mathbf{x}^j$ is within ε-distance of the set $L = \{\mathbf{x}^* \in \mathbb{R}^{n_\mathbf{x}} : 0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)\}$.*

Proof. Since our algorithm assures always that $\mathbf{x}^j - \mathbf{x}^{j+1} \to 0$, by using Ostrowski's theorem (Ostrowski, 1973, 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. For the details we refer to (Noll, 2012). □

5 A SMOOTH RELAXATION OF HANKEL NORM

This section is motivated by (Nesterov, 2007), 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. Nesterov's 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 non-convex situation. More precisely, we can try to solve problem (5), (12) by replacing the function $f(\mathbf{x}) = \lambda_1(Z(\mathbf{x}))$ by its smooth approximation

$$f_\mu(\mathbf{x}) := \mu \ln \left(\sum_{i=1}^{n} e^{\lambda_i(Z(\mathbf{x}))/\mu} \right),$$

where $\mu > 0$ is a tolerance parameter, n is the order of matrix Z, and where λ_i denotes the ith eigenvalue of a symmetric or Hermitian matrix. Then

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

with $q_i(Z)$ the ith column of the orthogonal matrix $Q(Z)$ from the eigendecomposition of symmetric matrix $Z = Q(Z)D(Z)Q(Z)^\top$. We obtain

$$\nabla f_\mu(\mathbf{x}) =$$
$$(\mathrm{Tr}(Z_1(\mathbf{x})^\top \nabla f_\mu(Z)), \ldots, \mathrm{Tr}(Z_{n_\mathbf{x}}(\mathbf{x})^\top \nabla f_\mu(Z)))^\top.$$

On the other hand,

$$f(\mathbf{x}) \leqslant f_\mu(\mathbf{x}) \leqslant f(\mathbf{x}) + \mu \ln n.$$

Therefore, to find an ε-solution $\bar{\mathbf{x}}$ of problem (12), we find an $\frac{\varepsilon}{2}$-solution of the smooth problem

$$\begin{array}{ll} \text{minimize} & f_\mu(\mathbf{x}) \\ \text{subject to} & c(\mathbf{x}) \leqslant 0 \end{array} \tag{13}$$

with $\mu = \frac{\varepsilon}{2 \ln n}$. Here we use this idea to initialize the non-smooth algorithm 3. The smoothed problem (13) can be solved using standard NLP software.

6 LAUNCHER IN ATMOSPHERIC FLIGHT

In this section we apply Algorithm 1 to design a MIMO PI controller for a satellite launcher in atmospheric flight. The linear model is described by

$$\begin{array}{l} \dot{x} = Ax + Bu \\ y = Cx \end{array} \tag{14}$$

where

$$A = \begin{bmatrix} Z_w & Z_q+U_0 & Z_\theta & Z_v & 0 & Z_\psi & Z_p & Z_\phi \\ M_w & M_q & 0 & 0 & M_r & 0 & M_p & 0 \\ 0 & T_q & 0 & 0 & T_r & 0 & 0 & 0 \\ Y_w & 0 & Y_\theta & Y_v & Y_r & Y_\psi & Y_p & Y_\phi \\ 0 & N_q & 0 & N_v & N_r & 0 & N_p & 0 \\ 0 & P_q & 0 & 0 & P_r & 0 & 0 & 0 \\ 0 & L_q & 0 & 0 & L_r & 0 & L_p & 0 \\ 0 & F_q & 0 & 0 & F_r & 0 & 1 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} Z_{\beta z} & M_{\beta z} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Y_{\beta y} & N_{\beta y} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & L_{\beta r} & 0 \end{bmatrix}^\top,$$

and numerical data in A, B are given in the Appendix. The state vector $x = [w\ q\ \theta\ v\ r\ \psi\ p\ \phi]^\top \in \mathbb{R}^8$ regroups

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).

The control signal is defined as $u = [\beta_z\ \beta_y\ \beta_r]^\top \in \mathbb{R}^3$ with

β_z = deflection of pitch nozzle actuator (deg),
β_y = deflection of yaw nozzle actuator (deg),
β_r = deflection of roll nozzle actuator (deg).

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 (McLean, 1990) about a steady state flight point

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

the procedure being explained in (Greensite, 1970).

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.
- Settling times around 2.5 seconds.

We use a set-point tracking control architecture with MIMO PI feedback as in Figure 1. Tunable matrix gains are therefore K_P and K_I.

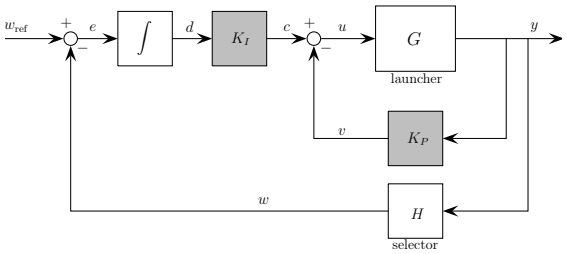

Figure 1: Control architecture with MIMO PI-controller.

For simulation we use the closed-loop transfer function $T_{w_{\text{ref}} \to y}(K)$ that requires the plant P_{sim}:

$$\begin{bmatrix} \frac{y(s)}{y(s)} \\ d(s) \end{bmatrix} = \left[\begin{array}{c|c} 0 & G(s) \\ \hline 0 & G(s) \\ s^{-1}I_3 & -s^{-1}HG(s) \end{array}\right] \begin{bmatrix} \frac{w_{\text{ref}}(s)}{u(s)} \end{bmatrix},$$

which is in closed-loop with the static output feedback control law

$$K: \quad u(s) = \begin{bmatrix} -K_P & K_I \end{bmatrix} \begin{bmatrix} y(s) \\ d(s) \end{bmatrix}.$$

Here $w = (\theta, \psi, \phi) = Hy$, $d = s^{-1}(w_{\text{ref}} - w)$, and

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

It is also convenient to introduce a second performance plant P_{perf} to assess the closed-loop channel $T_{w_{\text{ref}} \to e}(K)$. This requires

$$\begin{bmatrix} \frac{e(s)}{y(s)} \\ d(s) \end{bmatrix} = \left[\begin{array}{c|c} I_3 & -HG(s) \\ 0 & G(s) \\ \hline s^{-1}I_3 & -s^{-1}HG(s) \end{array}\right] \begin{bmatrix} \frac{w_{\text{ref}}(s)}{u(s)} \end{bmatrix}$$

along with the same control structure K.

For pole placement respectively eigenstructure assignment we shall therefore use

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

and the control law is defined by $K_a = W(C_a V)^{-1}$, $K_a = [-K_P\ K_I]$, where

$$W = [w_1 \dots w_9],\ V = [v_1 \dots v_9],$$
$$[A_a - \lambda_i I_{11} | B_a] \begin{bmatrix} v_i \\ w_i \end{bmatrix} = 0.$$

The pole placement is performed by using reference values of the second order system ξ and ω. We choose the desired damping $\xi = \frac{\sqrt{2}}{2}$, and frequencies

$$\omega_1 = 2.1, \omega_2 = 2.2, \omega_3 = 1.8,$$

which leads to the nominal modal set $\Lambda = \{\lambda_1^0, \ldots, \lambda_9^0\}$, where

$$\lambda_{1,2}^0 = -\omega_1 \left(\xi \pm j\sqrt{1-\xi^2}\right),$$
$$\lambda_{3,4}^0 = -\omega_2 \left(\xi \pm j\sqrt{1-\xi^2}\right),$$
$$\lambda_{5,6}^0 = -\omega_3 \left(\xi \pm j\sqrt{1-\xi^2}\right),$$
$$\lambda_7^0 = -3.5,\ \lambda_8^0 = -4,\ \lambda_9^0 = -4.5.$$

In the case of no constraint on eigenvectors v_i, to find the optimal controller K_{opt} we minimize the tracking error by starting Algorithm 1 at a random initial. The algorithm returns the controller K_{opt} with $\|T(P_{\text{perf}}, K_{\text{opt}})\|_H = 0.7135$, while the initial controller K_{init} gives $\|T(P_{\text{perf}}, K_{\text{init}})\|_H = 66.7208$. Figure 2 shows that decoupling is substantially improved. The variation of closed-loop poles from their nominal value is depicted in Figure 3.

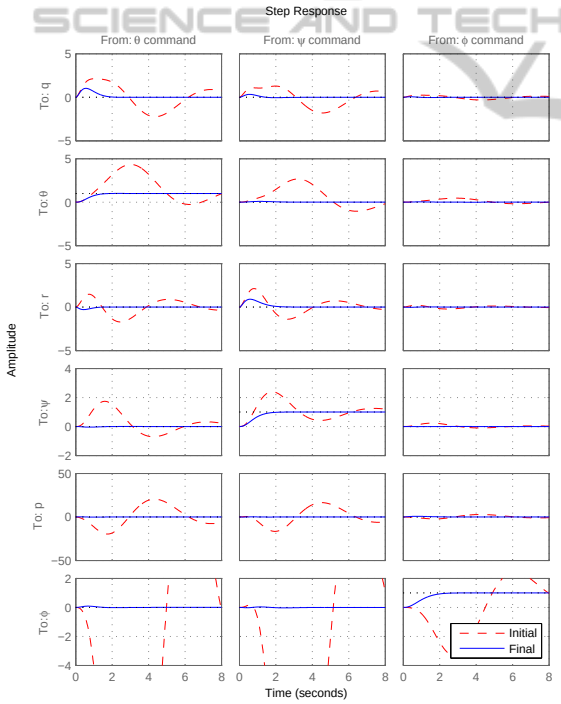

Figure 2: Control of a launcher in atmospheric flight. Initial and final controller obtained respectively by standard and optimized eigenstructure assignment in the case where eigenvectors are not structured. Decoupling is improved.

We now wish to design a closed-loop controller to get decoupling of the modes by choosing some structural constraints on eigenvectors v_i. As an illustration, the eigenvectors v_1 and v_2 are complex conjugate to each other and have zero entries in the rows corresponding to ψ and ϕ. The eigenvector v_4 is the

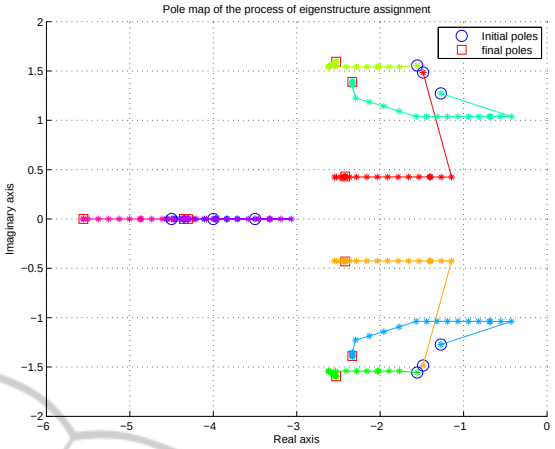

Figure 3: Itineraries of closed-loop poles in optimized eigenstructure assignment based on Hankel program (5).

complex conjugate of v_3 whose entries relative to θ and ϕ are taken as zero. The eigenvectors v_5 and v_6 are complex conjugate and have zero entries in the rows associated with θ and ψ. For the real modes, the eigenvectors are chosen as

$$v_7 = [* * 1 * * 0 * 0 * * *]^\top,$$
$$v_8 = [* * 0 * * 1 * 0 * * *]^\top,$$
$$v_9 = [* * 0 * * 0 * 1 * * *]^\top.$$

The controller K_{opt} computed by Algorithm 1 gives $\|T(P_{\text{perf}}, K_{\text{opt}})\|_H = 0.7360$, while the initial controller K_{init} obtained by standard assignment has $\|T(P_{\text{perf}}, K_{\text{init}})\|_H = 0.7787$. Figure 4 illustrates the improvement in step responses.

7 CONCLUSIONS

We have presented a new approach to partial eigenstructure assignment in output feedback control, which is *dynamic* in the sense that it allows the eigenelements (λ_i, v_i, w_i) to move in the neighborhood of their nominal elements $(\lambda_i^0, v_i^0, w_i^0)$ obtained by standard assignment. This gain of flexibility is used to optimize closed-loop stability and performance. Optimization is based on minimizing the Hankel norm of the performance channel, as this reduces system ringing. The efficiency of the new approach was demonstrated for control of a launcher in atmospheric flight.

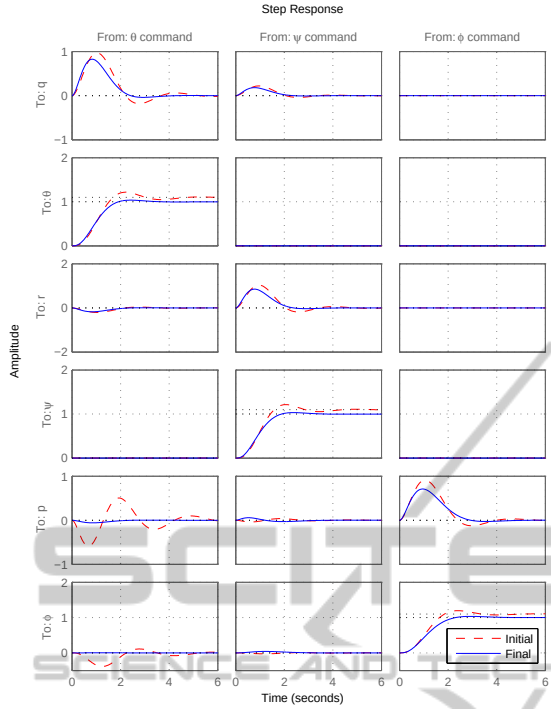

Figure 4: Control of a launcher in atmospheric flight. Initial and final controller obtained respectively by standard and optimized eigenstructure assignment in the case of structured eigenvectors.

REFERENCES

Apkarian, P. and Noll, D. (2006). Nonsmooth H_∞ synthesis. *IEEE Trans. Automat. Control*, 51(1):71–86.

Burke, J. V. and Overton, M. L. (1994). Differential properties of the spectral abscissa and the spectral radius for analytic matrix-valued mappings. *Nonlinear Anal.*, 23(4):467–488.

Clarke, F. H. (1981). Generalized gradients of Lipschitz functionals. *Adv. in Math.*, 40(1):52–67.

Clarke, F. H. (1983). *Optimization and Nonsmooth Analysis*. John Wiley & Sons, Inc., New York.

Gabarrou, M., Alazard, D., and Noll, D. (2013). Design of a flight control architecture using a non-convex bundle method. *Math. Control Signals Systems*, 25(2):257–290.

Greensite, A. L. (1970). *Elements of modern control theory*, volume 1 of *Control Theory*. Spartan Books, New York.

McLean, D. (1990). *Automatic flight control systems*. Prentice Hall, London.

Nesterov, Y. (2007). Smoothing technique and its applications in semidefinite optimization. *Math. Program., Ser. A*, 110(2):245–259.

Noll, D. (2010). Cutting plane oracles to minimize non-smooth non-convex functions. *Set-Valued Var. Anal.*, 18(3–4):531–568.

Noll, D. (2012). Convergence of non-smooth descent methods using the Kurdyka-Łojasiewicz inequality. Preprint.

Noll, D., Prot, O., and Rondepierre, A. (2008). A proximity control algorithm to minimize nonsmooth and nonconvex functions. *Pac. J. Optim.*, 4(3):571–604.

Ostrowski, A. M. (1973). *Solutions of Equations in Euclidean and Banach Spaces*. Academic Press, New York-London.

Overton, M. L. (1992). Large-scale optimization of eigenvalues. *SIAM J. Optim.*, 2(1):88–120.

Polak, E. (1997). *Optimization: Algorithms and Consistent Approximations*, volume 124 of *Applied Mathematical Sciences*. Springer-Verlag, New York.

Rockafellar, R. T. and Wets, R. J.-B. (1998). *Variational Analysis*. Springer-Verlag, Berlin.

APPENDIX

The numerical data for A and B used in (14) are gathered in the following Table 1.

Table 1: 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