Preprints of the 2012 20th Mediterranean Conference on Control & Automation (MED), Barcelona, Spain, July 3-6, 2012

Output Feedback Based Pole Confinement For Launch Vehicle Attitude Control

Vincent Dubanchet, David Saussié, Lahcen Saydy, Richard Gourdeau and Caroline Bérard

***Abstract*—A new and more efficient version of a previous algorithm based upon guardian maps to synthetize output feedback control laws is proposed in this paper. Explicit gradient calculations are used to implement well-known quasi-Newton methods, in particular BFGS method. The new algorithm proves more efficient and less computational demanding than its previous version. It is successfully applied to design launcher vehicle attitude control laws.**

I. Introduction

Output feedback stabilization remains a fundamental and challenging problem in linear control which is known to be in the class NP, and probably to be NP-hard [1]. As pointed out by [2], necessary and sufficient conditions leading to an efficient algorithm are still paramount.

The static output feedback (SOF) stabilization problem can typically be cast into a Bilinear Matrix Inequality (BMI) condition via a Lyapunov inequality such as the following one:

Find K and P such that $(A+BKC)^T P+P(A+BKC) \prec 0$, $P = P^T \succ 0$ where $\prec$ and $\succ$ respectively stand for negative semidefinite and positive semidefinite.

In [3], SOF is cast into the BMI context using the Hermite stability criterion but without the additional Lyapunov variables. In [4], the author used *direct search methods* to solve the underlying numerical analysis issues and proposed three multivariate direct search functions (`adsmax`, `mdsmax`, `nmsmax`) in his Matrix Computation Toolbox for MATLAB [5]. They were then tested in [6] to tackle the SOF problem. In [**?**], gradient sampling is used on the spectral abscissa function. The resulting the package (`HIFOO`) does have a nondeterministic aspect to it as the search directions depend on randomly sampled points. The algorithm may thus give different results for different runs. Finally, [7] proposed a deterministic way of proceeding with a new way to find descent steps which is based on subgradients of the spectral abscissa.

Here, SOF is considered in the more general setting of generalized stability, that is, pole confinement in a specific subset of the complex plane of interest (the open left half-plane and unit disk being two classical examples). Even if pole confinement can be expressed in terms of LMI constraints for the state feedback case ([8], [9]), BMI are still involved when it comes to SOF. Yet another characterization is available in terms of guardian maps ([10]) which led to a new algorithm that was used in [11] to design a pitch rate flight control law. The proposed algorithm was however partly based on an *ad hoc* procedure which did not take advantage of guardian map gradient information. Instead of using search direction derived from gradient computation, the variables were iteratively optimized one at a time and, although good results were obtained, the computational burden may prove to be prohitive for other applications. To improve the algorithm's efficiency, we rely on gradient computations based on Jacobi's formula for determinant derivatives as well as quasi-Newton methods [12].

Section 2 briefly presents the guardian map theory and some useful results. In Section 3, guardian map gradients w.r.t. controller gains are computed and used in a pole confinement algorithm based upon BFGS method. Section 4 presents the description of a launch vehicle attitude control problem, the model of which was provided by ASTRIUM-ST. Finally, Section 5 presents the synthesis of PD and PID controllers to meet the control requirements.

II. Guardian Maps

The guardian map approach was introduced in [10] as a unifying tool for the study of generalized stability of parameterized families of matrices or polynomials. Here, generalized stability means confinement of matrix eigenvalues or polynomial zeros to general open subsets of the complex plane.

A. Definition

Basically, guardian maps are scalar valued maps defined on the set of $n \times n$ real matrices (or n^{th}-order polynomials) that take non-zero values on the set of "stable" matrices (or polynomials) and vanish on its boundary. The description below will focus on families of matrices with the understanding that it applies to polynomials as well. We are hence interested in stability sets of the form:

$$S(\Omega) = \{A \in \mathbb{R}^{n\times n} | \, \sigma(A) \subset \Omega\} \tag{1}$$

where Ω is an open subset of the complex plane of interest, and $\sigma(A)$ denotes the spectrum of A.

Definition 1. *Let ν map $\mathbb{R}^{n\times n}$ into $\mathbb{C}$. We say that ν_Ω* guards *$S(\Omega)$ if for all $A \in \overline{S}(\Omega)$, the following equivalence holds:*

$$\nu_\Omega(A) = 0 \Leftrightarrow A \in \partial S(\Omega) \tag{2}$$

Vincent Dubanchet, David Saussié, Lahcen Saydy and Richard Gourdeau are with École Polytechnique de Montréal, Department of Electrical Engineering, Montréal, QC, H3T 1J4, Canada {vincent.dubanchet, david.saussie, lahcen.saydy, richard.gourdeau}@polymtl.ca

Caroline Bérard is with Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse, 31000, France, caroline.berard@isae.fr

978-1-4673-2529-5/12/$31.00 ©2012 IEEE

Here $\overline{S}$ denotes closure of the set S and ∂S its boundary.

See Appendix A for usual guardian maps.

B. Robust stability and the characterization of stabilizing gains:

Let $\{A(r) \in \mathbb{R}^{n\times n} | \, r \in \mathcal{U} \subset \mathbb{R}^k\}$ be a continuous family of matrices which depend on the parameter vector $r \in \mathcal{U}$ where $\mathcal{U}$ is pathwise connected.

Theorem 1. *(Saydy et al, [10]) Let $S(\Omega)$ be guarded by the map ν_Ω. The family $\{A(r) | \, r \in \mathcal{U}\}$ is stable relative to Ω if and only if:*

1) *it is nominally stable, i.e. $\exists \, r_0 \in \mathcal{U}$ such that $A(r_0) \in S(\Omega)$;*
2) *and $\forall r \in \mathcal{U}$, $\nu_\Omega(A(r)) \neq 0$.*

Corollary 1. *Let $S(\Omega)$ be guarded by the map ν_Ω and consider the family $\{A(r) | \, r \in \mathcal{U}\}$. Then the set $\mathcal{C}$ defined by:*

$$\mathcal{C} = \{r \in \mathbb{R}^k | \, \nu_\Omega(A(r)) = 0\} \tag{3}$$

divides the parameter space $\mathbb{R}^k$ into components $\mathcal{C}_i$ that are either stable or unstable relative to Ω. To see which situation prevails for a given component $\mathcal{C}_i$, one simply has to test $A(r)$ for any one vector in $\mathcal{C}_i$.

Example 1. Suppose that the closed-loop poles of a given system are specified by the polynomial:

$$p(s) = s^3 + k_1 s^2 + k_2 s + 1 \tag{4}$$

where k_1, k_2 denote some controller gains. If Ω is the conic sector $|\theta| < \pi/4$ (i.e. $\zeta > 0.707$), then one obtains (e.g. by applying (32) to the companion matrix corresponding to p):

$$\nu_\Omega(p) = 2k_2^3 - k_1^2 k_2^2 - 4k_1 k_2 + 2k_1^3 + 1 \tag{5}$$

Setting this quantity to 0 yields the 3 components in the parameter space (k_1, k_2) of Fig. 2. It can be verified that the set of all gains (k_1, k_2) which place all the closed-loop poles within the target damping zone is the component $\mathcal{C}_3$. Any other choice of the gains yields closed-loop poles which are not entirely within Ω. We arrive to this conclusion simply by testing three pairs (k_1, k_2) in $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ respectively.

C. One-parameter family stability test

Let $M(r) = M_0 + rM_1 + \ldots + r^m M_m$ where all $M_i \in \mathbb{R}^{n\times n}$ are constant matrices, M_0 is non singular and $r \in \mathbb{R}$. Let

$$r^- \doteq \sup\{r < 0 | \, \det(M(r)) = 0\} \text{ (or } -\infty \text{ if none exists)}$$
$$r^+ \doteq \inf\{r > 0 | \, \det(M(r)) = 0\} \text{ (or } +\infty \text{ if none exists)}$$

be the maximal perturbation bounds for nonsingularity of matrices around $r = 0$.

Lemma 1. *Let $A(r) = A_0 + rA_1 + \ldots + r^k A_k$ be a polynomial matrix in the uncertain parameter $r \in \mathbb{R}$ with given constant matrices $A_i \in \mathbb{R}^{n\times n}$ such that $A(0)$ is stable w.r.t. Ω and let $S(\Omega)$ be guarded by a map ν_Ω of the form $\nu_\Omega(A(r)) = \det(M(r))$ with $M(r)$ the polynomial matrix associated to $A(r)$ according to ν_Ω, then $A(r)$ is stable relative to Ω for $r \in]r^-; r^+[$. Furthermore, this interval is the largest open one containing 0.*

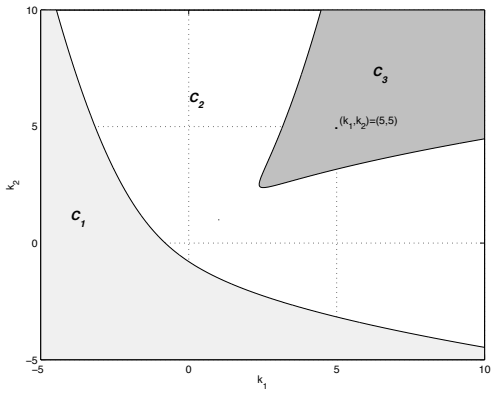

Figure 1. Set C_3 of all gains ensuring $\Omega-$stability

III. Pole Confinement Algorithm

We propose an algorithm that will possibly solve the following problem:

Problem 1. *Let $P(s)$ be an LTI model and $C[K](s)$ an LTI fixed structure controller with tunable control gains $\boldsymbol{K} = [K_j] \in \mathcal{U} \subset \mathbb{R}^k$. The problem is to tune $\boldsymbol{K}$ so that the closed-loop poles lie in a target stability region of the complex plane Ω_t of interest.*

Hypothesis 1. *Here, we focus for simplicity on stability regions Ω_t of the form (Fig. 2) :*

$$\Omega = \{z \in \mathbb{C} | \, \Re(z) < \alpha, \, |\pi - \angle z| < \theta\} \tag{6}$$

with $\zeta = \cos\theta$

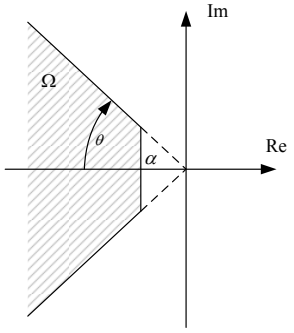

Figure 2. Target region Ω

The algorithm presented in [11] was partly based on an *ad hoc* procedure. Explicit calculations of gradients of guardian maps associated with the target region are used to implement a BFGS method. The new algorithm proves more efficient and less computational demanding than its previous version.

Next, we present some basic gradient computations for the guardian maps (31) and (32) corresponding to the above

target region Ω_t (see the appendix), then we present the algorithm itself derived from BFGS[1] method [12].

A. Gradient computation

Let $\boldsymbol{K} = [K_j]$ denote the vector of tunable parameters and let

$$F = F_0 + \sum_{j=1}^{k} K_j F_j \tag{7}$$

a matrix of appropriate size which depends on the $[K_j]$'s.

Then the determinant of (7) is a multivariate polynomial function of $\boldsymbol{K}$, the derivative of which is given by Jacobi's formula as:

$$\frac{\partial \det F}{\partial K_j} = \mathrm{tr}\left(\mathrm{adj}(F)\frac{\partial F}{\partial K_j}\right) = \mathrm{tr}\left(\mathrm{adj}(F)F_j\right) \tag{8}$$

where $\mathrm{adj}(F)$ denotes the adjugate of F and $\mathrm{tr}(F)$ its trace.

Let now the closed-loop stae space matrix be written as:

$$A_c = A_0 + \sum_{j=1}^{k} K_j A_j \tag{9}$$

1) Spectral abscissa constraint: A corresponding guardian map is given by (31). Let $\nu_1(\alpha, \boldsymbol{K}) = \det(A_c - \alpha I)$. Therefore we have:

$$\nabla\nu_1(\alpha, \boldsymbol{K}) = \frac{\partial \nu_1}{\partial \boldsymbol{K}}(\alpha, \boldsymbol{K}) = \left[\frac{\partial \nu_1}{\partial K_j}(\alpha, \boldsymbol{K})\right] \tag{10}$$

Using (8), one can deduce:

$$\nabla\nu_1(\alpha, \boldsymbol{K}) = \left[\mathrm{tr}\left(\mathrm{adj}(A')A_j\right)\right] \tag{11}$$

with $A' = A_c(\boldsymbol{K}) - \alpha I$.

Similarly, $\nu_2(\alpha, \boldsymbol{K}) = \det(A_c \odot I - \alpha I \odot I)$, one would find with the same procedure:

$$\nabla\nu_2(\alpha, \boldsymbol{K}) = \left[\mathrm{tr}\left(\mathrm{adj}(A' \odot I)(A_j \odot I)\right)\right] \tag{12}$$

2) Conic sector: We consider the guardian map (32). Let $\nu_3(\zeta, \boldsymbol{K}) = \det\left(A_c^2 \odot I + (1-2\zeta^2)A_c \odot A_c\right)$. Using (8) again, one can then calculate:

$$\begin{aligned}\nabla\nu_3(\zeta, \boldsymbol{K}) = \big[&\mathrm{tr}\left(\mathrm{adj}(A'')((A_cA_j + A_jA_c) \odot I + \ldots\right.\\ &\left.\ldots + 2(1-2\zeta^2)A_c \odot A_j)\right)\big]\end{aligned} \tag{13}$$

with $A'' = A_c(\boldsymbol{K})^2 \odot I + (1-2\zeta^2)A_c(\boldsymbol{K}) \odot A_c(\boldsymbol{K})$.

Remark 1. We do not consider the second factor of (32) as it is a particular case of ν_1 with $\alpha = 0$.

[1]Broyden-Fletcher-Goldfarb-Shanno

B. Algorithm description

In order to find satisfying gains that place the closed-loop poles in a target region Ω_t (parameters α_t, ζ_t), the algorithm alternates between two phases[11]:

1. At iteration l, definition of a region Ω_l and corresponding guardian map ν w.r.t. current $\boldsymbol{K}_l$;
2. Minimization of ν within the active component (Corollary 1) to find $\boldsymbol{K}_{l+1}$.

If $A_c(\boldsymbol{K}_l)$ is Hurwitz stable, we define Ω_l as (6) (see Step 1. of the algorithm in Tab. I). The corresponding guardian map is then:

$$\nu_{\alpha_l,\zeta_l}(\boldsymbol{K}) = \nu_1(\alpha_l, \boldsymbol{K})\nu_2(\alpha_l, \boldsymbol{K})\nu_3(\zeta_l, \boldsymbol{K}) \tag{14}$$

and its derivative w.r.t. $\boldsymbol{K}$ is:

$$\nabla\nu_{\alpha_l,\zeta_l} = \sum_{j=1}^{3}\left(\prod_{i=1, i\neq j}^{3}\nu_i\right)\nabla\nu_j \tag{15}$$

where $\nabla\nu_j$ are computed using (11), (12) and (13).

If $A_c(\boldsymbol{K}_l)$ is not Hurwitz stable, we take $\Omega_l = \{z \in \mathbb{C} | \Re(z) < \alpha_l\}$ with $\alpha_l \geq 0$, spectral abscissa of $A_c(\boldsymbol{K}_l)$. The corresponding guardian map is then:

$$\nu_{\alpha_l}(\boldsymbol{K}) = \nu_1(\alpha_l, \boldsymbol{K})\nu_2(\alpha_l, \boldsymbol{K}) \tag{16}$$

and its derivative w.r.t. $\boldsymbol{K}$ is:

$$\nabla\nu_{\alpha_l} = \nu_1\nabla\nu_2 + \nu_2\nabla\nu_1 \tag{17}$$

where $\nabla\nu_j$ are computed using (11) and (12).

C. Proposed algorithm

The algorithm is divided in two loops corresponding to the two phases mentionned in previous section. We first introduce algorithm notations then the algorithm itself.

1) Main loop notations:

l	counter
N_1	maximum number of iterations
$\boldsymbol{K}_l$	current value of the variables
$\varepsilon_{\boldsymbol{K}}$	tolerance for step size in gain spaces
$\alpha(A)$	spectral abscissa of matrix A
$\zeta(A)$	minimal damping of matrix A (if Hurwitz)

2) Minimization loop notations:

i	counter
N_2	maximum number of iterations
X_i	current value of the variables
ε	tolerance for the gradient norm
ν_i	objective function value at X_i
$\nabla\nu_i$	gradient of objective function at X_i
H_i	inverse of approximate Hessian matrix
d_i	i^{th} descent direction at X_i
β	coefficient of the Armijo line search condition
γ	coefficient of the curvature condition

Table I
POLE CONFINEMENT ALGORITHM

Set $\Omega_t = \Omega_{\alpha_t,\zeta_t}$, $\boldsymbol{K}_0 \in \mathcal{U} \subset \mathbb{R}^k$, $\epsilon, \epsilon_K > 0$, $\beta \in]0,1[$, $\gamma \in]0,1[$, $N_1, N_2 > 0$
Set counter $l \leftarrow 0$.
Main Loop

1. Compute $\alpha_l = \alpha(A_c(\boldsymbol{K}_l))$ and $\zeta_l = \zeta(A_c(\boldsymbol{K}_l))$ (if Hurwitz).
   If ($\alpha_l \leq \alpha_t$ **and** $\zeta_l \geq \zeta_t$) **or** $l > N_1$ **then Stop**
   Else $\alpha_l = \max(\alpha_l, \alpha_t)$ and $\zeta_l = \min(\zeta_l, \zeta_t)$.
   Define $\nu = \nu_{\alpha_l,\zeta_l}$ (if Hurwitz) or $\nu = \nu_{\alpha_l}$ if not.
2. Set $X_0 = \boldsymbol{K}_l$. Compute $c = \nabla\nu(\boldsymbol{K}_l)$ and set $H_0 = \frac{1}{\|c\|} I$ as inverse of initial Hessian matrix approximation.
3. **Minimization Loop** (over variable X_i)
   Set counter $i \leftarrow 0$.
   3.a **If** $||\nabla\nu_i|| < \epsilon$ **or** $i > N_2$ **then Stop Minimization Loop**. Go to 4.
   3.b Obtain the descent direction according to BFGS method:
   $$d_i = -H_i \nabla\nu_i$$
   3.c Find a step length t_i satisfying the Wolfe conditions (Armijo and curvature conditions):
   $$\nu(X_i + t_i d_i) \leq \nu_i + \beta\, t_i\, \nabla\nu_i^T d_i$$
   and
   $$|\nabla\nu(X_i + t_i d_i)^T d_i| \leq \gamma |\nabla\nu_i^T d_i|$$
   3.d Set $X_{i+1} \leftarrow X_i + t_i d_i$
   Compute $\nu_{i+1} \leftarrow \nu(X_{i+1})$
   Compute $\nabla\nu_{i+1} \leftarrow \nabla\nu(X_{i+1})$
   Compute H_{i+1} using Sherman-Morrison formula
   Set $i \leftarrow i+1$ and go back to I.3.a.
4. Set $\boldsymbol{K}_{l+1} \leftarrow X_i$
   If $||\boldsymbol{K}_{l+1} - \boldsymbol{K}_l|| < \epsilon_K$ **then Stop**
   Else set $l \leftarrow l+1$ and go back to 1.

3) Algorithm:

The algorithm is presented in Tab. I.

Remark 2. *Armijo line search (3.c).* Line search methods usually start with a step length $t = 1$ when the descent direction d is defined by $d = -H\,\nabla\nu$. But in the present case, depending on the local hessian approximation, starting with such a step length could provide a point outside the active component. Therefore we used the Lemma 1 to find the step length range $[t_{min}, t_{max}]$ that keeps the point $X + td$ inside the component. Our line search method starts then with the step length t_{max}.

IV. LAUNCH VEHICLE ATTITUDE CONTROL

Here we focus of the design of an output feedback control law for a launcher vehicle in atmospheric ascent, from time 25 s to time 60 s since lift-off. To control the attitude, launch vehicle dynamics are then generally described by "short-period" equations of motion whose parameters are changing greatly due to mass variation, velocity and altitude. The mathematical model is therefore time-dependent. In order to relax aerodynamic loads on the structure, the main constraint is generally to keep the angle of attack α small (no manoeuvre commands are considered here) [13], [14].

A. Model description

Relative orientation of the vehicle in the aerodynamic context is defined by the angle of attack α while the launch vehicle orientation is defined by the attitude angle θ. They are related by the equation:

$$\alpha = \theta + \gamma - \frac{W}{V_r}, \tag{18}$$

with W horizontal wind input, V_r launch vehicle relative speed and γ path angle. Under the assumption $\gamma = 0°$, the launcher model is then described along a nominal trajectory by the LTV model [15], [16]:

$$\begin{bmatrix} \ddot{\theta} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & a_{12} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \theta \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} W \\ \beta \\ \ddot{\beta} \end{bmatrix} \tag{19}$$

$$\begin{bmatrix} \theta \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \theta \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ d_{21} & 0 & 0 \end{bmatrix} \begin{bmatrix} W \\ \beta \\ \ddot{\beta} \end{bmatrix} \tag{20}$$

with β the actual nozzle deflection. The coefficients a_{ij}, b_{ij} and d_{ij} are time-varying coefficients and depend mainly on the launch vehicle relative speed and inertia. The inputs β and $\ddot{\beta}$ are generated by a 4^{th} order actuator:

$$\begin{bmatrix} \beta \\ \ddot{\beta} \end{bmatrix} = G(s)\beta_c \tag{21}$$

where β_c is the commanded nozzle deflection. Actuator dynamics are defined by two pairs of complex poles: $(\omega_1, \zeta_1) = (102, 0.67)$ and $(\omega_2, \zeta_2) = (173, 0.2)$. We consider six flight instants evenly distributed between 25 s and 60 s: $t_1 = 25$ s, $t_2 = 32$ s, ..., $t_6 = 60$ s.

B. Requirements

Only the input β_c and the output θ are available to design the controller (Fig. 3).

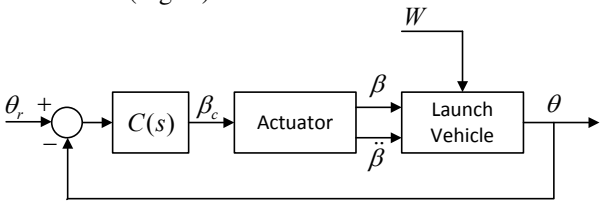

Figure 3. Attitude control system

We focus mainly on the following closed-loop pole specifications:

- Minimal damping for the launcher poles: $\zeta > 0.5$
- Maximum real part: $\alpha < -0.5$

For synthesis purpose, a balanced reduction is performed to simplify the model. As expected, the actuator poles are removed as they are much faster than the launcher poles. The 2^{nd} order reduced model is denoted as:

$$\dot{x} = A_r x + B_r \beta_c \tag{22}$$
$$\theta = C_r x + D_r \beta_c \tag{23}$$

V. Design Example

We seek to apply our algorithm to the synthesis of two classical controllers, namely, PD and PID controllers. The target region Ω_t is defined with $\alpha_t = -0.5$ and $\zeta_t = 0.5$.

A. PD synthesis

We first consider the synthesis of a PD controller with $\tau_1 = 1/30$ in order to limit the derivative action. Moreover we add a 1^{st} order filter (with $\tau_2 = 1/30$) to impose a roll-off constraint of $-20\,\mathrm{dB/dec}$.

$$C(s) = \left(K_p + \frac{K_d s}{\tau_1 s + 1}\right) \frac{1}{\tau_2 s + 1} \tag{24}$$

The closed-loop matrix A_c can then be written as:

$$A_c = A_o + B_o \begin{bmatrix} K_p & K_d \end{bmatrix} C_o \tag{25}$$

with

$$A_o = \begin{bmatrix} A_r & 0 & 0 \\ -\frac{C_r}{\tau_2} & -\frac{1}{\tau_2} & 0 \\ 0 \quad 0 & \frac{1}{\tau_1} & -\frac{1}{\tau_1} \end{bmatrix} \quad B_o = \begin{bmatrix} B_r \\ -\frac{D_r}{\tau_2} \\ 0 \end{bmatrix} \tag{26}$$
$$C_o = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\tau_1} & -\frac{1}{\tau_1} \end{bmatrix}$$

Starting (arbitrarily) from zero gains, the tuned values of the controller gains are gathered in Table II for the six flight instants. In each case, four iterations of the algorithm are needed to find satisfying gains that place the closed-loop poles in the target region (Fig. 4). Moreover the final gains are decreasing monotonically with flight instant. It is interesting to note that the closed-loop poles dispersion is quite similar from a flight instant to another one. Fig. 5 illustrates the different iterations of the algorithm till the final values inside the component of interest.

Table II
Tuned values of the PD gains

Flight instant	t_1	t_2	t_3	t_4	t_5	t_6
K_p	2.748	2.448	2.139	1.826	1.351	1.137
K_d	1.104	0.973	0.810	0.671	0.538	0.454
# of iterations	4	4	4	4	4	4

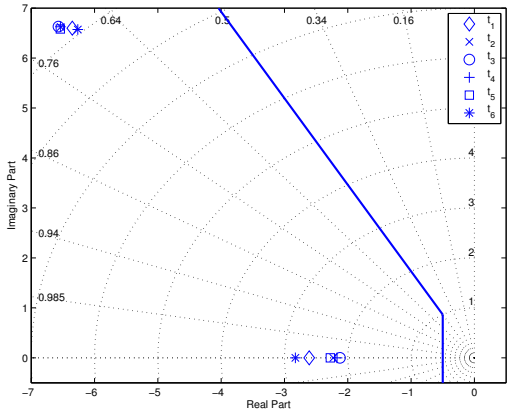

Figure 4. Closed-loop poles with PD controller

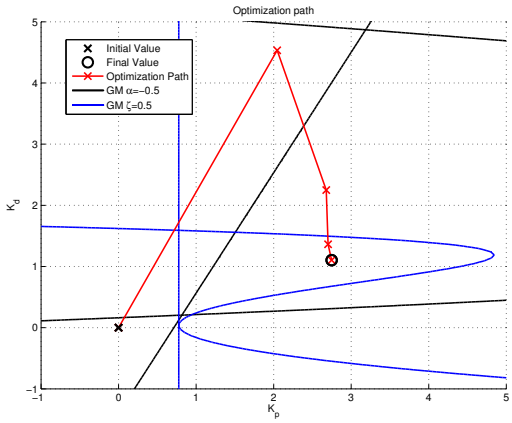

Figure 5. Optimization path for flight instant t_1

B. PID synthesis

Here,

$$C(s) = \left(K_p + \frac{K_i}{s} + \frac{K_d s}{\tau_1 s + 1}\right) \frac{1}{\tau_2 s + 1} \tag{27}$$

The closed-loop matrix A_c can then be written as:

$$A_c = A_o + B_o \begin{bmatrix} K_p & K_i & K_d \end{bmatrix} C_o \tag{28}$$

with

$$A_o = \begin{bmatrix} A_r & 0 & 0 & 0 \\ -\frac{C_r}{\tau_2} & -\frac{1}{\tau_2} & 0 & 0 \\ 0 \quad 0 & 1 & 0 & 0 \\ 0 \quad 0 & \frac{1}{\tau_1} & 0 & -\frac{1}{\tau_1} \end{bmatrix} \quad B_o = \begin{bmatrix} B_r \\ -\frac{D_r}{\tau_2} \\ 0 \\ 0 \end{bmatrix} \tag{29}$$
$$C_o = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\tau_1} & 0 & -\frac{1}{\tau_1} \end{bmatrix}$$

Starting again (arbitrarily) from zero gains, the tuned values of the controller gains are gathered in Table III for the six flight instants. Fig. 6 shows the corresponding closed-loop poles; they are all inside the target stability region.

Table III
TUNED VALUES OF THE PID GAINS

Flight instant	t_1	t_2	t_3	t_4	t_5	t_6
K_p	2.407	2.321	2.258	2.415	1.979	1.525
K_i	1.725	0.967	0.892	1.756	2.348	1.673
K_d	1.032	1.028	0.813	0.718	0.557	0.456
# of iterations	3	3	3	9	5	5

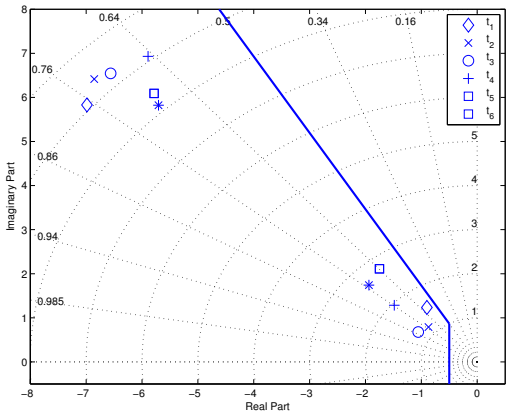

Figure 6. Closed-loop poles with PID controller

VI. CONCLUSION

This article presents a new version of a former algorithm that tunes the gains of a structured compensator so the closed-loop poles lie in a specific stability region. Explicit gradient computation and BFGS method define efficient descent directions that help fast convergence and alleviate computation burden compared to previous solution. The algorithm was successfully applied on a launch vehicle attitude control; the tuning of PD and PID controllers was performed on different flight instants along the trajectory.

VII. ACKNOWLEDGMENTS

The authors would like to thank ASTRIUM-ST who allowed the launch vehicle model to be used for this article. This work was supported by an NSERC Discovery Grant, RGPIN 121639.

REFERENCES

[1] V. Blondel and J. Tsitsiklis, "NP-hardness of some linear control design problems," *SIAM Journal on Control and Optimization*, vol. 35, no. 6, pp. 2118–2127, 1997.

[2] D. S. Bernstein, "Some open problems in matrix theory arising in linear systems and control," *Linear Algebra and its Applications*, vol. 162-164, pp. 409–432, 1992.

[3] D. Henrion, J. Löfberg, M. Kočvara, and M. Stingl, "Solving polynomial static output feedback problems with PENBMI," in *Proc. 44th IEEE Conference Decision and Ctrl. Eur. Ctrl. Conf. CDC-ECC'05*. IEEE, 2005, pp. 7581–7586.

[4] J. N. Higham, "Optimization by direct search in matrix computations," *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 2, pp. 317–333, April 1993.

[5] ——, "The matrix computation toolbox." See http://www.ma.man.ac.uk/ higham/mctoolbox, Version 1.2 released in September 2002.

[6] D. Henrion, "Solving static output feedback problems by direct search optimization," in *Proceedings of the 2006 IEEE Conference on Computer Aided Control Systems Design, CACSD*, IEEE, Ed., 2007, pp. 1534–1537.

[7] V. Bompart, P. Apkarian, and D. Noll, "Non-smooth techniques for stabilizing linear systems," in *2007 American Control Conference New York, NY, USA*. IEEE, 2007, pp. 1245–1250.

[8] M. Chilali and P. Gahinet, "h_∞ design with pole placement constraints," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 358–267, 1996.

[9] C. Scherer, M. Chilali, and P. Gahinet, "Multiobjective output feedback control via LMI optimization," *IEEE Transactions on Automatic Control*, vol. 42, no. 7, pp. 896–911, 1997.

[10] L. Saydy, A. Tits, and E. Abed, "Guardian maps and the generalized stability of parametrized families of matrices and polynomials," *Mathematics of Control, Signals and Systems*, vol. 3, no. 4, pp. 345–371, 1990.

[11] D. Saussié, L. Saydy, and O. Akhrif, "Pitch rate control with guardian map based algorithm," in *18th Mediterranean Conference on Control and Automation, 23-25 June 2010, Marrakech, Morocco*, 2010, pp. 1473 – 1478.

[12] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., ser. Springer Series in Operations Research and Financial. Springer Verlag, New York, NY, 2006.

[13] M. Hernandez, "Launcher study case: modal self -scheduling control," Master's thesis, ONERA-France, 2008.

[14] D. Saussié, V. Dubanchet, C. Bérard, L. Saydy, and R. Gourdeau, "Control of launcher in atmospheric ascent with guardian maps," in 8^{th} *International ESA Conference on Guidance, Navigation & Control Systems*, June 2011.

[15] J. H. Blakelock, *Automatic control of aircraft and missiles*. New York, NY: John Wiley & Sons, Inc., 1965.

[16] L. Greensite, *Analysis and design of space vehicle control system*. New York: Spartan Books, 1970.

[17] C. Stephanos, "Sur une extension du calcul des substitutions linéaires," *J. Math. Pures Appl.*, vol. 5, no. 6, pp. 73–128, 1900.

APPENDIX

A. Usual guardian maps

Guardian maps are given for classical regions of Fig. 7.

- Hurwitz stability (open left half-plane $\mathring{\mathbb{C}}_-$)

$$\nu_H(A) = \det(A \odot I)\det(A) \tag{30}$$

  where $\odot$ denotes the bialternate product [17].

- Spectral abscissa (open α-shifted left half-plane, i.e. $\mathrm{Re}(z) < \alpha$)

$$\nu_m(A) = \det\left(A \odot I - \alpha I \odot I\right)\det\left(A - \alpha I\right) \tag{31}$$

- Conic sector with inner angle 2θ.

$$\nu_d(A) = \det\left(A^2 \odot I + (1-2\zeta^2)A \odot A\right)\det\left(A\right) \tag{32}$$

  where $\zeta = \cos\theta$ is the limiting damping ratio.

- Schur stability (or more generally open disk of radius ω).

$$\nu_p(A) = \det\left(A \odot A - \omega^2 I \odot I\right)\det\left(A^2 - \omega^2 I\right) \tag{33}$$

A systematic way of constructing guardian maps for various Ω regions can be found in [10].

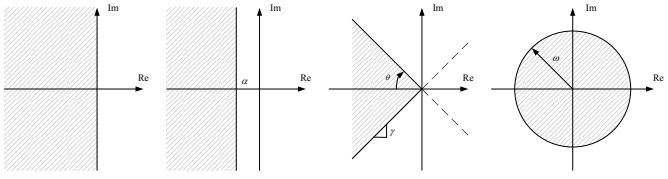

Figure 7. Stability regions