Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut Supérieur de l'Aéronautique et de l'Espace (ISAE)

Présentée et soutenue par :
Marion GABARROU

le lundi 26 novembre 2012

Titre :

Développement d'un algorithme de faisceau non convexe avec contrôle de proximité pour l'optimisation de lois de commande structurées.

École doctorale et discipline ou spécialité :
ED MITT : Mathématiques appliquées

Unité de recherche :
Institut de Mathématiques de Toulouse - Équipe MIP

Directeur(s) de Thèse :

Dominikus Noll et Daniel Alazard

Jury :
Samir Adly - Rapporteur
Daniel Alazard - Co-directeur de thèse
Pierre Apkarian - Examinateur
Dominikus Noll - Directeur de thèse
Olivier Prot - Examinateur
Michel Zasadzinski - Rapporteur

Table des matières

Table des figures

Remerciements

Je remercie mes directeurs de thèse Dominikus Noll et Daniel Alazard pour m'avoir donné l'opportunité de découvrir le monde de la recherche et de travailler sur un sujet enrichissant personnellement et valorisant. En effet l'optimisation est un maître mot dans nos sociétés et permet d'avoir un contact direct avec les applications diverses et variées. Je remercie mes rapporteurs Samir Adly et Michel Zasadzinski d'avoir rapporté ma thèse, d'y avoir consacré du temps et d'y avoir porté de l'intérêt. Je remercie Olivier Prot et Pierre Apkarian d'avoir accepté d'être examinateur dans mon jury.

Je remercie Aude Rondepierre et à nouveau Olivier Prot pour leur soutien scientifique et leurs encouragements apportés tout au long de ma thèse. Je remercie vivement Stanislas Larnier pour son soutien infaillible et précieux. Je remercie également ses parents Catherine et Patrick Larnier pour leur accueil chaleureux et leur bons petits plats. Je remercie Fabien Monfreda, Dao Ngoc Minh, Nguyen Thuy Lien, Yann Ameho que j'ai côtoyés régulièrement au sein de l'IMT ou de l'ONERA et que j'apprécie beaucoup.

Je remercie ma famille au sens large et en particulier mes parents Claude et Christian Gabarrou pour tout l'amour qu'ils m'ont donné et le cadeau inestimable d'être entourée, de grandir, de partager et d'évoluer auprès de mes frère et soeurs avec dans l'ordre chronologique Claire, Agathe, Amandine, Gabrielle et Franck.

1. Introduction

1.1. Contexte de la thèse

Cette thèse se situe à la croisée des chemins de l'automatique et de l'optimisation. Elle s'intéresse à l'asservissement des systèmes linéaires, et développe un algorithme d'optimisation non lisse (de fonctions localement lipschitziennes lower $\mathcal{C}^1$) destiné à synthétiser une loi de commande structurée à contre-réaction. Par loi de commande structurée, on entend loi de commande présentant un schéma d'action particulier, où seuls quelques paramètres sont à ajuster. Un exemple illustratif et certainement le plus connu et le plus répandu, est la loi P.I.D. qui combine une action proportionnelle, intégrale et dérivée et où les paramètres à ajuster sont les gains proportionnel, intégral et dérivé.

On peut citer trois grandes périodes dans l'histoire des outils développés pour l'analyse et la conception des systèmes asservis, la période classique de 1940 à 1960, la période moderne de 1960 à 1980, et la période néoclassique de 1980 à aujourd'hui. De manière synthétique et pour placer les idées, l'automatique classique donne une représentation fréquentielle d'un système, et décrit uniquement les relations entre les signaux d'entrées et de sorties du système à travers ce que l'on appelle à juste titre sa fonction de transfert. Tandis que l'automatique moderne donne une représentation temporelle d'un système, et décrit, outre son comportement entrée-sortie, sa dynamique interne à travers ce que l'on appelle sa représentation d'état. Elle apparaît donc en cela plus complète que la première mais ne la remplace pas pour autant. En effet l'automatique classique est plus favorable à la spécification des performances d'un système asservi. Citons entre autres la bande passante déterminant le domaine d'action de l'asservissement ou encore les notions de marge de phase et marge de gain pour mesurer la robustesse de la loi de commande. C'est pourquoi l'automatique actuelle unifie les théories classiques et modernes. Elle tire le meilleur parti de chacune d'entre elles. De l'automatique classique elle emprunte la richesse de l'analyse fréquentielle des systèmes. De l'automatique moderne elle hérite la simplicité et la puissance des méthodes de synthèse par variables d'état des asservissements. Elle permet également de définir une notion forte de stabilité, la stabilité interne, qui assure que les signaux internes au système reste bornés et ne peuvent endommager, et dans le pire des cas détruire, le système.

La période classique est marquée par des personnalités comme H.W. Bode, N.B. Nichols, W.R Evans et H.S. Black, dont les travaux fondent la théorie des asservissements linéaires dans le domaine fréquentiel. Ce qu'il est important de comprendre et de retenir

de cette théorie, est que les outils de conception d'une loi de commande sont graphiques (Lieu de Nyquist, Lieu de Bode, Lieu d'Evans, Abaque de Black-Nichols) parfois même empiriques, ce qui a favorisé le développement de méthodes de l'ingénieur. De plus, les lois produites sont simples et structurées, et les éléments de la structure ont un sens physique et une unité physique. En conséquence, le traitement et l'implémentation de ces lois en sont facilités, ce qui a d'autant plus participé à l'intégration de ces techniques dans le milieu industriel. Par contre la procédure de synthèse de la loi consiste en un réglage à la main de type essai-erreur, demandant au concepteur de jongler entre les différentes représentations graphiques de l'asservissement. Cette procédure devient impraticable lorsque le système à contrôler est complexe, et que le compromis à trouver pour satisfaire les différentes spécifications antagonistes du cahier des charges, si encore il existe, est inaccessible par une approche manuelle naturellement grossière. Il est alors apparu nécessaire de disposer d'une théorie de l'optimisation adéquate ainsi que des outils numériques efficaces associés.

Les techniques de commande modernes, introduisant le formalisme de représentation d'état, sont nées des problèmes de commande optimale apparaissant notamment dans l'industrie spatiale. Il est important de noter que la commande optimale crée une rupture avec la théorie classique puisqu'elle met l'optimisation au coeur du processus de synthèse de la loi de commande. Parmi les grands noms qui ont participé à la naissance et l'élaboration de cette théorie citons entre autres R. Bellman, L.S. Pontryagin et R. Kalman. En particulier les concepts d'observabilité et de commandabilité très utilisés dans l'automatique actuelle ont été introduits par R. Kalman et montrent l'importance de la représentation d'état.

Enfin l'avènement de la théorie de la commande robuste et de la synthèse H_∞ développée entre autres par G. Zames, J.C. Doyle et M.G. Safonov unifie les théories classique et moderne. La théorie de la synthèse H_∞ formule les problèmes d'analyse et de synthèse d'une loi de commande comme des problèmes d'optimisation et y associe des méthodes numériques de résolution. Elle permet de traiter la question de l'existence d'une loi qui satisfait un cahier des charges donné et le problème de construction de la loi. De cette manière, elle propose un outil systématique de synthèse permettant à l'ingénieur d'imposer des spécifications fréquentielles complexes et d'obtenir directement un diagnostic de faisabilité et une loi de commande appropriée. L'ingénieur peut ainsi se concentrer sur la recherche du meilleur compromis et analyser les limites de son système. D'un point de vue mathématique la loi de commande optimale s'exprime comme la solution d'équations algébriques (équations de Lyapunov, de Riccati), ou de problèmes de programmation semi-définie positive (SDP) encore appelés problèmes d'inégalité matricielle linéaire (LMI). Il est à noter que ces problèmes sont des problèmes d'optimisation convexe, ce qui signifie que l'optimisation est globale et que la recherche d'un point critique équivaut à celle du minimum global. Les problèmes LMI sont résolus numériquement par des algorithmes d'optimisation convexe non différentiables telles que les méthodes de faisceaux convexes ou par des algorithmes de point intérieur.

Les techniques optimales ont été implémentées dans le milieu industriel [1] mais souf-

frent de problèmes méthodologiques. D'une part, elles synthétisent des contrôleurs non structurés du même ordre que le système à commander. En conséquence, non seulement la loi n'a aucune signification physique et n'est pas lisible et intelligible par l'ingénieur, elle est qualifiée de boîte noire, mais de plus elle présente la même complexité que le système à commander, elle est qualifiée d'ordre plein. En pratique, la mise en oeuvre de la commande sur le système réel s'accompagne d'une étape de validation, ayant éventuellement pour conséquence la retouche du correcteur rendue difficile par son niveau de complexité et son manque de lisibilité. D'autre part il a été mis en évidence [2], [3, chap. 6], que la synthèse H_∞ peut faire apparaître des problèmes d'inversion locale ou totale du modèle dans le correcteur, ce qui soulève entre autres des problèmes de robustesse paramétriques. Ces problèmes sont développés et illustrés dans la section 1.2. Pour pallier au problème de contrôleur boîte noire, les travaux [4] proposent de formuler un correcteur H_∞ sous forme estimation commande, ce qui permet de donner un sens physique et une unité physique à toutes les variables qui définissent le correcteur. Malgré tout, l'ingénieur reste contraint à une structure de commande imposée.

Les recherches se sont alors tournées vers la synthèse de contrôleurs d'ordre réduit (l'ordre du contrôleur est strictement inférieur à l'ordre du système à commander) appelé synthèse H_∞ d'ordre fixé. Malheureusement le formalisme LMI ne permet pas de formuler ce type de problème. C'est pourquoi une formulation plus générale a été introduite appelée formulation d'inégalité matricielle bilinéaire (BMI).

Definition 1. *Le problème d'optimisation*

$$\begin{aligned} &\min_{x\in\mathbb{R}^m} c^\top x \\ &\text{s.c.} \quad F_0 + \sum_{i=1}^{m} F_i x_i + \sum_{i=1}^{m}\sum_{j=i}^{m} G_{ij} x_i x_j \succeq 0 \end{aligned} \tag{1.1}$$

où F_0, F_i, G_{ij} sont des matrices symétriques de dimension n, est appelé problème d'optimisation BMI.

Un problème de commande d'ordre fixé formulé par une BMI n'est plus un problème d'optimisation convexe et sa résolution est beaucoup plus difficile. L'idée qui s'est alors installée est celle de formuler un problème LMI, associé au problème BMI, dont l'optimum (nécessairement global par convexité) est un majorant de l'optimum global de ce dernier. Le problème ainsi défini, appelé approche par relaxation convexe, est généralement l'expression d'une condition suffisante pour le problème originel. La qualité de la relaxation est alors liée au pessimisme de la condition [5–7].

Dans tous les cas de figure, les formulations LMI et BMI font intervenir des variables auxiliaires au problème d'optimisation, dîtes variables de Lyapunov, particulièrement handicapantes du point de vue numérique. D'une part le nombre de ces variables croit comme le carré de l'ordre du système à commander, ce qui pose des difficultés en termes de puissance de calculs pour les systèmes de grande taille présentant un grand nombre d'états. D'autre part l'ordre de grandeur des variables de Lyapunov n'a aucune raison

d'être le même que celui des variables du correcteur ce qui peut poser de difficiles problèmes numériques.

Dans ce contexte, il est apparu nécessaire de développer des algorithmes d'optimisation dont les variables de synthèse sont uniquement celles qui définissent le contrôleur et qui vont au delà de la frontière fixée par la convexité en traitant directement la non convexité intrinsèque des problèmes de commande avec contrainte de structure. Pour s'affranchir de la difficulté de calculer l'optimum global d'une fonction non convexe, les algorithmes de commande structurée qui se sont développés ces dernières années sont des algorithmes d'optimisation locale. On pourrait croire que cette restriction est pénalisante mais l'expérience a montré que dans beaucoup de cas pratiques, la recherche d'un optimum local suffit. Les algorithmes de commande structurée actuellement disponibles dans le domaine public sont HIFOO [8, 9] en accès libre sur le site `http://www.cs.nyu.edu/overton/software/hifoo/` et HINFSTRUCT [10] disponible sur Matlab dans la Toolbox Robust Control à partir de la version 7.11 (R2010b). Ces algorithmes permettent entre autres de stabiliser et d'optimiser au sens de la norme H_∞ des systèmes linéaires à temps invariants.

1.2. Justification de la synthèse H_∞ structurée par l'exemple

L'objectif ici est de montrer l'importance, sur le plan méthodologique, de savoir isoler dans le processus de synthèse d'une loi de commande, des paramètres de réglage dimensionnants et, sur le plan algorithmique, de disposer d'outils d'optimisation qui travaillent sur un jeu de paramètres de réglage judicieusement choisis par l'ingénieur automaticien. A cette fin citons l'exemple académique traité dans [11] où le système à commander,

$$G(s) = \frac{1}{s^2 + 0.01s + 1}, \tag{1.2}$$

est un système linéaire du second ordre de fréquence propre $\omega_0 = 1$, de gain statique $K = 1$, et de coefficient d'amortissement $\xi = 0.01/2 \ll 1/\sqrt{2}$. Ce système présente donc une résonance aiguë à la fréquence $\omega_r = \omega_0\sqrt{1 - 2\xi^2}$, ce qui lui confère une dynamique bien particulière. On s'intéresse au problème de commande suivant :

Trouver une loi de commande K asservissant la réponse de G sur la consigne w_1 selon le schéma fonctionnel de la Figure 1.1 et selon les spécifications de performance :

$$\begin{cases} \text{la bande passante de l'asservissement est de } 10 \text{ radians par seconde,} \\ \text{l'erreur statique (réponse de } G \text{ à un échelon de } w_1) \text{ est inférieure à } 1\%. \end{cases} \tag{1.3}$$

Si l'on applique les techniques fréquentielles classiques de synthèse d'un correcteur, opérant sur la boucle ouverte $L = KG$ aux moyens d'outils graphiques tels que les

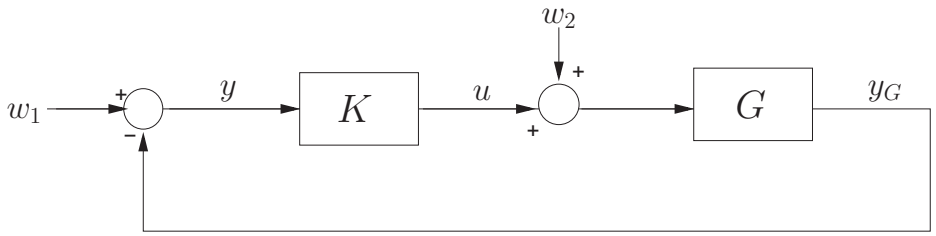


FIGURE 1.1.: Schéma fonctionnel d'un asservissement.

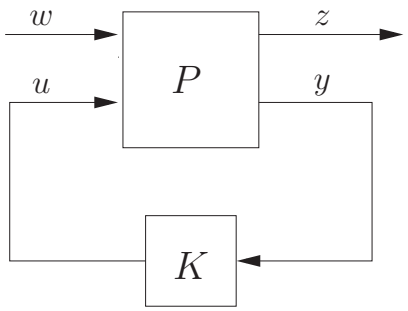


FIGURE 1.2.: Problème de commande standard. La fonction de transfert entre w et z se note $F_l(P, K)$ et s'appelle la transformée linéaire fractionnaire de P et de K.

diagrammes de Bode, Black et Nichols ou le lieu des racines, alors le problème (1.3) est résolu de manière simple, et sans mauvaise surprise sur les propriétés de la boucle fermée. En revanche, si l'on cherche à appliquer les techniques de synthèse H_∞, où la loi de commande s'exprime comme la solution d'équations algébriques (équations de Lyapunov, de Riccati), ou de problèmes de minimisation d'une fonction linéaire sous des contraintes matricielles de positivité (LMI), on montre dans [11] que la synthèse peut conduire à des solutions marginales. Ces techniques optimales s'appuient sur le formalisme du problème de commande standard présenté Figure 1.2 et visent à modeler en fréquence les fonctions caractéristiques qui décrivent le comportement entrée sortie des différents transferts de la boucle d'asservissement pris en compte dans le problème d'optimisation. Le système P de la forme standard comprend à la fois la dynamique du système à commander et les dynamiques des filtres de pondérations (inverses des gabarits imposés aux fonctions caractéristiques). Formellement le problème de synthèse H_∞ se formule ainsi :

$$\begin{array}{l} \text{Trouver un correcteur } K \text{ minimisant} \\ \|F_l(P,K)\|_\infty = \sup\limits_{\text{Re } s>0} \sigma_1[F_l(P(s),K(s))]. \end{array} \tag{1.4}$$

Dans notre exemple, le système à commander est G et la fonction caractéristique est le transfert $S(K) = (1 + GK)^{-1}$ entre la consigne w_1 et l'erreur de poursuite y. L'interprétation fréquentielle de la norme H_∞ permet de prendre en compte simplement les spécifications du problème de commande (1.3) par un gabarit

$$S_d(s) = \frac{s + 0.1}{s + 10} \tag{1.5}$$

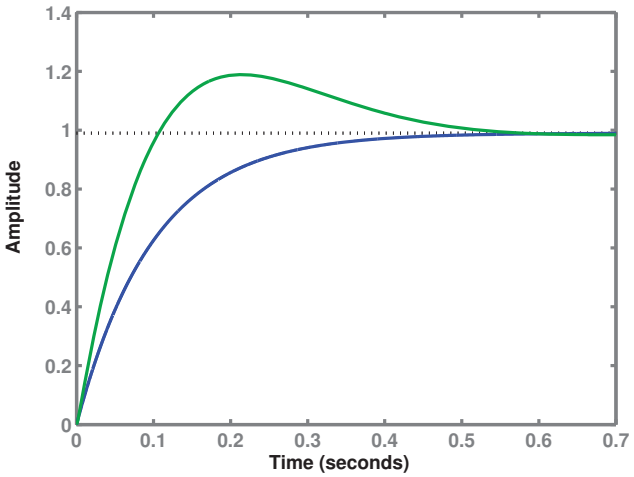


FIGURE 1.3.: Réponses indicielles de $(1+GK_1)^{-1}$ en bleue et de $(1+GK_2)^{-1}$ en vert

sur S (cf. annexe C). Dire que le correcteur K satisfait le gabarit S_d signifie que la norme H_∞ de $S_d^{-1}S(K)$ est inférieure à 1. Si la synthèse H_∞ fournit un correcteur K tel que $\left\|S_d^{-1}S(K)\right\|_\infty \ll 1$, alors d'une part celle-ci est réussie, et d'autre part on peut se dire que les spécifications imposés par S_d ne sont pas assez contraignantes et que l'on peut demander plus de performances. La loi de commande optimale qui vise à imposer le gabarit S_d à $S(K)$ et vers laquelle convergent tous les algorithmes de synthèse H_∞ d'ordre plein est de la forme [11] :

$$K_1(s) = \frac{k(s+0.01s+1)}{(s+0.1)(s^2/\omega^2+2\xi s/\omega+1)}. \tag{1.6}$$

Les paramètres k, ξ et ω ne dépendent que de la valeur ε des termes de régularisation et de la tolérance $\delta\gamma$ spécifiée dans l'algorithme utilisé pour la synthèse H_∞.

Bien que vérifiant les spécifications d'erreur statique (courbe bleue Figure 1.3) et de bande passante (courbe bleue Figure 1.4), le correcteur (1.6) ne présente pas les caractéristiques que l'on attend d'un asservissement au sens ingénierie du terme. En effet, on constate une inversion du système à contrôler dans le contrôleur, et de cette inversion découle deux inconvénients qui rendent inapplicable en pratique le correcteur (1.6). D'une part celui-ci est sensible au modèle G, ce qui soulève des problèmes de robustesse de la loi de commande aux variations de ce modèle. D'autre part les pôles mal amortis de G s'annulent avec les zéros de K_1 dans le produit GK_1. Alors que cette annulation s'effectue pour le transfert $S(K)$ qui est pris en compte dans la synthèse, il n'en va pas de même pour le transfert $G(1+KG)^{-1}$ entre une perturbation w_2 à l'entrée de G et sa sortie y_G, ce qui pose des problèmes de réjection de perturbations. La réponse impulsionnelle de $G(1+K_1G)^{-1}$ est catastrophique (courbe bleue Figure 1.5) puisque le mode oscillant de G évolue avec son comportement en boucle ouverte.

Il est légitime de se demander pourquoi une synthèse classique se protège naturellement du problème d'inversion locale ou totale du modèle dans le correcteur. La réponse

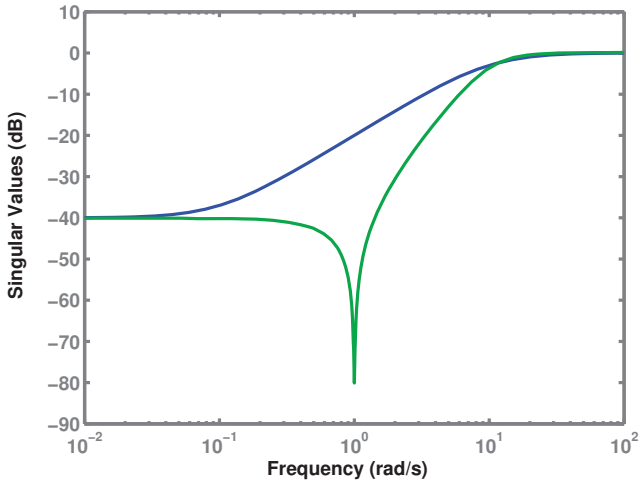


FIGURE 1.4.: Réponses fréquentielles de $(1+GK_1)^{-1}$ en bleue et de $(1+GK_2)^{-1}$ en vert

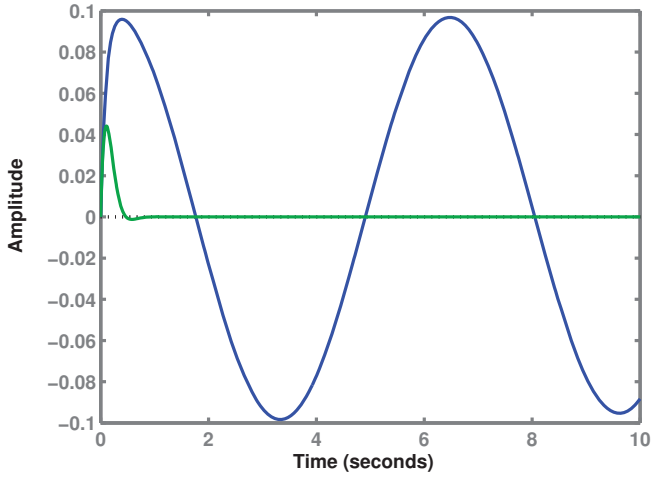


FIGURE 1.5.: Réponses impulsionnelles de $G(1+K_1G)^{-1}$ en bleue et de $G(1+K_2G)^{-1}$ en vert

réside dans la procédure essai-erreur, certes limitée, mais qui permet de trouver le correcteur le plus simple pour modeler la réponse fréquentielle du transfert en boucle ouverte $L = KG$ tout en maîtrisant la dynamique du système en boucle fermée. Une telle démarche demande au concepteur de jongler entre différentes représentations (réponse fréquentielle de la boucle ouverte, lieu des racines, ...). Cependant, elle cherchera toujours à amortir les pôles de G en boucle fermée plutôt que de les annuler avec des zéros de K. Dans [11] on explique comment une synthèse classique amène à choisir un correcteur de type proportionnel dérivé

$$K_2(s) = \frac{k_p + k_v s}{1 + \tau s} \tag{1.7}$$

avec $k_p = 100$, $k_v = 15$ et $\tau = 0.001$. On voit sur la Figure 1.4 que le système asservi par le correcteur (1.7) satisfait le gabarit S_d et sur la Figure 1.5 que celui-ci rejette de manière satisfaisante la perturbation w_2 en entrée de G. Alors que la réponse fréquentielle de $(1 + GK_1)^{-1}$ se confond avec celle du gabarit S_d, la réponse fréquentielle de $(1+GK_2)^{-1}$ asservie par le correcteur (1.7) est foncièrement différente du gabarit S_d et révèle une crevasse bénéfique liée à la résonance naturelle du système G. Ceci met en lumière l'une des difficultés majeures de l'utilisation du problème de commande standard, à savoir le choix des gabarits. La spécification du gabarit ne peut se faire sans avoir préalablement analysé le comportement dynamique du système. On peut également évaluer la sous-optimalité de $K_2(s)$ par rapport à $K_1(s)$ sur le problème H_∞

$$\left\| S_d^{-1}(1 + GK_2)^{-1} \right\|_\infty = 1.0643.$$

Cette valeur est très proche de l'optimum global, ce qui se traduit sur la Figure 1.4 par le fait que $(1+GK_2)^{-1}$ respecte qualitativement le gabarit. Suite à cet exemple, il apparaît intéressant d'étudier et de développer des algorithmes d'optimisation de lois de commande structurées venant compléter le savoir-faire ingénieur. Ce savoir-faire permettra d'analyser les propriétés dynamiques du système à asservir et de concevoir la structure de commande appropriée. Les gains de la structure seront ensuite optimisés. Le défaut des problèmes de commande structurée est qu'ils sont par nature non convexes. Les algorithmes développés depuis quelques années (Hifoo, Hinfstruct, Bundle) sont donc localement optimaux. Mais il se trouve que dans beaucoup de cas pratiques cela suffit.

On pourrait penser que le problème d'inversion mis en évidence dans cet exemple vient de la sur-paramétrisation du correcteur (1.6). En effet ce correcteur est d'ordre 3 alors que le correcteur (1.7) est d'ordre 1. L'optimalité globale de la loi de commande (1.6) se paye au prix fort de l'ordre plein de celle-ci (même nombre d'états que le système à commander P du problème de commande standard). Le système à commander est G d'ordre 2 augmenté du filtre de pondération S_d^{-1} d'ordre 1, ce qui donne un correcteur (1.6) d'ordre 3. Cette pensée est juste. Si l'on ajoute au problème de synthèse H_∞ (1.4) une contrainte structurelle sur la loi de commande, ici ce sera l'ordre 1, et que l'on applique un algorithme de commande structuré, ici ce sera l'algorithme présenté dans le chapitre 2, alors le problème d'inversion du modèle dans le correcteur disparait. Le comportement de la boucle fermée avec le correcteur optimisé

$$K_3 = \frac{1.04e04s + 5.886e04}{s + 611.6}, \tag{1.8}$$

est présenté Figure 1.6. Par ailleurs, d'autres problèmes peuvent apparaître dans le réglage des pondérations. En effet, on peut facilement montrer que la prise en compte d'une pondération fréquentielle supplémentaire sur le transfert $GS = G(I + KG)^{-1}$ permet d'éviter la compensation pôle-zéro mentionnée précédemment. Par contre le réglage de cette pondération n'est pas facile et n'a pas de lien direct avec les spécifications initiales. Il se révèle finalement plus laborieux que le réglage (même par une approche essai-erreur) des deux gains de la loi proportionnelle-dérivée.

Et si nous voulions spécifier une erreur statique nulle, comment l'intégrerions nous dans le gabarit ? Tout ingénieur en automatique sait parfaitement que, pour annuler l'erreur statique, il suffit de mettre un intégrateur dans le contrôleur K, s'il n'est pas déjà dans le système à contrôler G (cf. annexe C). Par contre, le formalisme de la synthèse H_∞ ne permet pas de prendre en compte ce type de spécification, puisqu'elle requiert l'utilisation d'un filtre de pondération stable. Or celui qu'il faudrait utiliser, $(s + 10)/s$, est instable. En pratique, $(s + 10)/s$ est régularisé en déplaçant le pôle à zéro d'un ε vers la gauche, i.e. en utilisant le filtre de pondération $(s + 10)/(s + \varepsilon)$ avec $\varepsilon > 0$ et $\varepsilon \approx 0$.

Et si nous rajoutions au problème initial une sortie mesurée sur la dérivée de $y_G(t)$, comment serait utilisé cette sortie par le contrôleur optimal ? Il faut bien noter que cela ne pose aucun problème de prendre en compte cette nouvelle mesure par l'approche fréquentielle classique : il suffit de répartir directement les gains proportionnel k_p et dérivé k_v sur les 2 mesures. On peut même dire que cela simplifie le problème. Par contre, la synthèse H_∞ d'ordre plein donne le correcteur

$$K_4(s) = \left[\frac{120319875081.7054(s^2 + 0.01s + 1)}{(s + 0.1)(s^2 + 1.569e05s + 1.216e10)} \quad 0 \right]$$

qui n'utilise pas la seconde mesure, ce qui n'est pas pertinent.

Avant de clore cette section, n'oublions pas de parler de la structure GNC (Guidage, Navigation, Commande) qui est largement utilisée pour concevoir l'asservissement d'un engin aéronautique. Cette architecture organise la commande selon plusieurs boucles imbriquées (voir Figure 3.1) et trouve sa justification dans le découplage fréquentiel des dynamiques lentes et rapides de l'avion. Une telle structure peut être remise en cause lorsque la spécification de performance fait que le découplage fréquentiel entre les boucles n'est plus valide. Le réglage d'une boucle dépend alors du réglage d'une autre boucle. Avant de remettre en cause cette structure, on peut se poser la question suivante. Est-on capable d'optimiser conjointement les différentes lois de commande intervenant dans l'asservissement pour satisfaire les objectifs de performance ? Pour pouvoir répondre à cette question, il est encore une fois nécessaire de disposer d'un algorithme d'optimisation de lois de commande structurées.

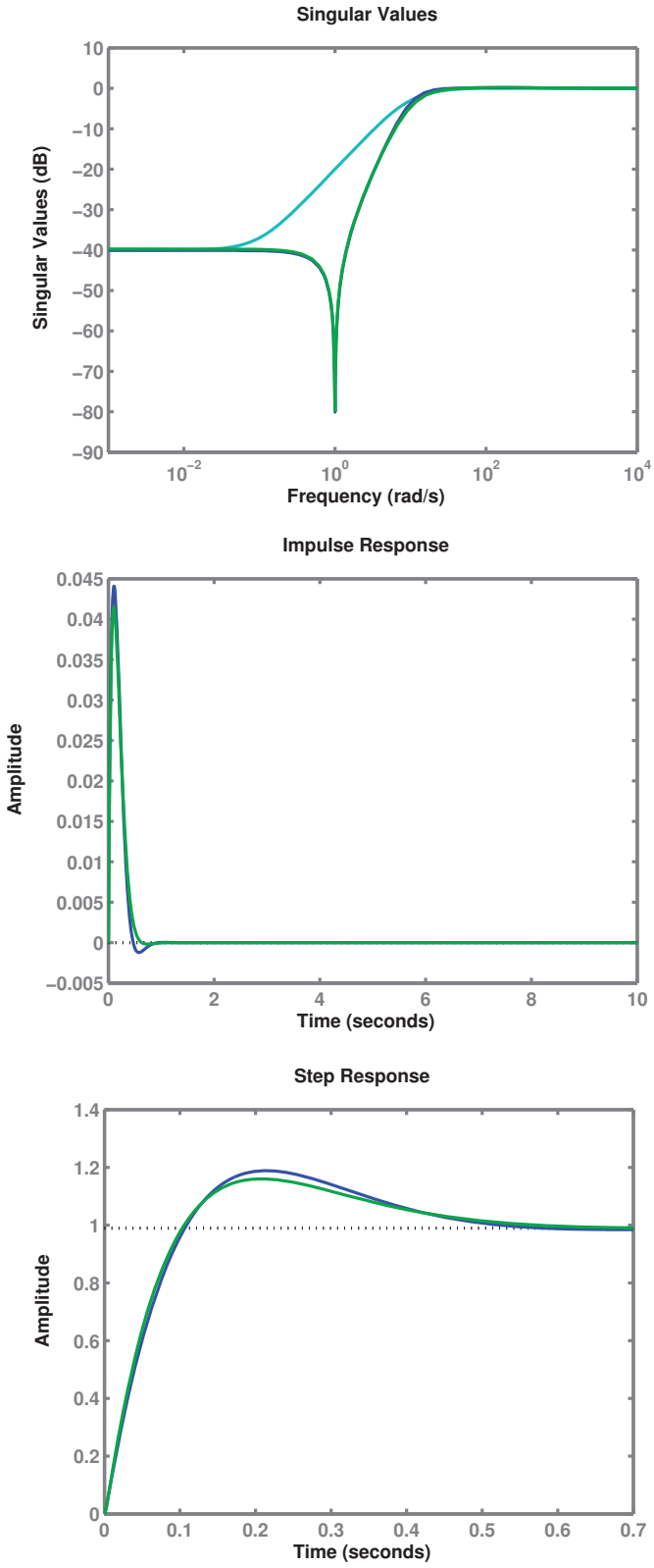


FIGURE 1.6.: Synthèse d'ordre 1. En bleu clair S_d, en bleu foncé K_2, en vert K_3.

1.3. Plan du manuscrit

L'ordre des chapitres de ce manuscrit ne correspond pas à l'ordre chronologique de la thèse. Les chapitres 3 et 4, qui ont été rédigés en anglais en vue de les publier dans des revues, sont chronologiquement antérieurs aux chapitres 1 et 2.

Le chapitre 2 a été construit lors de la rédaction du manuscrit de thèse pour présenter le nouvel algorithme étudié, développé et appliqué au cours de la thèse. La résolution de l'exemple académique de la section 1.2 a mis en évidence des phénomènes intéressants que l'on a alors décidés de présenter dans le chapitre 2.

Les applications traitées dans cette thèse portent toutes sur la synthèse d'une loi de commande d'un avion de transport en vol longitudinal. Le chapitre 3 expose un problème de synthèse conjointe de la loi de commande de vol et du pilote automatique en un point de vol donné. Le chapitre 4 présente la synthèse d'un contrôleur paramétré en vitesse et altitude asservissant une famille de modèles linéaires associée à un ensemble de points dans le domaine de vol.

Les annexes A, B et D détaillent le calcul des gradients et sous-gradients des fonctions utilisées dans les algorithmes non lisses de commande structurée. L'annexe C éclaire certains points énoncés dans la section 1.2. Enfin l'annexe E compare les performances de notre algorithme avec les solveurs Hifoo et Hinfstruct sur des problèmes issus de la librairie $Compl_eib$.

2. Algorithme de faisceau non convexe avec contrôle de proximité

2.1. Introduction

Considérons une fonction non lisse $f : \mathbb{R}^n \to \mathbb{R}$, $n \in \mathbb{N}^*$, i.e. une fonction $f \notin \mathcal{C}^1(\mathbb{R}^n)$ non continuement différentiable. Parmi les méthodes reconnues les plus efficaces pour résoudre les problèmes d'optimisation non lisses se trouvent les méthodes de faisceau [12–15]. Elles ont fait leurs preuves dans le cas convexe mais leur généralisation au cas non convexe est un thème de recherche actuel. De manière générale, elles se basent sur la donnée d'un oracle : pour un x donné dans $\mathbb{R}^n$, on sait calculer $f(x)$ et un sous-gradient arbitraire $g(x) \in \partial f(x)$ où $\partial f(x)$ est le sous-différentiel de Clarke [16] de f en x. D'un point de vue des applications, elles sont très intéressantes car elles n'exigent qu'une connaissance minimale de $\partial f(x)$. L'ensemble $\partial f(x) \subset \mathbb{R}^n$ contient, à l'instar du gradient $\nabla f(x) \in \mathbb{R}^n$ dans le cas régulier, une information variationnelle au premier ordre sur f. Par contre, $x \to g(x)$ n'est a priori pas continue, $-g(x)$ n'est pas nécessairement une direction de descente, et $g(x)$ n'est pas nécessairement nul même si x est un minimum local de f. En effet lorsque f est non lisse au point x alors d est une direction de descente pour f à partir de x si et seulement si $d^\top g < 0, \forall g \in \partial f(x)$. La condition nécessaire d'optimalité du premier ordre devient quant à elle $0 \in \partial f(x)$. L'information donnée par l'oracle en un seul point x est insuffisante pour produire un pas de descente satisfaisant pour f à partir de ce point. Il est alors nécessaire d'accumuler dans ce que l'on appelle un faisceau, de l'information variationnelle sur f, en faisant appel à l'oracle, dans un voisinage de ce point. Ce faisceau est enrichi et mis à jour de manière itérative et permet de définir un modèle local polyédral de f en x dont la minimisation fournit un pas d'essai. Dans cette thèse, on propose d'enrichir le faisceau en décalant vers le bas une tangente de f en un pas d'essai ne constituant pas un pas de descente satisfaisant. Le décalage est indispensable dans le cas non convexe pour préserver la consistance, on dit encore l'exactitude, du modèle vis à vis de f au point x.

2.2. Trame de l'algorithme

On présente dans cette section un algorithme de faisceau non convexe avec contrôle de proximité pour la minimisation d'une fonction $f : \mathbb{R}^n \to \mathbb{R}$ définie et localement lipschitzienne lower $\mathcal{C}^1$ sur son domaine de définition $\mathcal{D} \subset \mathbb{R}^n$.

Definition 2. *$f : \mathbb{R}^n \to \mathbb{R}$ est lower $\mathcal{C}^1$ en x_0 si il existe un ensemble K compact, un réel $\delta > 0$ et une fonction $L : B(x_0, \delta) \times K \to \mathbb{R}$ tels que $\forall y \in K$, $L(\cdot, y) \in \mathcal{C}^1(B(x_0, \delta))$ et tels que*

$$f(x) \;=\; \max\{L(x, y),\; y \in K\} \tag{2.1}$$

La propriété de régularité lipschitz de f donne l'existence en tout $x \in \mathbb{R}^n$ du sous-différentiel de Clarke $\partial f(x)$ de f en x. Elle nous permet ainsi de mettre au point un algorithme itératif de descente, utilisant le sous-différentiel de Clarke pour extraire une information variationnelle au premier ordre sur f. L'algorithme génère, à partir d'un point initial x^0, une suite $\{x^j\}$ d'itérés dits "sérieux" telle que pour tout $j \in \mathrm{N}^*$, $f(x^{j+1}) \leq f(x^j)$, et telle que tous ses points d'accumulation x^* vérifient la condition nécessaire d'optimalité du premier ordre $0 \in \partial f(x^*)$.

Le processus générant l'itéré sérieux x^{j+1} à partir de l'itéré sérieux courant x^j consiste en la résolution itérative de programmes quadratiques convexes $\mathcal{P}_k$

$$\mathcal{P}_k \qquad \min_{y \in \mathbb{R}^n} \quad \Phi_k(y, x^j) + \frac{\tau_k}{2} \left\| y - x^j \right\|^2, \tag{2.2}$$

où $\Phi_k(\cdot, x^j)$ se décompose en la somme d'une fonction polyédrale

$$\phi_k^{[1]}(\cdot, x^j) = \max\{a + g^T(\cdot - x^j) :\; (a, g) \in \mathcal{G}_k\},$$

$\mathcal{G}_k \subset \mathbb{R} \times \mathbb{R}^n$ un ensemble fini, et d'une fonction quadratique

$$\phi^{[2]}(\cdot, x^j) = \frac{1}{2}(\cdot - x^j)^\top Q(x^j)(\cdot - x^j)$$

telle que $\tau_k I + Q(x^j) \succ 0$. Les solutions y^k des problèmes $\mathcal{P}_k$ forment une suite $\{y^k\}$ qui converge vers x^{j+1}. La convergence a lieu en un nombre fini d'itérations k à condition que x^j ne soit pas déjà localement optimal pour f, ce qui est toujours le cas lorsque $0 \notin \partial f(x^j)$. En effet les méthodes d'optimisation non lisse s'appuient sur l'existence d'une direction de descente pour f en x^j. Si $0 \notin \partial f(x^j)$ alors en invoquant le théorème de séparation des convexes [17, p. 4], on montre qu'il existe un hyperplan affine H séparant strictement les deux ensembles convexes compacts $\partial f(x^j)$ et $\{0\}$ [16, p. 27]. Un élément $d \in \mathbb{R}^n$ définissant cet hyperplan, i.e. $H = \{x \in \mathbb{R}^n : x^\top d = c\}$ est une direction de descente pour f en x^j. C'est pourquoi on pose comme test d'arrêt :

$$\textit{Test d'arrêt} : \qquad 0 \in \partial f(x^j). \tag{2.3}$$

On fera la remarque importante que ce test d'arrêt est un outil théorique pour l'analyse de la convergence de l'algorithme. En pratique il est inapplicable si l'on ne connaît

pas le sous-différentiel dans sa totalité. Ce qui est alors implémenté est un test de "slow progress" et consiste à fixer des tolérances sur les erreurs relatives de la solution courante x^j et de la valeur courante $f(x^j)$. L'analyse de la convergence nous assure que la suite $\{x^j\}$ générée par l'algorithme, et construite à partir de l'information *unique* fournit par l'oracle, est telle que tous ses points d'accumulation sont des points critiques de f. La connaissance du sous-différentiel dans sa totalité ne servirait donc qu'au niveau du test d'arrêt. L'algorithme présente ainsi deux boucles, une boucle de compteur j appelée boucle externe de test d'arrêt (2.3) et une boucle de compteur k appelée boucle interne dont on va développer les mécanismes. Pour s'économiser en indices, on notera x au lieu de x^j et x^+ au lieu de x^{j+1}.

La fonction polyédrale $\phi_k^{[1]}(\cdot, x)$ est construite à partir d'un "faisceau" ou "paquet" $\mathcal{G}_k$ d'information variationnelle au premier ordre sur f dans un voisinage de x. Ces informations sont accumulées au cours des précédentes itérations $1, ..., k$ de la boucle interne et cette accumulation est caractéristique des méthodes de faisceau. Les branches du maximum $\phi_k^{[1]}(\cdot, x)$ sont appelées plans de coupe ou plans sécants. Dans le cas d'une méthode de faisceau convexe, un plan de coupe de f en $z \in \mathbb{R}^n$ est support affine de f en z de la forme $f(z) + g^\top(\cdot - z)$ avec $g \in \partial f(z)$. Un tel plan généralise la notion d'application "linéaire tangente" en application "sous-linéaire tangente" donnant naissance à la notion de "sous-différentiel" généralisant celle de "gradient" [12, chap. 5]. Le qualificatif "sous" fait appel à la notion de "relaxation", de propriété "moins forte" que la propriété de linéarité ou de différentiabilité. Dans tous les cas, que f soit convexe ou non, $\phi_k^{[1]}(\cdot, x)$ doit être consistant avec f en x, i.e. doit satisfaire la condition suivante

$$\textit{Exactitude} : \qquad \phi_k^{[1]}(x, x) = f(x) \quad \text{et} \quad \partial_1 \phi_k^{[1]}(x, x) \subset \partial f(x). \tag{2.4}$$

Remarquons que si $\phi_k^{[1]}(\cdot, x)$ est consistant avec f en x, $\Phi_k(\cdot, x)$ l'est également puisque $\partial_1 \Phi_k(y, x) = \partial_1 \phi_k^{[1]}(y, x) + Q(x)(y - x)$. Le sous-différentiel de $\phi_k^{[1]}(\cdot, x)$ en $y \in \mathbb{R}^n$ est, en tant que fonction polyédrale, très simple à calculer [16, p. 47] et vaut

$$\partial_1 \phi_k^{[1]}(y, x) = \{g \in \mathbb{R}^n : \ a + g^\top(y - x) = \phi_k^{[1]}(y, x), \ (a, g) \in \mathcal{G}_k\}.$$

La condition (2.4) est donc vérifiée si et seulement si :

. $\forall (a, g) \in \mathcal{G}_k \quad a \leq f(x)$,

. $\exists (a, g) \in \mathcal{G}_k$ tel que $a = f(x)$,

. $\forall (a, g) \in \mathcal{G}_k$ tel que $a = f(x)$, $g \in \partial f(x)$.

La fonction quadratique $\phi^{[2]}(\cdot, x)$ a pour vocation de constituer une approximation de la courbure de f au voisinage de x, i.e. $\phi^{[2]}(\cdot, x)$ contient une information du second ordre. Elle ne dépend que de l'itéré sérieux courant x et reste fixe pendant tout le processus de mise à jour de x en x^+. La convergence de l'algorithme requiert que l'opérateur $x \mapsto Q(x)$, $\mathbb{R}^n \to \mathbb{S}^n$, soit borné sur les ensembles bornés. Le modèle $\Phi_k(\cdot, x)$ constitue ainsi une approximation rudimentaire de f en x que l'on sait minimiser. C'est donc lui que l'on minimise pour générer un candidat y^k à la succession de x, et c'est pourquoi on le qualifie de modèle de "travail" de f en x à l'itération k de la boucle interne. Le candidat y^k sera élu successeur de x et deviendra le nouvel itéré sérieux x^+ si $\Phi_k(\cdot, x)$ constitue une "bonne" approximation de f dans un voisinage de x contenant y^k.

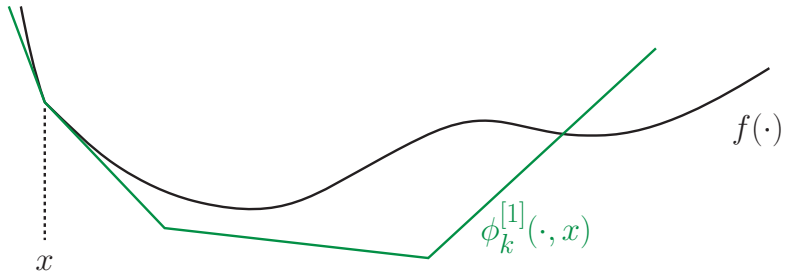


FIGURE 2.1.: $\phi_k^{[1]}(\cdot, x)$: modèle convexe non lisse d'ordre 1 de f en x.

On rappelle que y^k est l'unique solution du programme quadratique convexe $\mathcal{P}_k$ dont la fonction objectif est la somme de $\Phi_k(\cdot, x)$ avec ce que l'on appelle le terme de proximité $(\tau_k/2)\,\|\cdot - x\|^2$. Arrêtons nous quelques instants sur ce terme. Les raisons de son utilisation sont multiples. Tout d'abord, il permet de borner inférieurement le programme $\mathcal{P}_k$ puisque l'on impose $\tau_k I + Q(x) \succ 0$. La première partie $\phi_k^{[1]}(\cdot, x)$ du modèle de travail n'est bornée inférieurement que lorsque $0 \in \mathrm{co}\{g : (a, g) \in \mathcal{G}_k\}$, où co désigne l'enveloppe convexe. Il fait même beaucoup plus que borner le problème, il le rend strictement convexe assurant ainsi l'existence et l'unicité de la solution y^k à distance finie. Ensuite, il permet de remédier au phénomène d'instabilité des plans sécants qui rend l'algorithme intolérablement lent au voisinage d'un optimum local x^* [13, chap. 15]. Enfin, il permet de traiter la non convexité de f par une mise à jour adaptée de τ_k à chaque itération k agissant ainsi comme une contrainte de région de confiance.

La confiance que l'on porte au modèle de travail $\Phi_k(\cdot, x)$ est mesurée par le quotient

$$\textit{Test d'acceptation} : \qquad \rho_k \doteqdot \frac{f(x) - f(y^k)}{f(x) - \Phi_k(y^k, x)}, \tag{2.5}$$

qui calcule le rapport entre le progrès réel $f(x) - f(y^k)$ apporté par y^k et le progrès prédit ou attendu $f(x) - \Phi_k(y^k, x)$. Si $\Phi_k(\cdot, x)$ représente f avec précision en y^k, nous nous attentons à $\rho_k \approx 1$, mais nous acceptons y^k comme le nouvel x^+ dès que $\rho_k \geq \gamma$, où $0 < \gamma < 1$ est fixée au début de l'algorithme. Le pas y^k est alors qualifié de "pas sérieux". Si au contraire $\rho_k < \gamma$, alors y^k est rejeté et il est qualifié de "pas nul". A ce moment là, on itère sur k en construisant $\mathcal{P}_{k+1}$ de manière à ce que y^{k+1} soit "meilleur" que y^k, i.e. $\rho_{k+1} > \rho_k$. Plaçons nous dans le cas où $f(x) - \Phi_k(y^k, x)$ est > 0. Remarquons alors que si ρ_k est < 0 cela signifie que y^k est un pas de montée. Comme $0 < \gamma$, y^k sera bien rejeté ! Maintenant si $\rho_k \gg 1$ cela signifie que la décroissance réelle est bien plus importante que la décroissance prédite. Comme $\gamma < 1$, y^k sera bien accepté ! Enfin si y^k est un pas de descente mais qu'il fournit un progrès très faible, i.e. $0 < f(x) - f(y^k) \ll 1$, y^k sera tout de même accepté si $\rho_k \geq \gamma$, i.e. si le progrès prédit $f(x) - \Phi_k(y^k, x)$ est également très faible. Ici intervient vraiment le concept de confiance. Ce n'est pas tant la quantité de la décroissance $f(x) - f(y^k)$ qui fait de y^k un pas de descente "satisfaisant" que la qualité du modèle $\Phi_k(\cdot, x)$ vis à vis de f.

Nous allons maintenant expliquer les mises à jour $\Phi_k(\cdot, x) \leftarrow \Phi_{k+1}(\cdot, x)$ et $\tau_k \leftarrow \tau_{k+1}$ qui ont lieu lorsque y^k est un pas nul. Le modèle de travail est amélioré en intégrant dans

la partie d'ordre 1, $\phi_k^{[1]}(\cdot, x)$, un nouveau plan sécant, dont le rôle est de mettre hors jeu y^k. Dans le cas f convexe, un plan sécant est une tangente de f en y^k, i.e. un support affine de f en y^k : $m_k(y, x) = f(y^k) + g_k^\top(y - y^k)$ avec $g_k \in \partial f(y^k)$. Par convexité de f, tout plan sécant de f est un minorant de f et donc pour tout k, $\phi_k^{[1]}(\cdot, x)$ est une "sous-approximation" de f sur son domaine de définition. Par conséquent l'ajout d'un plan sécant quelconque dans le modèle de travail ne "brise" pas la propriété d'exactitude (2.4) imposée à $\phi_k^{[1]}(\cdot, x)$ et l'améliore toujours dans le sens où $\phi_k^{[1]}(\cdot, x) \leq \phi_{k+1}^{[1]}(\cdot, x) \leq f$. Sans la propriété de convexité de la fonction à minimiser, il est plus délicat d'obtenir un plan sécant approprié. En effet une tangente de f n'est plus nécessairement un minorant de f. Dans l'esprit de conserver l'information exacte sur le "taux d'accroissement" de f au pas rejeté y^k tout en conservant la propriété d'exactitude du modèle de travail, on a opté pour l'utilisation de tangentes de f décalées vers le bas. En voici la construction. En adéquation avec la décomposition du modèle de travail, nous décomposons f

$$f(\cdot) = f_1(\cdot, x) + f_2(\cdot, x),$$

où $f_2(\cdot, x) = \frac{1}{2}(\cdot - x)^\top Q(x)(\cdot - x)$ et $f_1 = f - f_2$. Pour un pas nul y^k donné, on choisit un sous gradient $g_k \in \partial_1 f_1(y^k, x)$. La fonction affine $t_k(\cdot) = f_1(y^k, x) + g_k^\top(\cdot - y^k)$ est alors une tangente de $f_1(\cdot, x)$ en y^k. On rappelle que l'on ne peut pas utiliser $t_k(\cdot)$ directement en tant que plan sécant puisque dans le cas f non convexe on ne peut même pas savoir si $t_k(x) \leq f(x)$. Nous définissons donc le décalage comme

$$s_k = [t_k(x) - f(x)]_+ + c\|y^k - x\|^2,$$

où $c > 0$ est une constante fixée au début de l'algorithme. Nous définissons ensuite le plan sécant comme

$$\textit{Plan sécant} : \qquad m_k(\cdot, x) = t_k(\cdot) - s_k. \tag{2.6}$$

Notons que $\nabla m_k(\cdot, x) = \nabla t_k(\cdot) = g_k$, tandis que $m_k(x, x) \leq f(x) - c\|y^k - x\|^2 \leq f(x)$. Le plan sécant se réécrit comme $m_k(\cdot, x) = a_k + g_k^\top(\cdot - x)$, où

$$a_k = t_k(x) - s_k = t_k(x) - [t_k(x) - f(x)]_+ - c\|y^k - x\|^2.$$

La tangente décalée $m_k(\cdot, x)$ contient une information plus complète que la tangente t_k puisqu'elle tient compte de la "proximité" de y^k avec x.

Les capacités de stockage et de traitement des données des ordinateurs étant limitées, on a recourt à une stratégie "d'agrégation" de plans sécants. Cette stratégie transforme le faisceau $\mathcal{G}_k$ de manière à en extraire les caractéristiques les plus pertinentes. Le programme

$$\begin{aligned} \min_{(t,y)\in\mathbb{R}\times\mathbb{R}^n} \; & t + \frac{1}{2}(y - x)^\top (Q(x) + \tau_k I)(y - x) \\ \text{s.c.} \quad & a + g^\top(y - x) \leq t, \quad (a, g) \in \mathcal{G}_k \end{aligned} \tag{2.7}$$

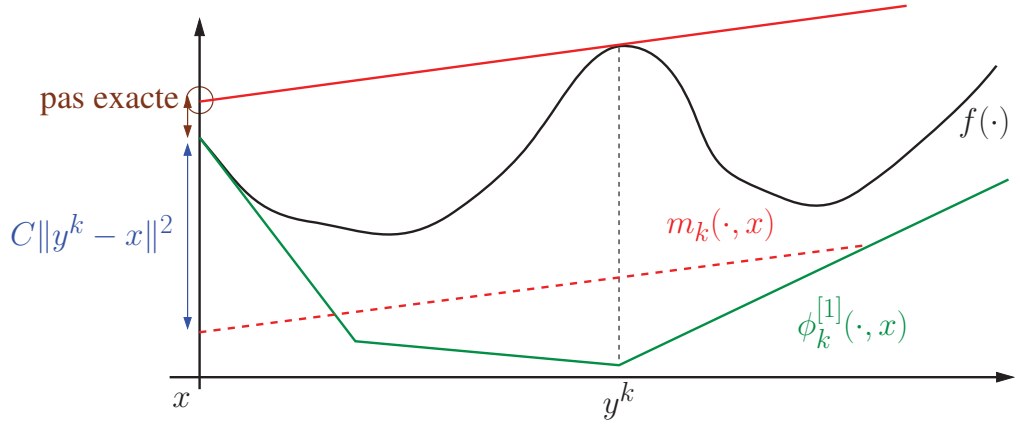


FIGURE 2.2.: $m_k(\cdot, x) = t_k(\cdot) - [t_k(x) - f(x)]_+ - C\|y^k - x\|^2$: tangente décalée de f en y^k.

est équivalent au programme (2.2) et a pour conditions nécessaires d'optimalité :

$$\begin{aligned} &\nabla_{(t,y)}(t + \frac{1}{2}(y-x)^\top (Q(x) + \tau_k I)(y-x)) \\ &\qquad + \sum_{(a,g)\in\mathcal{G}_k} \lambda_{(a,g)} \nabla_{(t,y)}(a + g^\top (y-x) - t) = 0, \\ &\lambda_{(a,g)} > 0 \quad \text{si} \quad a + g^\top (y-x) = t, \\ &\lambda_{(a,g)} = 0 \quad \text{sinon.} \end{aligned} \tag{2.8}$$

Ainsi y^k est solution de (2.2) si et seulement s'il existe $(a_1, g_1), \ldots, (a_r, g_r) \in \mathcal{G}_k$, $\lambda^k \in (\mathbb{R}_+^*)^r$ avec $\sum_{i=1}^r \lambda_i^k = 1$, tels que $a_i + g_i^\top (y^k - x) = \phi_k^{[1]}(y^k, x)$ pour $i = 1, \ldots, r$, et $(Q(x) + \tau_k I)(x - y^k) = \sum_{i=1}^r \lambda_i^k g_i$. On pose $g_k^* = \sum_{i=1}^r \lambda_i^k g_i$, $a_k^* = \sum_{i=1}^r \lambda_i^k a_i$ et $m_k^* = a_k^* + g_k^{*\top}(\cdot - x)$. On appelle g_k^* le sous-gradient agrégé et m_k^* le plan agrégé. Pour assurer la convergence de la méthode, il suffit que le faisceau $\mathcal{G}_{k+1}$ à l'itération $k+1$ contienne une paire exacte (a_0, g_0) avec $a_0 = f(x)$ et $g_0 \in \partial f(x)$ pour assurer l'exactitude, une paire (a_k, g_k) pour enrichir le faisceau et la paire agrégé (a_k^*, g_k^*) pour limiter le nombre de plans coupants dans le faisceau et ainsi limiter la complexité du modèle de travail.

Contrairement à une méthode de faisceau convexe où le paramètre de proximité τ peut être fixé une fois pour toute au début de l'algorithme, une méthode de faisceau non convexe utilise de manière dynamique ce paramètre tout au long de l'algorithme pour compenser la perte de convexité. Pour savoir si l'information extraite au pas nul y^k peut être utile à l'itération suivante $k+1$, on calcule le quotient

$$\widetilde{\rho}_k = \frac{f(x) - \Phi_{k+1}(y^k, x)}{f(x) - \Phi_k(y^k, x)} \tag{2.9}$$

qui évalue la distance entre les modèles de travail successifs en y^k. L'amélioration de la qualité du nouveau modèle de travail $\Phi_{k+1}(\cdot, x)$ est jugée insuffisante lorsque $\widetilde{\rho}_k \geq \widetilde{\gamma}$ où $\gamma < \widetilde{\gamma} < 1$ est fixée au début de l'algorithme. En effet comme Φ_{k+1} est supposé se rapprocher de f, la conjonction $(\rho_k < \gamma, \widetilde{\rho}_k \geq \widetilde{\gamma})$ nous dit que le progrès attendu est trop

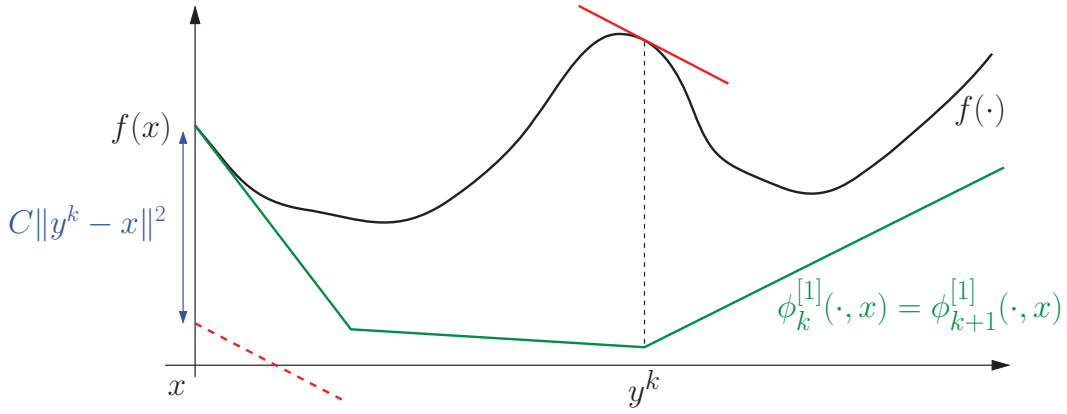


FIGURE 2.3.: Cas particulier où $\phi_{k+1}^{[1]}$ n'améliore pas $\phi_k^{[1]}$. Cause : non convexité de f.

marginal. C'est là que nous augmentons $\tau_{k+1} = 2\tau_k$ pour forcer les pas à se faire plus petits à la prochaine itération $k+1$. La situation opposée $\rho_k < \gamma$ et $\widetilde{\rho}_k < \widetilde{\gamma}$ est considérée comme encore ouverte. Gardant fixé $\tau_{k+1} = \tau_k$, nous comptons sur l'amélioration de Φ_{k+1} en ajoutant des plans sécants et le plan agrégé. En ce qui concerne la gestion du paramètre de proximité entre deux pas sérieux x et x^+, on procède comme suit. Dès qu'un pas sérieux x^+ est trouvé, nous avons besoin du paramètre τ pour la prochaine boucle interne. C'est fait via l'élément mémoire $\tau^\sharp$. Si $\rho_k \geq \Gamma$, où $0 < \gamma < \Gamma < 1$, alors nous diminuons le paramètre τ, parce que la concordance entre le modèle et la réalité est "bonne". Si $\gamma \leq \rho_k < \Gamma$, alors l'adéquation n'est "pas mauvaise", et nous gardons τ en tant que tel.

Dans le but de ne pas partir de "zéro" à chaque itération j de la boucle externe, on propose de recycler des plans sécants de l'itération j pour l'itération $j+1$. Dans les méthodes de faisceau convexes, le dernier faisceau $\mathcal{G}$ de la $j^{\text{ème}}$ boucle interne peut être intégré tel quel dans le premier faisceau $\mathcal{G}_1$ de la $j+1^{\text{ème}}$ boucle interne. La seule raison qu'a une méthode de faisceau convexe de ne pas garder tous les plans sécants accumulés est la capacité limitée de stockage des ordinateurs. Encore une fois, dans le cas non convexe, ce n'est plus possible puisque le plan $m(\cdot, x) = a + g^\top(\cdot - x)$ tel que $(a, g) \in \mathcal{G}$ est inutile en x^+, si $m(x^+, x) \geq f(x^+)$. Nous proposons donc de recycler le vieux plan $m(\cdot, x)$ en le nouveau plan $m(\cdot, x^+)$, où

$$m(\cdot, x^+) = m(\cdot, x) - s^+,$$

avec s^+ le décalage vers le bas en x^+. Soit

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

Typiquement on conserve tout $(a, g) \in \mathcal{G}$ associé à un multiplicateur de Lagrange strictement positif (2.7)-(2.8).

En conclusion on dira que les techniques de faisceau peuvent être vues comme un filtre sur le sous-différentiel qui extrait quelques sous-gradients sélectionnés avec soin.

Algorithm 1. Algorithme de faisceaux non convexe avec tangentes décalées.

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

1: **Initialiser la boucle externe**. Choisir un point initial x^1 et une matrice initiale $Q_1 = Q_1^\top$ avec $-qI \preceq Q_1 \preceq qI$. Initialiser le paramètre mémoire de contrôle $\tau_1^\sharp$ tel que $Q_1 + \tau_1^\sharp I \succ 0$. Poser $j = 1$.

2: **Test d'arrêt**. A l'itération j de la boucle externe et à l'itéré sérieux x^j, stopper si $0 \in \partial f(x^j)$. Sinon aller dans la boucle interne.

3: **Initialiser la boucle interne**. Poser le compteur de la boucle interne $k = 1$. Initialiser le paramètre de contrôle $\tau_1 = \tau_j^\sharp$. Construire le premier modèle de travail $\Phi_1(\cdot, x^j)$ en utilisant l'ensemble initial $\mathcal{G}_1$ et la matrice Q_j.

4: **Génération d'un pas d'essai.** A l'itération k de la boucle interne, résoudre le programme tangent

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

La solution est le nouveau pas d'essai y^k.

5: **Test d'acceptation du pas d'essai.** Vérifier si

$$\rho_k = \frac{f(x^j) - f(y^k)}{f(x^j) - \Phi_k(y^k, x^j)} \geq \gamma.$$

Si c'est le cas, poser $x^{j+1} = y^k$ (pas sérieux), quitter la boucle interne et aller à l'étape 8. Si ce n'est pas le cas (pas nul) continuer dans la boucle interne avec l'étape 6.

6: **Mise à jour du modèle de travail**. Générer un plan coupant $m_k(\cdot, x^j) = a_k + g_k^\top(\cdot - x^j)$ au pas nul y^k et à l'itération k en utilisant le décalage vers le bas. Calculer le plan agrégé $m_k^*(\cdot, x^j) = a_k^* + g_k^{*\top}(\cdot - x^j)$ en y^k. Construire $\mathcal{G}_{k+1} = \mathcal{G}_k \cup \{(a_k, g_k), (a_k^*, g_k^*)\}$. Dans le but de garder raisonnable la taille de $\mathcal{G}_{k+1}$ permettre de retirer quelques éléments de $\mathcal{G}_k$ appelé par le plan agrégé. Construire le nouveau modèle de travail $\Phi_{k+1}(\cdot, x^j)$ en utilisant l'ensemble $\mathcal{G}_{k+1}$ et la matrice Q_j.

7: **Mise à jour du paramètre de contrôle de proximité.** Vérifier si

$$\widetilde{\rho}_k = \frac{f(x^j) - \Phi_{k+1}(y^k, x^j)}{f(x^j) - \Phi_k(y^k, x^j)}.$$

$$\text{Poser} \quad \tau_{k+1} = \begin{cases} \tau_k, & \text{si } \widetilde{\rho}_k < \widetilde{\gamma} \\ 2\tau_k, & \text{si } \widetilde{\rho}_k \geq \widetilde{\gamma} \end{cases}$$

Incrémenter le compteur k de la boucle interne et continuer dans la boucle interne avec le step 4.

8: **Mise à jour de Q_j et de l'élément mémoire**. Mettre à jour la matrice $Q_j \to Q_{j+1}$ en respectant $Q_{j+1} = Q_{j+1}^\top$ et $-qI \preceq Q_{j+1} \preceq qI$. Ensuite stocker le nouvel élément mémoire

$$\tau_{j+1}^\sharp = \begin{cases} \tau_{k+1}, & \text{si } \gamma \leq \rho_k < \Gamma \qquad \text{(pas mauvais)} \\ \\ \frac{1}{2}\tau_{k+1}, & \text{si } \rho_k \geq \Gamma \qquad \text{(bon)} \end{cases}$$

Augmenter $\tau_{j+1}^\sharp$ si nécessaire pour assurer $Q_{j+1} + \tau_{j+1}^\sharp I \succ 0$. Si $\tau_{j+1}^\sharp > T$ alors poser $\tau_{j+1}^\sharp = T$. Incrémenter le compteur j de la boucle externe et retourner à l'étape 2.

La mise en place de l'algorithme 1 fait suite aux travaux [18] qui ont mené à l'algorithme 2. La différence entre les algorithmes 1 et 2 est que les plans sécants de l'algorithme 1 sont des tangentes décalées de f tandis que les plans sécants de l'algorithme 2 sont des tangentes d'un modèle idéal ϕ de f.

Definition 3. *[18] Une fonction $\phi : \mathbb{R}^n \times \Omega \to \mathbb{R}$ est appelée modèle idéal d'ordre 1 de f sur l'ensemble $\Omega \subset \mathbb{R}^n$ si pour tout $x \in \Omega$ la fonction $\phi(\cdot, x) : \mathbb{R}^n \to \mathbb{R}$ est convexe et les conditions suivantes sont satisfaites :*

(i) $\phi(x, x) = f(x)$ *et* $\partial_1\phi(x, x) \subset \partial f(x)$.

(ii) Pour tout $x \in \Omega$ *et tout* $\varepsilon > 0$ *il existe* $\delta > 0$ *tel que* $f(y) - \phi(y, x) \leq \varepsilon \|y - x\|$ *pour tout* $y \in \mathbb{R}^n$ *avec* $\|y - x\| \leq \delta$.

(iii) ϕ *est semi continue supérieurement sur* $\mathbb{R}^n \times \Omega$, *i.e.* $(y_j, x_j) \to (y, x)$ *dans* $\mathbb{R}^n \times \Omega$ *implique* $\limsup_{j\to\infty} \phi(y_j, x_j) \leq \phi(y, x)$.

Si $\Omega = \mathbb{R}^n$, *on dit simplement que* ϕ *est un modèle d'ordre 1 de* f.

Algorithm 2 Algorithme de faisceaux non convexe avec modèle idéal.

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

1: **Initialiser la boucle externe**. Choisir un point initial x^1 et une matrice initiale $Q_1 = Q_1^\top$ avec $-qI \preceq Q_1 \preceq qI$. Initialiser le paramètre mémoire de contrôle $\tau_1^\sharp$ tel que $Q_1 + \tau_1^\sharp I \succ 0$. Poser $j = 1$.

2: **Test d'arrêt**. A l'itération j de la boucle externe et à l'itéré sérieux x^j, stopper si $0 \in \partial f(x^j)$. Sinon aller dans la boucle interne.

3: **Initialiser la boucle interne**. Poser le compteur de la boucle interne $k = 1$. Initialiser le paramètre de contrôle $\tau_1 = \tau_j^\sharp$. Construire le premier modèle de travail convexe $\phi_1(\cdot, x^j)$ et poser $\Phi_1(y, x^j) = \phi_1(y, x^j) + \frac{1}{2}(y - x^j)Q_j(y - x^j)$.

4: **Génération d'un pas d'essai.** A l'itération k de la boucle interne, résoudre le programme tangent

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

La solution est le nouveau pas d'essai y^{k+1}.

5: **Test d'acceptation.** Vérifier si

$$\rho_k = \frac{f(x^j) - f(y^{k+1})}{f(x^j) - \Phi_k(y^{k+1}, x^j)} \geq \gamma.$$

Si c'est le cas, poser $x^{j+1} = y^{k+1}$ (pas sérieux), quitter la boucle interne et aller à l'étape 8. Si ce n'est pas le cas (pas nul) continuer dans la boucle interne avec l'étape 6.

6: **Mise à jour du paramètre de proximité.** Calculer le second paramètre de contrôle

$$\widetilde{\rho}_k = \frac{f(x^j) - \Phi(y^{k+1}, x^j)}{f(x^j) - \Phi_k(y^{k+1}, x^j)}.$$

$$\text{Poser} \quad \tau_{k+1} = \begin{cases} \tau_k, & \text{si } \widetilde{\rho}_k < \widetilde{\gamma} \\ 2\tau_k, & \text{si } \widetilde{\rho}_k \geq \widetilde{\gamma} \end{cases}$$

7: **Mise à jour du modèle de travail**. Construire un nouveau modèle de travail convexe $\phi_{k+1}(., x^j)$ en respectant les trois règles (exactitude, plan coupant, agrégation) basé sur le pas nul y^{k+1}. Ensuite augmenter le compteur k de la boucle interne et continuer la boucle interne avec l'étape 4.

8: **Mise à jour de Q_j et de l'élément mémoire**. Mettre à jour la matrice $Q_j \to Q_{j+1}$ en respectant $Q_{j+1} = Q_{j+1}^\top$ et $-qI \preceq Q_{j+1} \preceq qI$. Ensuite stocker le nouvel élément mémoire

$$\tau_{j+1}^\sharp = \begin{cases} \tau_{k+1}, & \text{si } \gamma \leq \rho_k < \Gamma \qquad \text{(pas mauvais)} \\ \\ \frac{1}{2}\tau_{k+1}, & \text{si } \rho_k \geq \Gamma \qquad \text{(bon)} \end{cases}$$

Augmenter $\tau_{j+1}^\sharp$ si nécessaire pour assurer $Q_{j+1} + \tau_{j+1}^\sharp I \succ 0$. Incrémenter le compteur j de la boucle externe et retourner à l'étape 2.

2.3. Application en synthèse H_∞

Ces algorithmes, bien que généraux, ont été conçu dans l'esprit de résoudre le problème de synthèse H_∞ *structuré* issu de l'automatique. Celui-ci propose de minimiser la norme H_∞ d'une transformée linéaire fractionnaire $F_l(P,K)$ [3] sur l'ensemble des contrôleurs K satisfaisant une contrainte de structure. Les systèmes P et K sont supposés linéaires à temps invariants :

$$P(s) = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \text{ avec } P_{ij}(s) = D_{ij} + C_i(sI - A)^{-1}B_j,$$

et

$$K(s) = A_K + B_K(sI - A_K)^{-1}B_K.$$

La matrice de transfert $P(s)$ a été partitionnée en quatre blocs associés aux quatre transferts $w \to z$, $w \to y$, $u \to z$, $u \to y$ (voir Figure 1.2). Pour décrire les propriétés variationnelles de $F_l(P,K)$, il est plus commode de se placer dans le cas d'un correcteur statique, i.e. $K = D_K$, ce qui n'est aucunement restrictif puisque via un changement de variable approprié on peut toujours se ramener à l'action d'un correcteur statique (cf. annexe D). Dans ce cadre, on montre que la fonction de transfert $F_l(P,K)$ s'écrit

$$F_l(P,K)(s) = D(K) + C(K)(sI - A(K))^{-1}B(K),$$

avec $A(K)$, $B(K)$, $C(K)$, $D(K)$ rationnelles en K et définies sur l'ensemble des K tel que $(I - D_{22}K)$ est inversible (cf. annexe D). De plus on montre que pour tout $s \in \mathbb{C}$, $F_l(P,\cdot)(s)$ est définie et différentiable sur

$$\mathcal{D}_{F_l(P,\cdot)(s)} = \{K \text{ tel que } \det(I - D_{22}K) \neq 0 \text{ et } s \notin \mathrm{Sp}(A(K))\}.$$

Parmi les spécifications de performance de la loi de commande K, celle qui revient toujours est la stabilité interne de la boucle fermée, i.e. le spectre $\mathrm{Sp}(A(K))$ de la matrice dynamique en boucle fermée est inclus dans $\mathbb{C}_-^*$. Lorsque $\mathrm{Sp}(A(K)) \subset \mathbb{C}_-^*$ on dit que K stabilise $F_l(P,K)$ ou que $F_l(P,K)$ est stable et l'on a

$$\sup_{\mathrm{Re}\, s>0} \sigma_1(F_l(P,K)(s)) = \sup_{\omega\in\mathbb{R}} \sigma_1(F_l(P,K)(j\omega)) < \infty$$

d'après le principe du maximum pour les fonctions analytiques. Pour un système LTI G, la quantité $\sup\{\sigma_1(G(j\omega)) : \omega \in \mathbb{R}\}$ est appelée norme L_∞ de G.

Dans les exemples traités ci-dessous, on choisit de considérer le carré de la norme H_∞, ce qui donne un problème d'optimisation strictement équivalent. Cette fonction est de la forme

$$f(x) = \max_{\omega\in\mathbb{R}} \lambda_1(F(x,\omega)), \tag{2.10}$$

où $F : \mathbb{R}^n \times \mathbb{R} \to \mathbb{H}^m$ et où λ_1 fait référence à l'opérateur plus grande valeur propre sur l'espace vectoriel complexe des matrices hermitiennes $\mathbb{H}^m$. Plus précisément

$$F(x,\omega) = F_l(P,K(x))(j\omega)F_l(P,K(x))(j\omega)^H.$$

On ne s'intéresse qu'aux paramétrisations du contrôleur $x \to K(x)$ définies et différentiables sur $\mathbb{R}^n$. Ainsi f est définie et pour tout $\omega \in \mathbb{R}$, $F(.,\omega)$ est différentiable sur

$$\mathcal{D}_f = \{x \in \mathbb{R}^n \text{ tel que le système } F_l(P, K(x)) \text{ est stable}\}\,. \tag{2.11}$$

On note $F'(x,\omega)$ la différentielle de $F(\cdot,\omega)$ en x et $F'(x,\omega)^*$ l'adjoint de $F'(x,\omega)$. On dira qu'une fréquence ω est active en x lorsque $f(x) = \lambda_1(F(x,\omega))$. Le calcul des fréquences actives en un point x donné est aussi important que celui de $f(x)$ puisque celles-ci permettent de caractériser le sous-différentiel $\partial f(x)$ de f en x (cf. annexe B). Le calcul de $f(x)$ et de ses fréquences actives est effectué par un algorithme de bissection utilisant un calcul hamiltonien [19, 20]. La convergence de l'algorithme [19] est quadratique, ce qui le rend efficace.

Le modèle idéal associé à la fonction (2.10) est

$$\phi(y,x) = \max_{\omega \in \mathbb{R}_+} \lambda_1(F(x,\omega) + F'(x,\omega)(y-x)). \tag{2.12}$$

On voit ici que la notion de modèle idéal est particulièrement adaptée pour traiter le problème de synthèse H_∞. En effet le modèle idéal vient naturellement en exploitant la structure très spéciale de f qui compose une fonction convexe avec une fonction différentiable. On laisse le lecteur se reporter à l'annexe B pour une description détaillée des propriétés variationnelles de $\lambda_1 : \mathbb{H}^m \to \mathbb{R}$. Cependant (2.12) a le défaut d'être gourmand en terme de coût de calcul [21, p. 16-19]. Dans [21] on explique que le calcul de $\phi(y,x)$ est 27 fois plus coûteux que celui de $f(y)$. Son utilisation est alors limitée par la taille de $F(x,\omega)$. Pour remédier à ce problème numérique, [21] propose un calcul d'une version simplifiée de ϕ.

2.4. Modèle idéal versus tangentes décalées

Lors des premiers tests avec l'algorithme 2 appliqué au problème du chapitre 3, on a été confronté à des problèmes numériques. On a découvert plus tard que la grande majorité des problèmes avait pour origine le résolution de la forme duale du programme quadratique tangent (QP). La routine utilisée pour résoudre le QP est la routine quadprog de matlab. La forme duale du QP fait intervenir une multiplication de matrices de la forme $A^\top A$ où les colonnes de A sont les sous-gradients des plans sécants du modèle de travail. Ce produit cause des problèmes de conditionnement et sa résolution numérique rencontrait des échecs. La résolution de la forme primale du QP s'est montrée beaucoup plus robuste. Lorsque l'on trace le graphe du modèle idéal dans sa version simplifiée, celui-ci n'est généralement pas convexe. De plus la mise en application du modèle idéal avait nécessité beaucoup de travail pour son calcul numérique [21]. D'un point de vue théorique, la construction du modèle idéal peut s'avérer plus délicate pour des fonctions autres que la norme H_∞. Comme on l'a fait remarquer précédemment, la norme H_∞ présente une structure propice à la mise en place d'un modèle idéal. La technique avec tangentes décalées enlève les problèmes théoriques et numériques de la construction et

Algorithme	$f(x^+)$	$\frac{\|x^+-x\|}{1+\|x\|}$	$\frac{\|f(x^+)-f(x)\|}{1+\|f(x)\|}$	Par. de proximité τ	Itér.
Tangentes décalées	1.82	2.66×10^{-5}	2.91×10^{-5}	1.00	1000
Modèle idéal	1.09	4.76×10^{-4}	6.71×10^{-5}	3.96×10^{-5}	1000

TABLE 2.1.: Résolution du problème (2.13).

de l'implémentation du modèle idéal. C'est pourquoi on s'est tourné vers cette technique alternative. Pour finir on ajoutera que, le fait que le graphe du modèle idéal ne soit pas convexe, n'a pas d'impact sur l'algorithme 2 si la tangente du modèle idéal est correcte et se place bien, puisque c'est bien elle qui vient enrichir le modèle de travail, qui à son tour génère un pas d'essai.

Dans cette section, ce sont les deux versions actuelles des algorithmes 1 et 2 qui sont confrontées sur le problème de la section 1.2 et sur le problème du chapitre 3.

2.4.1. Problème 1

Le problème de commande structurée présenté dans la section 1.2 s'énonce ainsi :

$$\begin{array}{c}\text{Trouver un correcteur } K \textit{ d'ordre 1} \text{ minimisant} \\ \left\|S_d^{-1}S(K)\right\|_\infty = \max\limits_{\text{Re } s>0} \sigma_1[S_d^{-1}(s)S(K)(s)].\end{array} \tag{2.13}$$

où $S(K)(s) = (1+G(s)K(s))^{-1}$, $S_d(s) = (s+0.1)(s+10)^{-1}$ et $G(s) = (s^2+0.01s+1)^{-1}$. On rappelle que $\left\|S_d^{-1}S(K)\right\|_\infty \leq 1$ signifie que $S(K)$ satisfait le gabarit S_d et donc que la synthèse a réussie. Observons que pour tout K, $\left\|S_d^{-1}S(K)\right\|_\infty \geq 1$. Par conséquent, le mieux que l'on puisse faire est de trouver K tel que $\left\|S_d^{-1}S(K)\right\|_\infty = 1$. Ici $F_l(P,K) = S_d^{-1}S(K)$ et la paramétrisation du contrôleur est $x = (x_1,...,x_4) \in \mathbb{R}^4 \to K(x) = x_4 + x_2(s-x_1)^{-1}x_3$. Le test d'arrêt utilisé est un test de "slow progress" défini comme suit

$$\text{Stopper si} \quad \frac{|x^+ - x|}{1+|x|} \leq 1 \times 10^{-6} \quad \text{et} \quad \frac{|f(x^+) - f(x)|}{1+|f(x)|} \leq 1 \times 10^{-6}.$$

Le nombre maximum d'itérations autorisées pour la boucle externe est 1000, et pour la boucle interne est 60. Pour cette première optimisation, on précise que les algorithmes 1 et 2 sont lancés sans recyclage de plans sécants. On présente Table 2.1 les résultats. On constate que l'utilisation du modèle idéal pour alimenter en plans sécants le modèle de travail donne de meilleurs résultats que l'utilisation de tangentes décalées de l'objectif. Analysons en détails le comportement des deux algorithmes. On présente Figure 2.4 en fonction des 100 premières itérations de la boucle externe, l'évolution de l'objectif, l'évolution du paramètre de contrôle de proximité et l'évolution de la taille des pas.

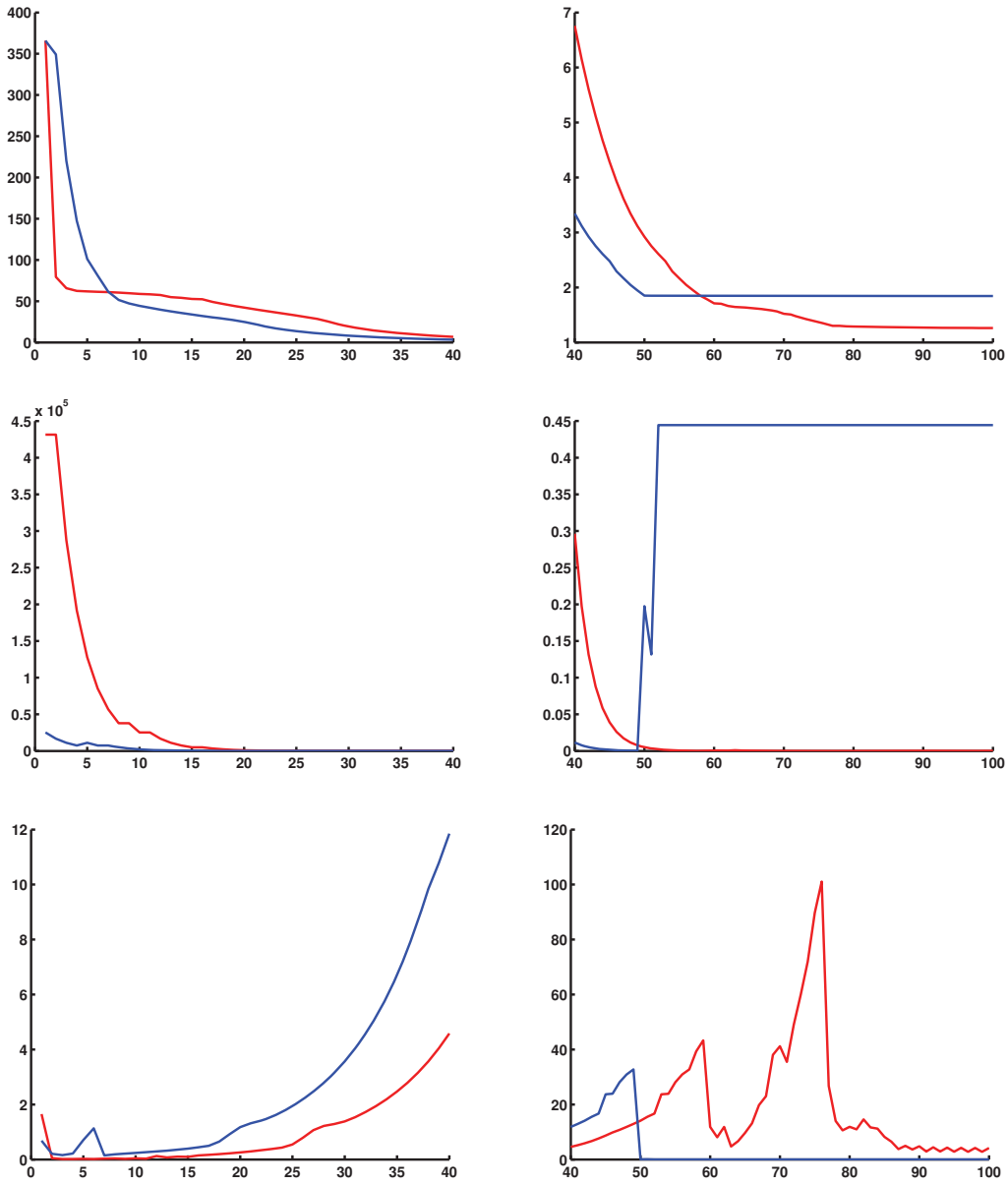


FIGURE 2.4.: Algorithme avec modèle idéal (en rouge) ou tangentes décalées (en bleu) sur le problème (2.13). A droite : itérations de 1 à 40, à gauche : itérations de 40 à 100. De haut en bas en fonction j : $f(x^j)$, $\tau_j^{\#}$, $|x^j - x^{j+1}|$.

L'algorithme 2 augmente beaucoup plus le paramètre τ dans la première itération $j = 1$, on remarque que cela lui fait faire un pas plus performant en regard de la décroissance de l'objectif, mais surtout il ne se bloque pas en fin d'optimisation. On se demande pourquoi.

Rappelons que le modèle idéal $\phi(., x)$ à l'itéré sérieux courant x est une approximation convexe local de f en x auquel on applique une méthode de faisceaux convexe pour créer et améliorer le modèle de travail $\phi_k(., x)$. Le paramètre de proximité τ_k permet de pallier au défaut de convexité de f et permet de prendre en compte le caractère local du modèle idéal ϕ. D'après la gestion du paramètre de proximité entre deux itérés sérieux, i.e. il est diminué si le pas sérieux est un "bon" pas, on peut dire que si τ tend vers 0, comme c'est le cas dans cette étude, alors ϕ est un bon modèle pour f, et tend à se rapprocher de f. Cela n'est pas forcément surprenant. En effet, on imagine assez facilement qu'en tout minimum local x^* d'une fonction continue, il existe un voisinage ouvert de x^* dans lequel f est convexe. Ainsi dans ce voisinage, l'algorithme avec modèle idéal revient à appliquer une méthode de faisceaux convexe à f. Rappelons que les tangentes décalées tiennent compte de la proximité du pas d'essai y^k avec l'itéré sérieux x et se décalent vers le bas avec le carré de la distance de y^k à x. Ainsi le décalage tend vers 0 quand y^k tend vers x. Or lorsque l'on regarde Figure 2.4 la taille des pas effectués par l'algorithme avec modèle idéal, on se rend compte que celui-ci fait de grands pas entre les itérations 50 et 80. Ce qui porte à croire que pour progresser de manière raisonnable vers l'optimum, il faut parcourir de "longues" distances et faire de "grands" pas. L'algorithme avec tangentes décalées ne progresse plus à partir de l'itération 50 et fait de tous petits pas. Pour finir le raisonnement, on pense que, comme c'est le cas en général, la fonction est non différentiable en un minimum local x^*. L'information générée aux pas nuls est donc essentielle pour progresser vers un point critique. Le décalage occasionné par de "grands pas" fait perdre, dans cette étude, trop d'information à proximité de x. En conséquence l'algorithme avec tangentes décalées est condamné à faire de petits pas.

Afin de valider ce raisonnement, on a relancé l'algorithme 1 sans décaler les tangentes de f et avec pour point initial l'itéré sérieux obtenu à la $100^{\text{ème}}$ itération de l'optimisation précédente. On présente les résultats en bleu Figure 2.5. L'optimisation s'est débloquée ($f(x^0) = 1.84$ et $f(x^{100}) = 1.06$), le paramètre τ tend vers 0 et les pas sont beaucoup plus grands. Ils sont même plus grands que les pas effectués par le modèle idéal. On présente en vert sur les mêmes figures, les effets du recyclage sur les performances de l'algorithme 1. La différence fondamentale apporté par le recyclage est le nombre d'itérations dans la boucle interne. Celui-ci est nettement inférieur. Ce qui prouve bien que le recyclage permet de transporter de l'information utile d'une itération de la boucle externe à la suivante et de ne pas repartir de "zéro". On précise que les plans recyclés, de la même manière que les tangentes de l'objectif aux pas nuls, n'ont pas été décalés. On rappelle que la fonction f est une fonction max

$$f(x) = \max_{\omega \in \mathbb{R}_+} f_\omega(x)$$

où $f_\omega(x) = \lambda_1(F(x, \omega))$. Elle n'est pas différentiable en un point x de son domaine de définition $\mathcal{D}_f$ si l'une des deux conditions suivantes est vérifiée (cf. annexe B) :

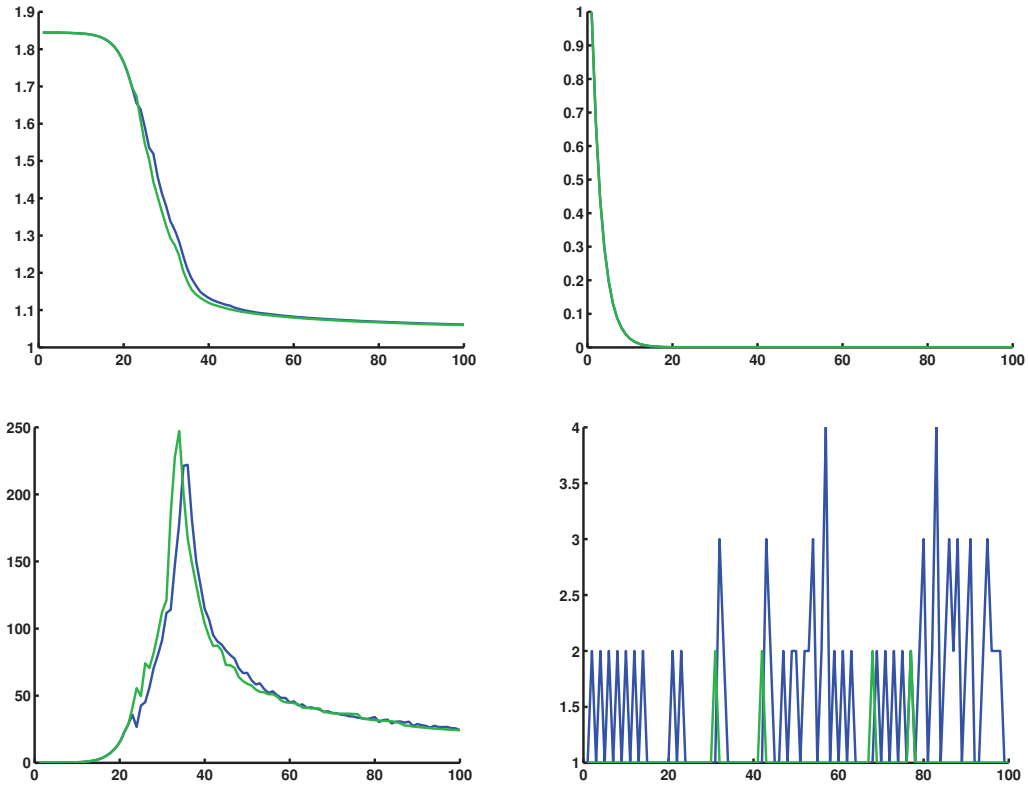


FIGURE 2.5.: En bleu : algorithme avec tangentes non décalées sans recyclage, en vert : algorithme avec tangentes non décalées et recyclage. En haut de gauche à droite en fonction de j : $f(x^j)$ et $\tau_j^{\#}$. En bas de gauche à droite en fonction de j : $|x^j - x^{j+1}|$ et nombre d'itér. dans la boucle interne.

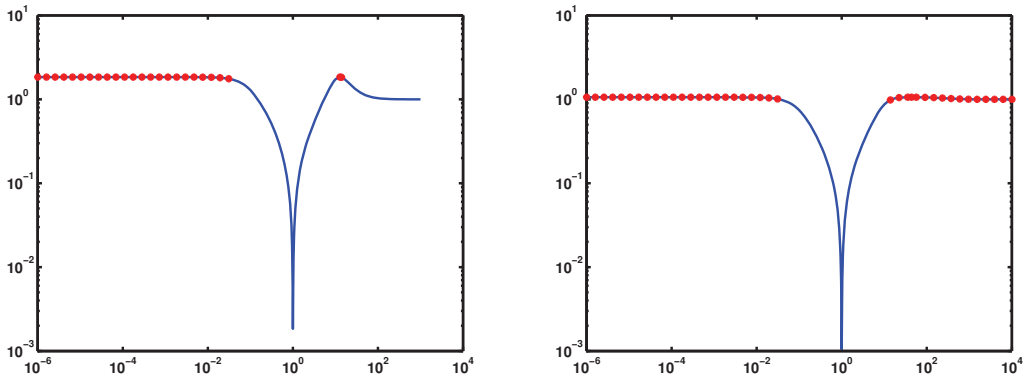


FIGURE 2.6.: $\omega \to f_\omega(x^j)$. De gauche à droite, $j = 1$ et $j = 100$. Les échelles sont logarithmiques. Les marqueurs rouges correspondent à des fréquences ω telles que $f_\omega(x) \geq 0.9 \times f(x)$.

– il existe plusieurs fréquences actives en x,
– il existe une fréquence ω active en x avec $\lambda_1(F(x,\omega))$ multiple.

Dans ce problème $F(x,\omega) \in \mathbb{C}$, donc $f_\omega(\cdot) = F(\cdot,\omega)$ est différentiable sur $\mathcal{D}_f$. Les problèmes de régularité de f ne peuvent provenir que de l'opérateur $\max$ sur ω. On présente Figure 2.6 le graphes des fonctions $\omega \to f_\omega(x^j)$ pour $j = 1$ et $j = 100$. Les marqueurs rouges correspondent à des fréquences ω, dîtes *quasi* actives, telles que $f_\omega(x) \geq 0.9 \times f(x)$. Clairement f est non lisse à la fois en x^1 et en x^{100}. De plus à l'itéré sérieux x^{100}, $f_\omega(x) \simeq 1$ sauf sur un plage de fréquences autour de $1rad.s^{-1}$. Ce qui signifie que la réponse fréquentielle $S(K)$ colle au gabarit S_d sauf au voisinage de $1rad.s^{-1}$ où elle présente une "crevasse" bénéfique qui tient compte des modes mal amortis $\xi = 0.005$ du système à commander G de fréquence propre $1rad.s^{-1}$ (cf. chapitre 1 et Figure 1.6).

Bien sûr le décalage est nécessaire dans le cas général comme le prouve l'échec de l'algorithme avec tangentes non décalées lancé avec le point initial de la première optimisation. L'algorithme atteint le nombre maximum d'itérations autorisées dans la première boucle interne ($j = 1$, $k = k_{\max}$, ici $k_{\max} = 60$) sans avoir trouvé de pas sérieux x^2 succédant au point initial x^1. Le décalage peut devenir handicapant dans une région convexe où l'on a besoin de faire de la distance pour progresser vers un point critique. Mais il est nécessaire lorsque l'on se meut dans une région non convexe. On présente Figure 2.7 le tracé de la fonction $\omega \to f_\omega(y^k)$ où y^k est le pas nul obtenu à $j = 1$ et $k = 22$. En pratique on ne construit pas un seul plan sécant au pas nul y^k, mais un ensemble de plans sécants associé à un ensemble de fréquences quasi actives en y^k. Ici on utilise la fréquence pic ω_1 et une fréquence quasi active ω_2. "La tangente non décalée" en y^k est alors un modèle polyédral de la forme

$$\widehat{m}_k(\cdot) = \max\{f_{\omega_i}(y^k) + g^k_{\omega_i}(\cdot - y^k),\ i = 1,2\},$$

avec $g^k_{\omega_i} = F'(y^k,\omega_i)$. Dans le cas général, i.e. $F(x,\omega) \notin \mathbb{C}$, f_ω est différentiable en $x \in \mathcal{D}_f$ si et seulement si $\lambda_1(F(x,\omega))$ est simple. Son gradient est alors $F'(x,\omega)^*(uu^H)$ avec u un vecteur propre unitaire de $F(x,\omega)$ associé à $\lambda_1(F(x,\omega))$. Lorsque $\lambda_1(F(x,\omega))$ est

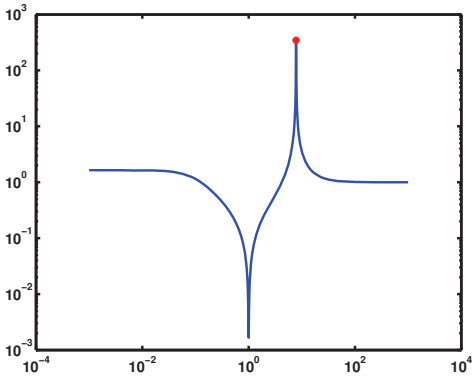


FIGURE 2.7.: $\omega \to f_\omega(y^k)$, $j = 1$, $k = 22$, en échelles logarithmiques.

multiple, on parle de sous-gradient g_ω de f_ω en x. Un tel g_ω est de la forme

$$F'(x,\omega)^* \left(Q_\omega Y_\omega Q_\omega^H\right) \in \partial f_\omega(x),$$

où les colonnes de Q_ω forment une base orthonormée du sous-espace propre de $F(x,\omega)$ associé à sa plus grande valeur propre, et Y_ω est hermitienne, positive et de trace 1 (cf. annexe B). Remarquons que pour

$$Y_\omega = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix},$$

on a $g_\omega = F'(x,\omega)^*(uu^H)$ avec u un vecteur propre unitaire de $F(x,\omega)$ associé à sa plus grande valeur propre. Ainsi $F'(x,\omega)^*(uu^H)$ convient toujours, que f_ω soit différentiable en x ou pas ! Dans un cas c'est le gradient, dans l'autre c'est un sous-gradient (parmi une infinité d'autres). On présente Figure 2.8 l'allure de la fonction à minimiser f, du modèle de travail $\phi_k^{[1]}$, du plan agrégé m_k^* et de la "tangente non décalée" $\widehat{m}_k$ dans la direction $y^k - x^1$. Les carrés rouges correspondent à des points $y \in \mathbb{R}^n$ n'appartenant pas à $\mathcal{D}_f$. On remarque que l'ajout de $\widehat{m}_k$ dans le nouveau modèle de travail $\phi_{k+1}^{[1]}$ "brise" la propriété d'exactitude puisque $\widehat{m}_k(x^1) > f(x^1)$! Sur la Figure 2.9 sont représentées les branches f_{ω_1} et f_{ω_2} au voisinage de y^k dans la direction $y^k - x^1$. Le besoin de décaler la tangente de la branche f_{ω_2} en y^k est dû à la concavité de celle-ci en y^k.

Abscisse spectrale

Le domaine de définition de f équation (2.11) n'est pas une contrainte classique d'un problème d'optimisation dans le sens où la contrainte est un ensemble ouvert. L'algorithme non lisse peut alors générer une suite $\{x^j\}$ convergeant vers un point non admissible x^* appartenant au bord $\partial\mathcal{D}_f$ du domaine de définition. De plus lorsque le programme quadratique tangent (2.2) fournit un pas d'essai $y^k \notin \mathcal{D}_f$, que fait-on ? Lors

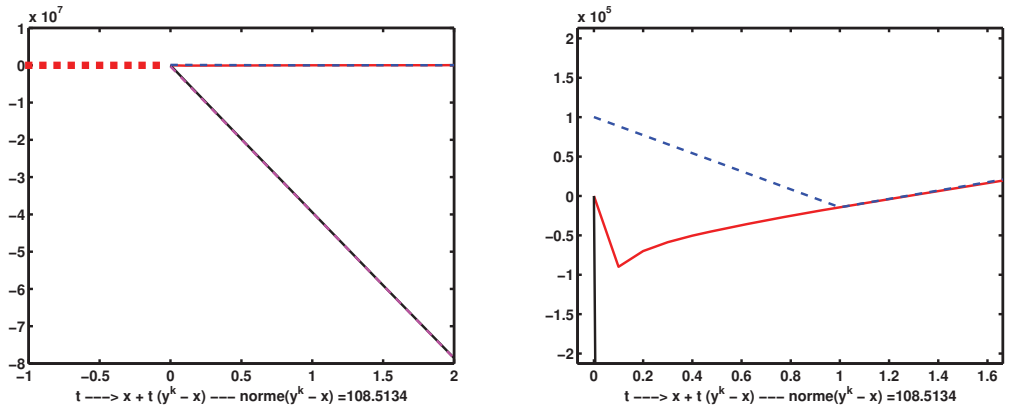


FIGURE 2.8.: La figure de droite est un zoom de la figure de gauche. Tracé de $t \to g(x^1 + t(y^k - x^1))$, où $g = f$ en rouge, $g = \phi_k(., x^1)$ en noir, $g = m_k^*(\cdot, x^1)$ en pointillé mauve, et $g = \widehat{m}_k(\cdot)$ en pointillé bleu.

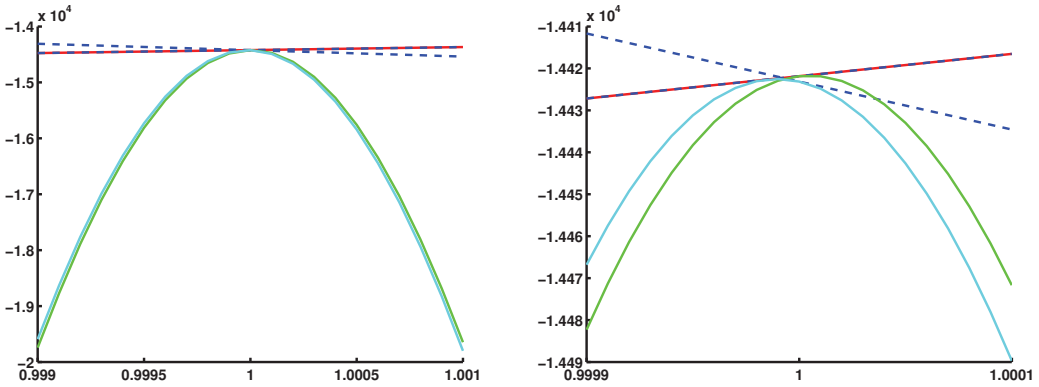


FIGURE 2.9.: La figure de droite est un zoom de la figure de gauche. Tracé de $t \to g(x^1 + t(y^k - x^1))$, où $g = f$ en rouge, $g = \widehat{m_k}(\cdot)$ en pointillé bleu, $g = f_{\omega_1}$ en vert et $g = f_{\omega_2}$ en cyan.

des applications en synthèse H_∞ [22, 23] de l'algorithme 2, la technique consistait à rejeter le pas y^k, à augmenter le paramètre de proximité τ et à relancer le programme quadratique tangent. Par continuité de f en $x^j \in \mathcal{D}_f$, il existe un voisinage ouvert de x^j inclus dans $\mathcal{D}_f$. L'idée est donc de diminuer la taille du pas jusqu'à obtention d'un pas stabilisant. Cependant aucune information sur les directions "déstabilisantes" n'est fournit au modèle de travail. Sur les exemples (2.13) et (2.14), cette technique fonctionne, le problème est résolu. Nous allons tout de même montrer de manière heuristique que l'ajout d'une information directionnelle dans le modèle de travail lors d'un pas déstabilisant permet d'améliorer la qualité de la trajectoire des itérés sérieux.

La stabilité de la boucle fermée peut se formuler à l'aide de la fonction abscisse spectrale

$$\begin{aligned} \alpha : E &\to \mathbb{R} \\ M &\mapsto \text{Re}\,(\lambda_1(M)) \end{aligned}$$

sur l'espace vectoriel réel E des matrices à coefficients réels de même dimension que la matrice dynamique en boucle fermée $A(x)$. Dire que $F_l(P, K(x))$ est stable revient à dire que $\alpha(A(x)) < 0$. Les propriétés variationnelles de l'opérateur plus grande valeur propre sur un ensemble de matrices non nécessairement hermitiennes sont autrement plus compliquées à appréhender et demande un bagage théorique plus poussé que celui nécessaire à l'étude de $\lambda_1 : \mathbb{H}^m \to \mathbb{R}$. La fonction α est continue mais non convexe et surtout non lipschitzienne en général. Plus précisément α est différentiable en M si et seulement si $\lambda_1(M)$ est simple et localement lipschitzienne en M si et seulement si $\lambda_1(M)$ est semi-simple, i.e. sa multiplicité géométrique est égale à sa multiplicité algébrique [24,25]. Le point positif est que α est presque partout différentiable et que la coalescence des valeurs propres a typiquement lieu en des minima locaux. Or nous ne cherchons pas à minimiser l'abscisse spectrale, nous cherchons seulement à la garder strictement négative.

En pratique, les cas où l'on a à la fois $y^k \notin \mathcal{D}_f$ et α non différentiable en y^k sont rares. Une première idée a été d'utiliser α en tant que fonction contrainte dans la fonction de progrès $F(\cdot, x)$ définie et commentée section 3.3.1. Mais l'abscisse spectrale et la fonction de progrès se sont révélées incompatibles pour la raison suivante. En adéquation avec le formalisme de la section 3.3.1, $c(x) = \alpha(A(x)) - \varepsilon$ avec $|\varepsilon| \ll 1$ et $\varepsilon < 0$. De part sa construction, $F(x, x) = 0$ pour tout x. Si l'itéré sérieux courant x est admissible, i.e. $c(x) < 0$, alors $F(y, x) = \max\{f(y) - f(x), c(y)\}$. Or, ce que l'on a observé sur les exemples traités est que, en règle général, lorsque y^k est admissible, $c(y^k)$ est très petit en valeur absolue et $\nabla c(y^k)$ est très petit en norme. La contrainte c est constamment active aux pas nuls et trop proche de la valeur $F(x, x)$ de la fonction $F(\cdot, x)$ à minimiser. La tangente décalée de l'abscisse spectrale en y^k vient "aplanir" le modèle de travail de manière critique ce qui a pour conséquence la génération de pas excessivement proches les uns des autres. La deuxième idée est heuristique et utilise une tangente décalée de l'abscisse spectrale uniquement aux pas non admissibles. Cette technique étant heuristique, elle n'a aucune raison de fonctionner à "tous les coups". On montre dans l'annexe A que le gradient de $\alpha \circ A$ en y^k est égale à

$$\nabla(\alpha \circ A)(y^k) = A'(y^k)^* \left(|u_1^H v_1|^{-2} \text{Re}\left(u_1^H v_1\right) \text{Re}\left(u_1 v_1^H\right) \right)$$

Algorithme 1	$f(x^+)$	$\frac{\lvert x^+-x\rvert}{1+\lvert x\rvert}$	$\frac{\lvert f(x^+)-f(x)\rvert}{1+\lvert f(x)\rvert}$	Par. de proximité τ	Itér.
sans abscisse spectrale	1.82	2.66×10^{-5}	2.91×10^{-5}	1.00	1000
avec abscisse spectrale	1.28	9.98×10^{-7}	9.90×10^{-7}	0.66	995

TABLE 2.2.: Effet des tangentes décalées de l'abscisse spectrale aux pas déstabilisants.

Algorithme	$f(x^+)$	$\frac{\lvert x^+-x\rvert}{1+\lvert x\rvert}$	$\frac{\lvert f(x^+)-f(x)\rvert}{1+\lvert f(x)\rvert}$	Par. de proximité τ	Itér.
Tangentes décalées	1.08	5.69×10^{-5}	1.99×10^{-5}	1.45	565
Modèle idéal	1.08	9.97×10^{-5}	7.37×10^{-7}	2.60×10^{-2}	86

TABLE 2.3.: Résolution du problème (2.14).

où u_1 et v_1 sont respectivement des vecteurs propres à gauche et à droite de $A(y^k)$ associés à $\lambda_1(A(y^k))$. Cette technique a été appliquée à l'exemple (2.13) dont on présente les résultats Table 2.2 et Figure 2.10.

On remarque que le paramètre de proximité τ n'est plus utilisé pour rétrécir les pas, que la fonction objectif f décroît plus vite et que les pas sont nettement plus grands. De plus l'algorithme a vérifié le test d'arrêt avant d'atteindre le nombre maximum d'itérations et donne en sortie un contrôleur plus performant en norme H_∞.

2.4.2. Problème 2

Le problème guidage commande est décrit en détail dans le chapitre 3. On donne juste ici la forme du programme d'optimisation non lisse 3.5.

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) := \max\limits_{i=1,\dots,4} \left\| W_i^{-1} T_{w_i \to z_i}(\mathbf{x}) \right\|^2_{\infty,\Omega_{\text{low}}} \\ \text{subject to} & c(\mathbf{x}) := \max\limits_{i=5,6} \left\| W_i^{-1} T_{w_i \to z_i}(\mathbf{x}) \right\|^2_{\infty,\Omega_{\text{high}}} - r^2 \leq 0 \\ & \mathbf{x} \in \mathbb{R}^n \end{array} \tag{2.14}$$

On prend $r = 1.08$. Pour traiter la contrainte c dans le programme (2.14) on fait appel à une fonction de progrès dont l'utilisation et le fonctionnement sont décrits dans la section 3.3.1. Les résultats sont présentés Table 2.3 et Figure 2.11. La différence notable entre les deux algorithmes est que celui avec modèle idéal converge en beaucoup moins d'itérations (86) que celui avec tangentes décalées (565).

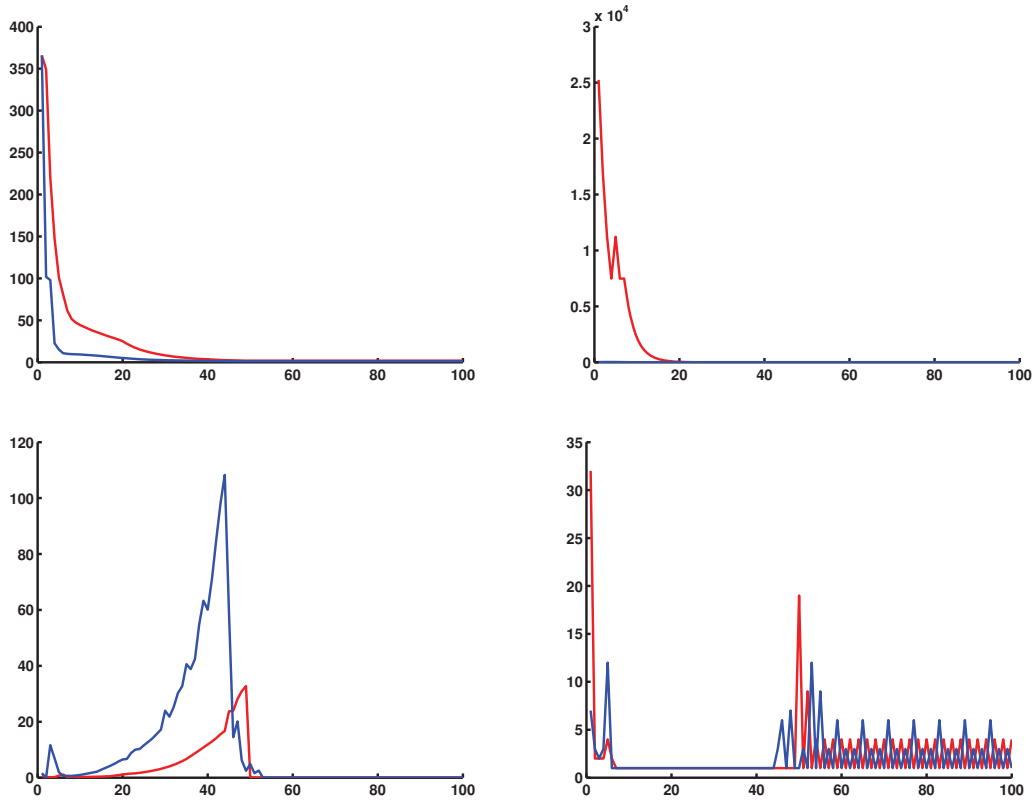


FIGURE 2.10.: Avec (en bleu) et sans (en rouge) tangentes décalées de l'abscisse spectrale aux pas déstabilisants. En haut de gauche à droite en fonction de j : $f(x^j)$ et $\tau_j^{\#}$. En bas de gauche à droite en fonction de j : $|x^j - x^{j+1}|$ et nombre d'itér. dans la boucle interne.

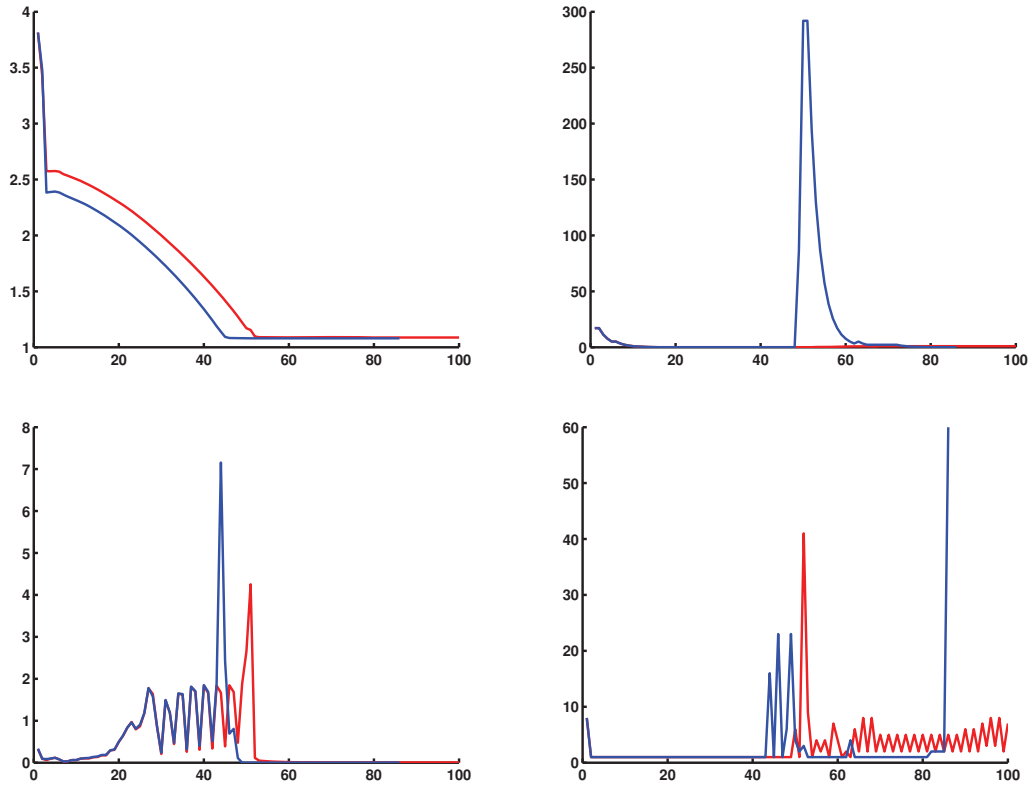


FIGURE 2.11.: Algorithme avec modèle idéal (en bleu) ou tangentes décalées (en rouge) sur le problème (2.14). En haut de gauche à droite en fonction de j : $\max(f(x^j), c(x^j))$ et $\tau_j^{\#}$. En bas de gauche à droite en fonction de j : $|x^j - x^{j+1}|$ et nombre d'itér. dans la boucle interne.

3. Design of a flight control architecture using a non-convex bundle method

3.1. Introduction

Automatic control of aircraft generally follows a scheme known as *guidance, navigation, and control* (GNC), which stipulates the use of architectures with interconnected control loops at different levels [26, 27]. Figure 3.1 presents such a multi-level control architecture for the case of longitudinal flight. The inner loop (the control loop) governs the short term dynamics in high frequency. It is represented by the *flight controller* in the red box. The outer loop (the guidance loop) serves to control the long term dynamics in low frequency, represented by the *autopilot* shown in the cyan boxes. Roughly, GNC can therefore be understood as a frequency decoupling strategy. In the case of longitudinal flight this decoupling dissociates short term rotational dynamics from long term translational modes.

An important feature in longitudinal flight is the switch between automatic and manual mode on the input of the low-level control loop. The pilot can at any moment de-activate the autopilot and switch to manual mode. Autopilot and flight controller therefore operate together in cruising mode, but in manual mode the commands of the pilot through the side-stick are interpreted as vertical load factor input references N_{z_c} and sent directly to the flight controller, which must then operate independently. In consequence, the two controllers have to be considered as decentralized units, but designed simultaneously to work satisfactory in automatic and manual mode. Due to lack of appropriate design techniques, current practice is to tune the two controller blocks independently, which leads to a lack of performance and robustness. The present work proposes a method which allows simultaneous synthesis of the full architecture.

The way we proceed is by translating simultaneous synthesis of both controller blocks into a non-smooth non-convex optimization program. We then present a non-smooth optimization method, prove its convergence, and use it to solve the control problem. Our algorithm expands on previous work [18, 28, 29] and develops the non-convex bundling technique originally put forward in [10, 28]. Here we use a progress function technique, which is motivated by older ideas for smooth problems in [30], and expands on the non-

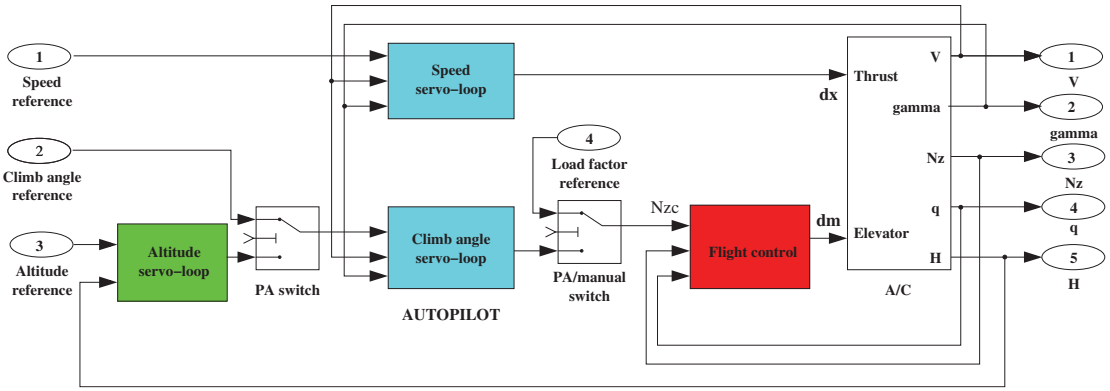


FIGURE 3.1.: Longitudinal control of an aircraft. The flight control loop (red box) controls the short term dynamics in high frequency. The autopilot (cyan boxes) controls the long term dynamics in low frequency.

smooth approach in [31]. We propose a new form of the non-convex cutting plane oracle, referred to as *down-shifted tangents*, which offers several advantages over previously used methods.

The structure of the paper is as follows. In section 3.2 we present the longitudinal control problem. Sections 3.3 – 3.4 present the non-convex bundle method and prove convergence. Section 3.5 goes back to the motivating application, gives specific information on how to compute Clarke subgradients, how to adapt the cutting plane strategy to the situation, and concludes with numerical results in longitudinal flight control.

3.2. Longitudinal flight

In this section we present the control application, going gradually from a concrete class of examples to a more abstract setting. Subsection 3.2.2 indicates how performance and robustness criteria are found, and subsection 3.2.3 presents a general setting which could be valid for other multi-objective H_∞-control problems.

3.2.1. Open-loop model

We consider an aircraft moving in the vertical plane (Figure 3.2). Its aerodynamic behavior, linearized around one particular flight point (Mach$= 0.7$, Altitude$= 5000\, ft$), is described by a set of equations of the form

$$\begin{bmatrix} \dot{x}_P \\ y_P \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_P \\ u \end{bmatrix} \tag{3.1}$$

where numerical data are given in the Appendix at the end of the chapter. Here the state is $x_P = [V,\ \gamma,\ \alpha,\ q,\ H]^T$, the control is $u = [d_x,\ d_m]^T$, and the output is $y_P = [V,\ \gamma,\ N_z,\ q,\ H]^T$. In particular,

- The states are aerodynamic speed $V\ [m/s]$, climb angle (or slope) $\gamma\ [rad]$, angle of attack $\alpha\ [rad]$, pitch rate $q = \dot{\theta} = \dot{\alpha} + \dot{\gamma}\ [rad/s]$, and altitude $H\ [m]$.
- The controls are engine thrust d_x ($\%$ of the maximal thrust) and elevator deflection $d_m\ [rad]$.
- The measurements are vertical load factor $N_z\ [m/s^2]$, and $[V, \gamma, q, H]$.

The longitudinal dynamics are characterized by 5 eigenvalues, which for the specific flight point chosen are

- $\lambda_{1,2} = -0.56 \pm 1.61\,j$ (i.e., pulsation : $1.7\,rad/s$ and damping ratio : 0.33) is the angel-of-attack (AoA) oscillation, also called short term mode. This mode mainly impacts the states α and q,
- $\lambda_{3,4} = -0.0039 \pm 0.064\,j$ (i.e., pulsation : $0.064\,rad/s$ and damping ratio : 0.06) is the phugoïd mode, also called long term mode. It mainly impacts the states γ and V,
- $\lambda_5 = -0.0026$ is the altitude convergence mode (a very long term mode). It mainly impacts the state H.

The structures of the command laws are presented in Figures 3.3 and 3.4. Practitioners prefer simple controller structures in order to address issues like saturation, interpolation of the controller according to flight operating conditions, and feedforward compensation adapted to the various aircraft configurations.

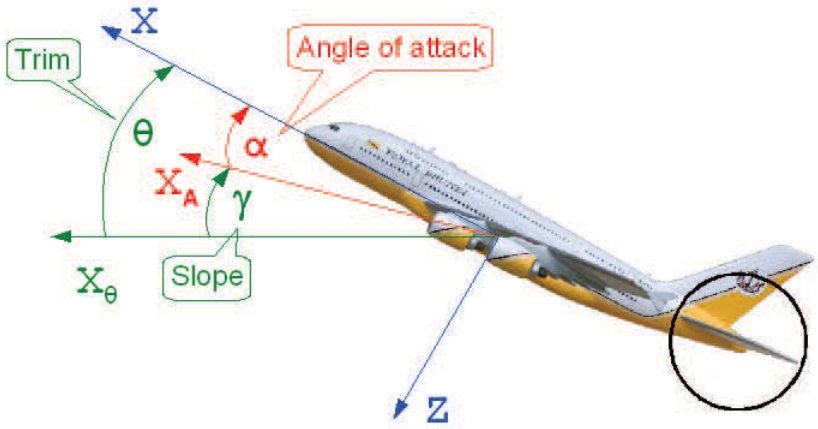


FIGURE 3.2.: Longitudinal motion of a civil aircraft.

The autopilot generates engine thrust d_x and the vertical load factor input reference N_{z_c}

$$K^{(1)}:\ \begin{bmatrix} d_x(s) \\ N_{z_c}(s) \end{bmatrix} = \begin{bmatrix} K_{p_{\text{vel}}} + \frac{K_{i_{\text{vel}}}}{s} & K_{\text{dec}} \\ 0 & K_{p_{\text{slope}}} \end{bmatrix} \begin{bmatrix} dV(s) \\ d\gamma(s) \end{bmatrix} \tag{3.2}$$

and involves a P-feedback to servo-loop the speed V, a PI-feedback to control the slope γ, and a P feedback for γ in order to decouple V from γ.

The flight control law governing the elevator deflection d_m reads

$$K^{(2)}:\ d_m(s) = F(s) \begin{bmatrix} K_p + \frac{K_i}{s+\varepsilon} & -K_v \end{bmatrix} \begin{bmatrix} N_{z_c}(s) - N_z(s) \\ q(s) \end{bmatrix} \tag{3.3}$$

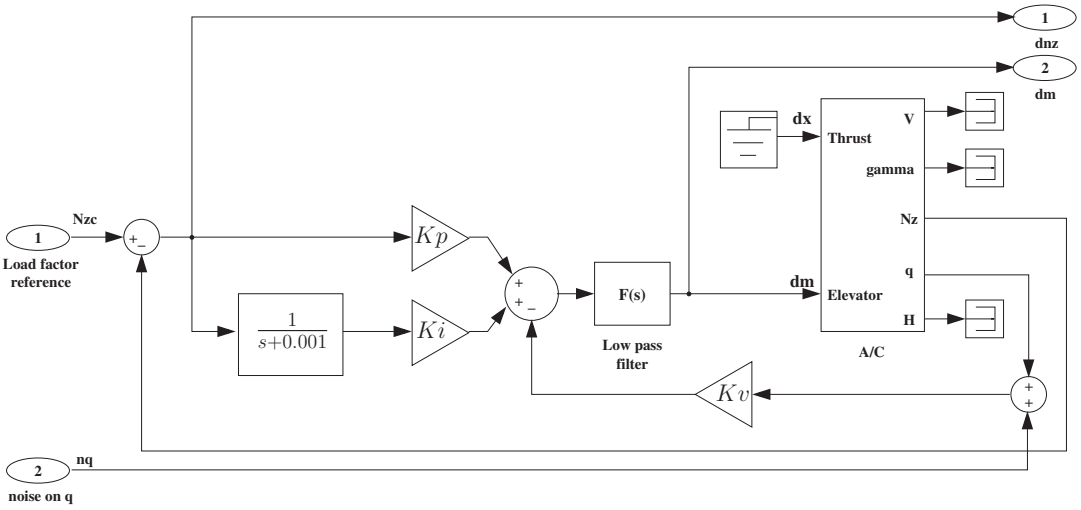


FIGURE 3.3.: Functional scheme of the flight control loop.

and combines a PI feedback to servo-loop the vertical load factor N_z with a P-feedback on the pitch rate q to damp the angle-of-attack (AoA) oscillation. In addition, the role of the low pass filter $F(s) = a/(s^2 + bs + a)$ is to prevent spill-over of unmodeled dynamics, caused mainly by flexible structural modes [32].

Remark 1. PDE-based models for flexible aircraft are currently developed, so future approaches might give better insight into the presently unmodeled structural modes. Validating such a model is outside the scope of the present contribution.

The goal is to optimize the controller gains grouped in the optimization variable

$$\mathbf{x} = [K_p; K_i; K_v; b; a; K_{p_{\text{slope}}}; K_{p_{\text{vel}}}; K_{i_{\text{vel}}}; K_{\text{dec}}],$$

in order to synthesize the two controller blocks $K^{(1)}$ and $K^{(2)}$ in such a way that performance and robustness requirements are met in automatic and manual mode.

Remark 2. In a conventional approach we would fix the low-pass filter F beforehand, and then design $K^{(1)}$ and the remaining parameters in $K^{(2)}$ separately. Our approach shows that it is preferable to design all elements simultaneously, as this leads to better performance. The conventional block-by-block design can then still be useful to initialize the optimization algorithm.

3.2.2. Controller specifications

Performance and robustness criteria are defined by introducing frequency weights on specific closed-loop transfer functions $T_i(\mathbf{x}, s) := T_{w_i \to z_i}(\mathbf{x}, s)$ between suitably chosen inputs w_i and outputs z_i. In this study we consider the six transfers $V_c \to dV$, $\gamma_c \to d\gamma$,

$\gamma_c \to dV$, $V \to d\gamma$, $N_{z_c} \to dN_z$, $(N_{z_c}, n_q) \to dm$. For each of these channels $w_i \to z_i$ we construct a state-space representation

$$P_i(s) : \begin{bmatrix} \dot{x}_i \\ z_i \\ y_i \end{bmatrix} = \begin{bmatrix} A^i & B_1^i & B_2^i \\ C_1^i & D_{11}^i & D_{12}^i \\ C_2^i & D_{21}^i & D_{22}^i \end{bmatrix} \begin{bmatrix} x_i \\ w_i \\ u_i \end{bmatrix}, \quad i = 1, \ldots, 6, \tag{3.4}$$

where $x_i \in \mathbb{R}^{n_i}$ is the state of representation P_i, $u_i \in \mathbb{R}^{m_2^i}$ the control input and $y_i \in \mathbb{R}^{p_2^i}$ the measured output. Observe that channels $i = 1, \ldots, 4$ concern the autopilot (3.2). Therefore, $\dim(u_1) = \cdots = \dim(u_4) = 2$, $\dim(y_1) = \cdots = \dim(y_4) = 2$, and we connect the same controller

$$u_i(s) = K^{(1)}(\mathbf{x}, s) y_i(s), \ i = 1, \ldots, 4$$

to the first four channels. Similarly, channels $i = 5, 6$ concern the flight controller (4.3), so that $\dim(u_5) = \dim(u_6) = 1$ and $\dim(y_5) = \dim(y_6) = 2$, and we connect the same controller

$$u_i(s) = K^{(2)}(\mathbf{x}, s) y_i(s), \ i = 5, 6$$

to the last two channels. Notice that $K^{(1)}$ depends on all 9 parameters in $\mathbf{x}$, whereas $K^{(2)}$ depends only on the flight control gains $(\mathbf{x}_1, \ldots, \mathbf{x}_5) = (K_p, K_i, K_v, b, a)$. This reflects the fact that we want $K^{(2)}$ independent of the autopilot in order to guarantee closed-loop performances during manual mode.

The rationale of these channels is as follows. The first specification for flight control is tracking of the load factor N_z. We use a template $W_5(s) = (s^2 + 4s) \,/\, (s^2 + 4s + 7)$ for $T_{N_{z_c} \to dN_z}(\mathbf{x}, s)$, where dN_z is the vertical load factor tracking error. In other words, we want $T_5 := W_5^{-1} T_{N_{z_c} \to dN_z}$ to be close to 1. The situation can be seen in Figure 3.7 left.

The second specification concerns robustness with regard to unmodeled dynamics. We want to cut off the command signal $d_m(s)$ in high frequency (roll-off). To do this, we impose the low pass template $W_6(s) = 25/\left(s^2 + \sqrt{2}\,5\, s + 25\right)$, which aims at shaping a second order roll-off beyond $5\, rad/s$, on $T_{(N_{z_c}, n_q) \to dm}(\mathbf{x}, s)$, where n_q is the pitch rate measurement noise. That means we want $T_6 = W_6^{-1}\, T_{(N_{z_c}, n_q) \to dm}$ close to 1, and this channel is visualized in Figure 3.7 right.

Remark 3. One can notice in Figure 3.7 that frequency-domain templates for T_5, T_6 need not be satisfied for pulsations under $0.1\, rad/s$. For the flight controller we are only interested in the high frequency band $\Omega_{\text{high}} = [0.1, \infty] rad/s$, as its performances concern the short-term dynamics only and are not affected even when templates are violated in very low frequency.

The specifications for the autopilot include tracking of speed and slope (climb angle). For that we introduce a template $W_1(s) = (s + 0.01) \,/\, (s + 0.2)$, which we use for $T_{V_c \to dV}(\mathbf{x})$ and $W_2 = \sqrt{2}(s + 0.01)/(s + 0.7)$ for $T_{\gamma_c \to d\gamma}(\mathbf{x})$, where dV, $d\gamma$ are the tracking errors of speed V and slope γ. We put $T_1 = W_1^{-1} T_{V_c \to dV}$ and $T_2 = W_2^{-1} T_{\gamma_c \to d\gamma}$, visualized in Figure 3.8, which we want as small as possible. Furthermore, we want to

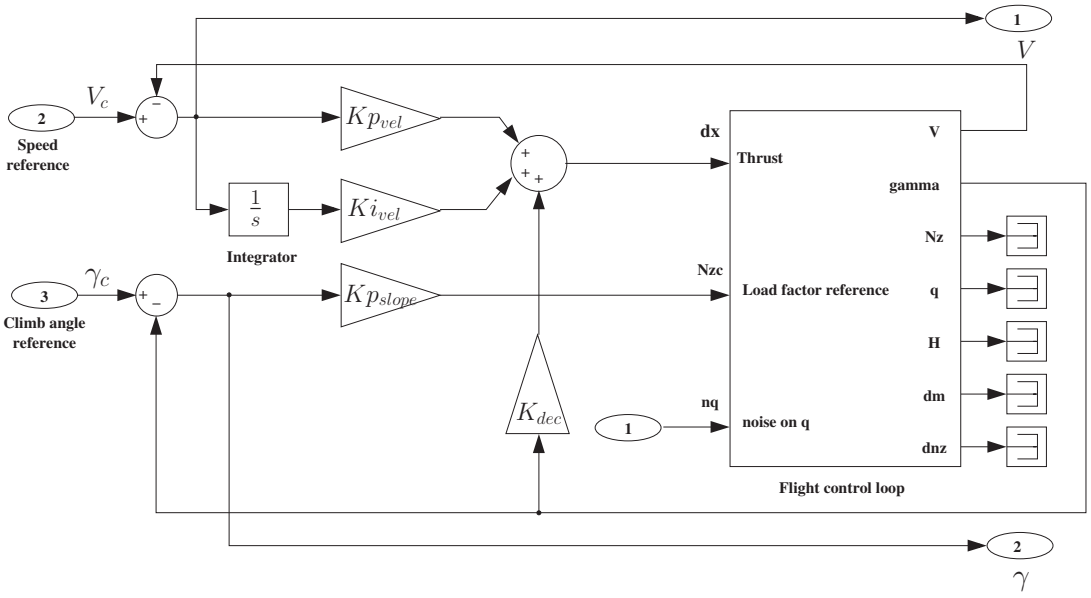


FIGURE 3.4.: Functional scheme of the guidance loop.

decouple speed and slope, and for that we impose the template 0.05 on $T_{\gamma_c \to dV}(\mathbf{x}, s)$ and $T_{V_c \to d\gamma}(\mathbf{x}, s)$. This defines T_3 and T_4, shown in Figure 3.9, which again should be small.

Remark 4. The autopilot controls the low frequency range, which means frequency-domain templates for $T_1, \ldots, T_4$ have only to be satisfied for frequencies within the low-frequency band $\Omega_{\text{low}} = [0.01, 10] rad/s$.

3.2.3. Optimization program

The performance and robustness specifications are now cast as an optimization program :

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) := \max\limits_{i=1,\ldots,4} \|T_i(\mathbf{x})\|^2_{\infty,\Omega_{\text{low}}} \\ \text{subject to} & c(\mathbf{x}) := \max\limits_{i=5,6} \|T_i(\mathbf{x})\|^2_{\infty,\Omega_{\text{high}}} - r^2 \leq 0 \\ & \mathbf{x} \in \mathbb{R}^n \end{array} \tag{3.5}$$

where objective f and constraint c represent weighted H_∞-norms on different frequency bands Ω_{low} and Ω_{high}, and where $r \approx 1$. Each of the branches in f and c has therefore the abstract form

$$f(\mathbf{x}) = \|T(\mathbf{x}, \cdot)\|^2_{\infty,\Omega} = \sup_{\omega \in \Omega} \lambda_1[\mathcal{F}(\mathbf{x}, \omega)] = \sup_{\omega \in \Omega} f(\mathbf{x}, \omega) \tag{3.6}$$

where $\lambda_1(X)$ is the maximum eigenvalue of the Hermitian matrix X, and where the mapping

$$\mathcal{F}(\mathbf{x}, \omega) = T(\mathbf{x}, j\omega) T(\mathbf{x}, j\omega)^H \tag{3.7}$$

is smooth in $\mathbf{x}$, jointly continuous in $(\mathbf{x}, \omega)$, and takes values in a space $\mathbb{H}$ of appropriately sized complex Hermitian matrices. This is due to the fact that $K(\mathbf{x})$ depends smoothly on the design parameter $\mathbf{x}$. Given the fact that the H_∞-norm is only defined for stable transfer functions, f and c are only defined on the set S of those parameters $\mathbf{x}$ where all $T_{w_i \to z_i}(\mathbf{x})$ are stable. In other words, program (3.5) has the hidden constraint $\mathbf{x} \in S$.

The salient point is that (3.5) is highly non-smooth due to the presence of the semi-infinite maximum eigenvalue function (3.6) in constraint and objective. We therefore develop a non-smooth progress function method to solve such programs algorithmically. A similar rationale was previously applied to mixed H_2/H_∞-control [31], where in contrast with (3.5) the objective function f was smooth. H_∞/H_∞-control with structured control laws $K(\mathbf{x})$ was pioneered in [33]. Optimization methods for the band-limited H_∞-norm were first discussed in [34].

Remark 5. In classical H_∞-loopshaping the use of the banded H_∞-norm is avoided mainly due to lack of methods to deal with it algorithmically. The advantage of working with banded norms is that the state-space dimension of the channel representations (3.4) is kept small. If one tries to adapt the templates W_i so that their effect is negligible outside the band Ω of interest, the state space dimension of the plants P^i increases.

Remark 6. Simple control architectures like (3.2), (4.3) are preferred by practitioners for various reasons. The building blocks are well-understood, and they are easier to hardware embed. It is therefore important to stress that it is precisely this need for simplicity which renders controller design difficult. Namely, designing an advanced full-order H_∞-controller would be much easier as it could be achieved e.g. by solving algebraic Riccati equations (AREs) or linear matrix inequalities (LMIs), but such controllers are as a rule useless in practice.

Remark 7. The gap between abstract H_∞-theory based on AREs on the one hand, and the need for practical controller structures to solve real problems on the other, has created a paradoxal situation, where controllers are tuned using heuristics, while the sophisticated techniques of H_∞-control cannot be brought to work. Our contribution helps to close this gap, as it allows to apply the H_∞-paradigm to structured controllers. We mention that this requires optimization techniques like (3.5), because even for a relatively simple structure like (3.2), (4.3) it is impossible to simply throw the blocks $K^{(1)}$, $K^{(2)}$ by hand, as there are 6 concurring performance and robustness specifications to satisfy.

3.3. Non-convex bundle method

In this section we present our non-smooth algorithm, discuss its constituents and rationale, and prove convergence. We consider an abstract version of (3.5),

$$\min\{f(\mathbf{x}) : c(\mathbf{x}) \leq 0, \mathbf{x} \in \mathbb{R}^n\}, \tag{3.8}$$

where $f, c : \mathbb{R}^n \to \mathbb{R}$ are locally Lipschitz functions. To solve (3.8) algorithmically, we assume that for every $\mathbf{x} \in \mathbb{R}^n$ we have the function value $f(\mathbf{x})$ and a Clarke subgradient $g \in \partial f(\mathbf{x})$ at our disposal, and similarly $c(\mathbf{x})$, $h \in \partial c(\mathbf{x})$. In cases where several subgradients are available, the method can be adapted to include this information.

3.3.1. Progress function and optimality conditions

We address program (3.8) by introducing a *progress function* $F(\cdot, \mathbf{x})$ at the current iterate $\mathbf{x}$,

$$F(\cdot, \mathbf{x}) = \max\{f(\cdot) - f(\mathbf{x}) - \mu c(\mathbf{x})_+, c(\cdot) - c(\mathbf{x})_+\}, \tag{3.9}$$

where $\mu > 0$ is fixed and $c(\mathbf{x})_+ = \max(c(\mathbf{x}), 0)$. The idea is as follows. Notice that $F(\mathbf{x}, \mathbf{x}) = 0$, where either the left branch $f(\cdot) - f(\mathbf{x}) - \mu c(\mathbf{x})_+$ or the right branch $c(\cdot) - c(\mathbf{x})_+$ of (3.9) is active at $\mathbf{x}$, i.e., attains the maximum, depending on whether $\mathbf{x}$ is feasible for (3.8) or not. If $c(\mathbf{x}) > 0$, meaning that $\mathbf{x}$ is infeasible, then the right hand term in (3.9) is active at $\mathbf{x}$, whereas the left hand term equals $-\mu c(\mathbf{x}) < 0$ at $\mathbf{x}$. Reducing $F(\cdot, \mathbf{x})$ below its value 0 at the current $\mathbf{x}$ therefore reduces constraint violation. The period when iterates $\mathbf{x}$ are infeasible is called phase I.

On the other hand, if $c(\mathbf{x}) \leq 0$, meaning that $\mathbf{x}$ is feasible, 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. This is phase II, where the true optimization of f takes place.

The following lemma, whose proof can be found in [31], gives an optimality test for program (3.8) based on the progress function. Recall that $\mathbf{x}^*$ satisfies the F. John necessary optimality conditions for program (3.8) if there exist $\lambda_0^* \geq 0, \lambda_1^* \geq 0$ with $\lambda_0^* + \lambda_1^* = 1$ such that $0 \in \lambda_0^* \partial f(\mathbf{x}^*) + \lambda_1^* \partial c(\mathbf{x}^*)$, $\lambda_1^* c(\mathbf{x}^*) = 0$, and $c(\mathbf{x}^*) \leq 0$. If in addition $\lambda_0^* > 0$, then $\mathbf{x}^*$ satisfies the Karush-Kuhn-Tucker conditions with associated Lagrange multiplier $\lambda_1^*/\lambda_0^* \geq 0$.

Lemma 1. *(Compare [31, Lemma 5.1]). Suppose* $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)$ *for some* $\mathbf{x}^* \in \mathbb{R}^n$*, where* ∂_1 *is the subdifferential with respect to the first coordinate. Then we have the following possibilities :*

1. *Either* $c(\mathbf{x}^*) > 0$*, in which case* $\mathbf{x}^*$ *is a critical point of* c*, called a critical point of constraint violation.*
2. *Or* $c(\mathbf{x}^*) \leq 0$*, in which case* $\mathbf{x}^*$ *satisfies the F. John necessary optimality conditions for program (3.8). In addition, there are two sub-cases :*
   a) *Either* $\mathbf{x}^*$ *is a Karush-Kuhn-Tucker point of (3.8).*
   b) *Or* $\mathbf{x}^*$ *fails to be a Karush-Kuhn-Tucker point. The latter can only happen when* $c(\mathbf{x}^*) = 0$ *and at the same time* $0 \in \partial c(\mathbf{x}^*)$. □

We plan to solve program (3.8) by constructing a sequence of iterates $\mathbf{x}^j$, such that $\mathbf{x}^{j+1}$ is a descent step for $F(\cdot, \mathbf{x}^j)$ away from $\mathbf{x}^j$. That is $F(\mathbf{x}^{j+1}, \mathbf{x}^j) < F(\mathbf{x}^j, \mathbf{x}^j) = 0$ in a qualified way. We expect $\mathbf{x}^j$ to converge to a point $\mathbf{x}^*$ satisfying $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)$. Lemma 1 tells us that $\mathbf{x}^*$ is a KKT point of program (3.8) as a rule. The exceptions from that rule are conditions 1. and 2b. Condition 1 gives the case where iterates $\mathbf{x}^j$ get stuck at a limit point $\mathbf{x}^*$ with value $c(\mathbf{x}^*) > 0$ in phase I. This is a critical point of constraint violation. (Condition 2b is the limiting case, where $c(\mathbf{x}^*) = 0$. This case was never observed in our experiments and appears unlikely in practice.) A first order method may indeed get trapped at such points, and in classical mathematical programming second order techniques are used to avoid them. Here we are working with a non-smooth program, where second order elements are not available. When critical points of constraint violation are encountered, we restart our method at a different initial guess.

When reducing constraint violation in phase I, a controlled increase in f not exceeding $\mu c(\mathbf{x})$ is granted. This helps the algorithm in not being trapped at infeasible critical points of f alone. For the theoretical justification see Section 3.4.

The algorithm used to compute solutions to (3.8) is shown schematically in Figure 3.4, and stated formally as Algorithm 3 in section 3.10. We subsequently describe its essential features.

3.3.2. Working model

We denote the current serious iterate of the algorithm by $\mathbf{x}$, or $\mathbf{x}^j$ if the counter j of the outer loop is used. If a new serious iterate is found, it will be denoted by $\mathbf{x}^+$, or $\mathbf{x}^{j+1}$. Serious iterates refer to the outer loop colored blue in Figure 3.4.

At the current iterate $\mathbf{x}$ we use approximations $F_k(\cdot, \mathbf{x})$ of the progress function $F(\cdot, \mathbf{x})$ called working models. Every working model satisfies $F_k(\mathbf{x}, \mathbf{x}) = 0$ and $\partial_1 F_k(\mathbf{x}, \mathbf{x}) \subset \partial_1 F(\mathbf{x}, \mathbf{x})$. Moreover, the F_k decompose into a polyhedral convex possibly non-smooth first-order part, $F_k^{[1]}(\cdot, \mathbf{x}) = \max_{(a,g)\in\mathcal{G}_k} a + g^\top(\cdot - \mathbf{x})$, and a nonconvex but smooth second-order part $F^{[2]}(\cdot, \mathbf{x}) = \frac{1}{2}(\cdot - \mathbf{x})^\top Q(\mathbf{x})(\cdot - \mathbf{x})$:

$$F_k(\cdot, \mathbf{x}) = \max_{(a,g)\in\mathcal{G}_k} a + g^\top(\cdot - \mathbf{x}) + \tfrac{1}{2}(\cdot - \mathbf{x})^\top Q(\mathbf{x})(\cdot - \mathbf{x}). \tag{3.10}$$

Here $\mathcal{G}_k \subset \mathbb{R}^n \times \mathbb{R}^n$ is a finite set, which we update continuously during the inner loop with counter k, colored yellow in Figure 3.4. In contrast, the second order term $F^{[2]}(\cdot, \mathbf{x}) = \frac{1}{2}(\cdot - \mathbf{x})^\top Q(\mathbf{x})(\cdot - \mathbf{x})$ is held fixed during the inner loop and only updated between serious steps $\mathbf{x} \to \mathbf{x}^+$. We allow $Q(\mathbf{x}) \in \mathbb{S}^n$ to be indefinite, and we assume that the operator $\mathbf{x} \mapsto Q(\mathbf{x})$, $\mathbb{R}^n \to \mathbb{S}^n$, is bounded on bounded sets. Our notation $F_k(\cdot, \mathbf{x}) = F_k^{[1]}(\cdot, \mathbf{x}) + F^{[2]}(\cdot, \mathbf{x})$ highlights that the second order part does not depend on k.

3.3.3. Tangent program

In the inner loop at serious iterate $\mathbf{x}$ we generate trial steps $\mathbf{y}^k$ indexed by the counter k of the inner loop, which are candidates to be elected as the new serious iterate $\mathbf{x}^+$. The trial step $\mathbf{y}^k$ is obtained by solving the convex tangent program

$$\min_{\mathbf{y}\in\mathbb{R}^n} F_k(\mathbf{y},\mathbf{x}) + \tfrac{\tau_k}{2}\left\|\mathbf{y}-\mathbf{x}\right\|^2. \tag{3.11}$$

Here τ_k is the proximity control parameter, which is updated during the inner loop. Convexity of (3.11) is assured because we require $Q(\mathbf{x}) + \tau_k I \succ 0$ for every k, where $\succ 0$ means positive definite. Observe that (3.11) is equivalent to the convex quadratic program (CQP)

$$\begin{array}{ll} \text{minimize} & t + \frac{1}{2}(\mathbf{y}-\mathbf{x})^\top (Q(\mathbf{x})+\tau_k I)(\mathbf{y}-\mathbf{x}) \\ \text{subject to} & a + g^\top(\mathbf{y}-\mathbf{x}) \le t \\ & (a,g) \in \mathcal{G}_k \end{array} \tag{3.12}$$

with unknown variable $(t,\mathbf{y}) \in \mathbb{R}^{1+n}$, which can be conveniently solved with standard CQP solvers.

The necessary optimality condition for (3.11) is $\tau_k(\mathbf{x}-\mathbf{y}^k) \in \partial_1 F_k(\mathbf{y}^k,\mathbf{x})$, or equivalently,

$$g_k^* := (Q(\mathbf{x})+\tau_k I)(\mathbf{x}-\mathbf{y}^k) \in \partial_1 F_k^{[1]}(\mathbf{y}^k,\mathbf{x}), \tag{3.13}$$

and we call g_k^* the aggregate subgradient. Equivalently, there exist pairs (a_1,g_1), …, $(a_r,g_r) \in \mathcal{G}_k$ and $\lambda_i > 0$, $\sum_{i=1}^r \lambda_i = 1$, such that

$$a_i + g_i^\top(\mathbf{y}^k-\mathbf{x}) = t_k,\ i=1,\dots,r \qquad (Q(\mathbf{x})+\tau_k I)(\mathbf{x}-\mathbf{y}^k) = \sum_{i=1}^r \lambda_i g_i, \tag{3.14}$$

where $t_k = F_k^{[1]}(\mathbf{y}^k,\mathbf{x})$. Putting $a_k^* = \sum_{i=1}^r \lambda_i a_i$, we call $m_k^*(\cdot,\mathbf{x}) = a_k^* + g_k^{*\top}(\cdot-\mathbf{x})$ the aggregate plane. We say that the subgradients $g_1,\dots,g_r$ are *called* by the aggregate subgradient, and that the planes $a_i + g_i^\top(\cdot-\mathbf{x})$ are called by the aggregate plane. An equivalent way to define the aggregate plane is to use (3.13) and choose a_k^* such that $m_k^*(\cdot,\mathbf{x}) = a_k^* + g_k^{*\top}(\cdot-\mathbf{x})$ has value $t_k = F_k^{[1]}(\mathbf{y}^k,\mathbf{x})$ at $\mathbf{y}^k$.

When building the new set $\mathcal{G}_{k+1}$ after a null step $\mathbf{y}^k$, we assure that $(a_k^*,g_k^*) \in \mathcal{G}_{k+1}$. This allows us to drop any of the older $(a_i,g_i) \in \mathcal{G}_k$ called by the aggregate pair.

3.3.4. Acceptance test

In order to decide whether the solution $\mathbf{y}^k$ of (3.11) is acceptable to become the new serious iterate $\mathbf{x}^+$ in the outer loop, we use the test

$$\rho_k = \frac{F(\mathbf{y}^k,\mathbf{x})}{F_k(\mathbf{y}^k,\mathbf{x})} \overset{?}{\ge} \gamma, \tag{3.15}$$

where $0 < \gamma < 1$ is fixed throughout. As usual, this test compares actual decrease and predicted decrease at $\mathbf{y}^k$. If F_k represents F accurately at $\mathbf{y}^k$, we expect $\rho_k \approx 1$, but we accept $\mathbf{y}^k$ as the new $\mathbf{x}^+$ already when $\rho_k \geq \gamma$. According to standard terminology in bundle methods, $\mathbf{y}^k$ is called a null step if $\rho_k < \gamma$, while the case $\rho_k \geq \gamma$, when $\mathbf{x}^+ = \mathbf{y}^k$, is referred to as a serious step.

3.3.5. Cutting planes

If the trial step $\mathbf{y}^k$ fails the acceptance test (3.15), then agreement between F and F_k at $\mathbf{y}^k$ was bad. In this case the inner loop has to continue, but we have to improve the quality of the next working model $F_{k+1}(\cdot, \mathbf{x})$ in order to do better at the next trial. Since the second order part $F^{[2]}(\cdot, \mathbf{x})$ of the model does not change during the inner loop k, we have to improve the first-order part $F^{[1]}_{k+1}(\cdot, \mathbf{x})$. In traditional bundle methods this is achieved by including a cutting plane into the new working model, whose role is to cut away the unsuccessful trial step $\mathbf{y}^k$. In the convex case cutting planes are simply tangents to the first-order part $F^{[1]}(\cdot, \mathbf{x})$ of the progress function $F(\cdot, \mathbf{x})$ at $\mathbf{y}^k$. Without convexity it is more delicate to obtain a suitable cutting plane. In this study we use downshifted tangents as substitutes for the traditional convex cutting planes. Here is the construction.

In accordance with the decomposition of the working model $F_k(\cdot, \mathbf{x})$, we decompose the progress function

$$F(\cdot, \mathbf{x}) = F^{[1]}(\cdot, \mathbf{x}) + F^{[2]}(\cdot, \mathbf{x}),$$

where $F^{[2]}(\cdot, \mathbf{x}) = \frac{1}{2}(\cdot - \mathbf{x})^\top Q(\mathbf{x})(\cdot - \mathbf{x})$ is the second-order part, and $F^{[1]} = F - F^{[2]}$ is the first-order part.

Given the null step $\mathbf{y}^k$, pick a subgradient $g_k \in \partial_1 F^{[1]}(\mathbf{y}^k, \mathbf{x})$. Then the affine function $t_k(\cdot) = F^{[1]}(\mathbf{y}^k, \mathbf{x}) + g_k^\top(\cdot - \mathbf{y}^k)$ is a tangent to $F^{[1]}(\cdot, \mathbf{x})$ at $\mathbf{y}^k$. Without convexity we may not use $t_k(\cdot)$ directly as a cutting plane. We do not even know whether $t_k(\mathbf{x}) \leq F^{[1]}(\mathbf{x}, \mathbf{x}) = F(\mathbf{x}, \mathbf{x}) = 0$, as would be the minimum requirement for a plane contributing to the new model $F^{[1]}_{k+1}(\cdot, \mathbf{x})$. We therefore define the down-shift as

$$s_k = [t_k(\mathbf{x})]_+ + c\|\mathbf{y}^k - \mathbf{x}\|^2, \tag{3.16}$$

where $c > 0$ is some small constant fixed at the beginning. Now we define the cutting plane as

$$m_k(\cdot, \mathbf{x}) = t_k(\cdot) - s_k. \tag{3.17}$$

Notice that $\nabla m_k(\cdot, \mathbf{x}) = \nabla t_k(\cdot) = g_k$, while $m_k(\mathbf{x}, \mathbf{x}) \leq -c\|\mathbf{y}^k - \mathbf{x}\|^2 \leq 0$. The cutting plane can also be written as $m_k(\cdot, \mathbf{x}) = a_k + g_k^\top(\cdot - \mathbf{x})$, where

$$a_k = t_k(\mathbf{x}) - s_k = t_k(\mathbf{x}) - [t_k(\mathbf{x})]_+ - c\|\mathbf{y}^k - \mathbf{x}\|^2.$$

The cutting plane depends on the full information $\mathbf{x}$, $\mathbf{y}^k$, and $g_k \in \partial_1 F^{[1]}(\mathbf{y}^k, \mathbf{x})$, whereas $t_k(\cdot)$ only depends on $\mathbf{y}^k$ and the specific subgradient g_k at $\mathbf{y}^k$. We assure that $\mathcal{G}_{k+1}$ contains the newly generated pair (a_k, g_k).

3.3.6. Exploiting the structure of the progress function

The construction of the cutting plane in section 3.3.5 does not fully exploit the structure of the first-order part $F^{[1]}$ of the progress function F. Namely, observe that

$$\begin{aligned} F^{[1]}(\cdot,\mathbf{x}) &= \max\left\{f(\cdot)-f(\mathbf{x})-\mu c(\mathbf{x})_+ - F^{[2]}(\cdot,\mathbf{x}), c(\cdot)-c(\mathbf{x})_+ - F^{[2]}(\cdot,\mathbf{x})\right\} \\ &=: \max\left\{F^{[11]}(\cdot,\mathbf{x}), F^{[12]}(\cdot,\mathbf{x})\right\}, \end{aligned} \tag{3.18}$$

and so far our construction only includes a down-shifted tangent to that part $F^{[1i]}$ of $F^{[1]}$ which is active at $\mathbf{y}^k$. It is beneficial to include also a down-shifted tangent to the inactive part. Indeed, suppose for instance $F_k^{[11]}(\mathbf{y}^k,\mathbf{x}) < F_k^{[12]}(\mathbf{y}^k,\mathbf{x})$. Then in section 3.3.5 we included a downshifted tangent to $F_k^{[12]}$ into $\mathcal{G}_{k+1}$. Now let $\widetilde{t}_k(\cdot)$ be a tangent to the inactive part $F_k^{[11]}$ at $\mathbf{y}^k$. Then we build $\widetilde{m}_k(\cdot,\mathbf{x}) = \widetilde{t}_k(\cdot) - \widetilde{s}_k$, where $\widetilde{s}_k = \left[\widetilde{t}_k(\mathbf{x})\right]_+ + c\|\mathbf{y}^k-\mathbf{x}\|^2$ just as in (3.16), that is, we downshift with respect to the value $F(\mathbf{x},\mathbf{x})=0$ at $\mathbf{x}$, and not with respect to the potentially lower value $F^{[11]}(\mathbf{x},\mathbf{x})$. This generalized cutting plane $\widetilde{m}_k$, when added into $\mathcal{G}_{k+1}$, may have some beneficial secondary effect. Even though it is inactive at $\mathbf{y}^k$, it may become active elsewhere, just as the branch $F^{[i]}$ of F inactive at $\mathbf{x}$ may become active as we move away from $\mathbf{x}$. The inactive plane $\widetilde{m}_k$ has therefore an anticipative effect, and we sometimes call these planes anticipated cutting planes.

3.3.7. Exactness and recycling

In order to guarantee $\partial_1 F_k(\mathbf{x},\mathbf{x}) \subset \partial_1 F(\mathbf{x},\mathbf{x})$ we keep at least one plane of the form $m_0(\cdot,\mathbf{x}) = g_0^\top(\cdot-\mathbf{x})$ in the model at all times k. We call m_0 an exactness plane, because it assures $F_k(\mathbf{x},\mathbf{x})=0$. Formally $(0,g_0)\in\mathcal{G}_k$ for all k. As it may happen that $\partial_1 F(\mathbf{x},\mathbf{x})$ is not singleton, we are free to add other exactness planes $(0,g')$, $g'\in\partial_1 F(\mathbf{x},\mathbf{x})$ into $\mathcal{G}_k$, for instance, one at each inner loop step k.

When a serious step $\mathbf{x}\to\mathbf{x}^+$ is made, the old working model is lost, and we will have to start $\mathcal{G}_1$ anew when the inner loop starts. This is in contrast with convex bundle methods, where all planes accumulated on the way may stay in $\mathcal{G}$ forever. The only reason to not keep them all is to avoid overflow. In contrast, in the nonconvex case we lose planes from previous serious steps for the following reason : the plane $m(\cdot,\mathbf{x}) = a + g^\top(\cdot-\mathbf{x})$ stored in $\mathcal{G}$ will in general be useless at $\mathbf{x}^+$, because we may have $m(\mathbf{x}^+,\mathbf{x}) \geq F(\mathbf{x}^+,\mathbf{x}^+)=0$. We therefore propose to recycle the old plane $m(\cdot,\mathbf{x})$ as

$$m(\cdot,\mathbf{x}^+) = m(\cdot,\mathbf{x}) - s^+,$$

with s^+ the downshift at $\mathbf{x}^+$. That is

$$s^+ = \left[m(\mathbf{x}^+,\mathbf{x})\right]_+ + c\|\mathbf{x}^+-\mathbf{x}\|^2.$$

Formally, if $(a,g)\in\mathcal{G}_{k_j}$ at the end of the j^{th} inner loop occurring at counter $k=k_j$, then let $a^+ = a - s^+$ and put $(a^+,g)\in\mathcal{G}_1$ at the beginning of the $(j+1)^{\text{st}}$ inner loop.

3.3.8. Management of the proximity parameter

At the core of algorithm 3 is the management of τ during the inner loop. According to step 7 the τ-parameter is never decreased during the inner loop. It is increased when $\rho_k < \gamma$, $\widetilde{\rho}_k \geq \widetilde{\gamma}$, and held constant when $\rho_k < \gamma$, $\widetilde{\rho}_k < \widetilde{\gamma}$. The test

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

where $\gamma < \widetilde{\gamma} < 1$ is fixed throughout, compares working models $F_{k+1}(\cdot, \mathbf{x})$ and $F_k(\cdot, \mathbf{x})$ at $\mathbf{y}^k$. If $\widetilde{\rho}_k < \widetilde{\gamma}$, then agreement between the two is bad, while $\widetilde{\rho}_k \geq \widetilde{\gamma}$ means it is not bad. The interpretation of step 7 is that $\rho_k < \gamma$ in tandem with $\widetilde{\rho}_k \geq \widetilde{\gamma}$ means F_k is far from F at $\mathbf{y}^k$, but at the same time F_k is reasonably close to F_{k+1} at $\mathbf{y}^k$. Now as F_{k+1} is supposed to make progress toward F, this constellation ($\rho_k < \gamma, \widetilde{\rho}_k \geq \widetilde{\gamma}$) tells us that the intended progress is too marginal. This is where we increase $\tau_{k+1} = 2\tau_k$ to force smaller steps at the next sweep $k+1$. The opposite situation $\rho_k < \gamma$ and $\widetilde{\rho}_k < \widetilde{\gamma}$ is considered as still open. Keeping $\tau_{k+1} = \tau_k$ fixed, we rely on improving F_{k+1} by adding cutting planes and the aggregate plane.

Observe that $F_{k+1}(\mathbf{y}^k, \mathbf{x}) \geq F_k(\mathbf{y}^k, \mathbf{x})$, because the aggregate plane, which contributes to F_{k+1}, knows the value of F_k at $\mathbf{y}^k$. Since $F_k(\mathbf{y}^k, \mathbf{x}) < 0$, the quotient $\widetilde{\rho}_k$ satisfies $\widetilde{\rho}_k \leq 1$.

3.3.9. Management of the proximity parameter between serious steps

As soon as a serious step $\mathbf{x} \to \mathbf{x}^+$ is made, we need to pass the τ-parameter on to the next inner loop. This is done via the memory element $\tau^\sharp$. We proceed as follows. If $\rho_k \geq \Gamma$, where $0 < \gamma < \Gamma < 1$, then we decrease the τ-parameter, as agreement between model and reality is *good*. If $\gamma \leq \rho_k < \Gamma$, then agreement is *not bad*, and we keep τ as is. This is organized in step 8. We re-set $\tau^\sharp = T$ if the preceding inner loop terminates with $\tau > T$. One can also dispense with this re-set, see [29] for details.

Algorithm 3 Proximity control algorithm for (3.8).

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

1: **Initialize outer loop**. Choose initial serious iterate $\mathbf{x}^1$ and initial 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 and serious iterate $\mathbf{x}^j$, 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 $k = 1$ and initialize $\tau_1 = \tau_j^\sharp$. Build working model $F_1(\cdot, \mathbf{x}^j)$ by using initial set $\mathcal{G}_1$ and matrix Q_j.

4: **Trial step generation.** At inner loop counter k solve tangent program

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

The solution is the new trial step $\mathbf{y}^k$.

5: **Acceptance test.** Check whether

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

If this is the case put $\mathbf{x}^{j+1} = \mathbf{y}^k$ (serious step), quit inner loop and goto step 8. If this is not the case (null step) continue inner loop with step 6.

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 downshift (3.17). Compute aggregate plane $m_k^*(\cdot, \mathbf{x}^j) = a_k^* + g_k^{*\top}(\cdot - \mathbf{x}^j)$ at $\mathbf{y}^k$. Build $\mathcal{G}_{k+1} = \mathcal{G}_k \cup \{(a_k, g_k), (a_k^*, g_k^*)\}$. In order to keep the size of $\mathcal{G}_{k+1}$ reasonable allow removing some of the elements of $\mathcal{G}_k$ called for by the aggregate plane. Build new working model $F_{k+1}(\cdot, \mathbf{x}^j)$.

7: **Update proximity control parameter.** Compute secondary control parameter

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

Then decide as follows. Put

$$\tau_{k+1} = \begin{cases} \tau_k, & \text{if } \widetilde{\rho}_k < \widetilde{\gamma} \qquad \text{(bad)} \\ 2\tau_k, & \text{if } \widetilde{\rho}_k \geq \widetilde{\gamma} \qquad \text{(too bad)} \end{cases}$$

Then increase inner loop counter k and continue inner loop with 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

$$\tau_{j+1}^\sharp = \begin{cases} \tau_{k+1}, & \text{if } \gamma \leq \rho_k < \Gamma \qquad \text{(not bad)} \\ \\ \frac{1}{2}\tau_{k+1}, & \text{if } \rho_k \geq \Gamma \qquad \text{(good)} \end{cases}$$

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

3.4. Convergence analysis

In this section we state and prove a convergence result for algorithm 3. We shall require the notion of lower C^1-functions introduced by Spingarn [35]. More generally, following [36], a locally Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}$ is called lower C^k at $\mathbf{x}_0$ if there exists a compact space K and a continuous function $F : B(\mathbf{x}_0, \delta) \times K \to \mathbb{R}$ for which all partial derivatives of order $\leq k$ with respect to $\mathbf{x}$ are also continuous, such that

$$f(\mathbf{x}) = \max_{\mathbf{y} \in K} F(\mathbf{x}, \mathbf{y}) \tag{3.19}$$

for every $\mathbf{x} \in B(\mathbf{x}_0, \delta)$. The function f is called lower C^k if it is lower C^k at every $\mathbf{x} \in \mathbb{R}^n$. According to [36] lower C^2-functions are lower C^k for every $k \geq 2$. On the other hand, the class of lower C^1-functions is strictly larger than lower C^2, and sufficiently large to include all practical situations.

Theorem 1. *Suppose the program data f and c in (3.8) are locally Lipschitz lower C^1-functions. In addition, let the following conditions be satisfied :*

(a) f is weakly coercive on the constraint set $\Omega = \{\mathbf{x} \in \mathbb{R}^n : c(\mathbf{x}) \leq 0\}$, i.e., if $\mathbf{x}^j$ is a sequence of feasible iterates with $\|\mathbf{x}^j\| \to \infty$, then $f(\mathbf{x}^j)$ is not monotonically decreasing.

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

Then the sequence of serious steps $\mathbf{x}^j$ generated by algorithm 3 is bounded. It either ends finitely with $0 \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$, or it is infinite, in which case every accumulation point $\mathbf{x}^*$ of $\mathbf{x}^j$ satisfies $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)$. In particular, $\mathbf{x}^*$ is either a critical point of constraint violation, or a KKT-point of (3.8).

Here, motivated by Lemma 1, we shall call $\mathbf{x}^*$ a critical point of constraint violation, if $0 \in \partial c(\mathbf{x}^*)$ in tandem with $c(\mathbf{x}^*) \geq 0$. The proof is divided into several Lemmas. The first step is to prove that the inner loop ends finitely. We write $\mathbf{x}$ for the current serious iterate $\mathbf{x}^j$, and Q for the matrix $Q(\mathbf{x}^j)$.

Lemma 2. *Suppose the inner loop at serious iterate $\mathbf{x}$ turns infinitely, i.e., $\rho_k < \gamma$ for all $k \in \mathbb{N}$. Then there exists $k_0 \in \mathbb{N}$ such that $\tau_k = \tau_{k_0}$ for all $k \geq k_0$.*

Proof: i) Suppose on the contrary that the control parameter is increased infinitely often. Then, as it is never decreased in the inner loop, we must have $\tau_k \to \infty$. We will show that this implies $0 \in \partial_1 F(\mathbf{x}, \mathbf{x})$, contradicting step 2 of the algorithm. Indeed, the inner loop is only entered when $0 \notin \partial_1 F(\mathbf{x}, \mathbf{x})$. Notice that when $\tau_k \to \tau_{k+1}$ is increased, we have $\widetilde{\rho}_k \geq \widetilde{\gamma}$, so we have an infinity of counters $k \in \mathcal{K}$ where this happens.

ii) Recall that by (3.13) the aggregate subgradient satisfies $g_k^* = (Q + \tau_k I)(\mathbf{x} - \mathbf{y}^k) \in \partial_1 F_k^{[1]}(\mathbf{y}^k, \mathbf{x})$. By the subgradient inequality we have

$$(\mathbf{x} - \mathbf{y}^k)^\top (Q + \tau_k I)(\mathbf{x} - \mathbf{y}^k) \leq F_k^{[1]}(\mathbf{x}, \mathbf{x}) - F_k^{[1]}(\mathbf{y}^k, \mathbf{x}) = -F_k^{[1]}(\mathbf{y}^k, \mathbf{x}). \tag{3.20}$$

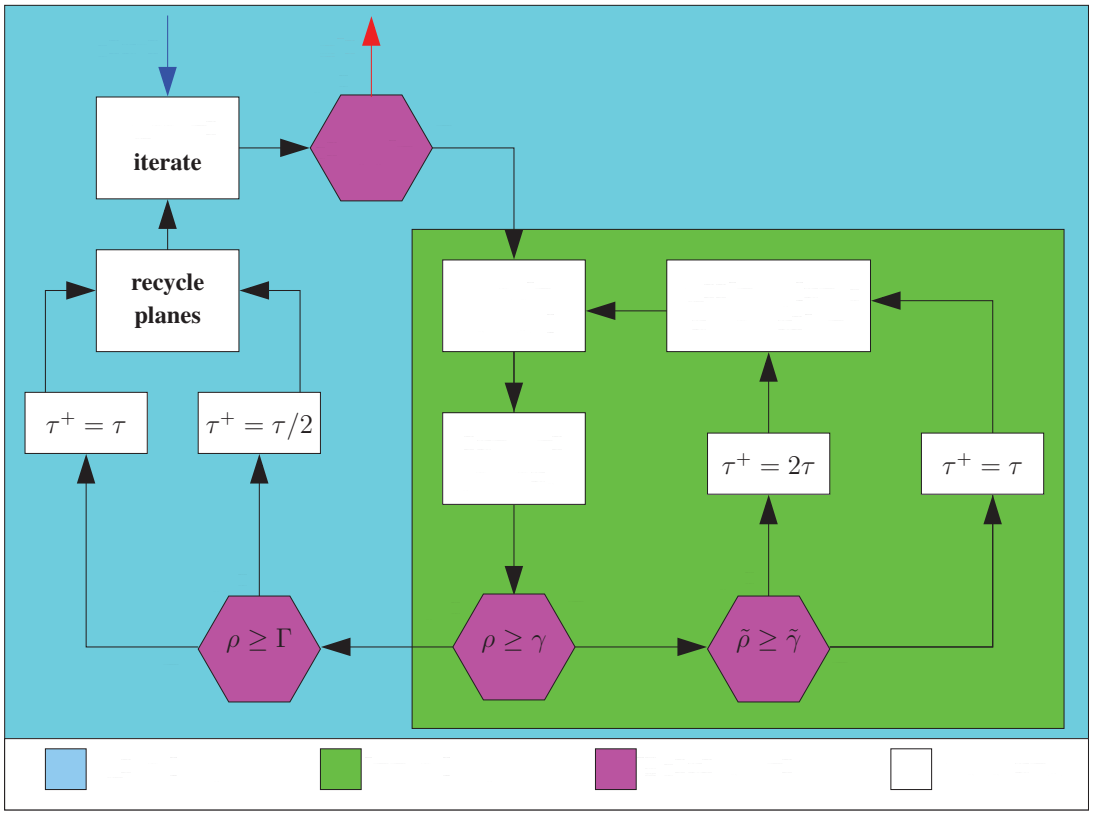


FIGURE 3.5.: Flowchart of proximity control algorithm

Recall that $m_0(\cdot, \mathbf{x}) \leq F_k^{[1]}(\cdot, \mathbf{x})$, where $m_0(\cdot, \mathbf{x}) = g_0^\top(\cdot - \mathbf{x})$ is the exactness plane at $\mathbf{x}$. Substituting this in (3.20) implies

$$(\mathbf{x} - \mathbf{y}^k)^\top (Q + \tau_k I)(\mathbf{x} - \mathbf{y}^k) \leq g_0^\top (\mathbf{x} - \mathbf{y}^k) \leq \|g_0\| \|\mathbf{x} - \mathbf{y}^k\|.$$

Since $\tau_k \to \infty$, the left hand side behaves asymptotically like $\tau_k \|\mathbf{x} - \mathbf{y}^k\|^2$. In other words, fixing $0 < \zeta < 1$, we may assume that it is minorized by $(1 - \zeta)\tau_k \|\mathbf{x} - \mathbf{y}^k\|^2$ for k large enough. After dividing a factor $\|\mathbf{x} - \mathbf{y}^k\|$ we obtain $(1 - \zeta)\tau_k \|\mathbf{x} - \mathbf{y}^k\| \leq \|g_0\|$, which implies boundedness of $\tau_k(\mathbf{x} - \mathbf{y}^k)$, and therefore also boundedness of the sequence g_k^*. As $\tau_k \to \infty$, we deduce $\mathbf{y}^k \to \mathbf{x}$ and $(\mathbf{x} - \mathbf{y}^k)^\top (Q + \tau_k I)(\mathbf{x} - \mathbf{y}^k) \to 0$.

iii) Subtracting $\frac{1}{2}(\mathbf{x} - \mathbf{y}^k)^\top Q(\mathbf{x} - \mathbf{y}^k)$ on both sides of (3.20) gives

$$\tfrac{1}{2}(\mathbf{x} - \mathbf{y}^k)^\top Q(\mathbf{x} - \mathbf{y}^k) + \tau_k \|\mathbf{x} - \mathbf{y}^k\|^2 \leq -F_k(\mathbf{y}^k, \mathbf{x}).$$

Fix $0 < \zeta < 1$. As $\tau_k \to \infty$, we have for $k \in \mathcal{K}$ sufficiently large

$$(1 - \zeta)\tau_k \|\mathbf{x} - \mathbf{y}^k\| \leq \|g_k^*\| \leq (1 + \zeta)\tau_k \|\mathbf{x} - \mathbf{y}^k\|.$$

Indeed, by the definition (3.13) of the aggregate subgradient g_k^* we have $\|g_k^*\|/(\tau_k \|\mathbf{x} - \mathbf{y}^k\|) = \|(\tau_k^{-1} Q + I)\|\mathbf{x} - \mathbf{y}^k\|/\|\mathbf{x} - \mathbf{y}^k\| \to 1$, in view of $\tau_k^{-1} \to 0$, hence $1 - \zeta < \|g_k^*\|/(\tau_k \|\mathbf{x} - \mathbf{y}^k\|) < 1 + \zeta$ for k large enough. A similar argument shows

$$\tfrac{1}{2}(\mathbf{x} - \mathbf{y}^k)^\top Q(\mathbf{x} - \mathbf{y}^k) + \tau_k \|\mathbf{x} - \mathbf{y}^k\|^2 \geq (1 - \zeta)\tau_k \|\mathbf{x} - \mathbf{y}^k\|^2$$

for $k \in \mathcal{K}$ large enough. Combining these estimates gives

$$-F_k(\mathbf{y}^k, \mathbf{x}) \geq \tfrac{1-\zeta}{1+\zeta}\|g_k^*\|\|\mathbf{x}-\mathbf{y}^k\|. \tag{3.21}$$

iv) Now we argue that $F_k(\mathbf{y}^k, \mathbf{x}) \to F(\mathbf{x}, \mathbf{x}) = 0$. Going back to the subgradient inequality (3.20), we see that the left hand side tends to 0 by iii). Hence

$$0 \leq \liminf(-F_k^{[1]}(\mathbf{y}^k, \mathbf{x})),$$

or equivalently, $\limsup F_k^{[1]}(\mathbf{y}^k, \mathbf{x}) \leq 0$. It therefore remains to prove

$$\liminf F_k^{[1]}(\mathbf{y}^k, \mathbf{x}) \geq 0.$$

To prove this, observe that $F_k^{[1]}(\mathbf{y}^k, \mathbf{x}) \geq m_0(\mathbf{y}^k, \mathbf{x})$ for the exactness plane $m_0(\cdot, \mathbf{x})$ at $\mathbf{x}$. Since $m_0(\mathbf{y}^k, \mathbf{x}) \to m_0(\mathbf{x}, \mathbf{x}) = 0$ due to iii), the claim follows.

v) Now let $\eta_k := \operatorname{dist}(g_k^*, \partial_1 F(\mathbf{x}, \mathbf{x}))$. We prove $\eta_k \to 0$. Using the subgradient inequality we have for a fixed vector $\mathbf{y}$

$$g_k^{*\top}(\mathbf{y}-\mathbf{y}^k) + F_k^{[1]}(\mathbf{y}^k, \mathbf{x}) \leq F_k^{[1]}(\mathbf{y}, \mathbf{x}) = m_{\mathbf{z}_k(\mathbf{y})}(\mathbf{y}, \mathbf{x}),$$

where $m_{\mathbf{z}_k(\mathbf{y})}(\cdot, \mathbf{x})$ is a cutting plane at $\mathbf{z}_k(\mathbf{y}) \in \{\mathbf{y}^1, \ldots, \mathbf{y}^k\}$ with respect to serious iterate $\mathbf{x}$, contributing to the build-up of model $F_k^{[1]}(\cdot, \mathbf{x})$, and exact at $\mathbf{y}$. In other words

$$m_{\mathbf{z}_k(\mathbf{y})}(\cdot, \mathbf{x}) = F^{[1]}(\mathbf{z}_k(\mathbf{y}), \mathbf{x}) + g_{\mathbf{z}_k(\mathbf{y})}^\top(\cdot - \mathbf{z}_k(\mathbf{y})) - s$$

where $g_{\mathbf{z}_k(\mathbf{y})} \in \partial_1 F^{[1]}(\mathbf{z}_k(\mathbf{y}), \mathbf{x})$ and where $s = s(\mathbf{z}_k(\mathbf{y}), \mathbf{x})$ is the downshift at $\mathbf{z}_k(\mathbf{y})$ with respect to $\mathbf{x}$. That is $s(\mathbf{z}_k(\mathbf{y}), \mathbf{x}) = t_{\mathbf{z}_k(\mathbf{y})}(\mathbf{x})_+ + c\|\mathbf{z}_k(\mathbf{y}) - \mathbf{x}\|^2$. Here $t_{\mathbf{z}_k(\mathbf{y})}(\mathbf{x}) = F^{[1]}(\mathbf{z}_k(\mathbf{y}), \mathbf{x}) + g_{\mathbf{z}_k(\mathbf{y})}^\top(\mathbf{x} - \mathbf{z}_k(\mathbf{y}))$. Substituting this gives

$$\begin{aligned} g_k^{*\top}(\mathbf{y}-\mathbf{y}^k) + F_k^{[1]}(\mathbf{y}^k, \mathbf{x}) &\leq F^{[1]}(\mathbf{z}_k(\mathbf{y}), \mathbf{x}) + g_{\mathbf{z}_k(\mathbf{y})}^\top(\mathbf{y} - \mathbf{z}_k(\mathbf{y})) - s(\mathbf{z}_k(\mathbf{y}), \mathbf{x}) \\ &= F^{[1]}(\mathbf{z}_k(\mathbf{y}), \mathbf{x}) + g_{\mathbf{z}_k(\mathbf{y})}^\top(\mathbf{y} - \mathbf{z}_k(\mathbf{y})) \\ &\quad - \left[F^{[1]}(\mathbf{z}_k(\mathbf{y}), \mathbf{x}) - g_{\mathbf{z}_k(\mathbf{y})}^\top(\mathbf{x} - \mathbf{z}_k(\mathbf{y}))\right]_+ - c\|\mathbf{z}_k(\mathbf{y}) - \mathbf{x}\|^2. \end{aligned} \tag{3.22}$$

There are two cases to discuss, $[\ldots]_+ > 0$ and $[\ldots]_+ = 0$. Consider $[\ldots]_+ > 0$ first. Then

$$g_k^{*\top}(\mathbf{y}-\mathbf{y}^k) + F_k^{[1]}(\mathbf{y}^k, \mathbf{x}) \leq g_{\mathbf{z}_k(\mathbf{y})}^\top(\mathbf{y} - \mathbf{x}) - c\|\mathbf{z}_k(\mathbf{y}) - \mathbf{x}\|^2.$$

Due to boundedness of the g_k^* and of the set of trial steps we may pass to a subsequence $\mathcal{K}'$ of $\mathcal{K}$ where $g_k^* \to g^*$ and $\mathbf{z}_k(\mathbf{y}) \to \mathbf{z}(\mathbf{y})$ for some $\mathbf{z}(\mathbf{y})$. From part iv) we know $F_k(\mathbf{y}^k, \mathbf{x}) \to F(\mathbf{x}, \mathbf{x}) = 0$. Hence, passing to the limit in the above estimate gives

$$g^{*\top}(\mathbf{y}-\mathbf{x}) \leq g_{\mathbf{z}(\mathbf{y})}^\top(\mathbf{y} - \mathbf{x}) - c\|\mathbf{z}(\mathbf{y}) - \mathbf{x}\|^2 \leq g_{\mathbf{z}(\mathbf{y})}^\top(\mathbf{y} - \mathbf{x}). \tag{3.23}$$

One can see from this relation that $\mathbf{y} \to \mathbf{x}$ implies $\mathbf{z}(\mathbf{y}) \to \mathbf{x}$, because the $g_{\mathbf{z}(\mathbf{y})}$ are bounded. Using this information in (3.23), and writing $\mathbf{e}(\mathbf{y}) = (\mathbf{y}-\mathbf{x})/\|\mathbf{y}-\mathbf{x}\|$, we have

$$g^{*\top}\mathbf{e}(\mathbf{y}) \leq g_{\mathbf{z}(\mathbf{y})}^\top\mathbf{e}(\mathbf{y}).$$

Fixing an arbitrary unit vector $\mathbf{e}$, we arrange convergence $\mathbf{y} \to \mathbf{x}$ in such a way that $\mathbf{e}(\mathbf{y}) = (\mathbf{y}-\mathbf{x})/\|\mathbf{y}-\mathbf{x}\| \to \mathbf{e}$. Passing to a subsequence, we may in addition have $g_{\mathbf{z}(\mathbf{y})} \to g_{\mathbf{x}}$ for some $g_{\mathbf{x}}$, where $g_{\mathbf{x}} \in \partial_1 F^{[1]}(\mathbf{x},\mathbf{x})$ by upper semicontinuity of the Clarke subdifferential. That shows $g^{*\top}\mathbf{e} \leq \max\{g^\top \mathbf{e} : g \in \partial_1 F^{[1]}(\mathbf{x},\mathbf{x})\}$, and by the Hahn-Banach theorem we deduce $g^* \in \partial_1 F^{[1]}(\mathbf{x},\mathbf{x})$. That shows $\eta_k \leq \|g_k^* - g^*\| \to 0$ in the case $[\dots]_+ > 0$.

It remains to discuss the case where $[\dots]_+ = 0$. Going back to (3.22), we may again pass to the limit $k \in \mathcal{K}'$ such that $g_k^* \to g^*$ and $\mathbf{z}_k(\mathbf{y}) \to \mathbf{z}(\mathbf{y})$ to obtain

$$\begin{aligned} g^{*\top}(\mathbf{y}-\mathbf{x}) &\leq F^{[1]}(\mathbf{z}(\mathbf{y}),\mathbf{x}) + g_{\mathbf{z}(\mathbf{y})}^\top(\mathbf{y}-\mathbf{z}(\mathbf{y})) - c\|\mathbf{z}(\mathbf{y})-\mathbf{x}\|^2 \\ &= F^{[1]}(\mathbf{z}(\mathbf{y}),\mathbf{x}) + g_{\mathbf{z}(\mathbf{y})}^\top(\mathbf{x}-\mathbf{z}(\mathbf{y})) + g_{\mathbf{z}(\mathbf{y})}^\top(\mathbf{y}-\mathbf{x}) - c\|\mathbf{z}(\mathbf{y})-\mathbf{x}\|^2 \\ &\leq g_{\mathbf{z}(\mathbf{y})}^\top(\mathbf{y}-\mathbf{x}) - c\|\mathbf{z}(\mathbf{y})-\mathbf{x}\|^2 \qquad (\text{using } t_{\mathbf{z}(\mathbf{y})}(\mathbf{x}) \leq 0). \end{aligned}$$

This shows again that $\mathbf{z}(\mathbf{y}) \to \mathbf{x}$ when $\mathbf{y} \to \mathbf{x}$. Now the proof proceeds as above, and we deduce $g^* \in \partial F^{[1]}(\mathbf{x},\mathbf{x})$ in the case $[\dots]_+ = 0$, too. That ends the proof of $\eta_k \to 0$.

vi) Let $\eta := \operatorname{dist}(0, \partial_1 F^{[1]}(\mathbf{x},\mathbf{x}))$. We have to prove $\eta = 0$. Assume on the contrary that $\eta > 0$. Using the definition of η_k choose $\tilde{g}_k \in \partial_1 F^{[1]}(\mathbf{x},\mathbf{x})$ such that $\|g_k^* - \tilde{g}_k\| = \eta_k$. Then $\|\tilde{g}_k\| \geq \eta$, hence $\|g_k^*\| \geq \eta - \eta_k > (1-\zeta)\eta$ for k large enough, given that $\eta_k \to 0$ by v) and $\eta > 0$. (Here $\zeta \in (0,1)$ is the parameter chosen in part iii)). Going back with this to (3.21) gives

$$-F_k(\mathbf{y}^k,\mathbf{x}) \geq \tfrac{(1-\zeta)^2}{1+\zeta}\eta\|\mathbf{x}-\mathbf{y}^k\|. \tag{3.24}$$

vi) Choose $\epsilon > 0$ such that

$$\epsilon < \frac{\eta(\widetilde{\gamma}-\gamma)(1-\zeta)^2}{(1+\zeta)^2}. \tag{3.25}$$

We claim that there exists $k(\epsilon)$ such that $F(\mathbf{y}^k,\mathbf{x}) \leq F_{k+1}(\mathbf{y}^k,\mathbf{x}) + (1+\zeta)\epsilon\|\mathbf{x}-\mathbf{y}^k\|$ for all $k \in \mathcal{K}$, $k \geq k(\epsilon)$.

Indeed, let $m_k(\cdot,\mathbf{x})$ be the cutting plane at $\mathbf{y}^k$, $M_k(\cdot,\mathbf{x}) = m_k(\cdot,\mathbf{x}) + \frac{1}{2}(\cdot-\mathbf{x})^\top Q(\cdot-\mathbf{x})$. Then $F_{k+1}(\mathbf{y}^k,\mathbf{x}) = M_k(\mathbf{y}^k,\mathbf{x})$ by construction. Moreover, $m_k(\cdot,\mathbf{x}) = t_k(\cdot) - s_k$, where $t_k(\cdot)$ is the tangent to $F^{[1]}(\cdot,\mathbf{x})$ at $\mathbf{y}^k$, and s_k is the corresponding downshift (3.16). That means

$$\begin{aligned} m_k(\cdot,\mathbf{x}) &= F^{[1]}(\mathbf{y}^k,\mathbf{x}) + g_k^\top(\cdot-\mathbf{y}^k) - s_k \\ &= F^{[1]}(\mathbf{y}^k,\mathbf{x}) + g_k^\top(\cdot-\mathbf{y}^k) - c\|\mathbf{x}-\mathbf{y}^k\|^2 - [t_k(\mathbf{x})]_+ . \end{aligned}$$

There are two cases to discuss, $[\dots]_+ > 0$ and $[\dots]_+ = 0$. Assuming first $t_k(\mathbf{x}) > 0$, we have

$$F^{[1]}(\mathbf{y}^k,\mathbf{x}) - m_k(\mathbf{y}^k,\mathbf{x}) = F^{[1]}(\mathbf{y}^k,\mathbf{x}) - F^{[1]}(\mathbf{x},\mathbf{x}) - g_k^\top(\mathbf{y}^k-\mathbf{x}) + c\|\mathbf{x}-\mathbf{y}^k\|^2.$$

According to [37, Thm. 2] a lower C^1-function is approximately convex in the following sense. For a sequence $\mathbf{y}^k \to \mathbf{x}$ there exists $k(\epsilon)$ such that $g_k^\top(\mathbf{x}-\mathbf{y}^k) \leq F^{[1]}(\mathbf{x},\mathbf{x}) - F^{[1]}(\mathbf{y}^k,\mathbf{x}) + \epsilon\|\mathbf{x}-\mathbf{y}^k\|$ for all $k \geq k(\epsilon)$. Substituting this gives

$$F^{[1]}(\mathbf{y}^k,\mathbf{x}) - m_k(\mathbf{y}^k,\mathbf{x}) \leq \epsilon\|\mathbf{x}-\mathbf{y}^k\| + c\|\mathbf{x}-\mathbf{y}^k\|^2.$$

Re-arranging $F^{[1]}(\mathbf{y}^k,\mathbf{x}) - m_k(\mathbf{y}^k,\mathbf{x}) = \big(F^{[1]}(\mathbf{y}^k,\mathbf{x}) + (\mathbf{y}^k-\mathbf{x})^\top Q(\mathbf{y}^k-\mathbf{x})\big)$ $-\big(m_k(\mathbf{y}^k,\mathbf{x}) + (\mathbf{y}^k-\mathbf{x})^\top Q(\mathbf{y}^k-\mathbf{x})\big) = F(\mathbf{y}^k,\mathbf{x}) - M_k(\mathbf{y}^k,\mathbf{x})$, we have

$$F(\mathbf{y}^k,\mathbf{x}) - M_k(\mathbf{y}^k,\mathbf{x}) \leq \epsilon\|\mathbf{x}-\mathbf{y}^k\| + c\|\mathbf{x}-\mathbf{y}^k\|^2 \leq (1+\zeta)\epsilon\|\mathbf{x}-\mathbf{y}^k\| \qquad (3.26)$$

for k large enough. This ends the case $[\dots]_+ > 0$. Notice that in the second case $t_k(\mathbf{x}) \leq 0$ we get an even better estimate $F(\mathbf{y}^k,\mathbf{x}) - M_k(\mathbf{y}^k,\mathbf{x}) = c\|\mathbf{x}-\mathbf{y}^k\|^2 \leq \epsilon\|\mathbf{x}-\mathbf{y}^k\|$ for large k, so (3.26) holds in both cases.

vii) Using (3.24) and (3.26) we now expand the parameter $\widetilde{\rho}_k$ as

$$\begin{aligned}
\widetilde{\rho}_k &= \rho_k + \frac{F(\mathbf{y}^k,\mathbf{x}) - M_k(\mathbf{y}^k,\mathbf{x})}{F(\mathbf{x},\mathbf{x}) - F_k(\mathbf{y}^k,\mathbf{x})} \\
&\leq \rho_k + \frac{(1+\zeta)^2\epsilon\|\mathbf{x}-\mathbf{y}^k\|}{(1-\zeta)^2\eta\|\mathbf{x}-\mathbf{y}^k\|} = \rho_k + \frac{(1+\zeta)^2\epsilon}{(1-\zeta)^2\eta} \\
&< \rho_k + \widetilde{\gamma} - \gamma < \widetilde{\gamma}
\end{aligned}$$

using the choice (3.25) of ϵ and $\rho_k < \gamma$. But this contradicts $\widetilde{\rho}_k \geq \widetilde{\gamma}$ for the infinitely many $k \in \mathcal{K}$. Hence $\eta > 0$ was an incorrect hypothesis, and we have shown $\eta = 0$. This ends the proof. □

Lemma 3. *Under the hypotheses of the theorem, the inner loop at serious iterate* $\mathbf{x}$ *ends finitely.*

Proof: From the previous Lemma 2 we deduce that if the inner loop turns infinitely, then $\rho_k < \gamma$ and $\tau_k = \tau$ for $k \geq k_0$. By step 7 of the algorithm this implies $\widetilde{\rho}_k < \widetilde{\gamma}$ for all $k \geq k_0$, so that we are in the situation analyzed in [18, Lemma 6.3], and the conclusion is that we must have $0 \in \partial_1 F(\mathbf{x},\mathbf{x})$. As this contradicts step 2 of the algorithm, the inner loop must be finite. □

Proof of Theorem 1 :

i) We first prove $F(\mathbf{x}^{j+1},\mathbf{x}^j) \to 0$ $(j \to \infty)$, along with boundedness of the sequence $\mathbf{x}^j$. Notice that by construction, $F(\mathbf{x}^{j+1},\mathbf{x}^j) \leq 0$ for every j. There are two cases to discuss.

Case I : $c(\mathbf{x}^j) > 0$ for every $j \in \mathbb{N}$. Here the sequence of serious iterates never becomes feasible, and the algorithm remains in phase I. Here we expect to converge to a critical point of constraint violation. Notice that in case I we have

$$F(\mathbf{x}^{j+1},\mathbf{x}^j) = \max\{f(\mathbf{x}^{j+1}) - f(\mathbf{x}^j) - \mu c(\mathbf{x}^j), c(\mathbf{x}^{j+1}) - c(\mathbf{x}^j)\} \leq 0,$$

which shows $c(\mathbf{x}^j)$ is monotonically decreasing. Therefore $c(\mathbf{x}^j) \to c(\mathbf{x}^*)$ for every accumulation point $\mathbf{x}^*$ of the $\mathbf{x}^j$, and from $c(\mathbf{x}^{j+1}) - c(\mathbf{x}^j) \leq F(\mathbf{x}^{j+1}, \mathbf{x}^j) \leq 0$ we obtain $F(\mathbf{x}^{j+1}, \mathbf{x}^j) \to 0$. We use hypothesis (b) to deduce that the sequence $\mathbf{x}^j$ is bounded.

Case II : There exists $j_0 \in \mathbb{N}$ such that $c(\mathbf{x}^{j_0}) \leq 0$. Then from that index j_0 onward we have

$$F(\mathbf{x}^{j+1}, \mathbf{x}^j) = \max\{f(\mathbf{x}^{j+1}) - f(\mathbf{x}^j), c(\mathbf{x}^{j+1})\} \leq 0,$$

hence $f(\mathbf{x}^{j+1}) \leq f(\mathbf{x}^j)$ and $c(\mathbf{x}^{j+1}) \leq 0$. The iterates therefore stay feasible for $j \geq j_0$, and the objective f is optimized, so that we are in phase II. In particular, the sequence $f(\mathbf{x}^j)$, $j \geq j_0$, is monotonically decreasing. Therefore, for every accumulation point $\mathbf{x}^*$ of the $\mathbf{x}^j$, we have $f(\mathbf{x}^j) \to f(\mathbf{x}^*)$. Then $\liminf_{j\to\infty} F(\mathbf{x}^{j+1}, \mathbf{x}^j) \geq \lim_{j\to\infty} f(\mathbf{x}^{j+1}) - f(\mathbf{x}^j) = 0$ in tandem with $F(\mathbf{x}^{j+1}, \mathbf{x}^j) \leq 0$ proves $F(\mathbf{x}^{j+1}, \mathbf{x}^j) \to 0$. Here we use hypothesis (a) to deduce that the sequence $\mathbf{x}^j$ is bounded.

ii) Suppose in the j^{th} inner loop the serious step is accepted at inner loop counter k_j, that is, $\mathbf{x}^{j+1} = \mathbf{y}^{k_j}$. We show that $\tau_{k_j}\|\mathbf{x}^j - \mathbf{x}^{j+1}\|^2 \to 0$ and also $\|\mathbf{x}^j - \mathbf{x}^{j+1}\|_{Q_j+\tau_{k_j}I} \to 0$. To see this, observe that by the optimality condition (3.13) we have $g_j^* = (Q_j + \tau_{k_j} I)(\mathbf{x}^j - \mathbf{x}^{j+1}) \in \partial_1 F_{k_j}^{[1]}(\mathbf{x}^{j+1}, \mathbf{x}^j)$, hence by the subgradient inequality

$$(\mathbf{x}^j - \mathbf{x}^{j+1})^\top (Q_j + \tau_{k_j} I)(\mathbf{x}^j - \mathbf{x}^{j+1}) \leq F_{k_j}^{[1]}(\mathbf{x}^j, \mathbf{x}^j) - F_{k_j}^{[1]}(\mathbf{x}^{j+1}, \mathbf{x}^j) = -F_{k_j}^{[1]}(\mathbf{x}^{j+1}, \mathbf{x}^j).$$

Subtracting $F^{[2]}(\mathbf{x}^{j+1}, \mathbf{x}^j) = \frac{1}{2}(\mathbf{x}^j - \mathbf{x}^{j+1})^\top Q_j (\mathbf{x}^j - \mathbf{x}^{j+1})$ on both sides gives

$$\tfrac{1}{2}(\mathbf{x}^j - \mathbf{x}^{j+1})^\top Q_j (\mathbf{x}^j - \mathbf{x}^{j+1}) + \tau_{k_j}\|\mathbf{x}^j - \mathbf{x}^{j+1}\|^2 \leq -F_{k_j}(\mathbf{x}^{j+1}, \mathbf{x}^j).$$

Now by the acceptance test, $-F_{k_j}(\mathbf{x}^{j+1}, \mathbf{x}^j) \leq -\gamma^{-1} F(\mathbf{x}^{j+1}, \mathbf{x}^j)$, we have

$$\tfrac{1}{2}(\mathbf{x}^j - \mathbf{x}^{j+1})^\top Q_j (\mathbf{x}^j - \mathbf{x}^{j+1}) + \tau_{k_j}\|\mathbf{x}^j - \mathbf{x}^{j+1}\|^2 \leq -\gamma^{-1} F(\mathbf{x}^{j+1}, \mathbf{x}^j).$$

Next we use the fact that $Q_j + \tau_{k_j} I \succ 0$, which allows us to regroup the portion $\frac{1}{2}(\mathbf{x}^j - \mathbf{x}^{j+1})^\top Q_j (\mathbf{x}^j - \mathbf{x}^{j+1}) + \frac{1}{2}\tau_{k_j}\|\mathbf{x}^{j+1} - \mathbf{x}^j\|^2$ on the left into the norm $\frac{1}{2}\|\mathbf{x}^{j+1} - \mathbf{x}^j\|^2_{Q_j+\tau_{k_j}I}$, so that altogether the left hand side is the sum of two squared norms :

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

But the term on the right converges to 0 by part i), and this proves simultaneously $\tau_{k_j}\|\mathbf{x}^{j+1} - \mathbf{x}^j\|^2 \to 0$ and $\|\mathbf{x}^{j+1} - \mathbf{x}^j\|^2_{Q_j+\tau_{k_j}I} \to 0$, as claimed.

iii) Let $\mathbf{x}^*$ be an accumulation point of the sequence $\mathbf{x}^j$ of serious iterates. We have to prove $0 \in \partial_1 F(\mathbf{x}^*, \mathbf{x}^*)$. Select an infinite subsequence $J \subset \mathbb{N}$ such that $\mathbf{x}^j \to \mathbf{x}^*$, $j \in J$. Recall that $g_j^* = (Q_j + \tau_{k_j} I)(\mathbf{x}^j - \mathbf{x}^{j+1})$ is the aggregate subgradient belonging to $\mathbf{x}^{j+1}$ in the j^{th} inner loop. We distinguish two cases. *Case 1 :* There exists $\theta > 0$ such that $\|g_j^*\| \geq \theta > 0$ for all $j \in J$. *Case 2 :* There exists an infinite $J' \subset J$ such that $g_{j'}^* \to 0$, $j' \in J'$. Case 1 will be discussed in paragraphs iv) – vii). Case 2 is considered in part viii).

iv) We discuss the first case $\|g_j^*\| \geq \theta > 0$ for all $j \in J$. We first show that this working hypothesis implies $\tau_{k_j} \to \infty$ $(j \in J)$. Indeed, suppose there exists an infinite subset $J' \subset J$ such that the τ_{k_j}, $j \in J'$, are bounded. Then, using boundedness of the Q_j and of the set of serious steps proved in i), we could extract a subsequence $J'' \subset J'$ such that $Q_j \to \bar{Q}$, $\mathbf{x}^j - \mathbf{x}^{j+1} \to \delta\mathbf{x}$, $\tau_{k_j} \to \bar{\tau}$ and therefore $g_j^* \to (\bar{Q} + \bar{\tau}I)\delta\mathbf{x}$, where consequently $\|(\bar{Q} + \bar{\tau}I)\delta\mathbf{x}\| \geq \theta > 0$. But also $(\mathbf{x}^j - \mathbf{x}^{j+1})^\top (Q_j + \tau_{k_j} I)(\mathbf{x}^j - \mathbf{x}^{j+1}) \to \delta\mathbf{x}^\top (\bar{Q} + \bar{\tau}I)\delta\mathbf{x} = 0$ as a consequence of ii). Since $\bar{Q} + \bar{\tau}I$ is symmetric and $\succeq 0$, this contradicts $\|(\bar{Q} + \bar{\tau}I)\delta\mathbf{x}\| > 0$. Hence the τ_{k_j}, $j \in J'$ could not be bounded. This shows $\tau_{k_j} \to \infty$, $j \in J$.

So far we know that $\mathbf{x}^j \to \mathbf{x}^*$ and $\tau_{k_j} \to \infty$ $(j \in J)$. Now let J^+ be the set of those indices $j \in J$ where the τ-parameter was increased at least once during the j^{th} inner loop, J^- the other indices in J, where τ remained unchanged. In other words, in view of step 3 of the algorithm,

$$J^+ = \{j \in J : \tau_{k_j} > \tau_j^\sharp\}, \qquad J^- = \{j \in J : \tau_{k_j} = \tau_j^\sharp\}.$$

Then J^- must be finite. Indeed, $\tau_{k_j} \to \infty$, $(j \in J)$, but $\tau_j^\sharp \leq T < \infty$ according to step 8 of the algorithm.

v) Working on the set J^+, let us assume that the τ-parameter was increased for the last time at stage $k_j - \nu_j$, where $\nu_j \geq 1$. That is

$$\tau_{k_j} = \tau_{k_j - 1} = \cdots = \tau_{k_j - \nu_j + 1} = 2\tau_{k_j - \nu_j}.$$

According to step 7 of the algorithm, we have

$$\rho_{k_j - \nu_j} < \gamma, \qquad \widetilde{\rho}_{k_j - \nu_j} \geq \widetilde{\gamma}.$$

Since $\tau_{k_j - \nu_j} \to \infty$, $(j \in J^+)$, boundedness of the subgradients $\widetilde{g}_j = (Q_j + \frac{1}{2}\tau_{k_j} I)(\mathbf{x}^j - \mathbf{y}^{k_j - \nu_j})$ shows $\mathbf{y}^{k_j - \nu_j} - \mathbf{x}^j \to 0$. Here boundedness of the $\widetilde{g}_j$ can be seen as follows. By the subgradient inequality,

$$(\mathbf{x}^j - \mathbf{y}^{k_j - \nu_j})^\top (Q_j + \tfrac{1}{2}\tau_{k_j} I)(\mathbf{x}^j - \mathbf{y}^{k_j - \nu_j}) \leq$$

$$F_{k_j - \nu_j}^{[1]}(\mathbf{x}^j, \mathbf{x}^j) - F_{k_j - \nu_j}^{[1]}(\mathbf{y}^{k_j - \nu_j}, \mathbf{x}^j) = -F_{k_j - \nu_j}^{[1]}(\mathbf{y}^{k_j - \nu_j}, \mathbf{x}^j).$$

Now the exactness plane at $\mathbf{x}^j$ has the form $m_0(\cdot, \mathbf{x}^j) = g_{0j}^\top (\cdot - \mathbf{x}^j)$ for some $g_{0j} \in \partial_1 F^{[1]}(\mathbf{x}^j, \mathbf{x}^j)$, and we have $m_0(\cdot, \mathbf{x}^j) \leq F_{k_j - \nu_j}^{[1]}(\cdot, \mathbf{x}^j)$ by construction of the working model. Using this we have

$$(\mathbf{x}^j - \mathbf{y}^{k_j - \nu_j})^\top (Q_j + \tfrac{1}{2}\tau_{k_j} I)(\mathbf{x}^j - \mathbf{y}^{k_j - \nu_j}) \leq g_{0j}^\top (\mathbf{x}^j - \mathbf{y}^{k_j - \nu_j}) \leq \|g_{0j}\| \|\mathbf{x}^j - \mathbf{y}^{k_j - \nu_j}\|.$$

As $\tau_{k_j} \to \infty$ and the Q_j are bounded, the left hand side behaves asymptotically like $\tau_{k_j} \|\mathbf{x}^j - \mathbf{y}^{k_j - \nu_j}\|^2$. So after dividing one factor, we have $\tau_{k_j} \|\mathbf{x}^j - \mathbf{y}^{k_j - \nu_j}\| \leq C \|g_{0j}\|$ for some constant $C > 0$. Since the $\mathbf{x}^j$ are bounded, so are the g_{0j}, and we deduce boundedness of $\tau_{k_j}(\mathbf{x}^j - \mathbf{y}^{k_j - \nu_j})$. This shows boundedness of the $\widetilde{g}_j$ and also $\mathbf{x}^j - \mathbf{y}^{k_j - \nu_j} \to 0$ because of $\tau_{k_j} \to \infty$.

vi) As $\mathbf{x}^j \to \mathbf{x}^*$, part v) implies $\mathbf{y}^{k_j-\nu_j} \to \mathbf{x}^*$, $j \in J^+$. Passing to a subsequence, we may assume $\widetilde{g}_j \to \widetilde{g}$ for some $\widetilde{g}$. We show $\widetilde{g} \in \partial_1 F^{[1]}(\mathbf{x}^*, \mathbf{x}^*)$. From the subgradient inequality,

$$\widetilde{g}_j^\top \mathbf{h} \;\leq\; F^{[1]}_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}+\mathbf{h}, \mathbf{x}^j) - F^{[1]}_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j). \tag{3.27}$$

From $\widetilde{\rho}_{k_j-\nu_j} \geq \widetilde{\gamma}$ we obtain

$$-\widetilde{\gamma}^{-1} F_{k_j-\nu_j+1}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) \geq -F_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j).$$

Adding $\frac{1}{2}(\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j})^\top Q_j(\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j})$ on both sides gives

$$-\widetilde{\gamma}^{-1} F_{k_j-\nu_j+1}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) + \tfrac{1}{2}(\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j})^\top Q_j(\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}) \geq -F^{[1]}_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j).$$

Combining this with (3.27) gives

$$\begin{aligned}\widetilde{g}_j^\top \mathbf{h} \leq F^{[1]}_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}+\mathbf{h}, \mathbf{x}^j) - \widetilde{\gamma}^{-1} F_{k_j-\nu_j+1}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) + \\ \tfrac{1}{2}(\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j})^\top Q_j(\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}).\end{aligned} \tag{3.28}$$

Since $\mathbf{y}^{k_j-\nu_j} - \mathbf{x}^j \to 0$, the rightmost term converges to 0 by boundedness of the Q_j. We claim that the term $\widetilde{\gamma}^{-1} F_{k_j-\nu_j+1}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j)$ converges to $\widetilde{\gamma}^{-1} F(\mathbf{x}^*, \mathbf{x}^*) = 0$.

It suffices to show

$$F^{[1]}_{k_j-\nu_j+1}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) \to 0,$$

because we already know that $F^{[2]}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) = \frac{1}{2}(\mathbf{y}^{k_j-\nu_j} - \mathbf{x}^j)^\top Q_j(\mathbf{y}^{k_j-\nu_j} - \mathbf{x}^j)$ converges to 0. Now recall $F^{[1]}_{k_j-\nu_j+1}(\mathbf{y}^{k_j+\nu_j}, \mathbf{x}^j) = m_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j)$ for a cutting plane $m_{k_j-\nu_j}(\cdot, \mathbf{x}^j)$ at $\mathbf{y}^{k_j-\nu_j}$ with regard to serious iterate $\mathbf{x}^j$. That means we have $m_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) \leq F^{[1]}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) \to F^{[1]}(\mathbf{x}^*, \mathbf{x}^*) = 0$, because cutting planes are downshifted tangents. Hence $\limsup m_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) \leq 0$. It therefore suffices to show $\liminf m_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) \geq F^{[1]}(\mathbf{x}^*, \mathbf{x}^*) = 0$. Now $m_{k_j-\nu_j}(\cdot, \mathbf{x}^j) = t_{k_j-\nu_j}(\cdot) - s_j$, where $t_{k_j-\nu_j}$ is a tangent to $F^{[1]}(\cdot, \mathbf{x}^j)$ at $\mathbf{y}^{k_j-\nu_j}$, and $s_j \geq 0$ is the down-shift. Clearly $t_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}) = F^{[1]}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) \to F^{[1]}(\mathbf{x}^*, \mathbf{x}^*) = 0$ by joint continuity of F and the fact that the second order term also goes to 0, so we can concentrate on proving $s_j \to 0$. Now

$$s_j = \big[t_{k_j-\nu_j}(\mathbf{x}^j)\big]_+ + c\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\|^2 \to 0$$

because $t_{k_j-\nu_j}(\mathbf{x}^j) = F^{[1]}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) + \widetilde{g}_j^\top(\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}) \to 0$ by the argument just used. This proves our claim $F_{k_j-\nu_j+1}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) \to 0$.

Going back with this information to (3.28), passing to the limit gives $\widetilde{g}^\top \mathbf{h}$ on the left hand side and $\ell := \limsup F^{[1]}_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}+\mathbf{h}, \mathbf{x}^j)$ on the right, we have $\widetilde{g}^\top \mathbf{h} \leq \ell$, and we proceed to analyze the terms $F^{[1]}_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}+\mathbf{h}, \mathbf{x}^j)$ occurring on the right of (3.28).

Observe that $F^{[1]}_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}+\mathbf{h}, \mathbf{x}^j) = m_{\mathbf{z}_j(\mathbf{h})}(\mathbf{y}^{k_j-\nu_j}+\mathbf{h}, \mathbf{x}^j)$ for one of the cutting planes contributing to the buildup of $F^{[1]}_{k_j-\nu_j}(\cdot, \mathbf{x}^j)$. By construction, $m_{\mathbf{z}_j(\mathbf{h})}(\cdot, \mathbf{x}^j) =$

$t_{\mathbf{z}_j(\mathbf{h})}(\cdot) - s_j$, where $t_{\mathbf{z}_j(\mathbf{h})}(\cdot)$ is a tangent to $F^{[1]}(\cdot, \mathbf{x}^j)$ at a null step $\mathbf{z}_j(\mathbf{h})$, and s_j is the downshift is with regard to this tangent and serious iterate $\mathbf{x}^j$. The $\mathbf{z}_j(\mathbf{h})$ are among the previous null steps which form a bounded set. We may therefore extract a subsequence with $\mathbf{z}_j(\mathbf{h}) \to \mathbf{z}(\mathbf{h})$ for some $\mathbf{z}(\mathbf{h})$. The tangent is of the form $t_{\mathbf{z}_j(\mathbf{h})}(\cdot) = F^{[1]}(\mathbf{z}_j(\mathbf{h}), \mathbf{x}^j) + g_{\mathbf{z}_j(\mathbf{h})}^\top(\cdot - \mathbf{z}_j(\mathbf{h}))$, where $g_{\mathbf{z}_j(\mathbf{h})} \in \partial_1 F^{[1]}(\mathbf{z}_j(\mathbf{h}), \mathbf{x}^j)$. Passing to another subsequence, we may assume $g_{\mathbf{z}_j(\mathbf{h})} \to g_{\mathbf{z}(\mathbf{h})} \in \partial_1 F^{[1]}(\mathbf{z}(\mathbf{h}), \mathbf{x}^*)$ by upper semicontinuity of the Clarke subdifferential.

Next observe that for this subsequence the downshift also converges $s_j \to s^*$, where s^* is the downshift of tangent $t_{\mathbf{z}(\mathbf{h})}(\cdot)$ at $\mathbf{z}(\mathbf{h})$ with subgradient $g_{\mathbf{z}(\mathbf{h})}$ at serious step $\mathbf{x}^*$. That shows $t_{\mathbf{z}_j(\mathbf{h})}(\mathbf{y}^{k_j-\nu_j} + \mathbf{h}) - s_j \to t_{\mathbf{z}(\mathbf{h})}(\mathbf{x}^* + \mathbf{h}) - s^* = m_{\mathbf{z}(\mathbf{h})}(\mathbf{x}^* + \mathbf{h}, \mathbf{x}^*) = \ell$. As usual there are two cases for s^*.

First consider the case $s^* = \big[t_{\mathbf{z}(\mathbf{h})}(\mathbf{x}^*)\big]_+ + c\|\mathbf{x}^* - \mathbf{z}(\mathbf{h})\|^2 = t_{\mathbf{z}(\mathbf{h})}(\mathbf{x}^*) + c\|\mathbf{x}^* - \mathbf{z}(\mathbf{h})\|^2$. Then $\widetilde{g}^\top\mathbf{h} \le \ell = g_{\mathbf{z}(\mathbf{h})}^\top\mathbf{h} - c\|\mathbf{x}^* - \mathbf{z}(\mathbf{h})\|^2 \le g_{\mathbf{z}(\mathbf{h})}^\top\mathbf{h}$. This shows that if $\mathbf{h} \to 0$, then $\mathbf{z}(\mathbf{h}) \to \mathbf{x}^*$. Consequently, $g_{\mathbf{z}(\mathbf{h})} \to g_{\mathbf{x}^*}$ for some $g_{\mathbf{x}^*} \in \partial_1 F^{[1]}(\mathbf{x}^*, \mathbf{x}^*)$. Now fixing a unit vector $\mathbf{e}$, we can steer $\mathbf{h} \to 0$ in such a way that $\mathbf{h}/\|\mathbf{h}\| \to \mathbf{e}$. That implies $\widetilde{g}^\top\mathbf{e} \le g_{\mathbf{x}^*}^\top\mathbf{e} \le \max\{g^\top\mathbf{e} : g \in \partial_1 F^{[1]}(\mathbf{x}^*, \mathbf{x}^*)\}$. The expression on the right is the support function of $\partial_1 F^{[1]}(\mathbf{x}^*, \mathbf{x}^*)$, and by Hahn-Banach, $\widetilde{g} \in \partial_1 F^{[1]}(\mathbf{x}^*, \mathbf{x}^*)$.

Next consider the case $s^* = c\|\mathbf{x}^* - \mathbf{z}\|^2$. Then $\widetilde{g}^\top\mathbf{h} \le F^{[1]}(\mathbf{z}, \mathbf{x}^*) + g_{\mathbf{z}}^\top(\mathbf{x}^* + \mathbf{h} - \mathbf{z}) - c\|\mathbf{x}^* - \mathbf{z}\|^2 \le g_{\mathbf{z}}^\top\mathbf{h} - c\|\mathbf{x}^* - \mathbf{z}\|^2 \le g_{\mathbf{z}}^\top\mathbf{h}$ using $[t_{\mathbf{z}}(\mathbf{x}^*)]_+ = 0$. That gives the same estimate as before, so the conclusion in both cases is $\widetilde{g} \in \partial_1 F^{[1]}(\mathbf{x}^*, \mathbf{x}^*)$.

vii) Let $\eta := \operatorname{dist}(0, \partial_1 F^{[1]}(\mathbf{x}^*, \mathbf{x}^*))$. We have to prove $\eta = 0$. Assume on the contrary that $\eta > 0$. Then $\|\widetilde{g}\| \ge \eta > 0$ for $\widetilde{g}$ found in part vi). Fix $0 < \zeta < 1$. Using $\widetilde{g}_j \to \widetilde{g}$ we have $\|\widetilde{g}_j\| \ge (1-\zeta)\eta$ for j large enough. Now, assuming first that $[\dots]_+ > 0$, we have

$$\begin{aligned} m_{k_j-\nu_j}(\cdot, \mathbf{x}^j) &= F^{[1]}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) + \widetilde{g}_j^\top(\cdot - \mathbf{y}^{k_j-\nu_j}) - s_j \\ &= F^{[1]}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) + \widetilde{g}_j^\top(\cdot - \mathbf{y}^{k_j-\nu_j}) - t_{k_j-\nu_j}(\mathbf{x}^j) - c\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\|^2 \\ &= \widetilde{g}_j^\top(\cdot - \mathbf{x}^j) - c\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\|^2. \end{aligned} \tag{3.29}$$

Therefore

$$F^{[1]}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) - m_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) =$$
$$F^{[1]}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) - \widetilde{g}_j^\top(\mathbf{y}^{k_j-\nu_j} - \mathbf{x}^j) + c\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\|^2.$$

Now choose $\epsilon > 0$ such that

$$\epsilon < \frac{(1-\zeta)^2(\widetilde{\gamma} - \gamma)\eta}{(1+\zeta)^2}. \tag{3.30}$$

Since f and g are lower C^1, the $F(\cdot, \mathbf{x}^j)$ are uniformly ϵ-convex in the sense that there exists $j(\epsilon)$ such that $\widetilde{g}_j^\top(\mathbf{y}^{k_j-\nu_j} - \mathbf{x}^j) \le F^{[1]}(\mathbf{x}^j, \mathbf{x}^j) - F^{[1]}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) + \epsilon\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\|$ for all $j \ge j(\epsilon)$, cf. [**?**, Thm. 2]. Substituting this in (3.29) at $\mathbf{y}^{k_j-\nu_j}$ gives

$$\begin{aligned} F^{[1]}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) - m_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) &\le \epsilon\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\| + c\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\|^2 \\ &\le (1+\zeta)\epsilon\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\| \end{aligned} \tag{3.31}$$

for j large enough. The case $[\dots]_+ = 0$ in (3.29) leads to the even stronger estimate $F^{[1]}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) - m_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) = c\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\|^2$, so we may continue with (3.31). Now, recall that $\widetilde{g}_j = (Q_j + \frac{1}{2}\tau_{k_j} I)(\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}) \in \partial_1 F^{[1]}_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j)$ gives

$$\widetilde{g}_j^\top(\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}) \le F^{[1]}_{k_j-\nu_j}(\mathbf{x}^j, \mathbf{x}^j) - F^{[1]}_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) = -F^{[1]}_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j).$$

Subtracting a quadratic term from both sides, we get

$$\tfrac{1}{2}(\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j})^\top Q_j(\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}) + \tfrac{1}{2}\tau_{k_j}\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\|^2 \le -F_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j).$$

As $\tau_{k_j} \to \infty$, we have

$$(1-\zeta)\tfrac{1}{2}\tau_{k_j}\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\| \le \|\widetilde{g}_j\| \le (1+\zeta)\tfrac{1}{2}\tau_{k_j}\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\|$$

and also

$$\tfrac{1}{2}(\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j})^\top Q_j(\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}) + \tfrac{1}{2}\tau_{k_j}\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\|^2 \ge (1-\zeta)\tfrac{1}{2}\tau_{k_j}\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\|^2.$$

Combining these gives

$$-F_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) \ge \frac{(1-\zeta)^2}{1+\zeta}\eta\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\|. \tag{3.32}$$

Combining (3.31) and (3.32) leads to

$$\begin{aligned}
\widetilde{\rho}_{k_j-\nu_j} &= \rho_{k_j-\nu_j} + \frac{F^{[1]}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j) - m_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j)}{-F_{k_j-\nu_j}(\mathbf{y}^{k_j-\nu_j}, \mathbf{x}^j)} \\
&\le \rho_{k_j-\nu_j} + \frac{(1+\zeta)^2\epsilon\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\|}{(1-\zeta)^2\eta\|\mathbf{x}^j - \mathbf{y}^{k_j-\nu_j}\|} \qquad \text{use (3.31) and (3.32)} \\
&\le \rho_{k_j-\nu_j} + \widetilde{\gamma} - \gamma < \widetilde{\gamma}, \qquad \text{use (3.30)}
\end{aligned}$$

contradicting $\widetilde{\rho}_{k_j-\nu_j} \ge \widetilde{\gamma}$ for the infinitely many $j \in J^+$. This shows that the hypothesis $\eta > 0$ was incorrect, hence $\eta = 0$, which ends the convergence proof in the case started in part iv).

viii) It remains to deal with the case $g_j^* \to 0$, $j \in J'$. Since g_j^* is a subgradient of $F_{k_j}(\cdot, \mathbf{x}^j)$ at $\mathbf{x}^{j+1}$, the subgradient inequality gives for any test vector $\mathbf{h}'$:

$$\begin{aligned}
g_j^{*\top}\mathbf{h}' &\le F^{[1]}_{k_j}(\mathbf{x}^{j+1} + \mathbf{h}', \mathbf{x}^j) - F^{[1]}_{k_j}(\mathbf{x}^{j+1}, \mathbf{x}^j) \\
&= F^{[1]}_{k_j}(\mathbf{x}^{j+1} + \mathbf{h}', \mathbf{x}^j) - F_{k_j}(\mathbf{x}^{j+1}, \mathbf{x}^j) + \tfrac{1}{2}(\mathbf{x}^j - \mathbf{x}^{j+1})^\top Q_j(\mathbf{x}^j - \mathbf{x}^{j+1}) \\
&= F^{[1]}_{k_j}(\mathbf{x}^{j+1} + \mathbf{h}', \mathbf{x}^j) - F_{k_j}(\mathbf{x}^{j+1}, \mathbf{x}^j) + \tfrac{1}{2}\|\mathbf{x}^j - \mathbf{x}^{j+1}\|^2_{Q_j+\tau_{k_j}I} - \tfrac{1}{2}\tau_{k_j}\|\mathbf{x}^j - \mathbf{x}^{j+1}\|^2 \\
&\le F^{[1]}_{k_j}(\mathbf{x}^{j+1} + \mathbf{h}', \mathbf{x}^j) - F_{k_j}(\mathbf{x}^{j+1}, \mathbf{x}^j) + \tfrac{1}{2}\|\mathbf{x}^j - \mathbf{x}^{j+1}\|^2_{Q_j+\tau_{k_j}I} \\
&\le F^{[1]}_{k_j}(\mathbf{x}^{j+1} + \mathbf{h}', \mathbf{x}^j) - \gamma^{-1}F(\mathbf{x}^{j+1}, \mathbf{x}^j) + \tfrac{1}{2}\|\mathbf{x}^j - \mathbf{x}^{j+1}\|^2_{Q_j+\tau_{k_j}I}.
\end{aligned}$$

Fixing another test vector $\mathbf{h}$, we put $\mathbf{h}' = \mathbf{x}^j - \mathbf{x}^{j+1} + \mathbf{h}$ and substitute it to obtain

$$\tfrac{1}{2}\|\mathbf{x}^j - \mathbf{x}^{j+1}\|^2_{Q_j+\tau_{k_j}I} + g_j^{*\top}\mathbf{h} \le F^{[1]}_{k_j}(\mathbf{x}^j + \mathbf{h}, \mathbf{x}^j) - \gamma^{-1}F(\mathbf{x}^{j+1}, \mathbf{x}^j).$$

Since $g_j^* \to 0$ by hypothesis, and $\|\mathbf{x}^j - \mathbf{x}^{j+1}\|_{Q_j+\tau_{k_j}I} \to 0$, $F(\mathbf{x}^{j+1}, \mathbf{x}^j) \to 0$ by part i), and we may therefore condense the above to

$$\epsilon_j \leq F_{k_j}^{[1]}(\mathbf{x}^j + \mathbf{h}, \mathbf{x}^j)$$

for every test vector $\mathbf{h}$, where $\epsilon_j = \frac{1}{2}\|\mathbf{x}^j - \mathbf{x}^{j+1}\|^2_{Q_j+\tau_{k_j}I} + g_j^{*\top}\mathbf{h} + \gamma^{-1}F(\mathbf{x}^{j+1}, \mathbf{x}^j) \to 0$.

Now recall that in the j^{th} inner loop $F_{k_j}^{[1]}(\cdot, \mathbf{x}^j)$ is constructed as a maximum of cutting planes, so there exists a null step $\mathbf{z}_j(\mathbf{h}) \in \{\mathbf{y}^1, \ldots, \mathbf{y}^{k_j-1}\}$ such that $F_{k_j}^{[1]}(\mathbf{x}^j + \mathbf{h}, \mathbf{x}^j) = m_{\mathbf{z}_j(\mathbf{h})}(\mathbf{x}^j + \mathbf{h}, \mathbf{x}^j)$ for the cutting plane at trial $\mathbf{z}_j(\mathbf{h})$ for serious iterate $\mathbf{x}^j$. Next recall that $m_{\mathbf{z}_j(\mathbf{h})}(\cdot, \mathbf{x}^j) = t_{\mathbf{z}_j(\mathbf{h})}(\cdot) - s_j$, where $t_{\mathbf{z}_j(\mathbf{h})}$ is a tangent to $F^{[1]}(\cdot, \mathbf{x}^j)$ at $\mathbf{z}_j(\mathbf{h})$, and s_j is the corresponding downshift. Since $t_{\mathbf{z}_j(\mathbf{h})}(\cdot) = F^{[1]}(\mathbf{z}_j(\mathbf{h}), \mathbf{x}^j) + g_{\mathbf{z}_j(\mathbf{h})}^\top(\cdot - \mathbf{z}_j(\mathbf{h}))$ for some $g_{\mathbf{z}_j(\mathbf{h})} \in \partial_1 F^{[1]}(\mathbf{z}_j(\mathbf{h}), \mathbf{x}^j)$, we have

$$\epsilon_j \leq F_{k_j}^{[1]}(\mathbf{x}^j + \mathbf{h}, \mathbf{x}^j) = F^{[1]}(\mathbf{z}_j(\mathbf{h}), \mathbf{x}^j) + g_{\mathbf{z}_j(\mathbf{h})}^\top(\mathbf{x}^j + \mathbf{h} - \mathbf{z}_j(\mathbf{h})) - s_j. \tag{3.33}$$

Here we have to discuss the two cases $s_j = t_{\mathbf{z}_j(\mathbf{h})} + c\|\mathbf{z}_j(\mathbf{h}) - \mathbf{x}^j\|^2$ and $s_j = c\|\mathbf{z}_j(\mathbf{h}) - \mathbf{x}^j\|^2$.

Starting with the first case, as the set of all trial steps visited during the run of the algorithm is bounded, we may extract a subsequence of J such that $\mathbf{z}_j(\mathbf{h}) \to \mathbf{z}(\mathbf{h})$ and $g_{\mathbf{z}_j(\mathbf{h})} \to g_{\mathbf{z}(\mathbf{h})}$. As $\mathbf{x}^j \to \mathbf{x}^*$, upper semi-continuity of the Clarke subdifferential gives $g_{\mathbf{z}(\mathbf{h})} \in \partial_1 F^{[1]}(\mathbf{z}(\mathbf{h}), \mathbf{x}^*)$. Moreover, as the downshift procedure is continuous in the data used, $s_j \to s$, where s is the downshift for tangent $t_{\mathbf{z}(\mathbf{h})}(\cdot)$ to $F^{[1]}(\cdot, \mathbf{x}^*)$ at $\mathbf{z}(\mathbf{h})$. In other words, $m_{\mathbf{z}(\mathbf{h})}(\cdot, \mathbf{x}^*) = t_{\mathbf{z}(\mathbf{h})}(\cdot) - s$ is the cutting plane which our method would compute at null step $\mathbf{z}$ for serious iterate $\mathbf{x}^*$ if the corresponding tangent used the subgradient $g_{\mathbf{z}(\mathbf{h})}$. Altogether, this implies

$$0 \leq F^{[1]}(\mathbf{z}(\mathbf{h}), \mathbf{x}^*) + g_{\mathbf{z}(\mathbf{h})}^\top(\mathbf{x}^* + \mathbf{h} - \mathbf{z}(\mathbf{h})) - s,$$

where $s = t_{\mathbf{z}(\mathbf{h})}(\mathbf{x}^*) + c\|\mathbf{z}(\mathbf{h}) - \mathbf{x}^*\|^2$. We obtain $0 \leq$

$$F^{[1]}(\mathbf{z}(\mathbf{h}), \mathbf{x}^*) + g_{\mathbf{z}(\mathbf{h})}^\top(\mathbf{x}^* + \mathbf{h} - \mathbf{z}(\mathbf{h})) - F^{[1]}(\mathbf{z}(\mathbf{h}), \mathbf{x}^*) - g_{\mathbf{z}(\mathbf{h})}^\top(\mathbf{x}^* - \mathbf{z}(\mathbf{h})) - c\|\mathbf{z}(\mathbf{h}) - \mathbf{x}^*\|^2,$$

which can be re-arranged as

$$0 \leq c\|\mathbf{z}(\mathbf{h}) - \mathbf{x}^*\|^2 \leq g_{\mathbf{z}(\mathbf{h})}^\top \mathbf{h}. \tag{3.34}$$

Since the set of all possible $g_{\mathbf{z}(\mathbf{h})}$ is bounded, the estimate shows that $\mathbf{z}(\mathbf{h}) \to \mathbf{x}^*$ when $\mathbf{h} \to \mathbf{0}$. Dividing by $\|\mathbf{h}\|$, we now have

$$0 \leq g_{\mathbf{z}(\mathbf{h})}^\top \frac{\mathbf{h}}{\|\mathbf{h}\|}.$$

Now fix a unit vector $\mathbf{e}$ and let $\mathbf{h} \to \mathbf{0}$ in such a way that $\mathbf{h}/\|\mathbf{h}\| \to \mathbf{e}$. From the previous we know that $\mathbf{z}(\mathbf{h}) \to \mathbf{x}^*$. Therefore, using the upper semi-continuity of the Clarke subdifferential, we may extract a subsequence such that $g_{\mathbf{z}(\mathbf{h})} \to g_{\mathbf{x}^*}$ for some $g_{\mathbf{x}^*} \in$

$\partial_1 F^{[1]}(\mathbf{x}^*, \mathbf{x}^*)$. We have therefore shown $0 \leq g_{\mathbf{x}^*}^\top \mathbf{e} \leq \max\{g^\top \mathbf{e} : g \in \partial_1 F^{[1]}(\mathbf{x}^*, \mathbf{x}^*)\}$. But the expression on the right is the Clarke directional derivative of $F^{[1]}(\cdot, \mathbf{x}^*)$ at $\mathbf{x}^*$ in direction $\mathbf{e}$. As $\mathbf{e}$ was arbitrary, we have shown that the Clarke directional derivative of $F^{[1]}(\cdot, \mathbf{x}^*)$ is non-negative in every direction, and this implies $0 \in \partial_1 F^{[1]}(\mathbf{x}^*, \mathbf{x}^*)$. This ends the proof in the case $[\dots]_+ > 0$.

It remains to discuss the case $[\dots]_+ = 0$. Going back to estimate (3.33), we observe that the downshift is $s_j = c\|\mathbf{z}_j(\mathbf{h}) - \mathbf{x}^j\|^2$. As before, $F_{k_j}^{[1]}(\mathbf{x}^j + \mathbf{h}, \mathbf{x}^j) = m_{\mathbf{z}_j(\mathbf{h})}(\mathbf{x}^j + \mathbf{h}, \mathbf{x}^j)$, and we now represent the cutting plane as $m_{\mathbf{z}_j(\mathbf{h})}(\cdot, \mathbf{x}^j) = m_{\mathbf{z}_j(\mathbf{h})}(\mathbf{x}^j, \mathbf{x}^j) + g_{\mathbf{z}_j(\mathbf{h})}^\top(\cdot - \mathbf{x}^j)$ for the same $g_{\mathbf{z}_j(\mathbf{h})} \in \partial_1 F^{[1]}(\mathbf{z}_j(\mathbf{h}), \mathbf{x}^j)$. Now as the tangent at $\mathbf{x}^j$ satisfies $t_{\mathbf{z}_j(\mathbf{h})}(\mathbf{x}^j) \leq 0$, we have $m_{\mathbf{z}_j(\mathbf{h})}(\mathbf{x}^j, \mathbf{x}^j) \leq -c\|\mathbf{z}_j(\mathbf{h}) - \mathbf{x}^j\|^2$. Therefore

$$\epsilon_j \leq F_{k_j}^{[1]}(\mathbf{x}^j + \mathbf{h}, \mathbf{x}^j) \leq -c\|\mathbf{x}^j - \mathbf{z}_j(\mathbf{h})\|^2 + g_{\mathbf{z}_j(\mathbf{h})}^\top \mathbf{h}.$$

Passing to the limits $\epsilon_j \to 0$, $\mathbf{x}^j \to \mathbf{x}^*$, $\mathbf{z}_j(\mathbf{h}) \to \mathbf{z}(\mathbf{h})$, $g_{\mathbf{z}_j(\mathbf{h})} \to g_{\mathbf{z}(\mathbf{h})}$ as in the previous case, we get $0 \leq -c\|\mathbf{z}(\mathbf{h}) - \mathbf{x}^*\|^2 + g_{\mathbf{z}(\mathbf{h})}^\top \mathbf{h}$. But now we are back in the situation (3.34), and the conclusion is the same. This ends the proof in case $[\dots]_+ = 0$. □

3.5. Application to flight control

In this section we switch back from the abstract optimization program to (3.5), discussing the elements needed to apply our algorithm.

3.5.1. The banded H_∞-norm

We start by discussing the banded H_∞-norm $f(\mathbf{x})$ in (3.6). The first observation is that f is lower C^1. We have the even stronger

Lemma 4. *Let f be a squared H_∞-norm (3.6) on a closed frequency band Ω. Then f is lower C^2 on the open set $S = \{\mathbf{x} \in \mathbb{R}^n : T_{w\to z}(\mathbf{x}, \cdot)$ is internally stable$\}$.*

Proof: The mapping $\mathcal{F} : \mathbb{R}^n \times \mathbb{S}^1 \to \mathbb{H}$ defined by (3.7) is of class C^2 in $\mathbf{x}$ and analytic in s for $\mathbf{x} \in S$. Indeed, the closed-loop matrices $A_{\rm cl} = A + BKC$, and similarly $B_{\rm cl}$, $C_{\rm cl}$, $D_{\rm cl}$, are affine functions of K, so that $\mathcal{F}(K, s)$ depends rationally on K and s. By construction (3.2), (4.3), the controller $K = K(\mathbf{x})$ depends rationally on $\mathbf{x}$, hence $\mathcal{F}(\mathbf{x}, s)$ depends rationally on $\mathbf{x}, s$. Since matrix inversion is allowed for $\mathbf{x} \in S$, the claim follows.

Writing the maximum eigenvalue as

$$\lambda_1(X) = \max\{Z \bullet X : Z \succeq 0, \mathrm{Trace}(Z) = 1\},$$

we have

$$f(\mathbf{x}) = \max_{\omega\in\Omega} f(\mathbf{x},\omega) = \max_{\omega\in\Omega} \max_{Z\succeq 0,\mathrm{Tr}(Z)=1} Z \bullet \mathcal{F}(\mathbf{x},\omega),$$

which is a representation of the form (3.19) with $(Z,\omega) \mapsto Z \bullet \mathcal{F}(\mathbf{x},\omega)$ of class C^2. The compact space is $K = \{Z \in \mathbb{H} : Z \succeq 0, \mathrm{Trace}(Z) = 1\} \times \Omega$. □

Computation of the H_∞-norm is based on the algorithm of Boyd *et al.* [19]. Computation of Clarke subgradients $g \in \partial f(\mathbf{x})$ was discussed in [10]. Notice that the peak frequencies $\Omega(\mathbf{x}) = \{\omega \in \Omega : f(\mathbf{x}) = f(\mathbf{x},\omega)\}$, obtained along with the function value $f(\mathbf{x})$, are needed to compute subgradients. Recall that the set $\Omega(\mathbf{x})$ of peak frequencies has a very special structure. We have

Lemma 5. *(Compare [19], [38, Lemma 1]). The set $\Omega(\mathbf{x})$ is either finite, or $\Omega(\mathbf{x}) = \Omega$.*

If $\mathbf{y}^k$ is a null step at serious step $\mathbf{x}$, then it is reasonable to enrich the working model $F_k(\cdot,\mathbf{x})$ by adding several cutting planes or near cutting planes of objective f and constraint c simultaneously. This may be done by building a finite set $\Omega_e(\mathbf{y}^k)$ of near active frequencies at $\mathbf{y}^k$, i.e., frequencies ω satisfying $f(\mathbf{y}^k) - \theta \le f(\mathbf{y}^k,\omega) < f(\mathbf{y}^k)$ for some threshold $\theta > 0$, and computing tangents to $f(\cdot,\omega)$ at $\mathbf{y}^k$. By Lemma 5 we assure that $\Omega_e(\mathbf{y}^k) \supset \Omega(\mathbf{y}^k)$ when $\Omega(\mathbf{y}^k)$ is finite, which it always is in practice. Similarly for tangents arising from the constraint c. These near tangents to F are then downshifted with respect to the current value $F(\mathbf{x},\mathbf{x}) = 0$ just as the regular tangent. Ways to select an extended set of frequencies $\Omega_e(\mathbf{y}^k)$ containing $\Omega(\mathbf{y}^k)$ are given in [10]. It is for instance wise to include the finitely many secondary peaks, that is, the local maxima of the curve $\omega \mapsto f(\mathbf{y}^k,\omega)$, because secondary peaks are candidates to become active at the next iteration. Ways to compute those are for instance given in [38].

3.5.2. Internal stability

The last issue we have to discuss before applying our algorithm to (3.5) concerns the hidden constraint $\mathbf{x} \in S = \{\mathbf{x} \in \mathbb{R}^n : T_i(\mathbf{x},\cdot), i = 1,\dots,6$ are internally stable$\}$, which is not dealt with explicitly in (3.8). Notice that S is an open set, so $\mathbf{x} \in S$ is not a constraint in the usual sense of optimization. The closed-loop channels T_i in (3.5) are obtained by substituting controllers $K^{(1)}$, $K^{(2)}$ into the corresponding plants (3.4), which provides closed-loop system matrices $A_i(\mathbf{x})$ whose stability we have to guarantee. Using the spectral abscissa $\alpha(A) = \max\{\mathrm{Re}(\lambda) : \lambda$ eigenvalue of $A\}$, we can replace internal stability by the inequality constraint

$$\max_{i=1,\dots,6} \alpha\left(A_i(\mathbf{x})\right) \le -\epsilon \tag{3.35}$$

for some small $\epsilon > 0$. In order to maintain stability of the iterates, we add the constraint (3.35) to program (3.5).

Notice that in our application the open-loop system is stable, and it is not too hard to tune the three blocks autopilot, flight controller, low-pass filter independently to find a stabilizing choice of parameters $\mathbf{x}_1$. In other situations, it may be necessary to compute an initial stabilizing iterate $\mathbf{x}_1$ satisfying (3.35) by solving an optimization program. Here one may use the method of Burke *et al.* [39], which consists in optimizing

$$\min_{\mathbf{x}\in\mathbb{R}^n} \max_{i=1,\dots,6} \alpha\left(A_i(\mathbf{x})\right) \tag{3.36}$$

using a descent method until $\mathbf{x}_1$ satisfying (3.35) is found.

Remark 8. As a rule it is easy to find a stabilizing controller for practical systems, those being designed to work correctly. However, from a purely mathematical point of view, *deciding* whether or not a stabilizing structured controller exists is NP-complete for most practical controller structures [40]. That means if one fails to find a stabilizing controller e.g. with program (3.36), or by using specific knowledge about the given application, then a proof that no stabilizing controller of the given structure exists will take exponential time (in the system order), and will therefore be difficult or even impossible to obtain.

3.5.3. Numerical results

In this section we present numerical tests obtained with our algorithm. In a first phase an initial stabilizing controller $\mathbf{x}_1 = [-0.1, -0.15, -1.0, 5\sqrt{2}, 25, -5.0, -0.05, -0.0035, 0]$ is found by a traditional design, where each of the blocks (PI, P, filter) is tuned manually and independently. The corresponding closed-loop channels are shown in blue in Figures 3.7, 3.8, 3.9. The six H_∞-norms involved are $\|T\|_\infty := (\|T_1\|_\infty, \dots, \|T_6\|_\infty)$ with $\|T\|_\infty = [1.0336e+00, 2.2775e+00, 3.0700e+00, 3.0359e-01, 1.1345e+00, 3.8147e+00]$, which means $f(\mathbf{x}_1) = 3.07^2, \quad c(\mathbf{x}_1) = 3.8147^2$. The algorithm is now run with the constraint $c(\mathbf{x}) = \max_{i=5,6} \|W_i^{-1} T_i(\mathbf{x}, \cdot)\|_\infty^2 - r^2 \leq 0$ with $r = 1.08$. We used the following two-stage stopping test. If the inner loop at $\mathbf{x}^j$ finds a serious iterate $\mathbf{x}^{j+1}$ satisfying

$$\frac{\|\mathbf{x}^j - \mathbf{x}^{j+1}\|}{1 + \|\mathbf{x}^j\|} < \mathrm{tol}, \tag{3.37}$$

then $\mathbf{x}^{j+1}$ is accepted as the final solution. On the other hand, if the inner loop is unable to find a serious step and provides three consecutive unsuccessful trial steps $\mathbf{y}^k$ satisfying

$$\frac{\|\mathbf{x}^j - \mathbf{y}^k\|}{1 + \|\mathbf{x}^j\|} < \mathrm{tol}, \tag{3.38}$$

or if a maximum number of 20 allowed steps k in the inner loop is reached, then we decide that $\mathbf{x}^j$ is already optimal. The second stopping criterion (3.38) is rarely invoked in our experiments. Both tests are based on the observation that $0 \in \partial_1 F(\mathbf{x}^j, \mathbf{x}^j)$ if and only if $\mathbf{y}^k = \mathbf{x}^j$ is solution of the tangent program (3.3.3), and on Lemmas 2, 3.

In our flight control example we use tol$= 2.0 \cdot 10^{-4}$, which induces the algorithm to stop based on (3.37) after 72 iterations within 379 seconds CPU. The relative progress of function and constraint at that stage are

$$|f(\mathbf{x}^{j+1})-f(\mathbf{x}^j)|/(1+|f(\mathbf{x}^j)|) = 1.3\cdot 10^{-5},\ |c(\mathbf{x}^{j+1})-c(\mathbf{x}^j)|/(1+|c(\mathbf{x}^j)|) = 6.9\cdot 10^{-5}.$$

The optimal controller was

$$\mathbf{x}^* = [-0.0937, -0.108, -0.648, 10.743, 34.335, -10.968, -0.218, -0.142, 0.0258]$$

with

$$\|T\|_\infty = [1.0181, 1.0890, 1.0257, 1.0890, 1.0273, 1.0800]$$

meaning $f(\mathbf{x}^*) = 1.089^2, c(\mathbf{x}^*) = 1.08^2$. In particular, the constraint is active, as it should be. The performance and robustness curves of $\mathbf{x}^*$ are shown in red in Figures 3.7 – 3.9. Time domain responses of $\mathbf{x}^*$ are shown in Figure 3.10.

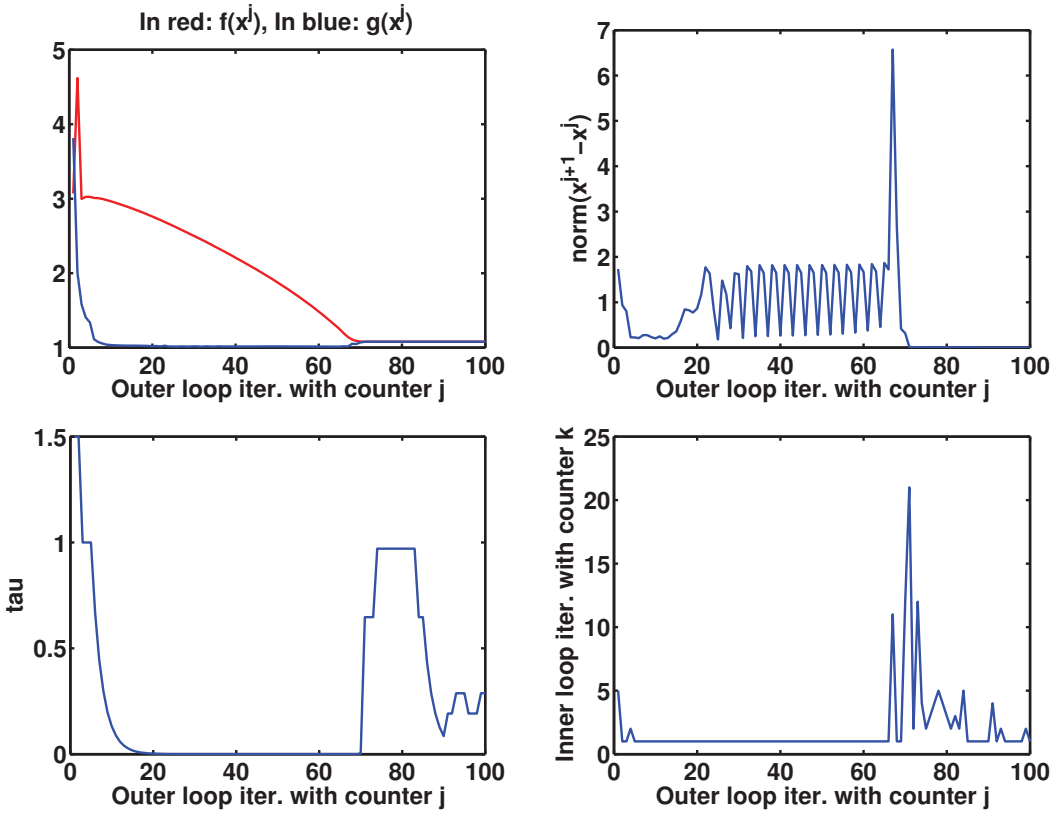


FIGURE 3.6.: Bearing of the algorithm. Top left shows $j \mapsto f(\mathbf{x}^j)$ (red) and $j \mapsto c(\mathbf{x}^j)$ (blue). Top right $j \mapsto \|x^{j+1} - x^j\|$ shows length of accepted serious step. Lower left shows $j \mapsto k_j$, the number of iterates of the inner loop. Lower right shows $j \mapsto \tau_j^\sharp$, the τ-parameter at serious steps. From iteration 72 onward progress is slight, the inner loop takes more time to find serious steps, and τ behaves more irregularly.

For the purpose of testing we considered smaller values of the tolerance tol in order to see how many iterations the algorithm needs to reach this precision. For instance, tol $= 1.12\cdot 10^{-4}$ leads already to 100 iterations, reached in 713 seconds CPU, tol $= 1.1\cdot 10^{-4}$

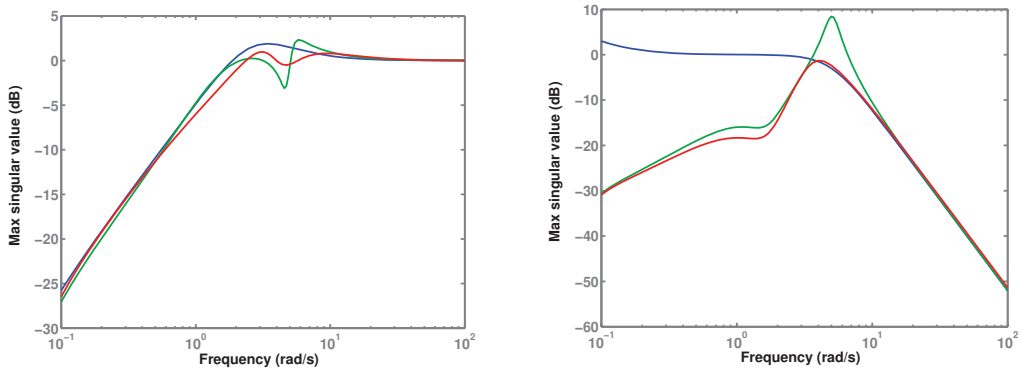


FIGURE 3.7.: Criteria for flight controller. Performance channel $T_{N_z \to dN_z}$ on the left assures good tracking of vertical load factor in the range $[10^{-1}, 10^{0}]$. Robustness channel $T_{n_q \to dm}$ on the right limits influence of noise on elevator deflection in the range $> 10^{1}$. Blue is template, green initial guess, red optimized. Both criteria are not relevant for frequencies below 10^{-1}.

leads to 119, tol $= 1.09 \cdot 10^{-4}$ to 138, tol $= 1.06 \cdot 10^{-4}$ to 169 iterations, highlighting the well-known fact that stopping is a delicate problem in non-smooth methods.

Figure 3.6 displays typical parameters of the algorithm during the first 100 iterations. From iteration 73 onwards the algorithm essentially stagnates, which leads to an increase in τ and k_j. Steplength at that stage becomes small, and progress is slight.

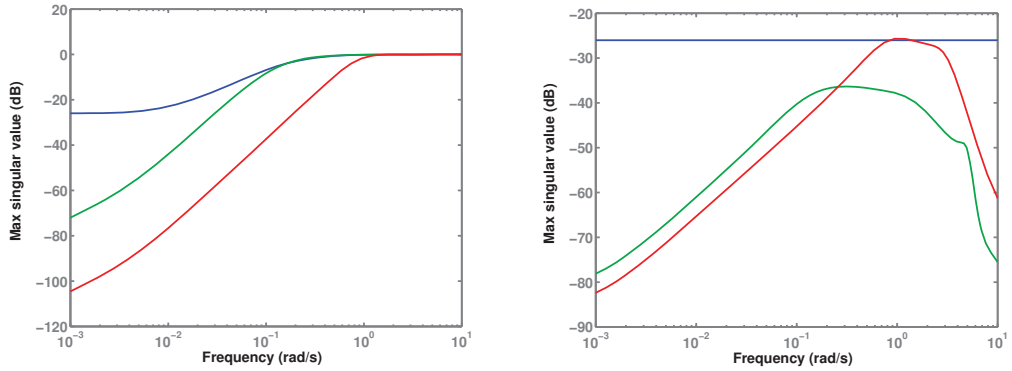


FIGURE 3.8.: Performance channels for autopilot. Velocity tracking error $T_{V \to dV}$ left and climb angle (slope) tracking error $T_{\gamma \to d\gamma}$ right are kept small for frequencies below 10^{-1}. Blue template, green before optimization, red after optimization.

The final experiment consists in inspecting step responses in closed loop.

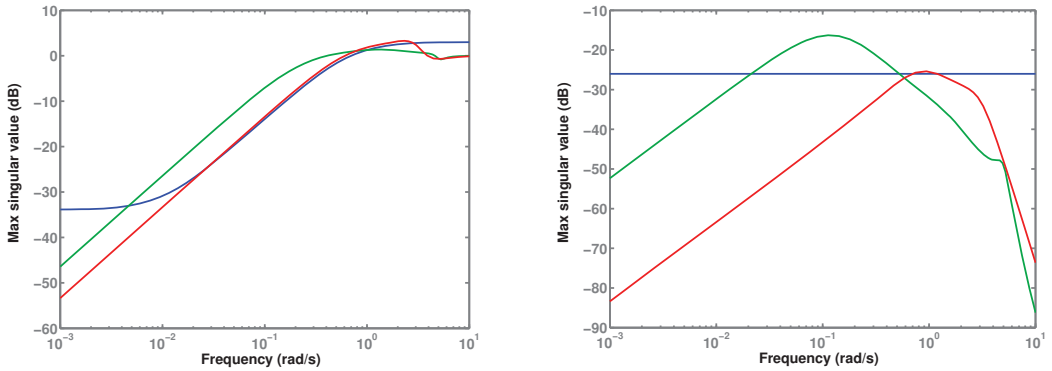


FIGURE 3.9.: Cross channels $\gamma \to dV$ (left) and $V \to d\gamma$ (right) for autopilot. The template -26dB is given in blue. Smallness of these responses assures decoupling of climb angle and velocity. The constant template indicates simply a weighting of the H_∞-norms. Decoupling increases the overall robustness of the design.

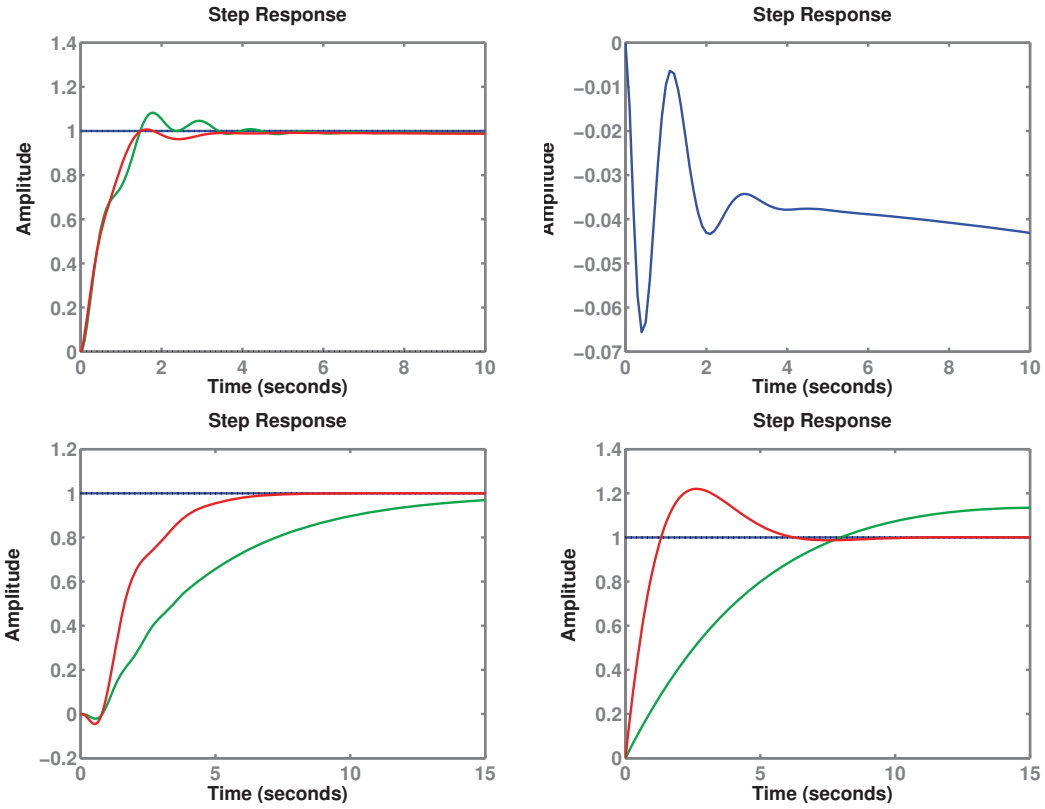


FIGURE 3.10.: Step responses for $\mathbf{x}^*$. At top, from left to right $T_{N_z \to dN_z}$, $T_{n_q \to dm}$. At bottom, from left to right $T_{\gamma \to d\gamma}$, $T_{V \to dV}$.

3.6. Conclusion

We have applied a nonconvex bundle algorithm to solve a multi-objective H_∞-control design problem (3.5), where the controller is structured. Convergence of the algorithm has been proved in the sense that every accumulation point $\mathbf{x}^*$ of the sequence of serious iterates is either a critical point of constraint violation, or a Karush-Kuhn-Tucker point. We have shown that the algorithm allows to solve the problem of simultaneous synthesis of flight controller and autopilot in longitudinal flight of aircraft.

The proposed technique has two advantages over the model-based bundle technique of [18], where an ideal model is used to compute cutting planes. In the case of the composite H_∞-norm (3.6), this ideal model is of the form

$$\phi(\cdot,\mathbf{x}) = \max_{\omega\in\mathbb{S}^1} \lambda_1\left(\mathcal{F}(\mathbf{x},\omega) + \mathcal{F}'(\mathbf{x},\omega)(\cdot-\mathbf{x})\right)$$

and has therefore the same structure as (3.6), but may be costly to compute if the system gets sizable. In [28] it was shown that computing $\phi(\mathbf{y},\mathbf{x})$ at a trial step $\mathbf{y}$ can be up to 27 times more expensive than computing the objective $f(\mathbf{y})$ itself. A second observation is that the new method seems less prone to rapid increase of the τ-parameter in the inner loop, which on average allows larger steps.

Appendix

The numerical data for the specific flight point Mach$= 0.7$, Altitude$= 5000\,ft$ used in (3.5) are

$$A = \begin{bmatrix} -0.0120 & -9.8040 & -14.8800 & 0 & 0 \\ 0.0004 & 0 & 0.8524 & 0 & -0.0000 \\ -0.0004 & 0 & -0.8524 & 1.0000 & 0.0000 \\ 0 & 0 & -2.6650 & -0.2783 & 0 \\ 0 & 234.1000 & 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 4.9580 & 0 \\ 0 & 0.3113 \\ 0 & -0.3113 \\ 0 & -4.9360 \\ 0 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 1.0000 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 \\ 0.0085 & 0 & 13.5409 & -0.7092 & -0.0001 \\ 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & & 1.0000 \end{bmatrix},$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & -5.1535 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

3.7. Mixed time/frequency domains control design

3.7.1. Frequency synthesis

Performance and robustness criteria are defined by introducing frequency weights on specific closed-loop transfer functions $T_{w_i \to z_i}(\mathbf{x}, s)$ between suitably chosen inputs w_i and outputs z_i. In this study we consider the six transfers $V_c \to dV$, $\gamma_c \to d\gamma$, $\gamma_c \to dV$, $V_c \to d\gamma$, $N_{z_c} \to dN_z$, $(N_{z_c}, n_q) \to dm$.

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) := \max\limits_{i=1,\dots,4} \left\| W_i^{-1} T_{w_i \to z_i}(\mathbf{x}) \right\|^2_{\infty,\Omega_{\text{low}}} \\ \text{subject to} & c(\mathbf{x}) := \max\limits_{i=5,6} \left\| W_i^{-1} T_{w_i \to z_i}(\mathbf{x}) \right\|^2_{\infty,\Omega_{\text{high}}} - r^2 \leq 0 \end{array} \tag{3.39}$$

Solution Analysis :
We take the following initial point to initialize program (3.39) :

```
Kp=-0.1;Kv=-1;Ki=-0.15;
a=25;b=sqrt(2)*5;
Kp_vit=-0.05;
Ki_vit=-0.0035;
Kp_pente=-5;
K_dec=0;
```

The optimized controller gains computed by the non convex bundle method to solve program (3.39) are

```
Kp=-9.3726e-02;
Ki=-1.0830e-01;
Kv=-6.4800e-01;
a=3.4335e+01;
b=1.0743e+01;
Kp_pente=-1.0968e+01;
Kp_vit=-2.1848e-01;
Ki_vit=-1.4199e-01;
K_dec=2.5822e-02;
```

We are not satisfied by the step response of the channel $V_c \rightarrow dV$, because of its overshoot. If we look at the performances of the step response we see that the percentage overshoot is about 23% and the settling time to get within 5% of the final value 1 is about 5.2 seconds. We then decide to minimize the overshoot but at the same time preserve

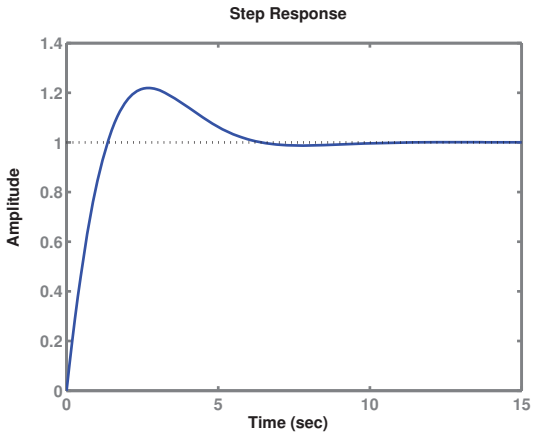


FIGURE 3.11.: step response of the channel $V_c \rightarrow dV$

the performances and robustness of the command law described in frequency domain. The overshoot is achieved at the time 2.7 seconds. So we define $t_0 = 2.7$ and the function $z(.,t_0) : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $z(\mathbf{x},t_0)$ is the value of the step response at time t_0 of the closed loop channel $V_c \rightarrow dV$, where the components of $\mathbf{x}$ are the values of the feedback gains.

3.7.2. Mixed frequency time domains synthesis

The goal now is to solve the optimization program mixing frequency/time constraints :

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) := z(\mathbf{x},t_0) \\ \text{subject to} & c(\mathbf{x}) := \max\limits_{i=1,\ldots,6} \left\| W_i^{-1} T_{w_i \rightarrow z_i}(\mathbf{x}) \right\|^2_{\infty,\Omega_{\text{high}}} - r^2 \leq 0 \end{array} \tag{3.40}$$

Program (3.40) takes into account the desire « to push down the hump » while maintaining the performances of the closed loop system in the frequency domain. The default of program (3.40) is that it does not take into account the settling time performance. Indeed look at the solution of program (3.40). Below, the figure 3.12 shows the evolution of the objective $z(.,t_0)$ during the iterations of the outer loop of the algorithm and the figure 3.13 shows the step response $z(\mathbf{x},.)$ for different controllers $\mathbf{x}$. The settling times for all of these curves are : $1.7s$ for the red one, $2.2s$ for the blue one, $4.6s$ for the magenta one and $7s$ for the goose poop one. So the goose poop one is not interesting because it degrades settling time performance. So the formulation (3.39) is not a good formulation. May be it is better to take into account the overshoot as a constraint of the optimization program instead of the objective. The figures just below presents the evolution of the other channels $\gamma_c \rightarrow d\gamma$, $\gamma_c \rightarrow dV$, $V_c \rightarrow d\gamma$, $N_{z_c} \rightarrow dN_z$, $(N_{z_c}, n_q) \rightarrow dm$,

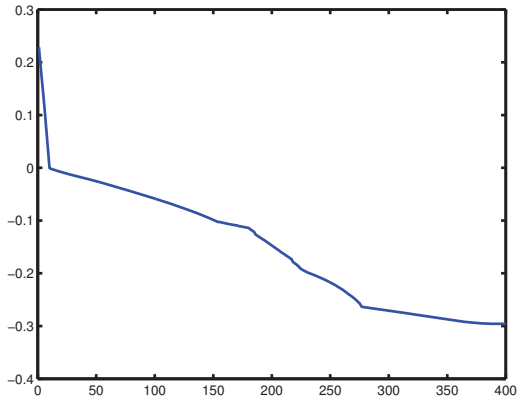


FIGURE 3.12.: x axis : Outer iterations, y axis : $z(x, t_0)$.

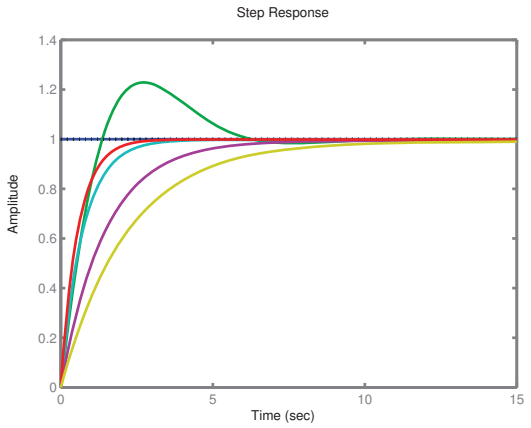
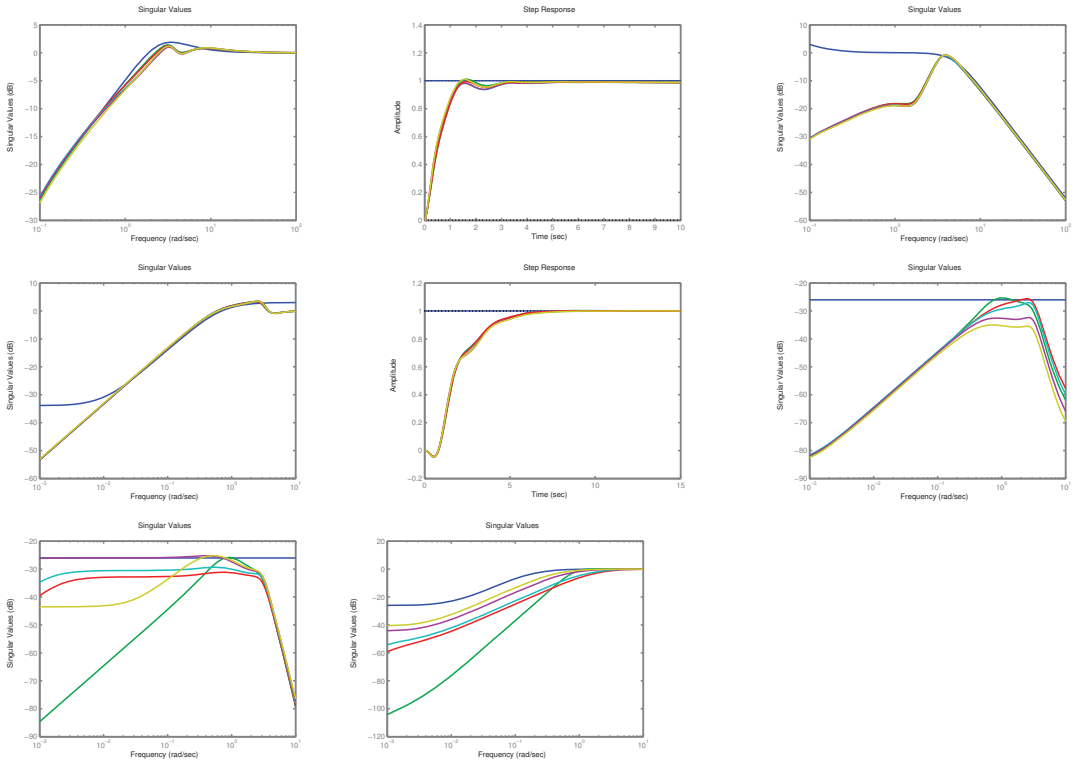


FIGURE 3.13.: x axis : seconds, y axis : $z(x, .)$ for x obtained at iterations 0, 10, 50, 200, 400.

in frequency domain and time domain. The controller x_{10} obtained at the iteration 10 gives a very good settling time, but we wonder if it is a physical solution, i.e. if the actuator dynamics can follow the command ... We take $1/(s+1)$ for the thrust dynamic and $400/(s^2+28s+400)$ for the elevator deflection dynamic. Let see the step response with the addition of these filters on the figure 3.14. The system to be controlled has now 3 more states. Here, the settling time of the goose poop curve is $4.6s$ and the overshoot is under 5%, so in these configuration the controller x_{400} is the best, but the frequency template $W(s) = 0.05$ is highly violated by the transfer $V_c \to d\gamma$.



3.7.3. Time constraint gradient

$z(.,t_0) : \mathbb{R}^n \to \mathbb{R}$. With respect to the standard form, we have

$$\begin{bmatrix} Z(\mathbf{x},s) \\ Y(\mathbf{x},s) \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \begin{bmatrix} W(s) \\ U(\mathbf{x},s) \end{bmatrix}, \; U(\mathbf{x},s) = K(\mathbf{x},s)Y(\mathbf{x},s).$$

Then by applying the differentiation rules we have for all $i = 1, ..., n$,

$$\begin{bmatrix} \partial Z(\mathbf{x},s)/\partial \mathbf{x}_i \\ \partial Y(\mathbf{x},s)/\partial \mathbf{x}_i \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \begin{bmatrix} 0 \\ \partial U(\mathbf{x},s)/\partial \mathbf{x}_i \end{bmatrix},$$

and

$$\partial U(\mathbf{x},s)/\partial \mathbf{x}_i = \partial K(\mathbf{x},s)/\partial \mathbf{x}_i \; Y(\mathbf{x},s) + K(\mathbf{x},s) \; \partial Y(\mathbf{x},s)/\partial \mathbf{x}_i.$$

Because of the Laplace transform is linear and injective we have,

$$\partial z(\mathbf{x},.)/\partial \mathbf{x}_i = L^{-1}(\partial Z(\mathbf{x},.)/\partial \mathbf{x}_i).$$

So to obtain the gradient of the function $z(.,t_0) : \mathbb{R}^n \to \mathbb{R}$ at the point $\mathbf{x}$, which is $(\partial z(\mathbf{x},t_0)/\partial \mathbf{x}_i)_{i=1}^n$, you can simulate the step response of the channel $w \to z$, which gives you $y(\mathbf{x},.)$, then you simulate the response of the plant $\tilde{P}_i$ (see figure 3.15) to obtain $\partial z(\mathbf{x},.)/\partial \mathbf{x}_i$.

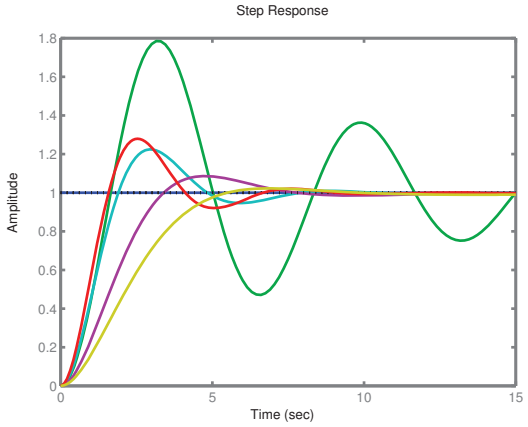
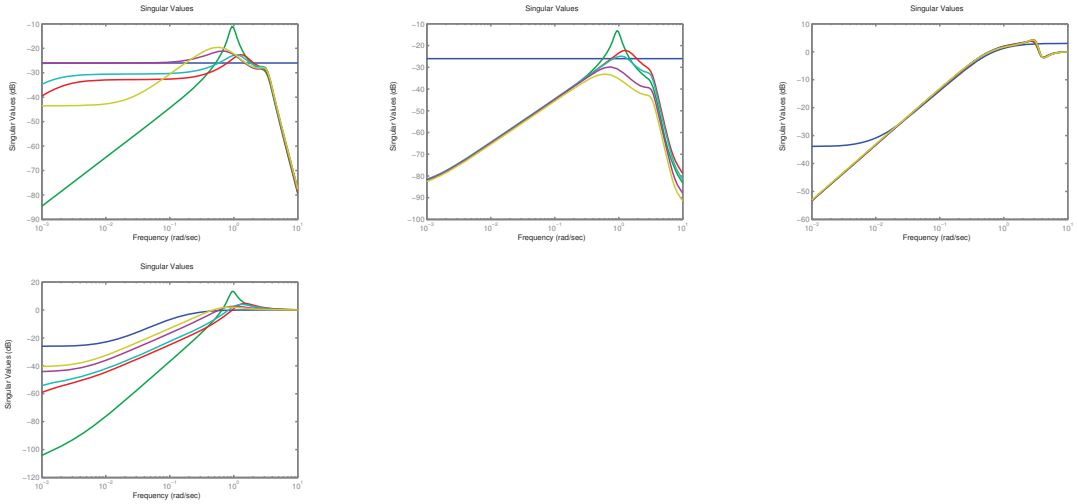


FIGURE 3.14.: x axis : seconds, y axis : $z(x,.)$ for x obtained at iterations 0, 10, 50, 200, 400.



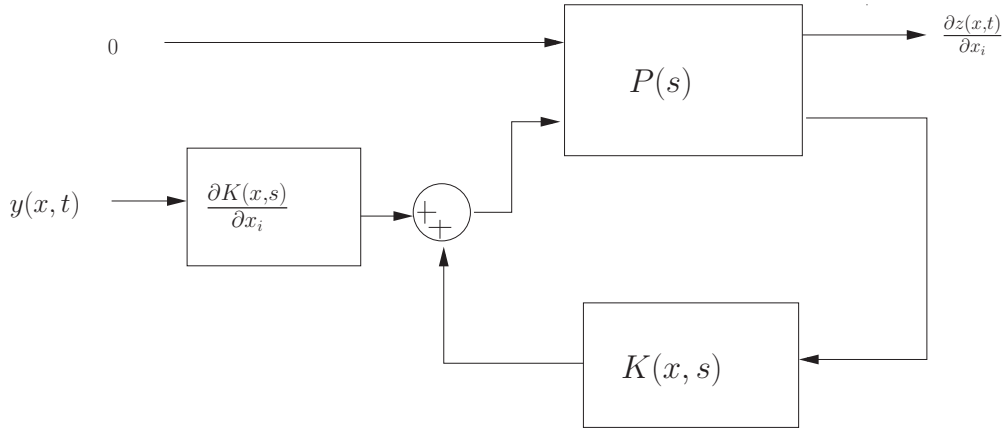


FIGURE 3.15.: $\tilde{P}_i$

4. Gain scheduled control law synthesis.

4.1. Longitudinal control problem over a flight envelope.

We consider an aircraft moving in the vertical plane. In this configuration, the aircraft is described by the state $x_P = (V,\ \gamma,\ \alpha,\ q,\ H)$ where $V\ [m/s]$ is the aerodynamic speed, $\gamma\ [rd]$ is the climb angle , $\alpha\ [rd]$ is the angle of attack, $q = \dot{\theta} = \dot{\alpha} + \dot{\gamma}\ [rd/s]$ is the pitch rate, and $H\ [m]$ is the altitude. The control by means of which we act on the aircraft is $u = (x,\ m)$ where x ($\%$ of the maximal thrust) is the engine thrust and $m\ [rd]$ is the elevator deflection. The measurement we collect to design a feedback is $y_P = (V,\ \gamma,\ N_z,\ q,\ H)$ where $N_z\ [m/s^2]$ is the vertical load factor. The aerodynamic behaviour of the aircraft is described locally by linearising the non-linear equations of the longitudinal motion around an equilibrium point (x_{Pe}, u_e). We recall some elementary definition and result. For a system of the general form

$$\dot{x} = f(x,u), \quad f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \text{sufficiently smooth}, \tag{4.1}$$

a point $(x_{P_e}, u_e) \in \mathbb{R}^n \times \mathbb{R}^m$ is an equilibrium point of (4.1) if by definition $f(x_{P_e}, u_e) = 0$. In practice, the research of an equilibrium point is performed numerically. Compute a zero of f is equivalent to resolve a set of non-linear equations. In order to apply an iterative algorithm like Newton algorithm, the set (4.1) which is under-determined, need to be completed to make it square. In pratice, as many independent linear equations as commands are added to the original set (4.1). In this application, the added equations are $V_e = V$ and $H_e = H$ for a given aerodynamic speed V and altitude H. You can find more information and references on the book [41]. In this configuration, the set of equilibrium points is parametrized by (V, H) and the linear model at point (V, H) writes

$$\begin{bmatrix} \delta\dot{x}_P(t) \\ \delta y_P(t) \end{bmatrix} = \begin{bmatrix} A(V,H) & B(V,H) \\ C(V,H) & D(V,H) \end{bmatrix} \begin{bmatrix} \delta x_P(t) \\ \delta u(t) \end{bmatrix}, \tag{4.2}$$

where $\delta x_P = x_P - x_{Pe}$, $\delta u = u - u_e$. The closed loop system is presented in Figure 4.1 and takes into account :
– the Tchebychev filter :

$$T(s) = \frac{1}{0.023s^2 + 0.1585s + 1},$$

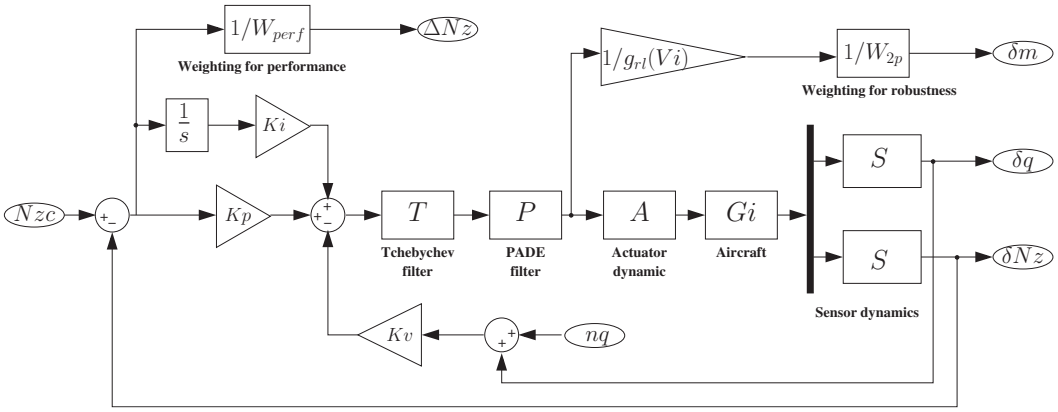


FIGURE 4.1.: Functional scheme of the flight control loop.

which gives robustness to unmodeled dynamics caused mainly by flexible structural modes [32],

– the 2 order PADE filter :

$$P(s) = \frac{(0.025^2/12)s^2 - (0.025/2)s + 1}{(0.025^2/12)s^2 + (0.025/2)s + 1},$$

which simulates a transmission delays of $25\ ms$,

– the elevator deflection dynamic :

$$A(s) = \frac{1}{0.07s + 1},$$

– the sensor dynamics :

$$S(s) = \frac{1}{0.1s + 1}.$$

The flight control law at flight point (V, H) governing the elevator deflection δm reads

$$\delta m(s) = \begin{bmatrix} K_p(V,H) + K_i(V,H)/(s+\varepsilon) & -K_v(V,H) \end{bmatrix} \begin{bmatrix} N_{z_c}(s) - \delta N_z(s) \\ \delta q(s) \end{bmatrix} \quad (4.3)$$

and combines a proportional integral (PI) feedback to servo-loop the vertical load factor δN_z with a proportional (P) feedback on the pitch rate δq to damp the angle-of-attack (AoA) oscillation. We want to compute three controller gain functions $K_p(V, H)$, $K_i(V, H)$, $K_v(V, H)$ in order to ensure performances and robustness of the closed loop (Figure 4.1) over the flight envelope $\mathcal{H}$ delimited by :

$$5000 < H < 35000 \text{ and } 160 < V < 300.$$

The performances are shaped by the template

$$W_{perf}(s) = \frac{s^2 + 1.33s + 0.001}{s^2 + s + 1}$$

on the transfer $N_{z_c} \rightarrow \Delta N_z = N_{z_c} - \delta N_z$, where $\delta N_z = N_z - N_{z_e}$ and N_{z_c} is the vertical load factor reference input. The robustness is shaped by the roll off template

$$W_{rl}(V, s) = g_{rl}(V) W_{2p}(s) \text{ with : } W_{2p}(s) = 25\frac{(s/500)^2 + 1.4(s/500) + 1}{s^2 + 5\sqrt{2}s + 25}$$

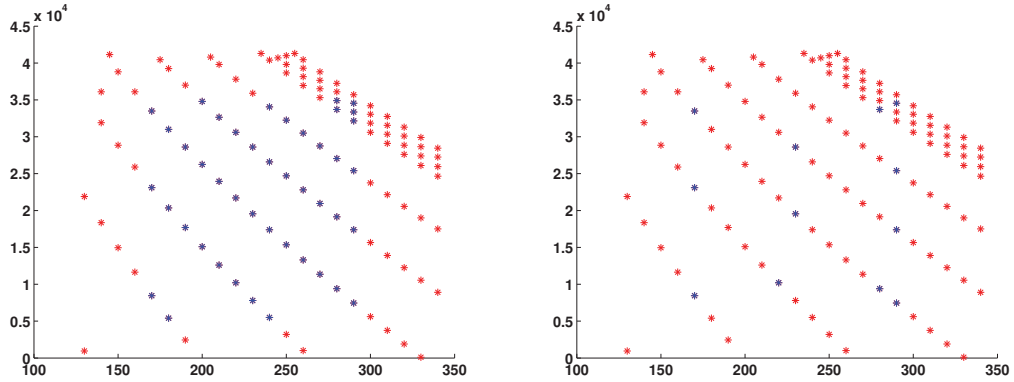


FIGURE 4.2.: Flight envelope.

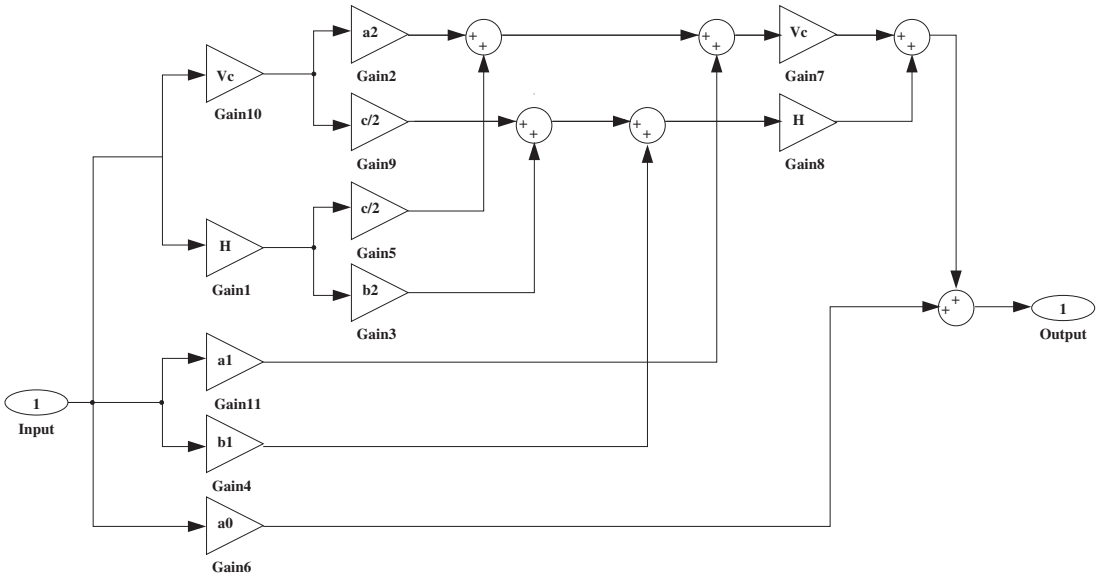


FIGURE 4.3.: Controller gain parametrized by (V, H) as a 2 degree polynomial function.

on the transfer $[N_{z_c}, n_q] \to \delta m$. The template W_{perf} is kept fixed over the flight envelope whereas the template W_{rl} depend on V. We allow larger amplitudes for the control at low speeds through the gain $g_{rl}(V)$ in order to be able to satisfy the performance template W_{perf} over the flight envelope. A collection of aircraft models data,

$$\left(G_i(s) = \left[\begin{array}{c|c} A_i & B_i \\ \hline C_i & D_i \end{array}\right]\right)_{i\in\mathcal{I}},$$

is provided by [42], on a grid $((V_i, H_i))_{i\in\mathcal{I}}$ of the flight envelope $\mathcal{H}$ (see the grid in Figure 4.2). In order to deal with a finite dimensional optimization problem, an a priori dependence law of the controller gains K_p, K_i, K_v with respect to (V, H) is chosen. In this application a 2 degree polynomial dependence is proposed :

$$a_0 + a_1 V + b_1 H + a_2 V^2 + b_2 H^2 + cVH. \tag{4.4}$$

The bloc diagram sketch of such a controller gain is in Figure 4.3. The optimization

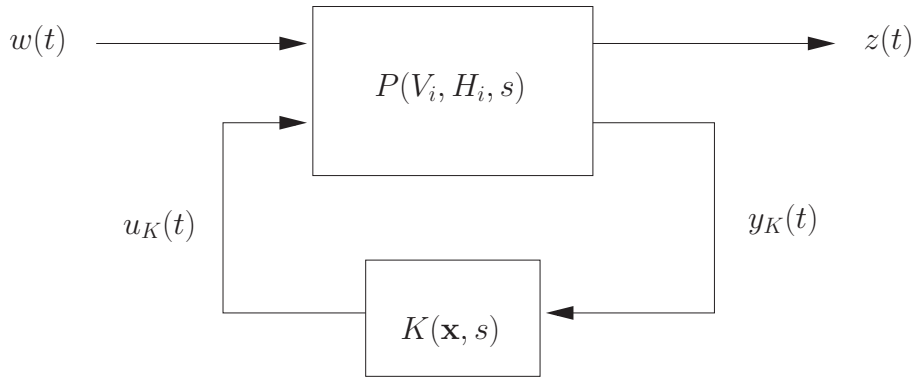


FIGURE 4.4.: Standard H_∞ formulation of the system $T^i_{w\to z}(\mathbf{x})$.

variables are grouped in the vector $\mathbf{x}$. To distinguish which coefficient belongs to which gain, we add to the form (4.4) the indices p, i and v to refer respectively to K_p, K_i and K_v :

$$\mathbf{x} = [a_{0p}, a_{1p}, b_{1p}, a_{2p}, b_{2p}, c_p, a_{0i}, a_{1i}, b_{1i}, a_{2i}, b_{2i}, c_i, a_{0v}, a_{1v}, b_{1v}, a_{2v}, b_{2v}, c_v]^\top. \quad (4.5)$$

The optimization program is cast as follow :

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x}) := \max_{i\in\mathcal{A}} \max \\ &\left\{ \left\| W_{perf}^{-1} T^i_{N_{z_c}\to\Delta N_z}(\mathbf{x}) \right\|_\infty^2, \left\| W_{rl}(V_i)^{-1} T^i_{[N_{z_c},n_q]\to\delta m}(\mathbf{x}) \right\|_\infty^2 \right\}, \end{aligned} \quad (4.6)$$

where $T^i_{N_{z_c}\to\Delta N_z}$, $T^i_{[N_{z_c},n_q]\to\delta m}(\mathbf{x})$ are the closed loop transfer functions at the flight point (V_i, H_i) and $\mathcal{A} \subset \mathcal{I}$. As regards the implementation of such a problem, we use the standard formulation of the H_∞ synthesis (see Figure 4.4). We model the closed loop transfer function $T^i_{w\to z}(\mathbf{x})$ as a system $P(V_i, H_i, s)$ containing the flight parameter V_i and H_i and which is looped back to the command law $U_K(s) = K(\mathbf{x}, s)Y_K(s)$ containing the optimization variables. To do this, we pull out the gains V_i and H_i of the controller gains K_p, K_i and K_v (see Figure 4.3) by adding some artificial inputs and outputs to the controller.

4.2. Solution analysis

The controller used to initialize the non-smooth algorithm, is a hand-tuning performed on the linear model Figure 4.1 corresponding to the mid-point of the flight envelope $V_c = 230$, $H = 19550$. According to (4.5) this tuning writes

$$\mathbf{x}_0 = [-5, 0, 0, 0, 0, 0, -7, 0, 0, 0, 0, 0, -1.5, 0, 0, 0, 0, 0]. \quad (4.7)$$

4.2.1. Case 1 : Design with all models in a restricted flight domain

The set $\mathcal{A}$ in (4.6) is the set of indices of flight point (V_i, H_i) in blue on the left of Figure 4.2. There are 44 points. The curves of the frequency and step responses are presented on Figure 4.5 and Figure 4.6. Initially, the maximum of the H_∞ norms in (4.6) over the set $\mathcal{A}$ at the initial point x_0 is $f(x_0) = 3.8295e{+}01$. After optimization, the maximum at the optimized point x^* is $f(x^*) = 1.2479e + 00$. We can note that the step responses are more grouped and their performances are more homogeneous in the optimal case than the initial case. For comparison, a synthesis is performed locally at each flight point of the grid $((V_i, H_i))_{i\in\mathcal{I}}$. That means, for all $i \in \mathcal{I}$, a couple of three real values of the controller gains, denoted by (K_p^i, K_i^i, K_v^i), is computed to minimize the max of the H_∞ norms of performance and robustness channels of the model Figure 4.1 corresponding to the flight point (V_i, H_i). Then, a 2 degree polynomial approximation in the least square sense of the point set $\{K_p^i, K_i^i, K_v^i,\ i \in \mathcal{I}\}$ is computed. This procedure is called local synthesis by opposition to the synthesis (4.6) called global synthesis. In Figure 4.7 we compare the parametrizations of the controller gains computed by the local and global synthesises evaluated on the following regular grid g_1

$$V = 160 : 5 : 300, \quad H = 5000 : 1000 : 35000$$

of the flight envelope $\mathcal{H}$. On the most part of the flight envelope, the surfaces corresponding to the global synthesis are above the local synthesis ones. For K_i the two surfaces are close for the values of speed V between 250 and 300 and they separate themselves when the speed is decreasing. For K_p the two surfaces have similar shapes only the curvature is different. For K_v the two surfaces are really different, while the local synthesis gain K_v increase with V, the global synthesis one decrease. In Figure 4.9, we compare the performance graphs generated by these parametrizations evaluated on the grid g_1 and also on the following smaller grid g_2

$$V = 175 : 5 : 290, \quad H = 8000 : 1000 : 33000$$

to do a zoom on the first one. As it can be expected, the local synthesis gives better control law performance than the global synthesis. To evaluate the performance at a flight point of the regular grid, we need a local model at this point, so a 2 degree polynomial approximation in the least square sense $G(V_c, H, s)$ of the model data set $\{G_i(s),\ i \in \mathcal{I}\}$ is computed as well. Figure 4.8 ensures the validity of the parametrized model $G(V_c, H, s)$. The blue curves are the step responses of $T^*_{N_{z_c}\to[\delta q,\delta N_z]}(x_0)$, where x_0 is a typical tuning of the controller for the midpoint (V^*=230,H^*=19550) of $\mathcal{H}$, that is $K_p = -2.4$, $K_i = -5$, $K_v = -1$. The other curves are the step responses of the model errors $T^i_{N_{z_c}\to[\delta q,\delta N_z]}(x_0) - T_{N_{z_c}\to[\delta q,\delta N_z]}(V_i, H_i, x_0)$, $i \in \mathcal{I}$. In order to avoid the creation of models outside of the flight envelope, and the use of points with no physical meaning, we restrict the considered domain (V, H) to the largest square inscribed in the flight envelope and centred on the nominal point $V = 230$, $H = 19550$. This restricted domain is plotted in blue on the left of Figure 4.2. However, it is possible to take into account the initial flight envelope in red in Figure 4.2 by setting two new parameters

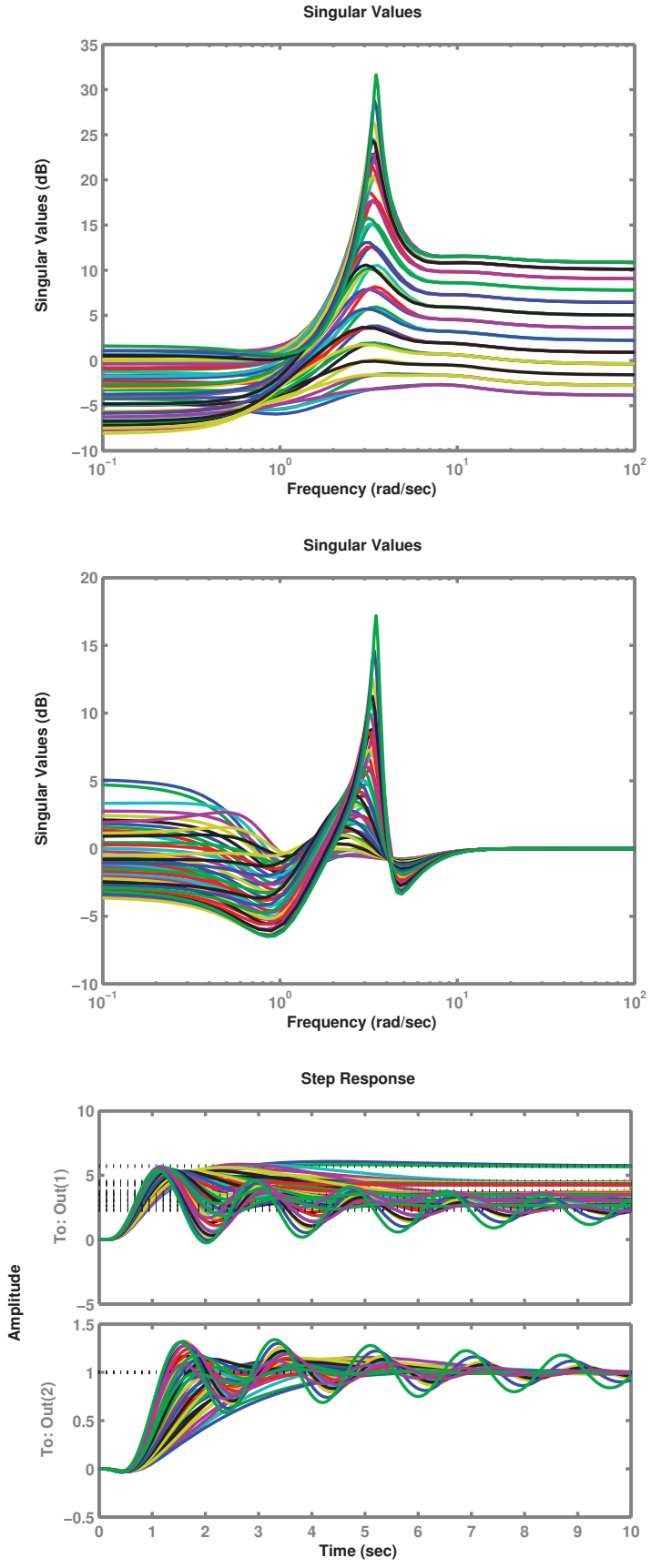


FIGURE 4.5.: Frequency responses of $T^i_{[N_{z_c},n_q]\to\delta m}(\mathbf{x}_0)$ at left and of $T^i_{N_{z_c}\to\delta N_z}(\mathbf{x}_0)$ at right and step response of $T^i_{N_{z_c}\to\delta N_z}(\mathbf{x}_0)$ at bottom, $i \in \mathcal{I}$, where $\mathbf{x}_0$ is the initial controller.

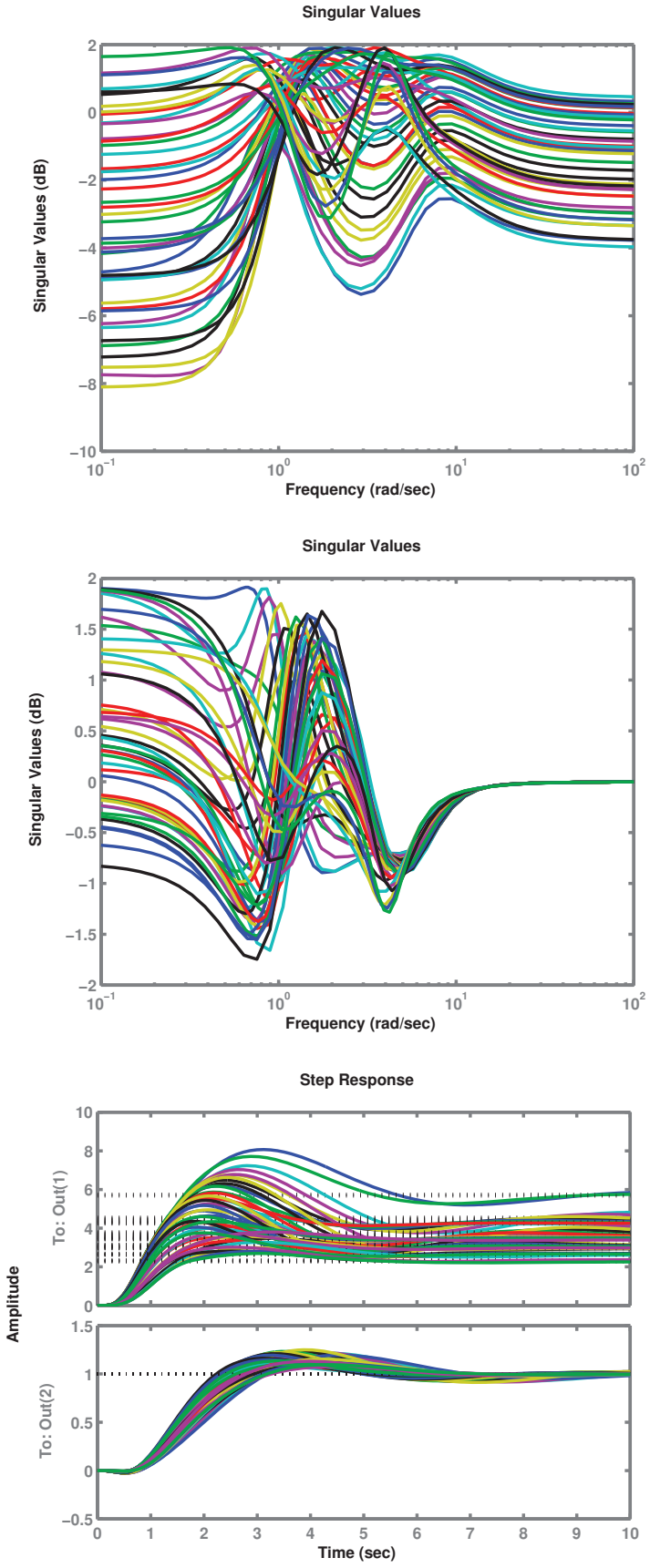


FIGURE 4.6.: Frequency analysis of $T^i_{N_{z_c} \to \delta N_z}(\mathbf{x}^*)$ at left and $T^i_{[N_{z_c}, n_q] \to \delta m}(\mathbf{x}^*)$ at right and step response of $T^i_{N_{z_c} \to \delta N_z}(\mathbf{x}^*)$ at bottom, $i \in \mathcal{I}$, where $\mathbf{x}^*$ is the optimized controller.

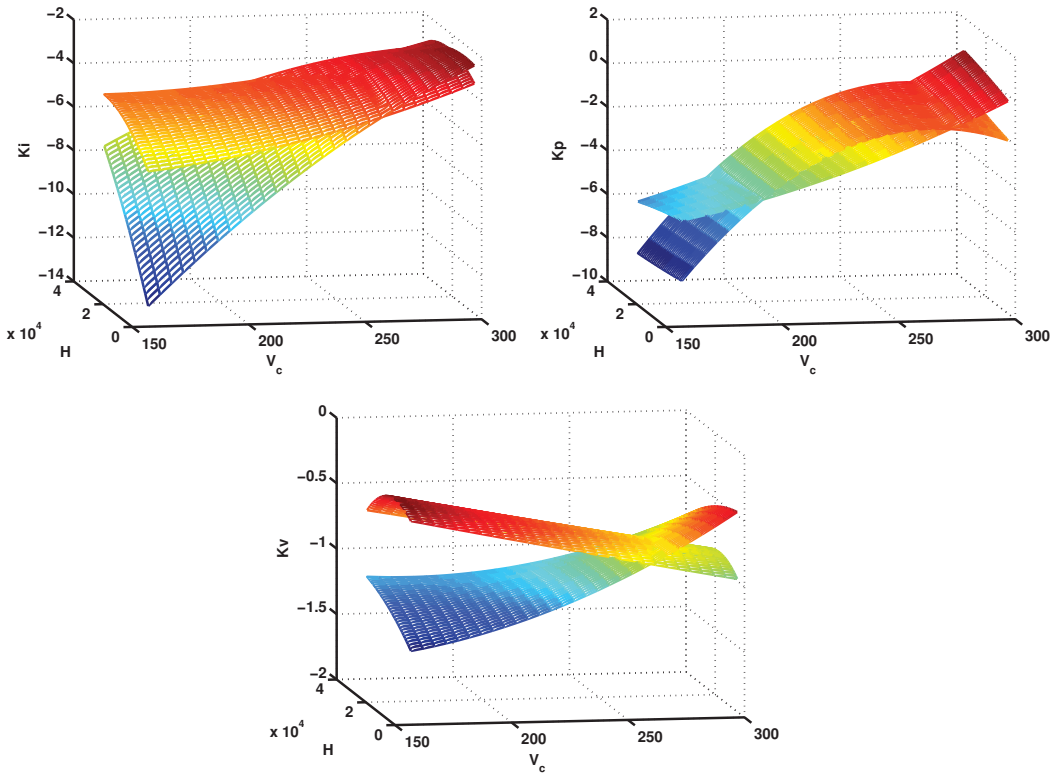


FIGURE 4.7.: Comparison between the parametrizations of K_i (left), K_p (right) and K_v (below) computed by local and global synthesises.

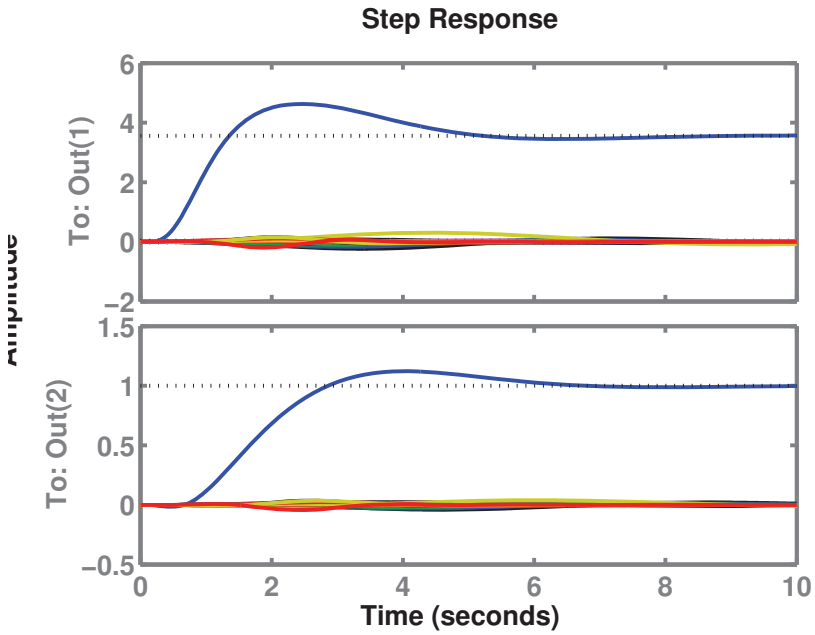


FIGURE 4.8.: Validation of the model $G(V_c, H, s)$ which approximates the sample $(G_i(s))_{i\in\mathcal{I}}$.

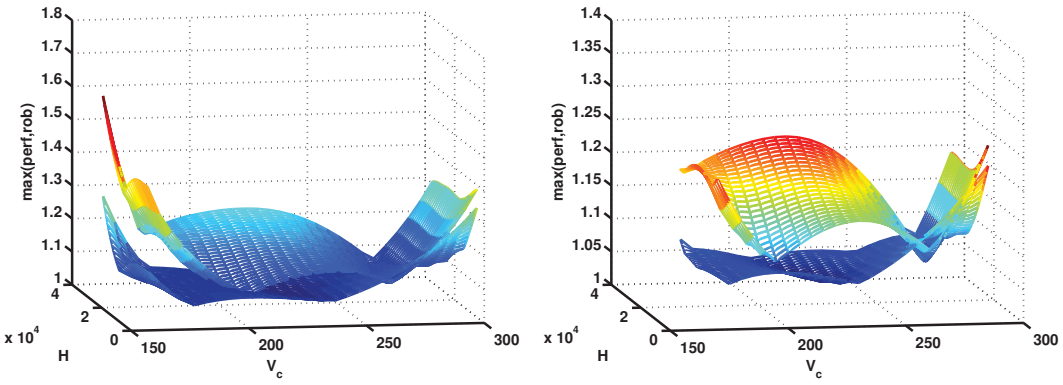


FIGURE 4.9.: Comparison between the performance graphs computed by evaluating the parametrizations of the controller gains on the regular grid g_1 on the left and g_2 on the right.

x_1 and x_2 and a LFT (linear fractional transformation) which allows to transform the square domain (x_1, x_2) into the non square domain (V, H).

4.2.2. Case 2 : Design with few models in a restricted flight domain

The set $\mathcal{A}$ in (4.6) is the set of indices of flight point (V_i, H_i) in blue on the right of Figure 4.2. There are 12 points. In this case, we consider much fewer points than Case 1 section 4.2.1. We want to know if this loss of information in the synthesis (4.6) penalized the performance over the flight envelope. We recall that in Case 1, we consider 44 flight points, so we minimize 88 H_∞ norms while in Case 2, we consider 12 flight points, so we minimize 24 H_∞ norms. In Figure 4.10 we compare the solutions of Case 1 and Case 2. If we compare the cases in the grid 1, bottom line of Figure 4.10, one can note that on the boundary of the flight envelope $V = 160$ there is peak for Case 1. However on the grid 2, top line of Figure 4.10, Case 1 is a little better.

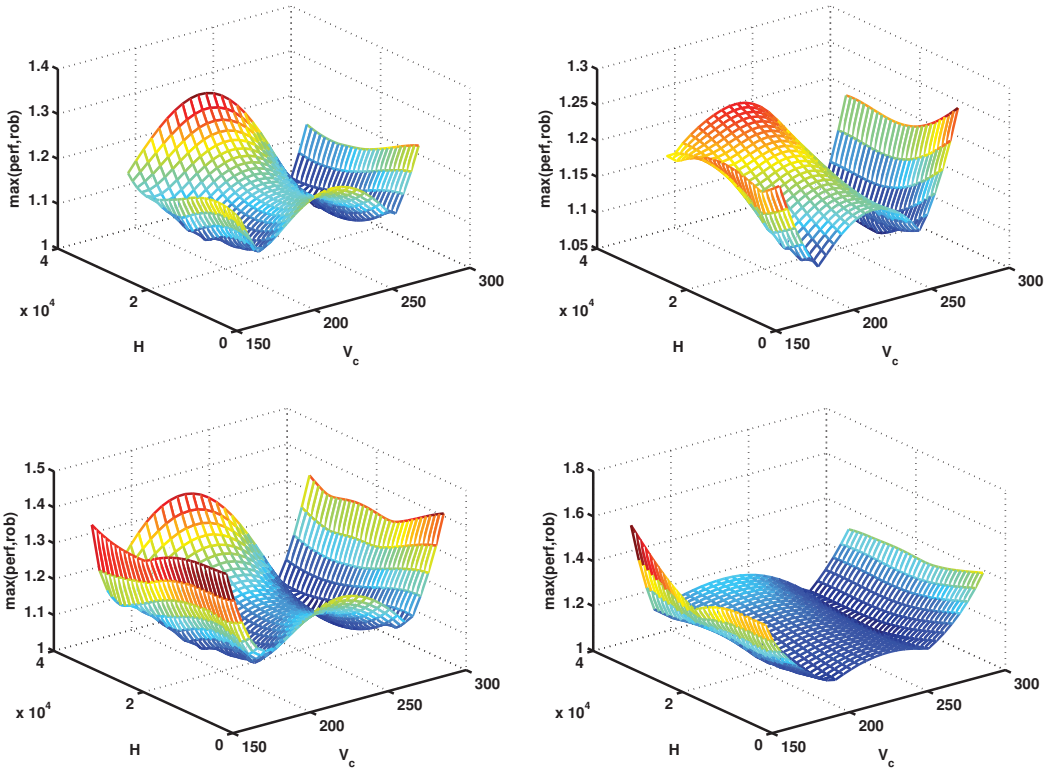


FIGURE 4.10.: On the left : Case 2, on the right : Case 1. On the top : grid g_2, on the bottom : grid g_1.

4.3. μ-analysis

You can find detailed explanations of the μ-analysis theory on the books [3], [43], [44] or on the Skew Mu toolbox documentation [45]. Consider a LTI system subject to different model uncertainties, and assume that the nominal system (i.e. without model uncertainties) is stable and satisfies a performance criterion. The purpose is to estimate the robustness margin, i.e. the maximal amount of model uncertainties for which the system satisfies this criterion. The μ-analysis theory deals with system shaped into $M-\Delta$ structure for robust stability analysis or $N-\Delta$ structure for robust performance analysis (see Figure 4.11). It is most generally possible to transform a specific uncertain plant into this standard interconnection structure : on the one hand, the transfer matrix $M(s)$ contains the dynamics of the nominal system (i.e. without any model uncertainty) and the way the various model perturbations enter the system. On the other hand, all model perturbations are gathered in the uncertain transfer matrix $\Delta(s)$, which has the following block diagonal structure :

$$\Delta = \text{diag}\{\Delta_1(s), ..., \Delta_m(s), \delta_1 I_{q_1}, ..., \delta_n I_{q_n}\}, \tag{4.8}$$

The block Δ_i is a transfer function of some neglected dynamics which is assumed to satisfy $\|\Delta_i\|_\infty \leqslant 1$, and the scalar δ_i is a real parametric uncertainty which is assumed to satisfy $\delta_i \in [-1, 1]$. These model uncertainties are normalized. We introduce the set

$$B\Delta = \{\Delta \text{ with the structure (4.8) and such that } \|\Delta\|_\infty < 1\},$$

which summarizes the normalization constraints on the neglected dynamics and on the parametric uncertainties.

4.3.1. Definition of the structured singular value μ

Here we deal with the question : If $M-0$ is stable, is the plant $M-\Delta$ stable for all $\Delta \in B\Delta$? First of all remark that the eigenvalues of the plant $N-\Delta$ are strictly the

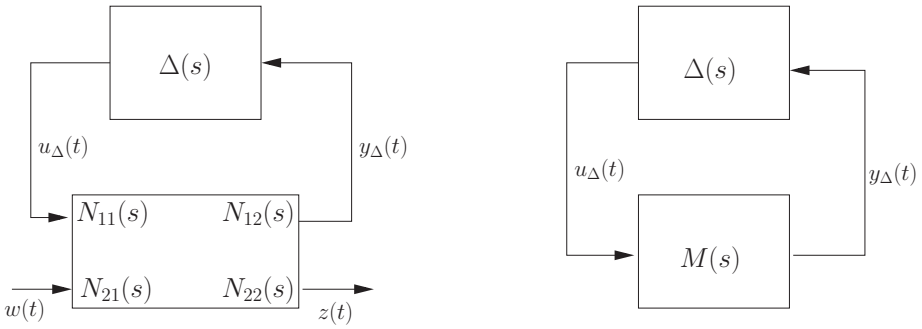


FIGURE 4.11.: On the left : $N-\Delta$ structure for robust performance analysis. On the right : $M-\Delta$ structure for robust stability analysis.

same than the eigenvalues of $M - \Delta$ with $M = N_{11}$. In the case of $\Delta = \text{diag}(\delta_i I_{q_i})_{i=1}^{n}$, the dynamic matrix of $M - \Delta$ is

$$A_{\text{cl}} = A + B_2(I - \Delta D_{22})^{-1}\Delta C_2,$$

with A, B_2, C_2, D_{22} such that $N_{11}(s) = D_{22}+C_2(sI-A)^{-1}B_2$. The matrix A is stable by assumption, so the matrix A_{cl} is also stable for $\Delta = 0$. Furthermore the eigenvalues of A_{cl} are continuous with respect to Δ. Therefore, by increasing k from 0, $M-\Delta$ becomes marginally stable (one or several poles are on the imaginary axis) before becoming unstable for some $\Delta \in kB\Delta$. The goal is to find the smallest value $k_m \in \mathbb{R}_+$, called the robustness margin, for which there exists $\Delta \in k_m B\Delta$ such that $M - \Delta$ is marginally stable. We can prove (see [3], page 135) that the characteristic polynomials of A and A_{cl} verify the following relation :

$$\frac{P_{cl}(s)}{P_{ol}(s)} = \frac{\det(I - \Delta M(s))}{\det(I - \Delta M(\infty))}. \tag{4.9}$$

We have $P_{ol}(s) \neq 0 \ \forall s \in \mathbb{C}_+$, and combined with the equation (4.9) we obtain the equivalence

$$\begin{aligned} (\forall s \in \mathbb{C}_+)\,(\forall \Delta \text{ for which } \det(I - \Delta M(\infty)) \neq 0) \\ (P_{cl}(s) = 0 \Leftrightarrow \det(I - \Delta M(s)) = 0)\,. \end{aligned} \tag{4.10}$$

The equation (4.10) says that $s^* \in \mathbb{C}_+$ is an eigenvalue of the plant $M - \Delta$, if and only if the matrix $I - \Delta M(s^*)$ is singular. It's why we define the structured singular value at frequency ω as :

$$\begin{aligned} \mu_\Delta(M(j\omega)) &= (\inf\{k \in \mathbb{R}_+ \ / \ \exists \Delta \in kB\Delta \text{ such that } \det(I - \Delta M(j\omega)) = 0\})^{-1}\,, \\ &= 0 \text{ if it does not exist such a } k. \end{aligned} \tag{4.11}$$

Finally $M - \Delta$ is asymptotically stable in the continuous domain $B\Delta$ if and only if the robustness margin

$$k_m \doteqdot \frac{1}{\max_{\omega\in[0,\infty]} \mu_\Delta(M(j\omega))}$$

is greater than one or equivalently the structured singular value $\mu = 1/k_m$ is smaller than one.

4.3.2. Definition of the skew mu μ^s

Here we deal with the question : Does the plant $N - \Delta$ verify the H_∞ performance $\|N - \Delta\|_\infty \leqslant 1$ for all $\Delta \in B\Delta$? In fact the robust performance problem can be equivalently transform into an augmented stability problem. The closed loop $N - \Delta$ writes

$$F_u(N, \Delta) = N_{22} + N_{21}\Delta\,(I - N_{11}\Delta)^{-1}\,N_{12}.$$

We can prove that for all $\omega \in \mathbb{R}_+$

$$\left(\mu_\Delta(N_{11}(j\omega)) < 1 \text{ and } \max_{\Delta \in B\Delta} \mu_{\Delta_1}(F_u(N(j\omega), \Delta)) < 1\right) \Longleftrightarrow \mu_{\Delta_2}(N(j\omega)) < 1, \tag{4.12}$$

where Δ_1 is a fictitious full complex block and $\Delta_2 = \text{diag}(\Delta, \Delta_1)$ is an augmented block diagonal matrix. The skew structured singular value at frequency ω is defined as :

$$\begin{aligned}\mu^s(N(j\omega)) = (\inf\{k \in \mathbb{R}_+ / \exists \Delta_2 = \text{diag}(\Delta, k\Delta_1) \text{ with } \Delta_i \in B\Delta_i \\ \text{and } \det(I - N(j\omega)\Delta) = 0\})^{-1}.\end{aligned} \tag{4.13}$$

4.3.3. Implementation

We use the Skew Mu Toolbox [45] to analyse the robustness of the auto-sequenced law synthesized using a non-smooth algorithm. The uncertain parameters of the closed loop Figure 4.1 are V and H. By construction, the auto-sequenced law ensures the stability and some performances of the closed loop over the grid $(V_i, H_i)_{i \in \mathcal{I}}$. But we want more, we want the stability and the performances to be ensured in the continuous flight envelope $\mathcal{H}$. It's why we do a μ-analysis.

For the controller obtained in case 1 subsection 4.2.1, an lower bound of the robustness margin is 1.29 so we can say that our command law stabilizes the simulation model over the all flight envelope. The worst case performance is 2.45.

5. Conclusion

Les travaux de cette thèse ont consisté à développer un algorithme de faisceaux non convexe avec contrôle de proximité et à l'appliquer à des problèmes pratiques de synthèse de lois de commande structurées issus de l'industrie aéronautique.

Le point de départ a été les travaux [18]. Ceux-ci proposent de définir, en chaque pas sérieux x, un modèle local convexe $\phi(.,x)$ de la fonction à minimiser f, et d'y appliquer une méthode de faisceaux convexe combinée à une gestion dynamique du paramètre de contrôle de proximité τ, pour générer le prochain itéré sérieux x^+. Typiquement, dans le cas où l'objectif f est la norme H_∞ d'une fonction de transfert T, le modèle idéal à l'itéré sérieux x, est la norme H_∞ de l'application linéaire tangente en x de T. Le paramètre de contrôle de proximité, nécessaire à la convergence d'une méthode de faisceaux quelconque, prend un rôle supplémentaire dans le cas non convexe. Il est traité de manière dynamique pour tenir compte, à chaque itération, du caractère local du modèle idéal, et de sa capacité à représenter convenablement la réalité f dans une certaine région de confiance. Cet algorithme a été appliqué avec succès sur un problème de minimisation d'une norme H_2 sous une contrainte de norme H_∞ [22] et sur un problème de minimisation d'une norme H_∞ [23].

Dans cette thèse, nous proposons une technique alternative, qui n'utilise plus d'intermédiaire entre l'objectif f et son modèle de travail Φ_k. Les tangentes décalées de f remplacent les tangentes de $\phi(.,x)$ dans la construction de Φ_k. De part sa structure très particulière qui compose une fonction convexe non différentiable avec une fonction non convexe différentiable, la norme H_∞ se prête bien à l'utilisation d'un modèle idéal, tout du moins théoriquement. En effet l'implémentation du modèle idéal a rencontré des difficultés techniques [23]. On montre dans [23] que le calcul de $\phi(y,x)$ est 27 fois plus coûteux que celui de $f(y)$. De plus, il n'est pas dit que pour une fonction quelconque autre que la norme H_∞, la construction théorique du modèle idéal soit aussi aisée. La technique avec tangentes décalées enlève les problèmes théoriques et numériques de la construction et de l'implémentation du modèle idéal. C'est pourquoi on s'est tourné vers cette technique alternative. Cependant on montre dans la section 2.4.1 que le modèle idéal présente des qualités, comme le fait que lorsque l'on entre dans un voisinage convexe d'un minimum local, la technique revient à appliquer une méthode de faisceaux convexe à f. La non convexité a priori de f ne permet pas de prendre directement ses tangentes pour construire le modèle de travail. En effet celles ci ne supportent plus la fonction par le bas, ou dit autrement, elles ne sont plus des minorants affines de f. Un décalage est alors nécessaire pour préserver la consistance de Φ_k vis à vis de f, i.e. $\Phi_k(x,x) = f(x)$ et $\partial\Phi_k(x,x) \subset \partial f(x)$.

D'un point de vue implémentation, on a intégré une technique de recyclage de plans sécants entre deux itérations successives de la boucle externe. Le recyclage permet de garder une mémoire et de transporter de l'information d'une itération à la suivante. Typiquement on garde les plans sécants du dernier modèle de travail minimisé associés à un multiplicateur de Lagrange strictement positif (cf. section 2.2). On a également intégré une technique heuristique de tangentes décalées de l'abscisse spectrale pour tenir compte de la contrainte de stabilité de la fonction de transfert en boucle fermée et améliorer la trajectoire des itérés sérieux. Ensuite on a opté pour une résolution de la forme primale du programme quadratique tangent pour des raisons de robustesse numérique. La forme duale fait intervenir une multiplication de matrices de la forme $A^\top A$ où les colonnes de A sont les sous-gradients des plans sécants du modèle de travail. On s'est rendu compte que ce produit causait des problèmes de conditionnement et des échecs de résolution numérique. Enfin le code a été généralisé à la minimisation d'un nombre arbitraire de normes H_∞ et à la prise en compte de problèmes d'optimisation avec et sans contraintes.

Les algorithmes ont été illustré et validé sur des applications aéronautiques pour lesquelles la structure de la loi de commande est primordiale. Nous nous sommes intéressés à la synthèse conjointe de l'auto-pilote et de la loi de commande de vol et à la synthèse d'une loi paramétrée en vitesse et altitude asservissant une famille de modèles linéaires associée à un ensemble de points dans le domaine de vol. La résolution de l'exemple académique de la section 2.4.1 a permis de mettre en évidence des caractéristiques intéressantes des algorithmes et a permis d'illustrer les concepts sous-jacents aux algorithmes non lisses et non convexes.

Des recherches sont à faire concernant le certificat de stabilité d'une loi de commande structurée dépendant de paramètres variants et concernant les lois de commande structurées robustes aux incertitudes structurées [46], [47].

A. Différentiabilité d'une valeur propre simple

Soit E l'espace vectoriel réel des matrices de taille $n \times n$ à coefficients réels. On note E^* son dual et $\langle \cdot, \cdot \rangle$ son produit scalaire canonique, i.e.

$$\forall A = (a_{ij}), B = (b_{ij}) \in E, \quad \langle A, B \rangle = \text{Tr}(A^T B) = \sum_{1 \leq i,j \leq n} a_{ij} b_{ij}. \tag{A.1}$$

Soient $A = (a_{ij}) \in E$ et λ^* une valeur propre simple de A. On définit

$$\begin{aligned} p : \mathbb{C} \times E &\longrightarrow \mathbb{C} \\ (\lambda, M) &\longrightarrow \chi_M(\lambda) \doteqdot \det(M - \lambda I). \end{aligned}$$

Comme λ^* est une valeur propre simple de A, $\chi_A(\lambda) = (\lambda - \lambda^*)Q(\lambda)$ avec $Q(\lambda^*) \neq 0$ et

$$\begin{aligned} \frac{\partial}{\partial \lambda} p(\lambda^*, A) &\doteqdot \lim_{\lambda \to \lambda^*} \frac{p(\lambda, A) - p(\lambda^*, A)}{\lambda - \lambda^*}, \\ &= \lim_{\lambda \to \lambda^*} Q(\lambda) = Q(\lambda^*) \neq 0. \end{aligned}$$

Ainsi

$$\begin{cases} p(\lambda^*, A) = 0, \\ \dfrac{\partial}{\partial \lambda} p(\lambda^*, A) \neq 0. \end{cases}$$

Sous réserve de montrer que p est de classe C^1 sur un ouvert contenant (λ^*, A), le théorème des fonctions implicites nous donne l'existence d'une fonction $g : E \to \mathbb{C}$ définie et différentiable sur un voisinage ouvert $U \in \mathcal{V}(A)$ de A tel que

$$p(g(M), M) = 0, \, \forall M \in U.$$

On cherche le gradient $\nabla g(A) \in E$ de g en A associé à la différentielle $g'(A) \in E^*$ de g en A et au produit scalaire canonique (A.1), i.e. on cherche $\nabla g(A)$ tel que $\langle \nabla g(A), \Delta A \rangle = g'(A)\,(\Delta A)$ pour tout $\Delta A \in E$. Son existence est assuré par le théorème de représentation de Riesz. On montre qu'il est égal à la matrice $(\partial g(A)/\partial a_{ij})$ des dérivées partielles de g en A. Par définition

$$\frac{\partial}{\partial a_{ij}} g(A) \doteqdot \lim_{\varepsilon \to 0} \frac{g(A + \varepsilon M_{ij}) - g(A)}{\varepsilon}$$

où M_{ij} est la matrice dont les coefficients sont tous nuls sauf le $(i,j)^{\text{ème}}$ qui vaut 1. Or

$$g(A + \varepsilon M_{ij}) - g(A) = \varepsilon g'(A)(M_{ij}) + o(|\varepsilon|),$$

d'où

$$\langle \nabla g(A), M_{ij} \rangle = g'(A)(M_{ij}) = \frac{\partial}{\partial a_{ij}} g(A).$$

Comme U est un voisinage ouvert de A, il existe un voisinage ouvert $V \in \mathcal{V}(0)$ de zéro tel que

$$(\forall \varepsilon \in V)(\forall 1 \leq i, j \leq n)(A + \varepsilon M_{ij} \in U).$$

Donc

$$\forall \varepsilon \in V, \exists v(\varepsilon) \neq 0_{\mathbb{C}^n}, (A + \varepsilon M_{ij} - g(A + \varepsilon M_{ij})I)v(\varepsilon) = 0.$$

On dérive par rapport à ε

$$(M_{ij} - \frac{d}{d\varepsilon} g(A + \varepsilon M_{ij})I)v(\varepsilon) + (A + \varepsilon M_{ij} - g(A + \varepsilon M_{ij})I)v'(\varepsilon) = 0.$$

Au point $\varepsilon = 0$, on a

$$(M_{ij} - \frac{\partial}{\partial a_{ij}} g(A)I)v^* + (A - \lambda^* I)v'(0) = 0,$$

où v^* est un vecteur propre de A associé à la valeur propre λ^*. Soit u^* un vecteur propre à gauche de A associé à la valeur propre λ^*. Alors ${u^*}^H A = \lambda^* {u^*}^H$ et

$${u^*}^H M_{ij} v^* - {u^*}^H \frac{\partial}{\partial a_{ij}} g(A) v^* + ({u^*}^H A - \lambda^* {u^*}^H)v'(0) = 0,$$

Comme ${u^*}^H A - \lambda^* {u^*}^H = 0$, on a

$$\frac{\partial}{\partial a_{ij}} g(A) = \frac{{u^*}^H M_{ij} v^*}{{u^*}^H v^*}.$$

Ainsi

$$\nabla g(A) = \frac{1}{{u^*}^H v^*} \left({u^*}^H M_{ij} v^* \right) = \frac{1}{{u^*}^H v^*} \left(\bar{u^*}^i v^{*j} \right).$$

Par abus de langage, on peut écrire λ^* en lieu et place de g.

A.1. Abscisse spectrale

Nous nous intéressons à la fonction *abscisse spectrale*

$$\begin{aligned} \alpha : E &\to \mathbb{R} \\ M &\mapsto \text{Re}\,(\lambda_1(M)) \end{aligned}$$

où $\text{Re}\,(\lambda_1(M))$ est la partie réelle de la plus grande valeur propre de M. On se donne $A \in E$ tel que $\lambda_1(A)$ est simple et on cherche à calculer $\nabla\alpha(A) \in E$. On note de manière abusive λ_1 la fonction implicite définie dans un voisinage V de A, qui à ΔA

tel que $A + \Delta A \in V$ associe la plus grande valeur propre de $A + \Delta A$. On a donc pour $\Delta A = (\Delta a_{ij})$ suffisamment petit

$$\lambda_1(A + \Delta A) = \lambda_1(A) + \langle \nabla \lambda_1(A), \Delta A \rangle + o(\|\Delta A\|),$$

avec

$$\langle \nabla \lambda_1(A), \Delta A \rangle = \mathrm{Tr}(\nabla \lambda_1(A)^\top \Delta A) = \frac{1}{{u_1}^H v_1} \sum_{1 \leq i,j \leq n} \bar{u_1}^i v_1{}^j \Delta a_{ij},$$

où u_1 et v_1 sont respectivement des vecteurs propres à gauche et à droite de A associé à $\lambda_1(A)$. Comme Re est une fonction $\mathbb{R}$-linéaire sur $\mathbb{C}$ et que $\mathrm{Re}\,(o\,(\|\Delta A\|)) = o\,(\|\Delta A\|)$ alors

$$\mathrm{Re}\,(\lambda_1(A + \Delta A)) - \mathrm{Re}\,(\lambda_1(A)) = \mathrm{Re}\,\langle \nabla \lambda_1(A), \Delta A \rangle + o\,(\|\Delta A\|)\,.$$

De plus ΔA étant à coefficients réels on a

$$\mathrm{Re}\,\langle \nabla \lambda_1(A), \Delta A \rangle = \langle \mathrm{Re}\,(\nabla \lambda_1(A))\,, \Delta A \rangle\,.$$

En conclusion

$$\alpha(A + \Delta A) - \alpha(A) = \langle \mathrm{Re}\,(\nabla \lambda_1(A))\,, \Delta A \rangle + o\,(\|\Delta A\|)\,,$$

ce qui nous permet d'affirmer que

$$\nabla \alpha(A) = \mathrm{Re}\,(\nabla \lambda_1(A)) = |u_1^H v_1|^{-2} \mathrm{Re}\left(u_1^H v_1\right) \mathrm{Re}\left(u_1 v_1^H\right).$$

Avant de clore cette annexe, considérons le cas où A dépend de manière différentiable de la variable $x \in \mathbb{R}^k$. Alors, la différentielle $(\alpha \mathrm{o} A)'(x)$ de l'application composée est égale à la composée $\alpha'(A(x)) \mathrm{o} A'(x)$ des différentielles. Mettons l'accent sur le fait que si l'on peut associer un gradient aux applications $(\alpha \mathrm{o} A)'(x)$ et $\alpha'(A(x))$, on ne le peut pas pour $A'(x)$! En effet celle-ci n'est pas une *forme* linéaire continue dans le sens où elle n'est pas à valeurs dans le corps $\mathbb{R}$ des scalaires de l'espace vectoriel $\mathbb{R}^k$. Par contre, on peut utiliser la notion d'adjoint d'une application linéaire pour caractériser le gradient de $\alpha \mathrm{o} A$ en x. On a pour tout $\Delta x \in \mathbb{R}^k$

$$\alpha'(A(x)) \mathrm{o} A'(x)\ (\Delta x) = \langle \nabla \alpha(A(x)), A'(x)(\Delta x) \rangle = \Delta x^\top A'(x)^* \nabla \alpha(A(x)),$$

où $A'(x)^* : E \to \mathbb{R}^k$ est l'adjoint de $A'(x)$. Ainsi

$$\nabla(\alpha \mathrm{o} A)(x) = A'(x)^* \nabla \alpha(A(x)).$$

D'un point de vue pratique, étant donné une application différentiable $A : x \to A(x)$, on calcule analytiquement le gradient de $\alpha \mathrm{o} A$ en posant le produit scalaire canonique sur E $\langle \nabla \alpha(A(x)), A'(x)(\Delta x) \rangle$ et en opérant de telle manière à isoler Δx dans un des arguments du produit scalaire canonique sur $\mathbb{R}^k$. L'autre argument est alors $\nabla(\alpha \mathrm{o} A)(x)$.

B. Sous-différentiel de la norme H_∞

On se propose ici de calculer le sous-différentiel de Clarke [16] de la norme H_∞

$$\|.\|_\infty : \mathcal{R} \longrightarrow \mathbb{R}_+, \quad G \longrightarrow \sup_{\omega \in \mathbb{R}_+} \sigma_1(G(j\omega)). \tag{B.1}$$

B.1. Notations et définitions

On note $\mathcal{R}$ l'espace vectoriel réel des fonctions G de la variable complexe s, rationnelles, à pôles dans le demi plan ouvert gauche, de la forme $G(s) = D+C(sI-A)^{-1}B$, où A, B, C, D sont des matrices à coefficients réels. On munit $\mathcal{R}$ de la norme H_∞ définie par (B.1). On note $\mathbb{H}^n$ l'ensemble des matrices hermitiennes de taille $n \in \mathrm{N}^*$ et on le munit du produit scalaire $A \bullet B \doteq \mathrm{Tr}(AB)$. On note $\mathbb{C}^n$ l'ensemble des n-uplets à coefficients complexes et on le munit du produit scalaire euclidien $< x, y > \doteq x^H y$ et de la norme induite $\|x\| = \sqrt{x^H x}$. Pour une matrice M à coefficients complexes, on note $\sigma_1(M)$ sa plus grande valeur singulière. Pour une matrice M carrée à coefficients complexes, on note $\lambda_1(M)$ sa plus grande valeur propre. Pour un nombre complexe $\mu \in \mathbb{C}$, $\bar{\mu}$ désigne son complexe conjugué. Pour un ensemble L, $\mathrm{co}\ L$ désigne l'enveloppe convexe de L.

Definition 4. *[16, p. 25-27] Soient E un espace de Banach, f une application de E dans $\mathbb{R}$, localement lipschitzienne au voisinage d'un point $x \in E$ donné, et soit v un autre élément de E. L'espace E^* désigne le dual topologique de E et $\langle ., . \rangle$ le crochet de dualité, i.e. $\forall y \in E$, $\forall \varphi \in E^*$, $\langle \varphi, y \rangle \doteq \varphi(y)$. Sous ces conditions la quantité*

$$f^o(x; v) \doteq \limsup_{y \to x, t \searrow 0} \frac{f(y + tv) - f(y)}{t}$$

existe et est bien définie. On l'appelle la dérivée directionnelle généralisée de f en x dans la direction v et constitue l'analogue non lisse de la dérivée directionnelle $f'(x; v)$. De même l'ensemble

$$\partial f(x) \doteq \{\zeta \in E^* : f^o(x; v) \geq \langle \zeta, v \rangle \ \textit{pour tout } v \textit{ dans } E\}$$

existe et est bien défini. On l'appelle le gradient généralisé de f en x ou encore le sous différentiel de Clarke de f en x et constitue l'analogue non lisse du gradient $\nabla f(x)$. Un élément $\zeta \in \partial f(x)$ s'appelle un sous-gradient de f en x.

Lorsque l'espace E de la Définition 4 est de Hilbert, on identifie E^* à E et $\langle .,. \rangle$ à un produit scalaire que l'on se donne sur E.

B.2. Position du problème

Le but recherché est d'appréhender et de comprendre le sous-différentiel de Clarke [16] de la norme H_∞

$$\|.\|_\infty : \mathcal{R} \longrightarrow \mathbb{R}_+, \quad G \longrightarrow \sup_{\omega \in \mathbb{R}_+} \sigma_1(G(j\omega)).$$

Une valeur singulière étant la racine carrée d'une valeur propre d'une matrice hermitienne, le calcul du sous-différentiel de la norme H_∞ passe par l'étude des propriétés variationnelles des valeurs propres de matrices hermitiennes. On calculera donc dans un premier temps $\partial \|G\|_\infty^2$ où

$$\|.\|_\infty^2 : \mathcal{R} \longrightarrow \mathbb{R}_+, \quad G \longrightarrow \sup_{\omega \in \mathbb{R}_+} \lambda_1(G(j\omega)G(j\omega)^H),$$

puis on appliquera la règle de la chaîne [16, p. 45]

$$\partial \|G\|_\infty = (2 \|G\|_\infty)^{-1} \partial \|G\|_\infty^2 .$$

Les propositions du calcul différentiel non lisse que nous allons appliquer dans la suite sont

Proposition 1. *[12, p. 267] Soient J un ensemble d'indices arbitraire, $(f_j)_{j \in J}$ une collection de fonctions convexes de $\mathbb{R}^n$ dans $\mathbb{R}$. Si*

$$f(x) \doteqdot \sup\{f_j(x) : j \in J\} < +\infty \text{ pour tout } x \in \mathbb{R}^n$$

alors f est convexe. Pour un x donné on appelle

$$\mathcal{J}(x) \doteqdot \{j \in J : f_j(x) = f(x)\}$$

l'ensemble des indices actifs en x. Si J est un ensemble compact dans un espace métrique, et que les fonctions $j \to f_j(x)$ sont semi-continues supérieurement pour tout $x \in \mathbb{R}^n$ alors

$$\partial f(x) = \{\cup \partial f_j(x) : j \in \mathcal{J}(x)\}.$$

Pour la proposition qui suit on a besoin de la notion de régularité au sens de Clarke [16, p. 39]. On dira qu'une fonction f d'un espace de Banach E dans $\mathbb{R}$ est régulière au sens de Clarke en $x \in E$ si pour tout $v \in E$ $f'(x; v)$ existe et est égale à $f^o(x; v)$.

Proposition 2. *[16, p. 47] Soit $(f_i)_{1 \leqslant i \leqslant p}$ une famille finie de fonctions définies sur un espace de Banach E, localement lipschitziennes et régulières au sens de Clarke en $x \in E$. Alors le maximum point par point $\max_i f_i$ est encore localement lipschitzien et régulier au sens de Clarke en $x \in E$, et*

$$\partial \left(\max_{1 \leqslant i \leqslant p} f_i \right)(x) = \operatorname{co} \{\cup \partial f_j(x) : j \in \mathcal{J}(x)\},$$

où $\mathcal{J}(x)$ désigne l'ensemble des indices j actifs en x, tels que $f_j(x) = \max_i f_i(x)$.

B.3. Sous-différentiel de $\lambda_1 : \mathbb{H}^n \to \mathbb{R}$

On expose ici quelques idées essentielles à la démonstration du résultat suivant. La fonction

$$\lambda_1 : \mathbb{H}^n \longrightarrow \mathbb{R}, \quad X \longrightarrow \lambda_1(X)$$

n'est pas différentiable en X si $\lambda_1(X)$ n'est pas de multiplicité 1 mais a la propriété d'être convexe sur $\mathbb{H}^n$ et donc de posséder un sous-différentiel de Clarke en tout $X \in \mathbb{H}^n$ [16]. Pour ce faire nous allons exprimer $\lambda_1(X)$ comme un maximum sur un ensemble convexe de fonctions linéaires en X. Soit donc $(v_i)_i$ une base orthonormée de vecteurs propres de X et soient λ_1,...,λ_n les valeurs propres correspondantes. La base $(v_i)_i$ existe toujours du fait que X est hermitienne. Tout vecteur $q \in \mathbb{C}^n$ se décompose alors dans la base $(v_i)_i$, $q = \sum_i \mu_i v_i$, $(\mu_i)_i \in \mathbb{C}^n$, $\sum_i \mu_i \bar{\mu}_i = \|q\|$. Ainsi

$$q^H X q = \sum_{i=1}^{n} \lambda_i \mu_i \bar{\mu}_i \leqslant \lambda_1 \|q\| ,$$

l'égalité ayant lieu lorsque q est un vecteur propre de X associé à λ_1 de norme 1. Ceci nous donne la caractérisation de λ_1 suivante

$$\begin{aligned} \lambda_1(X) &= \max\{q^H X q : q \in \mathbb{C}^n, \|q\| = 1\} \\ &= \max\{\text{Tr}(X q q^H) : q \in \mathbb{C}^n, \|q\| = 1\}, \end{aligned} \tag{B.2}$$

d'après la propriété de trace $q^H X q = \text{Tr}(q^H X q) = \text{Tr}(X q q^H)$. En remarquant que $\text{Tr}(X q q^H)$ est linéaire par rapport à $q q^H$, on peut réécrire le maximum (B.2) comme un maximum sur l'enveloppe convexe des $q q^H$ tels que $q^H q = 1$. Cette enveloppe convexe est égale à

$$\mathcal{C} = \{Z \in \mathbb{H}^n : Z \succeq 0,\ \text{Tr}(Z) = 1\} , \tag{B.3}$$

ce qui nous permet de conclure que

$$\lambda_1(X) = \max\{\text{Tr}(X Z) : Z \in \mathcal{C}\}. \tag{B.4}$$

L'établissement de (B.3) vient de ce que toute matrice hermitienne Z se décompose comme

$$Z = \sum_{i=1}^{n} \alpha_i q_i q_i^H, \tag{B.5}$$

où $(q_i)_i$ est une base orthonormée de vecteurs propres de Z et α_1,...,α_n sont les valeurs propres correspondantes. Pour le voir il suffit de faire le calcul,

$$Z q = \begin{pmatrix} q_1 & \cdots & q_n \end{pmatrix} \begin{pmatrix} \alpha_1 & & 0 \\ & \ddots & \\ 0 & & \alpha_n \end{pmatrix} \begin{pmatrix} q_1^H \\ \vdots \\ q_n^H \end{pmatrix} q = \sum_{i=1}^{n} \alpha_i q_i q_i^H q, \quad q \in \mathbb{C}^n.$$

Si de plus Z est positive et de trace 1 alors la combinaison (B.5) est convexe.

L'équation (B.4) est fondamentale et donne accès aux propriétés variationnelles de λ_1. Elle exprime de manière explicite le caractère convexe, en tant que maximum de fonctions linéaires, et à priori non lisse, en tant que maximum, de λ_1. Elle permet de plus d'accéder au sous-différentiel $\partial\lambda_1(X)$ en appliquant la Proposition 1 :

$$\begin{aligned}\partial\lambda_1(X) &= \text{co}\{\cup\{\nabla_X \text{Tr}(XZ)\} : Z \in \mathcal{C}, \text{Tr}(XZ) = \lambda_1(X)\}, \\ &= \text{co}\{Z \in \mathcal{C} : \text{Tr}(XZ) = \lambda_1(X)\},\end{aligned} \tag{B.6}$$

puisque $\nabla_X \text{Tr}(XZ) = Z$. Notons que l'application de la Proposition 1 se fait à travers l'assimilation de $\mathbb{H}^n$ à $\mathbb{R}^{n(n+1)}$ qui lui est isomorphe. La condition de semi-continuité supérieure des fonctions $Z \rightarrow$Tr(XZ) pour tout $X \in \mathbb{H}^n$ est bien vérifiée ainsi que la condition de compacité de $\mathcal{C}$ dans l'espace métrique $\mathbb{H}^n$. L'enveloppe convexe (B.6) est égale à (B.7). Pour le comprendre, prenons un Z dans $\mathcal{C}$ tel que $\text{Tr}(XZ) = \lambda_1(X)$ et décomposons Z suivant (B.5). Alors Tr(XZ)= $\sum_i \alpha_i q_i^H X q_i$= $\lambda_1(X)$. Comme pour tout i, $q_i^H X q_i \leqslant \lambda_1$ d'après (B.2) et (B.5) et que $\sum_i \alpha_i = 1$ d'après (B.3) et (B.5), on a alors pour tout i $q_i^H X q_i = \lambda_1$. A présent donnons nous un q_i et décomposons le dans la base $(v_j)_j$, soit $q_i = \sum_j \mu_{j,i} v_j$. Si $p \leqslant n$ est la multiplicité de $\lambda_1(X)$ alors

$$q_i^H X q_i = \sum_{j=1}^{n} \lambda_j \mu_{j,i} \bar{\mu}_{j,i} = \lambda_1 \sum_{j=1}^{p} |\mu_{j,i}|^2 + \sum_{j=p+1}^{n} \lambda_j |\mu_{j,i}|^2 = \lambda_1,$$

avec $\lambda_j < \lambda_1$ pour tout $j \geqslant p+1$. Par conséquent $\mu_{j,i} = 0$ pour tout $j \geqslant p+1$ et $\sum_{1 \leqslant j \leqslant p} |\mu_{j,i}|^2 = 1$. On en conclut que

$$Z = Q(X) Y Q(X)^H,$$

avec $Q(X)$ la matrice dont les colonnes forment une base orthonormée du sous espace propre de X associé à $\lambda_1(X)$ et

$$Y = \sum_{i=1}^{n} \alpha_i \begin{pmatrix} \mu_{1,i} \\ \vdots \\ \mu_{p,i} \end{pmatrix} \begin{pmatrix} \bar{\mu}_{1,i} & \cdots & \bar{\mu}_{p,i} \end{pmatrix}$$

hermitienne, positive et de trace 1. On laisse le soin au lecteur de terminer la démonstration qui nous amène à

$$\partial\lambda_1(X) = \{Q(X) Y Q(X)^H : Y \in \mathbb{H}^p, Y \succeq 0,\ \text{Tr}(Y) = 1\}. \tag{B.7}$$

On terminera cette section par la remarque suivante. La fonction λ_1 est différentiable en X si et seulement si $\partial\lambda_1(X)$ est réduit à un seul élément, son gradient, si et seulement si $\lambda_1(X)$ est de multiplicité 1. Dans ce cas là $\nabla\lambda_1(X) = uu^H$, où u est un vecteur propre de X associé à $\lambda_1(X)$ de norme 1.

B.3.1. Composition avec une application continûment différentiable

Plaçons nous dans le cadre où X dépend de manière continûment différentiable du k-uplets à coefficients réels $x \in \mathbb{R}^k$, $k \in \mathrm{N}^*$. Notons

$$f : \mathbb{R}^k \longrightarrow \mathbb{H}^n, \quad x \longrightarrow \lambda_1(X(x)),$$

l'application composée . Alors la règle de la chaîne [16, p. 45] nous dit que

$$\partial f(x) = X'(x)^* \partial \lambda_1(X(x)),$$

où $X'(x)^*$ est l'adjoint de la différentielle $X'(x)$ de X en x. Ce qui veut dire que pour tout sous-gradient $Z \in \partial\lambda_1(X(x))$, $X'(x)^* Z$ est un sous-gradient de f en x défini par

$$\forall y \in \mathbb{R}^k \quad < X'(x)^* Z, y > \doteq Z \bullet X'(x)(y).$$

B.4. Sous différentiel de $\|.\|_\infty^2 : \mathcal{R} \to \mathbb{R}_+$

La fonction $\|.\|_\infty^2$ est un supremum sur l'ensemble $\mathbb{R}_+$ de fonctions convexes sur $\mathcal{R}$. En effet pour tout $\omega \in \mathbb{R}_+$, la branche du supremum d'indice ω

$$f_\omega : \mathcal{R} \longrightarrow \sigma_1(G(j\omega))^2$$

est convexe d'après les propriétés de convexité et de positivité de σ_1 et les propriétés de convexité et de croissance sur $\mathbb{R}_+$ de la fonction carré. Toutefois on ne peut pas appliquer la Proposition 1 puisque l'ensemble $\mathbb{R}_+$ des indices des branches du supremum n'est pas compact dans l'espace métrique $\mathbb{R}_+$. Ce problème est résolu par le fait que l'ensemble

$$\mathcal{I}(G) \doteq \{\omega \in \mathbb{R}_+ : \sigma_1(G(j\omega)) = \|G\|_\infty\}$$

des fréquences actives de $\|.\|_\infty^2$ en G a la particularité d'être fini dès lors que G est dynamique (la matrice dynamique de sa représentation d'états est non vide) [38]. Dans ce cas là, on peut considérer que $\|.\|_\infty^2$ se comporte comme un maximum sur un ensemble fini $\mathcal{I}(G)$ de fonctions convexes dans un voisinage ouvert de G. Une fonction convexe étant localement lipschitzienne et régulière au sens de Clarke, on peut faire appel à la Proposition 2 pour établir que

$$\partial \|G\|_\infty^2 = \mathrm{co}\left\{\cup \partial\lambda_1(G(j\omega)G(j\omega)^H) : \omega \in \mathcal{I}(G)\right\}. \tag{B.8}$$

On peut également faire appel à la proposition 1 puisque un ensemble fini dans un espace métrique est toujours compact et que les fonctions $\omega \to f_\omega(G)$ sont continues pour tout $G \in \mathcal{R}$.

Pour calculer de manière précise et efficace la norme H_∞ de G ainsi que l'ensemble $\mathcal{I}(G)$ on utilise un algorithme de bissection basé sur un calcul hamiltonien [19, 20]. D'après (B.7), on a pour tout $\omega \in \mathcal{I}$

$$\partial\lambda_1(G(j\omega)G(j\omega)^H) = \{Q_\omega Y_\omega Q_\omega^H : Y_\omega = Y_\omega^H, Y_\omega \succeq 0, \mathrm{Tr}(Y_\omega) = 1\}, \tag{B.9}$$

avec Q_ω la matrice dont les colonnes forment une base orthonormée du sous-espace propre de $G(j\omega)\ G(j\omega)^H$ associé à sa plus grande valeur propre. On montre que l'enveloppe convexe de l'union des ensembles de la forme (B.9) avec $\omega \in \mathcal{I}(G)$, s'écrit

$$\partial \|G\|_\infty^2 = \left\{ \Phi_Y, Y = (Y_\omega)_{\omega\in\mathcal{I}(G)}, Y_\omega = Y_\omega^H, Y_\omega \succeq 0, \sum_{\omega\in\mathcal{I}(G)} \mathrm{Tr}(Y_\omega) = 1 \right\}, \tag{B.10}$$

où

$$\Phi_Y = \sum_{\omega\in\mathcal{I}(G)} Q_\omega Y_\omega {Q_\omega}^H.$$

L'ensemble (B.10) est trivialement convexe et les ensembles de la forme (B.9) sont trivialement inclus dans (B.10). Leur enveloppe convexe est donc incluse dans (B.10). L'inclusion est réciproque puisque Φ_Y se réécrit comme la combinaison convexe

$$\Phi_Y = \sum_{\substack{\omega\in\mathcal{I}(G)\\ \alpha_\omega\neq 0}} \alpha_\omega\ Q_\omega \left(\alpha_\omega^{-1} Y_\omega\right) {Q_\omega}^H, \text{ où } \alpha_\omega = \mathrm{Tr}(Y_\omega).$$

De la même manière que dans la section B.3.1, si l'on compose $\|.\|_\infty^2$ avec une application $x \to G(x)$ continûment différentiable, le sous-différentiel de l'application composée $\|.\|_\infty^2 \circ G$ en x est égal à $G'(x)^* \partial \|G(x)\|_\infty^2$.

C. Compléments de la section 1.2

C.1. Le gabarit S_d et la lecture des spécifications

Le gabarit fréquentiel S_d sur le transfert $w_1 \to y$ est :

$$\begin{aligned} S_d(s) &= \frac{s+0.1}{s+10}, \\ &= 1 - 9.9\frac{1}{s+10} \quad \text{(décomposition en éléments simples).} \end{aligned}$$

Ce gabarit modélise les spécifications de performance qui sont :
– une bande passante de $10\,rad/s$,
– une erreur statique inférieure à 1%.
On se place dans le cas où la réponse fréquentielle $T_{w_1 \to y}$ colle au gabarit S_d, i.e.

$$Y(s) = S_d(s)W_1(s),$$

et on regarde ce que cela implique sur le comportement fréquentiel et temporel du transfert $w_1 \to y$. Le noyau de convolution vaut

$$s_d(t) = \delta(t) - 9.9e^{-10t} \quad \text{(original de } S_d\text{)},$$

et la réponse temporelle vaut

$$y(t) = s_d * w_1\,(t) = w_1(t) - 9.9\int_0^t e^{-10(t-\tau)}w_1(\tau)d\tau. \tag{C.1}$$

La réponse indicielle est donnée par (C.1) où $w_1(t) = 1\ \forall t \geq 0$:

$$y(t) = 1 - 0.99\left(1 - e^{-10t}\right) = w_1(t) - y_G(t).$$

Comme $y_G(t) \to 0.99$ quand $t \to +\infty$, l'erreur statique vaut 1%. En ce qui concerne la bande passante de S_d, on rappelle que c'est l'intervalle de fréquences pour lequel

$$|S_d(j\omega)| \geq \frac{1}{\sqrt{2}}.$$

Comme $20\log_{10}(1/\sqrt{2}) \simeq -3$, on dit également que la bande passante est l'ensemble des fréquences où le gain est supérieur à $-3dB$. Comme S_d est un système d'ordre 1, sa bande passante se présente sous la forme $[\omega_c, +\infty[$ où ω_c est tel que $|S_d(j\omega_c)| = 1/\sqrt{2}$.

On trouve $\omega_c \simeq 10$. Le nombre 10 définit donc à la fois la constante de temps de la réponse indicielle, et la bande passante. Prêtons attention tout de même au fait que $\omega_c \simeq 10$ parce que 0.1 est relativement petit.

$$|S_d(j\omega_c)| = \frac{1}{\sqrt{2}} \Leftrightarrow |S_d(j\omega_c)|^2 = \frac{1}{2} \Leftrightarrow \frac{\omega^2 + 0.1^2}{\omega^2 + 10^2} = \frac{1}{2} \Leftrightarrow \omega^2 = 100 - 2 * 0.01$$

C.2. Inversion du modèle dans le correcteur

On appelle

$$N_G(s)N_K(s) + D_G(s)D_K(s) = 0. \tag{C.2}$$

l'équation caractéristique du système asservi Figure 1.1.

C.2.1. $(1 + GK)^{-1}$

Lorsque l'on fait tendre les paramètres ϵ et $\delta\gamma$ vers 0, le contrôleur (1.6) tend vers

$$K_0(s) = \frac{9.9(s^2 + 0.01s + 1)}{s + 0.1}$$

K_0 résout le problème (1.3) puisque

$$T_{w_1 \to y}(K_0) = \frac{1}{1 + GK_0} = \frac{s + 0.1}{s + 10} = S_d$$

Ainsi $(1+GK_0)^{-1}$ est un système d'ordre 1 ! Et a pour seul pôle $s = -10$. Le problème d'annulation pôle/zéro se présentant ici

$$\frac{1}{1 + GK_0} = \frac{\cancel{D_G} D_{K_0}}{\cancel{D_G} D_{K_0} + N_G \cancel{N_{K_0}}}$$

fait disparaître les pôles de $G(s)$ qui font partie des 3 solutions de l'équation caractéristique (C.2)

$$(s^2 + 0.01s + 1)(s + 0.1) + 9.9(s^2 + 0.01s + 1) = (s^2 + 0.01s + 1)(s + 10) = 0.$$

C.2.2. $G(1 + KG)^{-1}$

Dans le cas du transfert $w_2 \to y_G$, il n'y pas d'annulation pôle/zéro dans la fonction de transfert

$$\frac{G}{1 + K_0 G} = \frac{\frac{N_G}{D_G}}{1 + \frac{N_{K_0} N_G}{D_{K_0} D_G}} = \frac{N_G D_{K_0}}{D_G D_{K_0} + N_G N_{K_0}}.$$

Donc les pôles de $G(1+KG)^{-1}$ sont les solutions de l'équation caractéristique (C.2), i.e. le pôle $s=-10$ et les pôles mal amortis $s=\alpha$ et $s=\bar{\alpha}$ de $G(s)$! La réponse impulsionnelle de $G(1+KG)^{-1}$ est catastrophique puisque le mode oscillant du système à contrôler évolue avec son comportement naturel. Si on appelle $g(t)$ l'original de $G(1+KG)^{-1}$ alors

$$g(t) = 2e^{at}\left(a_1\cos(bt)+b_1\sin(bt)\right)+Ce^{-10t},$$

où a et b sont les parties réelle et imaginaire de α et a_1, b_1, C sont des constantes dans $\mathbb{R}$.

C.3. L'intégrateur : un outil de performance et de robustesse.

On se place dans la configuration de la Figure 1.1.

C.3.1. Annulation de l'erreur statique

On souhaite annuler l'erreur statique en régime permanent, i.e. on souhaite que pour une consigne $w_1(t)=1\ \forall t\geq 0$, l'erreur de poursuite $y(t)$ tende vers 0 lorsque t tend vers $+\infty$. Le théorème de la valeur finale nous dit que

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s).$$

Comme la transformée de Laplace de 1 vaut s^{-1} on a

$$\lim_{s\to 0} sY(s) = \lim_{s\to 0} s(1+G(s)K(s))^{-1}s^{-1} = (1+G(0)K(0))^{-1}.$$

Enfin comme

$$\frac{1}{1+GK} = \frac{\text{Den}_G\text{Den}_K}{\text{Den}_G\text{Den}_K+\text{Num}_G\text{Num}_K},$$

on a $\lim_{t\to\infty} y(t) = 0$ si et seulement si il n'y a pas d'annulation pôle/zéro dans le produit GK et que soit $\text{Den}_G(0)=0$, i.e. G possède un intégrateur, soit $\text{Den}_K(0)=0$, i.e. K possède un intégrateur.

C.3.2. Robustesse à l'incertitude sur le gain statique

D'après ce qui précède l'erreur statique est nulle si on met un intégrateur dans K et ce quelque soit la valeur du gain statique $G(0)$.

D. Transformée linéaire fractionnaire $F_l(P, K)$

On se place dans la configuration du problème de commande standard 1.2. On définit

$$P(s) = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \text{ avec } P_{ij}(s) = D_{ij} + C_i(sI - A)^{-1}B_j,$$

et

$$K(s) = A_K + B_K(sI - A_K)^{-1}B_K.$$

La matrice de transfert $P(s)$ a été partitionnée en quatre blocs associés aux quatre transferts $w \to z$, $w \to y$, $u \to z$, $u \to y$.

Definition 5. *[3, p. 8] La transformée linéaire fractionnaire de P et de K est la fonction de transfert en boucle fermée entre w et z :*

$$F_l(P, K) \doteqdot P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}.$$

D.1. Cas statique

On se place dans le cas où $K = D_K$, i.e. K ne possède pas de variable d'état. Sous cette hypothèse, la fonction de transfert en boucle fermée se réécrit

$$F_l(P, K)(s) = D(K) + C(K)(sI - A(K))^{-1}B(K) \tag{D.1}$$

avec

$$A(K) = A + B_2K(I - D_{22}K)^{-1}C_2, \quad B(K) = B_1 + B_2K(I - D_{22}K)^{-1}D_{21}$$
$$C(K) = C_1 + D_{12}K(I - D_{22}K)^{-1}C_2, \quad D(K) = D_{11} + D_{12}K(I - D_{22}K)^{-1}D_{21}.$$

Les matrices $A(K)$, $B(K)$, $C(K)$, $D(K)$ sont bien définies si $(I - D_{22}D_K)^{-1}$ existe. On dira donc que le problème de commande standard est *bien posé* si $(I - D_{22}D_K)$ est inversible. La régularité de $F_l(P, K)$ en tant que fonction de K est entièrement basée sur le résultat fondamental suivant. La fonction inverse

$$\begin{aligned} f : \mathcal{M}_n(\mathbb{C}) &\longrightarrow \mathcal{M}_n(\mathbb{C}) \\ M &\longrightarrow f(M) = M^{-1} \end{aligned}$$

est $\mathcal{C}^\infty$ sur son domaine de définition. En effet lorsque M est inversible, $M+\Delta M = M(I-M^{-1}\Delta M)$. Donc pour ΔM suffisamment petit ($\|M^{-1}\Delta M\|<1$) on a :

$$(M+\Delta M)^{-1} = (I-M^{-1}\Delta M)^{-1}M^{-1} = \left(\sum_{n=0}^{\infty}(M^{-1}\Delta M)^n\right)M^{-1}.$$

En tant que fonctions rationnelles en la variable K, $A(\cdot)$, $B(\cdot)$, $C(\cdot)$, $D(\cdot)$ sont $\mathcal{C}^\infty$ sur leur ensemble de définition, i.e. l'ensemble des matrices K telles que $(I-D_{22}K)$ est inversible. En posant $h_s : M \to sI-M$, on a $(sI-A(K))^{-1} = f \circ h \circ A\ (K)$. La fonction f est régulière en $h_s(A(K))$ si et seulement si $\det h_s(A(K)) \neq 0$ si et seulement si s n'est pas une valeur propre de $A(K)$. En conséquence, $F_l(P,\cdot)(s)$ est $\mathcal{C}^\infty$ en tant que composée de fonctions $\mathcal{C}^\infty$ sur l'ensemble

$$\mathcal{D}_{F_l(P,\cdot)(s)} = \{K \text{ tel que } \det(I-D_{22}K)\neq 0 \text{ et } s\notin \mathrm{Sp}(A(K))\}\,, \tag{D.2}$$

où $\mathrm{Sp}(A(K))$ est l'ensemble des valeurs propres de $A(K)$.

D.2. Cas dynamique

Le cas dynamique, i.e. A_K est non vide, se ramène au cas statique par augmentation des matrices de la représentation d'état de P. Si k désigne le nombre de lignes ou de colonnes de A_K alors

$$\begin{aligned}
&K \to \begin{pmatrix} A_K & B_K \\ C_K & D_K \end{pmatrix}, \quad A \to \begin{pmatrix} A & 0 \\ 0 & 0_k \end{pmatrix}, \quad B_1 \to \begin{pmatrix} B_1 \\ 0 \end{pmatrix}, \\
&C_1 \to \begin{pmatrix} C_1 & 0 \end{pmatrix}, \quad B_2 \to \begin{pmatrix} 0 & B_2 \\ I_k & 0 \end{pmatrix}, \quad C_2 \to \begin{pmatrix} 0 & I_k \\ C_2 & 0 \end{pmatrix}, \\
&D_{12} \to \begin{pmatrix} 0 & D_{12} \end{pmatrix}, \quad D_{21} \to \begin{pmatrix} 0 \\ D_{21} \end{pmatrix}, \quad D_{22} \to \begin{pmatrix} 0_k & 0 \\ 0 & D_{22} \end{pmatrix}.
\end{aligned} \tag{D.3}$$

E. Comparaison de l'algorithme 1 avec Hifoo et Hinfstruct sur des problèmes de Compleib

Dans cette annexe, nous considérons des tests issus de la librairie COnstrained Matrix-optimization Problem LIBrary $Compl_eib$ [48]. L'algorithme 1, Hifoo et Hinfstruct sont comparés avec leurs paramètres par défaut. Des tests comparatifs entre HIFOO et HINFSTRUCT sont fournis par Pierre Apkarian [49, 50] et Daniel Ankelhed [51].

Chaque code est utilisé en mode par défaut avec 3 points initiaux aléatoires.

Les tests correspondants à l'algorithme 1 ont été réalisé sur un Toshiba Portégé M800-10D, Intel Core 2, 2Ghz et 3Gb de mémoire avec Window Vista et Matlab R2012a. Les tests correspondants à HIFOO et HINFSTRUCT proviennent de *Comparison of HINFSTRUCT Matlab Robust Control Toolbox R2010b and HIFOO 3.0 with HANSO 2.0* [49]. Ces tests ont été réalisés avec un PC avec Winwos XP 32, Intel Core 2, 2Ghz et 4Gb de mémoire. Les deux PCs ont des capacités de calculs proches.

Les premières colonnes des tableaux suivants donnent l'acronyme utilisé et permettent d'identifier le modèle dans $Compl_eib$. Les secondes colonnes précisent l'ordre du système à commander, l'ordre du contrôleur et le nombre de variables de décision. Les colonnes 3, 4 et 5 fournissent les temps de calculs de respectivement l'algorithme 1, HIFOO et HINFSTRUCT. Les colonnes 4 et 5 sont extraites de [49]. Les colonnes 6, 7 et 8 fournissent les normes H_∞ de respectivement l'algorithme 1, HIFOO et HINFSTRUCT. Les colonnes 7 et 8 sont extraites de [49].

Le code de l'algorithme 1 n'est pas du tout optimisé du point de vue temps de calcul et il est encore en développement.

	order P/K/	cpu			H_∞ norm		
model	dim x	Algo. 1	HIFOO	HINFS.	Algo. 1	HIFOO	HINFS.
AC1	5/0/9	34.52	7.09	5.70	0.01	**0.00**	**0.00**
AC1	5/2/25	32.83	34.48	8.48	0.02	**0.00**	**0.00**
AC2	5/0/9	29.3	4.23	1.16	**0.11**	**0.11**	**0.11**
AC2	5/2/25	21.13	0.30	0.78	**0.11**	**0.11**	**0.11**
AC3	5/0/8	71.94	2.11	1.72	3.63	3.67	**3.57**
AC3	5/2/24	74.87	8.05	2.83	3.12	3.21	**2.98**
AC4	4/0/2	3.41	1.53	1.17	**0.94**	**0.94**	**0.94**
AC4	4/2/12	39.27	4.67	2.03	0.59	**0.56**	**0.56**
AC5	4/0/4	NaN	0.41	1.36	Inf	674.92	**664.97**
AC5	4/2/16	NaN	4.36	1.88	Inf	673.61	**665.10**
AC6	7/0/8	26.02	5.67	2.02	**4.11**	**4.11**	**4.11**
AC6	7/2/24	73.78	60.39	5.28	3.63	3.74	**3.52**
AC6	7/4/48	121.69	44.64	8.27	3.73	3.57	**3.45**
AC7	9/0/2	27.96	0.38	0.69	**0.07**	**0.07**	**0.07**
AC7	9/3/20	81.78	12.91	3.89	0.06	**0.04**	**0.04**
AC7	9/5/42	96.72	52.81	5.42	0.06	**0.04**	**0.04**
AC8	9/0/5	13.41	5.89	1.05	**2.01**	**2.01**	**2.01**
AC8	9/3/32	80.46	33.66	6.77	**1.62**	1.63	1.63
AC8	9/5/60	58.10	25.53	5.48	1.63	**1.62**	**1.62**
AC9	10/0/20	40.71	35.52	4.50	1.02	1.01	**1.00**
AC9	10/3/56	33.08	32.25	9.39	1.05	1.02	**1.00**
AC9	10/5/90	158.95	68.89	16.20	1.03	1.03	**1.00**
AC10	55/0/4	NaN	NaN	20.20	Inf	Inf	**13.24**
AC10	55/3/25	NaN	58.03	110.03	Inf	184.00	**6.49**
AC10	55/10/144	NaN	NaN	317.20	Inf	Inf	**7.88**
AC11	5/0/8	27.07	5.42	2.20	3.55	3.56	**2.81**
AC11	5/2/24	75.83	28.66	1.97	2.85	2.82	**2.81**
AC12	4/0/12	92.91	7.58	1.56	**0.32**	**0.32**	**0.32**
AC12	4/2/30	65.77	10.53	6.31	0.32	0.31	**0.02**
AC13	28/0/12	145.64	482.23	12.33	163.61	163.60	**163.30**
AC13	28/2/30	118.18	537.19	14.09	182.4	**156.30**	**156.30**
AC14	40/0/12	294.52	371.17	9.61	101.79	**101.76**	101.86
AC14	40/2/30	66.33	24.22	4.14	**100.00**	**100.00**	**100.00**
AC14	40/10/182	146.93	114.20	16.11	**100.00**	**100.00**	**100.00**
AC15	4/0/6	74.61	2.73	1.23	**15.17**	16.01	15.19
AC15	4/2/20	41.56	5.80	1.77	14.94	**14.86**	14.87
AC16	4/0/8	93.86	1.48	1.11	14.87	15.17	**14.86**
AC16	4/2/24	125.74	9.25	1.31	14.88	**14.86**	**14.86**
AC17	4/0/2	7.16	0.08	0.44	**6.61**	**6.61**	**6.61**
AC17	4/2/12	3.65	0.09	0.38	**6.61**	**6.61**	**6.61**
AC18	10/0/4	27.9	NaN	2.50	11.46	Inf	**10.70**
AC18	10/2/16	29	67.25	2.33	**7.18**	7.58	7.66
AC18	10/5/49	91.53	94.66	9.95	6.63	14.13	**6.11**

TABLE E.1.: Aicraft models

	order P/K/	cpu			H_∞ norm		
model	dim x	Algo. 1	HIFOO	HINFS.	Algo. 1	HIFOO	HINFS.
HE1	4/0/2	14.01	0.28	0.88	**0.15**	**0.15**	**0.15**
HE1	4/2/12	28.70	11.83	2.55	**0.08**	**0.08**	**0.08**
HE2	4/0/4	17.18	0.81	0.94	4.26	4.00	**3.90**
HE2	4/2/16	93.01	25.86	2.44	2.46	**2.43**	**2.43**
HE3	8/0/24	95.67	7.67	2.86	0.85	0.84	**0.81**
HE3	8/2/48	105.03	97.98	3.44	0.81	0.81	**0.80**
HE4	8/0/24	55.32	32.52	2.98	24.08	**22.84**	**22.84**
HE4	8/2/48	56.01	14.41	3.81	26.97	**22.84**	**22.84**
HE5	8/0/8	46.11	7.88	0.95	**8.90**	**8.90**	13.07
HE5	8/2/24	93.49	50.88	1.97	**2.16**	**2.16**	2.17
HE6	20/0/24	123.48	95.19	7.16	**192.35**	394.30	192.42
HE6	20/2/48	184.08	188.41	51.94	43.55	191.08	**15.77**
HE6	20/8/168	388.06	56.45	82.06	20.74	394.64	**2.50**
HE7	20/0/24	221.50	88.98	14.75	192.46	**192.39**	192.43
HE7	20/2/48	255.45	199.08	48.19	36.32	150.56	**25.66**
HE7	20/8/168	90.16	28.00	91.03	83.26	114.46	**2.85**
DIS1	8/0/16	86.51	7.81	1.42	4.18	**4.16**	4.17
DIS1	8/2/36	90.19	8.95	1.31	4.18	**4.16**	4.17
DIS1	8/4/64	84.59	129.31	3.11	4.19	**4.16**	4.17
DIS2	3/0/4	57.37	0.78	0.66	1.05	**1.03**	1.05
DIS2	3/2/16	77.83	10.86	1.53	1.00	**0.95**	**0.95**
DIS3	6/0/16	77.25	9.61	3.14	1.12	1.10	**1.06**
DIS3	6/2/36	109.87	15.20	2.59	1.11	**1.05**	**1.05**
DIS4	6/0/24	120.89	9.06	1.48	0.78	**0.74**	**0.74**
DIS4	6/2/48	126.71	7.05	2.09	0.78	0.74	**0.73**
DIS5	4/0/4	NaN	1.84	1.13	Inf	**1035.53**	**1035.53**
DIS5	4/2/16	NaN	33.03	2.64	Inf	678.05	**667.60**
JE1	30/0/15	127.51	236.72	39.83	25.5	23.45	**10.16**
JE1	30/2/35	65.11	404.41	53.59	46.60	16.08	**4.05**
JE1	30/8/143	102.21	35.92	107.31	88.33	98.05	**3.94**
JE2	21/0/9	51.06	152.48	14.08	543.04	**183.35**	183.57
JE2	21/2/25	56.83	309.44	38.72	1155.43	82.04	**73.65**
JE2	21/8/121	69.27	77.52	96.22	1088.13	258.62	**52.82**
JE3	24/0/18	97.98	134.69	11.09	**5.10**	**5.10**	**5.10**
JE3	24/2/40	155.43	304.47	34.05	2.92	2.94	**2.90**
JE3	24/8/154	83	124.98	25.95	2.92	3.22	**2.89**

TABLE E.2.: Aicraft and helicopter models

	order P/K/	cpu			H_∞ norm		
model	dim x	Algo. 1	HIFOO	HINFS.	Algo. 1	HIFOO	HINFS.
REA1	4/0/6	20.09	0.27	0.91	**0.87**	0.89	**0.87**
REA1	4/2/20	75.17	0.34	0.59	**0.86**	**0.86**	**0.86**
REA2	4/0/4	106.9	0.28	1.22	**1.15**	**1.15**	**1.15**
REA2	4/2/16	68.23	0.53	0.61	**1.13**	**1.13**	**1.13**
REA3	12/0/3	3.33	0.48	1.06	**74.25**	**74.25**	**74.25**
REA3	12/2/15	NaN	0.34	1.28	Inf	**74.25**	**74.25**
REA3	12/5/48	5.06	1.17	1.52	**74.25**	**74.25**	**74.25**
WEC1	10/0/12	75.35	52.79	2.84	7.35	6.39	**4.05**
WEC1	10/2/30	63.87	1.66	1.17	**3.64**	**3.64**	**3.64**
WEC1	10/4/56	82.49	107.63	2.61	3.64	**4.34**	3.64
WEC2	10/0/12	122.46	21.17	3.41	4.29	**4.25**	**4.25**
WEC2	10/2/30	131.28	15.38	1.28	**3.60**	**3.60**	**3.60**
WEC2	10/4/56	70.89	7.83	2.97	**3.60**	**3.60**	**3.60**
TG1	10/0/4	9.89	9.34	1.47	17.38	**12.85**	**12.85**
TG1	10/2/16	43.18	28.11	1.94	**3.47**	**3.47**	**3.47**
TG1	10/4/36	38.70	22.42	4.58	**3.47**	**3.47**	**3.47**
AGS	12/0/4	60.06	1.42	0.97	9.28	**8.17**	**8.17**
AGS	12/2/16	107.27	4.41	2.86	10.53	**8.17**	**8.17**
AGS	12/4/36	150.47	3.61	5.45	10.44	**8.17**	**8.17**
PAS	5/0/3	NaN	NaN	2.63	Inf	Inf	**0.00**
PAS	5/2/15	NaN	NaN	5.89	Inf	Inf	**0.00**
TMD	6/0/8	22.52	2.80	1.88	2.59	2.56	**2.52**
TMD	6/2/24	83.37	178.89	3.84	2.49	2.27	**2.15**
TMD	6/2/48	94.08	207.28	6.88	**2.14**	2.28	2.15
CM1	20/0/2	2.69	1.53	1.75	**0.82**	**0.82**	**0.82**
CM1	20/2/12	6.63	4.16	3.61	**0.82**	**0.82**	**0.82**
ROC1	9/1/9	33.57	12.84	1.55	**1.24**	**1.24**	**1.24**
ROC1	9/3/25	94.21	19.47	2.45	1.20	1.20	**1.19**
ROC1	9/5/49	118.69	50.64	4.11	1.20	**1.16**	1.19
ROC2	10/1/12	114.57	33.72	3.91	0.06	**0.05**	**0.05**
ROC2	10/3/30	45.25	18.66	5.77	2.62	0.05	**0.04**
ROC2	9/5/49	240.17	94.80	18.28	0.05	0.05	**0.04**
ROC3	11/1/25	NaN	NaN	15.41	Inf	Inf	**263.73**
ROC3	11/3/49	NaN	25.95	25.94	Inf	68087.54	**284.01**
ROC3	11/5/81	NaN	10.72	55.25	Inf	49001.69	**238.01**
ROC4	9/1/9	31.83	33.69	1.78	2049.63	**302.21**	**302.21**
ROC4	9/3/25	13.37	23.33	4.89	525.48	254.38	**155.33**
ROC4	9/5/49	NaN	24.56	8.55	Inf	232.01	**220.40**
ROC5	7/1/24	41.81	9.64	11.69	0.02	**0.00**	**0.00**
ROC5	7/3/48	42.39	9.19	16.03	0.02	**0.00**	**0.00**
ROC5	7/5/80	54.35	12.89	36.91	0.02	**0.00**	**0.00**

TABLE E.3.: Miscellaneous

	order P/K/	cpu			H_∞ norm		
model	dim x	Algo. 1	HIFOO	HINFS.	Algo. 1	HIFOO	HINFS.
NN1	3/0/2	12.10	NaN	NaN	**13.94**	Inf	Inf
NN1	3/2/12	NaN	1.64	1.83	Inf	14.43	**13.13**
NN2	2/0/1	2.77	0.25	0.69	**2.22**	**2.22**	**2.22**
NN2	2/2/9	13.97	0.73	0.77	1.77	**1.76**	**1.76**
NN3	4/0/1	NaN	NaN	NaN	Inf	Inf	Inf
NN3	4/2/9	NaN	NaN	2.98	Inf	Inf	**20.47**
NN4	4/0/6	86.64	2.17	1.27	1.38	**1.36**	1.37
NN4	4/2/20	73.91	5.00	1.84	1.31	**1.29**	**1.29**
NN5	7/0/2	5.74	0.78	0.66	326.37	**266.54**	**266.54**
NN5	7/2/12	24.74	7.19	1.73	**238.43**	239.19	241.18
NN6	9/0/4	NaN	20.34	1.59	Inf	**5602.70**	5611.29
NN6	9/2/18	NaN	84.30	7.14	Inf	**264.92**	270.03
NN7	9/0/4	NaN	4.92	0.95	Inf	**74.08**	**74.08**
NN7	9/3/38	NaN	81.05	6.78	Inf	42.54	**37.48**
NN8	3/0/4	52.78	1.58	0.80	**2.89**	**2.89**	**2.89**
NN8	3/3/25	71.94	2.16	1.42	2.41	2.37	**2.36**
NN9	5/0/6	43.84	5.91	1.34	**28.68**	29.21	28.80
NN9	5/3/30	NaN	48.67	6.02	Inf	14.06	**13.65**
NN11	16/0/15	29.69	0.23	1.38	0.10	0.10	**0.09**
NN11	16/2/35	42.24	133.14	11.66	0.13	0.04	**0.02**
NN12	6/0/4	NaN	NaN	NaN	Inf	Inf	Inf
NN12	6/2/16	NaN	56.45	7.23	Inf	**13.37**	29.28
NN13	6/0/4	22.99	19.25	2.28	14.35	**14.06**	**14.06**
NN13	6/2/16	53.90	33.94	3.88	11.01	11.65	**10.47**
NN14	6/0/4	58.20	10.78	1.27	22.72	**17.48**	**17.48**
NN14	6/2/16	34.76	53.36	5.20	9.89	9.79	**9.47**
NN15	3/0/4	12.82	0.08	1.20	**0.10**	0.11	**0.10**
NN15	3/2/16	14.24	0.39	1.11	**0.10**	**0.10**	**0.10**
NN16	8/0/16	104.18	25.03	1.91	**0.96**	**0.96**	**0.96**
NN16	8/2/36	322.76	19.50	2.56	1.22	**0.96**	**0.96**
NN17	3/0/2	3.39	0.72	0.47	12.33	**11.22**	**11.22**
NN17	3/0/12	25.17	3.09	1.20	5.43	5.28	**5.15**

TABLE E.4.: Miscellaneous

	order P/K/	cpu			H_∞ norm		
model	dim x	Algo. 1	HIFOO	HINFS.	Algo. 1	HIFOO	HINFS.
TF1	$7/0/8$	37.04	14.58	1.89	0.39	0.38	**0.32**
TF1	$7/2/24$	12.83	16.92	5.33	0.30	0.26	**0.25**
TF1	$7/4/48$	64.28	46.52	6.70	0.27	**0.25**	**0.25**
TF2	$7/0/6$	3.00	0.05	0.22	**5200.00**	**5200.00**	**5200.00**
TF2	$7/2/20$	3.05	0.05	0.31	**5200.00**	**5200.00**	**5200.00**
TF2	$7/4/42$	0.62	0.08	0.33	**5200.00**	**5200.00**	**5200.00**
TF3	$7/0/6$	37.29	6.86	2.08	0.54	**0.49**	0.52
TF3	$7/2/20$	7.8	25.77	5.88	0.97	0.27	**0.26**
TF3	$7/4/42$	140.27	21.92	6.59	0.28	0.27	**0.25**
CSE1	$20/0/20$	16.26	0.59	2.11	**0.02**	**0.02**	**0.02**
CSE1	$20/2/48$	31.31	2.44	2.83	**0.02**	**0.02**	**0.02**
CSE1	$20/8/180$	22.05	4.80	7.41	**0.02**	**0.02**	**0.02**
CSE2	$60/0/60$	61.53	4.31	12.55	**0.02**	**0.02**	**0.02**
CSE2	$60/2/128$	66.22	5.52	16.06	**0.02**	**0.02**	**0.02**
CSE2	$60/10/480$	267.97	61.36	59.63	**0.02**	**0.02**	**0.02**

TABLE E.5.: Miscellaneous

Bibliographie

[1] S. Mauffrey, P. Meunier, G. Pignié, A. Biard, and I. Rongier. H_∞ control for the ARIANE 5 PLUS launcher. 52^{nd} *International Astronautic Congress, Toulouse*, 2001.

[2] J. Sefton and K. Glover. Pole/zero cancellations in the general H_∞ problem with reference to a two block design. *Systems and Control Letters*, 14 :295–306, 1990.

[3] D. Alazard, C. Cumer, P. Apkarian, M. Gauvrit, and G. Ferreres. *Robustesse et Commande Optimale*. Cépaduès, 1999.

[4] D. Alazard and P. Apkarian. Exact oberserver-based structures for arbitrary compensators. *International Journal of Robust and Non-Linear Control*, 9 :101–118, 1999.

[5] C. Scherer, P. Gahinet, and M. Chilali. Multiobjective output-feedback control via LMI optimization. *IEEE Transactions on Automatic Control*, 42(7) :896–911, 1997.

[6] Y. Ebihara, D. Peaucelle, and D. Arzelier. Some conditions for convexifying static H_∞ control problems. Technical Report 10684, LAAS-CNRS, IFAC World Congress, Milan, Italy, 2011.

[7] D. Arzelier. *Théorie de Lyapunov, commande robuste et optimisation*, 2004. Habilitation à diriger des recherches, Spécialité Automatique.

[8] J. V. Burke, D. Henrion, A. S. Lewis, and M. L. Overton. HIFOO - a matlab package for fixed-order controller design and H_∞ optimization. In *Proceedings of the 5th IFAC Symposium on Robust Control Design*, July 2006.

[9] S. Gumussoy, D. Henrion, M. Millstone, and M.L. Overton. Multiobjective robust control with HIFOO 2.0. In *Proceedings of the IFAC Symposium on Robust Control Design*, June 2009.

[10] P. Apkarian and D. Noll. Nonsmooth H_∞ synthesis. *IEEE Transactions on Automatic Control*, 51 :71–86, 2006.

[11] D. Alazard. *Commandes robustes des systèmes flexibles : synthèse des travaux de recherche*, 2003. Habilitation à diriger des recherches, Spécialité Automatique Université Paul Sabatier, Toulouse, France.

[12] J.B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms I : Fundamentals*. Grundlehren der mathematischen Wissenschaften. Springer-Verlag, 1993.

[13] J.B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms II : Advanced theory and bundle methods*. Grundlehren der mathematischen Wissenschaften. Springer-Verlag, 1993.

[14] K.C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization*. Lecture Notes in Mathematics. Springer-Verlag, 1985.

[15] J.F. Bonnans, J.C. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical optimization : Theoretical and practical aspects*. Springer London, Limited, 2007.

[16] F.H. Clarke. *Optimization and nonsmooth analysis*. Canadian Math. Soc. Series. John Wiley & Sons, 1983.

[17] H. Brezis. *Analyse fonctionnelle : Théorie et applications*. Sciences sup. Dunod, 2005.

[18] D. Noll, O. Prot, and A. Rondepierre. A proximity control algorithm to minimize nonsmooth and nonconvex functions. *Pacific Journal of Optimization*, 4(3) :569–602, 2008.

[19] S. Boyd and V. Balakrishnan. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its L_∞-norm. *Systems and Control Letters*, 15 :1–7, 1990.

[20] S. Boyd, V. Balakrishnan, and P. Kabamba. A bisection method for computing the $\mathbf{H}_\infty$ norm of a transfer matrix and related problems. *Mathematics of Control, Signals, and Systems*, 2(3) :207–219, 1989.

[21] P. Apkarian, D. Noll, and O. Prot. A proximity control algorithm to minimize nonsmooth and nonconvex semi-infinite maximum eigenvalue functions. *Journal of Convex Analysis*, 16 :641–666, 2009.

[22] Pierre Apkarian, Dominikus Noll, and Aude Rondepierre. Mixed H_2/H_∞ control via nonsmooth optimization. In *CDC*, pages 6460–6465. IEEE, 2009.

[23] Dominikus Noll, Olivier Prot, and Pierre Apkarian. A proximity control algorithm to minimize nonsmooth and nonconvex semi-infinite maximum eigenvalue functions. *Journal of Convex Analysis*, 16(3-4) :641–666, 2009.

[24] M.L. Overton and R.S. Womersley. On minimizing the spectral radius of a nonsymmetric matrix function : Optimality conditions and duality theory. *SIAM Journal on Matrix Analysis and Applications*, 9 :473–498, 1988.

[25] J. V. Burke and M. L. Overton. Differential properties of the spectral abscissa and the spectral radius for analytic matrix-valued mappings. *Nonlinear Analysis : Theory, Methods and Applications*, 23 :467–488, August 1994.

[26] M.B. Tischler. *Advances In Aircraft Flight Control*. Series in Systems and Control. Taylor & Francis, 1996.

[27] B.L. Stevens and F.L. Lewis. *Aircraft Control and Simulation*. John Wiley & Sons, 1992.

[28] P. Apkarian, D. Noll, and O. Prot. A trust region spectral bundle method for nonconvex eigenvalue optimization. *SIAM Journal on Optimization*, 19(1) :281–306, 2008.

[29] D. Noll. Cutting plane oracles to minimize nonsmooth and nonconvex functions. *Journal of Set-Valued and Variational Analysis*, 18(3-4) :531–568, 2010.

[30] E. Polak. *Optimization : algorithms and consistent approximations*. Applied mathematical sciences. Springer-Verlag, 1997.

[31] Pierre Apkarian, Dominikus Noll, and Aude Rondepierre. Mixed H_2/H_∞ control via nonsmooth optimization. *SIAM J. Control and Optimization*, 47(3) :1516–1546, 2008.

[32] D. Alazard. Robust H_2 design for lateral flight control of a highly flexible aircraft. *Journal of Guidance, Control and Dynamics*, 25 :502–509, 2002.

[33] P. Apkarian and D. Noll. Nonsmooth optimization for multidisk H_∞ synthesis. *European Journal of Control*, 12(3) :229–244, 2006.

[34] Pierre Apkarian and Dominikus Noll. Nonsmooth optimization for multiband frequency domain control design. *Automatica*, 43(4) :724–731, 2007.

[35] J.E. Spingarn. Submonotone subdifferentials of lipschitz functions. *Transactions of the American Mathematical Society*, 264 :77–89, 1981.

[36] R. T. Rockafellar. *Convex analysis*. Princeton Mathematical Series. Princeton University Press, 1997.

[37] A. Daniilidis and P. Georgiev. Approximate convexity and submonotonicity. *Journal of Mathematical Analysis and Applications*, 291 :117–144, 2004.

[38] V. Bompart, D. Noll, and P. Apkarian. Second-order nonsmooth optimization for H_∞ synthesis. *Numerische Mathematik*, 107(3) :433–454, 2007.

[39] J. Burke, A.S. Lewis, and M.L. Overton. *Two numerical methods for optimizing matrix stability*. Research report (University of Waterloo. Faculty of Mathematics). Faculty of Mathematics, University of Waterloo, 2001.

[40] Vincent Blondel and John N. Tsitsiklis. Np-hardness of some linear control design problems. *SIAM J. Control Optim.*, 35(6) :2118–2127, November 1997.

[41] J.L. Boiffier. *The dynamics of flight : the equations*. Dynamics of Flight Series. John Wiley & Sons, 1998.

[42] G. Puyou. Données pour les travaux COCKPIT ONERA-LAAS. Technical report, ONERA-DCSD/AIRBUS, 2010.

[43] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.

[44] S. Skogestad and I. Postlethwaite. *Multivariable feedback control : analysis and design*. John Wiley, 2005.

[45] G. Ferreres and J.M. Biannic. The Skew Mu Toolbox. `http://www.onera.fr/staff-en/jean-marc-biannic/`.

[46] L. Hosseini-Ravanbod, D. Noll, and P. Apkarian. Robustness via structured H_∞/H_∞- synthesis. *International Journal of Control*, 84(5) :851–866, 2011.

[47] P. Apkarian. Nonsmooth μ synthesis. *International Journal of Robust and Nonlinear Control*, 21(8) :1493–1508, 2011.

[48] F. Leibfritz. COMPLeIB, COnstraint Matrix-optimization Problem LIbrary - a collection of test examples for nonlinear semidefinite programs, control system design and related problems. Technical report, Universitat Trier, 2004.

[49] P. Apkarian. Comparison of HINFSTRUCT Matlab Robust Control Toolbox R2010b and HIFOO 3.0 with HANSO 2.0. 2010. Available from : `http://pierre.apkarian.free.fr/hinfstructBenchmarking.html`.

[50] P. Apkarian. Comparison of HINFSTRUCT Matlab Robust Control Toolbox R2012a with HIFOO 3.5 with HANSO 2.1. 2012. Available from : `http://pierre.apkarian.free.fr/hinfstructBenchmarking.html`.

[51] Daniel Ankelhed. *On design of low order H-infinity controllers*. PhD thesis, Linköping UniversityLinköping University, Automatic Control, The Institute of Technology, 2011.

Cette thèse développe une méthode de faisceau non convexe pour la minimisation de fonctions localement lipschitziennes lower $\mathcal{C}^1$ puis l'applique à des problèmes de synthèse de lois de commande structurées issus de l'industrie aéronautique. Ici loi de commande structurée fait référence à une architecture de contrôle, qui se compose d'éléments comme les PIDs, combinés avec des filtres variés, et comprenant beaucoup moins de paramètres de réglage qu'un contrôleur d'ordre plein. Ce type de problème peut se formuler dans le cadre théorique et général de la programmation non convexe et non lisse. Parmi les techniques numériques efficaces pour résoudre ces problèmes non lisses, nous avons dans ce travail, opté pour les méthodes de faisceau, convenablement étendues au cas non convexe. Celles-ci utilisent un oracle qui, en chaque itéré x, retourne la valeur de la fonction et un sous-gradient de Clarke arbitraire. Afin de générer un pas de descente satisfaisant à partir de l'itéré sérieux courant, ces techniques stockent et accumulent de l'information, dans ce que l'on appelle le faisceau, obtenu à partir d'évaluations successives de l'oracle à chaque pas d'essai insatisfaisant. Dans cette thèse, on propose de construire le faisceau en décalant vers le bas une tangente de l'objectif en un pas d'essai ne constituant pas un pas de descente satisfaisant. Le décalage est indispensable dans le cas non convexe pour préserver la consistance, on dit encore l'exactitude, du modèle vis à vis de l'objectif. L'algorithme développé est validé sur un problème de synthèse conjointe du pilote automatique et de la loi des commandes de vol d'un avion civil en un point de vol donné et sur un problème de synthèse de loi de commande par séquencement de gain pour le contrôle longitudinal dans une enveloppe de vol.

This thesis develops a non convex bundle method for the minimization of lower $\mathcal{C}^1$ locally Lipschitz functions which it then applies to the synthesis of structured control laws for problems arising in aerospace control. Here a structured control law refers to a control architecture preferred by practitioners, which consist of elements like PIDs, combined with various filters, featuring significantly less tunable parameters than a full-order controller. This type of problem can be formulated under the theoretical and general framework of non convex and non smooth programming. Among the efficient numerical techniques to solve such non smooth problems, we have in this work opted for bundle methods, suitably extended to address non-convex optimization programs. Bundle methods use oracles which at every iterate x return the function value and one unspecified Clarke subgradient. In order to generate descent steps away from a current serious iterate, these techniques hinge on storing and accumulating information, called the bundle, obtained from successive evaluations of the oracle along the unsuccessful trial steps. In this thesis, we propose to build the bundle by shifting down a tangent of the objective at a trial step which is not a satisfactory descent step. The shift is essential in the non convex case in order to preserve the consistency, named also the exactitude, of the model with regard to the objective. The developed algorithm is validated on a synthesis problem combining the automatic pilot and the flight control law of a civil aircraft at a given flying point ; and a gain scheduled control law synthesis for the longitudinal control in a flight envelope.