

# Optimisation jointe d’une séquence et d’un filtre désadapté

Uy Hour TAN<sup>1,2</sup>, Olivier RABASTE<sup>3</sup>, Claude ADNET<sup>1</sup>, Jean-Philippe OVARLEZ<sup>2,3</sup>

<sup>1</sup>Thales Air Systems – Voie Pierre Gilles de Gennes, 91470 Limours-en-Hurepoix, France

<sup>2</sup>Laboratoire SONDRRA, CentraleSupélec – 3 rue Joliot Curie, 91192 Gif-sur-Yvette cedex, France

<sup>3</sup>ONERA, The French Aerospace Lab – Chemin de la Hunière, BP 80100, 91123 Palaiseau cedex, France

u.tan@laposte.net, olivier.rabaste@onera.fr, claude.adnet@thalesgroup.com,  
jean-philippe.ovarlez@onera.fr

**Résumé** – Dans cet article, on s’intéresse à l’optimisation jointe d’un signal et de son filtre désadapté associé. Ce problème a déjà été traité dans la littérature, mais de manière cyclique : l’optimisation se fait sur une des deux séquences, l’autre étant fixée, et inversement. Cet article propose deux algorithmes – avec deux problèmes d’optimisation – réalisant une optimisation jointe. Le premier introduit une fonction objectif avec un gabarit, qui permet de contrôler le niveau des lobes secondaires et la perte en gain de traitement. Le second utilise la propriété d’existence d’un filtre désadapté optimal minimisant l’ISL (*Integrated Sidelobe Level*). Les simulations effectuées montrent l’intérêt d’une optimisation jointe, et présentent des profils de lobes secondaires intéressants, notamment sur leur répartition selon l’algorithme considéré.

**Abstract** – This paper deals with the joint optimization of a signal and its associated mismatched filter. This question has already been addressed in the literature, by alternatively solving two optimization problems, one per sequence. In this article, two algorithms – and thus, two optimization problems – are introduced in order to optimize jointly both sequences. The first objective function contains a pattern, that allows to control the loss-in-processing gain and the sidelobe level. The other includes the optimal mismatched filter that minimizes the ISL (*Integrated Sidelobe Level*). A gradient descent is applied in both cases. As expected, simulations show that a joint optimization performs better than a separate one. Besides, while sidelobe levels are quite interesting in both cases, an optimization of the ISL presents amazingly low levels around the mainlobe.

## 1 Introduction

Le traitement classique radar se fait en corrélant le signal reçu à un filtre. Ce filtre est dit « adapté » lorsque celui-ci est identique au signal émis ; sinon, on parle de filtre « désadapté ». L’utilisation d’un filtre adapté permet de maximiser le rapport signal sur bruit sous l’hypothèse d’un bruit blanc. Cependant, sa sortie peut présenter des niveaux de lobes secondaires relativement élevés, qui peuvent être corrigés à l’aide d’un filtre désadapté, mais au prix d’une perte en gain de traitement.

La conception d’un signal ou d’un filtre est souvent perçue comme un problème d’optimisation, selon un critère comme l’ISL (*Integrated Sidelobe Level*). Cependant, la résolution de ces problèmes n’est pas aisée. En effet, le signal est, d’une part, soumis à la contrainte (non convexe) de module constant. D’autre part, le nombre de variables peut être assez important, rendant de nombreuses méthodes d’optimisation inefficaces en pratique (à l’instar des méthodes dites stochastiques). Dans cet article, une descente de gradient sera utilisée. Ce choix peut paraître surprenant, de par sa convergence locale, mais cet algorithme a montré des résultats remarquables dans des problèmes similaires [1]. L’optimisation du filtre désadapté est généralement plus souple, car celui-ci n’est soumis à aucune contrainte d’ensemble. D’ailleurs, pour une séquence donnée, il existe un filtre désadapté optimal qui minimise l’ISL [2, 3].

Plus récemment, plusieurs approches ont été proposées afin

d’optimiser la séquence et le filtre de manière cyclique. Deux problèmes d’optimisation – un par séquence – sont résolus alternativement. Par exemple, une relaxation des contraintes [4] rend la recherche d’une séquence SDP (semi-définie positive), tandis que dans [3], l’erreur quadratique moyenne de l’estimation du SINR est étudiée dans le domaine fréquentiel.

Cet article propose de traiter de manière jointe l’optimisation du couple signal/filtre. Deux algorithmes sont proposés, tous deux basés sur une descente de gradient. Le premier introduit un gabarit dans la fonction coût, permettant notamment de contrôler le niveau des lobes secondaires. La descente est effectuée simultanément sur la séquence et le filtre, comme dans [5]. Le second exploite l’existence du filtre désadapté optimal au sens de l’ISL, en l’insérant dans la fonction coût. La fonction ainsi définie modifie la séquence et le filtre implicitement.

Cet article présente, dans un premier temps, quelques rappels sur le filtrage adapté et désadapté. Deux méthodologies d’optimisation jointe seront ensuite définies, qui seront comparées à l’aide de simulations dans la section qui suivra.

## 2 Rappels sur le filtre désadapté

Cette section introduit quelques définitions, et rappelle la procédure permettant de calculer un filtre désadapté optimal au sens de l’ISL pour une séquence donnée.

## 2.1 Définitions

Soit  $\mathbf{s} = [s_1, s_2, \dots, s_N]^T$  un signal discrétisé de longueur  $N$ . Dans la suite, on suppose que  $\mathbf{s}$  est sujet à la contrainte de module constant. En d'autres termes,  $\mathbf{s}$  appartient à un tore de dimension  $N$ , noté  $\mathbf{T}^N$ , un ensemble non convexe. Soit  $\alpha_k$  la phase de l'élément  $s_k$ , telle que  $s_k := \exp(j2\pi\alpha_k)/\sqrt{N}$ , pour  $k \in \llbracket 1, N \rrbracket$ .

Le filtrage adapté consiste à comparer le signal  $\mathbf{s}$  à différentes versions décalées d'elle-même dans le temps. Cela revient à étudier la séquence  $\mathbf{y}$ , de longueur  $2N - 1$ , telle que :

$$\mathbf{y} = \mathbf{\Lambda}_N(\mathbf{s})\mathbf{s}^*, \quad (1)$$

où  $\mathbf{\Lambda}_K(\mathbf{s})$ , définie dans [2], est une matrice de convolution de taille  $K + N - 1 \times K$ , contenant des répliques de  $\mathbf{s}$  décalées dans le temps. D'un autre côté, la compression d'un signal  $\mathbf{s}$  à l'aide d'un filtre différent  $\mathbf{q}$ , de longueur  $K$ , est appelé filtrage désadapté :

$$\mathbf{y} = \mathbf{\Lambda}_K(\mathbf{s})\mathbf{q}. \quad (2)$$

Le filtre désadapté est moins contraint que le filtre adapté.  $\mathbf{q}$  est en effet un élément de  $\mathbb{C}^K$  (un ensemble convexe) et, de plus, sa longueur peut être différente de celle du signal. Ainsi, le filtre désadapté est généralement choisi plus long, ce qui génère plus de degrés de liberté. Sans perte de généralité, on supposera que  $K = N + 2p$ ,  $p \in \mathbb{N}$ , de sorte que  $K$  soit impair.

Le filtre adapté est connu pour maximiser le SNR (*Signal-to-Noise Ratio*) sous l'hypothèse d'un bruit blanc. L'utilisation d'un filtre désadapté entraîne donc nécessairement une perte en gain de traitement (LPG, pour *Loss in Processing Gain*). Cette perte est définie par :

$$\text{LPG} = 10 \log_{10} \left( \frac{\text{SNR}_{\text{dés.}}}{\text{SNR}_{\text{ad.}}} \right) = 10 \log_{10} \left( \frac{|\mathbf{q}^H \mathbf{s}|^2}{(\mathbf{q}^H \mathbf{q})(\mathbf{s}^H \mathbf{s})} \right).$$

Le LPG peut être mis sous la forme d'une contrainte – convexe, qui plus est – d'un problème d'optimisation [2] :

$$\text{LPG} \geq -10 \log_{10} \alpha \Leftrightarrow \mathbf{q}^H \mathbf{q} \leq \alpha \mathbf{s}^H \mathbf{s}. \quad (3)$$

## 2.2 Filtre désadapté optimal

Plusieurs critères ont été définis afin de comparer les performances de ces filtres : le PSLR ou encore l'ISL. Pour ce dernier, il est possible de calculer, pour une séquence donnée, un filtre désadapté optimal. Ceci est le sujet de cette section.

L'ISL (*Integrated Sidelobe Level*) représente la moyenne du niveau des lobes secondaires, en quelque sorte :

$$\begin{aligned} \mathbf{T}^N \times \mathbb{C}^K &\rightarrow \mathbb{R}^+ \\ (\mathbf{s}, \mathbf{q}) &\mapsto \text{ISL}(\mathbf{s}, \mathbf{q}) := \mathbf{y}^H \mathbf{F} \mathbf{y}, \end{aligned} \quad (4)$$

où  $\mathbf{y}$  est définie comme dans l'équation (2), et  $\mathbf{F}$  une matrice diagonale de taille  $K + N - 1$ , spécifiée par le vecteur  $[1, \dots, 1, 0, 1, \dots, 1]$  (un vecteur composé de 1, sauf à la position  $N + p$ , représentant le lobe principal).

Considérons l'ISL comme la fonction coût d'un problème d'optimisation, que l'on notera ( $P_{\text{ISL}}$ ) :

$$(P_{\text{ISL}}) \left\{ \begin{array}{l} \min_{\mathbf{q}} \text{ISL}(\mathbf{s}, \mathbf{q}) \\ \text{t.q.} \mathbf{s}^H \mathbf{q} = \mathbf{s}^H \mathbf{s}. \end{array} \right. \quad (5)$$

Ce problème admet une solution globale analytique, que l'on peut déterminer à l'aide des multiplicateurs de Lagrange [6] :

$$\mathbf{q}_{\text{ISL}}(\mathbf{s}) = \frac{(\mathbf{s}^H \mathbf{s}) (\mathbf{\Lambda}_K(\mathbf{s})^H \mathbf{F} \mathbf{\Lambda}_K(\mathbf{s}))^{-1} \mathbf{s}}{\mathbf{s}^H (\mathbf{\Lambda}_K(\mathbf{s})^H \mathbf{F} \mathbf{\Lambda}_K(\mathbf{s}))^{-1} \mathbf{s}}. \quad (6)$$

Cela signifie que, pour une séquence fixée  $\mathbf{s}$ , il existe un filtre désadapté  $\mathbf{q}_{\text{ISL}}(\mathbf{s})$  qui minimise le critère ISL. Il est nécessaire de remarquer que cette solution ne contrôle en rien la perte en gain de traitement. On peut donc introduire cette perte sous forme de contrainte (cf. (3)). Le problème qui en résulte n'admet plus *a priori* de solution analytique, mais reste un problème QCQP (*Quadratically Constrained Quadratic Program*) convexe [7].

Il est également possible de calculer un filtre désadapté optimal au sens du PSLR. Mais le problème d'optimisation associé n'admet semble-t-il aussi aucune solution analytique [2].

## 3 Algorithmes d'optimisation jointe

Dans cette section, deux algorithmes – et sous-entendu, deux problèmes de minimisation – sont proposés, optimisant de manière conjointe une séquence et son filtre désadapté associé. Le premier algorithme est basé sur une fonction coût comprenant un gabarit. Le second algorithme exploite l'existence du filtre désadapté optimal au sens de l'ISL.

### 3.1 Optimisation jointe avec un gabarit

L'utilisation d'une méthode de gradient pour optimiser simultanément la séquence  $\mathbf{s}$  et le filtre désadapté  $\mathbf{q}$  a déjà été présentée dans [5], où la fonction coût considérée fut le PSLR. Ici, on se propose d'inclure un gabarit. Ce gabarit a pour but de contrôler la perte en gain de traitement, et permet de définir des objectifs sur le niveau des lobes secondaires : un gabarit « plat » peut s'apparenter à une minimisation du PSLR, tandis qu'un autre peut privilégier une minimisation locale autour du pic principal. Ces exemples sont illustrés figure 1.

Pour une séquence  $\mathbf{s}$  de taille  $N$  et un filtre  $\mathbf{q}$  de taille  $K$ , la sortie après traitement est une séquence  $\mathbf{y}$  de longueur  $N + K - 1$ . Cela est donc également le cas pour le gabarit  $\mathbf{g}$ .

Le problème d'optimisation étudié est le suivant :

$$\left\{ \begin{array}{l} \min_{\mathbf{s}, \mathbf{q}} E_1(\mathbf{s}, \mathbf{q}) := \sum_{k=1}^{N+K-1} \left( |y_k/g_k|^{2\rho} - 1 \right)^2, \\ \mathbf{s} \in \mathbf{T}^N, \mathbf{q} \in \mathbb{C}^K, \end{array} \right. \quad (7)$$

avec  $\rho \in \mathbb{N}^*$ . On recherche ainsi le couple  $(\mathbf{s}, \mathbf{q})$  dont la corrélation  $\mathbf{y}$  est la plus proche du gabarit imposé  $\mathbf{g}$ .

Résoudre ce problème d'optimisation à l'aide d'une descente de gradient nécessite le calcul d'un... gradient. Pour cela, on se propose, dans un premier temps, de calculer les dérivées partielles selon les phases de  $\mathbf{s}$  – notées  $\alpha_k$  – afin de conserver la propriété de module constant :

$$\frac{\partial E_1}{\partial \alpha_k} = -2\rho \Im [s_k ((\boldsymbol{\gamma} \circ \mathbf{y}) * \mathbf{q})_k], \quad k \in \llbracket 1, N \rrbracket, \quad (8)$$

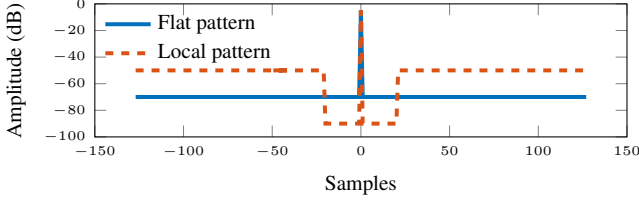


FIGURE 1 – Exemples de gabarits. Une perte (exagérée) de 5 dB est autorisée.

où  $\gamma_k = (|y_k/g_k|^{2\rho} - 1) |y_k/g_k|^{2\rho-1} (1/g_k^2)$ . Pour le filtre désadapté, qui ne présente pas ce type de contrainte, on considère les dérivées de Wirtinger (directement sur les éléments, cette fois-ci) :

$$\frac{\partial E_1}{\partial q_k} = \rho ((\gamma \circ \mathbf{y}) * \mathbf{s}^r), \quad k \in \llbracket 1, K + N - 1 \rrbracket. \quad (9)$$

Ici,  $\circ$  et  $*$  représentent les produits d'Hadamard et de convolution, tandis que  $(\cdot)^r$  désigne l'opérateur « inverse » (pour un vecteur  $\mathbf{a} := [a_1, \dots, a_N]$ ,  $\mathbf{a}^r := [a_N, \dots, a_1]$ ).

Après la convergence de la descente, le gabarit peut être légèrement modifié (réduction des « lobes secondaires » de 1 dB par exemple). Il paraîtrait logique d'utiliser un gabarit ayant les lobes secondaires les plus faibles possibles (voire irréalisables) mais, en pratique, la vitesse de convergence et la qualité des résultats proposés par l'algorithme en pâtissent. Il vaut donc mieux utiliser un gabarit atteignable, et le réajuster ensuite.

Enfin, on se propose de considérer deux conditions d'arrêt : d'une part, l'amélioration entre deux itérations se doit être bénéfique et quantifiable; d'autre part, il est nécessaire que la contrainte sur le lobe principal (et donc, sur la perte) soit respectée. La procédure complète est résumée dans le tableau 1.

Algorithme 1 : Optimisation jointe avec un gabarit

<b>Données</b>	Initialisation de la séquence $\mathbf{s}$ (aléatoire, par exemple) Initialisation du filtre désadapté $\mathbf{q}$
<b>Répéter</b>	1. Descente de gradient : – Calcul du gradient : $\nabla_{\mathbf{s}} E, \nabla_{\mathbf{q}} E$ – Recherche du pas optimal $\mu$ – Mise à jour de $\mathbf{s} : \mathbf{s}_k = \mathbf{s}_k \exp(-j\mu(\nabla_{\mathbf{s}} E)_k), k \in \llbracket 1, N \rrbracket$ – Mise à jour de $\mathbf{q} : \mathbf{q}_k = \mathbf{q}_k - \mu(\nabla_{\mathbf{q}} E)_k, k \in \llbracket 1, K \rrbracket$ – Itérer la procédure jusqu'à convergence
<b>Jusqu'à</b>	2. Modification du gabarit $\mathbf{g}$ Une condition d'arrêt est satisfaite.

TABLE 1 – Optimisation jointe avec un gabarit

### 3.2 Optimisation jointe au sens de l'ISL

Un algorithme itératif est également proposé dans cette section. Il exploite l'existence d'un filtre désadapté optimal qui minimise l'ISL, comme expliqué dans la section 2. Cette procédure peut également être étendue avec le critère du PSLR.

Afin d'effectuer une optimisation jointe, le problème ( $P_{\text{ISL}}$ ) devient :

$$(P_{\text{ISL}}^{(2)}) \left\{ \begin{array}{l} \min_{\mathbf{s}, \mathbf{q}} \text{ISL}(\mathbf{s}, \mathbf{q}) \\ \text{t.q. } \mathbf{s}^H \mathbf{q} = \mathbf{s}^H \mathbf{s}. \end{array} \right. \quad (10)$$

Remarquons que la seule différence par rapport au problème originel est l'introduction de la variable  $\mathbf{s}$  dans les variables à

optimiser. Cependant, ce dernier vit dans l'ensemble  $\mathbf{T}^N$  non convexe. Ce problème ne peut donc pas être résolu *a priori* de manière optimale. Cela dit, en séparant les deux variables d'optimisation, on obtient le problème suivant :

$$(P_{\text{ISL}}^{(2)}) \left\{ \begin{array}{l} \min_{\mathbf{s}} \min_{\mathbf{q}} \text{ISL}(\mathbf{s}, \mathbf{q}) \\ \text{t.q. } \mathbf{s}^H \mathbf{q} = \mathbf{s}^H \mathbf{s}. \end{array} \right. \quad (11)$$

Dans cette expression, on peut reconnaître le problème ( $P_{\text{ISL}}$ ). Comme mentionné précédemment, ce dernier admet une solution analytique, notée  $\mathbf{q}_{\text{ISL}}(\mathbf{s})$ . En insérant cette solution dans la fonction coût, le problème d'optimisation ne dépend plus que d'une seule variable :

$$(P_{\text{ISL}}^{(4)}) \left\{ \begin{array}{l} \min_{\mathbf{s}} E(\mathbf{s}) := \text{ISL}(\mathbf{s}, \mathbf{q}_{\text{ISL}}(\mathbf{s})) \\ \text{t.q. } \mathbf{s}^H \mathbf{q}_{\text{ISL}}(\mathbf{s}) = \mathbf{s}^H \mathbf{s}. \end{array} \right. \quad (12)$$

Une descente de gradient permet également d'obtenir des solutions acceptables de ce problème. Le calcul du vecteur gradient,  $\nabla E = (\partial_{\alpha_k} E)_{k \in \llbracket 1, N \rrbracket}$ , peut également être effectué de manière numérique. L'algorithme est décrit dans le tableau 2.

Algorithme 2 : Optimisation jointe au sens de l'ISL

<b>Données</b>	Initialisation de la séquence $\mathbf{s}$ Initialisation du filtre désadapté $\mathbf{q}$
<b>Répéter</b>	1. Calcul de $\mathbf{q}_{\text{ISL}}(\mathbf{s})$ , solution analytique du problème ( $P_{\text{ISL}}$ ) 2. Descente de gradient – Calcul du vecteur gradient $\nabla E$ – Recherche du pas optimal $\mu$ – Mise à jour de $\mathbf{s} : \mathbf{s}_k = \mathbf{s}_k \exp(-j\mu(\nabla E)_k), k \in \llbracket 1, N \rrbracket$
<b>Jusqu'à</b>	Une condition d'arrêt est satisfaite (sur l'amélioration après chaque itération)

TABLE 2 – Optimisation jointe prenant en compte le filtre désadapté optimal au sens de l'ISL

## 4 Simulations et résultats

Dans cette section, on présente quelques résultats obtenus avec les algorithmes décrits précédemment. La séquence  $\mathbf{s}$  est de longueur choisie  $N = 64$ , tandis que le filtre désadapté  $\mathbf{q}$  est trois fois plus long. On s'autorise une perte en gain de traitement de l'ordre de 1 dB. Pour résoudre les différents problèmes d'optimisation, le package CVX a été utilisé [7].

### 4.1 Résultats

La figure 2 illustre l'intérêt d'une optimisation jointe. En effet, elle compare les sorties après filtrage d'une séquence tirée aléatoirement puis optimisée soit de manière séquentielle, soit de manière jointe, comme présentée dans la section 3.1. (i) Optimisation séquentielle : une séquence est tirée de manière aléatoire, dont la sortie après filtrage adapté est représentée en bleue. Cette séquence est ensuite optimisée au sens du filtre adapté, comme dans [1]; le filtre désadapté optimal au sens du PSLR [2] est ensuite appliqué. (ii) Optimisation jointe : à partir de la même initialisation, une optimisation jointe est effectuée en appliquant la méthode présentée dans la section 3.1. Une différence de l'ordre de 10 dB peut être observée, ce qui n'est pas négligeable pour des séquences aussi courtes.

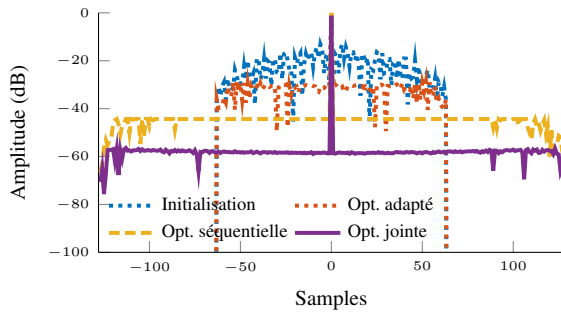


FIGURE 2 – Optimisation séquentielle ou jointe ?

La figure 3 présente les résultats obtenus avec les deux algorithmes d’optimisation jointe proposés, à partir de la même initialisation. En terme de PSLR, le premier algorithme est plus efficace, avec des lobes secondaires autour de -58 dB. Le profil obtenu dépend bien évidemment du gabarit choisi – plat, ici ; cependant, il est difficile de déterminer un gabarit avec un bon ISL. L’optimisation à l’aide de la seconde méthode présente un profil intéressant : elle propose des lobes secondaires plus bas autour du pic principal, et acceptables sur les bords (inférieurs à -40 dB).

## 4.2 Sur le choix de l’initialisation

Une descente de gradient converge vers le minimum local le plus proche. Le choix de l’initialisation a donc son importance, car il influencerait logiquement le résultat final. Ainsi, comme la figure 3, la figure 4 se propose d’établir une comparaison entre les deux méthodes. Cette fois, l’initialisation n’est plus aléatoire mais optimisée de sorte que le PSLR en sortie de filtre adapté soit le plus faible possible [1].

Outre une amélioration globale du niveau des lobes secondaires, c’est surtout leur répartition qui interpelle. Une optimisation à l’aide de la première méthode – qui recherche le PSLR le plus bas possible – nous offre des lobes secondaires autour de -66 dB, ce qui est intéressant pour des séquences de longueur  $N = 64$ . De l’autre côté, une optimisation au sens de l’ISL nous donne des résultats assez impressionnants, avec des lobes secondaires autour de -90 dB autour du pic principal, tandis que ceux sur les bords restent largement acceptables (inférieurs à -50 dB). Cette répartition est très intéressante, notamment dans des applications radars, car elle permettrait d’améliorer la détection des cibles faibles et proches, qui sont généralement noyées dans le clutter.

## 5 Conclusion

Dans cet article, plusieurs algorithmes ont été proposés, permettant d’optimiser de manière jointe une séquence et son filtre désadapté. Contrairement aux méthodes proposées dans la littérature, l’optimisation est effectuée de manière simultanée sur les deux séquences (et non pas l’une après l’autre). Les simulations ont montré des résultats encourageants, meilleurs qu’une optimisation cyclique. Le niveau des lobes secondaires pré-

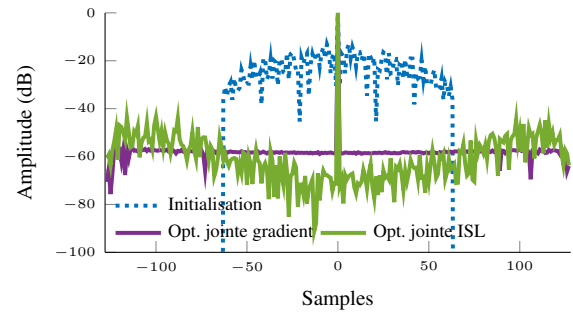


FIGURE 3 – Optimisation jointe (initialisation aléatoire)

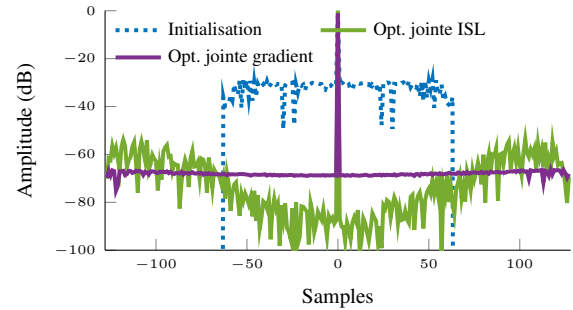


FIGURE 4 – Optimisation jointe (initialisation « optimisée »)

senté est relativement bon, même pour des séquences courtes. De plus, selon l’algorithme, la répartition des lobes secondaires est différente, mais chacune a son intérêt. Le premier algorithme propose des lobes secondaires faibles partout, tandis qu’une optimisation au sens de l’ISL nous offre des résultats impressionnants autour du lobe principal.

## Remerciements

U. Tan remercie la Direction Générale de l’Armement pour lui avoir accordé un financement.

## Références

- [1] J. M. Baden, M. S. Davis et L. Schmieder, *Efficient Energy Gradient Calculations for Binary and Polyphase Sequences*, 2015 IEEE Radar Conference (RadarCon), 2015.
- [2] O. Rabaste et L. Savy, *Mismatched Filter Optimization for Radar Applications Using Quadratically Constrained Quadratic Programs*, IEEE Transactions on Aerospace and Electronic Systems, 2015.
- [3] P. Stoica, H. He et J. Li, *Optimization of the Receive Filter and Transmit Sequence for Active Sensing*, IEEE Transactions on Signal Processing, 2012.
- [4] S. M. Karbasi, M. Radmard, M. M. Nayebi et M. H. Bastani, *Design of Multiple-Input Multiple-Output Transmit Waveform and Receive Filter for Extended Target Detection*, IET Radar, Sonar Navigation, 2015.
- [5] L. Xu, H. Liu, K. Yin, H. Zang, S. Zhou et H. Wang, *Joint Design of Phase Coded Waveform and Mismatched Filter*, 2015 IEEE Radar Conference, 2015.
- [6] K. R. Griep, J. A. Ritcey et J. J. Burlingame, *Poly-Phase Codes and Optimal Filters for Multiple User Ranging*, IEEE Transactions on Aerospace and Electronic Systems, 1995.
- [7] M. Grant et S. Boyd, *CVX : Matlab Software for Disciplined Convex Programming, version 2.1*, <http://cvxr.com/cvx>, 2014.