



HAL
open science

Méthodologies de conception de formes d'onde pour radars sol. Application au cas du radar MIMO.

Uy Hour Tan

► **To cite this version:**

Uy Hour Tan. Méthodologies de conception de formes d'onde pour radars sol. Application au cas du radar MIMO.. Traitement du signal et de l'image [eess.SP]. Université Paris Saclay (COMUE), 2019. Français. NNT : 2019SACLCO36 . tel-02388089

HAL Id: tel-02388089

<https://theses.hal.science/tel-02388089>

Submitted on 1 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Méthodologies de conception de formes d'onde pour radars sol Application au cas du radar MIMO

Thèse de doctorat de l'Université Paris-Saclay
préparée à CentraleSupélec

École doctorale n°580 : Sciences et Technologies de l'Information
et de la Communication (STIC)

Spécialité de doctorat : Traitement du signal (pôle 1)

Thèse présentée et soutenue à Gif-sur-Yvette le 13 juin 2019 par

Uy Hour TAN

Composition du jury :

Sylvie Marcos Directrice de recherche, CentraleSupélec (L2S)	Présidente
Éric Chaumette Professeur, ISAE-SUPAERO (DEOS)	Rapporteur
Braham Himed Docteur, <i>Technical Advisor</i> , US Air Force Research Laboratory	Rapporteur
Laurent Ferro-Famil Professeur, Université de Rennes 1 (IETR)	Examineur
Guillaume Ginolhac Professeur, Université de Savoie Mont-Blanc (LISTIC)	Examineur
Jean-Philippe Ovarlez Maître de recherche, CentraleSupélec (SONDRA)	
Ingénieur de recherche, ONERA (DEMR)	Directeur de thèse
Claude Adnet Docteur, ingénieur de recherche, Thales Land & Air Systems	Encadrant
Olivier Rabaste Docteur, ingénieur de recherche, ONERA (DEMR)	Encadrant

Remerciements

La coutume veut qu'un manuscrit commence par des remerciements. Je m'y apprête donc sobrement ici — *via* quelques anaphores et autres répétitions —, non qu'il le fasse, mais par envie. Il faut dire que j'ai eu la chance et l'honneur de côtoyer pendant toutes ces années des personnes exceptionnelles, sur le plan scientifique certes, mais surtout sur le plan humain.

Commençons par le commencement, avec mon directeur de thèse, Jean-Philippe Ovarlez. Jean-Phi, merci à toi de m'avoir soutenu et supporté pendant ces années. Ça y est, te voilà enfin libéré de moi ! Je tiens à remercier du fond du cœur mes deux encadrants pour leur suivi, Claude Adnet à Limours et Olivier Rabaste à Palaiseau. Claude, un immense merci pour ta grande disponibilité, tes nombreux conseils — sportifs ? — et ta persévérance. Olivier, tu t'es greffé par toi-même à ce projet, et tu t'y es impliqué comme jamais je n'aurais pu l'espérer. Sans toi, je ne pense pas que j'aurais souhaité terminer cette thèse. Je t'en suis infiniment reconnaissant. Merci aussi à François Polle pour ces quelques après-midi algorithmiques, à Jean-Paul Guyvarch qui a initié ce projet de thèse, et à ma chère Nadège.

Ensuite, j'exprime ma gratitude à l'ensemble des membres du jury, et en particulier aux deux rapporteurs, pour avoir accepté d'évaluer mon travail de thèse. Merci à Éric Chaumette de m'avoir embarqué dans ses bagages pour une nouvelle aventure.

L'impatience de retrouver une équipe, la déception et les remords de quitter l'autre : tels étaient les sentiments qui m'ont balancé ces années durant, partageant mon bureau entre Thales et le laboratoire Sondra.

Je tiens ainsi à remercier chaleureusement l'ensemble de mes collègues limouriens, et en particulier Fabien, pour ta gentillesse et nos discussions interminables (Ah ! la borne...); Marion, pour tes petites piques humoristiques et les belles découvertes; Yanis, notre encyclopédie curieuse mais intimidante; et Alice, la Parisienne inavouée. J'ai aussi une pensée pour le groupe des (presque) coureurs-grimpeurs : Daniel, Mathilde, Christophe, Narayan, Paquito, Laurence, Judith... Et petite dédicace à Achraf et à Mounir.

Mes amitiés vont également de l'autre côté de l'Yvette, à Eugénie, Orian, Pedro, Ammar et Bruno, camarades doctorants. J'admets volontiers chérir les moments partagés — toujours autour d'un café et de plusieurs madeleines — à converser, à plaisanter et à râler sur tout et n'importe quoi, mais aussi à nous soutenir mutuellement. J'estime d'ailleurs beaucoup cette entraide collective, si magnétique... et qui me manquera nécessairement. En outre, je remercie Thierry, Israel et Chengfang qui nous ont souvent tenus compagnie, ainsi que Joana et Alice pour leur accueil bienveillant. J'ajoute un grand, que dis-je, un immense merci à Virginie, pour tout le temps consacré à nous aider et à nous conseiller (et quelquefois à me remettre dans le droit chemin), toujours avec entrain... et un brin de maternité. Plus globalement, je suis reconnaissant envers le laboratoire Sondra et ses directeurs — M. Lesturgie (le pot ?), S. Azarian (le pot ?), S. Saillant — pour le soutien et l'hospitalité pendant toutes ces années.

A few words in English to thank all the people in Singapore who welcomed me: Adriaan Smits, Chin Kwok Quek and Chin Yuan Chong. I would also like to thank Camille, Teresa — and her snacks! —, Noel and Matthew who made my stay there more enjoyable.

Enfin, je me permets d'exprimer toute mon affection à mes parents si attentionnés, ma sœur si aimante et mon frère en or à sa façon (enfin presque...).

Table des matières

Remerciements	iii
Table des matières	vii
Table des figures	xi
Liste des tableaux	xiii
Notations	xv
Abréviations	xvii
Introduction	1
1 Introduction sur le radar	7
1.1 Généralités	7
1.1.1 Aux origines du radar	7
1.1.2 Principes de base	8
1.1.3 Le traitement radar	9
1.1.3.1 Filtrage adapté	11
1.1.3.2 Fonction d’ambiguïté	13
1.2 Le radar MIMO cohérent	15
1.2.1 Classification des radars	15
1.2.2 La fonction d’ambiguïté MIMO	19
1.2.2.1 Mise en place et notations	19
1.2.2.2 Le traitement classique	20
1.2.2.3 Illustration des différents traitements	23
1.2.3 Beampattern	23
1.2.4 Différentes formes d’onde	25
1.3 Objectifs de la thèse	30
2 De l’optimisation de la corrélation	33
2.1 La fonction de corrélation	33
2.1.1 Définition d’un code de phase	33
2.1.2 Filtrage adapté et corrélation	34
2.1.3 Mesure des lobes secondaires	36
2.1.4 Familles connues	36
2.1.4.1 Codes de Barker	37
2.1.4.2 Construction des codes de Frank	38
2.1.5 Un problème d’optimisation	39
2.2 Comparatif des méthodes d’optimisation	42
2.2.1 Méthodes heuristiques	43
2.2.1.1 Présentation	43
2.2.1.2 Méthodes évolutionnaires	45
2.2.1.3 Premiers résultats	48
2.2.2 Méthodes déterministes	53
2.2.2.1 Présentation	53

2.2.2.2	La descente de gradient	57
2.2.2.3	Résultats applicatifs	59
2.2.2.4	Sortir du point selle	64
2.3	Conclusion	68
3	Aux contraintes opérationnelles	71
3.1	Altération des codes de phase	71
3.1.1	Robustesse	71
3.1.2	Sur-échantillonnage	72
3.2	Contrainte spectrale	75
3.2.1	Modélisation d'un problème d'optimisation	75
3.2.2	Parenthèse sur l'optimisation multi-objectif	77
3.2.3	Exemple numérique	79
3.3	Phénomènes d'éclipse	80
3.3.1	À gauche...	82
3.3.2	... <i>et</i> à droite	85
3.4	Effets de la quantification	87
3.4.1	Introduction	87
3.4.2	Relaxation, optimisation discrète	88
3.4.2.1	Relaxation	88
3.4.2.2	Une optimisation discrète	89
3.5	Orthogonalité d'un ensemble de séquences	94
3.5.1	Mise en place	94
3.5.2	Quelques résultats	97
3.6	Bornes : jusqu'où peut-on aller ?	100
3.6.1	État de l'art sur les bornes	100
3.6.2	Généralisation de la borne de Levenshtein	102
3.6.3	Améliorations de la borne de Levenshtein	104
3.6.4	Comparaison avec les bornes existantes	106
3.6.4.1	Deux séquences	106
3.6.4.2	Trois séquences	107
3.6.5	Conclusion	108
3.7	Conclusion et perspectives	109
4	En passant par le filtre désadapté	113
4.1	Préliminaires	114
4.1.1	Définitions	114
4.1.2	Filtre désadapté optimal	116
4.1.2.1	Mesure générique	116
4.1.2.2	L' <i>Integrated Sidelobe Level</i>	117
4.1.3	Pour un ensemble de séquences	118
4.1.4	Résultats	120
4.1.4.1	Passage introductif	120
4.1.4.2	Galerie multi-séquence	123
4.1.4.3	Passerelle transitoire	127
4.1.5	Conclusion	130
4.2	Optimisation jointe d'une séquence et d'un filtre	130
4.2.1	Optimisation alternée	130
4.2.2	Optimisation jointe	132
4.2.2.1	Optimisation jointe avec gabarit	132
4.2.2.2	Optimisation jointe par réinjections	134
4.2.3	Résultats	135
4.2.4	Conclusions et perspectives	137
4.3	Bornes : parce qu'on n'ira pas loin	137

4.3.1	Une borne sur le filtrage désadapté	138
4.3.1.1	Définitions et notations	138
4.3.1.2	Une nouvelle borne	139
4.3.2	Quelques commentaires	142
4.3.2.1	Pondération optimale	142
4.3.2.2	Comparaisons sur les bornes	143
4.3.3	Conclusion	145
5	Aux codes de phase pour le radar MIMO cohérent	147
5.1	Le radar MIMO cohérent	148
5.1.1	Généralités et notations	148
5.1.2	Fonction d'ambiguïté	149
5.1.3	Écriture matricielle, discrétisation	150
5.1.4	Différentes formes d'onde	151
5.1.5	Beampattern	152
5.2	Un problème d'optimisation	153
5.2.1	Sur les lobes secondaires	154
5.2.2	Beampattern, directions d'intérêt	158
5.2.3	Le problème d'optimisation	159
5.2.3.1	Avec filtrage adapté	159
5.2.3.2	Avec filtrage désadapté	160
5.2.4	Derniers résultats	161
5.3	Conclusion	162
	Conclusion	171
	Le début de la fin	171
	La fin du début?	172
A	Bornes sur les corrélations apériodiques	175
A.1	Généralisation de la borne de Levenshtein	175
A.2	Amélioration de la borne de Levenshtein	179
A.3	Borne proposée pour deux séquences	180
B	Bornes sur le filtrage désadapté	183
B.1	Démonstration de la borne supérieure	184
B.2	Démonstration de la borne inférieure	185
B.3	Un petit calcul	186
	Bibliographie	189

Table des figures

I.1	Différents systèmes	1
I.2	L’union fait la force	3
I.3	Illustration des méthodes d’optimisation	3
I.4	Filtrage adapté, filtrage désadapté	4
1.1	Principe du radar	8
1.2	Détermination de la distance	9
1.3	Effet Doppler	10
1.4	Représentation des directions angulaires azimut φ et élévation θ	10
1.5	Approximation de la position angulaire	10
1.6	Illustration d’un signal bruité	11
1.7	Corrélation d’un signal bruité	12
1.8	Détection et corrélation	12
1.9	Résolution et corrélation	12
1.10	Fonction d’ambiguïté distance/Doppler pour un chirp	14
1.11	Une ambiguïté distance	15
1.12	Radar monostatique, radar bistatique, radar MIMO multistatique	16
1.13	Principe du réseau phasé classique	18
1.14	Principe du radar MIMO cohérent (1)	18
1.15	Principe du radar MIMO cohérent (2)	18
1.16	Vue d’ensemble de la station expérimentale RIAS	19
1.17	Traitement optimal d’un signal à émission colorée	22
1.18	Traitement supposé d’un signal à émission colorée	22
1.19	Position relative des antennes	22
1.20	Corrélation avec traitement à la réception	24
1.21	Corrélation avec traitement à l’émission et à la réception	24
1.22	Exemples de faisceaux pour un radar MIMO composé de $N_T = 12$ antennes	25
1.23	Fonction d’ambiguïté pour un codage FDMA	26
1.24	Fonction d’ambiguïté pour un codage TDMA	27
1.25	Fonction d’ambiguïté pour un codage fréquentiel hybride	27
1.26	Fonction d’ambiguïté pour un codage CDMA aléatoire (1)	28
1.27	Illustration d’une optimisation d’un codage fréquentiel	29
1.28	Fonction d’ambiguïté pour un codage CDMA aléatoire (2)	30
2.1	Représentation d’un code de phase	34
2.2	Représentation d’un code de Barker à 13 éléments	38
2.3	Niveau des lobes secondaires des codes de Frank	39
2.4	Représentation du fléau de la dimension	41
2.5	Séparabilité de la fonction Rastrigin	41
2.6	Propriétés géométriques des fonctions convexes	41
2.7	Une fonction convexe, une fonction d’autocorrélation	41
2.8	Représentation d’un point selle	42

2.9	Classification des méthodes métaheuristiques	44
2.10	Les différentes étapes d'un algorithme évolutionnaire	45
2.11	Illustration de l'algorithme CMA-ES	47
2.12	Optimisation à l'aide de méthodes stochastiques, $N = 63$	48
2.13	Optimisation à l'aide de méthodes stochastiques, $N = 1023$	49
2.14	Comparaison du PSL selon la longueur de la séquence N	49
2.15	Étude de convergence de deux algorithmes évolutionnaires	50
2.16	Efficacité de la méthode CMA-ES après 100 tirages différents	51
2.17	Choix de la norme de la fonction coût pour l'algorithme CMA-ES	52
2.18	Représentation de la fonction objectif pour $N = 4$	52
2.19	Illustration de l'algorithme à régions de confiance	54
2.20	Illustration de l'algorithme de descente de gradient	55
2.21	Optimisation à l'aide de méthodes déterministes, $N = 63$	60
2.22	Optimisation à l'aide de méthodes déterministes, $N = 1023$	60
2.23	Étude de convergence de quelques méthodes déterministes	61
2.24	Choix de la norme de la fonction coût pour deux méthodes déterministes	61
2.25	Choix de la norme de la fonction coût pour deux méthodes déterministes (2)	61
2.26	Répartition des lobes secondaires après une descente de gradient, $p = 8$	62
2.27	Répartition des lobes secondaires après une descente de gradient, $p = 8$	62
2.28	Temps d'exécution des deux algorithmes déterministes	62
2.29	Convergence de la descente de gradient en fonction du nombre d'évaluations	63
2.30	Marche aléatoire basée sur le gradient	66
2.31	Illustration d'une marche aléatoire basée sur le gradient	67
3.1	Représentation des phases d'un signal bruité.	73
3.2	Altération des phases d'un code polyphase (1)	73
3.3	Altération des phases d'un code polyphase (2)	73
3.4	Phases d'une séquence sur-échantillonnée	74
3.5	Corrélation d'une séquence sur-échantillonnée, selon différentes interpolations	74
3.6	Illustration d'un front de Pareto	77
3.7	Densité spectrale d'une séquence de longueur $N = 31$	79
3.8	Corrélation de séquences optimisées spectralement	80
3.9	Illustration d'un phénomène d'éclipse	81
3.10	Séquence de corrélation pour un signal éclipsé	82
3.11	Optimisation d'une séquence pour plusieurs coefficients d'éclipse	84
3.12	Lobes secondaires selon le coefficient d'éclipse	84
3.13	Différents types d'éclipse	85
3.14	Optimisation pour une éclipse à gauche seule	86
3.15	Optimisation simultanée pour une éclipse à gauche et à droite	86
3.16	Illustration de la quantification de phases	87
3.17	Corrélation d'un signal sous plusieurs niveaux de quantification	88
3.18	Performances de la quantification selon le nombre de racines de l'unité	90
3.19	Performances de la quantification selon la longueur de séquence	90
3.20	Représentation des éléments de l'ensemble $(R_n)^N$ sous la forme d'un arbre	92
3.21	Optimisation d'un ensemble de $M = 2$ séquences de longueur $N = 127$	98
3.22	Comparaison des lobes secondaires selon différents paramètres	99
3.23	Comparaison des premières séquences de corrélation	99
3.24	La nouvelle borne, fonction du paramètre D	106
3.25	Comparaison entre la borne de Levenshtein et la borne proposée, $M = 3$	109
3.26	Exemple de résultats obtenus avec le banc Hycam de l'ONERA	110
4.1	Filtrage désadapté optimal pour un <i>chirp</i> échantillonné	117
4.2	Filtrage désadapté optimal pour un code de phase	120
4.3	Corrélation pour différentes longueurs de filtres désadaptés	122

4.4	PSLR selon la longueur du filtre désadapté	122
4.5	PSLR selon la contrainte de perte	122
4.6	Niveau des lobes secondaires selon la longueur, cas mono-séquence	124
4.7	Optimisations locales du filtre désadapté	124
4.8	Optimisation d'un ensemble de $M = 2$ séquences, avec filtrage désadapté	125
4.9	PSLR selon différents paramètres, avec filtrage désadapté.	126
4.10	Filtrage désadapté optimal pour un code de phase	127
4.11	PSLR selon la longueur du filtre et la perte, initialisation optimisée	128
4.12	PSLR selon la longueur de séquence, cas mono-séquence	129
4.13	PSLR selon différents paramètres, filtrage désadapté et initialisation optimisée	129
4.14	Exemples de gabarits	132
4.15	Optimisation cyclique, ou optimisation jointe?	136
4.16	Comparatif des méthodes d'optimisation jointe	136
4.17	Comparatif des méthodes d'optimisation jointe, initialisation optimisée	136
4.18	Comparaison de différentes bornes en fonction de K , $M = 4$ séquences	144
4.19	Comparaison de différentes bornes en fonction de K , $M = 10$ séquences	144
4.20	Comportement des bornes en fonction du nombre de séquences	144
4.21	Comportement des bornes en fonction de la longueur des séquences	145
5.1	Principe du radar MIMO cohérent (1)	148
5.2	Principe du radar MIMO cohérent (2)	148
5.3	Fonction d'ambiguïté pour un codage FDMA	152
5.4	Exemples de faisceaux pour un radar MIMO composé de $N_T = 12$ antennes	153
5.5	Principe du réseau virtuel	154
5.6	Un exemple de directions d'intérêt, pour un faisceau focalisé	158
5.7	Fonction d'ambiguïté MIMO, séquences élémentaires aléatoires (1)	164
5.8	Fonction d'ambiguïté MIMO, séquences élémentaires aléatoires (2)	164
5.9	Fonction d'ambiguïté MIMO, séquences élémentaires pseudo-orthogonales	165
5.10	Fonction d'ambiguïté MIMO, séquences optimisées	165
5.11	Fonction d'ambiguïté MIMO, séquences optimisées, filtrage désadapté	166
5.12	Quelques coupes sur les fonctions d'ambiguïtés (1)	166
5.13	Quelques coupes sur les fonctions d'ambiguïtés (2)	167
5.14	Orthogonalité des séquences élémentaires	168
5.15	Diagramme obtenu après optimisation	169
5.16	Lobes secondaires selon le nombre de séquences	169
5.17	Lobes secondaires selon le nombre de séquences, avec filtrage désadapté	169
II.1	Fonction d'ambiguïté d'un réseau phasé classique	172
II.2	Faites l'amour, pas la guerre	173

Liste des tableaux

2.1	Algorithme d'optimisation alternée	46
2.2	Algorithme de recherche aléatoire locale	46
2.3	Algorithme de descente générique	57
2.4	Algorithme de descente avec <i>backtracking</i>	63
2.5	Algorithme d'exploration naïve	65
2.6	Algorithme simplifié d'une marche aléatoire	65
2.7	Comparaison des différentes échappatoires	68
3.1	Un algorithme à ne pas utiliser	89
3.2	Un algorithme à privilégier	89
3.3	Comparaison des bornes pour différentes valeurs de N , $M = 2$	107
3.4	Comparaison des bornes pour différentes valeurs de N , $M = 3$	107
4.1	Algorithme d'optimisation alternée	131
4.2	Algorithme d'optimisation avec gabarit	133
4.3	Optimisation jointe par réinjections avec une descente de gradient	134

Notations

x	Scalaire (caractère minuscule)
\mathbf{x}	Vecteur (caractère minuscule, en gras)
\mathbf{X}	Matrice (caractère majuscule, en gras)
c	Célérité de la lumière, $c = 299792458$ m/s
j	Unité imaginaire
\mathbf{k}	Vecteur d'onde
B	Bande de fréquence
N_R	Nombre de récepteurs
N_T	Nombre d'émetteurs
θ	Angle d'élévation
λ	Longueur d'onde
ν	Fréquence Doppler
τ	Retard
χ	Fonction d'ambiguïté
\mathbb{N}	Ensemble des entiers naturels
\mathbb{Z}	Ensemble des entiers relatifs
\mathbb{R}	Ensemble des réels
\mathbb{C}	Ensemble des complexes
$L^2(\mathbb{C})$	Espace des fonctions de carré intégrable
R_n	Ensemble des racines n -ième de l'unité $R_n := \left\{ \exp\left(\frac{2\pi jk}{n}\right), k \in \llbracket 0, n-1 \rrbracket \right\}.$
T^n	Hypertore de dimension n
Ω^n	Sphère de dimension n
\mathbf{I}_n	Matrice identité de taille n
\mathbf{E}_n	Matrice d'échange de taille n $\mathbf{E}_n = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}.$
$\mathbf{0}_{m,n}$	Matrice nulle de taille $m \times n$
$\mathbf{1}_{m,n}$	Matrice de taille $m \times n$, ne contenant que des 1
$\lfloor \cdot \rfloor$	Partie entière
$\lceil \cdot \rceil$	Partie entière par excès
$\llbracket a, b \rrbracket$	$x \in \llbracket a, b \rrbracket := \{x \in \mathbb{Z} : a \leq x \leq b, \forall (a, b) \in \mathbb{Z}^2\}$
$a \bmod b$	Reste de la division euclidienne de a par b

$\Re(\cdot)$	Partie réelle
$\Im(\cdot)$	Partie imaginaire
$\mathcal{N}(\mu, \sigma^2)$	Loi normale de moyenne μ et de variance σ^2
rect	Fenêtre rectangulaire Pour tout $x \in \mathbb{R}$, $\text{rect}(x) = \begin{cases} 1 & \text{si } x \in [0, 1[\\ 0 & \text{sinon.} \end{cases}$
$(\cdot)^*$	Opérateur conjugué
$(\cdot)^T$	Opérateur transposé
$(\cdot)^H$	Opérateur hermitien (transpose conjugué)
$\ \cdot\ _p$	Norme p En particulier, $\ \mathbf{x}\ _1 = \sum_i x_i $ et $\ \mathbf{x}\ _\infty = \max_i (x_i)$
\circ	Produit d'Hadamard (produit terme à terme)
$\text{circ}(\mathbf{x})$	Matrice circulante spécifiée par le vecteur \mathbf{x} $\text{circ}(\mathbf{x}) := \begin{bmatrix} x_1 & x_2 & \dots & x_{n-1} & x_n \\ x_n & x_1 & x_2 & & x_{n-1} \\ \vdots & x_n & x_1 & \ddots & \vdots \\ x_3 & & \ddots & \ddots & x_2 \\ x_2 & x_3 & \dots & x_n & x_1 \end{bmatrix}.$
$\text{diag}(\mathbf{x})$	Matrice diagonale spécifiée par le vecteur \mathbf{x}
$\text{diag}(\mathbf{X})$	Vecteur défini comme la diagonale de la matrice \mathbf{X}
$\text{rank}(\mathbf{X})$	Rang de la matrice \mathbf{X}
$\text{Tr}(\mathbf{X})$	Trace de la matrice \mathbf{X}
\mathcal{F}	Transformée de Fourier
\mathcal{L}	Opérateur Lagrangien
∇f	Gradient de la fonction f
$q \mapsto f_s(q)$	Application partielle de f en s associée à q Définie pour toute fonction f dépendant d'au minimum deux variables (ici, s et q).
*	Opérateur de convolution Pour deux fonctions f et g , $(f * g)(x) = \int_{-\infty}^{+\infty} f(x-t)g(t) dt.$

Abréviations

ACF	<i>Autocorrelation Function</i>
CCF	<i>Cross-Correlation Function</i>
CDMA	<i>Code Division Multiple Access</i>
CMA-ES	<i>Covariance Matrix Adaptation Evolution Strategy</i>
DBF	<i>Digital Beamforming</i>
DSO	<i>Defence Science Organisation</i>
EUSIPCO	<i>European Signal Processing Conference</i>
FDMA	<i>Frequency Division Multiple Access</i>
FFT	<i>Fast Fourier Transform</i>
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
GRETSI	<i>Groupe d'études du traitement du signal et des images</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IET	<i>Institution of Engineering and Technology</i>
IGO	<i>Information-Geometric Optimization</i>
INRIA	<i>Institut national de recherche en informatique et en automatique</i>
IRS	<i>International Radar Symposium</i>
ISL	<i>Integrated Sidelobe Level</i>
L-BFGS	<i>Limited-memory BFGS (Broyden-Fletcher-Goldfarb-Shanno)</i>
LCZS	<i>Low Correlation Zone Sequences</i>
LPG	<i>Loss-in-Processing Gain</i>
MF	<i>Merit Factor</i>
MIMO	<i>Multiple-Input Multiple-Output</i>
MISO	<i>Multiple-Input Single-Output</i>
NLFM	<i>Non-Linear Frequency Modulation</i>
NUS	<i>National University of Singapore</i>
ONERA	<i>Office national d'études et de recherches aérospatiales</i>
PAPR	<i>Peak-to-Average Power Ratio</i>
PSLR	<i>Peak-to-Sidelobe Level Ratio</i>
QCQP	<i>Quadratically Constrained Quadratic Program</i>
RADAR	<i>Radio Detection and Ranging</i>
RIAS	<i>Radar à impulsion et antenne synthétique</i>
SAR	<i>Synthetic Aperture Radar</i>
SER	<i>Surface équivalente radar</i>
SIMO	<i>Single-Input Multiple-Output</i>
SINR	<i>Signal-to-Interference-plus-Noise Ratio</i>
SISO	<i>Single-Input Single-Output</i>
SNR	<i>Signal-to-Noise Ratio</i>
SONDRA	<i>Supélec ONERA NUS DSO Research Alliance</i>
TDMA	<i>Time Division Multiple Access</i>
ZCZS	<i>Zero Correlation Zone Sequences</i>

Introduction

Cadre Le MIMO, *kezako?* MIMO est un acronyme pour *Multiple-Input Multiple-Output*, que l'on pourrait traduire par « entrées multiples, sorties multiples ». Ce terme décrit donc des systèmes composés de plusieurs éléments, éléments pouvant être utilisés de manière unique et indépendante. Ils s'opposent ainsi aux systèmes SISO (*Single-Input Single-Output*), comme l'illustre la Figure I.1 ci-dessous...

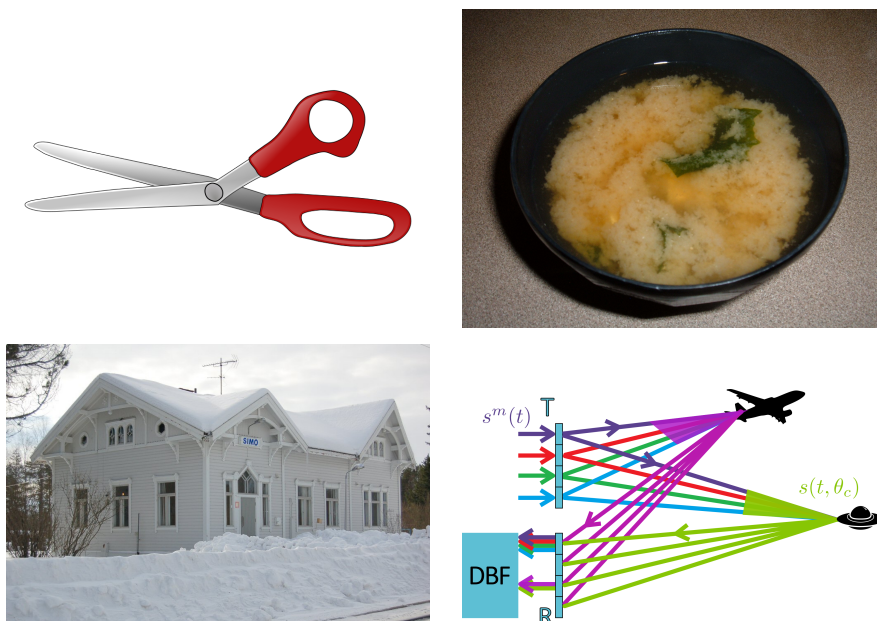


FIGURE I.1 – Différents systèmes

De haut en bas, de gauche à droite : un système SISO ; « Miam ! » ; la gare de Simo, en Finlande ; un système MIMO. Un radar MIMO est donc composé de plusieurs émetteurs, dont chacun a sa propre forme d'onde, représenté par une couleur différente.

Le concept MIMO a été introduit dans les années 70 dans le domaine des télécommunications, pour répondre à des problématiques dont certaines peuvent être transposées à notre époque. Face à l'essor des *smartphones* et autres technologies de communication, comment peut-on gérer l'augmentation du nombre d'utilisateurs ou la quantité de données requises et échangées ? Comment peut-on répondre au besoin de rapidité ou de débit ? La solution la plus naturelle consiste à utiliser plusieurs émetteurs (antennes relais) et plusieurs récepteurs (dans un même téléphone) en simultanée. Le radar MIMO s'est inspiré de cette solution, en ouvrant la possibilité d'émettre des formes d'onde différentes pour les différentes antennes d'émission d'un même radar (Figure I.1). Cette stratégie nécessite par contre une modification non négligeable de la chaîne de traitement radar, et de surcroît un choix de l'ensemble des signaux (les formes d'onde) émis.

De nombreuses études ont été effectuées sur la recherche de formes d’onde, appliquées en majorité pour le radar monostatique (le radar classique, en somme). Elles ont notamment été motivées par les avancées et les obstacles matériels. Le radar MIMO ne fait pas exception. Malheureusement, quelques difficultés s’y ajoutent : la plus évidente concerne la multiplicité des formes d’onde (une famille de signaux doit être déterminée) et les éventuelles interactions (à l’instar des mélanges entre signaux élémentaires, comme représenté à la Figure I.1) entre elles. Afin de prendre en compte l’ensemble des contraintes induites, une approche convenable consiste en la modélisation d’un problème d’optimisation, où les variables de recherche sont précisément ces formes d’onde.

Avec la puissance grandissante des moyens de calcul, l’optimisation a le vent en poupe, notamment avec ses dérivés, les plus connus étant le *Machine Learning* et le *Deep Learning*. L’optimisation est une branche des mathématiques qui s’intéresse à la modélisation (à partir d’équations) et à la résolution (de manière analytique ou numérique, à l’aide d’algorithmes) d’un problème, traduit sous forme d’une minimisation (ou encore d’une maximisation) d’une fonction objectif. Par exemple, concernant la génération de signaux, la fonction objectif peut se définir comme une mesure permettant d’établir les performances d’un radar *selon* les formes d’onde choisies. La recherche de son optimum, *i.e.* la résolution du problème de minimisation induit, offre des formes d’onde intéressantes.

Mais, qu’est ce qu’une forme d’onde « performante » ? Pour définir cela, il faut en revenir à l’essence même du radar : son fonctionnement et son traitement. *Ça s’en va, et ça revient.* Un radar émet un signal, une impulsion électromagnétique, dans son champ de vision. Son retour révèle la présence d’un objet potentiel, qualifié de « cible ». *C’est fait de tout petits rien.* Cependant, le signal perçu est généralement faible en comparaison, en terme d’énergie. Il faut donc effectuer un traitement, permettant de récupérer au mieux l’énergie du signal émis (par rapport au bruit). Le traitement maximisant cette énergie est appelé filtre adapté. Cela dit, la sortie après filtrage adapté se distingue selon le signal choisi : elle peut présenter des prééminences (appelées lobes secondaires) plus ou moins prononcées, pouvant engendrer une dégradation des capacités de détection ou des fausses alarmes. En d’autres termes, les performances d’une forme d’onde peuvent se mesurer à travers le niveau des lobes secondaires que celle-ci propose en sortie de filtre.

Finalement, le problème d’optimisation qui nous intéressera dans cette thèse consiste en la recherche d’un ensemble de séquences pour le radar MIMO (les variables), minimisant principalement le niveau des lobes secondaires après filtrage (la fonction objectif).

Contexte Revenons sur les radars MIMO. Si on y réfléchit, on pourrait naïvement mais légitimement insinuer que le plus sommaire d’entre eux consiste à rassembler côte à côte plusieurs radars. Alors, l’union fait-elle la force ? En ce sens, osons une petite transition maladroite. Cette thèse est issue d’une collaboration entre plusieurs entités (Figure I.2) : Thales (thèse initialement proposée par Thales Research and Technology et poursuivie par Thales Land & Air Systems), le laboratoire SONDRRA (CentraleSupélec, Université Paris-Saclay) et l’ONERA. Elle a été financée en partie par une bourse DGA-MRIS.

Contenu Le présent document a pour intitulé « Méthodologies de conception de formes d’onde pour radars sols — Application au cas du radar MIMO ». Les différentes expressions qui composent ce titre décrivent quelque peu le contexte et l’objectif de cette thèse.

- Le concept du radar MIMO cohérent constitue l’application principale de cette thèse.
- Habituellement mentionnée par l’anglicisme *waveform design*, la conception de formes d’onde mêle la recherche, l’évaluation et la sélection de signaux performants. Adaptés pour et appropriés au radar MIMO, ici.
- Pour cela, une modélisation sous la forme d’un problème de minimisation est suggérée. Elle nécessite l’expression d’une fonction objectif, mesurant les lobes secondaires de la fonction de corrélation. La résolution de ce problème fournit les signaux souhaités. Le manuscrit est organisé en cinq chapitres, suivant une certaine progression hiérarchique.



FIGURE I.2 – L’union fait la force

Le laboratoire SONdra est le fruit d’une collaboration entre Supélec, l’ONERA, la NUS (National University of Singapore) et le DSO (Defence Science Organisation).

Le problème d’intérêt sur les formes d’onde MIMO est atteint à partir de la suggestion d’un problème élémentaire, auquel on greffe des hypothèses et des contraintes au fur et à mesure.

- Le Chapitre 1 débute avec quelques généralités sur le traitement radar. Il s’intéresse ensuite au radar MIMO, en détaillant notamment la construction de la fonction d’ambiguïté MIMO cohérent, et en y essayant quelques formes d’onde connues. Ce chapitre n’a en aucun cas la prétention d’être complet, puisque son seul but est d’introduire les termes favorisant à la compréhension de la suite.
- Le Chapitre 2 propose un problème d’optimisation élémentaire. Basé sur le niveau des lobes secondaires des séquences de corrélation, sa résolution permet d’obtenir des signaux avantageux — des codes de phase, en particulier. Ce problème d’optimisation est élémentaire, car tous les autres problèmes de ce manuscrit en découlent. Mais il est difficile (non convexe) à résoudre. De ce fait, différentes méthodes d’optimisation seront présentées et comparées (Figure I.3).



FIGURE I.3 – Illustration des méthodes d’optimisation

A gauche : une descente de gradient. Descendre la pente d’une fonction vous permettra d’atteindre votre but... ou pas. A droite : une méthode stochastique. En cherchant (fusillant) partout, vous atteindrez votre cible... ou pas.

- Le Chapitre 3 présente quelques contraintes et variantes, pouvant s’incorporer au problème initial. Elles permettent de traiter des problématiques particulières, *via* des procédures d’optimisation, comme l’ajout de contraintes spectrales ou la robustesse aux phénomènes d’éclipse. Les effets de la quantification de phases sont aussi étudiées.
- Le Chapitre 4 porte sur un filtrage alternatif, qualifié de désadapté (Figure I.4) et caractérisé par une propriété forte intéressante : pour une séquence *donnée*, il existe un filtre désadapté optimal. Afin de générer des couples séquences/filtres performants, deux méthodologies d’optimisation jointe sont proposées. La première emploie un gabarit pour contrôler le niveau des lobes secondaires et la perte due à la désadaptation.



FIGURE I.4 – Filtrage adapté, filtrage désadapté

A gauche : ce qu'on cherche à voir (ou à faire, allégorie du signal émis, filtre adapté).

A droite : ce qu'on peut observer (signal reçu, bruité, avec une fausse alarme).

En bas : différents filtres facilitant la recherche de la cible (filtres désadaptés — visuels, ici).

Crédits : Warner Bros.

La seconde exploite la propriété sus-mentionnée.

- En regroupant l'ensemble de ces études, le Chapitre 5 révèle une procédure permettant d'obtenir des séquences appropriées pour le radar MIMO cohérent : le dénouement de cette thèse, en somme.
- En fil rouge, deux bornes ont été suggérées dans les Chapitres 3 et 4, afin de caractériser les performances atteignables. La première concerne le niveau des lobes secondaires d'un ensemble de séquences. En particulier, la borne de Levenshtein a été généralisée pour les séquences polyphases, et même légèrement améliorée. La seconde étudie le niveau des lobes d'un ensemble de séquences associées à des filtres désadaptés. À notre connaissance, il n'existait aucune borne à ce sujet...

Une conclusion résume tout cela, tente de susciter des questionnements et propose quelques perspectives. Enfin, deux petites annexes donnent des indications sur les démonstrations sur les bornes.

Publications Les travaux effectués durant cette thèse ont fait (et feront, espérons-le) l'objet de publications dans des congrès et/ou dans des revues, classées ci-dessous par ordre chronologique.

1. U. Tan, C. Adnet, O. Rabaste, F. Arlery, J.-P. Ovarlez, and J.-P. Guyvarch (2016b). Phase Code Optimization for Coherent MIMO Radar Via a Gradient Descent. In *2016 IEEE Radar Conference (RadarConf)*, pages 1–6 ;
2. F. Arlery, R. Kassab, U. Tan, and F. Lehmann (2016a). Efficient Gradient Method for Locally Optimizing the Periodic/Aperiodic Ambiguity Function. In *2016 IEEE Radar Conference (RadarConf)*, pages 1–6 ;
3. U. Tan, O. Rabaste, C. Adnet, F. Arlery, and J.-P. Ovarlez (2016c). Optimization Methods for Solving the Low Autocorrelation Sidelobes Problem. In *2016 17th International Radar Symposium (IRS)*, pages 1–5 ;

4. F. Arlery, R. Kassab, U. Tan, and F. Lehmann (2016b). Efficient Optimization of the Ambiguity Functions of Multi-Static Radar Waveforms. In *2016 17th International Radar Symposium (IRS)*, pages 1–6;
5. U. Tan, C. Adnet, O. Rabaste, F. Arlery, J.-P. Ovarlez, and J.-P. Guyvarch (2016a). Coherent MIMO Waveform Design with a Gradient Descent. In *SONDRA 4th Workshop*;
6. U. Tan, O. Rabaste, C. Adnet, and J.-P. Ovarlez (2017a). A Joint Optimization for Coherent MIMO Radar. In *International Conference on Radar Systems (Radar 2017)*, pages 1–6;
7. U. Tan, O. Rabaste, C. Adnet, and J.-P. Ovarlez (2017b). Optimisation jointe d’une séquence et d’un filtre désadapté. In *GRETSI 2017*;
8. U. Tan, O. Rabaste, C. Adnet, and J.-P. Ovarlez (2018). A Sequence-Filter Joint Optimization. In *26th European Signal Processing Conference (EUSIPCO 2018)*, pages 1–5;
9. F. Arlery, U. Tan, and O. Rabaste (2019). Generalization and Improvement of the Levenshtein Lower Bound for Aperiodic Correlation. *IEEE Transactions on Information Theory*, 65(2):1258–1266;
10. U. Tan, F. Arlery, O. Rabaste, F. Lehmann, and J.-P. Ovarlez (2019a). A New Lower Bound on the Maximum Correlation of a Set with Mismatched Filters. *IEEE Transactions on Information Theory*. (Submitted);
11. U. Tan, O. Rabaste, C. Adnet, and J.-P. Ovarlez (2019b). Designing Phase Codes for Coherent MIMO Radar. *IET Radar, Sonar and Navigation*. (To be submitted);
12. U. Tan, O. Rabaste, C. Adnet, and J.-P. Ovarlez (2019c). On the Eclipsing Phenomenon with Phase Codes. In *2019 International Radar Conference (Radar 2019)*. (Accepted).

Chapitre 1

Introduction sur le radar

1.1 Généralités

1.1.1 Aux origines du radar

Le radar est un système utilisant les ondes électromagnétiques pour détecter et localiser des « objets », tels que des avions, des navires... ou bien même des personnes.

La seconde moitié du XIX^{ème} siècle a vu l'essor des travaux sur l'électromagnétisme, portés notamment par les découvertes de James Maxwell [Skolnik, 2001]. En ont découlé les premières applications dans le domaine des télécommunications, ou encore les premiers systèmes de radionavigation. En outre, Nikola Tesla a remarqué que les ondes électromagnétiques permettaient de déceler et de mesurer le mouvement des objets métalliques [Darricau *et al.*, 2003]. Leur utilisation dans un système de détection a semblé ainsi adéquate, d'autant plus que, contrairement à ceux employant la radionavigation, la coopération des objets observés n'est pas nécessaire.

Les premiers systèmes opérationnels sont apparus simultanément dans les années 30, de part et d'autre de l'Atlantique, principalement à des fins militaires (défense aérienne et maritime). Notons une grande diversité dans le choix des paramètres : nombre et positionnement des antennes (système monostatique ou bistatique — ces termes seront précisés plus loin), fréquence du signal, durée de l'impulsion...

C'est lors de la Seconde Guerre Mondiale qu'a été adopté le nom pour lequel la détection électromagnétique est connue : le « radar », pour *RAdio Detection And Ranging*.

Depuis, la technologie a bien évolué et le radar s'est considérablement diversifié, comme en témoigne la liste suivante — très incomplète — des nombreuses applications civiles ou militaires [Skolnik, 2001] :

- radars de couverture aérienne : contrôle aérien, surveillance militaire ;
- radars de trajectoire : pistage d'une cible, stations aérospatiales ;
- radars d'atterrissage : altimètre radio ;
- radars de navigation : radars maritimes, voitures autonomes ;
- télédétection (plus connu sous l'anglicisme *Remote Sensing*) : cartes météorologiques, imagerie SAR (*Synthetic Aperture Radar*) ;
- cinémomètres de la police.

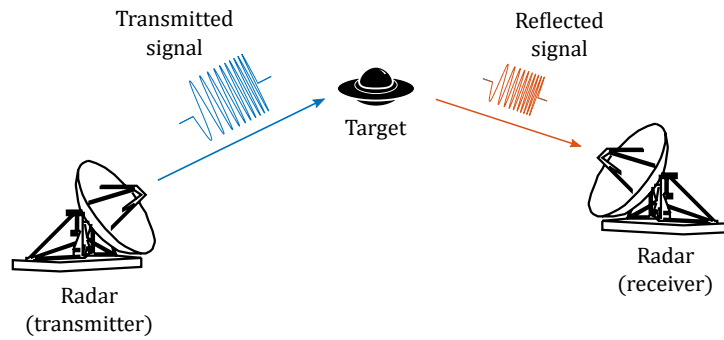


FIGURE 1.1 – Principe du radar
*Si une cible il y a, le récepteur reçoit le signal émis par l'émetteur
 (mais légèrement perturbé).*

1.1.2 Principes de base

Une bonne caractérisation du radar est donnée par son acronyme :

- *radio* : le radar utilise des ondes radioélectriques ;
- *detection* : son rôle premier est de détecter, *i.e.* de révéler la présence d'une cible ;
- *ranging* : ce terme évoque une notion de mesure car, après détection, le radar peut s'attarder à déterminer quelques paramètres de la cible, comme sa localisation ou sa vitesse. D'ailleurs, le terme *range* peut se traduire par « distance » ou bien « portée ».

Cette sous-section présente quelques notions sur le radar, de manière très simpliste, afin d'introduire les premières notations et autres terminologies. Pour plus de précisions, nous vous invitons à vous référer à [Darricau, 1994] et à [Skolnik, 2001].

Remarque 1.1. (*Cible*) Comme le sous-entend la précédente liste des applications radars, le terme « cible » n'a pas nécessairement une connotation militaire...

Détection Comme sus-mentionné, le but premier d'un radar est de détecter la présence d'une cible potentielle, en utilisant la propriété des ondes électromagnétiques à se réfléchir sur tout obstacle. Pour cela, le radar est composé d'au moins un émetteur et un récepteur, comme l'indique la Figure 1.1.

Localisation en distance La localisation en distance est obtenue en utilisant un principe analogue à celui utilisé par certains animaux (la chauve-souris par exemple) avec les ondes sonores : la mesure du décalage temporel τ entre l'émission et la réception d'un signal, dû au délai de propagation, représenté Figure 1.2. Ici, l'impulsion radio se propage dans l'espace à la célérité $c = 299\,792\,458$ m/s. En supposant que l'émetteur et le récepteur sont situés au même endroit, la distance d séparant le radar de la cible s'écrit (pour une cible immobile) :

$$d = c \frac{\tau}{2}. \quad (1.1)$$

Le facteur $1/2$ caractérise le trajet aller-retour de l'onde. Du fait de cette proportionnalité, la variable retard τ est également désignée abusivement par « distance »...

Vitesse La vitesse radiale de la cible v_r entraîne une modification du signal émis, comme l'illustre la Figure 1.3. Pour être plus précis, elle provoque un décalage de fréquence ν du signal émis, appelé « effet Doppler », qui vaut (putativement à $v_r \ll c$) :

$$\nu = \frac{2v_r f_c}{c}, \quad (1.2)$$

où f_c la fréquence porteuse du radar.

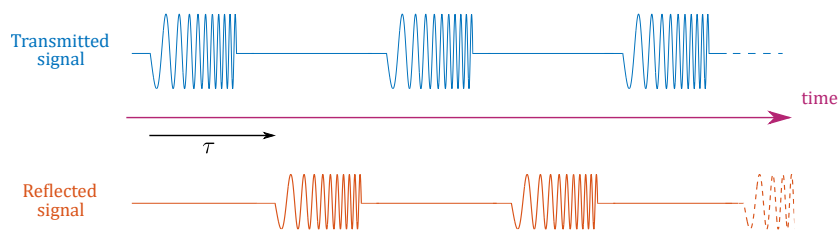


FIGURE 1.2 – Détermination de la distance

Le temps τ entre l'émission et la réception est proportionnel à la distance séparant le radar de la cible.

Localisation angulaire Outre la localisation en distance, il est généralement nécessaire de connaître la direction angulaire de la cible, définie par l'azimut φ et l'élévation θ (Figure 1.4). Il existe plusieurs méthodes, telles que l'écartométrie [Darricau, 1994] ou la triangulation [Poirot *et al.*, 1978], mais la plus accessible consiste à se baser sur la directivité du radar. Cette dernière décrit la disposition d'un radar à focaliser l'énergie émise dans un faisceau étroit. Une bonne directivité permet d'améliorer la localisation angulaire, comme le montre la Figure 1.5. Elle est en outre reliée à la dimension de l'antenne et à la longueur d'onde λ .

Autres S'il dispose des voies de réception adéquates, le radar peut également effectuer une analyse polarimétrique (horizontale, verticale, circulaire, *etc.*) de manière à déterminer et différencier les structures d'un environnement (bâtiments, végétation) [Couderc *et al.*, 2016], voire même imager celui-ci (imagerie SAR) afin d'observer, par exemple, les changements d'une scène au cours du temps [Mian *et al.*, 2018]. Ces multiples applications dépassent le cadre de cette thèse.

Remarque 1.2. (*Train d'impulsions*) Dans ce manuscrit, on ne considère que les radars à impulsions : un même signal, qualifié d'« impulsion », est émis et répété avec une certaine période, comme illustré à la Figure 1.2. On les oppose aux radars à émission dite continue.

1.1.3 Le traitement radar

Mais revenons à la composante distance, Figure 1.2. Celle-ci s'établit à partir du retard τ , qu'il faut ainsi déterminer. Malheureusement, sa lecture n'est pas si immédiate en pratique, à cause des perturbations qui polluent et modifient le signal reçu.

- Bilan énergétique. Un ratio important apparaît entre l'énergie émise et l'énergie reçue. Ce ratio, défini par l'équation radar [Darricau, 1994], est notamment proportionnel à d^4 (d étant la distance entre le radar et la cible), ce qui signifie que le signal que l'on souhaite déceler présente une puissance infime (relativement à celle émise)... Par ailleurs, il est influencé par la SER de la cible. Pour information, cette SER (Surface équivalente radar) décrit la capacité d'un objet à réfléchir les rayonnements électromagnétiques.
 - Bruit. Ce terme décrit un certain nombre de facteurs gênant la réception du signal utile, que l'on catégorise en trois espèces. Premièrement, le bruit thermique (ou bruit interne) est généré par la chaleur du récepteur, limitant ainsi sa sensibilité. Deuxièmement, l'environnement naturel (nuages, surface de la mer, sol...) donne naissance à des échos parasites, qui s'accumulent et se confondent dans le signal reçu. Leur accumulation est appelée « fouillis » ou « clutter ». Enfin, le brouillage consiste à produire un bruit intentionnel, afin de polluer et de cacher le retour de la cible.
 - Et d'autres altérations, comme le Doppler de la cible (*cf.* le paragraphe précédent).
- Ainsi, il est nécessaire d'effectuer une procédure de traitement spécifique afin d'extraire le signal émis du signal reçu. Cela est expliqué dans le paragraphe suivant.

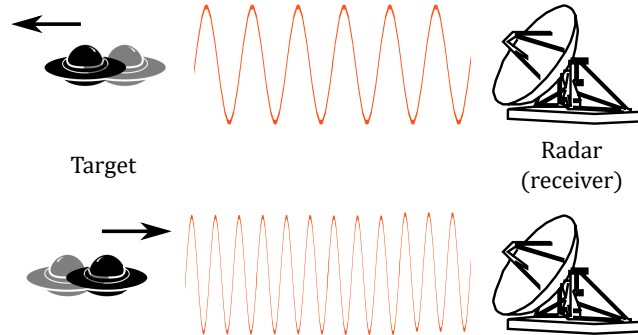


FIGURE 1.3 – Effet Doppler

En haut, la cible s'éloigne du récepteur, la fréquence diminue. En bas, l'inverse.

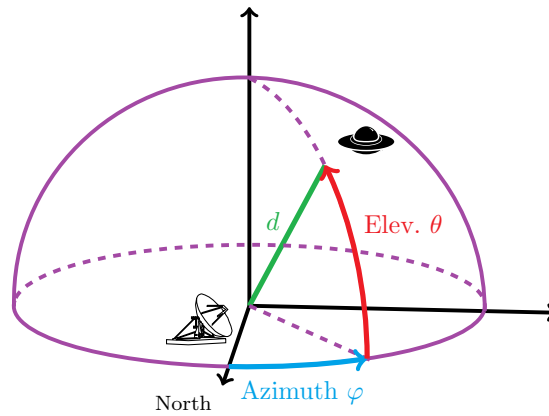


FIGURE 1.4 – Représentation des directions angulaires azimuth φ et élévation θ
Azimuth : angle dans le plan horizontal. Élévation : angle avec le plan horizontal.

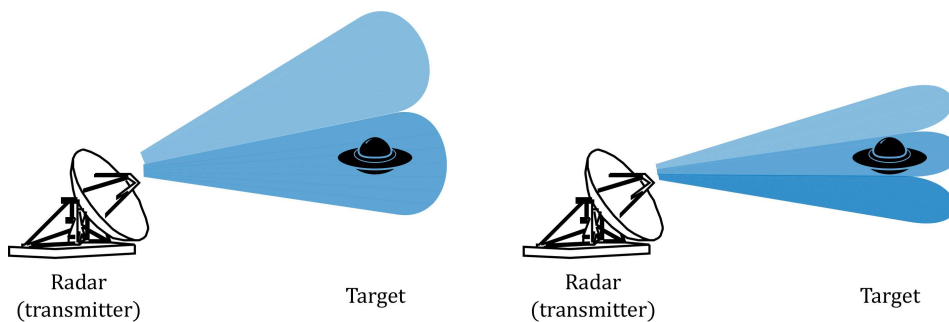


FIGURE 1.5 – Approximation de la position angulaire

Un radar peut explorer l'espace à l'aide d'un faisceau plus ou moins étroit, en le déplaçant électroniquement ou mécaniquement. La qualité de l'approximation de la position angulaire est liée à la directivité de l'antenne.

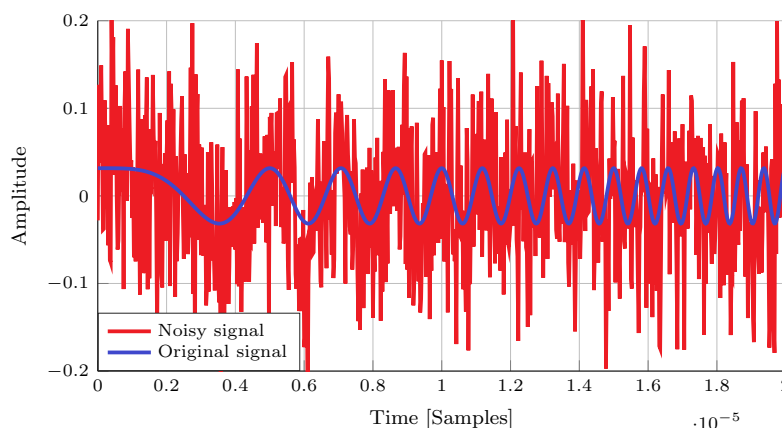


FIGURE 1.6 – Illustration d'un signal bruité

Le signal émis, une modulation linéaire de fréquence, est représenté en bleu. On y ajoute du bruit : le signal obtenu (« le signal reçu ») est en rouge.

1.1.3.1 Filtrage adapté

Afin de détecter la présence d'une cible, le traitement radar de base consiste à déceler la présence du signal émis dans le signal qui a été reçu. La probabilité de détection étant reliée au rapport signal sur bruit (ou SNR, pour *Signal-to-Noise Ratio*), il est d'intérêt de trouver le filtre qui permet de maximiser ce SNR. Ce dernier existe et est unique — dans le cas d'un bruit blanc additif — : il est nommé « filtre adapté » et correspond peu ou prou au signal émis [Levanon *et al.*, 2004, North, 1963]. En radar, on parle également de *compression d'impulsion*.

Le filtrage adapté consiste donc à calculer la corrélation entre le signal reçu et le signal émis, qu'on note $s \in L^2(\mathbb{C})$. Cela s'écrit :

$$\left\{ \begin{array}{l} \text{ACF} : L^2(\mathbb{C}) \rightarrow L^2(\mathbb{C}), \\ \quad s \mapsto \text{ACF}(s) := s * l^*(s), \\ \text{ACF}(s) : \mathbb{R} \rightarrow \mathbb{C}, \\ \quad \tau \mapsto \text{ACF}(s)(\tau) := \int_{-\infty}^{+\infty} s(t)s^*(t - \tau)dt. \end{array} \right. \quad (1.3)$$

* désigne la convolution, présentée dans les notations, tandis que $l : L^2(\mathbb{C}) \rightarrow L^2(\mathbb{C})$ est un opérateur « inverse », *i.e.* $l(s(t)) := s(-t)$ pour tout $t \in \mathbb{R}$. La fonction $\text{ACF}(s)$ est appelée fonction d'autocorrélation associée au signal s , et est bien définie mathématiquement car ce dernier est borné et à support fini.

Comment interpréter cette fonction ? Comme le montre l'expression (1.3), la corrélation est une fonction du paramètre retard. Elle évalue bien la ressemblance entre le signal reçu et le signal émis, mais en décalant ce dernier dans le temps. Ainsi, lorsque les deux signaux sont parfaitement synchronisés, on obtiendra une corrélation à son maximum, signe du retard τ adéquat, et donc de la distance avec la cible. Une illustration de cette fonction est présentée à la Figure 1.7 (celle-ci est associée avec la Figure 1.6).

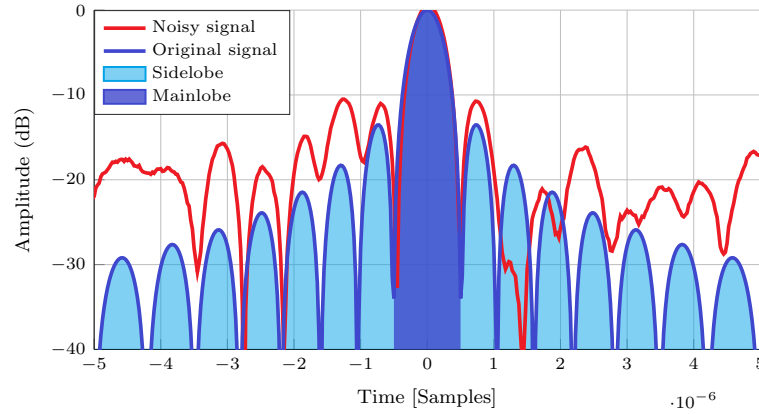


FIGURE 1.7 – Corrélation d'un signal bruité

Les lobes principaux et secondaires sont représentés pour le chirp non bruité. Malgré l'application d'un bruit additif gaussien, l'argument maximal de la fonction de corrélation est (relativement) identique.

Les courbes sont recentrées en 0, et l'axe des ordonnées est en dB.

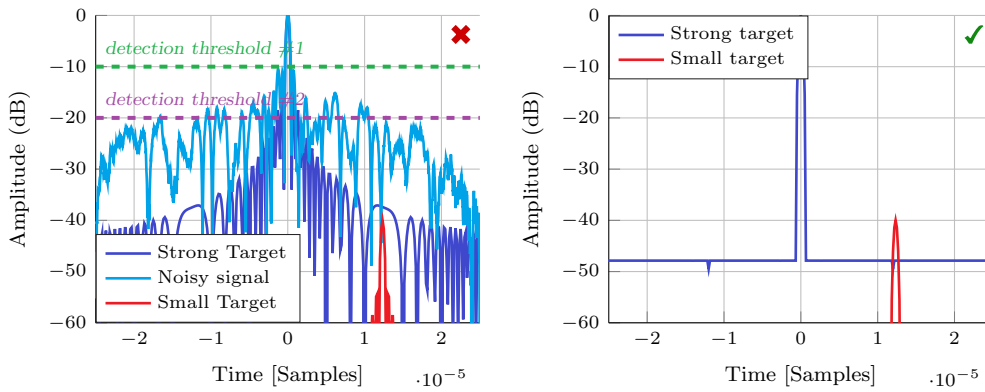


FIGURE 1.8 – Détection et corrélation

Des lobes secondaires trop importants peuvent induire des fausses alarmes (lobes secondaires au dessus du seuil de détection), ou enfouir la réponse d'une cible plus faible.

Ainsi, à gauche, la cible n'est pas détectée. Elle l'est à droite.

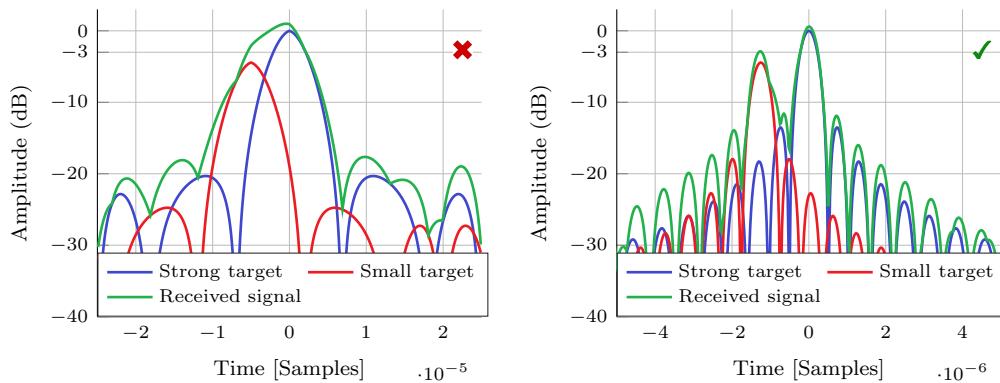


FIGURE 1.9 – Résolution et corrélation

Une mauvaise résolution (à gauche) ne permet pas de distinguer les deux lobes principaux : deux cibles proches peuvent être donc inséparables.

Considérons une modulation linéaire de fréquence [Levanon *et al.*, 2004] — usuellement connue sous le nom de *chirp*, que l'on peut traduire par « gazouillis » —, représentée en bleu sur la Figure 1.6, et jouant le rôle de signal émis. Sa fonction de corrélation est similaire à un sinus cardinal, comme le montre la Figure 1.7. En inspectant l'axe des abscisses, on retrouve logiquement qu'elle est maximale à retard nul. Cette situation se rencontrerait dans un monde parfait, où le signal reçu serait identique à celui émis. En *admettant* que tel n'est pas le cas, on ajoute à ce dernier un bruit additif gaussien, en rouge. Concevez que le résultat, présenté à la Figure 1.6, en rouge, n'est pas visuellement ressemblant. Néanmoins, en effectuant un filtrage adapté, l'argument maximal reste proche du retard nul (le filtrage adapté semble donc fonctionner).

Terminons ce paragraphe avec un peu de nomenclature sur la sortie de cette fonction de corrélation. D'une part, le « lobe principal » et les « lobes secondaires » sont illustrés à la Figure 1.7. D'autre part, elle permet de relater certaines performances d'un radar, comme l'indique la remarque suivante.

Remarque 1.3. (*Performances et corrélation*) *Ci-dessous des propos sur les performances de détection et la résolution d'un radar.*

- *Comme le montrait la Figure 1.7 (reprise à la Figure 1.8), le filtrage adapté a permis de déceler la présence d'une cible, via l'existence d'un pic principal dans la fonction de corrélation. S'il n'y en avait pas, nous aurions pu affirmer le contraire... Ce critère, plutôt qualitatif, se transcrit en pratique à l'aide d'un seuil de détection (Figure 1.8). Pour faire simple, disons que tout pic au-dessus du seuil définit une cible. Dès lors, un seuil trop haut pourrait négliger certaines cibles (Figure 1.8, seuil n° 1), tandis qu'un autre trop bas induirait des fausses alarmes (seuil n° 2). Son choix est donc assez délicat, et dépasse le cadre de cette thèse. Néanmoins, sachez qu'il est influencé par la statistique liée au bruit. Pour augmenter les performances de détection, il est préférable d'avoir les lobes secondaires les plus faibles possibles, comme le montre la Figure 1.8, droite, où la cible faible peut plus aisément être détectée.*
- *La résolution décrit la capacité à discerner et à dissocier une cible d'une autre. On l'évalue ici à l'aide de l'épaisseur du lobe principal, mesurée par convention à 3 dB (Figure 1.9). La finesse est préférable de surcroît, sous deux aspects. Sur la précision de l'estimation des paramètres d'une part, car la variance de l'estimation dépend de la largeur du lobe principal. Sur la capacité de discernement d'autre part, puisqu'un lobe trop épais pourrait mener à une confusion entre deux cibles proches (seule une cible sera considérée, Figure 1.9, gauche). Dans les chapitres suivants, nous étudierons des codes de phase, pour lesquels la résolution est notamment prédéterminée par la bande de fréquence allouée au radar.*

Finalement, certaines performances d'un signal peuvent être caractérisées sans prendre en compte le bruit ou toute autre perturbation. Pour cette raison, la suite de cette thèse ignorera ces considérations, bien qu'importantes en pratique.

1.1.3.2 Fonction d'ambiguïté

Comme l'a montré le paragraphe précédent, le filtre adapté, *via* la fonction de corrélation, permet de récupérer la composante distance d'une cible. De même, il serait possible de mesurer d'autres paramètres, à l'instar de la vitesse. Ce n'est pas ce que l'on fait en pratique : le traitement *de base* s'effectue en effet en supposant la phase liée au Doppler constante sur l'impulsion, et en considérant ce changement de phase uniquement d'impulsion à impulsion. Néanmoins, il est nécessaire d'évaluer l'influence de ladite composante sur le filtrage.

Reprenons ainsi l'expression (1.3) et incorporons la variable Doppler ν . Cela nous donne une fonction à deux variables (τ, ν) , appelée « fonction d'ambiguïté » de Woodward, du nom de la personne qui a formalisé le concept [Woodward, 1953]. Elle sera notée par la suite χ et s'écrit :

$$\chi(\tau, \nu) := \int_{-\infty}^{+\infty} s(t)s^*(t - \tau) \exp(j2\pi\nu t) dt. \quad (1.4)$$

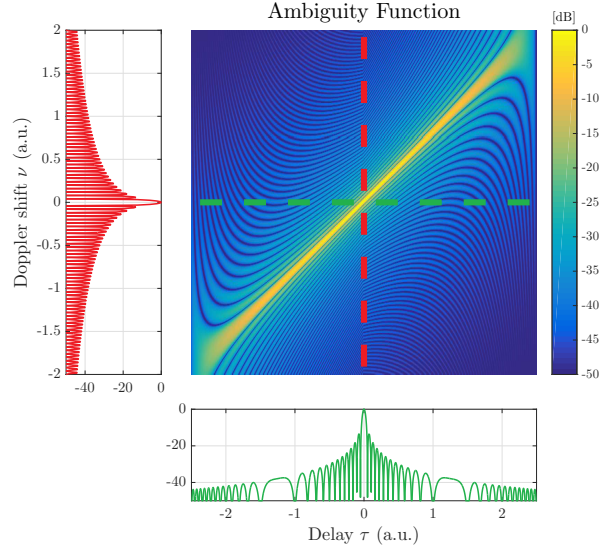


FIGURE 1.10 – Fonction d’ambiguïté distance/Doppler pour un chirp
*La valeur maximale se situe bien au centre.
 La position des coupes est représentée par un code couleur.*

La fonction d’ambiguïté se définit donc comme une extension de la fonction de corrélation, décrivant la sortie du filtre adapté classique en présence d’un signal provenant d’une cible régie d’un certain Doppler (sous le terme exponentiel). Elle reste par conséquent une bonne indicatrice des performances d’un radar. En ce sens, il est souvent admis qu’elle doit avoir l’aspect d’une punaise [San Antonio *et al.*, 2007] : maximale à la bonne hypothèse $(\tau, \nu) = (0, 0)$ — la Remarque 1.4 en témoigne — et la plus faible possible pour les autres choix de paramètres, afin d’éviter toute... ambiguïté.

Remarque 1.4. (*Conservation de l’énergie*) La fonction d’ambiguïté admet une kyrielle de propriétés, dont certaines sont citées dans [Skolnik, 2001]. Citons notamment les principales :

1. Par convention, un changement de coordonnées est appliqué de sorte que la cible est amenée en $(0, 0)$. La fonction d’ambiguïté est maximale en ce point et, par normalisation, on a $\chi(0, 0) = 1$.
2. A l’aide de quelques développements, on peut écrire :

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} |\chi(\tau, \nu)|^2 d\tau d\nu = 1. \quad (1.5)$$

A l’aide de la propriété précédente, on peut montrer qu’il n’est pas possible d’améliorer la fonction d’ambiguïté sans la détériorer ailleurs.

La Figure 1.10 représente la fonction d’ambiguïté pour notre *chirp*. Ici, l’axe horizontal correspond à une variation de la variable retard τ , l’axe vertical le Doppler ν . Sur les côtés sont présentées des coupes (*i.e.* une composante est fixée), pour $\tau = 0$ en rouge et $\nu = 0$ en vert. Cette dernière est d’ailleurs familière, puisqu’on peut reconnaître l’une des courbes de la Figure 1.7. Comme précisé ci-dessus, la cible est placée en $(0, 0)$, où elle atteint son maximum (d’une valeur normalisée de 0 dB, en jaune). Cependant, on s’aperçoit qu’il y a de nombreux points (τ, ν) où les valeurs de la fonction sont très élevées. Cette zone, appelée dans le jargon « lame de couteau » (spécifique à la modulation linéaire de fréquence), caractérise les ambiguïtés. En effet, comment savoir quel couple distance–Doppler est le plus crédible ?

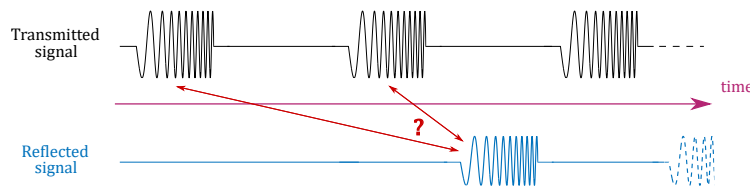


FIGURE 1.11 – Une ambiguïté distance
De quelle impulsion provient le retour ?

A une hypothèse temporelle donnée, on a certes une bonne idée de la vitesse. Mais comment déterminer si cette hypothèse est juste ?

Remarque 1.5. (*Ambiguïtés*) Le terme « ambiguïté » est lui-même ambigu, parce qu'il existe des ambiguïtés...

- ... dues à la forme d'onde, comme celle présentée par le chirp ;
- ... induites par les différentes impulsions (la Figure 1.11 est plus explicite).

Dans le même esprit, il existe également des phénomènes d'éclipse, où le signal émis n'est récupéré que partiellement. On discutera de cela plus en détails dans le Chapitre 3.

La fonction d'ambiguïté est définie selon les deux variables (τ, ν) . Bien qu'il n'apparaît pas en tant que tel, le choix du signal s est primordial, et influe directement sur les performances du système. Voilà pourquoi de nombreux travaux ont été et sont actuellement menés sur le design de formes d'onde... Comme cette thèse en fait !

1.2 Le radar MIMO cohérent

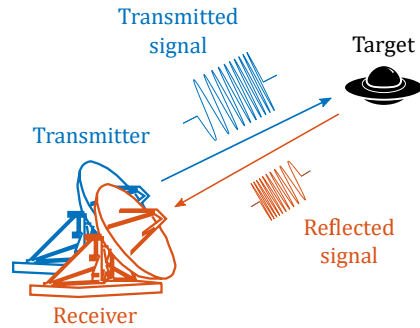
Le présent manuscrit a pour application objectif le radar MIMO cohérent. L'acronyme MIMO signifie *Multiple-Input Multiple-Output*, que l'on pourrait finalement traduire par « Entrées multiples, sorties multiples ». Qu'est-ce que cela peut-il caractériser ? Afin de répondre à cette interrogation, cette section introduit de manière succincte le radar MIMO cohérent, en trois points. On commence par catégoriser les différents radars selon la position et le nombre d'antennes qui les composent. Ensuite, on exhibe la fonction d'ambiguïté pour le radar MIMO cohérent, qui se trouve être légèrement différente de celle présentée à la Section 1.1.3.2. Enfin, on y essaie différentes formes d'onde, ce qui permettra de les présenter.

1.2.1 Classification des radars

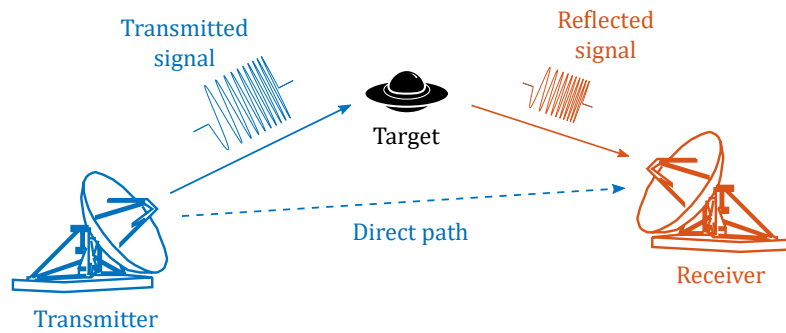
Débutons donc par une petite classification des radars, portant principalement sur la position et le nombre d'antennes. La liste suivante des caractéristiques est non exhaustive, et quelquefois non exclusive.

Primaire, secondaire Depuis le début — et la suite fera de même —, nous n'avons évoqué que les radars primaires, qui éclairent l'espace à l'aide de signaux électromagnétiques, afin de détecter et de localiser des cibles potentiellement non coopératives. On les oppose aux radars secondaires, qui interrogent et échangent des informations avec celles-ci (contrôle aérien). Mais ces derniers sont-ils vraiment des radars au sens premier du terme ?

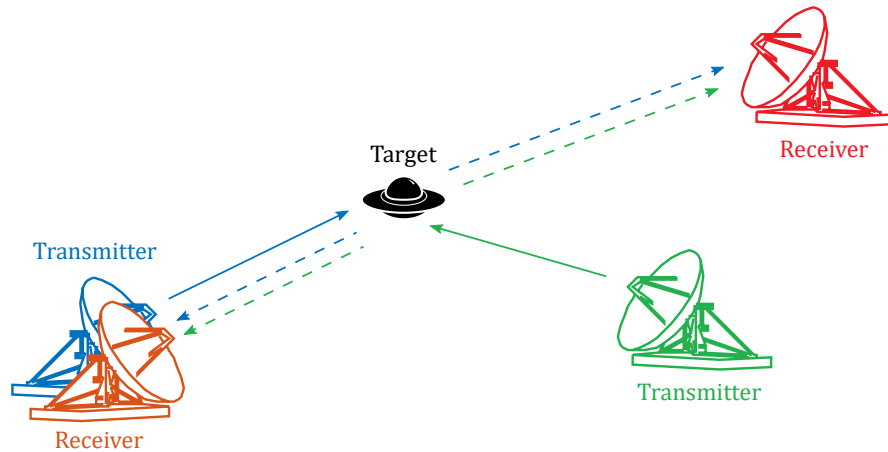
Actif, passif Depuis le début — et la suite fera de même —, nous n'avons évoqué que les radars... actifs, qui éclairent l'espace à l'aide de signaux électromagnétiques, afin de détecter et de localiser des cibles potentielles. Ils sont donc nécessairement composés d'un émetteur, dont ils maîtrisent le signal émis. Les radars passifs n'en ont pas : ils exploitent à la place des signaux d'opportunités (dont le but initial n'est pas la détection) comme ceux des diffusions radios ou de télévision.



Un radar monostatique : l'émetteur et le récepteur sont très proches (voire ne font qu'un). Pendant l'émission, le récepteur est inopérant.



Un radar bistatique : l'émetteur et le récepteur sont éloignés. Il faut veiller à ne pas prendre en compte le trajet direct...



Un radar MIMO multistatique, composé de plusieurs émetteurs et récepteurs qui peuvent être co-localisés ou non. Les récepteurs reçoivent les contributions de chacun des émetteurs.

FIGURE 1.12 – Radar monostatique, radar bistatique, radar MIMO multistatique

Monostatique, bistatique Se dit d'un radar dont l'émetteur et le récepteur sont *suffisamment* proches (radar monostatique) ou non (radar bistatique). Les éléments d'un radar monostatique perçoivent la cible sous le même angle (et la même SER *a fortiori*). Ce n'est pas le cas pour un radar bistatique ; en outre, les distances émetteur-cible et récepteur-cible sont potentiellement différentes. Une condition sur le terme « suffisamment », basée sur la distance entre les capteurs, est proposée dans [Fishler *et al.*, 2006], tandis que les deux configurations sont illustrées Figure 1.12.

SISO, MIMO... SISO est l'acronyme pour *Single-Input Single-Output*, que l'on pourrait traduire par « une seule entrée, une seule sortie ». Dans notre cadre, cela décrit un système radar avec un seul émetteur (l'entrée) et un seul récepteur (la sortie). Par extension, un radar MIMO, pour *Multiple-Input Multiple-Output*, est composé de plusieurs émetteurs et de plusieurs récepteurs [Lesturgie, 2011]. Il est en outre caractérisé par la diversité de ses formes d'onde, puisque chaque émetteur possède un signal qui lui est propre. Notez qu'on définit de la même manière les radars MISO et SIMO. Ces derniers comprennent les réseaux phasés « classiques », qui possèdent comme les radars MIMO plusieurs récepteurs et émetteurs, mais produisant le même signal (Figure 1.13).

Et il y a sans doute d'autres caractérisations : on pourrait par exemple distinguer un radar sol d'un radar maritime, du fait des différences matérielles... Quoi qu'il en soit, chacun des précédents attributs offre ses avantages et ses inconvénients. Pour en savoir plus, nous vous invitons à jeter un coup d'œil aux quelques références qui nous ont inspiré [Cherniakov, 2007, Darricau, 1994, Hack *et al.*, 2014]. Finalement, ce manuscrit se focalisera sur le radar MIMO primaire, actif, sol et cohérent (ou co-localisé [Li *et al.*, 2007], *i.e.* monostatique), qu'on oppose au radar MIMO distribué [Haimovich *et al.*, 2008] (multistatique). Il en existe également une version hybride [Fuhrmann *et al.*, 2009].

A ce propos, permettons-nous un bref interlude comparatif de ces deux systèmes MIMO. Mais avant, sachez, pour la petite histoire, que le concept MIMO a été suggéré dans les années 70 dans le domaine des télécommunications [Kaye *et al.*, 1970, Salz, 1985] — bien qu'il ne fut pas nommé comme tel. Il a ensuite été adopté dans le domaine radar dix ans plus tard avec le premier prototype français, le « RIAS », pour Radar à impulsion et antenne synthétique [Luce *et al.*, 1992]. Ce démonstrateur était composé de deux réseaux d'antennes circulaires, comme illustré à la Figure 1.16, pour la surveillance et le pistage.

MIMO statistique Le radar MIMO statistique (ou distribué) se base sur le principe de la « diversité spatiale ». Plusieurs antennes sont employées, suffisamment éloignées les unes des autres, qui illuminent de façon indépendante [Fishler *et al.*, 2006, Lehmann *et al.*, 2007] et observent sous de multiples coutures une même cible (par exemple, sa Surface équivalente radar varie selon l'angle à laquelle on la regarde). De plus, en tant que radar MIMO, chaque récepteur reçoit l'ensemble des contributions des émetteurs, comme l'illustre la Figure 1.12. Leur traitement est non cohérent : le rassemblement et le recouplement des données se fait seulement sur le module du signal reçu, temporellement. Il a été démontré dans [Haimovich *et al.*, 2008] que le radar MIMO statistique offre de meilleures performances de détection et de localisation [Lehmann *et al.*, 2006], et une meilleure robustesse envers les cibles mouvantes.

MIMO cohérent Par opposition, les antennes d'un radar MIMO cohérent (ou co-localisé) sont rapprochées. À l'émission, cela signifie de fait qu'une cible reçoit de manière simultanée l'ensemble des contributions des émetteurs. Ce signal résultant se trouve être en plus différent selon la direction d'émission (*cf.* Figure 1.14 et 1.15). À la réception, le traitement s'établit de façon cohérente, comme pour un réseau phasé. Nous préciserons tous ces points dans la sous-section suivante, où nous poserons en outre quelques équations caractérisant le radar MIMO cohérent. Notez que ce radar offre notamment une meilleure résolution angulaire [Li *et al.*, 2007] qu'un réseau phasé classique présentant un diagramme d'émission large.

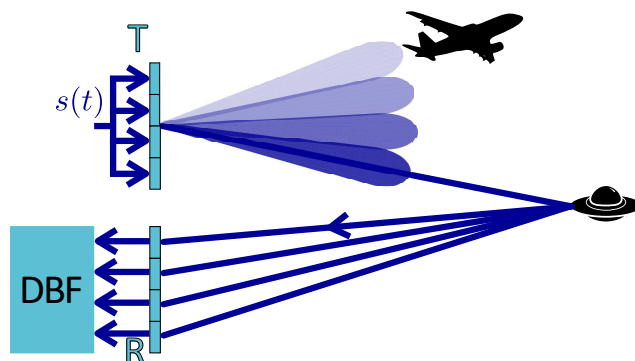


FIGURE 1.13 – Principe du réseau phasé classique
Le signal élémentaire est identique sur chaque antenne d'émission. Le faisceau émis est alors focalisé; et la couverture de l'espace se fait par balayage.

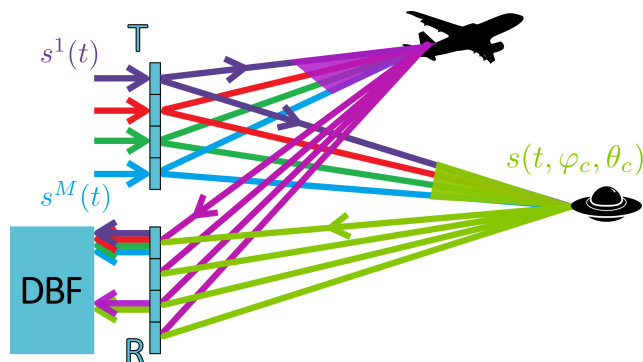


FIGURE 1.14 – Principe du radar MIMO cohérent (1)
Chaque émetteur émet sa propre forme d'onde, représentée par différentes couleurs. Chaque cible reçoit ainsi, selon sa position, une somme pondérée — potentiellement différente — des signaux élémentaires.

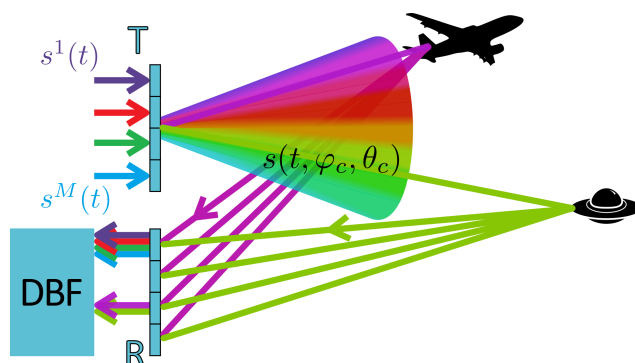


FIGURE 1.15 – Principe du radar MIMO cohérent (2)
*D'un point de vue macroscopique, le radar MIMO cohérent émet un signal différent dans chaque direction de l'espace.
 DBF : Digital Beamforming (formation de faisceaux par le calcul).*



FIGURE 1.16 – Vue d’ensemble de la station expérimentale RIAS
Photo ONERA. Crédits : [Darricau, 1994].

1.2.2 La fonction d’ambiguïté MIMO

Ce paragraphe modélise sous forme d’équations deux des caractéristiques du radar MIMO cohérent : l’expression du signal émis selon la direction angulaire, et la fonction d’ambiguïté MIMO. Quelques précisions sont données par exemple dans [Chen *et al.*, 2008, Chen, 2009, San Antonio *et al.*, 2007].

1.2.2.1 Mise en place et notations

Commençons par une présentation simplifiée du radar MIMO cohérent. Son principe est illustré à la Figure 1.14. Comme mentionné précédemment, un radar MIMO cohérent est composé de $N_T > 1$ antennes d’émission et de $N_R > 1$ antennes de réception, *suffisamment* proches les unes des autres. Chaque émetteur possède sa propre forme d’onde — ce qui signifie qu’elle peut être distincte de celle de ses voisins — et est représentée par une couleur différente sur la Figure 1.14. Le terme « émission colorée » est ainsi parfois utilisé pour parler du radar MIMO cohérent. Soit s^m le signal élémentaire émis par la m -ème antenne.

Remarque 1.6. Lorsque $N_T = N_R = 1$, on se retrouve avec un *Single-Input Single-Output* radar, le radar classique en somme.

Chaque cible reçoit une somme cohérente de ces signaux élémentaires. Cependant, selon la position angulaire (azimut φ_c , élévation θ_c) de cette cible par rapport au radar, ces signaux élémentaires peuvent être légèrement décalés en phase lors de la propagation de l’onde. De ce fait, le signal reçu par des cibles à des positions distinctes est potentiellement différent. En d’autres termes, le radar MIMO cohérent est un radar qui émet simultanément des signaux différents dans les différentes directions de l’espace (Figure 1.15). D’ailleurs, le signal émis par le radar dans la direction (φ_c, θ_c) , noté $s(\varphi_c, \theta_c)$, s’écrit, sous l’hypothèse bande étroite :

$$s(t, \varphi_c, \theta_c) = \sum_{m=1}^{N_T} e^{j\mathbf{x}_{T,m}^T \mathbf{k}(\varphi_c, \theta_c)} s^m(t), \quad (1.6)$$

où :

- $\mathbf{x}_{T,m} \in \mathbb{R}^3$ est la position de la m -ème antenne d’émission ;
- λ est la longueur d’onde du signal ;
- $\mathbf{k}(\varphi_c, \theta_c)$ est le vecteur d’onde dans la direction (φ_c, θ_c) , défini par :

$$\mathbf{k}(\varphi_c, \theta_c) := \frac{2\pi}{\lambda} [\cos(\varphi_c) \sin(\theta_c), \sin(\varphi_c) \sin(\theta_c), \cos(\theta_c)]^T; \quad (1.7)$$

- s^m est le signal élémentaire émis par la m -ème antenne.

La position des éléments à la réception introduit également un changement de phase. Ainsi, en notant $\mathbf{x}_{R,n}$ sa position, la n -ième antenne de réception reçoit le signal :

$$s_r^n(t, \tau_c, \varphi_c, \theta_c) := e^{j\mathbf{x}_{R,n}^T \mathbf{k}(\varphi_c, \theta_c)} s(t - \tau_c, \varphi_c, \theta_c), \quad (1.8)$$

où τ_c représente le retard induit par la position de la cible.

Remarque 1.7. (*Il manque comme quelque chose...*)

1. Pour les mêmes raisons que la section précédente, nous ignorons également les termes de bruit. Nous souhaiterons en effet caractériser les performances des formes d'onde pour le radar MIMO ici ; cela sera plus simple sans ajout d'aléatoire...
2. En outre, le signal reçu est généralement perturbé par un facteur égal à $e^{2\pi\nu_c t}$, où ν_c représente la fréquence Doppler de la cible, inhérente de sa vitesse. Dans la suite de ce manuscrit, et par hypothèse, nous ne considérerons pas ici ce facteur, car l'étude qui suit — dans les prochains chapitres — sera suffisamment difficile à résoudre sans ce paramètre... Citons néanmoins [Zhou et al., 2016], qui étudie la sensibilité des formes d'onde envers le Doppler.

1.2.2.2 Le traitement classique

Après avoir dévoilé l'expression d'un signal provenant d'un radar MIMO cohérent, on construit ci-dessous la fonction d'ambiguïté MIMO, en deux étapes.

Filtrage adapté Comme nous l'avons vu dans le paragraphe 1.1.3.1, le traitement radar usuel — la compression d'impulsion — consiste en une convolution dans le domaine temporel entre le signal reçu et le signal émis, ce dernier jouant le rôle de filtre. Pour un radar MIMO cohérent, le signal reçu est déterminé par les directions azimut-élévation (φ_c, θ_c) . Il convient que ce soit également le cas pour le filtre : des hypothèses angulaires (φ, θ) sont donc à poser. Ainsi, la sortie après filtrage adapté au niveau de la n -ème antenne de réception s'écrit :

$$\begin{aligned} y^n(\tau, \varphi, \theta, \tau_c, \varphi_c, \theta_c) &= \int_{-\infty}^{+\infty} s_r^n(t, \tau_c, \varphi_c, \theta_c) s^*(t - \tau, \varphi, \theta) dt \\ &= e^{j\mathbf{x}_{R,n}^T \mathbf{k}(\varphi_c, \theta_c)} \int_{-\infty}^{+\infty} s(t - \tau_c, \varphi_c, \theta_c) s^*(t - \tau, \varphi, \theta) dt \quad (1.9) \\ &= e^{j\mathbf{x}_{R,n}^T \mathbf{k}(\varphi_c, \theta_c)} \int_{-\infty}^{+\infty} s(t, \varphi_c, \theta_c) s^*(t - \tau, \varphi, \theta) dt. \end{aligned}$$

La dernière égalité est obtenue par des changements de variables sur les variables temporelles, à savoir $t \mapsto t - \tau_c$ et $\tau \mapsto \tau - \tau_c$ (et avec, dans le même temps, un bel abus de notations). Remarquez ainsi que la dépendance envers le paramètre τ_c devient sous-entendue.

Afin de prendre en compte l'ensemble des contributions des récepteurs, il est nécessaire d'effectuer ensuite un recalage, sous la même hypothèse angulaire. Cela nous donne :

$$\begin{aligned} \chi_2(\tau, \varphi, \theta, \varphi_c, \theta_c) &= \sum_{n=1}^{N_R} e^{-j\mathbf{x}_{R,n}^T \mathbf{k}(\varphi, \theta)} y^n(\tau, \varphi, \theta, \tau_c, \varphi_c, \theta_c) \\ &= \left(\sum_{n=1}^{N_R} e^{j(\mathbf{x}_{R,n}^T (\mathbf{k}(\varphi_c, \theta_c) - \mathbf{k}(\varphi, \theta)))} \right) \int_{-\infty}^{+\infty} s(t, \varphi_c, \theta_c) s(t - \tau, \varphi, \theta)^* dt \end{aligned} \quad (1.10)$$

La fonction χ_2 ainsi obtenue définit la fonction d'ambiguïté MIMO cohérent.

Elle dépend de cinq variables, voire six si on y ajoute le Doppler. Remarquez la présence de deux couples d'hypothèses angulaires azimut-élévation : le premier couple, (φ, θ) , porte sur le choix du filtre adapté ; le second sur l'origine du signal reçu, c'est-à-dire sur la position de la cible. Nous nous permettons d'insister sur le fait que ces couples ne sont pas nécessairement identiques, sauf à la bonne hypothèse... Par ailleurs, quelques propriétés de cette fonction d'ambiguïté MIMO sont proposées dans la littérature précédemment citée [Chen *et al.*, 2008, Chen, 2009, San Antonio *et al.*, 2007].

Décomposition du traitement Revenons à l'écriture (1.10), et développons-la à l'aide de l'expression du signal émis (1.6) :

$$\chi_2(\tau, \varphi, \theta, \varphi_c, \theta_c) = \left(\sum_{n=1}^{N_R} e^{j(\mathbf{x}_{R,n}^T \mathbf{k}(\varphi_c, \theta_c) - \mathbf{k}(\varphi, \theta))} \right) \times \left(\sum_{m, m'=1}^{N_T} e^{j(\mathbf{x}_{T,m}^T \mathbf{k}(\varphi_c, \theta_c) - \mathbf{x}_{T,m'}^T \mathbf{k}(\varphi, \theta))} \int_{-\infty}^{+\infty} s^m(t) (s^{m'}(t - \tau))^* dt \right). \quad (1.11)$$

Le résultat est plutôt sympathique car il permet de dévoiler les différentes composantes du traitement d'un signal MIMO.

1. Le premier terme correspond à un beamforming (Formation de faisceaux par le calcul en français, ou FFC) à la réception, dans la direction angulaire (φ, θ) . Remarquons que ce terme se factorise complètement : il est donc assez courant de considérer une fonction d'ambiguïté MIMO χ sans celui-ci :

$$\chi(\tau, \varphi, \theta, \varphi_c, \theta_c) = \sum_{m, m'=1}^{N_T} e^{j(\mathbf{x}_{T,m}^T \mathbf{k}(\varphi_c, \theta_c) - \mathbf{x}_{T,m'}^T \mathbf{k}(\varphi, \theta))} \int_{-\infty}^{+\infty} s^m(t) (s^{m'}(t - \tau))^* dt \dots$$

... ce qui ne signifie pas pour autant qu'il faudra l'ignorer par la suite. Notons que cette FFC à la réception est identique à celle d'un réseau phasé classique.

2. Le deuxième terme correspond à un traitement à l'émission (une sorte de FFC émission) pour la même hypothèse angulaire (φ, θ) . Il est cette fois absent du réseau phasé [Deng *et al.*, 2016]. Il permet de focaliser les phases des différents signaux émis et donc de récupérer la directivité angulaire à l'émission.
3. Le dernier terme évoque une compression d'impulsion entre deux signaux élémentaires $(s^m, s^{m'})$. Comme les valeurs de ces corrélations ne sont pas nécessairement identiques, cette étape ne peut pas être factorisée.

De ces observations, nous pouvons en déduire l'une des manières d'effectuer le traitement optimal d'un signal MIMO cohérent : un *beamforming* à la réception, suivi d'une application d'un ensemble de filtres adaptés, associés à des directions angulaires (φ, θ) . Ces deux étapes sont illustrées à la Figure 1.17. Finalement, la pluralité des signaux élémentaires dans un signal reçu et la multiplicité des informations apportées qu'elle entraîne ne sont exploitées qu'indirectement. En effet, nous n'allons pas nécessairement chercher à décomposer le signal reçu en ses composantes élémentaires, comme illustré à la Figure 1.18, puisque, on le répète, le traitement à l'émission ne peut pas toujours se séparer de l'étape du filtrage élémentaire. Cette procédure, naturelle à s'y méprendre, se rencontre parfois dans la littérature, comme ici [Guyvarch *et al.*, 2012]. Elle n'est optimale que dans certains cas particuliers (orthogonalité des signaux par exemple, ce qui est le cas dans l'article cité), que nous développerons dans le Chapitre 5.

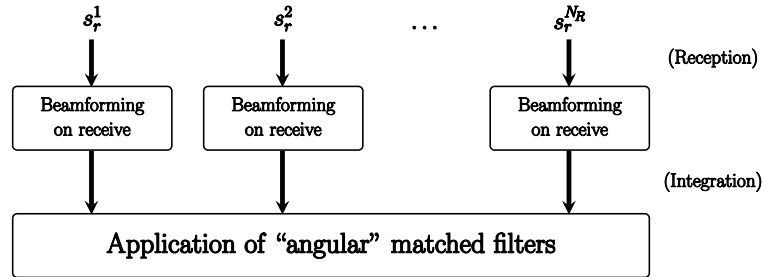


FIGURE 1.17 – Traitement optimal d'un signal à émission colorée
Le traitement à la réception peut se faire indifféremment en début ou en fin de traitement, au niveau de chaque récepteur.

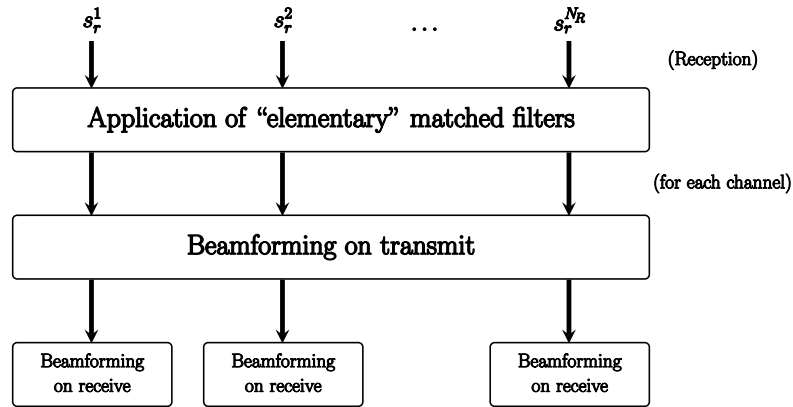


FIGURE 1.18 – Traitement supposé d'un signal à émission colorée
Ce traitement n'est optimal que dans certains cas particuliers.

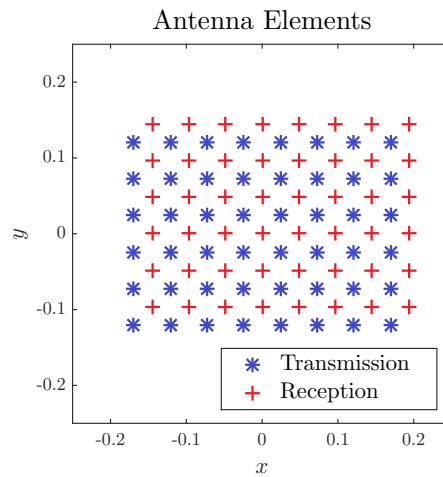


FIGURE 1.19 – Position relative des antennes
Antenne composée de 6×8 émetteurs, et autant de récepteurs.

1.2.2.3 Illustration des différents traitements

La fonction d'ambiguïté MIMO ajoute un terme supplémentaire par rapport à celle d'un réseau phasé classique. Comme expliqué précédemment, il correspond à un traitement à l'émission, appelé *beamforming*. Le présent paragraphe va simplement illustrer les effets de ces *beamformings* (puisque'il y en a un autre, celui à la réception), par un exemple jouet.

Pour information, l'anglicisme *beamforming*, que l'on traduit par « Formation de faisceaux par le calcul » (FFC), évoque également le balayage électronique. A la Figure 1.13, on a vu qu'il était nécessaire pour une antenne focalisée de déplacer son faisceau, dans le but de couvrir l'ensemble de l'espace de surveillance. Cela peut se faire mécaniquement, mais aussi électroniquement à l'aide de déphaseurs [Haupt, 2010]. Le *beamforming* désigne donc la modification des phases appliquée sur chaque antenne d'émission pour changer l'orientation du diagramme d'émission.

Du côté traitement, le *beamforming* indique la mise en évidence d'un signal recherché. Lorsque les récepteurs « écoutent », ils perçoivent des signaux provenant de toutes parts. Il est donc nécessaire d'effectuer quelques opérations numériques afin de se concentrer sur une direction à la fois. Et cela se voit bien sur l'expression (1.10) : si on veut se focaliser sur la direction (φ_c, θ_c) , il suffit de poser $(\varphi, \theta) = (\varphi_c, \theta_c)$, de sorte que les intégrales soient pondérées au maximum.

Une image vaut mille mots, alors considérons l'exemple suivant. Choisissons une antenne composée de $N_T = 48$ émetteurs, et d'autant de récepteurs. Leurs positions relatives sont représentées à la Figure 1.19. Chaque émetteur a un diagramme de rayonnement isotrope ; la distance minimale séparant deux éléments proches est de $\lambda/2$. Sur les Figures 1.20 et 1.21, on représente la corrélation (à retard nul) entre le signal émis dans la direction $(\varphi, \theta) = (0, 0)$ et les autres directions azimut-élévation, en appliquant au fur à mesure différents traitements.

- Figure 1.20 : traitement à la réception seulement. On retrouve le fait que la corrélation est maximale dans la direction $(0, 0)$. Logique ! Cependant, elle reste assez élevée dans certaines directions (« lobes secondaires » à -12.4 dB).
- Figure 1.21 : ajout du traitement à l'émission. Il facilite la focalisation autour de la direction d'intérêt, en pondérant les corrélations alentours. On observe en outre une légère réduction de la largeur du lobe principal.

Tel est le comportement du radar MIMO cohérent. Malgré une émission potentiellement large (*cf.* le paragraphe suivant) et une diversité à l'émission, la combinaison des deux formations de faisceaux (émission et réception) permet de récupérer convenablement le signal émanant d'une certaine direction.

1.2.3 Beampattern

Comme l'indique l'équation (1.6), le signal émis par un radar MIMO cohérent diffère selon les directions, ce qui sous-entend que l'antenne n'est pas nécessairement isotrope (l'énergie rayonnée dans chacune des directions peut également être fluctuante). Pour être plus précis, il a été montré dans [Fuhrmann *et al.*, 2008] que la densité de puissance normalisée P est une fonction de la direction angulaire θ_c (et de φ_c dans le cas général), et est définie par :

$$P(\theta_c) = \frac{1}{4\pi} \mathbf{a}_t^H(\theta_c) \mathbf{R}(0) \mathbf{a}_t(\theta_c). \quad (1.12)$$

L'expression précédente suggère que le *beampattern* (la forme du faisceau) est lié à l'inter-corrélation des signaux élémentaires au retard nul, *via* la matrice de covariance, notée ici $\mathbf{R}(0)$. Présentons ci-dessous deux cas.

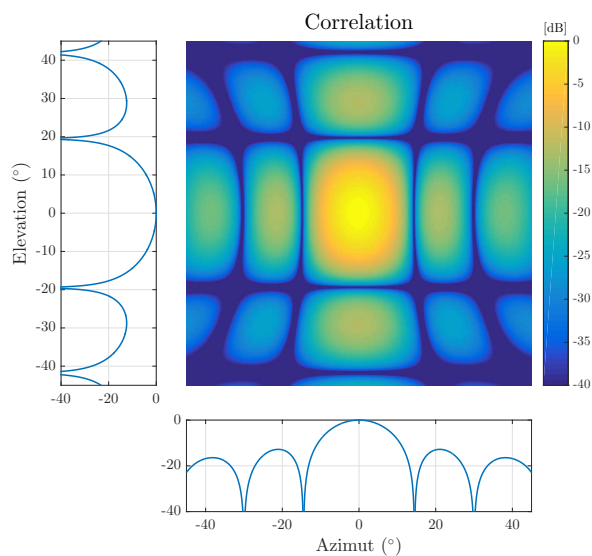


FIGURE 1.20 – Corrélation avec traitement à la réception
 Corrélation à retard nul par rapport à la séquence émise en $(\varphi, \theta) = (0, 0)$.
 Notez les fortes corrélations sur la coupe $\varphi = 0$ (la coupe de gauche).

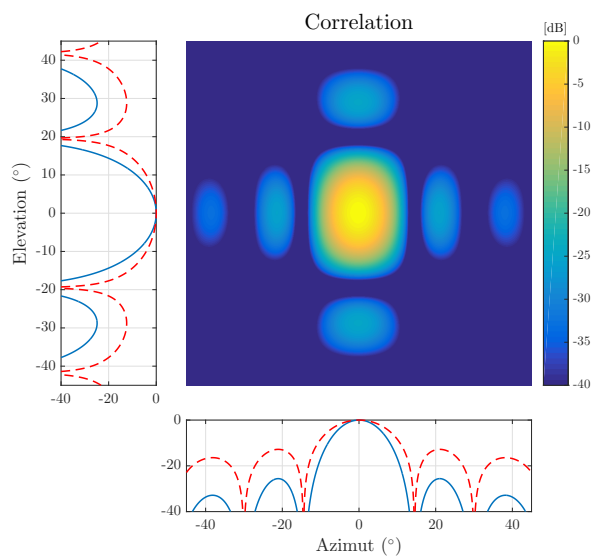


FIGURE 1.21 – Corrélation avec traitement à l'émission et à la réception
 Observez la réduction des « lobes secondaires », et la largeur plus fine du « lobe principal »,
 par rapport à la Figure 1.20.
 Les courbes de cette dernière sont en pointillés.

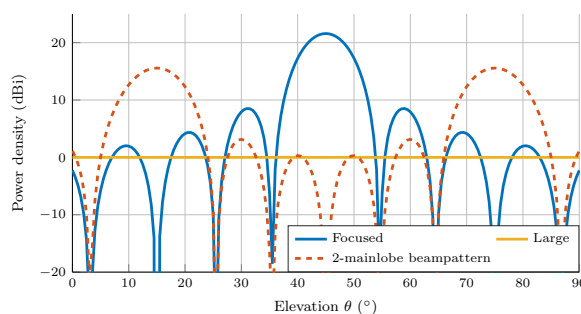


FIGURE 1.22 – Exemples de faisceaux pour un radar MIMO composé de $N_T = 12$ antennes. Le faisceau avec deux « lobes principaux » a été obtenu à l'aide de [Lipor et al., 2014].

- Si les signaux $\{\mathbf{s}^m\}$ sont identiques, alors $\mathbf{R}(0) = \mathbf{1}_{N_T, N_T}$. Un radar MIMO cohérent se réduit à un réseau phasé classique, et :

$$\begin{aligned}
 P(\theta_c) &= \frac{1}{4\pi} \mathbf{a}_t^H(\theta_c) \mathbf{1}_{N_T} \mathbf{a}_t(\theta_c) \\
 &= \frac{1}{4\pi} \left| \sum_{n=1}^{N_T} e^{-j\mathbf{x}_{T,n}^T \mathbf{k}(\theta_c)} \right|^2 \\
 &\sim \frac{1}{4\pi} \text{sinc}^2(N_T \theta_c).
 \end{aligned} \tag{1.13}$$

On retrouve le diagramme en « sinus cardinal », qui est potentiellement (et relativement) fin et focalisé.

- À l'opposé, des signaux parfaitement orthogonaux donnent $\mathbf{R} = \mathbf{I}_{N_T}$. On a cette fois :

$$\begin{aligned}
 P(\theta_c) &= \frac{1}{4\pi} \mathbf{a}_t^H(\theta_c) \mathbf{I}_{N_T} \mathbf{a}_t(\theta_c) \\
 &= \frac{1}{4\pi} \|\mathbf{a}_t^H(\theta_c)\|^2 \\
 &= \frac{1}{4\pi} N_T.
 \end{aligned} \tag{1.14}$$

Ce résultat est similaire pour tout θ_c , et se traduit par un faisceau plus large, omnidirectionnel disons.

Ces deux possibilités sont illustrées à la Figure 1.22, et montrent les avantages d'un radar MIMO cohérent à ce sujet. Contrairement à un réseau linéaire classique où, on le répète, le faisceau sera nécessairement focalisé — peu importe la forme d'onde —, la forme du faisceau d'un radar MIMO cohérent peut être adaptée selon les besoins opérationnels : d'un faisceau fin pour du pistage par exemple [Bar-Shalom et al., 2004, Pilté et al., 2017], à un autre, plus large, pour de la veille ou des applications de surveillance. La forme du faisceau étant régie par la matrice de corrélation $\mathbf{R}(0)$, il est en outre possible de modéliser des *beampatterns* beaucoup plus exotiques. En effet, à l'aide d'un problème d'optimisation, les auteurs de [Cheng et al., 2017, Guo et al., 2015, Lipor et al., 2014, Wang et al., 2012] ont proposé des faisceaux avec plusieurs « lobes principaux », comme dépeint en cette même figure.

1.2.4 Différentes formes d'onde

Pour l'instant, nous n'avons encore émis aucune hypothèse sur l'ensemble des signaux élémentaires $\{s^m\}$. Le paragraphe 1.1.3.2 a mentionné l'importance de la forme d'onde sur la fonction d'ambiguïté, et donc *a fortiori* sur les performances du système. Mais son choix est également déterminé par les besoins opérationnels (liés par exemple à l'environnement ou à la cible) et les contraintes matérielles (largeur de bande, etc.) [Blunt et al., 2016].

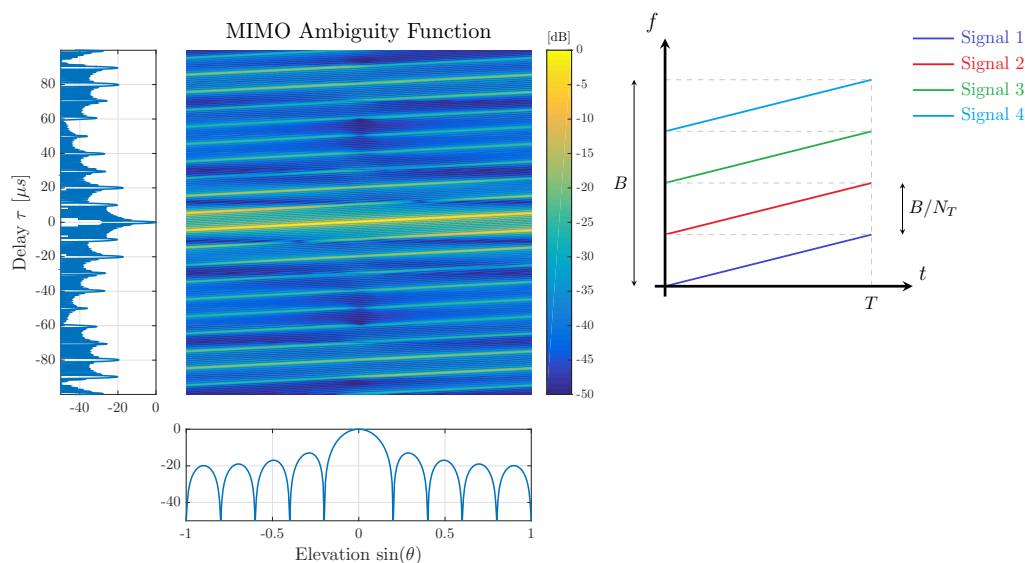


FIGURE 1.23 – Fonction d’ambiguïté pour un codage FDMA

A gauche : représentation de la fonction d’ambiguïté pour un codage FDMA.

Antenne composée de 10 émetteurs, sans traitement à la réception.

Chacun émet un chirp dans une bande de fréquence dédiée, comme schématisé à droite.

Cela dit, les bandes ne sont pas nécessairement disjointes pour ce codage.

Dans cette thèse, nous avons décidé de nous concentrer sur le codage *intrapulse*. Cette notion insinue d’une part l’utilisation de radars émettant dans un régime pulsé (par opposition à un régime continu). D’autre part, elle indique que le signal émis par une antenne est identique d’impulsion à impulsion. Il existe de nombreuses formes d’onde pour (ou, du moins, applicable pour) le radar MIMO cohérent. Un classement a été proposé dans [Blunt *et al.*, 2016, Guyvarch *et al.*, 2012, Rabaste *et al.*, 2013, Sun *et al.*, 2014, Xue *et al.*, 2012]. Permettons-nous de citer quelques-unes d’entre elles.

FDMA *Frequency Division Multiple Access*. Chaque antenne émet à une fréquence propre, différente de celle des autres. Avec un écart fréquentiel suffisant — par exemple égal à la bande [Rabaste *et al.*, 2013] — les séquences élémentaires sont orthogonales entre elles. Mais cela ne signifie en aucun cas que leur autocorrélation est parfaite. La Figure 1.23 représente une fonction d’ambiguïté distance/élévation pour le codage CDMA. Ici, chaque signal élémentaire est une modulation linéaire de fréquence (*chirp*) partageant la même largeur de bande (autrement dit, la bande totale a été subdivisée de façon égale). En outre, ces signaux sont décalés en fréquence de la largeur de bande élémentaire. On y devine un couplage important dans l’espace distance/angle, symbolisé par une diagonale forte. Cela signifie qu’il sera difficile d’estimer de manière précise ces deux paramètres...

Remarque 1.8. (*RIAS*) *Le premier radar MIMO, le RIAS, utilisait un codage FDMA.*

TDMA *Time Division Multiple Access*. Un même signal de base est émis alternativement sur chaque antenne mais à des instants différents. Cela entraîne une orthogonalité des signaux au retard nul. Lorsque ce signal est en plus une modulation linéaire de fréquence, mais jouissant d’une bande différente pour chaque émetteur, il y a également une orthogonalité entre les séquences élémentaires. La fonction d’ambiguïté induite ressemble à la Figure 1.24, où les valeurs autour de la coupe $\theta = 0$ sont très proéminentes : la composante retard sera ainsi très difficile à déterminer, car très ambiguë... Enfin, notons que, dans le cas du codage TDMA, le signal élémentaire n’est pas nécessairement fréquentiel.

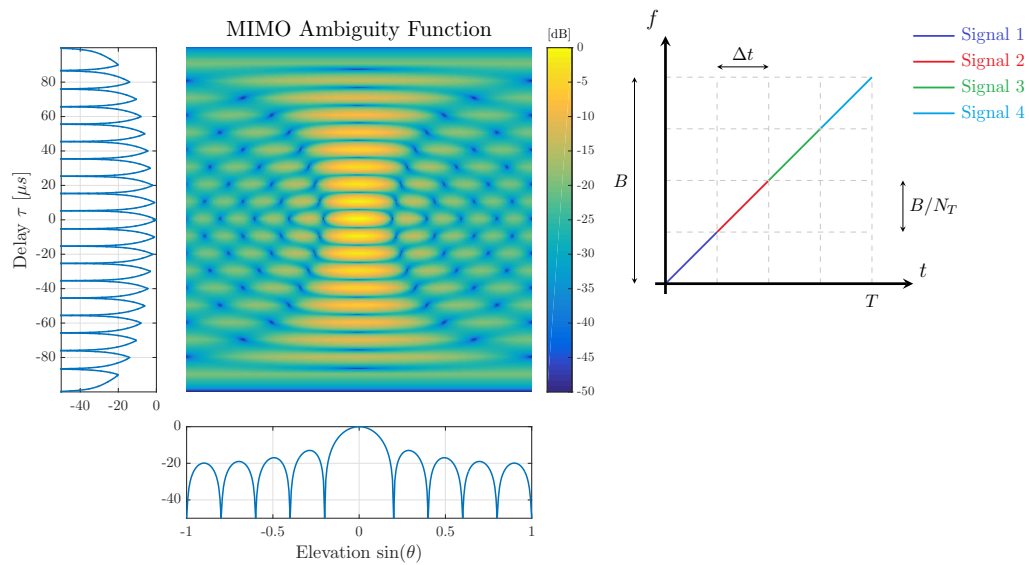


FIGURE 1.24 – Fonction d’ambiguïté pour un codage TDMA

A gauche : représentation de la fonction d’ambiguïté pour un codage TDMA.

Antenne composée de 10 émetteurs, sans traitement à la réception.

Chacun émet un chirp, mais à des temps différents, comme schématisé à droite.

Il est possible d’employer d’autres codes élémentaires, à l’instar des codes de phase.

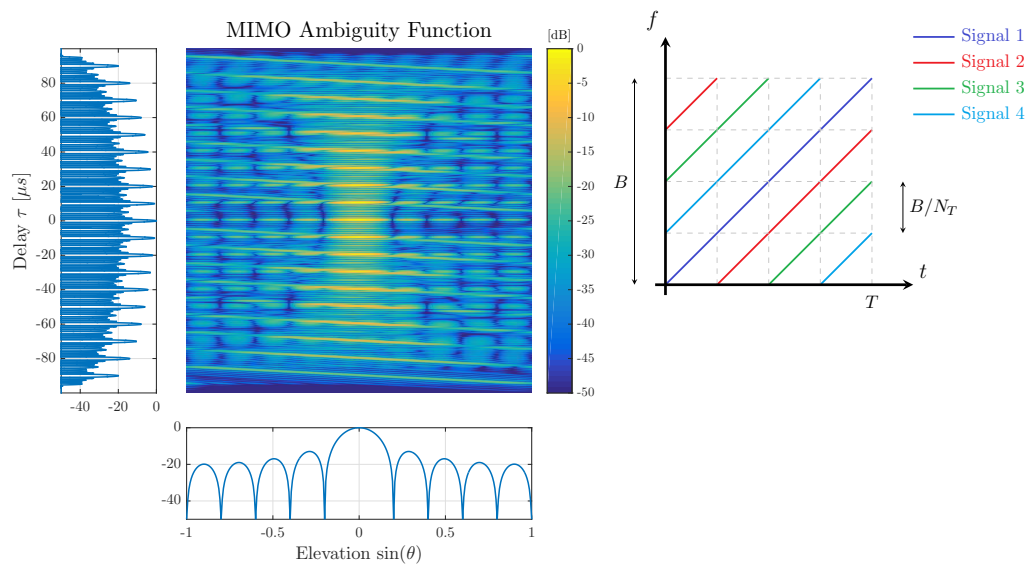
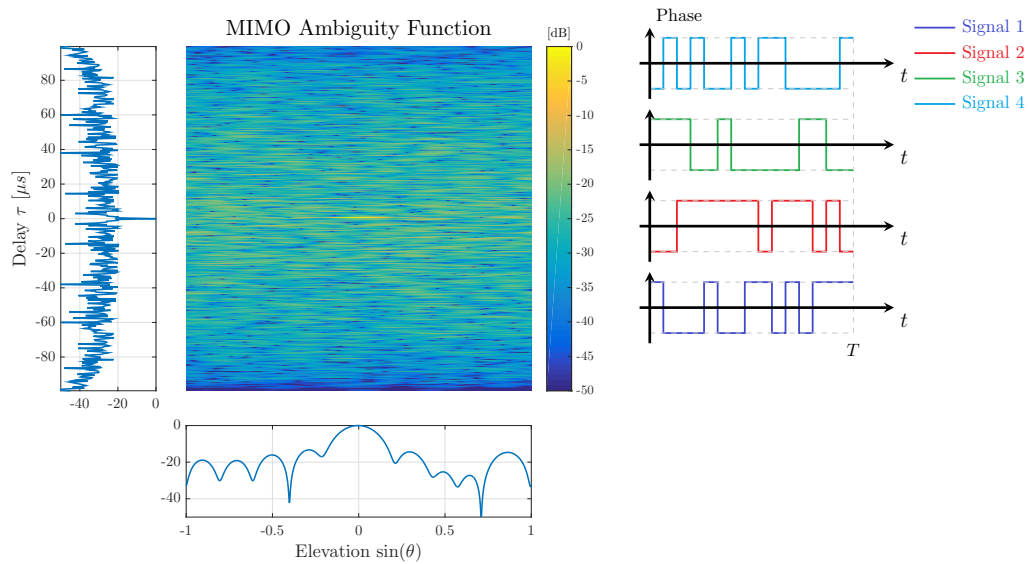


FIGURE 1.25 – Fonction d’ambiguïté pour un codage fréquentiel hybride

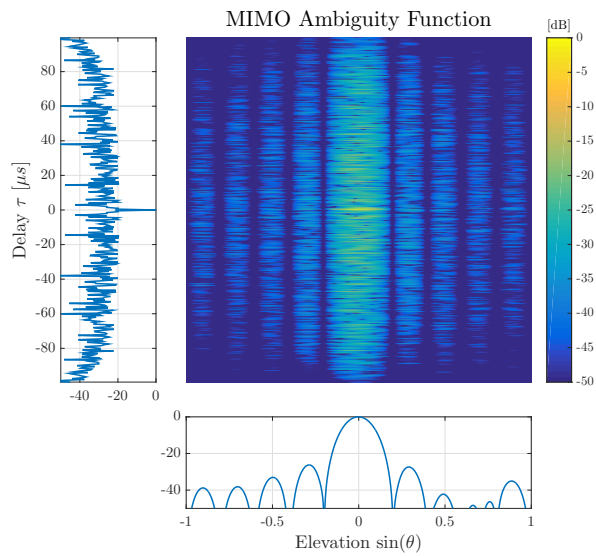
A gauche : représentation de la fonction d’ambiguïté pour un codage fréquentiel hybride.

Antenne composée de 10 émetteurs, sans traitement à la réception.

Chacun émet un chirp exploitant la bande totale, mais de manière décalée, comme schématisé à droite.



A gauche : représentation de la fonction d'ambiguïté pour un codage CDMA. Antenne composée de 10 émetteurs, sans traitement à la réception. Chacun émet son propre code de phase, tiré ici aléatoirement, comme schématisé à droite.



La même figure que ci-dessus, mais en ajoutant le traitement à la réception. Notez la différence.

FIGURE 1.26 – Fonction d'ambiguïté pour un codage CDMA aléatoire (1)

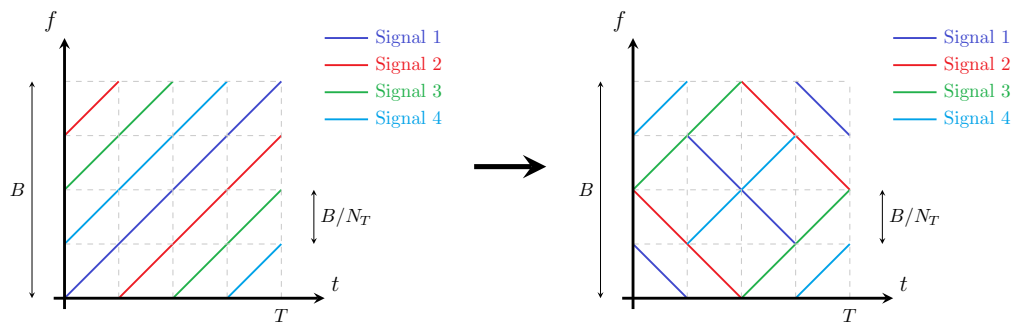


FIGURE 1.27 – Illustration d'une optimisation d'un codage fréquentiel

Hybrides À partir de ces deux codages, il est possible d'en construire des versions « hybrides », comme le *frequency-hopping* [Chen *et al.*, 2008, Han *et al.*, 2016], issu des télécommunications, ou les codes circulants [Babur *et al.*, 2016]. Ainsi, de nombreux travaux portent sur le choix du découpage fréquentiel ou temporel [Babur *et al.*, 2016, Chang *et al.*, 2018, Chen *et al.*, 2008, Gao *et al.*, 2016, Sammartino *et al.*, 2013], comme représenté Figure 1.27. D'ailleurs, un exemple d'une fonction d'ambiguïté est donné en Figure 1.25. On retrouve un comportement relativement similaire au précédent.

CDMA *Code Division Multiple Access*. Chaque antenne possède son propre code de phase (un signal ne jouant seulement sur la modulation de la phase). À ce sujet, la littérature y est assez riche, puisqu'elle propose...

- ... de nombreuses familles de séquences pseudo-orthogonales, définies à l'origine dans le domaine des télécommunications. Une liste est proposée ici [Arlery *et al.*, 2015].
- ... de nombreuses méthodologies pour générer des codes polyphases quasi-orthogonaux [Deng, 2004, Deng *et al.*, 2016, Esmaeili-Najafabadi *et al.*, 2017, Gold, 1967] [He *et al.*, 2012, He *et al.*, 2009, Jardak *et al.*, 2014, Kasami, 1966]. Le Chapitre 3 en fera d'ailleurs de même.

Une fonction d'ambiguïté pour des séquences polyphases tirées de façon aléatoire est donnée Figure 1.26. Celle-ci est en comparaison quelque peu différente, dans la mesure où le couplage est clairement moins prononcé : le lobe principal est d'ailleurs plutôt discret. Néanmoins, on note des lobes secondaires assez prononcés dans un vaste domaine distance/angle (par rapport à la Figure 1.23 par exemple).

Comme précisé dans [Sun *et al.*, 2014], chaque catégorie présente ainsi diverses attitudes envers la propriété d'orthogonalité. Or, plusieurs études (citons [Rabaste *et al.*, 2013] en plus de la référence précédente) ont établi que l'ensemble de ces formes d'onde souffrent d'un couplage distance/élévation plus ou moins prononcé. Cela se traduit par une augmentation de la variance de l'estimée de la position de la cible dans l'espace distance/angle, c'est-à-dire, en d'autres termes, une mesure moins précise. Reprenons par exemple la Figure 1.23. Il est assez clair que la fonction représentée ne ressemble en rien à une punaise. Elle dévoile à la place des stries diagonales assez prononcées, signe de ce couplage. En fait, il se trouve que les codes CDMA proposent *généralement* le couplage le moins prononcé (parmi les codes *intrapulses*), et une fonction d'ambiguïté MIMO approximativement découplée. Mais avec des lobes secondaires plus élevés en contrepartie. Ceci est l'une des raisons pour laquelle nous avons décidé de nous intéresser aux codes de phase dans cette thèse...

Remarque 1.9. (*Représentation de la fonction d'ambiguïté*) Dans le présent manuscrit, on se contentera d'une représentation de la fonction d'ambiguïté sous son profil distance/élévation. En toute rigueur, il nous aurait fallu présenter beaucoup plus de coupes, dans la mesure où celle-ci dépend de six paramètres.

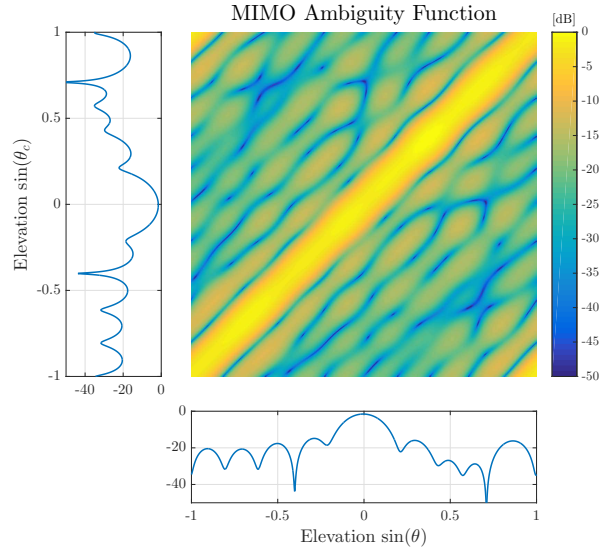


FIGURE 1.28 – Fonction d’ambiguïté pour un codage CDMA aléatoire (2)
*Cette fois-ci, la représentation est en angle de réception/angle de la cible (θ, θ_c) .
 Remarquez que la diagonale où $\theta = \theta_c$ présente heureusement la valeur maximale.*

Mais, par exemple, les images élévation/élévation se ressemblent beaucoup, qualitativement parlant, avec une diagonale « forte » (du fait d’une hypothèse angulaire correcte, i.e. $\theta_c = \theta$, cf. Figure 1.28).

1.3 Objectifs de la thèse

Ce manuscrit s’intitule « Méthodologies de conception de formes d’onde pour radars sols. Application au cas du radar MIMO ». Il s’intéresse donc à la génération de signaux, adaptés *spécifiquement* pour ledit radar, dans sa version co-localisée. Autrement dit, il s’emploie à répondre à la question suivante : « Comment générer des formes d’onde intéressantes et *spécifiques* au radar MIMO cohérent ? »

Mais d’où provient cette spécificité ? Un radar MIMO, co-localisé ou non, est composé de plusieurs émetteurs, capables (à notre avantage) d’émettre des signaux qui leur sont propres. Il nous est donc demandé de pouvoir générer de multiples formes d’onde, contrairement à un radar classique, qui n’en utilise qu’une seule. Alors qu’il existe de nombreuses familles de séquences (répertoriées à la Section 1.2.4), nous avons décidé dans cette thèse de nous concentrer uniquement sur le codage CDMA, notamment pour des raisons de couplage. Néanmoins, il semblerait que les codes de phase présentent des fonctions d’ambiguïté distance/élévation avec de forts lobes secondaires. Une optique de performances nous amène donc à la problématique suivante, plus représentative de ce manuscrit : « Comment déterminer les codes de phase les plus performants pour le radar MIMO cohérent ? »

L’objectif formulé, il ne nous reste plus qu’à soumettre une réponse sur le « Comment ? ». Dans le but d’obtenir des signaux polyphasés présentant des lobes secondaires acceptables, une procédure naturelle consiste à :

1. Établir une mesure quantifiant ces lobes. Elle offrira ainsi la possibilité de classer des signaux selon leurs performances.
2. Rechercher les meilleurs signaux sous cette mesure.

Mathématiquement, cela s'apparente à un problème d'optimisation [Floudas *et al.*, 2008], de minimisation plus précisément. Pour y faire face, de nombreuses méthodologies ont été suggérées dans la littérature, dont nous devons comparer les résultats proposés et la vitesse d'exécution (pour une potentielle utilisation en temps réel) sur notre affaire. Révétons d'emblée que la mesure proposée portera sur un critère d'orthogonalité, c'est-à-dire sur le niveau des lobes secondaires. Ainsi, dans cette thèse, on considère que le « Comment ? » se lit « Quelle est la méthode d'optimisation la plus adaptée ? » et que, finalement, les deux points précédents définissaient les deux objectifs principaux, précisés ci-dessous :

1. Modéliser une fonction de coût chiffrant, pour un ensemble de séquences, les lobes secondaires de la fonction d'ambiguïté MIMO cohérente. Et pas seulement...
2. Évaluer différentes méthodes d'optimisation sur le problème de minimisation induit.

Par une approche incrémentale, les différents chapitres — présentés dans l'introduction — essaieront d'apporter des éléments de réponse.

Chapitre 2

De l'optimisation de la corrélation

Le traitement radar classique s'effectue en recherchant dans le signal reçu la présence du signal émis. Ce procédé est appelé « filtrage adapté » et consiste, mathématiquement parlant, à évaluer la corrélation entre ces deux signaux. Dans un monde parfait (sans bruit, sans pertes...), une correspondance revient à étudier la fonction d'autocorrélation induite, associée au signal émis. Le niveau des lobes secondaires de cette dernière permet, entre autres, d'évaluer les performances de détection du système.

Ce chapitre s'intéresse à cette fonction de corrélation. Plus précisément, on se propose ici de rechercher des signaux présentant de faibles lobes secondaires. Pour cela, dans un premier temps, un problème d'optimisation est modélisé, dont le critère à minimiser est ce niveau même de lobes. On s'intéressera en particulier aux signaux polyphasés (ce choix a été expliqué dans le chapitre précédent).

Le problème ainsi défini est malheureusement non convexe, ce qui signifie que sa résolution n'est pas une chose aisée. On suggère donc dans un second temps de tester différentes méthodes d'optimisation : une méthode évolutionnaire (CMA-ES) et une autre, déterministe (une descente de gradient). Quelques simulations permettront de comparer leur efficacité.

La modélisation sous forme d'un problème d'optimisation est relativement flexible : des contraintes — définissant certains besoins opérationnels, par exemple — peuvent aisément être adjointes ; des extensions sont également possibles. Cela sera le sujet du chapitre suivant.

2.1 La fonction de corrélation

2.1.1 Définition d'un code de phase

Avant toute chose, définissons de manière analytique un code de phase.

Soit $s \in L^2(\mathbb{C})$ un signal de durée T . On considère une subdivision de ce signal en N moments de durée identique $T_e := T/N$. Si une phase $\varphi_n \in [0, 2\pi[$ est attribuée à chacun de ces moments, avec $n \in \llbracket 1, N \rrbracket$, l'enveloppe complexe de ce signal s'écrit [Levanon *et al.*, 2004] :

$$s(t) = \frac{1}{\sqrt{T}} \sum_{n=1}^N s_n \operatorname{rect} \left(\frac{t - (n-1)T_e}{T_e} \right), \quad (2.1)$$

où $s_n = \exp(j\varphi_n)$. La séquence $\mathbf{s} := [s_1, \dots, s_N]^T$ est appelée *code de phase* associé au signal s . Cependant, par abus de langage, « le signal s » fera aussi bien référence au signal s qu'à son code de phase \mathbf{s} . La Figure 2.1 donne un exemple de code de phase.

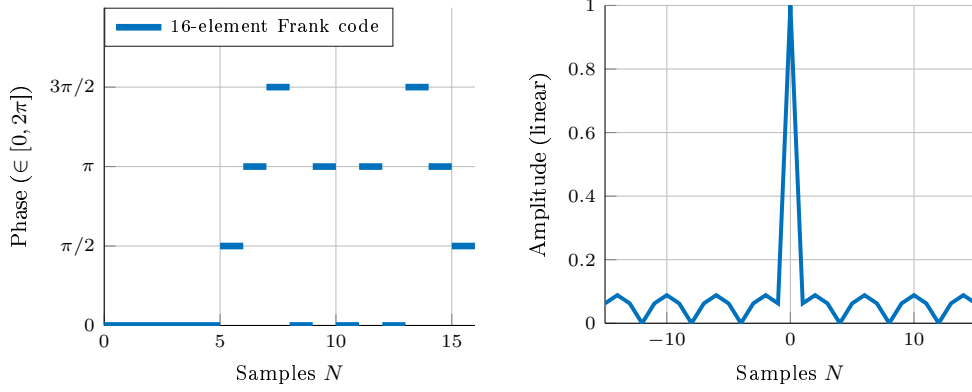


FIGURE 2.1 – Représentation d'un code de phase

Exemple d'un code de Frank à 16 éléments [Frank, 1963]. A gauche : représentation des phases. A droite : représentation de la fonction d'autocorrélation.

Remarque 2.1. (Contrainte de module constant)

1. La définition ci-dessus sous-entend qu'un code de phase n'est qu'une modulation de phase (i.e. l'amplitude reste constante). Cette hypothèse, désignée par « contrainte de module constant », a une origine physique [Wang et al., 2012] : elle implique que le radar opère dans un régime saturé afin de maximiser la puissance du signal en sortie, et ainsi d'éviter au mieux les distorsions de celui-ci. A moins d'une mention contraire, cette hypothèse sera toujours considérée dans ce manuscrit.
2. En ignorant la dimension temporelle, chaque élément de la séquence $\mathbf{s} = [s_1, \dots, s_N]^T$ sera défini, sans perte de généralité, par :

$$\forall n \in \llbracket 1, N \rrbracket, \quad s_n = \frac{1}{\sqrt{N}} \exp(j\varphi_n), \quad \varphi_n \in [0, 2\pi[, \quad (2.2)$$

de sorte que l'énergie de la séquence \mathbf{s} est constante et égale à 1. Sous ces conditions, la séquence \mathbf{s} appartient à un tore de dimension N , qu'on note dans ce manuscrit \mathbf{T}^N , un espace non convexe [do Carmo, 1992].

Remarque 2.2. (Échantillonnage) Le nombre de moments N influe généralement sur les paramètres du radar : en fait, il fixe même le produit BT , où B indique la largeur de la bande de fréquence allouée. Ce n'est donc pas un facteur de discrétisation à proprement dit (plus le nombre N est important, plus l'approximation d'un signal continu est précise).

2.1.2 Filtrage adapté et corrélation

Afin de détecter la présence d'une cible, le traitement radar de base consiste à déceler le signal émis dans celui qui a été reçu. Comme expliqué dans le chapitre précédent, cela équivaut à étudier la fonction de corrélation entre ces deux signaux. Lorsque le signal émis joue le rôle de filtre adapté, celle-ci s'écrit :

$$\left\{ \begin{array}{l} \text{ACF} : L^2(\mathbb{C}) \rightarrow L^2(\mathbb{C}), \\ s \mapsto \text{ACF}(s) := s * l^*(s), \\ \text{ACF}(s) : \mathbb{R} \rightarrow \mathbb{C}, \\ \tau \mapsto \text{ACF}(s)(\tau) := \int_{-\infty}^{+\infty} s(t)s^*(t-\tau)dt. \end{array} \right. \quad (2.3)$$

De la version continue peut être déduite une version discrète de la fonction d'autocorrélation, sans doute plus adaptée pour les codes de phase. L'image de $\mathbf{s} = [s_1, \dots, s_N]^T$ par la fonction d'autocorrélation discrète aperiodique (notée AACF, pour *Aperiodic Autocorrelation Function*) est également une séquence, mais de longueur $2N - 1$:

$$\begin{aligned} \text{AACF} : \mathbf{T}^N &\rightarrow \mathbb{C}^{2N-1}, \\ \mathbf{s} \mapsto (\text{AACF}(\mathbf{s}))_k &= (\mathbf{s} * \mathbf{s}^l)_k = \sum_{n=1}^N s_n s_{n+k-N}^*, \quad (1 \leq k \leq 2N - 1), \end{aligned} \quad (2.4)$$

où, par convention, la séquence \mathbf{s} est nulle en dehors de son support principal, *i.e.* $s_n = 0$ si $n \leq 0$ ou $n > N$. En outre, on pose $\mathbf{s}^l = [s_N, \dots, s_1]$.

Un peu de nomenclature : dans la suite du manuscrit, l'élément central, $(\text{AACF}(\mathbf{s}))_N$, sera reconnu comme le « lobe principal », tandis que les autres éléments seront des lobes secondaires. Par ailleurs, le terme *autocorrélation* peut désigner à la fois — mais de manière impropre, certes — la fonction d'autocorrélation AACF ou sa séquence en sortie $\text{AACF}(\mathbf{s})$.

Remarque 2.3. (*Expression de l'autocorrélation*)

1. La séquence d'autocorrélation $\text{AACF}(\mathbf{s})$ peut également se déterminer sous une forme matricielle [Rabaste et al., 2015]. Soit $\mathbf{\Lambda}_N(\mathbf{s})$ la matrice de convolution, de taille $(2N - 1) \times N$, telle que :

$$\mathbf{\Lambda}_N(\mathbf{s}) = \begin{bmatrix} s_N & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ s_1 & \cdots & \cdots & s_N \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & s_1 \end{bmatrix}. \quad (2.5)$$

Ceci entraîne $\text{AACF}(\mathbf{s}) = \mathbf{\Lambda}_{2N-1}(\mathbf{s})\mathbf{s}^*$.

2. La littérature emploie généralement les définitions ci-dessus pour la fonction d'autocorrélation discrète (citons par exemple [Baden et al., 2018, Welch, 1974]). Cependant, si on avait voulu étendre rigoureusement l'expression continue (2.3), il nous aurait fallu écrire :

$$(\text{AACF}(\mathbf{s}))_k = \sum_{n=1}^N s_n s_{n-k}^*, \quad (1 - N \leq k \leq N - 1). \quad (2.6)$$

Et attention aux indices ! Remarquons néanmoins que l'expression (2.6) est conjuguée de (2.4).

3. Soit $P : x \in \mathbb{R} \rightarrow \mathbb{C}$ un polynôme, tel que $P(x) = s_1x + s_2x^2 + \dots + s_Nx^N$. Les coefficients de la séquence d'autocorrélation peuvent se déduire en développant le produit $P(x)P^*(x^{-1})$. Cette astuce sera nullement utilisée dans ce manuscrit, mais si elle peut inspirer le lecteur pour la résolution du problème qui suit...

Remarque 2.4. (*Autocorrélation périodique*) Il existe également une autocorrélation dite « périodique ». Celle-ci a un sens lorsqu'on considère, par exemple, les émissions continues, en opposition aux émissions pulsées [Park et al., 2016]. Elle est ainsi souvent utilisée dans les applications télécoms.

L'hypothèse de périodicité s'écrit, pour une séquence \mathbf{s} , $s_n = s_{(n \bmod N)}$, pour tout $n \in \mathbb{N}$, ce qui induit une matrice de corrélation plus petite :

$$\mathbf{\Lambda}_N(\mathbf{s}) = \begin{bmatrix} s_N & s_1 & \cdots & s_{N-1} \\ s_{N-1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & s_1 \\ s_1 & \cdots & s_{N-1} & s_N \end{bmatrix}. \quad (2.7)$$

L'ensemble des résultats de cette thèse peut se transposer pour le cas périodique, mais cela ne sera guère développé...

2.1.3 Mesure des lobes secondaires

Le premier chapitre a évoqué l'importance de réduire les lobes secondaires de la séquence d'autocorrélation. De nombreuses grandeurs ont été définies afin de mesurer le niveau de ces lobes secondaires : citons notamment l'ISL (pour *Integrated Sidelobe Level*) ou encore le PSLR (*Peak-to-Sidelobe Level*). Le premier mesure l'énergie présente, après filtrage, dans les lobes secondaires, tandis que le second évalue le lobe secondaire le plus fort.

Soit $\mathbf{s} = [s_1, \dots, s_N]^T$ une séquence de longueur N . Pour plus de simplicité, notons $\mathbf{r} = \text{AACF}(\mathbf{s})$ le vecteur d'autocorrélation de longueur $2N - 1$; mais gardons en tête que \mathbf{r} dépend de \mathbf{s} !

L'*Integrated Sidelobe Level* est défini par :

$$\begin{aligned} \mathbf{T}^N &\rightarrow \mathbb{R}_+ \\ \mathbf{s} &\mapsto \text{ISL}(\mathbf{s}) := \mathbf{r}^H \mathbf{F} \mathbf{r}, \end{aligned} \quad (2.8)$$

où \mathbf{F} est une matrice diagonale de taille $2N - 1$, spécifiée par le vecteur $[1, \dots, 1, 0, 1, \dots, 1]$ (un vecteur composé uniquement de 1, sauf à la position N , représentant le lobe principal). Remarquons que l'ISL peut également s'écrire à l'aide d'une norme : $\text{ISL}(\mathbf{s}) = \|\mathbf{F}\mathbf{r}\|_2^2$, ce qui implique *a fortiori* la positivité de cette mesure.

Remarque 2.5. (*Merit Factor*) L'ISL est étroitement lié au « Merit Factor » MF, introduit par Golay [Golay, 1977] :

$$\text{MF}(\mathbf{s}) := \frac{|r_N|^2}{\text{ISL}(\mathbf{s})}. \quad (2.9)$$

Le *Peak-to-Sidelobe Level* est quant à lui défini par :

$$\begin{aligned} \mathbf{T}^N &\rightarrow \mathbb{R}_+ \\ \mathbf{s} &\mapsto \text{PSLR}(\mathbf{s}) := \max \left(|(F\mathbf{r})_1|, \dots, |(F\mathbf{r})_{2N-1}| \right) = \|\mathbf{F}\mathbf{r}\|_\infty. \end{aligned} \quad (2.10)$$

Les mesures présentées ci-dessus ne diffèrent finalement que sur le choix de la norme. Une généralisation peut donc être établie en considérant une mesure basée sur la norme ℓ_p [Cilliers et al., 2007], i.e. $f_p(\mathbf{s}) := \|\mathbf{F}\mathbf{r}\|_p$, pour $p \in \mathbb{N}^*$.

2.1.4 Familles connues

Avant de nous intéresser à l'optimisation de séquences polyphases selon les critères établis ci-dessus, de nombreux travaux ont été effectués sur les codes binaires : que ce soit dans le domaine des télécommunications (par exemple, utilisation des codes CDMA dans les réseaux 3G, CDMA signifiant *Code Division Multiple Access*), dans le radar (nous sommes sûrs que vous avez au moins un exemple sous la main) ou dans la cryptographie, avec certains algorithmes de chiffrages [Golomb et al., 2005]...

La littérature propose de nombreuses familles de séquences pseudo-orthogonales : citons par exemple les codes de Kasami [Kasami, 1966] ou encore les codes de Gold [Gold, 1967], familles de codes binaires présentant de bonnes propriétés d'orthogonalité (autocorrélation et intercorrélation — ce terme, qui évoque la corrélation entre deux séquences distinctes, sera détaillé dans une prochaine section —). Leur construction, plus ou moins élégante, est basée sur les m -séquences et la théorie de Galois. Celle-ci est détaillée dans [Golomb *et al.*, 2005]. Une liste exposant d'autres familles est disponible dans la référence précédente, mais aussi dans [Arlery *et al.*, 2015, Levanon *et al.*, 2004].

Une extension naturelle aux codes binaires est le passage aux codes polyphases. A des fins illustratives, on se propose dans cette sous-section de définir les codes de Barker [Bomer *et al.*, 1989, Borwein *et al.*, 2005] ainsi que de décrire la construction des codes de Frank [Levanon *et al.*, 2004].

2.1.4.1 Codes de Barker

Soit \mathbf{s} une séquence de longueur N et $\text{AACF}(\mathbf{s})$ la séquence d'autocorrélation associée, comme définie par l'expression (2.4), et donc de longueur $2N - 1$. La séquence \mathbf{s} est dite de « Barker » si chacun de ses lobes secondaires est borné par $1/N$, ce qui peut s'écrire :

$$\forall k \in \llbracket 1, 2N - 1 \rrbracket, k \neq N, \quad |(\text{AACF}(\mathbf{s}))_k| \leq 1/N. \quad (2.11)$$

En d'autres termes, un code de Barker est un code performant en terme de PSL, du moins en ce qui concerne l'optimisation de la séquence d'autocorrélation apériodique.

Remarque 2.6. (*Optimalité*) À partir de la Remarque 2.3 et de l'égalité (2.5), explicitons les termes de la séquence d'autocorrélation.

$$\text{AACF}(\mathbf{s}) = \Lambda_N(\mathbf{s})\mathbf{s}^* = \begin{bmatrix} s_N s_1^* \\ s_{N-1} s_1^* + s_N s_2^* \\ \vdots \\ s_1 s_{N-1}^* + s_2 s_N^* \\ s_1 s_N^* \end{bmatrix}. \quad (2.12)$$

Pour chacun de ces éléments, l'inégalité de Cauchy-Schwarz nous permet d'écrire :

$$(\text{AACF}(\mathbf{s}))_k^2 \leq (d/N)^2 \text{ pour } k = d \text{ ou } k = 2N - d, \text{ et avec } d \in \llbracket 1, N \rrbracket. \quad (2.13)$$

En particulier, l'inégalité précédente devient égalité quand $d = 1$ (pour des séquences sous la contrainte de module constant). Le module du premier et du dernier élément de la séquence d'autocorrélation est donc nécessairement égale à $1/N$. La définition (2.11) définit donc bien des séquences optimales, puisqu'on ne peut pas faire mieux (en terme de PSLR)...

A l'origine, cette définition fut proposée dans le cas des codes binaires. Malheureusement, il n'en existe que très peu : en effet, seuls sont connus ceux dont la longueur est inférieure ou égale à $N = 13$, tandis qu'il est couramment admis qu'il n'en existe pas pour des longueurs supérieures [Bomer *et al.*, 1989, Turyn, 1963]. Un code de Barker à 13 éléments est représenté Figure 2.2.

Par extension, cette définition est également appliquée dans le cas des codes polyphases. Contrairement aux codes binaires pour lesquels une « simple » énumération suffit, il est plus ardu de démontrer l'existence de codes de Barker dans le cas polyphase. Une étude statistique [Ein-Dor *et al.*, 2002] a montré qu'il en existe pour des séquences de longueur inférieure à $N = 32$. Celle-ci a été corroborée dans la foulée par la publication d'une liste [Borwein *et al.*, 2005, Levanon *et al.*, 2004].

Cependant, l'ensemble de ces études concerne des séquences relativement courtes. Qu'en est-il pour des séquences plus longues ? La question reste encore ouverte, et la sous-section qui suit ne va pas y répondre. Elle va malgré tout proposer des séquences intéressantes...

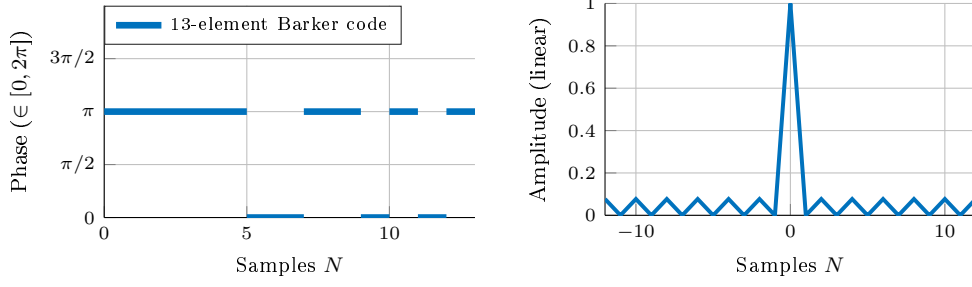


FIGURE 2.2 – Représentation d'un code de Barker à 13 éléments
A gauche : représentation des phases. A droite : représentation de la fonction d'autocorrélation. Observez le niveau maximal des lobes secondaires.

2.1.4.2 Construction des codes de Frank

Initialement présentés comme des séquences présentant de bonnes propriétés d'autocorrélation périodique [Frank *et al.*, 1962], les codes de Frank (du nom de l'un de ses créateurs) se comportent également de manière convenable dans le cas apériodique [Frank, 1963].

Les codes de Frank ne sont définis que pour des longueurs N qui sont des carrés parfaits. Ainsi, soit L un entier tel que $N = L^2$, et soit \mathbf{A} une matrice de taille $L \times L$ telle que :

$$\mathbf{A} = \frac{2\pi}{L} \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 2 & \cdots & L-1 \\ 0 & 2 & 4 & \cdots & 2(L-1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & L-1 & 2(L-1) & \cdots & (L-1)^2 \end{bmatrix}. \quad (2.14)$$

Notons par $\mathbf{a}_1, \dots, \mathbf{a}_L$ les colonnes de \mathbf{A} . Le vecteur de phase $\boldsymbol{\alpha} \in \mathbb{R}^N$, associé au code \mathbf{s} , est défini comme la concaténation de ces vecteurs colonnes.

Remarquons que, par construction, un code de Frank de longueur $N = L^2$ ne peut donc être constitué que de L phases distinctes, régulièrement réparties sur l'intervalle $[0, 2\pi[$. De ce fait, on dit que les codes de Frank sont *multiphases*.

Exemple *Le code de Frank à 16 éléments.*

Pour cet exemple, la matrice \mathbf{A} s'écrit :

$$\mathbf{A} = \frac{\pi}{2} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 \\ 0 & 2 & 4 & 6 \\ 0 & 3 & 6 & 9 \end{bmatrix}. \quad (2.15)$$

S'ensuit l'expression du vecteur de phase $\boldsymbol{\alpha}$:

$$\begin{aligned} \boldsymbol{\alpha} &= \left[0, 0, 0, 0, 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 0, \pi, 2\pi, 3\pi, 0, \frac{3\pi}{2}, 3\pi, \frac{9\pi}{2} \right] \\ &\equiv \left[0, 0, 0, 0, 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 0, \pi, 0, \pi, 0, \frac{3\pi}{2}, \pi, \frac{\pi}{2} \right]. \end{aligned} \quad (2.16)$$

La dernière ligne se justifie par le fait que l'on recherche, par convention, des phases comprises dans l'intervalle $[0, 2\pi[$.

Ce code, ainsi que sa séquence d'autocorrélation, sont représentés Figure 2.1. Une échelle linéaire — par opposition à une échelle logarithmique, majoritairement de mise dans ce manuscrit — a volontairement été choisie pour représenter l'amplitude des lobes, afin de souligner la régularité et le (relatif) bon niveau des lobes secondaires.

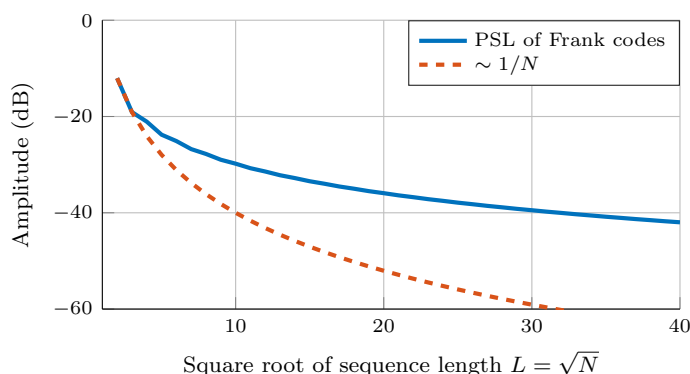


FIGURE 2.3 – Niveau des lobes secondaires des codes de Frank

On représente ici le niveau du lobe secondaire le plus fort (le PSLR) des codes de Frank selon leur longueur, définie ici par le paramètre $L = \sqrt{N}$.

Performances La Figure 2.3 représente les performances des codes de Frank, en terme de lobes secondaires, selon leur longueur. A titre d'exemple, un code de Frank de longueur $N = 64$ propose des lobes secondaires d'au mieux -27.8 dB, tandis que celui pour $N = 1024$ offre -40 dB. Le caractère décroissant de la courbe est rassurant, mais nous n'avons pas encore de point de comparaison à ce stade du chapitre... Nous pouvons simplement établir le fait que les codes de Frank sont des codes de Barker pour $L = 2, 3$.

Le principal inconvénient de toutes ces familles est leur rigidité : elles sont prédéfinies et, par conséquent, ne peuvent pas toujours répondre à des besoins opérationnels. Ceci dit, le principal point de comparaison de ces familles est le niveau de leurs lobes secondaires, et plus précisément leur *Peak-to-Sidelobe Level*. Ainsi, les lignes qui suivent proposent une méthodologie afin de rechercher d'autres séquences intéressantes. Espérons qu'elles fassent mieux !

2.1.5 Un problème d'optimisation

A l'aide des définitions précédentes, il semble assez raisonnable de déterminer un problème d'optimisation dont le but est de rechercher des séquences ayant un niveau de lobes secondaires « avantageux ». Il suffit en effet d'écrire :

$$(P_1) \begin{cases} \min_{\mathbf{s}} & f_p(\mathbf{s}), \\ \text{t.q.} & \mathbf{s} \in \mathbf{T}^N. \end{cases} \quad (2.17)$$

Le problème d'optimisation (P_1) est catégorisé comme un problème d'optimisation *avec contraintes* [Boyd *et al.*, 2004, Floudas *et al.*, 2008]. Il recherche les séquences \mathbf{s} qui permettent de minimiser la *fonction coût* f_p (ou *fonction objectif*), tout en respectant la contrainte de module constant, traduite ici par le tore \mathbf{T}^N .

Afin de reformuler cette contrainte — cela ne veut nullement dire qu'elle est ignorée —, définissons une fonction Φ qui associe à une séquence $\mathbf{s} = [s_1, \dots, s_N]^T$ son vecteur de phase, noté $\varphi = [\varphi_1, \dots, \varphi_N]^T$, c'est-à-dire $\varphi = \Phi(\mathbf{s})$. Le problème en question devient ainsi :

$$(P_2) \quad \min_{[s^1, \dots, s^N]^T} g_p(\varphi) := f_p(\Phi^{-1}(\varphi)). \quad (2.18)$$

Il existe de nombreuses méthodes permettant de résoudre un problème d'optimisation [Boyd *et al.*, 2004, Floudas *et al.*, 2008]. Certaines d'entre elles seront d'ailleurs appliquées dans la section suivante. Mais, avant cela, donnons quelques caractéristiques de notre petit problème...

Dimensionnalité Si l'on considère chacune des phases en tant que variable, leur nombre N peut être assez important : il peut aisément dépasser le millier en pratique. De ce fait, de nombreuses méthodes d'optimisation perdent de leur efficacité, comme notamment les méthodes dites « stochastiques », qui explorent l'espace de recherche.

Citons notamment un exemple présenté dans [Bellman, 2003], illustré Figure 2.4, et que l'on peut présenter ainsi : considérons un quadrillage régulier du segment $[0, 1]$ à l'aide de $10^2 = 100$ points. Si nous voulons faire de même dans l'espace $[0, 1]^{10}$, il nous faudrait alors 10^{20} points ! Ce qui n'est guère tenable...

D'ailleurs, les problèmes d'optimisation avec de nombreuses variables sont catégorisés en tant que problèmes à *grande échelle* [Hager *et al.*, 1994, Pillo *et al.*, 2006]. Certaines astuces peuvent être exploitées, comme la *séparabilité* [Tsurkov, 2013].

Séparabilité La séparabilité se formule de manière relativement claire par l'énoncé de sa définition formelle. Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite séparable si :

$$\arg \min_{(x_1, \dots, x_n)} f(x_1, \dots, x_n) = \left(\arg \min_{x_1} f(x_1, \dots), \dots, \arg \min_{x_n} f(\dots, x_n) \right). \quad (2.19)$$

Cela signifie que le minimum d'une fonction séparable peut être découvert à l'aide de n sous-problèmes d'optimisation, ce qui facilite *a priori* la tâche. Malheureusement, notre problème (P_1) n'est clairement pas séparable, comme en témoigne la définition de la fonction d'auto-corrélation (2.4). Prenons l'exemple suivant : la fonction Rastrigin, connue pour évaluer des méthodes d'optimisation et représentée à la Figure 2.5, est définie par :

$$\begin{aligned} & [-5.12, 5.12]^n \rightarrow \mathbb{R} \\ \mathbf{x} \mapsto f(\mathbf{x}) & := 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)). \end{aligned} \quad (2.20)$$

Cette fonction est séparable additivement. Il suffit en effet de poser une fonction g à variable réelle telle que $g : x \mapsto g(x) = 10 + x^2 - 10 \cos(2\pi x)$, de sorte que $f(\mathbf{x}) = \sum_{i=1}^n g(x_i)$.

Convexité Une fonction f à valeurs réelles est dite convexe si elle satisfait :

$$f(\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y}), \quad (2.21)$$

pour tout $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, avec $\lambda \in [0, 1]$. Elle est *strictement* convexe si l'inégalité est stricte. Les ensembles convexes se définissent de manière similaire. Géométriquement, cela signifie que le segment entre les points $(\mathbf{x}, f(\mathbf{x}))$ et $(\mathbf{y}, f(\mathbf{y}))$ est « au-dessus » de son graphe, tandis que les tangentes en tout point sont « en-dessous ». Cela est illustré Figure 2.6. Cette comparaison n'est pas si anodine, car les fonctions convexes admettent une propriété fort sympathique. Un minimum *local* est également un minimum *global*. Notons par ailleurs qu'il est unique si la fonction est strictement convexe [Boyd *et al.*, 2004].

Une branche en optimisation, appelée sobrement « optimisation convexe », s'intéresse aux problèmes d'optimisation dits convexes, c'est-à-dire ceux dont la fonction et l'espace d'étude sont convexes [Bonnans *et al.*, 2013, Boyd *et al.*, 2004, Nocedal *et al.*, 2006], et recherche en particulier des algorithmes adaptés à leur résolution. Mais, malheureusement — encore une fois —, nos deux problèmes ne le sont pas, car ni la fonction de coût (malgré le fait qu'elle soit basée sur une norme), ni l'espace de recherche (le tore \mathbf{T}^N) ne sont convexes, comme représenté Figure 2.7.

Dérivabilité Comme nous le verrons dans la section suivante, de nombreuses méthodes d'optimisation requièrent le gradient de la fonction objectif. Ce dernier est souvent utilisé pour déterminer la qualité du minimum (minimum local/global, point selle) ainsi que la direction d'exploration [Boyd *et al.*, 2004, Nocedal *et al.*, 2006]. Nous évoquerons ce dernier point dans la section suivante.

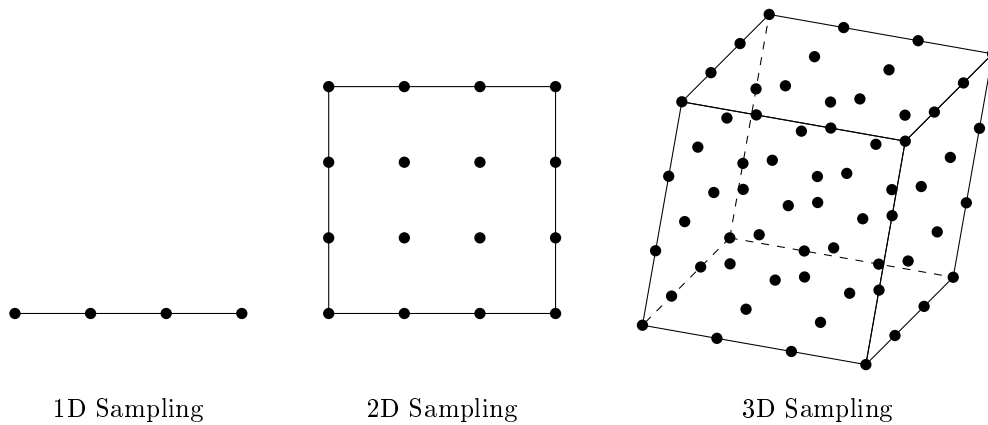


FIGURE 2.4 – Représentation du fléau de la dimension
 Le nombre d'échantillons s'accroît de façon exponentielle avec la dimension...

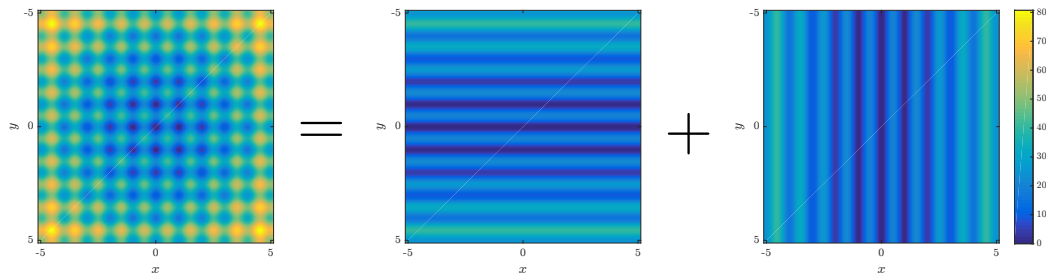


FIGURE 2.5 – Séparabilité de la fonction Rastrigin
 La fonction Rastrigin, représentée à gauche en dimension 2, peut se décomposer selon chacune de ses variables.

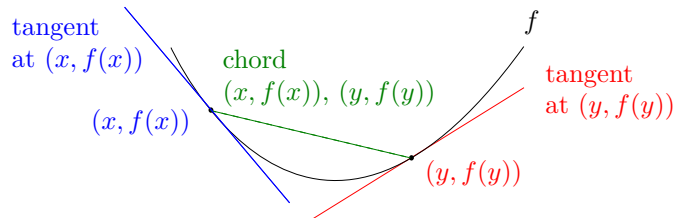


FIGURE 2.6 – Propriétés géométriques des fonctions convexes
 Toute tangente est en dessous de la courbe de la fonction f , tandis que la corde entre deux points est au-dessus.

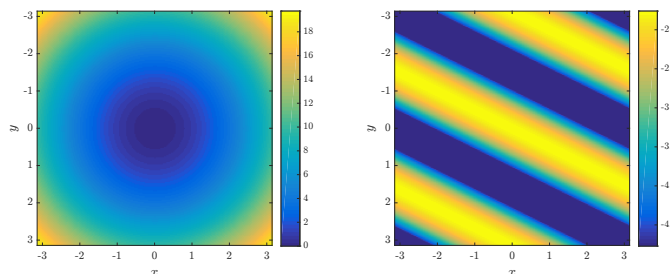


FIGURE 2.7 – Une fonction convexe, une fonction d'autocorrélation
 Une fonction convexe, ici à gauche, ne présente qu'un seul minimum global, tandis que la fonction de corrélation, non-convexe et représentée ici pour $N = 3$, en possède plusieurs.

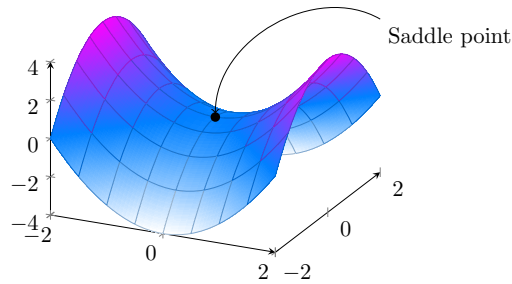


FIGURE 2.8 – Représentation d'un point selle

Un point selle — ici, en $(0,0)$ — a la particularité d'être « à la fois » un minimum et un maximum local, selon les points de vue... Remerciements : Ammar Mian.

Soit f une fonction dérivable. Notons ∇f son gradient. On suppose que f est localement convexe autour du point \mathbf{x} . A l'aide d'un développement de Taylor au premier ordre, on a, pour tout point \mathbf{y} au voisinage de \mathbf{x} (cf. Figure 2.6) :

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}). \quad (2.22)$$

Pour que \mathbf{x} soit un minimum local — ce qui peut s'écrire $f(\mathbf{x}) \leq f(\mathbf{y})$ pour tout \mathbf{y} proche de \mathbf{x} —, il est donc nécessaire que $\nabla f(\mathbf{x}) = 0$. Cette condition est connue comme la *condition de premier ordre* [Boyd *et al.*, 2004]. Insistons sur le fait que celle-ci ne garantit en aucun cas que \mathbf{x} est un optimum local (sauf en dimension 1), mais seulement un point stationnaire (ou point selle, comme illustré Figure 2.8). Il existe également une *condition de second ordre* : si la hessienne de f (la matrice de ses dérivées secondes) est semi-définie positive en un point \mathbf{x} , alors ce dernier est un minimum local.

Notre fonction de coût f_p n'est pas dérivable au sens strict du terme (*i.e.* holomorphe). Elle admet cependant l'existence de dérivées partielles, ce qui sera, malgré tout, suffisant pour appliquer les méthodes de recherche linéaire.

Si nous rassemblons toutes ces remarques, nous pouvons conclure que la résolution de notre problème d'optimisation ne sera pas si simple : la forte dimensionnalité ne nous permettra pas d'appliquer convenablement l'ensemble des algorithmes d'optimisation, tandis qu'il nous sera difficile de garantir l'optimalité de notre candidat à cause de la non-convexité. Cela peut donc expliquer les nombreux travaux sur les recherches de séquences parfaites. Pour anecdote, il y a toujours une session sur le *waveform design* dans une conférence radar. Sans doute due à l'émergence du MIMO et des radars cognitifs, certes... En ce sens, la section suivante recherchera les méthodes d'optimisation les plus adaptées à notre problème.

2.2 Comparatif des méthodes d'optimisation

Que ce soit pour une optimisation à grande échelle ou non, les méthodes d'optimisation peuvent se catégoriser selon différents critères [Cavazzuti, 2012, Floudas *et al.*, 2008].

- Déterministe ou stochastique. Y a-t-il une part d'aléatoire dans le processus de recherche ? Une méthode stochastique répond positivement (logique !) à cette question [Dréo *et al.*, 2006, Glover *et al.*, 2003].
- Avec ou sans contraintes. Le problème d'optimisation présente-t-il des contraintes sur les variables ? La grande majorité des méthodes déterministes ne sont pas de prime abord les plus efficaces avec les contraintes [Cavazzuti, 2012], sauf dans des cas très particuliers (contraintes linéaires, fonction et espace convexes). Les méthodes stochastiques se contentent *a minima* d'un ensemble de définition. Une relaxation, à l'instar d'une pénalisation, peut être employée pour supprimer les contraintes.

- Simple ou multi-objectif. Le problème d'optimisation est-il défini par plusieurs fonctions objectifs ? Les méthodes multi-objectifs permettent de définir le front de Pareto associé au problème ; le front de Pareto est une représentation du compromis entre les différentes fonctions objectifs [Branke *et al.*, 2008, Collette *et al.*, 2004, Désidéri, 2009].
- Local ou global. La convergence de l'algorithme est-elle locale ou globale ? L'ensemble des méthodes déterministes converge vers un minimum local, sauf dans des cas particuliers (fonction convexe) [Boyd *et al.*, 2004].
- Optimisation discrète ? On parle d'optimisation discrète lorsque les variables ne sont pas continues. L'optimisation discrète est considérée comme un domaine à part entière en optimisation [Korte *et al.*, 2006].

Les méthodes d'optimisation à grande échelle sont généralement identiques à celles utilisées dans le cas « habituel » [Hager *et al.*, 1994, Pillo *et al.*, 2006, Tsurkov, 2013]. La taille pénalise surtout les méthodes stochastiques, comme mentionné dans le paragraphe précédent.

Cette section s'intéresse à la résolution de notre problème d'optimisation... enfin, presque. Le terme *résolution* est probablement assez maladroit — mais on s'entêtera à l'employer de nombreuses fois dans ce manuscrit ! — dans la mesure où il nous sera impossible de déterminer, quelle que soit la méthode, si la séquence candidate est le/un optimum ou non (on pourra espérer au mieux le caractère local). De ce fait, il nous faut tester différentes méthodes d'optimisation, afin de comparer les candidats qu'elles nous proposent.

On se penchera d'abord sur les méthodes stochastiques, avec notamment l'application de l'algorithme CMA-ES, acronyme pour *Covariance Matrix Adaptation Evolution Strategy* [Hansen, 2016]. Cet algorithme n'a pas encore été employé pour notre problème, à notre connaissance. Ensuite, on évoquera les méthodes déterministes basées sur le gradient, avec la suggestion d'une procédure permettant d'échapper aux points stationnaires.

2.2.1 Méthodes heuristiques

2.2.1.1 Présentation

Intéressons nous d'abord aux méthodes dites *méta-heuristiques*. [Blum *et al.*, 2003] ont établi une liste de leurs principales caractéristiques :

1. Les métaheuristiques sont des stratégies itératives « guidant » le processus de recherche. Leur but est d'explorer efficacement l'espace de recherche afin de trouver des solutions (quasi-) optimales.
2. Quasi-optimales, car elles essaient d'approcher au mieux l'optimum, sans pour autant le connaître. Ce sont donc à ce titre des méthodes approximatives (et accessoirement, non déterministes pour la plupart).
3. Des mécanismes peuvent être incorporés afin de ne pas être piégé par des minima locaux.
4. Les méthodes métaheuristiques se doivent de ne pas être spécifiques à un problème d'optimisation en particulier. Elles sont donc adaptées pour les optimisations dites « boîtes noires » (*black-box optimization*), où l'expression analytique de la fonction de coût n'est pas connue.

Il existe une grande diversité de méthodes comme illustré à la Figure 2.9. Elles se différencient pour la plupart par le choix de la méthodologie d'*exploration*, et donc du choix du *voisinage* [Dréo *et al.*, 2006]. Sous ces deux termes sont regroupés les mécanismes permettant de définir et de choisir un candidat. Pour plus de détails, nous vous invitons à considérer [Dréo *et al.*, 2006] ou [Glover *et al.*, 2003].

- Population ou à parcours : à chaque itération, l'algorithme fait-il évoluer un seul point ? Se base-t-il sur toute une population ?
- Emploi de mémoire ou non : l'algorithme se base-t-il sur les itérations précédentes pour générer les candidats suivants ?

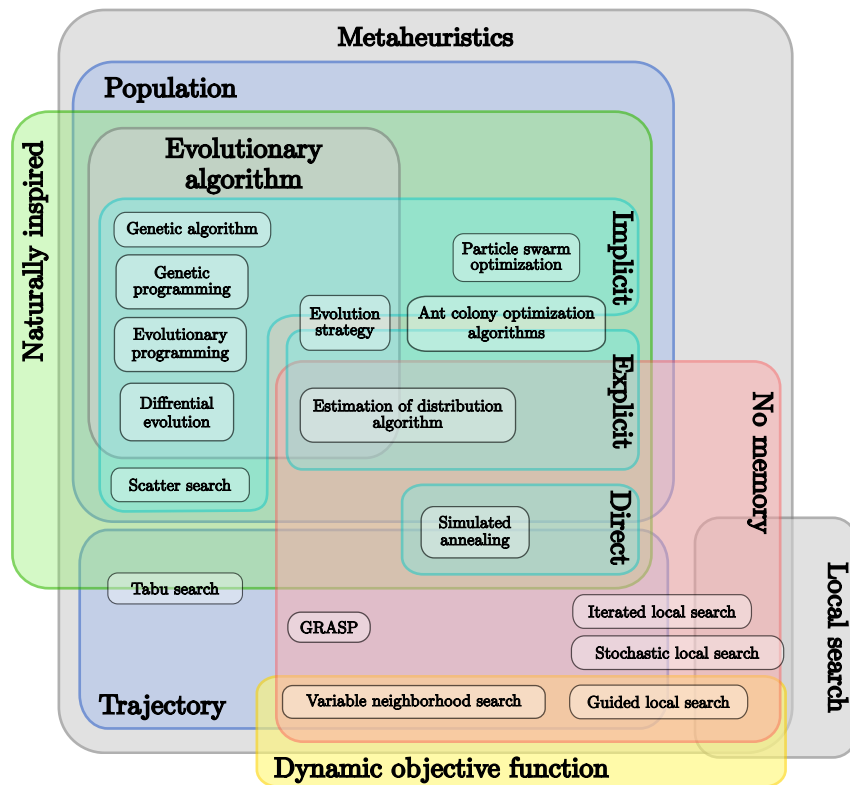


FIGURE 2.9 – Classification des méthodes métaheuristiques
 Crédits : Johann Dreó [Dreó et al., 2006].

- Implicite ou explicite : la génération d'un candidat se fait-elle selon une distribution connue ou non ? Pour le second cas, on peut penser à certains algorithmes génétiques : bien que la loi de génération peut être définie (*via* une loi normale, par exemple), le choix de ses paramètres est régi par la valeur des candidats.

Nombre de ces méthodes ont été employées dans la littérature, pour la recherche de codes de phase. Citons notamment, sans ordre particulier : la recherche tabou [Farnane *et al.*, 2018, Liu *et al.*, 2012a], l'optimisation par essais particuliers [Jamil *et al.*, 2016], l'algorithme de colonies de fourmis [Malekzadeh *et al.*, 2012, Reddy *et al.*, 2014] ou bien les algorithmes génétiques ou évolutionnaires [Dam *et al.*, 2004, Deng, 2004, Liu *et al.*, 2006]...

Il aurait pu être utile de comparer l'ensemble de ces méthodes pour notre problème, mais il faut avouer que cela nous prendrait beaucoup de temps. Dans cette sous-section, nous allons mettre l'emphase sur les algorithmes évolutionnaires. Plus précisément, nous allons en comparer deux :

- Le premier, assez simpliste, est une recherche aléatoire. Ce choix a été motivé par le fait que cet algorithme est celui qui a initialement été utilisé chez Thales pour optimiser des formes d'onde pour le radar MIMO [Guyvarch *et al.*, 2012].
- Le second, beaucoup plus évolué, est l'algorithme CMA-ES, pour *Covariance Matrix Adaptation Evolution Strategy* [Hansen, 2016]. Développé par l'INRIA, il se caractérise par la génération de la population selon une gaussienne. Si la loi n'est pas spécifiée, cela donne un algorithme très généraliste, IGO (pour *Information-Geometric Optimization*), qui permet de généraliser pas mal d'algorithmes évolutionnaires [Ollivier *et al.*, 2011].
- Reconnaissons que nous n'incluons pas les résultats pour le recuit simulé. Nous n'avons pas trouvé de loi de température qui permet d'améliorer significativement les résultats par rapport à une recherche aléatoire.

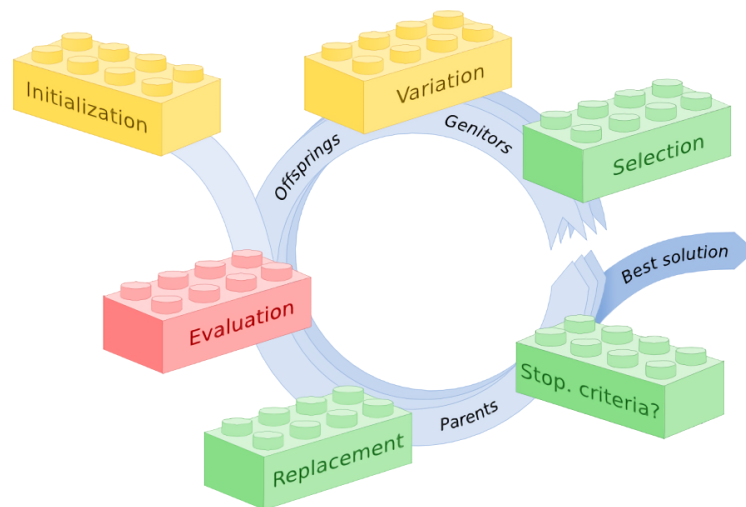


FIGURE 2.10 – Les différentes étapes d'un algorithme évolutionnaire
 Crédits : <http://eodev.sourceforge.net/>

2.2.1.2 Méthodes évolutionnaires

Comme son nom l'indique, un algorithme évolutionnaire est inspiré de la théorie éponyme de Darwin. Elle se caractérise par l'évolution d'une population, selon deux opérateurs : la *sélection*, une étape de choix et de remplacement des éléments de la population, et la *variation* qui modifie la population à l'aide d'opérateurs de *mutation* et de croisements. De ce fait, les meilleurs éléments de la population, au sens de la fonction coût, ont plus de chance d'être conservés, d'où l'analogie avec la théorie de l'évolution.

Les différentes étapes d'un algorithme évolutionnaire sont illustrées Figure 2.10 ; seule l'étape d'initialisation ne fait pas partie du processus itératif [Dréo *et al.*, 2006]...

0. Initialisation : génération d'une population composée de $\mu \in \mathbb{N}^*$ éléments — appelés « parents » — selon une certaine loi, généralement de manière aléatoire, tout simplement.
1. Sélection : sélection de ρ parents parmi les nouveaux, pour l'étape de reproduction. Cette désignation peut par exemple s'effectuer de manière aléatoire. Notez qu'il est assez courant de choisir les μ parents, *i.e.* $\rho = \mu$.
2. Reproduction : à partir de ces ρ parents, génération de λ enfants, par une application des opérateurs de croisement et/ou de mutation. Le croisement consiste en une variation des parents. L'opérateur de mutation ajoute une part d'aléatoire sur les enfants.
3. Évaluation : évaluation de chaque enfant de la population par la fonction objectif. Comme rappelé précédemment, l'expression analytique de cette dernière n'est pas nécessaire : une optimisation *black-box* est ainsi possible.
4. Remplacement : parmi une certaine partie de la population, sélection des μ parents. « Certaine », car le choix des parents peut se faire sur des ensembles distincts, selon les algorithmes.
 - On parle d'un algorithme $(\mu + \lambda)$ -ES lorsque la sélection se fait sur l'ensemble comprenant les μ parents *et* les λ enfants.
 - On parle d'un algorithme (μ, λ) -ES lorsque l'on choisit les μ parents *parmi* les λ enfants.
 - Et il y a d'autres variantes...

Généralement, en ce qui concerne les algorithmes génétiques, ce sont les meilleurs — ceux qui présentent la fonction coût la plus faible — qui sont plébiscités.

- ?. Les conditions d'arrêt sont assez diverses : limite sur le nombre d'itérations, le nombre d'évaluations par la fonction coût ou le temps de calcul, faible amélioration pour un nombre d'itérations donné, équité des parents au regard de la fonction coût [Hansen, 2016]...

La différenciation entre les méthodes évolutionnaires s'établit par le choix des définitions (valeurs des différents paramètres, loi d'échantillonnage...) pour chacune de ces étapes.

Remarque 2.7. *Le croisement entre plusieurs parents se traduit habituellement par une combinaison linéaire entre eux. Ce point est assez délicat dans le cas d'optimisation de séquences, car la somme de deux séquences présentant de bonnes propriétés d'orthogonalité ne résulte pas nécessairement en une troisième avec de bonnes propriétés !*

Considérons un exemple jouet. Il n'existe que deux codes de Barker binaires de longueur $N = 4$ [Levanon et al., 2004] dont les phases sont les suivantes : $\varphi^1 := [0, 0, 0, \pi]$ et $\varphi^2 := [0, 0, \pi, \pi]$. Malheureusement, la somme de ces deux vecteurs phases ne retourne pas l'un deux, ce qui signifie nécessairement que la corrélation est dégradée...

De ce fait, l'étape de croisement se révèle être simplement dans notre cas une manière d'échantillonner, sans arrière-pensée sur les performances des parents.

Recherche aléatoire Si nous reprenons les notations définies précédemment, la recherche aléatoire est un algorithme (1 + 1)-ES. Son principe est assez sommaire : un candidat est tiré de manière aléatoire. S'il offre une réduction de la fonction coût, ce candidat est conservé. Son principe est détaillé dans le Tableau 2.1, avec les notations de notre problème (P_2).

Algorithme Recherche aléatoire	
Données	Initialisation d'un vecteur de phase φ
Répéter	1. Tirage aléatoire d'un candidat $\varphi' \in [0, 2\pi[N$ 2. Évaluation du candidat par la fonction de coût g_p Sélection si amélioration : $\varphi = \varphi'$.
Jusqu'à	Un critère d'arrêt est atteint.

TABLE 2.1 – Algorithme d'optimisation alternée

Une petite alternative consiste à effectuer une recherche aléatoire locale, qui ne modifie qu'une seule phase à la fois. Cette alternative sera utilisée par la suite ; elle est expliquée dans le Tableau 2.2.

Algorithme Recherche aléatoire locale	
Données	Initialisation d'un vecteur de phase $\varphi = [\varphi_1, \dots, \varphi_N]^T$
Répéter	1. <i>Génération d'un candidat</i> $\varphi' = [\varphi'_1, \dots, \varphi'_N]^T$: Tirage aléatoire d'un indice $n \in \llbracket 1, N \rrbracket$ Tirage aléatoire d'une phase $\varphi \in [0, 2\pi[$ Génération du candidat : $\varphi' = \varphi, \varphi'_n = \varphi$ 2. <i>Évaluation du candidat par la fonction coût</i> g_p Sélection si amélioration : $\varphi = \varphi'$.
Jusqu'à	Un critère d'arrêt est atteint.

TABLE 2.2 – Algorithme de recherche aléatoire locale

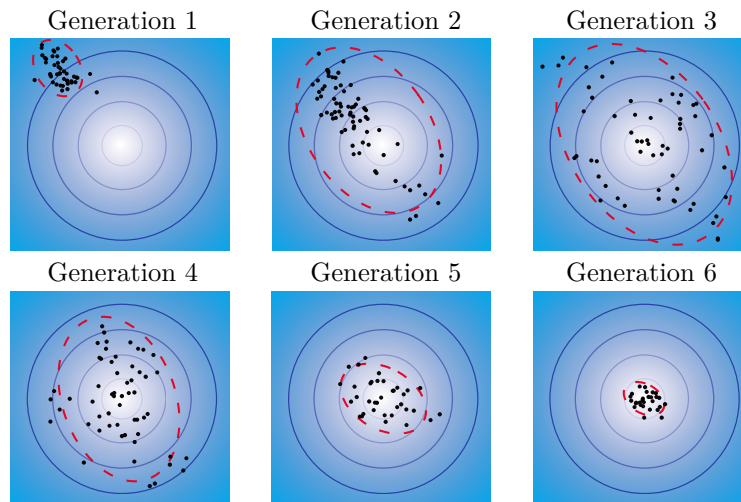


FIGURE 2.11 – Illustration de l'algorithme CMA-ES

L'ellipse décrit la distribution de la population. Les antécédents des faibles valeurs de la fonction objectif sont représentés d'une couleur claire. Remerciements : Ammar Mian.

CMA-ES Il est assez délicat de détailler l'ensemble des paramètres de l'algorithme CMA-ES sans exposer les nombreux concepts périphériques requis ; les plus curieux sont invités à lire [Hansen, 2016, Ollivier *et al.*, 2011]. Malgré tout, nous allons essayer d'expliquer les grandes lignes de cet algorithme à l'aide de l'illustration 2.11.

CMA-ES est un algorithme $(\mu/\mu_w, \lambda)$: le terme μ/μ_w indique que la génération des enfants se fait à partir d'une version pondérée des μ parents. Plus précisément, les enfants sont tirés de manière aléatoire depuis une loi gaussienne ; les paramètres de cette gaussienne sont notamment déterminés par la qualité des parents. La Figure 2.11 montre ainsi que les enfants sont contenus dans une ellipse.

Examinons de manière clinique la Figure 2.11. Une première génération est tirée de manière aléatoire selon une gaussienne, représentée ici par une ellipse. Les candidats les plus intéressants « se situent » au sud-est, ce qui entraîne *de facto* une déviation de la distribution (génération 2 et 3). Par suite, la population commence à se concentrer autour d'un minimum : l'ellipse se réduit petit à petit, et l'algorithme finit par converger.

Une fois qu'un minimum a été « atteint », une autre exploration de l'espace de recherche est réalisée, si le nombre d'évaluations maximal le permet (non représenté sur la figure). Cela s'exprime par une augmentation de la taille de l'ellipse, ou par une nouvelle initialisation.

Avantages et inconvénients Nous avons déjà plus ou moins mentionné les avantages et les inconvénients de ces méthodes évolutionnaires. Nous en rappelons quelques-uns ci-dessous, en guise de conclusion.

- Les méthodes stochastiques peuvent aisément s'appliquer pour n'importe quel problème d'optimisation, car aucune hypothèse n'est nécessaire sur la fonction objectif : dérivabilité, convexité... En outre, l'expression analytique de cette dernière n'est pas requise, ce qui confère à ces méthodes une utilisation possible dans des problèmes dites « boîtes noires ».
- La majorité d'entre elles présente une convergence *globale*, par ergodicité. Du fait que l'exploration se fait sur l'ensemble de recherche tout entier, il y a une chance (certes, infime) que le minimum global soit atteint...
- Mais, en pratique, il est difficile de déterminer la nature d'un candidat, et surtout son caractère optimal.
- Les problèmes en grande dimension peuvent réduire drastiquement leurs performances (malédiction de la dimension).

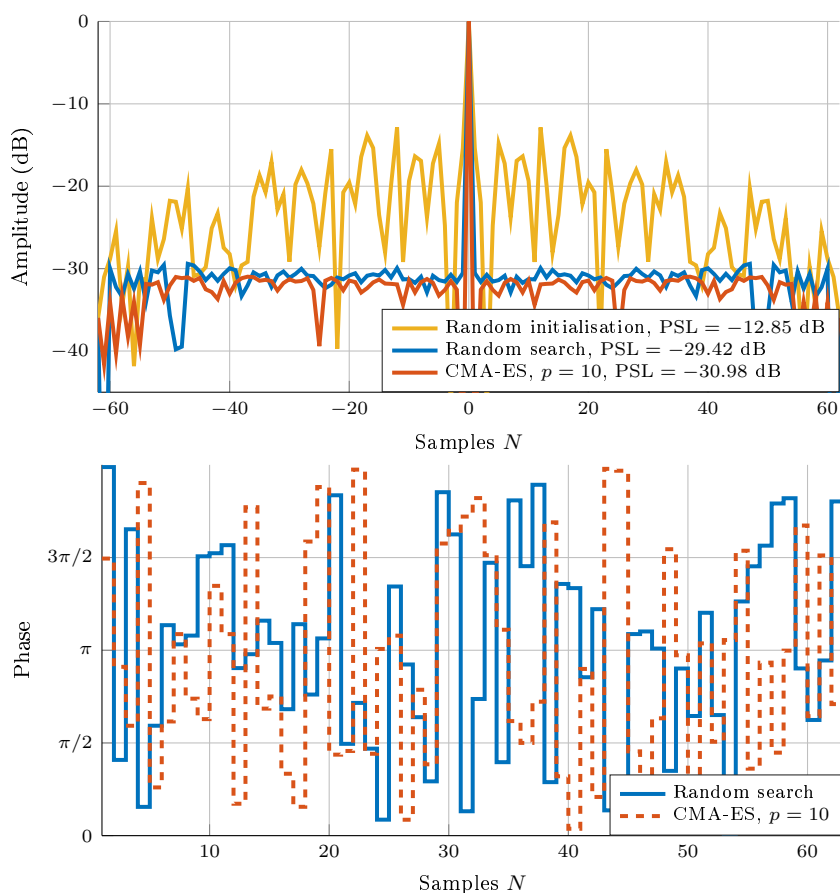


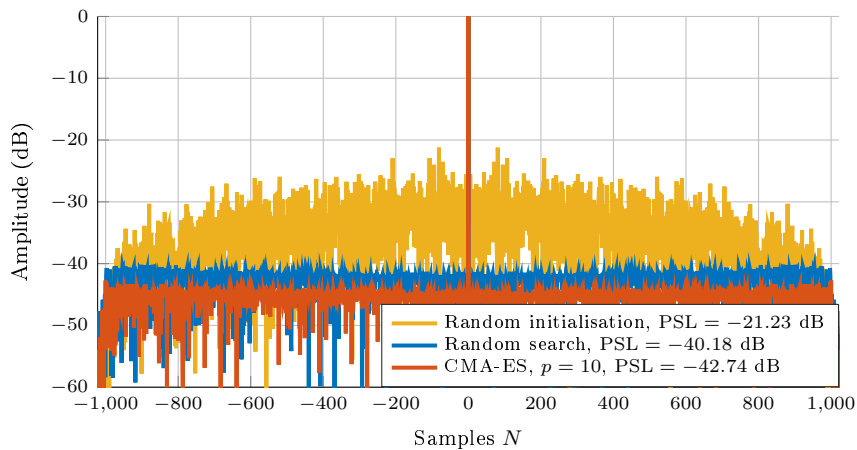
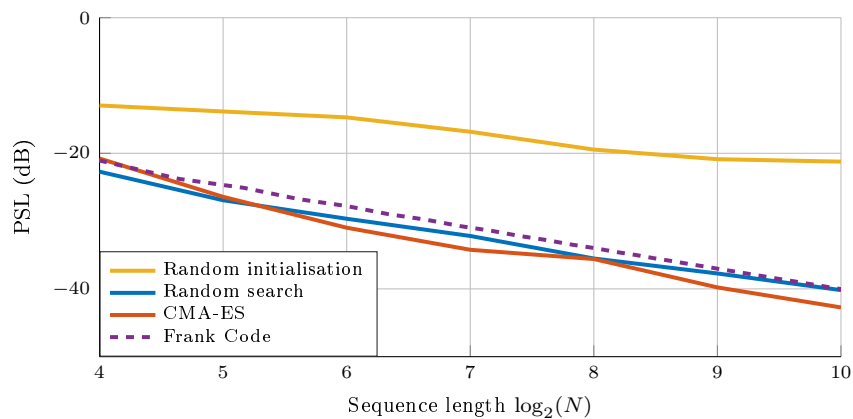
FIGURE 2.12 – Optimisation à l'aide de méthodes stochastiques, $N = 63$
 En haut : représentation des séquences d'autocorrélation. En bas : représentation des phases de chaque séquence. Oui, c'est le chaos.

2.2.1.3 Premiers résultats

Il est maintenant temps de reprendre notre problème d'optimisation (P_2). Ce paragraphe s'accorde à appliquer une méthode de recherche aléatoire — initialement utilisée par Thales pour les codes de phase — ainsi que l'algorithme CMA-ES sur différents exemples. On s'intéresse ici à l'optimisation du PSL, plus couramment utilisé dans les applications radars. Les observations qui en suivent pourront probablement s'extrapoler aux autres normes (l'ISL par exemple).

De nombreuses séquences d'autocorrélation sont reproduites dans les figures qui suivent. Bien que ce ne sont pas les premières que nous présentons dans ce manuscrit, rappelons deux évidences. D'une part, l'objectif est généralement d'obtenir une séquence qui s'apparente à un Dirac, avec une forte valeur au « centre », et les plus faibles possibles autour, notamment à des fins de performances. D'autre part, cette forte valeur symbolise, en somme, le pic principal. Son module est d'ailleurs constant, dû à la contrainte éponyme.

Performances Commençons par l'optimisation d'une séquence de longueur $N = 63$, représentée Figure 2.12. En guise d'initialisation, un vecteur de phases est tiré de manière aléatoire, selon une loi uniforme entre 0 et 2π . Ce dernier exhibe des lobes secondaires assez irréguliers, dont le plus élevé est aux alentours des -13 dB. L'application de nos deux méthodologies évolutionnaires confère une belle amélioration, d'environ 17 dB (selon le lobe le

FIGURE 2.13 – Optimisation à l'aide de méthodes stochastiques, $N = 1023$ FIGURE 2.14 – Comparaison du PSL selon la longueur de la séquence N
Affichage du meilleur PSL obtenu après 10 instances

plus haut, PSL oblige), avec un léger avantage pour CMA-ES de 1.5 dB. Notons que ce gain n'est « que » de l'ordre de 2–3 dB par rapport à un code de Frank à 64 éléments (–27.8 dB, Figure 2.3). Le choix du paramètre p , fixé à 10 pour CMA-ES, sera discuté ultérieurement.

Une initialisation identique, une amélioration similaire, mais qu'en est-il du code de phase obtenu? Le vecteur de phase de chacune des séquences obtenues est visualisé sur le bas de la Figure 2.12. Cette figure est assez éloquent : de nombreuses phases diffèrent et, les deux solutions trouvées sont très différentes...

Effectuons la même étude, mais cette fois-ci pour le cas $N = 1023$, Figure 2.13. Compte-tenu de l'initialisation — aléatoire, comme précédemment — une amélioration est encore notable. Cependant, le fossé s'est creusé entre, d'un côté, les codes de Frank et la méthode de recherche aléatoire, aux alentours de –40 dB et, de l'autre, CMA-ES, en avance de presque 3 dB. Peut-on ainsi dire que la longueur de la séquence est en cause? La figure qui suit investigate cela.

La Figure 2.14 représente le meilleur PSL obtenu selon la dimension de la variable s . Pour cela, 10 optimisations successives ont été opérées et ce, pour différentes longueurs de séquences. Les différentes méthodes se retrouvent dans un mouchoir de poche, et nous pouvons supputer que l'apport du CMA-ES se fait réellement sentir en grande dimension. Mais, heureusement, l'amélioration envers l'initialisation s'accroît dans tous les cas, ce qui nous rassure sur l'intérêt de l'optimisation.

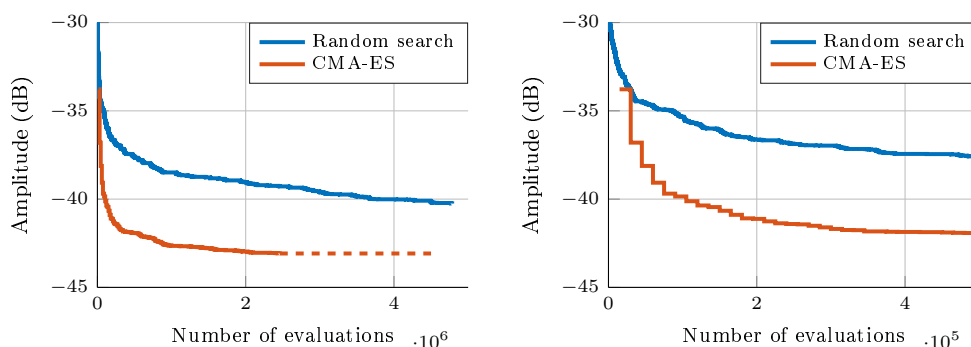


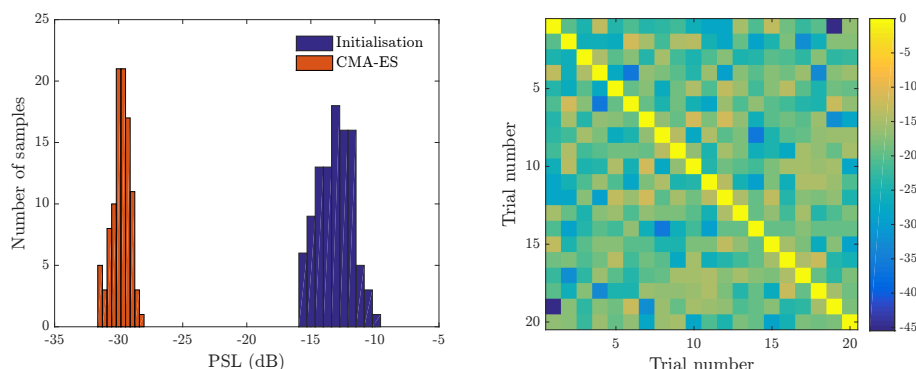
FIGURE 2.15 – Étude de convergence de deux algorithmes évolutionnaires
 A gauche : les pointillés marquent un prolongement constant de la courbe ; l'algorithme s'est en effet arrêté sur un critère de convergence.
 A droite : zoom sur les premières évaluations de la courbe de gauche.

Convergence L'ensemble des résultats présentés ci-dessus n'est le fruit que d'une seule instance, choisie sans raison apparente. Il aurait été plus rigoureux d'effectuer de nombreux tirages pour solidifier nos conclusions... Mais, combien ? Difficile à dire, pour des problèmes de cette dimension. Nous espérons de *bona fides* que ces résultats soient représentatifs du ressenti que nous avons eu lors de leur implémentation. Cela dit, nous allons essayer malgré tout d'analyser les qualités de convergence de nos deux algorithmes, en mentionnant deux constatations.

La première concerne la vitesse de convergence et se sert de la Figure 2.15. Cette dernière représente le niveau des lobes secondaires du meilleur candidat courant selon le nombre d'évaluations de la fonction objectif, *i.e.* le nombre de candidats visités. Après une belle chute dans les premières itérations, la décroissance se fait beaucoup plus lente pour la recherche aléatoire. Il est même probable que celle-ci ne soit pas terminée... L'algorithme CMA-ES propose la même conduite, mais de manière plus prononcée : la convergence vers un bon candidat — bien meilleur que celui obtenu par la recherche aléatoire à la fin de cette expérimentation — est beaucoup plus rapide. S'ensuit une période d'alternance entre accroissement et focalisation du domaine d'exploration, ce qui peut plausiblement expliquer la difficulté à découvrir des bonifications.

La seconde s'intéresse à la fiabilité des résultats proposés par l'algorithme CMA-ES. Pour cela, nous avons réalisé, Figure 2.16, 100 instances d'une optimisation d'une séquence de longueur $N = 63$, avec chacune une initialisation différente. Malgré l'hétérogénéité de ces dernières, le résultat obtenu est relativement uniforme, avec des solutions qui tournent aux alentours de -30 dB. Il est intéressant de noter la constance de cet algorithme : aucune instance n'a échoué, dans le sens où toutes les optimisations ont apporté un gain avéré. Par ailleurs, la matrice de covariance entre les séquences obtenues (ce qui équivaut à une corrélation à retard nul) montre une disparité approximative de ces solutions. Pouvons-nous ainsi conclure que les différents minima locaux se situent à un niveau quasi-équivalent ? Cette question nous renvoie naturellement aux questions encore ouvertes sur la répartition et le niveau des minima de notre fonction objectif...

Un dernier mot sur la convergence, mais algorithmique cette fois : il est difficile de comparer le coût calculatoire, car les deux algorithmes ont des comportements inverses. L'algorithme de recherche aléatoire nécessite de nombreuses itérations, mais chacune ne coûte qu'une seule évaluation de la fonction objectif. Le comportement est inverse pour CMA-ES.



	Moyenne	Écart-type
Initialisation	-13.14 dB	1.33 dB
CMA-ES	-29.84 dB	0.72 dB
Gain	16.69 dB	1.49 dB

FIGURE 2.16 – Efficacité de la méthode CMA-ES après 100 tirages différents
A gauche : répartition du PSL de chaque tirage, avant et après optimisation.
A droite : corrélation entre les 20 premières séquences obtenues.
En bas : quelques valeurs statistiques. Séquences de longueur $N = 63$.

Choix de la norme Intéressons-nous maintenant au choix de la norme ℓ_p lors de l'application de l'algorithme CMA-ES. Nous rappelons que nous recherchons toujours dans cette section une séquence avec le meilleur *Peak-to-Sidelobe Level*, c'est-à-dire qui est avantageuse envers la fonction coût f_∞ . Cependant, lorsqu'on applique cet algorithme en faisant varier le paramètre p , nous obtenons souvent quelque chose qui ressemble à la Figure 2.17.

Le tracé des séquences d'autocorrélation est clair : la séquence optimisée selon la fonction coût f_{10} ($p = 10$) présente un meilleur PSL que celle optimisée précisément pour ce critère ! Ce comportement profitable ne se retrouve pas pour n'importe quelle valeur de p , comme l'illustre la courbe qui suit que nous n'avons pas encore commenté (Figure 2.18). L'interrogation qui vient naturellement concerne le choix optimal de cette norme. La seconde partie de la Figure 2.17 tente d'y répondre.

Une séquence est tirée de manière aléatoire, et joue le rôle d'initialisation pour différents problèmes basés sur (P_1) . Ceux-ci se distinguent par le choix du paramètre p , modifiant ainsi la norme appliquée sur le vecteur d'autocorrélation. Cette procédure est effectuée une dizaine de fois et les résultats sont reportés Figure 2.17. Cette figure est également assez explicite : une minimisation sous une « petite » norme ne permet pas d'obtenir une séquence avec un bon PSL, ce qui semble assez logique. Le caractère décroissant de la courbe est également assez attendu, puisqu'une augmentation du paramètre p nous rapproche de la norme infinie. Cependant, une petite remontée est observée à la fin (la dernière valeur de la courbe représente la norme infinie). Bien qu'il est difficile d'en tirer des enseignements avec si peu de tirages, est-ce malgré tout un signe ?

Conclusion Ces premiers résultats sont assez encourageants. Le problème semble potentiellement bien posé puisque sa résolution nous permet d'obtenir des séquences intéressantes. Un avantage certain est donné à l'algorithme CMA-ES en comparaison de la recherche aléatoire : celui-ci propose de meilleures solutions, et cela à l'aide de moins d'évaluations. Le second semble (légèrement plus) en pâtir lorsque la dimension du problème devient important. Remarquons que, en pratique, l'optimisation d'une séquence de longueur $N = 1023$ prend quelques heures sur un ordinateur personnel.

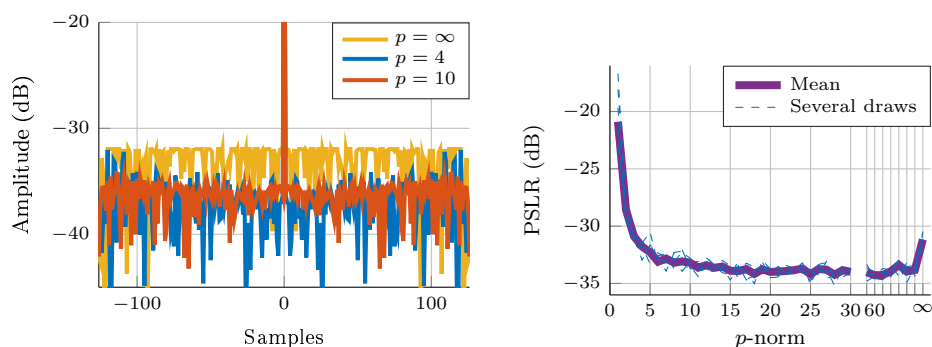


FIGURE 2.17 – Choix de la norme de la fonction coût pour l'algorithme CMA-ES
A gauche : séquences d'autocorrélation après une optimisation d'une séquence de longueur $N = 127$ selon différentes normes (le pic principal est bien à 0 dB).
A droite : niveau de PSLR moyen selon la norme choisie après différentes optimisations, $N = 127$. L'axe des abscisses varie de 1 à 30, avec en supplément des valeurs pour $p \in \{40, 60, 80, 100, 200, 500, 1000\}$, et la norme ℓ_∞ en dernier.

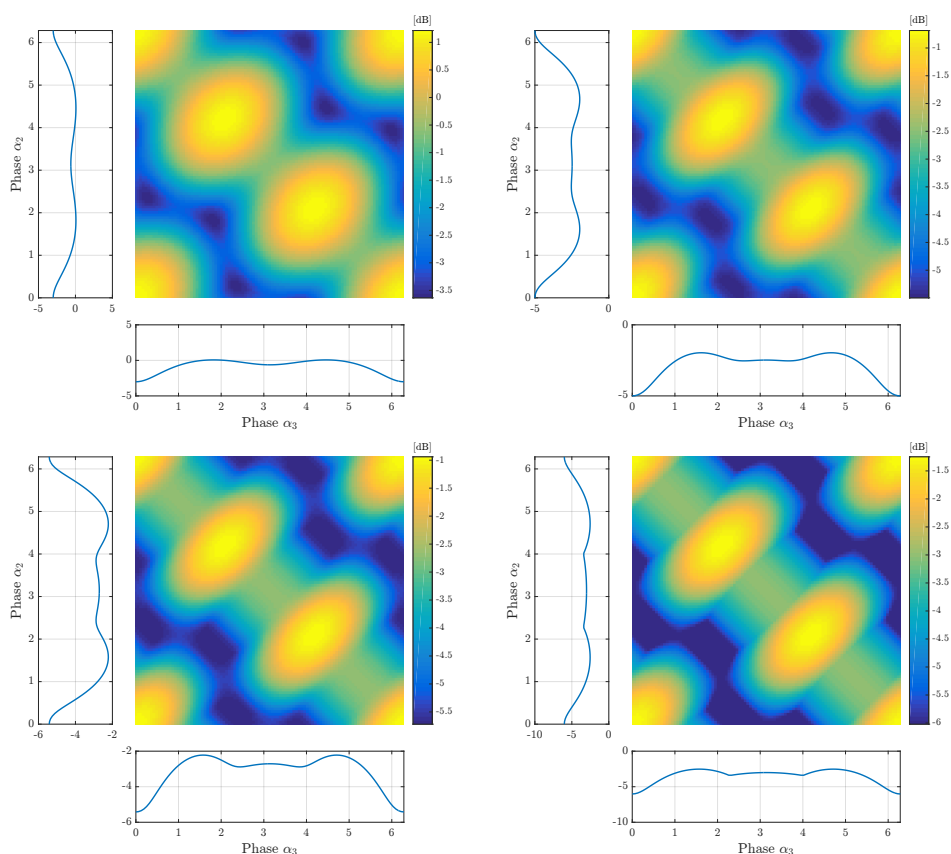


FIGURE 2.18 – Représentation de la fonction objectif pour $N = 4$
De haut en bas, de gauche à droite : norme $\ell_2, \ell_6, \ell_{10}, \ell_\infty$.
Étant donné qu'il est difficile de représenter une fonction en dimension 4, nous avons fixé la première et la dernière composante angulaire à 0.

2.2.2 Méthodes déterministes

Après s'être intéressé aux méthodes heuristiques, évoquons maintenant les méthodes dites déterministes.

2.2.2.1 Présentation

Commençons par une petite présentation des méthodes déterministes adaptées pour une optimisation sans contrainte. La majorité d'entre elles se fondent sur des principes de convexité. Malheureusement, toutes les fonctions ne sont pas convexes... Pour pallier à cela, il est concevable de supposer qu'elles le sont toutes, mais localement. Les méthodes déterministes présentent donc pour la plupart une convergence locale, contrairement aux méthodes heuristiques.

Il existe deux familles de méthodes déterministes : les méthodes de recherche linéaire, dont la plus connue est la descente de gradient, et les méthodes basées sur un modèle.

Méthodes basées sur un modèle Elles utilisent un... modèle de la fonction de coût. L'étape de minimisation est effectuée sur ce modèle. Ce dernier est ainsi choisi de sorte que ses propriétés permettent de détecter aisément son minimum. Citons, par exemple :

- Les algorithmes à régions de confiance qui construisent itérativement un modèle quadratique (donc convexe) local. Ce dernier nécessite le calcul des dérivées de la fonction de coût. En outre, il est possible d'établir une équivalence entre les algorithmes à régions de confiance et les méthodes de descente [Nocedal *et al.*, 2006]. Une illustration de cet algorithme est donnée Figure 2.19.
- Les méthodes comme le krigeage [Laurenceau, 2008] qui se basent sur une interpolation (locale ou globale) de la fonction objectif. Elles nécessitent donc un certain nombre d'évaluations afin de se rapprocher au mieux de la fonction coût initial.

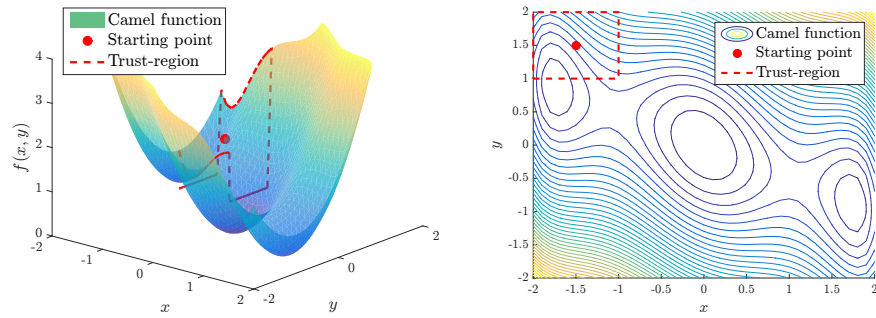
Les méthodes basées sur un modèle n'ont pas été considérées dans le cadre de cette thèse : la construction d'un modèle a semblé assez coûteux et peu efficace en grande dimension. A notre connaissance, il n'existe en effet aucune propriété efficace sur le minimum d'un polynôme multivarié. Si vous en connaissez une, souvenez-vous que la fonction de corrélation est un ensemble de polynômes...

Méthodes de recherche linéaire Les méthodes de recherche linéaire ont également une nature itérative [Bonnans *et al.*, 2013, Boyd *et al.*, 2004]. Elles recherchent le minimum de la fonction objectif le long d'une droite, ce qui revient donc à l'optimisation d'une fonction à une seule dimension. Cette droite est généralement caractérisée par le/les dérivées de la fonction coût. Citons par exemple [Baden *et al.*, 2018] dans lequel une descente de gradient est employée pour optimiser des séquences binaires et/ou polyphases. Plus de détails seront donnés par la suite, car cette méthode fera l'objet d'une section à part entière. Une illustration est également proposée à la Figure 2.20.

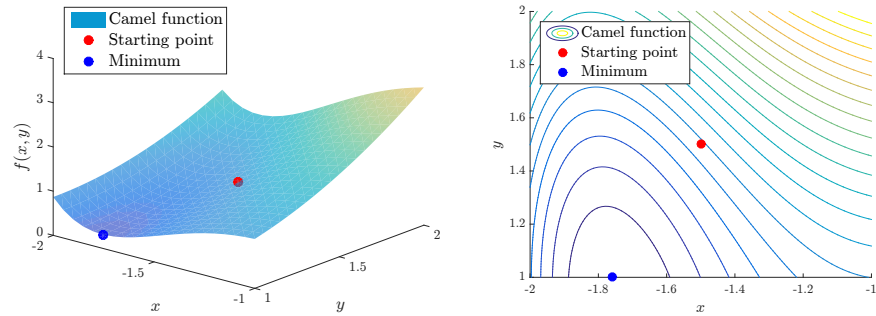
Exemple La fonction *three-hump camel back*. Cette fonction, dont le nom pourrait être traduit par « dos de chameau à trois bosses » est une fonction habituellement utilisée dans les *benchmarks* sur les méthodes d'optimisation. Celle-ci est définie par :

$$\begin{aligned}
 &[-2, 2]^2 \rightarrow \mathbb{R} \\
 &(a, b) \mapsto f(a, b) := 2a^2 - 1.05a^4 + \frac{1}{6}a^6 + ab + b^2,
 \end{aligned}
 \tag{2.23}$$

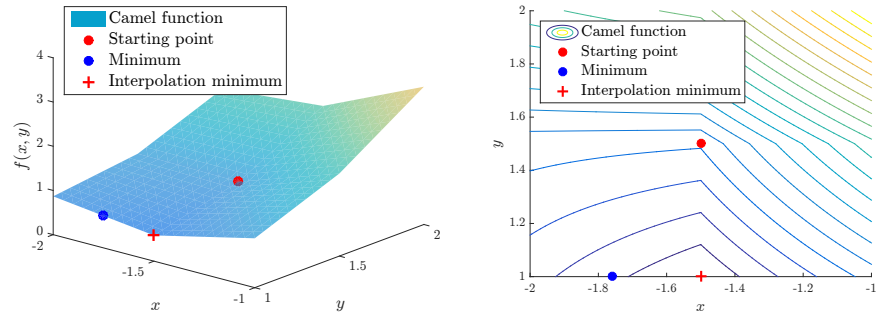
et admet un minimum global en $(0, 0)$. Utilisons cette fonction pour illustrer la méthode à région de confiance (Figure 2.19) et la méthode de descente de gradient (Figure 2.20).



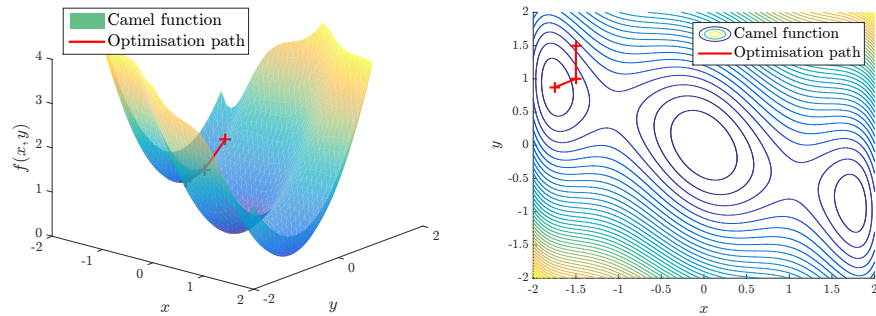
A partir d'une initialisation, une région de confiance locale est définie.



Une représentation de la région de confiance sélectionnée.

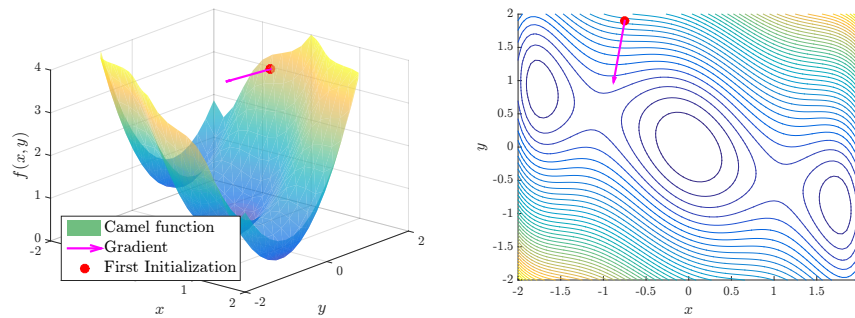


Une interpolation de la fonction dans cette région est effectuée. Le minimum de celle-ci est déterminé. Notez que le minimum de l'interpolant n'est pas le même que celui de l'interpolé...

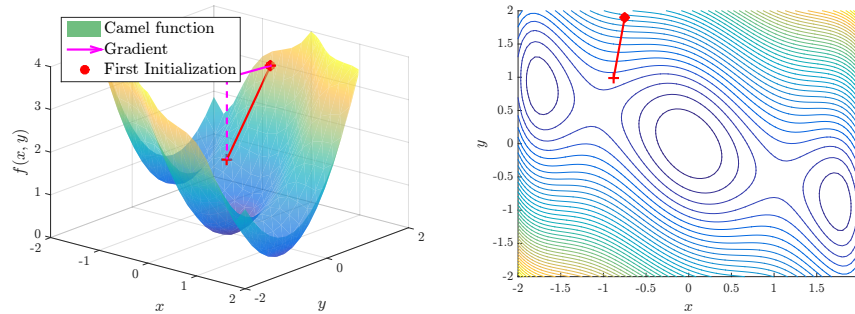


Ce minimum est choisi comme nouveau point de départ. Le processus est répété jusqu'à convergence. Une convergence locale, ici.

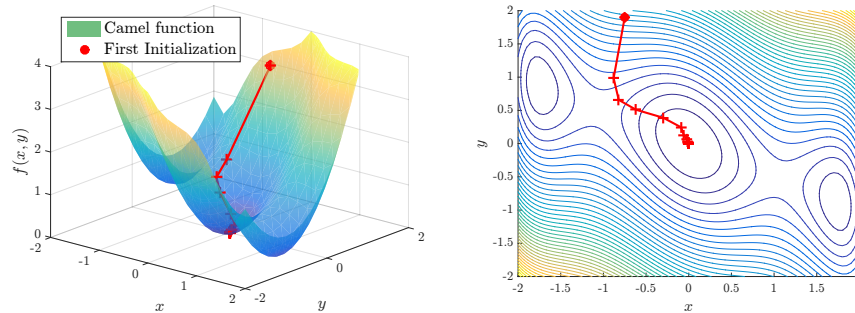
FIGURE 2.19 – Illustration de l'algorithme à régions de confiance
A gauche : représentation sous forme d'une surface. A droite : représentation des contours de la fonction.



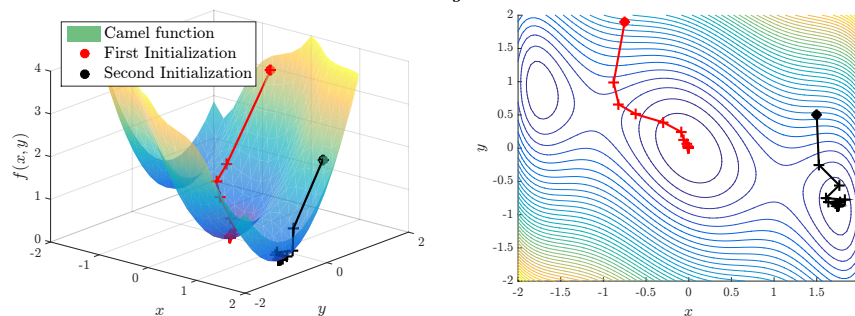
Le calcul du gradient à partir d'un point (ici, l'initialisation) donne une direction de descente.



Le minimum sur cette droite (en théorie, celui-ci doit être au voisinage du point précédent, ce qui n'est pas forcément le cas en pratique) est choisi comme nouveau point de départ.



Le procédé est répété jusqu'à convergence. Une descente de gradient peut ainsi converger vers le minimum global...



... ou bien vers le minimum local, selon le choix de l'initialisation.

FIGURE 2.20 – Illustration de l'algorithme de descente de gradient
A gauche : représentation sous forme d'une surface. A droite : représentation des contours de la fonction.

Pénalisation Les problèmes d'optimisation (P_1) et (P_2) sont équivalents d'un point de vue résultats. Mais, d'un point de vue optimisation, l'un est un problème d'optimisation avec contraintes, l'autre sans. De ce fait, l'application des méthodes sus-citées pour un problème d'optimisation avec contraintes nécessite une petite transformation, que nous présentons ici. Plus de détails sont donnés dans [Boyd *et al.*, 2004, Nocedal *et al.*, 2006], avec notamment quelques preuves sur les conditions d'optimalité.

Un problème d'optimisation avec contraintes se définit par :

$$(P) \begin{cases} \min_{\mathbf{x}} & f(\mathbf{x}), \\ \text{t.q.} & c_i(\mathbf{x}) = 0, \quad i \in \mathcal{I}, \\ & c_i(\mathbf{x}) \leq 0, \quad i \in \mathcal{E}, \end{cases} \quad (2.24)$$

où la première ligne de contrainte représente les contraintes d'égalité, et la seconde les contraintes de... vous pouvez le déduire. Pour plus de simplicité, rassemblons-les sous la forme d'un vecteur :

$$\mathbf{c}(\mathbf{x}) = [c_1(\mathbf{x}), \dots, c_{|\mathcal{E}|+|\mathcal{I}|}(\mathbf{x})]^T. \quad (2.25)$$

Le Lagrangien \mathcal{L} associé au problème (P) est :

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{x}), \quad (2.26)$$

où $\boldsymbol{\lambda} \in \mathbb{R}^{|\mathcal{E}|+|\mathcal{I}|}$ est connu sous le nom de multiplicateur de Lagrange. Les points stationnaires du Lagrangien sont également ceux de la fonction de coût initiale f (en ignorant la variable $\boldsymbol{\lambda}$, bien entendu). Pour les trouver, deux possibilités s'offrent à nous. La première est de rechercher directement le minimum du Lagrangien, dans la mesure où le problème n'a plus de contraintes. La seconde est de considérer le problème dual de (P) . Pour cela, il nous faut tout d'abord définir le dual de la fonction objectif, qui est également une fonction, mais seulement selon le paramètre $\boldsymbol{\lambda}$:

$$q(\boldsymbol{\lambda}) := \inf_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}). \quad (2.27)$$

Ce dernier nous permet d'écrire le problème d'optimisation dual (P') :

$$(P') \begin{cases} \max_{\boldsymbol{\lambda}} & q(\boldsymbol{\lambda}), \\ \text{t.q.} & \lambda_i \geq 0, \quad i \in \llbracket 1, |\mathcal{E}| + |\mathcal{I}| \rrbracket. \end{cases} \quad (2.28)$$

La résolution du problème dual (P') nous dévoile le paramètre $\boldsymbol{\lambda}$ optimal et potentiellement, par suite, l'optimum de la fonction f . Accessoirement, une borne sur la valeur prise par l'optimum de la fonction f peut également être déterminée.

Remarque 2.8. (*Méthode des points intérieurs*) Le Lagrangien (2.26) prend en compte les contraintes de manière linéaire. Il est également possible d'« ajouter » une fonction barrière, qui permet, en quelque sorte, de pondérer cette contrainte [Wright, 2005]. Par exemple, la fonction barrière logarithmique s'écrit :

$$B(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \boldsymbol{\lambda}^T \ln(\mathbf{c}). \quad (2.29)$$

Cette méthodologie n'a pas été considérée dans ce manuscrit : il fallait déjà essayer de montrer l'efficacité des méthodes déterministes sans contraintes avant...

Avantages et inconvénients Comme pour les méthodes heuristiques, concluons ce paragraphe avec quelques considérations.

- De nombreux algorithmes déterministes se basent sur le gradient de la fonction coût : on parle d'ailleurs d'*optimisation différentielle*. Des hypothèses de régularité sont ainsi nécessaires (l'existence de dérivées partielles est le minimum requis). Si connaissance il y a, la précision de l'algorithme ne peut en être que meilleure.
- Les algorithmes déterministes garantissent une convergence vers un minimum local ou, au pire, vers un point selle. Cette convergence est généralement en pratique très rapide, si le calcul des dérivées n'est pas prohibitif.

2.2.2.2 La descente de gradient

Un algorithme de descente Le principe d'un algorithme de descente est de rechercher de proche en proche des points qui permettent de réduire la valeur prise par la fonction objectif f . En d'autres termes, à partir d'un point \mathbf{x} , on souhaite trouver un point $\mathbf{y} := \mathbf{x} + \gamma\Delta\mathbf{x}$, $\gamma > 0$, tel que $f(\mathbf{y}) \leq f(\mathbf{x})$. Si cela était possible, on obtiendrait l'algorithme suivant [Boyd *et al.*, 2004] :

Algorithme Descente générique	
Données	Initialisation \mathbf{x}
Répéter	1. Détermination d'une direction de descente $\Delta\mathbf{x}$ 2. Choix du pas $\gamma > 0$ 3. Mise à jour : $\mathbf{x} = \mathbf{x} + \gamma\Delta\mathbf{x}$
Jusqu'à	Un critère d'arrêt est atteint.

TABLE 2.3 – Algorithme de descente générique

Il ne reste donc qu'à définir la direction de descente $\Delta\mathbf{x}$. Souvenez-vous, la majorité des algorithmes déterministes se base sur l'hypothèse que la fonction objectif est localement convexe. Les algorithmes de descente ne font pas exception. Pour tout \mathbf{y} suffisamment proche de \mathbf{x} , un développement limité nous donne :

$$f(\mathbf{y}) = f(\mathbf{x} + \gamma\Delta\mathbf{x}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{x} + \gamma\Delta\mathbf{x} - \mathbf{x}). \quad (2.30)$$

Pour que $f(\mathbf{y}) \leq f(\mathbf{x})$, il faut donc nécessairement que $\gamma\nabla f(\mathbf{x})^T\Delta\mathbf{x} \leq 0$. Ainsi, un choix naturel d'une direction de descente est de considérer l'opposé du gradient :

$$\Delta\mathbf{x} = -\nabla f(\mathbf{x}). \quad (2.31)$$

Cette direction est l'essence de la descente de gradient.

Alternatives Mais revenons à l'algorithme en lui-même, rappelé à la Table 2.3. Celui-ci est composé de plusieurs étapes, dont nous donnons ici quelques détails et présentons d'autres alternatives.

0. L'initialisation sera par la suite choisie de manière aléatoire. Ce n'est pas forcément un choix judicieux, dans la mesure où cela influence grandement le résultat final...
1. Le choix du gradient en tant que direction de descente est le plus naturel, mais pas forcément le plus efficace en terme de convergence [Boyd *et al.*, 2004]. Si la fonction objectif le permet, nous pouvons également considérer la direction de Newton :

$$\Delta\mathbf{x} = -(\nabla^2 f(\mathbf{x}))^{-1} \nabla f(\mathbf{x}), \quad (2.32)$$

mais cela signifie que la fonction objectif est au moins deux fois dérivables. Le calcul du Hessien peut cependant être relativement coûteux. Il existe donc des alternatives qui approximent cette dérivée seconde, appelées « méthodes de Quasi-Newton » ; la plus connue est appelée BFGS, d'après les initiales de ses inventeurs (Broyden, Fletcher, Goldfarb et Shannon) et est détaillée dans [Nocedal *et al.*, 2006].

2. Le choix du pas $\gamma > 0$ est également important, puisqu'il influe sur la convergence théorique (avec un pas trop important, on peut passer d'un minimum à un autre ; l'algorithme perdra ainsi sa propriété locale) et pratique (avec un pas trop petit, l'algorithme peut prendre beaucoup de temps à converger). Le pas optimal à chaque itération est en fait solution d'un problème de minimisation à une dimension [Boyd *et al.*, 2004] :

$$\gamma = \operatorname{argmin}_{\gamma' > 0} f(\mathbf{x} + \gamma'\Delta\mathbf{x}). \quad (2.33)$$

Une approximation de cette solution peut être déterminée à l'aide des règles d'Armijo ou de Wolfe [Bonnans *et al.*, 2013]. Alternativement, la notion de voisinage peut être exprimée à l'aide d'un *backtracking*. Pour cela, γ est réduit petit à petit, jusqu'à ce que la condition $f(\mathbf{x} + \gamma\Delta\mathbf{x}) \leq f(\mathbf{x})$ soit atteinte.

- ?. L'algorithme s'arrête quand un point selle ou un minimum local est atteint. Deux conditions sont communément utilisées pour décrire ce phénomène : la faible valeur de la norme du gradient au point courant, ou un pas γ qui n'est viable qu'en étant quasi nul.

Remarque 2.9. (*Et, dans le cas complexe, ça fonctionne ?*) Si nous prenons l'écriture (P_1) de notre problème d'optimisation, la fonction de coût f_p est de variables complexes (les éléments du code de phase) et à valeurs réelles. En l'occurrence, pour appliquer une descente de gradient, il suffit que f_p admette l'existence de dérivées partielles, ce qui est heureusement le cas. La direction de descente est ainsi donnée par [Li *et al.*, 2008] :

$$\Delta\mathbf{s} = -2\gamma \frac{\partial f_p}{\partial \mathbf{s}^*}. \quad (2.34)$$

Calcul de gradient De ce fait, afin d'appliquer une descente de gradient sur la fonction objectif f_p , nous nous devons de calculer la différentielle de cette dernière. La norme ℓ_p , et donc la fonction f_p *a fortiori*, n'est pas une fonction holomorphe, car elle ne satisfait pas les conditions de Cauchy-Riemann [Kreutz-Delgado, 2009]. Elle est cependant \mathbb{R} -différentiable pour tout $\mathbf{z} \in \mathbb{C}^N \setminus \{\mathbf{0}\}$, ce qui est heureusement suffisant pour appliquer une descente de gradient [Brandwood, 1983, van den Bos, 1994]. Les dérivées formelles de ℓ_p — communément appelées « dérivées de Wirtinger » — sont définies par :

$$\frac{\partial}{\partial \mathbf{z}} \|\mathbf{z}\|_p = J_{\|\cdot\|_p}(\mathbf{z}) = \left(\frac{\mathbf{z} \circ |\mathbf{z}|^{p-2}}{\|\mathbf{z}\|_p^{p-1}} \right)^H \quad \text{et} \quad \frac{\partial}{\partial \bar{\mathbf{z}}} \|\mathbf{z}\|_p = \left(\frac{\partial}{\partial \mathbf{z}} \|\mathbf{z}\|_p \right). \quad (2.35)$$

En supposant que les variables \mathbf{s} et \mathbf{s}^* sont indépendantes — bien que ce ne soit pas forcément nécessaire [Ten Cate, 2015] —, nous pouvons aisément écrire à partir de (2.4) :

$$\frac{\partial(\text{AACF}(\mathbf{s}))_k}{\partial s_n} = \bar{s}_{k+n-N}, \quad \frac{\partial(\text{AACF}(\mathbf{s}))_k}{\partial \bar{s}_n} = s_{-k+n+N}, \quad (2.36)$$

ce qui nous donne les matrices jacobiniennes suivantes :

$$J_{\text{AACF}(\mathbf{s})}(\mathbf{s}) = \begin{bmatrix} 0 & \cdots & 0 & \bar{s}_1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ddots & & \vdots \\ \bar{s}_1 & \cdots & \cdots & \bar{s}_N \\ \vdots & & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \bar{s}_N & 0 & \cdots & 0 \end{bmatrix} = \mathbf{E}_{2N-1} \bar{\Lambda}_N(\mathbf{s}), \quad (2.37)$$

$$\bar{J}_{\text{AACF}(\mathbf{s})}(\mathbf{s}) = \Lambda_N(\mathbf{s}),$$

où \mathbf{E}_{2N-1} est la matrice dite d'échange de taille $2N - 1$. Celle-ci est définie comme une matrice identité dont les lignes (ou les colonnes) sont inversées, permettant une inversion des lignes par une multiplication à gauche, ou une inversion des colonnes si elle est à droite.

Ensuite, le théorème des fonctions composées [Hjørungnes, 2011] permet de déterminer la matrice jacobienne de f_p en fonction de \mathbf{s} :

$$\begin{aligned}
J_{f_p}(\mathbf{s}) &= J_{\|\cdot\|_p}(\mathbf{F} \text{AACF}(\mathbf{s})) \mathbf{F} J_{\text{AACF}(\mathbf{s})}(\mathbf{s}) + J_{\|\cdot\|_p}(\mathbf{F} \text{AACF}(\mathbf{s}))^* \mathbf{F} J_{\text{AACF}(\mathbf{s})}(\mathbf{s}) \\
&= J_{\|\cdot\|_p}(\mathbf{F} \text{AACF}(\mathbf{s})) \mathbf{F} J_{\text{AACF}(\mathbf{s})}(\mathbf{s}) + (J_{\|\cdot\|_p}(\mathbf{F} \text{AACF}(\mathbf{s})))^* \mathbf{F} (\bar{J}_{\text{AACF}(\mathbf{s})}(\mathbf{s}))^* \\
&= J_{\|\cdot\|_p}(\mathbf{F} \text{AACF}(\mathbf{s})) \mathbf{F} \mathbf{J}_{2N-1} \bar{\Lambda}_N + (J_{\|\cdot\|_p}(\mathbf{F} \text{AACF}(\mathbf{s})))^* \mathbf{F} \bar{\Lambda}_N \\
&= 2 \left\{ (J_{\|\cdot\|_p}(\mathbf{F} \text{AACF}(\mathbf{s})))^* \mathbf{F} \bar{\Lambda}_N \right\}.
\end{aligned} \tag{2.38}$$

La dernière égalité s'obtient à l'aide d'une propriété de la séquence d'autocorrélation :

$$\begin{aligned}
\text{AACF}(\mathbf{s})^* &= \mathbf{E}_{2N-1} \text{AACF}(\mathbf{s}) \\
\implies (J_{\|\cdot\|_p}(\mathbf{F} \text{AACF}(\mathbf{s})))^* &= J_{\|\cdot\|_p}(\mathbf{F} \text{AACF}(\mathbf{s})) \mathbf{E}_{2N-1}.
\end{aligned} \tag{2.39}$$

Du fait de la contrainte de module constant, la séquence \mathbf{s} peut se définir uniquement à l'aide de son vecteur de phase φ . En remarquant que $J_{f_p}(\bar{\mathbf{s}}) = (J_{f_p}(\mathbf{s}))^*$, nous pouvons conclure ce calcul à l'aide d'un petit changement de variable :

$$J_{g_p}(\varphi) = -2\Im (J_{f_p}(\mathbf{s}) \text{diag}(\mathbf{s})). \tag{2.40}$$

Ce résultat correspond peu ou prou à l'expression obtenue dans [Baden *et al.*, 2015] ou dans [Tan *et al.*, 2016b], où le calcul est effectué avec les dérivées partielles selon les parties réelles et imaginaires de \mathbf{s} .

Remarque 2.10. (*Séquences binaires*) Si \mathbf{s} est une séquence binaire (composée uniquement de -1 et de 1), on peut remarquer que les termes utilisés pour le calcul précédent sont tous à valeurs réelles. En particulier, cette observation inclut la jacobienne J_{g_p} , qui est même nulle : ainsi, toute séquence binaire est un point stationnaire de notre problème...

2.2.2.3 Résultats applicatifs

Après ces petits calculs, nous pouvons utiliser une descente de gradient sur notre problème d'optimisation. Plus précisément, les résultats suivants ont été obtenus à l'aide d'un algorithme comprenant une étape de *backtracking*, permettant de déterminer le pas à chaque itération. Cet algorithme est résumé plus loin, dans le Tableau 2.4.

Performances Commençons par une optimisation de deux séquences, respectivement de longueur $N = 63$ et $N = 1023$. Les initialisations sont aléatoires et identiques à celles tirées à la Section 2.2.1.3. Nous présentons les résultats obtenus aux Figures 2.21 et 2.22. Malgré leur caractère local, les méthodes basées sur le gradient apportent un bénéfice — inattendu, il nous faut l'avouer. Pour la première séquence, les méthodologies proposées se tiennent dans un mouchoir de poche, avec un petit avantage pour l'algorithme CMA-ES. L'apport des dérivées secondes ne semble pas flagrant ici, avec une perte de l'ordre de 0.5 dB. Ces observations restent valables avec des séquences plus longues (ici, $N = 1023$, et voir par ailleurs la Figure 2.23).

Choix de la norme La légende de la Figure 2.21 introduit le paramètre p dans le processus d'optimisation. A l'instar de l'algorithme CMA-ES, le choix de la norme a également une influence sur les résultats obtenus, bien que l'on recherche ceans des séquences proposant le meilleur *Peak-to-Sidelobe Level*. En outre, il est plus difficile de manœuvrer le gradient de la norme infinie... Les deux figures précédentes ont été présentées avec une sélection optimale de la norme (celle-ci est d'ailleurs détaillée en Figure 2.24, gauche). Mais cette norme doit-elle être identique pour toutes les séquences ? Pour le savoir, on se propose d'optimiser une centaine de séquences de différentes longueurs, et d'en étudier les résultats.

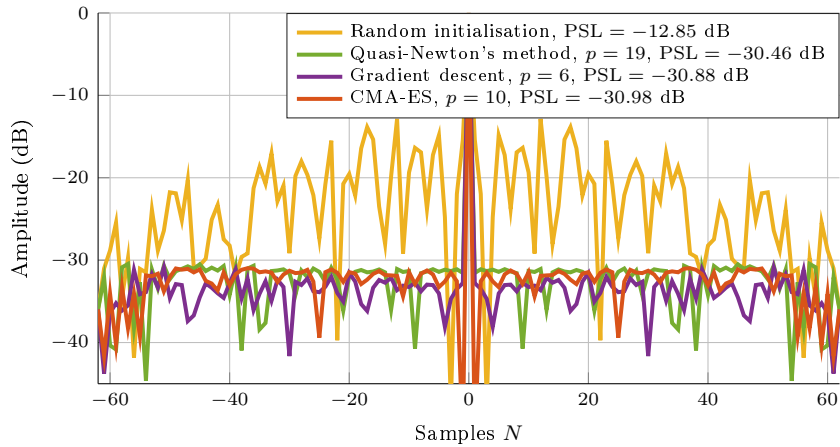
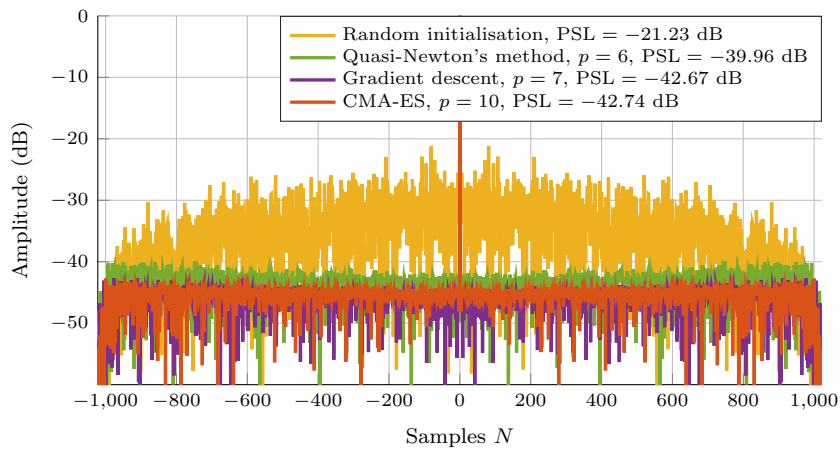
FIGURE 2.21 – Optimisation à l'aide de méthodes déterministes, $N = 63$ 

FIGURE 2.22 – Optimisation à l'aide de méthodes déterministes, $N = 1023$
 Les initialisations sont identiques à celles des Figures 2.12 et 2.13, respectivement.
 Les résultats de l'algorithme CMA-ES sont donnés à titre indicatif.
 Les normes ℓ_p ont été choisies de sorte à obtenir le meilleur PSL possible.

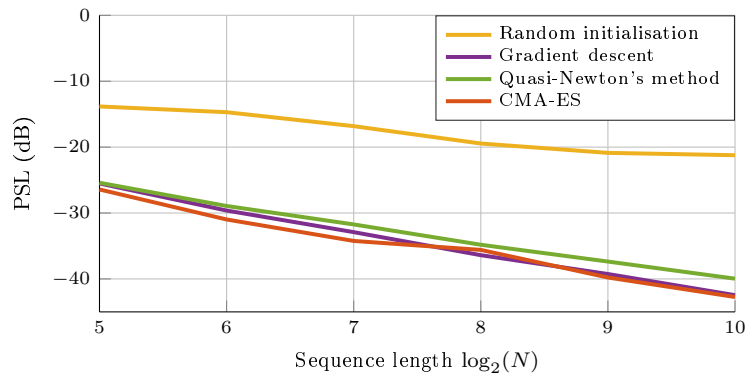


FIGURE 2.23 – Étude de convergence de quelques méthodes déterministes
Affichage du meilleur PSL obtenu après 100 instances, $p = 8$ fixé. Remarquez que l'on gagne environ 3 dB en doublant la longueur de la séquence.

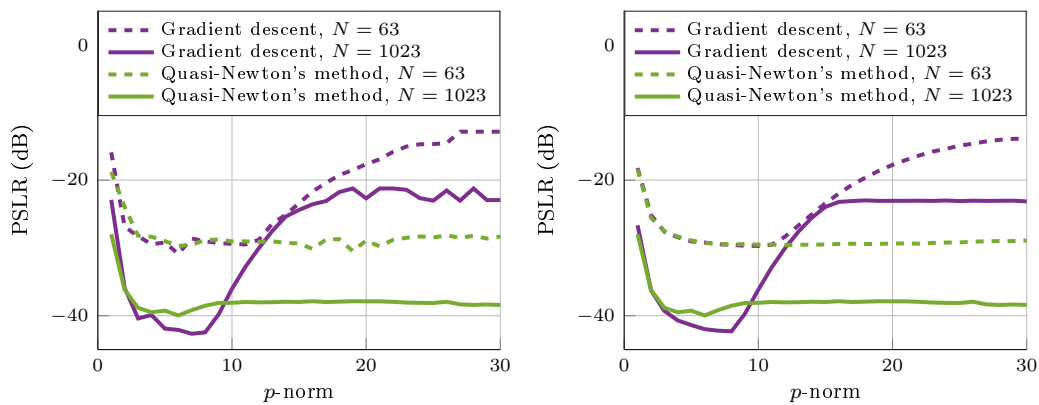


FIGURE 2.24 – Choix de la norme de la fonction coût pour deux méthodes déterministes
A gauche : PSL selon la norme p pour les Figures 2.21 et 2.22.
A droite : PSL moyen selon la norme p après 100 instances d'optimisation.

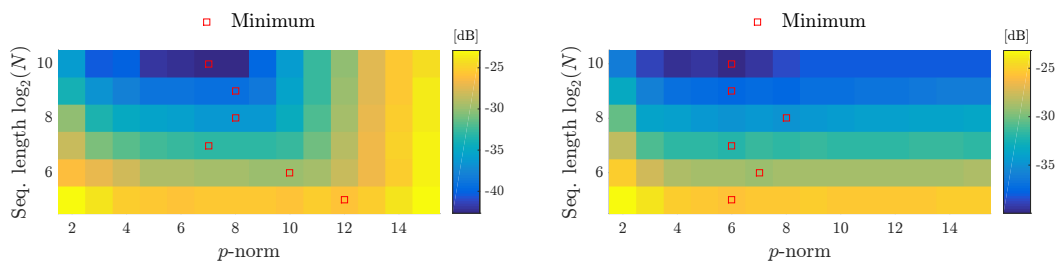


FIGURE 2.25 – Choix de la norme de la fonction coût pour deux méthodes déterministes (2)
A gauche : choix de la norme p pour la descente de gradient.
A droite : choix de la norme p pour une méthode de Quasi-Newton.
Les résultats ont été obtenus après 100 instances d'optimisation.

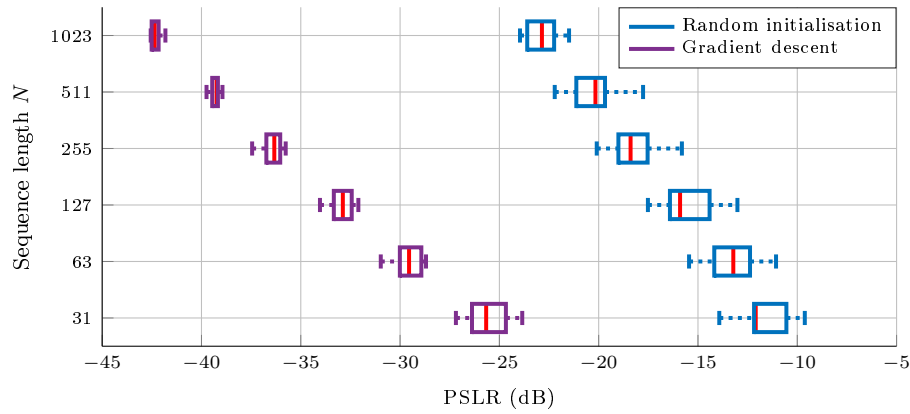


FIGURE 2.26 – Répartition des lobes secondaires après une descente de gradient, $p = 8$
 La boîte représente le deuxième et troisième quartile, tandis que la ligne rouge définit la valeur médiane.

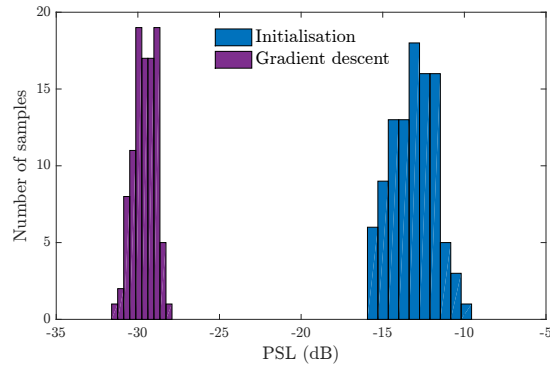


FIGURE 2.27 – Répartition des lobes secondaires après une descente de gradient, $p = 8$
 Zoom sur la répartition pour des séquences de longueur $N = 63$.

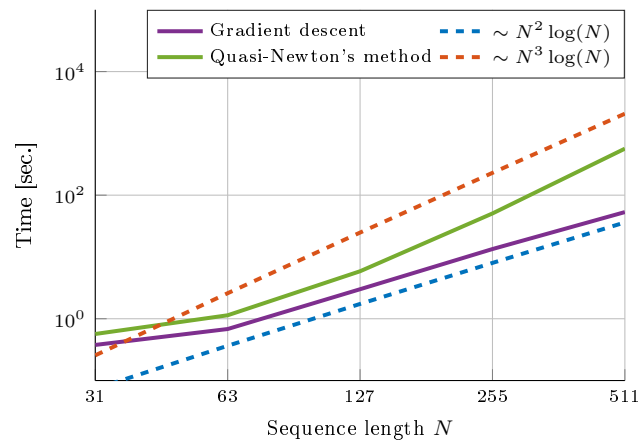


FIGURE 2.28 – Temps d'exécution des deux algorithmes déterministes
 Échelle logarithmique. Temps mesuré à l'aide d'un processeur E5-2643 @3.4 Ghz. Deux fonctions sont données à titre indicatif.

Algorithme Descente avec *backtracking*

Données	Initialisation aléatoire de la séquence de phases φ Initialisation du pas $\gamma = 1$
Répéter	<ol style="list-style-type: none"> 1. Détermination de la direction de descente Calcul de la jacobienne $J_{g_p}(\varphi)$ 2. Évaluation de candidats par <i>backtracking</i> Premier candidat : $\varphi' := \varphi - \gamma J_{g_p}(\varphi)$ Amélioration de la fonction coût g_p ? (Oui) Aller directement au point 3. (Non) <i>Backtracking</i> sur le pas Tant qu'il n'y a pas d'amélioration sur g_p : Réduction du pas : $\gamma = 0.9\gamma$ Détermination d'un nouveau candidat $\varphi' := \varphi - \gamma J_{g_p}(\varphi)$ 3. Mise à jour : $\varphi = \varphi'$
Jusqu'à	Un critère d'arrêt est atteint.

TABLE 2.4 – Algorithme de descente avec *backtracking*

L'algorithme ci-dessus explicite les paramètres initiaux utilisés lors des simulations : le pas initial $\gamma = 1$, le facteur de réduction de ce dernier est fixé à 0.9.

Par ailleurs, un critère d'arrêt supplémentaire concerne la valeur minimale du pas.

Que pouvons-nous en tirer ? La Figure 2.24 confirme nos impressions. Il semble bien y avoir une corrélation entre la définition de la fonction objectif et les lobes secondaires de la séquence résultante. Mais cette relation s'avère différente selon la méthode d'optimisation. Celle de la méthode de Quasi-Newton est relativement familière, et fait penser à la Figure 2.17 : une décroissance au départ, un plateau constant ensuite. La méthode de Quasi-Newton possède ainsi une certaine souplesse dans le choix de la norme. La descente de gradient « classique » propose quant à elle trois phases :

- une forte décroissance (ici, jusqu'à $p = 8$) qui nous offre en sa fin des séquences dont le PSL est plus intéressant que celui obtenu *in fine* par les méthodes de Quasi-Newton ;
- une remontée, suivie d'une phase constante. La constance de la courbe indique qu'aucune optimisation n'a en fait été effectuée. Une explication plausible se rapporte au mauvais conditionnement possible du problème, ce qui entraînerait que la direction donnée par le gradient est peu informative...

Afin de déterminer une norme convenable, nous menons la même étude mais pour différentes

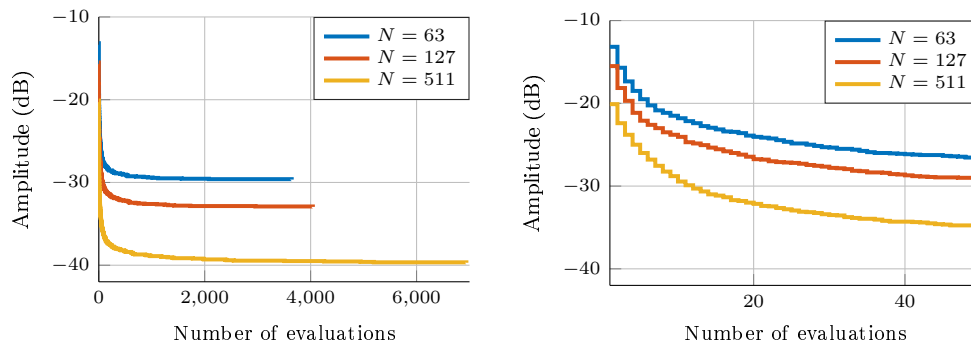


FIGURE 2.29 – Convergence de la descente de gradient en fonction du nombre d'évaluations

Moyenne après 100 tirages.

A droite : zoom sur les premières évaluations de la courbe de gauche.

longueurs (Figure 2.25). Il en vient qu'un prétendant crédible pour la descente de gradient est la norme ℓ_8 . Bien que $p = 6$ ait montré de meilleures performances, la norme ℓ_8 peut également convenir pour son compère de second ordre, du fait de la zone constante. En ce sens, les simulations qui suivent seront effectuées sous cette norme.

Convergence Après avoir choisi une norme, il nous est maintenant possible d'analyser la qualité des résultats obtenus. Curieusement, nous allons encore une fois nous répéter : « Malgré l'hétérogénéité [des initialisations], le résultat obtenu est relativement uniforme, avec des solutions qui tournent aux alentours de -30 dB ». Cette phrase a été prononcée pour l'algorithme CMA-ES dans le cas $N = 63$. Elle reste cependant pertinente, que ce soit dans ce cas précis ou pour d'autres longueurs de séquences — sauf à propos de la valeur annoncée de 30 dB, évidemment ! —, comme le montrent les Figures 2.26 et 2.27. En outre, une condensation du niveau des lobes secondaires est observée, relative à N .

Cette assertion a initialement permis d'établir une certaine constance dans les résultats obtenus. Cela étant, elle est beaucoup plus forte ici : en effet, une descente de gradient converge vers le point selle le plus proche. Pourrait-on annoncer que chaque point selle a un niveau semblable ? Cela est difficile à affirmer, étant donné la dimension du problème. Cependant, toutes ces observations vont en ce sens...

Remarque 2.11. (*Complexité*)

1. La jacobienne de la fonction objectif (2.40) s'écrit selon la fonction d'autocorrélation elle-même, et avec quelques produits matriciels. La fonction d'autocorrélation pouvant s'écrire à l'aide de FFT (Fast Fourier Transform, ou transformée de Fourier rapide) [Baden et al., 2018], il est possible de calculer chaque direction de descente avec une complexité en $\mathcal{O}(N \log N)$. En n'oubliant pas le côté itératif, la complexité de l'algorithme peut se déduire à l'aide de la Figure 2.28.
2. Il est intéressant de noter que l'application d'une descente de gradient pour une séquence de longueur $N = 1023$ prend habituellement quelques minutes (cf. Figure 2.28), contre quelques heures pour une réalisation de l'algorithme CMA-ES. Et cela pour un niveau de performances similaire.
3. D'ailleurs, la vitesse de convergence de ces algorithmes aurait pu être améliorée en pratique. Du côté de la descente de gradient, les dernières évaluations sont nombreuses pour une maigre amélioration, comme le montre la Figure 2.29. L'utilisation d'une règle de Wolfe [Bonnans et al., 2013] aurait sûrement été plus judicieuse qu'un backtracking. Du côté de la méthode de Quasi-Newton, un calcul analytique de la hessienne permettrait d'enlever l'étape de son approximation.

2.2.2.4 Sortir du point selle

Comme nous l'avons vu dans une section précédente, une condition d'arrêt majeure porte sur la norme du vecteur gradient au point courant. L'algorithme s'arrête en effet lorsque la séquence courante \mathbf{s} est un point stationnaire, *i.e.* $J_{f_p}(\mathbf{s}) = \mathbf{0}$. Cependant, être un point stationnaire ne signifie en rien que celui-ci est un minimum local ; d'ailleurs, en grande dimension, il est fort à parier que ce ne soit pas le cas. Cela garantit seulement le titre de « point selle » (une illustration d'un point selle est donnée Figure 2.8).

En fixant certaines dimensions, un point selle est un minimum local ; en en fixant d'autres, c'est un maximum local. Cela signifie qu'il existe des points présentant une fonction objectif plus faible en son voisinage. De ce fait, nous voulons exploiter cette particularité pour obtenir de meilleurs points, tout en conservant la rapidité d'exécution originelle. Pour cela, plusieurs méthodologies sont proposées dans ce paragraphe.

Algorithme Exploration naïve	
Données	Initialisation \mathbf{x}
Répéter	1. Descente de gradient, jusque convergence Mise à jour du point \mathbf{x} 2. Classification de \mathbf{x} : point selle, minimum local ? 3. Si point selle : exploration mono-dimensionnelle Sélection du meilleur candidat, appelé \mathbf{x}
Jusqu'à	Un critère d'arrêt est atteint.

TABLE 2.5 – Algorithme d'exploration naïve

Algorithme Marche aléatoire (Metropolis-Hastings)	
Données	Initialisation \mathbf{x} Fonction d'acceptation P
Répéter	1. Génération d'un candidat \mathbf{x}' 2. Évaluation par la fonction d'acceptation $P(\mathbf{x}' \mathbf{x})$ 3. Génération d'un nombre aléatoire $u \in [0, 1]$ 4. Si $P(\mathbf{x}' \mathbf{x}) > u$, adoption du point \mathbf{x}'
Jusqu'à	Un critère d'arrêt est atteint.

TABLE 2.6 – Algorithme simplifié d'une marche aléatoire
Notez qu'un point est toujours conservé s'il améliore la fonction objectif.

Intuitivement, l'utilisation des méthodes de second ordre semblait pertinente, dans la mesure où leur convergence est de meilleure qualité [Boyd *et al.*, 2004]. En outre, souvenez-vous qu'un minimum local est caractérisé par une matrice hessienne semi-définie positive. Les simulations précédentes ne corroborent malheureusement pas ce pressentiment.

Une deuxième idée consiste à perturber légèrement la direction de descente en ajoutant une petite quantité — appelée *bruit* — au vecteur gradient. En d'autres termes, cette direction est donnée par :

$$\Delta \mathbf{s} = -\nabla f_p(\mathbf{s}) + \epsilon. \quad (2.41)$$

ϵ dénote le bruit, que nous devons déterminer empiriquement. Cette idée s'appuie sur le fait qu'une descente classique converge vers un point selle ; en modifiant légèrement le chemin, il est possible de l'éviter, permettant ainsi de continuer la chute... Une preuve de convergence (mais toujours vers un minimum local) a été suggérée dans [Ge *et al.*, 2015].

Comme indiqué ci-dessus, un point selle est un maximum local sous certaines dimensions. Ainsi, après convergence d'un algorithme de descente, une idée naïve consiste à explorer au voisinage chacune des dimensions, séparément. Lors de l'examen d'une dimension suggérant le caractère maximal de notre candidat, il existera nécessairement un autre point améliorant la fonction coût. En outre, si le changement apporté est suffisamment avantageux, une descente de gradient peut à nouveau être effectuée. Cela se résume sous le processus itératif suivant de la Table 2.5. La classification du candidat peut s'effectuer de manière grossière à l'aide de la norme du gradient (en ce point), ou de manière plus rigoureuse avec les valeurs propres de la hessienne. Le critère d'arrêt le plus courant consiste à mesurer l'amélioration de la fonction coût *post*-exploration.

Un algorithme hybride Si l'on résume en quelques mots l'ensemble des précédents résultats, nous pouvons dire que les méthodes de descente proposent de bonnes séquences, tandis que les méthodes stochastiques *peuvent* légèrement faire mieux, mais avec un coût calculatoire beaucoup plus élevé. Que se passe-t-il si nous essayons de coupler les deux ?

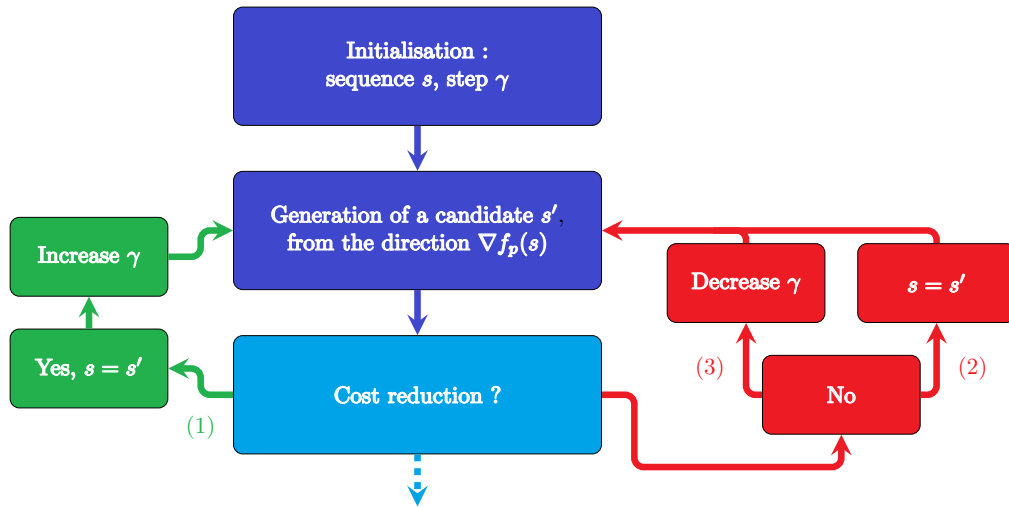


FIGURE 2.30 – Marche aléatoire basée sur le gradient

Cela donne une méthode hybride, et cela a déjà été employé dans [Milewski *et al.*, 2006] par exemple. La méthode que nous proposons ici est basée sur la marche aléatoire, proposée par Metropolis *et al.* en 1953 [Metropolis *et al.*, 1953, Robert, 2015]. Le principe de la marche aléatoire — également qualifiée de « marche de l’ivrogne » — est assez explicite. L’exploration de l’espace des solutions se fait de manière aléatoire. Cependant, pour une optimisation, on l’enrichit d’une petite subtilité : chaque candidat visité ne sera pas nécessairement une étape dans le « trajet » final, dans la mesure où cela dépend d’une fonction d’*acceptation*. Cette dernière se définit par son intitulé, et dépend généralement de la valeur de la fonction objectif à l’itération précédente. Le Tableau 2.6 résume tout cela.

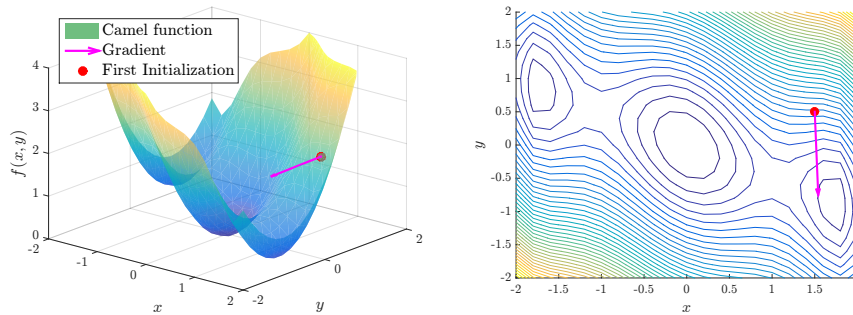
Remarque 2.12. (*Recherche aléatoire*) Il est possible d’instaurer une certaine loi pour la génération d’un candidat. Ainsi, avec les bons paramètres — une loi d’acceptation qui n’accepte que les progrès par exemple — il est possible de retrouver la recherche aléatoire locale (Tableau 2.2).

Ainsi, on propose ici d’effectuer une marche pseudo-aléatoire, sur la droite définie par le gradient. De ce fait, l’exploration se fait sur un espace beaucoup plus réduit (après chaque acceptation, cet espace est modifié) mais dont on est certain qu’il y a des points réduisant la fonction coût. Nous ajoutons également notre grain de sel : la valeur du pas γ est liée à l’étape d’acceptation de l’itération précédente.

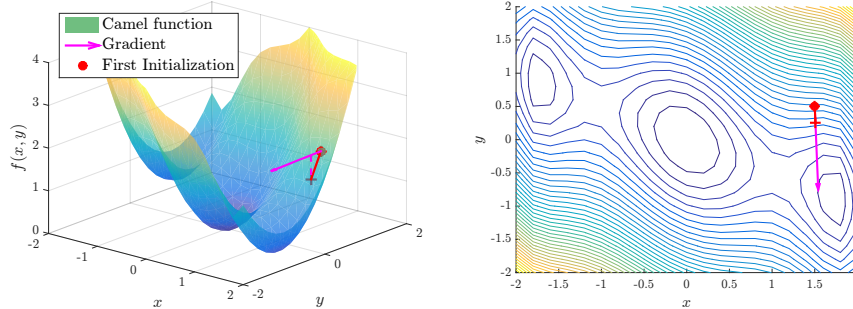
1. γ est augmenté, toujours de manière aléatoire, si le candidat apporte une amélioration. Cela permet d’explorer de nouveaux horizons...
2. γ est laissé tel quel si le candidat est conservé, malgré une dégradation de la fonction objectif. Bien entendu, la probabilité de tomber dans ce cas de figure dépend de la dégradation...
3. Sinon, γ est réduit — toujours d’un facteur aléatoire — ce qui nous permet d’augmenter nos chances d’obtenir un meilleur point (cela revient en quelque sorte à une itération de *backtracking*).

L’algorithme obtenu est schématisé Figure 2.30 et illustré Figure 2.31.

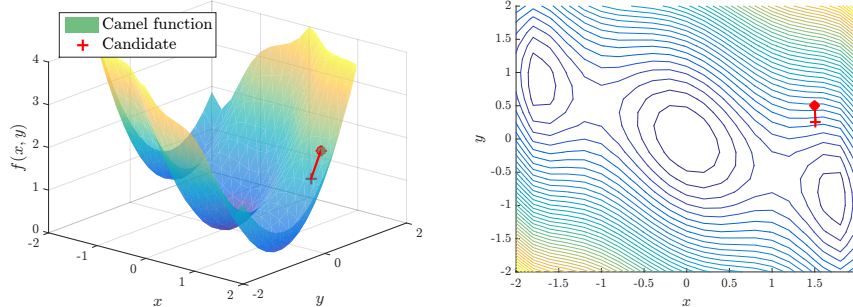
Remarque 2.13. (*Une méthode déterministe ? Euh...*) Bien que la méthodologie proposée se base sur une direction de descente, une marche aléatoire est bel et bien une méthode stochastique.



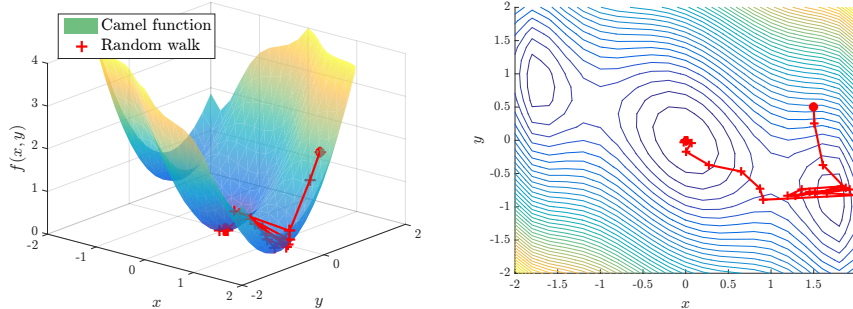
Le calcul du gradient à partir d'un point (ici, l'initialisation) donne une direction de descente.



Un point, choisi aléatoirement sur cette droite, est évalué.



Ici, il apporte une amélioration de la fonction coût ; il est donc sélectionné à coup sûr.



Le procédé est répété jusqu'à convergence. Nous avons réussi à nous échapper du minimum local, dans cet exemple.

FIGURE 2.31 – Illustration d'une marche aléatoire basée sur le gradient
A gauche : représentation sous forme d'une surface. A droite : représentation des contours de la fonction.

	$N = 63$	$N = 1023$
Initialisation	-12.85 dB	-21.23 dB
CMA-ES	-30.98 dB	-42.74 dB
Gradient descent	-30.88 dB	-42.66 dB
Quasi-Newton's method	-30.46 dB	-39.96 dB
Noisy gradient	-30.41 dB	-42.17 dB
Saddle exploration	-31.16 dB	-43.03 dB
Random Walk	-31.24 dB	-42.69 dB

TABLE 2.7 – Comparaison des différentes échappatoires

Résultats Terminons cette digression par une petite application numérique, Table 2.7. Nous comparons ainsi ces différents algorithmes sur les deux exemples qui nous suivent en cette section. Il nous faut reconnaître que nous n'avons pas trouvé une loi fiable sur ϵ qui nous permettrait de soutenir le gradient bruité pour notre problème. Les séquences obtenues restent cependant acceptables, au niveau des méthodes de Quasi-Newton. Sinon, l'exploration dimensionnelle ou la marche aléatoire permettent de grappiller des poussières de dB, dépassant ainsi l'algorithme CMA-ES, pour un coût calculatoire bien plus faible. Mais, entre ces deux méthodes (à savoir l'exploration mono-dimensionnelle ou la marche aléatoire), le choix est plus difficile. La première a la certitude de converger vers un minimum local, tandis que la seconde permet d'explorer des points alentours et potentiellement meilleurs.

Ces observations semblent ainsi montrer qu'un minimum local de notre fonction objectif est relativement proche d'un de ses points selles. Cela souligne encore une fois la fiabilité *éventuelle* de la descente de gradient ; l'éventualité réside dans le fait que nous ne connaissons pas le niveau du minimum global...

2.3 Conclusion

Ce chapitre s'est proposé de résoudre le problème d'optimisation que nous avons défini précédemment. Il est rassurant de noter que ce dernier s'avère bien posé, dans la mesure où sa résolution permet d'obtenir des séquences dont l'autocorrélation affiche l'aspect souhaité. Le choix de la norme semble influencer les résultats.

De nombreuses méthodes d'optimisation ont été définies et appliquées dans la littérature, mais le but principal de cette section était d'effectuer un comparatif entre méthodes déterministes et stochastiques. En matière de performances brutes, celles étudiées se valent plus ou moins, avec éventuellement un léger avantage pour l'algorithme CMA-ES.

Les algorithmes stochastiques ont l'avantage d'une grande simplicité à l'utilisation. Ils ne nécessitent en effet que de l'expression (et encore...) de la fonction objectif. Leur temps de convergence est par contre assez long. Par opposition, leurs homologues déterministes, avec notamment les méthodes de descente de gradient, sont clairement moins coûteux, pour peu que l'expression analytique des dérivées soit connue. Ils sont donc compatibles avec une utilisation en temps réel, par exemple. Les performances de la descente de gradient sont très légèrement en deçà ; une recherche dimensionnelle naïve ou l'ajout d'aléatoire (*cf.* la marche aléatoire) peut compenser cela.

Une descente de gradient converge généralement vers un point selle. Cependant, l'ensemble des optimisations effectuées montre que ces points selles présentent des niveaux de lobes secondaires relativement équivalents (et intéressants de surcroît). L'application d'une recherche mono-dimensionnelle dans le voisinage révèle que le minimum local le plus proche n'est pas si éloigné que cela...

Finalement, cette étude nous encourage à employer une descente de gradient afin de rechercher des formes d'onde. Mais il faut prendre cette conclusion avec précaution. Seul un nombre fini d'optimisation a été effectué, ce qui n'est assurément pas représentatif de l'espace des séquences. De plus, aucune propriété n'est vraiment connue sur les séquences d'autocorrélation et sur le minimum global (et sa valeur) de notre fonction objectif, mis à part une borne qui sera évoquée plus loin. Si une telle propriété était connue, peut-être que ce chapitre n'aurait pas lieu d'être...

Pour en être convaincu, il nous aurait fallu éventuellement tester d'autres approches. Nous pensons notamment aux méthodes de pénalisation, voire même de relaxation ; il existe d'ailleurs un terme dédié pour la relaxation sur le module de la séquence, appelé *PAPR*, pour *Peak-to-Average Power Ratio*. Ou voir le problème sous un autre angle, en étudiant par exemple la variété des séquences (qui n'est qu'un tore \mathbf{T}^N de dimension finie).

Une partie de ce chapitre a fait l'objet d'une publication [Tan et al., 2016c].

Chapitre 3

Aux contraintes opérationnelles

Le chapitre précédent a montré qu'il est possible, à partir d'un problème d'optimisation assez *simple*, de retrouver des séquences présentant de bonnes autocorrélations. « Simple » au sens de l'apparence, bien sûr, et non de sa résolution ! D'ailleurs, la fonction de coût est primordiale, car elle traduit, avec les contraintes, les besoins opérationnels. En ce sens, la modélisation d'un problème sous la forme d'une optimisation se révèle assez flexible car il suffit donc, en quelque sorte, de définir ladite fonction.

Dans ce chapitre, on se propose ainsi de procéder à quelques modifications sur notre problème initial. Cela nous aidera à établir quelques observations, sans ordre apparent, sans quelconque relation, sur :

- la robustesse des codes de phase et l'influence du sur-échantillonnage ;
- l'introduction d'une contrainte spectrale ;
- les phénomènes d'éclipse ;
- les effets de la quantification.

L'ensemble de ces études ne concerne que le cas mono-séquence. Mais rappelez-vous : cette thèse a pour finalité les radars MIMO, ce qui sous-entend l'utilisation de plusieurs séquences. Ainsi, nous verrons dans une autre section qu'il est relativement aisé d'étendre notre problème afin d'obtenir des familles de séquences pseudo-orthogonales.

Enfin, *quid* de leurs performances ? Le chapitre précédent a comparé différentes méthodes d'optimisation, ne décrivant ainsi que des performances atteignables. Est-ce possible de faire mieux ? Quelle est la limite ? Une dernière section décrira et comparera les bornes à ce sujet.

3.1 Altération des codes de phase

Commençons cette section par une petite étude sur la robustesse des codes de phase, et ce, envers deux facteurs. Le premier concerne les phases de la séquence émise elle-même. Que se passe-t-il si le signal émis est légèrement différent de celui qu'on avait prédéterminé ? Une baisse des performances est logiquement attendue, mais à quelle teneur ? Le second concerne le sur-échantillonnage du signal et l'interpolation de ses phases.

3.1.1 Robustesse

Considérons une séquence \mathbf{s} de longueur $N = 63$ (cette séquence provient de la Section 2.2.2.4, pour être tout à fait transparent), et notons par $\boldsymbol{\varphi} \in [0, 2\pi[^N$ son vecteur de phase associé, de même longueur. On se propose d'opérer une légère perturbation de ces phases, comme suit. Soit $\boldsymbol{\alpha}$ un bruit additif gaussien sur la phase. Chaque élément α_n , pour $n \in \llbracket 1, N \rrbracket$, suit une loi normale $\alpha_n \sim \mathcal{N}(0, (\pi\sigma/180)^2)$, de sorte que l'écart-type soit explicitement proportionnel à un changement de phase en degrés. Le signal bruité a donc pour vecteur de phase $\boldsymbol{\alpha} + \boldsymbol{\varphi}$; un exemple est présenté à la Figure 3.1.

Afin de caractériser les effets de cette dégradation sur les performances, une simulation est effectuée. Elle consiste, à partir d'une séquence donnée, à faire évoluer le bruit par le biais du paramètre σ ; pour chacune de ces valeurs, 10000 instances sont réalisées. Les résultats moyens sont présentés en Figure 3.2 et 3.3.

Pour commencer, on étudie le lobe secondaire le plus élevé de l'autocorrélation de la séquence bruitée, représenté en bleu sur la Figure 3.2. Le caractère croissant de cette courbe est attendu. Il est logique qu'une modification des phases entraîne une remontée des lobes secondaires. Cependant, celle-ci est assez rapide : une perturbation d'un degré (en écart-type) entraîne un déficit de l'ordre de 0.5 dB, tandis qu'une autre de 5° dégrade de 4 dB, ce qui commence à faire beaucoup. Les codes polyphasés ne semblent donc pas très robustes...

Supposons maintenant que le signal émis par un radar soit malencontreusement bruité. À la réception, le traitement sera involontairement désadapté, puisque le filtre utilisé — le signal original, non bruité — est différent du signal émis. Cela dit, la courbe rouge de la Figure 3.2 (le niveau des lobes secondaires après filtrage « désadapté ») indique que les performances ne sont clairement pas détériorées, par rapport à un filtrage adapté (la séquence bruitée avec elle-même, en bleu). De plus, la perte entraînée par la désadaptation n'est pas si conséquente (en vert, Figure 3.3) : par exemple, on note une dégradation de 0.1 dB pour un changement de 10 degrés. Finalement, pour cette simulation, il semblerait que les changements de phase ne conduisent pas à une perte importante en terme de LPG (*Loss-in-Processing Gain*), mais à une remontée rapide de PSLR. Notez qu'on évoquera plus en détail le filtrage désadapté dans le prochain chapitre.

3.1.2 Sur-échantillonnage

Intéressons-nous maintenant à l'échantillonnage d'un signal polyphasé. Lorsqu'on considère le code de phase associé, l'échantillonnage est généralement effectué à chaque saut de phase, et ce de manière régulière. Que se passe-t-il lorsque cette discrétisation est différente ? Si celle-ci est plus grossière, il est assez probable que les performances soient dégradées, la séquence en question étant différente de l'originale. Dans le cas d'une discrétisation plus fine, nous nous proposons d'évaluer deux alternatives, listées ci-dessous.

- Une interpolation constante par morceaux. Chaque phase est répétée plusieurs fois.
- Une interpolation cubique. Un polynôme est construit (ici, d'ordre 3) dans le domaine des phases, à partir de la séquence initiale. Un échelonnement régulier permet de définir la nouvelle séquence sur-échantillonnée [Stoer *et al.*, 2013]. Cela s'apparente à l'utilisation d'un filtre passe-bas en pratique.

A des fins illustratives, la Figure 3.4 dépeint les phases d'une séquence de longueur $N = 63$, sur-échantillonnée d'un facteur 15, selon les deux interpolations. Quant à la Figure 3.5, les séquences d'autocorrélation sont représentées. Ces dernières semblent, en terme de lobes secondaires, assez semblables. On a même l'impression que la séquence d'autocorrélation initiale (en bleu, sur la figure) « relie » les points les plus élevés de la courbe rouge (sur-échantillonnage linéaire) — ce qui est attendu. La séquence obtenue par une interpolation cubique présente néanmoins des lobes secondaires assez prononcés, notamment autour du lobe principal.

Un mot d'ailleurs sur ce lobe principal : un épaissement de celui-ci est à noter, comparé au lobe sous forme triangulaire, signature d'un code de phase discret. Comme nous l'avons vu dans le chapitre précédent, la résolution d'un radar est notamment déterminée à partir de la mesure du lobe principal à -3 dB, appelée et notée $\theta_{3\text{dB}}$. De ce fait, il faut donc veiller à ce que la résolution du radar soit bien définie selon les caractéristiques du radar (donc, à peu près T/N), et avec le bon échantillonnage...

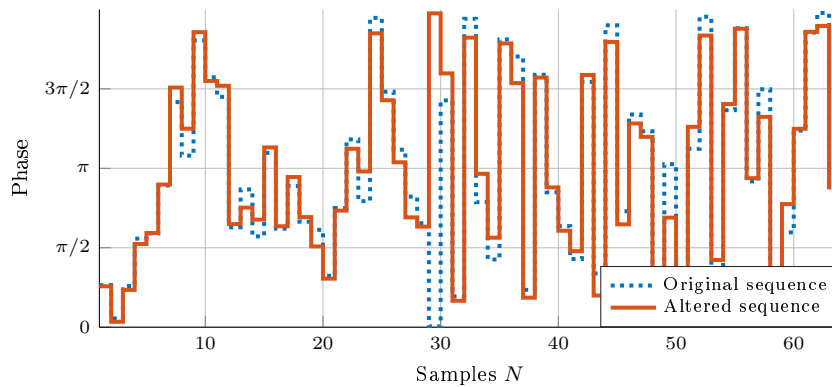


FIGURE 3.1 – Représentation des phases d'un signal bruité.
 Pour que cet exemple soit assez explicite, $\sigma^2 = 15$ ici.

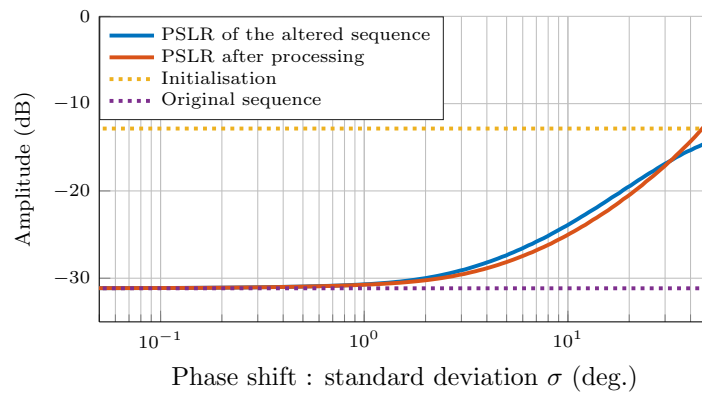


FIGURE 3.2 – Altération des phases d'un code polyphase (1)
 En bleu : le PSLR moyen de la séquence perturbée après 10000 instances.
 En rouge : le PSLR moyen après filtrage de la séquence altérée par la séquence initiale,
 sans prendre en compte la perte.

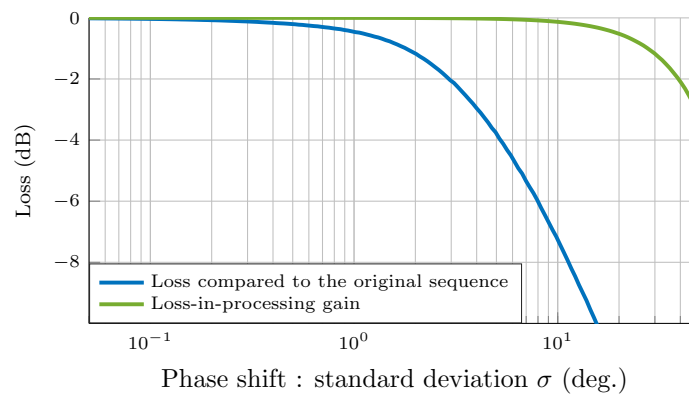


FIGURE 3.3 – Altération des phases d'un code polyphase (2)
 En bleu : mesure de la dégradation du PSLR comparé à la séquence initiale. En vert : perte
 due à la désadaptation (lors du filtrage).

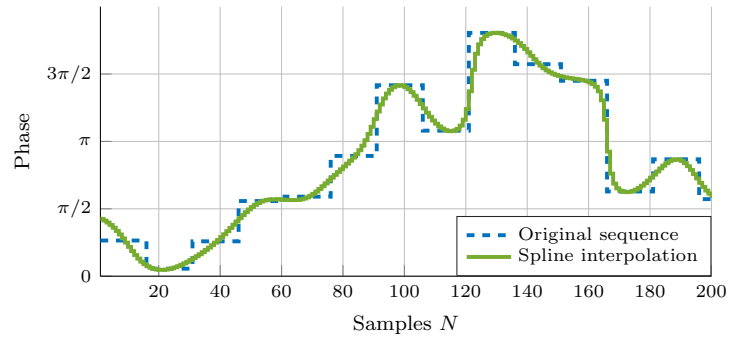


FIGURE 3.4 – Phases d'une séquence sur-échantillonnée
Sur-échantillonnage d'une séquence de longueur $N = 63$ d'un facteur 15. Zoom sur les premières phases. L'interpolation constante par morceau n'est pas représentée ici, car elle serait confondue avec les phases de la séquence originale.

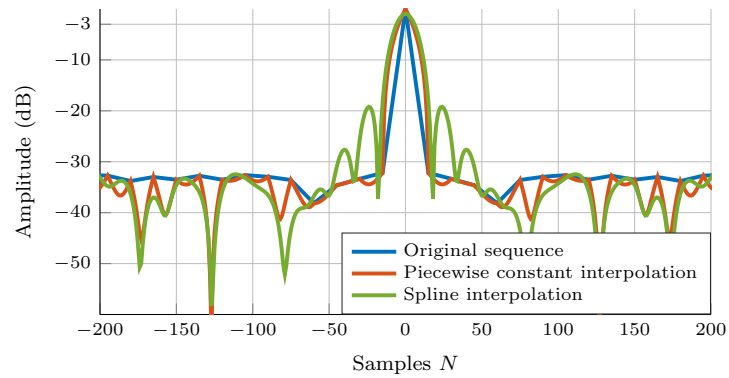


FIGURE 3.5 – Corrélation d'une séquence sur-échantillonnée, selon différentes interpolations
Sur-échantillonnage d'une séquence de longueur $N = 63$ d'un facteur 15. Focalisation sur le lobe principal.

3.2 Contrainte spectrale

Le problème d'optimisation (P_1) défini à la Section 2.1.5 est composé de deux fonctions : une mesurant les lobes secondaires de la séquence, et une autre définissant la contrainte de module constant. Il est assez courant dans la littérature d'y ajouter une contrainte *spectrale*. Cette contrainte semble superflue dans une étude sur l'orthogonalité d'une séquence. Elle reste cependant nécessaire en pratique, dans une application radar par exemple, pour limiter l'énergie émise hors-bande, afin notamment de ne pas polluer les bandes réservées à d'autres usages (dans les communications par exemple) [NTIA, 2000].

Ainsi, dans ce paragraphe, on se propose de montrer qu'il est possible de contraindre le spectre d'une séquence, par le biais d'un problème d'optimisation. Il nous faudra malgré tout garder en tête que l'image par la fonction d'autocorrélation est importante. Pour cela, on s'inspirera des travaux effectués par Stoica *et al.* dans [He *et al.*, 2012], où l'optimisation a été effectuée dans le domaine spectral.

3.2.1 Modélisation d'un problème d'optimisation

Spectre Soit $\mathbf{s} = [s_1, \dots, s_N]^T$ une séquence de longueur N . On rappelle, d'une part, que la transformée de Fourier discrète de \mathbf{s} est également une séquence de même longueur, définie par :

$$(\mathcal{F}(\mathbf{s}))_k = \sum_{n=1}^N s_n e^{-2i\pi(k-1)\frac{n-1}{N}}, \quad 1 \leq k \leq N, \quad (3.1)$$

où, par abus de notation, \mathcal{F} dénote également l'opérateur de Fourier discret. Le vecteur $\mathcal{F}(\mathbf{s})$ spécifie le spectre du signal \mathbf{s} . D'autre part, on peut définir l'énergie spectrale de ce dernier à l'aide d'un opérateur, que l'on note E :

$$E(\mathbf{s}) := \|\mathcal{F}(\mathbf{s})\|_2^2. \quad (3.2)$$

Afin d'améliorer la précision spectrale, on effectue un *zero-padding* de la séquence \mathbf{s} , que l'on note $\tilde{\mathbf{s}}$. Cela consiste à effectuer un prolongement par des zéros sur la droite, *i.e.*, :

$$\tilde{\mathbf{s}} = [s_1, \dots, s_N, \underbrace{0, \dots, 0}_{\tilde{N}-N}]^T, \quad \text{avec } \tilde{N} > N, \quad (3.3)$$

de sorte que la séquence $\tilde{\mathbf{s}}$ ainsi construite soit donc de longueur \tilde{N} .

Il ne reste plus qu'à définir les bandes spectrales que l'on souhaite minimiser. On s'aide pour cela d'un vecteur de pondération, appelé \mathbf{w} et de longueur \tilde{N} , tel que :

$$\begin{cases} \sum_{k=1}^{\tilde{N}} w_k = 1, \\ w_k \geq 0, \quad k \in \llbracket 1, \tilde{N} \rrbracket. \end{cases} \quad (3.4)$$

Le rôle de la pondération \mathbf{w} est ainsi d'attribuer un coefficient au vecteur spectral $\mathcal{F}(\tilde{\mathbf{s}})$: les bandes que nous voulons minimiser sont assignées d'une certaine valeur (généralement identique), tandis que les autres sont nulles. Par exemple, si on souhaite minimiser l'énergie dans une « première » bande (cela est illustré plus loin à la Figure 3.7), il suffit de poser :

$$\begin{cases} w_k = \frac{1}{\lfloor 0.1\tilde{N} \rfloor}, & \text{si } k \in \llbracket 1, \lfloor 0.1\tilde{N} \rfloor \rrbracket, \\ w_k = 0, & \text{sinon.} \end{cases} \quad (3.5)$$

Ainsi, dans l'expression ci-dessus, on essaie de réduire l'énergie spectrale contenue dans les premiers 10% de la séquence sur-échantillonnée.

Soit \mathbf{F}_f une matrice diagonale de taille \tilde{N} , définie par le vecteur de pondération \mathbf{w} . Nous pouvons finalement écrire un problème d'optimisation pour contraindre le spectre d'une séquence \mathbf{s} , comme suit :

$$(P_3) \begin{cases} \min_{\mathbf{s}} & E_1(\mathbf{s}) := \|\mathbf{F}_f \mathcal{F}(\tilde{\mathbf{s}})\|_2^2 \\ \text{t.q.} & \mathbf{s} \in \mathbf{T}^N. \end{cases} \quad (3.6)$$

Le problème d'optimisation ci-dessus se concentre ainsi seulement sur les éléments de la séquence de Fourier à minimiser.

Remarque 3.1. (Sur la pondération) L'équation (3.5) est un exemple sommaire de pondération. Mais il est également possible :

- d'interdire plusieurs bandes de fréquences disjointes ;
- de définir une pondération non binaire, notamment par le biais d'une fonction, afin de faciliter l'optimisation ;
- de minimiser l'énergie hors-bande.

Tout cela dépend bien sûr des besoins opérationnels !

Une optimisation locale Rappelons le problème d'optimisation initial (P_1) qui traite des lobes secondaires d'une séquence :

$$(P_1) \begin{cases} \min_{\mathbf{s}} & f_p(\mathbf{s}) := \|\mathbf{F}\mathbf{r}\|_p, \\ \text{t.q.} & \mathbf{s} \in \mathbf{T}^N. \end{cases} \quad (3.7)$$

où, si nous nous répétons, « \mathbf{F} est une matrice diagonale de taille $2N - 1$, spécifiée par le vecteur $[1, \dots, 1, 0, 1, \dots, 1]$ (un vecteur composé uniquement de 1, sauf à la position N , représentant le lobe principal) ». Cela ne vous rappelle-t-il rien ? La matrice \mathbf{F} est d'une certaine façon une pondération sur l'ensemble des éléments de la séquence d'autocorrélation, ôté du lobe principal. Ainsi, afin d'effectuer une optimisation locale autour du lobe principal, il suffit de considérer une matrice dont la diagonale est sous la forme :

$$\begin{array}{c} \text{Pour ignorer} \\ \text{le lobe principal} \\ \downarrow \\ [0, \dots, 0, 1, \dots, 1, 0, 1, \dots, 1, 0, \dots, 0] \\ \uparrow \qquad \qquad \qquad \uparrow \\ \text{Pour ignorer} \\ \text{les bords} \end{array} \quad (3.8)$$

Pour la suite, soient \mathbf{F}_3 la matrice diagonale basée sur (3.8) et (P_4) le problème d'optimisation associé, inspiré de (P_1). Appelons E_2 sa fonction objectif.

Remarque 3.2. (Low Correlation Zone Sequences) La résolution du problème (P_4) permet d'obtenir des séquences dont les lobes secondaires sont optimisés dans une certaine zone, autour du lobe principal. Il existe une littérature [Chung et al., 2010] consacrée à la recherche de ces séquences, appelées LCZS, pour Low Correlation Zone Sequences. En outre, dans le cas périodique, on s'intéresse même aux ZCZS (Zero Correlation Zone Sequences).

Une optimisation multi-objectif Finalement, nous avons deux problèmes d'optimisation entre nos mains, que nous réécrivons ci-dessous :

$$(P_3) \begin{cases} \min_{\mathbf{s}} & E_1(\mathbf{s}) := \|\mathbf{F}_f \mathcal{F}(\tilde{\mathbf{s}})\|_2^2 \\ \text{t.q.} & \mathbf{s} \in \mathbf{T}^N, \end{cases} \quad (P_4) \begin{cases} \min_{\mathbf{s}} & E_2(\mathbf{s}) := \|\mathbf{F}_3 \mathbf{r}\|_p, \\ \text{t.q.} & \mathbf{s} \in \mathbf{T}^N. \end{cases} \quad (3.9)$$

Le premier, (P_3), concerne la contrainte spectrale, tandis que le second, (P_4), évalue les lobes secondaires d'une séquence. Il nous reste maintenant à savoir comment traiter ces deux problèmes. Séparément ? Simultanément ? Le paragraphe suivant évoque brièvement différentes possibilités, bien qu'elles ne seront pas comparées.

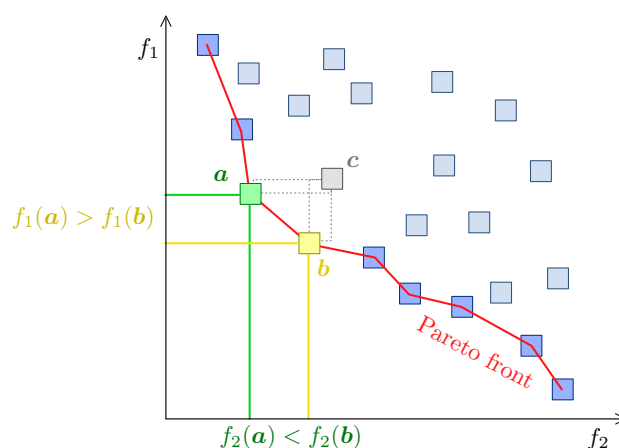


FIGURE 3.6 – Illustration d'un front de Pareto

Les points \mathbf{a} et \mathbf{b} sont dits « efficaces » : la réduction de la valeur selon une des fonctions objectifs — ici, f_1 et f_2 — entraîne nécessairement une augmentation de l'autre. L'ensemble de ces points efficaces est appelé *Front de Pareto*. Crédits : Johann Dréo.

3.2.2 Parenthèse sur l'optimisation multi-objectif

Il est assez difficile d'évoquer l'optimisation multi-objectif en quelques mots, mais nous nous y risquons malgré tout ici. Le sujet est en effet assez vaste, et peut-être plus connu sous le terme « aide à la décision », croisé notamment en économie — bien que ce dernier évoque en fait le cas particulier où l'ensemble des solutions est discret. Pour les plus curieux, nous vous invitons à jeter un coup d'œil à [Branke *et al.*, 2008, Collette *et al.*, 2004], qui ont nourri cette parenthèse. Un problème d'optimisation multi-objectif élémentaire s'écrit :

$$\begin{aligned} \min_{\mathbf{x}} \quad & \{f_1(\mathbf{x}), \dots, f_m(\mathbf{x})\} \\ \text{t.q.} \quad & \mathbf{x} \in \mathcal{S} \end{aligned} \quad (3.10)$$

Comme vous l'avez sans doute deviné, le terme « multi-objectif » se rapporte ainsi au nombre de fonctions objectifs que nous voulons optimiser simultanément, avec $m > 1$, contrairement à ce que nous avons pu voir précédemment. De plus, le vecteur $\mathbf{f}(\mathbf{x}) := [f_1(\mathbf{x}), \dots, f_m(\mathbf{x})]^T$ est appelé *vecteur objectif*.

Optimalité Ce dernier permet de comparer deux solutions potentielles \mathbf{x} et \mathbf{y} . Comme il est sans aucun doute délicat de donner une relation d'ordre entre deux vecteurs, les notions de « domination » et de « front de Pareto » ont été définies. On dit que \mathbf{x} domine \mathbf{y} si :

1. La solution \mathbf{x} est aussi bonne que \mathbf{y} pour l'ensemble des objectifs, ce qui peut s'écrire $\forall i \in \llbracket 1, m \rrbracket, (\mathbf{f}(\mathbf{x}))_i \leq (\mathbf{f}(\mathbf{y}))_i$,
2. \mathbf{x} est meilleure que \mathbf{y} sur un objectif au moins, *i.e.* $\exists i \in \llbracket 1, m \rrbracket, \mathbf{f}_i(\mathbf{x}) < \mathbf{f}_i(\mathbf{y})$.

Un vecteur \mathbf{x} est dit optimal au sens de Pareto s'il n'en existe aucun autre qui le domine. Le front de Pareto est défini comme l'ensemble des solutions optimales au sens de Pareto, *i.e.* l'ensemble des solutions dominantes. Le front de Pareto représente ainsi un certain compromis, comme l'illustre la figure ci-dessous.

Méthodes d'optimisation Un problème d'optimisation multi-objectif admet donc un ensemble de solutions. La difficulté réside donc dans la capacité à déterminer cet ensemble, ou un élément contenu dans celui-ci. Pour cela, il existe de nombreuses approches, classées généralement en trois catégories : les approches *a priori*, *progressives* et *a posteriori*. L'énumération qui suit les caractérise en quelques mots et présente quelques exemples de méthodes d'optimisation.

Les méthodes dites *a priori*. Le compromis entre les différentes fonctions objectifs est déterminé en amont. La procédure d'optimisation propose ainsi une seule solution, reflet de ce compromis. Les méthodes d'optimisation *a priori* sont les plus courantes ; parmi elles :

- La méthode de pondération des fonctions objectifs est l'approche la plus naïve. Elle regroupe l'ensemble des fonctions objectifs, de sorte que le problème devienne mono-objectif :

$$\begin{cases} \min_{\mathbf{x}} & \{f_1(\mathbf{x}), \dots, f_m(\mathbf{x})\} \\ \text{t.q.} & \mathbf{x} \in \mathcal{S}, \end{cases} \implies \begin{cases} \min_{\mathbf{x}} & \sum_{i=1}^m w_i f_i(\mathbf{x}) \\ \text{t.q.} & \mathbf{x} \in \mathcal{S}. \end{cases} \quad (3.11)$$

Le vecteur de pondération \mathbf{w} suit les conditions suivantes :

$$\begin{cases} \sum_{i=1}^m w_i = 1, \\ w_i \geq 0, \quad i \in \llbracket 1, m \rrbracket. \end{cases} \quad (3.12)$$

Remarquez qu'il est possible qu'un coefficient puisse être nul, si la fonction de coût associée est estimée secondaire. Il ne reste plus qu'à résoudre le problème à l'aide des méthodes présentées à la Section 2.2...

- La méthode de la distance à un objectif de référence consiste à minimiser, comme son nom l'indique, la distance envers un vecteur objectif utopique, qu'on note ici $\mathbf{g} \in \mathbb{R}^m$. En utilisant une distance usuelle, cela s'écrit :

$$\begin{cases} \min_{\mathbf{x}} & \{f_1(\mathbf{x}), \dots, f_m(\mathbf{x})\} \\ \text{t.q.} & \mathbf{x} \in \mathcal{S}, \end{cases} \implies \begin{cases} \min_{\mathbf{x}} & (\sum_{i=1}^m |f_i(\mathbf{x}) - g_i|^p)^{1/p} \\ \text{t.q.} & \mathbf{x} \in \mathcal{S}. \end{cases} \quad (3.13)$$

D'autres distances ont été définies à cet effet [Collette *et al.*, 2004].

- L'ordonnancement lexicographique se propose de minimiser les fonctions objectifs les unes après les autres. On commence ainsi par résoudre :

$$\begin{cases} \min_{\mathbf{x}} & f_1(\mathbf{x}) \\ \text{t.q.} & \mathbf{x} \in \mathcal{S}. \end{cases} \quad (3.14)$$

Soit f_1^* le minimum obtenu. On place cette valeur sous la forme d'une contrainte, et on s'occupe du deuxième objectif :

$$\begin{cases} \min_{\mathbf{x}} & f_2(\mathbf{x}) \\ \text{t.q.} & f_1(\mathbf{x}) = f_1^*, \\ & \mathbf{x} \in \mathcal{S}. \end{cases} \quad (3.15)$$

Cette procédure est répétée jusqu'à la m -ème fonction objectif. Ainsi, comme son nom l'indique, elle privilégie donc le critère f_1 , puis f_2 , *etc.*

Les méthodes progressives font, quant à elles, intervenir un opérateur durant le processus d'optimisation. Celui-ci oriente la recherche vers des solutions qui satisfassent les compromis qu'il souhaite opérer. Ces méthodes ont donc l'inconvénient d'accaparer le décideur pendant toute la durée de calcul ; elles sont donc très peu utilisées. Néanmoins, présentons l'une d'entre elles, baptisée méthode du compromis par substitution.

- La méthode du compromis par substitution propose de se préoccuper de chaque fonction objectif une à une, en transformant les autres en contraintes. Pour cela, soient $\varepsilon \in \mathbb{R}^m$ et $\mathbf{w} = \mathbf{0}_m$ deux vecteurs de contraintes. Il nous faut résoudre :

$$\begin{cases} \min_{\mathbf{x}} & f_1(\mathbf{x}) \\ \text{t.q.} & f_i(\mathbf{x}) \leq \varepsilon_i + w_i, \quad \forall i \in \llbracket 2, m \rrbracket, \\ & \mathbf{x} \in \mathcal{S}. \end{cases} \quad (3.16)$$

La suite s'apparente à un jeu d'équilibre, dicté par l'opérateur. Dans un premier temps, le vecteur ε se doit d'être suffisamment relâché — mais pas trop ! — pour que (3.16) admette des solutions réalisables. Dans un second temps, certaines composantes de \mathbf{w} sont réduites (respectivement, augmentées) afin de mettre en exergue les fonctions coûts associées.

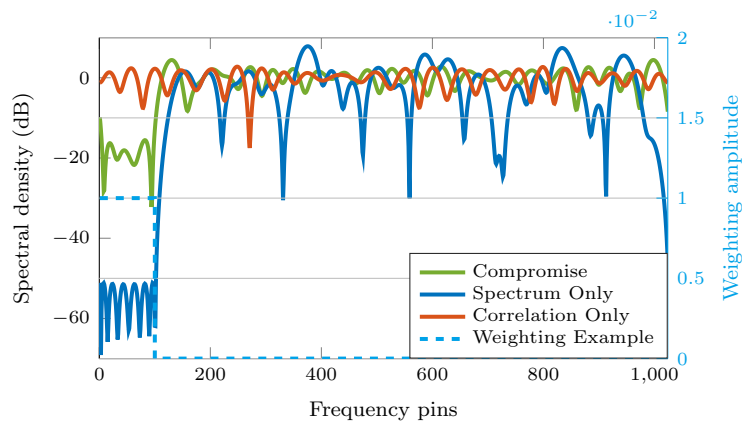


FIGURE 3.7 – Densité spectrale d’une séquence de longueur $N = 31$
Représentation d’une FFT sur-échantillonnée.
Normalisation selon la moyenne spectrale.
La bande « supprimée » concerne les premières valeurs en fréquence.

Cette procédure se répète jusqu’à ce que le décideur considère que la solution obtenue est suffisamment représentative de ses désirs.

Enfin, les méthodes *a posteriori* proposent un ensemble de solutions réalisables, généralement bien réparties, modélisant ainsi le front de Pareto. Le choix de la solution se fait donc... *a posteriori*. Elles ont l’avantage d’éviter l’étape de la définition du compromis, mais endurent quelques difficultés supplémentaires :

- La méthode *a posteriori* la plus simple consiste à effectuer plusieurs instances d’une méthode *a priori*. Par exemple, appliquer la méthode de pondération des fonctions objectifs en faisant varier la pondération w permet de définir convenablement l’ensemble de compromis.
- La multiplicité des instances induit nécessairement un temps de calcul accru.
- Par ailleurs, comment représenter de manière convenable l’ensemble de solutions à un opérateur — humain, de surcroît ? Lorsqu’il y a précisément deux fonctions coûts, un graphe semble approprié, comme celui de la Figure 3.6. Mais sinon...

Ce paragraphe avait pour but d’illustrer les possibilités immenses de l’optimisation multi-objectif, qu’on effleura à peine dans cette thèse. La parenthèse est désormais fermée.

3.2.3 Exemple numérique

Concluons ce paragraphe sur les contraintes spectrales par la résolution du problème multi-objectif défini en (3.9). Deux fonctions objectifs confectionnent notre problème : la première traite de la composante spectrale, la seconde de la séquence d’autocorrélation. Afin d’illustrer l’idée de compromis, les Figures 3.7 et 3.8 présentent à la fois les résultats d’une optimisation mono-objectif (en ignorant son homologue) et d’une autre, multi-objectif. Pour ce dernier, nous utiliserons une méthode de pondération des fonctions objectifs, mais avec une petite subtilité : la pondération w n’est pas constante mais dépend du niveau des lobes secondaires pour chacune des contraintes. L’optimisation a été effectuée sur une séquence de longueur $N = 31$.

A la Figure 3.7, on se propose de réduire une seule bande spectrale comprenant les « 100 premiers » échantillons (pour information, $\tilde{N} = 1024$). Pour une séquence non optimisée pour ce critère — la courbe rouge sur la Figure 3.7, focalisée sur la corrélation — la densité spectrale apparaît relativement constante. Quant aux autres séquences, elles exhibent une différence sensible entre cette première bande et les autres échantillons. Cela nous rassure, car cela montre que le problème est bien posé, dans une certaine mesure.

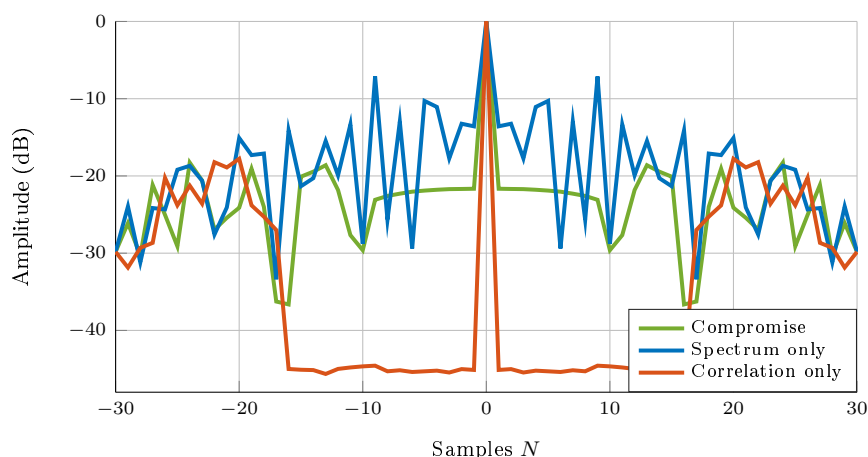


FIGURE 3.8 – Corrélation de séquences optimisées spectralement
L'optimisation est effectuée localement, autour du pic principal.

A la Figure 3.8, une optimisation locale autour du lobe principal a été effectuée. Après adaptation, les commentaires du paragraphe précédent restent adéquats... Ajoutons alors simplement que des lobes secondaires aux alentours de -45 dB ont été obtenus, ce qui est assez intéressant pour des séquences aussi courtes. Dans une optique radar, une optimisation locale est utile lorsqu'on veut différencier du fouillis les cibles proches.

Les résultats d'une optimisation mono-objectif décrivent une borne approximative pour le problème multi-objectif, mais atteignable en pratique. Pour une séquence de longueur $N = 31$, il est possible de s'attendre à des lobes secondaires autour de -45 dB, tout en sachant que cela entraîne probablement — et malheureusement — un spectre non régulé. Une augmentation de ces lobes secondaires offrira une certaine latitude pour un contrôle du fantôme. Et *vice versa*. Ceci est donc une histoire de compromis qui, en pratique, doit refléter les besoins opérationnels. Un exemple numérique est d'ailleurs donné sur les deux figures, en vert.

Remarque 3.3. *(Sur le pourquoi de l'optimisation locale.) Contrairement aux sections précédentes, l'optimisation de la séquence de corrélation a été effectuée localement. Ce choix a été motivé pour deux raisons simples. Premièrement, aucune minimisation locale n'a encore été présentée dans ce chapitre. Deuxièmement, la contrainte spectrale influe de manière assez prépondérante sur le système. Pour que les résultats puissent être probants, il nous a fallu relâcher une contrainte... Notez qu'une relaxation sur la contrainte de module constant (ou, en d'autres termes, sur le PAPR) aurait pu être considérée.*

Ce paragraphe illustre la possibilité de contrôler simultanément l'énergie spectrale en quelques bandes et les lobes secondaires. Bien que ce critère est foncièrement important dans la pratique, il sera omis par la suite, afin de soulager les optimisations futures.

3.3 Phénomènes d'éclipse

Intéressons-nous maintenant à toute autre chose : les phénomènes d'éclipse. La définition d'une éclipse dans un dictionnaire classique [Larousse, 2015] nous offre : « Absence, disparition temporaire [...] de quelqu'un, d'une activité ». Mais, quel est le lien avec les formes d'onde? Pour cela, revenons à un principe sommaire d'un radar, illustré à la Figure 3.9.

Un radar classique est aveugle lorsqu'il émet (oui, les sens d'un radar sont relativement chamboulés). En effet, soit l'émission et la réception sont effectuées par une seule et même antenne, soit l'antenne d'émission interfère avec son homologue.

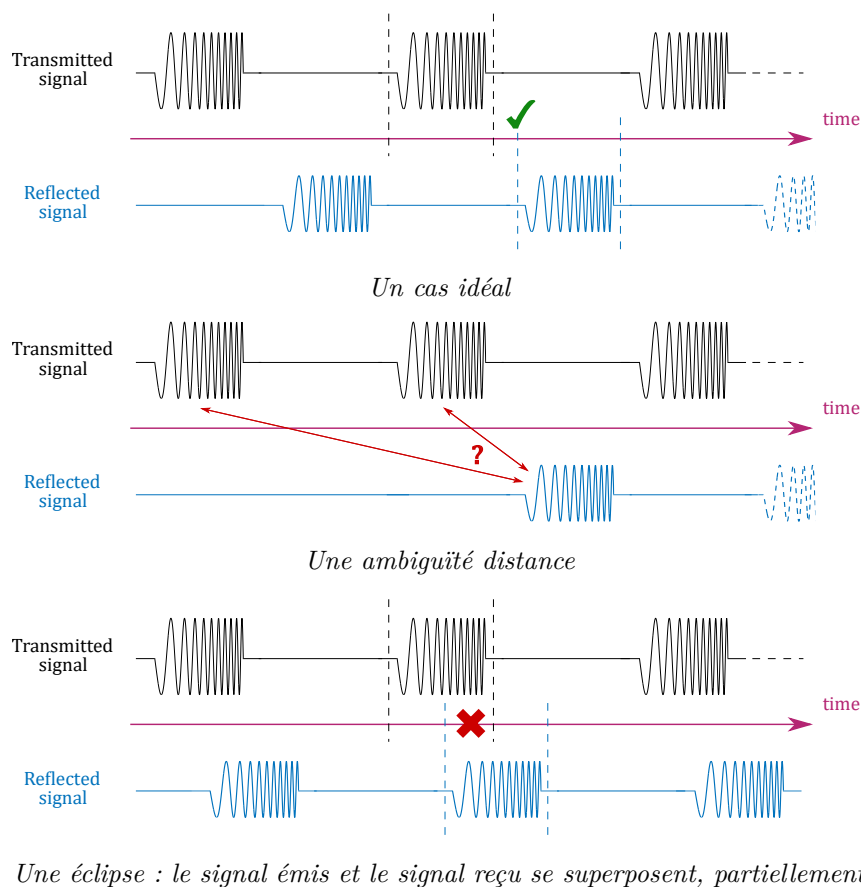


FIGURE 3.9 – Illustration d'un phénomène d'éclipse

En se rappelant qu'un radar émet généralement de manière pulsée, quelles sont les situations temporelles qui s'offrent à nous ?

- Le cas idéal consiste à ce que l'écho du signal émis soit perçu pendant la période de répétition *correspondante*. La position de la cible peut être estimée convenablement.
- Lorsque la correspondance ne peut-être établie, il y a *ambiguïté distance*. Par exemple, sur la Figure 3.9, au centre, il est impossible de déterminer si l'écho provient de la première impulsion ou de la seconde...
- Enfin, il est possible que le signal réfléchi soit reçu au cours d'une émission, et donc qu'il y ait une intersection temporelle entre l'émission et la réception, comme le montre le bas de la Figure 3.9. Dans ce cas, le début de l'impulsion n'est pas enregistré par le radar. C'est ce qu'on appelle une « éclipse » [Blunt *et al.*, 2008]. Signalons qu'on ne considère que les éclipses partielles, puisqu'une éclipse totale correspondrait à une non-réception...

Finalement, le phénomène d'éclipse se définit donc par une perte partielle d'informations à la réception. Cette perte se manifeste sous différentes formes à l'étape de traitement, comme le montre la Figure 3.10. Sur celle-ci, on applique un filtre adapté sur le signal reçu, malgré le fait que ce dernier soit tronqué de moitié (une éclipse à 50%). Pour le *chirp*, on note une perte énergétique et un épaississement du lobe principal (lié à la diminution de la bande du signal intrinsèque à la troncature du signal). Pour un code polyphasé, on y ajoute une remontée des lobes secondaires, mais on n'observe pas nécessairement une augmentation de la largeur du lobe principal.

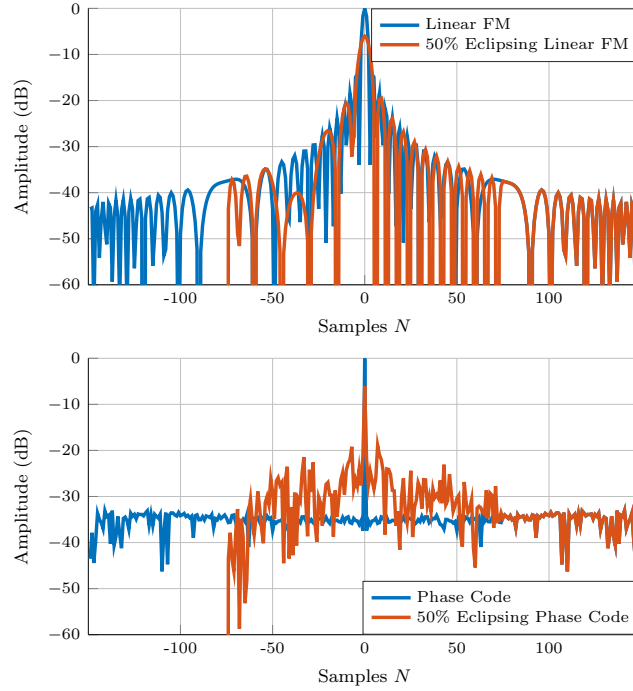


FIGURE 3.10 – Séquence de corrélation pour un signal éclipse

En haut : un chirp. En bas : un code de phase.

Notez l'épaississement du lobe principal pour le chirp, la remontée des lobes secondaires pour le code de phase, et la perte due à la désadaptation.

La perte énergétique est intrinsèque à la désadaptation. Elle est d'ailleurs identique pour toute séquence à module constant subissant la « même » éclipse. Cependant, on s'aperçoit que la modulation linéaire de fréquence conserve ses lobes secondaires, ce qui ne semble pas le cas pour les codes de phase. Si bien qu'on peut se demander s'il existe des codes de phase robustes aux éclipses. Ceci est le sujet du paragraphe suivant. Avant cela, signalons qu'un travail similaire a été effectué dans le cas de modulations non-linéaires [Alabaster *et al.*, 2017, Blunt *et al.*, 2008].

3.3.1 À gauche...

Modélisation Soit $\mathbf{s} = [s_1, \dots, s_N]^T$ une séquence de longueur N . Une éclipse en début de portée (à gauche) se traduit par une absence partielle d'informations en début de signal ; le signal reçu $\mathbf{s}_{\text{ecl.}}$ est donc sous la forme :

$$\mathbf{s}_{\text{ecl.}}^i = [0, \dots, 0, s_{i+1}, \dots, s_N]^T \in \mathbb{C}^N. \quad (3.17)$$

L'indice i compte ainsi le nombre d'éléments manquants, par rapport à la séquence \mathbf{s} . On dénommera le ratio i/N le coefficient d'éclipse.

Remarque 3.4. (*Éclipse et échantillonnage*) Un code de phase correspond à une certaine discrétisation temporelle, comme nous avons pu le voir dans la toute première définition de ce chapitre. En pratique, il est évidemment peu probable que la coupure due à une éclipse tombe parfaitement entre deux échantillons. Ce phénomène peut être corrigé par un jeu d'échantillonnage (cf. la Section 3.1), et sera donc ignoré par la suite, par simplicité.

Le traitement à la réception s'effectue comme si le signal $\mathbf{s}_{\text{ecl.}}^i$ est plein. La séquence en sortie, appelée \mathbf{y}^i et de longueur $2N - 1$, s'écrit :

$$\mathbf{y}^i := \mathbf{\Lambda}_N(\mathbf{s}) (\mathbf{s}_{\text{ecl.}}^i)^* , \quad (3.18)$$

où $\mathbf{\Lambda}_N(\mathbf{s})$ est la matrice de corrélation définie à la Remarque 2.3.

Notez que l'expression ci-dessus permet de démontrer la constance (selon le coefficient d'éclipse) du lobe principal, indépendamment de la séquence. Il suffit de poser :

$$y_N^i = \sum_{n=i+1}^N s_n s_n^* = \frac{N-i}{N}. \quad (3.19)$$

Ainsi, lorsqu'on mesure les lobes secondaires, il faut bien veiller à normaliser les corrélations par ce facteur y_N^i ... D'ailleurs, comme toute section de ce chapitre, nous pourrions conclure en invoquant le problème d'optimisation associé :

$$\begin{cases} \min_{\mathbf{s}} & f_p^i(\mathbf{s}) := \left\| \frac{1}{y_N^i} \mathbf{F} \mathbf{y}^i \right\|_p , \\ \text{t.q.} & \mathbf{s} \in \mathbf{T}^N . \end{cases} \quad (3.20)$$

Il nous faut cependant y ajouter notre petit grain de sel. En effet, le coefficient d'éclipse est inhérent de la position de la cible, et donc variable en pratique. Optimiser une séquence pour un seul coefficient n'a donc aucun sens ! Il vaut mieux considérer plusieurs cas de figure simultanément. Soit I un ensemble d'indices i — l'indice i porte sur le nombre d'éléments manquants. On a alors :

$$\begin{cases} \min_{\mathbf{s}} & \{f_p^i(\mathbf{s})\}_{i \in I} \\ \text{t.q.} & \mathbf{s} \in \mathbf{T}^N . \end{cases} \quad (3.21)$$

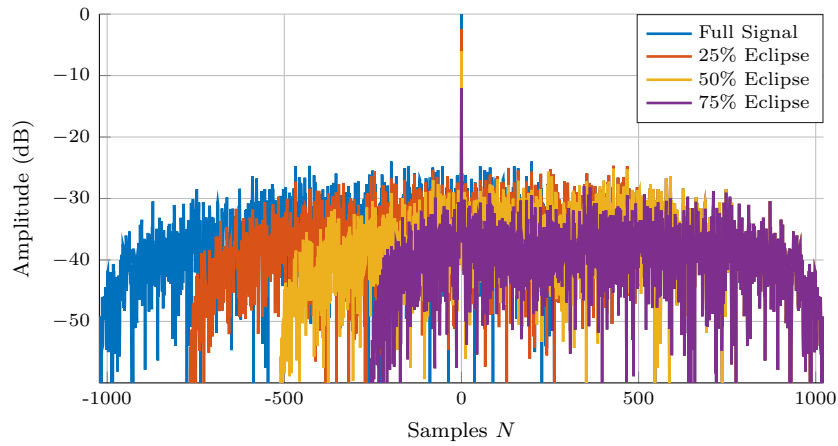
Le problème (3.21) est certes un problème d'optimisation multi-objectif, mais il n'est pas nécessaire ici de se compliquer la tâche. Contrairement à la section précédente, chaque fonction coût mesure une même quantité, à savoir le niveau des lobes secondaires. De ce fait, une simple somme non pondérée suffit à notre bonheur :

$$\begin{cases} \min_{\mathbf{s}} & \left(\sum_{i \in I} (f_p^i(\mathbf{s}))^p \right)^{1/p} , \\ \text{t.q.} & \mathbf{s} \in \mathbf{T}^N . \end{cases} \quad (3.22)$$

Les exemples suivants seront basés sur la résolution de ce problème.

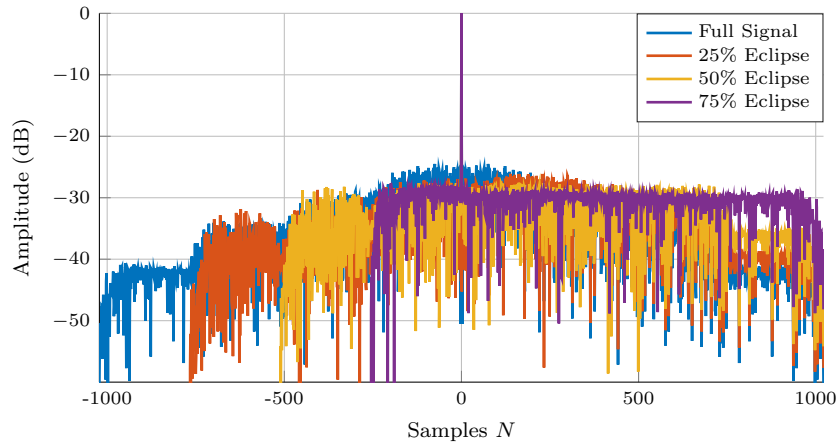
Exemple La Figure 3.11 présente les résultats d'une optimisation pour une séquence de longueur $N = 1023$ et ce, pour quatre coefficients d'éclipse : 0% (ce qui revient à la séquence complète), 25%, 50% et 75%. L'optimisation a été effectuée avec l'algorithme CMA-ES. En ce qui concerne l'initialisation, le niveau des lobes secondaires présente la même fluctuation que le coefficient d'éclipse : si l'un augmente, l'autre aussi. Dans le cas de la séquence obtenue, les lobes secondaires présentent une étonnante uniformité. Cela étant, la procédure que nous proposons apporte une certaine différence, de 4 à 11 dB, selon les optimisations, ce qui nous rassure sur son utilité éventuelle. Notons que le lobe secondaire le plus élevé (pour la séquence complète) est aux alentours des -25 dB, soit environ 18 dB d'écart par rapport à une séquence optimisée pour l'autocorrélation seule. Compromis quand tu nous tiens.

Sur cet exemple, nous nous sommes focalisés sur quatre ratios d'éclipse. Mais qu'en est-il pour d'autres ? La Figure 3.12 dépeint le PSLR selon différents coefficients d'éclipse. Logiquement, on constate une remontée des lobes secondaires pour les coefficients qui n'ont pas été pris en compte lors du processus d'optimisation. Le palier délimité par l'initialisation aléatoire reste heureusement hors d'atteinte. En revanche, l'ensemble des ratios considérés pour l'optimisation détonne...



Initialisation aléatoire.

*Les courbes ne sont pas normalisées afin d'illustrer la perte sur le lobe principal.
Faites attention lorsque vous voulez comparer les lobes secondaires.*



Séquence optimisée. Les courbes sont cette fois normalisées.

FIGURE 3.11 – Optimisation d'une séquence pour plusieurs coefficients d'éclipse

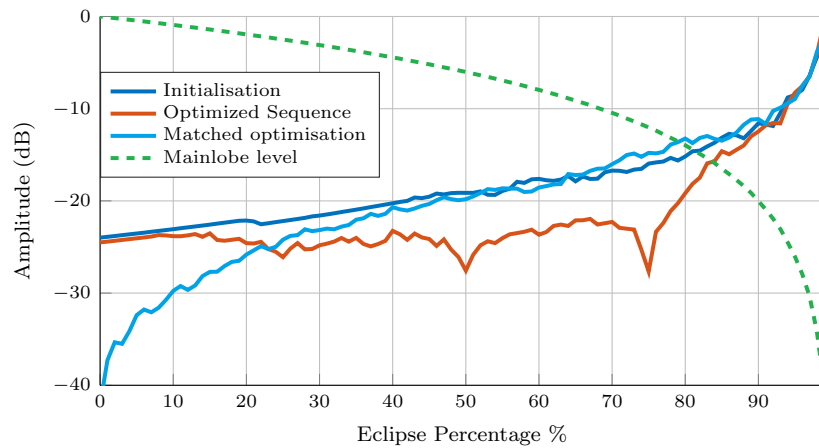


FIGURE 3.12 – Lobes secondaires selon le coefficient d'éclipse

La robustesse au phénomène d'éclipse est également dessinée pour une séquence optimisée au sens du filtre adapté.

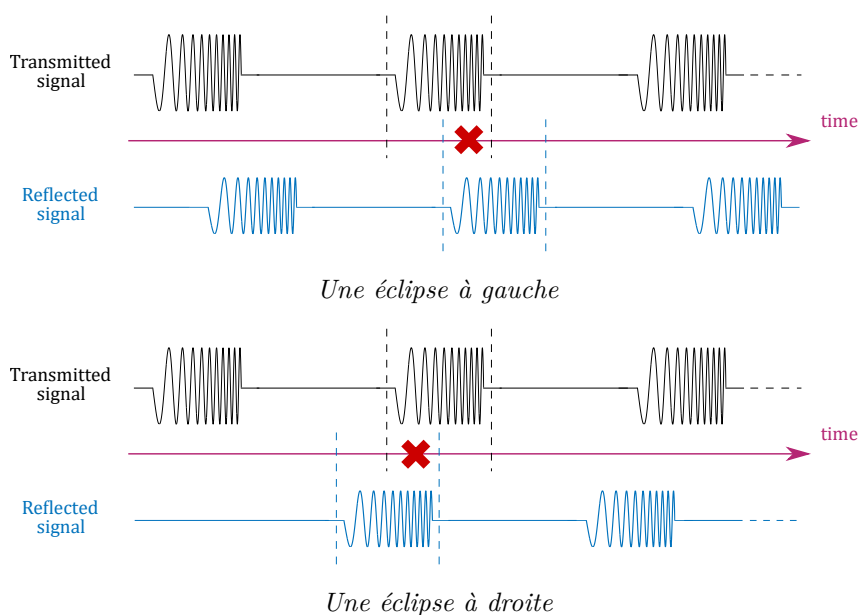


FIGURE 3.13 – Différents types d'éclipse
La perte d'information n'est pas située au même endroit...

3.3.2 ... et à droite

Dans le paragraphe précédent, nous n'avons considéré qu'un seul type d'éclipse, que l'on peut décrire à l'aide du substantif « à gauche ». En effet, il existe également une éclipse à droite (*i.e.* en fin de portée), comme l'illustre la Figure 3.13. De ce fait, il est préférable que le signal émis soit robuste dans les deux sens. Pour cela, on procède à une méthodologie similaire. Le signal reçu s'écrit :

$$\begin{aligned} \mathbf{s}_{\text{ecl.ga.}}^i &= [0, \dots, 0, s_{i+1}, \dots, s_N]^T \in \mathbb{C}^N, & (\text{éclipse à gauche}) \\ \mathbf{s}_{\text{ecl.dr.}}^j &= [s_1, \dots, s_{N-j}, 0, \dots, 0]^T \in \mathbb{C}^N, & (\text{éclipse à droite}) \end{aligned} \quad (3.23)$$

où les entiers i et j dérivent du coefficient d'éclipse.

Soient J un ensemble d'indices et g_p^j la fonction objectif associée à la séquence $\mathbf{s}_{\text{ecl.dr.}}^j$. Le problème d'optimisation qui nous intéresse est le suivant :

$$\begin{cases} \min_{\mathbf{s}} & \left(\sum_{i \in I} (f_p^i(\mathbf{s}))^p + \sum_{j \in J} (g_p^j(\mathbf{s}))^p \right)^{1/p}, \\ \text{t.q.} & \mathbf{s} \in \mathbf{T}^N. \end{cases} \quad (3.24)$$

Il n'y a bien sûr pas de pondération à ajouter : il n'y a aucune raison pour qu'une éclipse soit plus importante qu'une autre.

Afin d'éclaircir l'intérêt d'une optimisation à double sens, on se propose de comparer deux optimisations de séquences de longueur $N = 127$, une sans (une éclipse à gauche seule, Figure 3.14), une avec (Figure 3.15). Les résultats ne sont guère surprenants. Sur l'éclipse à gauche, l'optimisation « unique » prend l'avantage, avec une différence de l'ordre de 3 dB. Mais, en se spécialisant, le problème (3.22) ne contrôle en rien les lobes secondaires pour les éclipses à droite, qui s'affolent complètement. Ce n'est évidemment pas le cas de l'optimisation pour les deux types d'éclipses (3.24). Toutefois, comme mentionné, cela entraîne une légère hausse du niveau moyen des lobes secondaires.

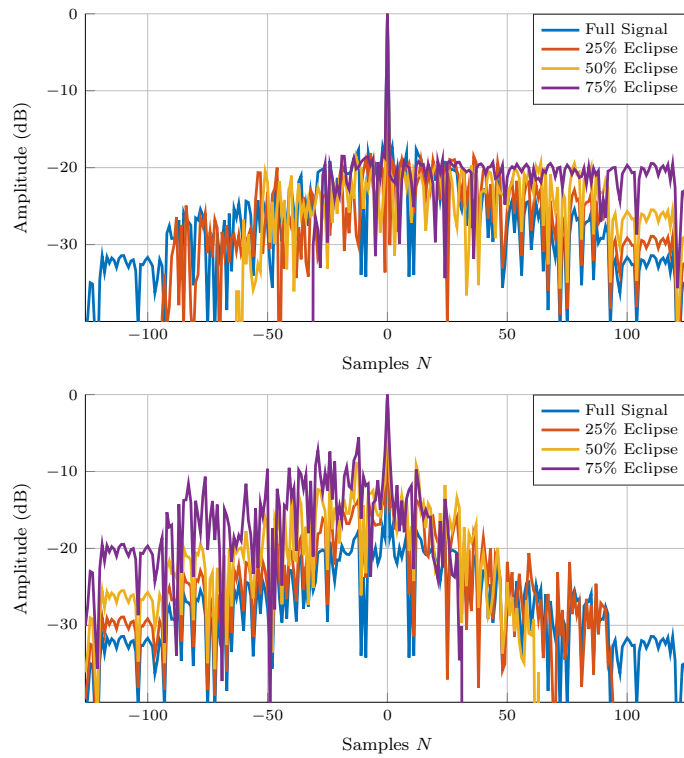


FIGURE 3.14 – Optimisation pour une éclipse à gauche seule
En haut : éclipse à gauche. En bas : éclipse à droite.

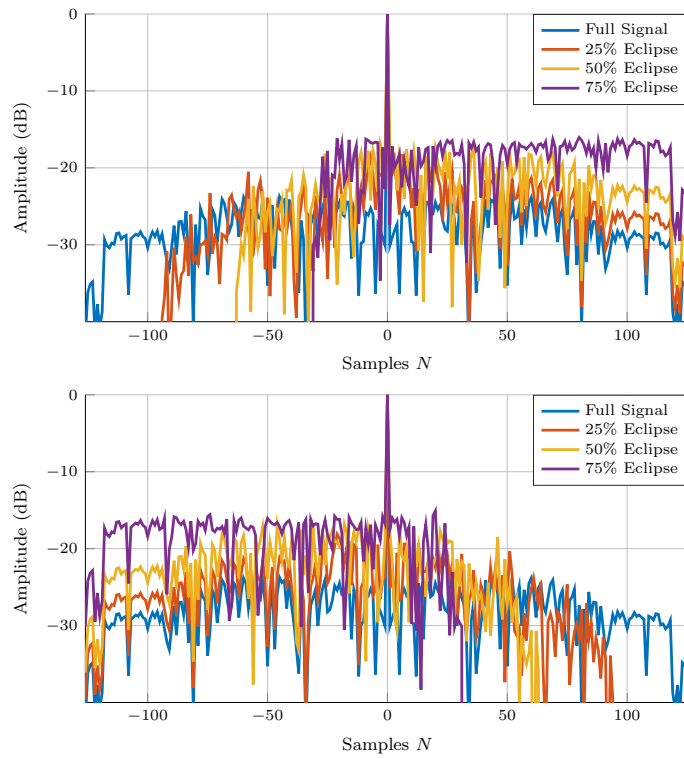


FIGURE 3.15 – Optimisation simultanée pour une éclipse à gauche et à droite
En haut : éclipse à gauche. En bas : éclipse à droite.

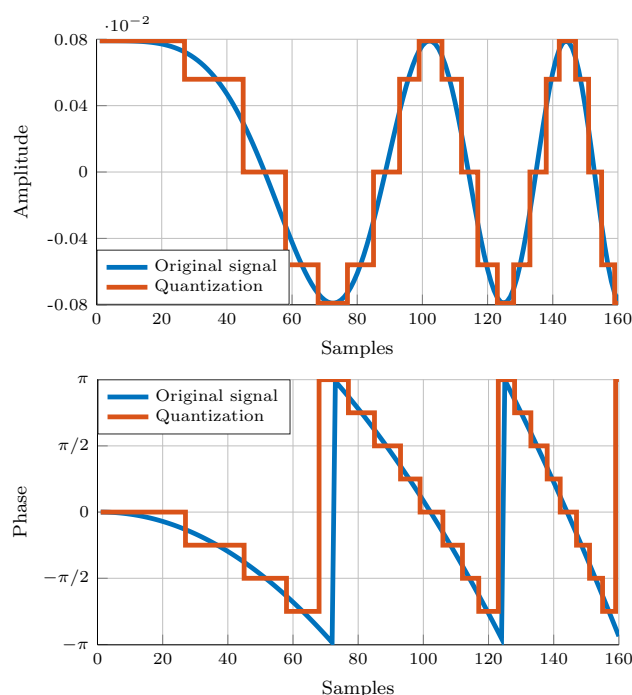


FIGURE 3.16 – Illustration de la quantification de phases
Quantification à 8 éléments de phase.

Pour obtenir un signal numérique, un signal analogique est échantillonné et quantifié.

3.4 Effets de la quantification

3.4.1 Introduction

Soit $\mathbf{s} = [s_1, \dots, s_N]^T \in \mathbf{T}^N$ une séquence polyphase de longueur N . On rappelle que la contrainte de module constant nous permet d'écrire :

$$\forall n \in \llbracket 1, N \rrbracket, \quad s_n = \frac{1}{\sqrt{N}} \exp(j\varphi_n), \quad \varphi_n \in [0, 2\pi[. \quad (3.25)$$

Les éléments de phase φ_n appartiennent à l'intervalle (continu) $[0, 2\pi[$. Cependant, pour maints facteurs — conversion analogique–numérique, déphaseurs quantifiés entre autres —, certains radars ne peuvent émettre qu'un ensemble fini de phases. Le signal émis qui en résulte n'est alors qu'une approximation de la séquence \mathbf{s} d'origine. Cette limitation est appelée « quantification ». Dans ce paragraphe, on s'intéresse donc à la quantification sur les phases. Sachez qu'il en existe également sur l'amplitude.

Comme le montre la Figure 3.16, la quantification d'une séquence consiste en un échantillonnage et une approximation de ses phases, et donc de ses éléments, d'un espace continu à un espace discret, appelé dictionnaire.

Soit n un entier. L'ensemble des racines n -ièmes de l'unité, que l'on note R_n , est généralement plébiscité en tant que dictionnaire. Il est défini par :

$$R_n = \left\{ \exp\left(\frac{2\pi jk}{n}\right), k \in \llbracket 0, n-1 \rrbracket \right\}. \quad (3.26)$$

Cet ensemble est composé de n éléments, chacun de même module. Il peut donc aisément être employé pour définir des séquences respectant la contrainte de module constant.

Qui dit approximation dit dégradation probable des performances. A titre d'illustration, on considère un code de phase de longueur $N = 127$, fruit d'une descente de gradient.

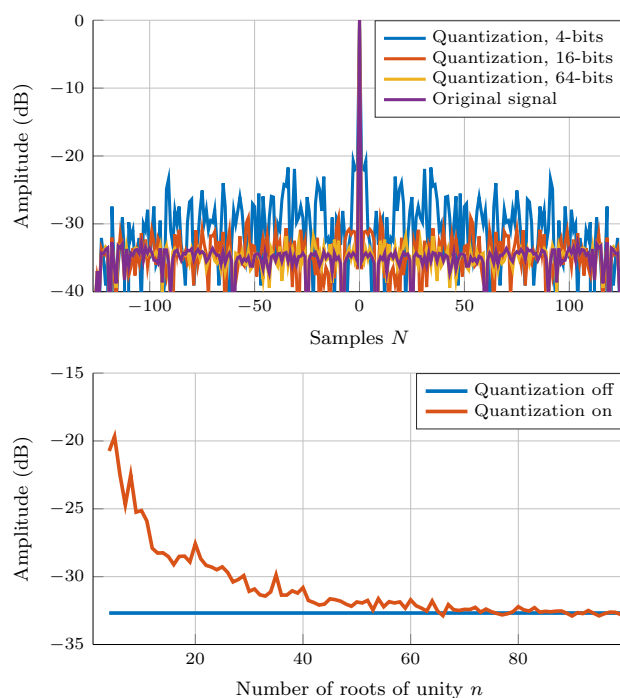


FIGURE 3.17 – Corrélation d'un signal sous plusieurs niveaux de quantification
En haut : séquences de corrélation.
En bas : PSLR en fonction du nombre de racines de l'unité n .

On quantifie ce signal à différents niveaux, en faisant varier le nombre de racines n . Les différentes séquences de corrélation sont représentées à la Figure 3.17. Cette dernière est assez prévisible : plus le nombre de bits n est élevé, mieux les lobes secondaires sont conservés. Par exemple, on constate une perte de 1 dB pour 31 phases, de 3 dB pour 17 phases. En outre, on peut voir qu'il est suffisant d'en considérer une cinquantaine, mais ce nombre est peut-être trop important en pratique...

Remarque 3.5. (*Distorsion*) La quantification introduit une certaine erreur entre le signal quantifié et le signal d'origine. Cette erreur est appelée « distorsion » : il existe d'ailleurs des études statistiques à ce sujet [Darricau, 1994]...

3.4.2 Relaxation, optimisation discrète

Le présent manuscrit est friand d'optimisation, et ce paragraphe ne doit pas faire exception. Dans l'optique de rechercher des séquences à un certain degré de quantification, deux solutions s'offrent à nous. La plus naturelle — après tout ce que nous avons vu — consiste à traiter le problème relaxé, *i.e.* sur des ensembles de recherche continus, puis à effectuer une quantification. La seconde étudie directement le problème d'optimisation discret.

3.4.2.1 Relaxation

Soit $Q_n : \mathbf{T}^N \rightarrow (R_n)^N$ un opérateur de quantification. La figure précédente a déjà évoqué les effets liés à une relaxation. Est-il possible de faire mieux ? La réponse est évidemment positive. Il suffit par exemple de modifier légèrement l'algorithme de descente de gradient, en incorporant à la dérobée le processus de quantification dans l'étape de *backtracking*. Rappelons que le *backtracking* consiste à évaluer différents candidats en réduisant le pas γ . De ce fait, on se propose d'évaluer parallèlement les quantifications de chaque candidat visité.

On ne change donc ni le déroulement de l'algorithme, ni sa convergence. Mais cette astuce nous permet de jauger plusieurs candidats, nous offrant ainsi plus de latitudes.

Algorithme Descente de gradient relaxée inefficace	
Données	Initialisation \mathbf{x}
Répéter	<ol style="list-style-type: none"> 1. Détermination d'une direction de descente $\Delta\mathbf{x}$ 2. Choix du pas $\gamma > 0$ 3. Mise à jour : $\mathbf{x} = \mathbf{x} + \gamma\Delta\mathbf{x}$ 4. Quantification : $\mathbf{x} = Q_n(\mathbf{x})$
Jusqu'à	Un critère d'arrêt est atteint.

TABLE 3.1 – Un algorithme à ne pas utiliser
Une quantification au cours du processus itératif perturbe la convergence...

Algorithme Descente de gradient relaxée	
Données	Initialisation $\mathbf{x}, \mathbf{y} := Q_n(\mathbf{x})$
Répéter	<ol style="list-style-type: none"> 1. Détermination d'une direction de descente $\Delta\mathbf{x}$ 2. <i>Backtracking</i> pour le choix du pas $\gamma > 0$. Répéter : <ul style="list-style-type: none"> – Candidat : $\mathbf{x}' = \mathbf{x} + \gamma\Delta\mathbf{x}$ – Aucune amélioration ? Réduction de γ – Candidat : $\mathbf{y}' = Q_n(\mathbf{x}')$ – Amélioration ? Mise à jour : $\mathbf{y} = \mathbf{y}'$ 3. Mise à jour : $\mathbf{x} = \mathbf{x}'$
Jusqu'à	Un critère d'arrêt est atteint.

TABLE 3.2 – Un algorithme à privilégier
... tandis qu'étudier la version relaxée de tous les candidats rencontrés semble offrir de meilleurs résultats.

Afin de confirmer nos dires, les Figures 3.18 et 3.19 comparent les résultats d'une quantification directe et des deux méthodologies présentées ci-dessus. En faisant varier le nombre d'éléments du dictionnaire n ou la longueur de la séquence N , on peut constater un potentiel écart entre la première méthode, la quantification de la séquence d'origine et la seconde méthode, souvent en faveur de cette dernière. Mais, malgré ces améliorations, l'écart reste néanmoins important avec la séquence d'origine.

Cela était déjà visible à la Figure 3.17, mais on l'aperçoit encore mieux ici : le degré de quantification ne va pas nécessairement dans le même sens que le niveau des lobes secondaires (le PSLR), notamment pour les petites valeurs de n . La tendance qui s'y dégage « plus le nombre de bits n est élevé, mieux les lobes secondaires sont conservés », prononcée une page plus haut, reste néanmoins vraie.

3.4.2.2 Une optimisation discrète

Effectuer une relaxation a l'avantage d'être peu coûteux en temps de calcul. Mais cela n'est pas sans défaut. D'une part, une relaxation n'offre aucune indication sur la nature de la solution obtenue (minimum local...). D'autre part, l'aspect discret n'est nullement exploité. Ainsi, essayons maintenant de résoudre directement le problème d'optimisation discret. Celui-ci s'écrit (la différence se situe dans l'espace de recherche) :

$$\left\{ \begin{array}{l} \min_{\mathbf{s}} f_p(\mathbf{s}), \\ \text{t.q. } \mathbf{s} \in \mathbf{T}^N, \end{array} \right. \implies \left\{ \begin{array}{l} \min_{\mathbf{s}} f_p(\mathbf{s}), \\ \text{t.q. } \mathbf{s} \in (R_n)^N. \end{array} \right. \quad (3.27)$$

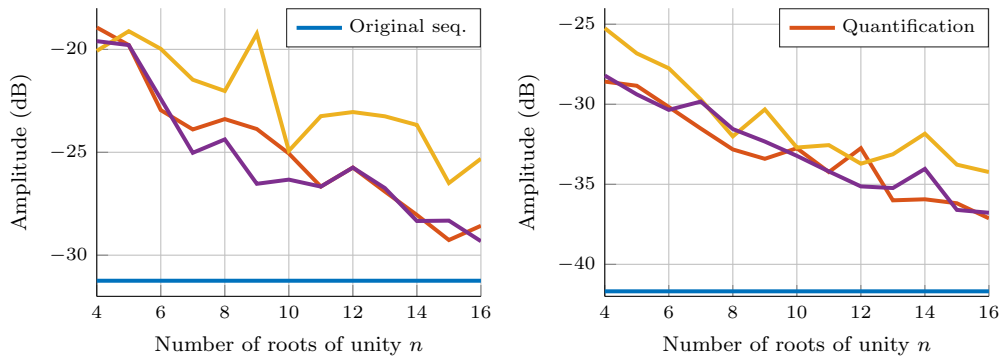


FIGURE 3.18 – Performances de la quantification selon le nombre de racines de l'unité
A gauche : séquence de longueur $N = 115$. A droite : séquence de longueur $N = 1024$.
Les légendes des Figures 3.18 et 3.19 sont communes.

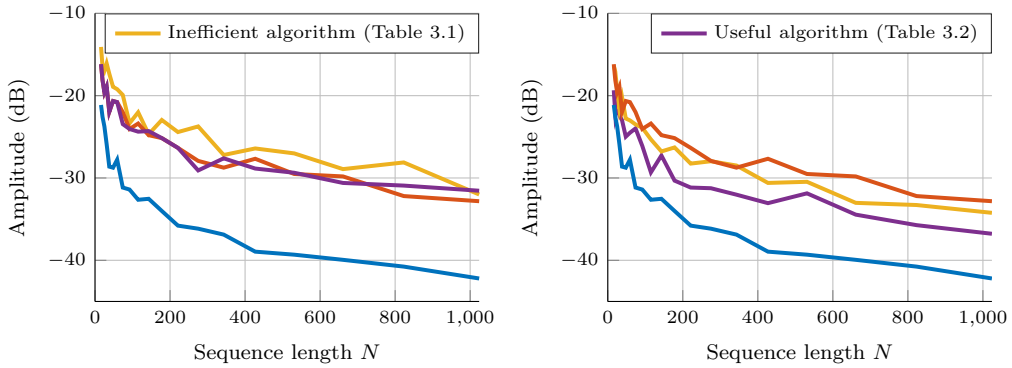


FIGURE 3.19 – Performances de la quantification selon la longueur de séquence
A gauche : quantification de 8 bits. A droite : quantification de 16 bits.
Les légendes des Figures 3.18 et 3.19 sont communes.

Le problème initial est non convexe et donc compliqué à résoudre. D'ailleurs, malgré ces quelques pages, nous n'y sommes littéralement pas parvenus. Son homologue n'est pas mieux loti : il est *NP-difficile* [Cygan *et al.*, 2017], comme tout problème de minimisation discret non linéaire [Hemmecke *et al.*, 2010]. Cela signifie qu'il n'existe pas — à ce jour, et jusqu'à preuve du contraire — d'algorithme en temps polynomial permettant de déterminer si une séquence est un minimum global. Nous ne sommes guère plus avancés...

Ceci dit, rien ne nous empêche d'employer les méthodologies d'optimisation discrète. Évaluer l'ensemble des possibilités est bien sûr proscrit, à cause de la dimension du problème (*cf.* la Section 2.1.5). On se propose alors d'appliquer les algorithmes les plus courants pour les programmes non linéaires : le *branch-and-bound* (ou « séparation et évaluation » en français). Il est difficile d'expliquer ces derniers en quelques mots de manière rigoureuse, tant cela requiert de nombreux pré-requis, notamment en théorie des graphes.

Ici, on se contentera donc de quelques descriptions et d'un schéma explicatif. Pour plus de détails, nous vous invitons à jeter un coup d'œil à [Korte *et al.*, 2006]. L'algorithme *branch-and-bound* est bien défini en ces termes.

- Le « branchement » ou la « séparation » évoque le fait de partitionner un ensemble de solutions réalisables en plusieurs sous-ensembles, non vides.
- Pour chaque sous-ensemble obtenu par branchement, une « borne » inférieure sur la fonction objectif peut être définie pour les éléments de ce sous-ensemble.

Le premier point sous-entend une réorganisation des solutions réalisables sous la forme d'un arbre, où chaque « nœud » pourrait correspondre à un élément. Ici, une petite subtilité s'impose : chaque nœud représente en fait un *couple* d'éléments. Pour expliquer cela, il suffit de reprendre l'expression de la séquence d'autocorrélation, sur laquelle est basée notre fonction objectif :

$$\Lambda_{2N-1}(\mathbf{s})\mathbf{s}^* = \begin{bmatrix} s_N s_1^* \\ s_{N-1} s_1^* + s_N s_2^* \\ \vdots \\ s_1 s_{N-1}^* + s_2 s_N^* \\ s_1 s_N^* \end{bmatrix}. \quad (3.28)$$

Le premier élément de la séquence de corrélation (et, accessoirement, le dernier) s'exprime à l'aide de deux éléments, s_1 et s_N . Pour le deuxième élément de la séquence de corrélation, on y *ajoute* en plus s_2 et s_{N-1} . Et ainsi de suite. Au bout du compte, lorsqu'on construira cet arbre, on commencera par définir le couple (s_1, s_N) , puis le couple (s_2, s_{N-1}) , comme le dépeint la Figure 3.20. Notez que, si la longueur de la séquence \mathbf{s} est impaire, les feuilles (les nœuds les plus profonds de l'arbre) ne caractérisent qu'un seul élément.

Ensuite, il faut explorer cet arbre. Pas en sa totalité, car cela revient à étudier l'ensemble des solutions réalisables, ce qui est clairement trop coûteux. Par conséquent, il faut un moyen d'éviter certaines séquences : ceci est le rôle de la borne inférieure. Chaque nœud — qui définit une solution partielle — est ainsi évalué et comparé à cette borne. Si cette dernière est inférieure, il est inutile de suivre les séquences qui découlent de ce nœud, car elles présenteront nécessairement une valeur de fonction coût plus élevée. Reste donc à définir cette borne : le plus simple est de considérer la valeur d'une séquence, tirée aléatoirement ou à l'aide de méthodologies comme une descente de gradient relaxée...

Finalement, l'algorithme de *branch-and-bound* se déroule de la façon suivante :

- La borne est initialisée à l'aide d'une certaine séquence (aléatoire, par exemple).
- On parcourt l'arbre, en profondeur de préférence (et non pas en largeur, c'est-à-dire d'étage en étage). On évalue la séquence partielle induite par chaque nœud, et on ébranche si nécessaire (on ignore les solutions suivantes provenant de ce nœud).
- Lorsqu'une feuille est atteinte, on obtient une séquence complète. Si amélioration il y a, la borne est mise à jour en conséquence.
- On s'arrête lorsque toutes les feuilles restantes ont été visitées, ou bien lorsqu'on le désire, tout simplement. Cela est pratique (utilisation en temps réel), mais on ne saura en aucun cas si l'optimum a été atteint.

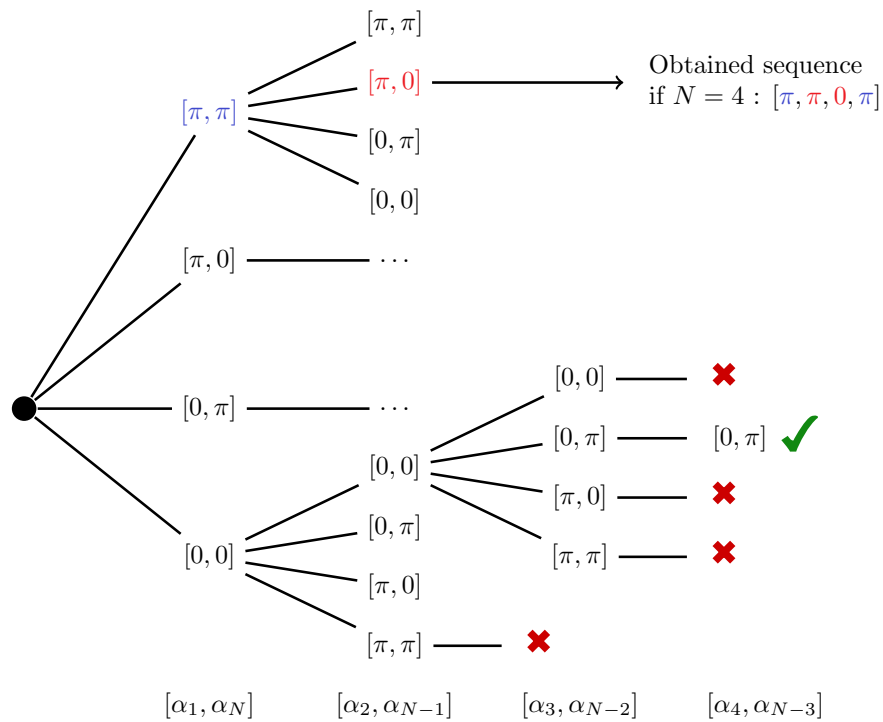


FIGURE 3.20 – Représentation des éléments de l'ensemble $(R_n)^N$ sous la forme d'un arbre

En haut : un exemple de séquence pour $N = 4$.

En bas : le déroulement de l'algorithme pour l'exemple présenté dans le corps du texte.

Pour plus d'esthétique, ce ne sont pas les éléments de $(R_n)^N$ qui composent les nœuds, mais leurs phases.

Exemple À titre d'illustration, considérons l'exemple jouet suivant. Supposons que l'on souhaite optimiser, au sens du PSLR, une séquence binaire de longueur $N = 8$. La liste suivante essaie de décrire au mieux l'exploration de l'arbre, de manière très laconique. L'exemple est également représenté à la Figure 3.20.

- Initialisation choisie : $\alpha = [0, 0, 0, 0, \pi, 0, 0, 0]$. Borne : $g_\infty(\alpha) = 3/8$.
- Exploration. Nœud courant : $(0, 0)$.
Séquence de phases : $\alpha' = [0, ?, ?, ?, ?, ?, ?, 0]$.
Séquence de corrélation : $[1/8, ?, \dots, ?, 1/8]$.
Évaluation : $g_\infty(\alpha') = 1/8 < g_\infty(\alpha)$. On continue donc l'exploration en profondeur.
Remarquez qu'on n'élaguera jamais à l'issue du premier étage...
- Exploration. Nœud précédent : $(0, 0)$. Nœud courant : $(0, 0)$.
Séquence de phases : $\alpha' = [0, 0, ?, ?, ?, ?, 0, 0]$.
Évaluation : $g_\infty(\alpha') = 2/8 < g_\infty(\alpha)$. On continue l'exploration.
- Exploration. Nœuds précédents : $(0, 0) - (0, 0)$. Nœud courant : $(0, 0)$.
Séquence de phases : $\alpha' = [0, 0, 0, ?, ?, 0, 0, 0]$.
Évaluation : $g_\infty(\alpha') = 3/8 \not< g_\infty(\alpha)$. On élague. Cela n'est pas étonnant, puisqu'on risque d'atteindre la séquence à l'origine de notre borne...
- On poursuit alors l'exploration, mais au niveau du même étage.
Exploration. Nœuds précédents : $(0, 0) - (0, 0)$. Nœud courant : $(0, \pi)$.
Séquence de phases : $\alpha' = [0, 0, 0, ?, ?, \pi, 0, 0]$.
Évaluation : $g_\infty(\alpha') = 2/8$. On continue l'exploration.
- On a atteint le dernier étage. Les quatre séquences qui la composent sont évaluées.
Parmi elles, on a $\alpha' = [0, 0, 0, \pi, 0, \pi, 0, 0]$.
Évaluation : $g_\infty(\alpha') = 2/8 < g_\infty(\alpha)$.
La séquence est retenue, et la borne est mise à jour : $\alpha = \alpha'$.
- On remonte d'un étage, et on reprend notre exploration.
Exploration. Nœuds précédents : $(0, 0) - (0, 0)$. Nœud courant : $(\pi, 0)$.
Séquence de phases : $\alpha' = [0, 0, \pi, ?, ?, 0, 0, 0]$.
Évaluation : $g_\infty(\alpha') = 2/8 \not< g_\infty(\alpha)$. On élague. Remarquez que nous avons choisi de considérer les inégalités strictes, car nous ne souhaitons pas nécessairement tous les optima...
- Et ainsi de suite, jusqu'à la dernière feuille.

On s'arrêtera là pour cet exemple. En effet, la séquence de phases α que nous avons trouvée se trouve être un des optima...

La boucle est bouclée ? Appliquons maintenant la méthodologie pour une séquence de longueur... Ah non, en fait. Bien que connus, au jour d'aujourd'hui, pour être efficaces sur de nombreux problèmes *NP-difficile* [Korte *et al.*, 2006], les algorithmes de *branch-and-bound* demeurent néanmoins beaucoup trop coûteux pour notre problème.

Pourquoi ? Dans la pratique — et cela peut même se voir dans notre exemple pathologique — les étapes d'ébranchages ne s'effectuent qu'au niveau des derniers étages. Cela signifie que la majorité des éléments de l'espace de départ sera visité, ce que nous voulons éviter. La faute, sans doute, à beaucoup de paramètres : une fonction objectif difficilement séparable, l'expression de la séquence d'autocorrélation qui ne se dévoile qu'en ses feuilles et rend la structure arborescente inutilisable, une borne pas assez informative pour les séquences partielles, un mauvais alignement des astres... Ou tout simplement parce que le choix des paramètres ne convient pas.

Résoudre un problème d'optimisation discret évoque les premières recherches sur les séquences orthogonales, les m -séquences, les codes de Barker, les codes de Gold *etc.* comme celles que nous avons pu voir dans le chapitre précédent. On comprend d'ailleurs pourquoi ces derniers ont été découverts en exploitant des astuces (sur la théorie des groupes, notamment), et non en résolvant des problèmes d'optimisation.

Après toutes ces études, on s'aperçoit que les séquences multiphases restent suffisamment compétentes (mais légèrement en retrait) par rapport aux codes polyphases. Mais cela, pour le cas mono-séquence...

3.5 Orthogonalité d'un ensemble de séquences

A ce stade du chapitre, nous nous sommes intéressés uniquement à la corrélation d'une seule séquence. Cependant, cette thèse est intitulée « Mise en place de méthodologies de conception de formes d'onde pour radars sols — Application au cas du radar MIMO ». Le terme MIMO suggère l'utilisation d'un ensemble de séquences. Ainsi, on se propose dans cette section de rechercher des ensembles pseudo-orthogonaux. Nous verrons qu'il est en fait relativement aisé d'étendre les travaux des précédentes sections...

3.5.1 Mise en place

Fonctions de corrélation Afin d'effectuer un parallèle avec l'étude précédente, commençons par rappeler quelques définitions, provenant de la Section 2.1.2. Soient $s \in L^2(\mathbb{C})$ et $l : L^2(\mathbb{C}) \rightarrow L^2(\mathbb{C})$ un opérateur « inverse », *i.e.* $l(s(t)) := s(-t)$ pour tout $t \in \mathbb{R}$. La fonction d'autocorrélation est définie comme suit.

$$\left\{ \begin{array}{l} \text{ACF} : L^2(\mathbb{C}) \rightarrow L^2(\mathbb{C}), \\ s \mapsto \text{ACF}(s) := s * l^*(s), \\ \text{ACF}(s) : \mathbb{R} \rightarrow \mathbb{C}, \\ \tau \mapsto \text{ACF}(s)(\tau) := \int_{-\infty}^{+\infty} s(t)s^*(t-\tau)dt. \end{array} \right. \quad (3.29)$$

Il existe également une version discrète. Pour une séquence $\mathbf{s} = [s_1, \dots, s_N]^T$, on a :

$$\begin{aligned} \text{AACF} : \mathbf{T}^N &\rightarrow \mathbb{C}^{2N-1}, \\ \mathbf{s} \mapsto (\text{AACF}(\mathbf{s}))_k &= \sum_{n=1}^N s_n s_{n+k-N}^*, \quad (1 \leq k \leq 2N-1), \end{aligned} \quad (3.30)$$

où, par convention, la séquence \mathbf{s} est nulle en dehors de son support principal.

La fonction d'autocorrélation mesure, en quelque sorte, la ressemblance entre un signal et une réplique décalée dans le temps. De la même manière, la fonction d'*inter-corrélation*, notée ci-après CCF pour *Cross-Correlation Function*, est tout simplement la comparaison entre deux signaux. Soit $q \in L^2(\mathbb{C})$.

$$\left\{ \begin{array}{l} \text{CCF} : L^2(\mathbb{C}) \times L^2(\mathbb{C}) \rightarrow L^2(\mathbb{C}), \\ (s, q) \mapsto \text{CCF}(s, q) := s * l(q), \\ \text{CCF}(s, q) : \mathbb{R} \rightarrow \mathbb{C}, \\ \tau \mapsto \text{CCF}(s, q)(\tau) := \int_{-\infty}^{+\infty} s(t)q^*(t-\tau)dt. \end{array} \right. \quad (3.31)$$

Soit $\mathbf{q} = [q_1, \dots, q_N]^T$. La version discrète est également jumelle.

$$\begin{aligned} \text{ACCF} : \mathbf{T}^N \times \mathbf{T}^N &\rightarrow \mathbb{C}^{2N-1}, \\ (\mathbf{s}, \mathbf{q}) &\mapsto (\text{ACCF}(\mathbf{s}, \mathbf{q}))_k = \sum_{n=1}^N s_n q_{n+k-N}^*, \quad (1 \leq k \leq 2N-1). \end{aligned} \quad (3.32)$$

Remarquons qu'on a également $\text{ACCF}(\mathbf{s}, \mathbf{q}) = \Lambda_{2N-1}(\mathbf{s})\mathbf{q}^*$. Pour plus de simplicité — et de cohérence avec ce qui suivra — nous avons considéré l'inter-corrélation pour deux séquences de même longueur. Cela n'est pas forcément nécessaire ! La fonction d'inter-corrélation n'est finalement qu'une généralisation. En effet, remplacer q par s dans les définitions ci-dessus nous offre la fonction d'autocorrélation...

Lobes secondaires La fonction d'autocorrélation a déjà été intronisée dans ce manuscrit lorsqu'on a évoqué le traitement radar. Une bonne forme d'onde se doit d'exhiber une forte réponse impulsionnelle, en d'autres termes un bon ratio entre le lobe principal et les lobes secondaires. Voilà pourquoi on souhaite des séquences d'autocorrélation ressemblant à un Dirac (une valeur forte unique).

Lorsqu'on a plusieurs signaux, on souhaite également être capable de les reconnaître, mais aussi de les dissocier. Reconnaître, c'est l'autocorrélation ; dissocier, c'est l'inter-corrélation. En ce sens, il faut les corrélations croisées les plus faibles, avec un profil plat en somme. D'ailleurs, ces dernières ne produisent aucun lobe principal, mais « que » des secondaires. Il nous reste donc à définir une mesure pour ces lobes secondaires. Nous n'allons pas faire dans l'original. Cependant, il faut simplement veiller à ne pas oublier le cas particulier où la fonction d'inter-corrélation est égale à la fonction d'autocorrélation, car cette dernière doit épargner le lobe principal. Pour cela, on commence par extrapoler dangereusement le symbole de Kronecker. Pour deux séquences \mathbf{s} et \mathbf{q} , on pose :

$$\delta_{\mathbf{s}, \mathbf{q}} := \begin{cases} 1 & \text{si } \mathbf{s} = \mathbf{q}, \\ 0 & \text{sinon.} \end{cases} \quad (3.33)$$

Comme auparavant, soit $\mathbf{F}_{\mathbf{s}, \mathbf{q}}$ une matrice diagonale de taille $2N-1$, spécifiée par le vecteur $[1, \dots, 1, 1 - \delta_{\mathbf{s}, \mathbf{q}}, 1, \dots, 1]$ (un vecteur composé uniquement de 1, sauf à la position N , représentant l'éventuel lobe principal, dans le cas de l'autocorrélation). Avec un petit abus de notation, on peut suggérer une mesure générique des lobes secondaires pour une paire de séquences (\mathbf{s}, \mathbf{q}) :

$$h_p(\mathbf{s}, \mathbf{q}) := \left\| \mathbf{F}_{\mathbf{s}, \mathbf{q}} \text{ACCF}(\mathbf{s}, \mathbf{q}) \right\|_p. \quad (3.34)$$

Lorsque $\mathbf{s} = \mathbf{q}$, on retrouve la mesure introduite pour une séquence d'autocorrélation.

Remarque 3.6. (*Orthogonalité et orthogonalité*) Dans ce manuscrit, l'orthogonalité entre deux séquences se caractérise par une séquence d'inter-corrélation nulle. Quelques articles dans la littérature (par exemple, [Xiang et al., 2015]) définissent l'orthogonalité comme un produit scalaire nul. Il faut donc faire attention !

Un problème d'optimisation Avec toutes les cartes en main, nous pouvons déterminer un problème d'optimisation dont le but est de rechercher une famille de séquences quasi-orthogonales. Soit $S = \{\mathbf{s}^1, \dots, \mathbf{s}^M\}$ un ensemble de M séquences de même longueur N . La minimisation qui nous intéresse est la suivante :

$$(P_5) \begin{cases} \min_S & f_p(S) = \left(\sum_{m_1, m_2=1}^M \left(h_p(\mathbf{s}^{m_1}, \mathbf{s}^{m_2}) \right)^p \right)^{1/p}, \\ \text{t.q.} & \mathbf{s}^m \in \mathbf{T}^N, \quad \forall m \in \llbracket 1, M \rrbracket. \end{cases} \quad (3.35)$$

La fonction f_p est relativement compréhensible : elle cumule les lobes secondaires pour toutes les paires de séquences $(\mathbf{s}^{m_1}, \mathbf{s}^{m_2})$ de l'ensemble S . Remarquez qu'on a la décomposition suivante :

$$f_p(S) = \left(\sum_{m=1}^M \left(h_p(\mathbf{s}^m, \mathbf{s}^m) \right)^p + \sum_{\substack{m_1, m_2=1 \\ m_1 \neq m_2}}^M \left(h_p(\mathbf{s}^{m_1}, \mathbf{s}^{m_2}) \right)^p \right)^{1/p}, \quad (3.36)$$

où la première somme correspond aux lobes secondaires de l'autocorrélation, et la seconde aux lobes secondaires de l'inter-corrélation.

Malheureusement, rien n'a changé : le problème est toujours non convexe, avec un nombre de variables accru à cause de la cardinalité des séquences... En se basant sur les conclusions de la Section 2.2, la sous-partie suivante présentera les résultats obtenus à l'aide d'une descente de gradient. Mais, pour l'appliquer, il faut effectuer avant un petit calcul de gradient.

Un petit calcul Comme vu à l'équation 3.36, il est possible de décomposer la fonction coût en deux termes : l'un représentant l'autocorrélation, l'autre l'inter-corrélation. Le gradient du premier a déjà été calculé précédemment (*cf.* la Section 2.2.2.2); reste donc à calculer celui du second. Afin de simplifier les choses, on s'occupera de la fonction intermédiaire h_p . Soient $\mathbf{s}, \mathbf{q} \in \mathbf{T}^N$ deux séquences différentes, et $\boldsymbol{\alpha}, \boldsymbol{\beta}$ leurs vecteurs de phases respectifs. On pose $\mathbf{r} = \text{ACCF}(\mathbf{s}, \mathbf{q})$.

A l'aide du théorème de dérivation des fonctions composées [Hjørungnes, 2011], nous pouvons définir le gradient partiel de h_p selon la phase $\boldsymbol{\alpha}$ à partir des dérivées partielles selon les parties réelles et imaginaires de \mathbf{s} :

$$\forall n \in \llbracket 1, N \rrbracket, \quad \frac{\partial h_p(\mathbf{s}, \mathbf{q})}{\partial \alpha_n} = \Re(s_n) \frac{\partial h_p(\mathbf{s}, \mathbf{q})}{\partial \Re(s_n)} - \Im(s_n) \frac{\partial h_p(\mathbf{s}, \mathbf{q})}{\partial \Im(s_n)}. \quad (3.37)$$

Il nous faut donc expliciter les dérivées partielles du terme de droite. Pour cela, mettons en évidence la présence de $\Re(s_n)$ et $\Im(s_n)$ dans l'expression (3.34) :

$$h_p(\mathbf{s}, \mathbf{q}) = h(\mathbf{s}, \mathbf{q})^{1/p} \text{ avec } h(\mathbf{s}, \mathbf{q}) := \|\mathbf{r}\|_p^p = \sum_{k=1}^{2N-1} \left(\Re(r_k)^2 + \Im(r_k)^2 \right)^{p/2}. \quad (3.38)$$

Cela nous permet d'écrire (les dérivées selon les parties imaginaires se définissent de manière similaire) :

$$\begin{cases} \frac{\partial h_p(\mathbf{s}, \mathbf{q})}{\partial \Re(s_n)} = \frac{1}{p} \frac{\partial h(\mathbf{s}, \mathbf{q})}{\partial \Re(s_n)} h(\mathbf{s}, \mathbf{q})^{1/p-1}, \\ \frac{\partial h(\mathbf{s}, \mathbf{q})}{\partial \Re(s_n)} = p \sum_{k=1}^{2N-1} \left(\Re(r_k) \frac{\partial \Re(r_k)}{\partial \Re(s_n)} + \Im(r_k) \frac{\partial \Im(r_k)}{\partial \Re(s_n)} \right) \left(\Re(r_k)^2 + \Im(r_k)^2 \right)^{p/2-1}. \end{cases} \quad (3.39)$$

Repartons de (3.32), et décomposons-le :

$$\begin{cases} \Re(r_k(\mathbf{s}, \mathbf{q})) = \sum_{n=1}^N \Re(s_n) \Re(q_{n+k-N}) + \Im(s_n) \Im(q_{n+k-N}), \\ \Im(r_k(\mathbf{s}, \mathbf{q})) = \sum_{n=1}^N -\Re(s_n) \Im(q_{n+k-N}) + \Im(s_n) \Re(q_{n+k-N}), \end{cases} \quad (3.40)$$

ce qui entraîne les dérivées partielles suivantes :

$$\begin{aligned} \frac{\partial \Re(r_k)}{\partial \Re(s_n)} &= \Re(q_{n+k-N}), & \frac{\partial \Re(r_k)}{\partial \Im(s_n)} &= \Im(q_{n+k-N}), \\ \frac{\partial \Im(r_k)}{\partial \Re(s_n)} &= -\Im(q_{n+k-N}), & \frac{\partial \Im(r_k)}{\partial \Im(s_n)} &= \Re(q_{n+k-N}). \end{aligned} \quad (3.41)$$

En insérant (3.41) dans (3.39), nous obtenons :

$$\begin{aligned}\frac{\partial h(\mathbf{s}, \mathbf{q})}{\partial \Re(s_n)} &= p \sum_{k=1}^{2N-1} |r_k|^{p-2} \left(\Re(r_k) \Re(q_{n+k-N}) - \Im(r_k) \Im(q_{n+k-N}) \right) \\ &= p \sum_{k=1}^{2N-1} |r_k|^{p-2} \Re(r_k q_{n+k-N}), \\ \frac{\partial h(\mathbf{s}, \mathbf{q})}{\partial \Im(s_n)} &= p \sum_{k=1}^{2N-1} |r_k|^{p-2} \Im(r_k q_{n+k-N}).\end{aligned}\tag{3.42}$$

Enfin, nous rassemblons l'ensemble de ces résultats pour obtenir :

$$\frac{\partial h_p(\mathbf{s}, \mathbf{q})}{\partial \alpha_n} = - \sum_{k=1}^{2N-1} \frac{|r_k|^{p-2}}{\|\mathbf{r}\|_p^{p-1}} \Im\left(s_n (r_k q_{n+k-N})^*\right).\tag{3.43}$$

De la même manière, on trouve :

$$\frac{\partial h_p(\mathbf{s}, \mathbf{q})}{\partial \beta_n} = - \sum_{k=1}^{2N-1} \frac{|r_k|^{p-2}}{\|\mathbf{r}\|_p^{p-1}} \Re\left(s_n r_k (q_{n-k+N})^*\right).\tag{3.44}$$

3.5.2 Quelques résultats

Terminons ce paragraphe avec quelques résultats applicatifs. La résolution du problème d'optimisation (P_5) nous offre *a priori* des familles de séquences quasi-orthogonales. Sachez avant toute chose qu'il en existe certaines, prédéterminées dans la littérature, à l'instar des codes de Gold [Gold, 1967] ou de Kasami [Kasami, 1966] (mais ces codes ont été optimisés pour le cas des corrélations périodiques). Une liste est d'ailleurs disponible ici [Arlery *et al.*, 2015].

L'ensemble des résultats qui suivent ont été obtenus à l'aide d'une descente de gradient, comme mentionné précédemment. Le but de ce paragraphe est d'exhiber des performances atteignables en pratique ; il n'y aura donc aucun débat sur la convergence, car celle-ci est probablement — et au mieux — locale ! Commençons ainsi par l'optimisation d'un ensemble de deux séquences $S = \{\mathbf{s}^1, \mathbf{s}^2\}$ de longueur $N = 127$. En cet ensemble, trois séquences sont à considérer :

- les deux séquences d'autocorrélation associées à \mathbf{s}^1 et \mathbf{s}^2 ;
- la séquence d'inter-corrélation entre ces dernières.

Tout cela est représenté à la Figure 3.21. Outre le gain d'environ 10 dB par rapport à une initialisation aléatoire, on peut s'apercevoir que le niveau des lobes secondaires est plus ou moins identique pour chacune des corrélations. En même temps, c'est ce qu'on a voulu : on a considéré un problème d'optimisation mono-objectif et non pondéré... Les séquences ainsi obtenues sont dites *pseudo-orthogonales*.

Sur les Figures 3.22 et 3.23, on se propose d'anticiper les performances qu'on peut atteindre. Pour cela, on procède à une cinquantaine d'instances d'optimisation et, pour chacune d'entre elles, on fait varier le nombre de séquences M et leur longueur N . Permettons-nous deux observations et/ou pressentiments :

- doubler la longueur de séquence permet de gagner 3 dB ;
- le niveau des lobes secondaires semble proportionnel à l'inverse du nombre de séquences de l'ensemble.

Mais tout cela n'est que supputations... En ce sens, il existe des bornes analytiques sur le PSLR, dépendant des paramètres M et N . Ceci est le sujet de la prochaine section !

*Cette sous-section a fait l'objet d'une publication [Tan *et al.*, 2016b].*

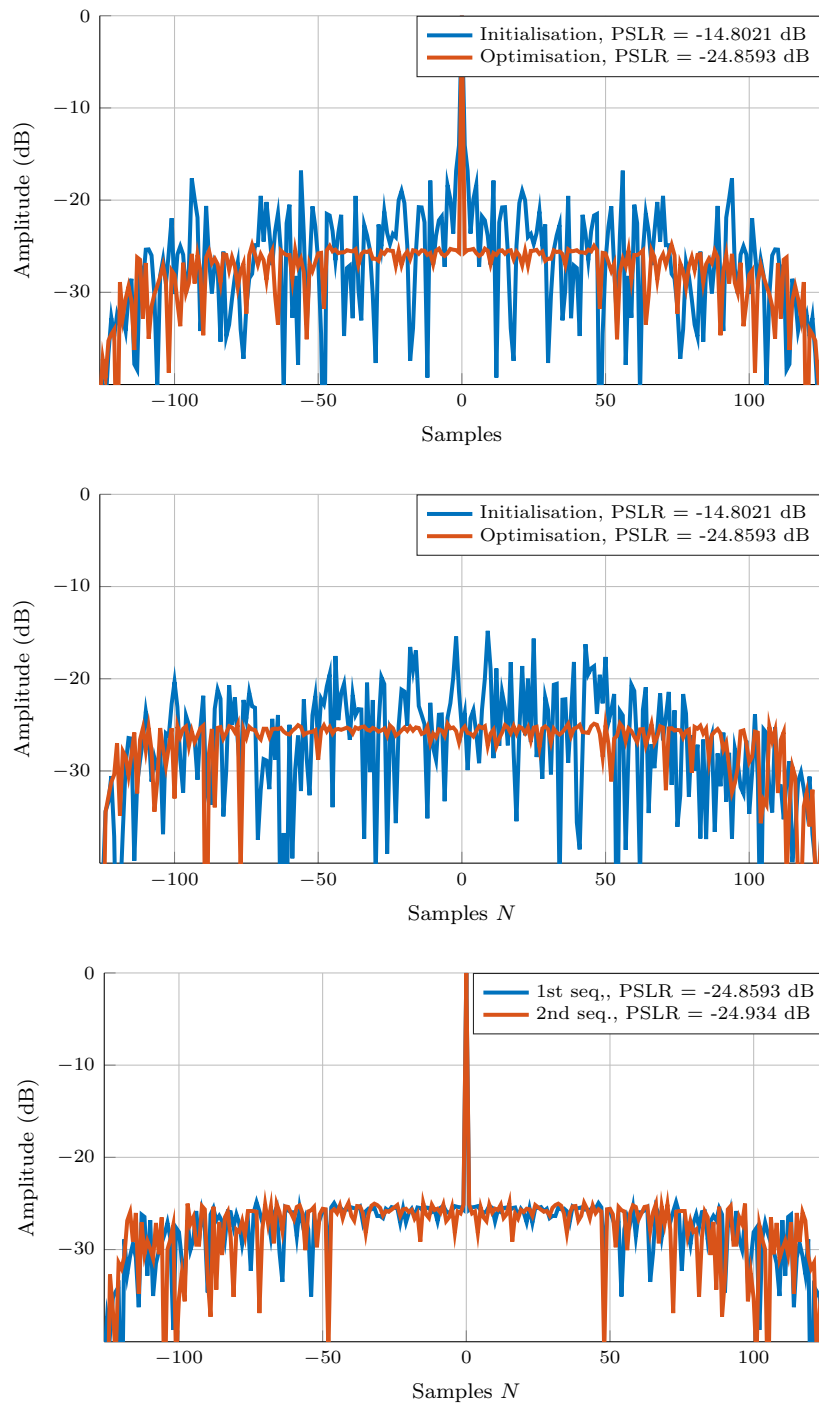


FIGURE 3.21 – Optimisation d'un ensemble de $M = 2$ séquences de longueur $N = 127$

En haut : autocorrélation de la 1ère séquence.

Au centre : inter-corrélation entre les deux séquences.

En bas : autocorrélation des deux séquences optimisées.

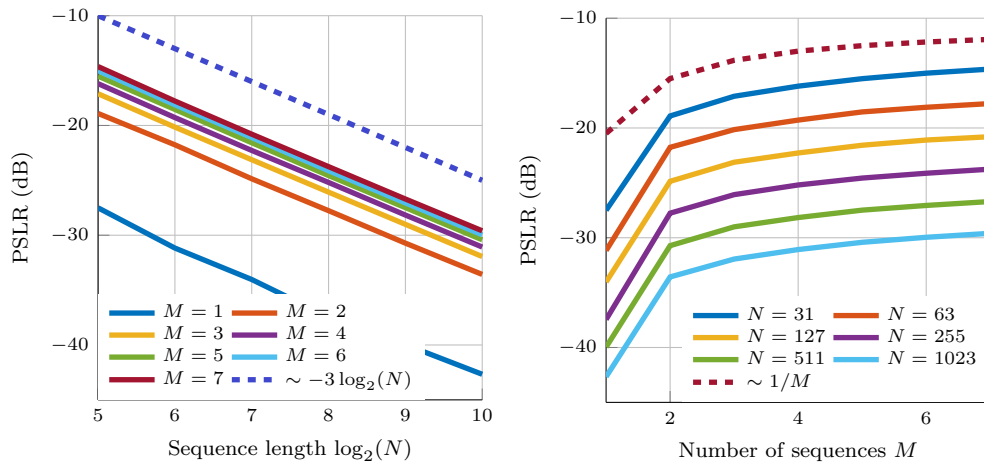


FIGURE 3.22 – Comparaison des lobes secondaires selon différents paramètres
A gauche : variation de la longueur de séquence N .
A droite : variation du nombre de séquences de l'ensemble M .

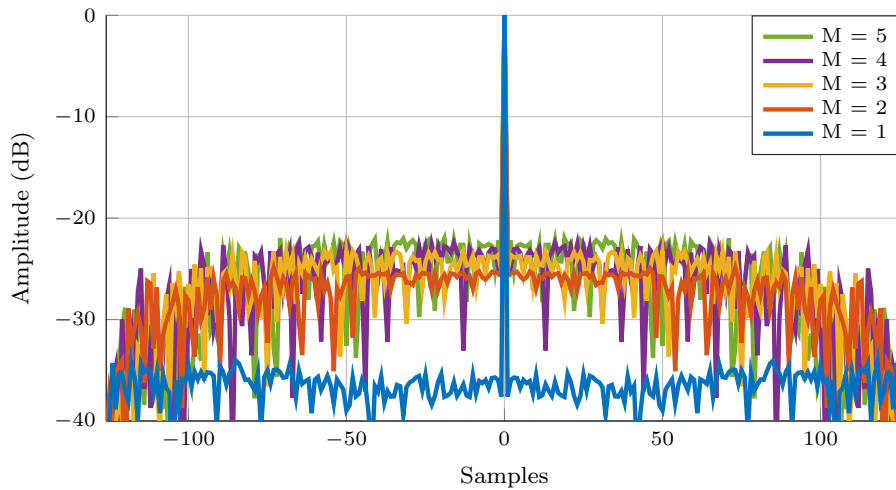


FIGURE 3.23 – Comparaison des premières séquences de corrélation
Séquences de longueur $N = 127$.
Variation du nombre d'éléments de l'ensemble.

3.6 Bornes : jusqu'où peut-on aller ?

La section précédente a introduit une optimisation d'un ensemble de séquences. Les quelques simulations effectuées nous ont offert des convictions sur les performances — le niveau des lobes secondaires — atteignables en pratique. Nous n'avons cependant aucune certitude sur leur optimalité (*Existe-t-il de meilleures séquences ?*), dans la mesure où peu de propriétés sur la fonction de corrélation et ses optima sont connues.

En ce sens, plusieurs bornes ont été définies sur le niveau des lobes secondaires d'un ensemble, la première étant la borne de Welch [Welch, 1974] dans les années 70, suivie par la borne de Levenshtein [Levenshtein, 1998] 25 ans plus tard. Cette dernière n'avait été démontrée que dans le cas d'un ensemble de séquences binaires ou de séquences basées sur les racines n -ème l'unité [Boztaş, 1998]. Sous ces conditions, la borne de Levenshtein est plus resserrée que la borne de Welch (sauf dans certains cas particuliers). Cependant, ce chapitre a pour sujet d'étude les codes polyphases. Pouvons-nous trouver une borne dans ce cas précis ?

Ainsi, cette section s'intéresse donc à ces bornes analytiques. Elle commence par un bref rappel sur les bornes définies dans la littérature. Ensuite, elle prouve que la borne de Levenshtein est *valable* pour les séquences polyphases. Enfin, on suggère une petite astuce permettant de bonifier la borne la plus serrée à ce jour. Les performances de cette « nouvelle » borne seront illustrées avec des exemples.

Le contenu de cette section a fait l'objet d'une publication [Arlery et al., 2019]. L'ensemble des démonstrations, très calculatoires, s'y trouvent. Elles sont reprises dans l'Annexe A.

Remarque 3.7. (*Codebooks*) Comme mentionné dans une remarque précédente, l'orthogonalité est associée ici aux séquences de corrélation : les bornes que nous allons présenter dans cette section se basent donc sur cette hypothèse de travail. Il existe cependant dans la littérature des bornes homonymes où l'orthogonalité n'est seulement définie que par le produit scalaire. Il faut donc veiller à bien différencier ces deux cas de figure...

3.6.1 État de l'art sur les bornes

Commençons par énoncer quelques définitions. Afin d'être cohérent sur la littérature à ce sujet, certaines notations seront adaptées ici. Soit $\{\mathbf{s}^m\}_{m \in \llbracket 1, M \rrbracket}$ un ensemble de M séquences de longueur N sous la contrainte de module constant. Pour deux séquences \mathbf{s}^{m_1} et \mathbf{s}^{m_2} de cet ensemble, on note par $\boldsymbol{\theta}_{\mathbf{s}^{m_1}}$ la séquence d'autocorrélation et par $\boldsymbol{\theta}_{\mathbf{s}^{m_1}, \mathbf{s}^{m_2}}$ la séquence d'inter-corrélation.

Les bornes auxquelles nous nous intéressons ici concernent le niveau du lobe secondaire le plus élevé parmi l'ensemble des autocorrélations et inter-corrélations. Ce dernier sera noté θ_{\max} et est défini par :

$$\theta_{\max}^2 = \max \left\{ \max_{\substack{k \\ m_1 \neq m_2}} |\theta_{\mathbf{s}^{m_1}, \mathbf{s}^{m_2}}(k)|^2, \max_{\substack{k \neq N \\ m_1}} |\theta_{\mathbf{s}^{m_1}}(k)|^2 \right\}. \quad (3.45)$$

Les bornes les plus utilisées ont été apportées, par ordre chronologique, par Welch [Welch, 1974] puis Levenshtein [Levenshtein, 1998]. La première est définie par :

$$\theta_{\max}^2 \geq \frac{M-1}{M(2N-1)-1}. \quad (3.46)$$

La borne de Welch n'est donc pas informative pour $M = 1$. La seconde introduit un petit vecteur de pondération, noté \mathbf{w} et de longueur $2N - 1$, qu'on applique sur chaque élément

des séquences de corrélation. Ce dernier doit vérifier la *condition de pondération* suivante :

$$\sum_{k=1}^{2N-1} w_k = 1, \text{ avec } w_k \geq 0 \text{ si } k \in \llbracket 1, 2N-1 \rrbracket, \quad (3.47)$$

$$w_k = 0 \text{ sinon.}$$

La borne de Levenshtein s'écrit :

$$\theta_{\max}^2 \geq \frac{1}{N^2} \left(N - \frac{Q_{2N-1} \left(\mathbf{w}, \frac{N(N-1)}{M} \right)}{1 - \frac{1}{M} \sum_{k=1}^{2N-1} w_k^2} \right), \quad (3.48)$$

où \mathbf{w} est un vecteur de pondération vérifiant la condition de pondération (3.47), et :

$$\begin{cases} Q_{2N-1}(\mathbf{w}, a) = a \sum_{k=1}^{2N-1} w_k^2 + \sum_{s,t=1}^{2N-1} l_{s,t,N} w_s w_t, \\ l_{s,t,N} = \min(|s-t|, 2N-1-|s-t|). \end{cases} \quad (3.49)$$

La borne de Levenshtein [Levenshtein, 1998] a été initialement définie pour les séquences binaires. Sa démonstration a été étendue dans le même esprit [Boztaş, 1998, Peng *et al.*, 2004] aux racines n -ièmes de l'unité. Mais elle ne permet pas, telle quelle, de démontrer sa validité à n'importe quelle séquence polyphasée, car elle fait intervenir le cardinal de l'ensemble des séquences possibles, qui n'est pas dénombrable dans le cas des séquences polyphases. En utilisant une stratégie différente de celle de [Boztaş, 1998] et de [Levenshtein, 1998], mélangeant les démonstrations originelles de Welch et de Levenshtein, nous allons en fait prouver dans le paragraphe suivant cette validité [Arlery *et al.*, 2019]. Sans transition, signalons enfin que la borne de Levenshtein est plus précise que celle de Welch dans le cas $M \geq 4, N \geq 2$.

A ce stade, les études qui ont suivi se sont majoritairement consacrées à rechercher des vecteurs de pondération permettant d'apporter une borne de Levenshtein resserrée. Par exemple, dans le cas particulier d'un vecteur de pondération constant, ce qui s'écrit :

$$w_k = \begin{cases} 1/K & \text{pour } k \leq K, \\ 0 & \text{sinon,} \end{cases} \quad (3.50)$$

la borne de Levenshtein devient :

$$\theta_{\max}^2 \geq \frac{1}{N^2} \left(\frac{3NMK - 3N^2 - MK^2 + M}{3(KM - 1)} \right). \quad (3.51)$$

La partie de droite étant fonction du paramètre K , il est possible d'en calculer l'optimum, ce qui nous donne :

$$\theta_{\max}^2 \geq \frac{1}{N^2} \left(N - \frac{2N}{\sqrt{3M}} \right) \quad \text{pour } M \geq 3. \quad (3.52)$$

Par ailleurs, la borne de Levenshtein induit un problème de minimisation sur les vecteurs de pondération \mathbf{w} [Levenshtein, 1999]. La forme $Q_{2N-1}(\mathbf{w}, a)$ étant quadratique, Levenshtein lui-même a pu en obtenir une borne plus serrée :

$$\theta_{\max}^2 \geq \frac{1}{N^2} \left(N - \left\lceil \frac{\pi N}{\sqrt{8M}} \right\rceil \right) \quad \text{pour } 5 \leq M \leq N^2. \quad (3.53)$$

D'autres vecteurs de pondérations [Liu *et al.*, 2017, Liu *et al.*, 2013, Liu *et al.*, 2014c] ont été suggérés, le troisième article proposant la borne de Levenshtein la plus serrée à notre connaissance.

3.6.2 Généralisation de la borne de Levenshtein

Ce paragraphe introduit les notations qui nous seront utiles par la suite, et présente les grandes lignes montrant que la borne de Levenshtein peut s'appliquer dans le cas des ensembles de séquences polyphases. Cette généralisation s'effectue en deux étapes : on commence par établir une borne supérieure et une borne inférieure sur la norme d'une matrice contenant l'ensemble des éléments des séquences de corrélations. On en déduit ensuite de manière élémentaire une borne sur le PSLR de l'ensemble.

Soit $\{\mathbf{s}^m\}_{m \in \llbracket 1, M \rrbracket}$ un ensemble de M séquences unimodulaires de longueur N . On considère la matrice \mathbf{S} de taille $[M(2N - 1)] \times [2N - 1]$ suivante :

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}^1 \\ \mathbf{S}^2 \\ \vdots \\ \mathbf{S}^M \end{bmatrix} \quad (3.54)$$

avec \mathbf{S}^m , $m \in \llbracket 1, M \rrbracket$, une matrice carrée de taille $2N - 1$ définie par :

$$\mathbf{S}^m = \begin{bmatrix} s_1^m & s_2^m & \cdots & s_{N-1}^m & s_N^m & 0 & 0 & \cdots & 0 \\ 0 & s_1^m & s_2^m & \cdots & s_{N-1}^m & s_N^m & 0 & & \vdots \\ 0 & 0 & \ddots & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & s_1^m & s_2^m & \cdots & s_{N-1}^m & s_N^m \\ s_N^m & 0 & & & 0 & s_1^m & s_2^m & & s_{N-1}^m \\ s_{N-1}^m & s_N^m & 0 & & & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & 0 & s_1^m & s_2^m \\ s_2^m & \cdots & s_{N-1}^m & s_N^m & 0 & \cdots & \cdots & 0 & s_1^m \end{bmatrix}. \quad (3.55)$$

La matrice \mathbf{S}^m contient donc des versions décalées de la séquence \mathbf{s}^m .

Remarque 3.8. (Matrices circulantes) Soit $\mathbf{x} = [x_1, \dots, x_n]$ un vecteur de longueur n . On peut définir une matrice circulante à partir d'une application, notée circ , et telle que :

$$\left\{ \begin{array}{l} \mathbb{C}^n \rightarrow \mathbb{C}^{n \times n} \\ \mathbf{x} \mapsto \text{circ}(\mathbf{x}) = \begin{bmatrix} x_1 & x_2 & \cdots & x_{n-1} & x_n \\ x_n & x_1 & x_2 & & x_{n-1} \\ \vdots & x_n & x_1 & \ddots & \vdots \\ x_3 & & \ddots & \ddots & x_2 \\ x_2 & x_3 & \cdots & x_n & x_1 \end{bmatrix} \end{array} \right. \quad (3.56)$$

Remarquez ainsi que la matrice \mathbf{S}^m est circulante et spécifiée par le vecteur $[s^m, \mathbf{0}_{N-1,1}]^T$.

Soit \mathbf{S}_k^m , $k \in \llbracket 1, 2N - 1 \rrbracket$ la k -ème ligne de la matrice \mathbf{S}^m . On peut remarquer que chaque valeur de corrélation peut se retrouver à l'aide d'un produit scalaire entre ces lignes. De ce fait, ces valeurs sont forcément contenues dans une matrice carrée $\mathbf{R} := \mathbf{S}\mathbf{S}^H$ de taille $M(2N - 1)$.

La borne de Levenshtein a introduit un vecteur de pondération \mathbf{w} . Il peut donc être utile de considérer des versions pondérées des matrices que nous venons d'introduire :

$$\begin{cases} \tilde{\mathbf{S}}_k^m = \mathbf{S}_k^m \sqrt{w_k} & k \in \llbracket 1, 2N-1 \rrbracket \text{ et } m \in \llbracket 1, M \rrbracket, \\ \tilde{\mathbf{S}} = \begin{bmatrix} \tilde{\mathbf{S}}^1 \\ \tilde{\mathbf{S}}^2 \\ \vdots \\ \tilde{\mathbf{S}}^M \end{bmatrix}, \\ \tilde{\mathbf{R}} = \tilde{\mathbf{S}} \tilde{\mathbf{S}}^H. \end{cases} \quad (3.57)$$

Par exemple, dans le cas particulier où la pondération \mathbf{w} est non-nulle pour les K premiers éléments uniquement, la matrice $\tilde{\mathbf{R}}$ ne contiendra que les valeurs de corrélations considérant les K premiers décalages. Par ailleurs, cette matrice $\tilde{\mathbf{R}}$ vérifie les propriétés suivantes :

Lemme 3.1. (*Borne supérieure*) La norme de Frobenius de la matrice $\tilde{\mathbf{R}}$ admet la borne supérieure suivante :

$$\|\tilde{\mathbf{R}}\|_F^2 \leq M^2 \theta_{\max}^2 + M (1 - \theta_{\max}^2) \sum_{k=1}^{2N-1} w_k^2, \quad (3.58)$$

Lemme 3.2. (*Borne inférieure*) ... et la borne inférieure qui suit :

$$\|\tilde{\mathbf{R}}\|_F^2 \geq \frac{M^2}{N^2} \left(N - \sum_{s,t=1}^{2N-1} l_{s,t,N} w_s w_t \right). \quad (3.59)$$

avec $l_{s,t,N} = \min(|t-s|, 2N-1-|t-s|)$.

A partir de ces deux lemmes, on peut en déduire une borne sur θ_{\max}^2 .

Théorème 3.1. Soit \mathbf{w} un vecteur vérifiant la condition de pondération (3.47). Pour tout ensemble de M séquences unimodulaires de même longueur N , une borne inférieure sur le lobe secondaire le plus élevé s'écrit :

$$\theta_{\max}^2 \geq \frac{1}{N^2} \left[N - \frac{Q_{2N-1} \left(\mathbf{w}, \frac{N(N-1)}{M} \right)}{1 - \frac{1}{M} \sum_{k=1}^{2N-1} w_k^2} \right], \quad (3.60)$$

avec $Q_{2N-1}(\mathbf{w}, a) = a \sum_{k=1}^{2N-1} w_k^2 + \sum_{s,t=1}^{2N-1} l_{s,t,N} w_s w_t$
et $l_{s,t,N} = \min(|s-t|, 2N-1-|s-t|)$.

Toutes les démonstrations se trouvent dans [Arlery *et al.*, 2019], et sont reprises dans l'Annexe A. Sachez simplement qu'elles exploitent la structure des matrices \mathbf{S}_m et l'hypothèse d'unimodularité. Finalement, cette expression est strictement identique à la borne de Levenshtein, précédemment présentée. Cela signifie que cette dernière est non seulement valable pour les séquences binaires, mais elle l'est aussi pour toutes les séquences unimodulaires. En outre, cela implique que les vecteurs de pondérations définis pour obtenir une borne plus resserrée [Levenshtein, 1999, Liu *et al.*, 2014c] le sont également.

3.6.3 Améliorations de la borne de Levenshtein

Il est assez aisé de voir le lien entre le Théorème 3.1 et les bornes précédemment mentionnées. Ces dernières portent sur la norme de Frobenius de la matrice $\tilde{\mathbf{R}}$, une matrice qui, on le rappelle, contient l'ensemble des valeurs de corrélations. Cependant, lorsque l'on étudie leurs démonstrations, ce sont des astuces sur la combinatoire du problème qui sont utilisées. Aucune propriété sur la structure des séquences de corrélation n'est utilisée.

Une séquence d'inter-corrélation s'écrit :

$$\boldsymbol{\theta}_{\mathbf{s}^l, \mathbf{s}^m} = \begin{bmatrix} \mathbf{s}_N^l(\mathbf{s}_1^m)^* \\ \mathbf{s}_{N-1}^l(\mathbf{s}_1^m)^* + \mathbf{s}_N^l(\mathbf{s}_2^m)^* \\ \vdots \\ \mathbf{s}_1^l(\mathbf{s}_{N-1}^m)^* + \mathbf{s}_2^l(\mathbf{s}_N^m)^* \\ \mathbf{s}_1^l(\mathbf{s}_N^m)^* \end{bmatrix}. \quad (3.61)$$

Pour chacun de ces éléments, on peut aisément écrire, à l'aide de l'inégalité de Cauchy-Schwarz (pour $d \in \llbracket 1, N \rrbracket$ et $d \neq N$ si $l = m$) :

$$|\theta_{\mathbf{s}^l, \mathbf{s}^m}(k)|^2 \leq (d/N)^2 \text{ pour } k = d, \text{ ou } k = 2N - d, \quad \forall (l, m). \quad (3.62)$$

En particulier, l'inégalité précédente devient égalité lorsque $d = 1$. Si on considère le parallèle avec le traitement radar, le paramètre d compte le nombre des derniers retards.

En insérant cette propriété dans la borne supérieure de la matrice $\tilde{\mathbf{R}}$, nous obtenons :

Lemme 3.3. (Borne supérieure, prise en compte des D derniers retards)

$$\|\tilde{\mathbf{R}}\|_F^2 \leq M^2 \theta_{\max}^2 + M (1 - \theta_{\max}^2) \sum_{k=1}^{2N-1} w_k^2 - \sum_{d=1}^D \left(M^2 \left(\theta_{\max}^2 - \frac{d^2}{N^2} \right) \sum_{\substack{i,j=1 \\ l_{i,j,N}=N-d}}^{2N-1} w_i w_j \right). \quad (3.63)$$

La démonstration est donnée dans l'Annexe A.2. Dans le cas où $D = 0$, cette borne supérieure est identique à la précédente. Mais il est possible de prouver qu'elle est plus précise si :

$$\exists (w_i, w_j) \text{ t.q. } \sum_{\substack{i,j=1 \\ l_{i,j,N}=N-d}}^{2N-1} w_i w_j \neq 0 \quad \text{et} \quad \exists d \text{ t.q. } \theta_{\max}^2 \geq \frac{d^2}{N^2}. \quad (3.64)$$

Finalement, afin d'obtenir la borne supérieure la plus serrée, il suffit de rechercher la valeur de D qui minimise le terme de droite de l'expression (3.63)... Quoiqu'il arrive, la nouvelle borne supérieure nous permet d'améliorer (ou, au pire, d'égaliser) la borne de Levenshtein. Mais, avant d'écrire la nouvelle borne, nous avons besoin de définir les matrices \mathbf{A}_d et \mathbf{L} telles que :

$$\begin{cases} \sum_{\substack{i,j=1 \\ l_{i,j,N}=N-d}}^{2N-1} w_i w_j = \mathbf{w}^T \mathbf{A}_d \mathbf{w}, \\ \sum_{s,t=1}^{2N-1} l_{s,t,N} w_s w_t = \mathbf{w}^T \mathbf{L} \mathbf{w}. \end{cases} \quad (3.65)$$

Un exemple de ces matrices est donné ci-dessous.

Exemple illustratif : Soit $N = 3$. La matrice \mathbf{L} est de taille 5×5 ($2N - 1 \times 2N - 1$ en fait) et est définie comme suit :

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & 2 & 2 & 1 \\ 1 & 0 & 1 & 2 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 2 & 1 & 0 & 1 \\ 1 & 2 & 2 & 1 & 0 \end{bmatrix} = \text{circ}([0, 1, 2, 2, 1]^T). \quad (3.66)$$

Les matrices \mathbf{L} et \mathbf{A}_d sont liées par la relation suivante :

$$(\mathbf{A}_d)_{i,j} = \begin{cases} 1 & \text{si } \mathbf{L}_{i,j} = N - d, \\ 0 & \text{sinon,} \end{cases} \quad (3.67)$$

ce qui nous donne :

$$\begin{aligned} \mathbf{A}_1 &= \text{circ}([0, 0, 1, 1, 0]^T), \\ \mathbf{A}_2 &= \text{circ}([0, 1, 0, 0, 1]^T). \end{aligned} \quad (3.68)$$

La nouvelle borne s'écrit sous ces notations :

Théorème 3.2. *Soit \mathbf{w} un vecteur vérifiant la condition de pondération (3.47). Pour tout ensemble de M séquences unimodulaires de même longueur N , une borne inférieure sur le lobe secondaire le plus élevé s'écrit :*

$$\theta_{\max}^2 \geq \max_D \frac{1}{N^2} \left[N - \frac{\tilde{Q}\left(\mathbf{w}, \frac{N(N-1)}{M}, \sum_{d=1}^D (d^2 - N)\mathbf{A}_d\right)}{1 - \mathbf{w}^T \left(\frac{1}{M}\mathbf{I} + \sum_{d=1}^D \mathbf{A}_d\right) \mathbf{w}} \right] \quad (3.69)$$

avec $\tilde{Q}(\mathbf{w}, a, \mathbf{B}) = \mathbf{w}^T (a\mathbf{I} + \mathbf{B} + \mathbf{L}) \mathbf{w}$.

Les démonstrations sont toujours dans [Arlery *et al.*, 2019]. Ce théorème a été obtenu en considérant la nouvelle borne supérieure fournie par le Lemme 3.3. Leur précision va en fait de pair : si cette borne supérieure sur $\|\tilde{\mathbf{R}}\|_F^2$ est plus serrée (par rapport à la version initiale, présentée dans la section précédente), alors la borne finale sur le PSLR le sera nécessairement. Ainsi, afin d'obtenir une petite amélioration, il suffit de jouer avec le paramètre D . C'est ce qu'indique le corollaire suivant :

Corollaire 3.1. *Soit B_{Lev} la valeur de la borne de Levenshtein pour des paramètres M , N et \mathbf{w} fixés. S'il existe une valeur d telle que \mathbf{w} vérifie (3.64) et $B_{\text{Lev}} \geq d^2/N^2$, alors la borne définie par le théorème 3.2 est plus serrée que la borne de Levenshtein pour la même pondération \mathbf{w} .*

Le Corollaire 3.1 présente une condition suffisante sur le choix du paramètre d pour obtenir une meilleure borne que celle de Levenshtein. Si une telle valeur existe — et que les conditions du corollaire sont vérifiées —, cela implique que :

$$B_{\text{Lev}} \geq \frac{d^2}{N^2} \implies d \in \left[0, \lfloor N\sqrt{B_{\text{Lev}}} \rfloor \right], \quad (3.70)$$

où $\lfloor \cdot \rfloor$ dénote la partie entière. En d'autres termes, toutes les valeurs de d appartenant à cet intervalle nous offre une borne plus serrée. Nécessairement, l'une d'entre elles apporte la plus serrée ; une approximation de cette dernière est donnée par la plus grande valeur de d qui satisfait :

$$B_{\text{Lev}} \geq \frac{d^2}{N^2} \implies D_{\max} \approx \lfloor N\sqrt{B_{\text{Lev}}} \rfloor. \quad (3.71)$$

Cependant, veuillez noter que l'ensemble des valeurs apportant une amélioration ne sont pas nécessairement comprises dans cet intervalle, comme le montre la Figure 3.24.

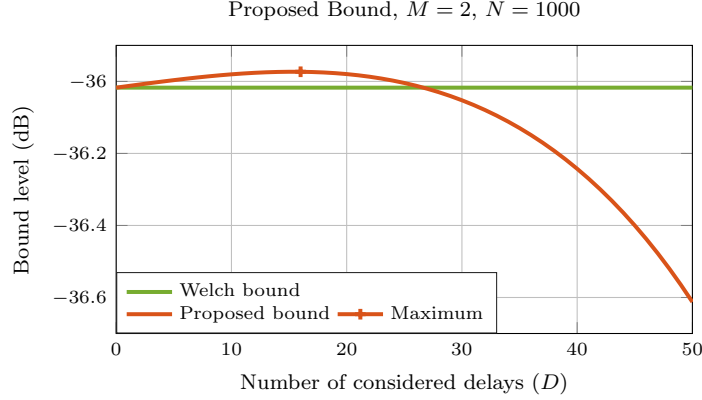


FIGURE 3.24 – La nouvelle borne, fonction du paramètre D
 En faisant varier le paramètre D , on peut déterminer une borne plus serrée.
 Dans le cas $M = 2$, la borne la plus serrée était celle de Welch.

Remarque 3.9. (Une borne atteignable ?) « Est-il possible d’atteindre la borne de Levenshtein ? » Comme expliqué dans [Liu et al., 2012b], ceci est une question ouverte. Une première réponse, en partie négative, peut être donnée, et ce, à partir du Corollaire 3.1. En effet, s’il existe un moyen de resserrer la borne de Levenshtein, cela signifie logiquement qu’elle ne peut être égale...

Finalement, une petite astuce semble améliorer les choses. Mais, est-ce vraiment le cas ? On se propose dans la section qui suit de faire quelques comparatifs pour les cas $M = 2$ et $M = 3$ (les autres seront en fait très similaires).

3.6.4 Comparaison avec les bornes existantes

La borne définie par le Théorème 3.2 dépend de plusieurs paramètres : ceux, pratiques, comme le nombre d’éléments de l’ensemble M , la longueur des séquences N , mais aussi ceux intrinsèques aux bornes, comme le vecteur de pondération \mathbf{w} ou le nombre de retards considérés D . Les performances de chacune des bornes dépendent du nombre de séquences [Levenshtein, 1999, Liu et al., 2013, Liu et al., 2014c] : par exemple, la borne la plus serrée dans le cas $M = 2$ est la borne de Welch, tandis que Levenshtein l’est pour les cas $M \geq 3$. Dans ces cas précis, il est temps d’effectuer une comparaison avec notre borne...

3.6.4.1 Deux séquences

Il a été prouvé que la borne de Levenshtein la plus serrée dans le cas $M = 2$ est obtenue avec des pondérations constantes, *i.e.* $w_k = 1/(2N - 1)$ pour $k = 1, \dots, 2N - 1$ [Levenshtein, 1999]. Avec cette pondération, il se trouve que la borne de Levenshtein est égale à la borne de Welch, qui s’écrit :

$$B_{\text{Welch}} = \frac{1}{4N - 3}. \quad (3.72)$$

Cependant, ce vecteur de pondération vérifie clairement la condition (3.64). Cela signifie qu’il existe $d \geq 1$ (il est assez aisé de vérifier que $d = 1$ fonctionne) tel que $B_{\text{Welch}} \geq d^2/N^2$ pour $N \geq 3$. Une application du Corollaire 3.1 nous permet de montrer que la borne que nous proposons est plus serrée que la borne de Welch pour $M = 2$, ce qui n’était pas possible avec la borne de Levenshtein, quelle que soit la pondération. D’ailleurs, cette nouvelle borne s’écrit :

$$\theta_{\max}^2 \geq \frac{1}{3N^2} \frac{3MN^2 - 3N^2 - MD(D + 1)(2D + 1)}{M(2N - 2D - 1) - 1}. \quad (3.73)$$

N	D_{\max}	Proposed (dB)	Welch (dB)	Gradient (dB)
10	1	-15.36	-15.68	-12.55
50	3	-22.77	-22.94	-20.09
100	5	-25.86	-25.99	-23.31
500	11	-32.94	-33.00	-30.41
1000	15	-35.97	-36.02	-33.41
10000	50	-46.01	-46.02	-37.76*

TABLE 3.3 – Comparaison des bornes pour différentes valeurs de N , $M = 2$.
 Le D_{\max} a été choisi de manière approximative à l'aide de l'expression (3.71).
 On peut observer un écart de 3 dB entre la borne et une application pratique.
 * : Hum, cette valeur semble douteuse.

N	D_{\max}	Proposed (dB)	Levenshtein (dB)	Gradient (dB)
10	1	-14.15	-14.28	-11.58
50	4	-21.36	-21.40	-18.77
100	6	-24.40	-24.43	-22.03
500	13	-31.42	-31.43	-28.85
1000	18	-34.43	-34.44	-31.65
10000	58	-44.442	-44.445	-36.76*

TABLE 3.4 – Comparaison des bornes pour différentes valeurs de N , $M = 3$.
 Le D_{\max} a été choisi de manière approximative à l'aide de l'expression (3.71).
 * : Hum, cette valeur semble douteuse.

Le calcul de l'expression (3.73) est donné en annexe. La Figure 3.24 illustre le comportement de notre borne envers le paramètre D . Il est possible d'observer qu'une meilleure borne est obtenue en considérant de 1 à 25 retards ; l'optimum est obtenu pour $D = 15$. Cette valeur peut être calculée approximativement :

$$\frac{15^2}{N^2} \approx 2,25 \cdot 10^{-4} < B_{\text{Welch}} \approx 2,5 \cdot 10^{-4} < \frac{16^2}{N^2} \approx 2,56 \cdot 10^{-4}. \quad (3.74)$$

Le Tableau 3.3 compare la borne de Welch avec notre borne pour certaines valeurs de N . La borne proposée est effectivement légèrement plus serrée.

3.6.4.2 Trois séquences

Cette fois-ci, la borne la plus serrée est celle de Levenshtein, associée avec les pondérations appelées *Positive-Cycle-of-a-Sine-Wave* par Liu *et al.* [Liu *et al.*, 2014c]. Ces pondérations sont définies par :

$$w_k = \begin{cases} \tan\left(\frac{\pi}{2K}\right) \sin\left(\frac{\pi(k-1)}{K}\right) & \text{si } k \in \llbracket 1, K \rrbracket, \\ 0 & \text{sinon.} \end{cases} \quad (3.75)$$

Avec ces pondérations, la borne de Levenshtein associée atteint son optimum pour une certaine valeur de K , qu'on note K_{opt} . Mais, est-elle plus lâche que la borne que nous proposons ? Pour répondre à cette question, il suffit de vérifier si le corollaire 3.1 est applicable.

Comme mentionnée dans [Liu *et al.*, 2014c], la borne de Levenshtein avec les pondérations de Liu est donnée par :

$$B(K) = \frac{1}{N^2} \left[N - \frac{NK(N-1) \tan^2\left(\frac{\pi}{2K}\right) + 2MQ}{2M - K \tan^2\left(\frac{\pi}{2K}\right)} \right], \quad (3.76)$$

$$\text{avec } Q = \begin{cases} \frac{K}{4} \left[1 - \tan^2\left(\frac{\pi}{2K}\right) \right] & \text{pour } K \in \llbracket 2, N \rrbracket, \\ -\frac{3K - 4N + 2}{4} - \frac{K}{4} \tan^2\left(\frac{\pi}{2K}\right) + \frac{K - N - 1}{2} \cos\left(\frac{N\pi}{K}\right) \\ + \left[\frac{2K - 2N + 1}{4} \tan\left(\frac{\pi}{2K}\right) + \frac{3}{4 \tan\left(\frac{\pi}{2K}\right)} \right] \sin\left(\frac{N\pi}{K}\right) & \text{sinon.} \end{cases}$$

On peut montrer que cette borne est une fonction croissante sur son intervalle de définition, dans les cas où $M = 3$ et $M = 4$. Ainsi, K_{opt} est nécessairement plus grand ou égal à N . En calculant $B(N)$, on peut aisément vérifier que la condition (3.64) est vérifiée, ce qui nous permet d'affirmer que notre borne est encore une fois plus serrée que la borne de Levenshtein. Des analyses plus poussées doivent être effectuées dans le cas $M \geq 5$, car la valeur de K_{opt} peut être plus petite que N .

La Figure 3.25 compare les différentes bornes avec la pondération *Positive-Cycle-of-a-Sine-Wave* [Liu *et al.*, 2014c] dans le cas $M = 3$. Le paramètre D a été estimé à l'aide de l'approximation (3.71). Sur cette figure, on peut notamment voir que la valeur K_{opt} est bien supérieure à la longueur de séquence N . Par ailleurs, le Tableau 3.4 donne quelques valeurs numériques.

Selon les applications, il peut être nécessaire de considérer l'ensemble des corrélations, *i.e.*, $K = 2N - 1$. Pour ce cas précis, Liu *et al.* ont développé un autre vecteur de pondération [Liu *et al.*, 2017]. Pour tout entier q :

$$w_k = \frac{1}{2N - 1} \left(1 + \frac{\cos\left(\frac{2\pi(k+q)}{2N-1}\right)}{\cos\left(\frac{\pi}{2N-1}\right)} \right), \quad k \in \llbracket 1, 2N - 1 \rrbracket. \quad (3.77)$$

La Figure 3.25 compare également les bornes avec ces pondérations. Les pondérations *Positive-Cycle-of-a-Sine-Wave* apportent toujours une borne globalement plus précise (à l'aide d'un choix judicieux du paramètre K) mais, si on considère l'ensemble des corrélations, la tendance s'inverse. Cela n'est pas si surprenant, car les pondérations introduites dans [Liu *et al.*, 2017] ont précisément été développées pour ce cas précis. Néanmoins, dans les deux cas, la borne que nous proposons reste légèrement plus précise.

3.6.5 Conclusion

Dans cette section plus théorique, nous avons présenté deux bornes sur le niveau maximal des lobes secondaires d'un ensemble : la borne de Welch et la borne de Levenshtein. La première a l'avantage d'avoir une formulation simple et élégante, tandis que la seconde — dépendant d'un vecteur de pondération — est généralement plus serrée. Mais elle n'était établie que pour les séquences binaires et les séquences basées sur les racines de l'unité. De ce fait, la première chose que nous avons démontré concerne sa validité pour les séquences unimodulaires.

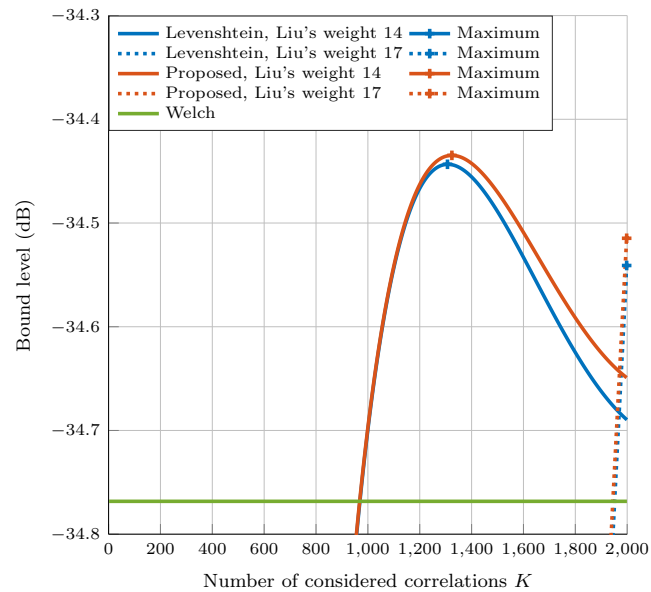


FIGURE 3.25 – Comparaison entre la borne de Levenshtein et la borne proposée, $M = 3$. « Liu's weight 14 » réfère aux pondérations de cet article [Liu et al., 2014c], « Liu's weight 17 » de celui-ci [Liu et al., 2017].

La démonstration originelle de la borne de Levenshtein prend notamment en compte des observations combinatoires sur le problème. En y ajoutant des considérations sur la structure spécifique des séquences de corrélation, on peut définir une borne plus serrée. Le Corollaire 3.1 donne d'ailleurs des conditions assez simples pour montrer que tel est le cas. La borne ainsi obtenue est, à notre connaissance, la borne la plus serrée à ce jour sur les ensembles de séquences. En ce sens, tout vecteur de pondération permettant d'améliorer la borne de Levenshtein améliore nécessairement notre borne.

Avoir pu améliorer la borne de Levenshtein signifie que celle-ci ne peut être atteinte. Par contre, la nouvelle borne le pourrait... En outre, les exemples numériques évoquent une différence de 3 dB entre les séquences obtenues à l'aide d'une descente de gradient et les bornes (sauf en grande dimension ?), quelque soit la longueur de séquence. Un signe d'espoir, ou simple coïncidence ?

3.7 Conclusion et perspectives

« Optimisation de la corrélation d'un code de phase ». En ces termes se décrit assez succinctement l'objectif de ce chapitre et du précédent. *Corrélation*. La fonction de corrélation est une fonction de base du traitement du signal. Elle permet de caractériser les performances d'un signal radar, comme la résolution ou la détection, notamment par le biais du niveau des lobes secondaires. *Optimisation*. Dans le but de rechercher un signal présentant un bon niveau de lobes secondaires, nous avons suggéré une méthodologie basée sur un problème d'optimisation. La fonction coût retourne le niveau des lobes secondaires des séquences de corrélation associées. *Codes de phase*. Les variables d'études considérées décrivent un code de phase. Ce choix a été influencé par la suite de cette étude. En outre, les codes de phase définissent en quelque sorte une approximation de tout signal échantillonné...

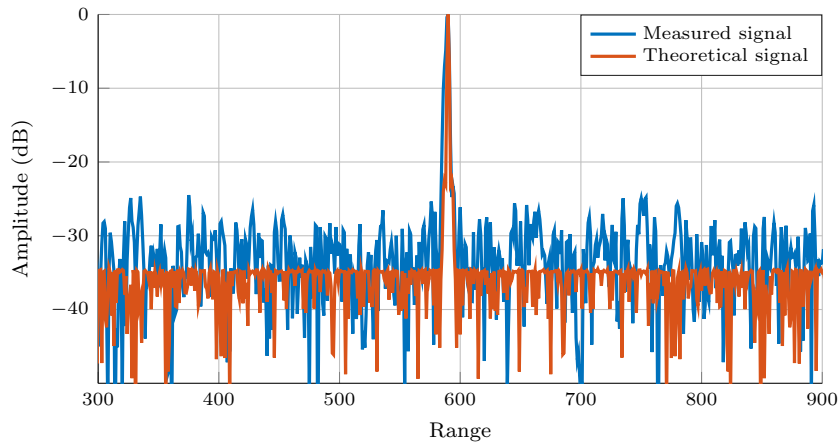


FIGURE 3.26 – Exemple de résultats obtenus avec le banc Hycam de l’ONERA
La séquence émise est un code de phase optimisée dans le sens du filtrage adapté, avec des contraintes sur le spectre. L’axe des abscisses est arbitraire.

Le problème en question est malheureusement non convexe, doté de nombreuses variables... donc difficile à résoudre. Aucune méthode ne peut certifier — à notre connaissance — la découverte du minimum global. Ainsi, on s’est proposé de comparer quelques méthodes stochastiques et déterministes. Chacune a ses qualités et ses défauts, mais se valent en terme de niveau de lobes secondaires. L’application d’une méthode stochastique est relativement aisée, ne nécessitant que des sorties de la fonction coût. Les méthodes stochastiques sont par contre relativement lentes à converger. La vitesse est une qualité des méthodes basées sur le gradient. Mais elles requièrent, comme leurs noms l’indiquent, des hypothèses sur la continuité de la fonction coût. De plus, la convergence est au mieux locale.

La modélisation d’un problème d’optimisation admet une certaine flexibilité. Il est en effet assez facile d’y greffer des besoins et des contraintes supplémentaires : contraintes spectrales, contraintes sur le Doppler, prise en compte de la quantification ou des effets d’éclipse... Mais cela a un coût : une dégradation certaine des performances initiales, et donc un compromis à définir. Par exemple, l’optimisation de l’orthogonalité d’un ensemble de séquences s’est faite naturellement. Nous verrons d’ailleurs par la suite qu’il en sera de même pour les codes de phase pour le radar MIMO cohérent.

Quelle que soit l’optimisation, nous présentons des performances en terme de lobes secondaires dont nous sommes certains qu’elles sont atteignables. Peut-on faire mieux ? Peut-être. Pour s’en convaincre, un ensemble de bornes a été développé sur le niveau maximal des lobes d’un ensemble. Bien qu’il n’est guère sûr que l’on puisse égaler ces bornes, on peut déceler une différence de l’ordre de 3 dB avec les exemples numériques. Les possibilités d’amélioration restent donc limitées.

Dans tous les cas, beaucoup de pistes ont été laissées en suspens. Tout d’abord, il faudrait se pencher sur les résultats de la descente de gradient en très grande dimension, où l’écart avec les bornes est important. Ensuite, en terme d’optimisation, de nombreuses méthodes sont encore à considérer, à l’instar des méthodes de pénalisation comme les méthodes de points intérieurs, ou de la relaxation du problème avec une contrainte sur le PAPR. En outre, pourquoi ne pas voir le problème sous un autre angle, comme étudier le problème dans la variété \mathbf{T}^N ? On pourrait appeler ça « Opium », pour *OPTimisation In Unit Modulus spaces...* Cela pourrait éventuellement nous aider à trouver des propriétés utiles sur la fonction de corrélation. Enfin, pourquoi toujours considérer une optimisation ? Pourrait-on approcher le problème autrement ? Les codes de Gold ou les m -séquences sont par exemple fondées sur des théories algébriques. Avis aux amateurs. Nous espérons simplement que ces pages apporteront une pierre à l’édifice, bien que le problème reste encore et toujours ouvert...

Concluons ce chapitre en mentionnant qu'une application réelle a été effectuée avec le banc Hycam de l'ONERA. Cet essai s'est présenté sous forme de détection de cibles d'opportunités (avions domestiques de l'aéroport d'Orly) à l'aide d'un seul code de phase, optimisé spectralement. La détection a été positive (Figure 3.26), ce qui confirme le fait que les codes de phase peuvent être utilisés dans un radar. Néanmoins, nous n'avons pu comparer les résultats obtenus avec d'autres formes d'onde, et nous avons observé une perte quantifiable par rapport aux résultats théoriques (mais qui est peut-être due à une mauvaise connaissance des paramètres du radar, comme le facteur d'échantillonnage).

Chapitre 4

En passant par le filtre désadapté

Le Chapitre 2 a eu pour sujet la fonction de corrélation. Dans une optique radar, cela sous-entend l'utilisation d'un filtre adapté. Ce dernier consiste en une convolution entre le signal reçu et le signal émis — qui joue le rôle de filtre — et est défini pour maximiser le rapport signal-sur-bruit (*SNR*) [Levanon *et al.*, 2004]. En contrepartie, la séquence en sortie de filtre peut présenter des lobes secondaires assez élevés. Deux options immédiates s'offrent à nous. La première consiste à rechercher des séquences intéressantes sur ce plan : le chapitre susmentionné s'en est occupé. La seconde est celle utilisée en pratique ; elle prône l'utilisation d'un autre filtre, qualifié de « désadapté ».

Un filtre, et non pas *le* filtre, car ce dernier n'est pas unique. En effet, n'importe quel signal peut, cette fois, embrasser la vocation de filtre. Il n'est d'ailleurs nullement contraint, que ce soit sur sa norme ou son support par exemple. Le filtre désadapté est donc en quelque sorte une généralisation du filtre adapté. On attend donc de son emploi un apport sur les lobes secondaires, au prix d'une perte intrinsèque liée à la désadaptation.

Cela étant, il a été prouvé qu'une séquence admet un filtre désadapté optimal pour un certain critère donné (comme l'ISL ou le PSLR), car ce dernier est solution d'un problème d'optimisation convexe [Rabaste *et al.*, 2015].

La recherche d'un filtre désadapté n'étant plus un problème, comment déterminer le couple séquence–filtre permettant de maximiser les performances de détection ? En se basant sur cette propriété d'optimalité, de nombreux travaux favorisent les algorithmes cycliques [Karbasi *et al.*, 2015, Stoica *et al.*, 2012] pour traiter le problème d'optimisation associé. Un algorithme cyclique consiste à résoudre alternativement plusieurs sous-problèmes d'optimisation, pour lesquels l'une des séquences est la variable à optimiser, pendant que les autres sont fixées. L'optimisation est donc rarement effectuée sur les deux séquences simultanément...

Le présent chapitre est composé de trois parties. La première, préliminaire, instaure les notations et donne quelques (re-)définitions. On rappellera notamment la procédure qui nous acheminera vers le filtre désadapté optimal, qu'on étendra pour un ensemble de séquences. La seconde partie introduira un algorithme d'optimisation jointe, permettant d'optimiser *simultanément* la séquence et le filtre désadapté, tout en conservant ce caractère optimal. Enfin, à l'instar de la Section 3.6, on se proposera de définir une borne sur le niveau des lobes secondaires que l'on peut atteindre avec un filtre désadapté. Ce type de borne n'a, à notre connaissance, jamais été suggéré. Le comportement de celle-ci nous a d'ailleurs surpris...

4.1 Préliminaires

Cette petite section, préliminaire, commence par introduire de manière analytique le filtre désadapté. De nombreux parallèles peuvent être établis avec les expressions dans le cas adapté. Ensuite, elle explique comment déterminer un filtre désadapté optimal selon un critère donné, pour une séquence donnée. Cela passe par un problème d'optimisation — très familier — qui se trouve être convexe. Le cas particulier de l'ISL sera par ailleurs présenté. Enfin, on étendra cette procédure à des ensembles de séquences.

4.1.1 Définitions

Soit $\mathbf{s} = [s_1, s_2, \dots, s_N]^T$ un code de phase de longueur N . On supposera dans ce chapitre que la séquence \mathbf{s} est soumise à la contrainte de module constant : chaque élément s_n est de même amplitude et n'est donc distinct que par sa phase $\varphi_n \in [0, 2\pi[$:

$$s_n = \frac{1}{\sqrt{N}} \exp(j\varphi_n), \text{ pour } n \in \llbracket 1, N \rrbracket. \quad (4.1)$$

En d'autres termes, \mathbf{s} appartient à un tore de dimension N , noté \mathbf{T}^N , un ensemble non convexe [do Carmo, 1992]. Sauf mention contraire, on posera dans la suite $s_n = 0$ pour tout $n \notin \llbracket 1, N \rrbracket$.

Comme nous l'avons vu dans le Chapitre 2, le filtrage adapté consiste à comparer le signal \mathbf{s} à des versions décalées de lui-même dans le temps. Cela revenait à étudier la séquence AACF(\mathbf{s}), de longueur $2N - 1$, telle que :

$$\text{AACF}(\mathbf{s}) = \mathbf{\Lambda}_N(\mathbf{s})\mathbf{s}^*, \quad (4.2)$$

où $\mathbf{\Lambda}_K(\mathbf{s})$, est une matrice de convolution de taille $(K + N - 1) \times K$, $K \in \mathbb{N}^*$ — initialement présentée à la Remarque 2.3 pour le cas particulier $K = N$ —, contenant des répliques de \mathbf{s} décalées dans le temps :

$$\mathbf{\Lambda}_K(\mathbf{s}) := \begin{bmatrix} s_N & 0 & \dots & \dots & \dots & \dots & 0 \\ \vdots & s_N & \ddots & & & & \vdots \\ s_2 & & \ddots & 0 & & & \vdots \\ s_1 & s_2 & \dots & s_N & 0 & \dots & 0 \\ 0 & s_1 & \ddots & \vdots & s_N & \ddots & \vdots \\ \vdots & \ddots & \ddots & s_2 & & \ddots & 0 \\ 0 & \dots & 0 & s_1 & s_2 & & s_N \\ \vdots & & & 0 & s_1 & \ddots & \vdots \\ \vdots & & & & \ddots & \ddots & s_2 \\ 0 & \dots & \dots & \dots & \dots & 0 & s_1 \end{bmatrix}. \quad (4.3)$$

La compression du signal \mathbf{s} à l'aide d'un filtre différent \mathbf{q} , de longueur K , est appelé filtrage désadapté. En étendant les notations du cas adapté, on pose :

$$\begin{aligned} \text{AXCF} : \mathbf{T}^N \times \mathbb{C}^K &\rightarrow \mathbb{C}^{K+N-1}, \\ (\mathbf{s}, \mathbf{q}) &\mapsto \text{AXCF}(\mathbf{s}, \mathbf{q}) = \mathbf{\Lambda}_K(\mathbf{s}^*)\mathbf{q}. \end{aligned} \quad (4.4)$$

L'expression (4.4) ressemble fort à celle de l'inter-corrélation mais, lors de l'étape de filtrage, ne recherche-t-on pas un profil digne d'une autocorrélation ? En la considérant seule, difficile de trancher. On se satisfera donc de la nomenclature volontairement ambiguë AXCF...

Notons que la longueur d'un filtre désadapté peut différer de celle de la séquence \mathbf{s} ; le filtre désadapté est généralement choisi plus long, apportant plus de degrés de liberté dans son choix. Sans perte de généralité, on supposera que $K = N + 2l$, $l \in \mathbb{N}$, de sorte que la longueur de la séquence en sortie $\text{AXCF}(\mathbf{s}, \mathbf{q})$ soit impaire. Par ailleurs, \mathbf{q} peut prendre librement ses valeurs dans \mathbb{C}^K , un ensemble convexe. Cette propriété contribuera à générer des filtres désadaptés optimaux selon certains critères, comme il sera expliqué dans le paragraphe suivant.

Remarque 4.1. (*Indices et retards*) Une expression équivalente à (4.4) est la suivante (mais attention aux indices !):

$$(\text{AXCF}(\mathbf{s}, \mathbf{q}))_k = \sum_{n=1}^K s_{n-k-l}^* q_n \quad (-N-l < k < N+l). \quad (4.5)$$

Remarque 4.2. (*Cas périodique*) Dans le cas périodique, la matrice $\mathbf{\Lambda}_N(\mathbf{s})$ est la suivante :

$$\mathbf{\Lambda}_N(\mathbf{s}) = \begin{bmatrix} s_N & s_1 & \cdots & s_{N-1} \\ s_{N-1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & s_1 \\ s_1 & \cdots & s_{N-1} & s_N \end{bmatrix}. \quad (4.6)$$

Remarquez qu'il est ici inutile de prendre un filtre désadapté plus long que N . En effet :

$$\begin{aligned} (\text{PXCF}(\mathbf{s}, \mathbf{q}))_k &= \sum_{n=1}^K s_{n-k-l}^* q_n \\ &= \sum_{\kappa=0}^{\lceil K/N-1 \rceil} \sum_{n=1}^N s_{n+\kappa N-k-l}^* q_{n+\kappa N} \\ &= \sum_{n=1}^N s_{n-k-l}^* \sum_{\kappa=0}^{\lceil K/N-1 \rceil} q_{n+\kappa N} \quad \text{car } \mathbf{s} \text{ est périodique,} \end{aligned} \quad (4.7)$$

sous l'hypothèse que $q_n = 0$ en dehors de son support principal. Ainsi, un filtre désadapté \mathbf{q} de longueur K équivaut à un filtre désadapté \mathbf{q}' de longueur N , dont les composantes sont :

$$q'_n = \sum_{\kappa=0}^{\lceil K/N-1 \rceil} q_{n+\kappa N}. \quad (4.8)$$

L'ensemble de ce chapitre peut se transposer pour le cas périodique, mais cela ne sera guère développé...

Le filtre adapté est défini comme celui maximisant le rapport signal sur bruit (SNR, pour *Signal-to-Noise Ratio*) [Levanon et al., 2004], sous l'hypothèse d'un bruit blanc additif. L'utilisation d'un autre filtre entraîne donc nécessairement une perte en gain de traitement, appelée LPG, pour *Loss in Processing Gain*. Cela s'apparente à une perte d'énergie du lobe principal, qui s'écrit :

$$\begin{aligned} \text{LPG} &= 10 \log_{10} \left(\frac{\text{SNR}_{\text{désadapté}}}{\text{SNR}_{\text{adapté}}} \right) \\ &= 10 \log_{10} \left(\frac{|\mathbf{q}^H \mathbf{s}|^2}{(\mathbf{q}^H \mathbf{q})(\mathbf{s}^H \mathbf{s})} \right) \leq 0. \end{aligned} \quad (4.9)$$

Notez le petit abus de dimension, qu'on utilisera souvent par commodité. En effet, pour que le produit scalaire entre \mathbf{s} et \mathbf{q} puisse se faire, il faut encadrer le premier avec un vecteur contenant l zéros, à gauche et à droite :

$$\mathbf{s} = [\mathbf{0}_{1,l}, s_1, \dots, s_N, \mathbf{0}_{1,l}]^T. \quad (4.10)$$

4.1.2 Filtre désadapté optimal

L'emploi d'un filtre désadapté a pour objectif initial de réduire les lobes secondaires lors du traitement. De manière similaire à la Section 2.1.3, on peut définir une grandeur générique, basée sur une norme, afin de mesurer le niveau de ces lobes secondaires. En considérant celle-ci comme une fonction objectif d'un problème d'optimisation, il est possible de générer un filtre désadapté optimal sous ce critère. En particulier, le paragraphe s'attardera sur le cas de l'ISL (pour *Integrated Sidelobe Level*), car le problème d'optimisation associé admet une propriété fort intéressante...

4.1.2.1 Mesure générique

Pour une séquence \mathbf{s} et son filtre désadapté \mathbf{q} , on peut définir une mesure générique des lobes secondaires de la façon suivante, basée sur la norme ℓ^p [Cilliers *et al.*, 2007] :

$$\begin{aligned} \mathbf{T}^N \times \mathbb{C}^K &\rightarrow \mathbb{R}_+ \\ (\mathbf{s}, \mathbf{q}) &\mapsto f_p(\mathbf{s}, \mathbf{q}) := \left\| \mathbf{F} \mathbf{A} \mathbf{X} \mathbf{C} \mathbf{F}(\mathbf{s}, \mathbf{q}) \right\|_p, \quad \text{pour } p \in \mathbb{N}^*, \end{aligned} \quad (4.11)$$

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

Remarque 4.3. (*PSLR et normalisation*) Le *Peak-to-Sidelobe Level Ratio (PSLR)* exprime la différence entre le lobe principal et le lobe secondaire le plus haut :

$$\text{PSLR}(\mathbf{s}, \mathbf{q}) := \frac{1}{|\mathbf{s}^H \mathbf{q}|} f_\infty(\mathbf{s}, \mathbf{q}). \quad (4.12)$$

Dans le cas particulier où $\mathbf{s}^H \mathbf{q} = \mathbf{s}^H \mathbf{s} = 1$, on obtient donc $\text{PSLR}(\mathbf{s}, \mathbf{q}) = f_\infty(\mathbf{s}, \mathbf{q})$. Parfois, le PSLR est défini avec la partie droite de l'expression (4.12) mais mise au carré. L'ISL sera évoqué dans le paragraphe suivant.

Remarque 4.4. (*Application partielle*) Soit \mathbf{s} une séquence appartenant à l'hypertore \mathbf{T}^N . L'« application partielle de f_p en \mathbf{s} associée à \mathbf{q} », notée par la suite $f_{p,\mathbf{s}}$, s'exprime par :

$$\begin{aligned} \mathbb{C}^K &\rightarrow \mathbb{R}_+ \\ \mathbf{q} &\mapsto f_{p,\mathbf{s}}(\mathbf{q}) = f_p(\mathbf{s}, \mathbf{q}). \end{aligned} \quad (4.13)$$

En clair, cela consiste à étudier la fonction d'origine f_p selon une seule variable, ici \mathbf{q} , les autres étant fixées. L'application partielle $f_{p,\mathbf{s}}$ se définit de la même manière. Remarquons que, par convexité de la norme dans un espace de dimension finie, les fonctions partielles $f_{p,\mathbf{s}}$ et $f_{p,\mathbf{q}}$ se trouvent être convexes (mais l'espace dans lequel vit la séquence \mathbf{s} ne l'est toujours évidemment pas).

Dans le but de rechercher un filtre désadapté \mathbf{q} , écrivons à l'aide d'une application partielle le problème de minimisation induit, que l'on note (P_p) :

$$(P_p) \begin{cases} \min_{\mathbf{q}} & f_{p,\mathbf{s}}(\mathbf{q}) = f_p(\mathbf{s}, \mathbf{q}) \\ \text{t.q.} & \mathbf{s}^H \mathbf{q} = \mathbf{s}^H \mathbf{s}. \end{cases} \quad (4.14)$$

La contrainte $\mathbf{s}^H \mathbf{q} = \mathbf{s}^H \mathbf{s}$ est ici nécessaire pour discréditer les solutions multiples engendrées par un même point, ainsi que la solution triviale.

La convexité individuelle de tous ses éléments (l'espace de départ \mathbb{C}^K , la fonction de coût basée sur la norme, la contrainte linéaire) rend ce problème convexe. De ce fait, les minima locaux réalisables de la fonction objectif sont les solutions de notre problème (P_p) . Cela signifie que toute séquence \mathbf{s} admet un filtre désadapté optimal $\mathbf{q}_p(\mathbf{s})$ — au moins un, en fait, si nous voulons être rigoureux — qui minimise le critère f_p . Mais, à notre connaissance, son expression analytique n'est malheureusement pas connue. La solution peut cependant être approchée numériquement, à l'aide d'algorithmes tels que la méthode des points intérieurs [Grant *et al.*, 2014, Rabaste *et al.*, 2015].

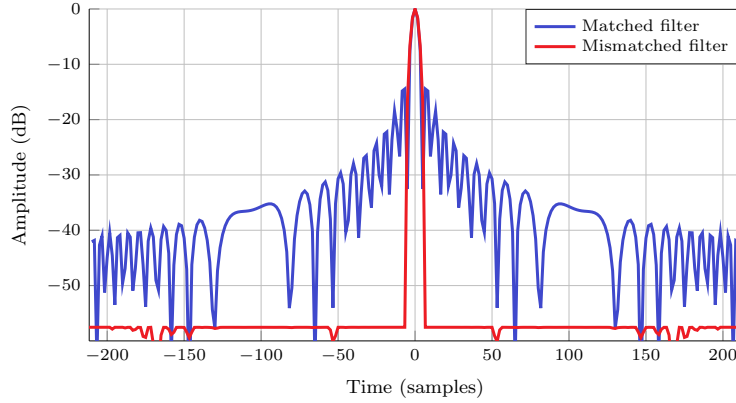


FIGURE 4.1 – Filtrage désadapté optimal pour un *chirp* échantillonné
 Filtre désadapté de longueur $3N$. Perte autorisée de 1 dB.
 Les corrélations sont normalisées. Représentation partielle.

Remarque 4.5. Le problème (P_∞) , au carré près, peut être mis sous la forme d'un problème quadratique convexe (ou, plus précisément, QCQP, pour *Quadratically Constrained Quadratic Program*) [Rabaste et al., 2015] :

$$\begin{cases} \min_{\mathbf{q}, t} & t \\ \text{t.q.} & \mathbf{s}^H \mathbf{q} = \mathbf{s}^H \mathbf{s} \\ & \mathbf{q}^H \boldsymbol{\lambda}_{N+p+i}^H(\mathbf{s}) \boldsymbol{\lambda}_{N+p+i}(\mathbf{s}) \mathbf{q} \leq t \quad \text{pour } |i| \geq 1. \end{cases} \quad (4.15)$$

où $\boldsymbol{\lambda}_k(\mathbf{s})$ est la k -ème ligne de la matrice $\boldsymbol{\Lambda}_K(\mathbf{s})$. Sous cette écriture, chaque contrainte — convexe, encore une fois — concerne un élément de la séquence en sortie de filtre.

Le problème précédent (P_p) ne contrôle en rien la perte due à la désadaptation (le LPG). Pour remédier à cela, il est possible de mettre cette perte sous la forme d'une contrainte d'un problème d'optimisation, comme l'indique la Remarque 4.6. Le problème qui en résulte reste heureusement convexe.

Remarque 4.6. Supposons que l'on souhaite une perte inférieure à α dB, $\alpha > 0$. Cela signifie que :

$$\begin{aligned} \text{LPG} &\geq -\alpha \quad \text{car } \text{LPG} \leq 0, \\ \Leftrightarrow 10^{\alpha/10} \mathbf{s}^H \mathbf{s} &\geq \mathbf{q}^H \mathbf{q}. \end{aligned} \quad (4.16)$$

La seconde expression peut être utilisée en tant que contrainte — convexe, qui plus est, puisque c'est une contrainte d'inégalité sur une forme quadratique définie positive — d'un problème d'optimisation, permettant ainsi de contrôler le LPG.

La Figure 4.1 représente les séquences en sortie pour une modulation linéaire de fréquence (*chirp*) après un filtrage adapté et un filtrage désadapté. Notez que le second a été défini au sens du PSLR, comme en témoigne le profil « plat » obtenu.

4.1.2.2 L'Integrated Sidelobe Level

Étudions le cas particulier de l'*Integrated Sidelobe Level*. L'ISL, de son petit nom, prend son sens en présence de fouillis pour, par exemple, réduire les effets de nombreux échos (dûs, notamment, aux multi-trajets) [Dai et al., 2008].

Comme mentionné ci-dessus, l'ISL peut se définir à partir de l'équation (4.11) :

$$\begin{aligned} \mathbf{T}^N \times \mathbb{C}^K &\rightarrow \mathbb{R}_+ \\ (\mathbf{s}, \mathbf{q}) &\mapsto \text{ISL}(\mathbf{s}, \mathbf{q}) := f_2(\mathbf{s}, \mathbf{q})^2 \\ &= \text{AXCF}(\mathbf{s}, \mathbf{q})^H \mathbf{F} \text{AXCF}(\mathbf{s}, \mathbf{q}). \end{aligned} \quad (4.17)$$

De la même façon, considérons le problème d'optimisation associé, que l'on note (P_{ISL}) :

$$(P_{\text{ISL}}) \begin{cases} \min_{\mathbf{q}} & \text{ISL}(\mathbf{s}, \mathbf{q}) \\ \text{t.q.} & \mathbf{s}^H \mathbf{q} = \mathbf{s}^H \mathbf{s}. \end{cases} \quad (4.18)$$

La résolution de ce problème de minimisation permet d'obtenir le filtre désadapté $\mathbf{q}_{\text{ISL}}(\mathbf{s})$, associé à la séquence \mathbf{s} , offrant l'ISL le plus bas. Nous avons choisi d'évoquer dans cette section ce cas particulier car le problème (P_{ISL}) admet une solution globale analytique, notée $\mathbf{q}_{\text{ISL}}(\mathbf{s})$, que l'on peut déterminer à l'aide des multiplicateurs de Lagrange [Griep *et al.*, 1995] :

$$\mathbf{q}_{\text{ISL}}(\mathbf{s}) = \frac{(\mathbf{s}^H \mathbf{s}) (\boldsymbol{\Lambda}_K(\mathbf{s})^H \mathbf{F} \boldsymbol{\Lambda}_K(\mathbf{s}))^{-1} \mathbf{s}}{\mathbf{s}^H (\boldsymbol{\Lambda}_K(\mathbf{s})^H \mathbf{F} \boldsymbol{\Lambda}_K(\mathbf{s}))^{-1} \mathbf{s}}. \quad (4.19)$$

Pour démontrer cela, écrivons le Lagrangien [Boyd *et al.*, 2004] du problème (P_{ISL}) :

$$\begin{aligned} \mathbb{C}^K \times \mathbb{C} &\rightarrow \mathbb{C} \\ (\mathbf{q}, \nu) &\mapsto \mathcal{L}(\mathbf{q}, \nu) = \mathbf{q}^H \boldsymbol{\Lambda}_K(\mathbf{s})^H \mathbf{F} \boldsymbol{\Lambda}_K(\mathbf{s}) \mathbf{q} + \nu (\mathbf{s}^H \mathbf{q} - \mathbf{s}^H \mathbf{s}) \\ &= \mathbf{q}^H \mathbf{A} \mathbf{q} + \nu (\mathbf{s}^H \mathbf{q} - \mathbf{s}^H \mathbf{s}), \end{aligned} \quad (4.20)$$

avec $\mathbf{A} = \boldsymbol{\Lambda}_K(\mathbf{s})^H \mathbf{F} \boldsymbol{\Lambda}_K(\mathbf{s})$. Recherchons ensuite ses points stationnaires, c'est-à-dire les couples (\mathbf{q}, ν) qui annulent le gradient de \mathcal{L} :

$$\begin{cases} \nabla_{\mathbf{q}} \mathcal{L}(\mathbf{q}, \nu) = 2\mathbf{A} \mathbf{q} + \nu \mathbf{s} = \mathbf{0} \\ \nabla_{\nu} \mathcal{L}(\mathbf{q}, \nu) = \mathbf{s}^H \mathbf{q} - \mathbf{s}^H \mathbf{s} = 0. \end{cases} \quad (4.21)$$

On peut vérifier que la matrice \mathbf{A} est définie positive donc inversible. En multipliant la première ligne par $\mathbf{s}^H \mathbf{A}^{-1}$ et, après quelques développements, on obtient le résultat souhaité :

$$\begin{cases} \nu = -\frac{\mathbf{s}^H \mathbf{s}}{\mathbf{s}^H \mathbf{A}^{-1} \mathbf{s}}, \\ \mathbf{q} = \frac{(\mathbf{s}^H \mathbf{s}) (\boldsymbol{\Lambda}_K(\mathbf{s})^H \mathbf{F} \boldsymbol{\Lambda}_K(\mathbf{s}))^{-1} \mathbf{s}}{\mathbf{s}^H (\boldsymbol{\Lambda}_K(\mathbf{s})^H \mathbf{F} \boldsymbol{\Lambda}_K(\mathbf{s}))^{-1} \mathbf{s}}. \end{cases} \quad (4.22)$$

De ce fait, l'ISL est intéressante car le problème d'optimisation associé admet une expression analytique, peu coûteuse. Nous exploiterons d'ailleurs cette propriété dans les prochaines pages. Avant cela, signalons que l'ajout d'une contrainte sur le LPG résulte en un problème dont la solution analytique n'est *a priori* pas connue, mais qui reste toutefois convexe.

4.1.3 Pour un ensemble de séquences

De manière analogue à la recherche d'un ensemble de séquences orthogonales (ceci fut l'objet de la Section 3.5), on se propose d'étendre les définitions ci-dessus à un ensemble de séquences, associé à un ensemble de filtres désadaptés. Ces deux ensembles se doivent d'atteindre au mieux les propriétés d'orthogonalité idéales. Ainsi, pour une séquence donnée, cela consiste en une bonne « autocorrélation » avec son filtre homologue, et une bonne intercorrélation avec les autres filtres. On ne considère donc nullement les corrélations entre séquences ou entre filtres.

Soient $S = \{\mathbf{s}^1, \dots, \mathbf{s}^M\} \subset \mathbf{T}^N$ un ensemble de M séquences de même longueur N et $Q = \{\mathbf{q}^1, \dots, \mathbf{q}^M\}$ un ensemble de filtres désadaptés, de même longueur K . Sans perte de généralités, on attribue à la séquence \mathbf{s}^m le filtre \mathbf{q}^m , pour tout $m \in \llbracket 1, M \rrbracket$. Une mesure naturelle des lobes secondaires entre une séquence \mathbf{s}^{m_1} et un filtre désadapté \mathbf{q}^{m_2} s'écrit :

$$h_p(\mathbf{s}^{m_1}, \mathbf{q}^{m_2}) := \left\| \mathbf{F}_{\delta_{m_1, m_2}} \text{AXCF}(\mathbf{s}^{m_1}, \mathbf{q}^{m_2}) \right\|_p, \quad \text{pour } p \in \mathbb{N}^*, \quad (4.23)$$

où δ dénote le symbole de Kronecker et $\mathbf{F}_{\delta_{m_1, m_2}}$ est une matrice diagonale de taille $K + N - 1$, spécifiée par le vecteur $[1, \dots, 1, 1 - \delta_{m_1, m_2}, 1, \dots, 1]$ (un vecteur composé uniquement de 1, sauf à la position $N + l$, si $m_1 = m_2$). Pas de surprise de ce côté-là. Deux petites remarques sur ces définitions. D'une part, la fonction h_p est convexe, pour les mêmes raisons que nous avons exprimées précédemment. D'autre part, en affectant un filtre désadapté à une séquence, on peut différencier l'autocorrélation de l'inter-corrélation. Même s'il était clairement sous-entendu dans les parties précédentes qu'on souhaitait étudier la première, rien ne nous le garantissait lorsqu'on examine isolément l'expression (4.4).

Pas de surprise également du côté du problème d'optimisation. Avec, derechef, un abus de notation, on décide de poser :

$$(P) \begin{cases} \min_Q f_{p,S}(\mathbf{q}^1, \dots, \mathbf{q}^M) = \sum_{m_1, m_2}^M h_{p,S}(\mathbf{s}^{m_1}, \mathbf{q}^{m_2}), \\ \text{t.q. } (\mathbf{s}^m)^H \mathbf{q}^m = (\mathbf{s}^m)^H \mathbf{s}^m, \quad \forall m \in \llbracket 1, M \rrbracket, \end{cases} \quad (4.24)$$

où la fonction $h_{p,s}$ est l'application partielle de h_p en \mathbf{s} associée au filtre \mathbf{q} . La fonction coût $f_{p,S}$ s'écrit comme une somme de fonctions convexes par composantes. L'est-elle pour autant pour tout ensemble Q ? Oui, et la remarque suivante va nous aider à le constater.

Remarque 4.7. (*Optimisation multi-objectif*) *A l'instar du chapitre précédent, l'optimisation d'un ensemble de séquences et des filtres désadaptés associés est un problème d'optimisation multi-objectif. Il existe donc plusieurs façons de poser le problème : l'expression (4.24) est l'une d'entre elles. Mais on aurait pu par exemple considérer le lobe secondaire le plus haut de l'ensemble, c'est-à-dire :*

$$f_{p,S}(\mathbf{q}^1, \dots, \mathbf{q}^M) := \max_{m_1, m_2} h_{p,S}(\mathbf{s}^{m_1}, \mathbf{q}^{m_2}), \quad (4.25)$$

Le problème d'optimisation induit reste heureusement convexe, et permet ainsi de tomber sur un autre point du front de Pareto du problème d'optimisation multi-critère. Dans les simulations qui suivront, nous avons employé les deux fonctions de coût, et la première a donné de meilleurs résultats en terme de PSLR moyen (sans raison connue)...

Remarque 4.8. (*Convexité, écriture matricielle*) *La fonction objectif $f_{p,S}$ du problème d'optimisation (4.24) a été définie à l'aide d'une somme dont les opérandes $h_{p,S}(\mathbf{s}^{m_1}, \mathbf{q}^{m_2})$ ne dépendent que d'un seul filtre désadapté. Afin de démontrer la convexité de $f_{p,S}$, il est plus simple de décrire ces dernières en fonction de la globalité de ses variables. D'ailleurs, notons par \mathbf{Q} la concaténation verticale des filtres désadaptés :*

$$\mathbf{Q} = \left[(\mathbf{q}^1)^T, \dots, (\mathbf{q}^M)^T \right]^T \in \mathbb{C}^{MK}. \quad (4.26)$$

Pour tout $m_1, m_2 \in \llbracket 1, M \rrbracket$, soit $\mathbf{H}_S^{m_1, m_2}$ une matrice par blocs telle que :

$$\mathbf{H}_S^{m_1, m_2} := \left[\underbrace{\mathbf{0}_{K+N-1, K}, \dots, \mathbf{0}_{K+N-1, K}}_{m_2 - 1 \text{ bloc}(s)}, \mathbf{F}_{\delta_{m_1, m_2}} \mathbf{\Lambda}_K((\mathbf{s}^{m_1})^*), \underbrace{\mathbf{0}_{K+N-1, K}, \dots, \mathbf{0}_{K+N-1, K}}_{M - m_2 \text{ bloc}(s)} \right].$$

Bloc en position m_2
↓

A partir d'une fonction $H_{p,S}^{m_1, m_2}$ qui en découle :

$$H_{p,S}^{m_1, m_2}(\mathbf{Q}) := \left\| \mathbf{H}_S^{m_1, m_2} \mathbf{Q} \right\|_p, \quad (4.27)$$

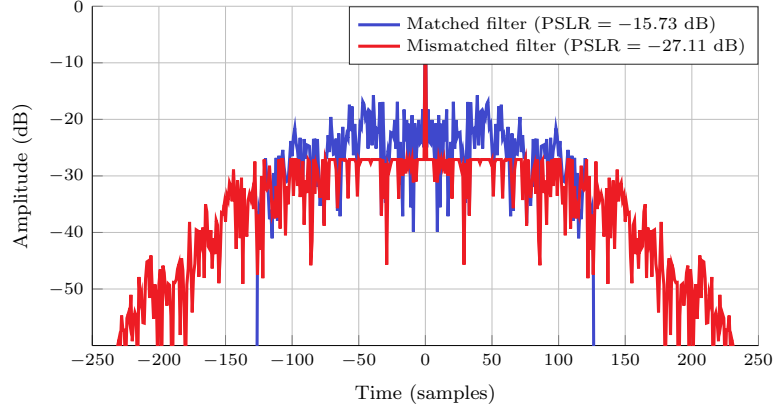


FIGURE 4.2 – Filtrage désadapté optimal pour un code de phase
*Initialisation aléatoire de longueur $N = 127$. Filtre désadapté de longueur $3N$.
 Perte autorisée de 1 dB. Perte effective de 1 dB.*

on peut facilement vérifier que :

$$h_p(\mathbf{s}^{m_1}, \mathbf{q}^{m_2}) = H_{p,S}^{m_1, m_2}(\mathbf{Q}). \quad (4.28)$$

Finalement, la fonction objectif $f_{p,S}$ s'exprime bien à l'aide d'une somme de composantes convexes sur l'ensemble de ses variables. Elle l'est donc tout autant [Boyd et al., 2004].

Ainsi, comme pour le cas mono-varié, le problème d'optimisation (P) est convexe. Cela signifie qu'il est relativement accessible de déterminer, pour un ensemble de séquences $S = \{\mathbf{s}^1, \dots, \mathbf{s}^M\}$, son ensemble optimal de filtres désadaptés associés, pour un certain critère.

Remarque 4.9. (*Applications avec Doppler*) Une application possible concerne la détection d'une cible dont la vitesse est inconnue [Rabaste et al., 2015]. Cette vitesse induit un décalage en phase du signal émis ; on peut donc par exemple envisager un ensemble de séquences composé du signal émis, et de ce même signal sous différents Doppler.

4.1.4 Résultats

Le présent paragraphe se consacre à la résolution du problème d'optimisation pour la recherche d'un filtre désadapté optimal. Elle présente ainsi une kyrielle de résultats, qui varient selon un paramètre. En y réfléchissant, ce paragraphe a plus ou moins été introduit avec la Figure 4.1, qui applique un filtre désadapté optimal sur un *chirp*...

4.1.4.1 Passage introductif

De même, appliquons un filtre désadapté optimal sur un code de phase, Figure 4.2. En tant que figure introductive, on donne une pléthore de détails sur sa mise en place, en vrac.

- L'optimisation a été réalisée sur un code de phase de longueur $N = 127$, initialisé aléatoirement et sous la contrainte de module constant.
- Quant au filtre désadapté, il a été choisi trois fois plus long, car cette longueur offre — en matière de lobes secondaires — les meilleures performances. Ce point sera d'ailleurs évoqué dans une prochaine figure.
- Sauf mention contraire, on s'intéresse ici et dans les figures suivantes au critère du PSLR, ce qui est peu ou prou équivalent à la résolution du problème (P_∞).
- Ce dernier étant convexe, un algorithme à convergence locale comme une descente de gradient [Boyd et al., 2004] pourrait permettre de conclure (à condition de trouver une façon de considérer la contrainte sur la perte). Néanmoins, en pratique, nous avons

utilisé CVX, « une bibliothèque permettant de définir et de résoudre des problèmes convexes » [Grant *et al.*, 2014].

- La désadaptation entraîne une perte en gain de traitement qui, comme nous avons pu le remarquer, peut être contrôlée à l’aide d’une contrainte. Celle-ci a été fixée et atteinte à 1 dB. Signalons de ce fait que les courbes de la Figure 4.2, ainsi que les suivantes, ont été normalisées, afin qu’on puisse comparer les lobes secondaires.
- Enfin, permettons-nous quelques observations sur les performances finales. Le filtre désadapté optimal a apporté un gain d’environ 11 dB par rapport à une compression d’impulsion classique (de -15.7 dB à -27 dB) avec, toujours, cette signature « plate », représentative d’une optimisation sur le PSLR.

Longueur du filtre Commençons par une variation de la longueur du filtre désadapté. Les résultats sont présentés aux Figures 4.3 et 4.4. La première présente le niveau des corrélations en sortie pour plusieurs filtres désadaptés de différentes longueurs. Le résultat est relativement attendu : la longueur du filtre se coudoie avec le niveau des lobes secondaires. Ce comportement se confirme avec les courbes décroissantes de la seconde figure, qui représente le PSLR en fonction de la longueur additionnelle l (pour rappel, le filtre désadapté est ainsi de taille $K = N + 2l$). De toute manière, il ne pouvait en être autrement. En effet, le « pire » consisterait à ce que les courbes soient constantes : le caractère optimal impliquerait alors que les différents filtres soient identiques à celui de taille N , mais entourés de zéros (pour le problème du PSLR, il faut également faire attention à la normalisation).

Remarquons sur la Figure 4.4 un effet de plateau sur les dernières longueurs de filtres. Cette particularité semble spécifique à ce code de phase ; nous allons d’ailleurs prochainement justifier cette assertion à l’aide d’un contre-exemple. Néanmoins, cette figure permet de préciser un argument sur la longueur du filtre. Faut-il nécessairement prendre le plus long ? On aurait tendance à répondre par la positive pour des questions de performances. En pratique, il faut modérer en ajoutant qu’un filtre trop long tend à étaler les lobes secondaires sur une distance plus grande, et peut apporter ceux provenant d’une cible lointaine... Le critère de l’ISL est d’ailleurs fortement affecté : plus le filtre est long, plus l’ISL va augmenter, sans nécessairement diminuer le PSLR...

Pertes Sur les figures précédentes, la contrainte sur la perte en gain de traitement (LPG) a été fixée à 1 dB. On se demande maintenant si celle-ci a une influence sur les performances que l’on peut atteindre. Pour cela, on se propose, à partir d’une même initialisation, de résoudre le problème (P_∞) en jouant sur la valeur de cette contrainte. La Figure 4.5 en résulte, traçant le niveau des lobes secondaires normalisé (*i.e.* les pertes ne sont pas prises en compte) selon la valeur de ladite contrainte. Deux cas de figure sont observables.

- Le premier correspond à une zone où la contrainte sur la perte en gain de traitement est atteinte. Cette « augmentation » du LPG a permis de réduire les lobes secondaires de manière significative.
- Le second cas de figure correspond à un stade où la contrainte sur la perte n’est plus atteinte. Ici, nous sommes arrivés à — pour la longueur de filtre considéré — la valeur minimale du PSLR possible. La relaxation sur le LPG ne permet plus de l’améliorer.

En pratique, il ne faut pas oublier que le consentement d’une perte induit une détection plus difficile des cibles noyées dans le bruit...

Performances mono-séquences Intéressons nous maintenant aux performances du filtrage désadapté. Pour cela, on se propose de mesurer le gain apporté par le filtrage désadapté (comparé à un filtrage adapté) selon la longueur de la séquence. Les résultats moyens sur une cinquantaine d’instances tirées aléatoirement sont illustrés à la Figure 4.6. Les variations de ces courbes, respectivement décroissantes et croissantes, sont attendues, étant donné l’ajout de degrés de libertés (longueur de la séquence *a minima* pour la première, longueur du filtre pour la seconde).

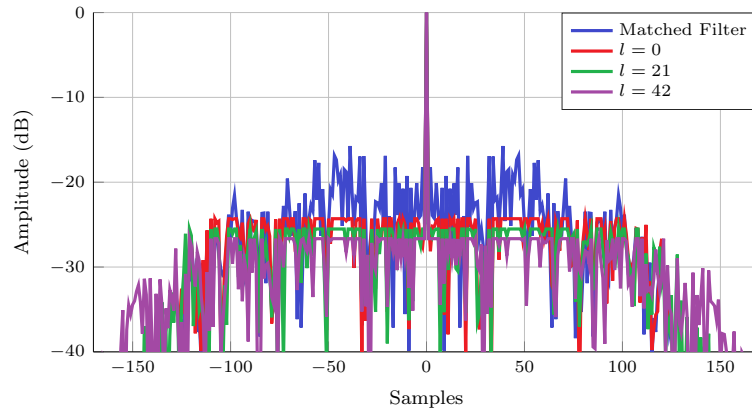


FIGURE 4.3 – Corrélation pour différentes longueurs de filtres désadaptés
Veillez noter un gain par rapport à la compression d'impulsion classique, même pour un filtre de même longueur.

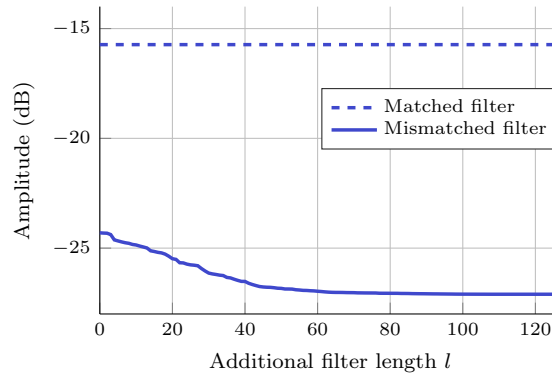


FIGURE 4.4 – PSLR selon la longueur du filtre désadapté
*Pour les deux figures : initialisation aléatoire de longueur $N = 127$.
 Perte autorisée de 1 dB.*

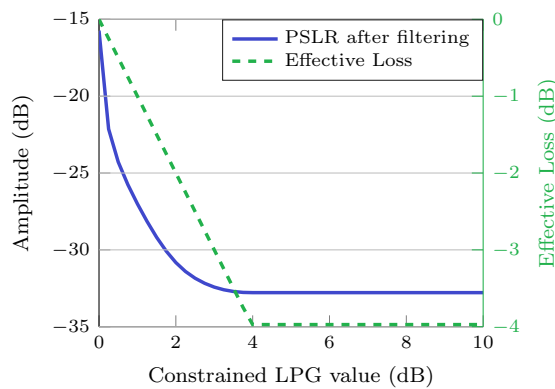


FIGURE 4.5 – PSLR selon la contrainte de perte
*Initialisation aléatoire de longueur $N = 127$, identique aux précédentes.
 Axe des ordonnées de gauche : PSLR normalisé (en bleu).
 Axe des ordonnées de droite : LPG obtenu (en vert).
 Filtre désadapté de longueur $3N$.*

Optimisation locale, élargissement du lobe principal Souvenez-vous : la Section 3.2 a introduit le concept d'optimisation locale. Cela consiste à se préoccuper essentiellement des lobes secondaires autour du lobe principal, tout en ignorant les autres. Cette prévenance est à considérer dans la pratique lorsqu'on veut se focaliser sur les cibles proches — tout en partant de l'hypothèse qu'il n'y a aucune cible forte lointaine. Dans ladite section, l'optimisation a été effectuée sur la séquence \mathbf{s} , à l'aide de coefficients de pondérations. Comme on s'intéresse ici à la génération du filtre désadapté \mathbf{q} , il n'est pas nécessaire, d'une part, que la séquence \mathbf{s} ait initialement des caractéristiques locales, et d'autre part, d'inclure un vecteur de pondération. En effet, ce rôle est déjà pris en compte par la matrice \mathbf{F} , et plus précisément par sa diagonale. En reprenant les écritures précédentes (provenant de la Section 3.2) pour une optimisation locale, cette dernière peut être mise sous la forme :

$$\left[\underbrace{0, \dots, 0}_{\text{Pour ignorer les bords}}, 1, \dots, 1, \underbrace{0, 1, \dots, 1}_{\text{Pour ignorer le lobe principal}}, 0, \dots, 0 \right].$$

Pour un élargissement du lobe principal, il suffit de relâcher la contrainte sur celui-ci :

$$\left[1, \dots, 1, \underbrace{0, \dots, 0}_{\text{Pour relâcher le lobe principal}}, 1, \dots, 1 \right].$$

Il est possible de faire cela de manière plus élégante en « arrondissant » le lobe, c'est-à-dire en définissant une progression plus douce. Il suffit pour cela d'ajouter des contraintes sur les valeurs du lobe principal, contraintes qui s'avèrent également être convexes.

Des exemples d'optimisations locales sont présentées à la Figure 4.7. La première consiste en une réduction des lobes secondaires autour du lobe principal. La zone considérée, exagérément petite pour mettre en valeur les effets de cette optimisation, est composée d'une quarantaine d'éléments, par rapport aux $N = 127$ de la séquence d'origine, tandis que le filtre désadapté est choisi trois fois plus long. Remarquez que nous avons, en quelque sorte, obtenue une *ZCZ* (*Zero Correlation Zone*), avec des lobes dérisoires autour des -150 dB. Le niveau des autres lobes est équivalent à une initialisation aléatoire. Par ailleurs, la seconde figure dépeint le fait qu'une relaxation autour du *mainlobe* n'offre pas toujours un avantage si important que cela...

Conclusion Ces premières simulations ont permis d'illustrer les avantages à employer un filtre désadapté. Un côté pratique, puisqu'il est relativement aisé — et peu coûteux — d'obtenir un filtre désadapté optimal selon un critère donné, par convexité. Au demeurant, la question sur le choix de la méthode d'optimisation à appliquer ne s'est même pas posée. Une souplesse certaine, car le filtrage désadapté offre des libertés supplémentaires en ses paramètres (aucune contrainte de module constant, longueur du filtre, perte autorisée), ce qui apporte généralement un gain non négligeable sur les performances.

4.1.4.2 Galerie multi-séquence

La Section 4.1.3 a suggéré une extension de la définition d'un filtre désadapté optimal pour un ensemble de séquences. Bien que cette définition n'est pas nécessairement unique « Pourquoi diable faire une somme sur les différentes fonctions objectifs ? », on se propose ici

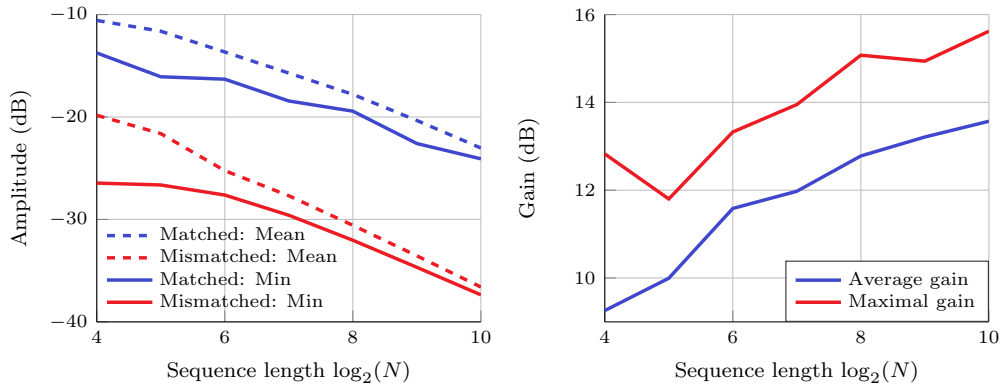


FIGURE 4.6 – Niveau des lobes secondaires selon la longueur, cas mono-séquence
A gauche : PSLR selon la longueur de la séquence N .
A droite : gain selon cette même longueur, par rapport à un filtrage adapté.
Résultats obtenus après 50 instances, initialisées aléatoirement.
Filtre désadapté de longueur $3N$. Perte autorisée de 1 dB.

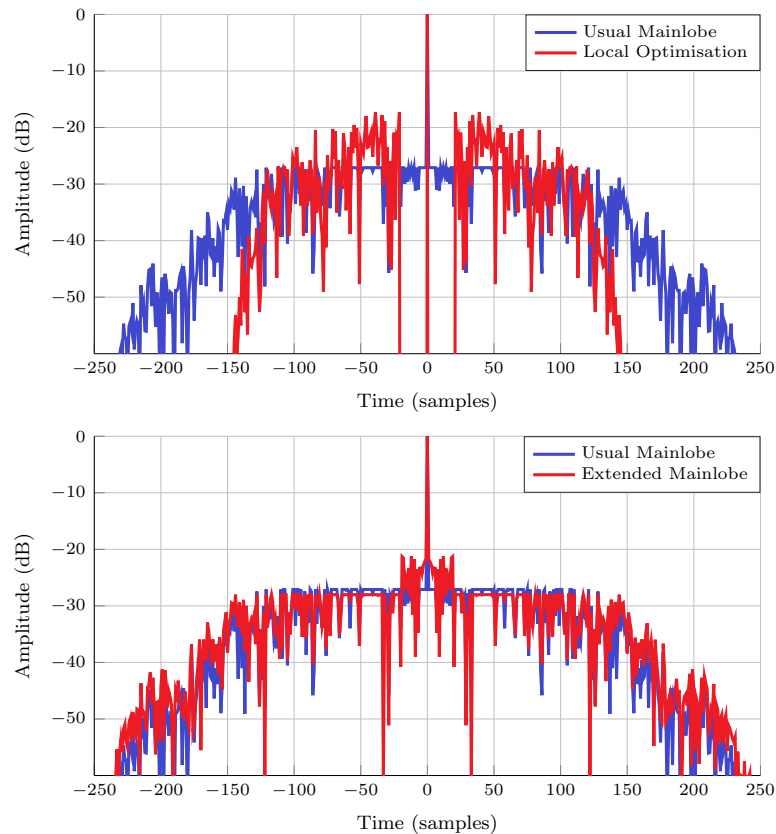


FIGURE 4.7 – Optimisations locales du filtre désadapté
En haut : optimisation locale autour du lobe principal. La région autour du lobe principal, exagérément petite (à ne pas faire en pratique), présente des lobes secondaires aux alentours de -150 dB.
En bas : relaxation des contraintes autour du lobe principal.

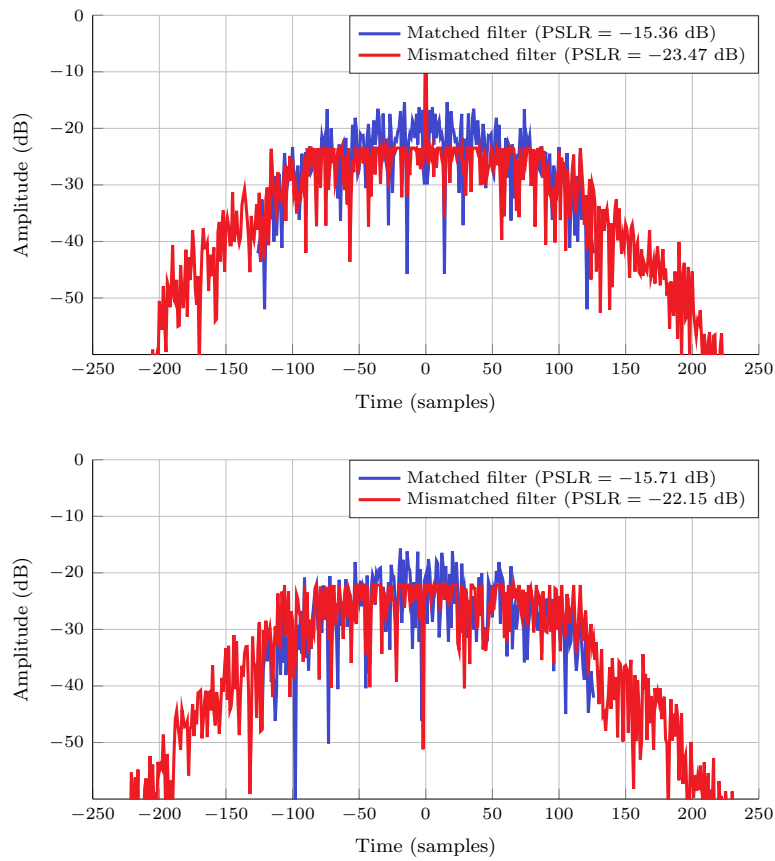


FIGURE 4.8 – Optimisation d’un ensemble de $M = 2$ séquences, avec filtrage désadapté
En haut : « autocorrélation » de la première séquence.
En bas : « inter-corrélation » entre la première séquence et le second filtre.
Initialisation aléatoire. Perte autorisée de 1 dB. Filtre désadapté de longueur $3N$.

d’effectuer quelques simulations afin d’illustrer les solutions en sortie. Comme leur attitude est plus ou moins similaire à celui du cas mono-séquence, on restera très synthétique...

Débutons par une optimisation de deux séquences, disons s^1 et s^2 , associées à deux filtres désadaptés. Pour chaque séquence, on souhaite qu’elle présente une bonne « autocorrélation » avec son conjoint, et une bonne « inter-corrélation » avec les autres. La Figure 4.8 représente ainsi les corrélations d’une séquence de longueur $N = 127$. Remarquons simplement que le gain par rapport à un filtrage adapté est moins important que dans le cas précédent, ce qui est légitime puisque le problème est beaucoup plus contraint. Mais il reste cependant appréciable : ici, il est de l’ordre de 7 dB. Le profil de la seconde séquence est relativement similaire.

On s’amuse ensuite à faire fluctuer le nombre M et la longueur N des séquences d’un ensemble. Les résultats, obtenus après une centaine d’instances initialisées aléatoirement, sont présentés à la Figure 4.9. Permettons-nous deux constatations, très similaires à celles insinuées à la Section 3.5, dans le cas adapté :

- Doubler la longueur de séquence permet de gagner environ 3 dB. On retrouve le même comportement que dans le cas adapté, malgré l’utilisation d’un filtre désadapté trois fois plus étendu.
- En augmentant le nombre de séquences dans l’ensemble, l’apport du filtrage désadapté s’amenuise. Comme mentionné plus tôt, on peut justifier cela par le fait que les filtres désadaptés deviennent de plus en plus contraints.

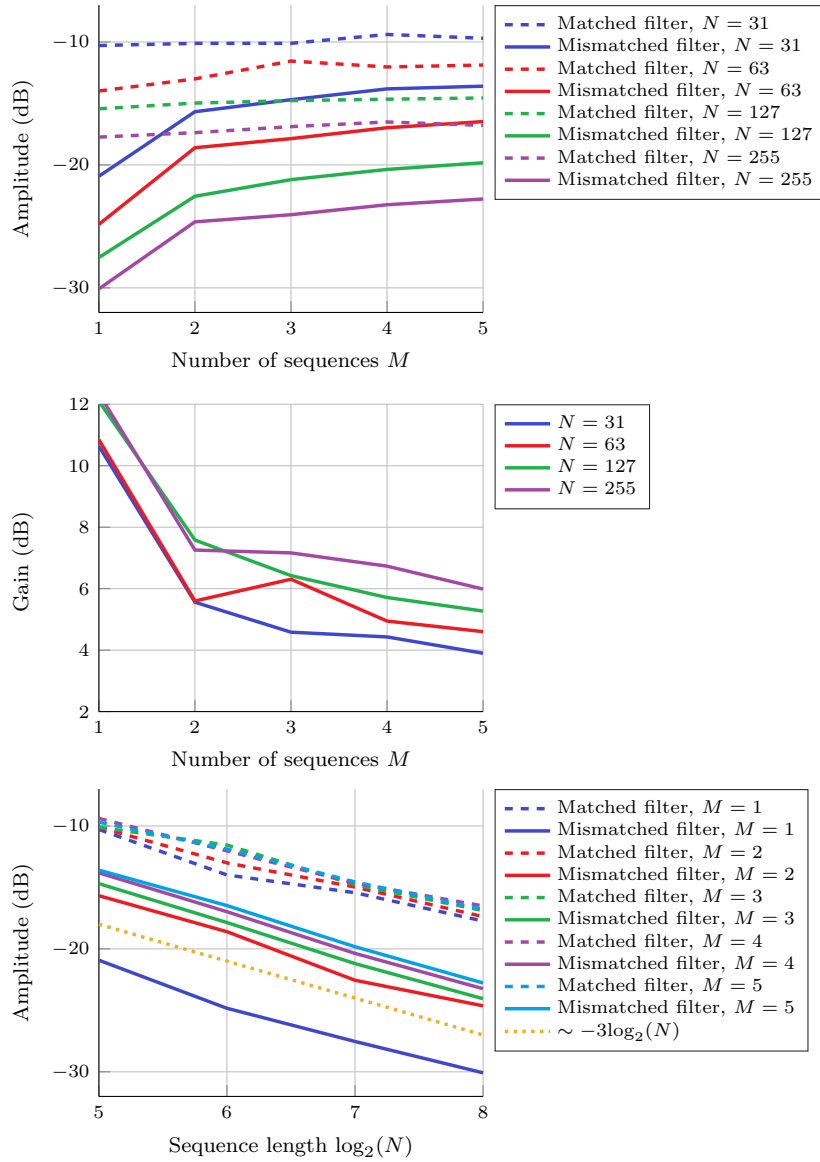


FIGURE 4.9 – PSLR selon différents paramètres, avec filtrage désadapté.

Résultats obtenus après 100 instances. Initialisations aléatoires.

Filtre désadapté de longueur $3N$. Perte autorisée de 1 dB.

En haut : lobes secondaires selon le nombre de séquences M de l'ensemble.

Au centre : différence (gain) entre le filtrage adapté et le filtrage désadapté.

En bas : lobes secondaires selon la longueur de séquence N .

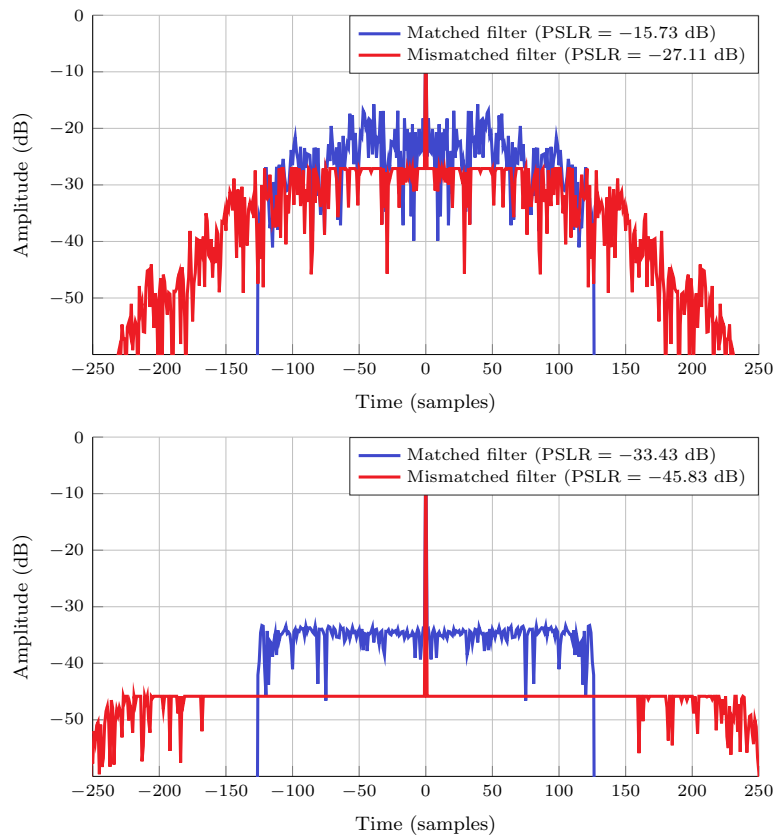


FIGURE 4.10 – Filtrage désadapté optimal pour un code de phase
En haut : initialisation aléatoire d’une séquence de longueur $N = 127$.
En bas : initialisation optimisée au sens du filtre adapté. Filtre désadapté de longueur $3N$.
Perte autorisée de 1 dB. Perte effective de 0.5 dB.

4.1.4.3 Passerelle transitoire

Les résultats antérieurs permettent d’apprécier les avantages du filtrage désadapté optimal : facilité de sa mise en œuvre, performances accrues. Mais nous ne nous sommes peut-être pas assez évertués à insister sur le joug de l’optimalité envers la séquence initiale : il existe un filtre désadapté optimal *selon* un critère, et, surtout, pour une séquence *donnée*. Cela est sans doute bête, mais un filtre désadapté associé à une séquence n’est certainement pas optimal pour une autre...

Optimisation séquentielle Dans les chapitres précédents, nous avons pu identifier un certain écart sur les sorties des compressions d’impulsion entre une initialisation aléatoire et une séquence optimisée. On pourrait dans la même veine mettre en opposition les optimisations « adapté » et « désadapté » — nous allons d’ailleurs brièvement le faire en une phrase ! —, mais pourquoi ne pas coupler les deux ? Les figures suivantes s’occupent de cela.

Pour cela, on se propose tout d’abord de reprendre une étude précédente. Le haut de la Figure 4.10 est donc, à la légende près, identique à la Figure 4.2. Pour rappel, cette dernière comparait les corrélations d’une séquence s^0 de longueur $N = 127$, initialisée aléatoirement, sur laquelle on avait appliqué un filtre adapté ou un filtre désadapté optimal. Le lobe secondaire le plus élevé était aux alentours des -27 dB.

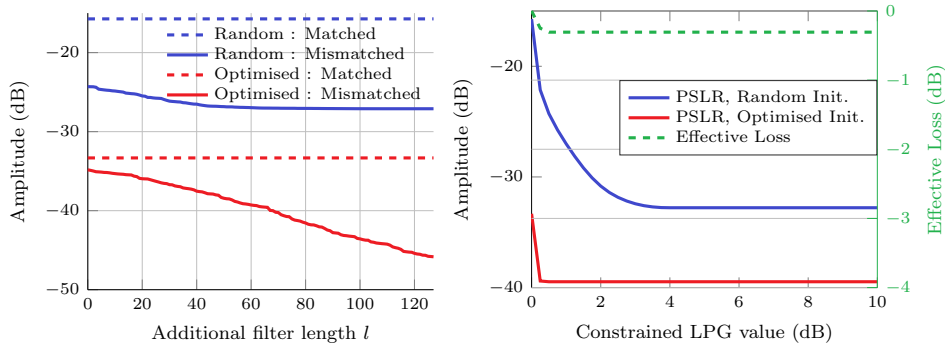


FIGURE 4.11 – PSLR selon la longueur du filtre et la perte, initialisation optimisée
 A gauche : selon la longueur additionnelle l du filtre désadapté. Perte autorisée de 1 dB.
 A droite : selon la contrainte de perte. Filtre désadapté de longueur $3N$.

Repartons ensuite de la séquence \mathbf{s}^0 . On se propose de l’optimiser dans le sens du filtre adapté, à l’aide d’une descente de gradient, comme expliqué dans le Chapitre 2. La séquence qui en découle, que l’on nomme ici \mathbf{s} , n’a pas nécessairement un lien direct avec son aînée. D’ailleurs, celle-ci présente un PSLR de l’ordre de -33 dB (Figure 4.10, en bleue) : pour cet exemple précis, on peut ainsi décerner un petit avantage pour l’optimisation de la séquence elle-même, et non pas de son filtre.

Cela dit, on pourrait également appliquer un filtre désadapté optimal sur la nouvelle séquence \mathbf{s} . En procédant comme tel, il est intéressant de noter que le lobe le plus nuisible affiche -45.8 dB : une amélioration finalement équivalente à celle fournie par le filtre désadapté appliqué à la séquence initiale \mathbf{s}^0 , et cela malgré une séquence d’origine de meilleure qualité. Avec du recul, on peut se dire que la méthodologie que nous venons de relater nous a permis de « gagner » 30 dB...

Résonances La méthodologie présentée ci-dessus semble offrir un couple séquence/filtre relativement intéressant (néanmoins sans aucune réelle preuve de convergence, malheureusement). Si on se résume, elle est composée de deux étapes : l’optimisation d’une séquence dans le sens du filtre adapté, suivie d’une application d’un filtre désadapté optimal.

Effectuons une brève étude sur les performances atteignables à l’aide de celle-ci. Pour commencer, la Figure 4.11 se propose de faire varier la longueur du filtre et la contrainte sur la perte liée à la désadaptation. Pour la première situation, une décroissance des lobes secondaires est observable ; anticipable, certes, mais avec en plus une absence des effets de plateau comme à la Section 4.1.4.1. Comme mentionné précédemment, il est intéressant de noter que le gain apporté par un filtrage désadapté est au final assez similaire, que ce soit pour une séquence aléatoire ou une séquence optimisée, malgré les différences initiales. Pour la seconde situation, un échelonnement de la contrainte sur la perte, on retrouve le stade où cette dernière est — involontairement — ignorée (sauf pour des LPG très faibles). Il ne semble *a priori* pas si utile de s’autoriser une grande perte, ce qui concorde avec les besoins pratiques...

Les Figures 4.12 et 4.13 se focalisent sur deux paramètres intrinsèques aux séquences \mathbf{s} , à savoir leur nombre et leur longueur. Mis à part la différence sur les lobes, mentionnée maintes fois dans ce paragraphe, on retrouve le comportement décrit dans le cas d’une initialisation aléatoire, à savoir... les commentaires de la page 127. Ajoutons simplement que le filtrage désadapté s’avère être réellement performant dans le cas mono-séquence seulement lorsqu’il est appliqué aux séquences optimisées. On observe effectivement dans ce cas un gain très léger dès que le nombre de séquences considérées est supérieur ou égal à 2. On essaiera de justifier ce comportement plus loin dans ce chapitre...

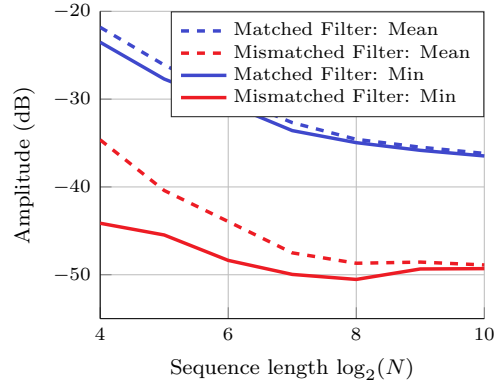


FIGURE 4.12 – PSLR selon la longueur de séquence, cas mono-séquence
Initialisation optimisée. Filtre désadapté de longueur $3N$. Perte autorisée de 1 dB.

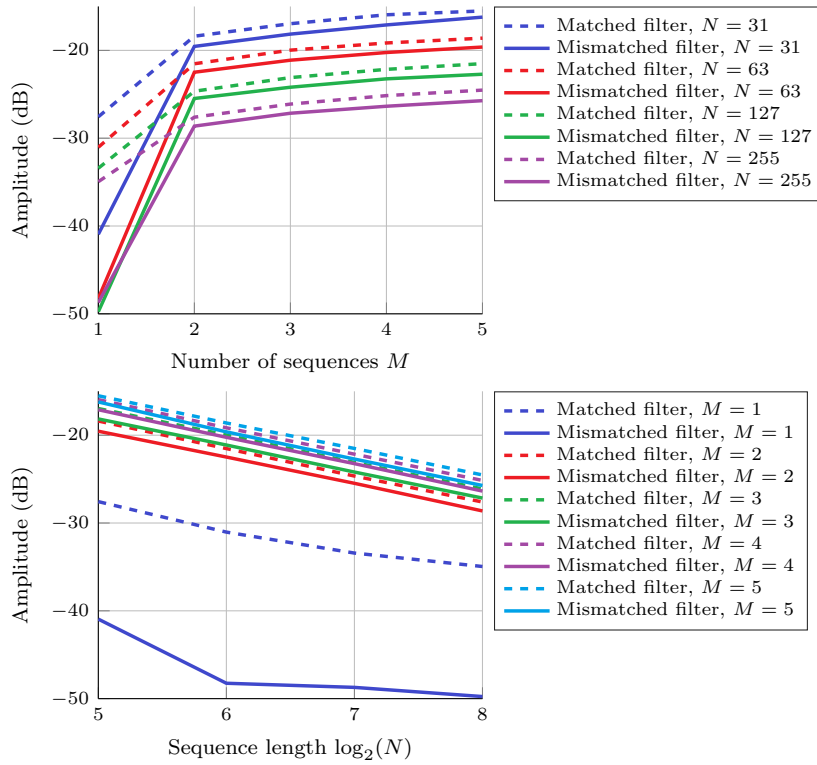


FIGURE 4.13 – PSLR selon différents paramètres, filtrage désadapté et initialisation optimisée

Résultats obtenus après 100 instances. Initialisations optimisées.

Filtre désadapté de longueur $3N$. Perte autorisée de 1 dB.

En haut : lobes secondaires selon le nombre de séquences M de l'ensemble.

En bas : lobes secondaires selon la longueur de séquence N .

4.1.5 Conclusion

Dans une optique radar, le filtrage consiste à établir une comparaison — lire, une corrélation — entre le signal reçu et un filtre. On parle de filtrage adapté lorsque ce dernier est peu ou prou une réplique du signal émis, et de filtrage désadapté sinon. D'ailleurs, tel était le sujet de cette section.

Le filtre désadapté offre certaines libertés : il ne souffre pas de la contrainte de module constant, il y a possibilité d'avoir une différence de taille entre la séquence et lui-même... Mais, surtout, le filtre désadapté vit dans un espace convexe. De ce fait, le problème d'optimisation prospectant un filtre désadapté l'est aussi *a fortiori*. Il est donc relativement aisé, pour une séquence *donnée*, de déterminer son filtre désadapté optimal, *selon* un certain critère. Dans la même veine, nous avons proposé une extension pour un ensemble de séquences et leurs filtres désadaptés associés. Dans les deux cas, il est possible d'ajouter une contrainte, convexe de surcroît, afin de contrôler la perte due à la désadaptation.

À l'aide de quelques simulations, nous avons pu observer qu'un filtre désadapté offrait un gain notable par rapport à une compression d'impulsion classique. Et ce, quel que soit le choix du paramètre : longueur du filtre, perte autorisée... Deux points restent néanmoins à élucider. Premièrement, en proposant une optimisation séquentielle — optimisation de la séquence dans le sens du filtre adapté, puis application d'un filtre désadapté optimal — on obtient des performances accrues. Cela souligne le fait que le filtre désadapté est optimal pour une séquence *donnée*, mais pas n'importe laquelle. Comment alors choisir un couple séquence/filtre idéal ? Deuxièmement, le filtre désadapté semble perdre de son intérêt pour un ensemble de séquences. Est-ce vraiment le cas ? On tentera d'élucider ces deux points dans les sections suivantes.

Une partie de ces résultats a été présentée dans [Tan et al., 2016c].

4.2 Optimisation jointe d'une séquence et d'un filtre

La section précédente s'est intéressée à la recherche d'un filtre désadapté optimal, mais ceci pour une séquence donnée. Dans l'optique d'atteindre un bon niveau de lobes secondaires, le choix de cette séquence a son importance. Comme l'illustre la Section 4.1.4, une séquence dont la sortie après filtrage adapté a été améliorée semble rester plus avantageuse qu'une autre tirée aléatoirement. Mais alors, comment choisir cette séquence ? La présente section tente de répondre à cette question, en proposant deux protocoles permettant d'optimiser, à la fois, la séquence et le filtre : une optimisation dite alternée, et une autre qu'on qualifiera de jointe.

Une écriture générique d'un problème d'optimisation jointe est la suivante :

$$(P_{\text{Joint}}) \begin{cases} \min_{\mathbf{s}, \mathbf{q}} & f_p(\mathbf{s}, \mathbf{q}) \\ \text{t.q.} & \mathbf{s}^H \mathbf{q} = \mathbf{s}^H \mathbf{s}, \\ & 10^{\alpha/10} \mathbf{s}^H \mathbf{s} \geq \mathbf{q}^H \mathbf{q}. \end{cases} \quad (4.29)$$

où f_p est une mesure générique des lobes secondaires définie à l'équation (4.11). Remarquons que la seule modification par rapport au problème précédent (P_p) se situe au niveau des variables d'optimisation, avec l'introduction de la séquence \mathbf{s} .

4.2.1 Optimisation alternée

Un algorithme d'optimisation alternée [Stoica *et al.*, 2004, Yang *et al.*, 2010] est un algorithme itératif qui consiste en la résolution successive de plusieurs problèmes d'optimisation. Chacun d'entre eux touche une partie disjointe des variables, les autres étant fixées comme paramètres. Dans notre cas, il convient assez naturellement de considérer un problème d'op-

Algorithme Optimisation alternée	
Données	Initialisation de la séquence \mathbf{s}
Répéter	1. Résolution exacte du problème $(P_{\text{Alt}(\mathbf{q})})$ Mise à jour du filtre désadapté \mathbf{q}
	2. Résolution du problème $(P_{\text{Alt}(\mathbf{s})})$ Mise à jour de la séquence \mathbf{s}
Jusqu'à	Un critère d'arrêt est atteint.

TABLE 4.1 – Algorithme d'optimisation alternée
Notez que la première étape permet d'obtenir un filtre désadapté optimal.

timisation pour la séquence \mathbf{s} , et un second pour le filtre désadapté \mathbf{q} :

$$(P_{\text{Alt}(\mathbf{q})}) \left\{ \begin{array}{l} \min_{\mathbf{q}} f_{p,\mathbf{s}}(\mathbf{q}) = f_p(\mathbf{s}, \mathbf{q}) \\ \text{t.q. } \mathbf{s}^H \mathbf{q} = \mathbf{s}^H \mathbf{s}, \\ 10^{\alpha/10} \mathbf{s}^H \mathbf{s} \geq \mathbf{q}^H \mathbf{q}. \end{array} \right. \quad \text{et} \quad (P_{\text{Alt}(\mathbf{s})}) \left\{ \begin{array}{l} \min_{\mathbf{s}} f_{p,\mathbf{q}}(\mathbf{s}) = f_p(\mathbf{s}, \mathbf{q}) \\ \text{t.q. } \mathbf{s}^H \mathbf{q} = \mathbf{s}^H \mathbf{s}. \end{array} \right. \quad (4.30)$$

Le problème de gauche, $(P_{\text{Alt}(\mathbf{q})})$, ne vous rappelle-t-il rien ? Celui-ci se définit comme la recherche d'un filtre désadapté à partir d'une séquence donnée, et peut donc s'apparenter aux différents problèmes traités dans la section précédente (prise en compte de la contrainte sur la perte par exemple). Quant au problème de droite, la variable d'optimisation est la séquence \mathbf{s} , qui vit par définition dans un espace non convexe. La résolution exacte de ce problème est particulièrement difficile mais, comme on l'a vu dans les chapitres précédents, de nombreuses méthodes (descente de gradient, relaxation...) permettent de dénicher des solutions acceptables.

Le terme « optimisation cyclique » est également utilisé pour évoquer une optimisation alternée [Soltanianian *et al.*, 2013]. Cela prend tout son sens car l'algorithme est en effet itératif : on résout les deux problèmes à la suite, et on répète ensuite cette opération jusqu'à convergence. Le Tableau 4.1 résume ces grandes lignes. Remarquez au passage que la procédure décrite dans la Section 4.1.4.3 se révèle être en fait une itération d'un algorithme cyclique...

Remarque 4.10. *Deux critères d'arrêt sont communément utilisés :*

- une limitation du nombre d'itérations, permettant d'obtenir un résultat dans un temps imparti,
- un critère d'arrêt sur la convergence (peu de modifications entre deux itérations par exemple).

L'emploi d'une optimisation cyclique peut se justifier par une convergence algorithmique [Wu *et al.*, 2015]. En effet, soit $(\mathbf{s}_i, \mathbf{q}_i)$ le couple séquence-filtre obtenu lors de la i -ème itération, et soit $(f_i)_{i \in \mathbb{N}}$ une suite de valeurs telle que $f_i := f_p(\mathbf{s}_i, \mathbf{q}_i)$. On peut écrire :

$$f_p(\mathbf{s}_i, \mathbf{q}_i) \leq f_p(\mathbf{s}_{i-1}, \mathbf{q}_i) \leq f_p(\mathbf{s}_{i-1}, \mathbf{q}_{i-1}). \quad (4.31)$$

Ces inégalités se justifient en examinant chacune des étapes de l'algorithme. L'itération i commence à partir du couple $(\mathbf{s}_{i-1}, \mathbf{q}_{i-1})$. La résolution du problème $(P_{\text{Alt}(\mathbf{q})})$ nous donne un filtre désadapté \mathbf{q}_i qui, nécessairement, réduit la fonction coût (dans le cas contraire, l'algorithme a convergé). En d'autres termes, cela s'écrit : $f(\mathbf{s}_{i-1}, \mathbf{q}_i) \leq f(\mathbf{s}_{i-1}, \mathbf{q}_{i-1})$, ce qui correspond à l'inégalité de droite. On peut démontrer de la même façon la seconde inégalité. La convergence de la suite $(f_i)_i$ est due au fait qu'elle peut être minorée par 0, car la fonction f_p est basée sur une norme. Cependant, d'un point de vue analytique, il est difficile de spécifier la nature du résultat obtenu, bien que celle-ci sera sans doute influencée par le choix de la méthode d'optimisation pour résoudre le problème $(P_{\text{Alt}(\mathbf{s})})$.

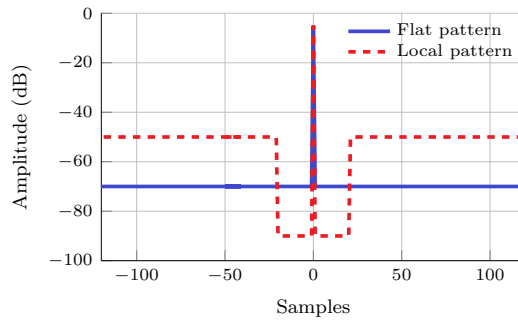


FIGURE 4.14 – Exemples de gabarits

Le gabarit décrit la forme de la séquence de corrélation que l'on souhaite. La perte est notamment modélisée par le pic principal, non normalisée. Un gabarit « plat » permet de se focaliser sur le PSLR, mais il est plus difficile d'en définir un autre pour l'ISL...

Un exemple d'optimisation cyclique sur une séquence et son filtre a été proposée dans [Karbasi *et al.*, 2015] ou [Soltanian *et al.*, 2013].

La séparation en deux sous-problèmes nous permet d'appliquer la propriété d'existence d'un filtre désadapté optimal. Mais cela signifie également que la séquence \mathbf{s} et le filtre désadapté \mathbf{q} ne sont jamais déterminés de manière simultanée. Est-il possible de faire autrement ? Le paragraphe suivant s'intéresse au problème d'optimisation jointe.

4.2.2 Optimisation jointe

Ainsi, on introduit dans ce paragraphe deux procédures afin d'optimiser de manière jointe (*i.e.* de manière simultanée) une séquence et son filtre désadapté associé. On commencera par une résolution à l'aide d'un gabarit : on appliquera spontanément un algorithme d'optimisation (à l'instar d'une descente de gradient) sur une fonction qui inclut donc un gabarit, et dont les variables recherchées sont les deux séquences \mathbf{s} et \mathbf{q} . Le gabarit en question permettra entre autres de contrôler le niveau des lobes secondaires et les pertes en gain de traitement. Ensuite, on adaptera la fonction objectif du problème (P_{joint}), afin de pouvoir exploiter la propriété d'existence du filtre désadapté optimal. Le LPG sera pris en compte à l'aide d'une contrainte, comme précédemment. Cette procédure sera désignée par la locution « optimisation par réinjections ».

4.2.2.1 Optimisation jointe avec gabarit

Pourquoi séparer les deux variables ? A deux, c'est toujours mieux, non ? De ce fait, on se propose de résoudre directement le problème d'optimisation jointe, afin d'obtenir \mathbf{s} et \mathbf{q} simultanément. Une méthode similaire a déjà été proposée dans [Xu *et al.*, 2015], où la fonction mesure le niveau du lobe secondaire le plus élevé. Cependant, les contraintes sont gérées à l'aide de coefficients de pondérations, qu'il faut déterminer empiriquement.

Ici, on modélise les objectifs souhaités sur le niveau des lobes secondaires et la perte due à la désadaptation à l'aide d'un gabarit, l'optimisation ayant ainsi pour but de s'approcher au mieux de ce dernier. Comme le sous-entend la Figure 4.14, le gabarit se rapporte à la séquence de corrélation. Le lien avec les lobes secondaires est donc évident ; en ce qui concerne les pertes, il suffit de reprendre sa définition analytique (ci-dessous en échelle linéaire) :

$$\text{LPG} = \frac{|\mathbf{q}^H \mathbf{s}|^2}{(\mathbf{q}^H \mathbf{q})(\mathbf{s}^H \mathbf{s})}. \quad (4.32)$$

Au dénominateur, on retrouve la norme euclidienne de chacune des séquences. En émettant quelques hypothèses...

Algorithme Optimisation jointe avec gabarit avec une descente de gradient	
Données	Initialisation de la séquence \mathbf{s} Initialisation du filtre désadapté \mathbf{q} Définition du gabarit objectif \mathbf{b}
Répéter	1. Descente de gradient <ul style="list-style-type: none"> — Calcul des dérivées partielles $\nabla_{\mathbf{s}}(f)$ et $\nabla_{\mathbf{q}}(f)$ — Détermination du pas $\gamma > 0$ — Mise à jour de \mathbf{s} à l'aide de $\nabla_{\mathbf{s}}(f)$ — Mise à jour de \mathbf{q} à l'aide de $\nabla_{\mathbf{q}}(f)$ — Normalisation de \mathbf{q} — Répéter jusque convergence 2. Calibration du gabarit \mathbf{b} si nécessaire
Jusqu'à	Un critère d'arrêt est atteint.

TABLE 4.2 – Algorithme d'optimisation avec gabarit
Présentée ici avec une descente de gradient.
Les dérivées partielles sont explicitées dans [Tan et al., 2018].

- La séquence \mathbf{s} est soumise à la contrainte de module constant, ce qui permet d'écrire $\mathbf{s}^H \mathbf{s} = 1$.
 - A l'aide d'une normalisation (nécessaire dans l'algorithme du gradient), le filtre désadapté \mathbf{q} est un élément de la sphère Ω^K — et non pas nécessairement du tore! — d'où $\mathbf{q}^H \mathbf{q} = 1$.
- ... on peut calibrer le LPG par le biais du numérateur, l'amplitude du pic principal. Plus précisément, l'écart entre l'amplitude du pic principal et la valeur assignée au gabarit en ce point permet donc de contrôler le LPG.

A partir de toutes ces considérations, on peut enfin écrire le problème d'optimisation qui nous intéresse. Soient \mathbf{s} une séquence de longueur N et \mathbf{q} son filtre désadapté associé, de longueur K . Comme la séquence de corrélation AXCF(\mathbf{s}, \mathbf{q}) est de longueur $N + K - 1$, le gabarit \mathbf{b} doit être de même longueur. Plus précisément, il appartient à l'ensemble \mathbb{R}_+^{N+K-1} . Afin de rechercher le couple (\mathbf{s}, \mathbf{q}) s'approchant au mieux du gabarit, on pose par exemple :

$$\left\{ \begin{array}{l} \min_{\mathbf{s}, \mathbf{q}} \quad f(\mathbf{s}, \mathbf{q}) := \sum_{k=1}^{N+K-1} \left(\left| \frac{(\text{AXCF}(\mathbf{s}, \mathbf{q}))_k}{b_k} \right|^{2p} - 1 \right)^2, \\ \text{t.q.} \quad \mathbf{s} \in \mathbf{T}^N, \mathbf{q} \in \Omega^K. \end{array} \right. \quad (4.33)$$

Quid du niveau des lobes secondaires du gabarit? Il est évident qu'on les aimerait les plus bas possibles. Pourtant, en pratique, la convergence des algorithmes d'optimisation en pâtit. Il vaut mieux ainsi choisir un gabarit objectif atteignable en pratique, puis le réajuster si nécessaire (par exemple, une réduction des lobes de 1 dB).

L'ensemble de la procédure est présentée à la Table 4.2.

Remarque 4.11. (*Calibration*) La calibration employée dans la procédure à la Table 4.2 est relativement naïve. Dans l'optique d'avoir le meilleur résultat possible...

- ... les « lobes secondaires » du gabarit sont réduits itérativement (par exemple, de 1 dB), tant que ce dernier est atteint. Le couple de séquences (\mathbf{s}, \mathbf{q}) est conservé.
- À un moment, le gabarit ne pourra nécessairement plus être atteint. En pratique, cela s'observe par une dégradation du LPG (le LPG est plus mauvais que celui souhaité) : la contrainte associée est éventuellement ignorée, et ce afin de respecter les lobes secondaires du gabarit. L'étape de calibration s'arrête, et on se contente du dernier résultat retenu.

Algorithme Optimisation jointe par réinjections avec une descente de gradient

Données	Initialisation de la séquence \mathbf{s}
Répéter	1. Descente de gradient <ul style="list-style-type: none"> — Calcul du gradient* $\nabla_{\alpha} f$ — Détermination du pas $\gamma > 0$ — Mise à jour de $\mathbf{s} : s_n = s_n \exp(-j\mu(\nabla_{\alpha} f)_n)$, $n \in \llbracket 1, N \rrbracket$
Jusqu'à	Un critère d'arrêt est atteint.

TABLE 4.3 – Optimisation jointe par réinjections avec une descente de gradient

* : Dans le cas où l'expression du filtre désadapté optimal n'est pas connue, le calcul du gradient peut s'effectuer numériquement, à l'aide d'un taux d'accroissement par exemple. Ce qui implique le calcul d'un filtre \mathbf{q} pour chaque terme $(\nabla_{\alpha} f)_n$, très dispendieux...

— Évidemment, si aucune solution n'a été définie, on « relève » les lobes secondaires du gabarit...

4.2.2.2 Optimisation jointe par réinjections

La procédure précédente optimise de manière jointe une séquence et son filtre désadapté associé. Cependant, elle n'exploite pas la propriété d'existence d'un filtre désadapté optimal. On se propose donc dans ce paragraphe de corriger cela... On rappelle que le problème d'optimisation jointe s'écrit — repris de l'équation (4.29) — :

$$(P_p^{(1)}) \begin{cases} \min_{\mathbf{s}, \mathbf{q}} & f_p(\mathbf{s}, \mathbf{q}) \\ \text{t.q.} & \mathbf{s}^H \mathbf{q} = \mathbf{s}^H \mathbf{s}, \\ & \mathbf{s} \in \mathbf{T}^N, \mathbf{q} \in \mathbb{C}^K. \end{cases} \quad (4.34)$$

Notons deux petites différences par rapport au problème initial (P_p) (4.14). La première se situe au niveau des variables d'optimisation, avec l'ajout de la séquence \mathbf{s} . La deuxième concerne sa résolution : du fait que \mathbf{s} appartient à un tore, un ensemble non convexe, il n'est pas aisé de résoudre de manière globale le problème ci-dessus. Un peu comme dans le Chapitre 2. Il existe évidemment encore moins une expression analytique, à notre connaissance... En séparant le problème d'optimisation en deux, on obtient :

$$(P_p^{(2)}) \begin{cases} \min_{\mathbf{s}} \min_{\mathbf{q}} & f_p(\mathbf{s}, \mathbf{q}) \\ \text{t.q.} & \mathbf{s}^H \mathbf{q} = \mathbf{s}^H \mathbf{s}, \\ & \mathbf{s} \in \mathbf{T}^N, \mathbf{q} \in \mathbb{C}^K. \end{cases} \quad (4.35)$$

Sous cette expression, on peut en quelque sorte identifier le problème (P_p) . Comme mentionné précédemment, ce dernier admet une solution optimale, que l'on peut déterminer numériquement. Soit $\mathbf{q}_p(\mathbf{s})$ cette solution. En l'insérant dans la fonction objectif, on obtient un problème d'optimisation mono-séquence :

$$(P_p^{(3)}) \begin{cases} \min_{\mathbf{s}} & f(\mathbf{s}) := f_p(\mathbf{s}, \mathbf{q}_p(\mathbf{s})) \\ \text{t.q.} & \mathbf{s} \in \mathbf{T}^N. \end{cases} \quad (4.36)$$

A partir de la fonction coût du problème $(P_p^{(3)})$, la définition du filtre désadapté se fait de façon intrinsèque. Ce dernier est d'ailleurs optimal pour le critère de f_p . En outre, le problème d'optimisation obtenu se révèle être similaire à ceux rencontrés dans le Chapitre 2.

Le Tableau 4.3 explicite une descente de gradient associée à une optimisation jointe par réinjections. α dénote le vecteur de phase associé à la séquence \mathbf{s} .

Remarque 4.12. (*Élucubrations*)

1. *Il existe des différences avec une optimisation cyclique : la plus apparente se situe au niveau de la génération des candidats. En effet, considérons une itération de l'algorithme cyclique où le filtre \mathbf{q} a été déterminé. Ce dernier est optimal pour la dernière séquence \mathbf{s} en date, mais pas nécessairement pour les différents candidats à l'itération en cours ! Ce qui n'est pas le cas pour le problème d'optimisation indirecte. Néanmoins, son étape de génération est donc beaucoup plus coûteuse.*
2. *En outre, en insérant la contrainte de perte, il n'existe aucune solution analytique connue à notre connaissance, même pour l'ISL, pour l'optimisation du filtre \mathbf{q} connaissant la séquence \mathbf{s} . Pourtant, les résultats qui suivent se baseront malgré tout sur le critère de l'ISL, car ce dernier nous a offert les performances les plus intéressantes. En appliquant le même procédé pour un p relativement grand, il faut avouer que nous avons décroché des résultats clairement moins bons. Et nous ignorons la raison...*

4.2.3 Résultats

Cette section présente quelques résultats obtenus à l'aide des algorithmes précédemment présentés, à savoir un algorithme cyclique, ainsi que les procédures jointes par gabarit et réinjections que nous avons proposées. Dans tous les cas, l'optimisation sera mise en œuvre à l'aide d'une descente de gradient. On comparera, comme pour la Section 4.1.4, l'influence d'une initialisation optimisée ou non pour le filtrage adapté.

D'un point de vue pratique, les séquences tirées seront de longueur $N = 63$, et les filtres désadaptés choisis trois fois plus longs. La perte autorisée sera de 1 dB. Par ailleurs, pour déterminer le filtre désadapté optimal au sens du PSLR, la bibliothèque CVX a été utilisée de nouveau.

Initialisation aléatoire La Figure 4.15 illustre l'intérêt d'une optimisation jointe. Elle compare ainsi les résultats obtenus après une optimisation cyclique et une optimisation jointe (ici, une optimisation avec un gabarit plat), pour une même initialisation aléatoire. Il y a peu de choses à dire, tant la figure est explicite : on remarque une différence de l'ordre de 10 dB, à l'avantage de l'optimisation conjointe. Pour un algorithme cyclique, la question de l'initialisation se pose. En effet, après les deux premières étapes (optimisation d'une séquence puis du filtre), les séquences changent peu. Est-ce dû au choix de la méthode d'optimisation ?

Pour plus de clarté, on compare dans une autre figure (Figure 4.16) les deux méthodologies d'optimisation jointe pour la même initialisation. Pour rappel, l'optimisation avec gabarit a pour objectif une minimisation du lobe secondaire le plus élevé (d'où le gabarit plat). Pour l'optimisation avec réinjections, on s'est intéressé à l'ISL, comme le mentionne la remarque précédente. Les normes diffèrent, et cela se ressent *a fortiori* sur le profil des résultats. En effet, on note des lobes secondaires légèrement plus bas autour du lobe principal, avec une légère remontée sur les bords (autour de -40 dB).

Influence de l'initialisation La descente de gradient est souvent caractérisée à juste titre comme un algorithme d'optimisation à convergence locale. Le choix de l'initialisation a sans doute une importance manifeste, mais cela dépend du problème en question. Dans le Chapitre 2, le choix de la séquence initiale n'a pas semblé avoir une grande influence sur le niveau des lobes secondaires de la séquence obtenue après une optimisation dans le sens du filtre adapté. Cette observation est inadéquate et même contraire à la Section 4.1.4.3, mais cette dernière portait simplement sur l'optimisation d'un filtre désadapté, pour une séquence donnée.

Pour déterminer la constatation la plus appropriée dans le cas d'une optimisation jointe, on se propose d'effectuer la même procédure que celle du paragraphe sus-cité. Pour rappel, cela consiste en une optimisation préalable de l'initialisation dans le sens du filtre adapté. Les résultats sont présentés à la Figure 4.17.

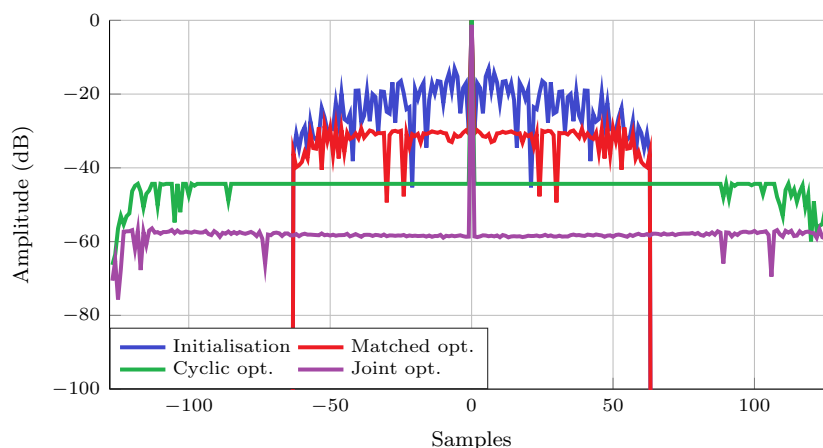


FIGURE 4.15 – Optimisation cyclique, ou optimisation jointe ?
Matched opt. : optimisation dans le sens du filtre adapté seulement.
Initialisation aléatoire de longueur $N = 63$

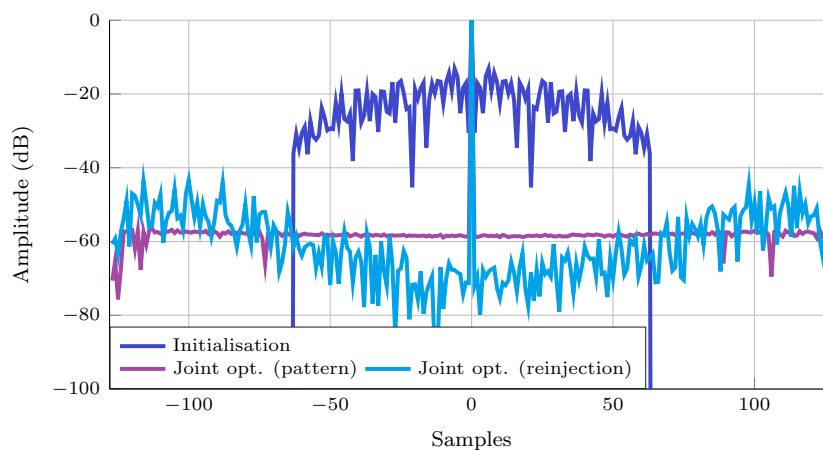


FIGURE 4.16 – Comparatif des méthodes d'optimisation jointe
Initialisation identique à celle de la figure précédente.

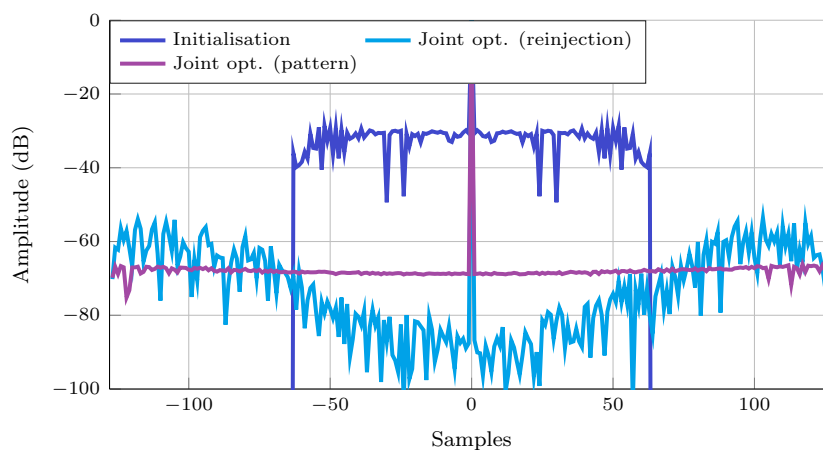


FIGURE 4.17 – Comparatif des méthodes d'optimisation jointe, initialisation optimisée
Initialisation optimisée de longueur $N = 63$.
Veuillez noter le gain de quelques dB par rapport à la figure ci-dessus...

On amorce donc une modification de l'initialisation aléatoire de la Figure 4.16, puis on évalue les deux méthodologies d'optimisation jointe. D'un point de vue qualitatif, les deux figures sont similaires ; cela n'est pas une surprise, puisque les paramètres sélectionnés (gabarit, pertes, norme ℓ_p) le sont également. Cependant, d'un point de vue quantitatif, on note quelques améliorations. Une optimisation avec gabarit offre toujours une distribution uniforme, mais avec un bénéfice de 10 dB, passant de -56 dB à -66 dB. De l'autre côté, le fruit d'une optimisation par réinjections dans le sens de l'ISL est vraiment à croquer, avec des lobes secondaires allant jusqu'à -90 dB autour du pic principal. Ceux sur les bords restent assurément acceptables (-53 dB). A notre connaissance, de tels niveaux n'ont jamais été atteints dans la littérature. Ce profil particulier peut réellement être intéressant dans des applications radars, car cela permettrait d'améliorer la détection des petites cibles proches qui peuvent être noyées dans un fort *clutter*.

4.2.4 Conclusions et perspectives

Terminons cette section avec quelques mots récapitulatifs. Nous avons observé dans la précédente section l'apport du filtrage désadapté par rapport à la compression d'impulsion, mais pour une séquence donnée. Néanmoins, lorsqu'on s'intéresse à la génération de formes d'onde, il est d'intérêt de prendre en compte dans l'optimisation la séquence d'émission \mathbf{s} . Nous nous sommes intéressés à une optimisation jointe, qui consiste à optimiser *simultanément* la séquence \mathbf{s} et son filtre désadapté \mathbf{g} , et avons proposé deux procédés. L'un est fondé sur l'emploi d'un gabarit, permettant de se donner des objectifs en terme de lobes secondaires et de contrôler les pertes dues à la désadaptation. L'autre, fondé sur la réinjection de l'expression optimale du filtre désadapté, optimise conjointement les deux séquences tout en conservant la propriété d'optimalité du filtre désadapté. Les simulations effectuées nous persuadent de l'utilité des deux méthodes.

- En effet, la première permet — après tout, nous l'avons plus ou moins imposé — d'avoir un niveau uniforme de lobes secondaires. Ce dernier est généralement plus intéressant que celui obtenu par des algorithmes cycliques.
- Tandis que la seconde offre un profil très intrigant, avec des lobes très atténués autour du lobe principal, tout en étant très corrects sur les bords, malgré la courte longueur des séquences. Cela est certes lié à l'ISL...

Tout cela est encourageant, mais il va falloir procéder à quelques études complémentaires sur le sujet. Premièrement, nous avons choisi par hypothèse une descente de gradient pour sa rapidité et sa facilité de mise en œuvre, mais aussi parce qu'elle était relativement convaincante lors de l'étude des séquences de corrélation. Il faudrait, pour être rigoureux, accomplir quelque chose de similaire à ce chapitre, en essayant par exemple diverses méthodes, en variant les paramètres, ou en déterminant des initialisations pertinentes. Deuxièmement, une extension pour des ensembles de séquences et leurs filtres désadaptés associés serait appréciable, à l'instar de la section précédente. Troisièmement, nous ne savons pas pourquoi une optimisation avec réinjections, appliquée avec le critère du PSLR, donne de moins bons résultats qu'avec une méthode à gabarits. Enfin, rappelons la non-convexité du problème, en raison de la séquence \mathbf{s} . Cela implique que nous n'avons pas l'assurance d'avoir atteint l'optimum global. Les résultats présentés sont atteignables en pratique, mais cela signifie aussi que l'on peut potentiellement faire mieux. Est-ce possible de caractériser ce « mieux » ? A l'aide d'une borne peut-être ?

Les travaux de cette section ont été publiés dans [Tan et al., 2018].

4.3 Bornes : parce qu'on n'ira pas loin

Afin d'évaluer les performances d'un ensemble de séquences, le chapitre précédent a introduit quelques bornes portant sur le niveau de leurs lobes secondaires : la borne de Welch, mais aussi celle de Levenshtein qui, pour rappel, a été au passage améliorée. Ici, nous avons

commencé à établir un parallèle en suggérant à la Section 4.1.3 une écriture d'une optimisation pour un couple d'ensembles séquences et filtres désadaptés associés. Poursuivons en essayant de définir une borne pour les performances de ces dernières.

Mais cette fois, le constat est simple. Il n'existe à notre connaissance aucune borne sur ce sujet. Ainsi, on se propose dans cette section d'en développer une nouvelle. Sa démonstration, et *a fortiori* son expression, sont inspirées de celles de la borne de Levenshtein (*cf.* la Section 3.6). Elles reposent donc sur l'encadrement de la norme d'une matrice contenant toutes les valeurs des corrélations. Tout cela fera l'objet d'un premier paragraphe. Le second étudiera son comportement ; une comparaison sera d'ailleurs effectuée avec la borne obtenue dans le cas adapté. Les démonstrations sont dans l'Annexe B.

4.3.1 Une borne sur le filtrage désadapté

4.3.1.1 Définitions et notations

Commençons par énumérer quelques définitions. Afin d'être cohérent avec la littérature sur le sujet, certaines notations seront adaptées ici : il faudra donc veiller à ne pas confondre certains paramètres... Soit $\{\mathbf{s}^m\}_{m \in \llbracket 1, M \rrbracket}$ un ensemble composé de M séquences de même longueur N_s , sous la contrainte de module constant (l'énergie de chaque séquence est unitaire). Soit $\{\mathbf{q}^m\}_{m \in \llbracket 1, M \rrbracket}$ l'ensemble de leurs filtres désadaptés associés. Sans perte de généralité, on supposera que leur longueur — similaire pour chacune — est égale à $N_q := N_s + 2N_l$, $N_l \in \mathbb{N}$. Ci-dessous une liste de (re-)définitions.

1. Soit $N_\theta := N_s + N_q - 1$. Pour une séquence \mathbf{s}^m et un filtre \mathbf{q}^l , on note par $\boldsymbol{\theta}^{m,l} \in \mathbb{C}^{N_\theta}$ leur séquence de corrélation :

$$\boldsymbol{\theta}^{m,l} = \text{AXCF}(\mathbf{s}^m, \mathbf{q}^l). \quad (4.37)$$

2. Comme sous-entendu dans l'introduction du paragraphe 4.1.3, on considère l'existence d'un lobe principal seulement pour une corrélation entre une séquence \mathbf{s}^m et son filtre désadapté associé \mathbf{q}^m . Soit $N_k := N_s + N_l - 1$ l'indice dans la séquence $\boldsymbol{\theta}^{m,m}$ définissant ce lobe. Séparer le module de la phase du lobe principal s'écrit, pour tout $m \in \llbracket 1, M \rrbracket$:

$$\theta_{N_k}^{m,m} = |\theta_{N_k}^{m,m}| e^{j\varphi_m}, \quad (4.38)$$

où $\varphi_m \in [-\pi, \pi[$ est un élément de phase. Le produit d'une séquence par une phase ne changeant ni les niveaux de corrélation ni la perte en gain de traitement, on peut toujours trouver un autre ensemble de filtres désadaptés $\{\mathbf{q}'^m\}_{m \in \llbracket 1, M \rrbracket}$ tel que :

$$\begin{cases} \{\mathbf{q}'^m = \mathbf{q}^m e^{-j\varphi_m}, m \in \llbracket 1, M \rrbracket\}, \\ \theta_{N_k}'^{m,m} = |\theta_{N_k}^{m,m}| \in \mathbb{R}^+. \end{cases} \quad (4.39)$$

Il est assez clair que les filtres \mathbf{q}^m et \mathbf{q}'^m permettent d'obtenir les mêmes niveaux de lobes secondaires. Cette propriété sera nécessaire dans une prochaine démonstration.

3. On définit ici le PSLR comme le lobe secondaire le plus élevé pour l'ensemble des corrélations :

$$\begin{aligned} \theta_{\max}^2 &= \max \left\{ \max_{k, m \neq l} \left(|\theta_k^{m,l}|^2 \right), \max_{k \neq N_k, m} \left(|\theta_k^{m,m}|^2 \right) \right\}, \\ \text{PSLR} &= \frac{\theta_{\max}^2}{\max_m \left(|\theta_{N_k}^{m,m}|^2 \right)}. \end{aligned} \quad (4.40)$$

4. S'ensuit le LPG (sans oublier la contrainte de module constant sur \mathbf{s}^m) :

$$\text{LPG}_m = \frac{|\theta_{N_k}^{m,m}|^2}{E_{\mathbf{q}^m}^2}, \quad (4.41)$$

où $E_{q^m}^2 := (\mathbf{q}^m)^H \mathbf{q}^m$ est l'énergie de \mathbf{q}^m . Si le niveau de chaque lobe principal est identique, remarquez que le LPG est complètement déterminé par cette énergie.

4.3.1.2 Une nouvelle borne

On peut maintenant s'attaquer à la nouvelle borne sur le niveau des lobes secondaires d'un ensemble de séquences et de leurs filtres désadaptés associés. La démonstration est très similaire à celle que nous avons proposée dans le chapitre précédent pour la borne de Levenshtein, dans le cas du filtrage adapté.

Dans cette démonstration, on considère seulement les paires d'ensembles $\{\mathbf{s}^m\}_{m \in \llbracket 1, M \rrbracket}$ et $\{\mathbf{q}^m\}_{m \in \llbracket 1, M \rrbracket}$ telles que les termes de corrélation $\theta_{N_k}^{m,m}$ soient réels. Cette hypothèse semble restrictive de prime abord mais, comme mentionné à la page précédente, n'importe quelle paire (*i.e.* en particulier celles dont l'un des lobes principaux est complexe) peut se résoudre en ce cas, par de simples déphasages. Cela est résumé en la proposition suivante :

Proposition 4.1. (*Extension de la borne pour n'importe quel filtre complexe*) *Considérons les ensembles de filtres désadaptés $\{\mathbf{q}^m\}_m$ tels que chaque lobe principal $\theta_{N_k}^{m,m}$ soit un réel positif, pour tout $m \in \llbracket 1, M \rrbracket$. Supposons qu'il en existe une borne inférieure B sur le niveau des lobes secondaires le plus élevé telle que $\theta_{\max}^2 \geq B$ pour tout ensemble $\{\mathbf{q}^m\}_m$. Alors la borne B s'applique également pour tout ensemble $\{\mathbf{q}^m\}_m$ tel que $\theta_{N_k}^{m,m}$ soit complexe.*

Démonstration. Supposons qu'il existe un ensemble de filtres désadaptés $\{\mathbf{q}^m\}_m$ tel que le lobe secondaire le plus élevé θ_{\max}^2 vérifie $\theta_{\max}^2 < B$. Pour chaque filtre \mathbf{q}^m , $m \in \llbracket 1, M \rrbracket$, on décompose le lobe principal induit selon le module et la phase comme suit :

$$\theta_{N_k}^{m,m} := \rho^m e^{j\varphi_m} \quad \text{avec } \rho^m \in \mathbb{R}^+. \quad (4.42)$$

Posons un autre ensemble de filtres désadaptés $\{\mathbf{q}'^m\}_m$ tel que :

$$\begin{aligned} \mathbf{q}'^m &:= \mathbf{q}^m e^{-j\varphi_m}, \\ \implies \theta_{N_k}^{m,m} &= \rho^m. \end{aligned} \quad (4.43)$$

Il est évident que chaque filtre désadapté \mathbf{q}'^m offre les mêmes lobes secondaires que \mathbf{q}^m (à une phase près). Ainsi, le lobe secondaire le plus élevé vérifie aussi $\theta_{\max}^2 < B$. Or, $\theta_{N_k}^{m,m}$ est un réel positif; il satisfait donc également $\theta_{\max}^2 \geq B$ par hypothèse, ce qui contredit l'assertion de départ. Il ne peut donc exister aucun ensemble de séquences $\{\mathbf{q}^m\}_m$ tel que $\theta_{\max}^2 < B$. \square

Soient \mathbf{S}^m et \mathbf{Q}^l deux matrices carrées, pour tout $(m, l) \in \llbracket 1, M \rrbracket^2$. Chaque matrice contient des versions décalées de \mathbf{s}^m et \mathbf{q}^l . Cependant, leur définition dépend de la périodicité du signal. Dans le cas aperiodique, \mathbf{S}^m et \mathbf{Q}^l sont de dimension N_θ :

$$\mathbf{S}^m = \begin{bmatrix} s_1^m & \cdots & \cdots & s_{N_s}^m & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & s_1^m & & s_{N_s}^m & 0 & & & & & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & & & & & & & & & \vdots \\ \vdots & & & & & & & & & & & & \vdots \\ \vdots & & & & & & & & & & & & \vdots \\ \vdots & & & & & & & & & & & & \vdots \\ \vdots & & & & & & & & & & & & \vdots \\ \vdots & & & & & & & & & & & & \vdots \\ \vdots & & & & & & & & & & & & \vdots \\ \vdots & & & & & & & & & & & & \vdots \\ 0 & & & & & & & & & & & & s_{N_s}^m \\ s_{N_s}^m & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & 0 & & & & & & & & & & & \vdots \\ s_2^m & \cdots & s_{N_s}^m & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & s_1^m \end{bmatrix}, \quad (4.44)$$

$$\mathbf{Q}^l = \begin{bmatrix} q_{N_l+1}^l & \dots & \dots & \dots & \dots & q_{N_q}^l & 0 & \dots & 0 & q_1^l & \dots & q_{N_l}^l \\ q_{N_l}^l & \dots & \dots & \dots & \dots & q_{N_q}^l & 0 & \dots & \dots & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & q_1^l \\ q_1^l & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & q_1^l & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \vdots \\ \vdots & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & q_{N_q}^l \\ q_{N_q}^l & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & q_{N_q-1}^l \\ q_{N_q-1}^l & q_{N_q}^l & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \vdots \\ q_{N_l+2}^l & \dots & \dots & \dots & \dots & q_{N_q}^l & 0 & \dots & 0 & q_1^l & \dots & q_{N_l+1}^l \end{bmatrix}. \quad (4.45)$$

Constatons que les deux précédentes matrices sont circulantes, puisqu'elles représentent les séquences \mathbf{s}^m et \mathbf{q}^l décalées dans le temps. La définition d'une matrice circulante est donnée à la Remarque 3.8, et on a :

$$\begin{cases} \mathbf{S}^m = \text{circ}([\mathbf{s}^m, \mathbf{0}_{N_q-1,1}]) \\ \quad = \text{circ}(\mathbf{S}_1^m), \\ \mathbf{Q}^l = \text{circ}(\mathbf{Q}_1^l). \end{cases} \quad (\text{la première ligne de la matrice } \mathbf{S}^m) \quad (4.46)$$

Dans le cas périodique, ces matrices sont de taille N_s (cf. la Remarque 4.2) :

$$\begin{cases} \mathbf{S}^m = \text{circ}(\mathbf{s}^m), \\ \mathbf{Q}^l = \text{circ}(\mathbf{q}^l). \end{cases} \quad (4.47)$$

Les matrices \mathbf{S}^m et \mathbf{Q}^l permettent de générer l'ensemble des valeurs de corrélation. En effet, si on pose les matrices de concaténation \mathbf{S} et \mathbf{Q} de taille $MN_\theta \times N_\theta$ suivantes :

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}^1 \\ \mathbf{S}^2 \\ \vdots \\ \mathbf{S}^M \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \mathbf{Q}^1 \\ \mathbf{Q}^2 \\ \vdots \\ \mathbf{Q}^M \end{bmatrix}, \quad (4.48)$$

toutes ces valeurs de corrélation se retrouvent dans une matrice carrée \mathbf{R} de taille MN_θ définie par :

$$\mathbf{R} = \mathbf{S}\mathbf{Q}^H = \begin{bmatrix} \mathbf{S}^1(\mathbf{Q}^1)^H & \dots & \mathbf{S}^1(\mathbf{Q}^M)^H \\ \vdots & \ddots & \vdots \\ \mathbf{S}^M(\mathbf{Q}^1)^H & \dots & \mathbf{S}^M(\mathbf{Q}^M)^H \end{bmatrix}, \quad (4.49)$$

de sorte que :

$$\mathbf{S}^m(\mathbf{Q}^l)^H = \text{circ}\left(\left[\theta_{N_k}^{m,l}, \theta_{N_k-1}^{m,l}, \dots, \theta_1^{m,l}, \theta_{N_\theta}^{m,l}, \dots, \theta_{N_k+1}^{m,l}\right]\right).$$

En d'autres termes, le bloc à la position (m, l) contient l'ensemble des corrélations entre la m -ème séquence et le l -ème filtre.

A l'instar de la borne de Levenshtein [Levenshtein, 1998], il est possible de considérer une pondération de ces termes. Soit $\mathbf{w} \in \mathbb{R}^{N_\theta}$ un vecteur vérifiant la *condition de pondération* (3.47), que l'on rappelle ci-dessous :

$$\sum_{k=1}^{N_\theta} w_k = 1, \quad \text{avec} \quad w_k \geq 0 \text{ si } k \in \llbracket 1, N_\theta \rrbracket, \quad (4.50)$$

$$w_k = 0 \text{ sinon.}$$

En particulier, il est souvent supposé que la pondération \mathbf{w} est non nulle pour les K premiers éléments uniquement, ce qui s'écrit :

$$\sum_{k=1}^K w_k = 1 \text{ tel que } \begin{cases} w_k \geq 0 & \text{si } k \in \llbracket 1, K \rrbracket, \\ w_k = 0 & \text{sinon.} \end{cases} \quad (4.51)$$

Multiplier chaque ligne de \mathbf{S}^m et de \mathbf{Q}^l par un coefficient nous donne :

$$\begin{cases} \tilde{\mathbf{S}}_i^m := \mathbf{S}_i^m \sqrt{w_i}, \\ \tilde{\mathbf{Q}}_j^l := \mathbf{Q}_j^l \sqrt{w_j}, \end{cases} \quad \text{pour tout } (i, j) \in \llbracket 1, N_\theta \rrbracket^2, (l, m) \in \llbracket 1, M \rrbracket^2, \quad (4.52)$$

ce qui nous permet d'écrire une version pondérée des matrices précédentes :

$$\tilde{\mathbf{R}} = \tilde{\mathbf{S}} \tilde{\mathbf{Q}}^H \text{ avec } \tilde{\mathbf{S}} = \begin{bmatrix} \tilde{\mathbf{S}}^1 \\ \vdots \\ \tilde{\mathbf{S}}^M \end{bmatrix} \text{ et } \tilde{\mathbf{Q}} = \begin{bmatrix} \tilde{\mathbf{Q}}^1 \\ \vdots \\ \tilde{\mathbf{Q}}^M \end{bmatrix}. \quad (4.53)$$

Si on se réfère à la démonstration de la borne de Levenshtein, il convient maintenant de déterminer un encadrement de la norme de la matrice $\tilde{\mathbf{R}}$. Cela nous permettra de déduire simplement une borne sur les lobes secondaires.

Lemme 4.1. (*Borne supérieure*) *La norme de Frobenius de la matrice $\tilde{\mathbf{R}}$ admet la borne supérieure suivante :*

$$\|\tilde{\mathbf{R}}\|_F^2 \leq \sum_{m=1}^M (|\theta_{N_k}^{m,m}|^2 - \theta_{\max}^2) \mathbf{w}^T \mathbf{w} + M^2 \theta_{\max}^2. \quad (4.54)$$

Lemme 4.2. (*Borne inférieure*) *... et la borne inférieure qui suit :*

$$\|\tilde{\mathbf{R}}\|_F^2 \geq \frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{l_{N_s, N_q, K}}, \quad (4.55)$$

où $l_{N_s, N_q, K} = \min(N_s + N_q - 1, N_s + K - 1, MK)$.

La borne supérieure s'obtient de la même façon que le Lemme 1 de [Levenshtein, 1999]. La démonstration est néanmoins proposée à l'Annexe B.1, tandis que celle de la borne inférieure est dans l'Annexe B.2. En rassemblant ces deux résultats, on peut aisément déduire une borne sur θ_{\max}^2 (le calcul se situe aussi en annexe).

Théorème 4.1. *Soit \mathbf{w} un vecteur vérifiant la condition de pondération (3.47). Pour tout ensemble de M séquences unimodulaires de même longueur N_s , pour tout ensemble de filtres désadaptés associés de même longueur N_q , une borne inférieure sur le lobe secondaire le plus élevé s'écrit :*

$$\theta_{\max}^2 \geq \frac{1}{M^2 - M \mathbf{w}^T \mathbf{w}} \left[\frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{l_{N_s, N_q, K}} - \mathbf{w}^T \mathbf{w} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \right], \quad (4.56)$$

pour $K \in \llbracket 1, N_\theta \rrbracket$, et avec $l_{N_s, N_q, K} = \min(N_s + N_q - 1, N_s + K - 1, MK)$.

Le théorème ci-dessus est présenté pour un K fixé. Pour être complet, on peut obtenir une borne plus serrée en considérant toutes les valeurs que ce paramètre peut prendre :

$$\theta_{\max}^2 \geq \max_{K \in \llbracket 1, N_\theta \rrbracket} \frac{1}{M^2 - M \mathbf{w}^T \mathbf{w}} \left[\frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{l_{N_s, N_q, K}} - \mathbf{w}^T \mathbf{w} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \right]. \quad (4.57)$$

4.3.2 Quelques commentaires

La borne suggérée dans le paragraphe précédent dépend de plusieurs paramètres : ceux qui sont intrinsèques à l'ensemble des séquences, à savoir leur nombre M ou leur longueur N_s , mais aussi ceux introduits par la désadaptation, comme le LPG (indirectement contenu dans $\theta_{N_k}^{m,m}$). Et n'oublions pas le vecteur de pondération \mathbf{w} . Ainsi, on se propose dans ce paragraphe de discuter de l'influence de chacun de ces paramètres, en commençant par la fin. On conclura donc par une comparaison avec les bornes du cas adapté.

4.3.2.1 Pondération optimale

L'expression du Théorème 4.1 peut se réécrire de la façon suivante (cf. Annexe B.3) :

$$\theta_{\max}^2 \geq \left(\frac{1}{M} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \right) \left[1 - \frac{M}{M - \mathbf{w}^T \mathbf{w}} \frac{l_{N_s, N_q, K} - \frac{|\sum_{m=1}^M \theta_{N_k}^{m,m}|^2}{M \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2}}{l_{N_s, N_q, K}} \right]. \quad (4.58)$$

Sous cette écriture, il est plus simple de reconnaître la relation inverse entre la pondération et la borne, pour un K fixé : plus petit est le terme $\mathbf{w}^T \mathbf{w}$, plus serrée est la borne. Cette assertion n'était pas nécessairement vraie dans le cas de la borne du filtre adapté, car le terme $l_{s,t,N}$ (homologue à $l_{N_s, N_q, K}$) était lié à la pondération. A partir de l'inégalité de Cauchy-Schwarz, on a :

$$\mathbf{w}^T \mathbf{w} = \sum_{k=1}^K w_k^2 \geq \frac{\left(\sum_{k=1}^K w_k \right)^2}{K} = \frac{1}{K}. \quad (4.59)$$

De cela, on peut dire que la pondération optimale est atteinte lorsque l'inégalité de Cauchy-Schwarz est égalité, c'est-à-dire lorsque $\mathbf{w}^T \mathbf{w} = 1/K$. En d'autres termes, \mathbf{w} est la pondération constante, rappelée ci-dessous :

$$w_k = \begin{cases} 1/K & \text{si } k \leq K, \\ 0 & \text{sinon.} \end{cases} \quad (4.60)$$

De ce fait, en appliquant ce vecteur spécifique sous les mêmes hypothèses, on obtient une borne plus serrée, décrite par le Théorème 4.1 :

Lemme 4.3. (Borne sur les lobes secondaires, pondération constante)

$$\theta_{\max}^2 \geq \left(\frac{1}{M} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \right) \left[1 - \frac{MK}{MK - 1} \frac{l_{N_s, N_q, K} - \frac{|\sum_{m=1}^M \theta_{N_k}^{m,m}|^2}{M \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2}}{l_{N_s, N_q, K}} \right]. \quad (4.61)$$

Le résultat obtenu est assez fort. Il signifie qu'il existe une pondération optimale pour cette borne. Cela diffère donc de la borne de Levenshtein (dans le cas adapté) où la pondération optimale reste à déterminer, et fait toujours l'objet de travaux dans la littérature...

Remarquons que l'expression ci-dessus ne dépend que des éléments $\theta_{N_k}^{m,m}$. Ceux-ci sont proportionnels à l'énergie des différents filtres désadaptés et à la perte en gain de traitement, — cf. l'égalité (4.41) —, qui ne sont contraintes par aucune hypothèse. Dans la suite, on peut donc poser sans perte de généralités :

$$\theta_{N_k}^{m,m} = 1 \quad \forall m \in \llbracket 1, M \rrbracket. \quad (4.62)$$

La borne devient alors :

$$\theta_{\max}^2 \geq \frac{MK - l_{N_s, N_q, K}}{(MK - 1)l_{N_s, N_q, K}}. \quad (4.63)$$

4.3.2.2 Comparaisons sur les bornes

Dans ce paragraphe, on se propose de comparer le comportement de la présente borne avec deux bornes pour le filtrage adapté : celle de Welch [Welch, 1974] et de Levenshtein [Arlery *et al.*, 2019, Levenshtein, 1998].

Inspirée de cette dernière, la borne sur le filtrage désadapté — avec une pondération constante, Lemme 4.3 — est également définie selon un paramètre K , le nombre de valeurs de corrélations considérées. La Figure 4.18 illustre l'influence de celui-ci par un simple exemple : un ensemble de $M = 4$ séquences de longueur $N_s = 1024$, avec une variation de la longueur des filtres désadaptés associés ($N_q = N_s$ pour la figure de gauche, $N_q = 3N_s$ pour celle de droite). Rappelons que la borne inférieure sur les lobes secondaires la plus serrée est ici déterminée par le maximum de chaque courbe. Dans le premier cas, la borne optimale est atteinte en prenant en considération l'ensemble des valeurs, *i.e.* $K = N_\theta$, contrairement au cas adapté. On y note d'ailleurs une différence de 0.5 dB. Cette première assertion est en revanche erronée dans le second cas. Néanmoins, on retrouve le fait qu'un filtre plus long permet d'offrir hypothétiquement de meilleures performances, avec 1.8 dB d'écart entre les deux bornes désadaptées. Ces observations sont-elles toujours vérifiées ? Inspectons un exemple avec la même longueur de séquence, mais un ensemble plus fourni.

En effet, on considère à la Figure 4.19 des ensembles composés de $M = 10$ séquences. Mais, cette fois, les deux courbes (du cas désadapté) présentent une borne optimale identique, obtenue à l'aide d'une même valeur de K , malgré différentes longueurs de filtres désadaptés. Un signe d'indépendance envers N_q (ne tenant seulement que sur la borne) ? Après tout, dans cet exemple, un filtre plus long n'apporterait pas nécessairement une amélioration. Ce phénomène, que nous avons déjà croisé dans le cas périodique (la Remarque 4.2), peut s'expliquer ici à partir de l'expression du terme $l_{N_s, N_q, K}$, que l'on rappelle ci-dessous :

$$l_{N_s, N_q, K} = \min(N_s + N_q - 1, N_s + K - 1, MK). \quad (4.64)$$

En effet, ce paramètre est le seul qui *s'exprime* en fonction de la longueur N_q dans l'expression de la borne. Mais, selon les situations, il n'en dépend pas forcément, d'où les bornes identiques. Au passage, on note un gain — si l'on peut parler de « gain » — d'environ 1.3 dB envers le filtre adapté.

En parlant d'amélioration, autorisons nous quelques mots sur le comportement des bornes en fonction du nombre et de la longueur des séquences de l'ensemble. Ainsi, la Figure 4.20 représente la valeur optimale des bornes (adapté et désadapté) selon cette cardinalité. Leur allure est clairement prévisible, et rejoint les observations effectuées à la Section 4.1.4.3 : le bénéfice possible apporté par le filtrage désadapté décroît au fur et à mesure. Par exemple, on note un écart notable de 3 dB pour un ensemble de deux séquences, tandis que les courbes semblent converger asymptotiquement vers la même limite... Comme mentionné auparavant, il est pressenti que la cardinalité est en cause. En effet, les degrés de liberté supplémentaires amenés par les filtres désadaptés compensent de moins en moins toutes les contraintes liées aux séquences de corrélation qui, elles, s'accroissent en « M^2 ». Ils deviennent même *a fortiori* négligeables. Effectivement, lorsqu'on ne se concentre que sur la composante temporelle, il est très peu probable en pratique que l'on souhaite optimiser une centaine de filtres simultanément. Cependant, cette situation prend tout son sens dans un cas pratique — à tout hasard, l'optimisation de séquences pour le radar MIMO — où il faut prendre en compte les filtres angulaires, les filtres *Doppler*, *etc.*

Alors, étudions quelques exemples réalisables, avec la Figure 4.21. On compare ainsi les résultats obtenus par une optimisation présentée à la Section 4.1.3 avec la borne de Levenshtein du cas adapté, et la borne proposée dans ce paragraphe. Souvenez-vous : lorsqu'on a présenté une comparaison (la Section 3.6.4 du chapitre précédent) entre une optimisation

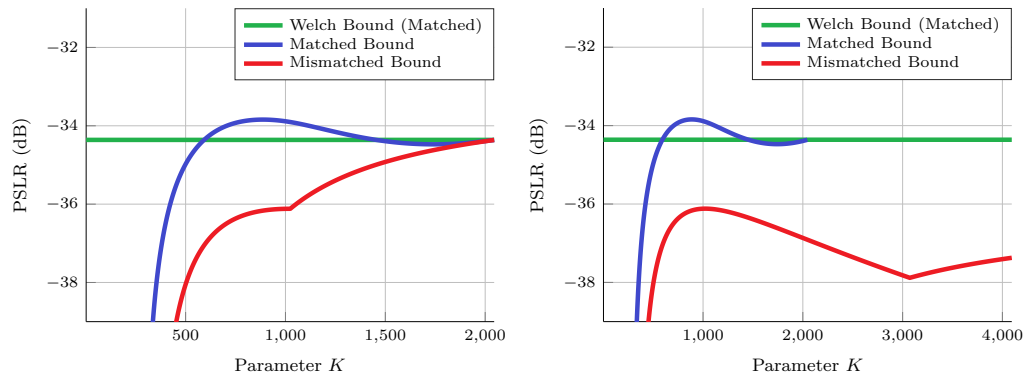


FIGURE 4.18 – Comparaison de différentes bornes en fonction de K , $M = 4$ séquences Séquences de longueur $N_s = 1024$. A gauche : filtre désadapté de longueur $N_q = N_s$. A droite : filtre désadapté de longueur $N_q = 3N_s$. On rappelle que la borne est définie par la valeur la plus haute.

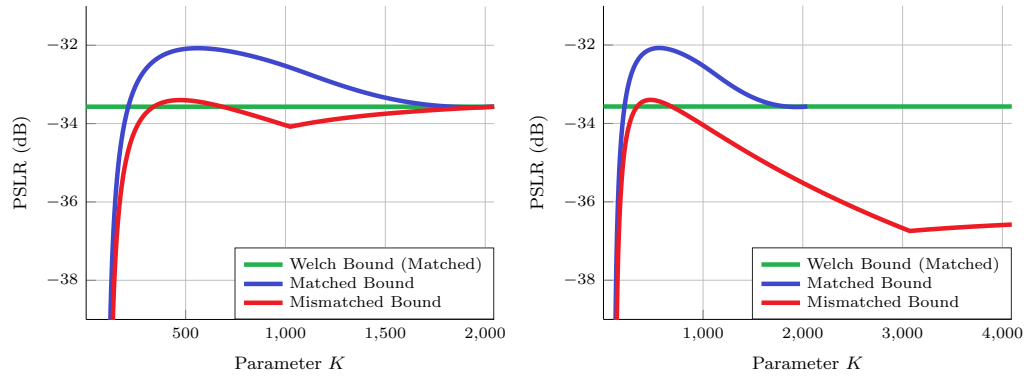


FIGURE 4.19 – Comparaison de différentes bornes en fonction de K , $M = 10$ séquences Séquences de longueur $N_s = 1024$. A gauche : filtre désadapté de longueur $N_q = N_s$. A droite : filtre désadapté de longueur $N_q = 3N_s$. On rappelle que la borne est définie par la valeur la plus haute.

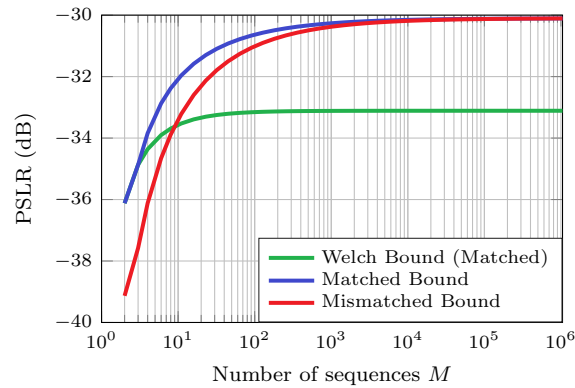


FIGURE 4.20 – Comportement des bornes en fonction du nombre de séquences Séquences de longueur $N_s = 1024$. Filtre désadapté trois fois plus long.

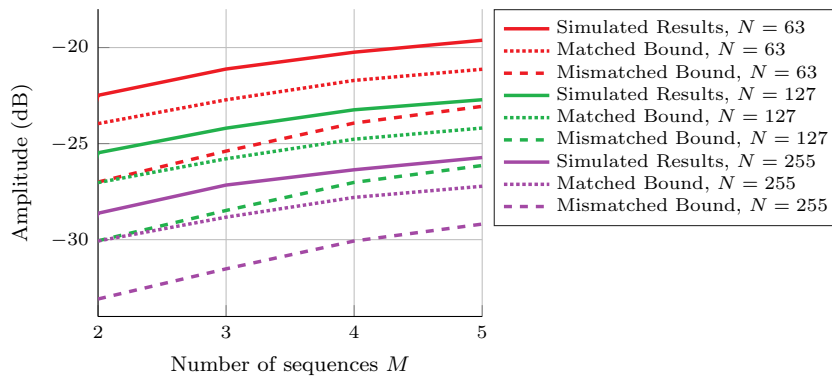


FIGURE 4.21 – Comportement des bornes en fonction de la longueur des séquences
 Les séquences simulées ont été obtenues à partir de la procédure décrite à la Section 4.1.3.

de séquences et la borne de Levenshtein, on observait une différence d'environ 3 dB, quelle que soit la longueur. Ici, on retrouve quelque chose de similaire. En effet, on constate d'une part une différence plus ou moins constante de 1.5 dB avec la borne de Levenshtein, ainsi non-égalée, malgré l'apport du filtrage désadapté. De l'autre côté, l'écart est d'environ 5 dB au départ avec la borne dans le cas désadapté, mais cet écart décroît avec le nombre de séquences.

4.3.3 Conclusion

A notre connaissance, il n'existait aucune borne sur le niveau des lobes secondaires d'un ensemble de séquences associé à des filtres désadaptés. Cela est désormais chose faite, avec la contribution de cette section. La borne ainsi présentée est assez générale : elle s'applique en effet pour des ensembles de séquences unimodulaires, et reste valable avec des corrélations périodiques ou aperiodiques. En outre, peu de contraintes ont été imposées sur les filtres. Sa démonstration est inspirée de celle de Levenshtein, que nous avons suggérée dans le chapitre précédent. Il est donc peu surprenant que les deux expressions se ressemblent beaucoup. Mais cette ressemblance nous interroge.

Du côté du pessimisme, la similitude peut justifier la constatation principale de ce chapitre. A savoir que le filtrage désadapté n'offre qu'un bénéfice contenu par rapport au filtrage classique, hormis dans le cas mono-séquence. Les degrés de liberté supplémentaires qu'il procure semblent être contrebalancés par les contraintes intrinsèques aux corrélations.

De l'autre côté, on pourrait évoquer le fait que les commentaires ci-dessus décrivent la conduite de ces bornes envers les différents paramètres. Ils ne donnent cependant aucune indication sur la précision. Sont-elles serrées ? A quel point ? Elles sont peut-être inaccessibles en pratique, mais cette assertion n'est basée sur aucun fondement. Seulement du pessimisme ? Une étude dans ce sens est donc nécessaire.

Néanmoins, nous avons rencontré un exemple où la borne était identique malgré des longueurs variables de filtres. Pourtant, le sens commun nous convaincrat qu'il ne peut y avoir qu'amélioration. Argument qui va dans le sens d'une borne trop relâchée...

Chapitre 5

Aux codes de phase pour le radar MIMO cohérent

Comme son acronyme l'indique, un radar MIMO se compose de plusieurs émetteurs et de plusieurs récepteurs. Selon la position des antennes, on distingue le radar MIMO *cohérent* [Li *et al.*, 2007] (ou co-localisé) du radar MIMO dit *statistique* [Haimovich *et al.*, 2008] (ou distribué). Dans ce chapitre, on s'intéresse au premier, dans lequel les éléments sont suffisamment proches les uns des autres, et plus précisément au design des formes d'onde.

Contrairement à un réseau phasé classique, chaque antenne émet une forme d'onde qui lui est propre. Cela se caractérise entre autres par une certaine diversité spatiale [Li *et al.*, 2009] (une forme d'onde potentiellement différente par direction d'émission) mais aussi par une variété sur la forme du faisceau émis [Aldayel *et al.*, 2017, Geng *et al.*, 2016]. En effet, lorsque les signaux sont identiques (avec uniquement une loi de phase linéaire d'antenne à antenne) — comme sur les réseaux phasés —, le faisceau est focalisé, tandis qu'il est large lorsque les signaux élémentaires sont orthogonaux. Ainsi, le choix des formes d'onde dépend des applications et des besoins pratiques. Par exemple, dans le domaine du pistage [Bar-Shalom *et al.*, 2004, Daniel *et al.*, 2014], la position de la cible est connue *a priori* : une focalisation offre une meilleure gestion énergétique. Par opposition, une émission large est plus adaptée pour une mission de surveillance ou un radar de veille. La dégradation des performances due à la non-focalisation peut être compensée par une illumination plus longue [Guyvarch *et al.*, 2012]. Le radar MIMO apporte donc une certaine flexibilité, puisqu'il est capable de générer les deux types d'émission. Et plus encore [Cheng *et al.*, 2017, Lipor *et al.*, 2014]...

Quel que soit le radar — MIMO ou non d'ailleurs —, la corrélation entre les signaux élémentaires joue un rôle important dans les performances d'un radar, comme nous avons pu le voir dans le Chapitre 2. Cependant, pour le radar MIMO cohérent, on y ajoute un petit grain de sable : le signal reçu n'est pas une réplique parmi l'un de ces signaux, mais plutôt une sommation pondérée [San Antonio *et al.*, 2007], qui diffère selon la direction de la cible. En ce sens, il serait donc intéressant de pouvoir séparer des signaux provenant de directions distinctes. Ainsi, de nombreuses études dans la littérature supposent que l'ensemble des signaux élémentaires sont parfaitement orthogonaux, de sorte que l'on puisse décomposer le signal reçu en ces différentes composantes. Malheureusement, des signaux parfaitement orthogonaux n'existent pas (*cf.* la borne de Welch à la Section 3.6.1).

De ce fait, on se propose dans ce chapitre d'établir une petite méthodologie permettant de générer des formes d'onde adaptées pour le radar MIMO cohérent. Comme vous pouvez vous y attendre, on modélisera cela sous la forme d'un problème d'optimisation qui repose sur des contraintes d'orthogonalité. Cela permettra de souligner les disparités entre une optimisation pour un radar MIMO cohérent et un radar MIMO statistique.

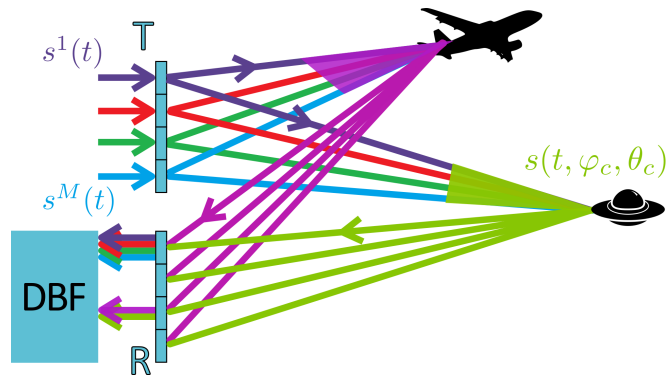


FIGURE 5.1 – Principe du radar MIMO cohérent (1)

Chaque émetteur émet sa propre forme d'onde, représentée par différentes couleurs. Chaque cible reçoit ainsi, selon sa position, une somme pondérée — potentiellement différente — des signaux élémentaires.

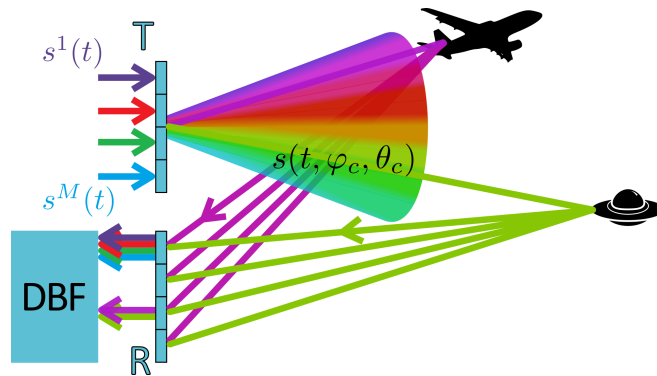


FIGURE 5.2 – Principe du radar MIMO cohérent (2)

D'un point de vue macroscopique, le radar MIMO cohérent émet un signal différent dans chaque direction de l'espace.

DBF : Digital Beamforming (formation de faisceaux par le calcul).

Ce chapitre sera composé de trois parties. La première commencera par introduire les notations et les équations régissant le radar MIMO cohérent — il y aura d'ailleurs quelques redites par rapport au premier chapitre. La deuxième modélisera ensuite le problème de génération de formes d'onde pour un radar MIMO cohérent sous la forme d'un problème d'optimisation. Cette partie se présente donc comme une application directe des chapitres précédents. Enfin, la dernière partie exhibera ensuite quelques résultats applicatifs.

5.1 Le radar MIMO cohérent

5.1.1 Généralités et notations

Commençons par une présentation simplifiée du radar MIMO cohérent. Son principe est illustré à la Figure 5.1. Comme mentionné dans l'introduction, un radar MIMO cohérent est composé de $N_T > 1$ antennes d'émission et de $N_R > 1$ antennes de réception, *suffisamment* proches les unes des autres. Chaque émetteur possède sa propre forme d'onde — ce qui signifie qu'elle peut être distincte de celle de ses voisins — et est représentée par une couleur différente sur la Figure 5.1. Le terme « émission colorée » est ainsi parfois utilisé pour parler du radar MIMO cohérent. Soit s^m le signal élémentaire émis par la m -ème antenne.

Chaque cible potentielle reçoit une somme cohérente de ces signaux élémentaires. Cependant, selon la position angulaire (azimut φ_c , élévation θ_c) de cette cible par rapport au radar, ces signaux élémentaires peuvent être légèrement décalés en phase. De ce fait, le signal reçu par des cibles à des positions distinctes est potentiellement différent (Figure 5.2). En d'autres termes, le radar MIMO cohérent est un radar qui émet simultanément des signaux différents dans les différentes directions de l'espace. D'ailleurs, le signal émis par le radar dans la direction (φ_c, θ_c) , noté $s(\varphi_c, \theta_c)$, s'écrit :

$$s(t, \varphi_c, \theta_c) = \sum_{m=1}^{N_T} e^{j\mathbf{x}_{T,m}^T \mathbf{k}(\varphi_c, \theta_c)} s^m(t), \quad (5.1)$$

où :

- $\mathbf{x}_{T,m}$ est la position de la m -ème antenne d'émission ;
- λ est la longueur d'onde du signal ;
- $\mathbf{k}(\varphi_c, \theta_c)$ est le vecteur d'onde dans la direction (φ_c, θ_c) , défini par :

$$\mathbf{k}(\varphi_c, \theta_c) := \frac{2\pi}{\lambda} [\cos(\varphi_c) \sin(\theta_c), \sin(\varphi_c) \sin(\theta_c), \cos(\theta_c)]^T; \quad (5.2)$$

- s^m est le signal élémentaire émis par la m -ème antenne.

La position des éléments à la réception introduit également un changement de phase. Ainsi, en notant $\mathbf{x}_{R,n}$ sa position, la n -ième antenne de réception reçoit :

$$s_r^n(t, \varphi_c, \theta_c) = e^{j\mathbf{x}_{R,n}^T \mathbf{k}(\varphi_c, \theta_c)} s(t - \tau, \varphi_c, \theta_c), \quad (5.3)$$

où τ représente le retard induit par la position de la cible. Comme maintes fois mentionné précédemment, nous n'avons pas pris en compte le Doppler ou encore le bruit dans l'expression du signal reçu.

5.1.2 Fonction d'ambiguïté

Le traitement classique radar est appelé compression d'impulsion. Cela consiste à identifier la présence du signal émis dans le signal reçu, reflétant ainsi l'existence potentielle d'une cible. Pour cela, une comparaison (*i.e.* une convolution) sur le domaine temporel est effectuée entre le signal reçu et une réplique décalée temporellement du signal émis. Ce dernier joue le rôle de filtre, que l'on connaît habituellement sous le nom de *filtre adapté*, et permet de maximiser le rapport signal sur bruit (SNR) [Levanon *et al.*, 2004]. Pour une hypothèse de retard τ , la sortie après filtrage adapté s'écrit :

$$\chi(\tau, \varphi, \theta, \varphi_c, \theta_c) = \int_{-\infty}^{+\infty} s(t, \varphi_c, \theta_c) \underbrace{s^*(t - \tau, \varphi, \theta)}_{\substack{\text{filtre adapté associé} \\ \text{à la direction } (\varphi, \theta)}} dt. \quad (5.4)$$

Ainsi, un filtre adapté MIMO dépend de la direction azimut-élévation (φ, θ) , comme l'indique l'équation (5.4). De ce fait, le traitement d'un signal MIMO ne s'effectue pas à l'aide d'un seul filtre mais avec plusieurs, chacun étant associé à une direction, entraînant un coût calculatoire supplémentaire. L'ensemble de ces directions (habituellement contenues dans la zone de vision du radar) sera appelé directions d'intérêt par la suite. Soulignons donc que le filtrage ne s'effectue pas nécessairement avec les séquences élémentaires s^m (sauf dans des cas très particuliers).

La fonction d'ambiguïté MIMO [San Antonio *et al.*, 2007] indique la valeur de la réponse impulsionnelle en sortie de filtre, mais selon un ensemble de paramètres. Le choix du filtre adapté correspond à une hypothèse sur les variables angulaires (φ, θ) ; une autre hypothèse doit également se faire sur l'origine du signal reçu, donc sur la position angulaire de la cible

(φ_c, θ_c) . Nous nous permettons d'insister (encore une fois) sur le fait que ces couples ne sont pas nécessairement identiques, sauf à la bonne hypothèse.

Comme indiqué au Chapitre 1, il est préférable d'étudier la fonction d'ambiguïté en prenant l'ensemble des contributions des récepteurs, en lieu et place des signaux émis. Après quelques développements (*cf.* Section 1.2.2.2), cela nous donne la fonction d'ambiguïté χ_2 suivante :

$$\chi_2(\tau, \varphi, \theta, \varphi_c, \theta_c) = \left(\sum_{n=1}^{N_R} e^{j(\mathbf{x}_{R,n}^T \mathbf{k}(\varphi_c, \theta_c) - \mathbf{k}(\varphi, \theta))} \right) \times \left(\sum_{m, m'=1}^{N_T} e^{j(\mathbf{x}_{T,m}^T \mathbf{k}(\varphi_c, \theta_c) - \mathbf{x}_{T,m'}^T \mathbf{k}(\varphi, \theta))} \int_{-\infty}^{+\infty} s^m(t) (s^{m'}(t - \tau))^* dt \right). \quad (5.5)$$

Les différents termes qui composent l'expression (5.5) font transparaître les différentes étapes de traitement d'un signal MIMO cohérent avec, pour commencer, une FFC réception, suivie (généralement) d'une FFC émission en simultanée avec une compression d'impulsion. Pour plus de simplicité, seule la composante élévation θ sera conservée dans la suite de l'étude — apparaissant notamment dans $\mathbf{k}(\varphi, \theta)$.

5.1.3 Écriture matricielle, discrétisation

Écriture matricielle Remarquons qu'il est assez courant d'utiliser une notation matricielle. Soient $\mathbf{s}(t) = [s^1(t), \dots, s^{N_T}(t)]$ un vecteur concaténant l'ensemble des signaux élémentaires à l'instant t , $\mathbf{a}_t(\theta)$ et $\mathbf{a}_r(\theta)$ les matrices de phases pour l'émission et la réception :

$$\mathbf{a}_t(\theta) := \begin{bmatrix} e^{-j\mathbf{x}_{T,1}^T \mathbf{k}(\theta)} \\ e^{-j\mathbf{x}_{T,2}^T \mathbf{k}(\theta)} \\ \vdots \\ e^{-j\mathbf{x}_{T,N_T}^T \mathbf{k}(\theta)} \end{bmatrix}, \quad \mathbf{a}_r(\theta) := \begin{bmatrix} e^{-j\mathbf{x}_{R,1}^T \mathbf{k}(\theta)} \\ e^{-j\mathbf{x}_{R,2}^T \mathbf{k}(\theta)} \\ \vdots \\ e^{-j\mathbf{x}_{R,N_R}^T \mathbf{k}(\theta)} \end{bmatrix}. \quad (5.6)$$

On peut remarquer que :

$$s(t, \theta) = \mathbf{s}(t) \mathbf{a}_t^*(\theta), \quad \chi(\tau, \theta, \theta_c) = \mathbf{a}_t^H(\theta) \left(\int_{-\infty}^{+\infty} \mathbf{s}^T(t) \mathbf{s}^*(t - \tau) dt \right) \mathbf{a}_t(\theta_c), \quad (5.7)$$

et que :

$$\chi_2(\tau, \theta, \theta_c) = \left(\mathbf{1}_{1, N_R} \mathbf{a}_r(\theta_c) \mathbf{a}_r^H(\theta) \mathbf{1}_{N_R, 1} \right) \chi(\tau, \theta, \theta_c). \quad (5.8)$$

Discrétisation Par la suite, nous allons considérer des signaux discrets à module constant. Pour cela, on décompose un signal s^m de durée $T \in \mathbb{R}_+^*$ en $N \in \mathbb{N}^*$ échantillons de même durée $\Delta t := T/N$. Par approximation, on attribue une phase $\varphi_n^m \in [0, 2\pi[$ à chacun de ces échantillons, de sorte que l'enveloppe complexe du signal est :

$$s^m(t) \approx \frac{1}{\sqrt{T}} \sum_{n=1}^N s_n^m \text{rect} \left(\frac{t - (n-1)\Delta t}{\Delta t} \right), \quad (5.9)$$

où $s_n^m = \exp(j\varphi_n^m)$. On identifie la séquence $\mathbf{s}^m := [s_1^m, \dots, s_N^m]^T$ au code de phase associé au signal s^m (cependant, par abus de notation, on emploiera souvent le terme « signal » pour qualifier la séquence \mathbf{s}^m). Soit $\mathbf{S} = [\mathbf{s}^1, \dots, \mathbf{s}^{N_T}] \in \mathbb{C}^{N \times N_T}$ la concaténation de tous les signaux élémentaires discrétisés.

Remarque 5.1. (*Approximation et codes de phase*) A ce stade, on a approché un signal s^m en un code de phase. Cela ne veut donc bien sûr pas dire que s^m soit initialement un code de phase, bien qu'on s'intéressera à eux par la suite...

En sus des signaux, on considère la version discrète de la fonction de corrélation, que l'on note ACCF comme expliqué à la Section 3.5. Cela nous permet enfin d'écrire la version discrétisée de la fonction d'ambiguïté MIMO :

$$\begin{aligned} \mathbf{s}(\theta) &= \mathbf{S}\mathbf{a}_t^*(\theta), \\ \chi(k, \theta, \theta_c) &= \left(\text{AACF}(\mathbf{s}(\theta), \mathbf{s}(\theta_c)) \right)_k, \\ &= \mathbf{a}_t^H(\theta) \mathbf{R}(k) \mathbf{a}_t(\theta_c), \end{aligned} \quad (5.10)$$

où les matrices carrées $\mathbf{R}(k)$ de taille N_T définissent les corrélations entre les signaux élémentaires à un retard k donné. Elles sont donc au nombre de $2N - 1$ et sont définies par :

$$\mathbf{R}(k)_{m,m'} = \left(\text{AACF}(\mathbf{s}^m, \mathbf{s}^{m'}) \right)_k. \quad (5.11)$$

5.1.4 Différentes formes d'onde

Pour l'instant, nous n'avons encore émis aucune hypothèse sur l'ensemble des signaux élémentaires $\{s^m\}$. Le paragraphe 5.1.2 a mentionné l'importance de la forme d'onde sur la fonction d'ambiguïté, et donc *a fortiori* sur les performances du système. Mais son choix est également déterminé par les besoins opérationnels (liés par exemple à l'environnement ou à la cible) et les contraintes matérielles (largeur de bande, *etc.*) [Blunt *et al.*, 2016].

Il existe de nombreuses formes d'onde pour le radar MIMO cohérent. Un classement, en particulier pour les codes *intrapulses*, a été proposé dans [Sun *et al.*, 2014], et repris à la Section 1.2.4. Résumons-en ci-dessous les grandes lignes :

- TDMA (*Time Division Multiple Access*). Un même signal est émis alternativement sur chaque antenne mais à des instants différents. Il y a donc orthogonalité des signaux au retard nul seulement.
- FDMA (*Frequency Division Multiple Access*). Chaque antenne émet à une fréquence particulière, différente de celle des autres. Les séquences élémentaires peuvent être orthogonales entre elles (selon l'écart fréquentiel entre les antennes), mais cela ne signifie en aucun cas que leur autocorrélation est parfaite...
- CDMA (*Code Division Multiple Access*). Chaque antenne possède son propre code de phase. Il existe de nombreuses familles de séquences pseudo-orthogonales définies dans la littérature, à l'origine pour le domaine des télécommunications. Une liste peut être trouvée ici [Arlery *et al.*, 2015].
- Quelques formes hybrides, comme le codage par *frequency-hopping* [Han *et al.*, 2016] (également issu des télécoms) ou les codes circulants [Babur *et al.*, 2016].

Chaque catégorie présente ainsi diverses attitudes envers la propriété d'orthogonalité. Or, plusieurs études [Rabaste *et al.*, 2013, Sun *et al.*, 2014] ont établi que l'ensemble de ces formes d'onde souffrent d'un couplage distance/élévation plus ou moins prononcé. Cela se traduit par une augmentation de la variance de l'estimée de la position de la cible, c'est-à-dire, en d'autres termes, une mesure moins précise. Sur l'exemple de la Figure 1.23 (et reprise à la Figure 5.3 par commodité), chaque signal élémentaire est un *chirp* partageant la même largeur de bande. Il est assez clair que la fonction représentée ne ressemble en rien à une punaise. Elle dévoile à la place des stries diagonales assez prononcées, signe de ce couplage. En fait, il se trouve que ce sont les codes CDMA qui proposent le couplage le moins prononcé, mais avec des lobes secondaires plus élevés en contrepartie. Pour cette raison, nous avons décidé de nous intéresser aux codes de phase dans cette thèse...

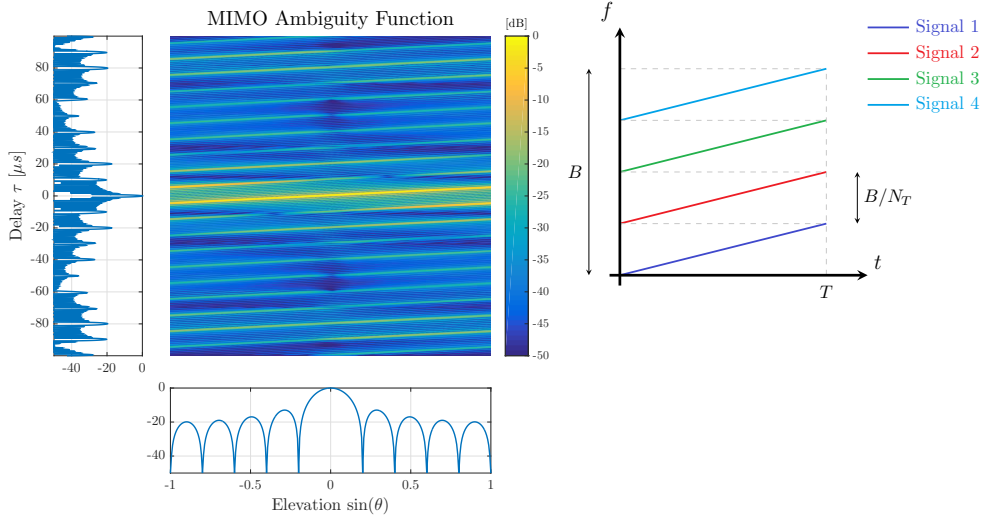


FIGURE 5.3 – Fonction d’ambiguïté pour un codage FDMA

A gauche : représentation de la fonction d’ambiguïté pour un codage FDMA.

Antenne composée de 10 émetteurs, sans traitement à la réception.

Chacun émet un chirp dans une bande de fréquence dédiée, comme schématisé à droite.

5.1.5 Beam pattern

Comme l’indique l’équation (5.1), le signal émis par un radar MIMO cohérent diffère selon les directions, ce qui sous-entend que l’antenne n’est pas nécessairement isotrope (l’énergie rayonnée dans chacune des directions peut également être fluctuante). Pour être plus précis, il a été montré dans [Fuhrmann *et al.*, 2008] que la densité de puissance normalisée P est une fonction de la direction angulaire θ_c (et de φ_c dans le cas général), et est définie par :

$$P(\theta_c) = \frac{1}{4\pi} \mathbf{a}_t^H(\theta_c) \mathbf{R}(0) \mathbf{a}_t(\theta_c). \quad (5.12)$$

L’expression précédente suggère que le *beam pattern* est lié à la matrice de covariance des signaux élémentaires (les détails ont été donnés à la Section 1.2.3).

- Si les signaux $\{\mathbf{s}^m\}$ sont identiques, alors un radar MIMO cohérent se réduit à un réseau phasé classique. Il émet ainsi un diagramme en « sinus cardinal », qui est potentiellement (et relativement) fin et focalisé.
- À l’opposé, des signaux parfaitement orthogonaux se traduisent par un faisceau plus « large », omnidirectionnel disons (si les antennes élémentaires le sont également).

Ces deux possibilités sont illustrées à la Figure 5.4, et montrent les avantages d’un radar MIMO cohérent à ce sujet. Contrairement à un réseau linéaire classique où, on le répète, le faisceau sera nécessairement focalisé — peu importe la forme d’onde, dès lors que la phase appliquée soit linéaire —, la forme du faisceau d’un radar MIMO cohérent peut être adaptée selon les besoins opérationnels : d’un faisceau fin pour du pistage par exemple [Bar-Shalom *et al.*, 2004, Pilté *et al.*, 2017], à un autre, plus large, pour de la veille ou des applications de surveillance. La forme du faisceau étant régie par la matrice de corrélation $\mathbf{R}(0)$, il est en outre possible de modéliser des *beam patterns* beaucoup plus exotiques. En effet, à l’aide d’un problème d’optimisation, les auteurs de [Cheng *et al.*, 2017, Guo *et al.*, 2015, Lipor *et al.*, 2014, Wang *et al.*, 2012] ont proposé des faisceaux avec plusieurs « lobes principaux », comme dépeint en cette même figure.

Remarque 5.2. (*Réseau virtuel et diagramme d’antenne*) Soient N_T et N_R le nombre d’émetteurs et de récepteurs de notre radar MIMO, respectivement. Rappelons les quelques hypothèses théoriques qui nous intéressent dans cette remarque.

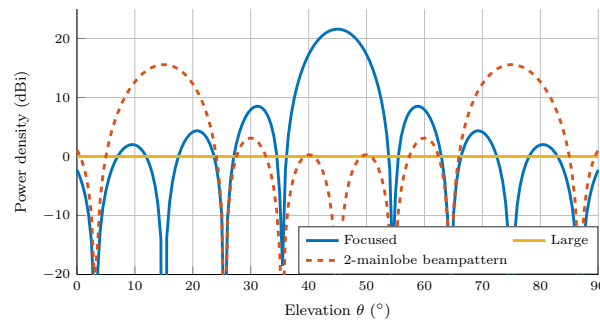


FIGURE 5.4 – Exemples de faisceaux pour un radar MIMO composé de $N_T = 12$ antennes. Le faisceau avec deux « lobes principaux » a été obtenu à l'aide de [Lipor et al., 2014].

1. Chaque récepteur reçoit N_T contributions provenant de chacun des émetteurs. Cette hypothèse est caractéristique des radars MIMO, cohérents ou non.
2. Les N_R récepteurs captent le même signal, à un décalage de phase près cependant ; ce décalage est dû à leur position physique.

Du point de vue de l'ensemble du radar, il y a donc $N_T N_R$ séquences à considérer. En les traitant séparément, on pourrait se convaincre que cela équivaut à travailler avec $N_T N_R$ systèmes indépendants, dont les coordonnées traduisent les décalages de phase. C'est le principe du « réseau virtuel » (virtual array dans la langue de Shakespeare), illustré à la Figure 5.5. Ce parallèle permet de mettre en évidence quelques propriétés de notre radar MIMO [Chen, 2009, Davis et al., 2014].

1. Ainsi on a, semble-t-il, équivalence (en terme de diagramme d'antenne — car il y a tout de même une différence de puissance) entre un système à $N_T + N_R$ éléments et un autre à $N_T N_R$ éléments : le MIMO offre un possible gain matériel, à compenser par les différentes interactions entre antennes.
2. Le réseau d'antennes n'est, après tout, qu'une antenne plus large. Dans le cas d'un réseau lacunaire (linéaire, comme à la Figure 5.5), cela entraîne une amélioration de la résolution angulaire, puisqu'il y a proportionnalité entre la dimension de l'antenne et sa résolution (Section 1.1).

Couronnons cette remarque sur une note plus négative : le réseau virtuel s'échafaude sur la condition qu'on puisse récupérer et séparer les différentes séquences élémentaires. Comme nous le verrons par la suite, cela n'est pas si évident...

5.2 Un problème d'optimisation

Résumons en quelques mots la Section 5.1 :

- Un radar MIMO cohérent émet un signal qui correspond à une somme pondérée des signaux élémentaires. Cette somme varie selon la direction d'émission (φ_c, θ_c) . De ce fait, le filtrage adapté s'effectue à l'aide d'un ensemble de filtres, chacun dépendant d'une direction d'intérêt (φ, θ) . Cet ensemble ne correspond donc généralement pas aux signaux élémentaires s^m .
- En jouant sur les signaux élémentaires et sur leur matrice de covariance, un radar MIMO cohérent a la capacité d'émettre un faisceau focalisé ou large, selon ses envies, mais aussi les contraintes opérationnelles.
- Il existe plusieurs catégories de formes d'onde. Parmi elles, nous avons décidé dans cette thèse de nous concentrer sur les codes de phase.

Afin de modéliser un problème d'optimisation permettant de générer des codes de phase pour le radar MIMO cohérent, il nous faut donc prendre en compte tous ces points. Entre les besoins et l'optimisation, une petite traduction s'impose...

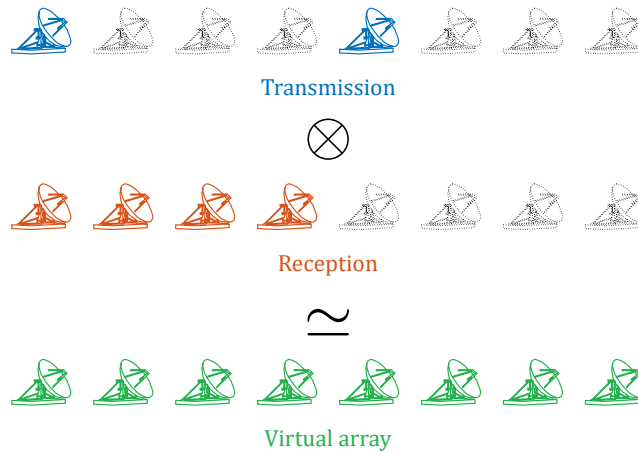


FIGURE 5.5 – Principe du réseau virtuel

Si le placement des antennes est judicieusement choisi, le réseau virtuel se compose d'antennes émission/réception dont les positions sont définies par les convolutions de celles de leur parenté.

1. Bien que l'on recherche les signaux élémentaires s^m , on se doit d'étudier le niveau des lobes secondaires des séquences émises en différentes directions (φ_c, θ_c) . Avant cela, il nous faut aussi définir ces directions d'intérêt.
2. Dans le même temps, le *beam pattern* doit être contrôlé.
3. Les variables d'études seront des séquences polyphases, avec une contrainte sur le module constant, comme expliqué à la Section 2.1.

Finalement, la présente section se compose de parties reprenant chacun de ces points. Mais avant de la commencer, jetons un coup d'œil à ce qu'en dit la littérature à ce sujet, avec une petite revue en quelques lignes...

3. La contrainte de module constant est généralement de rigueur dans la recherche de séquences. Bien qu'il est possible d'en considérer une relaxation...
2. Il y a une pléthore de travaux sur la gestion du *beam pattern* : sur la forme du faisceau comme mentionné précédemment, s'appliquant sur des codages polyphases [Aldayel *et al.*, 2016, Aubry *et al.*, 2016, Hu *et al.*, 2009, Stoica *et al.*, 2008] ou fréquentiels [Li *et al.*, 2017, Liu *et al.*, 2014b]... La fonction de coût employée concerne souvent le SINR (*Signal-to-Interference-plus-Noise Ratio*).
1. En ce qui concerne l'optimisation de la fonction d'ambiguïté MIMO, la littérature se concentre majoritairement dans le cas fréquentiel (*cf.* [Babur *et al.*, 2016] ou encore [Hassanien *et al.*, 2017]). Mais, dans le cas polyphase, elle se fait beaucoup plus rare. Citons néanmoins [Liu *et al.*, 2014a, Wang *et al.*, 2012, Zang *et al.*, 2014] : ces derniers travaux se rapprochent au mieux de ce que nous allons faire dans ce chapitre.
0. Le *beam pattern* et la fonction d'ambiguïté MIMO ne sont pas les seuls critères employés. Par exemple, les auteurs de [Tang *et al.*, 2016] ont généré des formes d'onde selon les fluctuations de Swerling de la cible, tandis que ceux de [Zang *et al.*, 2015, Mendes Ruiz *et al.*, 2015] ont joué avec la forme et le positionnement des antennes.

5.2.1 Sur les lobes secondaires

Pour un radar MIMO cohérent, la compression d'impulsion d'un signal s'effectue à l'aide d'un banc de filtres. En effet, à chaque filtre correspond une direction d'émission θ , comme l'indique l'équation (5.4). Il nous faut ainsi en définir plusieurs : soient Θ^f l'ensemble de ces

directions, qualifiées d'intérêt par la suite, et $S(\Theta^f)$ les signaux associés, tels que :

$$\begin{cases} \Theta^f := \{\theta_1^f, \dots, \theta_{|\Theta^f|}^f\}, \\ S(\Theta^f) := \{\mathbf{s}(\theta_1^f), \dots, \mathbf{s}(\theta_{|\Theta^f|}^f)\}. \end{cases} \quad (5.13)$$

La Remarque 5.4 ci-dessous porte sur le nombre de directions $|\Theta^f|$.

La fonction d'ambiguïté χ (5.4) dépend de trois paramètres : la position angulaire de la cible θ_c , et les hypothèses sur la « distance » τ et l'élévation θ du filtre. Afin d'obtenir des performances optimales, le couple radar/signal se doit d'en présenter une sous la forme d'une punaise. En d'autres termes, cela signifie que χ ...

- ... atteint son maximum lorsque les hypothèses sont vérifiées, *i.e.* $\theta = \theta_c$ et $\tau = 0$,
- ... est minimale ailleurs.

Malheureusement, une telle fonction n'existe pas, à cause de la propriété de conservation de l'énergie [San Antonio *et al.*, 2007]. Néanmoins, l'objectif de notre optimisation sera de se rapprocher au mieux de celle-ci. D'ailleurs, si on y réfléchit à deux fois, l'énumération précédente se révèle être des conditions sur l'orthogonalité de l'ensemble $S(\Theta^f)$, la fonction d'ambiguïté étant basée sur des corrélations. Autrement dit, on se retrouve avec un problème d'orthogonalité similaire à celui présenté à la Section 3.5 du Chapitre 2.

En reprenant les notations de ladite section, soit h_p une mesure générique des lobes secondaires de corrélation, telle que, pour une paire de séquence (\mathbf{s}, \mathbf{q}) :

$$h_p(\mathbf{s}, \mathbf{q}) := \left\| \mathbf{F}_{\mathbf{s}, \mathbf{q}} \text{ACCF}(\mathbf{s}, \mathbf{q}) \right\|_p, \quad (5.14)$$

où les définitions de la matrice $\mathbf{F}_{\mathbf{s}, \mathbf{q}}$ et de la fonction d'inter-corrélation AACF ont été données précédemment. Optimiser l'orthogonalité d'un ensemble de séquences $S = \{\mathbf{s}^1, \dots, \mathbf{s}^M\}$ consiste à considérer toutes les corrélations induites par chaque paire, ce qui nous amène à la fonction v_p suivante :

$$\begin{aligned} (\mathbf{T}^N)^M &\rightarrow \mathbb{R}_+ \\ (\mathbf{s}^1, \dots, \mathbf{s}^M) &\mapsto v_p(\mathbf{s}^1, \dots, \mathbf{s}^M) := \left(\sum_{m_i, m_j=1}^M \left(h_p(\mathbf{s}^{m_i}, \mathbf{s}^{m_j}) \right)^p \right)^{1/p}. \end{aligned} \quad (5.15)$$

Dans notre cas, la quantité qui nous intéresse est la suivante (avec un petit abus de notation sur la variable d'entrée) :

$$v_p(S(\Theta^f)) = \left(\sum_{\theta_i^f, \theta_j^f \in \Theta^f} \left| \sum_{n=1}^{N_R} e^{j(\mathbf{x}_{R,n}^T (\mathbf{k}(\theta_i^f) - \mathbf{k}(\theta_j^f)))} \right|^p \left(h_p(\mathbf{s}(\theta_i^f), \mathbf{s}(\theta_j^f)) \right)^p \right)^{1/p}. \quad (5.16)$$

Remarquez qu'un terme supplémentaire s'est glissé, compte tenu de l'expression (5.15). Ce coefficient — présent dans la fonction d'ambiguïté complète χ_2 — permet de prendre en compte les apports de la FFC à la réception, et pondère ainsi certaines corrélations. Ainsi, l'orthogonalité porte bien sur $S(\Theta^f)$, mais il ne faut pas oublier que nous recherchons les séquences élémentaires ! Le changement de variables est heureusement immédiat ; il suffit en effet de poser :

$$\begin{aligned} (\mathbf{T}^N)^{N_T} &\rightarrow (\mathbb{C}^N)^{|\Theta^f|} \\ (\mathbf{s}^1, \dots, \mathbf{s}^{N_T}) &\mapsto u(\mathbf{s}^1, \dots, \mathbf{s}^{N_T}) := \left(\mathbf{S}\mathbf{a}_t(\theta_1^f), \dots, \mathbf{S}\mathbf{a}_t(\theta_{|\Theta^f|}^f) \right), \end{aligned} \quad (5.17)$$

où $\mathbf{S} = [\mathbf{s}^1, \dots, \mathbf{s}^{N_T}]$ est une concaténation des vecteurs de l'ensemble, pour que, finalement, la fonction à minimiser f_p s'écrive :

$$\begin{aligned} (\mathbf{T}^N)^{N_T} &\rightarrow \mathbb{R}_+ \\ (\mathbf{s}^1, \dots, \mathbf{s}^{N_T}) &\mapsto f_p(\mathbf{s}^1, \dots, \mathbf{s}^{N_T}) := v_p(u(\mathbf{s}^1, \dots, \mathbf{s}^{N_T})). \end{aligned} \quad (5.18)$$

En résumé, la fonction f_p ...

1. Prend les séquences élémentaires $\mathbf{s}^1, \dots, \mathbf{s}^{N_T}$ en tant que variable ;
2. Effectue une transformation afin de considérer les séquences émises dans les directions d'intérêt (la fonction u) ;
3. Évalue l'orthogonalité entre ces différentes séquences, par le biais de la fonction v_p .

Remarque 5.3. (Et l'orthogonalité sur les séquences élémentaires ?) Supposons que l'on souhaite récupérer les différentes composantes du signal reçu. On considérerait les séquences élémentaires en tant que filtres, et le traitement s'effectuerait en trois étapes, comme suit.

1. Sur chaque antenne de réception, disons la n -ème ici, on recherche la signature de la séquence élémentaire $s^{m'}$, pour une hypothèse de retard τ :

$$\begin{aligned} y^{n,m'}(\tau, \tau_c, \varphi_c, \theta_c) &= \int_{-\infty}^{+\infty} s_r^n(t, \varphi_c, \theta_c) s^{m'}(t - \tau)^* dt \\ &= e^{j\mathbf{x}_{R,n}^T \mathbf{k}(\varphi_c, \theta_c)} \sum_{m=1}^{N_T} e^{j\mathbf{x}_{T,m}^T \mathbf{k}(\varphi_c, \theta_c)} \int_{-\infty}^{+\infty} s^m(t) s^{m'}(t - \tau)^* dt. \end{aligned} \quad (5.19)$$

2. On effectue ensuite un traitement à l'émission, sous l'hypothèse angulaire (φ, θ) . En rassemblant l'ensemble des contributions élémentaires précédentes, on a :

$$\begin{aligned} y^n(\tau, \varphi, \theta, \varphi_c, \theta_c) &= \sum_{m'=1}^{N_T} e^{-j\mathbf{x}_{T,m'}^T \mathbf{k}(\varphi, \theta)} y^{n,m'}(\tau, \varphi_c, \theta_c) \\ &= e^{j\mathbf{x}_{R,n}^T \mathbf{k}(\varphi_c, \theta_c)} \sum_{m,m'=1}^{N_T} e^{j(\mathbf{x}_{T,m}^T \mathbf{k}(\varphi_c, \theta_c) - \mathbf{x}_{T,m'}^T \mathbf{k}(\varphi, \theta))} \\ &\quad \int_{-\infty}^{+\infty} s^m(t) s^{m'}(t - \tau)^* dt. \end{aligned} \quad (5.20)$$

3. De façon similaire, on applique le « traitement à la réception » pour la même hypothèse angulaire. L'expression ci-dessous regroupe l'ensemble des sorties des récepteurs.

$$\sum_{n=1}^{N_R} e^{-j\mathbf{x}_{R,n}^T \mathbf{k}(\varphi, \theta)} y^n(\tau, \varphi, \theta, \varphi_c, \theta_c) = \chi_2(\tau, \varphi, \theta, \varphi_c, \theta_c). \quad (5.21)$$

Finalement, on retrouve la fonction d'ambiguïté (5.5). Cette approche est ainsi équivalente au traitement présenté précédemment (FFC réception suivie d'un filtrage adapté « angulaire »). Mais ne vous y méprenez pas : les séquences élémentaires jouent correctement leur rôle de filtre à condition que des hypothèses angulaires soient également posées. La nécessité de leur orthogonalité n'intervient donc pas directement dans le traitement.

En clair, l'orthogonalité des séquences élémentaires...

- permet de factoriser le traitement à l'émission ;
- permet de définir un faisceau large [Fuhrmann et al., 2008] ;
- est préférable dans le cas d'un radar MIMO distribué [Arlery et al., 2015].

Remarque 5.4. (Nombre de filtres et voisinage) Par la suite, nous allons prendre un nombre de filtres lié au nombre d'émetteurs, plus précisément $|\Theta_f| = N_T - 1$. Pourquoi prendre si peu de filtres dans le processus d'optimisation ? Il est pourtant vrai qu'augmenter leur nombre accroîtrait la précision... Mais, dans ce cas, il faut faire attention aux effets de voisinage. Pour expliquer ce phénomène, il est plus simple de se pencher sur un cas pathologique.

Considérons que l'ensemble Θ^f est composé d'un très grand nombre d'éléments (pas une infinité, sinon le problème serait mal défini). D'un côté, on pourrait se dire que le problème d'optimisation n'est pas plus contraint que cela, car optimiser l'autocorrélation d'un signal dans une direction entraîne nécessairement celle de la voisine. Mais qu'en est-il de l'inter-corrélation entre ces deux signaux ? Il y a fort à parier que ces signaux se ressemblent. Ainsi, minimiser leur inter-corrélation consisterait à minimiser l'énergie de leurs autocorrélations ; il en suit que le problème d'optimisation consisterait à rechercher des signaux permettant d'émettre des zéros partout... ce que nous ne souhaitons absolument pas (d'ailleurs, cela risque d'être compliqué avec la contrainte d'énergie imposée)...

Finalement, on choisira $|\Theta_f| = N_T - 1$. Cela provient du nombre de directions élémentaires orthogonales que l'on peut former avec un réseau de N_T antennes, comme l'illustre la Figure 5.4.

Calcul de la dérivée Nous n'allons pas dans ce paragraphe donner la formule explicite du gradient de la fonction f_p , car nous l'avons calculée dans [Tan *et al.*, 2016b]. Cependant, à l'aide des définitions précédentes, nous pouvons donner quelques indications.

La fonction f_p est une composition entre u et v_p . Le théorème des fonctions composées [Hjørungnes, 2011] nous permet d'écrire :

$$J_{f_p}(\mathbf{S}) = J_{v_p}(u(\mathbf{S}))J_u(\mathbf{S}) + J_{v_p}(u(\mathbf{S})^*)J_{u^*}(\mathbf{S}) \quad (5.22)$$

Nous devons donc calculer les jacobiniennes des fonctions u et v_p .

Commençons par considérer la version matricielle de la fonction u , que l'on note, par abus de notation, de la même manière :

$$\mathbb{C}^{N \times N_T} \rightarrow \mathbb{C}^{N \times |\Theta_f|} \\ \mathbf{S} \mapsto u(\mathbf{S}) := \mathbf{S} \mathbf{A}_t(\Theta_f) \quad \text{avec } \mathbf{A}_t(\Theta_f) = \left[\mathbf{a}_t(\theta_1^f), \dots, \mathbf{a}_t(\theta_{|\Theta_f|}^f) \right]. \quad (5.23)$$

Par un opérateur de vectorisation [Hjørungnes, 2011], sa jacobienne s'écrit :

$$J_u(\mathbf{S}) = \begin{bmatrix} \frac{\partial u_{1,1}}{\partial s_{1,1}} & \frac{\partial u_{1,1}}{\partial s_{2,1}} & \dots & \frac{\partial u_{1,1}}{\partial s_{N,1}} & \frac{\partial u_{1,1}}{\partial s_{1,2}} & \dots & \frac{\partial u_{1,1}}{\partial s_{N,N_T}} \\ \frac{\partial u_{2,1}}{\partial s_{1,1}} & & & & & & \vdots \\ \vdots & & & & & & \vdots \\ \frac{\partial u_{N,1}}{\partial s_{1,1}} & & & & & & \vdots \\ \frac{\partial u_{1,2}}{\partial s_{1,1}} & & & & & & \vdots \\ \vdots & & & & & & \vdots \\ \frac{\partial u_{N,|\Theta_f|}}{\partial s_{1,1}} & \dots & \dots & \dots & \dots & \dots & \frac{\partial u_{N,|\Theta_f|}}{\partial s_{N,N_T}} \end{bmatrix} \in \mathbb{C}^{[N|\Theta_f|] \times [NN_T]}. \quad (5.24)$$

On peut aisément montrer que $J_u(\mathbf{S}) = \mathbf{A}_t(\Theta_f)^T \otimes \mathbf{I}_N$ et que $J_{u^*}(\mathbf{S}) = \mathbf{0}$.

En ce qui concerne la jacobienne de la fonction v_p , $J_{v_p}(u(\mathbf{S}))$, la Section 3.5 nous donne suffisamment d'éléments pour qu'on puisse la déduire. Notez simplement que, par vectorisation, celle-ci est un vecteur ligne de longueur $N|\Theta_f|$.

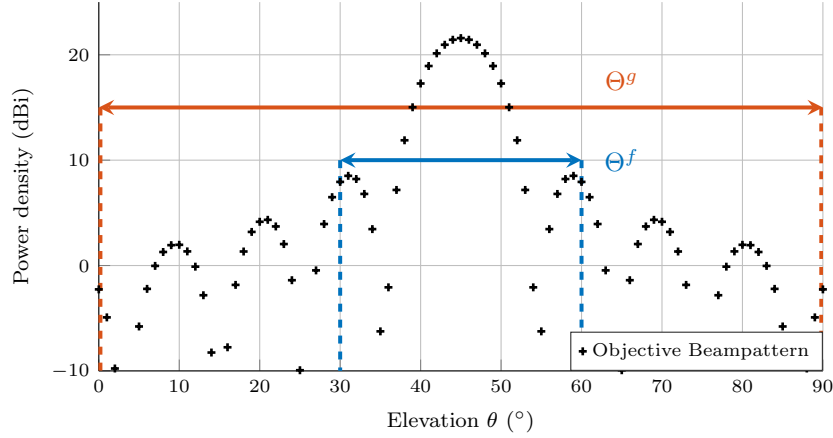


FIGURE 5.6 – Un exemple de directions d'intérêt, pour un faisceau focalisé
Remarque que la discrétisation angulaire pour les corrélations, notée Θ^f , n'est pas nécessairement identique à celle pour la forme du faisceau.

Finalement, tous les éléments de l'Expression (5.22) sont définis. Le lecteur peut donc conclure ce calcul de la jacobienne, portant sur les éléments des séquences $\mathbf{s}^1, \dots, \mathbf{s}^{N_T}$. Soient $\varphi^1, \dots, \varphi^{N_T}$ leurs vecteurs de phases respectifs, et φ leur concaténation (sous la forme d'un unique vecteur). Nous pouvons également — avec un petit abus de notation — calculer la jacobienne selon ces phases [Hjørungnes, 2011] :

$$J_{f_p}(\varphi) = -2\Im (J_{f_p}(\mathbf{s}) \text{diag}(\mathbf{s})) . \quad (5.25)$$

5.2.2 Beampattern, directions d'intérêt

Par la diversité sur les formes d'onde qu'il propose, un radar MIMO co-localisé peut produire différentes formes de faisceaux, comme illustré à la Figure 5.4. Focalisé ou large, le choix du *beampattern* est évidemment déterminé par l'application, les propriétés de la cible, *etc.* En pratique, si le radar a la possibilité de modifier en temps réel les formes d'onde émises, on peut considérer une procédure de détection comme suit :

- Surveillance et détection. On commence par émettre large, et cela afin d'observer plusieurs directions simultanément, le but étant de détecter une cible potentielle. Cette étape est utile pour extraire les cibles lentes du *clutter*, ou lorsque une grande résolution en Doppler est requise [Guyvarch *et al.*, 2012].
- Focalisation et mesure. On focalise cette fois-ci en direction de la cible, voire même des cibles, si le radar le permet (*i.e.* si le radar possède un nombre suffisant d'émetteurs). Cela permet d'augmenter le ratio signal-à-bruit de la cible, et donc d'améliorer *a fortiori* les estimations sur les paramètres de la cible comme sa position, sa vitesse ou sa trajectoire [Li *et al.*, 2009].

Dans le but de procéder à ces différentes étapes, il est nécessaire de rechercher une méthodologie permettant de générer une forme de faisceau adaptée à chacune de ces configurations. Il a été expliqué dans [San Antonio *et al.*, 2007] que le *beampattern* se modélise en jouant sur la matrice des corrélations — à retard nul — des séquences élémentaires. Les auteurs de [Lipor *et al.*, 2014] ont, quant à eux, proposé une stratégie s'appuyant sur les transformées de Fourier. Ici, on se contentera d'un simple problème d'optimisation, fondé sur un gabarit définissant le profil recherché.

Soit Θ^g une discrétisation angulaire du champ de vision du radar. Celle-ci se doit d'être la plus large possible — tout en étant réalisable par le radar, tout de même —, car elle doit pouvoir décrire toutes les configurations possibles. Soit \mathbf{b} un vecteur réel de taille $|\Theta^g|$ décrivant les valeurs que doit prendre le faisceau souhaité.

On pose :

$$\begin{cases} \Theta^g := \{\theta_1^g, \dots, \theta_{|\Theta^g|}^g\} \\ \mathbf{b} := [b(\theta_1^g), \dots, b(\theta_{|\Theta^g|}^g)]^T. \end{cases} \quad (5.26)$$

Une façon naturelle d'atteindre cet objectif est de mesurer la distance qui sépare ce dernier du *beampattern* courant, ce qui s'écrit à l'aide de la fonction g_p suivante :

$$(\mathbf{T}^N)^{N_T} \rightarrow \mathbb{R}_+ \\ (\mathbf{s}^1, \dots, \mathbf{s}^{N_T}) \mapsto g_p(\mathbf{s}^1, \dots, \mathbf{s}^{N_T}) := \frac{1}{|\Theta^g|} \left(\sum_{i=1}^{|\Theta^g|} |b(\theta_i^g) - P(\theta_i^g)|^p \right)^{1/p}, \quad (5.27)$$

où P représente la densité de puissance comme définie à l'équation (5.12). Dans le cas $p = 2$, on dit que g_2 caractérise l'erreur quadratique moyenne envers le faisceau objectif \mathbf{b} .

Remarque 5.5. (*Ressemblance*) Afin de déterminer la ressemblance du *beampattern* courant à l'objectif, la fonction coût g_p se propose de mesurer la distance qui les sépare. Il est évidemment possible d'en considérer d'autres, à l'instar de celle qu'on a évoquée dans la section sur l'optimisation jointe (Section 4.2.2.1).

Un exemple de vecteur objectif sur le *beampattern* est proposé à la Figure 5.6. Remarquez que, d'une part, la discrétisation établie par l'ensemble Θ^g peut se permettre d'être relativement détaillée, contrairement au précédent ensemble Θ^f (voir la Remarque 5.4 à ce sujet). La justification est assez sommaire : plus il y a de points, plus la mesure de distance est précise. D'autre part, les deux ensembles angulaires ne sont pas nécessairement identiques. Mais ils sont en fait liés, et l'exemple le plus parlant est celui présenté sur cette même figure. En effet, il est inutile d'optimiser les corrélations en des directions dont nous voulons minimiser la densité de puissance. Nous avons plus à gagner à ce que ces directions d'intérêts soient concentrées autour du lobe principal énergétique.

5.2.3 Le problème d'optimisation

5.2.3.1 Avec filtrage adapté

Les deux paragraphes précédents ont suggéré des fonctions permettant respectivement d'évaluer les corrélations des séquences émises et de mesurer la ressemblance de la forme du faisceau à un objectif. Écrire le problème d'optimisation associé n'est après tout qu'une question de synthèse... Commençons donc par rassembler les pré-requis :

- Discrétisation angulaire Θ^g selon la directivité du radar ;
- *Beampattern* objectif \mathbf{b} , notamment déterminé par les besoins opérationnels ;
- Directions d'intérêts Θ^f , selon les deux points précédents ;
- Quelques considérations pratiques, comme le nombre de couleurs M (le nombre de séquences élémentaires différentes) en fonction du nombre d'émetteurs N_T , ou la longueur des séquences N .

Le problème d'optimisation que l'on propose de résoudre, permettant de générer des codes de phase pour le radar MIMO cohérent, est le suivant :

$$(P_1) \quad \begin{cases} \min_{(\mathbf{s}^1, \dots, \mathbf{s}^{N_T})} f_p(\mathbf{s}^1, \dots, \mathbf{s}^{N_T}) + \mu g_p(\mathbf{s}^1, \dots, \mathbf{s}^{N_T}) \\ \text{t.q. } \mathbf{s}^m \in \mathbf{T}^N, \quad \forall m \in \llbracket 1, N_T \rrbracket, \end{cases} \quad (5.28)$$

où $\mu \in \mathbb{R}_+$ est un facteur de pénalisation. Ce problème regroupe toutes les caractéristiques (et donc les défauts) des problèmes d'optimisation qu'on a pu rencontrer précédemment. Premièrement, il reste toujours difficile à résoudre (*cf.* la Section 2.1.5), car :

- Doté de nombreuses variables, si l'on considère la globalité des phases composant chacune des séquences ;
- Non séparable à notre connaissance, ce qui ne nous permet pas de passer outre la grande dimensionnalité ;
- Non convexe à cause de la contrainte de module constant, ce qui nous empêche de déterminer sereinement l'optimum global.

En comparant quelques méthodes d'optimisation (Section 2.2), nous avons pu conclure que les méthodes déterministes et stochastiques se valent plus ou moins pour ce problème, avec une certaine efficacité. Deuxièmement, le problème (P_1) est multi-objectif (Section 3.2.2). Ici, on a choisi la solution la plus simple, consistant à cumuler l'ensemble des fonctions objectifs. Mais il vaut mieux considérer d'autres méthodologies si on y greffe d'autres contraintes ou besoins, comme sur le spectre (Section 3.2). Troisièmement, il manque l'utilisation du filtrage désadapté... Et ceci est le sujet du paragraphe suivant.

5.2.3.2 Avec filtrage désadapté

Le passage à une optimisation avec des filtres désadaptés se fait assez naturellement : il suffit en effet de les incorporer dans les mesures de corrélation. Ainsi, nous ne nous y étendrons pas ici, puisque cela fut de surcroît l'objet du chapitre précédent.

Pourtant, par souci de complétude, nous allons établir quelques (re-)définitions. Comme leurs homologues adaptés, nous décidons d'attribuer un filtre désadapté par direction d'intérêt, comme explicité ci-après.

$$\begin{cases} \Theta^f := \{\theta_1^f, \dots, \theta_{|\Theta_f|}^f\}, \\ S(\Theta^f) := \{\mathbf{s}(\theta_1^f), \dots, \mathbf{s}(\theta_{|\Theta_f|}^f)\}, \\ Q(\Theta^f) := \{\mathbf{q}(\theta_1^f), \dots, \mathbf{q}(\theta_{|\Theta_f|}^f)\}. \end{cases} \quad (5.29)$$

On supposera par simplicité que chaque filtre désadapté est de même longueur $K := N + 2l$, avec $l \in \llbracket 0, N \rrbracket$. Par ailleurs, seule la fonction auxiliaire v_p se doit d'être modifiée afin de prendre en considération les corrélations avec les filtres désadaptés :

$$v_p(S(\Theta_f), Q(\Theta_f)) := \left(\sum_{\theta_i^f, \theta_j^f \in \Theta_f} \left| \sum_{n=1}^{N_R} e^{j(\mathbf{x}_{R,n}^T (\mathbf{k}(\theta_i^f) - \mathbf{k}(\theta_j^f)))} \right|^p \left(h_p(\mathbf{s}(\theta_i^f), \mathbf{q}(\theta_j^f)) \right)^p \right)^{1/p}. \quad (5.30)$$

La fonction objectif f_p devient, avec un abus de notation sur les variables d'entrée :

$$\begin{aligned} (\mathbf{T}^N)^{N_T} \times (\mathbb{C}^K)^{|\Theta_f|} &\rightarrow \mathbb{R}_+ \\ (S, Q(\Theta_f)) &\mapsto f_p(S, Q(\Theta_f)) := v_p(u(S), Q(\Theta_f)). \end{aligned} \quad (5.31)$$

Le problème d'optimisation associé s'écrit alors :

$$(P_2) \begin{cases} \min_{(S, Q(\Theta_f))} & f_p(S, Q(\Theta_f)) + \mu g_p(S) \\ \text{t.q.} & \mathbf{s}^m \in \mathbf{T}^N, & \forall m \in \llbracket 1, N_T \rrbracket, \\ & \mathbf{q}(\theta_n^f) \in \mathbb{C}^K, & \forall n \in \llbracket 1, |\Theta_f| \rrbracket, \\ & 10^{\alpha/10} (\mathbf{s}(\theta_n^f))^H \mathbf{s}(\theta_n^f) \geq (\mathbf{q}(\theta_n^f))^H \mathbf{q}(\theta_n^f), & \forall n \in \llbracket 1, |\Theta_f| \rrbracket, \\ & (\mathbf{s}(\theta_n^f))^H \mathbf{q}(\theta_n^f) = (\mathbf{s}(\theta_n^f))^H \mathbf{s}(\theta_n^f), & \forall n \in \llbracket 1, |\Theta_f| \rrbracket. \end{cases} \quad (5.32)$$

Achevons ce paragraphe avec quelques observations sur ce problème.

- Les variables d'optimisation sont toujours les séquences élémentaires, auxquelles nous avons ajouté des filtres désadaptés. Ces deux ensembles ne sont pas *nécessairement* du même cardinal (et n'oublions pas que les filtres désadaptés peuvent être plus longs que les séquences).
- Le *beam pattern* n'est guère influencé par les filtres désadaptés.
- Les deux dernières lignes du problème (P_2) définissent des contraintes sur la perte en gain de traitement (LPG). Évidemment, cette perte se mesure selon les signaux émis dans les directions d'intérêt...
- Le problème partiel recherchant à optimiser les filtres désadaptés seuls est convexe.
- Il est possible d'appliquer des algorithmes d'optimisation jointe.

5.2.4 Derniers résultats

Dans ce paragraphe, on se propose d'illustrer les effets d'une optimisation avec un petit exemple. Il consiste à rechercher des codes de phase pour le radar MIMO cohérent, et à comparer leurs performances par rapport à une initialisation aléatoire. Les résultats sont présentés sous la forme d'un catalogue de figures, se trouvant aux pages 164–169.

Pour cette simulation, nous avons considéré les paramètres suivants :

- L'antenne, linéaire, est composée de $N_T = 4$ émetteurs, et d'autant de récepteurs. La distance minimale entre deux émetteurs est de $\lambda/2$.
- Les séquences élémentaires sont de longueur $N = 255$.
- Quant aux filtres désadaptés, ils sont choisis trois fois plus long. On accepte une perte de 1 dB.
- Quatre directions d'intérêt (dans la composante élévation) ont été considérées. Parmi elles, on ne présentera seulement ci-dessous des résultats pour la direction $\theta = 0$ (les résultats pour les autres directions sont relativement semblables).
- Comme à l'accoutumée, la cible est placée en $(\tau_c, \theta_c) = (0, 0)$.

La « résolution » du problème (5.28) a été effectuée à l'aide d'une descente de gradient. Comme expliqué dans le Chapitre 2, cette méthode semble proposer des résultats viables avec un coût calculatoire raisonnable.

Initialisation aléatoire Passons maintenant à une description sommaire des figures qui suivent. On débute en tirant un ensemble de $N_T = 4$ séquences, de longueur $N = 255$, de manière aléatoire. Leur fonction d'ambiguïté MIMO distance/élévation est représentée Figure 5.7, sans le traitement à la réception pour commencer. On observe le caractère spécifique des codes CDMA (Section 1.2.4), à savoir des lobes secondaires assez bien répartis, mais relativement hauts — que nous devons donc réduire! — sur l'ensemble du domaine d'étude. Notons d'ailleurs que le lobe secondaire le plus élevé sur la coupe $\theta = 0$ s'élève ici aux alentours des -18.8 dB. On applique ensuite un traitement à la réception (Figure 5.8) pour retrouver l'amincissement du lobe principal.

Séquences pseudo-orthogonales Ensuite, on étudie l'influence de l'orthogonalité des séquences élémentaires. Cette hypothèse est souvent de rigueur dans la littérature (lorsque sont évoquées les performances d'un radar MIMO), bien que nous avons vu qu'elle permettait seulement d'obtenir une émission large. Pour cela, on considère des séquences pseudo-orthogonales, obtenues à l'aide de la procédure décrite à la Section 3.5. La Figure 5.14 représente en outre l'autocorrélation de la première séquence élémentaire (les autres séquences d'autocorrélation sont du même acabit), tandis que la fonction d'ambiguïté MIMO induite est à la Figure 5.9. Qualitativement parlant, cette dernière ressemble beaucoup à la précédente; la coupe définie par $\theta = 0$ offre même de surcroît des lobes secondaires similaires. Décevant, mais attendu. On observe en effet sur cette coupe la sommation de toutes les autocorrélations et inter-corrélations des séquences élémentaires. Or, dans le cas des corrélations aperiodiques, nous savons que l'orthogonalité « parfaite » est inaccessible avec des

codes de phase : les lobes secondaires ne sont pas nuls et s'accumulent lors de la chaîne de traitement, d'où le profil observé.

Optimisation proposée En commençant avec la même initialisation, on applique la procédure proposée dans ce chapitre, basée sur la résolution du problème d'optimisation (5.32). Pour cet exemple, le gabarit employé vise une émission large. La fonction d'ambiguïté obtenue est représentée à la Figure 5.10. Cette fois, une différence se fait sentir au niveau de la direction d'intérêt $\theta = 0$, où l'on peut mesurer le lobe plus élevé à -29.8 dB, soit un gain non négligeable d'une dizaine de dB environ (par rapport à l'initialisation). Les autres lobes secondaires semblent être — approximativement — conservés. Ceci nous encourage sur le bénéfice apporté par une optimisation, bien qu'il faudrait d'autres simulations pour appuyer nos dires... On se propose dans un second temps, à partir des séquences élémentaires que nous venons d'acquérir, d'appliquer une batterie de filtres désadaptés, chacun associés aux mêmes directions d'intérêt. Les filtres sont choisis de longueur $3N$, tandis qu'on s'autorise une perte due à la désadaptation de 1 dB. Le résultat obtenu est reproduit Figure 5.11. C'est encore au niveau de la coupe $\theta = 0$ que la différence se fait sentir : les lobes secondaires sont constants et de l'ordre des -43.6 dB.

Quelques observations Concluons cet exemple avec quelques courbes/coupes. Premièrement, nous avons observé l'essentiel des distinctions entre les multiples optimisations au niveau des directions d'intérêt. Ainsi, la coupe $\theta = 0$ est reprise à la Figure 5.12; y est ajouté en sus une coupe dans une direction non optimisée ($\theta = -20^\circ$, Figure 5.13), où le niveau des lobes n'a malheureusement pas été amélioré. Deuxièmement, nous avons vu avec la Figure 5.9 que la *pseudo*-orthogonalité des séquences élémentaires ne permettait pas d'offrir de bonnes performances avec notre radar. Un examen des corrélations élémentaires des séquences proposées va dans ce sens : l'orthogonalité n'est en effet pas vraiment manifeste. Néanmoins, le diagramme de notre antenne reste relativement plat, comme l'illustre la Figure 5.15. Troisièmement, nous essayons de caractériser les performances atteignables de notre radar MIMO, toujours sur la coupe $\theta = 0$, en faisant varier la longueur des séquences N ainsi que le nombre d'antennes N_T (toujours séparées de $\lambda/2$). Des résultats sont proposés Figures 5.16 et 5.17. Nous pouvons observer que :

- Pour une longueur de séquence donnée, l'augmentation du nombre d'antennes ne réduit en rien les performances sur cette coupe. En effet, ce n'est pas la cardinalité des séquences élémentaires qui entre en jeu, mais plutôt la multiplicité des combinaisons linéaires selon les directions d'émission. En fait, cette augmentation améliore même potentiellement ces performances, du fait du traitement à la réception.
- Doubler la longueur de séquence permet, semble-t-il, de gagner 2 dB, et non 3... On retrouve ce comportement approximatif avec les filtres désadaptés. Coïncidence ?

Une conclusion sur l'ensemble de ces résultats sera effectuée dans le paragraphe suivant.

5.3 Conclusion

Dans ce chapitre, nous nous sommes intéressés à l'optimisation de formes d'onde — des codes de phase, pour être exact — mais dans le cadre de l'application principale de notre thèse, à savoir le radar MIMO cohérent.

La méthodologie proposée est relativement simple. Elle se base, encore une fois, sur la recherche d'un ensemble de séquences pseudo-orthogonales (MIMO oblige). Cependant, cette propriété d'orthogonalité ne concerne pas directement les séquences élémentaires. En effet, la particularité du radar MIMO cohérent réside en son émission. Plus précisément, le signal émis diffère potentiellement selon les directions. Cela a notamment deux conséquences. D'une part, le traitement s'établit à l'aide de filtres correspondant à chacune de ces directions. Ainsi, la procédure d'optimisation doit se consacrer à améliorer l'orthogonalité de ces signaux émis (et non celle des signaux élémentaires). D'autre part, le diagramme d'antenne induit est

donc variable. Il n'est donc pas nécessairement prédéterminé, comme sur un réseau phasé classique. De ce fait, il peut être intéressant de le contrôler, *via* les formes d'onde émises, afin de coller au mieux aux besoins opérationnels.

Sous ces considérations, nous avons suggéré un petit problème d'optimisation. Les résultats, obtenus relativement rapidement sur simulation à l'aide d'une descente de gradient, soulignent l'intérêt d'une optimisation. Ils affichent une amélioration notable, mais seulement au niveau des directions d'intérêt ; en outre, l'emploi d'un filtre désadapté permet d'améliorer quelque peu les performances.

Mais ce n'est que le début. De nombreuses tâches restent à accomplir, notamment en ce qui concerne l'ajout de contraintes, pour mieux cerner la réalité. Citons d'ailleurs quelques objectifs à court terme :

- Optimiser des corrélations avec des *beampatterns* plus exotiques (*e.g.* focalisation dans deux directions simultanément) [Wang *et al.*, 2012].
- Optimiser de manière jointe les séquences élémentaires et les filtres désadaptés, comme ce que nous avons proposé [Tan *et al.*, 2017a].
- Étendre cette méthodologie dans le cadre des modulations non linéaires de fréquence (*Non Linear Frequency Modulation*, NLFM) [Gao *et al.*, 2016]. Cela nécessite cependant de trouver une méthodologie pour faire abstraction des effets de couplage...

Et quelques objectifs à long terme :

- Gérer les antennes non-linéaires (à l'instar des antennes *co-primes* [Pribic, 2015]), et en particulier les antennes à deux dimensions.
- Prendre en compte le diagramme de veille [Briheche *et al.*, 2018] ou la position des capteurs (en tant que variables) dans l'optimisation [Mendes Ruiz *et al.*, 2015].
- Ajouter des contraintes matérielles (spectre des signaux élémentaires (Section 3.2), couplage entre les antennes par exemple [Yang *et al.*, 2011]).
- Prendre en considération la composante angulaire azimut, ainsi que le Doppler.

Mais, toute cette optimisation est-elle réellement viable ? Le dernier point mentionné est en effet assez critique, pour deux raisons. *Primo*, l'optimisation proposée a été effective dans l'espace distance/élévation. Cependant, la fonction d'ambiguïté présente un volume constant [Chen *et al.*, 2008]. Cette idée exprime la notion de compromis, souvent rencontrée dans le domaine radar. Si bien qu'une amélioration dans cet espace a nécessairement dégradé les performances autre part. *Secundo*, la prise en compte d'autres composantes augmentera significativement le nombre de filtres à considérer. Cela peut être délicat à gérer pour un radar, notamment pour une utilisation *online*... Sous ces deux arguments, difficile d'être rassurés par l'emploi de codes de phase pour un radar MIMO. Si l'applicatif le permet, il serait fort nécessaire de relaxer les contraintes (en distance par exemple)...

Ce chapitre a été présenté lors de deux congrès [Tan et al., 2016b, Tan et al., 2017a].

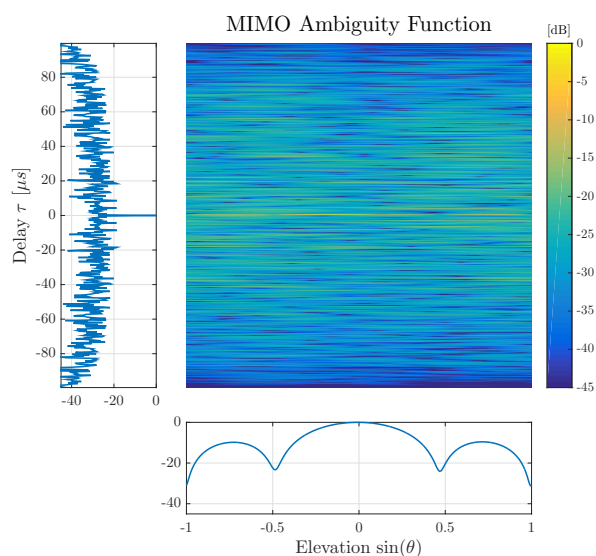


FIGURE 5.7 – Fonction d’ambiguïté MIMO, séquences élémentaires aléatoires (1)
Fonction d’ambiguïté sans le traitement à la réception.
 Antenne composée de $N_T = 4$ émetteurs et $N_R = 4$ récepteurs.
 Séquences tirées aléatoirement de longueur $N = 255$.

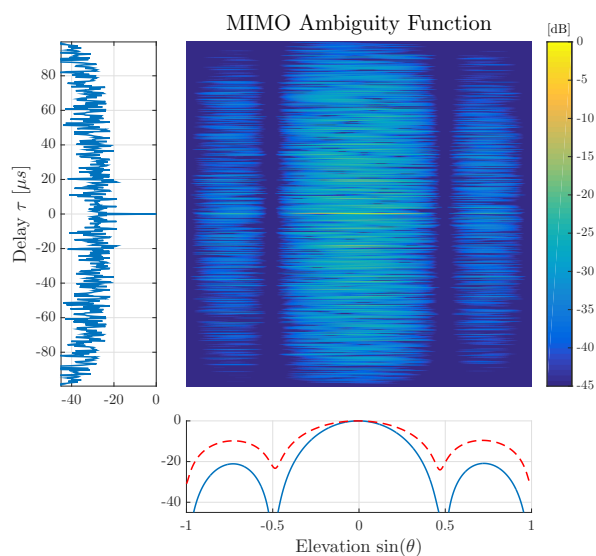


FIGURE 5.8 – Fonction d’ambiguïté MIMO, séquences élémentaires aléatoires (2)
Fonction d’ambiguïté avec ajout du traitement à la réception.
 Pointillés rouges : coupe à retard nul, provenant de la Figure 5.7.
 Antenne composée de $N_T = 4$ émetteurs et $N_R = 4$ récepteurs.
 Séquences tirées aléatoirement de longueur $N = 255$.

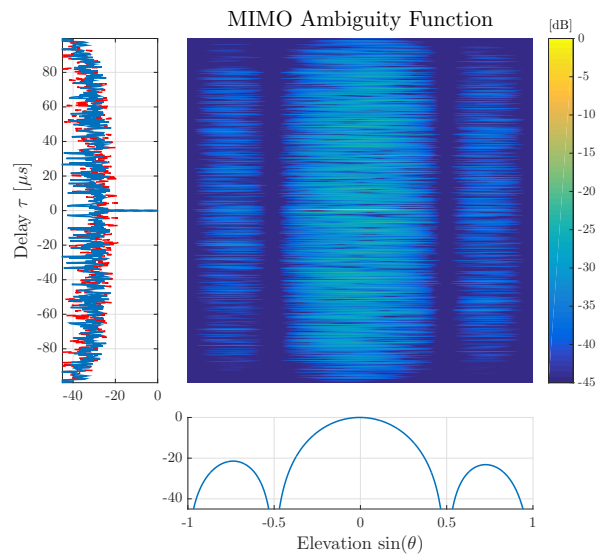


FIGURE 5.9 – Fonction d’ambiguïté MIMO, séquences élémentaires pseudo-orthogonales
Séquences obtenues à l’aide de la procédure décrite à la Section 3.5.

Pointillés rouges : coupe à $\theta = 0$ de l’initialisation aléatoire (Figure 5.7), obtenue sans FFC réception.

Antenne composée de $N_T = 4$ émetteurs et $N_R = 4$ récepteurs.

Séquences de longueur $N = 255$.

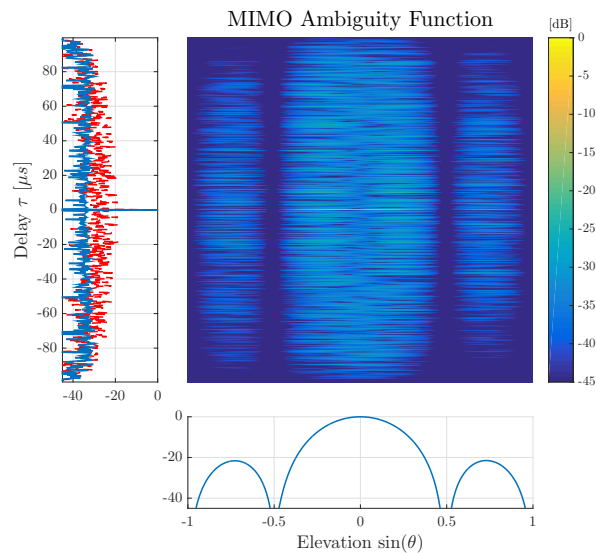


FIGURE 5.10 – Fonction d’ambiguïté MIMO, séquences optimisées
Séquences obtenues après résolution du problème (5.28).

Pointillés rouges : coupe à $\theta = 0$ de l’initialisation aléatoire (Figure 5.7).

Antenne composée de $N_T = 4$ émetteurs et $N_R = 4$ récepteurs.

Séquences de longueur $N = 255$.

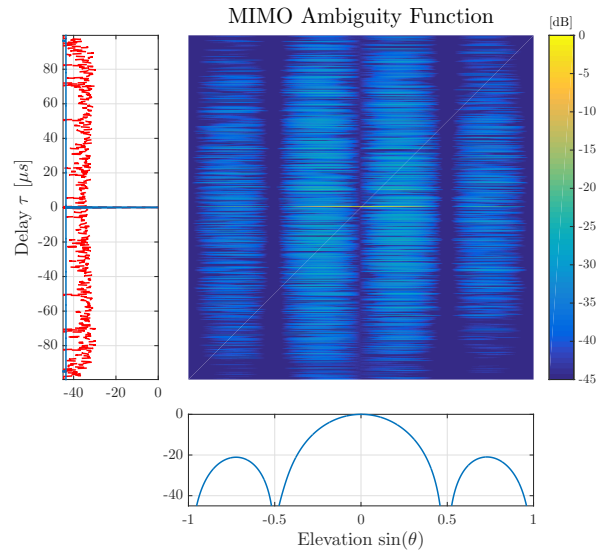


FIGURE 5.11 – Fonction d’ambiguïté MIMO, séquences optimisées, filtrage désadapté
Séquences de la Figure 5.10, mais avec application d’un filtrage désadapté
Pointillés rouges : coupe à $\theta = 0$ des séquences optimisées (Figure 5.10).
Antenne composée de $N_T = 4$ émetteurs et $N_R = 4$ récepteurs.
Séquences de longueur $N = 255$, filtres désadaptés de longueur $3N$.
Perte autorisée et effective de 1 dB.

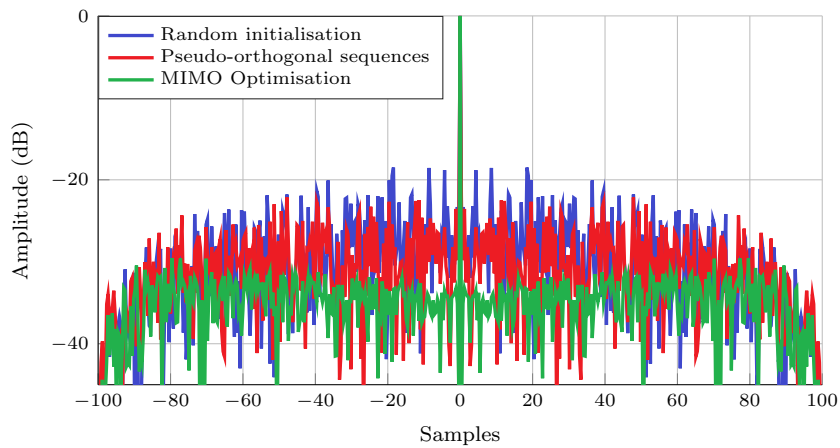


FIGURE 5.12 – Quelques coupes sur les fonctions d’ambiguïtés (1)
Coupe dans la direction d’intérêt $\theta = 0$.

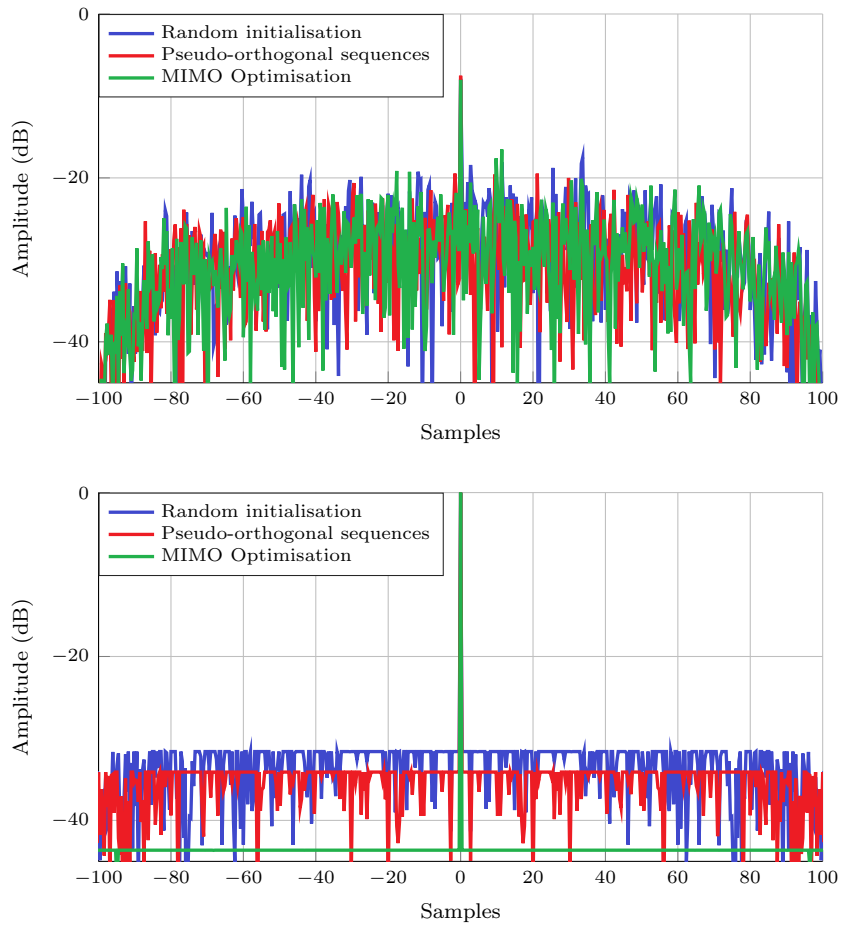


FIGURE 5.13 – Quelques coupes sur les fonctions d’ambiguïtés (2)

En haut : coupe dans la direction $\theta = -20^\circ$.

En bas : coupe dans la direction d’intérêt $\theta = 0$, avec application d’un filtre désadapté.

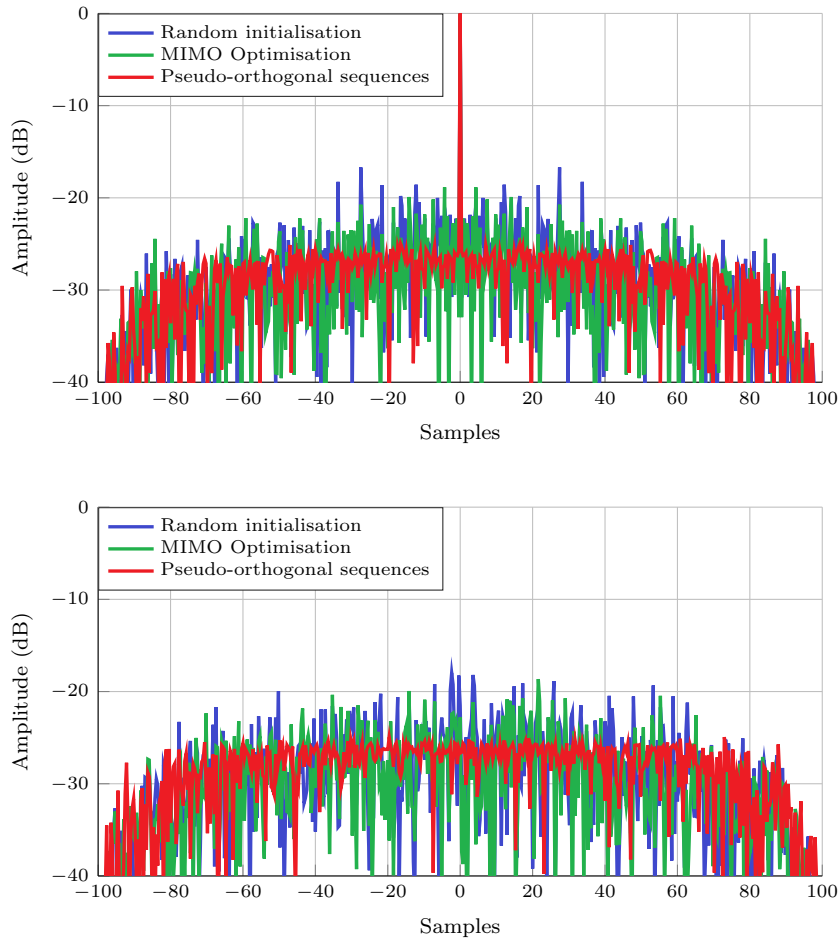


FIGURE 5.14 – Orthogonalité des séquences élémentaires
En haut : autocorrélation de la première séquence élémentaire.
En bas : inter-corrélation entre la première et la deuxième séquence élémentaire.

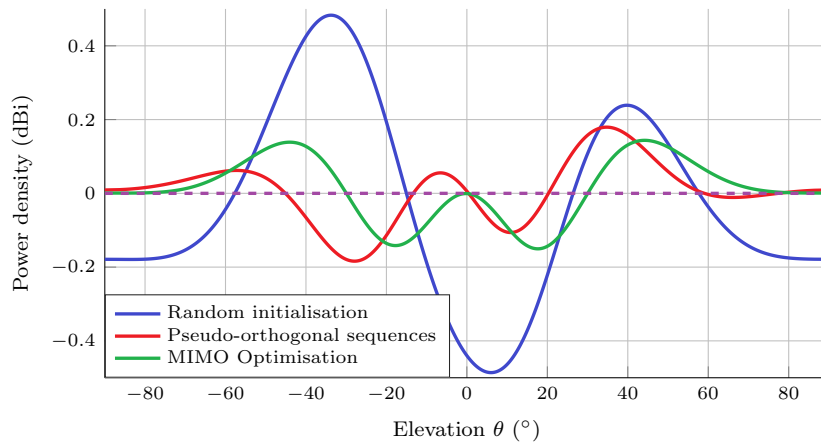


FIGURE 5.15 – Diagramme obtenu après optimisation
*L'objectif était de considérer un gabarit plat (en pointillés).
 Niveau relatif représenté (en dBi, d'où la valeur nulle).*

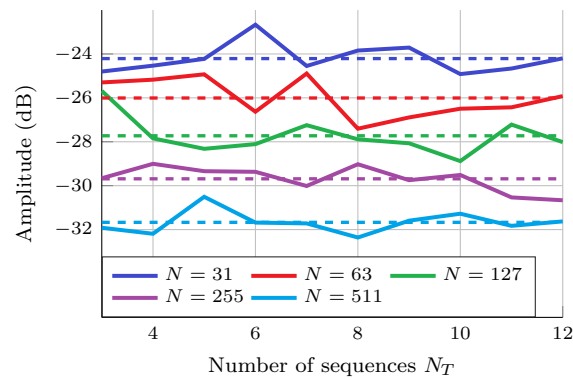


FIGURE 5.16 – Lobes secondaires selon le nombre de séquences
*Niveau des lobes secondaires sur la direction d'intérêt $\theta = 0$.
 La moyenne (sur les différentes longueurs) est représentée en pointillés.*

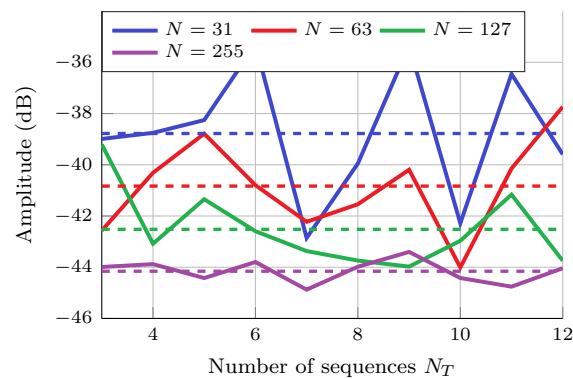


FIGURE 5.17 – Lobes secondaires selon le nombre de séquences, avec filtrage désadapté
Niveau des lobes secondaires sur la direction d'intérêt $\theta = 0$.

*La moyenne est représentée en pointillés.
 Les exemples choisis ne sont pas très réguliers...*

Conclusion

Le début de la fin

« Méthodologies de conception de formes d'onde pour radars sols — Application au cas du radar MIMO cohérent ». Tel est l'intitulé de ce manuscrit qui expose, avec une certaine largeur, le contexte et l'objectif des quelques pages précédentes. Plus concrètement, l'œuvre de cette thèse a consisté en la modélisation d'un problème d'optimisation (la sus-mentionnée « méthodologie »), dont la résolution offre des codes de phase (les « formes d'onde ») adéquats pour le radar MIMO cohérent.

Pour cela, une approche incrémentale a été suivie, reflétée par les différents chapitres. Nous sommes partis d'un problème d'optimisation élémentaire, permettant de générer des codes de phase acceptables pour un radar monostatique, que nous avons...

- étendu, pour un ensemble de séquences (MIMO oblige) ;
- spécifié, en ajoutant des contraintes opérationnelles (à l'instar de la gestion des bandes spectrales) ;
- généralisé, en prenant en compte le traitement à l'aide de filtres désadaptés ;
- adapté pour le radar MIMO cohérent.

De ces travaux, quelques observations et suggestions ont été effectuées. Elles sont résumées dans les lignes suivantes.

De l'optimisation de la corrélation... Le problème d'optimisation élémentaire consiste à minimiser les lobes secondaires d'une séquence de corrélation. Ce problème, bien connu de la littérature, est plutôt difficile à résoudre, que ce soit analytiquement ou numériquement. Il est en effet non convexe, non séparable, et doté de nombreuses variables. Ainsi, on s'est proposé de comparer quelques méthodes d'optimisation avec, en particulier, la descente de gradient (déterministe) et l'algorithme CMA-ES (stochastique). Elles se distinguent par leur commodité d'utilisation, mais les résultats se valent en terme de performances brutes.

... aux contraintes opérationnelles... À partir d'un problème d'optimisation, l'ajout de contraintes est relativement aisé. Il suffit en effet de modéliser une fonction définissant le besoin opérationnel... et le tour est joué. Cela nous a notamment permis d'établir une contrainte sur le spectre, et d'observer les effets des phénomènes d'éclipse et de quantification. Les problèmes induits sont beaucoup plus complexes à résoudre, et les résultats s'en ressentent, avec une petite dégradation des performances. Une belle illustration du compromis et de la flexibilité de la méthodologie.

... en passant par le filtre désadapté... Le filtrage désadapté a spontanément l'image d'un traitement avantageux, grâce à l'ajout de degrés de liberté (longueur du filtre variable) et sa non-soumission à la contrainte de module constant. Dans le cas mono-séquence, la différence est souvent assez flagrante avec la compression d'impulsion usuelle. En ce sens, nous avons suggéré une procédure d'optimisation jointe (*i.e.* une recherche simultanée de la séquence et de son filtre) qui offre des résultats intéressants, comparés à ceux de la littérature.

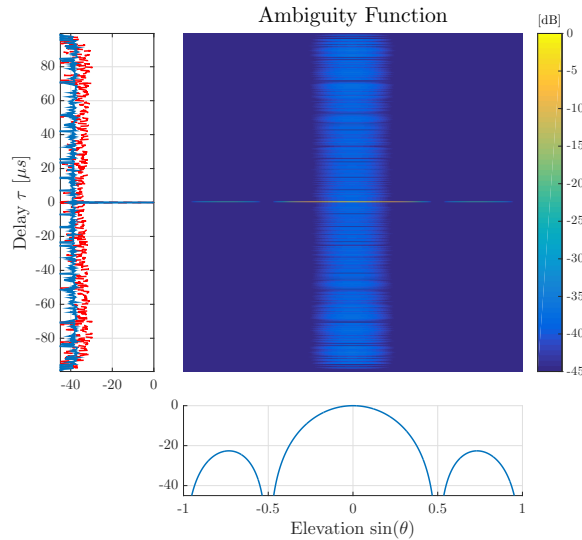


FIGURE II.1 – Fonction d’ambiguïté d’un réseau phasé classique
Séquence de longueur $N = 255$ optimisée à l’aide d’une descente de gradient.
Pointillés rouges : coupe à $\theta = 0$ de la séquence optimisée (Figure 5.10).
Antenne composée de $N_T = 4$ émetteurs et $N_R = 4$ récepteurs.

Dans le cas multi-séquence, cette différence est beaucoup plus ténue. Il semblerait que les degrés de liberté supplémentaires ne permettent pas de compenser la multiplication des contraintes (les séquences d’inter-corrélations). D’ailleurs, une borne sur le niveau maximal des lobes secondaires d’un ensemble de séquences, associé à des filtres désadaptés, a été proposée, appuyant cette hypothèse.

... aux codes de phase pour le radar MIMO cohérent Fort de cette expérience, nous avons finalement pu définir un problème d’optimisation permettant de générer des formes d’onde adéquats pour le radar MIMO cohérent. Comme stipulé maintes fois, le principe d’orthogonalité porte sur les séquences émises par l’intégralité du radar, et non sur les séquences élémentaires. La flexibilité de la modélisation facilite l’ajout de contraintes, notamment sur la forme du faisceau ou l’emploi de filtres désadaptés. Quelques simulations ont permis de souligner l’intérêt d’une optimisation. Mais ce n’est que le début...

La fin du début ?

En effet, il y a encore tant de choses à effectuer ! De nombreuses pistes ont été laissées en suspens, dont les principales sont énumérées ici.

- Ajouter d’autres méthodes d’optimisation à notre comparatif, bien que nous ne nous attendons pas à une révolution. Nous pensons entre autres aux méthodes dites de points intérieurs [Wright, 2005], de raffinements [Aldayel *et al.*, 2016] ou celles adaptées pour l’optimisation multi-objectif [Branke *et al.*, 2008]. Continuer en-sus les travaux sur le problème quantifié.
- Étudier le problème sous un aspect plus théorique, *i.e.* rechercher des propriétés sur la fonction de corrélation et son minimum global (ce que nous avons entrepris avec le calcul des bornes). On pourrait par exemple envisager d’analyser le problème en passant par la variété relativement connue \mathbf{T}^N (l’hypertore de dimension N , qui équivaut ici aux séquences de module constant).

- Accélérer l'algorithme d'optimisation jointe (séquence–filtre), et l'étendre pour un ensemble de séquences, bien que le gain ne sera probablement pas admirable.

En ce qui concerne l'optimisation des codes de phase pour le radar MIMO cohérent, des travaux sont en cours sur l'application de la méthodologie pour des antennes à deux dimensions, tout en prenant en compte des *beam patterns* plus exotiques (*e.g.* focalisation dans deux directions simultanément). Parallèlement, une interrogation se pose quant à la commodité des codes de phase et de son optimisation pour le radar MIMO cohérent :

- L'optimisation proposée a été réalisée dans l'espace distance/élévation, sans prendre en compte le Doppler. Il faudra également considérer la composante azimut pour des antennes non linéaires. Dans leurs domaines, les performances se sont probablement dégradées, à cause de la propriété de conservation de l'énergie.
- La prise en compte de ces différentes composantes augmentera significativement le nombre de filtres à considérer, et le nombre de calculs à effectuer. Cela pourra être délicat à gérer pour un radar, notamment pour une utilisation en temps réel.
- L'utilisation d'un radar MIMO cohérent est nécessairement plus coûteuse que celle de son lointain cousin, le réseau phasé classique. Pour ce dernier, chaque émetteur émet la même forme d'onde. Effectuons une comparaison hasardeuse, en optimisant cette dernière. La fonction d'ambiguïté obtenue est représentée en haut de la page précédente, et semble favorable pour le réseau phasé classique! Il faut néanmoins prendre cette comparaison avec précaution : les diagrammes d'antennes diffèrent, la directivité à l'émission est absente, le réseau phasé nécessitant de balayer l'ensemble de l'espace d'étude...

L'utilisation d'un ensemble de codes fréquentiels est peut-être une solution à creuser. Encore faut-il pouvoir faire abstraction des effets de couplage.



FIGURE II.2 – Faites l'amour, pas la guerre

Annexe A

Bornes sur les corrélations apériodiques

Cette annexe pose quelques démonstrations sur les propriétés proposées à la Section 3.6, concernant les bornes sur les corrélations apériodiques. Deux points ont été évoqués (et sont donc à démontrer) : la généralisation de la borne de Levenshtein pour les codes polyphases, et une petite amélioration de cette dernière.

A.1 Généralisation de la borne de Levenshtein

Afin de généraliser la borne de Levenshtein, nous sommes passés par une borne supérieure et une borne inférieure sur la norme d'une certaine matrice $\tilde{\mathbf{R}}$ (définie dans la section susmentionnée), que nous rappelons ci-dessous :

Lemme 3.1. (*Borne supérieure*) La norme de Frobenius de la matrice $\tilde{\mathbf{R}}$ admet la borne supérieure suivante :

$$\|\tilde{\mathbf{R}}\|_F^2 \leq M^2 \theta_{\max}^2 + M (1 - \theta_{\max}^2) \sum_{k=1}^{2N-1} w_k^2, \quad (3.58)$$

Lemme 3.2. (*Borne inférieure*) ... et la borne inférieure qui suit :

$$\|\tilde{\mathbf{R}}\|_F^2 \geq \frac{M^2}{N^2} \left(N - \sum_{s,t=1}^{2N-1} l_{s,t,N} w_s w_t \right). \quad (3.59)$$

avec $l_{s,t,N} = \min(|t-s|, 2N-1-|t-s|)$.

A partir de ces deux lemmes, il est relativement aisé — ce n'est qu'une question de réécriture — d'en déduire une borne sur θ_{\max}^2 , le lobe secondaire le plus haut, mise sous la forme d'un théorème :

Théorème 3.1. Soit \mathbf{w} un vecteur vérifiant la condition de pondération (3.47). Pour tout ensemble de M séquences unimodulaires de même longueur N , une borne inférieure sur le lobe secondaire le plus élevé s'écrit :

$$\theta_{\max}^2 \geq \frac{1}{N^2} \left[N - \frac{Q_{2N-1} \left(\mathbf{w}, \frac{N(N-1)}{M} \right)}{1 - \frac{1}{M} \sum_{k=1}^{2N-1} w_k^2} \right], \quad (3.60)$$

$$\text{avec } Q_{2N-1}(\mathbf{w}, a) = a \sum_{k=1}^{2N-1} w_k^2 + \sum_{s,t=1}^{2N-1} l_{s,t,N} w_s w_t$$

$$\text{et } l_{s,t,N} = \min(|s-t|, 2N-1-|s-t|).$$

Ainsi, il ne nous reste qu'à démontrer l'encadrement de la norme de Frobenius de $\tilde{\mathbf{R}}$. Pour la borne supérieure, le résultat est similaire au Lemme 1 de [Levenshtein, 1999]. Pour la borne inférieure, on commence par écrire :

$$\begin{aligned} \|\tilde{\mathbf{R}}\|_F^2 &= \|\tilde{\mathbf{S}}\tilde{\mathbf{S}}^H\|_F^2 = \|\tilde{\mathbf{S}}^H\tilde{\mathbf{S}}\|_F^2 \\ &= \sum_{k,k'=1}^{2N-1} \left| \sum_{m=1}^M \sum_{i=1}^{2N-1} S_{i,k}^m (S_{i,k'}^m)^* w_i \right|^2 \\ &\geq \sum_{k=1}^{2N-1} \left(\sum_{m=1}^M \sum_{i=1}^{2N-1} |S_{i,k}^m|^2 w_i \right)^2 \end{aligned} \quad (\text{A.1})$$

où la dernière inégalité est obtenue en retirant plusieurs termes (ou, autrement dit, en ne conservant que les termes tels que $k = k'$) dans la première somme. Le développement du côté droit s'établit par un argument de construction. En effet, la structure circulante des différentes matrices \mathbf{S}^m est similaire par construction, ce qui nous offre :

$$|S_{i,k}^l|^2 = |S_{i,k}^m|^2 \quad \text{pour tout } i, k \in \llbracket 1, 2N-1 \rrbracket, \text{ pour tout } l, m \in \llbracket 1, M \rrbracket. \quad (\text{A.2})$$

De plus, selon la position des éléments, la propriété d'unimodularité nous donne :

$$|S_{i,k}^m|^2 = \begin{cases} 1/N & \text{si on tombe sur un élément de la séquence } \mathbf{s}^m, \\ 0 & \text{sinon.} \end{cases} \quad (\text{A.3})$$

De cela, la précédente inégalité (A.1) devient :

$$\|\tilde{\mathbf{R}}\|_F^2 \geq M^2 \sum_{k=1}^{2N-1} \left(\sum_{i=1}^{2N-1} |S_{i,k}^m|^2 w_i \right)^2. \quad (\text{A.4})$$

À ce stade, les démonstrations de Levenshtein [Levenshtein, 1999] et de Welch [Welch, 1974] emploient l'inégalité de Cauchy-Schwarz. Cependant, nous n'allons pas faire de même : nous allons plutôt exploiter de nouveau la structure des matrices \mathbf{S}^m , en remarquant que chacune de leurs colonnes contient exactement N éléments non nuls, ce qui implique :

$$\sum_{i=1}^{2N-1} |S_{i,k}^m|^2 w_i = \begin{cases} \sum_{i=1}^k w_i/N + \sum_{i=N+k}^{2N-1} w_i/N & \text{si } k \leq N-1, \\ \sum_{i=k-N+1}^k w_i/N & \text{si } k \geq N. \end{cases} \quad (\text{A.5})$$

Incorporer ce résultat dans (A.4) nous permet de conclure :

$$\begin{aligned} \|\tilde{\mathbf{R}}\|_F^2 &\geq \frac{M^2}{N^2} \left[\sum_{k=1}^{N-1} \left(\sum_{i=1}^k w_i + \sum_{i=N+k}^{2N-1} w_i \right)^2 + \sum_{k=N}^{2N-1} \left(\sum_{i=k-N+1}^k w_i \right)^2 \right] \\ &= \frac{M^2}{N^2} \left[N - \sum_{s,t=1}^{2N-1} l_{s,t,N} w_s w_t \right], \end{aligned} \quad (\text{A.6})$$

avec $l_{s,t,N} = \min(|t-s|, 2N-1-|t-s|)$. La dernière égalité est démontrée ci-dessous !

Étape intermédiaire. Montrons que :

$$\sum_{k=1}^{N-1} \left(\sum_{i=1}^k w_i + \sum_{i=N+k}^{2N-1} w_i \right)^2 + \sum_{k=N}^{2N-1} \left(\sum_{i=k-N+1}^k w_i \right)^2 = \sum_{s,t=1}^{2N-1} (N - l_{s,t,N}) w_s w_t. \quad (\text{A.7})$$

En remarquant que :

$$\sum_{i=1}^k w_i = \sum_{i=\kappa}^k w_i \quad \text{pour } k \in \llbracket 1, N-1 \rrbracket, \quad (\text{A.8})$$

avec $\kappa = \max(1, k - N + 1)$, on peut développer le terme de gauche de (A.7) comme suit :

$$\begin{aligned} & \sum_{k=1}^{N-1} \left(\sum_{i=1}^k w_i + \sum_{i=N+k}^{2N-1} w_i \right)^2 + \sum_{k=N}^{2N-1} \left(\sum_{i=k-N+1}^k w_i \right)^2 \\ &= 2 \sum_{k=1}^{N-1} \sum_{s=1}^k \sum_{t=N+k}^{2N-1} w_s w_t + \sum_{k=1}^{N-1} \left(\sum_{i=N+k}^{2N-1} w_i \right)^2 + \sum_{k=1}^{2N-1} \left(\sum_{i=\max(1, k-N+1)}^k w_i \right)^2. \end{aligned} \quad (\text{A.9})$$

Soit $\mathbf{M}_{s_1, t_1}^{s_2, t_2}$ une matrice carrée de taille $2N - 1$. Elle est basée sur la matrice nulle, sauf au niveau de la sous-matrice située entre les lignes s_1 et s_2 et les colonnes t_1 et t_2 , qui ne contient que des éléments égaux à 1 :

$$\mathbf{M}_{s_1, t_1}^{s_2, t_2} = \begin{bmatrix} 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{array}{l} \leftarrow s_1 \\ \\ \leftarrow s_2 \\ \\ \end{array} \quad (\text{A.10})$$

$$\begin{array}{cc} \uparrow & \uparrow \\ t_1 & t_2 \end{array}$$

Avec une telle matrice, pour tout vecteur $\mathbf{w} \in \mathbb{R}^{2N-1}$, on a :

$$\sum_{s=s_1}^{s_2} \sum_{t=t_1}^{t_2} w_s w_t = \mathbf{w}^T \mathbf{M}_{s_1, t_1}^{s_2, t_2} \mathbf{w}, \quad (\text{A.11})$$

de sorte que l'ensemble des termes de (A.9) se réécrit à l'aide de ces matrices $\mathbf{M}_{s_1, s_2}^{t_1, t_2}$:

$$\begin{aligned} & \sum_{k=1}^{2N-1} \left(\sum_{i=\kappa}^k w_i \right)^2 = \mathbf{w}^T \left[\sum_{k=1}^{2N-1} \mathbf{M}_{\kappa, \kappa}^{k, k} \right] \mathbf{w}, \\ & \sum_{k=1}^{N-1} \left(\sum_{i=N+k}^{2N-1} w_i \right)^2 = \mathbf{w}^T \left[\sum_{k=1}^{N-1} \mathbf{M}_{N+k, N+k}^{2N-1, 2N-1} \right] \mathbf{w}, \\ & 2 \sum_{k=1}^{N-1} \sum_{s=1}^k \sum_{t=N+k}^{2N-1} w_s w_t = \mathbf{w}^T \left[\sum_{k=1}^{N-1} \left(\mathbf{M}_{1, N+k}^{k, 2N-1} + \mathbf{M}_{N+k, 1}^{2N-1, k} \right) \right] \mathbf{w}. \end{aligned} \quad (\text{A.12})$$

À la vue des différents membres de droite, nous ne semblons guère plus avancés. En fait, leur développement est relativement aisé. Pour cela, il nous faut définir quatre matrices intermédiaires, explicitées ci-dessous :

\mathbf{M}_1 , une matrice Toeplitz de taille $N \times N$, spécifiée par le vecteur $[N, N - 1, \dots, 1]$;

$$\begin{aligned}
 \mathbf{M}_2 &= \begin{bmatrix} 0 & 1 & 2 & \cdots & N-1 \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & 2 \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}; \\
 \mathbf{M}_3 &= \begin{bmatrix} N-1 & N-2 & \cdots & 2 & 1 \\ N-2 & N-2 & & \vdots & \vdots \\ \vdots & & \ddots & \vdots & \vdots \\ 2 & \cdots & \cdots & 2 & \vdots \\ 1 & \cdots & \cdots & \cdots & 1 \end{bmatrix}; \\
 \mathbf{M}_4 &= \begin{bmatrix} 1 & \cdots & \cdots & \cdots & 1 \\ \vdots & 2 & \cdots & \cdots & 2 \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & N-2 & N-2 \\ 1 & 2 & \cdots & N-2 & N-1 \end{bmatrix}.
 \end{aligned} \tag{A.13}$$

L'ensemble de ces matrices permet d'effectuer plusieurs assertions, pouvant évidemment se démontrer de façon plus rigoureuse et fastidieuse :

$$\begin{aligned}
 \sum_{k=1}^{2N-1} \mathbf{M}_{\kappa, \kappa}^{k, k} &= \begin{bmatrix} \mathbf{M}_1 & \mathbf{M}_2^T \\ \mathbf{M}_2 & \mathbf{M}_3 \end{bmatrix}, \\
 \sum_{k=1}^{N-1} \mathbf{M}_{N+k, N+k}^{2N-1, 2N-1} &= \begin{bmatrix} \mathbf{0}_{N, N} & \mathbf{0}_{N, N-1} \\ \mathbf{0}_{N-1, N} & \mathbf{M}_4 \end{bmatrix}, \\
 \sum_{k=1}^{N-1} \left(\mathbf{M}_{1, N+k}^{k, 2N-1} + \mathbf{M}_{N+k, 1}^{2N-1, k} \right) &= \begin{bmatrix} \mathbf{0}_{N-1, N-1} & \mathbf{M}_2 \\ \mathbf{M}_2^T & \mathbf{0}_{N, N} \end{bmatrix},
 \end{aligned} \tag{A.14}$$

Enfin, on peut montrer que la somme de ces matrices nous donne une autre matrice Toeplitz, de dimension $2N - 1$ cette fois-ci, définie par le vecteur $[N, N - 1, \dots, 2, 1, 1, 2, \dots, N - 1]$.

Ainsi, son élément à la position (s, t) est égal à $N - l_{s,t,N}$, avec :

$$l_{s,t,N} = \min(|t - s|, 2N - 1 - |t - s|), \quad (\text{A.15})$$

ce qui nous permet de conclure la démonstration :

$$\sum_{k=1}^{N-1} \left(\sum_{i=1}^k w_i + \sum_{i=N+k}^{2N-1} w_i \right)^2 + \sum_{k=N}^{2N-1} \left(\sum_{i=k-N+1}^k w_i \right)^2 = \sum_{s,t=1}^{2N-1} (N - l_{s,t,N}) w_s w_t. \quad (\text{A.16})$$

A.2 Amélioration de la borne de Levenshtein

Pour améliorer la borne de Levenshtein, nous allons cette fois-ci travailler sur la borne supérieure (3.59). La norme de Frobenius de la matrice $\tilde{\mathbf{R}}$ peut se développer en ses termes comme suit :

$$\begin{aligned} \|\tilde{\mathbf{R}}\|_F^2 &= \sum_{l,m=1}^M \sum_{i,j=1}^{2N-1} \left| \tilde{\mathbf{S}}_i^l (\tilde{\mathbf{S}}_j^m)^H \right|^2 \\ &= \sum_{l,m=1}^M \sum_{i,j=1}^{2N-1} |\theta_{\mathbf{s}^l, \mathbf{s}^m}(l_{i,j,N})|^2 w_i w_j. \end{aligned} \quad (\text{A.17})$$

Par construction, $\tilde{\mathbf{R}}$ est une matrice carrée de dimension $2N - 1$, contenant l'ensemble des corrélations entre les séquences \mathbf{s}^m . Ainsi, la contrainte sur l'énergie nous informe sur ses coefficients diagonaux :

$$|\theta_{\mathbf{s}^m, \mathbf{s}^m}(0)|^2 = 1, \quad \forall m \in \llbracket 1, 2N - 1 \rrbracket, \quad (\text{A.18})$$

tandis que les différentes corrélations aperiodiques vérifient :

$$|\theta_{\mathbf{s}^l, \mathbf{s}^m}(i - j)|^2 \leq \frac{d^2}{N^2} \quad \text{when } |i - j| = N - d \quad \forall (l, m). \quad (\text{A.19})$$

Autrement dit, chaque élément sommatoire est borné :

- le pic principal de chaque autocorrélation avec (A.18) ;
- les différentes inter-corrélations pour les D derniers retards, à l'aide de (A.19) ;
- le reste, par θ_{\max}^2 .

En appliquant l'ensemble de propriétés, une simple énumération nous permet d'obtenir une borne plus serrée sur la norme de Frobenius de $\tilde{\mathbf{R}}$:

$$\begin{aligned} \|\tilde{\mathbf{R}}\|_F^2 &= \sum_{l,m=1}^M \sum_{i,j=1}^{2N-1} |\theta_{\mathbf{x}^l, \mathbf{x}^m}(l_{i,j,N})|^2 w_i w_j \\ &\leq M^2 \theta_{\max}^2 \sum_{i,j=1}^{2N-1} w_i w_j + M (1 - \theta_{\max}^2) \sum_{i=1}^{2N-1} w_i^2 \\ &\quad - M^2 \left(\theta_{\max}^2 - \frac{1}{N^2} \right) \sum_{\substack{i,j=1 \\ l_{i,j,N}=N-1}}^{2N-1} w_i w_j \\ &\quad - \dots \\ &\quad - M^2 \left(\theta_{\max}^2 - \frac{D^2}{N^2} \right) \sum_{\substack{i,j=1 \\ l_{i,j,N}=N-1}}^{2N-1} w_i w_j. \end{aligned} \quad (\text{A.20})$$

La preuve se termine en utilisant la condition de pondération (3.47) :

$$\|\tilde{\mathbf{R}}\|_F^2 \leq M^2 \theta_{\max}^2 + M (1 - \theta_{\max}^2) \sum_{k=1}^{2N-1} w_k^2 - \sum_{d=1}^D \left(M^2 \left(\theta_{\max}^2 - \frac{d^2}{N^2} \right) \sum_{\substack{i,j=1 \\ l_{i,j,N}=N-d}}^{2N-1} w_i w_j \right).$$

A.3 Borne proposée pour deux séquences

Cette petite annexe, très calculatoire, dévoile l'expression de la borne proposée dans le cas particulier d'un ensemble à deux séquences, avec la pondération constante, c'est-à-dire :

$$w_i := \frac{1}{2N-1} \quad \text{pour tout } i \in \llbracket 1, 2N-1 \rrbracket. \quad (\text{A.21})$$

Rappelons que, pour tout vecteur de pondération \mathbf{w} , le Théorème 3.2 nous donne la borne supérieure suivante sur le PSLR :

$$\theta_{\max}^2 \geq \frac{1}{N^2} \left[N - \frac{\tilde{Q} \left(\mathbf{w}, \frac{N(N-1)}{M}, \sum_{d=1}^D (d^2 - N) \mathbf{A}_d \right)}{1 - \mathbf{w}^T \left(\frac{1}{M} \mathbf{I} + \sum_{d=1}^D \mathbf{A}_d \right) \mathbf{w}} \right] \quad (\text{A.22})$$

avec $\tilde{Q}(\mathbf{w}, a, \mathbf{B}) = \mathbf{w}^T (a\mathbf{I} + \mathbf{B} + \mathbf{L}) \mathbf{w}$. Ainsi, il nous suffit d'expliciter les composants de cette expression.

Pour commencer, appréhendons le terme $\mathbf{w}^T \mathbf{L} \mathbf{w}$ qui s'écrit :

$$\mathbf{w}^T \mathbf{L} \mathbf{w} = \sum_{s,t=1}^{2N-1} l_{s,t,N} w_s w_t, \quad (\text{A.23})$$

avec $l_{s,t,N} = \min(|t-s|, 2N-1-|t-s|)$. Comme mentionné dans le corps du texte, la matrice \mathbf{L} est circulante :

$$\mathbf{L} = \text{circ}([0, 1, \dots, N-1, N-1, \dots, 1]^T). \quad (\text{A.24})$$

Et, de ce fait, on peut voir que :

$$\mathbf{w}^T \mathbf{L} \mathbf{w} = \frac{1}{(2N-1)^2} \sum_{s,t=1}^{2N-1} L_{s,t}. \quad (\text{A.25})$$

En exploitant la structure particulière de \mathbf{L} , on a :

$$\begin{aligned} \mathbf{w}^T \mathbf{L} \mathbf{w} &= \frac{1}{(2N-1)^2} \sum_{s=1}^{2N-1} \left(\sum_{k=0}^{N-1} k + \sum_{k=1}^{N-1} k \right) \\ &= \frac{N(N-1)}{2N-1}. \end{aligned} \quad (\text{A.26})$$

Étudions maintenant $\mathbf{w}^T \mathbf{A}_d \mathbf{w}$:

$$\begin{aligned} \mathbf{w}^T \mathbf{A}_d \mathbf{w} &= \sum_{\substack{i,j=1 \\ l_{i,j,N}=N-d}} w_i w_j \\ &= \frac{1}{(2N-1)^2} \sum_{\substack{i,j=1 \\ |i-j|=N-d \\ \text{or } |i-j|=N+d-1}}^{2N-1} 1 \end{aligned} \quad (\text{A.27})$$

(suite du calcul)

$$\begin{aligned} \mathbf{w}^T \mathbf{A}_d \mathbf{w} &= \frac{2}{(2N-1)^2} \left[\sum_{s=1}^{N+d-1} 1 + \sum_{s=1}^{N-d} 1 \right] \\ &= \frac{2}{2N-1}. \end{aligned} \quad (\text{A.28})$$

De ce calcul, on peut aisément en déduire que :

$$\begin{cases} \mathbf{w}^T \left(\sum_{d=1}^D \mathbf{A}_d \right) \mathbf{w} = \frac{2D}{2N-1}, \\ \mathbf{w}^T \left(\sum_{d=1}^D d^2 \mathbf{A}_d \right) \mathbf{w} = \frac{2}{2N-1} \frac{D(D+1)(2D+1)}{6}. \end{cases} \quad (\text{A.29})$$

En rassemblant tous ces résultats, on retrouve la borne présentée à la Section 3.6.4, à savoir :

$$\theta_{\max}^2 \geq \frac{1}{3N^2} \frac{3MN^2 - 3N^2 - MD(D+1)(2D+1)}{M(2N-2D-1) - 1}. \quad (\text{A.30})$$

Annexe B

Bornes sur le filtrage désadapté

Le Chapitre 4 a proposé une borne sur le niveau des lobes secondaires d'un ensemble de séquences avec leurs filtres désadaptés associés. Cette annexe donne les grandes lignes de sa démonstration (sans rappeler les notations). Sa structure est similaire à la précédente : nous avons en effet considéré une borne supérieure et une borne inférieure de la matrice $\tilde{\mathbf{R}}$ (définie dans la Section 4.3), que nous rappelons ci-dessous :

Lemme 4.1. (*Borne supérieure*) La norme de Frobenius de la matrice $\tilde{\mathbf{R}}$ admet la borne supérieure suivante :

$$\|\tilde{\mathbf{R}}\|_F^2 \leq \sum_{m=1}^M (|\theta_{N_k}^{m,m}|^2 - \theta_{\max}^2) \mathbf{w}^T \mathbf{w} + M^2 \theta_{\max}^2. \quad (4.54)$$

Lemme 4.2. (*Borne inférieure*) ... et la borne inférieure qui suit :

$$\|\tilde{\mathbf{R}}\|_F^2 \geq \frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{l_{N_s, N_q, K}}, \quad (4.55)$$

où $l_{N_s, N_q, K} = \min(N_s + N_q - 1, N_s + K - 1, MK)$.

En rassemblant ces deux résultats, on peut en déduire une borne sur θ_{\max}^2 , le lobe secondaire le plus élevé parmi l'ensemble des séquences de corrélation :

Théorème 4.1. Soit \mathbf{w} un vecteur vérifiant la condition de pondération (3.47). Pour tout ensemble de M séquences unimodulaires de même longueur N_s , pour tout ensemble de filtres désadaptés associés de même longueur N_q , une borne inférieure sur le lobe secondaire le plus élevé s'écrit :

$$\theta_{\max}^2 \geq \frac{1}{M^2 - M \mathbf{w}^T \mathbf{w}} \left[\frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{l_{N_s, N_q, K}} - \mathbf{w}^T \mathbf{w} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \right], \quad (4.56)$$

pour $K \in \llbracket 1, N_q \rrbracket$, et avec $l_{N_s, N_q, K} = \min(N_s + N_q - 1, N_s + K - 1, MK)$.

Nous n'allons pas faire dans l'original. La preuve de la borne supérieure est relativement similaire à ce qui s'est fait dans [Levenshtein, 1999] ; elle est néanmoins présentée ici par commodité. Il ne nous reste ensuite qu'à démontrer la borne inférieure, puisque le théorème devient une implication triviale.

B.1 Démonstration de la borne supérieure

Commençons par rappeler quelques définitions. La matrice $\tilde{\mathbf{R}}$ a été construite par blocs, comme suit :

$$\tilde{\mathbf{R}} = \tilde{\mathbf{S}}\tilde{\mathbf{Q}}^H = \begin{bmatrix} \tilde{\mathbf{S}}^1(\tilde{\mathbf{Q}}^1)^H & \dots & \tilde{\mathbf{S}}^1(\tilde{\mathbf{Q}}^M)^H \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{S}}^M(\tilde{\mathbf{Q}}^1)^H & \dots & \tilde{\mathbf{S}}^M(\tilde{\mathbf{Q}}^M)^H \end{bmatrix}. \quad (\text{B.1})$$

Pour tout $(i, j) \in \llbracket 1, N_\theta \rrbracket^2$ et $(m, l) \in \llbracket 1, M \rrbracket^2$, soient $\tilde{\mathbf{S}}_i^m$ et $\tilde{\mathbf{Q}}_j^l$ la i -ème ligne de la matrice $\tilde{\mathbf{S}}^m$ et la j -ème ligne de la matrice $\tilde{\mathbf{Q}}^l$, respectivement. $\tilde{\mathbf{S}}_i^m$ dénote la version pondérée de \mathbf{S}_i^m , c'est-à-dire :

$$\tilde{\mathbf{S}}_i^m = \mathbf{S}_i^m \sqrt{w_i}. \quad (\text{B.2})$$

En partant de sa définition, la norme de Frobenius de $\tilde{\mathbf{R}}$ s'écrit :

$$\|\tilde{\mathbf{R}}\|_F^2 = \sum_{m=1}^M \sum_{l=1}^M \sum_{i=1}^{N_\theta} \sum_{j=1}^{N_\theta} \left| \mathbf{S}_i^m (\mathbf{Q}_j^l)^H \right|^2 w_i w_j. \quad (\text{B.3})$$

Décomposons cette somme en trois cas disjoints :

$$\begin{aligned} \|\tilde{\mathbf{R}}\|_F^2 &= \sum_{m=1}^M \sum_{\substack{l=1 \\ l=m}}^M \sum_{i=1}^{N_\theta} \sum_{\substack{i=1 \\ i=j}}^{N_\theta} \left| \mathbf{S}_i^m (\mathbf{Q}_j^l)^H \right|^2 w_i w_j \\ &\quad + \sum_{m=1}^M \sum_{\substack{l=1 \\ l=m}}^M \sum_{i=1}^{N_\theta} \sum_{\substack{i=1 \\ i \neq j}}^{N_\theta} \left| \mathbf{S}_i^m (\mathbf{Q}_j^l)^H \right|^2 w_i w_j \\ &\quad + \sum_{m=1}^M \sum_{\substack{l=1 \\ l \neq m}}^M \sum_{i=1}^{N_\theta} \sum_{j=1}^{N_\theta} \left| \mathbf{S}_i^m (\mathbf{Q}_j^l)^H \right|^2 w_i w_j. \end{aligned} \quad (\text{B.4})$$

Et bornons chaque « ligne » séparément :

$$\begin{aligned} \sum_{m=1}^M \sum_{\substack{l=1 \\ l=m}}^M \sum_{i=1}^{N_\theta} \sum_{\substack{i=1 \\ i=j}}^{N_\theta} \left| \mathbf{S}_i^m (\mathbf{Q}_j^l)^H \right|^2 w_i w_j &= \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \mathbf{w}^T \mathbf{w}, \\ \sum_{m=1}^M \sum_{\substack{l=1 \\ l=m}}^M \sum_{i=1}^{N_\theta} \sum_{\substack{i=1 \\ i \neq j}}^{N_\theta} \left| \mathbf{S}_i^m (\mathbf{Q}_j^l)^H \right|^2 w_i w_j &\leq M \theta_{\max}^2 (1 - \mathbf{w}^T \mathbf{w}), \\ \sum_{m=1}^M \sum_{\substack{l=1 \\ l \neq m}}^M \sum_{i=1}^{N_\theta} \sum_{j=1}^{N_\theta} \left| \mathbf{S}_i^m (\mathbf{Q}_j^l)^H \right|^2 w_i w_j &\leq M(M-1) \theta_{\max}^2. \end{aligned} \quad (\text{B.5})$$

En rassemblant l'ensemble de ces résultats, on obtient le résultat souhaité, *i.e.* :

$$\|\tilde{\mathbf{R}}\|_F^2 \leq \sum_{m=1}^M (|\theta_{N_k}^{m,m}|^2 - \theta_{\max}^2) \mathbf{w}^T \mathbf{w} + M^2 \theta_{\max}^2. \quad (\text{B.6})$$

B.2 Démonstration de la borne inférieure

Soit $\{\lambda_i, i \in \llbracket 1, \text{rank}(\tilde{\mathbf{R}}) \rrbracket\}$, l'ensemble des valeurs propres non nulles de la matrice $\tilde{\mathbf{R}}$. Elles permettent de définir la norme de Frobenius de cette dernière :

$$\|\tilde{\mathbf{R}}\|_F^2 = \sum_{i=1}^{\text{rank}(\tilde{\mathbf{R}})} |\lambda_i|^2 \dots \quad (\text{B.7})$$

... ainsi que sa trace :

$$\text{Tr}(\tilde{\mathbf{R}}) = \sum_{i=1}^{\text{rank}(\tilde{\mathbf{R}})} \lambda_i \dots \quad (\text{B.8})$$

qui est aussi égale, par construction, à :

$$\begin{aligned} \text{Tr}(\tilde{\mathbf{R}}) &= \sum_{m=1}^M \sum_{i=1}^{N_\theta} \theta_{N_k}^{m,m} w_i \\ &= \sum_{m=1}^M \theta_{N_k}^{m,m}. \end{aligned} \quad (\text{B.9})$$

L'inégalité de Cauchy-Schwarz permet de relier les deux expressions ci-dessus :

$$\begin{aligned} |\text{Tr}(\tilde{\mathbf{R}})|^2 &\leq \text{rank}(\tilde{\mathbf{R}}) \left(\sum_{i=1}^{\text{rank}(\tilde{\mathbf{R}})} |\lambda_i|^2 \right) \\ &= \text{rank}(\tilde{\mathbf{R}}) \|\tilde{\mathbf{R}}\|_F^2. \end{aligned} \quad (\text{B.10})$$

D'où :

$$\|\tilde{\mathbf{R}}\|_F^2 \geq \frac{|\text{Tr}(\tilde{\mathbf{R}})|^2}{\text{rank}(\tilde{\mathbf{R}})}. \quad (\text{B.11})$$

À ce stade, on se doit donc de déterminer une borne supérieure sur le rang de $\tilde{\mathbf{R}}$. Pour cela, il faut se souvenir que $\tilde{\mathbf{R}} = \tilde{\mathbf{S}}\tilde{\mathbf{Q}}$. En outre, la structure particulière de ces deux matrices nous donne (en supposant que $N_q \geq N_s$) :

$$\begin{aligned} \text{rank}(\tilde{\mathbf{S}}) &= \begin{cases} \min(MK, N_s + K - 1) & \text{si } K \leq N_y \\ \min(MK, N_s + N_q - 1) & \text{sinon,} \end{cases} \\ \text{rank}(\tilde{\mathbf{Q}}) &= \min(MK, N_q + \min(K, N_s) - 1), \\ \text{rank}(\tilde{\mathbf{R}}) &\leq \min(\text{rank}(\tilde{\mathbf{S}}), \text{rank}(\tilde{\mathbf{Q}})) \\ &= \min(MK, N_s + K - 1, N_s + N_q - 1). \end{aligned} \quad (\text{B.12})$$

Posons $l_{N_s, N_q, K} := \min(N_s + N_q - 1, N_s + K - 1, MK)$. En regroupant les résultats (B.11) et (B.12), nous arrivons à la conclusion promise :

$$\|\tilde{\mathbf{R}}\|_F^2 \geq \frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{l_{N_s, N_q, K}}. \quad (\text{B.13})$$

Remarque B.1. (Cas général) Dans [Tan et al., 2019a], nous présentons une borne un peu plus générale dans la mesure où c'est la norme de la matrice $\tilde{\mathbf{R}}^{\text{op}}$ qui est encadrée. Ici, la notation $(\cdot)^{\text{op}}$ appliquée à une matrice dénote l'opérateur élevant chacun de ses éléments à la puissance p .

B.3 Un petit calcul

Détendons nous dans cette section avec un petit jeu de réécriture. Montrons, à partir de ces deux bornes, qu'on peut passer à l'expression du Théorème 4.1 :

$$\theta_{\max}^2 \geq \frac{1}{M^2 - M\mathbf{w}^T\mathbf{w}} \left[\frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{l_{N_s, N_q, K}} - \mathbf{w}^T\mathbf{w} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \right] \dots \quad (4.56)$$

... pour ensuite arriver à l'Expression (4.58) (nécessaire à la Section 4.3.2.1) :

$$\theta_{\max}^2 \geq \left(\frac{1}{M} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \right) \left[1 - \frac{M}{M - \mathbf{w}^T\mathbf{w}} \frac{l_{N_s, N_q, K} - \frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{M \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2}}{l_{N_s, N_q, K}} \right]. \quad (4.58)$$

Nous n'avons pas 36 000 solutions : partons de l'inégalité induite par l'encadrement de la norme de la matrice $\tilde{\mathbf{R}}$, et allons-y gaiement ! La suite se fera sans commentaire...

$$\begin{aligned} \frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{l_{N_s, N_q, K}} &\leq \sum_{m=1}^M (|\theta_{N_k}^{m,m}|^2 - \theta_{\max}^2) \mathbf{w}^T\mathbf{w} + M^2\theta_{\max}^2 \\ \frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{l_{N_s, N_q, K}} &\leq M^2\theta_{\max}^2 - M\theta_{\max}^2 \mathbf{w}^T\mathbf{w} + \mathbf{w}^T\mathbf{w} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \\ \implies \theta_{\max}^2 (M^2 - M\mathbf{w}^T\mathbf{w}) &\geq \frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{l_{N_s, N_q, K}} - \mathbf{w}^T\mathbf{w} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \\ \theta_{\max}^2 &\geq \frac{1}{M^2 - M\mathbf{w}^T\mathbf{w}} \left[\frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{l_{N_s, N_q, K}} - \mathbf{w}^T\mathbf{w} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \right]. \end{aligned}$$

Ceci conclut la première étape. Pour la seconde, il « suffit » de montrer que le côté droite de (4.56) est identique à celui de (4.58).

$$\begin{aligned} &\frac{1}{M^2 - M\mathbf{w}^T\mathbf{w}} \left[\frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{l_{N_s, N_q, K}} - \mathbf{w}^T\mathbf{w} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \right] \\ &= \frac{1}{M(M - \mathbf{w}^T\mathbf{w})} \left[\frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2 - l_{N_s, N_q, K} \mathbf{w}^T\mathbf{w} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2}{l_{N_s, N_q, K}} \right] \\ &= \frac{1}{M(M - \mathbf{w}^T\mathbf{w})} \left[\frac{\sum_{m=1}^M |\theta_{N_k}^{m,m}|^2}{l_{N_s, N_q, K}} \left(\frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{\sum_{m=1}^M |\theta_{N_k}^{m,m}|^2} - l_{N_s, N_q, K} \mathbf{w}^T\mathbf{w} \right) \right] \\ &= \left(\frac{1}{M} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \right) \left[\frac{1}{l_{N_s, N_q, K} (M - \mathbf{w}^T\mathbf{w})} \left(\frac{\left| \sum_{m=1}^M \theta_{N_k}^{m,m} \right|^2}{\sum_{m=1}^M |\theta_{N_k}^{m,m}|^2} - l_{N_s, N_q, K} \mathbf{w}^T\mathbf{w} \right) \right] \end{aligned}$$

(suite du calcul précédent)

$$\begin{aligned}
&= \left(\frac{1}{M} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \right) \left[\frac{\frac{|\sum_{m=1}^M \theta_{N_k}^{m,m}|^2}{\sum_{m=1}^M |\theta_{N_k}^{m,m}|^2} - l_{N_s, N_q, K} \mathbf{w}^T \mathbf{w} + M l_{N_s, N_q, K} - M l_{N_s, N_q, K}}{l_{N_s, N_q, K} (M - \mathbf{w}^T \mathbf{w})} \right] \\
&= \left(\frac{1}{M} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \right) \left[1 + \frac{\frac{|\sum_{m=1}^M \theta_{N_k}^{m,m}|^2}{\sum_{m=1}^M |\theta_{N_k}^{m,m}|^2} - M l_{N_s, N_q, K}}{l_{N_s, N_q, K} (M - \mathbf{w}^T \mathbf{w})} \right] \\
&= \left(\frac{1}{M} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \right) \left[1 + \frac{-M \left(l_{N_s, N_q, K} - \frac{|\sum_{m=1}^M \theta_{N_k}^{m,m}|^2}{M \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2} \right)}{(M - \mathbf{w}^T \mathbf{w}) l_{N_s, N_q, K}} \right] \\
&= \left(\frac{1}{M} \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2 \right) \left[1 - \frac{M}{M - \mathbf{w}^T \mathbf{w}} \frac{l_{N_s, N_q, K} - \frac{|\sum_{m=1}^M \theta_{N_k}^{m,m}|^2}{M \sum_{m=1}^M |\theta_{N_k}^{m,m}|^2}}{l_{N_s, N_q, K}} \right].
\end{aligned}$$

Ceci termine cette annexe.

Bibliographie

- [Alabaster *et al.*, 2017] C. M. Alabaster and E. J. Hughes (2017). Generating Pulse Compression Waveforms Robust to Eclipsing. In *International Conference on Radar Systems (Radar 2017)*, pages 1–6.
- [Aldayel *et al.*, 2016] O. Aldayel, V. Monga, and M. Rangaswamy (2016). Successive QCQP Refinement for MIMO Radar Waveform Design Under Practical Constraints. *IEEE Transactions on Signal Processing*, 64(14):3760–3774.
- [Aldayel *et al.*, 2017] O. Aldayel, V. Monga, and M. Rangaswamy (2017). Transmit MIMO Beampattern Design under Constant Modulus and Spectral Interference Constraints. In *2017 IEEE Radar Conference (RadarConf)*, pages 1131–1136.
- [Arlery *et al.*, 2016a] F. Arlery, R. Kassab, U. Tan, and F. Lehmann (2016a). Efficient Gradient Method for Locally Optimizing the Periodic/Aperiodic Ambiguity Function. In *2016 IEEE Radar Conference (RadarConf)*, pages 1–6.
- [Arlery *et al.*, 2016b] F. Arlery, R. Kassab, U. Tan, and F. Lehmann (2016b). Efficient Optimization of the Ambiguity Functions of Multi-Static Radar Waveforms. In *2016 17th International Radar Symposium (IRS)*, pages 1–6.
- [Arlery *et al.*, 2015] F. Arlery, M. Klein, and F. Lehmann (2015). Utilization of Spreading Codes as Dedicated Waveforms for Active Multi-Static Primary Surveillance Radar. In *Radar Symposium (IRS), 2015 16th International*, pages 327–332.
- [Arlery *et al.*, 2019] F. Arlery, U. Tan, and O. Rabaste (2019). Generalization and Improvement of the Levenshtein Lower Bound for Aperiodic Correlation. *IEEE Transactions on Information Theory*, 65(2):1258–1266.
- [Aubry *et al.*, 2016] A. Aubry, A. D. Maio, and Y. Huang (2016). MIMO Radar Beampattern Design Via PSL/ISL Optimization. *IEEE Transactions on Signal Processing*, 64(15):3955–3967.
- [Babur *et al.*, 2016] G. Babur, P. Aubry, and F. Le Chevalier (2016). Simple Transmit Diversity Technique for Phased Array Radar. *IET Radar, Sonar Navigation*, 10(6):1046–1056.
- [Baden *et al.*, 2015] J. M. Baden, M. S. Davis, and L. Schmieder (2015). Efficient Energy Gradient Calculations for Binary and Polyphase Sequences. In *Radar Conference (Radar-Con), 2015 IEEE*, pages 0304–0309.
- [Baden *et al.*, 2018] J. M. Baden, B. O’Donnell, and L. Schmieder (2018). Multiobjective Sequence Design via Gradient Descent Methods. *IEEE Transactions on Aerospace and Electronic Systems*, 54(3):1237–1252.
- [Bar-Shalom *et al.*, 2004] Y. Bar-Shalom, X. Li, and T. Kirubarajan (2004). *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. Wiley.
- [Bellman, 2003] R. Bellman (2003). *Dynamic Programming*. Dover Books on Computer Science Series. Dover Publications.
- [Blum *et al.*, 2003] C. Blum and A. Roli (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Comput. Surv.*, 35(3):268–308.

- [Blunt *et al.*, 2008] S. D. Blunt, K. Gerlach, and E. L. Mokole (2008). Pulse Compression Eclipsing-Repair. In *2008 IEEE Radar Conference*, pages 1–5.
- [Blunt *et al.*, 2016] S. D. Blunt and E. L. Mokole (2016). Overview of Radar Waveform Diversity. *IEEE Aerospace and Electronic Systems Magazine*, 31(11):2–42.
- [Bomer *et al.*, 1989] L. Bomer and M. Antweiler (1989). Polyphase Barker Sequences. *Electronics Letters*, 25(23):1577–1579.
- [Bonnans *et al.*, 2013] J. Bonnans, J. Gilbert, C. Lemarechal, and C. Sagastizábal (2013). *Numerical Optimization: Theoretical and Practical Aspects*. Universitext. Springer Berlin Heidelberg.
- [Borwein *et al.*, 2005] P. Borwein and R. Ferguson (2005). Polyphase Sequences with Low Autocorrelation. *IEEE Transactions on Information Theory*, 51(4):1564–1567.
- [Boyd *et al.*, 2004] S. Boyd and L. Vandenberghe (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA.
- [Boztaş, 1998] S. Boztaş (1998). New Lower Bounds on Aperiodic Cross-Correlation of Codes Over the n th Roots of Unity. Dept. Math., R. Melbourne Inst. Technol., Melbourne, Australia, Res. Rep. 13.
- [Brandwood, 1983] D. H. Brandwood (1983). A Complex Gradient Operator and its Application in Adaptive Array Theory. *Communications, Radar and Signal Processing, IEE Proceedings F*, 130(1):11–16.
- [Branke *et al.*, 2008] J. Branke, K. Deb, and K. Miettinen (2008). *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Genetic algorithms and evolutionary computation. Springer.
- [Briheche *et al.*, 2018] Y. Briheche, F. Barbaresco, F. Bennis, and D. Chablat (2018). Optimisation of Radar Search Patterns in Localised Clutter and Terrain Masking under Direction-Specific Scan Update Rates Constraints. *IET Radar, Sonar Navigation*, 12(12):1429–1436.
- [Cavazzuti, 2012] M. Cavazzuti (2012). *Optimization Methods: From Theory to Design Scientific and Technological Aspects in Mechanics*. Springer Berlin Heidelberg.
- [Chang *et al.*, 2018] G. Chang, A. Liu, C. Yu, Y. Ji, Y. Wang, and J. Zhang (2018). Orthogonal Waveform With Multiple Diversities for MIMO Radar. *IEEE Sensors Journal*, 18(11):4462–4476.
- [Chen *et al.*, 2008] C. Chen and P. P. Vaidyanathan (2008). MIMO Radar Ambiguity Properties and Optimization Using Frequency-Hopping Waveforms. *IEEE Transactions on Signal Processing*, 56(12):5926–5936.
- [Chen, 2009] C.-Y. Chen (2009). *Signal Processing Algorithms for MIMO Radar*. PhD thesis, California Institute of Technology.
- [Cheng *et al.*, 2017] Z. Cheng, Z. He, S. Zhang, and J. Li (2017). Constant Modulus Waveform Design for MIMO Radar Transmit Beampattern. *IEEE Transactions on Signal Processing*, 65(18):4912–4923.
- [Cherniakov, 2007] M. Cherniakov, editor (2007). *Bistatic Radar*. John Wiley & Sons, Ltd.
- [Chung *et al.*, 2010] J.-S. Chung and J.-S. No (2010). Low Correlation Zone Sequences. In C. Carlet and A. Pott, editors, *Sequences and Their Applications – SETA 2010*, pages 1–29, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Cilliers *et al.*, 2007] J. E. Cilliers and J. C. Smit (2007). Pulse Compression Sidelobe Reduction by Minimization of Lp-Norms. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3):1238–1247.
- [Collette *et al.*, 2004] Y. Collette and P. Siarry (2004). *Multiobjective Optimization: Principles and Case Studies*. Decision Engineering. Springer Berlin Heidelberg.

- [Couderc *et al.*, 2016] O. Couderc, L. Thirion-Lefevre, and R. Guinvarc'h (2016). Is it Possible to Retrieve the Complex Permittivities Using Double Bounce? In *2016 IEEE International Symposium on Antennas and Propagation (APSURSI)*, pages 1993–1994.
- [Cygan *et al.*, 2017] M. Cygan, M. Mucha, K. Wegrzycki, and M. Włodarczyk (2017). On Problems Equivalent to (min, +)-Convolution. *CoRR*, abs/1702.07669.
- [Dai *et al.*, 2008] X. Dai, Y. Bai, and L. Chen (2008). Shedding New Light on Merit Factor. In *VTC Spring 2008 - IEEE Vehicular Technology Conference*, pages 1776–1780.
- [Dam *et al.*, 2004] H. H. Dam, H. J. Zepernick, and H. Luders (2004). Polyphase Sequence Design Using a Genetic Algorithm. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 3, pages 1471–1474 Vol.3.
- [Daniel *et al.*, 2014] A. Daniel and D. C. Popescu (2014). MIMO Radar Waveform Design for Multiple Extended Targets using Compressed Sensing. In *2014 IEEE Radar Conference*, pages 0567–0572.
- [Darricau, 1994] J. Darricau (1994). *Physique et théorie du radar*. Sodipe. (in French).
- [Darricau *et al.*, 2003] J. Darricau and Y. Blanchard (2003). Histoire du radar dans le monde puis en France. *Revue de l'Électricité et de l'Électronique*. (in French).
- [Davis *et al.*, 2014] M. S. Davis, G. A. Showman, and A. D. Lanterman (2014). Coherent MIMO Radar: the Phased Array and Orthogonal Waveforms. *IEEE Aerospace and Electronic Systems Magazine*, 29(8):76–91.
- [Deng, 2004] H. Deng (2004). Polyphase Code Design for Orthogonal Netted Radar Systems. *IEEE Transactions on Signal Processing*, 52(11):3126–3135.
- [Deng *et al.*, 2016] H. Deng, Z. Geng, and B. Himed (2016). MIMO Radar Waveform Design for Transmit Beamforming and Orthogonality. *IEEE Transactions on Aerospace and Electronic Systems*, 52(3):1421–1433.
- [Désidéri, 2009] J.-A. Désidéri (2009). Multiple-Gradient Descent Algorithm (MGDA). Technical Report RR-6953, INRIA.
- [do Carmo, 1992] M. do Carmo (1992). *Riemannian Geometry*. Mathematics: Theory & Applications. Birkhäuser Boston.
- [Dréo *et al.*, 2006] J. Dréo, A. Chatterjee, A. Pérowski, P. Siarry, and E. Taillard (2006). *Metaheuristics for Hard Optimization: Methods and Case Studies*. Springer Berlin Heidelberg.
- [Ein-Dor *et al.*, 2002] L. Ein-Dor, I. Kanter, and W. Kinzel (2002). Low Autocorrelated Multiphase Sequences. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 65:020102.
- [Esmaeili-Najafabadi *et al.*, 2017] H. Esmaeili-Najafabadi, M. Ataei, and M. F. Sabahi (2017). Designing Sequence With Minimum PSL Using Chebyshev Distance and its Application for Chaotic MIMO Radar Waveform Design. *IEEE Transactions on Signal Processing*, 65(3):690–704.
- [Farnane *et al.*, 2018] K. Farnane, K. Minaoui, and D. Aboutajdine (2018). Local Search Algorithm for Low Autocorrelation Binary Sequences. In *2018 4th International Conference on Optimization and Applications (ICOA)*, pages 1–5.
- [Fishler *et al.*, 2006] E. Fishler, A. Haimovich, R. S. Blum, L. J. Cimini, D. Chizhik, and R. A. Valenzuela (2006). Spatial Diversity in Radars-models and Detection Performance. *IEEE Transactions on Signal Processing*, 54(3):823–838.
- [Floudas *et al.*, 2008] C. Floudas and P. Pardalos (2008). *Encyclopedia of Optimization*. Encyclopedia of Optimization. Springer US.
- [Frank, 1963] R. Frank (1963). Polyphase Codes with Good Nonperiodic Correlation Properties. *IEEE Transactions on Information Theory*, 9(1):43–45.

- [Frank *et al.*, 1962] R. Frank, S. Zadoff, and R. Heimiller (1962). Phase shift pulse codes with good periodic correlation properties (Corresp.). *IRE Transactions on Information Theory*, 8(6):381–382.
- [Fuhrmann *et al.*, 2008] D. R. Fuhrmann and G. S. Antonio (2008). Transmit Beamforming for MIMO Radar Systems using Signal Cross-correlation. *IEEE Transactions on Aerospace and Electronic Systems*, 44(1):171–186.
- [Fuhrmann *et al.*, 2009] D. R. Fuhrmann, J. P. Browning, and M. Rangaswamy (2009). Ambiguity Function Analysis for the Hybrid MIMO Phased-array Radar. In *2009 IEEE Radar Conference*, pages 1–6.
- [Gao *et al.*, 2016] C. Gao, K. C. Teh, A. Liu, and H. Sun (2016). Piecewise LFM Waveform for MIMO Radar. *IEEE Transactions on Aerospace and Electronic Systems*, 52(2):590–602.
- [Ge *et al.*, 2015] R. Ge, F. Huang, C. Jin, and Y. Yuan (2015). Escaping From Saddle Points - Online Stochastic Gradient for Tensor Decomposition. *CoRR*, abs/1503.02101.
- [Geng *et al.*, 2016] Z. Geng, H. Deng, and B. Himed (2016). Waveform Design for Coherent MIMO Radar Radiation Management and Transmit-Receive Beam Refocusing. In *2016 IEEE Radar Conference (RadarConf)*, pages 1–5.
- [Glover *et al.*, 2003] F. Glover and G. Kochenberger (2003). *Handbook of Metaheuristics*. International Series in Operations Research & Management Science. Springer US.
- [Golay, 1977] M. Golay (1977). Sieves for Low Autocorrelation Binary Sequences. *IEEE Transactions on Information Theory*, 23(1):43–51.
- [Gold, 1967] R. Gold (1967). Optimal Binary Sequences for Spread Spectrum Multiplexing (Corresp.). *IEEE Transactions on Information Theory*, 13(4):619–621.
- [Golomb *et al.*, 2005] S. Golomb and G. Gong (2005). *Signal Design for Good Correlation: For Wireless Communication, Cryptography, and Radar*. Cambridge University Press.
- [Grant *et al.*, 2014] M. Grant and S. Boyd (2014). CVX: Matlab Software for Disciplined Convex Programming, version 2.1. <http://cvxr.com/cvx>.
- [Griep *et al.*, 1995] K. R. Griep, J. A. Ritcey, and J. J. Burlingame (1995). Poly-phase Codes and Optimal Filters for Multiple User Ranging. *IEEE Transactions on Aerospace and Electronic Systems*, 31(2):752–767.
- [Guo *et al.*, 2015] L. Guo, H. Deng, B. Himed, T. Ma, and Z. Geng (2015). Waveform Optimization for Transmit Beamforming With MIMO Radar Antenna Arrays. *IEEE Transactions on Antennas and Propagation*, 63(2):543–552.
- [Guyvarch *et al.*, 2012] J.-P. Guyvarch, L. Savy, and F. Le Chevalier (2012). Space-Time Diversity for Active Antenna Systems. In F. Gini, A. De Maio, and L. Patton, editors, *Waveform Design and Diversity for Advanced Radar Systems*, chapter 12, pages 317–341. IET Radar, Sonar and Navigation Series.
- [Hack *et al.*, 2014] D. E. Hack, L. K. Patton, B. Himed, and M. A. Saville (2014). Detection in Passive MIMO Radar Networks. *IEEE Transactions on Signal Processing*, 62(11):2999–3012.
- [Hager *et al.*, 1994] W. W. Hager, D. Hearn, and P. M. Pardalos, editors (1994). *Large Scale Optimization*. Springer US.
- [Haimovich *et al.*, 2008] A. M. Haimovich, R. S. Blum, and L. J. Cimini (2008). MIMO Radar with Widely Separated Antennas. *IEEE Signal Processing Magazine*, 25(1):116–129.
- [Han *et al.*, 2016] K. Han and A. Nehorai (2016). Jointly Optimal Design for MIMO Radar Frequency-Hopping Waveforms Using Game Theory. *IEEE Transactions on Aerospace and Electronic Systems*, 52(2):809–820.
- [Hansen, 2016] N. Hansen (2016). The CMA Evolution Strategy: A Tutorial. *CoRR*, abs/1604.00772.

- [Hassanien *et al.*, 2017] A. Hassanien, B. Himed, and B. D. Rigling (2017). A Dual-Function MIMO Radar-Communications System Using Frequency-Hopping Waveforms. In *2017 IEEE Radar Conference (RadarConf)*, pages 1721–1725.
- [Haupt, 2010] R. Haupt (2010). *Antenna Arrays: A Computational Approach*. Wiley - IEEE. Wiley.
- [He *et al.*, 2012] H. He, J. Li, and P. Stoica (2012). *Waveform Design for Active Sensing Systems: A Computational Approach*. Cambridge University Press.
- [He *et al.*, 2009] H. He, P. Stoica, and J. Li (2009). Designing Unimodular Sequence Sets With Good Correlations—Including an Application to MIMO Radar. *IEEE Transactions on Signal Processing*, 57(11):4391–4405.
- [Hemmecke *et al.*, 2010] R. Hemmecke, M. Köppe, J. Lee, and R. Weismantel (2010). Non-linear Integer Programming. In M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, editors, *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, pages 561–618. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Hjørungnes, 2011] A. Hjørungnes (2011). *Complex-Valued Matrix Derivatives: With Applications in Signal Processing and Communications*. Cambridge University Press.
- [Hu *et al.*, 2009] L. B. Hu, H. W. Liu, S. H. Zhou, and S. J. Wu (2009). Convex Optimization Applied to Transmit Beampattern Synthesis and Signal Waveform Design for MIMO Radar. In *2009 IET International Radar Conference*, pages 1–5.
- [Jamil *et al.*, 2016] M. Jamil and H. J. Zepernick (2016). Sequence Design for Radar Applications Using Particle Swarm Optimization. In *2016 International Conference on Advanced Technologies for Communications (ATC)*, pages 210–214.
- [Jardak *et al.*, 2014] S. Jardak, S. Ahmed, and M. Alouini (2014). Generation of Correlated Finite Alphabet Waveforms Using Gaussian Random Variables. *IEEE Transactions on Signal Processing*, 62(17):4587–4596.
- [Karbasi *et al.*, 2015] S. M. Karbasi, M. Radmard, M. M. Nayebi, and M. H. Bastani (2015). Design of Multiple-Input Multiple-Output Transmit Waveform and Receive Filter for Extended Target Detection. *IET Radar, Sonar Navigation*, 9(9):1345–1353.
- [Kasami, 1966] T. Kasami (1966). Weight Distribution Formula for Some Class of Cyclic Codes. Technical report, University of Illinois.
- [Kaye *et al.*, 1970] A. Kaye and D. George (1970). Transmission of Multiplexed PAM Signals Over Multiple Channel and Diversity Systems. *IEEE Transactions on Communication Technology*, 18(5):520–526.
- [Korte *et al.*, 2006] B. Korte and J. Vygen (2006). *Combinatorial Optimization: Theory and Algorithms*. Algorithms and Combinatorics. Springer.
- [Kreutz-Delgado, 2009] K. Kreutz-Delgado (2009). The Complex Gradient Operator and the CR-Calculus. *ArXiv e-prints*.
- [Larousse, 2015] Larousse (2015). Le grand Larousse illustré. online. <https://www.larousse.fr/>.
- [Laurenceau, 2008] J. Laurenceau (2008). *Kriging-Based Response Surfaces for Aerodynamic Shape Optimisation*. Theses, Institut National Polytechnique de Toulouse - INPT.
- [Lehmann *et al.*, 2007] N. H. Lehmann, E. Fishler, A. Haimovich, R. Blum, D. Chizhik, L. Cimini, and R. Valenzuela (2007). Evaluation of Transmit Diversity in MIMO-Radar Direction Finding. *IEEE Transactions on Signal Processing*, 55(5):2215–2225.
- [Lehmann *et al.*, 2006] N. H. Lehmann, A. M. Haimovich, R. S. Blum, and L. Cimini (2006). High Resolution Capabilities of MIMO Radar. In *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, pages 25–30.
- [Lesturgie, 2011] M. Lesturgie (2011). Some Relevant Applications of MIMO to Radar. In *2011 12th International Radar Symposium (IRS)*, pages 714–721.

- [Levanon *et al.*, 2004] N. Levanon and E. Mozeson (2004). *Radar Signals*. Wiley - IEEE. Wiley.
- [Levenshtein, 1998] V. Levenshtein (1998). New Lower Bounds on Aperiodic Crosscorrelation of Binary Codes. In *Proceedings of the IEEE International Symposium on Information Theory*, page 471.
- [Levenshtein, 1999] V. Levenshtein (1999). New Lower Bounds on Aperiodic Crosscorrelation of Binary Codes. *IEEE Trans. Inf. Theory*, 45(1):284–288.
- [Li *et al.*, 2008] H. Li and T. Adalı (2008). Complex-Valued Adaptive Signal Processing Using Nonlinear Functions. *EURASIP Journal on Advances in Signal Processing*, 2008(1):765615.
- [Li *et al.*, 2017] H. Li, Y. Zhao, Z. Cheng, and D. Feng (2017). Correlated LFM Waveform Set Design for MIMO Radar Transmit Beampattern. *IEEE Geoscience and Remote Sensing Letters*, 14(3):329–333.
- [Li *et al.*, 2007] J. Li and P. Stoica (2007). MIMO Radar with Colocated Antennas. *IEEE Signal Processing Magazine*, 24(5):106–114.
- [Li *et al.*, 2009] J. Li and P. Stoica (2009). MIMO Radar: Diversity Means Superiority. In J. Li and P. Stoica, editors, *MIMO Radar Signal Processing*, chapter 1, pages 1–64. Wiley - IEEE Press.
- [Lipor *et al.*, 2014] J. Lipor, S. Ahmed, and M. S. Alouini (2014). Fourier-Based Transmit Beampattern Design Using MIMO Radar. *IEEE Transactions on Signal Processing*, 62(9):2226–2235.
- [Liu *et al.*, 2006] B. Liu, Z. He, J. Zeng, and B. Liu (2006). Polyphase Orthogonal Code Design for MIMO Radar Systems. In *2006 CIE International Conference on Radar*, pages 1–4.
- [Liu *et al.*, 2012a] D. Liu, Y. Liu, and H. Cai (2012a). Orthogonal Polyphase Code Sets Design for MIMO Radar Using Tabu Search. In *2012 IEEE International Conference on Intelligent Control, Automatic Detection and High-End Equipment*, pages 123–127.
- [Liu *et al.*, 2014a] H. Liu, S. Zhou, H. Zang, and Y. Cao (2014a). Two Waveform Design Criteria for Colocated MIMO Radar. In *2014 International Radar Conference*, pages 1–5.
- [Liu *et al.*, 2014b] J. Liu, H. Li, and B. Himed (2014b). Joint Optimization of Transmit and Receive Beamforming in Active Arrays. *IEEE Signal Processing Letters*, 21(1):39–42.
- [Liu *et al.*, 2012b] Z. Liu and Y. Guan (2012b). Meeting the Levenshtein Bound with Equality by Weighted-Correlation Complementary Set. In *Proceedings of the IEEE International Symposium on Information Theory*, pages 1010–1013.
- [Liu *et al.*, 2017] Z. Liu, Y. L. Guan, and W. H. Mow (2017). Asymptotically Locally Optimal Weight Vector Design for a Tighter Correlation Lower Bound of Quasi-Complementary Sequence Sets. *IEEE Transactions on Signal Processing*, 65(12):3107–3119.
- [Liu *et al.*, 2013] Z. Liu, Y. L. Guan, U. Paramalli, and S. Boztaş (2013). Quadratic Weight Vector for Tighter Aperiodic Levenshtein Bound. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 3130–3134.
- [Liu *et al.*, 2014c] Z. Liu, U. Paramalli, Y. Guan, and S. Boztaş (2014c). A New Weight Vector for a Tighter Levenshtein Bound on Aperiodic Correlation. *IEEE Trans. Inf. Theory*, 60(2):1356–1366.
- [Luce *et al.*, 1992] A. S. Luce, H. Molina, D. Muller, and V. Thirard (1992). Experimental Results on RIAS Digital Beamforming Radar. In *Radar 92. International Conference*, pages 74–77.
- [Malekzadeh *et al.*, 2012] M. Malekzadeh *et al.* (2012). *Intelligent Computing Technology: 8th International Conference, ICIC 2012, Huangshan, China, July 25-29, 2012. Proceedings*, chapter Optimization of Orthogonal Poly Phase Coding Waveform Based on Bees Algorithm and Artificial Bee Colony for MIMO Radar. Springer Berlin Heidelberg.

- [Mendes Ruiz *et al.*, 2015] P. Mendes Ruiz, I. Hinojosa, R. Guinvarc'h, and R. L. Haupt (2015). Size Constraint in Design of Concentric Ring Array. In *2015 IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting*, pages 2455–2456.
- [Metropolis *et al.*, 1953] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092.
- [Mian *et al.*, 2018] A. Mian, J. Ovarlez, G. Ginolhac, and A. M. Atto (2018). A Robust Change Detector for Highly Heterogeneous Multivariate Images. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3429–3433.
- [Milewski *et al.*, 2006] A. Milewski and D. Szczegieliński (2006). A Hybrid Optimization Method for Designing Polyphase Sequences. *Hydroacoustics*, 9:125–132.
- [Nocedal *et al.*, 2006] J. Nocedal and S. Wright (2006). *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York.
- [North, 1963] D. O. North (1963). An Analysis of the Factors which Determine Signal/Noise Discrimination in Pulsed-Carrier Systems. *Proceedings of the IEEE*, 51(7):1016–1027.
- [NTIA, 2000] NTIA (2000). Federal Radar Spectrum Requirements. NTIA : National Telecommunications and Information Administration.
- [Ollivier *et al.*, 2011] Y. Ollivier, L. Arnold, A. Auger, and N. Hansen (2011). Information-Geometric Optimization Algorithms: A Unifying Picture via Invariance Principles. working paper or preprint.
- [Park *et al.*, 2016] K. H. Park, H. Y. Song, D. S. Kim, and S. W. Golomb (2016). Optimal Families of Perfect Polyphase Sequences From the Array Structure of Fermat-Quotient Sequences. *IEEE Transactions on Information Theory*, 62(2):1076–1086.
- [Peng *et al.*, 2004] D. Y. Peng and P. Z. Fan (2004). Generalised Sarwate Bounds on the Aperiodic Correlation of Sequences over Complex Roots of Unity. *IEE Proceedings - Communications*, 151(4):375–382.
- [Pillo *et al.*, 2006] G. Pillo and M. Roma (2006). *Large-Scale Nonlinear Optimization*. Non-convex Optimization and Its Applications. Springer US.
- [Pilté *et al.*, 2017] M. Pilté, S. Bonnabel, and F. Barbaresco (2017). Tracking the Frenet-Serret Frame Associated to a Highly Maneuvering Target in 3D. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1969–1974.
- [Poirot *et al.*, 1978] J. L. Poirot and G. Arbid (1978). Position Location: Triangulation Versus Circulation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-14(1):48–53.
- [Pribic, 2015] R. Pribic (2015). Angular Information Resolution from Co-prime Arrays in Radar. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 2840–2844.
- [Rabaste *et al.*, 2015] O. Rabaste and L. Savy (2015). Mismatched Filter Optimization for Radar Applications Using Quadratically Constrained Quadratic Programs. *IEEE Transactions on Aerospace and Electronic Systems*, 51(4):3107–3122.
- [Rabaste *et al.*, 2013] O. Rabaste, L. Savy, M. Cattenoz, and J. P. Guyvarch (2013). Signal Waveforms and Range/Angle Coupling in Coherent Colocated MIMO Radar. In *Radar (Radar), 2013 International Conference on*, pages 157–162.
- [Reddy *et al.*, 2014] B. R. Reddy and U. M (2014). Design of orthogonal waveform for MIMO radar using Modified Ant Colony Optimization Algorithm. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2554–2559.
- [Robert, 2015] C. P. Robert (2015). The Metropolis-Hastings Algorithm.

- [Salz, 1985] J. Salz (1985). Digital Transmission over Cross-coupled Linear Channels. *AT&T Technical Journal*, 64(6):1147–1159.
- [Sammartino *et al.*, 2013] P. F. Sammartino, C. J. Baker, and H. D. Griffiths (2013). Frequency Diverse MIMO Techniques for Radar. *IEEE Transactions on Aerospace and Electronic Systems*, 49(1):201–222.
- [San Antonio *et al.*, 2007] G. San Antonio, D. R. Fuhrmann, and F. C. Robey (2007). MIMO Radar Ambiguity Functions. *IEEE Journal of Selected Topics in Signal Processing*, 1(1):167–177.
- [Skolnik, 2001] M. Skolnik (2001). *Introduction to Radar Systems*. Electrical Engineering Series. McGraw-Hill.
- [Soltanalian *et al.*, 2013] M. Soltanalian, B. Tang, J. Li, and P. Stoica (2013). Joint Design of the Receive Filter and Transmit Sequence for Active Sensing. *IEEE Signal Processing Letters*, 20(5):423–426.
- [Stoer *et al.*, 2013] J. Stoer, R. Bartels, W. Gautschi, R. Bulirsch, and C. Witzgall (2013). *Introduction to Numerical Analysis*. Springer New York.
- [Stoica *et al.*, 2012] P. Stoica, H. He, and J. Li (2012). Optimization of the Receive Filter and Transmit Sequence for Active Sensing. *IEEE Transactions on Signal Processing*, 60(4):1730–1740.
- [Stoica *et al.*, 2008] P. Stoica, J. Li, and X. Zhu (2008). Waveform Synthesis for Diversity - Based Transmit Beampattern Design. *IEEE Transactions on Signal Processing*, 56(6):2593–2598.
- [Stoica *et al.*, 2004] P. Stoica and Y. Selen (2004). Cyclic Minimizers, Majorization Techniques, and the Expectation-Maximization Algorithm: a Refresher. *IEEE Signal Processing Magazine*, 21(1):112–114.
- [Sun *et al.*, 2014] H. Sun, F. Brigui, and M. Lesturgie (2014). Analysis and Comparison of MIMO Radar Waveforms. In *Radar Conference (Radar), 2014 International*, pages 1–6.
- [Tan *et al.*, 2016a] U. Tan, C. Adnet, O. Rabaste, F. Arlery, J.-P. Ovarlez, and J.-P. Guyvarch (2016a). Coherent MIMO Waveform Design with a Gradient Descent. In *SONDRA 4th Workshop*.
- [Tan *et al.*, 2016b] U. Tan, C. Adnet, O. Rabaste, F. Arlery, J.-P. Ovarlez, and J.-P. Guyvarch (2016b). Phase Code Optimization for Coherent MIMO Radar Via a Gradient Descent. In *2016 IEEE Radar Conference (RadarConf)*, pages 1–6.
- [Tan *et al.*, 2019a] U. Tan, F. Arlery, O. Rabaste, F. Lehmann, and J.-P. Ovarlez (2019a). A New Lower Bound on the Maximum Correlation of a Set with Mismatched Filters. *IEEE Transactions on Information Theory*. (Submitted).
- [Tan *et al.*, 2016c] U. Tan, O. Rabaste, C. Adnet, F. Arlery, and J.-P. Ovarlez (2016c). Optimization Methods for Solving the Low Autocorrelation Sidelobes Problem. In *2016 17th International Radar Symposium (IRS)*, pages 1–5.
- [Tan *et al.*, 2017a] U. Tan, O. Rabaste, C. Adnet, and J.-P. Ovarlez (2017a). A Joint Optimization for Coherent MIMO Radar. In *International Conference on Radar Systems (Radar 2017)*, pages 1–6.
- [Tan *et al.*, 2017b] U. Tan, O. Rabaste, C. Adnet, and J.-P. Ovarlez (2017b). Optimisation jointe d’une séquence et d’un filtre désadapté. In *GRETSI 2017*.
- [Tan *et al.*, 2018] U. Tan, O. Rabaste, C. Adnet, and J.-P. Ovarlez (2018). A Sequence-Filter Joint Optimization. In *26th European Signal Processing Conference (EUSIPCO 2018)*, pages 1–5.
- [Tan *et al.*, 2019b] U. Tan, O. Rabaste, C. Adnet, and J.-P. Ovarlez (2019b). Designing Phase Codes for Coherent MIMO Radar. *IET Radar, Sonar and Navigation*. (To be submitted).

- [Tan *et al.*, 2019c] U. Tan, O. Rabaste, C. Adnet, and J.-P. Ovarlez (2019c). On the Eclipsing Phenomenon with Phase Codes. In *2019 International Radar Conference (Radar 2019)*. (Accepted).
- [Tang *et al.*, 2016] B. Tang, J. Tang, and Y. Zhang (2016). Design of MIMO Radar Waveforms for Rician Target Detection. *IET Radar, Sonar Navigation*, 10(9):1583–1593.
- [Ten Cate, 2015] A. Ten Cate (2015). The Wirtinger Partial Derivatives: no Need for the Independence of a Variable and its Conjugate.
- [Tsurkov, 2013] V. Tsurkov (2013). *Large-scale Optimization: Problems and Methods*. Applied Optimization. Springer US.
- [Turyn, 1963] R. Turyn (1963). On Barker Codes of Even Length. *Proceedings of the IEEE*, 51(9):1256–1256.
- [van den Bos, 1994] A. van den Bos (1994). Complex Gradient and Hessian. *IEE Proceedings - Vision, Image and Signal Processing*, 141(6):380–383.
- [Wang *et al.*, 2012] Y. C. Wang, X. Wang, H. Liu, and Z. Q. Luo (2012). On the Design of Constant Modulus Probing Signals for MIMO Radar. *IEEE Transactions on Signal Processing*, 60(8):4432–4438.
- [Welch, 1974] L. Welch (1974). Lower Bounds on the Maximum Cross-Correlation of signals (Corresp.). *IEEE Transactions of Information Theory*, 20(3):397–399.
- [Woodward, 1953] P. Woodward (1953). Theory of Radar Information. *Transactions of the IRE Professional Group on Information Theory*, 1(1):108–113.
- [Wright, 2005] M. Wright (2005). The Interior-Point Revolution in Optimization: History, Recent Developments, and Lasting Consequences. *Bulletin (New Series) of the American Mathematical Society*, 42:39–56.
- [Wu *et al.*, 2015] H. Wu, Z. Song, H. Fan, and Q. Fu (2015). Joint Design of Transmitted Sequence and Receiving Filter in the Presence of Doppler Shifts. *Circuits, Systems, and Signal Processing*, 34(8):2621–2641.
- [Xiang *et al.*, 2015] C. Xiang, C. Ding, and S. Mesnager (2015). Optimal Codebooks From Binary Codes Meeting the Levenshtein Bound. *IEEE Transactions on Information Theory*, 61(12):6526–6535.
- [Xu *et al.*, 2015] L. Xu, H. Liu, K. Yin, H. Zang, S. Zhou, and H. Wang (2015). Joint Design of Phase Coded Waveform and Mismatched Filter. In *2015 IEEE Radar Conference*, pages 32–36.
- [Xue *et al.*, 2012] M. Xue, J. Li, and P. Stoica (2012). MIMO Radar Waveform Design. In F. Gini, A. De Maio, and L. Patton, editors, *Waveform Design and Diversity for Advanced Radar Systems*, chapter 4, pages 89–120. IET Radar, Sonar and Navigation Series.
- [Yang *et al.*, 2011] Y. Yang and R. S. Blum (2011). Phase Synchronization for Coherent MIMO Radar: Algorithms and Their Analysis. *IEEE Transactions on Signal Processing*, 59(11):5538–5557.
- [Yang *et al.*, 2010] Y. Yang, R. S. Blum, Z. He, and D. R. Fuhrmann (2010). MIMO Radar Waveform Design via Alternating Projection. *IEEE Transactions on Signal Processing*, 58(3):1440–1445.
- [Zang *et al.*, 2014] H. Zang, H. Liu, S. Zhou, and X. Wang (2014). MIMO Radar Waveform Design Involving Receiving Beamforming. In *2014 International Radar Conference*, pages 1–4.
- [Zang *et al.*, 2015] H. Zang, S. Zhou, X. Lv, Y. Cao, L. Xu, and H. Liu (2015). Joint Optimization of Waveforms and Transmit Array for Colocated MIMO Radar. In *2015 IEEE Radar Conference*, pages 374–378.
- [Zhou *et al.*, 2016] S. Zhou, H. Liu, and H. Z. H. Su (2016). Doppler Sensitivity of MIMO Radar Waveforms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(5):2091–2110.

Titre : Méthodologies de conception de formes d'onde pour radars sol – Application au cas du radar MIMO

Mots-clés : formes d'onde, optimisation, radars MIMO

Résumé : Cette thèse se focalise sur le concept du radar MIMO co-localisé. L'acronyme MIMO — pour *Multiple-Input Multiple-Output* — indique l'utilisation de plusieurs émetteurs et de plusieurs récepteurs, tandis que le terme "co-localisé" signifie que ces éléments sont étroitement espacés. Chaque émetteur envoie une forme d'onde qui lui est propre : un radar MIMO émet donc simultanément un ensemble de signaux. Cette thèse a ainsi pour but d'établir une méthodologie permettant de générer cet ensemble de signaux, tout en respectant certaines contraintes opérationnelles. Cela nous permettra de déterminer les apports éventuels de ce radar. Nous nous sommes intéressés en particulier aux codes de phase, pour des raisons de couplage (qu'on peut traduire ici par la capacité, lors du traitement, à distinguer la position angulaire d'une cible de sa distance).

La méthodologie proposée se synthétise simplement en une modélisation sous la forme d'un problème d'optimisation. Contrairement à la littérature et à des précédents résultats théoriques, nous avons décidé d'évaluer l'orthogonalité des signaux émis par le radar en différentes directions, et non l'orthogonalité des signaux élémentaires. Ce problème, plus réaliste, est malheureusement non-convexe et à grande échelle : un *benchmark* sur différentes méthodes d'optimisation nous a permis de constater l'efficacité des algorithmes basés sur le gradient.

Optimiser l'orthogonalité sous-entend l'utilisation de filtres adaptés. Cependant, en pratique, le traitement radar s'effectue à l'aide de filtres désadaptés. Nous suggérons ainsi un problème d'optimisation jointe, permettant de générer de manière simultanée un ensemble de formes d'onde (pour le radar MIMO, entre autres) et les filtres désadaptés associés. Des simulations ont permis de montrer l'efficacité de la méthode. Celle-ci est en particulier préférable aux algorithmes cycliques habituellement utilisés.

Title: Implementation of Waveform Design Methods for Ground MIMO Radars

Keywords: MIMO radars, optimisation, waveform design

Abstract: This thesis deals with coherent MIMO radars. MIMO stands for *Multiple-Input Multiple-Output*, meaning that several transmitters and several receivers are used, closely-spaced in a coherent MIMO radar. Each transmitter has its own signal, providing waveform diversity. This thesis aims for defining a way to generate a set of sequences, specific for this radar, while satisfying practical constraints. It may help to determine the potential contribution of a MIMO radar. Only phase codes are concerned here, because they suffer less from the range/angle coupling effect.

A simple framework is introduced, based on an optimisation problem. While literature often involves the orthogonality of the elementary signals (because of theoretical aspects), it is suggested here to consider the orthogonality of signals from different directions of the surveillance space. Unfortunately, the obtained optimisation problem is non-convex and has a lot of variables. A benchmark on a simpler problem notifies us that gradient-based algorithms are surprisingly efficient.

An optimisation of the correlation function corresponds to a processing with matched filters. However, in practice, mismatched filters are usually employed. A joint optimisation problem is suggested accordingly, in order to generate simultaneously a set of sequences (e.g. MIMO radar signals) and their associated mismatched filters. Obtained results are quite promising: as expected, a joint optimisation seems to perform better than a cyclic one, usually employed.

