



Adaptation d'un algorithme de deuxième ordre pour la
détection de pulse sans information de quadrature par le
principe du temps de vol

Mémoire

Simon East-Lavoie

Maîtrise en génie électrique
Maître ès sciences (M.Sc.)

Québec, Canada

© Simon East-Lavoie, 2015

Résumé

Par le principe du temps vol, un système émettant un pulse peut mesurer à quelle distance se trouve des cibles en calculant les délais d'arrivée des échos retournés par les obstacles. Des situations de détection complexes doivent être résolues, telles que deux cibles dont les échos se superposent partiellement. L'algorithme de détection développé a donc pour objectif de distinguer des cibles rapprochées entre elles, de façon fiable et précise, avec une bonne immunité au bruit, sur des signaux exclusivement réels, où seule l'information sur l'amplitude de l'enveloppe du signal est disponible.

L'algorithme développé sera basé sur l'algorithme MUSIC. Ce dernier est inutilisable tel quel, dû à la nature des signaux. Une adaptation est tout d'abord élaborée, puis optimisée. Cette version de l'algorithme surpasse les performances des algorithmes de notre partenaire industriel et des méthodes de détection généralement employées et est en mesure de distinguer les échos de cibles rapprochées entre elles.

Abstract

By using the time-of-flight principle, a system emitting a pulse is able to measure the distance of a target by calculating the echoes' delays returned by the obstacles. Some complex detection situations must be solved, such as two targets producing overlapping echoes. The goal of the detection algorithm is to distinguish targets with overlapping echoes, with a good precision and a good immunity to noise, using real signals, which only the envelope's amplitude information is available.

The created algorithm is based on the MUSIC algorithm. The later is not working as it is, because of the signals' properties. An adaptation is created, and then optimized. The most substantial improvement comes from the decorrelation processing applied on the signals' covariance matrix. The effect is a decorrelation of the sources, allowing the algorithm to distinguish targets with overlapping echoes. Also, most of the decorrelation techniques help to detect echoes with low SNRs. Another improvement concerns the measurement resolution, which is better than just the sample period. The algorithm's performances exceed those of our industrial partner algorithms and those of commonly used detection methods.

The ultimate goal of the project is to integrate the developed algorithm into our industrial partner's system. It has to be real time application, and to respect the cost and resources constraints of the system. Consequently, some optimizations of the algorithm were required. Some specific properties of the covariance matrix allowed a decrease of the memory space to save its data. This way, the number of matrix's data saved represents less than 5% of the initial covariance matrix. Another optimization is done by using an iterative method for the eigenvalue decomposition, accelerating significantly the processing time.

Finally, the algorithm's performances coming out of the comparative tests completed between the adapted MUSIC algorithm and our industrial partner's algorithms demonstrate that the project's goals are fulfilled. The developed algorithm can solve the situation where two targets produce overlapping echoes, while providing a good noise immunity.

Table des matières

Résumé	iii
Abstract	v
Table des matières	vii
Liste des tableaux	xi
Liste des figures	xiii
Liste des symboles	xv
1 Introduction	1
1.1 Mise en contexte	1
1.1.1 Description du système	1
1.1.2 Description des signaux	2
1.2 Problématique et buts du projet	5
1.3 Organisation du mémoire	6
2 Algorithmes de détection potentiels	7
2.1 Seuillage	7
2.1.1 Direct	7
2.1.2 Avec corrélation ou accumulation des réponses de l'historique	9
2.2 Seuillage sur la dérivée seconde et positionnement sur la dérivée première	12
2.3 Corrélation	13
2.3.1 Adaptation de la corrélation pour des délais d'arrivée	14
2.4 Méthodes de deuxième ordre statistique	16
2.4.1 Approximation de R_{XX}	16
2.4.2 Maximum Likelihood (Vraisemblance maximale)	17
2.4.3 MUSIC	19
2.5 Conclusion sur les algorithmes de détection potentiels	21
3 Adaptation de MUSIC	23
3.1 Signaux du système	23
3.2 MUSIC pour des délais d'arrivée	25
3.3 Adaptation des signaux et du modèle	25
3.4 Formation de la matrice de covariance	26
3.5 Décomposition propre et séparation des sous-espaces propres	31

3.5.1	Discrimination des valeurs propres sources	31
	Problèmes engendrés par le choix de la méthode de construction de la matrice de covariance	32
	Critère d'Akaike	33
	Critère par seuillage simple	35
	Conclusion sur la séparation des sous-espaces propres	35
3.6	Pseudospectre	36
3.6.1	Obtention du projecteur dans l'espace bruit	36
3.6.2	Obtention du pseudospectre	36
3.7	Exemples de résultats obtenus	37
3.8	Conclusion sur l'algorithme MUSIC adapté	40
4	Optimisations des performances et implantation	41
4.1	Optimisation des performances	41
4.1.1	Prétraitement de décorrélation	41
	Échos en doublons dans la matrice de covariance	42
	Superposition décalée	46
	Modulation anti-diagonale	50
	Modulation par matrice modèle	57
4.1.2	Comparaison des lissages	59
	SNR de détection minimal	60
	Séparation d'échos partiellement superposés	62
	Conclusion sur l'élaboration de la matrice de covariance	64
	Choix du modèle pour l'obtention du pseudospectre	65
4.1.3	Critère conditionnel double de sélection des valeurs propres sources	66
	Sélection des valeurs propres sources pré-pseudospectre	67
	Détermination du nombre de sources post-pseudospectre	69
	Effets d'un mauvais dimensionnement du sous-espace propre source	70
	Cas particulier : réponses sans aucun écho	72
4.1.4	Conclusion sur le critère de détermination du nombre de sources	73
4.1.5	Précision au-delà de l'échantillon près	73
4.2	Optimisation pour l'implantation	75
4.2.1	Stockage de la matrice de covariance	76
4.2.2	Décomposition propre	79
	Comparaison versus Matlab™	82
4.3	Conclusion sur les optimisations	83
5	Tests et résultats de l'algorithme	85
5.1	Algorithmes comparatifs	85
5.1.1	Corrélation	85
5.1.2	<i>Double wavelet</i>	86
5.1.3	<i>RMS conv</i>	86
5.1.4	MUSIC	86
5.2	Séparation des échos superposés	86
5.2.1	Comparaison à d'autres algorithmes	87
5.2.2	Séparation d'échos selon le SNR	88

5.3	Rapport signal à bruit de détection minimum	89
5.4	Précision	92
5.4.1	Exactitude des mesures	92
5.4.2	Dispersion des mesures	93
5.5	Résolution	94
5.6	Conclusion des tests et résultats	95
6	Conclusion	99
A	Annexe	103
A.1	Définition et calcul du SNR	103
A.2	Adaptation	105
A.2.1	Résultats des tests	105
	Précision des algorithmes : exactitude des mesures	105
A.2.2	Précision des algorithmes : dispersion des mesures	107
	Résultats de séparation d'échos avec les données du 9 janvier 2014	108
	Bibliographie	111

Liste des tableaux

3.1	Valeurs propres selon le nombre de sources, $K = 8$, $SNR = 24.8dB$	32
4.1	Taux de détection des différents pré-traitements de décorrélation selon différents SNRs, $K = 8$	61
4.2	Taux de succès obtenus selon la méthode de construction de $\hat{\mathbf{R}}_{XX}$	64
4.3	Comparaison des valeurs propres, $e_{max} = 0.0001$	82
5.1	Taux de discrimination d'échos chevauchés des algorithmes selon l'écart inter-échos	87
5.2	Taux de réussites pour la séparation des échos superposés à différents SNRs pour MUSIC-adapté	89
5.3	Taux de détection à ± 1 échantillon des algorithmes en fonction du nombre de pré-accumulations des réponses, historiques de 4 réponses	90
5.4	Taux de détection à ± 1 échantillon des algorithmes en fonction du nombre de pré-accumulations des réponses, historiques de 2 réponses	91
5.5	Écart moyen des mesures en fonction du pas déphasage δ à différents SNRs	95

Liste des figures

1.1	Portée du système versus la longueur de la réponse enregistrée	2
1.2	Exemple d'acquisition d'un historique à 2 signaux	4
2.1	Détection par seuillage	8
2.2	Seuillage sur 2 groupes de 2 échos superposés	9
2.3	Réponses corrélées entre elles	10
2.4	Réponses accumulées	11
2.5	Résultats pour un SNR de 24.8 dB	13
2.6	Exemples de vecteurs de délais pour différentes valeurs de τ	15
2.7	Résultats de corrélation	15
2.8	Résultats pour MLH adapté en temps avec lissage spatial	18
3.1	Historique de 4 réponses, 3 échos présents aux positions $\tau_1 = 19$, $\tau_2 = 118$ et $\tau_3 = 336$	24
3.2	Exemples de vecteurs de délais pour différentes valeurs de τ	27
3.3	Résultats obtenus de MUSIC à différents SNR, utilisant un historique à 8 signaux	28
3.4	Comparaison de MUSIC sans (à gauche) et avec (à droite) <i>forward smoothing</i>	30
3.5	Premier vecteur propre pour 4 sources	32
3.6	Fonction $AIC(m)$ pour des réponses à 3 échos	34
3.7	Dérivée de la fonction $AIC(m)$ pour des réponses à 3 échos	34
3.8	Résultats pour une seule cible à différents SNR, $K = 4$	38
3.9	Résultats pour deux échos partiellement superposés, $K = 4$, $SNR = 30$ dB	39
4.1	Matrice de covariance présentant 2 doublons	42
4.2	Matrices $\hat{\mathbf{R}}_{XX}$ agrandies, 2 échos	43
4.3	Pseudospectres obtenus pour un historique de signaux où les cibles sont séparés de 1.25 m pour différentes matrices $\hat{\mathbf{R}}_{XX}$	44
4.4	$\hat{\mathbf{R}}_{XX,1}$ et $\hat{\mathbf{R}}_{XX,2}$ agrandies autour de l'écho	45
4.5	$\hat{\mathbf{R}}_{XX,3}$ agrandie autour des échos	45
4.6	Pseudospectre agrandi obtenu pour une matrice à 2 échos séparés par 2 échantillons, aucun doublon	46
4.7	Exemple de distribution de la puissance suite à une superposition décalée avec chevauchement	48
4.8	Représentation de l'emplacement des sous-matrices $\hat{\mathbf{R}}_{XX,p}$ dans la matrice de covariance $\hat{\mathbf{R}}_{XX}$, superposition décalée avec chevauchement	48
4.9	Exemple de superposition par chevauchement, 8 signaux dans l'historique à -1.2 dB, 1 seul écho, $L_p = 16$ pour le lissage	49

4.10	Représentation de l'emplacement des éléments des sous-matrices $\hat{\mathbf{R}}_{XX,p}$ dans la matrice de covariance $\hat{\mathbf{R}}_{XX}$, superposition décalée sans chevauchement	50
4.11	Matrices en exemple du principe voulant que les échos des anti-diagonales d'une matrice de covariance ait la forme du pulse au carré	51
4.12	Comparaison des formes d'un écho	52
4.13	État de la matrice $\hat{\mathbf{R}}_{XX}$ avant la modulation anti-diagonale, 8 signaux à $SNR = 4.8$, 2 échos	54
4.14	Vue du dessus d'un écho dans la matrice de covariance où la forme modulante est un pulse \mathbf{g}	55
4.15	Vue du dessus d'un écho dans la matrice de covariance où la variance du pulse modulant est multiplié par $\sqrt{2}$	56
4.16	Comparaison des pseudospectres pour 2 formes modulantes, 8 signaux à $SNR = 14.8$ dB, 1 écho	56
4.17	Matrices aux différentes étapes de la modulation par la matrice des modèles	58
4.18	Effet de la modulation sur la forme des échos de $\hat{\mathbf{R}}_{XX,modul}$	59
4.19	Comparaison des matrices $\hat{\mathbf{R}}_{XX}$, 8 signaux, $SNR = 10$ dB, 1 écho	60
4.22	Matrices $\hat{\mathbf{R}}_{XX}$ agrandies autour de l'écho pour différentes méthodes de création, réponses de l'historique à -2.7 dB	62
4.23	Réponses enregistrées pour des cibles séparées de 5 mètres	63
4.24	Forme du pulse selon différents exposants	66
4.25	Effet d'un sous-espace propre \mathbf{E}_s sous-dimensionné	71
4.26	Effet d'un sous-espace propre \mathbf{E}_s surdimensionné sur le pseudospectre obtenu	71
4.27	Effet d'un sous-espace propre \mathbf{E}_s surdimensionné sur le pseudospectre obtenu par des SNRs faibles	72
4.28	Exemple de déphasage sous-échantillon du modèle de pulse, où $\delta = 0.3$ échantillon	74
4.29	Exemple de sous-pseudospectre obtenu pour un écho à la position $\tau_2 + i\delta = 332.8$	75
4.30	Exemple de matrice de covariance démontrant les diagonales utiles	76
4.31	Matrice de covariance agrandie autour d'un écho, pré-traitement de décorrélation par superposition décalée avec chevauchement	77
4.32	Modèle de pulse	78
4.33	Comparaison des 5 vecteurs propres de plus grande importance pour les 2 méthodes	81
5.1	Écart moyen des mesures avec la distance réelle de la cible	93
5.2	Écart type des mesures selon la distance de la cible	94
A.1	Données en exemple pour le calcul du SNR	104

Liste des symboles

- $\mathbf{A}(\phi)$: matrice des vecteurs de direction
- $\mathbf{a}(\phi)$: vecteurs de direction
- $\mathbf{a}(\tau)$: vecteur de délai
- AIC : amplitude de la fonction du critère d'Akaike
- \mathbf{d}_i : i^e diagonale d'une matrice
- d_k : distance de la cible k
- \mathbf{E}_n : sous-espace propre bruit
- \mathbf{E}_s : sous-espace propre signal
- \mathbf{g} : modèle de forme gaussienne du pulse
- I : matrice identité
- K : nombre de réponses dans l'historique
- L_p : longueur d'une sous-section de réponse lors d'un pré-traitement de décorrélation
- M : nombre de sources
- N : nombre de capteurs
- n : indice entier de périodes d'échantillonnage d'une réponse
- \mathbf{n} : vecteur du bruit
- P : nombre de sous-matrices formées lors d'un pré-traitement de décorrélation
- P_{corr} : niveau de corrélation
- \mathbf{P}_{MLH} : pseudospectre de Maximum Likelihood
- \mathbf{P}_{MUSIC} : pseudospectre MUSIC
- \mathbf{P}_n : projecteur dans l'espace bruit
- $\hat{\mathbf{R}}_{NN}$: approximation de la matrice de covariance du bruit
- $\hat{\mathbf{R}}_p$: sous-matrice formée lors d'un pré-traitement de décorrélation
- \mathbf{R}_{SS} : matrice de covariance des sources ou matrice de corrélation des signaux
- \mathbf{R}_{XX} : matrice de covariance

- $\hat{\mathbf{R}}_{XX}$: matrice de covariance approximée
- \mathbf{S} : matrice des enveloppes complexes des sources
- \mathbf{s} : vecteur du signal
- t_0 : instant d'émission du pulse
- T_k : période de temps entre t_0 et le retour de l'écho #k.
- T_e : période d'échantillonnage
- \mathbf{v} : vecteur propre
- \mathbf{V} : matrice des vecteurs propres
- v_p : vitesse de propagation
- \mathbf{w}_{haar} : vecteur d'une vaguelette de Haar
- \mathbf{x} : vecteur d'une réponse
- \mathbf{X} : matrice d'un historique de réponses
- \mathbf{x}_{acc} : vecteur d'une somme de réponses
- \mathbf{x}_s : somme des réponses au carré de l'historique
- \mathbf{x}_{sum} : sommes des réponses de l'historique
- τ : délai d'arrivée d'un écho
- ϕ : angle d'arrivée
- α : amplitude complexe d'une source
- σ^2 : variance du bruit
- λ : valeur propre
- Λ : matrice des valeurs propres
- δ : pas de décalage sous-échantillon de l'écho du vecteur de délai modèle

Liste des acronymes et notations mathématiques

Acronymes

- dB : décibel
- FPGA : Field-Programmable gate array
- EVD : EigenValue Decomposition (décomposition propre)
- MLH : Maximum LikeliHood
- MUSIC : MUltiple SIgnal Classification
- RADAR : RAdio Detection And Ranging
- SNR : Signal to Noise Ratio (rapport signal à bruit)
- SONAR : SOund Navigation And Ranging

Notations mathématiques

- H : transposée complexe conjuguée
- m_X : nombre de rangées d'une matrice \mathbf{X}
- n_X : nombre de colonnes d'une matrice \mathbf{X}
- T : transposée
- \mathbf{X} : matrice, si elle est construite à partir de vecteurs, ceux-ci sont disposés sur les colonnes
- \mathbf{x} : vecteur

Chapitre 1

Introduction

1.1 Mise en contexte

Plusieurs types de systèmes permettent de déterminer la distance entre eux-mêmes et une cible. Les plus connus sont le RADAR, acronyme de RAdio Detection And Ranging, et le SONAR, SOund Navigation And Ranging. Munis d'un émetteur, ces systèmes produisent une impulsion ou un pulse, qui après avoir été réfléchis par les obstacles sur leur trajet, retourneront vers le système sous forme d'échos. Le but de ces systèmes est de mesurer le délai d'arrivée des échos pour ultimement établir la distance à laquelle l'obstacle se trouve. Lorsque la vitesse de propagation v_p du milieu est connue, il suffit de mesurer avec suffisamment de précision le temps écoulé entre l'émission du pulse et le retour d'un écho pour déterminer la distance à laquelle se trouve l'obstacle. En effet, l'équation liant la distance entre la cible k et le système au temps est $x_k = v_p \times T_k$. Comme la distance réelle parcourue est l'aller-retour, la distance de la cible d_k est déterminée par $d_k = \frac{v_p \times T_k}{2}$, où T_k est la période de temps entre l'émission et le retour de l'écho.

1.1.1 Description du système

Ce projet est issu d'une étroite collaboration avec un partenaire industriel. Le but poursuivi est d'utiliser un système équipé d'un pulseur et d'un capteur pour déterminer les délais d'arrivée des échos avec la meilleure résolution possible. Le système utilisé émet un pulse de forme gaussienne connue. Afin d'alléger la lecture de la suite de ce document, un pulse signifiera toujours un pulse de forme gaussienne représenté par une fonction de la forme $f(x) = ae^{-\frac{(x-b)^2}{2c^2}}$. Le capteur est échantillonné suite à chaque émission de pulse. Le système travaille sur la forme d'onde complète (full waveform). Autrement dit, l'échantillonnage du capteur se poursuit jusqu'à ce qu'un certain nombre d'échantillons soit enregistrés, et l'ensemble de ces échantillons est mis en mémoire constituant une réponse, disponible dans son entièreté pour les traitements subséquents. Cette façon de faire permet de travailler sur une forme d'onde entière, plutôt que sur un seul échantillon à la fois, ce qui laisse la possibilité d'appliquer des méthodes de détec-

tion supplémentaires. La fréquence d'échantillonnage doit être stable et fiable puisqu'elle fait office d'horloge. Elle détermine par le fait même quelle est la distance parcourue par le pulse entre 2 échantillons. Plus le nombre d'échantillons enregistrés à chaque pulse est grand, plus la portée en distance du système est grande. La figure 1.1 en illustre un exemple.

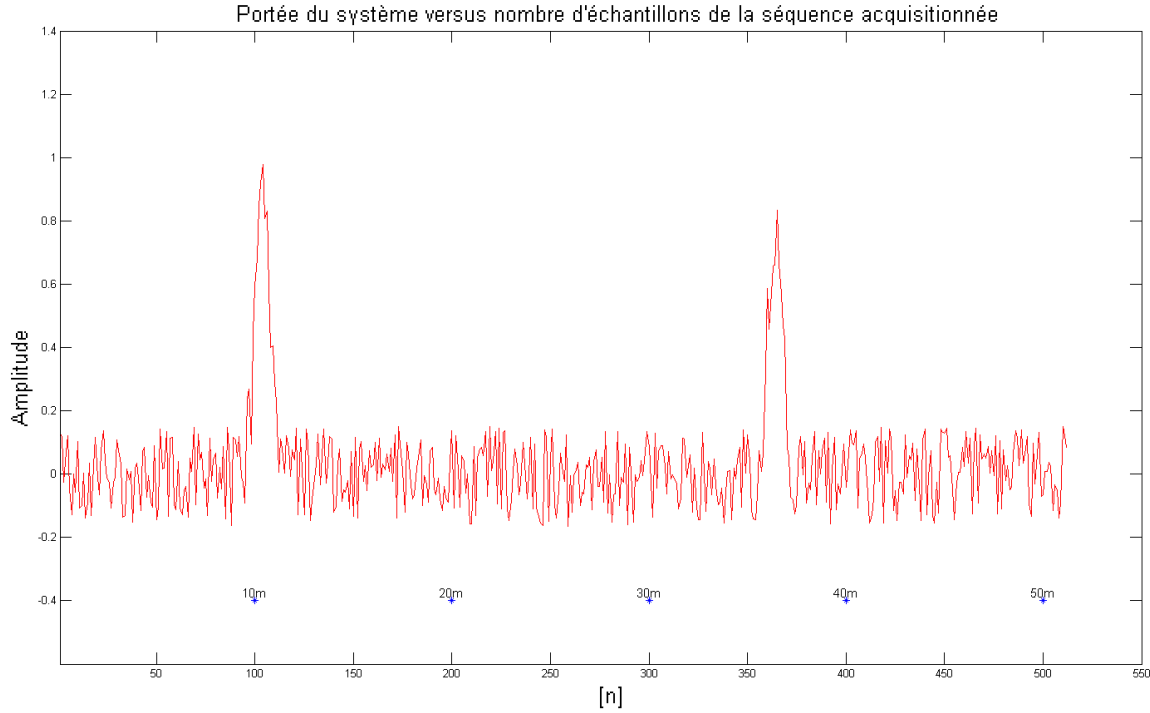


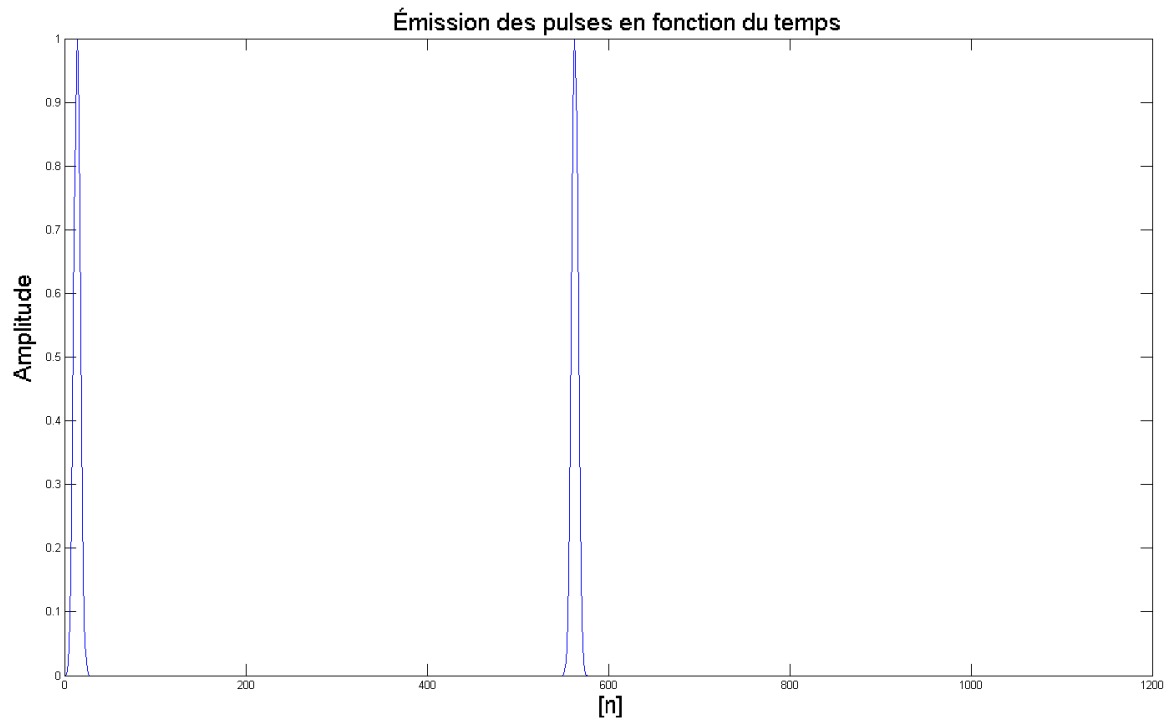
FIGURE 1.1 – Portée du système versus la longueur de la réponse enregistrée

Sur l'axe horizontal du graphique, l'indice n correspond au nombre entier de périodes d'échantillonnage T_e écoulées depuis le début de l'acquisition de la réponse. Les marqueurs en forme d'astérisque au-dessus de l'axe des x et sous la réponse affichent les distances équivalentes à chaque dizaine de mètres en terme de portée correspondant au nombre d'échantillons de la réponse. Comme on peut le voir, un système ajusté pour n'enregistrer que des réponses de 200 échantillons, pour la même fréquence d'échantillonnage, ne serait pas en mesure de détecter la cible située à une distance d'environ 37 mètres puisque sa portée est limitée à 20 mètres.

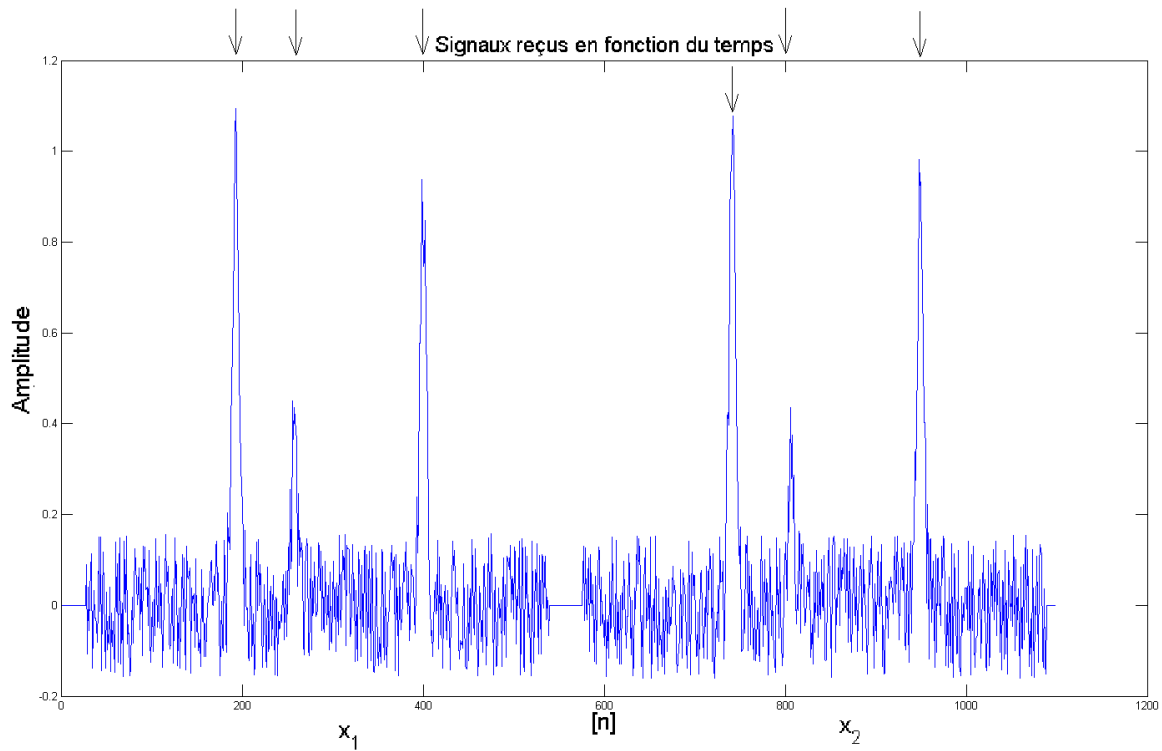
1.1.2 Description des signaux

Le système utilisé émet des pulses à une fréquence de répétition (fréquence d'émission des pulses, à ne pas confondre avec la fréquence porteuse) suffisamment élevée pour que la scène observée soit considérée comme immobile pendant plusieurs émissions. Il est donc possible de bâtir un historique contenant un certain nombre de signaux où les cibles se trouvent au même endroit et seul le bruit change. La figure 1.2 est un exemple de 2 émissions/acquisitions où 3

cibles ont renvoyé des échos. Les positions des échos sont indiquées par les flèches se trouvant entre les 2 graphiques.



(a) Émission de 2 pulses, où le sommet du pulse correspond à t_0 , soit le début de l'acquisition des signaux



(b) 2 signaux acquisitionnés suite à l'émission des 2 pulses, où les échos sont identifiés par des flèches

FIGURE 1.2 – Exemple d'acquisition d'un historique à 2 signaux

Pour déterminer le délai d'arrivée des échos, il est essentiel de déterminer une référence t_0 à partir duquel le temps écoulé est compté. Cet instant t_0 , où $\tau = 0$ correspond au sommet du pulse émis. Le délai d'un écho correspond donc au nombre d'échantillons séparant son sommet de celui du pulse émis, ce nombre d'échantillons est représenté par Δ_n . Comme la période d'échantillonnage est T_e , le délai d'un écho #k est égal à $\tau_k = \Delta_{n_k}$.

1.2 Problématique et buts du projet

- L'objectif principal de ce projet est d'estimer la distance de tous les objets retournant un écho, avec la meilleure résolution possible, même si leurs échos se superposent, en limitant la complexité calculatoire de l'algorithme pour éventuellement l'incorporer dans un système à faible coût.
- La résolution des mesures représente le degré de précision pouvant être obtenu des mesures, comme par exemple une résolution millimétrique serait préférée à une résolution au centimètre près. Une résolution précise des mesures est souhaitable dans plusieurs applications. Notamment lorsque la vitesse d'un objet est calculée à partir des mesures, une résolution grossière des distances peut produire une approximation peu fiable de la vitesse.
- Les objets doivent être repérés même si ceux-ci sont rapprochés les uns des autres. Cette situation représente un défi puisque les échos retournés ont une certaine largeur, pouvant entraîner une superposition partielle des échos et mener à des détections manquantes.
- Au final, l'algorithme de détection développé devrait être viable dans un système, comprenant principalement un FPGA. Il en découle que le nombre de calculs nécessaires à obtenir une mesure doit être raisonnable et optimisé. Un tel système à faible coût ne peut se permettre de contenir une grande puissance de calcul tels que plusieurs processeurs. De plus, le système visé par l'implantation est d'une taille réduite, limitant les possibilités de matériel électronique pouvant y être intégré.
- Comme le projet se déroule en collaboration étroite avec une entreprise, l'acquisition de signaux réels et les tests sur des situations réelles furent grandement facilités. En contrepartie, certaines informations sensibles par rapport au système, à l'algorithme ou aux optimisations d'implantation seront volontairement retenues.

1.3 Organisation du mémoire

Le chapitre 2 de ce mémoire se veut une revue des algorithmes ayant le potentiel d'effectuer des détections de positions d'échos sur les signaux réels utilisés dans le cadre de ce travail. Le chapitre 3 présente l'algorithme choisi pour atteindre les objectifs de détection fixés, ainsi que l'adaptation nécessaire à son fonctionnement sur le type particulier de signaux dont nous disposons. Le chapitre 4 présente d'une part les optimisations maximisant les performances de l'algorithme, et d'autre part, les optimisations calculatoires facilitant son implantation. Le dernier chapitre 5 avant la conclusion expose d'abord les performances de l'algorithme adapté devant différents critères de performance. Finalement, une comparaison avec d'autres algorithmes fréquemment retrouvés pour ce type de traitement est également complétée.

Chapitre 2

Algorithmes de détection potentiels

L'algorithme de détection a comme premier but de vérifier la présence ou l'absence d'échos dans le signal reçu. Il doit également indiquer l'emplacement des échos trouvés. En entrée, il reçoit un bloc contenant un minimum de 2 réponses aux pulses provenant de la même scène. Autrement dit, ces réponses aux pulses seraient théoriquement identiques si le bruit était complètement supprimé. Pour la suite de ce texte, les réponses aux pulses référeront au terme réponses, pour alléger la lecture. Les réponses obtenues, \mathbf{x}_i , sont des vecteurs stockés sur les colonnes du bloc de signaux \mathbf{X} , rebaptisé historique des réponses. Ce dernier est de dimensions $[m_X \times n_X]$. Tout au long de ce mémoire de maîtrise, les dimensions seront telles que $m_{nomdelamatrice}$ indique le nombre de lignes de la matrice et $n_{nomdelamatrice}$ indique le nombre de colonnes de la matrice. En sortie, l'algorithme doit minimalement indiquer si des échos sont repérés ou non, et à quelles positions si tel est le cas. Il peut fournir des informations supplémentaires sur l'écho trouvé telles que son rapport signal à bruit, ses pentes de montée ou de descente et sa largeur. Les performances des algorithmes sont évaluées pour différents paramètres comme la précision obtenue, la résolution, la séparation d'échos superposés, le rapport signal à bruit nécessaire à la détection (pour une définition du rapport signal à bruit des signaux utilisés, se référer en annexe à la section A.1) et le coût en calculs.

Plusieurs algorithmes ont un potentiel intéressant et présentent différents avantages. Pour chacun d'eux, une brève description de leur fonctionnement est faite. Les avantages et inconvénients de chacun sont aussi mis en évidence.

2.1 Seuillage

2.1.1 Direct

Cette méthode de détection des échos consiste simplement à déterminer un niveau limite. Lorsque la réponse dépasse ce niveau, un écho est détecté et le point maximal correspond au centre de l'écho. La simplicité est le principal avantage de cet algorithme car ses performances

sont limitées. En effet, si le niveau du seuil est trop haut, des échos pourraient ne pas être détectés. À l'inverse, un niveau trop bas engendre de fausses détections causées par le bruit. Les performances en terme de fiabilité des détections ou du rapport signal à bruit minimal sont compromises. Le cas de 2 échos partiellement superposés pose rapidement problème. Dès que les échos sont suffisamment chevauchés, le niveau entre les 2 ne redescend plus sous le seuil critique et la méthode ne détecte plus qu'un seul écho. De plus, puisque le sommet trouvé après un dépassement du seuil correspond à la position de l'écho, la précision est fortement affectée par le bruit. La figure 2.1 illustre un exemple des problèmes liés au choix du seuil.

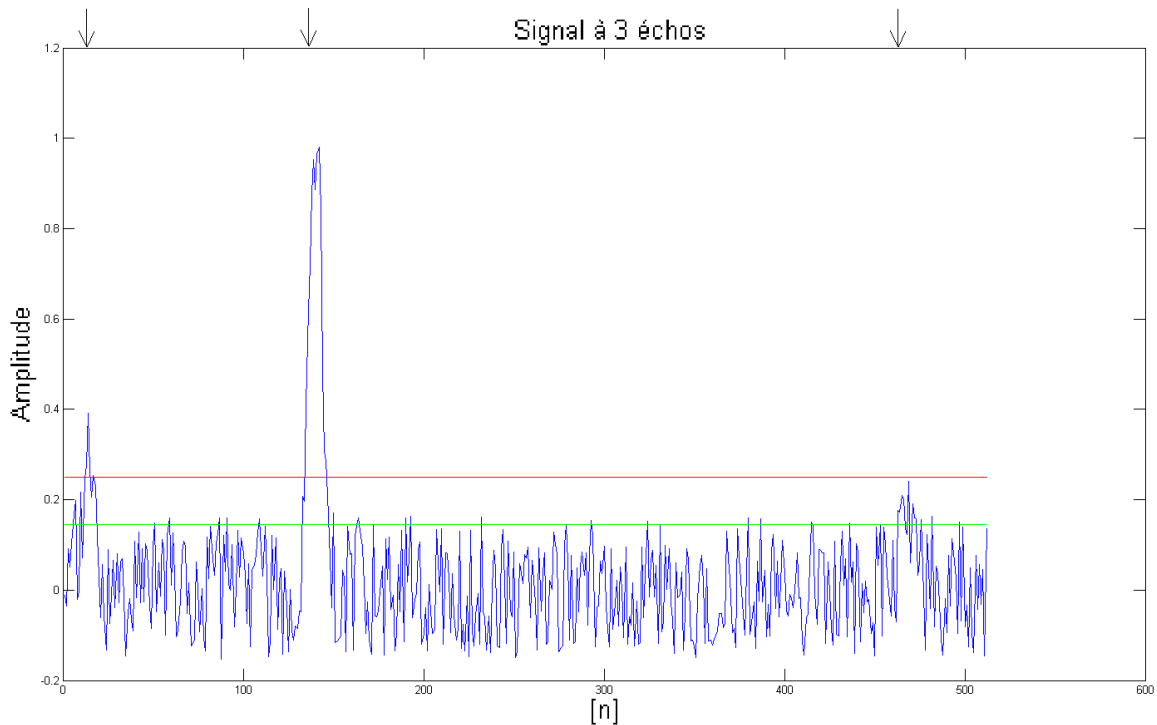


FIGURE 2.1 – Détection par seuillage

Sur la figure précédente 2.1, un niveau de seuillage plus conservateur est affiché en rouge, alors qu'un deuxième niveau de seuillage plus agressif est en vert. Les positions des 3 échos sont indiquées via des flèches au-dessus du graphique. Ce cas présente bien les compromis exigés par cet algorithme. Tout d'abord, pour les 2 cas, lorsque le seuillage est dépassé, le maximum trouvé au-dessus de cet détection n'est pas nécessairement la position exacte de l'écho dû au bruit. La précision obtenue de cet algorithme est très fortement dégradée lorsque le SNR des échos est affaibli. Deuxièmement, ce cas illustre bien le compromis entre fausse détection et détection manquante. Pour le seuil en rouge, aucune fausse détection n'est présente. Malheureusement, ce seuil est trop haut pour le troisième écho, qui lui ne sera pas détecté. À l'opposé, le seuil en vert détecte tous les échos, mais produira également un grand nombre de fausses détections dues aux pointes de bruit dépassant ce seuil. Dépendant des applications, il peut être très

difficile de fixer un seuil qui sera le compromis parfait entre fausses détections et cibles non détectées.

Le cas où deux échos sont partiellement superposés cause aussi problème pour cet algorithme.

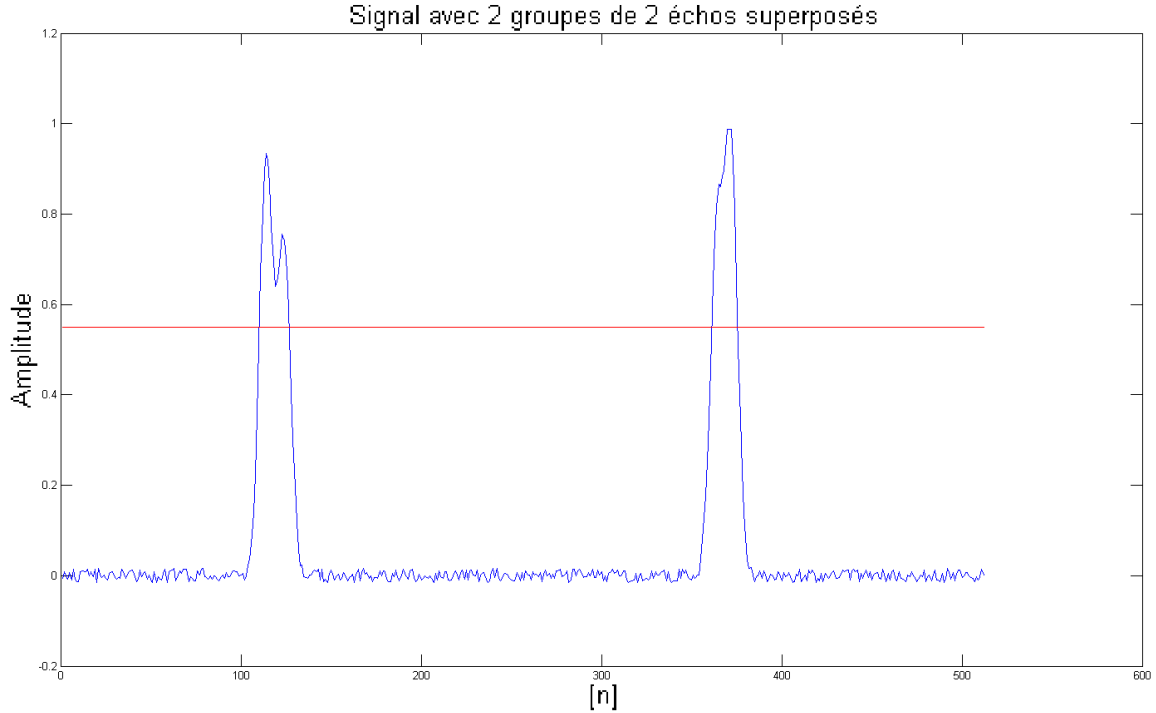


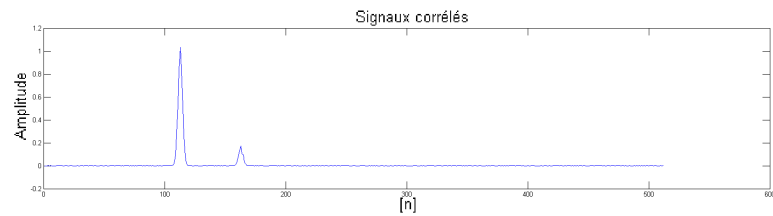
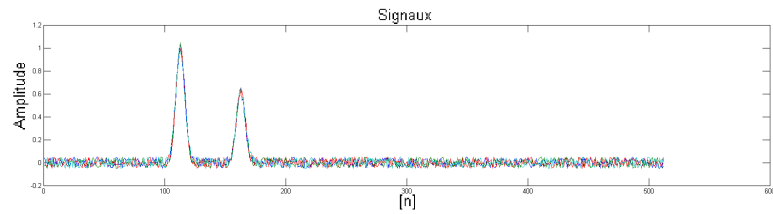
FIGURE 2.2 – Seuillage sur 2 groupes de 2 échos superposés

Sur la figure 2.2, le premier groupe d'échos est formé de 2 échos partiellement superposés, mais suffisamment séparés pour que leurs sommets soient toujours apparents. Le deuxième groupe d'échos présente un écart inter-sommets plus faible. Ainsi, seul un changement de pente sur la montée indique la présence de 2 échos. Comme le seuillage indique un écho à la position maximale d'un passage au-dessus du seuil, dans les 2 cas présentés précédemment, un seul écho sera détecté. En effet, pour les 2 groupes d'échos, le creux entre les 2 échos ne redescend pas sous le seuil, menant à 2 cibles non-détectées.

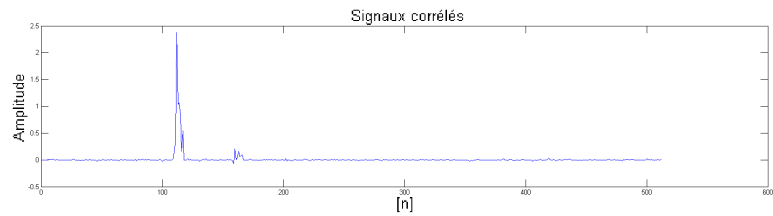
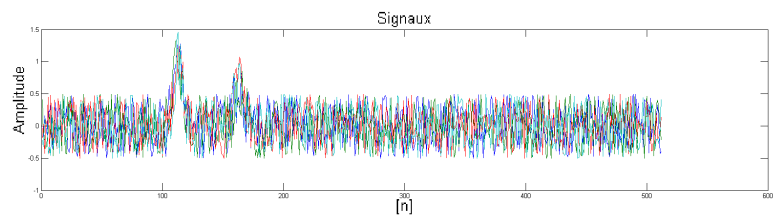
2.1.2 Avec corrélation ou accumulation des réponses de l'historique

Le seuillage avec corrélation est exactement la même méthode que le seuillage simple, sauf qu'un prétraitement est ajouté. Comme le système dispose d'un historique de quelques réponses du même scénario, la corrélation entre elles procure un meilleur SNR. Ainsi, le seuillage s'effectue sur le résultat de la corrélation \mathbf{x}_{xy} plutôt que sur les réponses individuelles directement.

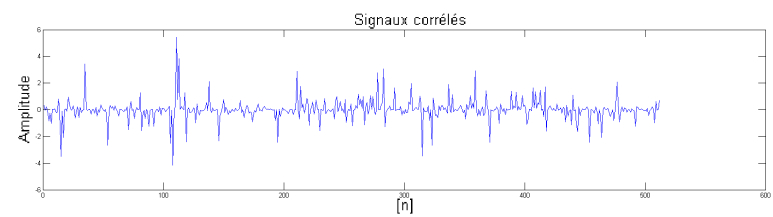
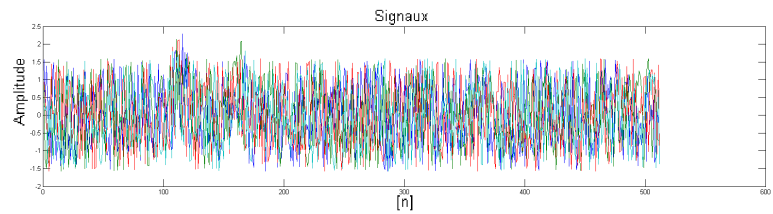
Des exemples de 4 réponses corrélées entre elles pour fournir un meilleur SNR sont présentés à la figure 2.3.



(a) SNR = 24.8 dB



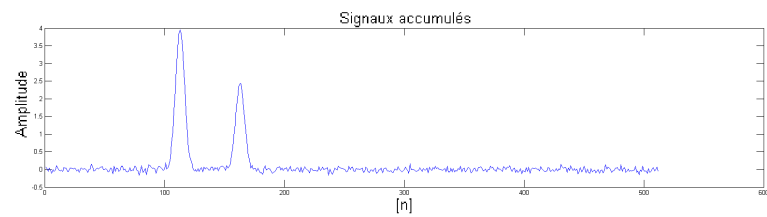
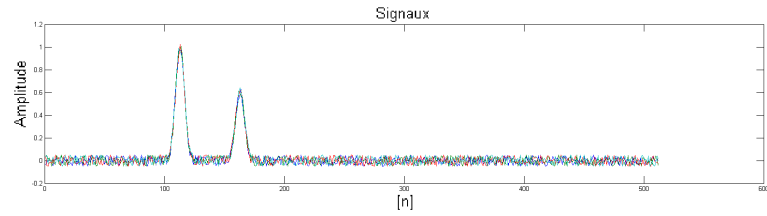
(b) SNR = 4.8 dB



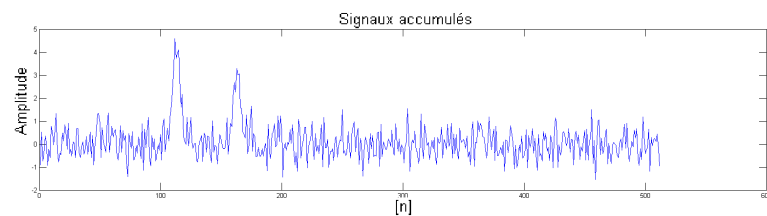
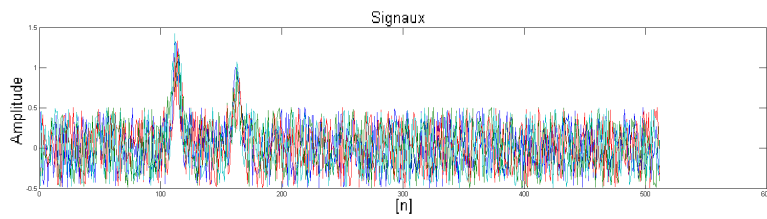
(c) SNR = -5.2 dB

FIGURE 2.3 – Réponses corrélées entre elles

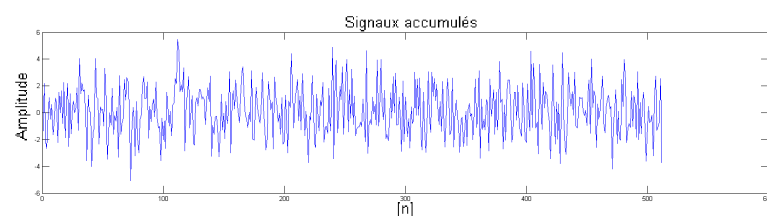
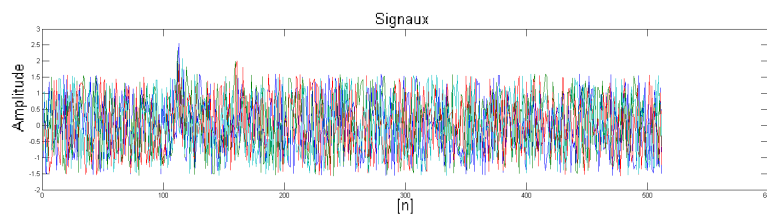
Des exemples de 4 réponses accumulées sont présentés à la figure 2.4.



(a) SNR = 24.8 dB



(b) SNR = 4.8 dB



(c) SNR = -5.2 dB

FIGURE 2.4 – Réponses accumulées

La corrélation des réponses est obtenue via la multiplication d'Hadamard, ou élément par

élément.

$$\mathbf{x}_{xy} = \bigodot_{i=1}^{n_x} \mathbf{x}_i \quad (2.1)$$

Au lieu de corrélérer entre eux les réponses de l'historique, il peut être bénéfique de les sommer.

$$\mathbf{x}_{acc} = \sum_{i=1}^{n_x} \mathbf{x}_i \quad (2.2)$$

Que ce soit la corrélation ou l'accumulation des réponses de l'historique qui soit utilisée, le SNR résultant est meilleur que le SNR des réponses initiales. La corrélation peut donner un effet plus marqué mais a le désavantage de déformer la forme des échos. Par exemple, lorsque 3 réponses sont corrélées ensemble, les échos ont une forme de gaussienne au cube plutôt qu'une gaussienne. L'accumulation des échos a un effet moindre sur le SNR mais a l'avantage de conserver la forme gaussienne des échos. Bien qu'amoindris grâce à un meilleur SNR, les problèmes liés à cette méthode sont les mêmes que pour le seuillage simple : difficulté à déterminer le niveau du seuil et à détecter 2 échos partiellement superposés, position exacte de l'écho incertaine.

2.2 Seuillage sur la dérivée seconde et positionnement sur la dérivée première

Cette méthode est le point de départ d'un algorithme de détection développé par notre partenaire industriel. De plus amples informations sont disponibles à la section 5.1.2 du chapitre 5. L'algorithme développé à l'interne de l'entreprise utilise le même principe que la vaguelette présentée ci-dessous, mais est optimisée spécifiquement aux signaux et pulses du système. Cet algorithme servira de comparaison plus tard, à la section 5.1 du chapitre 5, où les résultats de différents algorithmes sont présentés.

L'algorithme présenté ici utilise une vaguelette (wavelet) de Haar modifiée, telle que

$$\mathbf{w}_{haar} = [0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0 \ -0.1 \ -0.1 \ -0.1 \ -0.1 \ -0.1] \quad (2.3)$$

convoluée avec la réponse au pulse, le résultat est la dérivée filtrée de ce dernier. En appliquant 2 fois la convolution à la réponse, la dérivée seconde est obtenue.

$$\mathbf{x}'_{haar} = \mathbf{x} * \mathbf{w}_{haar} \quad (2.4)$$

$$\mathbf{x}''_{haar} = \mathbf{x}'_{haar} * \mathbf{w}_{haar} \quad (2.5)$$

La figure 2.5 est un exemple de résultats pour cette méthode.

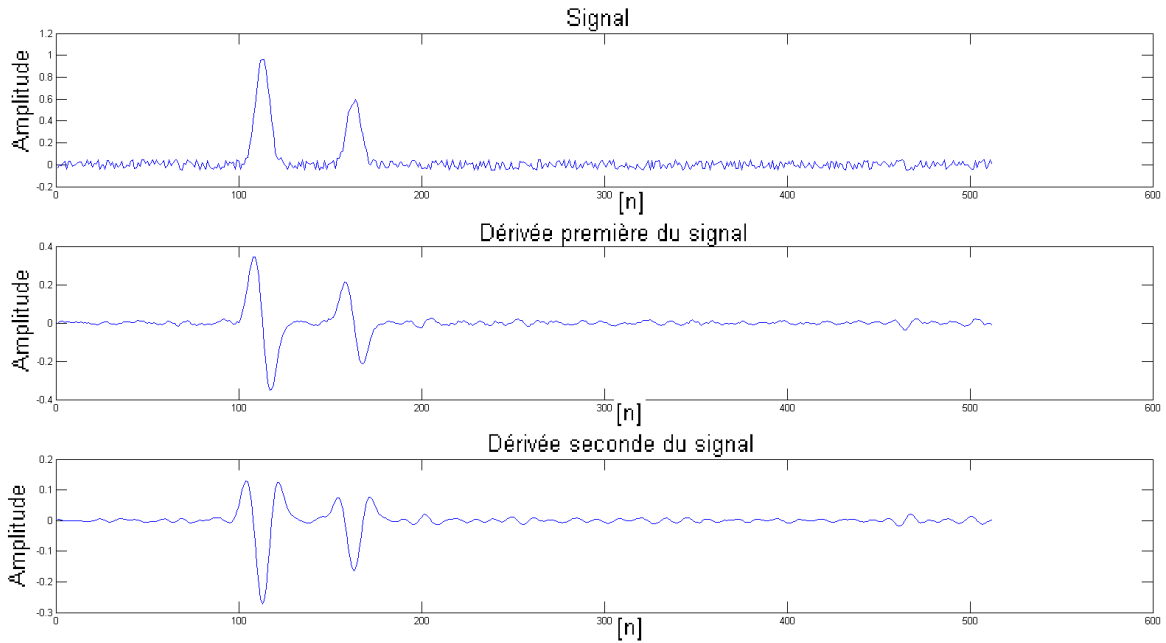


FIGURE 2.5 – Résultats pour un SNR de 24.8 dB

Sur la deuxième dérivée, les sommets sont devenus des creux. Le seuillage effectué sur la deuxième dérivée doit donc rechercher les minimums. En utilisant un nombre de points plus élevé dans la vaguelette de Haar, un effet de filtrage est obtenu, sur la première comme sur la deuxième dérivée, améliorant la précision du positionnement des échos grâce à un bruit à hautes fréquences réduit. Un avantage important de cette méthode est la résolution élevée du positionnement. Une fois le creux détecté sur la deuxième dérivée, il suffit de se reporter à cet échantillon sur la première dérivée et d'effectuer une interpolation linéaire pour trouver le passage par zéro, qui correspond au sommet de l'écho, mais avec une résolution bien au-delà de la période d'échantillonnage près.

2.3 Corrélation

Cet algorithme est utilisé, entre autres, pour déterminer des directions d'arrivée des signaux sur un réseau d'antennes, tel que décrit dans cette section de notes de cours de R. Adve[1]. Chacune des N antennes est considérée comme un capteur. À chacune des périodes d'échantillonnage T_e , une acquisition de chacun des capteurs est effectuée. Ainsi, le vecteur résultant d'une acquisition \mathbf{x} a une dimension de $[N \times 1]$, où N correspond au nombre de capteurs du réseau. Pour cette application, les réponses reçues sont exprimées par :

$$\mathbf{x}(kT_e) = \sum_{m=1}^M \alpha_m \mathbf{s}(\phi_m, kT_e) + \mathbf{n}(kT_e) \quad (2.6)$$

$$(2.7)$$

où \mathbf{n} est le vecteur contenant le bruit, \mathbf{s} est le vecteur signal, de dimensions $[N \times 1]$. Le paramètre ϕ_m , qui est celui à déterminer, correspond à l'angle d'arrivée des M signaux incidents. La variable α est l'amplitude attribuée à la source m et M est le nombre de sources des retours incidents. L'indice k correspond à la k^e séquence d'échantillonnage des capteurs, soit la k^e réponse.

Comme le produit de $\mathbf{a}^H(\phi)\mathbf{s}(\phi_m)$ est maximal si $\phi = \phi_m$, il est possible de connaître le niveau de corrélation pour chacun des angles d'arrivée via :

$$P_{corr}(\phi) = \mathbf{a}^H(\phi)\mathbf{x} \quad (2.8)$$

où $\mathbf{a}(\phi)$ est un modèle parfait d'un signal incident selon un angle ϕ .

Cette méthode telle que décrite devrait être adaptée pour utiliser des signaux sans phase et déterminer le délai d'arrivée plutôt que l'angle d'arrivée.

2.3.1 Adaptation de la corrélation pour des délais d'arrivée

Selon la définition des signaux du projet présentée à la section 3.3 du prochain chapitre, le niveau de corrélation pour chacun des délais d'arrivée est désormais obtenu par :

$$P_{corr}(\tau) = \mathbf{a}(\tau)^T \mathbf{x} \quad (2.9)$$

La transposée T est utilisée plutôt que la transposée complexe conjuguée H car les signaux du système pour des délais d'arrivée sont entièrement réels. Les vecteurs de délais $\mathbf{a}(\tau)$ sont construits à partir d'un modèle parfait d'un retour obtenu d'une source, et selon un délai de τ . Ce dernier correspond au nombre de périodes d'échantillonnage T_e séparant le sommet de l'écho de t_0 . Un exemple de quelques vecteurs de délais modèles pour différentes valeurs de τ se trouve à la figure 2.6.

Un maximum sera atteint lorsque $\tau = \tau_k$, le délai de la source k . Le vecteur $\mathbf{a}(\tau)$ contient un retour modèle, tel qu'un seul un écho à un SNR infini, ayant un délai d'arrivée de τ échantillons.

La figure 2.7 est un exemple de résultats pour cette méthode.

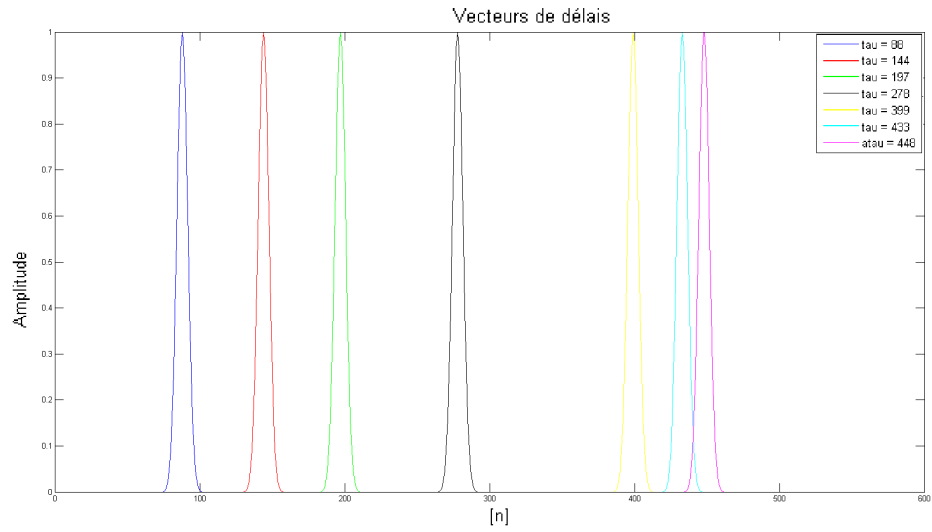


FIGURE 2.6 – Exemples de vecteurs de délais pour différentes valeurs de τ

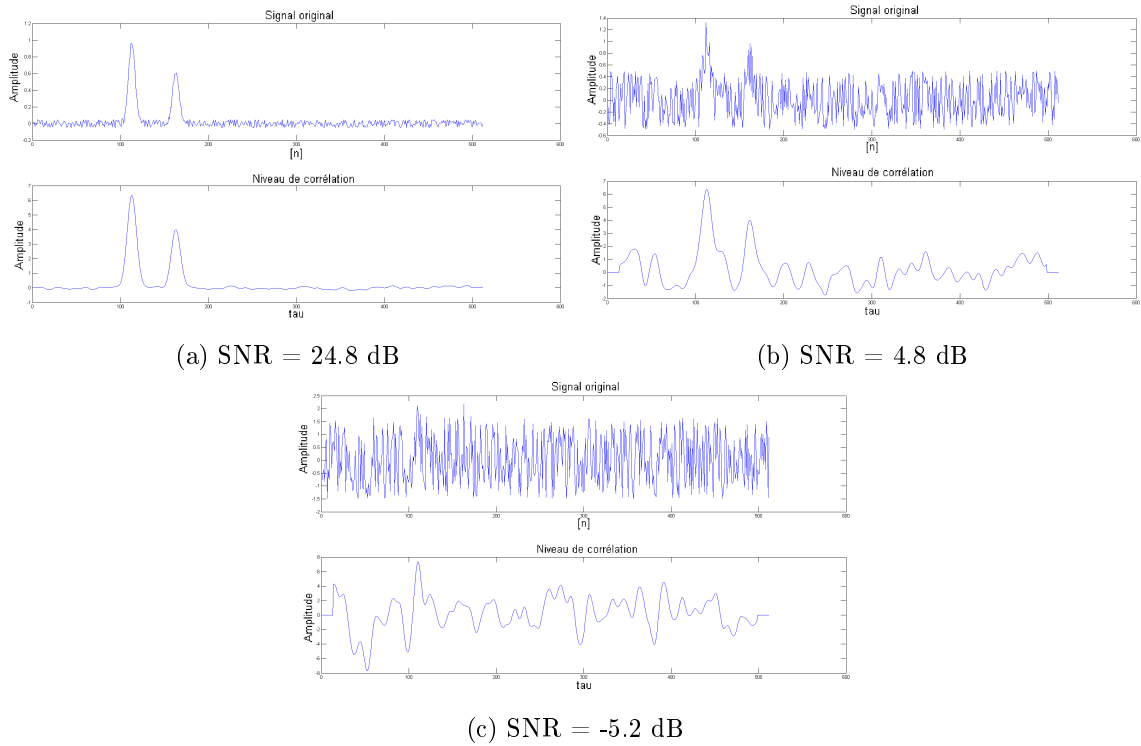


FIGURE 2.7 – Résultats de corrélation

Les indices de τ correspondant à un pic suffisamment élevé sont les positions des échos dans le signal.

2.4 Méthodes de deuxième ordre statistique

Les méthodes de deuxième ordre, telles que MLH et MUSIC, sont utilisées avec des réseaux d'antennes, principalement pour déterminer des directions d'arrivée de signaux.

Ces méthodes utilisent des propriétés statistiques de deuxième ordre du signal, telles que la corrélation ou bien la variance par exemple. Elles sont en général plus performantes que les méthodes du premier ordre vues précédemment, mais également plus coûteuses en calculs. Elles fonctionnent en général avec un SNR minimal de détection plus faible encore, mais surtout elles procurent une résolution significativement plus élevée. Ces qualités en font donc des algorithmes très intéressants pour la détermination de délais d'arrivée. Bien entendu, elles ne sont pas utilisables telles quelles dû aux types de signaux réels du présent projet, mais une adaptation en temps pour signaux réels pourrait permettre leur utilisation.

Pour les 2 méthodes exposées, MLH et MUSIC, la première étape est l'obtention d'une matrice de covariance.

2.4.1 Approximation de R_{XX}

Tel que mentionné précédemment à la section 2.3, les épreuves obtenues d'un réseau d'antenne comportant N capteurs s'expriment :

$$\mathbf{x}(kT_e) = \sum_{m=1}^M \alpha_m \mathbf{s}(\phi_m, kT_e) + \mathbf{n}(kT_e) \quad (2.10)$$

$$(2.11)$$

où \mathbf{n} est le vecteur contenant le bruit, \mathbf{s} est le vecteur signal, de dimensions $[N \times 1]$. Le paramètre ϕ_m , qui est celui à déterminer, correspond à l'angle d'arrivée des M signaux incidents. La variable α_m est l'amplitude attribuée à la source m et M est le nombre de sources des retours incidents. L'indice k correspond à la k^e séquence d'échantillonnage des capteurs, donc la k^e réponse. Une épreuve $\mathbf{x}(kT_e)$ est un vecteur des échantillons acquis sur chacun des capteurs du réseau d'antennes à l'instant kT_e . À chaque période d'échantillonnage T_e , ce sont toutes les antennes qui sont échantillonnées simultanément. Après qu'un certain nombre d'épreuves K ait été enregistré suite à l'observation de la scène présentée au système de capteurs, un historique des épreuves reçues est formé et fourni à l'algorithme qui les traitera.

L'historique \mathbf{X} des épreuves reçues du système est défini par :

$$\mathbf{X} = [\mathbf{x}_1 \quad \dots \quad \mathbf{x}_K] \quad (2.12)$$

où les \mathbf{x}_k sont les épreuves enregistrées. Le nombre d'épreuves enregistrées K correspond au nombre de colonnes de la matrice \mathbf{X} , donc $n_X = K$.

La matrice de covariance requise est définie par :

$$\mathbf{R}_{XX} = \mathbb{E}[XX^H] \quad (2.13)$$

où $\mathbb{E}[\]$ signifie l'espérance mathématique de la variable entre crochets.

De par la définition des signaux \mathbf{x} vue ci-dessus, la matrice de corrélation peut aussi être écrite comme :

$$\mathbf{R}_{XX} = \mathbf{A}\mathbf{R}_{SS}\mathbf{A}^H + \sigma^2\mathbf{I} \quad (2.14)$$

La matrice \mathbf{R}_{SS} est la matrice de covariance des signaux provenant des sources, parfois nommée matrice de corrélation des signaux. La variance du bruit est représentée par σ^2 et \mathbf{I} est une matrice identité de taille $m_X \times m_X$, où m_X correspond au nombre d'échantillons d'une épreuve \mathbf{x}_k , soit le nombre de capteurs N dans le cas où l'application est le traitement d'un réseau d'antennes. La matrice $\mathbf{A} = [\mathbf{a}(\phi_1) \ \dots \ \mathbf{a}(\phi_M)]$ contient les vecteurs de direction $\mathbf{a}(\phi)$. La matrice $\mathbf{S} = [\mathbf{s}_1(\phi) \ \dots \ \mathbf{s}_M(\phi)]$ représente les M enveloppes complexes provenant des M sources. La matrice \mathbf{A} , dans le cas où elle sert à représenter la matrice de covariance, ne contient que les vecteurs de directions des sources présentes dans le signal. Étant donné que ces méthodes de deuxième ordre servent au traitement spatial d'antenne, les vecteurs $\mathbf{a}(\phi)$ deviennent les vecteurs de balayage, notamment au moment de tracer les spectres et les pseudospectres, car ils représentent le modèle des signaux reçus sur le réseau de capteurs en fonction de la direction d'arrivée.

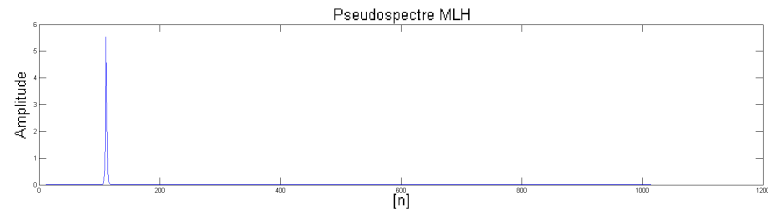
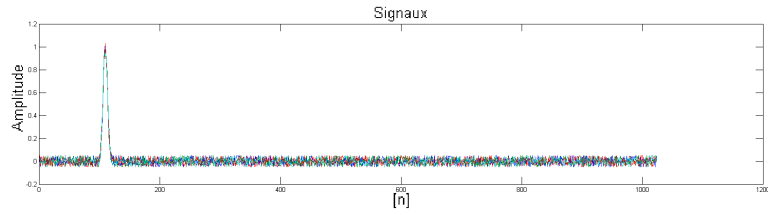
Il n'est cependant pas possible d'obtenir la matrice \mathbf{R}_{XX} définie par l'espérance mathématique. En effet, le nombre d'épreuves à enregistrer serait infini. Le système a tout d'abord une contrainte de mémoire disponible, mais plus important encore, la scène doit paraître immobile pour le système. La période de temps requise pour enregistrer un grand nombre d'épreuves invalide cette approximation d'immobilisme. Ainsi, les algorithmes présentés dans les prochaines sections utilisent une approximation de la matrice de covariance définie par :

$$\hat{\mathbf{R}}_{XX} = \frac{1}{K} \sum_{i=1}^K \mathbf{x}_i \mathbf{x}_i^H \quad (2.15)$$

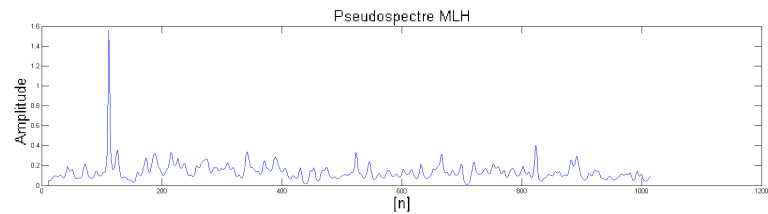
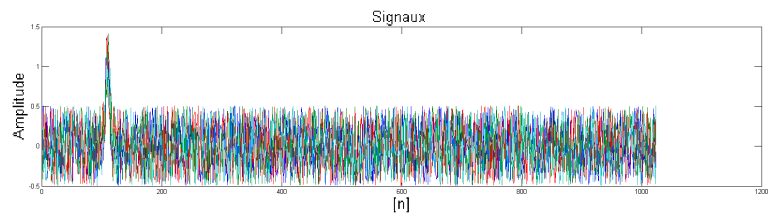
$$\hat{\mathbf{R}}_{XX} = \frac{1}{K} \mathbf{X} \mathbf{X}^H \quad (2.16)$$

2.4.2 Maximum Likelihood (Vraisemblance maximale)

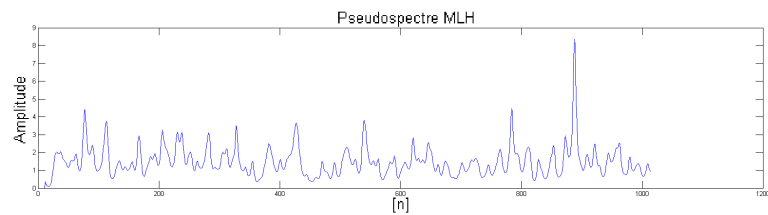
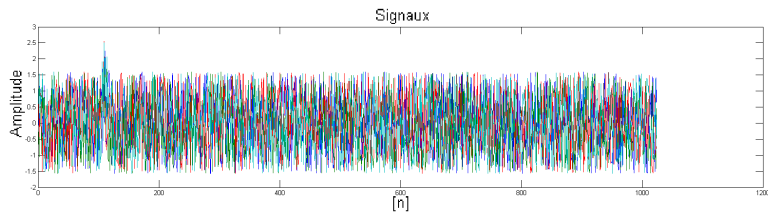
L'algorithme MLH (Maximum Likelihood) peut procurer des détections malgré de faibles SNR et une bonne résolution lors de la superposition partielle d'écho, en faisant donc un algorithme intéressant pour déterminer des délais d'arrivée.



(a) 4 signaux, $SNR = 24.8 \text{ dB}$



(b) 4 signaux, $SNR = 4.8 \text{ dB}$



(c) 4 signaux, $SNR = -5.2 \text{ dB}$

FIGURE 2.8 – Résultats pour MLH adapté en temps avec lissage spatial

En partant de la définition des signaux de directions d'arrivée présentée à la section précédente 2.3 et en se basant également sur les articles de R. Adve [1], M. K. Ibrahim[7] et S.

Shahbazpanahi, A. B. Gershman & G. B. Giannakis [17], l’algorithme MLH de Capon peut être résumé ainsi.

Dans un contexte où la direction d’arrivée est à déterminer, la matrice de covariance est approximée comme à la section précédente 2.4.1. Par la suite, le spectre est obtenu via :

$$P_{MLH}(\phi) = \frac{1}{\mathbf{a}^H(\phi)\hat{\mathbf{R}}_{XX}^{-1}\mathbf{a}(\phi)} \quad (2.17)$$

où $\mathbf{a}(\phi)$ réfère à la définition des signaux vue à la section 2.3 précédente.

Dans cette version, l’algorithme n’est pas utilisé pour des délais d’arrivée. Une adaptation est nécessaire pour y arriver, particulièrement pour les signaux utilisés dans le cadre de ce projet qui ne fournissent qu’une information sur l’amplitude de l’enveloppe. L’adaptation de l’algorithme pour obtenir des délais d’arrivée à partir de signaux réels n’est pas suffisante. En effet, la formation de la matrice de covariance via $\hat{\mathbf{R}}_{XX} = \frac{1}{K}\mathbf{X}\mathbf{X}^H$ constitue toujours un problème. MLH adapté en temps est fonctionnel seulement si la matrice de covariance est créée en utilisant une des méthodes de prétraitement de décorrélation, présentées à la section 4.1.1 du chapitre 3. La figure 2.8 présente des résultats alors obtenus avec MLH.

Cet algorithme offre une bonne séparation des échos partiellement superposés tout en ayant un SNR minimal de détection compétitif. Néanmoins, l’inversion matricielle nécessaire au calcul du spectre est exigeante en terme de calculs.

2.4.3 MUSIC

MUSIC, acronyme de Multiple Signal Classification, est un algorithme fréquemment utilisé pour déterminer des directions d’arrivée (ϕ), tel que décrit dans les article de Schmidt[16] et Rao & Rao[10]. Si son utilisation est si répandue, c’est en grande partie grâce à la haute résolution que cet algorithme procure. Pour du traitement d’antenne, autant la résolution sur un angle d’arrivée que la distinction entre deux sources ayant des directions d’arrivée rapprochées offrent des performances élevées. Certains algorithmes peuvent performer de façon semblable sur un aspect de la qualité de détection, mais au détriment des autres aspects ou au coût d’une complexité calculatoire plus gourmande.

L’application poursuivie dans la création initiale de MUSIC était le traitement de réseau d’antennes. Dans ce contexte, l’algorithme utilise un historique d’épreuves provenant de la même scène, considérée comme immobile. Cet historique $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_K]$, tel que décrit précédemment à la section 2.4.1, sert à former une matrice de covariance approximée $\hat{\mathbf{R}}_{XX}$ selon la méthode discutée à la section 2.4.1.

En reprenant à partir de la formulation initiale de la matrice de covariance :

$$\mathbf{R}_{XX} = \mathbf{A}\mathbf{R}_{SS}\mathbf{A}^H + \sigma^2\mathbf{I} \quad (2.18)$$

on assume que $M < N$, donc que le nombre de retours incidents sur le réseau de capteurs soit inférieur au nombre de capteurs. Ainsi, la matrice $\mathbf{A}\mathbf{R}_{SS}\mathbf{A}^H$, de mêmes dimensions que \mathbf{R}_{XX} , soit $[N \times N]$, a un rang égal à M , et par conséquent un espace nul de dimension $N - M$. Les colonnes de \mathbf{A} sont orthogonales à cet espace nul. Ces propriétés seront très utiles ultérieurement pour déterminer le nombre de sources incidentes. La matrice $\hat{\mathbf{R}}_{NN}$ est une approximation de la covariance du bruit, théoriquement représentée par $\sigma^2\mathbf{I}$.

La matrice de covariance approximée peut également être exprimée sous la forme d'une sommation de ses vecteurs propres pondérés par leurs valeurs propres associées :

$$\mathbf{R}_{XX} = \sum_{i=1}^N \lambda_i \mathbf{v}_i \mathbf{v}_i^H \quad (2.19)$$

Après la formation de la matrice de covariance à partir des K épreuves enregistrées dans l'historique \mathbf{X} , l'étape suivante de l'algorithme est la décomposition propre de cette matrice \mathbf{R}_{XX} . Suite à la décomposition propre, N valeurs propres λ_i sont obtenues et stockées par ordre croissant sur la diagonale principale de la matrice des valeurs propres $\mathbf{\Lambda}$, les autres éléments de cette dernière étant égaux à 0. Pour chacune des valeurs propres, un vecteur propre \mathbf{v}_i associé est obtenu. Ceux-ci sont stockés sur les colonnes de la matrice des vecteurs propres $\mathbf{V} = [\mathbf{v}_1 \dots \mathbf{v}_{m_X}]$. Comme N vecteurs propres de longueur N sont calculés, la matrice \mathbf{V} est de dimension $[N \times N]$. Les vecteurs propres sont classés dans la matrice dans le même ordre que les valeurs propres. Autrement dit, le vecteur propre associé à la plus faible valeur propre (λ_1) occupe la première colonne de la matrice \mathbf{V} alors que le vecteur propre associé à la valeur propre la plus élevée (λ_N) occupe la dernière colonne de \mathbf{V} .

Tel que mentionné auparavant, la matrice \mathbf{R}_{XX} est formé d'un espace nul de dimensions $N - M$. Ainsi, ce sont $N - M$ valeurs propres qui seront minimales et de valeurs assez semblables, d'une amplitude théoriquement égale à σ^2 , la variance du bruit, et auront tendance à tendre vers 0. Ainsi, ce sont M valeurs propres qui auront une amplitude plus élevée. Les M vecteurs propres associés à ces valeurs propres élevées sont ceux contenant les M retours des sources. En séparant les valeurs propres élevées, associées aux retours des sources, des valeurs propres faibles, associées au bruit, ce sont deux sous-espaces propres qui en résulte. Le premier sous-espace propre, baptisé \mathbf{E}_s , est formé des M vecteurs propres associés aux valeurs propres élevées. Cette nouvelle matrice \mathbf{E}_s , de dimensions $[N \times M]$, est appelé sous-espace propre signal. D'autre part, les vecteurs propres restants, associées aux valeurs propres quasi-nulles, forment le sous-espace propre bruit \mathbf{E}_n , de dimensions $[N \times (N - M)]$.

L'étape finale de l'algorithme MUSIC est l'obtention d'un pseudospectre. Ce dernier affichera M pointes indiquant les directions d'arrivée des M sources incidentes sur le réseau de capteurs dans le cas du traitement d'antennes. Ce pseudospectre est calculé selon :

$$P_{MUSIC}(\phi) = \frac{1}{\mathbf{a}^H(\phi)\mathbf{E}_n\mathbf{E}_n^H\mathbf{a}(\phi)} \quad (2.20)$$

où $\mathbf{a}(\phi)$ est le vecteur modèle de direction d'arrivée du signal pour la direction ϕ . Rappelons ici que les vecteurs directifs $\mathbf{a}(\phi_m)$ des M sources sont orthogonaux à ceux du sous-espace propre bruit \mathbf{E}_n , lorsque l'angle ϕ du vecteur modèle \mathbf{a} concorde avec la direction d'arrivée d'une source. Comme toutes les composantes de retours de sources sont absentes de \mathbf{E}_n , lorsque l'angle ϕ correspond à celui d'une source, la multiplication matricielle entre le modèle $\mathbf{a}(\phi)$ et \mathbf{E}_n donnera 0. Une division vers 0 donne un résultat qui tend vers l'infini, expliquant pourquoi les pointes du pseudospectre peuvent avoir des amplitudes excessivement élevées.

Lorsque les espaces sont bien séparés, la détection de sources sur le pseudospectre est aisée, fiable, précise, en plus de posséder la capacité de discriminer 2 sources dont les directions d'arrivée sont très rapprochées. Ces attributs en font donc un algorithme des plus intéressants. Tel que vu depuis le début, cet algorithme est appliqué à du traitement d'antenne, avec des signaux complexes possédant une phase. Il n'est malheureusement pas applicable sous cette forme pour des déterminer des délais d'arrivée sur des signaux réels. Une adaptation serait nécessaire pour y arriver. Celle-ci est présentée au prochain chapitre.

2.5 Conclusion sur les algorithmes de détection potentiels

Les algorithmes de premier ordre ont une complexité calculatoire généralement simple. Ceux-ci sont donc facilement implantables sur des systèmes temps réel. Malheureusement, ils ont des performances limitées. Certains algorithmes de premier ordre peuvent obtenir des bons résultats sur un aspect de la qualité des détections, comme la précision par exemple, mais trop souvent au détriment des autres qualités de détection, telles que la séparation des échos superposés, la détection par faibles SNR, ou encore la fiabilité des détections.

Comme le but du projet est d'améliorer tous les aspects de la qualité d'une détection, et plus particulièrement de discriminer au maximum des échos qui se chevauchent, l'algorithme idéal ne ferait pas de compromis et offrirait une résolution élevée. Ces particularités se retrouvent beaucoup plus du côté des algorithmes de deuxième ordre que de premier ordre. Dans cette optique, une analyse des algorithmes de deuxième ordre ainsi que des tests comparatifs ont dirigé notre choix vers l'algorithme MUSIC.

La disparité entre les signaux du traitement d'antenne et ceux pour des délais d'arrivée empêche l'utilisation directe l'algorithme. C'est pour cette raison qu'une adaptation de MUSIC pour déterminer des délais d'arrivée sur des signaux réels est proposée au chapitre suivant.

Chapitre 3

Adaptation de MUSIC

L'algorithme que nous devons développer dans le cadre de ce projet de mémoire a pour but de détecter les échos dans le signal, mais les détections doivent être de la plus grande qualité possible. Parmi les différentes méthodes initialement testées, MUSIC présente les meilleures détections en se basant sur la précision, la résolution, la fiabilité, la discrimination d'échos partiellement superposés et le SNR minimal atteint, bien qu'elle ne soit pas dans les méthodes des plus économes en calculs. C'est donc l'algorithme de prédilection pour ce projet.

Dans sa version classique, présentée dans le chapitre précédent à la section 2.4.3, l'algorithme est utilisé pour déterminer des directions d'arrivée. Il n'est donc pas utilisable dans cette forme. L'algorithme MUSIC est donc adapté dans le domaine du temps, pour des signaux réels ayant l'amplitude pour seule information, dans le but de déterminer les délais d'arrivée. Il est aussi optimisé aux contraintes particulières des signaux et des contraintes du présent projet.

3.1 Signaux du système

Avant de débiter l'adaptation de l'algorithme, il est de mise de décrire les signaux obtenus via le système de notre partenaire industriel. Tel que présenté dans la figure 1.2 de l'introduction, le système émet plusieurs pulses à une fréquence de répétition suffisamment élevée pour que la scène paraisse immobile, de façon à acquérir plusieurs réponses aux pulses. Ces réponses sont fournies à l'algorithme de détection par blocs, sous forme d'un historique de réponses. L'historique contenant les réponses aux K derniers pulses émis constitue la matrice $\mathbf{X} = [\mathbf{x}_1 \ \dots \ \mathbf{x}_K]$. Les \mathbf{x}_i sont des vecteurs colonnes contenant les réponses enregistrées. Un exemple d'historique à 4 réponses est présenté à la figure 3.1.

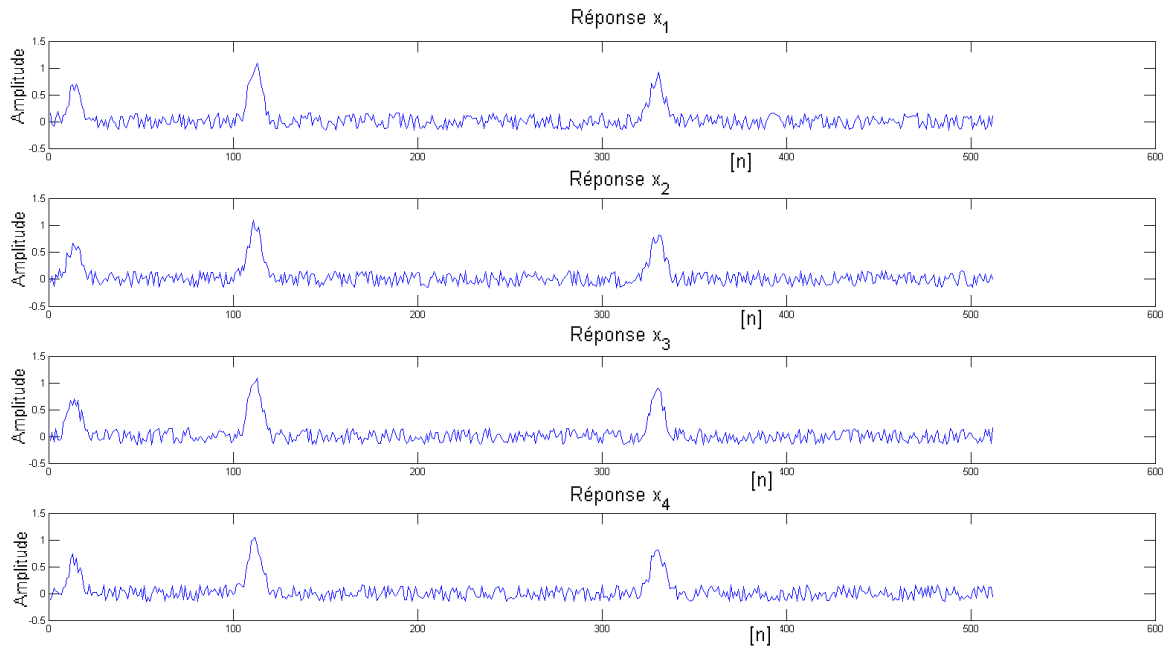


FIGURE 3.1 – Historique de 4 réponses, 3 échos présents aux positions $\tau_1 = 19$, $\tau_2 = 118$ et $\tau_3 = 336$

Comme on peut le voir, les échos ont les mêmes délais d'arrivée d'une réponse à l'autre, seul le bruit est différent.

Les réponses \mathbf{x}_i enregistrées par le système sont constituées de m_x échantillons, souvent 512 ou 1024 échantillons, mais ce nombre est ajustable et peut différer d'un système à l'autre. Pour déterminer les délais d'arrivée des échos d'une réponse, le temps de référence t_0 correspond au sommet du pulse émis (non visible sur la figure précédente, mais visible sur la figure 1.2). Ainsi, le délai d'arrivée d'un écho converti en temps vaut donc $\tau[s] = \Delta n T_e$, où Δn est le nombre d'échantillons séparant le sommet de l'écho et le sommet du pulse émis, T_e est la période d'échantillonnage.

Caractéristique particulière des signaux La caractéristique la plus importante des signaux utilisés est qu'ils ne fournissent qu'une information, soit l'amplitude de l'enveloppe. Lorsque la fréquence de la porteuse du pulse est très élevée par rapport à la fréquence d'échantillonnage, il n'est plus possible de récupérer l'information de phase du signal. Ainsi, seule l'enveloppe de la pulse peut être récupérée. Les échantillons sont donc des nombres réels représentant l'amplitude de l'enveloppe.

3.2 MUSIC pour des délais d'arrivée

Des adaptations de l'algorithme MUSIC pour déterminer des délais d'arrivée via le principe du temps de vol ont été élaborées par le passé. Des articles de Bruckstein, Shan et Kailath[2], Pallas et Jourdain[11] et Grenier, Pigeon et Turner[5] sont des exemples d'adaptations en temps permettant la localisation haute résolution d'échos. Dans sa version initiale pour des directions d'arrivée, les signaux fournis à l'algorithme MUSIC étaient les suivants :

$$\mathbf{x}_k = \sum_{m=1}^M \alpha_{m,k} \mathbf{s}_k(\phi_m) + \mathbf{n}_k \quad (3.1)$$

Les réponses \mathbf{x}_k sont les épreuves de la scène enregistrées aux périodes d'échantillonnage, où $t_e = kT_e$. La variable α représente l'amplitude complexe du signal reçu \mathbf{s} . Le bruit à l'instant d'échantillonnage kT_e est \mathbf{n}_k .

L'utilisation de MUSIC pour déterminer des délais d'arrivée avec des signaux complexes est faite avec des signaux du type :

$$\mathbf{x}(t) = \sum_{i=1}^M \alpha_i \mathbf{s}(t - \tau_i) + \mathbf{n}(t) \quad (3.2)$$

où $\alpha_i = |\alpha_i|e^{j\phi_i}$. Les amplitudes α_i et phases ϕ_i sont aléatoires et les signaux $\mathbf{s}(t)$ sont complexes. Le nombre de retours correspond au paramètre M . Lorsqu'utilisé pour des signaux temporels, la longueur du vecteur \mathbf{x} est dorénavant m_X , et non plus le nombre de capteurs N du réseau comme pour le traitement d'antenne. Le nombre d'échantillons m_X dépend de la durée de l'acquisition d'une réponse suite à l'émission d'un pulse. Pour une même période d'échantillonnage T_e , le nombre d'échantillons enregistrés des réponses doit augmenter pour allonger la portée du système. À chaque période d'échantillonnage T_e , le pulse émis parcourt une distance telle que $d_e = v_p T_e$, où v_p est la vitesse de propagation dans le milieu. Ainsi, la distance de la cible est connue selon $d_k = \frac{\Delta n_k d_e}{2}$, où n_k est le nombre d'échantillons séparant le sommet de l'écho de l'écho initial situé à t_0 . La division par 2 provient du fait qu'un écho retourné au système a parcouru 2 fois la distance séparant le système de la cible.

Cette adaptation est un premier pas des modifications à apporter à l'algorithme pour l'adapter aux contraintes du projet. Comme les signaux à utiliser sont entièrement réels, l'algorithme de détection MUSIC dans le domaine du temps doit être réadapté, ce qui est fait aux sections suivantes.

3.3 Adaptation des signaux et du modèle

Bien que les adaptations dans le domaine du temps de MUSIC permettent de déterminer des délais d'arrivées, les signaux utilisés sont toujours complexes et comportent donc les informations d'amplitude et de phase. Pour ce projet, il est essentiel de distinguer que les signaux

sont uniquement réels, et ne comportent que l'information d'amplitude. Cette distinction par rapport aux anciennes adaptations rend le modèle utilisé dans l'algorithme invalide.

Les signaux réels du projet sont définis par :

$$\mathbf{x} = \sum_{m=1}^M \alpha_m \mathbf{s}(\tau_m) + \mathbf{n} \quad (3.3)$$

où α_m est l'amplitude attribuée à chacun des M échos d'un pulse envoyé ; \mathbf{n} est le vecteur du bruit ; \mathbf{s} est le vecteur du signal contenant l'écho gaussien et τ_m est le délai de l'écho dans la réponse. Toutes les entités servant à représenter le signal dans le domaine temporel sont réelles puisqu'aucune information de phase n'est disponible. Le nombre d'échantillons m_X contenus dans le vecteur \mathbf{x} est spécifié par le système dépendant de la portée souhaitée.

Dans la version classique de MUSIC, à la section 2.4.3 du précédent chapitre, les vecteurs de direction de la matrice $\mathbf{A} = [\mathbf{a}(\phi_1) \dots \mathbf{a}(\phi_M)]$ sont des modèles des sources pour chacune des directions ϕ des M sources de retours incidents. Le vecteur $\mathbf{a}(\phi_k)$ est un modèle de signal parfait qui résulterait d'une source située à un angle ϕ_k . Ces modèles sont bien différents pour des signaux réels. La matrice contenant les vecteurs de direction, qui seront rebaptisés les vecteurs de délais, est définie telle que :

$$\mathbf{A} = [\mathbf{a}(\tau_1) \dots \mathbf{a}(\tau_M)] \quad (3.4)$$

$$(3.5)$$

Les vecteurs de délais ont le même nombre d'échantillons que les réponses aux pulses et représentent un retour parfait d'une source et qui serait retardé d'un délai τ_k qu'on peut exprimer en termes du nombre entier de périodes d'échantillonnage (ou plus simplement en échantillons). En utilisant τ comme paramètre de balayage, on obtient alors $\mathbf{a}(\tau)$ qui est un vecteur modèle de délai utilisé au moment de tracer le pseudospectre, tel qu'expliqué à la section 3.6. La forme du modèle d'écho est celle d'un pulse gaussien, de forme connue, qui aurait subi une réflexion parfaite. La forme est connue puisqu'elle provient de l'émetteur du système. Mathématiquement, un vecteur de délai est représenté par :

$$\mathbf{a}(\tau) = e^{\frac{-(n-\tau)^2}{2\sigma^2}} \quad (3.6)$$

Dans cette équation, σ représente la variance du pulse, n est l'indice correspondant au nombre entier de périodes d'échantillonnage T_e écoulées, τ est le délai de l'écho en termes de périodes d'échantillonnage T_e .

3.4 Formation de la matrice de covariance

La première grande étape de l'algorithme est la création d'une matrice de covariance. Cette étape fut introduite à la section 2.4.1 du précédente chapitre 2. Tout comme pour le traitement

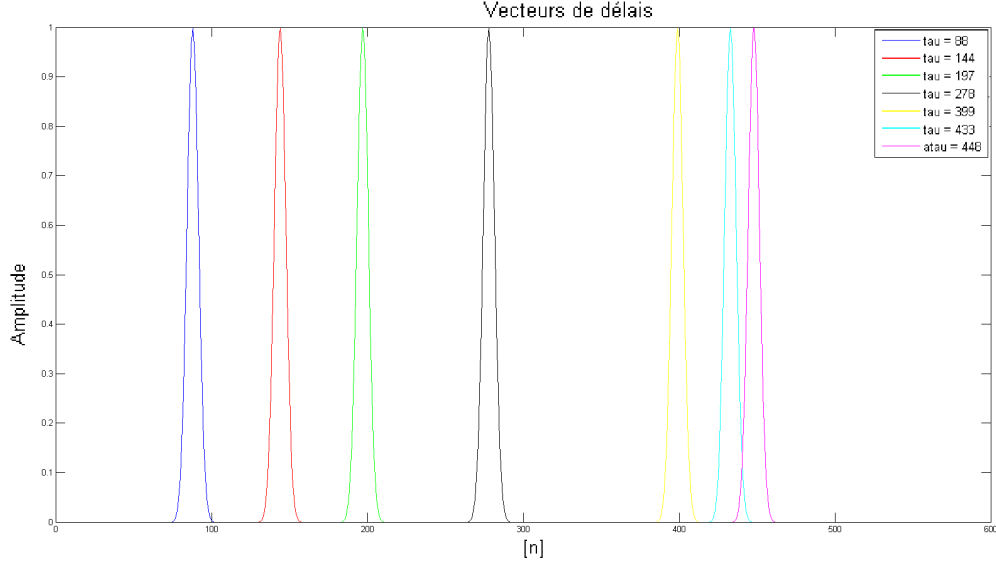


FIGURE 3.2 – Exemples de vecteurs de délais pour différentes valeurs de τ

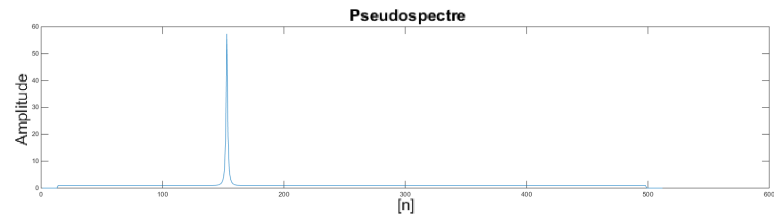
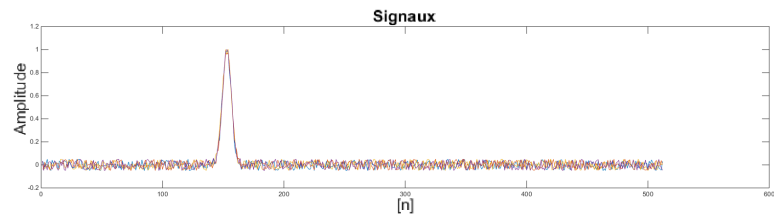
d'antenne, il est impossible d'obtenir cette matrice via l'espérance mathématique : $\mathbf{R}_{XX} = \mathbb{E}[\mathbf{X}\mathbf{X}^T]$. La matrice de covariance est donc approximée, utilisant l'historique des réponses aux pulses tel que

$$\hat{\mathbf{R}}_{XX} = \mathbf{X}\mathbf{X}^T \quad (3.7)$$

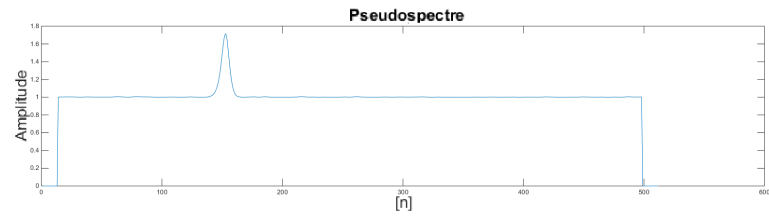
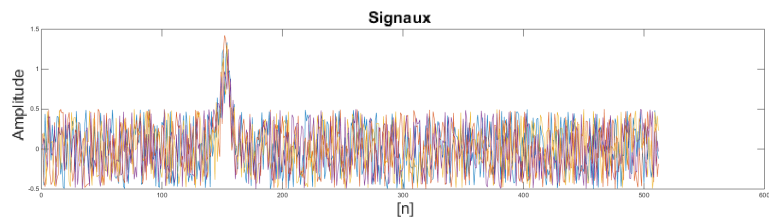
Contrairement au traitement d'antenne, les vecteurs \mathbf{x}_i de l'historique \mathbf{X} sont généralement longs. Dans le cas où le nombre d'échantillons des réponses m_X vaut 512, la matrice de covariance résultante est de dimensions $[512 \times 512]$, pour un total de 262 144 éléments. On voit très bien le problème de complexité calculatoire se présenter au fur et à mesure que la portée, donc la longueur des réponses \mathbf{x}_i augmente.

Toutefois, cette façon de faire est fonctionnelle comme le démontre les résultats obtenus à différents rapports signal à bruit à la figure 3.3.

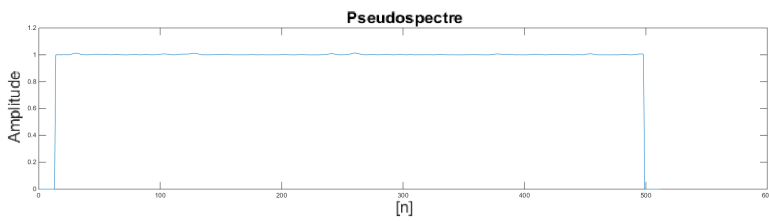
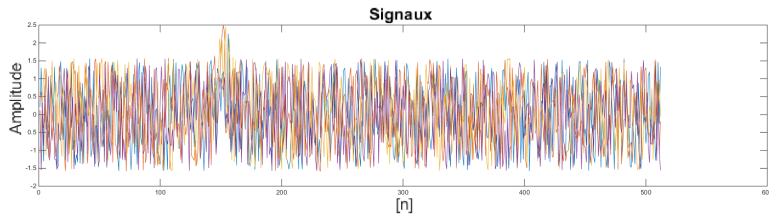
En traitement d'antenne, cette approximation de $\hat{\mathbf{R}}_{XX}$ procure des résultats satisfaisants seulement si les sources sont peu ou pas du tout corrélées, autrement dit, les sources doivent être indépendantes. Dans le cas contraire, les performances de l'algorithme sont fortement affectées. Pour le cas d'estimation du délai des échos, ces derniers sont inévitablement corrélés entre eux, car outre le bruit, les réponses d'un même historique sont des copies les unes des autres. Une modification à la création de la matrice de covariance, appelée lissage spatial permet de restaurer les performances malgré des sources fortement corrélées, voire cohérentes, dans le cas de la détermination de directions d'arrivée en traitement d'antenne. Un des prétraitements de décorrélation souvent utilisé est le lissage de type Forward-only smoothing, tel que décrit dans cet article de Rao et Hari[15]. Ce dernier est transposé dans le domaine des délais d'arrivée pour vérifier s'il modifie les performances.



(a) $SNR = 24.8 \text{ dB}$



(b) $SNR = 4.8 \text{ dB}$



(c) $SNR = -5.2 \text{ dB}$

FIGURE 3.3 – Résultats obtenus de MUSIC à différents SNR, utilisant un historique à 8 signaux

Forward-only smoothing

En transposant ce type de lissage spatial pour des délais d'arrivée, il faut débiter par sous-diviser la matrice de l'historique des réponses \mathbf{X} en P sous-matrices \mathbf{X}_p , contenant chacune L_P vecteurs de réponses. Avec :

$$P = n_X - L_P + 1 \quad (3.8)$$

On comprend qu'un vecteur réponse \mathbf{x}_i peut être présent dans plus d'une sous-matrice \mathbf{X}_p . La première sous-matrice contient $\mathbf{X}_1 = [\mathbf{x}_1 \quad \dots \quad \mathbf{x}_{L_P}]$, la deuxième sous-matrice contient $\mathbf{X}_2 = [\mathbf{x}_2 \quad \dots \quad \mathbf{x}_{L_P+1}]$, et ainsi de suite.

La première étape est d'approximer des sous-matrices de corrélation $\hat{\mathbf{R}}_p$ pour chacune des sous-matrices \mathbf{X}_p via :

$$\hat{\mathbf{R}}_p^{fs} = \frac{1}{L_P} \mathbf{X}_p \mathbf{X}_p^T \quad (3.9)$$

La matrice de corrélation lissée finale correspond à la somme moyennée des sous-matrices $\hat{\mathbf{R}}_p$:

$$\hat{\mathbf{R}}_{XX,fs} = \frac{1}{P} \sum_{p=1}^P \hat{\mathbf{R}}_p^{fs} \quad (3.10)$$

Comme les réponses utilisées pour ce projet sont toujours réelles, cette méthode n'a aucun impact sur les performances, autant au niveau du SNR minimal de détection que sur la séparation des échos partiellement superposés. En fait une légère baisse des performances est observée au niveau de l'amplitude des sommets du pseudospectre, tel que sur le prochain exemple à la figure 3.4, effectué avec 8 signaux d'un SNR à 4.8 dB et $L_P = 4$ pour le lissage spatial forward smoothing.

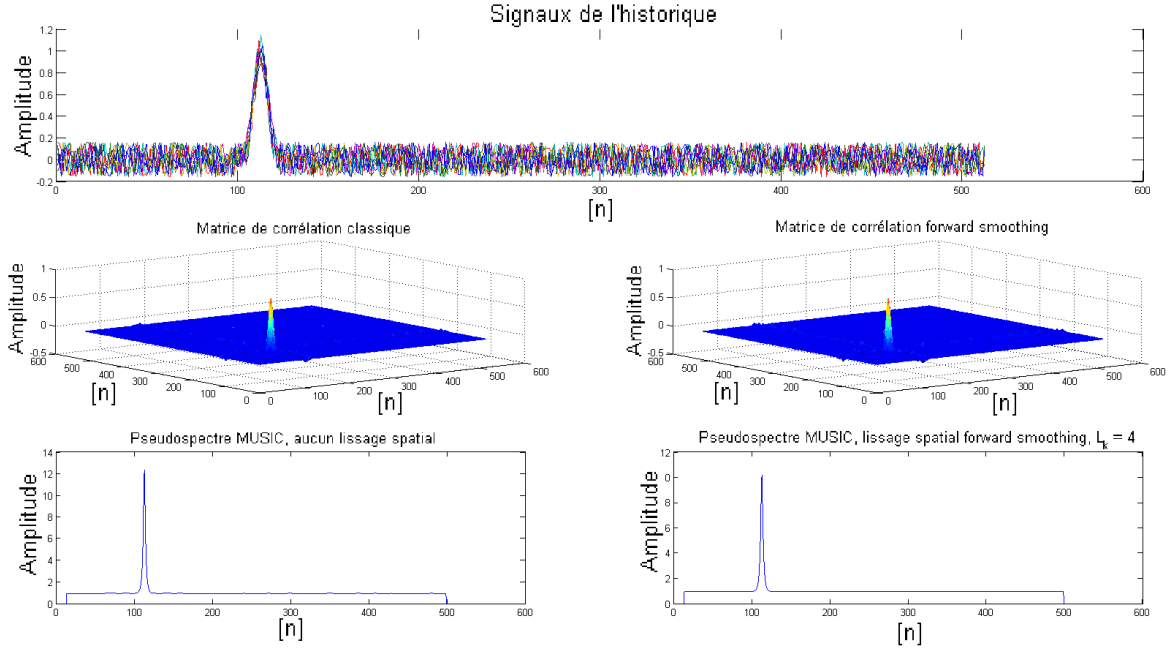


FIGURE 3.4 – Comparaison de MUSIC sans (à gauche) et avec (à droite) *forward smoothing*

En détermination d'angles d'arrivée, le principe du lissage spatial est le même qu'un filtre moyennneur, qui va adoucir la matrice $\hat{\mathbf{R}}_{XX}$. De cette façon, elle s'approche d'une matrice Toeplitz (où tous les éléments d'une même diagonale sont égaux), matrice souhaitée puisqu'elle correspond au cas où toutes les sources sont indépendantes.

Comme la construction de la matrice de covariance $\hat{\mathbf{R}}_{XX}$ pour l'estimation des délais est différente, il s'avère au final que ce sont toujours les mêmes points qui sont multipliés entre eux, il n'y a que très peu de différence entre la création avec et sans lissage spatial forward. Prenons en exemple une matrice $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \mathbf{x}_3]$. Selon la définition $\hat{\mathbf{R}}_{XX} = \frac{1}{n_X} \sum_{i=1}^{n_X} \mathbf{x}_i \mathbf{x}_i^T$, la matrice sans lissage spatial serait calculée comme $0.33333(\mathbf{x}_1 \mathbf{x}_1^T + \mathbf{x}_2 \mathbf{x}_2^T + \mathbf{x}_3 \mathbf{x}_3^T)$. Avec le lissage spatial forward, avec $L_P = 2$ le calcul serait plutôt $0.5(0.5(\mathbf{x}_1 \mathbf{x}_1^T + \mathbf{x}_2 \mathbf{x}_2^T) + 0.5(\mathbf{x}_2 \mathbf{x}_2^T + \mathbf{x}_3 \mathbf{x}_3^T))$. La petite différence sur l'amplitude du pseudospectre vient du fait que les réponses au début et à la fin de la matrice \mathbf{X} , soit \mathbf{x}_1 et \mathbf{x}_3 dans l'exemple précédent, sont moins utilisées que les autres (\mathbf{x}_2 dans l'exemple précédent) lors du lissage forward.

Au final, le prétraitement de décorrélation de type Forward-only ne procure aucun gain. Un autre traitement de décorrélation fut testé, tel qu'expliqué dans un article de D. Grenier & É. Bossé[4], et dont les résultats ne démontrent pas les performances escomptées. Par contre, d'autres méthodes permettent d'obtenir des matrices de covariance rendant l'algorithme plus performant, celles-ci seront présentées à la section 4.1.1 du chapitre 4 concernant les optimisations.

3.5 Décomposition propre et séparation des sous-espaces propres

La décomposition propre de la matrice $\hat{\mathbf{R}}_{XX}$ produit une matrice des valeurs propres $\mathbf{\Lambda}$ et une matrice des vecteurs propres \mathbf{V} . Les valeurs propres obtenues sont disposées sur la diagonale principale de la matrice $\mathbf{\Lambda}$, en ordre décroissant. Les autres valeurs de cette matrice sont des zéros. La matrice \mathbf{V} contient les vecteurs propres sur ses colonnes. Chacun des vecteurs propres est lié à la valeur propre associée. Ainsi, la matrice de covariance est représentée par :

$$\hat{\mathbf{R}}_{XX} = \sum_{i=1}^{m_X} \lambda_i \mathbf{v}_i \mathbf{v}_i^T \quad (3.11)$$

Puisque la matrice $\hat{\mathbf{R}}_{XX}$ est symétrique et qu'elle contient exclusivement des valeurs réelles, ses valeurs et vecteurs propres sont uniquement composés de nombres réels. Si un nombre complexe apparaît, c'est qu'un problème sous-jacent est apparu, et probablement lié à la méthode de création de la matrice de covariance ou à la méthode de décomposition propre.

Dans un premier temps, la décomposition propre est effectuée via la fonction de Matlab™. Une méthode moins coûteuse en calcul est néanmoins présentée à la section 4.2.2 du chapitre 4 sur l'optimisation d'implantation.

3.5.1 Discrimination des valeurs propres sources

Une fois les valeurs propres obtenues, l'étape suivante est de déterminer le nombre de sources d'échos présentes, dans le but ultime de séparer le sous-espace propre source \mathbf{E}_s du sous-espace propre bruit \mathbf{E}_n . Lorsqu'une valeur propre est élevée, la contribution du vecteur propre associé à la matrice est d'autant plus grande. Les valeurs propres liées aux échos retournés sont plus élevées que les valeurs propres liées au niveau de bruit moyen. Les valeurs propres les moins élevées tendent vers zéro. Ainsi, le nombre de sources correspond au nombre de valeurs propres supérieures au seuil des valeurs propres du bruit. Cependant, lorsque plusieurs échos sont présents, se superposent, ou se noient dans le bruit, il devient plus complexe de séparer les valeurs propres des sources des valeurs propres du bruit. La discrimination des valeurs propres est une étape importante puisque la formation des sous-espaces propres en découle, tel qu'expliqué à la section 2.4.3. Le sous-espace à obtenir est celui des sources, soit \mathbf{E}_s , qui inclut les M vecteurs propres liés aux M plus grandes valeurs propres représentant les sources. La mise en place d'un critère de sélection des valeurs propres sources doit être effectuée avec attention car les répercussions d'une erreur de ce critère se font sentir sur plusieurs aspects des performances de l'algorithme.

Problèmes engendrés par le choix de la méthode de construction de la matrice de covariance

Telle que construite à la section précédente 3.4 via $\hat{\mathbf{R}}_{XX} = \mathbf{X}\mathbf{X}^T$, aucune décorrélation des échos n'est faite. Cela se répercute sur le résultat de la décomposition propre. Le tableau 3.1 montre les 5 plus grandes valeurs propres obtenues suite à la décomposition propre de la matrice de covariance, formée à partir de 8 signaux synthétiques d'un SNR égal à 24.8 dB.

Valeurs propres obtenues pour différents nombres d'échos retournés				
Nombre d'échos	1	2	3	4
Valeurs propres				
# 1	52.17	101.30	148.13	197.12
# 2	0.48	0.52	0.48	0.48
# 3	0.47	0.43	0.46	0.47
# 4	0.43	0.41	0.43	0.41
# 5	0.42	0.41	0.40	0.39

TABLE 3.1 – Valeurs propres selon le nombre de sources, $K = 8$, $SNR = 24.8dB$

Comme le démontre le tableau 3.1, il y a toujours une seule valeur propre source, peu importe le nombre d'échos des réponses, pour une matrice de covariance obtenue par $\mathbf{X}\mathbf{X}^T$. Le problème est que le nombre de valeurs propres élevées devrait refléter le nombre d'échos. Par conséquent, la décomposition propre ne sépare pas les sources en les plaçant dans un sous-espace spécifique de dimension égale à leur nombre. Toutes les sources se retrouvent donc dans le même vecteur propre, comme le montre la figure 3.5.

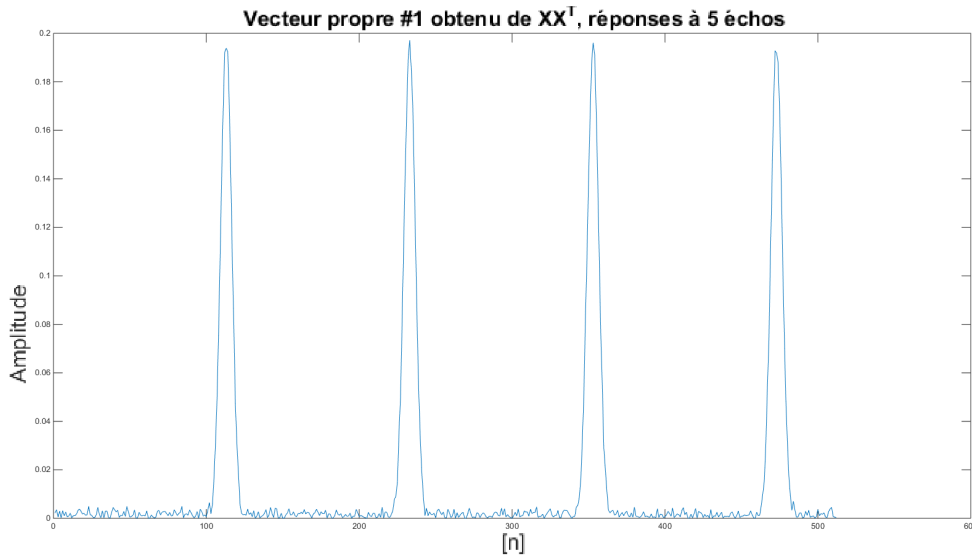


FIGURE 3.5 – Premier vecteur propre pour 4 sources

Lorsque toutes les sources se retrouvent dans le même vecteur, il est impossible d'en connaître leur nombre. De plus, les performances de l'algorithme sont réduites sur beaucoup d'aspects,

notamment sur la discrimination d'échos chevauchés en partie et sur le SNR de détection minimal. Des méthodes de décorrélation des sources lors de la construction de la matrice de covariance seront élaborées à la section 4.1.1.

Critère d'Akaike

Pour séparer les sous-espaces propres source et bruit, il est nécessaire d'évaluer quelles valeurs propres représentent une contribution des sources. Le critère d'Akaike est donc utilisé pour déterminer combien de valeurs propres sont associées à des sources. En traitement d'antenne, un critère de séparation des espaces propres fréquemment utilisé est le critère d'information d'Akaike. Ce critère, et bien d'autres, sont plus longuement expliqués dans ces articles de P.Chen, T.-J. Wu & J. Yang[3], Liavas & Regalia[9], traitant de la sélection du nombre de sources. Ce critère fut testé sur les résultats d'une décomposition propre d'une matrice ayant subi un prétraitement de décorrélation parmi ceux qui seront présentés à la section 4.1.1, où les sources se retrouvent dans des vecteurs propres séparés, comparativement à $\hat{\mathbf{R}}_{\mathbf{X}\mathbf{X}}$ de la section 3.5.1. Le critère fut implémenté comme :

$$AIC(m) = -2 \log(-n_X(m_X - m) \times \log(\sum_{i=m+1}^{m_X} \frac{\lambda_i}{m_X - m})) \quad (3.12)$$

$$m = [0, \dots, m_X] \quad (3.13)$$

où m est la variable du nombre de sources à déterminer. Rappelons ici que m_X et n_X sont les dimensions de la matrice de l'historique de signaux \mathbf{X} , donc m_X est le nombre d'échantillons de réponses et n_X est le nombre de réponses enregistrées dans l'historique. Les valeurs propres, en ordre décroissant, sont représentées par λ_i pour $i = [1, \dots, m_X]$.

Pour le traitement d'antenne, le nombre de sources présentes M est indiqué par la valeur de m qui minimise la fonction $AIC(m)$. Cependant, cette méthode ne fonctionne pas très bien dans le cas de délais d'arrivée sur signaux réels. Comme le démontre la figure 3.6, le nombre de valeurs propres sources n'est pas indiqué par le minimum de la fonction.

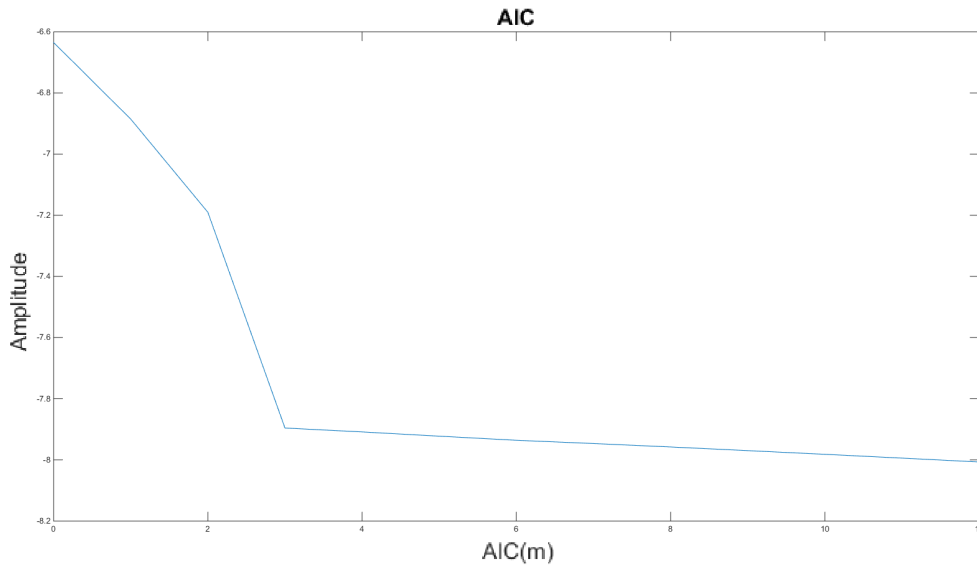


FIGURE 3.6 – Fonction $AIC(m)$ pour des réponses à 3 échos

Cependant, le changement de pente le plus important de la fonction $AIC(m)$ se situe bien vis-à-vis du nombre de valeurs propres sources. Ainsi, le critère devient fonctionnel en affirmant que le nombre de sources M correspond au minima de la dérivée de la fonction $AIC(m)$. La figure 3.7 présente la dérivée de la fonction présentée sur la figure 3.6, et le nombre de sources est bel et bien indiqué par le minimum atteint par cette courbe.

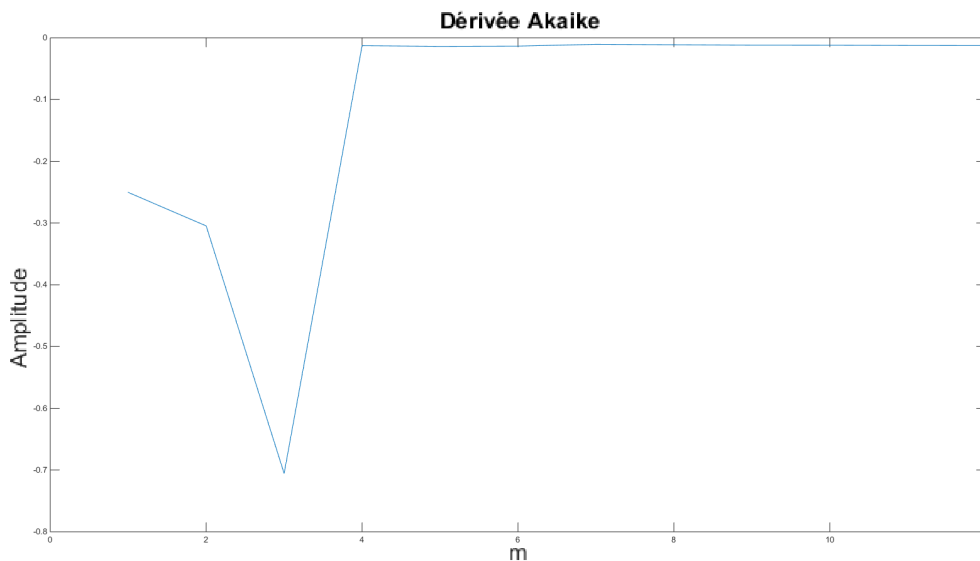


FIGURE 3.7 – Dérivée de la fonction $AIC(m)$ pour des réponses à 3 échos

Bien que fonctionnel, ce critère adapté d'Akaike ne performe pas bien dans certaines circonstances. Lorsque le SNR des signaux diminue, sans s'approcher de la limite de détection de

l'algorithme, le critère perd complètement sa fonctionnalité, malgré qu'une observation rapide des valeurs propres permet de déterminer sans difficulté le nombre de sources. Ce critère, en plus d'exiger beaucoup de calculs, ne procure pas des performances satisfaisantes.

Critère par seuillage simple

Un critère des plus simples pour séparer \mathbf{E}_s et \mathbf{E}_n est d'établir une valeur de seuil. Au-delà du seuil, la valeur propre représente une source et le vecteur propre associé est inclus dans le sous-espace propre source. Sous le seuil, la valeur propre représente du bruit et le vecteur propre associé appartient au sous-espace propre bruit. Les avantages de cette méthode sont certainement sa simplicité et son faible coût en calculs.

En observant plusieurs scénarios et les valeurs propres obtenues, un seuil a été fixé de façon à obtenir un bon compromis sur différents types de situations. Voici donc les résultats obtenus avec ce critère, avec 8 réponses par historique :

1. Pour un nombre d'échos variant entre 1 et 10, disposés sans qu'il y ait de superposition entre eux, et dont les amplitudes sont identiques, le critère fonctionne sur la plage de SNR allant de l'infini à -0.2 dB.
2. Lorsqu'aucun écho n'est présent, le critère s'emballé et donne un nombre de sources M très élevé, il faut donc prévoir un mécanisme de prévention pour ce cas.
3. Lorsque 2 échos se rapprochent, le critère fonctionne jusqu'à un écart de 12 échantillons, ce qui est tout de même raisonnable considérant le prétraitement de décorrélation utilisé. Si l'écart se réduit davantage, le critère confond les 2 sources. Pour que le critère détecte des échos plus fortement chevauchés, il faudrait abaisser la valeur du seuil, ce qui aurait pour conséquence de réduire la plage de SNR fonctionnel du point 1. Autrement dit, sa limite ne serait plus de -0.2 dB, mais deviendrait plus élevée.
4. Lorsque 2 échos bien séparés n'ont pas une amplitude semblable, le critère ignore souvent le plus faible des 2. En fait, à un SNR de 24.8 dB, un rapport de 1 pour 0.63 était la limite pour le critère établi. Lorsque l'amplitude du second écho diminue en-deçà de 63% du premier, il est ignoré.

Ce critère, bien que facilement utilisable, a des limitations importantes, particulièrement lorsque les amplitudes des échos d'une même réponse ont une différence importante. Pourtant, en réalité, la source de moindre amplitude est bel et bien présente. L'utilisateur doit donc choisir avec soin le seuil fixé, en fonction des aspects des performances à améliorer (minimiser les fausses détections, maximiser les cibles trouvées, discriminer des échos chevauchés...).

Conclusion sur la séparation des sous-espaces propres

La séparation des sous-espaces propres \mathbf{E}_s et \mathbf{E}_n est loin d'être évidente. L'élaboration d'un critère de discrimination idéal n'est pas tâche aisée. Plusieurs critères performants furent

élaborés pour du traitement d'antenne, mais il est impossible de réutiliser ces derniers tels quels dans le cadre du présent projet. En effet, la taille des signaux à traiter ainsi que leur nature particulière (aucune information de phase) oblige une adaptation ou tout simplement le développement d'un nouveau critère en fonction des objectifs visés. Dans le cadre de ce projet, une tentative d'élaboration d'un critère de sélection des valeurs propres sources plus performant sera fait à la section 4.1.3.

3.6 Pseudospectre

Le calcul produisant le pseudospectre est exécuté à la suite de la décomposition propre de l'algorithme. Le pseudospectre obtenu affichera des pointes aux endroits où des échos sont détectés dans les signaux de l'historique de départ.

3.6.1 Obtention du projecteur dans l'espace bruit

Suite à la décomposition propre de la matrice de covariance, les espaces propres source \mathbf{E}_s et bruit \mathbf{E}_n ont été séparés après estimation du nombre de sources (échos) présentes. Le pseudospectre pourrait être immédiatement tracé via :

$$P_{MUSIC}(\tau) = \frac{1}{\mathbf{a}(\tau)^T \mathbf{E}_n \mathbf{E}_n^T \mathbf{a}(\tau)} \quad (3.14)$$

mais il est préférable de passer une étape intermédiaire : l'obtention du projecteur dans l'espace bruit. Deux raisons justifient ce choix. La première d'entre elles est que les vecteurs propres de l'espace propre bruit \mathbf{E}_n sont associés à des valeurs propres pouvant être quasi-nulles. Le calcul numérique de ces vecteurs propres peut parfois s'avérer instable. La deuxième raison est une économie de calculs. En effet, comme les vecteurs propres de \mathbf{E}_n sont beaucoup plus nombreux que ceux de \mathbf{E}_s , il est plus coûteux en calculs d'effectuer la multiplication matricielle $\mathbf{E}_n \mathbf{E}_n^T$ que $\mathbf{E}_s \mathbf{E}_s^T$.

Considérant que les vecteurs propres sont orthogonaux et normés, car les matrices de covariance sont toujours symétriques, la relation suivante est valide :

$$\mathbf{E}_n \mathbf{E}_n^T + \mathbf{E}_s \mathbf{E}_s^T = \mathbf{I} \quad (3.15)$$

où \mathbf{I} est une matrice identité de dimensions $[m_X \times m_X]$. Finalement, en posant que le projecteur dans l'espace bruit $\mathbf{P}_n = \mathbf{E}_n \mathbf{E}_n^T$, ce dernier est obtenu via :

$$\mathbf{P}_n = \mathbf{I} - \mathbf{E}_s \mathbf{E}_s^T \quad (3.16)$$

3.6.2 Obtention du pseudospectre

L'obtention du pseudospectre s'effectue selon l'équation :

$$P_{MUSIC}(\tau) = \frac{1}{\mathbf{a}(\tau)^T \mathbf{P}_n \mathbf{a}(\tau)} \quad (3.17)$$

où τ correspond au délai séparant d'échantillonnage, et allant de 1 à m_X . Rappelons que lors de l'acquisition des réponses aux pulses, la période d'échantillonnage T_e correspond à la période de temps écoulée entre 2 échantillons. Le délai entre deux échantillons τ_e d'un modèle y est donc égal, d'où $\tau_e = T_e$; les vecteurs de délais $\mathbf{a}(\tau)$ sont une représentation d'un retour parfait d'un écho dont le retard correspond à τ périodes d'échantillonnage T_e .

Le pseudospectre n'est pas un spectre à proprement dit car l'amplitude des pics est relative et n'est pas une entité, telle que la puissance par exemple. En effet, un pic du pseudospectre peut être plus élevé que son voisin alors que la puissance de la source qu'il représente est plus faible que l'autre. Une plus grande amplitude d'une pointe signifie simplement que l'orthogonalité entre le modèle de délai $\mathbf{a}(\tau)$ et le sous-espace bruit \mathbf{E}_n est meilleure. Une autre façon de voir les choses serait de dire que le contenu du sous-espace propre source \mathbf{E}_s correspond mieux avec le contenu du modèle. Comme \mathbf{P}_n correspond à une matrice dont \mathbf{E}_s est soustrait, il se trouve que plus $\mathbf{a}(\tau)$ tend à ressembler à \mathbf{E}_s , plus il sera orthogonal à \mathbf{E}_n car les deux sous-espaces propres sont parfaitement orthogonaux entre eux.

Une fois le pseudospectre obtenu, il ne reste qu'à effectuer une détection des pics pour connaître l'emplacement des échos à l'échantillon près.

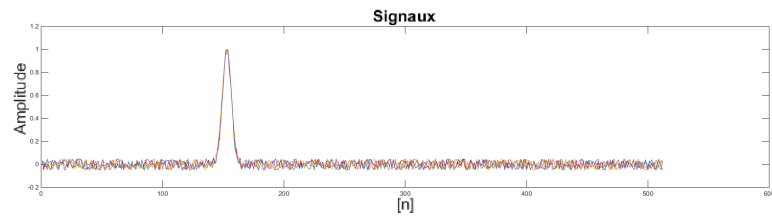
Si le nombre de vecteurs propres appartenant à \mathbf{E}_s est sous-estimé lors de la séparation des sous-espaces, le nombre de pics présents sur le pseudospectre sera réduit, ayant pour conséquence de ne pas détecter certaines cibles. À l'inverse, si le nombre de vecteurs propres inclus dans \mathbf{E}_s est surestimé, des pics supplémentaires apparaissent, pouvant mener à de fausses détections. Toutefois, ces pics supplémentaires sont généralement de plus faibles amplitudes que celui des échos. Lorsque le SNR des échos est à la limite de la détection, les pics des échos et les faux pics peuvent toutefois avoir des amplitudes similaires.

3.7 Exemples de résultats obtenus

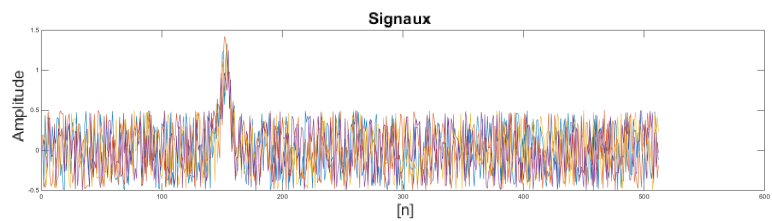
La figure 3.8 montre les résultats obtenus avec l'adaptation de l'algorithme MUSIC pour une seule cible à différents SNRs. Bien que les résultats puissent paraître satisfaisants, il est possible d'obtenir des détections avec des SNRs plus faibles encore.

La figure 3.9 montre ce qui survient au niveau du pseudospectre lorsque les deux échos se chevauchent de plus en plus. Tout d'abord, même si le SNR des réponses est plus élevé que sur le premier graphique de la figure 3.8, l'amplitude des pics sur le pseudospectre est beaucoup plus faible, compliquant la détection lorsque plus d'une cible est présente. Ensuite, on remarque qu'à un écart entre les cibles de 10 échantillons, le pseudospectre a confondu les 2 échos, malgré que les sommets de ceux-ci soient encore clairement distinguables dans les signaux. La haute

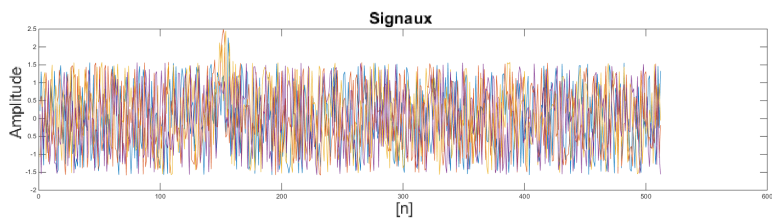
résolution ordinairement attribuée à l'algorithme MUSIC n'est donc pas au rendez-vous.



(a) $SNR = 20\text{ dB}$

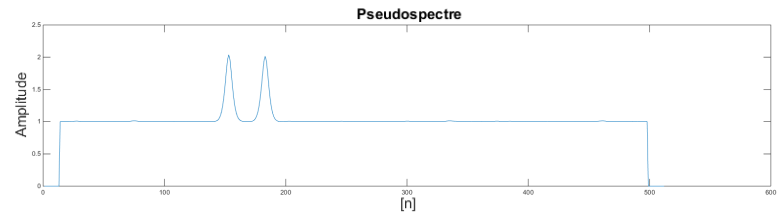
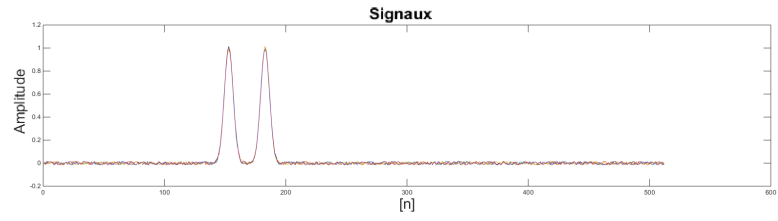


(b) $SNR = 0\text{ dB}$

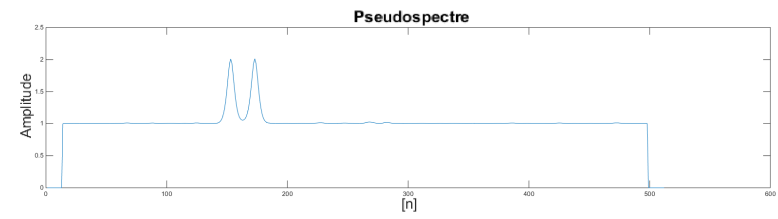
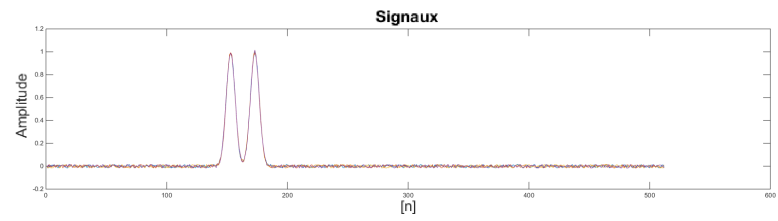


(c) $SNR = -10\text{ dB}$

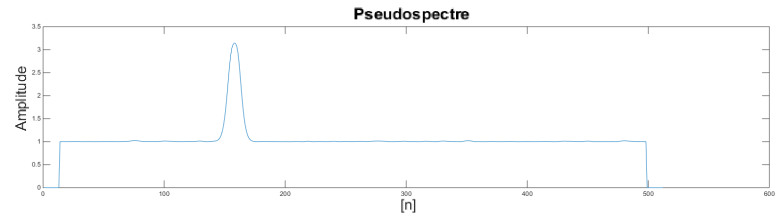
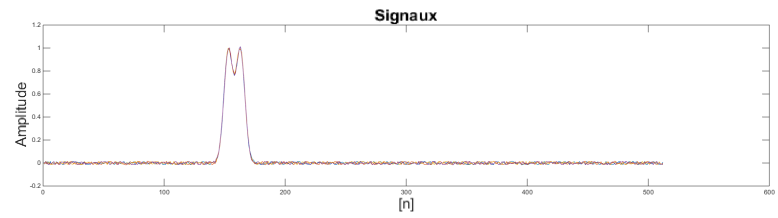
FIGURE 3.8 – Résultats pour une seule cible à différents SNR, $K = 4$



(a) Écart inter-sommets= $30 T_e$



(b) Écart inter-sommets= $20 T_e$



(c) Écart inter-sommets= $10 T_e$

FIGURE 3.9 – Résultats pour deux échos partiellement superposés, $K = 4$, $SNR = 30 \text{ dB}$

3.8 Conclusion sur l’algorithme MUSIC adapté

Le but de ce chapitre était de démontrer qu’il est possible d’élaborer une adaptation temporelle de MUSIC pour des signaux entièrement réels. Cette adaptation se réalise tout d’abord en assumant une définition des signaux où la seule information est l’amplitude de l’enveloppe, aucune information de phase ou sur la porteuse n’est disponible. Les vecteurs de direction traditionnellement utilisés pour le traitement d’antenne deviennent des vecteurs de délais, représentant l’allure du signal reçu sur un capteur dans le temps en fonction du délai d’arrivée plutôt que l’allure du signal sur un réseau de capteurs à un instant précis en fonction de sa direction d’arrivée. Dans cette adaptation, la formation de la matrice de covariance s’effectue de la même façon que dans l’algorithme original, mais donne souvent lieu à des matrices de grandes tailles. Cela est dû au fait que le nombre d’échantillons enregistrés dans le temps est bien souvent plus élevé que le nombre de capteurs en traitement d’antenne, car il faut voir que d’une certaine façon, les périodes d’échantillonnage sont aux délais d’arrivée ce que les capteurs sont au traitement d’antenne par analogie. De par la nature des signaux et de la matrice de covariance résultante, la décomposition propre produit en tout temps des valeurs et vecteurs propres entièrement réels. Finalement, comme les signaux sont définis par des entités réelles, le modèle utilisé pour tracer le pseudospectre a lui aussi été converti à un modèle réel, où le paramètre τ définit le délai du retour du modèle.

Bien que fonctionnel, l’algorithme ne présente pas la pleine mesure de ses performances. Plusieurs modifications et ajouts permettent d’optimiser l’algorithme. Celles-ci seront présentées au prochain chapitre. Elles concernent non seulement les performances, mais également une réduction des coûts d’implantation, car dans son état actuel, l’algorithme a une complexité calculatoire et un coût en ressources trop élevés.

Chapitre 4

Optimisations des performances et implantation

Le chapitre 3 précédent présente une version adaptée en temps de l'algorithme MUSIC pour déterminer des délais d'arrivée sur des signaux exclusivement réels. Bien que cette version soit fonctionnelle, elle ne permet pas à l'algorithme de fournir des résultats à la hauteur de son plein potentiel. En terme de performances, plusieurs modifications permettent d'augmenter son immunité au bruit, de réduire le SNR nécessaire à la détection, de mieux discriminer des échos chevauchés et d'obtenir une précision supérieure. L'aspect de la complexité calculatoire doit également être travaillé pour penser implanter l'algorithme sur un système temps réel à faible coût. Ce chapitre présente en premier lieu les modifications influençant à la hausse les performances. La deuxième section du chapitre présente des pistes de solution pour accélérer l'algorithme et faciliter son implantation réelle.

4.1 Optimisation des performances

Cette section présente les modifications apportées à l'algorithme MUSIC pour que la qualité de ses détections soit meilleure. Les aspects visés sont la précision, le SNR de détection minimal et la séparation d'échos partiellement superposés.

4.1.1 Prétraitement de décorrélation

La toute première étape de l'algorithme est l'élaboration de la matrice de covariance, approximée via $\hat{\mathbf{R}}_{XX}$, présentée à la section 2.4.1. Toutefois, des sources corrélées affectent à la baisse les performances, notamment dû à l'apparition de doublons dans la matrice de covariance. Le lissage spatial était une solution efficace en traitement d'antenne, mais s'est avéré inutile en détermination de délais d'arrivée dû aux signaux réels utilisés. Néanmoins, une version adaptée du lissage spatial est possible. En fait, plusieurs méthodes ont été élaborées et testées. Celles-ci sont présentées ci-dessous, et des tests comparatifs sont disponibles plus loin.

Échos en doublons dans la matrice de covariance

Un doublon est un écho dont le sommet n'est pas centré sur la diagonale principale de la matrice. La présence de doublons dans la matrice de covariance $\hat{\mathbf{R}}_{XX}$ est un paramètre affectant grandement les performances de l'algorithme au niveau de la séparation des échos. Ceux-ci sont indésirables car ils détériorent les performances de l'algorithme. Ils apparaissent uniquement lorsque plus d'un écho est présent dans la réponse au pulse, à cause de la corrélation entre les échos. La figure 4.1 illustre l'apparition des doublons. Les réponses utilisées pour former la matrice de covariance présentaient 2 échos. Les doublons sont indiqués par des flèches. Ceux-ci n'ont pas leurs sommets centrés sur un point de la diagonale principale, alors que c'est le cas pour les échos à conserver.

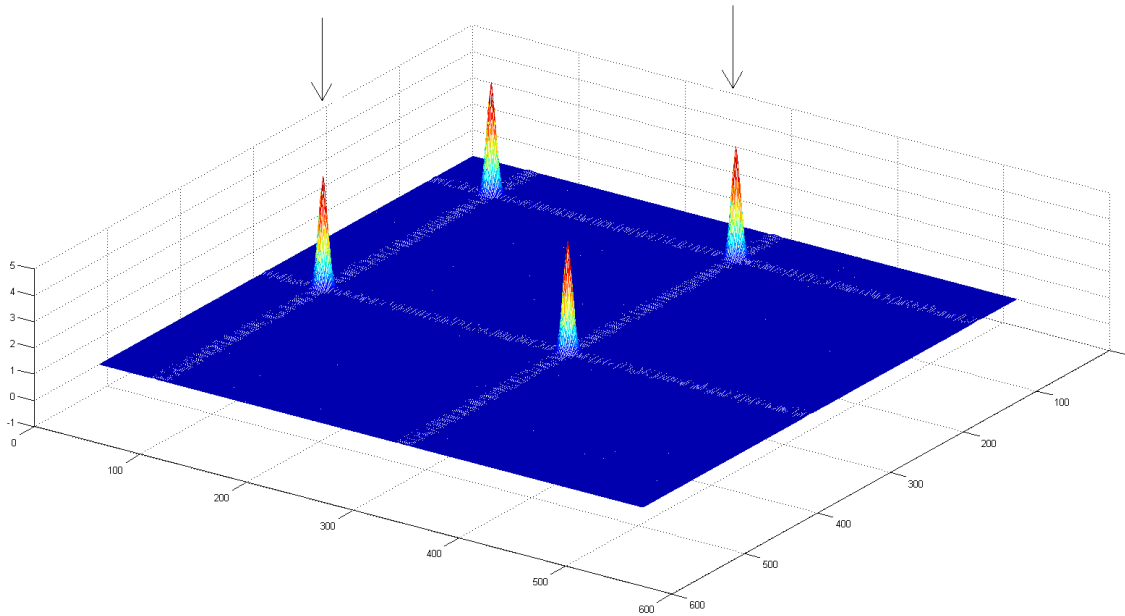


FIGURE 4.1 – Matrice de covariance présentant 2 doublons

Les méthodes de construction de la matrice de covariance ne produisent pas toutes des doublons ayant la même importance. En effet, certaines ont pour effet de réduire l'amplitude de ceux-ci. La figure 4.2 montre en exemple des matrices $\hat{\mathbf{R}}_{XX}$ obtenues pour chacune des méthodes de création, lorsque les échos sont séparés de 12 échantillons, 8 réponses à un SNR de 11.2 dB sont utilisées pour créer les matrices.

Certaines méthodes réduisent ou éliminent les doublons plus que d'autres. La méthode classique via $\mathbf{X}\mathbf{X}^T$ les conserve tous, alors que la méthode classique avec mise à zéro et la superposition décalée sans chevauchement vont conserver uniquement les portions de doublons à l'intérieur de la fenêtre de lissage spatial, les autres doublons seront éliminés. La superposition avec chevauchement produit le même effet, auquel s'ajoute une réduction de l'amplitude des doublons, grâce à l'accumulation. Finalement, la modulation anti-diagonale ne produit aucun doublon et procure les meilleures performances au point de vue de la discrimination d'échos su-

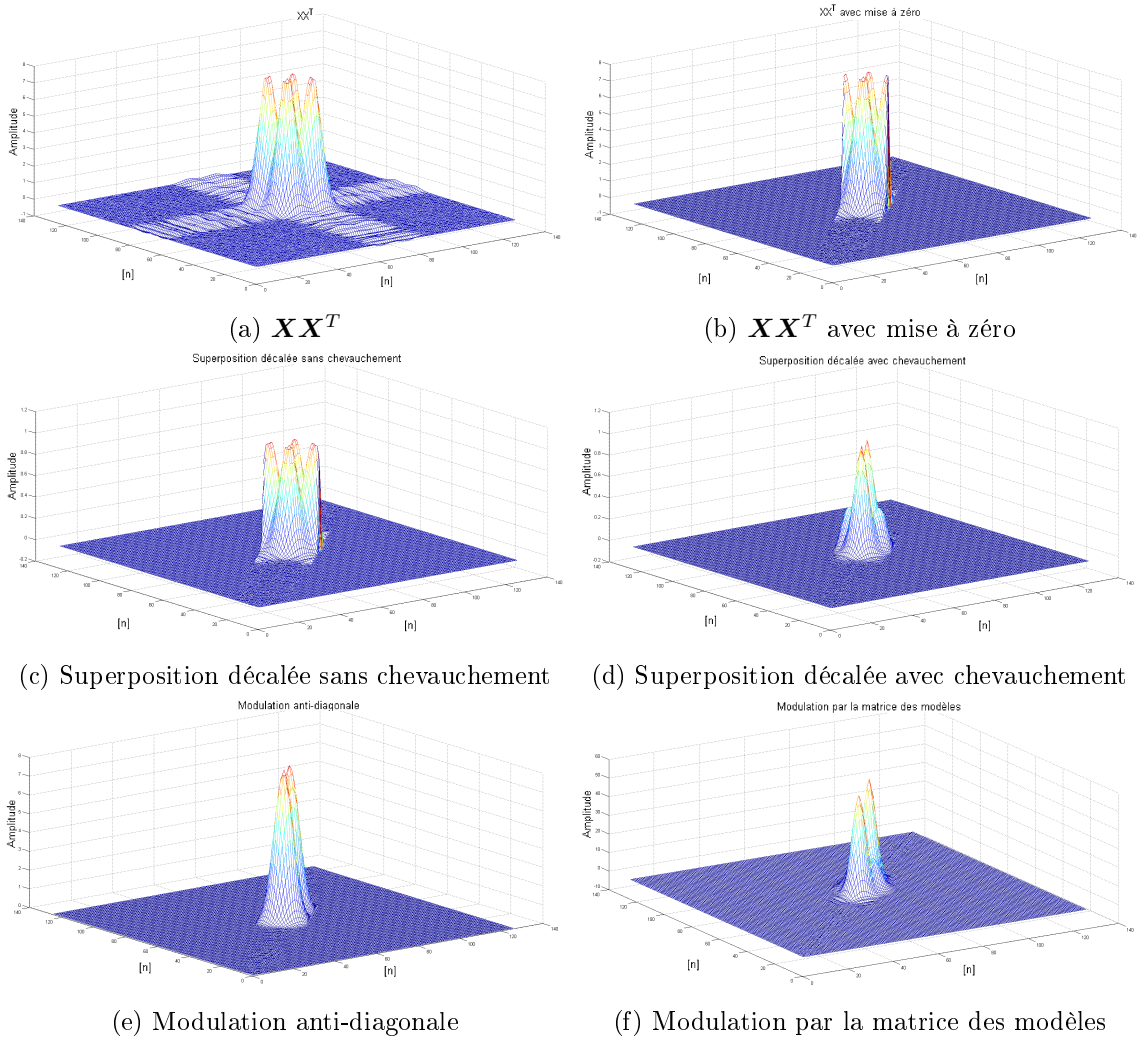


FIGURE 4.2 – Matrices $\hat{\mathbf{R}}_{XX}$ agrandies, 2 échos

perposés. En contre-partie, cette dernière produira des échos déformés dans la matrice lorsque l'écho reçu différera du modèle utilisé pour la modulation. Il peut parfois en résulter des pertes de performances de l'algorithme, notamment au niveau du SNR minimal atteint.

En comparant les graphiques de la figure 4.2 avec les performances atteintes par les méthodes d'élaboration de la matrice de covariance au niveau de la discrimination d'échos superposés, on peut établir un lien certain entre l'importance des doublons dans la matrice et les performances atteintes. Plus les doublons sont apparents, moins la séparation des échos chevauchés sera efficace. Bref, l'élimination des doublons lors de la création est souhaitable.

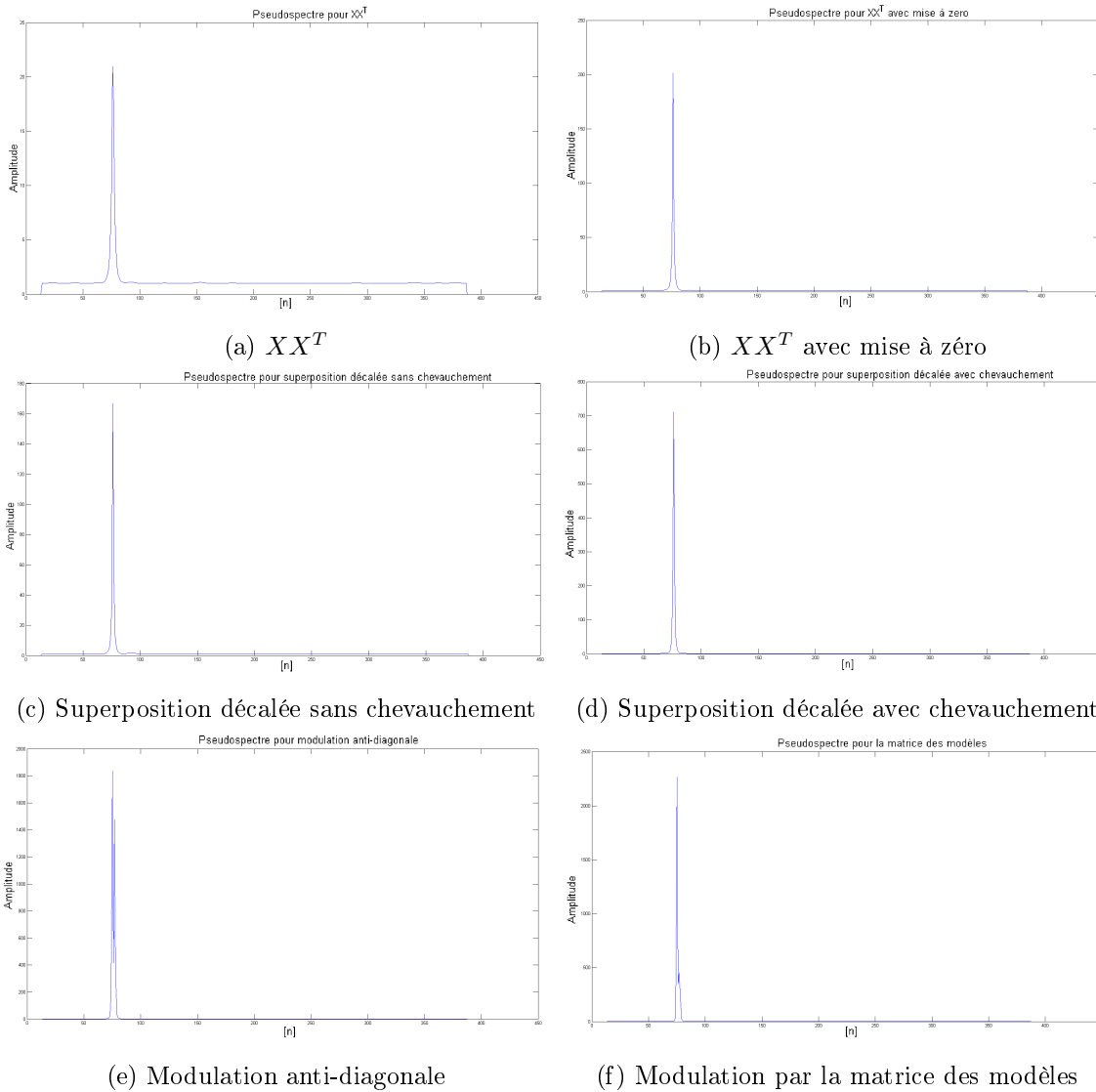


FIGURE 4.3 – Pseudospectres obtenus pour un historique de signaux où les cibles sont séparés de $1.25 m$ pour différentes matrice $\hat{\mathbf{R}}_{XX}$

Comme on peut le voir sur les graphiques de la figure 4.3, seules les matrices sans aucun doublon ou avec une forte atténuation de ceux-ci, créées via la modulation anti-diagonale ou la modulation par la matrice des modèles, engendrent un pseudospectre où 2 sommets sont bien présents.

Un test intéressant pour trouver les limites de l’algorithme quant à la séparation d’échos chevauchés est de combiner 2 matrices à un écho pour obtenir une matrice $\hat{\mathbf{R}}_{XX}$ à deux échos sans doublon. Ce faisant, il n’y a donc plus de corrélation entre les échos. La procédure est la suivante :

- Une première matrice $\hat{\mathbf{R}}_{XX,1}$ est créée via $\mathbf{X}\mathbf{X}^T$, avec le sommet de l'écho positionné au point (113,113), comme sur le graphique de gauche de la figure 4.4.
- Une deuxième matrice $\hat{\mathbf{R}}_{XX,2}$ est aussi créée via $\mathbf{X}\mathbf{X}^T$, avec le sommet de l'écho positionné au point (115,115), comme sur le graphique de droite de la figure 4.4.
- La matrice $\hat{\mathbf{R}}_{XX,3}$ résulte de la sommation de $\hat{\mathbf{R}}_{XX,1} + \hat{\mathbf{R}}_{XX,2}$, ayant 2 échos séparés de deux échantillons, et où aucun doublon n'est présent, comme sur la figure 4.5.

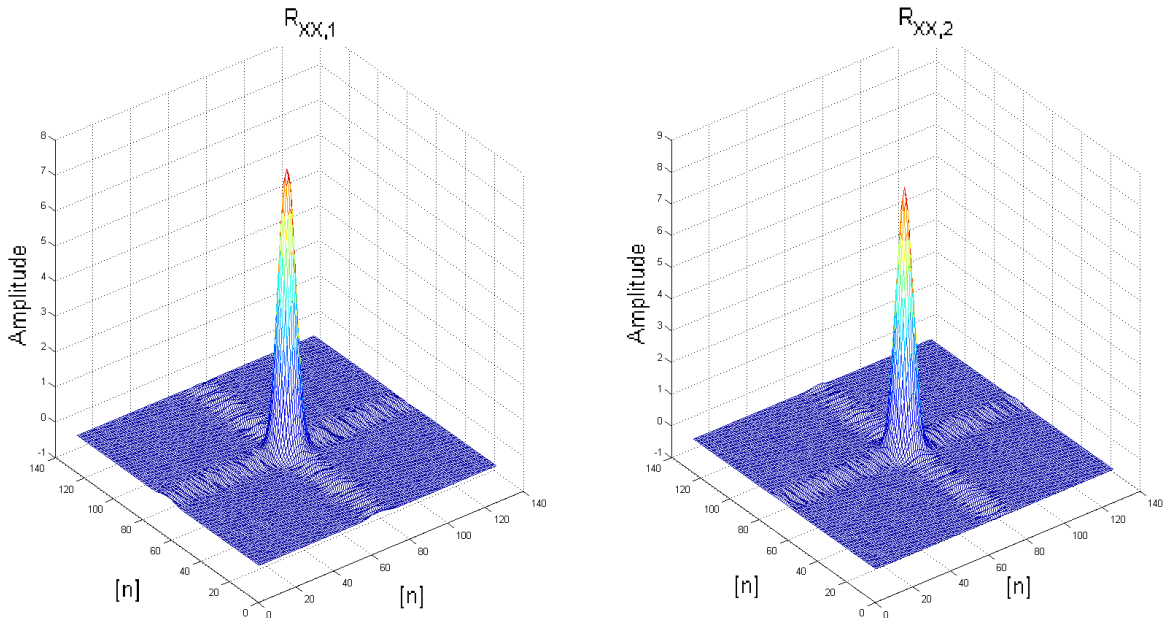


FIGURE 4.4 – $\hat{\mathbf{R}}_{XX,1}$ et $\hat{\mathbf{R}}_{XX,2}$ agrandies autour de l'écho

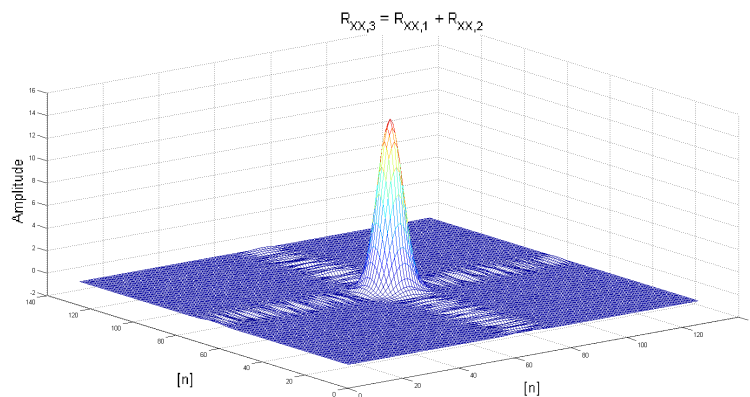


FIGURE 4.5 – $\hat{\mathbf{R}}_{XX,3}$ agrandie autour des échos

Par la suite, la matrice $\hat{\mathbf{R}}_{XX,3}$ fournie au reste de l'algorithme qui produit le pseudospectre de la figure 4.6.

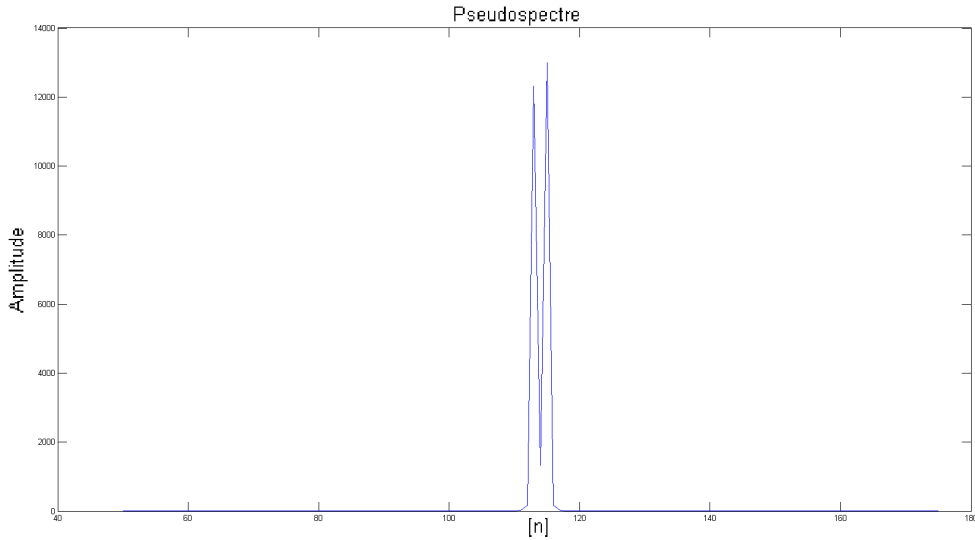


FIGURE 4.6 – Pseudospectre agrandi obtenu pour une matrice à 2 échos séparés par 2 échantillons, aucun doublon

En utilisant des signaux ayant un bon rapport signal bruit, il est possible de distinguer 2 échos séparés de 2 échantillons seulement lorsqu'aucun doublon n'est présent, d'où l'importance de les minimiser et d'idéalement les éliminer. Évidemment, il n'est pas possible de réellement procéder de cette façon puisque lorsque 2 échos sont présents, ils le sont pour toutes les réponses de l'historique.

Superposition décalée

Ce type de prétraitement est mieux adapté aux signaux d'amplitude seulement. Il se décline en 2 versions, telles que présentées ci-dessous. À la section précédente 3.4, le lissage spatial de type forward-only divisait l'historique des réponses \mathbf{X} en sous-matrices contenant un certain nombre de vecteurs pour créer des sous-matrices de covariance \hat{R}_p . Avec le lissage de type superposition décalée, le principe de sous-division de \mathbf{X} est toujours présent, mais sur les lignes plutôt que sur les colonnes. Suite à ce prétraitement, les performances de MUSIC sont améliorées.

Avec chevauchement

Cette étape de prétraitement débute par le découpage des réponses \mathbf{x}_k en sous-sections $\mathbf{x}_{p,k}$ de longueur L_p . Celles-ci doivent être au moins égales à la longueur d'un pulse. P sous-sections $\mathbf{x}_{p,k}$ sont ainsi obtenues, où

$$P = m_X - L_p + 1 \quad (4.1)$$

Ainsi, le découpage de la réponse k de l'historique \mathbf{x}_k serait tel que

$$\mathbf{x}_{1,k} = \begin{bmatrix} x_{1,k} \\ \vdots \\ x_{L_p,k} \end{bmatrix} \quad \mathbf{x}_{2,k} = \begin{bmatrix} x_{2,k} \\ \vdots \\ x_{L_p+1,k} \end{bmatrix} \quad \dots \quad \mathbf{x}_{P,k} = \begin{bmatrix} x_{P,k} \\ \vdots \\ x_{m_X,k} \end{bmatrix}$$

où m_X est le nombre d'échantillons de chacune des réponses.

L'étape suivante consiste à créer des vecteurs $\mathbf{s}_{p,k}$ de longueur m_X pour effectuer un lissage spatial lors de la création de la matrice de covariance. Ces signaux sont définis par :

$$\mathbf{s}_{1,k} = \begin{bmatrix} \mathbf{x}_{1,k} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{s}_{2,k} = \begin{bmatrix} 0 \\ \mathbf{x}_{2,k} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \dots \quad \mathbf{s}_{P,k} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \mathbf{x}_{P,k} \end{bmatrix}$$

L'étape finale est la création de la matrice de covariance approximée par :

$$\hat{\mathbf{R}}_{XX} = \frac{1}{L_p \times n_X} \sum_{p=1}^P \sum_{k=1}^{n_X} \mathbf{s}_{p,k} \mathbf{s}_{p,k}^T \quad (4.2)$$

où P est le nombre de sous-sections, et $n_X = K$, soit le nombre d'épreuves enregistrées dans l'historique \mathbf{X} .

Voici un exemple à la figure 4.7, à échelle réduite, démontrant la distribution des accumulations dans la matrice. La matrice en exemple est de dimension $[10 \times 10]$. Les sections ont une longueur L_p de 3 échantillons. La portion noire représente uniquement des zéros. Ensuite, la portion grise foncée ne comporte pas d'accumulation alors que la portion grise pâle comporte 1 accumulation. La portion blanche comporte 2 accumulations, soit la somme de 3 sous-matrices $\hat{\mathbf{R}}_{XX,p}$. Évidemment, le nombre de régions ainsi que le nombre d'accumulations est tributaire de la valeur de L_p , mais le principe reste le même.

La matrice suivante démontre bien l'emplacement des sous-matrices $\hat{\mathbf{R}}_{XX,p}$ dans la matrice de covariance. L'encadré rouge correspond à $p = 1$, vert pour $p = 2$, blue pour $p = 3$ et jaune pour $p = 4$. Cette matrice de covariance n'est pas à l'échelle des signaux du système. Son paramètre de lissage spatial L_p vaut 3, donc les sous-matrices ont des dimensions de $[3 \times 3]$ échantillons.

Comme certains éléments des matrices obtenues par $\mathbf{s}_{p,k} \mathbf{s}_{pki}^T$ sont additionnés avec une partie des éléments de $\mathbf{s}_{p+1,k} \mathbf{s}_{p+1,k}^T$, c'est pourquoi ce lissage par superposition décalée est dit avec chevauchement. En utilisant cette méthode, la matrice de covariance $\hat{\mathbf{R}}_{XX}$ est majoritairement emplie par des zéros. En effet, elle ne possède que $2L_p - 1$ diagonales où les éléments sont non-nuls, soit une diagonale principale et $L_p - 1$ sous-diagonales de part et d'autres. La figure 4.9 est un exemple de comparaison entre ce type de lissage et aucun lissage.

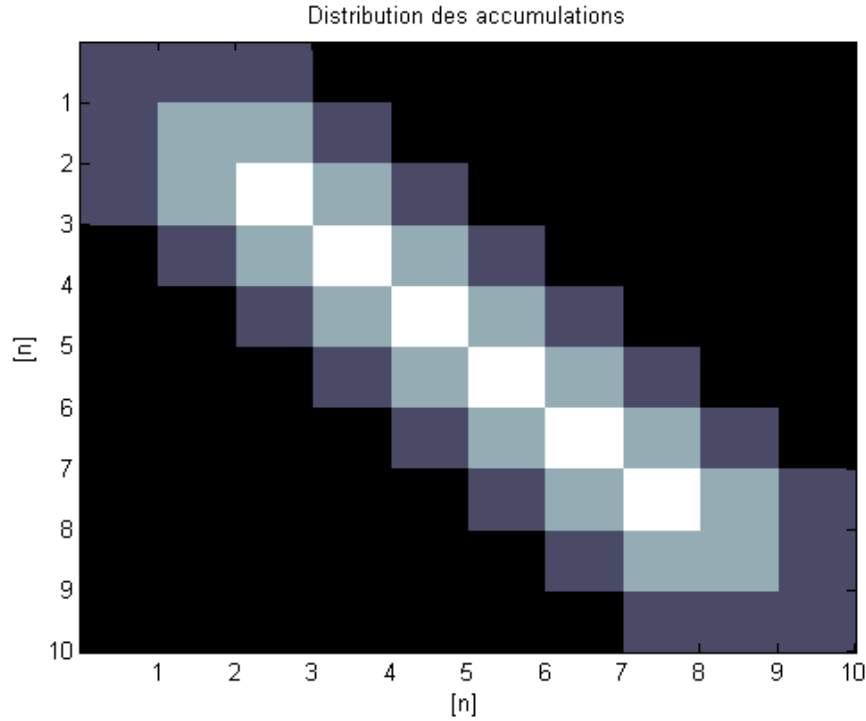


FIGURE 4.7 – Exemple de distribution de la puissance suite à une superposition décalée avec chevauchement

$$\begin{bmatrix}
 R_{1,1} & R_{1,2} & R_{1,3} & R_{1,4} & R_{1,5} & R_{1,6} \\
 R_{2,1} & R_{2,2} & R_{2,3} & R_{2,4} & R_{2,5} & R_{2,6} \\
 R_{3,1} & R_{3,2} & R_{3,3} & R_{3,4} & R_{3,5} & R_{3,6} \\
 R_{4,1} & R_{4,2} & R_{4,3} & R_{4,4} & R_{4,5} & R_{4,6} \\
 R_{5,1} & R_{5,2} & R_{5,3} & R_{5,4} & R_{5,5} & R_{5,6} \\
 R_{6,1} & R_{6,2} & R_{6,3} & R_{6,4} & R_{6,5} & R_{6,6}
 \end{bmatrix}$$

FIGURE 4.8 – Représentation de l'emplacement des sous-matrices $\hat{\mathbf{R}}_{XX,p}$ dans la matrice de covariance $\hat{\mathbf{R}}_{XX}$, superposition décalée avec chevauchement

Tel que le démontre la comparaison des pseudospectres, le lissage par superposition décalée procure des meilleurs résultats, notamment en supportant un SNR plus faible. Des comparaisons plus détaillées entre les méthodes de création de la matrice de covariance sont disponibles plus loin, à la section 4.1.2.

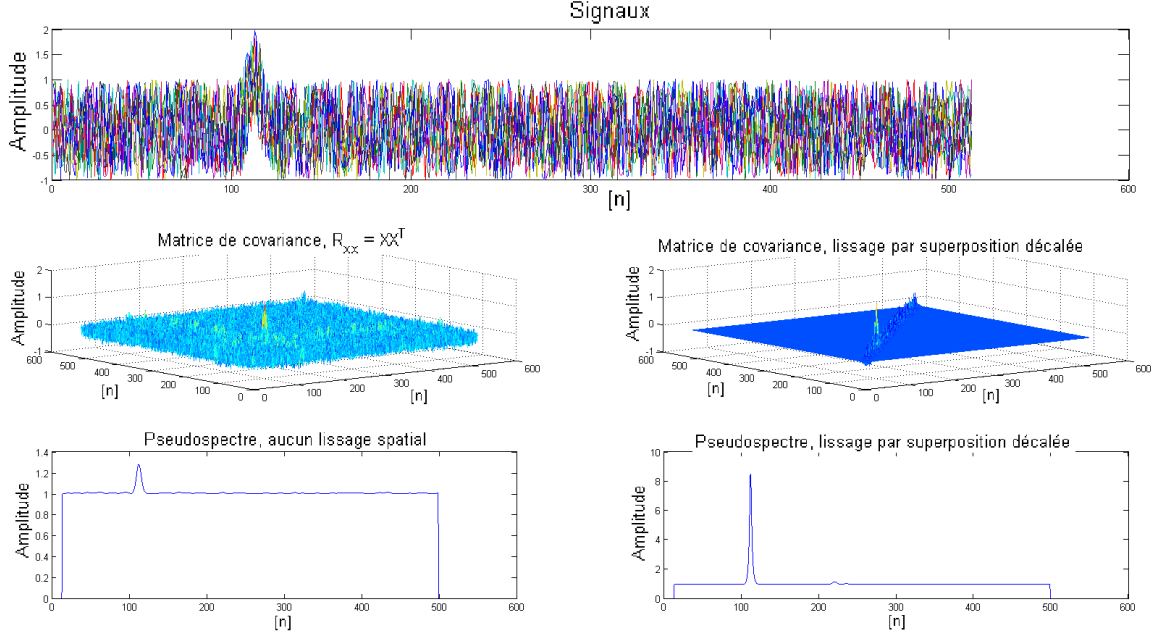


FIGURE 4.9 – Exemple de superposition par chevauchement, 8 signaux dans l’historique à -1.2 dB, 1 seul écho, $L_p = 16$ pour le lissage

Sans chevauchement

Cette méthode utilise le même principe que la méthode de décorrélation par superposition décalée avec chevauchement sauf que les éléments non-nuls déjà présents dans $\hat{\mathbf{R}}_{XX}$ via $\mathbf{s}_{p,k}\mathbf{s}_{p,k}^T$ ne sont plus additionnés avec les éléments de $\mathbf{s}_{p+1,k}\mathbf{s}_{p+1,k}^T$. Seuls les éléments obtenus par cette dernière correspondant à des positions à éléments nuls de $\hat{\mathbf{R}}_{XX}$ sont ajoutés.

À la section précédente 4.1.1 sur la superposition décalée avec chevauchement, les vecteurs $\mathbf{s}_{p,k}$ étaient remplis de 0, sauf pour une sous-section égale à une portion de longueur L_p du vecteur de réponse \mathbf{x}_k . Ces vecteurs sont également le point de départ pour la version sans chevauchement de la superposition décalée. Avec ces vecteurs, des matrices de covariance intermédiaires sont créées selon :

$$\hat{\mathbf{R}}_{XX,p} = \sum_{k=1}^{n_X} \mathbf{s}_{p,k}\mathbf{s}_{p,k}^T \quad (4.3)$$

La matrice de covariance approximée finale $\hat{\mathbf{R}}_{XX}$ est une sommation des matrices intermédiaires. Toutefois, à chacune des sommations d’une matrice intermédiaire, seuls les éléments de la matrice finale ayant une valeur de 0 subissent la sommation. Ceux ayant une valeur non nulle la conserve et ne subissent pas la sommation. L’obtention de la matrice de covariance

approximée finale peut s'exprimer ainsi :

$$\hat{\mathbf{R}}_{XX}(i, j) = \begin{cases} \hat{\mathbf{R}}_{XX,p}(i, j) & \text{lorsque } \sum_{k=1}^{p-1} \hat{\mathbf{R}}_{XX,k}(i, j) = 0 \\ \hat{\mathbf{R}}_{XX}(i, j) & \text{autrement} \end{cases} \quad (4.4)$$

La matrice ci-dessous illustre, à une échelle réduite, comment les blocs calculés sont disposés dans la matrice de covariance. Le paramètre L_p de lissage est de 3 pour cet exemple. Le carré en rouge est le bloc de départ, soit $p = 1$. Par la suite, les blocs verts ($p = 2$), bleus ($p = 3$) et jaunes ($p = 4$) sont successivement ajoutés. Comme on peut le voir, aucun chevauchement n'est présent d'un bloc à l'autre.

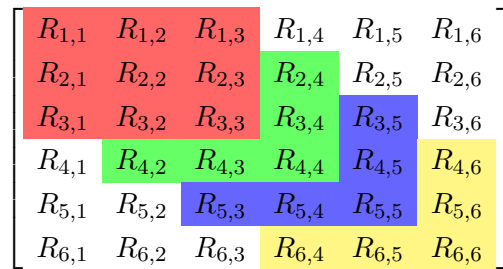


FIGURE 4.10 – Représentation de l'emplacement des éléments des sous-matrices $\hat{\mathbf{R}}_{XX,p}$ dans la matrice de covariance $\hat{\mathbf{R}}_{XX}$, superposition décalée sans chevauchement

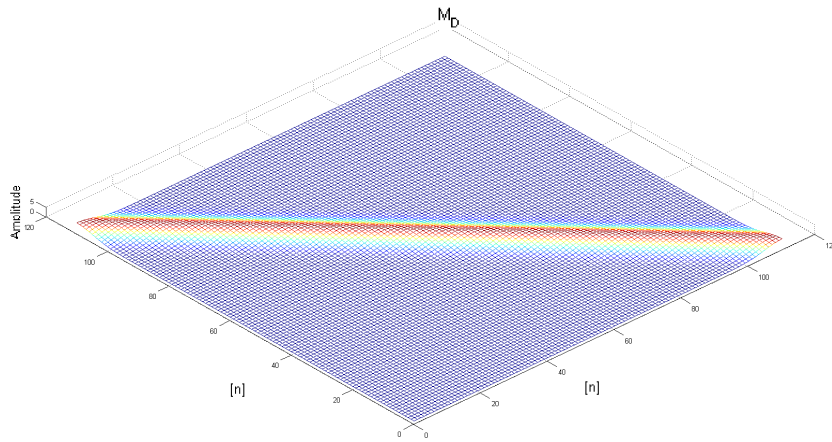
Cette version sans chevauchement nécessite moins de calculs que la version avec chevauchement car les éléments non-nuls ne subissent pas une nouvelle sommation.

Modulation anti-diagonale

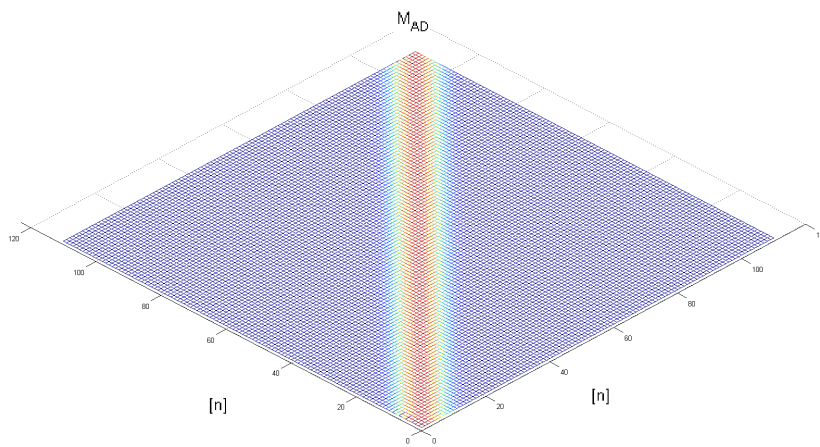
Cette méthode se veut une solution aux problèmes engendrés par l'apparition d'échos en doublons, discutés à la section 4.1.1. Jusqu'à présent, toutes les méthodes de création de la matrice $\hat{\mathbf{R}}_{XX}$ détaillées placent la somme des réponses au carré sur la diagonale principale. En effet, lors de la présentation initiale de MUSIC pour des angles d'arrivée à la section 2.4.3 du chapitre 2, l'approximation de la matrice de covariance était définie par

$$\hat{\mathbf{R}}_{XX} = \frac{1}{n_X} \sum_{k=1}^{n_X} \mathbf{x}_k \mathbf{x}_k^H \quad (4.5)$$

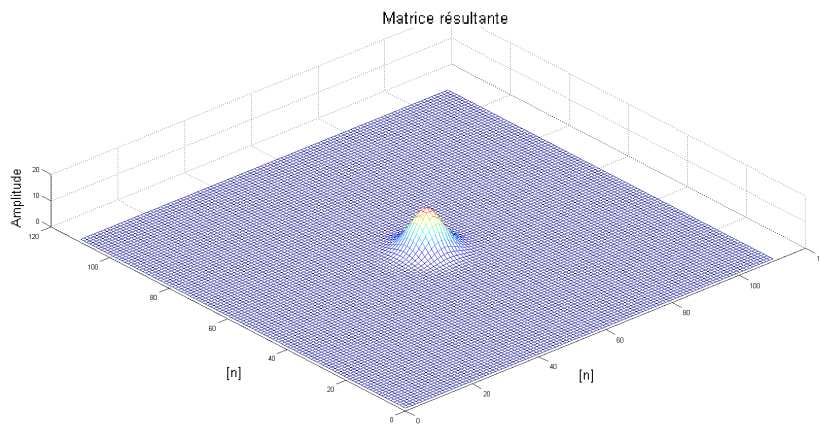
En détaillant le calcul pour chacun des points de la diagonale principale de la matrice de covariance, on observe que chacun de ces éléments est une somme des éléments au carré des signaux. Par exemple, le premier élément de la matrice vaut $\hat{\mathbf{R}}_{XX}(1, 1) = \mathbf{x}_1^2(1) + \dots + \mathbf{x}_{n_X}^2(1)$. Cette affirmation est uniquement valide pour les matrices de covariance créées pour une application de détermination des délais d'arrivée sur signaux réels.



(a) Matrice où la réponse au carré est répliquée sur les diagonales



(b) Matrice où la réponse au carré est répliquée sur les anti-diagonales



(c) Matrice résultante de $M_D \circ M_{AD}$

FIGURE 4.11 – Matrices en exemple du principe voulant que les échos des anti-diagonales d’une matrice de covariance ait la forme du pulse au carré

En observant les matrices de covariance créées, nous avons réalisé que la forme observée d’un

écho sur l'anti-diagonale de celui-ci est celle d'un pulse au carré, exactement comme sur la diagonale. Un exemple visuel de cette constatation est présenté à la figure 4.11, où un écho en 3 dimensions est obtenu via une multiplication d'Hadarnard (point à point) entre une première matrice \mathbf{M}_D , où les diagonales sont des copies de la réponse au carré, et une deuxième matrice \mathbf{M}_{AD} , où les anti-diagonales sont des reproduction de la réponse au carré. La matrice résultante affiche bel et bien un écho semblable à celui obtenu selon $\mathbf{X}\mathbf{X}^T$.

Afin de bien visualiser que la forme de l'écho sur le sens anti-diagonal est la même que sur le sens diagonal, une matrice de covariance est créée à partir d'un historique de 4 réponses avec un bon SNR selon $\hat{\mathbf{R}}_{XX}$. La figure 4.12 montre que la forme de l'écho sur l'anti-diagonale est bel et bien la même que celle sur la diagonale et que celle des réponses au carré sommées.

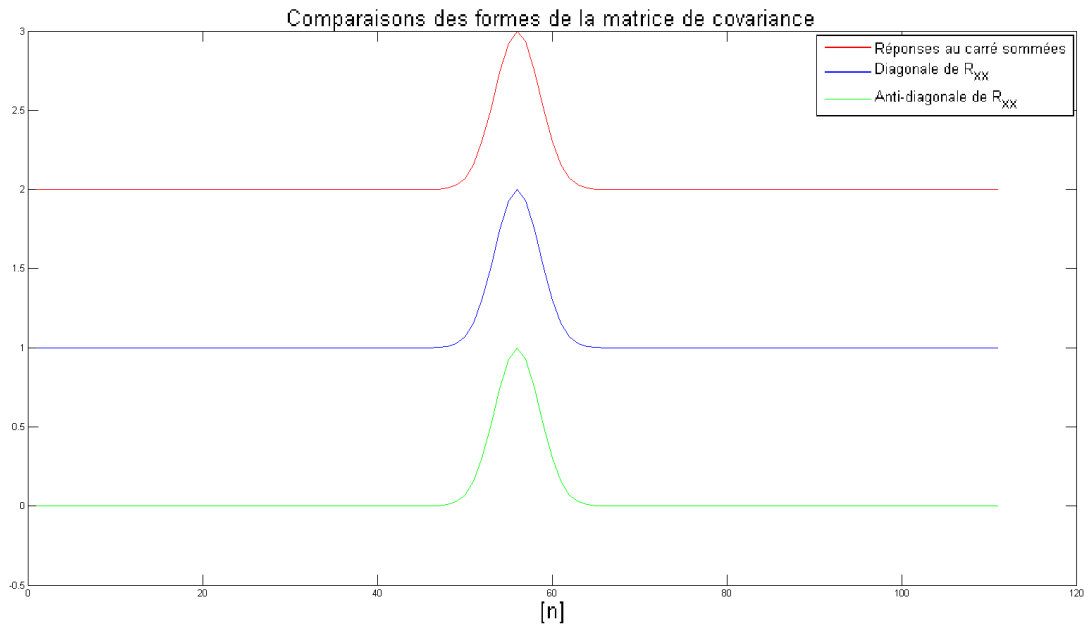


FIGURE 4.12 – Comparaison des formes d'un écho

Afin de conserver les propriétés originales de l'écho dans la matrice de covariance tout en éliminant les doublons, la création de la matrice de covariance par la modulation anti-diagonale va conserver la forme au carré de l'écho dans la matrice. La formation de celle-ci débute d'abord par la sommation des réponses au carré de l'historique.

$$\mathbf{x}_s = \frac{1}{n_X} \sum_{k=1}^{n_X} \mathbf{x}_k^2 \quad (4.6)$$

À partir de ce dernier, une séquence interpolée est obtenue, telle que :

$$\mathbf{x}_{s,odd}(n) = \frac{\mathbf{x}_s(n) + \mathbf{x}_s(n+1)}{2} \quad (4.7)$$

$$n = [1, \dots, m_X - 1] \quad (4.8)$$

Avant de placer les éléments dans la matrice de covariance, il est nécessaire de définir les notations utilisées pour ses diagonales. Celles-ci seront définies comme des vecteurs colonnes \mathbf{d}_i , disposés dans la matrice de cette façon :

$$\hat{\mathbf{R}}_{XX} = \begin{bmatrix} \mathbf{d}_0(1) & \mathbf{d}_1(1) & \dots & & \mathbf{d}_{m_X-1}(1) \\ \mathbf{d}_{-1}(1) & \dots & & & \\ \dots & & \dots & \dots & \dots \\ & & & \dots & \mathbf{d}_1(m_X - 1) \\ \mathbf{d}_{-(m_X-1)}(1) & \dots & \mathbf{d}_{-1}(m_X - 1) & \mathbf{d}_0(m_X) & \end{bmatrix}$$

La prochaine étape consiste à disposer la séquence \mathbf{x}_s sur les diagonales d'indices pairs de la matrice $\hat{\mathbf{R}}_{XX}$ et la séquence $\mathbf{x}_{s,odd}$ sur les diagonales d'indices impairs. Comme les diagonales n'ont pas toutes la même longueur, il est nécessaire d'éliminer un échantillon à chaque fois qu'on s'éloigne d'un indice de la diagonale principale. Comme ce ne serait pas symétrique de supprimer un échantillon à la séquence \mathbf{x}_s à chaque itération sur les diagonales, c'est pour cette raison que la séquence $\mathbf{x}_{s,odd}$ a été créée. Ainsi, pour les diagonales avec i paire :

$$\mathbf{d}_i = [\mathbf{x}_s(1 + 0.5|i|) \quad \dots \quad \mathbf{x}_s(m_X - 0.5|i|)] \quad (4.9)$$

$$i = [-(m_X - 2) \quad -(m_X - 4) \quad \dots \quad 0 \quad \dots \quad m_X - 4 \quad m_X - 2] \quad (4.10)$$

et pour les diagonales avec i impaires :

$$\mathbf{d}_i = [\mathbf{x}_{s,odd}(0.5 + 0.5|i|) \quad \dots \quad \mathbf{x}_{s,odd}(m_X - 0.5 - 0.5|i|)] \quad (4.11)$$

$$i = [-(m_X - 1) \quad \dots \quad -1 \quad 1 \quad \dots \quad m_X - 1] \quad (4.12)$$

À ce moment, la matrice a l'allure montrée à la figure 4.13 :

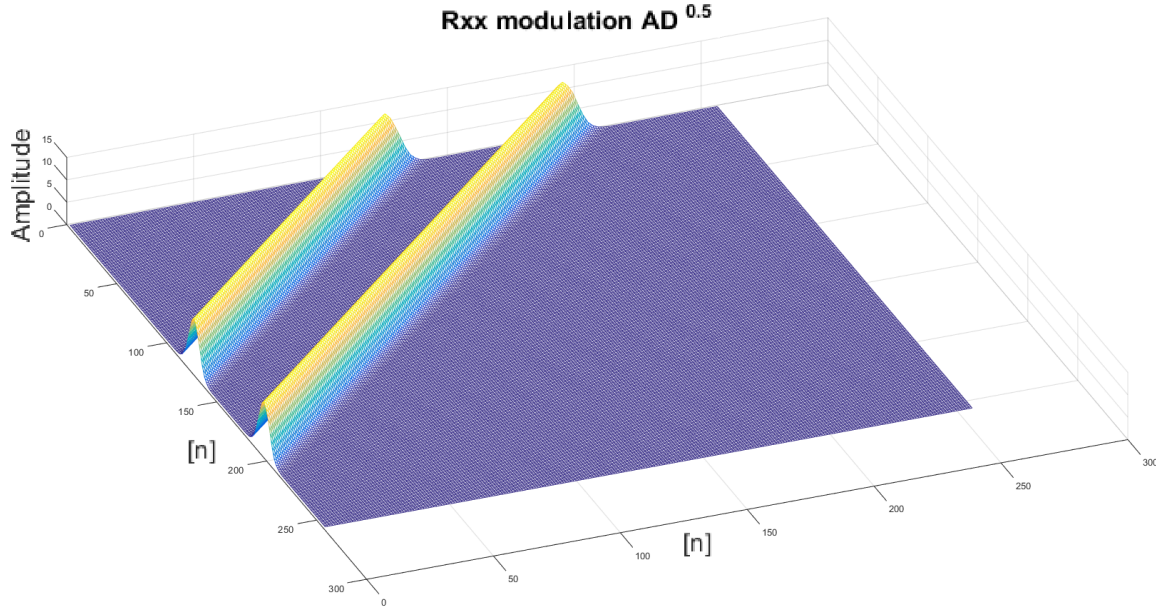


FIGURE 4.13 – État de la matrice $\hat{\mathbf{R}}_{XX}$ avant la modulation anti-diagonale, 8 signaux à $SNR = 4.8$, 2 échos

Le but est d'obtenir une matrice où les échos se retrouvent centrés sur un point de la diagonale principale, tout comme la méthode classique via $\hat{\mathbf{R}}_{XX} = \mathbf{X}\mathbf{X}^T$, mais avec le lissage. La modulation de l'amplitude des diagonales est le moyen d'y arriver. En effet, comme la forme des échos est connue, soit une gaussienne dont les paramètres sont également connus, il faut qu'un écho dans la matrice ait cette forme au carré selon toutes les directions. Pour ce faire, les diagonales sont modulées en amplitude via un vecteur de modulation multiplié point à point avec les anti-diagonales de la matrice formée précédemment. La forme modulante gaussienne doit avoir l'amplitude de son sommet égale à l'unité. La forme modulante appliquée aux diagonales est représentée par le vecteur \mathbf{g} . Celle-ci est très semblable aux vecteurs de délais $\mathbf{a}(\tau)$ mais en diffère essentiellement sur deux aspects. Tout d'abord, la forme pulse contenue dans le vecteur \mathbf{g} est toujours centrée dans ce dernier. Ensuite, la longueur utilisée de \mathbf{g} est variable en fonction de l'anti-diagonale à moduler. Lors de la modulation de l'anti-diagonale principale, le vecteur \mathbf{g} a un nombre d'échantillons utiles à la modulation maximale, tandis qu'en modulant les toutes dernières anti-diagonales secondaires, soit les coins de la matrice, un seul échantillon de \mathbf{g} est utilisé. Le vecteur \mathbf{g} est représenté selon :

$$\mathbf{g} = e^{\frac{-i^2}{2\sigma^2}} \quad (4.13)$$

$$i = [-(m_X - 1), \dots, m_X - 1] \quad (4.14)$$

où i correspond à l'indice de la diagonale \mathbf{d}_i . Chacune des diagonales est multipliée par l'amplitude du point de la gaussienne correspondant à sa position. Ainsi, pour une forme modulante

correspondant au pulse lui-même, les diagonales paires modulées sont définies par :

$$\mathbf{d}_i = \mathbf{g}(\lceil 0.5m_g + i \rceil) \times [\mathbf{x}_s(1 + 0.5|i|) \quad \dots \quad \mathbf{x}_s(m_X - 0.5|i|)] \quad (4.15)$$

$$i = [-(m_X - 2) \quad -(m_X - 4) \quad \dots \quad 0 \quad \dots \quad m_X - 4 \quad m_X - 2] \quad (4.16)$$

et les diagonales impaires modulées par :

$$\mathbf{d}_i = \mathbf{g}(\lceil 0.5m_g + i \rceil) \times [\mathbf{x}_{s,odd}(0.5 + 0.5|i|) \quad \dots \quad \mathbf{x}_{s,odd}(m_X - 0.5 - 0.5|i|)] \quad (4.17)$$

$$i = [-(m_X - 1) \quad \dots \quad -1 \quad 1 \quad \dots \quad m_X - 1] \quad (4.18)$$

Lorsque la forme modulante est celle du pulse, le résultat en deux dimensions dans la matrice de covariance est montré à la figure 4.14.

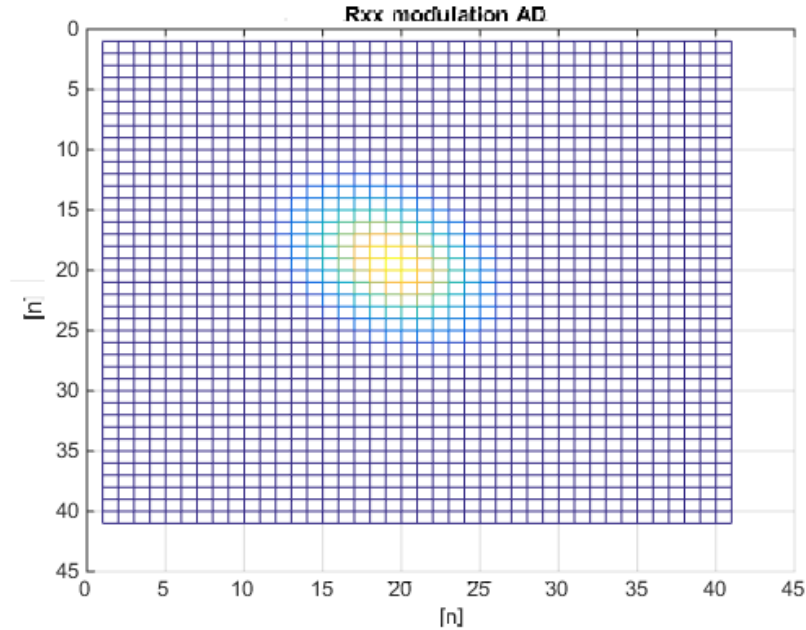


FIGURE 4.14 – Vue du dessus d’un écho dans la matrice de covariance où la forme modulante est un pulse \mathbf{g}

Le pulse ainsi obtenu est plus mince dans la direction anti-diagonale que sur celle diagonale. La modulation par la forme de la racine carrée d’un pulse ($\sqrt{\mathbf{g}}$) produit un écho symétrique selon ces 2 directions, tel que montré à la figure 4.15. La forme de la racine carré du modèle du pulse est équivalente à la forme d’un pulse dont le paramètre de la variance de la gaussienne σ^2 égalerait la variance du pulse original multiplié par la racine carré de 2.

L’exemple de la figure 4.16, utilisant les mêmes réponses pour les 2 cas, démontre que le pic de la position de l’écho sur le pseudospectre est passablement plus élevé lorsque la forme modulante utilise la racine carré d’un pulse ($\sqrt{\mathbf{g}}$), comme à la figure 4.15 que la forme du pulse

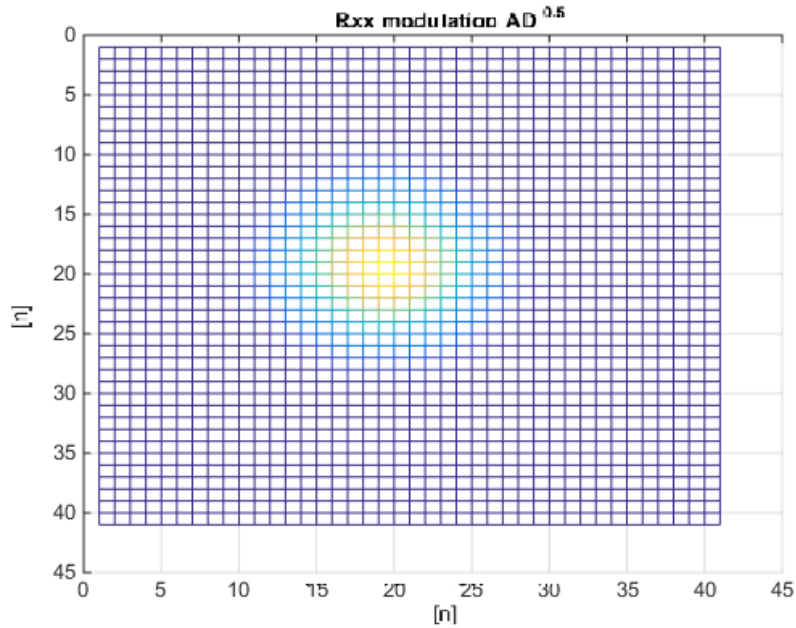


FIGURE 4.15 – Vue du dessus d'un écho dans la matrice de covariance où la variance du pulse modulant est multiplié par $\sqrt{2}$

directement (\mathbf{g}), comme à la figure 4.14. Cette différence dans les amplitudes est un indicateur que le vecteur propre obtenu est beaucoup plus près du modèle car la matrice de covariance a des éléments proportionnels au produit des signaux.

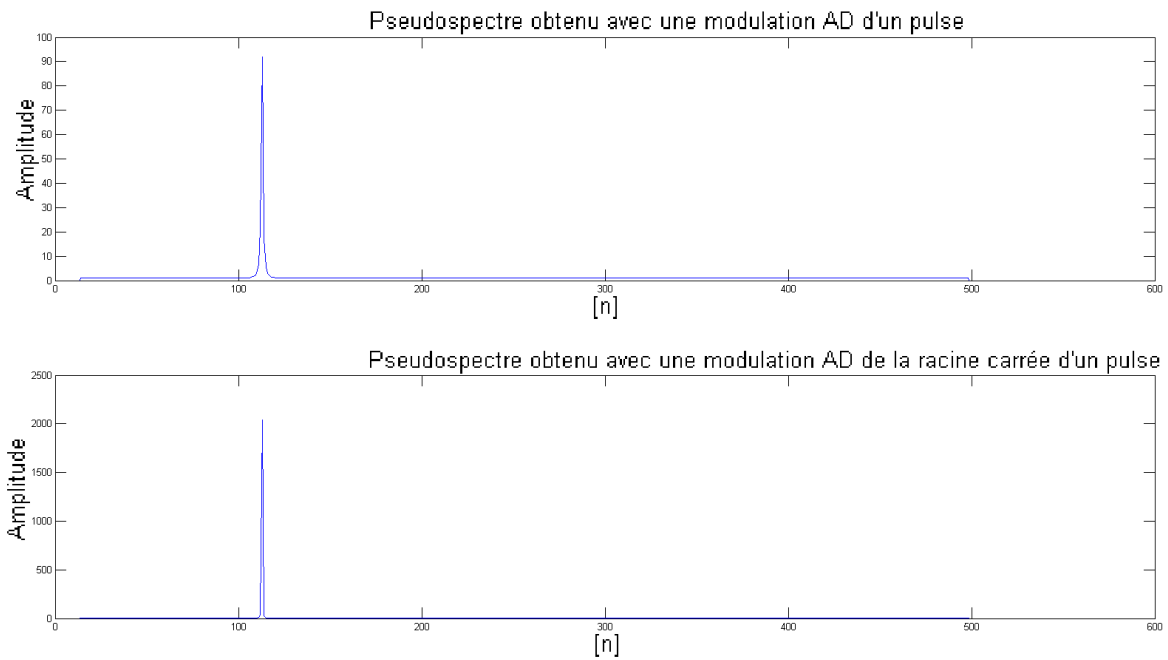


FIGURE 4.16 – Comparaison des pseudospectres pour 2 formes modulantes, 8 signaux à $SNR = 14.8 \text{ dB}$, 1 écho

Modulation par matrice modèle

Cette modulation de la matrice de covariance a également pour principal objectif d'éliminer les doublons, causant une baisse des performances de l'algorithme. Cette méthode se déroule en trois étapes : l'obtention de la matrice de covariance classique, l'élaboration de la matrice des modèles puis finalement l'obtention de la matrice finale modulée.

L'obtention de la matrice de covariance classique résulte de :

$$\hat{\mathbf{R}}_{XX,0} = \frac{1}{n_X} \mathbf{X} \mathbf{X}^T \quad (4.19)$$

Si plus d'un écho est présent dans les réponses de l'historique \mathbf{X} , l'apparition d'échos doublons est inévitable. Ces doublons proviennent du fait que les échos sont corrélés. Les détails à propos des doublons sont mentionnés précédemment à la section 4.1.1.

La matrice des modèles est produite par :

$$\mathbf{R}_{XX,model} = \sum_{k=1}^{m_X} \mathbf{a}(\tau_k) \mathbf{a}(\tau_k)^T \quad (4.20)$$

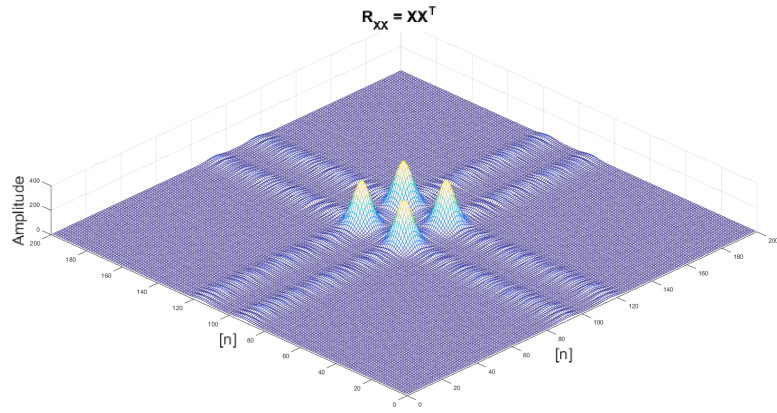
où $\mathbf{a}(\tau_k)$ est un vecteur modèle d'écho décalé de τ_k échantillons. Cette matrice peut être vue comme une succession de pulses, dont les sommets sont séparés d'un seul échantillon, centrés sur la diagonale principale. En fait, c'est exactement de cette manière que la matrice de modulation est créée : par le biais d'une somme du produit vectoriel d'une réponse dont l'écho est déplacé d'un échantillon selon τ_k à chaque itération.

La matrice finale modulée est le produit matriciel d'Hadamard de la matrice de covariance classique et de la matrice de modèles :

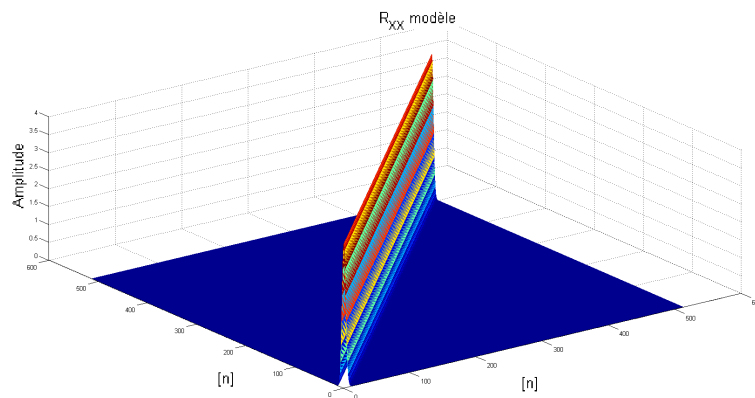
$$\hat{\mathbf{R}}_{XX,modul} = \hat{\mathbf{R}}_{XX,0} \circ \mathbf{R}_{XX,model} \quad (4.21)$$

Cette méthode ressemble en partie à la modulation anti-diagonale présentée précédemment. En effet, les deux méthodes modulent la matrice de covariance par une forme gaussienne avec un certain exposant sur le sens anti-diagonale de la matrice. Cependant, la modulation par la matrice des modèles ne comporte pas d'interpolation, comme la modulation anti-diagonale le requiert pour les diagonales impaires. Ainsi, la matrice finale obtenue via la modulation par la matrice des modèles subit moins d'approximations, et est par conséquent de meilleure qualité.

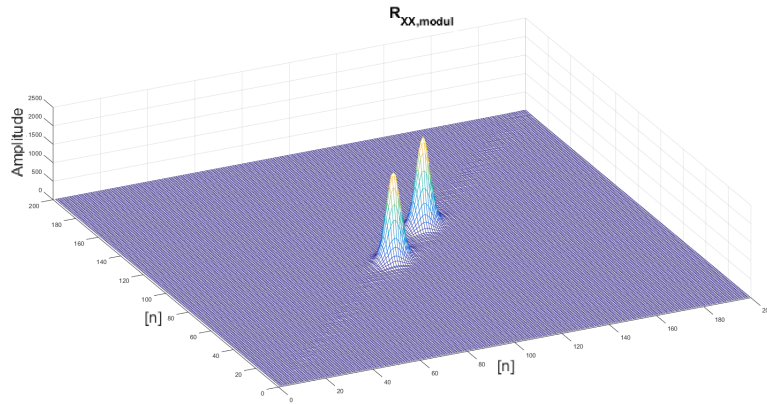
La figure 4.17 est un exemple des matrices obtenues aux 3 étapes de la création de la matrice de covariance finale.



(a) $\hat{R}_{XX,0}$



(b) $R_{XX, \text{modèle}}$



(c) $\hat{R}_{XX, \text{modul}}$

FIGURE 4.17 – Matrices aux différentes étapes de la modulation par la matrice des modèles

Comme les figures de $\hat{R}_{XX,0}$ et $\hat{R}_{XX, \text{modul}}$ le démontrent, la modulation par la matrice de modèles réduit fortement l'importance des doublons dans la matrice finale. Cependant, cette modulation modifie la forme d'un écho dans la matrice de covariance. En effet, l'écho de départ de la matrice $\hat{R}_{XX,0} = \mathbf{X}\mathbf{X}^T$ a une forme ronde. La modulation par la matrice modèle produit

un amincissement sur le sens anti-diagonale mais pas sur le sens diagonale. L'écho modulé a donc une forme ovale. En observant soit une ligne ou une colonne de ce dernier dans la matrice modulée, la forme est celle d'un pulse exposant $\sqrt{2}$. Autrement dit, cette forme est équivalente à un pulse dont la variance σ^2 serait égale à la variance du pulse originale divisée par la racine carrée de la racine carrée de 2. Quelques formes extraites d'une matrice formée à partir de signaux à un rapport signal bruit très bon sont montrées à la figure 4.18 :

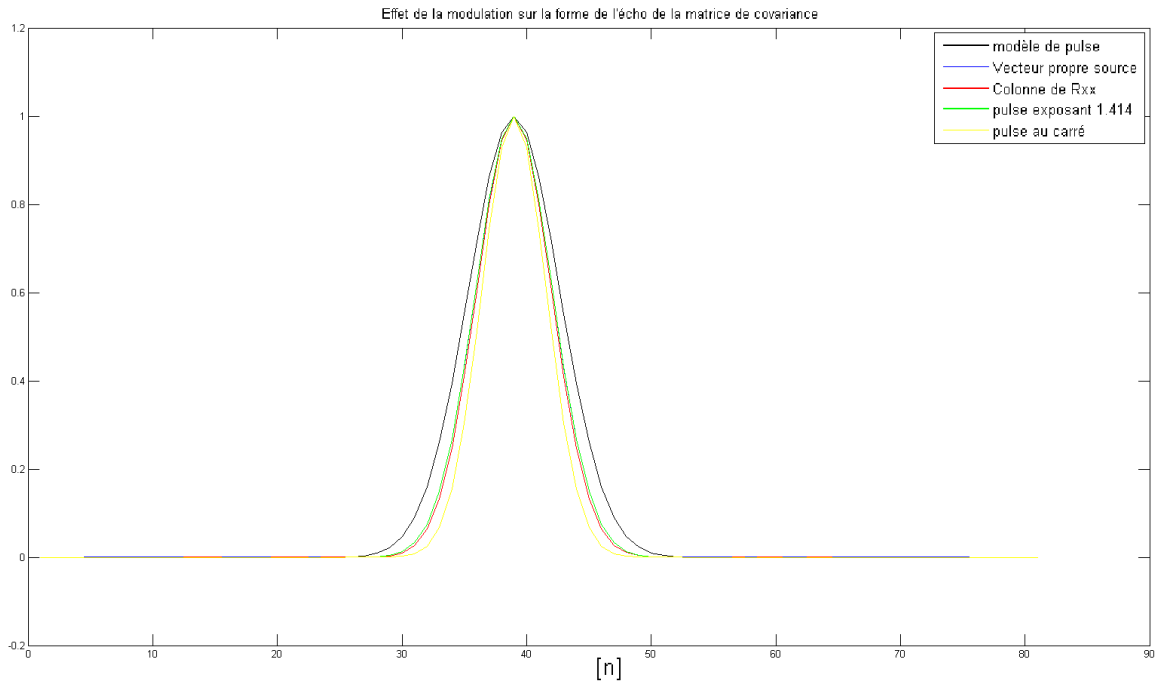


FIGURE 4.18 – Effet de la modulation sur la forme des échos de $\hat{\mathbf{R}}_{XX,modul}$

Sur la figure 4.18, on voit que la forme la plus large est celle du pulse modèle à la puissance 1, en noir. Les courbes bleue et verte sont parfaitement superposées, et correspondent respectivement au vecteur propre de la source et à un modèle de pulse dont l'exposant vaut $\sqrt{2}$. Finalement, la forme la plus étroite est celle en jaune, correspondant à un modèle de pulse au carré. Ainsi, la forme à rechercher lors du calcul du pseudospectre n'est plus un modèle de pulse, mais bien un modèle de pulse exposant 1.42. Les détails de cette étape sont expliqués à la section suivante 4.1.2 sur le pseudospectre.

4.1.2 Comparaison des lissages

Comme les méthodes de création de la matrice de covariance $\hat{\mathbf{R}}_{XX}$ avec pré-traitement de décorrélation ne procurent pas les mêmes performances, une comparaison de celles-ci est nécessaire pour déterminer laquelle sera choisie pour faire partie de l'algorithme. Deux aspects seront évalués : soit le SNR minimum des réponses en entrée pour que l'algorithme effectue la

détection de l'écho, et la distance à laquelle deux cibles rapprochées sont distinguées séparément.

SNR de détection minimal

Le but de ce test est de vérifier quelle méthode de création de $\hat{\mathbf{R}}_{XX}$ lissée permet à l'algorithme de détecter un écho dont le SNR est le plus faible possible. Cet aspect de la détection est certainement un des plus important car, en général, plus la cible est loin et plus le SNR de l'écho retourné est faible. Ainsi, plus l'algorithme de détection supporte de faibles SNR, plus la portée du système est grande. Les détails du calcul du SNR d'un écho sont disponibles à l'annexe A.1.

Avant de détailler la procédure de test, une version alternative à la méthode classique $\hat{\mathbf{R}}_{XX} = \mathbf{X}\mathbf{X}^T$ doit être ajoutée à la liste des méthodes testées. Comme les échos ont toujours leurs sommets centrés sur un point de la diagonale principale de la matrice, il en découle que leurs empreintes en deux dimensions ne dépasseront pas un nombre de diagonales, de part et d'autres de la diagonale principale. Ce nombre correspond à la moitié du nombre d'échantillons composant les échos. Ainsi, il est donc possible de conserver uniquement les éléments de ces diagonales et forcer tous les autres éléments de la matrice à zéro. Selon la définition des diagonales de la matrice faite à la section précédente 4.1.1, la matrice est créée via $\hat{\mathbf{R}}_{XX} = \mathbf{X}\mathbf{X}^T$, puis les diagonales superflues sont mises à zéros, comme sur la figure 4.19 :

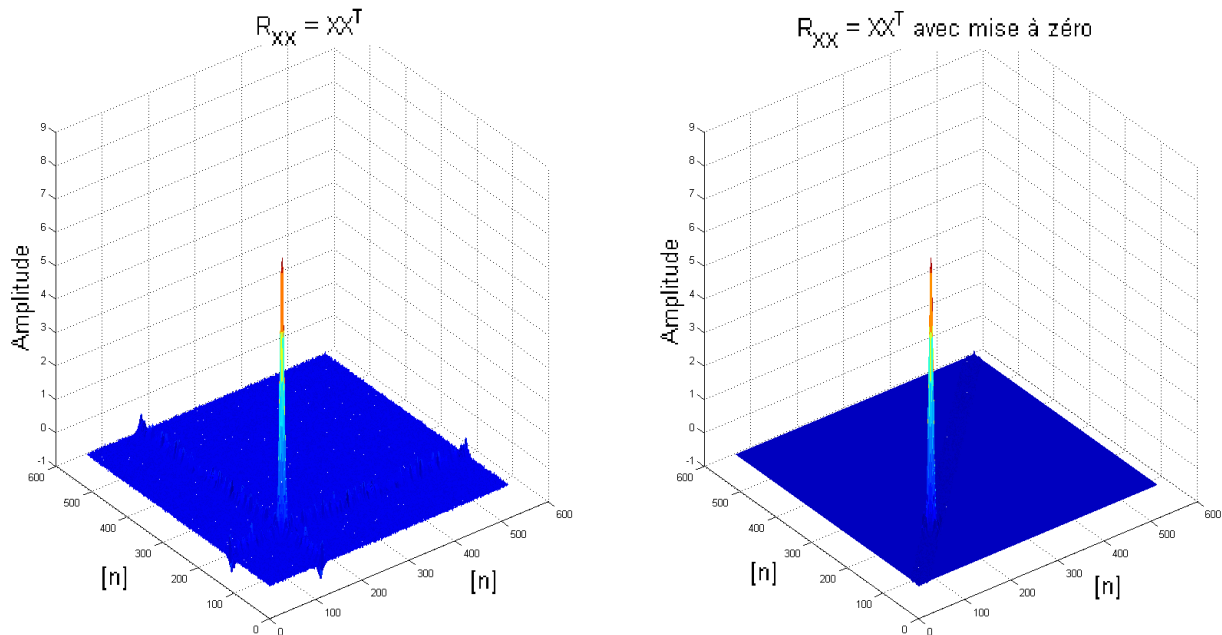


FIGURE 4.19 – Comparaison des matrices $\hat{\mathbf{R}}_{XX}$, 8 signaux, $SNR = 10dB$, 1 écho

Ainsi :

$$\mathbf{d}_k = 0 \quad \text{pour} \quad |k| = \lfloor 0.5m_G \rfloor, \dots, m_X - 1 \quad (4.22)$$

où m_G représente le nombre d'échantillons de l'écho modèle du pulse G .

La méthodologie du test est la suivante. Tout d'abord, 1000 essais contenant chacun 8 réponses sont créés de façon synthétique. Ces réponses contiennent un écho, à un SNR indiqué dans la première colonne du tableau ci-dessous, et dont la position est connue. Ensuite, pour chacun des essais, des matrices de covariance sont créées par les différentes méthodes et traitées par le reste de l'algorithme de manière semblable. Ainsi, toutes les versions doivent traiter les mêmes réponses, pour que la comparaison soit équitable entre les méthodes. Le pic obtenu sur le pseudospectre est analysé. Si ce dernier est détecté à la bonne position ou à ± 1 échantillon, la détection est considérée comme valide. Elle est invalide si elle est à l'extérieur de cette plage. Le tableau ci-dessous présente le pourcentage de détections valides pour chacune des méthodes d'élaboration de la matrice $\hat{\mathbf{R}}_{XX}$ en fonction du SNR des réponses des historiques.

Taux de détection des méthodes de création de $\hat{\mathbf{R}}_{XX}$ à différents SNR						
Méthode	$\mathbf{X}\mathbf{X}^T$	$\mathbf{X}\mathbf{X}^T$ avec mise à zéro	Super- position pas d'accu- mulation	Super- position avec accu- mulation	Modulation anti- diagonale	Modulation par matrice des modèles
SNR						
-2.7 dB	94.4 %	99.7 %	99.6 %	99.8 %	98.9 %	99 %
-4.2 dB	78.7 %	98.0 %	98.1 %	98.1 %	94.2 %	97.7 %
-5.7 dB	63.0 %	93.4 %	92.1 %	94.3 %	83.9 %	92.4 %
-7.2 dB	45.3 %	77.7 %	73.1 %	82.6 %	67.7 %	82.3 %
-8.7 dB	28.5 %	54.0 %	49.1 %	61.3 %	49.4 %	62.1 %
-10.2 dB	17.8 %	32.9 %	30.3 %	38.4 %	30.0 %	39 %
-11.7 dB	10.9 %	18.3 %	15.7 %	20.4 %	19.1 %	24.8 %

TABLE 4.1 – Taux de détection des différents pré-traitements de décorrélation selon différents SNRs, $K = 8$

Pour ce qui est de détecter les échos par faible SNR, ce sont la superposition décalée avec chevauchement et la modulation par la matrice des modèles qui procurent les meilleurs résultats. Si le but du système est uniquement d'obtenir la plus grande portée possible, alors ce sont les 2 méthodes à privilégier. Toutefois, d'autres critères servent à juger de la qualité des détections et sont comparés plus bas.

La figure 4.22 est un exemple des matrices $\hat{\mathbf{R}}_{XX}$ obtenues à -2.7 dB :

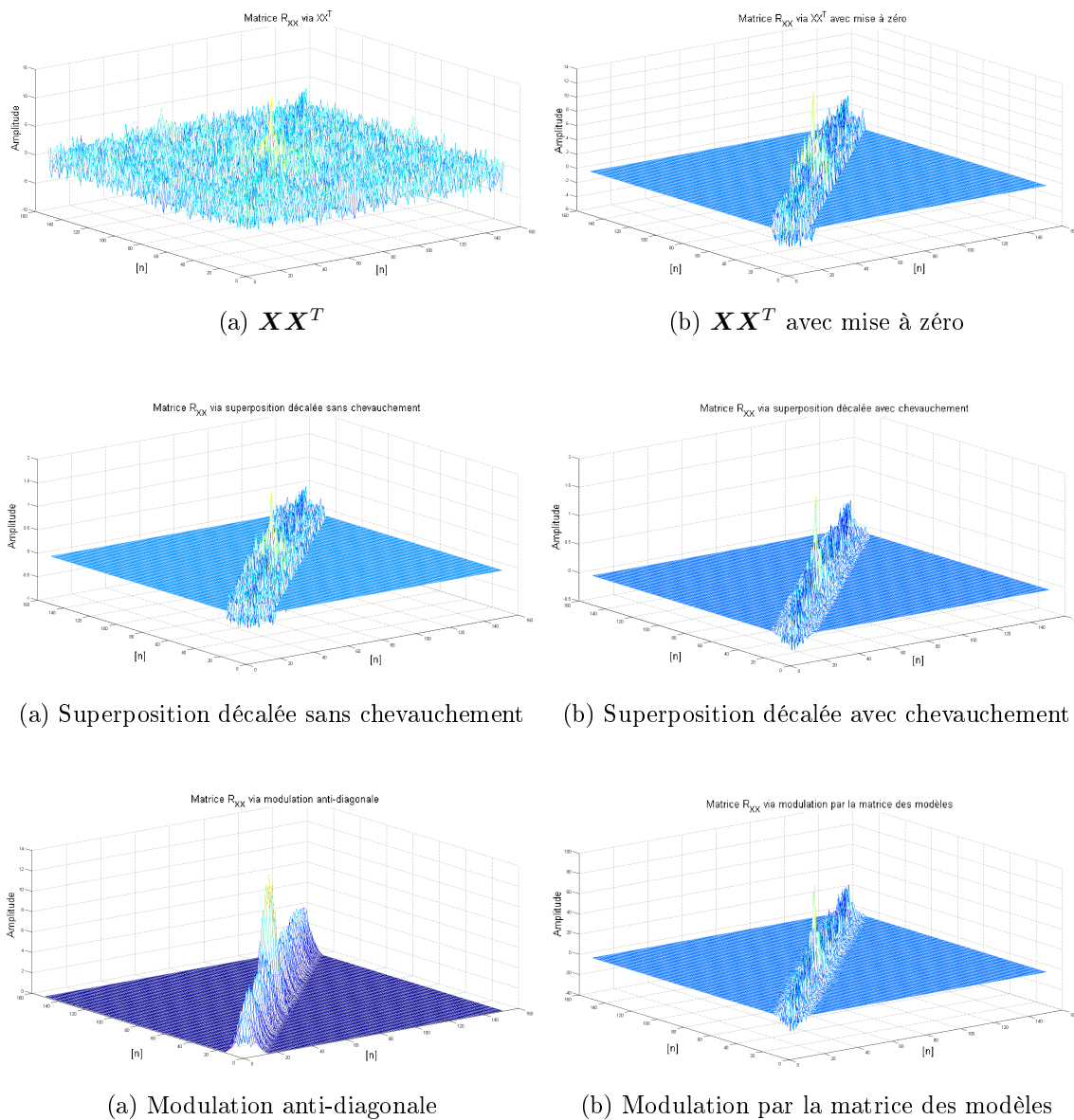


FIGURE 4.22 – Matrices $\hat{\mathbf{R}}_{XX}$ agrandies autour de l'écho pour différentes méthodes de création, réponses de l'historique à -2.7 dB

Séparation d'échos partiellement superposés

Un des buts à atteindre de l'algorithme est d'être en mesure de distinguer le plus possible deux cibles dont les échos se superposent partiellement. Les différentes méthodes d'élaboration de la matrice de covariance $\hat{\mathbf{R}}_{XX}$ ont des effets sur la décomposition propre de l'algorithme, menant à une variation des performances. Pour le processus de sélection de la meilleure méthode, il est important de comparer la limite de chacune.

La méthodologie de test est la suivante. Des essais d'une situation réelle ont tout d'abord été

enregistrés. La scène consistait donc en 2 cibles posées à une distance aisément atteinte par le système, et séparées d'une certaine distance. Pour une même distance de séparation, le système enregistrait entre 250 et 350 réponses. L'algorithme utilise un historique de 4 réponses pour produire un pseudospectre. L'algorithme réutilise les 3 dernières réponses lors de l'essai suivant. Ainsi, pour 100 réponses enregistrées, 97 pseudospectres en résultent. Par la suite, une analyse du pseudospectre vérifie que les endroits où les cibles sont présentes affichent bien 2 pics, avec une légère tolérance sur la distance séparant les 2 pics. Si un pic est manquant, l'essai est considéré comme un échec. Finalement, il suffit de calculer le pourcentage d'essais valides pour obtenir le taux de succès de détection des 2 cibles avec échos partiellement superposés pour chacune des méthodes de création de la matrice $\hat{\mathbf{R}}_{XX}$. Un exemple des réponses enregistrées lorsque les cibles étaient distancées de 5 mètres apparaît sur la figure 4.23. Le tableau 4.2 indique les taux de succès obtenus selon la méthode de construction de $\hat{\mathbf{R}}_{XX}$.

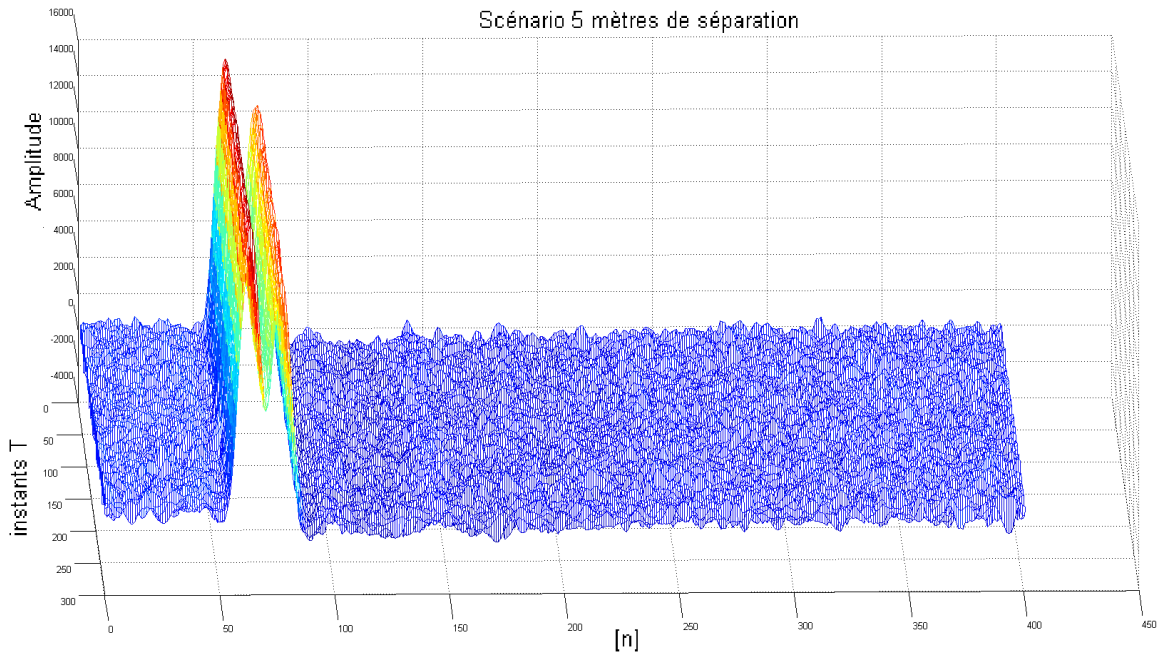


FIGURE 4.23 – Réponses enregistrées pour des cibles séparées de 5 mètres

Tout d'abord, pour la méthode classique $\mathbf{X}\mathbf{X}^T$, le taux de détection ne se détériore pas régulièrement. Les signaux réels peuvent avoir un SNR variable et certains aléas que les signaux synthétiques n'ont pas, expliquent le phénomène.

Ce sont finalement les méthodes de création de la matrice de covariance par modulation anti-diagonale et par la matrice des modèles qui séparent le mieux des échos partiellement superposés. Étant très sensibles, la sélection des valeurs propres, expliquée à la section suivante 3.5, devient plus importante. En effet, dans le cas où un seul écho est présent, que celui-ci a une largeur plus grande que prévue et que 2 plutôt qu'une seule valeur propre sont choisies, il est

Taux de séparation d'échos partiellement chevauchés						
Méthode	$\mathbf{X}\mathbf{X}^T$	$\mathbf{X}\mathbf{X}^T$ avec mise à zéro	Super- position pas d'accu- mulation	Super- position avec accu- mulation	Modulation anti- diagonale	Modulation par matrice des modèles
Écart						
5 m	37.23 %	100 %	100 %	100 %	100 %	100 %
4.75 m	41.79 %	100 %	100 %	100 %	100 %	100 %
4.5 m	39.04 %	100 %	100 %	100 %	100 %	100 %
4.25 m	21.65 %	100 %	100 %	100 %	100 %	100 %
4 m	0 %	100 %	100 %	100 %	100 %	100 %
3.75 m	0 %	100 %	100 %	100 %	100 %	100 %
3.5 m	0.71 %	100 %	4.59 %	100 %	100 %	100 %
3.25 m	1.08 %	42.65 %	0 %	100 %	100 %	100 %
3 m	0.72 %	0 %	0 %	100 %	100 %	100 %
2.75 m	0 %	0 %	0 %	100 %	100 %	100 %
2.5 m	0 %	0 %	0.36 %	100 %	100 %	100 %
2.25 m	0 %	0 %	0 %	100 %	100 %	100 %
2 m	0 %	0 %	0 %	92.79 %	100 %	100 %
1.75	0 %	0 %	0 %	19.68 %	100 %	100 %
1.5 m	0 %	0 %	0 %	0 %	100 %	100 %
1.25 m	0 %	0 %	0 %	0 %	100 %	99.37 %

TABLE 4.2 – Taux de succès obtenus selon la méthode de construction de $\hat{\mathbf{R}}_{\mathbf{X}\mathbf{X}}$

possible que le pseudospectre affiche 2 pointes rapprochées autour de la position du sommet de l'écho plutôt qu'une seule pointe, menant à une fausse détection.

Conclusion sur l'élaboration de la matrice de covariance

La méthode d'élaboration de la matrice de covariance a un impact important sur les performances globales de l'algorithme. Elle doit donc être choisie de façon à maximiser l'aspect de la détection souhaité par l'utilisateur. Ainsi, pour ce projet, la méthode de modulation par la matrice des modèles est celle choisie pour la suite. En effet, cette méthode permet à la fois de détecter les échos malgré de faibles SNR tout en discriminant des échos superposés dont les sommets sont très rapprochés.

Tout dépendant des contraintes d'implantation de l'algorithme dans un système donné, les possibilités d'optimisation sur la méthode d'élaboration de la matrice de covariance peuvent également devenir un critère de sélection important. En effet, les dimensions des matrices de covariance sont grandes et peuvent être problématiques. Les méthodes pouvant réduire la taille de la mémoire de stockage requises sont donc avantageuses. De plus, le nombre de calculs à l'élaboration diffère d'une méthode à l'autre, faisant ainsi un autre paramètre à tenir en

compte dans le choix d'une méthode de création de $\hat{\mathbf{R}}_{XX}$.

La méthode de prétraitement de décorrélation de modulation par la matrice modèle serait celle qui présente le moins de compromis sur les différents aspects de la qualité des détections. Avec la superposition décalée avec accumulation, ces deux prétraitements présentent les meilleurs SNRs de détection minimaux. Au niveau de la discrimination d'échos partiellement superposés, ce sont la modulation anti-diagonale et la modulation par la matrice modèle qui performant le mieux. Il est à noter que l'aspect de la détection qui bénéficie le plus du prétraitement de décorrélation est certainement celui-ci. En effet, les performances de séparation d'échos pour une élaboration classique de la matrice de covariance via $\mathbf{X}\mathbf{X}^T$ sont fortement inférieures. Ainsi, puisque la modulation par la matrice modèle est la méthode qui performe le plus sur le plus grand nombre d'aspects, c'est le prétraitement à privilégier pour la plupart des cas. Toutefois, ce n'est pas la méthode qui procure les valeurs propres sources les plus distinctives de celles du bruit. Un utilisateur ayant pour principal objectif de déterminer le nombre de cibles devrait idéalement se tourner vers un autre pré-traitement de décorrélation.

Choix du modèle pour l'obtention du pseudospectre

Lors de l'obtention du pseudospectre, selon :

$$P_{MUSIC}(\tau) = \frac{1}{\mathbf{a}(\tau)^T \mathbf{P}_n \mathbf{a}(\tau)} \quad (4.23)$$

les modèles de vecteurs de délais $\mathbf{a}(\tau)$ doivent être judicieusement choisis car ceux-ci ont un effet sur les performances de l'algorithme. Dans le cas présent, la méthode d'élaboration de la matrice de covariance choisie est la modulation par la matrice des modèles, décrite à la section précédente 4.1.1. Une des particularités de cette méthode est de produire une déformation de l'écho, menant à une forme du pulse exposant $\sqrt{2}$. Cette forme se retrouve sur la ligne et la colonne de l'écho dans la matrice de covariance, mais également au niveau du vecteur propre source. Ainsi, la forme à rechercher est idéalement la même que cette dernière. Donc les vecteurs délais $\mathbf{a}(\tau)$ doit utiliser un modèle d'écho décalé, dont la forme correspond au pulse exposant $\sqrt{2}$.

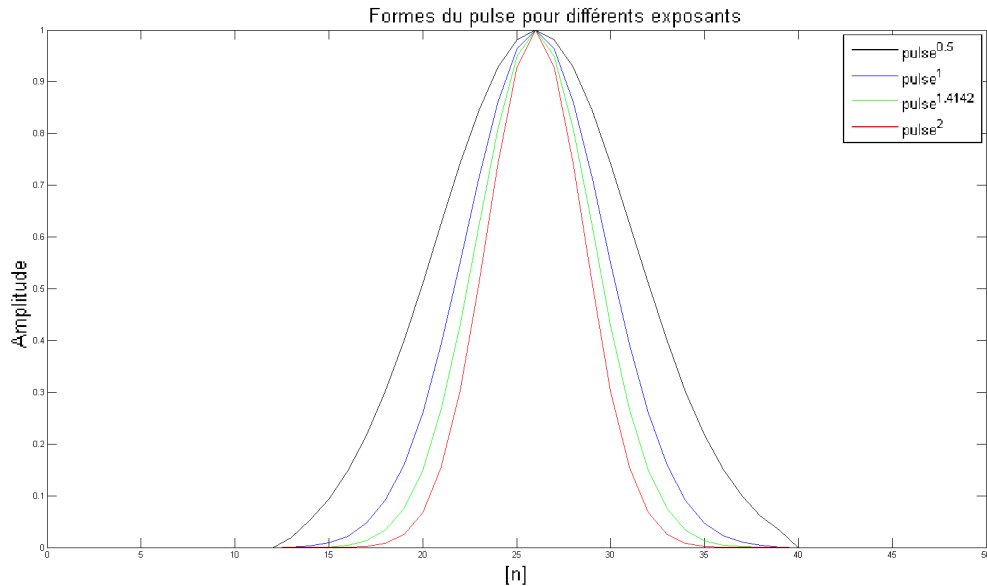


FIGURE 4.24 – Forme du pulse selon différents exposants

On comprend que la forme de pulse retrouvée dans le vecteur propre source est dépendante de la forme de l'écho dans la matrice de covariance, et par conséquent de la méthode d'élaboration de $\hat{\mathbf{R}}_{\mathbf{X}\mathbf{X}}$ utilisée. Il est donc important d'ajuster la forme des pulses de vecteurs de délais à la forme des échos de la matrice de covariance. Plus l'exposant ou la variance d'un pulse augmente, plus sa forme est amincie ; en voici un exemple sur la figure 4.24.

4.1.3 Critère conditionnel double de sélection des valeurs propres sources

Suite à la décomposition propre de la matrice de covariance approximée $\hat{\mathbf{R}}_{\mathbf{X}\mathbf{X}}$, l'étape suivante est de trier les valeurs propres selon la provenance de leur contribution, soit une source ou soit du bruit. Le critère de sélection des valeurs propres a comme fonction de distinguer quelles valeurs propres sont des contributions des sources, et par conséquent d'en déterminer leur nombre. Les vecteurs propres associés à ces valeurs propres sources formeront le sous-espace propre source, essentiel à la création du projecteur sur l'espace bruit \mathbf{P}_n , tel que discuté à la section 3.6.

À la section 3.5.1, deux critères pour déterminer le nombre de sources M sont présentés. Bien que fonctionnels, ceux-ci requéraient d'importants compromis sur certains aspects des performances de l'algorithme, comme le nombre de fausses détections ou encore la discrimination d'échos partiellement superposés. On a donc développé un nouveau critère dit conditionnel double.

Ce critère utilise à son avantage certaines propriétés des signaux temporels réels et de l'utilisation du système de notre partenaire industriel. Il devrait donc être adapté pour quiconque voudrait l'utiliser dans un contexte différent, mais il reste tout de même un bon exemple de

ce qui peut être fait de façon simple. Ce critère est dit double car il s'effectue en 2 temps : suite à la décomposition propre de la matrice de covariance approximée et suite à l'obtention du pseudospectre.

Sélection des valeurs propres sources pré-pseudospectre

Ce critère, bien que simple, a pour but de déterminer le nombre de vecteurs propres à inclure dans le sous-espace propre source \mathbf{E}_s . Le critère agit immédiatement après la décomposition propre en jugeant chacune des valeurs propres de façon à déterminer si elle est la contribution d'une source ou du bruit. Comme ce critère n'est pas optimal dans toutes les situations pouvant survenir, on lui laisse une tendance permissive pour s'assurer que tous les vecteurs propres ayant à faire partie de \mathbf{E}_s ont bel et bien été sélectionnés. Cette façon de faire s'appuie sur deux raisons principales. Tout d'abord, l'effet engendré par un vecteur propre en trop dans \mathbf{E}_s est moindre et a des effets bien moins dramatiques sur les résultats des détections que l'inverse. Ensuite, le nombre de cibles exact sera obtenu à une étape subséquente, soit après le critère de détermination du nombre de cibles post-pseudospectre, décrit plus bas.

Le critère pré-pseudospectre fonctionne sur la base d'une condition ayant 3 sous-parties. En obtenant les valeurs propres λ_i en ordre décroissant, le pseudocode décrivant le critère serait le suivant :

- $M_1 = 0$
- Boucle $i = [1, \dots, m_X - 1]$
 - SI $\{(\lambda_i > 50 \text{ OU } \lambda_i > 1.2\lambda_{i+1}) \text{ ET } \lambda_i > 1\}$
 - $M_1 = i$
 - SINON
 - Fin Boucle
 - Fin SI
- Fin Boucle
- $M_1 = M_1 + 1$

où M_1 indique le nombre estimé de valeurs propres associées à des sources selon ce critère.

Chacune des 4 sous-parties du critère a un rôle pour différentes situations. Tout d'abord $\lambda_i > 50$, le seuil est fixé à 50 car généralement, avec un historique à 8 réponses, les valeurs propres sont au-delà de cette valeur, sauf lorsque les échos se superposent, que le SNR se dégrade passablement ou qu'un écho a une amplitude beaucoup inférieure à un autre écho présent dans la même réponse. Il est à noter que cette valeur de seuil doit être proportionnelle au nombre de signaux contenus dans l'historique. La sous-partie $|\lambda_i| > 1.2|\lambda_{i+1}|$ sert justement à détecter les échos dont l'amplitude est beaucoup plus faible que celle de leur voisin précédent.

Elle est également utile lors de la superposition partielle des échos. Cependant, comme cette sous-partie pourrait laisser une valeur propre très faible, mais supérieure à 1.2 fois sa suivante, le troisième sous-critère $\lambda_i > 1$ est ajouté pour empêcher ce type de cas.

La dernière étape présentée dans le pseudocode ci-dessus ($M = M + 1$) est simplement une marge de manoeuvre pour réduire les chances qu'une source soit ignorée. Non seulement la détermination finale du nombre de sources, expliquée au paragraphe suivant va recalibrer cet ajout, mais il est moins dommageable de surévaluer la taille de \mathbf{E}_s que l'inverse, tel qu'expliqué plus bas à la section portant sur les effets d'un mauvais fonctionnement du critère de sélection des valeurs propres sources. Le précurseur de MUSIC, un algorithme développé par Pisarenko[13], n'utilisait qu'un seul vecteur propre pour former le sous-espace propre bruit \mathbf{E}_n , soit celui associé à la plus petite valeur propre.

Ce critère est une approximation laxiste du nombre de sources ayant pour but d'inclure suffisamment de vecteurs propres dans \mathbf{E}_s . On s'assure ainsi d'inclure toutes les contributions sans toutefois permettre à un nombre déraisonnable de vecteurs propres associés au bruit s'intégrer à \mathbf{E}_s . Malgré tout, le critère conditionnel sur les valeurs propres réussit à dénombrer un nombre exact de sources sur une plage de circonstances non négligeable. En utilisant des historiques de 8 réponses, il est donc entièrement fonctionnel dans les situations suivantes :

1. Lorsque des cibles non superposées ont des amplitudes semblables, la plage de SNR couverte s'étend de l'infini à -4.2 dB.
2. Lorsque des échos ont des amplitudes très différentes, le plus petit peut produire une valeur propre se confondant avec le bruit, en faisant une situation complexe malgré un SNR élevé des réponses enregistrées. Avec un bon SNR (disons de 35 dB), il repère des échos ayant un rapport d'amplitude pouvant aller jusqu'à 1/0.14.
3. Pour des échos de bonnes amplitudes (SNR des signaux à 35 dB), il détecte deux cibles jusqu'à un écart inter-échos de 26 périodes d'échantillonnage. Cette valeur est certes élevée, mais l'erreur du critère est d'ajouter une valeur propre aux sources. Ainsi, les deux cibles apparaissent sur le pseudospectre. Pour un écart-inter-échos allant de 17 à 26 périodes d'échantillonnage, la taille de \mathbf{E}_s est surévaluée. Pour un écart-inter-échos allant de 11 à 16 périodes d'échantillonnage, l'évaluation du nombre de sources redevient exacte avant de devenir finalement non fonctionnelle.

Suite à l'obtention des valeurs propres admises comme étant une contribution des sources par le critère de sélection pré-pseudospectre, le sous-espace propre \mathbf{E}_s est formé à partir des M_1 vecteurs propres associés aux M_1 plus grandes valeurs propres. Le processus restant de l'algorithme s'effectue normalement jusqu'à l'obtention du pseudospectre. Ce dernier sera ensuite utilisé par le critère de détermination du nombre de sources post-pseudospectre, donnant un nombre de sources détectées final.

Détermination du nombre de sources post-pseudospectre

S'il est possible d'utiliser ce critère post-pseudospectre, c'est grâce aux particularités de fonctionnement du système de notre partenaire industriel. Il ne peut donc pas être toujours possible d'élaborer ce genre de critère. Ce dernier, comme son nom l'indique, agit suite à l'obtention du pseudospectre. En fait, il s'agit d'un simple seuil s'appuyant sur la propriété qu'un vecteur propre en trop inclus dans \mathbf{E}_s produit des pics de faibles amplitudes sur le pseudospectre comparativement à un vecteur propre étant réellement la contribution d'une source. Cette propriété est détaillée à la section ci-dessous.

Une détection des pics du pseudospectre doit être préalablement effectuée à l'utilisation du critère. On définit donc un vecteur P_{peaks} contenant les positions des pics sur le pseudospectre P_{MUSIC} . Celui-ci compare tout simplement la valeur atteinte par le sommet du pic à la valeur d'amplitude moyenne du pseudospectre. Son fonctionnement sous forme de pseudocode serait le suivant :

— $M_2 = 0$

— Boucle $p = [1, \dots, m_{P_k}]$

$$\text{SI } P_{MUSIC}(P_{peaks}(p)) > 1.75 \frac{1}{m_{P_{MUSIC}}} \sum_{i=1}^{m_{P_{MUSIC}}} P_{MUSIC}(i)$$

$$M_2 = M_2 + 1$$

Fin SI

— Fin Boucle

où M_2 est le nombre estimé de sources détectées final. Le nombre de pics sur le pseudospectre est m_{P_k} . Dans le cas présenté ici, le seuil du critère stipule que si l'amplitude du pic vaut plus de 1.75 fois la valeur moyenne de l'amplitude moyenne du pseudospectre, ce pic provient d'une source et non du bruit. Un utilisateur désirant diminuer le plus possible les fausses détections aurait tendance à augmenter ce seuil. À l'opposé, lorsque l'objectif est de détecter toutes les cibles à n'importe quel prix, alors le seuil pourrait être abaissé.

L'ajout de ce critère post-pseudospectre augmente les performances de la détermination du nombre de sources par rapport à l'utilisation du critère de sélection des valeurs propres sources pré-pseudospectre utilisé seul. Les résultats des deux critères combinés sont les suivants :

1. La plage de SNR couverte pour des échos ne se superposant pas et ayant des amplitudes semblables est maintenant de l'infini à -5.2 dB. Le gain est seulement de 1 dB par rapport au critère sur les valeurs propres uniquement. L'avantage ne se trouve donc pas ici.
2. Au niveau des échos ayant des amplitudes différentes, leur ratio peut maintenant atteindre 1/0.07, deux fois mieux que ce qui était possible pour le critère sur les valeurs propres seul. Le gain ici est intéressant.

3. S'ils ont une amplitude similaire, les échos peuvent maintenant être comptabilisés malgré un écart aussi mince que 10 périodes d'échantillonnage. Avec la version de prétraitement de décorrélation utilisée sur la matrice de covariance approximée, cette dernière valeur est la limite absolue des performances de l'algorithme, même en forçant manuellement les vecteurs propres de \mathbf{E}_s . Le critère fonctionne donc très bien dans cette situation de superposition partielle d'échos. À titre comparatif, le critère sur les valeurs propres seulement commençait à montrer des signes de défaillance à 26 échantillons d'écart.

Pour obtenir un pseudospectre parfait, c'est-à-dire affichant uniquement les pics correspondant aux sources, il suffirait de reformer le sous-espace propre \mathbf{E}_s en y incluant uniquement les M_2 vecteurs propres associés aux M_2 plus grandes valeurs propres et à retracer le pseudospectre à partir de cette version de \mathbf{E}_s mise à jour.

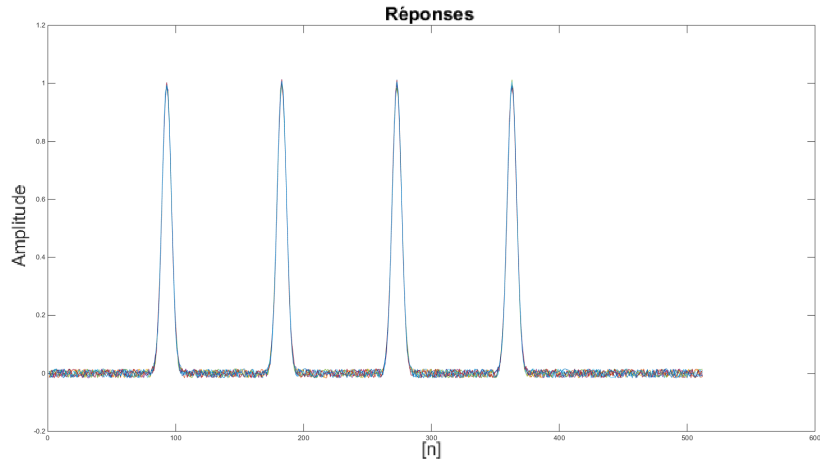
Effets d'un mauvais dimensionnement du sous-espace propre source

Le fonctionnement du critère conditionnel double pour la sélection des valeurs propres est basé sur le principe qu'un sous-espace propre source \mathbf{E}_s surdimensionné est préférable à un nombre de vecteurs sous-évalué.

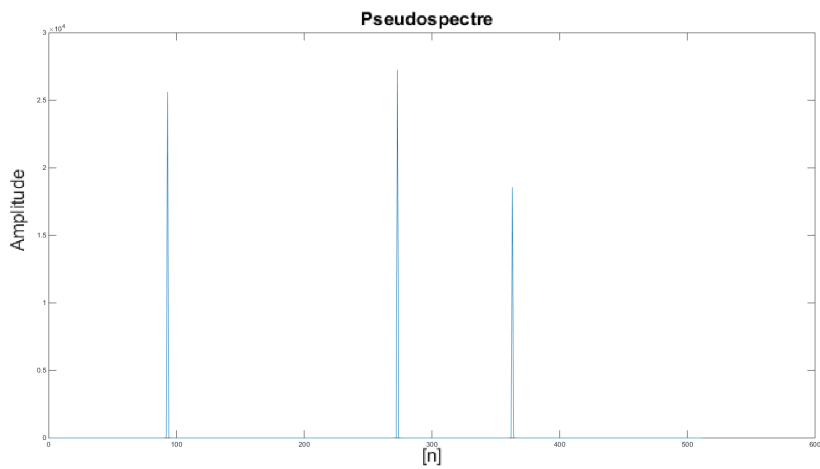
Il est impossible de récupérer une cible lorsque son vecteur propre est rejeté dans \mathbf{E}_n plutôt que dans \mathbf{E}_s . En voici un exemple patent à la figure 4.25 lorsqu'un historique contient des réponses affichant 4 échos et que \mathbf{E}_s se limite aux 3 plus importants vecteurs propres plutôt que 4.

Bien que 4 échos sont clairement présents dans les réponses, le pseudospectre n'affiche que 3 pics car \mathbf{E}_s est de dimensions $[m_X \times 3]$ et contient seulement les contributions de 3 sources. La 4^e est donc perdue. Il en résulte donc une détection manquante, ce qui peut être un manquement grave dans plusieurs situations.

À l'opposé, les effets d'un sous-espace propre source surdimensionné sont bien moindres et plus facilement gérables. La figure 4.26 montre un pseudospectre obtenu à partir des réponses montrées à la figure 4.25 avec un sous-espace propre \mathbf{E}_s dont la taille est doublée par rapport à ce qu'elle devrait être. En effet, 8 vecteurs propres y sont inclus, et par conséquent 4 sont des contributions du bruit.



(a) Réponses de l'historique



(b) Pseudospectre obtenu

FIGURE 4.25 – Effet d'un sous-espace propre \mathbf{E}_s sous-dimensionné

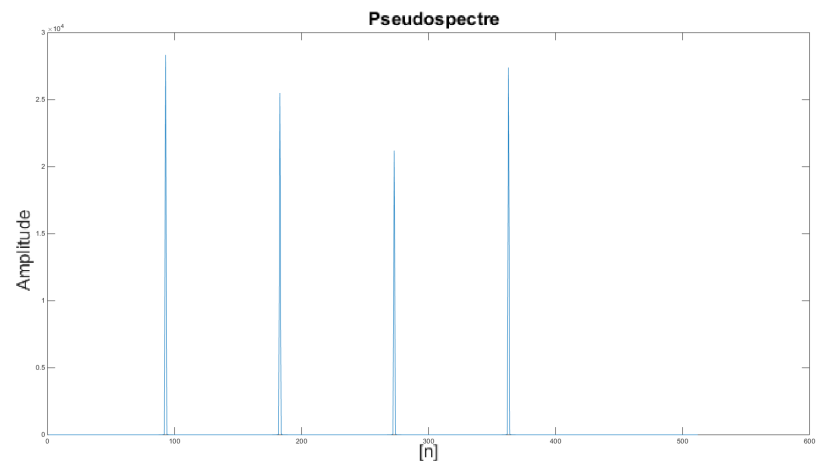


FIGURE 4.26 – Effet d'un sous-espace propre \mathbf{E}_s surdimensionné sur le pseudospectre obtenu

Le pseudospectre obtenu contient bel et bien uniquement 4 pics identifiant les 4 échos présents dans les réponses formant l'historique. L'ajout en trop des vecteurs propres bruit dans \mathbf{E}_s n'a aucune répercussion dans ce cas-ci. Cependant, lorsque le SNR des réponses se dégrade, d'autres pics peuvent apparaître, comme à la figure 4.27.

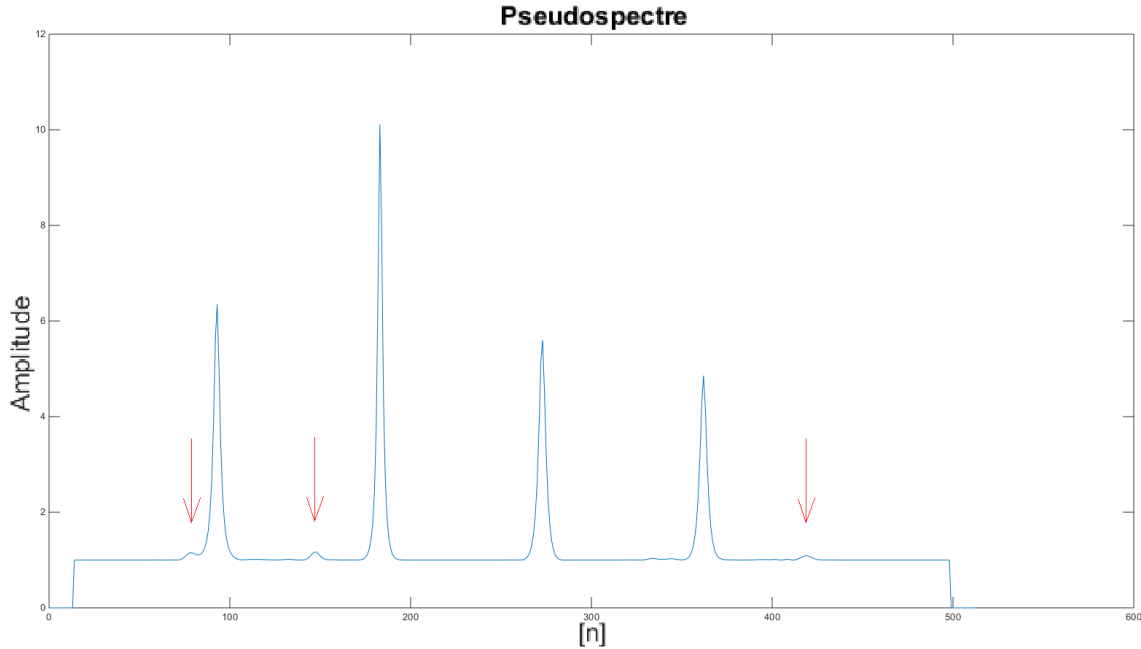


FIGURE 4.27 – Effet d'un sous-espace propre \mathbf{E}_s surdimensionné sur le pseudospectre obtenu par des SNRs faibles

Cette fois, avec des réponses très bruitées et un espace propre source de 8 vecteurs (4 sont en trop), des pics autres que ceux des échos sont apparus. Bien que l'algorithme soit à quelques décibels seulement de son SNR de détection minimal, les pics des fausses détections ont une amplitude beaucoup plus faibles que les pics des sources. Ainsi, par un seuillage simple, le rejet des fausses détections devient aisé.

Cas particulier : réponses sans aucun écho

Dans le cas où les réponses d'un historique ne contiendrait aucun écho, ce qui survient lorsqu'aucune cible n'est présente dans le champ de vue à une distance plus courte ou égale à la portée du système, la première section du critère donnera un nombre de sources M_1 bien supérieur à la réalité (aux alentours de 12 lors des tests). Toutefois, comme aucun des vecteurs inclus dans \mathbf{E}_s ne forme une base orthogonale avec le modèle $\mathbf{a}(\tau)$, les pics du pseudospectre affichera une multitude de pics, mais tous de faibles amplitudes. À ce moment, la deuxième portion du critère, celle qui agit sur le pseudospectre, détectera l'absence de sources et $M_2 = 0$.

4.1.4 Conclusion sur le critère de détermination du nombre de sources

Comme l'utilisation des critères de séparation des espaces propres rencontrés fréquemment en traitement d'antenne, tels que le critère d'information d'Akaike ou Bayésien, nécessiteraient des adaptations majeures, et qu'un seuillage simple offre des performances limitées (section 3.5.1), un critère conditionnel en deux étapes peut s'avérer être une solution simple et efficace. Pour un utilisateur dont le principal objectif de l'algorithme est de déterminer le nombre de sources, il est prêt à attendre après la formation du pseudospectre pour obtenir les meilleures performances.

Les principaux gains de ce critère conditionnel double par rapport au seuillage simple de la section 3.5.1 se font particulièrement sentir à de faibles SNRs et lorsque des échos d'une même réponse ont des amplitudes très différentes. Toutefois, il est possible d'ajuster les sous-parties du critère pour optimiser certaines situations, souvent au détriment de d'autres, il faut donc faire le meilleur compromis possible.

4.1.5 Précision au-delà de l'échantillon près

La résolution de la mesure en distance atteint l'échantillon près suite à l'obtention du pseudospectre, à la section 3.6 du chapitre 3. Cependant, ce n'est pas toujours le cas. De plus avec un bon SNR, il pourrait être possible d'améliorer la résolution pour qu'elle soit encore meilleure. Une solution serait d'améliorer les performances du système (échantillonnage plus rapide, réduction de la largeur des pulses émis), ce qui se traduit souvent par des changements majeurs. Une autre solution à ce problème est d'ajouter une étape à l'algorithme, qui permettrait de connaître plus précisément la position de l'écho, même entre 2 échantillons. Effectivement, bien que le sommet numérique du modèle de l'écho correspond à un certain échantillon de la réponse, il est fort probable que le sommet réel de l'écho se situe entre 2 échantillons numériques.

Une fois que les positions des cibles sont repérées sur le pseudospectre balayé par pas d'un seul échantillon, un autre pseudospectre plus précis est réalisé autour de la position entière de l'écho précédemment trouvé, soit sur la plage $\tau_k \pm 0.5T_e$. Celui-ci est tracé à l'aide d'un décalage sous-échantillon du modèle de pulse. Les sous-pseudospectres sont obtenus via :

$$\mathbf{P}_{MUSIC}(\tau_{i,k}) = \frac{1}{\mathbf{a}(\tau_k + i\delta)^T \mathbf{P}_n \mathbf{a}(\tau_k + i\delta)} \quad (4.24)$$

où k correspond à un sommet du pseudospectre et va de 1 à M , M étant le nombre de sources d'échos. Ainsi, τ_k correspond au délai du k^e écho, et vaut un nombre entier de périodes d'échantillonnage. Le pas de décalage est δ du modèle utilisé, et il correspond à une fraction

d'une période d'échantillonnage, où I est le nombre de pas de décalage. Ainsi, $I = 1 + \frac{1}{\delta}$. Un exemple de pulse décalé utilisé pour la création de $\mathbf{a}(\tau_i)$ est disponible à la figure 4.28. La légende indique le décalage appliqué à chacun des modèles de pulse.

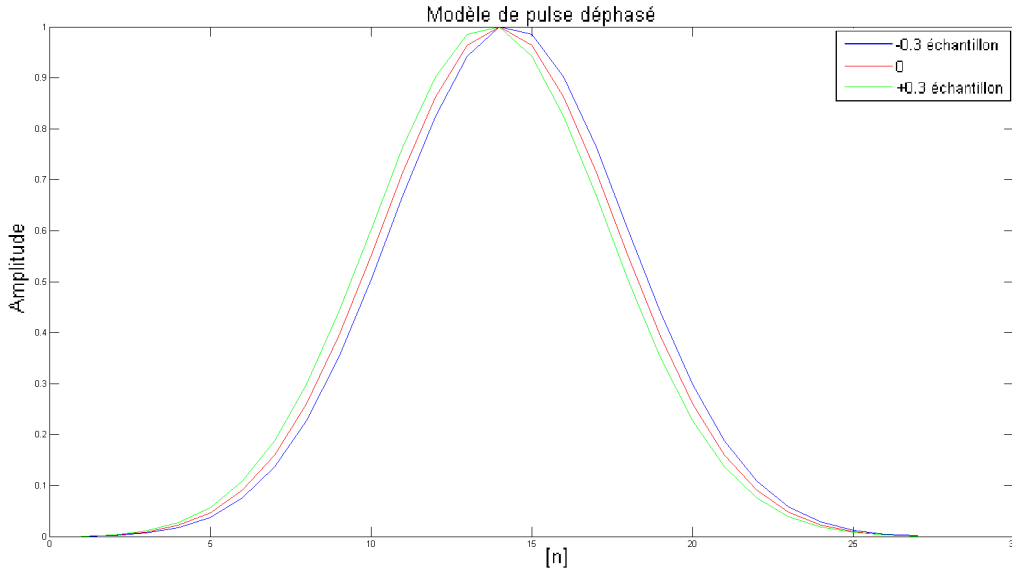


FIGURE 4.28 – Exemple de déphasage sous-échantillon du modèle de pulse, où $\delta = 0.3$ échantillon

Cette méthode exploite le fait que lorsque le décalage du modèle utilisé et celui de l'écho reçu sont identiques, à $\tau_a = \tau_x$, la correspondance entre les deux est à son maximum, ce qui engendre un pic plus prononcé sur le sous-pseudospectre. Ce principe est identique à celui produisant le pseudospectre, mais sur une échelle plus fine.

Le graphique de la figure 4.29 est un exemple de résultats obtenus pour l'obtention du sous-pseudospectre via décalage successif du modèle. Dans ce cas, le SNR des réponses de l'historique est de 30 dB. Sur le pseudospectre, le premier pic ($k = 1$) est situé à la position $\tau_1 = 112$, et le deuxième pic ($k = 2$) à la position $\tau_2 = 333$. Pour ce second pic, un sous-pseudospectre est tracé en utilisant un pas de déphasage $\delta = 0.01$ et $I = 101$. Ce sous-pseudospectre indique que la position exacte du second pic est 332.8.

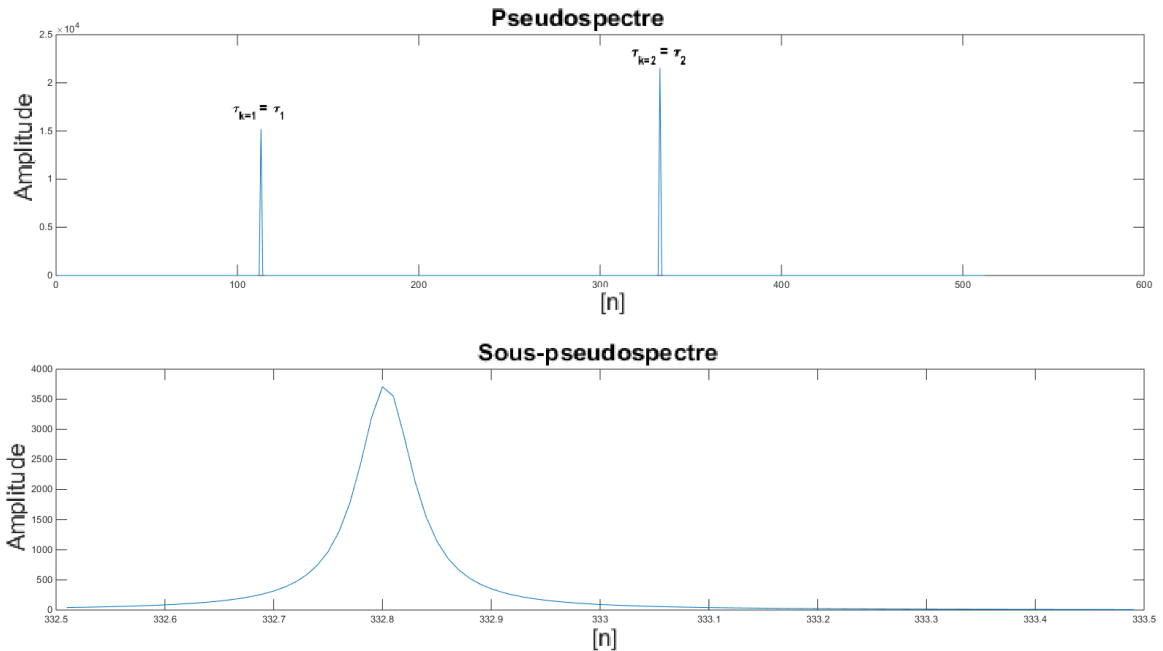


FIGURE 4.29 – Exemple de sous-pseudospectre obtenu pour un écho à la position $\tau_2 + i\delta = 332.8$

Évidemment, le pas de décalage sous-échantillon δ utilisé pour le balayage de $\mathbf{a}(\tau_k + i\delta)$ a un impact important sur la résolution de mesure obtenue. Plus le pas est fin, plus la mesure est précise. Si le pas δ vaut un demi échantillon, alors la position de l'écho ne peut pas être déterminée plus précisément qu'au demi échantillon près, tandis qu'avec $\delta = 0.01$, la position de l'écho peut être déterminée au centième d'échantillon près. Cependant, plus le pas est réduit, plus le nombre de décalages de $\mathbf{a}(\tau_i)$ est grand, augmentant le temps de réponse et le nombre de calculs nécessaires. De plus, lorsque le SNR des échos reçus diminue, l'incertitude sur la position de l'écho augmente à un point où une trop grande finesse du balayage sous-échantillon devient inutile.

4.2 Optimisation pour l'implantation

Certaines portions de l'algorithme peuvent présenter des complexités calculatoires élevées, alors que d'autres requièrent d'importantes quantités de ressources. Ce sont deux limitations à l'implantation de l'algorithme. L'utilisation de cet algorithme pour une application en temps réel oblige une efficacité et une diminution des calculs nécessaires pour que l'obtention d'un résultat soit rapide. Le système d'implantation vise un coût raisonnable, il est donc impossible d'utiliser une quantité importante de matériel. Les prochaines sections présentent des simplifications à certains calculs de l'algorithme et des rationalisations des ressources requises.

4.2.1 Stockage de la matrice de covariance

La matrice de covariance $\hat{\mathbf{R}}_{XX}$ créée au début de l'algorithme peut s'avérer très grande et présenter un problème de stockage, comme sur un système FPGA par exemple. Pour des signaux de 1024 échantillons, la matrice de covariance comporte déjà 1 048 576 éléments. Avec l'augmentation de la portée du système de détection, la taille des matrices peut augmenter dramatiquement et rendre l'algorithme non implantable.

Une caractéristique particulière des matrices de covariance produites par des signaux temporels entièrement réels permet toutefois de réduire grandement le nombre d'éléments de la matrice à stocker. Pour toutes les méthodes de création de $\hat{\mathbf{R}}_{XX}$ présentées à la section précédente 4.1.1, seulement quelques diagonales sont dignes d'intérêt. En effet, seuls les échos dont le sommet est centré sur un élément de la diagonale principale de la matrice de covariance sont valides. Les autres, si présents, sont des doublons nuisibles, comme il est expliqué à la section 4.1.1. Ainsi, le nombre de diagonales contenant l'information essentielle correspond à la largeur d'un écho, comme le montre la figure 4.30.

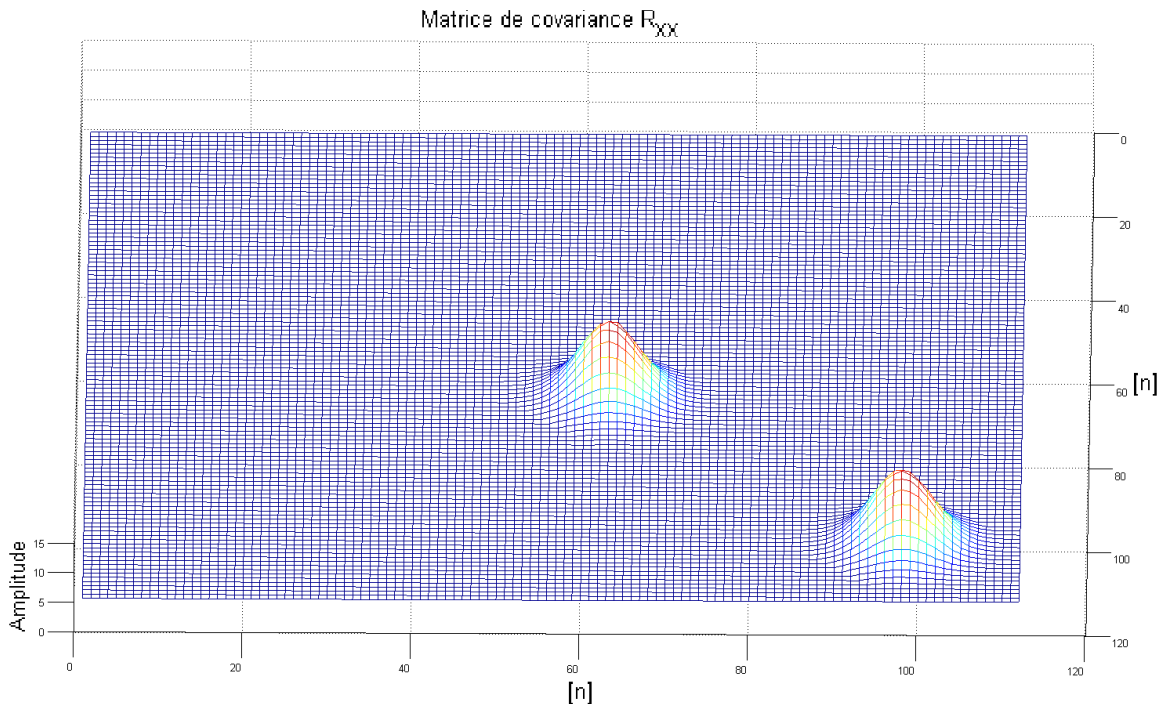


FIGURE 4.30 – Exemple de matrice de covariance démontrant les diagonales utiles

La figure 4.31 est un exemple de matrice de covariance agrandie autour d'un écho, dont le sommet est centré à la position (113, 113). Le pré-traitement de décorrélation utilisé dans ce cas est la superposition décalée avec accumulation. On peut ainsi constater que seules quelques

diagonales autour de la diagonale principale ont des valeurs non nulles. Le reste de la matrice contient uniquement des 0.

107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	0
2.2126e-04	2.6380e-04	2.2087e-04	1.6120e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-3.4529e-04	-4.3787e-04	-4.6389e-04	-3.9839e-04	-2.3620e-04	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-5.3205e-04	-6.8390e-04	-7.4656e-04	-7.4382e-04	-5.9202e-04	-3.3437e-04	0	0	0	0	0	0	0	0	0	0	0	0	0
-2.8686e-04	-3.4634e-04	-3.8979e-04	-4.2238e-04	-3.7001e-04	-2.7801e-04	-1.4517e-04	0	0	0	0	0	0	0	0	0	0	0	0
9.9758e-04	0.0013	0.0016	0.0017	0.0017	0.0014	9.7711e-04	4.6539e-04	0	0	0	0	0	0	0	0	0	0	0
0.0037	0.0050	0.0062	0.0069	0.0069	0.0062	0.0048	0.0031	0.0014	0	0	0	0	0	0	0	0	0	0
0.0099	0.0134	0.0167	0.0191	0.0197	0.0183	0.0153	0.0110	0.0066	0.0027	0	0	0	0	0	0	0	0	0
0.0205	0.0279	0.0354	0.0410	0.0433	0.0414	0.0359	0.0275	0.0185	0.0102	0.0040	0	0	0	0	0	0	0	0
0.0388	0.0531	0.0680	0.0799	0.0857	0.0836	0.0746	0.0596	0.0428	0.0266	0.0137	0.0050	0	0	0	0	0	0	0
0.0681	0.0937	0.1210	0.1435	0.1558	0.1543	0.1407	0.1157	0.0864	0.0573	0.0332	0.0162	0.0053	0	0	0	0	0	0
0.0937	0.1516	0.1972	0.2356	0.2583	0.2592	0.2400	0.2014	0.1548	0.1068	0.0661	0.0362	0.0158	0.0046	0	0	0	0	0
0.1210	0.1972	0.3011	0.3623	0.4006	0.4060	0.3806	0.3245	0.2546	0.1807	0.1165	0.0680	0.0334	0.0131	0.0038	0	0	0	0
0.1435	0.2356	0.3623	0.5118	0.5697	0.5823	0.5514	0.4759	0.3794	0.2748	0.1822	0.1108	0.0580	0.0256	0.0099	0.0027	0	0	0
0.1558	0.2583	0.4006	0.5697	0.7445	0.7661	0.7316	0.6379	0.5148	0.3789	0.2564	0.1603	0.0874	0.0411	0.0178	0.0064	0.0016	0	0
0.1543	0.2592	0.4060	0.5823	0.7661	0.9254	0.8898	0.7822	0.6376	0.4751	0.3267	0.2085	0.1169	0.0573	0.0265	0.0108	0.0035	0.0010	0
0.1407	0.2400	0.3806	0.5514	0.7316	0.8898	1.0041	0.8889	0.7306	0.5499	0.3828	0.2482	0.1421	0.0716	0.0345	0.0149	0.0055	0.0022	0
0.1157	0.2014	0.3245	0.4759	0.6379	0.7822	0.8889	0.9236	0.7645	0.5802	0.4079	0.2678	0.1557	0.0801	0.0397	0.0179	0.0070	0.0031	0
0.0864	0.1548	0.2546	0.3794	0.5148	0.6376	0.7306	0.7645	0.7426	0.5676	0.4024	0.2669	0.1571	0.0821	0.0416	0.0193	0.0078	0.0037	0
0.0573	0.1068	0.1807	0.2748	0.3789	0.4751	0.5499	0.5802	0.5676	0.5092	0.3634	0.2430	0.1445	0.0765	0.0393	0.0186	0.0078	0.0038	0
0.0332	0.0661	0.1165	0.1822	0.2564	0.3267	0.3828	0.4079	0.4024	0.3634	0.3046	0.2050	0.1230	0.0657	0.0343	0.0165	0.0070	0.0036	0
0.0162	0.0362	0.0680	0.1108	0.1603	0.2085	0.2482	0.2678	0.2669	0.2430	0.1621	0.0978	0.0527	0.0277	0.0135	0.0059	0.0030	0.0015	0
0.0053	0.0158	0.0334	0.0580	0.0874	0.1169	0.1421	0.1557	0.1571	0.1445	0.1230	0.0978	0.0694	0.0377	0.0199	0.0098	0.0043	0.0023	0
0	0.0046	0.0131	0.0256	0.0411	0.0573	0.0716	0.0801	0.0821	0.0765	0.0657	0.0527	0.0377	0.0240	0.0128	0.0063	0.0028	0.0015	0
0	0	0.0038	0.0099	0.0178	0.0265	0.0345	0.0397	0.0416	0.0393	0.0343	0.0277	0.0199	0.0128	0.0081	0.0040	0.0018	0.0011	9.8638e-04
0	0	0	0.0027	0.0064	0.0108	0.0149	0.0179	0.0193	0.0186	0.0165	0.0135	0.0098	0.0063	0.0040	0.0024	0.0011	6.0658e-04	3.6174e-04
0	0	0	0	0.0016	0.0035	0.0055	0.0070	0.0078	0.0078	0.0070	0.0059	0.0043	0.0028	0.0018	0.0011	6.6828e-04	3.6174e-04	2.5308e-04
0	0	0	0	0	0.0010	0.0022	0.0031	0.0037	0.0038	0.0036	0.0030	0.0023	0.0015	9.8638e-04	6.0658e-04	3.6174e-04	2.5308e-04	1.9164e-05
0	0	0	0	0	0	2.2205e-04	4.2884e-04	5.8048e-04	6.3723e-04	6.0627e-04	5.4451e-04	4.0798e-04	2.7903e-04	1.8223e-04	1.3487e-04	9.6366e-05	8.1094e-05	6.0558e-05
0	0	0	0	0	0	0	3.0341e-04	5.4007e-04	6.7603e-04	6.7940e-04	6.3471e-04	4.8510e-04	3.3058e-04	1.9160e-04	1.2910e-04	4.3134e-05	1.9164e-05	1.2910e-05
0	0	0	0	0	0	0	0	1.1294e-04	1.8016e-04	2.2565e-04	2.0683e-04	1.6809e-04	1.2929e-04	9.8471e-05	5.4292e-05	1.6851e-05	2.6342e-05	1.6851e-05
0	0	0	0	0	0	0	0	0	1.4350e-04	2.1003e-04	2.4507e-04	1.9807e-04	1.4677e-04	9.6870e-05	7.2741e-05	4.2361e-05	2.4696e-05	1.4350e-05
0	0	0	0	0	0	0	0	0	0	3.6301e-04	5.2763e-04	5.2501e-04	4.1008e-04	2.8611e-04	1.6980e-04	7.5738e-05	4.2189e-05	3.6301e-05
0	0	0	0	0	0	0	0	0	0	0	-8.4623e-06	-8.8790e-06	-3.0923e-06	4.9246e-06	5.3298e-06	-1.8414e-06	1.4363e-05	-8.4623e-06
0	0	0	0	0	0	0	0	0	0	0	0	5.9164e-05	7.1730e-05	6.0602e-05	4.5079e-05	2.2707e-05	2.0050e-05	5.9164e-05

FIGURE 4.31 – Matrice de covariance agrandie autour d’un écho, pré-traitement de décorrélation par superposition décalée avec chevauchement

La notation utilisée pour identifier les diagonales de la matrice de covariance est la même qu’à la section précédente 4.1.1, soit :

$$\hat{R}_{XX} = \begin{bmatrix} \mathbf{d}_0(1) & \mathbf{d}_1(1) & \dots & & \mathbf{d}_{m_X-1}(1) \\ \mathbf{d}_{-1}(1) & \dots & & & \\ \dots & & \dots & & \dots \\ & & & \dots & \mathbf{d}_1(m_X - 1) \\ \mathbf{d}_{-(m_X-1)}(1) & \dots & \mathbf{d}_{-1}(m_X - 1) & \mathbf{d}_0(m_X) & \end{bmatrix}$$

La figure 4.30 démontre bien que le nombre de diagonales occupées par tous les échos présents et utiles se limite au nombre d’échantillons nécessaires pour reproduire la forme d’un pulse. En définissant un vecteur pulse \mathbf{p} , contenant le modèle du pulse émis par le système, et comportant m_p échantillons, on peut considérer que seules la diagonale principale et les $\frac{m_p-1}{2}$ diagonales de part et d’autre de la diagonale principale contiennent les informations utiles et sont à conserver.

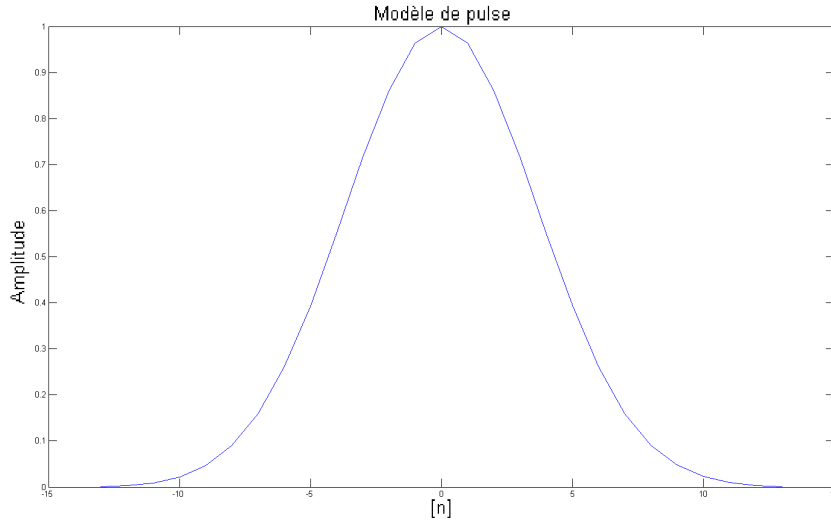


FIGURE 4.32 – Modèle de pulse

L'exemple de pulse à la figure 4.32, composé de 27 échantillons, produirait une matrice dont seulement les 27 diagonales de la matrice contiennent les informations pertinentes, soit la diagonale principale et les 13 diagonales de part et d'autre.

Gain supplémentaire par la symétrie matricielle

Toutefois, il est encore possible de réduire le nombre d'éléments de la matrice $\hat{\mathbf{R}}_{XX}$ à stocker. Pour le cas particulier des matrices de covariance, celles-ci sont toujours symétriques. Par conséquent, en stockant uniquement, la diagonale principale et les $\frac{m_p-1}{2}$ diagonales au-dessus ou au-dessous, toutes les informations utiles sont enregistrées.

Pour des réponses de 512 échantillons de longueur, la matrice de covariance, de dimensions $[512 \times 512]$, contient au total 262144 éléments. Avec la méthode de sauvegarde présentée précédemment et avec des pulses d'une durée de 27 échantillons, le nombre d'éléments sauvegardés est maintenant de 7077. La taille de mémoire requise pour stocker ces éléments représente donc 2.7% de la mémoire requise pour la matrice de covariance entière.

Lorsque la longueur des réponses augmente à 1024 échantillons, la matrice de covariance comporte 1048576 éléments. La méthode de sauvegarde réduit le nombre d'éléments à stocker à 14245, ce qui représente 1.36% de la matrice initial. Plus la portée du système est grande et plus cette méthode de sauvegarde engendre de gain en terme de réduction relative de mémoire.

Cas particulier d'échos étirés

Certaines cibles ont pour effet d'étirer dans le temps l'écho qu'il retourne. Autrement dit, l'écho retourné occupe plus d'échantillons que le pulse émis par le système. Ainsi, l'empreinte

dans la matrice de covariance de ce type d'écho est plus grande que pour un écho qui n'a pas été étiré. Par précaution, il peut être souhaitable d'enregistrer quelques diagonales supplémentaires pour s'assurer de ne pas perdre de l'information lorsque des échos ont subi un étirement temporel.

4.2.2 Décomposition propre

La décomposition propre (EVD, pour eigenvalue decomposition) de la matrice de covariance est non seulement une étape cruciale de l'algorithme, mais cette étape peut être coûteuse en temps et en ressources. Si on prend l'exemple d'implantation de Kumar & Mazhari[14], le temps et les ressources nécessaires pour obtenir la décomposition propre des matrices de la taille de celles engendrées par les signaux temporels de longue durée rendent son implantation laborieuse et exigent du matériel coûteux. Une particularité de l'algorithme permet toutefois d'alléger cette étape. Lors de la présentation du pseudospectre de MUSIC adapté, à la section 3.6 du chapitre 3, on concluait qu'il n'était pas nécessaire d'obtenir les vecteurs propres du bruit, formant le sous-espace propre \mathbf{E}_n . Seuls les vecteurs propres associés aux valeurs propres élevées, formant le sous-espace propre source \mathbf{E}_s , sont nécessaires puisque l'obtention du pseudospectre peut se faire via la projection dans l'espace bruit \mathbf{P}_n égale à :

$$\mathbf{P}_n = \mathbf{I} - \mathbf{E}_s \mathbf{E}_s^T \quad (4.25)$$

En règle générale, pour des applications de délais d'arrivée, le nombre de vecteurs formant \mathbf{E}_s est beaucoup plus petit que celui de \mathbf{E}_n . À titre d'exemple, les tests effectués avec le système de notre partenaire industriel indiquent que la taille de \mathbf{E}_s représente pratiquement toujours moins de 1% de celle de tous les vecteurs propres (\mathbf{E}_s et \mathbf{E}_n combinés). Une décomposition propre partielle de $\hat{\mathbf{R}}_{XX}$ qui permettrait d'obtenir seulement les vecteurs propres associés aux valeurs propres élevées (associées aux sources) formant \mathbf{E}_s économiserait une quantité de calculs considérable. La décomposition propre itérative est une solution envisageable ici puisque seules les plus grandes ou les plus petites valeurs propres sont recherchées. Son utilisation dans des situations usuelles, alors que tous les vecteurs propres sont recherchés, n'est pas très indiquée, car les multiples itérations nécessaires à l'obtention de chaque vecteur propre la rendrait extrêmement lente. Lorsqu'une décomposition propre totale est requise, des méthodes plus sophistiquées ont été développées, dont une multitude se retrouvent dans le livre de Golub & Van Loan[6]. Plusieurs méthodes itératives existent, comme le démontre cet article de l'Université de Waterloo[12]. Toutefois, celle utilisée de manière pédagogique qui semblait la plus appropriée à nos besoins est tirée du livre Applied Numerical Methods for Digital Computation[8]. La décomposition propre itérative peut être utilisée sur les matrices de covariance selon les étapes suivantes :

1. Tout d'abord, le lien entre la matrice et un vecteur propre est le suivant : $\hat{\mathbf{R}}_{XX} \mathbf{v}_i = \lambda_i \mathbf{v}_i$, où λ_i est la i^e valeur propre et \mathbf{v}_i son vecteur propre associé. La décomposition propre

itérative a besoin d'un vecteur de départ \mathbf{v}_0 , de même dimension que les vecteurs propres \mathbf{v}_i . Ce vecteur \mathbf{v}_0 peut généralement être d'un contenu quelconque, pour autant qu'il ne soit pas un vecteur nul. Dans certaines situations, son contenu peut être choisi pour accélérer la méthode.

2. La multiplication matricielle de $\hat{\mathbf{R}}_{XX}\mathbf{v}_0 = \mathbf{w}$ donne un vecteur \mathbf{w} .
3. Ensuite, on effectue une boucle telle que

$$\text{Normalisation de } \mathbf{w} \tag{4.26}$$

$$\mathbf{w}' = \hat{\mathbf{R}}_{XX}\mathbf{w} \tag{4.27}$$

$$\mathbf{e} = |\mathbf{w} - \mathbf{w}'| \tag{4.28}$$

$$\mathbf{w} = \mathbf{w}' \tag{4.29}$$

Il est à noter que le vecteur \mathbf{w}' doit être normalisé à chaque itération. Le vecteur \mathbf{w}' converge vers la combinaison $\lambda\mathbf{v}$ ayant la valeur propre la plus élevée. La condition d'arrêt de la boucle peut se faire sur le vecteur d'erreur \mathbf{e} , selon la précision souhaitée. La valeur d'erreur maximale trouvée à l'intérieur du vecteur d'erreur \mathbf{e} , baptisée e_{max} , est comparée à la tolérance permise. Lors de la normalisation de \mathbf{w}' , le facteur multiplicatif est égal à la valeur propre, car $\mathbf{w}' = \lambda\mathbf{v}$.

4. À la section 3.5, on expliquait que $\hat{\mathbf{R}}_{XX} = \sum_{i=1}^{m_X} \lambda_i \mathbf{v}_i \mathbf{v}_i^T$. Pour trouver la valeur propre suivante $i + 1$, il faut soustraire à la matrice la contribution la valeur propre trouvée à l'itération en cours i . La matrice de covariance devient $\hat{\mathbf{R}}_{XX} = \hat{\mathbf{R}}_{XX} - \lambda_i \mathbf{v}_i \mathbf{v}_i^T$. Avec cette nouvelle matrice, on retourne à l'étape 1 et on refait toutes les étapes pour trouver la combinaison valeur et vecteur propres suivante.

Pour résumer la décomposition propre itérative et la clarifier, le pseudocode suivant énumère les étapes.

- Boucle $i = 1 : M$ pour trouver les M valeurs et vecteurs propres désirés

$$\mathbf{w} = \hat{\mathbf{R}}_{XX}\mathbf{v}_0$$

Boucle tant que $e_w > e_{max}$, où e_w est l'erreur maximale obtenue du vecteur d'erreur \mathbf{e} , et e_{max} est l'erreur maximale acceptable

Normalisation de \mathbf{w}

$$\mathbf{w}' = \hat{\mathbf{R}}_{XX}\mathbf{w}$$

$$\mathbf{e} = |\mathbf{w} - \mathbf{w}'|$$

$$\mathbf{w} = \mathbf{w}'$$

Fin de la boucle

$$\lambda \mathbf{v}_i = \mathbf{w}, \text{ en normalisant } \mathbf{w}$$

$$\hat{\mathbf{R}}_{XX} = \hat{\mathbf{R}}_{XX} - \lambda_i \mathbf{v}_i \mathbf{v}_i^T$$

- Fin de la boucle

Comme l’algorithme produit les valeurs et vecteurs propres par ordre décroissant, il est possible d’arrêter la boucle principale de la décomposition propre itérative avec le critère de séparation du sous-espace propre. En effet, ce dernier détermine si une valeur propre fait partie de \mathbf{E}_s ou non. Par conséquent, lorsque la dernière valeur propre produite échoue face au critère, la boucle peut être stoppée puisque les valeurs propres suivantes sont toutes inévitablement plus faibles. Cette façon de faire est bien utile lorsque le nombre de sources est variable.

Résultats

À partir d’un historique de 8 réponses à un seul écho de SNR valant 24.8 dB, la matrice de covariance est approximée de la plus simple façon, via $\hat{\mathbf{R}}_{XX} = \mathbf{X}\mathbf{X}^T$. C’est à partir de ces données que les résultats suivants sont obtenus. La décomposition propre de référence est celle du logiciel Matlab™. La décomposition propre itérative a comme vecteur de départ \mathbf{v}_0 aux éléments nuls, sauf un de ceux-ci, n’importe lequel, dont la valeur vaut 1. L’erreur e_{max} permise vaut 0.0001. La figure 4.33 compare les 5 vecteurs propres de plus grande importance obtenus pour les 2 méthodes. Le tableau 4.3 compare les valeurs propres associées tout en indiquant le nombre d’itérations nécessaires à l’obtention de chaque combinaison valeur et vecteur propres.

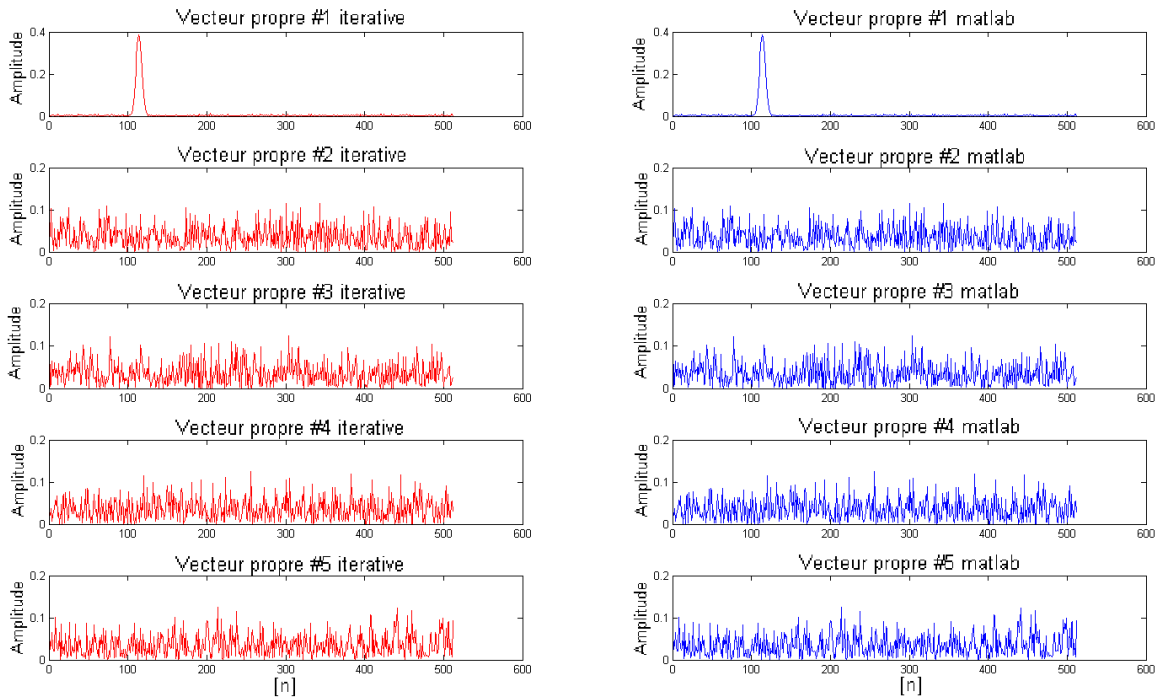


FIGURE 4.33 – Comparaison des 5 vecteurs propres de plus grande importance pour les 2 méthodes

Selon la dernière colonne du tableau, on remarque que l’obtention itérative des vecteurs propres associés au bruit nécessite généralement un nombre beaucoup plus élevé d’itérations que les vecteurs propres liés aux sources pour une même erreur tolérée. Cette propriété particulière

pourrait éventuellement être intégré au critère de discrimination des valeurs propres pour l'améliorer.

Comparaison des valeurs propres				
Valeur propre	EVD matlab	EVD itérative	% d'écart	Itérations requises
# 1	5238.1324	5238.1339	0.00003 %	4
# 2	49.3773	49.3763	0.002 %	142
# 3	48.0192	48.0186	0.0013 %	119
# 4	44.9691	44.9692	0.00038 %	239
# 5	43.4350	43.4357	0.0016 %	131

TABLE 4.3 – Comparaison des valeurs propres, $e_{max} = 0.0001$

Comparaison versus Matlab™

À titre indicatif, une comparaison avec la fonction de décomposition propre $eig()$ du logiciel Matlab™ est effectuée. Ce sont 10000 matrices de covariance qui sont fournies aux 2 méthodes de décomposition propre. La méthode $eig()$ doit effectuer la décomposition complète alors que la méthode itérative se limite à 12 vecteurs propres, car il serait très improbable qu'il y ait plus de 12 sources. La propriété du type de signaux utilisés voulant que le sous-espace propre \mathbf{E}_s soit de taille bien moindre que le sous-espace propre bruit \mathbf{E}_n est ici mise à profit. Les signaux contiennent 5 sources, le prétraitement de décorrélation est la modulation par la matrice des modèles et l'erreur maximale tolérée sur les vecteurs propres est de 0.0001, les signaux sont formés par 512 échantillons.

Pour la fonction $eig()$, ce sont 22.07 décompositions propres par seconde qui sont effectuées. La décomposition propre itérative permet d'augmenter cette fréquence à 33.41, un gain de 51.4 %. En abaissant l'erreur tolérée sur les vecteurs propres à 0.001, la méthode itérative augmente sa cadence à 47.2 décomposition par seconde, ce qui est 2 fois plus rapide que la version que la fonction de Matlab™. De plus, comme les opérations effectuées pour la méthode itérative sont sur des vecteurs, et donc hautement parallélisables, le gain sera certainement augmenté sur un système profitant de cet avantage à son maximum. Il faut aussi savoir que la fonction de Matlab™ nécessite un ordinateur (donc difficilement implantable sur FPGA, ne serait-ce que par la mémoire requise) et est optimisée selon le type de matrice passé en argument.

4.3 Conclusion sur les optimisations

Le but de ce chapitre était d'améliorer l'algorithme, autant du point de vue des performances que de la difficulté d'implantation.

Par rapport aux performances, l'algorithme procure des résultats bien meilleurs en y incluant les optimisations qu'avec la version initiale de MUSIC adapté présentée au chapitre 3. Une des optimisations parmi les plus marquantes est certainement l'ajout du prétraitement de décorrélation à la formation de la matrice de covariance approximée $\hat{\mathbf{R}}_{XX}$. Parmi les gains en découlant, on note une diminution d'environ 3 dB sur le SNR de détection minimal. Toutefois, le gain le plus fulgurant concerne certainement la discrimination d'échos superposés. Le test sur signaux réels démontre que d'un taux de succès d'environ 40% à un écart inter-échos de $5m$ obtenu avec la méthode classique peut être amélioré jusqu'à près de 100% à un écart inter-échos de $1.25m$ avec le pré-traitement de décorrélation de la modulation par la matrice modèle. Ainsi, sans dire que le pré-traitement de décorrélation est essentiel à l'algorithme, il lui est incontournable pour en tirer le maximum.

Par la suite, un autre point sensible de l'algorithme a été abordé : la sélection des valeurs propres sources pour la formation du sous-espace propre source. Le critère de sélection des valeurs propres doit être soigneusement choisi par l'utilisateur. En fonction de l'objectif ultime de l'algorithme, le critère peut donner une tangente aux détections de l'algorithme pour optimiser un aspect de la détection voulu, comme la minimisation des fausses détections par exemple. Dans ce chapitre, le critère proposé se voulait tout de même simple, visant une implantation facilitée. Ce critère conditionnel à 4 sous-parties fonctionne sur une plage de conditions plus étendue qu'un critère par seuillage simple, et ce plus particulièrement lorsque des échos d'une même réponse ont des amplitudes différentes. Le critère par seuillage simple est limité à un rapport des amplitudes de $1/0.63$, alors que le critère conditionnel double peut gérer des rapports d'amplitudes allant jusqu'à $1/0.07$. D'autre part, si un utilisateur veut optimiser un certain aspect des détections, il est possible de le faire en ajustant les paramètres de chacune des sous-parties du critère.

Un autre aspect des détections optimisé dans ce chapitre est la précision. La version initiale du chapitre 3 produisait un pseudospectre par pas d'une période d'échantillonnage. La précision des mesures était ainsi limitée à l'échantillon près. L'ajout de la précision sous-échantillon via la production d'un sous-pseudospectre autour d'un sommet à l'aide du balayage d'un modèle $\mathbf{a}(\tau)$ par pas de déphasage plus fin que l'échantillon permet d'obtenir une précision théoriquement limitée uniquement par la précision du sous-espace bruit. Pour la très grande majorité des tests effectués, le pas de déphasage était de 0.01 échantillon, donnant une précision de mesure au centième d'échantillon près. Le gain à ce niveau est donc appréciable, surtout que la précision est une qualité des détections souvent recherchée.

D'un point de vue de l'implantation de l'algorithme, la mise en mémoire proposée dans ce

chapitre pour la matrice de covariance $\hat{\mathbf{R}}_{XX}$ minimise grandement l'espace qu'elle requiert. Basée sur les propriétés de symétrie de la matrice et de l'emplacement des échos centrés sur les éléments de la diagonale principale, il devient possible de conserver une fraction de la matrice sans perdre aucune information utile. Plus les dimensions de la matrice sont grandes, plus cette méthode est avantageuse. L'économie de mémoire peut faire la différence lors d'une implantation sur un système en temps réel limité en ressources.

Un autre point facilitant l'implantation est d'utiliser la méthode de décomposition propre itérative. Tout d'abord, celle-ci réduit l'espace mémoire requis en obtenant uniquement quelques vecteurs propres, soit ceux étant des contributions des sources, plutôt que la matrice entière de tous les vecteurs propres. De plus, même en exigeant une fine précision, un gain important est observé par rapport à la fonction du logiciel Matlab™. En fait, la méthode itérative était 50% plus rapide. Ce gain est davantage élevé lors d'une implantation réelle car ses opérations mathématiques sont en très grande partie parallélisables.

Les performances globales de l'algorithme sont définitivement supérieures lorsque les optimisations décrites à ce chapitre sont incorporés à ce dernier. Ces optimisations procurent les caractéristiques habituellement attribuées à MUSIC, soit une haute résolution et une bonne immunité au bruit. Un autre point les rendant quasi-incontournables est que soit elles accélèrent l'algorithme ou bien elles améliorent les performances tout en demandant une augmentation minime en ressources.

Chapitre 5

Tests et résultats de l'algorithme

Le but de ce chapitre est de quantifier les gains réalisés par l'utilisation de l'algorithme MUSIC adapté pour détecter des délais d'arrivée sur des signaux n'ayant que pour seule information l'amplitude de l'enveloppe. Pour ce faire, cet algorithme est comparé à d'autres algorithmes couramment utilisés pour la même application. Tous ces algorithmes sont soumis à différents tests, révélant leur performance face à un paramètre de la qualité de détection.

5.1 Algorithmes comparatifs

Pour plusieurs tests, l'algorithme MUSIC-adapté se comparera à d'autres algorithmes. Le but de ces comparaisons est de quantifier les améliorations obtenues de MUSIC-adapté face aux algorithmes utilisés dans le même contexte. La corrélation est un algorithme général tandis que Double wavelet et RMS conv sont particuliers à notre partenaire industriel.

5.1.1 Corrélation

L'algorithme de détection par corrélation choisi est l'adaptation de celui-ci présentée à la section 2.3.1 du chapitre 2. Il s'agit d'un estimateur d'ordre 1 (par opposition aux algorithmes utilisant la matrice de covariance, qui sont d'ordre 2). Un spectre de corrélation est obtenu via :

$$P_{corr}(\tau) = \mathbf{s}(\tau)^T \mathbf{x}_{sum} \quad (5.1)$$

où le vecteur modèle de délai $\mathbf{s}(\tau)$ correspond à un modèle d'écho parfait reçu selon un délai de τ échantillons. Un exemple de ces vecteurs est disponible à la figure 3.2. Le vecteur \mathbf{x}_{sum} est tout simplement la somme de signaux de l'historique \mathbf{X} , obtenu par :

$$\mathbf{x}_{sum} = \sum_{i=1}^{n_X} \mathbf{x}_i \quad (5.2)$$

Pour tous les tests effectués, le nombre d'échos M à repérer dans les signaux est connu. Ainsi, les M sommets du spectre de corrélation P_{corr} les plus élevés sont ceux qui seront considérés comme les positions détectées par cet algorithme.

5.1.2 *Double wavelet*

Cet algorithme est propre à notre partenaire industriel, il ne sera donc pas décrit en détails, secret industriel oblige. Néanmoins, celui-ci débute par le passage successifs de 2 vaguelettes ayant un contenu adapté aux signaux reçus et au pulse émis. Finalement, c'est un seuillage sur la réponse résultante qui détecte la présence des échos. Cet algorithme permet d'obtenir une précision bien supérieure à l'échantillon près.

5.1.3 *RMS conv*

Cet algorithme est également propre à notre partenaire industriel, ses détails ne seront pas dévoilés, pour la même raison que précédemment. Ce dernier débute par une analyse de l'énergie du signal contenu dans une fenêtre glissante. Un seuillage sur le résultat de cette analyse détecte la présence d'échos. Finalement, c'est un traitement par vaguelettes qui détermine la position des échos. Cet algorithme permet lui aussi d'obtenir une précision au-delà de l'échantillon près, mais demeure d'ordre 1.

5.1.4 MUSIC

Pour cette section des tests et résultats, MUSIC-adapté est utilisé de la façon suivante. Tout d'abord, la matrice de covariance $\hat{\mathbf{R}}_{\mathbf{X}\mathbf{X}}$ est obtenue par la méthode de modulation par la matrice des modèles. Suite à la décomposition propre, le nombre de vecteurs appartenant au sous-espace signal \mathbf{E}_s est déterminé par le critère conditionnel double, présenté à la section 4.1.3.

5.2 Séparation des échos superposés

Un des défis à relever pour un algorithme de détection est de distinguer deux échos partiellement superposés, tel que mentionné dans l'introduction au projet. Ces tests permettent non seulement de vérifier si un des principaux objectifs du projet est atteint, soit de permettre au système de détecter des cibles rapprochées l'une de l'autre, mais aussi quantifient les performances pour ce paramètre. Le premier test compare les différents algorithmes de détection tandis que le deuxième test quantifie les performances de séparation d'échos en fonction du SNR de ceux-ci.

5.2.1 Comparaison à d'autres algorithmes

Les réponses utilisées sont les mêmes que pour le test présenté à la section précédente 4.1.2. Ces réponses sont enregistrées avec le système de notre partenaire industriel. La scène consistait en 2 cibles, disposées face au système à une distance facilement atteinte par le système, avec un écart variable entre elles. Pour chacun des écarts testés, entre 1000 et 1250 réponses ont été enregistrées. Les historiques étaient formés de 4 réponses pour ce test comparatif. Les 3 dernières réponses du dernier historique sont réutilisées dans l'historique de l'itération suivante, telle une fenêtre glissante. Un essai est considéré réussi lorsqu'un algorithme détecte séparément les deux cibles. À la fin de la série, le taux de succès est obtenu pour un écart particulier des 2 cibles. L'écart entre les 2 cibles est converti en terme de périodes d'échantillonnage séparant les 2 échos. Il est indiqué dans la colonne de gauche du tableau.

Comparaison des algorithmes pour la séparation d'échos superposés				
Algorithmes	MUSIC-adapté	<i>Double Wavelet</i>	<i>RMS conv</i>	Corrélation
Écart				
T_e				
13.33	100 %	100 %	100 %	100 %
12	100 %	100 %	100 %	0 %
11.33	100 %	95.87 %	100 %	0 %
10.67	100 %	0 %	0 %	0 %
10	100 %	0.84 %	0 %	0 %
⋮	⋮	⋮	⋮	⋮
5.33	100 %	0 %	0 %	0 %
5.067	100 %	0 %	0 %	0 %
4.8	100 %	0 %	0 %	0 %
4.53	33.48 %	0 %	0 %	0 %
4.27	0.32 %	0%	0 %	0 %
4	47.63 %	0 %	0 %	0 %
3.73	2.91 %	0 %	0 %	0 %
3.47	0 %	0 %	0 %	0 %
3.2	0 %	0 %	0 %	0 %
2.93	0 %	0 %	0 %	0 %
2.67	0 %	0 %	0 %	0 %

TABLE 5.1 – Taux de discrimination d'échos chevauchés des algorithmes selon l'écart inter-échos

Il est clair que l'algorithme MUSIC-adapté sépare des échos plus rapprochés que les autres

méthodes. En ce sens, un des objectifs du projet est atteint. Une telle sensibilité à reconnaître une superposition d'échos a toutefois un mauvais côté. Lorsqu'un pulse est réfléchi sur une cible qui étire l'écho, un problème survient si le nombre de vecteurs propres inclus dans le sous-espace propre source est supérieur au nombre d'échos véritables dans les signaux. À ce moment, le pseudospectre peut afficher deux pointes vis-à-vis la position de l'écho étiré, d'où l'apparition d'une fausse détection. Le critère de discrimination des valeurs propres devient un point crucial de l'algorithme à ce moment. Avec le bon nombre de valeurs propres identifiées comme appartenant aux sources, le pseudospectre affichera alors le bon nombre de pointes malgré un écho de forme élargie.

Pour les résultats de MUSIC-adapté, on constate une remontée inattendue du taux de succès à la distance de séparation de 1.5 *m*. Cette remontée soudaine peut être considérée comme une donnée aberrante, car toutes les valeurs de part et d'autres frôlent le 0%. Une seule hypothèse pourrait expliquer cette hausse, mais sa probabilité n'est pas très élevée. Celle-ci suppose que les deux échos sont à une distance correspondant à un multiple exact de la distance parcourue par l'onde sur une période d'échantillonnage (appelée phase nulle). On remarque que l'écart inter-échos correspond à un entier lorsque le taux de succès remonte inopinément. Ainsi, à ce moment, si le premier écho est de phase quasi-nulle, le second le sera également. Comme le pseudospectre est tracé par pas d'une période d'échantillonnage, et que le modèle du pulse utilisé est de phase nulle, il serait normal d'obtenir un meilleur résultat car le modèle de $\mathbf{a}(\tau)$ correspond mieux à l'écho de \mathbf{E}_s si les deux ont la même phase. Cela expliquerait pourquoi le taux de succès remonte. Toutefois, il s'agirait d'une coïncidence.

5.2.2 Séparation d'échos selon le SNR

Ce test ne concerne que l'algorithme MUSIC-adapté uniquement car étant l'algorithme choisi pour ce projet, il est utile de connaître la dégradation de la capacité à discriminer des échos chevauchés en fonction d'un SNR se dégradant. Le but de celui-ci est de quantifier la dégradation des performances en terme de discrimination d'échos partiellement superposés lorsque le SNR des signaux diminue. Les réponses utilisées pour ce test sont synthétiques (par simulations). Les historiques contiennent 4 réponses ayant chacune 2 échos de même amplitude. Le SNR de ces réponses est indiqué dans la première colonne du tableau 5.2 alors que le nombre d'échantillons séparant le sommet des 2 échos est indiqué sur la première ligne. Pour un SNR et un écart inter-échos donnés, 1000 historiques sont fournis à l'algorithme. Chaque historique forme un essai. Lorsque ce dernier produit un pseudospectre affichant 2 pointes séparées, aux positions respectives des échos plus une légère incertitude, cet essai est considéré comme réussi. Le tableau 5.2 indique donc le taux de succès de l'algorithme à détecter les 2 échos en fonction de leur rapport signal à bruit.

Taux de réussites pour la séparation des échos superposés à différents SNRs pour MUSIC-adapté						
	Écart [échantillons]					
	4	5	6	7	8	9
SNR						
44.8 dB	100 %	100 %	100 %	100 %	100 %	100 %
34.8 dB	100 %	100 %	100 %	100 %	100 %	100 %
24.8 dB	100 %	100 %	100 %	100 %	100 %	100 %
19.8 dB	100 %	100 %	100 %	100 %	100 %	100 %
13.8 dB	97 %	100 %	100 %	100 %	100 %	100 %
7.8 dB	53.3 %	86.2 %	98.7 %	100 %	100 %	100 %
1.8 dB	16.2 %	32.2 %	61.2 %	88 %	98.3 %	99.9 %
-4.2 dB	0.08 %	2.3 %	6.8 %	23.3 %	52.2 %	73.8 %
-10.2 dB	0 %	0.4 %	0.2 %	1.5 %	2.1 %	3.6 %

TABLE 5.2 – Taux de réussites pour la séparation des échos superposés à différents SNRs pour MUSIC-adapté

Il faut rappeler que pour ces tests, aucun critère de discrimination des sous-espaces propres \mathbf{E}_s et \mathbf{E}_n n'est en place. Comme le nombre d'échos présents dans les réponses aux pulses est préalablement connu, un nombre égal de vecteurs propres est directement placé dans le sous-espace source \mathbf{E}_s . De ce fait, les performances affichées dans le tableau ci-dessus sont donc optimales. En réalité, avec un SNR qui se dégrade, il devient de plus en plus complexe de déterminer le nombre d'échos retournés. Ainsi, il est probable qu'en réalité, le critère de discrimination des sous-espaces fassent défaut par moment, ne permettant pas d'obtenir des performances aussi élevées que celles de ce test, où le nombre d'échos est préalablement connu.

5.3 Rapport signal à bruit de détection minimum

Ce test est effectué sur des réponses aux pulses réelles obtenues grâce au système de notre partenaire industriel. L'acquisition des mesures consistait à placer une cible à une distance connue et à enregistrer un nombre élevé de réponses, toujours au-delà de 1000. Comme le système de notre partenaire industriel est paramétrable, il était possible de diminuer le SNR des réponses sans déplacer la cible. Chacune des réponses fournies par le système a été pré-accumulée un certain nombre de fois. Plus le nombre d'accumulations est élevé et meilleur est le SNR. Ainsi, sans bouger la cible, il a été possible d'enregistrer la même scène mais en modifiant le nombre d'accumulations des réponses. Ce nombre est indiqué dans la colonne de gauche des tableaux 5.3 et 5.4 ci-dessous. Le SNR moyen des réponses enregistrées est indiqué dans la deuxième colonne, et est calculé selon la définition présenté en annexe A.1.

Pour le premier test, les historiques contiennent 4 réponses, et chaque historique traité forme un essai. Les historiques de deuxième test sont composés de 2 réponses. Lorsqu'un algorithme détecte l'écho à ± 1 période d'échantillonnage de la position exacte de ce dernier, l'essai est valide, autrement il est invalide. Le taux de détection pour un nombre d'accumulations donné est indiqué dans le tableau 5.3 pour chacun des algorithmes testés. Pour chaque nombre d'accumulations testé, ce sont toujours plus de 1000 historiques réels qui sont fournis aux algorithmes.

Pour ce premier test, des historiques de 4 réponses étaient formés, dont les 3 dernières du présent historique sont réutilisées pour former l'historique de l'itération suivante, à la manière d'une fenêtre glissante.

Comparaison du SNR de détection minimal sur signaux réels, $K=4$					
	Algorithmes	MUSIC	<i>Double Wavelet</i>	<i>RMS conv</i>	Corrélation
Accumulations	SNR moyen				
32	10.83 dB	98.16 %	18.82 %	90.24 %	79.89 %
28	14.41 dB	86.24 %	73.31 %	84.15 %	94.55 %
24	14.19 dB	81.41 %	78.00 %	65.65 %	94.67 %
20	12.22 dB	82.46 %	49.97 %	67.50 %	92.90 %
16	13.55 dB	78.32 %	50.28 %	48.66 %	95.69 %
12	7.90 dB	91.65 %	8.10 %	26.82 %	98.36 %
10	11.53 dB	82.40 %	50.18 %	26.35 %	98.86 %
8	9.60 dB	87.69 %	21.63 %	10.93 %	93.10 %
6	10.10 dB	83.3 %	21.09 %	10.55 %	89.84 %
5	5.85 dB	83.45 %	2.70 %	2.53 %	81.71 %
4	6.06 dB	84.64 %	1.57 %	2.23 %	94.72 %
2	0.45 dB	78.70%	0.32 %	1.30 %	85.51 %
1	-1.95 dB	51.91 %	0 %	0 %	64.56 %

TABLE 5.3 – Taux de détection à ± 1 échantillon des algorithmes en fonction du nombre de pré-accumulations des réponses, historiques de 4 réponses

Autant dans le tableau 5.3 que le tableau 5.4, on peut remarquer que le SNR de signaux ne diminue pas toujours suivant le nombre d'accumulations. Cette contradiction est explicable par deux raisons. Tout d'abord, le SNR des échos est calculé selon la position de ce dernier. Toutefois, cette position peut légèrement changer d'un enregistrement à l'autre. Ainsi, quelques points de l'écho et du bruit peuvent être intervertis dans le calcul, abaissant le SNR réel de l'écho. Ensuite, au lieu d'enregistrer une grande quantité de réponses ayant été préaccumulées une seule fois et de les accumuler un nombre de fois désiré par la suite lors du test, nous avons recalibré le nombre d'accumulations de l'appareil et enregistré des séries de réponses différentes

à chaque nombre d'accumulations. Ainsi, les réponses des différents nombres d'accumulations ont été enregistrées à différents moments. Par conséquent, les variantes de l'environnement peuvent avoir variées (lumière de la fenêtre, niveau de bruit, etc...). De par cette raison, il est possible que le SNR des réponses ait réellement varié, expliquant aussi pourquoi les taux de succès peuvent momentanément remonter alors que le nombre de pré-accumulations baisse. Néanmoins, le but de ce test est d'effectuer une comparaison entre les algorithmes. Dans cette optique, les résultats sont fiables car pour un même nombre de pré-accumulations, les algorithmes fonctionnent dans les mêmes conditions sur les mêmes réponses, rendant la comparaison inter-algorithme valide.

Afin d'augmenter la difficulté d'effectuer une détection, les historiques seront formés uniquement de 2 réponses pour le test suivant.

Comparaison du SNR de détection minimal sur signaux réels, $K=2$					
	Algorithmes	MUSIC	<i>Double Wavelet</i>	<i>RMS conv</i>	Corrélation
Accumulations	SNR moyen				
32	10.83 dB	94.41 %	12.77 %	55.02 %	83.48 %
28	14.41 dB	85.60 %	43.43 %	37.82 %	92.69 %
24	14.19 dB	84.67 %	40.14 %	25.16 %	93.18 %
20	12.22 dB	84.29 %	21.47 %	25.64 %	88.88 %
16	13.55 dB	81.45 %	17.28 %	14.95 %	91.47 %
12	7.90 dB	84.65 %	1.71 %	6.70 %	93.05 %
10	11.53 dB	84.33 %	9.51 %	6.76 %	94.56 %
8	9.60 dB	80.25 %	1.49 %	1.95 %	84.50 %
6	10.10 dB	79.82 %	2.34 %	2.24 %	82.26 %
5	5.85 dB	69.04 %	0 %	0.26 %	72.96 %
4	6.06 dB	75.27 %	0 %	0.25 %	83.43 %
2	0.45 dB	53.45 %	0 %	0.32 %	68.47 %
1	-1.95 dB	35.07 %	0 %	0 %	38.78 %

TABLE 5.4 – Taux de détection à ± 1 échantillon des algorithmes en fonction du nombre de pré-accumulations des réponses, historiques de 2 réponses

À la vue de ces résultats, il est évident que l'algorithme MUSIC adapté et optimisé dispose d'un net avantage sur les algorithmes de premier ordre. Seule la corrélation semble bien performer. Il faut toutefois relativiser ses performances. En effet, si les 3 autres algorithmes n'ont préalablement aucun indice du nombre de sources à détecter, ce n'est pas le cas pour la corrélation. Cet algorithme n'ayant pas de critère de détermination du nombre de sources, un seul pic, son plus élevé, était choisi comme unique détection. Cette façon de faire a favorisé cet algorithme puisqu'un critère par seuillage par exemple, aurait exigé un compromis entre le

taux de détection et le nombre de fausses détection. En effet, pour obtenir un aussi bon taux de succès, son niveau de seuil aurait dû être assez bas, causant bien des fausses détections, ce qui n'est pas le cas pour les 3 autres algorithmes.

En conclusion, l'algorithme MUSIC-adapté fournit non seulement des détections par faibles SNRs, mais ses détections ont un niveau de fiabilité élevé.

5.4 Précision

La précision des mesures peut être un des critères des plus importants pour un utilisateur du système de détection. Le prochain test compare les performances de MUSIC-adapté face aux autres algorithmes.

Comme ce test utilise des données réelles, le système de détection de notre partenaire industriel est encore utilisé pour les enregistrer. Une seule cible était disposée à une distance connue, puis une série de réponses de la scène est mise en mémoire pour chacune des distances testées. Celles-ci sont comprises entre 10 et 25 mètres, par incrément de 25 centimètres. Pour chacune d'elles, ce sont entre 150 et 300 réponses qui sont enregistrées.

Une fois les mesures de distances obtenues des algorithmes, celles-ci sont filtrées, de façon à éliminer les fausses détections.

5.4.1 Exactitude des mesures

Une statistique intéressante à analyser est l'exactitude des mesures d'un algorithme. Celle-ci est obtenue en calculant la différence entre la mesure obtenue et la distance réelle à laquelle la cible était du système. Pour chacune des distances de la cible, une moyenne de ces différences est calculée. Il est à noter qu'une absence de détection et que les fausses détections ne sont pas tenues en compte dans le calcul de l'exactitude moyenne. Ainsi, la figure ci-dessous illustre l'écart moyen entre la distance réelle et les mesures obtenues de chacun des algorithmes selon la distance de la cible. Il faut savoir que le rapport signal à bruit des échos retournés diminue au fur et à mesure que la distance de la cible augmente. Il est donc normal d'observer une détérioration des performances de tous les algorithmes pour une cible de plus en plus lointaine. Les données de ce graphique sont disponibles en annexe à la section A.2.1.

Il ne ressort aucun algorithme exceptionnel quant à la précision obtenue. En effet, ils démontrent tous des performances semblables, sauf peut être Double Wavelet, qui éprouve plus de difficultés lorsque le SNR des signaux diminue. MUSIC-adapté démontre de bonnes performances, sa courbe se situant généralement quelque peu sous les autres. Son exactitude de

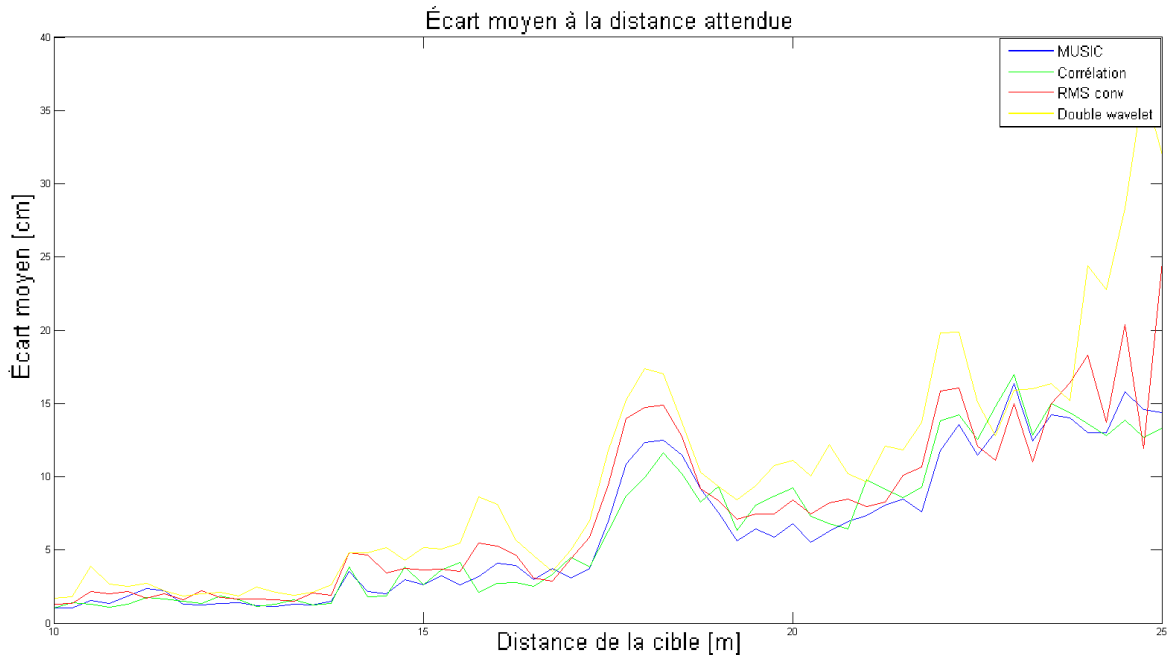


FIGURE 5.1 – Écart moyen des mesures avec la distance réelle de la cible

mesure serait donc légèrement meilleure.

Pour les distances de la cible situées entre environ 17 et 19 mètres, une augmentation subite de l'écart moyen selon la distance est présente et ne suit pas la tendance régulière de l'augmentation de l'écart. Cette irrégularité est observée pour tous les algorithmes. Ainsi, la source de l'erreur proviendrait de la prise des mesures. Pour cette plage de distances, tous les algorithmes fournissaient des distances plus courtes que la distance attendue, et non de plus grands écarts de part et d'autre de la distance attendue. Comme la cible était placée manuellement à la distance désirée sur un ruban à mesurer souple, il se peut très bien qu'un déplacement du ruban ait pu se produire. En plus, si celui qui place la cible a une tendance à la poser quelques centimètres devant la distance réelle, il devient fort possible que la distance de référence, qui est mesurée, soit inexacte, et plus fortement dans la plage des distances de 17 à 19 mètres.

5.4.2 Dispersion des mesures

Une autre statistique intéressante découlant de ces tests est la dispersion des mesures obtenues pour un algorithme. Les données de ce graphique sont disponibles en annexe à la section A.2.2. Pour une distance donnée de la cible, il est vrai qu'un algorithme peut donner une distance quelque peu à côté de la distance réelle, mais un autre aspect de la précision est de détecter la cible toujours à la même distance. Lors de l'enregistrement des données réelles, une fois la cible disposée à la distance souhaitée, celle-ci est totalement immobile. La dispersion des mesures est

simplement calculée selon l'écart type des mesures obtenues d'un algorithme. Une dispersion des mesures minimale est une qualité recherchée d'un algorithme de détection. En effet, plus la dispersion est grande et moins la fiabilité d'une mesure est bonne. De plus, une cible immobile peut avoir une fausse apparence de mouvement si les mesures de position utilisées pour calculer sa vitesse sont différentes d'un essai à l'autre. La figure ci-dessous affiche l'écart type obtenu en fonction de la distance à laquelle se situe la cible pour chacun des algorithmes.

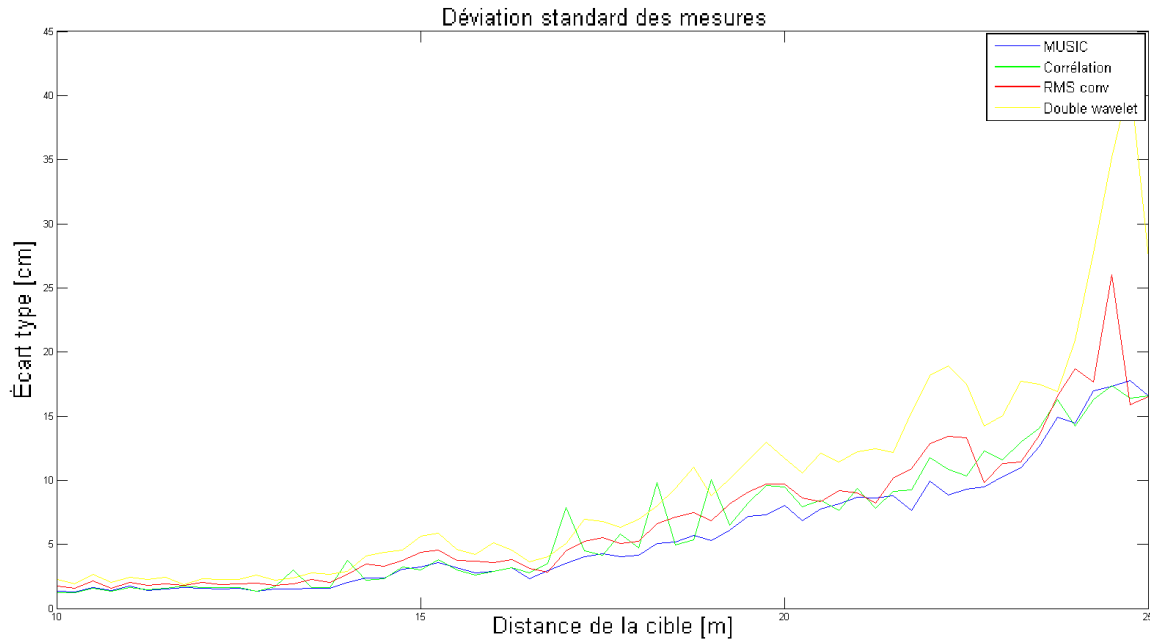


FIGURE 5.2 – Écart type des mesures selon la distance de la cible

Tout comme pour l'exactitude des mesures, aucun algorithme ne ressort avec éclat du lot. Toutefois, MUSIC-adapté affiche la courbe la plus basse pratiquement partout, indiquant donc une dispersion des mesures plus faible.

5.5 Résolution

La précision des mesures est un paramètre important de la qualité des détections. L'étape de l'algorithme MUSIC-adapté ayant le plus d'impact sur la précision obtenue est la résolution sous-échantillon, présentée à la section 4.1.5. À cette étape, le pas de déphasage δ détermine quelle variation sous-échantillon subit le modèle de pulse de $\mathbf{a}(\tau)$ à chaque essai.

Le but de ce test est d'observer l'impact du paramètre δ sur la précision des mesures. Pour chacun des rapports signal à bruit indiqué dans la colonne de gauche, ce sont 500 historiques contenant 4 signaux au SNR indiqué qui sont fournis à l'algorithme. Le pas de déphasage δ utilisé est indiqué sur la première ligne du tableau. Les cases du tableau indique l'erreur

moyenne, en fraction de la période d'échantillonnage T_e , entre le résultat obtenu de l'algorithme et la position exacte de l'écho. Afin d'être certain de vérifier toute la plage de déphasages possibles, le déphasage de l'écho des réponses est aléatoire d'un essai à l'autre.

Écart-moyen des mesures en fraction de période d'échantillonnage				
δ	0.125	0.0313	0.0078	0.0020
SNR				
50 dB	0.0159	0.0077	0.0017	0.0014
40 dB	0.0158	0.0083	0.0046	0.0041
30 dB	0.0206	0.0146	0.0135	0.0132
20 dB	0.0452	0.0426	0.0429	0.0450
10 dB	0.1189	0.1207	0.1178	0.1161
0 dB	0.4022	0.3882	0.3816	0.3936

TABLE 5.5 – Écart moyen des mesures en fonction du pas déphasage δ à différents SNRs

Un pas de déphasage δ très fin a un impact de plus en plus négligeable sur la précision moyenne des mesures au fur et à mesure que le SNR des signaux s'abaisse. Avec des historiques de 4 réponses, il n'y a pratiquement aucun gain à utiliser un pas plus fin que 0.125 échantillon lorsque le SNR est ou égal ou inférieur à 20 dB. Toutefois, lorsque le SNR augmente, un pas de déphasage plus fin procure un net avantage. À 50 dB par exemple, l'écart moyen à la mesure exacte est réduit d'au moins un facteur 11 entre $\delta = 0.125$ et $\delta = 0.0020$. Il faut donc choisir le bon compromis entre la finesse de δ et le coût en calculs en fonction de l'objectif du système de détection et des conditions de signaux attendues.

5.6 Conclusion des tests et résultats

Avant de passer à l'implantation d'un algorithme, il est primordial d'en évaluer les bénéfices. Ainsi, ce chapitre avait pour principal objectif de mesurer les gains procurés par l'algorithme MUSIC adapté en temps pour des signaux réels. Par la même occasion, certains tests procurent des données évaluant les limites et les performances de l'algorithme dans plusieurs situations. Pour ce faire, le nouvel algorithme MUSIC-adapté a été comparé aux algorithmes de notre partenaire industriel, double wavelet et rms conv, et à un algorithme de premier ordre, soit la corrélation adaptée aux délais d'arrivée, présenté à la section 2.3.

Une situation les plus complexes à gérer lors de détections est le cas où des échos sont en partie superposés. Une meilleure discrimination de ce type de cible était d'ailleurs un des principaux objectifs de ce projet de maîtrise et a orienté le choix d'algorithme vers MUSIC. Un test de discrimination de 2 cibles sur signaux réels a permis une comparaison entre les algorithmes de leurs taux de succès de séparation en fonction de la distance séparant les 2 cibles. Les résultats

obtenus par l'algorithme de corrélation en font un dernier de classe. Viennent ensuite à égalité les 2 algorithmes de notre partenaire industriel. Lors de la présentation des algorithmes de 2^e ordre, dont MUSIC fait partie, à la section 2.4, la haute résolution de ces algorithmes était mise en évidence et elle était une raison majeure justifiant le choix de ce type d'algorithme. Ainsi, MUSIC-adapté a réussi à séparer, avec un taux de succès de 100%, deux cibles séparées par 4.8 périodes d'échantillonnage, ce qui est 2.36 fois et 2.78 fois plus rapproché que les résultats respectifs des algorithmes de notre partenaire industriel et de la corrélation. Ainsi, l'algorithme MUSIC-adapté est fidèle à sa réputation, produisant une résolution supérieure et comblant un des principaux objectif du projet, soit d'obtenir une meilleure discrimination des cibles présentant des échos partiellement superposés. Finalement, il est important de savoir que la capacité de discrimination d'échos chevauchés s'abaisse en même temps que le SNR des signaux. Grossièrement, pour obtenir un taux de succès de 100% avec MUSIC-adapté, et des historiques de 4 réponses, l'écart inter-échos s'agrandit d'environ 2 échantillons à chaque baisse de 6 dB des réponses de l'historique.

Le second aspect des détections évalué dans ce chapitre est le seuil de SNR minimal nécessaire à la détection. Cet aspect de la détection est important puisqu'il augmente la portée utile d'un système de détection tout en permettant de détecter plus de cibles. Le test de comparaison est effectué sur des signaux réels, où une seule cible est présente, et dont l'écho est de plus en plus faible. Par rapport aux algorithmes double wavelet et rms conv, MUSIC-adapté abaisse le seuil limite de détection d'au moins 10 dB. Autrement dit, pour un bruit de même amplitude, il peut détecter un écho dont l'amplitude est 3.16 fois plus faible. L'algorithme de la corrélation affiche des performances comparables à celle de MUSIC. Toutefois, celui-ci a été avantaagé car seul le sommet le plus élevé a été considéré. Ceci est un avantage majeur car on ne devrait pas connaître a priori le nombre d'échos présents dans les réponses. Ainsi, pour que la corrélation affiche les mêmes performances que celles présentées à ce test, il faudrait un critère de seuil assez bas, menant à une augmentation des fausses détections. Bref l'algorithme MUSIC-adapté permet d'obtenir des détections par faibles SNRs tout en limitant les fausses détections.

Le test de la précision des mesures est divisé en 2 parties. Le premier test consistait à obtenir l'écart moyen des distances données par les algorithmes avec la distance réelle de la cible, autrement dit il servait à comparer l'exactitude des mesures des algorithmes. Le deuxième test mesurait la fiabilité, ou la constance si on peut dire, des mesures des algorithmes en calculant l'écart type en fonction de la distance de la cible. Les conclusions tirées de ces 2 tests sont les mêmes. Tout d'abord, aucun des algorithmes n'est exceptionnel, que ce soit positivement ou négativement. En fait, leurs exactitudes et constances en fonction d'un SNR se dégradant se ressemblent fortement. On pourrait dire que MUSIC-adapté a une tendance à mieux performer et que double wavelet accuse un léger retard sur le groupe, mais aucun des algorithmes ne procure une précision ressortant du lot.

Le dernier test concerne uniquement l'algorithme MUSIC. Comme l'étape de la résolution

sous-échantillon, présentée dans le chapitre des optimisations à la section 4.1.5, a un impact significatif sur la précision obtenue, les résultats permettent de déterminer l'utilité d'un pas de déphasage plus fin en fonction du SNR. Les données de ce tableau le prouvent : plus le SNR des signaux à traiter diminue et moins la finesse du pas de déphasage δ a un impact significatif. Sur les signaux synthétiques testés, on constate que sous la barre des 20 dB pour des historiques à 4 signaux, il n'y a aucun avantage à utiliser $\delta = 0.0020$ plutôt que $\delta = 0.1250$. Toutefois, avec des signaux de $SNR = 50$ dB, la précision peut être plus de 11 fois meilleure pour ces mêmes 2 valeurs de pas de déphasage. Comme un pas plus fin augmente le nombre de calculs nécessaires à l'obtention d'un résultat, il faut donc s'assurer de ne pas choisir un pas trop fin pour les conditions du système. Un pas trop grossier n'est pas souhaitable non plus car une incertitude supplémentaire est ajoutée aux mesures.

En résumé, l'algorithme MUSIC-adapté apporte des gains majeurs sur les aspects de la discrimination d'échos partiellement superposées et sur le SNR minimal de détection, tout en limitant les fausses détections et en procurant une précision comparable ou quelque peu supérieure aux autres algorithmes comparés.

Chapitre 6

Conclusion

Démarrage du projet

Lors la première rencontre d'ingénierie avec notre partenaire industriel, ce dernier nous a informés des problèmes rencontrés avec son appareil de détection de pulses par le principe du temps de vol. Le scénario de détection posant une difficulté majeure survenait lorsque 2 cibles étaient rapprochées de telle façon que leurs échos retournés se superposent partiellement dans les signaux reçus. Cette situation est délicate car elle déjoue les algorithmes de détection plus communs, tel que le seuillage par exemple, et mène à la non détection d'une des 2 cibles. À partir de ce constat, le principal mandat de ce projet de maîtrise était donc de résoudre le problème de la détection des échos partiellement superposés. D'autres facettes des méthodes de détection utilisées par l'entreprise étaient aussi à améliorer, comme le rapport signal à bruit minimum requis pour obtenir une détection et le nombre de fausses détections. Pour englober tous ces aspects de la performance des détections, le but du projet était donc d'élaborer un algorithme de détection du délai d'arrivée de pulse sans information de phase. Celui-ci devait procurer en premier lieu une haute résolution pour la séparation d'échos se chevauchant, être capable de repérer des cibles dont le SNR de leur écho est faible tout en ayant une complexité calculatoire limitée pour permettre une implantation sur un système de détection en temps réel de faible coût.

Démarche suivie et résultats obtenus

Le processus de l'élaboration d'un algorithme apportant une solution aux problèmes décrits par les partenaires de l'entreprise a débuté par une revue des algorithmes existants. Ceux liés à l'application visé ont été étudiés, mais aussi des algorithmes provenant de d'autres applications. Parmi ceux-ci, les méthodes de deuxième ordre ont ressorti du lot parce qu'elles possède la propriété de fournir une haute résolution aux détections. Au final, c'est l'algorithme MUSIC qui performait le mieux dans les conditions d'utilisation prévues. Ce dernier était à la base utilisé pour du traitement d'antenne, sous forme d'un réseau de capteurs. Bien que des applications étaient déjà existantes pour des applications de détermination de délais

d'arrivée selon le principe du temps de vol, celui-ci étaient toutes conçues en fonction de signaux complexes, donc ayant une information de phase. Malheureusement, les signaux du système de détection de notre partenaire industriel ont seulement l'information d'amplitude de l'enveloppe des signaux, rendant inopérant les versions trouvées de MUSIC.

Une adaptation de l'algorithme MUSIC était indispensable pour le rendre opérant sur le type de signaux dont nous disposions. La première étape fut de transformer le modèle des signaux. Des vecteurs de directions $\mathbf{a}(\phi)$ représentant les signaux selon un certain angle d'arrivée sur le réseau de capteurs, nous sommes passés à des vecteurs de délais $\mathbf{a}(\tau)$, représentant l'allure d'un signal dont le pulse est retardé selon une période τ . C'est autour de ces derniers que le passage à un algorithme fonctionnel s'est articulé. Le résultat est un algorithme MUSIC fonctionnel avec des signaux réels, sans information de quadrature.

En traitement d'antenne, une amélioration importante apportée au fonctionnement de l'algorithme est le lissage spatial. Ce dernier modifie la manière dont est créée la matrice de covariance et permet à l'algorithme d'opérer malgré des signaux en entrée fortement corrélés, voire cohérents. Le lissage spatial tel qu'utilisé en traitement d'antenne n'étant pas profitable aux délais d'arrivée sur signaux d'amplitude seulement, d'autres prétraitements de décorrélation de la matrice de covariance ont été élaborés. Parmi ceux-ci, la modulation par la matrice des modèles semble être le choix numéro un, autant pour la discrimination des échos partiellement superposés que pour l'atteinte du plus bas SNR critique de détection. Pour la séparation d'échos chevauchés, le pré-traitement de décorrélation permet de discriminer 2 échos chevauchés par un écart 4 fois plus mince que sans pré-traitement. Au point de vue du seuil minimal de rapport signal à bruit pour détecter un écho, c'est un gain de 3 dB qui est apporté par ce même pré-traitement de décorrélation.

Le pré-traitement de décorrélation a par la suite un effet favorable sur la décomposition propre de la matrice de covariance. Sans celui-ci, toutes les contributions des sources se retrouvent dans le même vecteur propre, et une seule valeur propre affiche une contribution des sources. Dans ce cas, il est impossible de déterminer le nombre de sources. La décorrélation corrige ce problème et un critère de sélection des valeurs propres sources permet de bien séparer les 2 sous-espaces propres source \mathbf{E}_s et bruit \mathbf{E}_n . Dans le chapitre 4 sur les optimisations, un critère conditionnel double est proposé. Ce dernier a l'avantage d'être simple, et par conséquent d'être implanté efficacement. Il fonctionne sur une très grande variété de situations et offre un gain notable par rapport à un critère par seuillage simple, comme pour des échos d'un même signal ayant des amplitudes très différentes. Dans ce cas, le rapport des amplitudes peut être 2 fois plus élevé avec le critère conditionnel double qu'avec un seuil. Finalement, comme il comporte plusieurs sous-parties liées à différentes situations, il est facilement ajustable en vue d'optimiser certains aspects de la qualité des détections.

Une autre optimisation majeure de l'algorithme est l'ajout de la précision au-delà de l'échan-

tillon près. Cette étape post-pseudospectre permet d'obtenir une résolution de mesure limitée uniquement par la précision du sous-espace bruit. Dans les faits, la résolution obtenue est moindre car la présence de bruit dans les signaux rend inutile une résolution sous-échantillon très fine. Pour des historiques à 4 réponses à un SNR inférieur à 20 dB, l'écart moyen des mesures avec la distance réelle demeure le même, malgré que la résolution sous-échantillon passe de 0.125 à 0.002. La conclusion est que plus le SNR augmente, plus une résolution sous-échantillon fine apporte de meilleurs résultats. À 50 dB par exemple, l'écart moyen est réduit par un facteur supérieur à 11 lorsque le pas de balayage sous-échantillon passe de 0.125 à 0.002.

Les optimisations des ressources utilisées par l'algorithme se retrouvent au niveau du stockage de la matrice de covariance et de la décomposition propre. La propriété de la matrice de covariance de contenir les informations utiles uniquement sur les quelques diagonales de part de d'autre de la diagonale principale permet de rejeter une majeure partie des triangles inférieurs et supérieurs. De plus, ces matrices étant en tout temps symétriques, on peut rejeter soit les diagonales utiles du triangle supérieur ou inférieur. Plus les dimensions de $\hat{\mathbf{R}}_{XX}$ sont grandes, plus la réduction de l'espace de stockage est significative. À titre indicatif, pour une matrice de dimensions $[1024 \times 1024]$, les éléments enregistrés représentent 1.36% des éléments de la matrice initiale. Du côté de la décomposition propre, c'est la propriété liant le projecteur du sous-espace bruit \mathbf{P}_n au sous-espace source \mathbf{E}_s qui permet de réduire les ressources matérielles et d'accélérer la décomposition propre. Comme $\mathbf{P}_n = \mathbf{I} - \mathbf{E}_s \mathbf{E}_s^T$, il n'est pas nécessaire d'effectuer une décomposition propre complète puisque seuls les vecteurs propres sources, associées aux valeurs propres de plus grandes valeurs, sont requis. Ainsi, la décomposition propre itérative obtient les vecteurs propres des contributions les plus élevées en quelques itérations sur chacun des vecteurs. Sans parallélisation totale puisqu'effectuée sur Matlab™, une comparaison permet d'observer une rapidité 50% supérieure pour la décomposition propre itérative par rapport à la fonction eig() de Matlab™.

La comparaison de l'algorithme MUSIC adapté et optimisé permet de conclure que ce dernier a permis à ce projet de maîtrise d'atteindre les objectifs fixés initialement. Le principal but à atteindre était de mieux distinguer 2 cibles dont les échos se superposent partiellement car cette situation est problématique du fait qu'une des cibles peut facilement être ignorée par le système de détection. Avec des signaux réels enregistrés, l'algorithme MUSIC-adapté sépare des cibles dont l'écart entre les échos est 2.36 fois plus mince que pour son plus proche concurrent. Le gain de performance est notable à ce niveau. Une autre qualité recherchée d'un algorithme de détection est le rapport signal à bruit requis pour effectuer une détection. Avec MUSIC-adapté, le SNR minimal de détection est inférieur d'au moins 10 dB à celui des 2 algorithmes présentant des taux de fausses détections semblables. Un autre algorithme affichait un seuil minimal comparable à MUSIC, mais dont le taux de fausses détections s'avère plus élevé.

Globalement, l'adaptation de MUSIC a procuré un algorithme résolvant le cas de détection complexe d'échos en partie superposés. Ses performances procurent une amélioration de pratiquement tous les aspects de la qualité des détections du système de détection. Les optimisations pour l'implantation proposées seront d'une grande aide pour ajouter cet algorithme dans le système de détection en temps réel et à un coût raisonnable.

Futur du projet

À court terme, l'étape incontournable est d'implanter l'algorithme adapté et optimisé sur le système de notre partenaire industriel. La courte période de temps allouée au calcul d'un résultat pour l'algorithme ainsi que le peu de ressources matérielles disponibles sur le système font donc de cette implantation un véritable défi. Quelques idées d'optimisation des ressources qui n'ont pas été abordées dans ce mémoire feront l'objet de tests, et seront éventuellement partie prenante du design final dans le cas où les tests sont concluants. Parmi celles-ci, une réduction de l'espace mémoire requis pour le projecteur sur le sous-espace bruit P_n serait souhaitable car les dimensions de cette matrice sont élevées, et donc problématiques pour le système. D'autre part, l'obtention d'une résolution sous-échantillon peut utiliser une structure plus efficace.

À plus long terme, il y a quelques optimisations des performances qui n'ont pas été élaborées plus que de mettre l'idée sur papier. Celles-ci concernent différents aspects des détections. Le premier pas dans cette direction sera donc de les tester en simulation afin de prouver leur bon fonctionnement et leur gain. Si une optimisation apporte un gain suffisamment important, il est certain que nous essayerons de l'ajouter au design du système de détection.

Une application utilisant l'algorithme allant au-delà de la simple détection des cibles serait de reconnaître grossièrement différents types de cibles. L'extraction de certaines caractéristiques des échos pourrait potentiellement être faite à l'intérieur même de l'algorithme. Comme une corrélation existe entre certains types de cibles et les caractéristiques de l'écho, il serait intéressant d'effectuer un triage grossier des cibles.

En ajoutant une couche logicielle au-dessus de l'algorithme, la reconnaissance des cibles pourrait être raffinée grâce à un système à plusieurs capteurs. Le système actuel de notre partenaire industriel comporte quelques pixels seulement, mais les versions futures en prévoient beaucoup plus. Ainsi, en regardant les résultats de tous les capteurs, on obtient une image basse résolution avec une information de profondeur. En analysant les détections inter-capteurs, il serait possible de reconnaître des formes plus élaborées qu'en se limitant à corrélérer les paramètres des échos avec des types de cibles, tel que suggéré au paragraphe précédent. La combinaison des deux méthodes serait fort probablement l'idéal.

À long terme et sur un plan plus académique, une publication scientifique sera envisagée dès que les bénéfices pour l'entreprise partenaire seront protégés.

Annexe A

Annexe

A.1 Définition et calcul du SNR

Comme un écho reçu n'est pas un signal continu mais plutôt ponctuel, il peut être moins évident de déterminer son SNR. Pour tout ce mémoire de maîtrise, la définition du SNR d'un écho se réfère à la définition suivante, tirée de la page 44 du livre Introduction to RADAR systems de Skolnik[18].

$$SNR = \frac{A}{\psi_0^{1/2}} = \frac{\textit{amplitude du signal}}{\textit{valeur rms bruit}} \quad (\text{A.1})$$

La figure A.1 présente les données utilisées pour un exemple de calcul du SNR. Le premier graphique est la contribution de l'écho seule. Le deuxième graphique affiche la contribution du bruit uniquement. Le troisième graphique est le signal qui résulterait du retour de cet écho dans ce bruit.

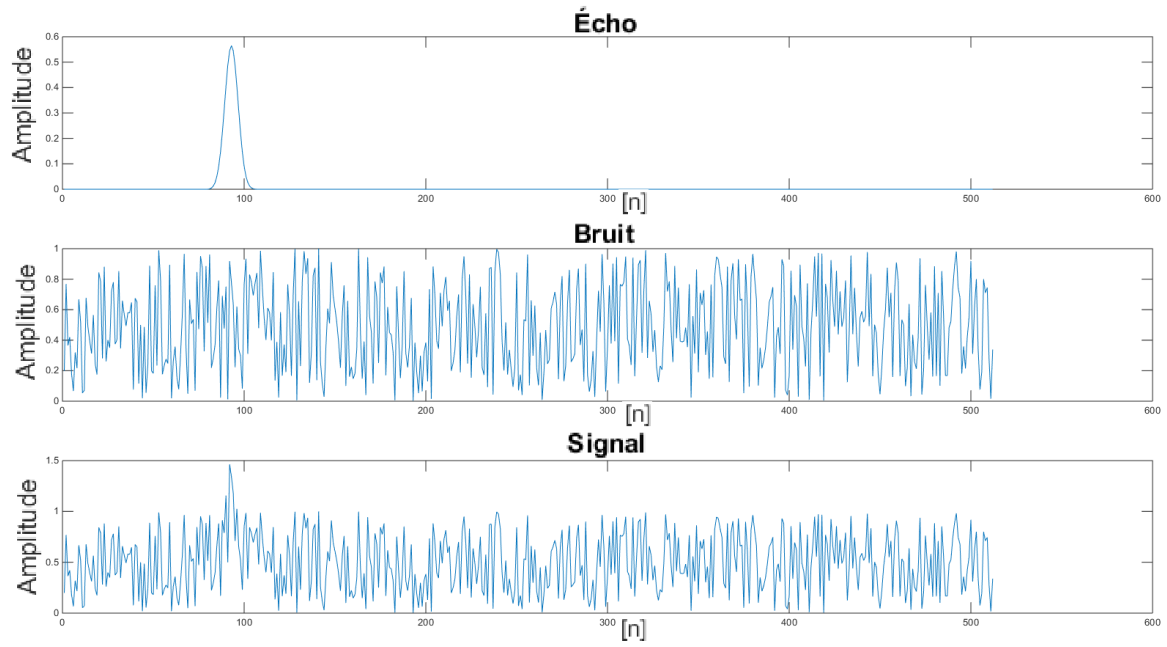


FIGURE A.1 – Données en exemple pour le calcul du SNR

Dans ce cas, l'amplitude de l'écho A est de 0.5654. De plus, la valeur RMS du bruit vaut elle aussi $\psi_0^{1/2} = 0.5654$. Ainsi, le SNR sera de $SNR = 0.5654/0.5654 = 1$, et converti en décibels, il vaut $SNR = 20\log(1) = 0 \text{ dB}$. Le dernier graphique de la figure illustre l'addition du bruit et de l'écho, formant ainsi le signal.

A.2 Adaptation

A.2.1 Résultats des tests

Précision des algorithmes : exactitude des mesures

Comparaison de l'exactitude de la distance détectée des algorithmes				
Algorithmes	MUSIC	Double Wavelet	RMS conv	Corrélation
Distance				
10 m	1.04 cm	1.70 cm	1.31 cm	1.01 cm
10.25 m	1.03 cm	1.82 cm	1.35 cm	1.41 cm
10.5 m	1.54 cm	3.91 cm	2.15 cm	1.28 cm
10.75 m	1.34 cm	2.65 cm	1.99 cm	1.09 cm
11 m	1.84 cm	2.53 cm	2.16 cm	1.31 cm
11.25 m	2.36 cm	2.73 cm	1.72 cm	1.77 cm
11.5 m	2.19 cm	2.19 cm	2.02 cm	1.63 cm
11.75 m	1.28 cm	1.88 cm	1.60 cm	1.49 cm
12 m	1.26 cm	2.03 cm	2.20 cm	1.34 cm
12.25 m	1.37 cm	2.11 cm	1.75 cm	1.84 cm
12.5 m	1.38 cm	1.87 cm	1.64 cm	1.60 cm
12.75 m	1.18 cm	2.49 cm	1.63 cm	1.12 cm
13 m	1.12 cm	2.12 cm	1.62 cm	1.30 cm
13.25	1.32 cm	1.92 cm	1.49 cm	1.62 cm
13.5 m	1.23 cm	2.11 cm	2.05 cm	1.25 cm
13.75 m	1.48 cm	2.61 cm	1.92 cm	1.37 cm
14 m	3.53 cm	4.82 cm	4.83 cm	3.82 cm
14.25 m	2.17 cm	4.79 cm	4.66 cm	1.82 cm
14.5 m	2.03 cm	5.14 cm	3.44 cm	1.88 cm
14.75 m	3.00 cm	4.30 cm	3.75 cm	3.86 cm
15 m	2.64 cm	5.15 cm	3.64 cm	2.64 cm
15.25 m	3.21 cm	5.06 cm	3.67 cm	3.64 cm
15.5 m	2.64 cm	5.46 cm	3.53 cm	4.14 cm
15.75 m	3.17 cm	8.63 cm	5.49 cm	2.11 cm
16 m	4.09 cm	8.11 cm	5.28 cm	2.70 cm
16.25 m	3.93 cm	5.67 cm	4.64 cm	2.77 cm
16.5 m	3.00 cm	4.63 cm	3.07 cm	2.53 cm
16.75 m	3.75 cm	3.55 cm	2.89 cm	3.35 cm

Algorithmes	MUSIC	Double Wavelet	RMS conv	Corrélation
Distance				
17 m	3.06 cm	5.01 cm	4.41 cm	4.50 cm
17.25 m	3.76 cm	7.01 cm	5.87 cm	3.83 cm
17.5 m	6.93 cm	11.76 cm	9.45 cm	6.28 cm
17.75 m	10.84 cm	15.24 cm	13.97 cm	8.69 cm
18 m	12.35 cm	17.40 cm	14.73 cm	9.95 cm
18.25 m	12.49 cm	17.00 cm	14.89 cm	11.63 cm
18.5 m	11.49 cm	13.73 cm	12.74 cm	10.19 cm
18.75 m	9.19 cm	10.28 cm	9.18 cm	8.28 cm
19 m	7.57 cm	9.34 cm	8.38 cm	9.32 cm
19.25 m	5.63 cm	8.44 cm	7.11 cm	6.32 cm
19.5 m	6.46 cm	9.41 cm	7.44 cm	8.08 cm
19.75 m	5.86 cm	10.78 cm	7.44 cm	8.68 cm
20 m	6.80 cm	11.10 cm	8.43 cm	9.24 cm
20.25 m	5.54 cm	10.03 cm	7.44 cm	7.29 cm
20.5 m	6.27 cm	12.19 cm	8.23 cm	6.80 cm
20.75 m	6.93 cm	10.21 cm	8.49 cm	6.45 cm
21 m	7.35 cm	9.66 cm	7.95 cm	9.82 cm
21.25 m	8.07 cm	12.09 cm	8.26 cm	9.16 cm
21.5 m	8.45 cm	11.83 cm	10.10 cm	8.56 cm
21.75 m	7.60 cm	13.69 cm	10.67 cm	9.29 cm
22 m	11.78 cm	19.83 cm	15.83 cm	13.83 cm
22.25 m	13.57 cm	19.85 cm	16.08 cm	14.24 cm
22.5 m	11.47 cm	15.08 cm	12.08 cm	12.56 cm
22.75 m	13.03 cm	12.80 cm	11.10 cm	14.76 cm
23 m	16.38 cm	15.90 cm	14.98 cm	16.97 cm
23.25 m	12.42 cm	16.02 cm	11.04 cm	12.85 cm
23.5 m	14.24 cm	16.38 cm	14.97 cm	15.01 cm
23.75 m	14.03 cm	15.18 cm	16.41 cm	14.37 cm
24 m	13.01 cm	24.42 cm	18.28 cm	13.62 cm
24.25 m	12.99 cm	22.77 cm	13.71 cm	12.82 cm
24.5 m	15.81 cm	28.34 cm	20.39 cm	13.85 cm
24.75 m	14.56 cm	36.66 cm	11.95 cm	12.69 cm
25 m	14.36 cm	32.00 cm	24.66 cm	13.30 cm

A.2.2 Précision des algorithmes : dispersion des mesures

Comparaison de la dispersion des distances des algorithmes				
Algorithmes	MUSIC	Double Wavelet	RMS conv	Corrélation
Distance				
10 m	1.33 cm	2.28 cm	1.74 cm	1.26 cm
10.25 m	1.31 cm	1.94 cm	1.57 cm	1.22 cm
10.5 m	1.66 cm	2.66 cm	2.13 cm	1.60 cm
10.75 m	1.39 cm	2.01 cm	1.56 cm	1.37 cm
11 m	1.77 cm	2.42 cm	2.03 cm	1.65 cm
11.25 m	1.41 cm	2.25 cm	1.80 cm	1.44 cm
11.5 m	1.53 cm	2.46 cm	1.89 cm	1.58 cm
11.75 m	1.66 cm	1.88 cm	1.80 cm	1.76 cm
12 m	1.59 cm	2.33 cm	2.03 cm	1.63 cm
12.25 m	1.53 cm	2.25 cm	1.84 cm	1.62 cm
12.5 m	1.56 cm	2.26 cm	1.93 cm	1.64 cm
12.75 m	1.37 cm	2.63 cm	2.00 cm	1.34 cm
13 m	1.50 cm	2.21 cm	1.78 cm	1.71 cm
13.25	1.52 cm	2.36 cm	1.93 cm	2.98 cm
13.5 m	1.56 cm	2.75 cm	2.24 cm	1.63 cm
13.75 m	1.57 cm	2.67 cm	2.03 cm	1.65 cm
14 m	2.01 cm	2.89 cm	2.66 cm	3.74 cm
14.25 m	2.38 cm	4.07 cm	3.47 cm	2.18 cm
14.5 m	2.39 cm	4.39 cm	3.30 cm	2.34 cm
14.75 m	3.06 cm	4.56 cm	3.73 cm	3.26 cm
15 m	3.26 cm	5.64 cm	4.40 cm	3.02 cm
15.25 m	3.60 cm	5.87 cm	4.54 cm	3.82 cm
15.5 m	3.16 cm	4.63 cm	3.75 cm	3.00 cm
15.75 m	2.76 cm	4.21 cm	3.72 cm	2.58 cm
16 m	2.88 cm	5.13 cm	3.55 cm	2.90 cm
16.25 m	3.18 cm	4.57 cm	3.79 cm	3.20 cm
16.5 m	2.33 cm	3.65 cm	3.14 cm	2.75 cm
16.75 m	2.94 cm	4.05 cm	2.82 cm	3.50 cm
17 m	3.53 cm	5.07 cm	4.49 cm	7.90 cm
17.25 m	4.03 cm	6.98 cm	5.21 cm	4.49 cm
17.5 m	4.24 cm	6.76 cm	5.52 cm	4.13 cm
17.75 m	4.02 cm	6.33 cm	5.09 cm	5.82 cm

Comparaison de la dispersion des distances des algorithmes				
Algorithmes	MUSIC	Double Wavelet	RMS conv	Corrélation
Distance				
18 m	4.17 cm	6.94 cm	5.24 cm	4.73 cm
18.25 m	5.04 cm	7.98 cm	6.63 cm	9.81 cm
18.5 m	5.16 cm	9.34 cm	7.15 cm	4.95 cm
18.75 m	5.70 cm	11.01 cm	7.47 cm	5.36 cm
19 m	5.32 cm	8.80 cm	6.83 cm	10.04 cm
19.25 m	6.08 cm	10.11 cm	8.13 cm	6.47 cm
19.5 m	7.17 cm	11.55 cm	9.10 cm	8.21 cm
19.75 m	7.31 cm	12.94 cm	9.68 cm	9.60 cm
20 m	8.06 cm	11.68 cm	9.68 cm	9.46 cm
20.25 m	6.82 cm	10.56 cm	8.64 cm	7.95 cm
20.5 m	7.76 cm	12.10 cm	8.35 cm	8.44 cm
20.75 m	8.17 cm	11.42 cm	9.19 cm	7.67 cm
21 m	8.68 cm	12.24 cm	9.03 cm	9.34 cm
21.25 m	8.60 cm	12.46 cm	8.21 cm	7.79 cm
21.5 m	8.78 cm	12.15 cm	10.18 cm	9.16 cm
21.75 m	7.66 cm	15.33 cm	10.93 cm	9.26 cm
22 m	9.92 cm	18.16 cm	12.83 cm	11.74 cm
22.25 m	8.87 cm	18.94 cm	13.40 cm	10.87 cm
22.5 m	9.33 cm	17.50 cm	13.32 cm	10.33 cm
22.75 m	9.47 cm	14.22 cm	9.84 cm	12.28 cm
23 m	10.28 cm	15.05 cm	11.31 cm	11.59 cm
23.25 m	10.97 cm	17.73 cm	11.41 cm	12.99 cm
23.5 m	12.62 cm	17.48 cm	13.51 cm	14.08 cm
23.75 m	14.94 cm	16.93 cm	16.56 cm	16.30 cm
24 m	14.43 cm	20.90 cm	18.71 cm	14.21 cm
24.25 m	17.00 cm	27.71 cm	17.68 cm	16.31 cm
24.5 m	17.30 cm	35.17 cm	26.03 cm	17.37 cm
24.75 m	17.78 cm	41.20 cm	15.86 cm	16.43 cm
25 m	16.58 cm	27.41 cm	16.53 cm	16.56 cm

Résultats de séparation d'échos avec les données du 9 janvier 2014

Les signaux utilisés sont les mêmes que pour le test présenté à la section précédente 4.1.2. Ces signaux sont enregistrés avec le système de notre partenaire industriel. La scène consistait en 2 cibles, disposées face au système à une distance facilement atteinte par le système, avec un écart variable entre elles. Pour chacun des écarts testés, entre 250 et 350 réponses furent enregistrées.

Les historiques de signaux étaient formés de 4 réponses pour ce test comparatif. Les 3 dernières réponses du dernier historique sont réutilisées dans l'historique de l'essai suivant, telle une fenêtre glissante. L'essai est réussi lorsque l'algorithme détecte séparément les deux cibles. À la fin de la série, le taux de succès est obtenu pour un écart particulier des 2 cibles.

Comparaison des algorithmes pour la séparation d'échos superposés				
Algorithmes	MUSIC	Double Wavelet	RMS conv	Corrélation
Écart				
5 m	100 %	100 %	100 %	100 %
4.75 m	100 %	100 %	100 %	96.64 %
4.5 m	100 %	100 %	100 %	98.29 %
4.25	100 %	100 %	100 %	57.32 %
4 m	100 %	39.38 %	36.56 %	4.38 %
3.75 m	100 %	40 %	4.76 %	1.90 %
3.5 m	100 %	53 %	7.77 %	1.06 %
3.25 m	100 %	4.66 %	0.36 %	0 %
3 m	100 %	0 %	0 %	0 %
2.75 m	100 %	0 %	0 %	0 %
2.5 m	100 %	0 %	0 %	0 %
2.25 m	100 %	0 %	0 %	0 %
2 m	100 %	0 %	0 %	0 %
1.75 m	100 %	0 %	0 %	0 %
1.5 m	100 %	0 %	0 %	0 %
1.25 m	99.69 %	0 %	0 %	0 %

Bibliographie

- [1] R. Adve : Direction of Arrival Estimation. Department of Electrical and Computer Engineering, University of Toronto, Canada, 2013.
- [2] A.M. Bruckstein, T.J. Shan et T. Kailath : A time-domain signal resolution problem. In International Conference on Acoustics, Speech and Signal Processing, pages 1665–1668, Stanford University, 1986.
- [3] P. Chen, T.-J. Wu et J. Yang : A comparative study of model selection criteria for the number of signals. IET Radar, Sonar and Navigation, pages 180–188, 2007.
- [4] D. Grenier et É. Bossé : Decorrelation performance of deese and spatial smoothing techniques for direction-of-arrival problems. IEEE Transactions on Signal Processing, 44(6), 1996.
- [5] D. Grenier, É. Pigeon et R. M. Turner : High-range resolution mono-frequency pulsed radar for the identification of approaching targets using subsampling and the music algorithm. IEEE Signal Processing Letters, 3(6):179–181, june 1996.
- [6] G. H. Golub et C. F. Van Loan : Matrix Computations. The Johns Hopkins University Press, 4th édition, 2013.
- [7] M. K. Ibrahim : New high-resolution pseudospectrum estimation method. IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-35(7), 1987.
- [8] M.L. James, G.M. Smith et J.C. Wolford : Applied Numerical Methods for Digital Computation. Harper and Row, New York, 3rd édition, 1985.
- [9] A. P. Liavas et P. A. Regalia : On the behavior of information theoretic criteria for model order selection. IEEE transactions on signal processing, 49(8):1689–1695, août 2001.
- [10] T. N. Rao et V. S. Rao : Implementation of music algorithm for a smart antenna system for mobile communications. International Journal of Scientific and Engineering Research, 2(12), 2011.
- [11] M.A. Pallas et G. Jourdain : Estimation de retard et méthodes à haute résolution. 1987.

- [12] M. Panju : Iterative Methods for Computing Eigenvalues and Eigenvectors. University of Waterloo, 2011. The Waterloo Mathematics Review.
- [13] V.F. Pisarenko : The retrieval of harmonics from a covariance function. Geophysical Journal, 33(3):347 – 366, 1973.
- [14] J. Kumar et B. Mazhari : Spectral estimation using a music algorithm, 2005. Indian Institute of Technology, Nios II Embedded Processor Design Contest.
- [15] B.D. Rao et K.V.S. Hari : Effect of spatial smoothing on the performance of music and the minimum-norm method. IEEE Proceedings, 137(6):449–458, december 1990.
- [16] R. O. Schmidt : Multiple emitter location and signal parameter estimation. IEEE Transactions on Antennas and Propagation, AP-34(3):276–280, March 1986.
- [17] S. Shahbazpanahi, A. B. Gershman et G. B. Giannakis : Semiblind multiuser mimo channel estimation using capon and music techniques. IEEE Transactions on Signal Processing, 54(9), 2006.
- [18] M. I. Skolnik : Introduction to RADAR Systems. McGRAW-HILL International Editions, 3rd édition, 2001.