Application de Méthodes de Clustering sur le Détecteur GLRT

Reihan Mazouz

Résumé :

Ce rapport explore différentes méthodes de clustering appliquées à un détecteur GLRT (Generalized Likelihood Ratio Test). Ces approches permettent de caractériser des signaux en regroupant des points détectés par le GLRT.

Novembre 2024

Table des matières

1 Introduction

Le *Generalized Likelihood Ratio Test* (GLRT) est une méthode statistique utilisée en traitement du signal pour détecter des signaux d'intérêt dans des environnements bruités. Appliqué sur des spectrogrammes temps-fréquence, il permet d'identifier des cellules de spectre où la présence d'un signal est statistiquement significative. Cependant, le GLRT ne permet pas de regrouper ou d'analyser ces points de manière globale, ce qui complique l'interprétation des résultats de détection.

Problématique : Comment regrouper les points détectés par le GLRT pour identifier des structures de signal dans les spectrogrammes ?

L'objectif de cette étude est de comparer différentes méthodes de clustering appliquées aux sorties du GLRT. Ces méthodes visent à regrouper les points détectés en clusters cohérents sensées représenter les signaux détectés.

Les contraintes majeures à considérer sont :
- **Robustesse face au bruit :** Les clusters doivent refléter des signaux réels et non des artefacts générés par le bruit.
- **Détection de structures complexes :** Les algorithmes doivent être capables de reconnaître des motifs non linéaires, des courbes ou des regroupements non convexes.
- **Gestion des signaux entrelacés :** Les méthodes doivent pouvoir distinguer et séparer des signaux qui se superposent en temps et en fréquence.
- **Efficacité algorithmique :** Les méthodes doivent être adaptées pour traiter des spectrogrammes de grande taille, comportant un grand nombre de points détectés.

Ce rapport évalue une série d'algorithmes de clustering appliqués aux sorties du GLRT. Chaque méthode est testée sur des spectrogrammes générés artificiellement et comparée selon des critères de précision, de robustesse, de scalabilité et de capacité à extraire des structures pertinentes.

2 Modélisation Mathématique et Méthodes d'Évaluation

Formulation Analytique du Problème

Considérons un spectrogramme temps-fréquence, noté $\mathbf{Z} \in \mathbb{C}^{M \times N}$, où M représente le nombre de cellules fréquentielles et N le nombre de cellules temporelles. Chaque cellule $\mathbf{Z}(i,j)$ contient une valeur complexe correspondant à l'énergie spectrale en fréquence f_i et temps t_j.

Le détecteur *Generalized Likelihood Ratio Test* (GLRT) est appliqué à chaque cellule pour calculer une statistique de détection donnée par :

$$|Z(i,j)| = \sqrt{\mathrm{Re}(\mathbf{Z}(i,j))^2 + \mathrm{Im}(\mathbf{Z}(i,j))^2},$$

où $\mathrm{Re}(\mathbf{Z}(i,j))$ et $\mathrm{Im}(\mathbf{Z}(i,j))$ sont respectivement les parties réelle et imaginaire de $\mathbf{Z}(i,j)$.

Un seuil de détection γ est défini pour séparer les cellules contenant un signal d'intérêt des cellules dominées par le bruit, ce seuil sera réglé en fonction de la probabilité de fausse alarme souhaitée. La condition de détection est alors donnée par :

$$|Z(i,j)| > \gamma.$$

Le GLRT fournit donc un ensemble de points détectés, noté $\mathcal{D} = \{(i,j) \mid \lambda(i,j) > \gamma\}$, où chaque point représente une cellule potentiellement associée à un signal.

Problème de Clustering

Le problème consiste à regrouper les points $(i, j) \in \mathcal{D}$, en clusters cohérents reflétant les structures globales des signaux de l'image temps fréquence. Matériellement, cela revient à rechercher un ensemble de clusters $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$, où chaque cluster $\mathcal{C}_k$ est une sous-partie de l'ensemble des points détectés $\mathcal{D}$, avec les propriétés suivantes :

$$\bigcup_{k=1}^{K} \mathcal{C}_k \subseteq \mathcal{D}.$$

Contrairement à une partition stricte, il est important de noter deux spécificités dans ce problème :

- **Filtrage des points de bruit :** Certains points détectés par le GLRT ne sont pas associés à un cluster valide, notamment ceux résultant du bruit. Ces points sont exclus de l'ensemble des clusters $\bigcup_{k=1}^{K} \mathcal{C}_k$.
- **Clusters avec intersection potentielle :** Lorsqu'il existe des signaux s'entremêlant dans le spectrogramme, certains points peuvent être associés à plusieurs clusters, créant ainsi des intersections non vides entre les clusters :

$$\mathcal{C}_i \cap \mathcal{C}_j \neq \emptyset, \quad \text{pour certains } i \neq j.$$

Évaluation des Méthodes de Clustering

Pour évaluer la qualité des algorithmes de clustering appliqués aux sorties du GLRT, les métriques suivantes sont définies :

- **Taux de détection (*Detection Rate*) :** Un signal est considéré comme détecté si l'intersection entre le signal réel et au moins un cluster généré est non nulle. Soit $\mathcal{S}_i$ l'ensemble des points correspondant au i-ème signal réel et $\mathcal{C}$ l'ensemble des clusters détectés. Le taux de détection est donné par :

$$\text{Taux de détection} = \frac{|\{\mathcal{S}_i \mid \exists \mathcal{C}_k \in \mathcal{C}, \mathcal{S}_i \cap \mathcal{C}_k \neq \emptyset\}|}{|\{\mathcal{S}_i\}|}.$$

- **Taux de fausse alarme (*False Alarm Rate*) :** Cette métrique mesure la proportion de clusters générés par l'algorithme qui ne sont pas réellement associés à un signal. Soit $\mathcal{C}$ l'ensemble des clusters détectés et $\mathcal{C}_{\text{valide}}$ l'ensemble des clusters correspondant à des signaux réels. Le taux de fausse alarme est donné par :

$$\text{Taux de fausse alarme} = \frac{|\mathcal{C} \setminus \mathcal{C}_{\text{valide}}|}{|\mathcal{C}|}.$$

- **Erreur de caractérisation :** Les paramètres t_0 (temps d'arrivée), f_c (fréquence porteuse), Δt (durée) et Δf (largeur de bande) d'un signal sont estimés à partir des indices des points appartenant à un cluster. Pour un cluster donné $\mathcal{C}_k$, les paramètres estimés sont calculés comme suit :

$$t_0^{\text{estimé}} = \min_{(i,j)\in\mathcal{C}_k} t_j, \quad f_c^{\text{estimé}} = \frac{\sum_{(i,j)\in\mathcal{C}_k} f_i}{|\mathcal{C}_k|},$$

$$\Delta t^{\text{estimé}} = \max_{(i,j)\in\mathcal{C}_k} t_j - \min_{(i,j)\in\mathcal{C}_k}, \quad \Delta f^{\text{estimé}} = \max_{(i,j)\in\mathcal{C}_k} f_i - \min_{(i,j)\in\mathcal{C}_k}.$$

Les erreurs de caractérisation sont définies par rapport aux valeurs réelles ($t_0^{\text{réel}}$, $f_c^{\text{réel}}$, $\Delta t^{\text{réel}}$, $\Delta f^{\text{réel}}$) comme suit :

$$\text{Erreur en } t_0 = \frac{|t_0^{\text{estimé}} - t_0^{\text{réel}}|}{t_0^{\text{réel}}}, \quad \text{Erreur en } f_c = \frac{|f_c^{\text{estimé}} - f_c^{\text{réel}}|}{f_c^{\text{réel}}}.$$

$$\text{Erreur en } \Delta t = \frac{|\Delta t^{\text{estimé}} - \Delta t^{\text{réel}}|}{\Delta t^{\text{réel}}}, \quad \text{Erreur en } \Delta f = \frac{|\Delta f^{\text{estimé}} - \Delta f^{\text{réel}}|}{\Delta f^{\text{réel}}}.$$

— **Complexité calculatoire :** La complexité algorithmique est évaluée empiriquement sur des données de test en mesurant les temps de calcul. Bien qu'il soit possible de formuler analytiquement la complexité théorique de chaque méthode, les résultats empiriques sont privilégiés pour simplifier cette étude. Elle ne sera parfois pas pertinente car certaines méthodes sont optimisées et certaines ne le sont pas.

— **Complexité de réglage du modèle :** La facilité d'utilisation des algorithmes est évaluée qualitativement en fonction du nombre de paramètres requis et de leur sensibilité aux données. Les algorithmes nécessitant un ajustement précis ou un grand nombre de réglages seront considérés comme plus complexes.

3 Protocole de test

3.1 Test sur un ensemble de données défini par son SNR

L'ensemble de données utilisé pour les tests par SNR est constitué de spectrogrammes de taille 512*512 contenant trois types de signaux distincts :

— **Signal LFM (Linear Frequency Modulation) :**

— **Signal NLFM (Non-Linear Frequency Modulation) :**

— **Signal PSK (Phase Shift Keying) :**

Les signaux sont générés de manière à être **spatialement séparés** dans le spectrogramme, garantissant l'absence d'entrelacements.

Les performances sont mesurées à l'aide des fonctions d'évaluation définies précédemment :

— Le **taux de détection**

— Le **taux de fausse alarme**

— Les **erreurs de caractérisation**

— Le **temps moyen de calcul**

Chaque algorithme est testé sur un ensemble de données simulés contenant 100 spectrogrammes générés avec des signaux de paramètres aléatoires à chaque niveau de SNR :

— **Haut SNR** : 20 dB.

— **Moyen SNR** : 12 dB.

— **Faible SNR** : 10 dB.

— **Très faible SNR** : 8 dB.

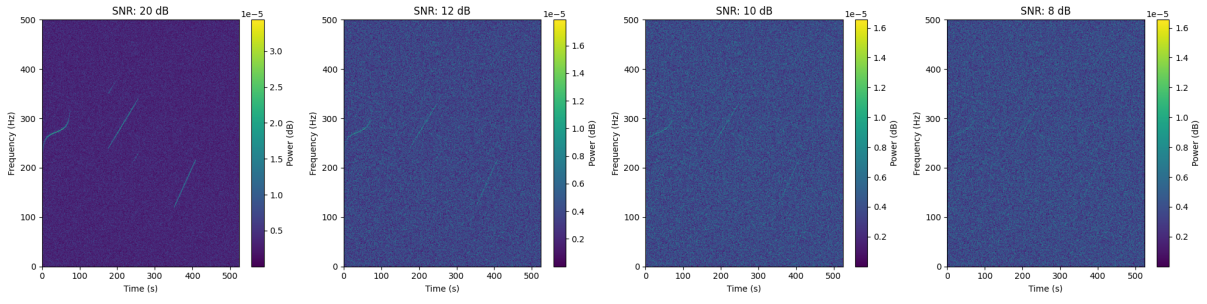


FIGURE 1 – Exemples de spectrogrammes générés pour différents niveaux de SNR : (a) 20 dB, (b) 12 dB, (c) 10 dB, (d) 8 dB. Les signaux LFM, P4, et LFM sont séparés dans le plan TF.

3.2 Évaluation pour les signaux entrelacés

Pour évaluer la capacité des algorithmes à détecter et regrouper des signaux entrelacés, une analyse visuelle est effectuée sur un spectrogramme en fort SNR contenant des signaux LFM et NLFM qui s'entremêlent dans le plan temps fréquence. Ces cas sont représentatifs de scénarios complexes où les signaux occupent des régions partiellement superposées dans le spectrogramme.

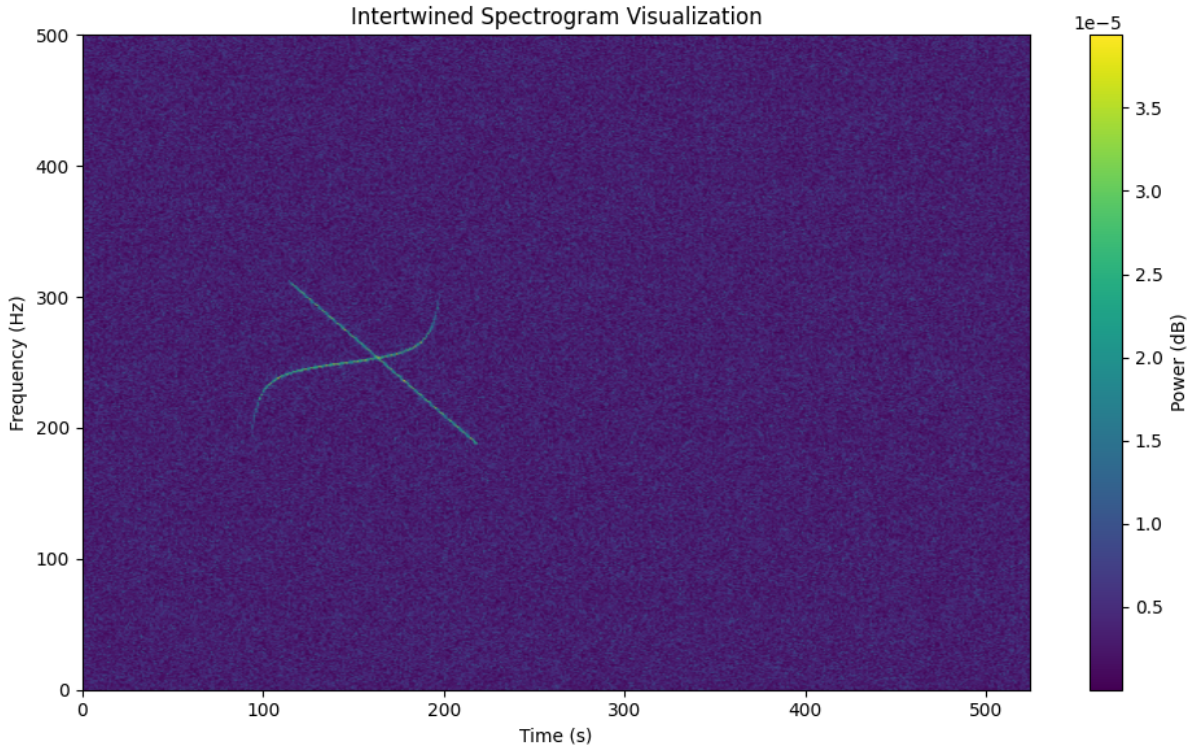


FIGURE 2 – Exemple de spectrogramme contenant des signaux entrelacés (LFM et NLFM) dans le plan temps-fréquence.

3.3 Protocole

Les différentes méthodes évaluées dans cette étude dépendent de plusieurs paramètres spécifiques à chaque algorithme. Ces paramètres, sans formules analytiques explicites, peuvent être complexes à sélectionner, car leur valeur optimale est souvent influencée par des facteurs externes lié à la donnée (comme le SNR).
Pour garantir une évaluation équitable, une phase préalable d'ajustement est réalisée afin de fixer les paramètres de chaque méthode en utilisant un dataset de donnée de bruit pure. L'objectif étant de maintenir le même taux de fausse alarme pour chaque algorithme.

En général, l'un des paramètres à déterminer est le seuil γ du test GLRT. Ce seuil impacte directement le nombre de points détectés dans le spectrogramme, influençant ainsi la performance des méthodes de clustering. Les figures ci-dessous illustrent les résultats du GLRT pour différents niveaux de SNR et seuils γ, mettant en évidence l'importance de ce choix.

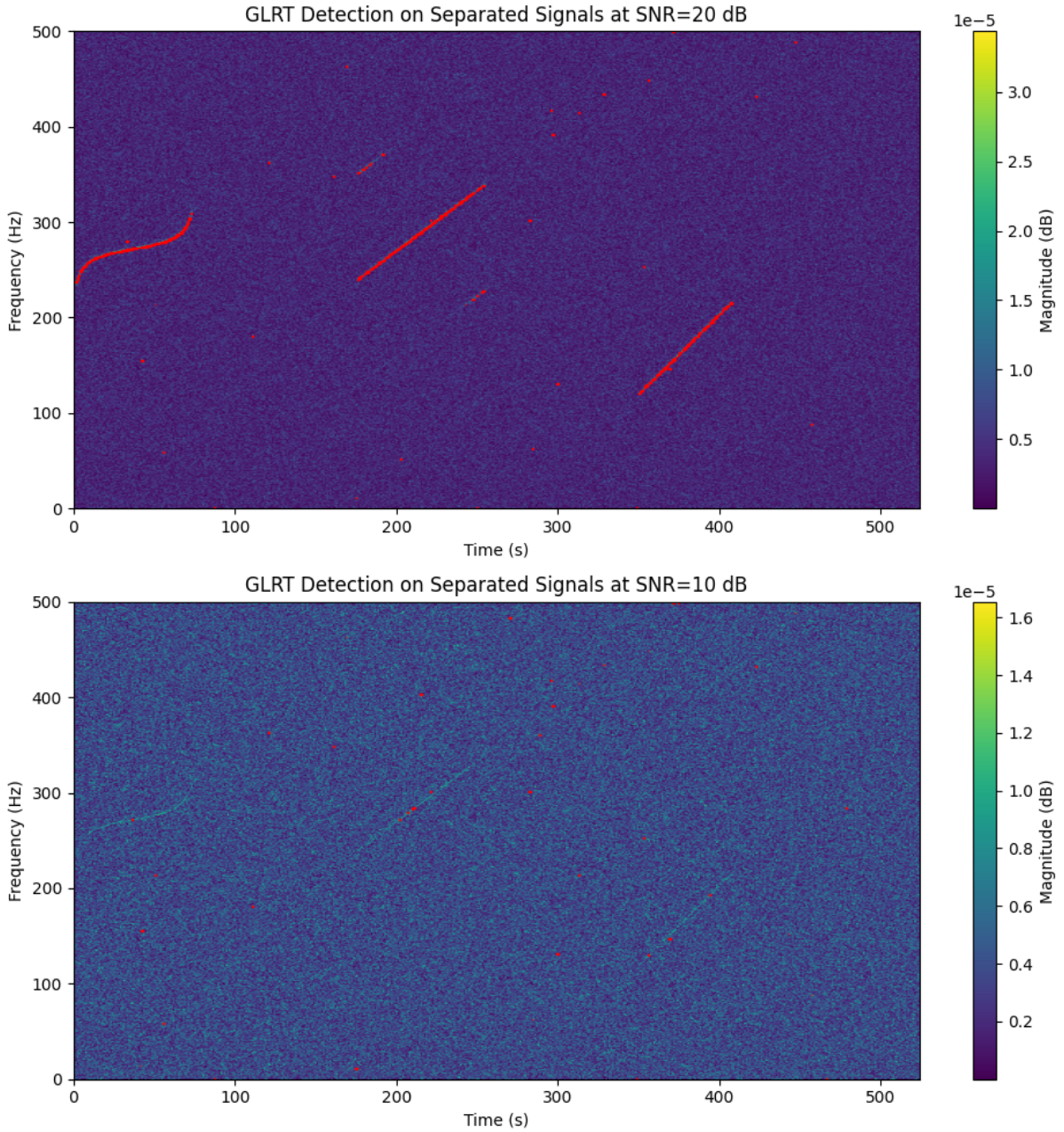


FIGURE 3 – Résultats du GLRT pour des SNRs de 20 dB et de 10 dB avec un seuil de détection fixé à $pfa = 1e - 4$. Les points détectés par le GLRT sont indiqués en rouge.

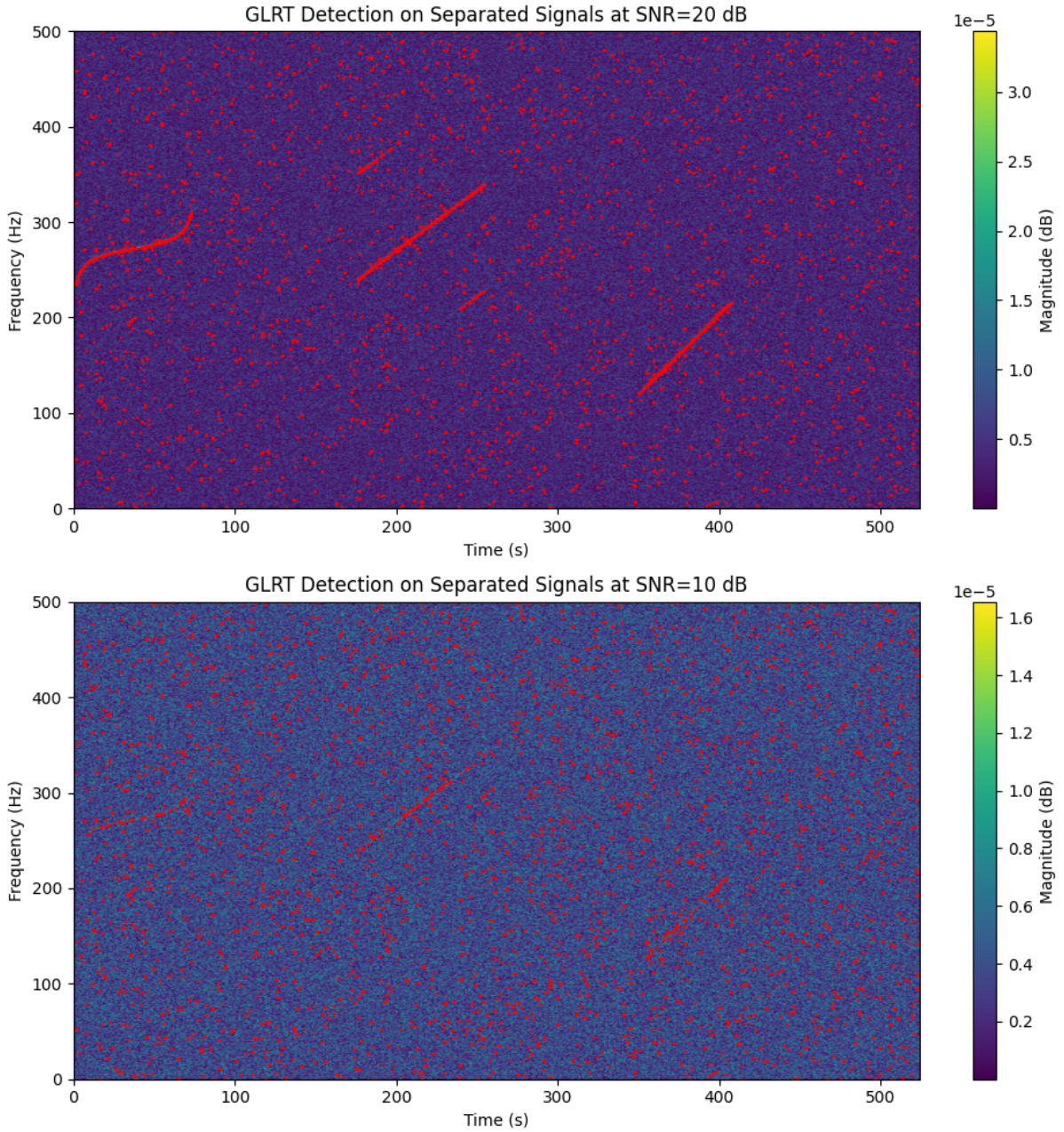


FIGURE 4 – Résultats du GLRT pour des SNRs de 20 dB et de 10 dB avec un seuil de détection fixé à $pfa = 1e - 2$. Les points détectés par le GLRT sont indiqués en rouge.

4 Méthodes de Clustering

Dans ce qui suit, des méthodes de clustering sont présentés. On y détaille pour la plupart leur principe de fonctionnement, les principaux paramètres de chaque méthode et une illustration des résultats obtenus sur une donnée test pour chaque SNR considéré (20 dB, 12 dB, 10 dB, 8 dB).

4.1 Grid-GLRT

Le **Grid-GLRT** est une méthode de détection basée sur un GLRT appliqué à un spectrogramme temps-fréquence quadrillé. Cette méthode consiste à diviser le spectre temps-fréquence $Z \in \mathbb{C}^{M \times N}$ en des grilles de cellules de tailles fixées, et à appliquer le test GLRT sur chaque grille pour détecter les régions contenant un signal significatif.

Principe de fonctionnement

1. **Quadrillage du spectrogramme :** L'image temps-fréquence Z est divisée en grilles de taille définie par les paramètres : pas de fréquences et pas de temps. Ces paramètres contrôlent la résolution spatiale de l'analyse.
2. **Test GLRT sur chaque grille :** Pour chaque grille, on réalise le GLRT en calculant une statistique de détection basée sur l'énergie moyenne des coefficients Z_{ij} présents dans la grille. La statistique est comparée à un seuil γ, fixé en fonction du taux de fausse alarme (PFA). Si l'énergie dépasse γ, la grille est considérée comme contenant un signal.
3. **Fusion des grilles proches :** Les grilles adjacentes ou suffisamment proches spatialement sont fusionnées pour former des clusters. La distance utilisée pour déterminer la proximité entre deux grilles est un paramètre.
4. **Filtrage :** Seuls sont conservés les clusters de dimension suffisantes, ce seuil sur la dimension étant un autre paramètre.

Analyse des hyperparamètres

Les performances de la méthode Grid-GLRT dépendent des paramètres suivants :
— **les grilles sélectionnées :** Une grille plus fine offre une meilleure résolution spatiale, détectant plus précisément les régions contenant du signal. Cependant, cela augmente le nombre de grilles à analyser, ce qui peut réduire l'efficacité algorithmique.
— **Seuil de détection (γ) :** Des seuils trop élevés conduisent à une sous-détection (faibles taux de détection), tandis que des seuils trop bas augmentent les fausses alarmes.
— **Distance de fusion :** Une distance de fusion trop petite peut fragmenter les clusters en plusieurs petites entités, tandis qu'une distance trop grande peut fusionner des régions contenant des signaux distincts.
— **Seuil sur la dimension :** Un seuil trop petit élimine les signaux de petites tailles

Illustrations

La figure 5 montre des exemples de l'application de la méthode Grid-GLRT sur des spectrogrammes à différents niveaux de SNR (20 dB, 12 dB) et avec différentes tailles de grille (4×4, 4×1, 1×4). La PFA par grille a été fixée à $1e^{-4}$.

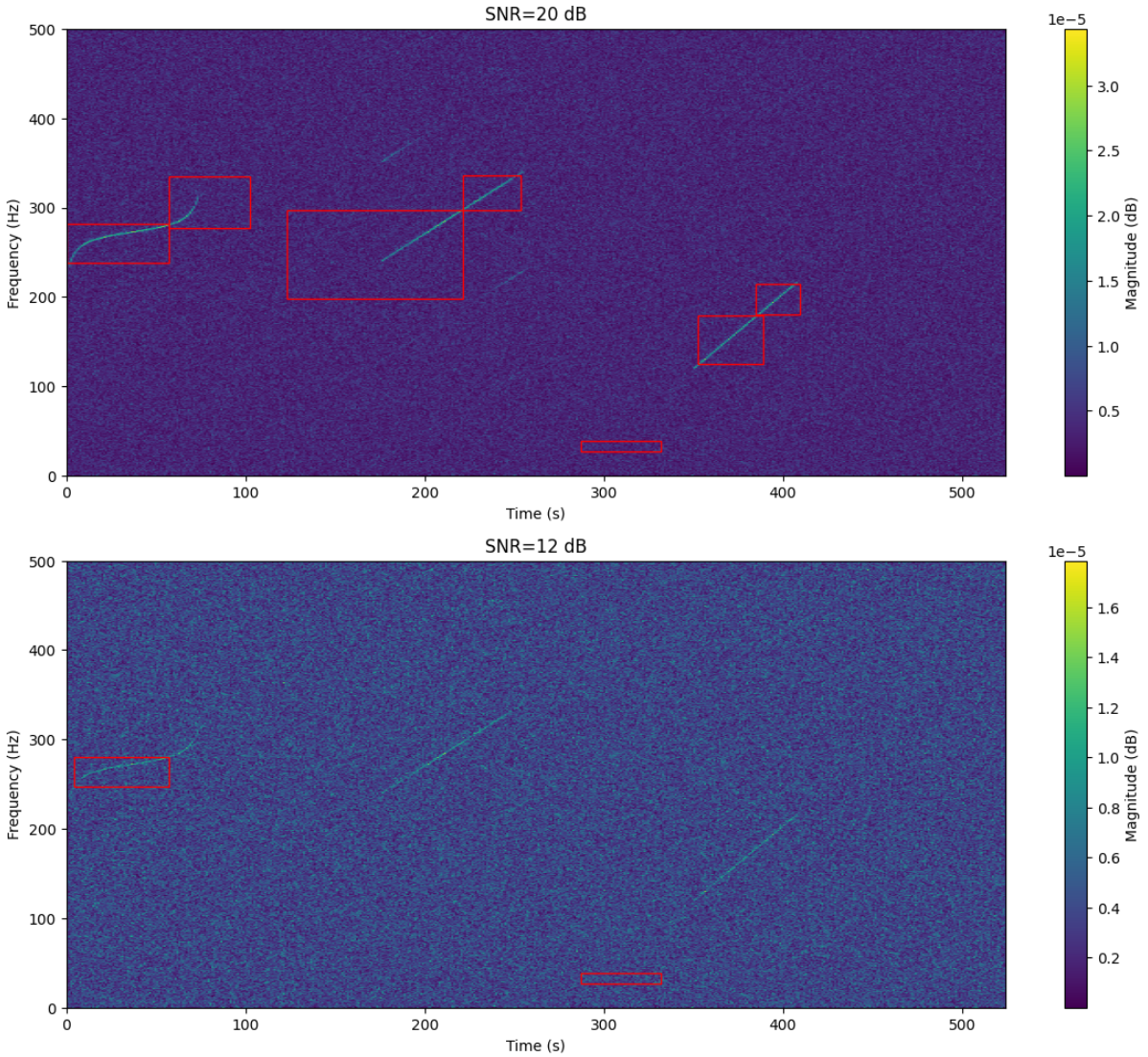


FIGURE 5 – Résultats de la méthode Grid-GLRT pour différents niveaux de SNR. Les rectangles rouges indiquent les clusters détectés.

4.2 Sliding-Box-GLRT

La méthode **Sliding-Box-GLRT** est une alternative au Grid-GLRT. Elle utilise une fenêtre glissante pour explorer le spectrogramme temps-fréquence $Z \in \mathbb{C}^{M \times N}$. Cette méthode applique le *Generalized Likelihood Ratio Test* (GLRT) sur chaque fenêtre pour détecter les régions contenant des signaux significatifs.

Principe de fonctionnement

1. **Exploration par fenêtre glissante :** Des fenêtres de taille fixe sont déplacées sur l'image temps-fréquence avec des pas spécifiques.
2. **Test GLRT pour chaque fenêtre :** La statistique GLRT est calculée en utilisant l'énergie moyenne des coefficients Z_{ij} dans la fenêtre. Un seuil γ, ajusté en fonction de la taille de la fenêtre, est utilisé pour décider si la fenêtre contient un signal significatif.
3. **Fusion des fenêtres adjacentes :** Les fenêtres détectées comme contenant un signal sont fusionnées si elles sont spatialement proches ou s'overlap. Une distance de fusion est utilisée comme critère de proximité.
4. **Filtrage par taille minimale :** Les boîtes de dimensions inférieures à un seuil spécifié sont éliminées.

Analyse des hyperparamètres

La méthode Sliding-Box-GLRT repose sur les mêmes hyperparamètres que Grid-GLRT. L'hyperparamètre sur le pas en temps et en fréquence de la fenêtre glissante, qui n'est pas présent dans la méthode Grid-GLRT, permet de jouer sur la résolution de l'algorithme. La complexité calculatoire est fortement impacté par ce paramètre.

Illustrations

La figure 6 présente les résultats obtenus avec Sliding-Box-GLRT sur des spectrogrammes pour différents niveaux de SNR et tailles de fenêtre.

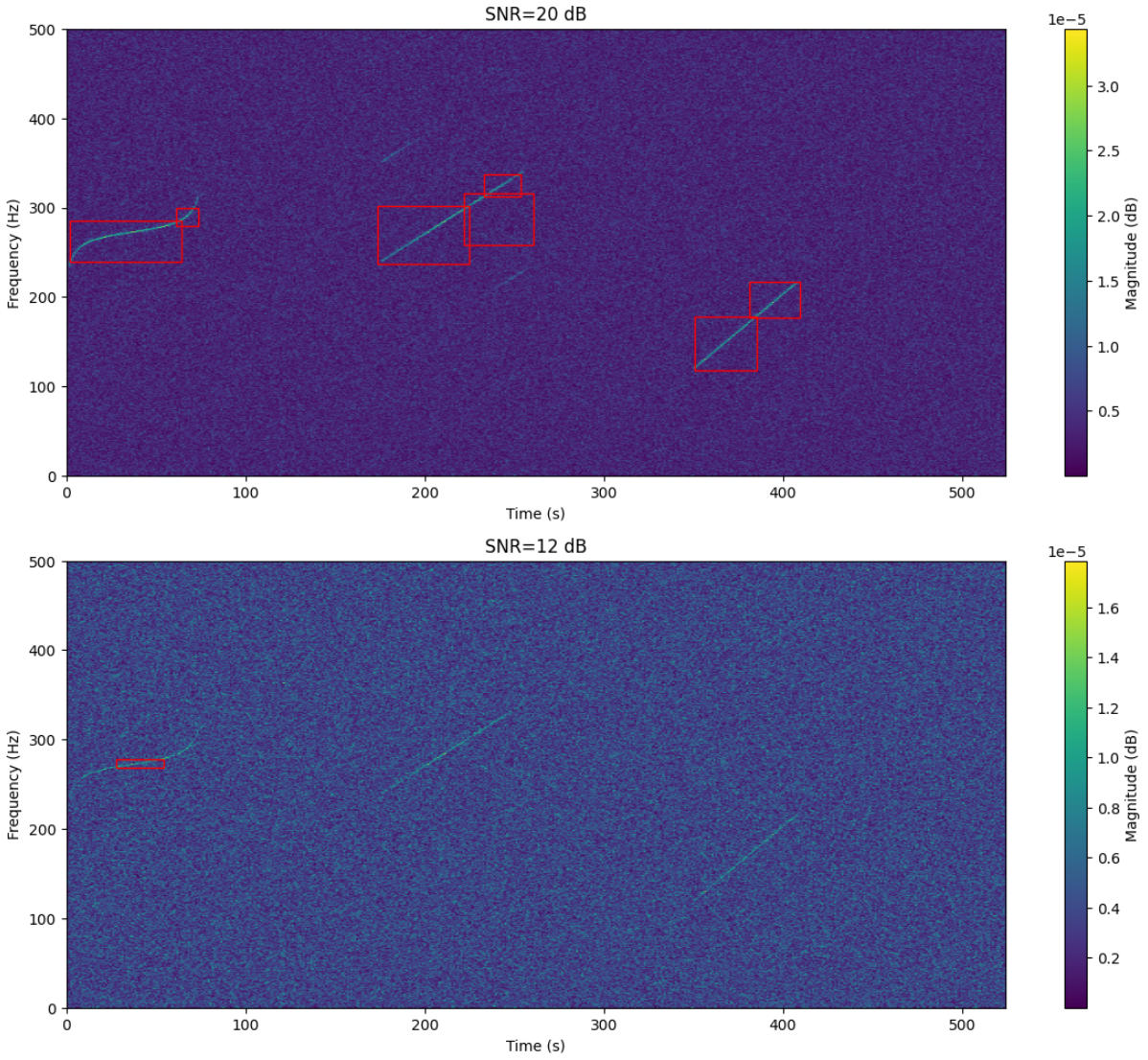


FIGURE 6 – Résultats de la méthode Sliding-Box-GLRT pour différents niveaux de SNR. Les rectangles rouges indiquent les fenêtres détectées. Les fenêtres utilisées sont $(4 \times 4, 4 \times 1, 1 \times 4)$. La PFA par grille a été fixée à $1e^{-6}$.

4.3 Extension-GLRT

La méthode **Extension-GLRT** repose sur une approche itérative pour détecter et étendre des chemins significatifs dans un spectrogramme temps-fréquence $Z \in \mathbb{C}^{M \times N}$. Cette méthode construit dynamiquement des clusters en partant d'un point initial, en ajoutant les voisins présentant la plus grande vraisemblance, tout en respectant un critère GLRT ajusté.

Principe de fonctionnement

1. **Initialisation :**
   — Chaque point du spectre Z est évalué pour déterminer s'il dépasse un seuil de détection initial γ.
   — Si la statistique de vraisemblance pour un point dépasse γ, un cluster est créé avec ce point comme origine.
2. **Recherche du meilleur voisin :**
   — À chaque étape, le voisin avec la plus grande valeur de vraisemblance est identifié parmi les voisins non encore visités.
   — Les voisins incluent les cellules adjacentes et diagonales.
3. **Extension du chemin :**
   — Le voisin identifié est ajouté au chemin si la statistique GLRT calculée pour le chemin étendu dépasse un seuil ajusté.
   — Ce processus itératif continue jusqu'à ce qu'aucun voisin valide ne puisse être ajouté.
4. **Finalisation du cluster :**
   — Lorsque le chemin ne peut plus être étendu, le cluster résultant est ajouté à la liste des clusters détectés.
   — Tous les points du cluster sont marqués comme visités pour éviter les doublons.
5. **Filtrage des clusters :**
   — Les clusters contenant moins d'un certain nombre de points sont rejetés pour éliminer les détections parasites.
   — Ce seuil garantit que seuls les clusters significatifs sont conservés, c'est un des paramètres de la méthode.

Analyse des hyperparamètres

L'efficacité de la méthode Extension-GLRT dépend des hyperparamètres suivants :
— **Probabilité de fausse alarme du test GLRT**
— **Taille minimale des clusters (min_points) :** Ce paramètre garantit que seuls les clusters contenant un nombre significatif de points sont conservés. Les petits clusters, souvent associés au bruit, sont ainsi éliminés.

Il n'y a donc que très peu d'hyperparamètres ce qui est avantageux, cependant la complexité calculatoire de la méthode est élevée.

Illustrations

La figure 7 montre les résultats de l'application de l'Extension-GLRT sur des spectrogrammes pour différents niveaux de SNR. Les clusters significatifs, correspondant à des chemins détectés, sont indiqués par des couleurs différentes.

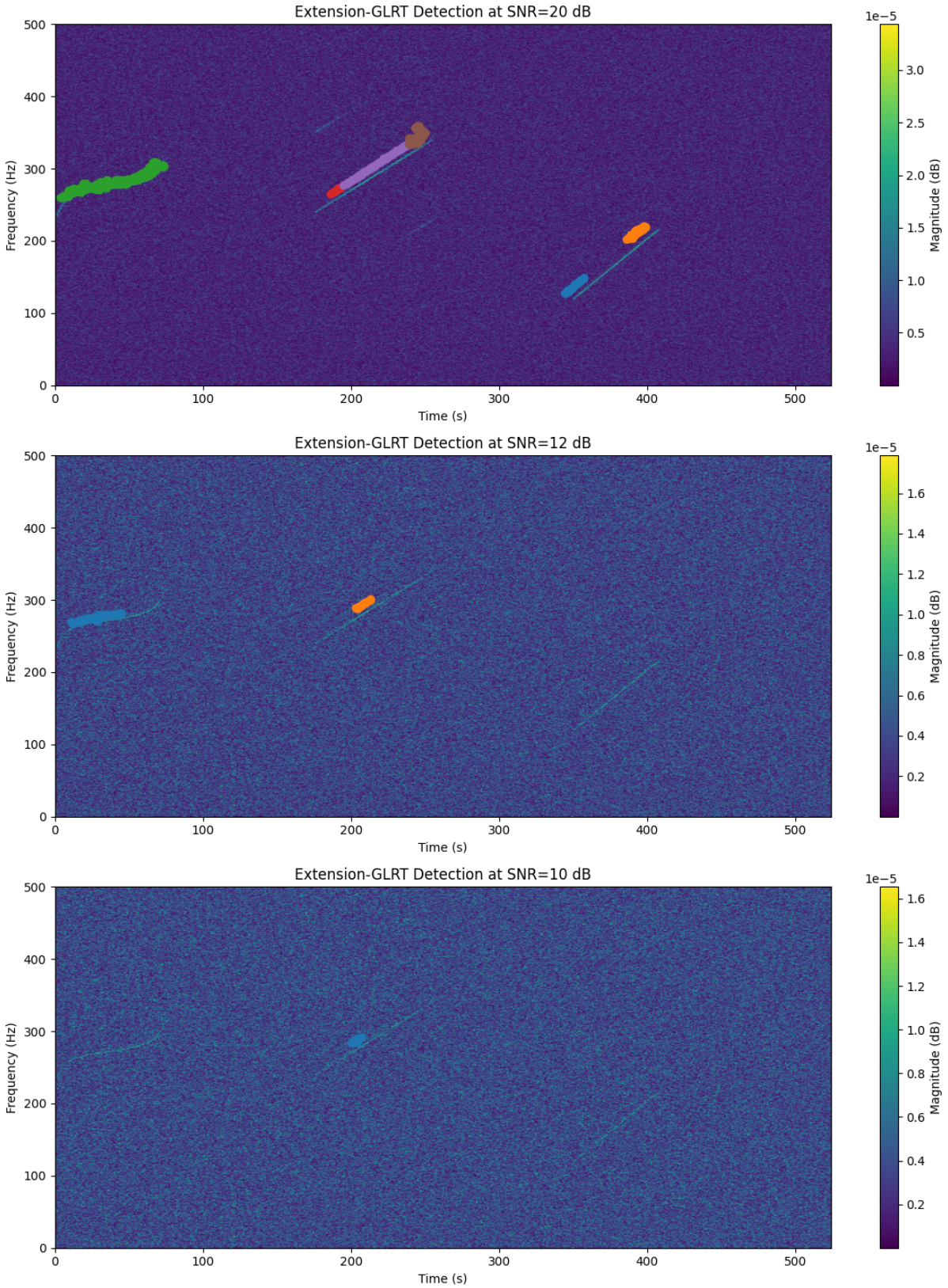


FIGURE 7 – Résultats de la méthode Extension-GLRT sur des spectrogrammes à différents niveaux de SNR. Les clusters détectés sont représentés par des couleurs différentes. On a utilisé les hyperparamètres suivants : $PFA = 1e^{-}6, minpoints = 10$ pour le SNR 20 dB et $PFA = 1e^{-}3, minpoints = 15$ pour les SNRs 12 et 10 dB

4.4 Paths-GLRT

La méthode **Paths-GLRT** se base sur la même idée que GLRT-extension qui est d'utiliser la continuité du signal dans le plan temps fréquence. L'objectif étant d'être moins calculatoire que la méthode précédente, elle va directement considéré les chemins entre deux points détéctés et les évaluer par le GLRT.

Principe de fonctionnement

Le **Paths-GLRT** se décompose en plusieurs étapes principales :

1. **Initialisation :**
   — Chaque point (i, j) du spectrogramme est évalué par le GLRT
2. **Construction de chemins :**
   — À partir d'un point candidat, la méthode cherche à connecter ce point à son voisin le plus proche parmi les autres candidats, en minimisant la distance euclidienne.
   — Les segments entre deux points candidats sont validés en utilisant un critère GLRT sur N points.
3. **Validation des segments :**
   — Un segment est ajouté au chemin si sa statistique de vraissemblance dépasse le seuil du GLRT.
   — Si un segment échoue au critère, le chemin courant est finalisé, et la méthode tente de démarrer un nouveau chemin.
4. **Finalisation et filtrage :**
   — Les chemins détectés sont ajoutés à la liste finale des clusters si leur taille dépasse un seuil.

Analyse des hyperparamètres

Les hyperparamètres de cette méthode sont identiques à ceux de la méthode Extension-GLRT. Cependant, en observant les résultats obtenus, on aimerait ajouter un hyperparamètre qui relierai les cluster juxtaposés afin d'améliorer la caractérisation.

Illustrations

La figure 8 illustre les résultats de l'application de la méthode **Paths-GLRT** sur des spectrogrammes pour différents niveaux de SNR (20 dB, 12 dB). Les chemins détectés, correspondant à des clusters significatifs, sont représentés par des points de couleurs différentes.

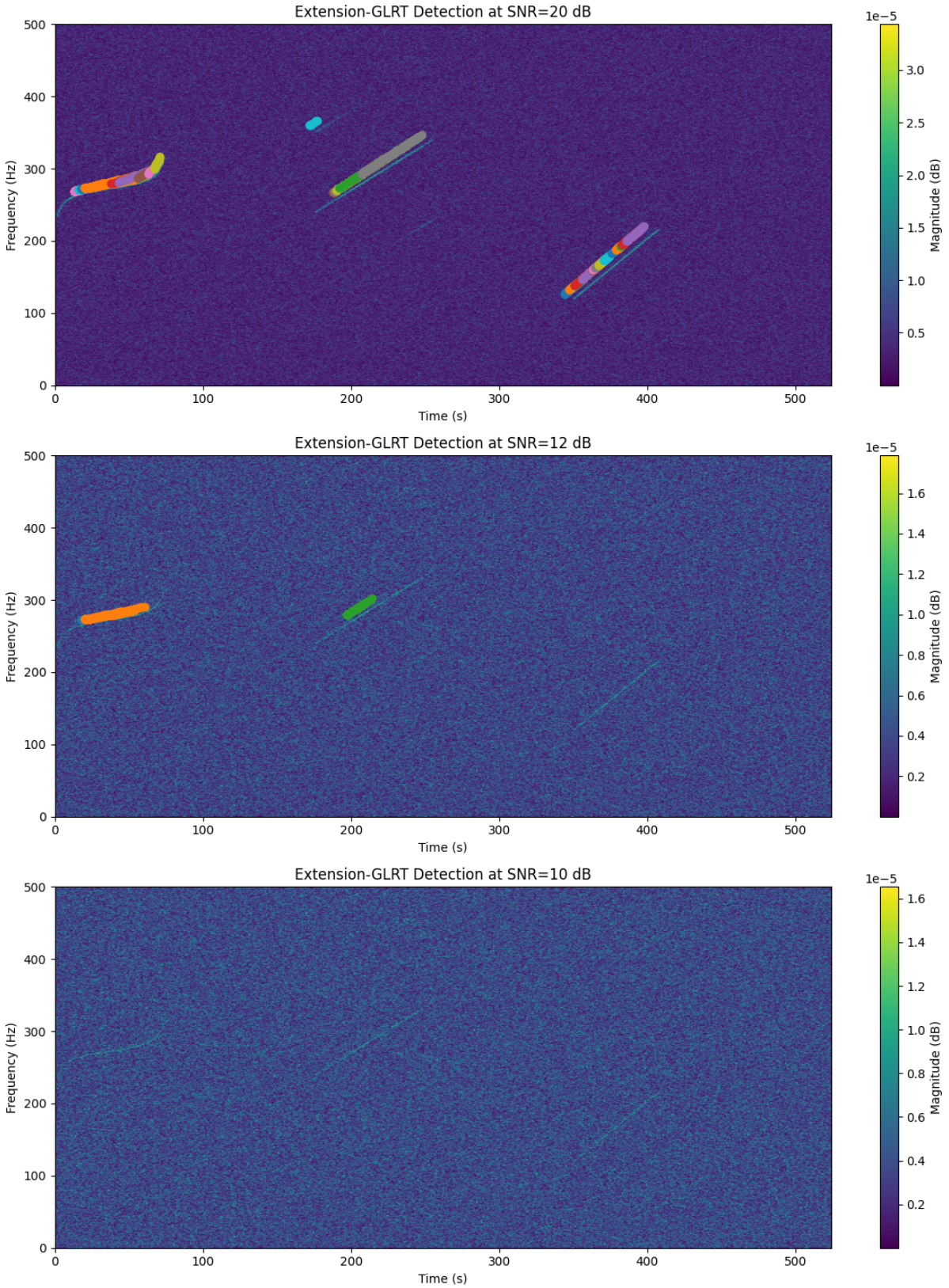


FIGURE 8 – Résultats de la méthode Paths-GLRT sur des spectrogrammes à différents niveaux de SNR. Chaque chemin détécté (cluster) a sa propre couleur. On a utilisé les hyperparamètres suivants : $PFA = 1e^{-}6, minpoints = 10$ pour le SNR 20 dB et $PFA = 1e^{-}4, minpoints = 15$ pour les SNRs 12 et 10 dB

4.5 GLRT + Hough Transform

Cette méthode combine le GLRT avec la transformée de Hough pour détecter des structures linéaires dans un spectrogramme temps-fréquence. Deux approches principales sont présentées : une transformée de hough fenêtrée et un GLRT sur transformée de Hough globale.

Référence : scikit-image: Hough Transform Example.

4.5.1 Approche Fenêtrée

Cette approche applique la transformée de Hough sur des fenêtres glissantes extraites du spectrogramme binaire obtenu après un GLRT point par point.

Étapes principales :

1. **GLRT point par point :** Le GLRT est appliqué sur chaque cellule du spectrogramme, et les amplitudes inférieures au seuil sont mises à zéro.
2. **Fenêtrage :** Le spectrogramme est découpé en fenêtres de taille fixe (freq_window, time_window), avec un pas (step_freq, step_time).
3. **Transformée de Hough locale :** Une transformée de Hough est appliquée dans chaque fenêtre pour détecter des lignes significatives.

Résultats : Bien que cette méthode permette de détecter des lignes, elle présente des limitations, notamment dans des environnements à faible SNR, et a des difficultés à distinguer des signaux spécifiques comme les NLFM.

Illustration : La figure 9 illustre des résultats obtenus avec cette approche.

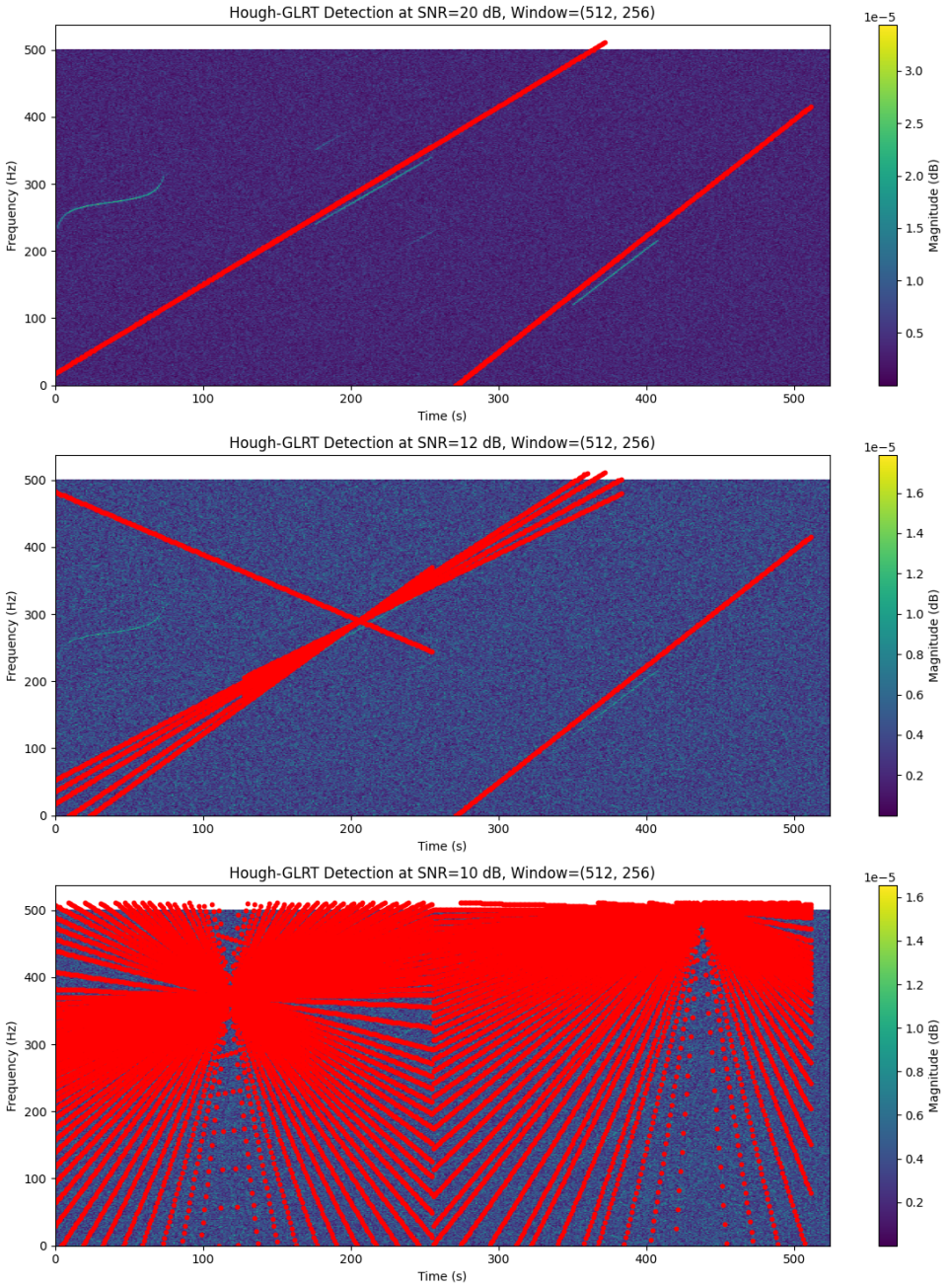


FIGURE 9 – Résultats de l'approche GLRT + Hough Transform fenêtrée. Les lignes détectées sont indiquées en rouge. Les paramètres incluent une PFA de $1e^{-6}$ et les valeurs par défaut de `hough_line_peaks` de `scikit-image`.

4.5.2 Approche Intégrée

L'approche intégrée applique une transformée de Hough sur tout le spectrogramme, sélectionne les N lignes les plus significatives, puis effectue un GLRT glissant sur ces lignes.

Étapes principales :

1. **Transformée de Hough globale :** Une transformée de Hough est appliquée sur le spectrogramme binaire après seuillage par le GLRT point par point.
2. **Sélection des lignes significatives :** Les N lignes ayant les scores d'accumulateur Hough les plus élevés sont retenues.
3. **GLRT glissant :** Un GLRT est appliqué sur des fenêtres glissantes le long de chaque ligne pour détecter les clusters significatifs.

Avantages et limitations : L'approche intégrée offre une meilleure localisation des clusters tout en réduisant le bruit grâce à une analyse plus ciblée des lignes significatives. Cependant, elle nécessite un réglage précis des paramètres, tels que le nombre de lignes N, les seuils γ du GLRT et la taille des fenêtres, pour maintenir un compromis entre complexité calculatoire et précision de détection.

Analyse des paramètres : L'efficacité de cette approche repose sur plusieurs paramètres :

- **Nombre de lignes significatives (N) :** Un nombre trop faible peut ignorer des structures importantes, tandis qu'un nombre trop élevé augmente la complexité computationnelle.
- **Seuils du GLRT (γ) :** Les seuils du GLRT point par point et du GLRT sur les fenêtres doivent être choisis en fonction du compromis entre détection et fausse alarme, généralement déterminés par une probabilité de fausse alarme (PFA).
- **Taille et pas des fenêtres :** Une fenêtre trop grande peut inclure du bruit ou des signaux non pertinents, tandis qu'un pas trop petit augmente le coût computationnel sans gain significatif en précision.
- **Résolutions ρ et θ :** Une résolution fine pour ρ et θ permet une meilleure précision des lignes détectées, mais augmente le coût calculatoire et peut rendre l'accumulateur Hough plus sensible au bruit.

Illustration : La figure 10 illustre les résultats de cette approche.

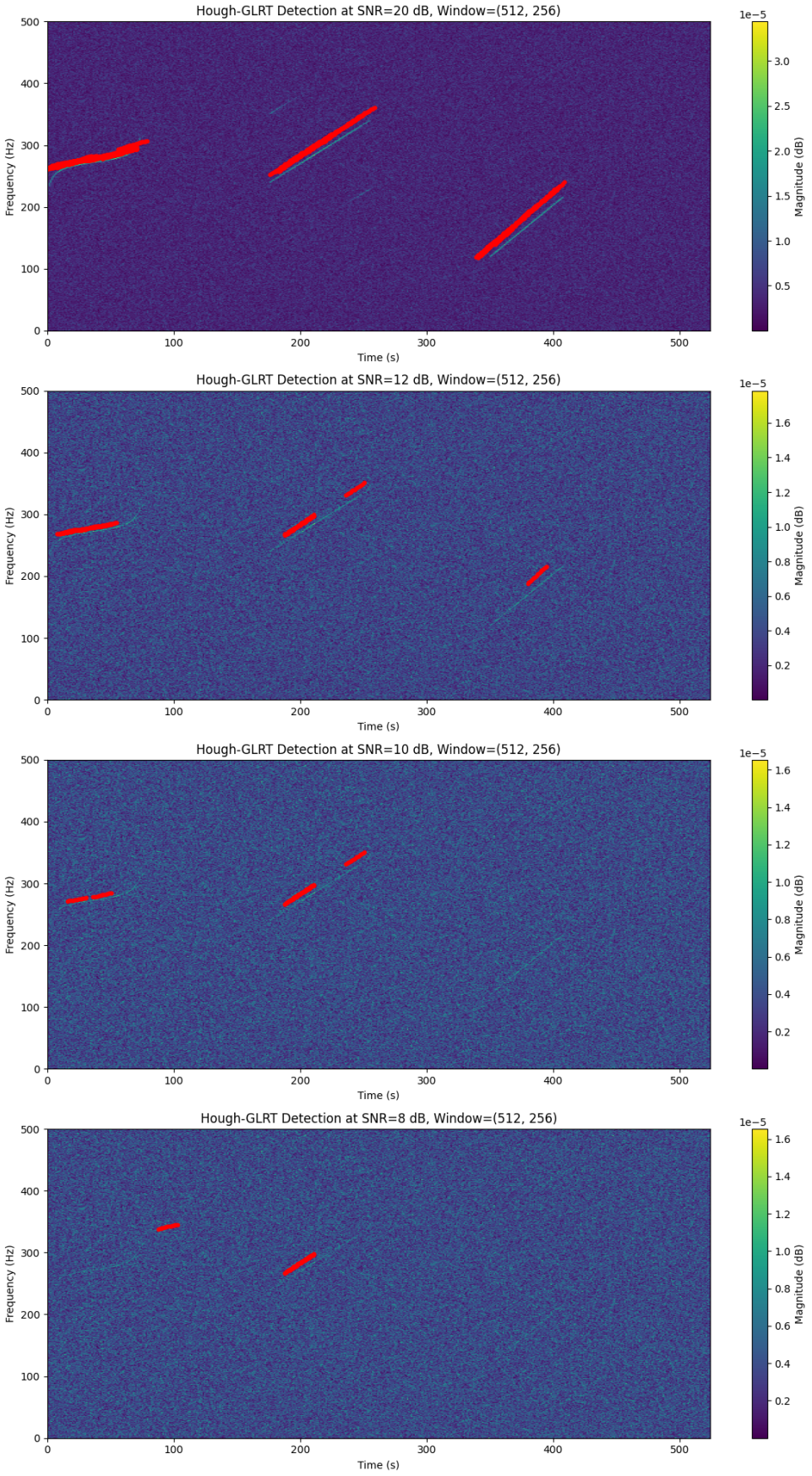


FIGURE 10 – Résultats de l'approche GLRT + Hough Transform intégrée. Les clusters détectés sont indiqués en rouge. Paramètres : $N = 100$ lignes, taille de fenêtre $= 16$, pas $= 4$, PFA point par point $= 1e^{-2}$, PFA sur fenêtre $= 1e^{-5}$.

4.6 GLRT + DBSCAN

Cette méthode réalise le seuillage du spectrogramme par le GLRT cellule par cellule, puis regroupe les points en cluster à l'aide de l'algorithme *DBSCAN* (*Density-Based Spatial Clustering of Applications with Noise*).

Référence : scikit-learn: DBSCAN.

Étapes principales

1. **GLRT cellule par cellule**
2. **Clustering avec DBSCAN :** L'algorithme *DBSCAN* regroupe les points candidats en clusters basés sur leur densité. Les clusters de points proches sont identifiés, tandis que les points isolés sont classifiés comme du bruit.

Analyse des paramètres

Les performances de cette méthode dépendent des choix des paramètres suivants :
- **Seuil du GLRT**
- ***eps* (distance maximale) :** Ce paramètre contrôle la distance maximale entre deux points pour qu'ils soient considérés comme voisins. Une valeur trop faible peut fragmenter des clusters, tandis qu'une valeur trop élevée peut regrouper des clusters distincts.
- ***min_ samples* (taille minimale des clusters) :** Ce paramètre définit le nombre minimum de points nécessaires pour former un cluster. Une valeur trop faible peut détecter des clusters de bruit, tandis qu'une valeur trop élevée peut ignorer des clusters significatifs de petite taille.

Illustration :

La figure 11 illustre les résultats obtenus en appliquant la méthode GLRT + DBSCAN . Les clusters détectés sont représentés par des regroupements de points colorés.

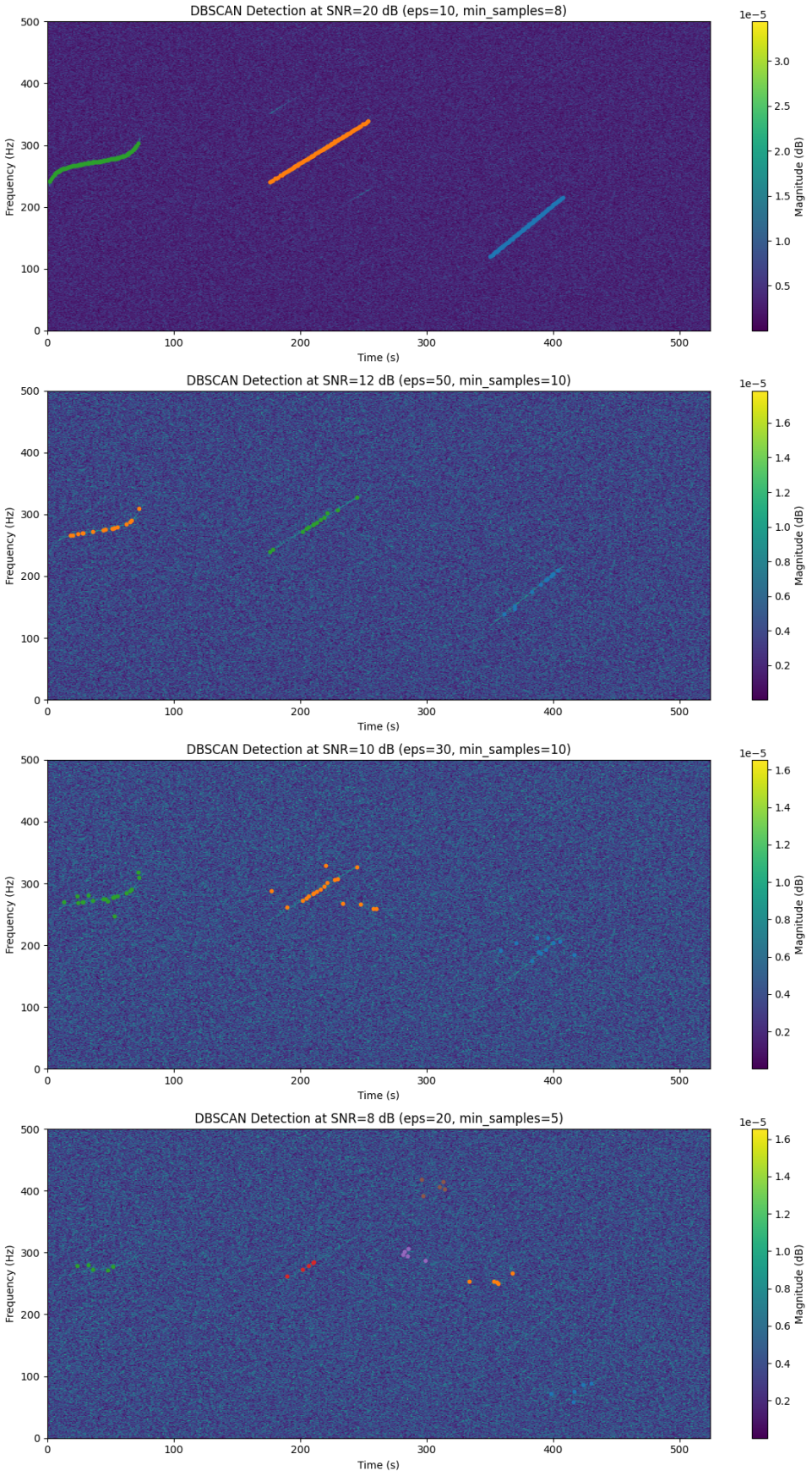


FIGURE 11 – Résultats de la méthode GLRT + DBSCAN. Les points candidats identifiés par le GLRT sont regroupés en clusters de différentes couleurs. Paramètres utilisés : Pour le SNR 20dB : eps = 10, min_samples = 8, $PFA = 1e^{-6}$, Pour le SNR 12 dB : eps = 30, min_samples = 10, $PFA = 1e^{-4}$, Pour le SNR 10 dB : eps = 50, min_samples = 10, $PFA = 1e^{-3}$, Pour le SNR 8 dB : eps = 20, min_samples = 5, $PFA = 1e^{-3}$,.

4.7 GLRT + Clustering

Cette section présente plusieurs méthodes de regroupement (*clustering*) utilisées en complément du *Generalized Likelihood Ratio Test* (GLRT) pour analyser les points détectés dans un spectrogramme temps-fréquence. L'objectif est d'identifier des clusters significatifs en regroupant les cellules ayant des caractéristiques similaires. Quatre méthodes principales de clustering sont explorées : **Graph-Based Clustering**, **Spectral Clustering**, **Mean-Shift Clustering** et **k-Means Clustering**.

4.7.1 Méthode Graph-Based Clustering

Dans cette approche, les points détectés par le GLRT sont modélisés comme des nœuds d'un graphe pondéré, où les arêtes représentent la distance entre les points. Deux variantes sont proposées :

- **Clustering par modularité de Louvain** : Cette méthode optimise une fonction de modularité pour diviser le graphe en communautés.
- **Clustering par arbre couvrant minimal (MST)** : Les clusters sont obtenus en partitionnant l'arbre couvrant minimal du graphe.

Référence : louvain-community: Louvain.

Référence : NetworkX: Minimum Spanning Tree.

4.7.2 Méthode Spectral Clustering

Le *Spectral Clustering* construit une matrice de similarité basée sur les distances entre points détectés et applique une factorisation spectrale pour regrouper les points en un nombre prédéfini de clusters (k). Cette méthode nécessite de définir à l'avance le nombre de clusters.

Référence : Scikit-learn: Spectral Clustering.

4.7.3 Méthode Mean-Shift Clustering

Le *Mean-Shift Clustering* ne requiert pas de spécifier à l'avance le nombre de clusters. Les points détectés sont regroupés en suivant les modes de densité locale dans l'espace des caractéristiques. Le paramètre clé est la largeur de bande, qui contrôle la portée des regroupements.

Référence : Scikit-learn: Mean-Shift Clustering.

4.7.4 Méthode k-Means Clustering

Le *k-Means Clustering* regroupe les points détectés en k clusters en minimisant la distance intra-cluster.

Référence : Scikit-learn: K-Means Clustering.

4.7.5 Résultats et Illustrations :

Les résultats des différentes méthodes sont illustrés à la figure 12, appliquées sur des spectrogrammes temps-fréquence à différents niveaux de SNR.

En SNR élevé, les méthodes de clustering montrent des limitations importantes, en particulier pour différencier les points de bruit des signaux. Les clusters détectés incluent souvent des points isolés.

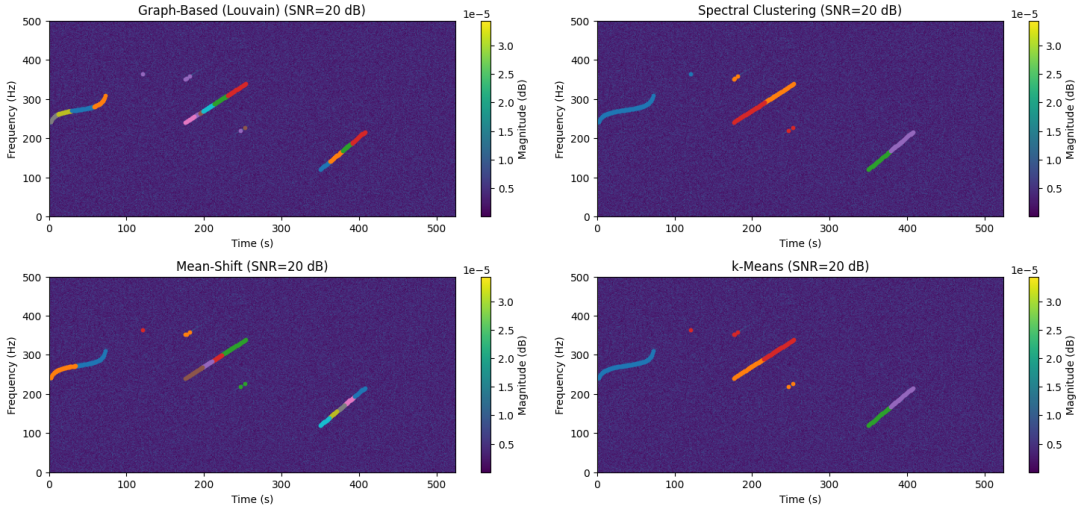


FIGURE 12 – Comparaison des méthodes de clustering après GLRT avec une fausse alarme de $1e^{-6}$ pour un SNR de 20 dB. Les clusters détectés sont représentés en différentes couleurs.

4.8 GLRT + k-NN Clustering

La méthode **GLRT + k-NN Clustering** combine l'analyse par le GLRT avec un algorithme de regroupement basé sur les k-plus-proches voisins (k-NN).

Référence : Scikit-learn: K-NN Clustering.

Étapes principales

1. **GLRT cellule par cellule**
2. **Construction du graphe des voisins :** Un graphe est construit où chaque point est un nœud, et une arête est ajoutée entre deux nœuds si leur distance est inférieure à une valeur maximale spécifiée. Les k-plus-proches voisins de chaque point sont considérés pour cette connexion.
3. **Détection des clusters :** Les composantes connexes du graphe sont identifiées comme des clusters potentiels.
4. **Filtrage des clusters par taille :** Seuls les clusters contenant un nombre de points supérieur ou égal à un seuil sont retenus comme clusters valides.

Paramètres

— γ : Seuil sur le GLRT.
— $n_{\text{neighbors}}$: Nombre de voisins considérés pour chaque point.
— max_neighbor_distance : Distance maximale pour relier deux points dans le graphe.
— size_threshold : Taille minimale d'un cluster pour qu'il soit retenu.

Illustration :

La figure 13 montre un exemple de détection de clusters en utilisant la méthode GLRT + k-NN Clustering sur un spectrogramme temps-fréquence.

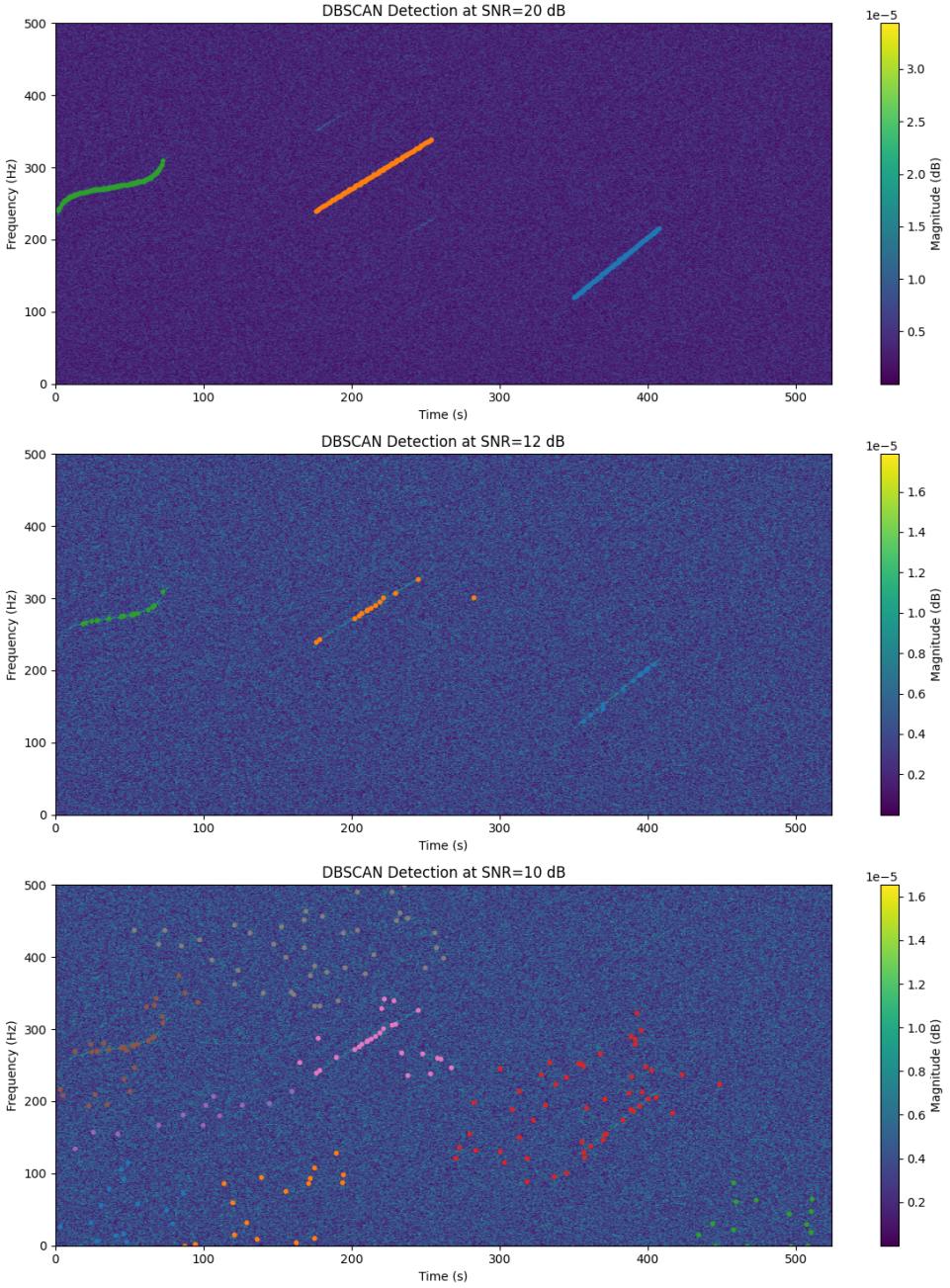


FIGURE 13 – Résultats de la méthode GLRT + k-NN Clustering. Les clusters valides sont représentés en différentes couleurs. Paramètres pour le SNR 20 dB : $nombredevoisins = 8$, $disancemaximale = 10$, $tailleminimale = 8$, $PFA = 1e^{-6}$, pour le SNR 12 dB : $nombredevoisins = 10$, $disancemaximale = 50$, $tailleminimale = 10$, $PFA = 1e^{-4}$, pour le SNR 10 dB : $nombredevoisins = 30$, $disancemaximale = 30$, $tailleminimale = 10$, $PFA = 1e^{-3}$.

5 Comparaison des méthodes

Dans cette section, nous comparons les performances des méthodes suivante :
— **GLRT + DBSCAN**
— **GLRT + k-NN**
— **GLRT + Extension**
— **GLRT + Paths**
— **GLRT + Hough Transform**

5.1 Analyse des performances sur différents niveaux de SNR

Cette première partie présente les performances des algorithmes pour différents niveaux de SNR. Chaque méthode est évaluée en termes de taux de détection, de taux de fausse alarme, et d'erreurs de caractérisation.

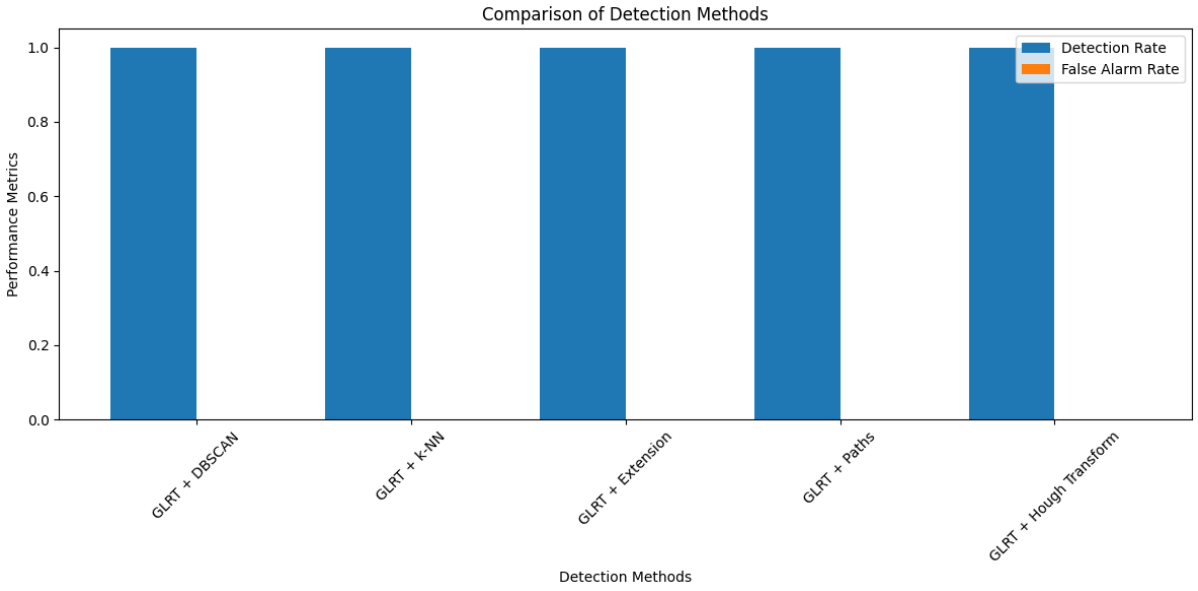


FIGURE 14 –

5.2 Analyse de la complexité calculatoire

5.2.1 Définition des Variables

— $\boldsymbol{Z}$: Le spectre temps-fréquence du signal, représenté par une matrice complexe de dimensions (n_f, n_t), où :
  — n_f est le nombre de points en fréquence.
  — n_t est le nombre de points en temps.
— p_f : Pas de fréquence utilisé dans les fenêtrages.
— p_t : Pas de temps utilisé dans les fenêtrages.
— d_s : Distance seuil pour fusionner les boîtes adjacentes.
— m : Nombre total de boîtes détectées

5.2.2 Box-GLRT avec Fusion des Boîtes

L'algorithme **Sliding-Window-GLRT with box merging** est une détection GLRT par boîte glissante (ou quadrillage avec un pas égal à la taille de la fenêtre) suivi d'une étape de

fusion des boîtes adjacentes pour regrouper les détections proches. L'analyse de sa complexité se décompose en plusieurs étapes principales :

— **Détection sur la fenêtre**
Cette étape applique la méthode GLRT sur chaque grille définie par sa taille en fréquence g_f et en temps g_t. La complexité de cette étape est proportionnelle au nombre total de grilles :

$$O(g_f \times g_t)$$

— **Détection sur le spectre**
On répète l'étape précédente autant de fois qu'il y a de fenêtres sur le spectre :

$$O\left(\frac{n_f}{p_f} \times \frac{n_t}{p_t}\right)$$

— **Fusion des boîtes adjacentes**
Une fois les détections effectuées, les boîtes adjacentes sont fusionnées si leur distance est inférieure au seuil. La fusion est effectuée en comparant chaque boîte avec toutes les autres, ce qui donne :

$$O(m^2)$$

— **Filtrage des boîtes par taille minimale**
Enfin, les boîtes restantes après fusion (m') dont les dimensions sont inférieures aux seuils définis sont éliminées :

$$O(m')$$

Complexité Totale : En combinant les étapes précédentes, la complexité globale de l'algorithme est donnée par :

$$O\left(\frac{n_f}{p_f} \times \frac{n_t}{p_t}\right) O(g_f \times g_t) + O(m^2)$$

Remarque 1 : m' ne fait plus partie de l'équation en remarquant que $m' =< m =< m^2$

Remarque 2 : Pour un quadrillage, cette expression se simplifie en :

$$O(n_f \times n_t) + O(m^2)$$

Remarque 3 : Pour une application à plusieurs grilles et en notant n_g le nombre de grilles, cette expression devient :

$$O(n_g)O(n_f \times n_t) + O(m^2)$$

5.2.3 GLRT avec Extension

L'algorithme **GLRT with Extension** détecte des clusters dans une image temps-fréquence en appliquant un critère GLRT sur des chemins étendus. Chaque chemin est construit dynamiquement en ajoutant le voisin avec la plus grande valeur de vraisemblance, tout en respectant un seuil GLRT ajusté à chaque itération. L'analyse de sa complexité calculatoire ext compliquée car elle repose sur les probabilité de détection et de fausse alarme du GLRT à chaque itérations. On abandonne l'idée de trouver une forme analytique à sa complexité calculatoire.

On peut néanmoins seuiller cette complexité au cas le plus simple, lorsqu'il n'y a pas de signal dans notre spectre :

$$O(n_f \times n_t \times p_{fa} \times (c_1 \times p_{fa}) \times (c_2 \times p_{fa})...)$$

où l'on a noté c_k le nombre moyen de voisins pour un cluster de k points.

Le cas où il existe un signal qui occupe n_c points dans le spectre. Dans ce cas la complexité devient :

$$O((n_f \times n_t - n_c) \times p_{fa} \times (c_1 \times p_{fa}) \times (c_2 \times p_{fa}) ...) + O(n_c \times p_d \times (c_{s,1} \times p_d + c_{\overline{s},1} \times p_{fa}) \times (c_{s,2} \times p_d + c_{\overline{s},2} \times p_{fa}) ...)$$

Où l'on a noté $c_{s,k}$ le nombre moyen de voisins contenant du signal pour k points et $c_{\overline{s},k}$ le nombre moyen de voisins ne contenant pas du signal qui est aussi égal à $c_k - c_{s,k}$. La première partie considère les points faussement attribué à du signal, la seconde les points où le GLRT détecte effectivement le signal.

Remarque : En plus d'avoir une complexité calculatoire difficile à exprimer, cet algorithme n'est pas parallélisable car les opérations sont interdépendantes.

5.2.4 GLRT avec chemins linéaires

L'algorithme **GLRT avec chemins linéaires** détermine les clusters en évaluant le GLRT sur les chemins (lignes) reliant 2 points détectés. L'analyse de sa complexité calculatoire se décompose comme suit :

- **Calcul initial du GLRT pour chaque point :**

$$O(n_f \times n_t)$$

- **Extension des chemins valides :**
  Pour chaque cellule candidate, l'algorithme trouve un chemin valide en cherchant le voisin le plus proche et en validant le segment intermédiaire à l'aide du GLRT.
  - **Recherche du voisin le plus proche :** Chaque point candidat compare sa distance à tous les autres candidats. Si le nombre total de candidats est c, la recherche du voisin pour un point a une complexité de $O(c)$. Répéter cette opération pour tous les candidats donne une complexité globale de :

$$O(c^2)$$

  - **Validation des segments intermédiaires :** On récupère tous les points traversé par la ligne qui rejoints les deux points puis on calcul le GLRT sur cette ensemble de points. Si chaque segment contient en moyenne l points, la validation de tous les segments est proportionnelle à :

$$O(c \times l)$$

- **Filtrage des chemins :**
  Enfin, les chemins moins de points que le seuil sont éliminés. Cette étape nécessite de parcourir tous les chemins générés, et sa complexité est proportionnelle au nombre total de chemins détectés k :

$$O(k)$$

Complexité Totale : En combinant les différentes étapes, la complexité globale de l'algorithme est donnée par :

$$O(n_f \times n_t) + O(c^2) + O(c \times l) + O(k)$$

où c est le nombre de candidats, l est la taille moyenne d'un segment et k est le nombre de chemins finaux.

Remarque 1 : Ici c est en réalité égale à $n_s p_d (n_t n_f - n_s) p_{fa}$

Remarque 2 : La complexité est dominée par la recherche du voisin le plus proche ($O(c^2)$) et peut être optimisée à l'aide de structures adaptées (e.g., k-d trees) ou de méthodes heuristiques pour limiter la recherche.

Remarque 3 : Pour des chemins long dans des régions spécifiques (par exemple des signaux très long), la validation des segments intermédiaires ($O(c \times l)$) peut également devenir un facteur significatif.

Remarque 4 : D'après les deux premières remarques, la complexité calculatoire est très impacté par le seuil du premier GLRT qui détermine le nombre de points c .

5.2.5 Hough-GLRT

L'algorithme **Hough-GLRT** combine une détection GLRT avec une transformée de Hough pour identifier les lignes significatives dans un spectre temps-fréquence. Puis un second GLRT appliqué à un fenêtrage sur les lignes détéctées. L'analyse de sa complexité peut être divisée en plusieurs étapes clés correspondant aux phases principales de l'algorithme.

Étape 1 : Premier GLRT cellule par cellule

$$O(n_f \times n_t)$$

Étape 2 : Construction de l'accumulateur Hough Pour chaque angle θ, la méthode calcule les distances ρ associées à chaque point significatif du spectre. Les calculs sont effectués pour tous les angles et tous les points non nuls, soit :

$$O(n_\theta \times c)$$

où :
- n_θ est le nombre de pas d'angles θ, soit $n_\theta = \frac{\pi}{\theta}$.
- c est le nombre de points extraits par le premier GLRT.

Étape 3 : Sélection des lignes les plus probables Cette étape utilise un tri partiel pour identifier les n_l lignes ayant les plus grandes valeurs dans l'accumulateur Hough. La complexité dépend du nombre de lignes à trier dans l'accumulateur Hough ($n_{total,l}$ qui est forcément inférieur à la taille de l'accumulateur) :

$$O(\log(n_{total,l}) \times n_l)$$

Étape 4 : Calcul GLRT et regroupement Pour chaque ligne détectée, l'algorithme calcule le GLRT sur des fenêtres glissantes.
Une fenêtre de taille w glisse le long de chaque ligne. Le nombre d'opérations dépend du nombre de points par ligne d_l, de w, et du pas s :

$$O\left(d_l \times \frac{w}{s}\right)$$

Pour n_l lignes détectées :

$$O\left(n_l \times d_l \times \frac{w}{s}\right)$$

Complexité totale La complexité totale est obtenue en sommant les contributions de toutes les étapes précédentes. Sous les hypothèses où c (le nombre de points significatifs) est proportionnel à $n_f \times n_t$, on obtient :

$$O(n_f \times n_t) + O(n_\theta \times c) + O(\log(n_{total,l}) \times n_l) + O(n_l \times d_l \times \frac{w}{s})$$

En remplaçant c par $k \times n_f \times n_t$ (où k est le ratio des points significatifs) :

$$O(nfnt) + O(nc) = O(n_f \times n_t \times (1 + n_\theta \times k)) = O(n_f \times n_t \times n_\theta \times k)$$

Et donc la complexité totale devient :

$$O(n_f \times n_t \times n_\theta \times k) + O(n_l \times (\log(n_{total,l}) + d_l \times \frac{w}{s}))$$

Remarque : La complexité totale est coupée en deux. L'optimisation peut se faire d'un côté sur le pas de l'angle de la transformée de Hough ou sur le seuil du GLRT et de l'autre sur le seuil sur le nombre de lignes ou sur la taille de la fenêtre et son pas.

5.2.6 GLRT + DBSCAN

L'algorithme **GLRT + DBSCAN** combine une détection GLRT appliquée cellule par cellule dans le spectre temps-fréquence avec l'algorithme DBSCAN pour regrouper les points détectés en clusters. L'analyse de la complexité peut être divisée en deux étapes principales.

Étape 1 : Calcul du GLRT cellule par cellule

$$O(n_f \times n_t)$$

Étape 2 : Clustering avec DBSCAN L'algorithme DBSCAN est appliqué aux points candidats, c'est-à-dire ceux qui dépassent le seuil γ. Le nombre de ces points est noté $c = k \cdot n_f \cdot n_t$, où k est le ratio de points retenus par rapport au total. Pour calculer la complexité de DBSCAN on se base sur la source suivante : DBSCAN Wiki

— **Recherche de voisins (ε-voisinage)** :
Pour chaque point candidat, DBSCAN recherche tous les points situés dans un rayon ε. Dans le pire des cas (sans structure optimisée), cette opération nécessite de comparer chaque point avec tous les autres, donnant une complexité quadratique :

$$O(c^2)$$

Cependant, en utilisant une structure optimisée comme un arbre, cette étape peut être réduite à $O(c \cdot \log c)$.

— **Regroupement des points en clusters** :
Une fois les voisins identifiés, DBSCAN regroupe les points en clusters en effectuant une exploration des voisins. Cette étape est linéaire par rapport au nombre de points c :

$$O(c)$$

Ainsi, la complexité totale de DBSCAN est donnée par :

$$O(c^2 + c)$$

ou, avec une structure optimisée :

$$O(c \cdot \log c + c)$$

Complexité totale En combinant les deux étapes, la complexité totale de l'algorithme obtimisé est donnée par :

$$O(n_f \cdot n_t) + O(c \log(c) + c)$$

En remplaçant $c = k \cdot n_f \cdot n_t$, la complexité devient :

$$O(n_f \cdot n_t) + O(k \cdot n_f \cdot n_t \cdot \log(k \cdot n_f \cdot n_t))$$

Complexité dominante Sous l'hypothèse où k est faible (nombre limité de points candidats), la complexité est dominée par le terme quadratique :

$$O(n_f \cdot n_t \cdot (1 + k \log(k \cdot n_f \cdot n_t))$$

Remarque : En supposant k très petit, la complexité calculatoire de cet algorithme est dominée par le GLRT point par point. Sinon il est dominé par DBSCAN d'une compléxité égale à $O(c \log(c))$. Quoi qu'il en soit, la complexité calculatoire de cet algorithme reste très faible.

5.2.7 GLRT + k-NN

L'algorithme **GLRT + k-NN** combine une détection GLRT cellule par cellule avec un clustering basé sur les k-plus-proches voisins (k-NN) pour regrouper les points détectés en clusters. L'analyse de sa complexité se décompose en trois étapes principales :

Étape 1 : Calcul initial du GLRT cellule par cellule

$$O(n_f \cdot n_t)$$

Étape 2 : Recherche des k-plus-proches voisins Une fois les points candidats extraits ($c = k \cdot n_f \cdot n_t$), l'algorithme des k-NN est appliqué. La complexité de cette étape est donnée par :

- **Construction de la structure de recherche :**
  Les points candidats sont organisés dans une structure optimisée pour les recherches, telle qu'un arbre k-d. La complexité de cette phase dépend du nombre de points candidats c et est généralement :
  $$O(c \cdot \log c)$$
- **Recherche des voisins pour chaque point :**
  Pour chaque point candidat, k voisins sont identifiés en explorant la structure. Cette étape a une complexité proportionnelle au nombre de points candidats c et aux voisins recherchés v :
  $$O(c \cdot v)$$

Ainsi, la complexité totale de l'étape de recherche des k-NN est :

$$O(c \cdot \log c + c \cdot v)$$

Étape 3 : Construction du graphe et clustering Une fois les voisins identifiés, les points sont regroupés en clusters en construisant un graphe, où chaque point est connecté à ses k voisins si la distance entre eux est inférieure à un seuil. Cette étape comprend :

- **Construction du graphe :**
  Le graphe est construit en connectant les k-voisins de chaque point, avec une complexité :
  $$O(c \cdot v)$$

— **Identification des composantes connectées :**
Les clusters sont extraits en identifiant les composantes connectées dans le graphe. Cette étape est linéaire par rapport au nombre de points candidats c :

$$O(c)$$

— **Filtrage des clusters :**
Les clusters trop petits (de taille inférieure à `size_threshold`) sont éliminés. Cette opération a une complexité proportionnelle au nombre de clusters, noté n_c, ce qui est toujours à c :

$$O(n_c)$$

Ainsi, la complexité totale de l'étape de clustering est donnée par :

$$O(c \cdot v + c + n_c) = O(c \cdot v)$$

Complexité totale En combinant les étapes précédentes, la complexité totale de l'algorithme est donnée par :

$$O(n_f \cdot n_t + c \cdot \log c + c \cdot v)$$

En remplaçant c par $k \cdot n_f \cdot n_t$, on a :

$$O(n_f \cdot n_t \cdot (1 + klog(c) + v)) = O(n_f \cdot n_t \cdot (klog(c) + v))$$

Remarque : Lorsque le nombre de points extrait par le GLRT est faible, alors la complexité calculatoire est donnée par $O(n_f \cdot n_t \cdot v)$ sinon elle est donnée par $O(clog(c))$

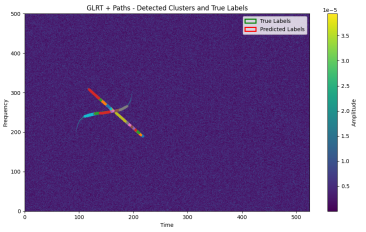
5.2.8 Tableau des complexités mesurées

Algorithme	**Temps moyen (s)**
GLRT + DBSCAN	0.0047
GLRT + k-NN	1.7152
GLRT + Extension	19.2653
GLRT + Paths	1.9012
GLRT + Hough Transform	0.4781
Grid GLRT	0.0078
Sliding-Window GLRT	0.638

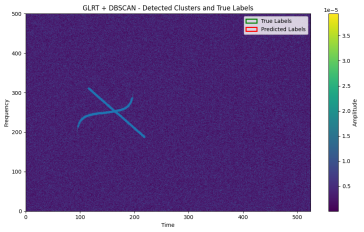
TABLE 1 – Comparaison des temps d'exécution moyens pour chaque méthode.

5.3 Analyse des performances sur des données entrelacées

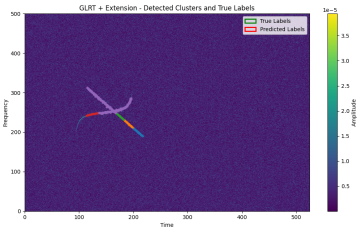
5.3.1 Illustrations



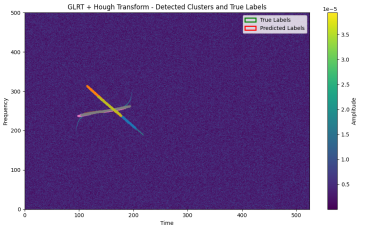
(a) GLRT + Paths



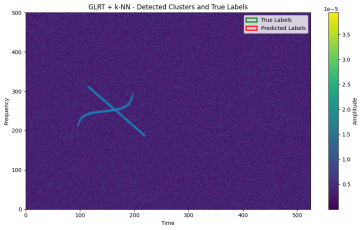
(b) GLRT + DBSCAN



(c) GLRT + Extension



(d) GLRT + Hough Transform



(e) GLRT + k-NN

FIGURE 15 – Résultats de détection pour des données entrelacées avec différentes méthodes à un SNR de 20 dB.

5.3.2 Analyse

Les méthodes de détections de chemins linéaires tels que la méthode utilisant la transformée de Hough ou GLRT+Paths permettent de distinguer les deux signaux à fort SNR. DBSCAN et KNN détectent bien les deux signaux mais l'assimilent à un seul cluster.

5.4 Conclusion