UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et de génie informatique

RÉDUCTION DE L'ÉGO-BRUIT DE ROBOTS

Mémoire de maitrise
Spécialité : génie électrique

Pierre-Olivier Lagacé

Sherbrooke (Québec) Canada

Juin 2023

MEMBRES DU JURY

François Grondin

Directeur

François Ferland

Codirecteur

François Michaud

Évaluateur

Adina Panchea

Évaluateur

RÉSUMÉ

En robotique, il est désirable d'équiper les robots du sens de l'audition afin de mieux interagir avec les utilisateurs et l'environnement. Cependant, le bruit causé par les actionneurs des robots, nommé égo-bruit, réduit considérablement la qualité des segments audios. Conséquemment, la performance des techniques de reconnaissance de la parole et de détection d'évènements sonores est limitée par la quantité de bruit que le robot produit durant ses mouvements. Le bruit généré par les robots diffère considérablement selon l'environnement, les moteurs, les matériaux utilisés et même selon l'intégrité des différentes composantes mécaniques. L'objectif du projet est de concevoir un modèle de réduction d'égo-bruit robuste utilisant plusieurs microphones et d'être capable de le calibrer rapidement sur un robot mobile.

Ce mémoire présente une méthode de réduction de l'égo-bruit combinant l'apprentissage de gabarit de matrice de covariance du bruit à un algorithme de formation de faisceau de réponses à variance minimum sans distorsion. L'approche utilisée pour l'apprentissage des matrices de covariances permet d'enregistrer les caractéristiques spatiales de l'égo-bruit en moins de deux minutes pour chaque nouvel environnement. L'algorithme de faisceau permet, quant à lui, de réduire l'égo-bruit du signal bruité sans l'ajout de distorsion non-linéaire dans le signal résultant. La méthode est implémentée sous *Robot Operating System* pour une utilisation simple et rapide sur différents robots.

L'évaluation de cette nouvelle méthode a été effectuée sur un robot réel dans trois environnements différents : une petite salle, une grande salle et un corridor de bureau. L'augmentation du ratio signal-bruit est d'environ 10 dB et est constante entre les trois salles. La réduction du taux d'erreur des mots de la reconnaissance vocale se situe entre 30 % et 55 %. Le modèle a aussi été testé pour la détection d'évènements sonores. Une augmentation de 7 % à 20 % de la précision moyenne a été mesurée pour la détection de la musique, mais aucune augmentation significative pour la parole, les cris, les portes qui ferment et les alarmes. La méthode proposée permet une utilisation plus accessible de la reconnaissance vocale sur des robots bruyants.

De plus, une analyse des principaux paramètres a permis de valider leurs impacts sur la performance du système. Les performances sont meilleures lorsque le système est calibré avec plus de bruit du robot et lorsque la longueur des segments utilisés est plus longue. La taille de la Transformée de Fourier rapide à court terme (*Short-Time Fourier Transform*) peut être réduite pour réduire le temps de traitement du système. Cependant, la taille de cette transformée impacte aussi la résolution des caractéristiques du signal résultant. Un compromis doit être faire entre un faible temps de traitement et la qualité du signal en sortie du système.

Mots-clés : Égo-bruit, robot, rehaussement audio, évènement audio inconnu

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ACRONYMES

Acronyme	Définition
AP	Précision moyenne (*Average Precision*)
ASR	Reconnaissance vocale (*Automatic Speech Recognition*)
BSS	Séparation des sources à l'aveugle (*Blind Source Separation*)
DOA	Direction d'arrivée (*Direction of Arrival*
ICA	Analyse des composantes interdépendante (*Independant Component Analysis*
IMCRA	Contrôle récursif minime moyen amélioré (*Improved Minimal Controlled Recursive Averaging*)
iSTFT	Transformée de Fourier inverse rapide à court terme (*inverse Short-Time Fourier Transform*)
ILRMA	Analyse des matrices indépendantes à bas rang (*Independent Low-Rank Matrix Analysis*)
MFCC	Cœffient cepstraux de la fréquence de Mel (*Mel-Frequency-Cepstral-Cœfficients*)
MFT	Théorie des caractéristiques manquante (*Missing Feature Theory*)
MNMF	Factorisation de matrice non négative à plusieurs canaux (*Multichannel Non-Negative Matrix Factorization*)
MVDR	Réponse à variance minimum sans distorsion (*Minimum Variance Distortionless Response*)
NMF	Factorisation de matrice non négative (*Non-Negative Matrix Factorization*)
PCA	Analyse des composantes principales (*Principal Component Analysis*)
PO-KSVD	Décomposition en valeurs singulières optimisée (*Phase optimized singular value decomposition*)
PSD	Densité spectrale de puissance (*Power Spectral Density*)
ROS	Système d'exploitation de robot (*Robot Operating System*)
SCM	Matrice de covariance spatiale (*Spatial Covariance Matrix*)

SDR	Ratio signal-distortion (*Signal-to-Distortion Ratio*)
SNR	Ratio signal-bruit (*Signal-to-Noise Ratio*)
STFT	Transformée de Fourier rapide à court terme (*Short-Time Fourier Transform*)
SVM	Machine à vecteur de support (*Support Vector Machine*)
TFS	Filtrage temps-fréquence (*Time-Frequency Spatial Filtering*)
UAV	Véhicule aérien sans pilote (*Unmanned Aerial Vehicles*)
UGV	Véhicule terreste sans pilote (*Unmanned Ground Vehicles*)
WER	Taux d'erreur des mots (*Word Error Rate*)

CHAPITRE 1

INTRODUCTION

L'arrivée des robots autonomes, autant dans l'industrie que dans les milieux domestiques, permet aux utilisateurs de bénéficier d'une aide pour accomplir certaines tâches. Il serait souvent nécessaire de donner des commandes vocales à ces systèmes pour mettre à jour les objectifs à atteindre. Par exemple, il serait désirable de demander vocalement à un aspirateur autonome de quitter la cuisine alors que les occupants mangent. La reconnaissance vocale permet de traduire la parole en texte pouvant être interprété par les systèmes informatisés. Toutefois, les performances des techniques traditionnelles en reconnaissance de la parole se heurtent aux bruits ambiants qui détériorent la qualité des signaux.

En robotique, les signaux vocaux sont également contaminés par l'égo-bruit, ce qui rend la reconnaissance de la parole plus difficile [30]. L'égo-bruit est composé du bruit généré par les éléments mécaniques et électriques des composantes du robot lui-même. En reprenant l'exemple du robot aspirateur, les actionneurs aux roues et la pompe produisent un bruit important. En équipant ce robot de microphones, il est difficile, voire impossible, de les éloigner de ces sources de bruits. La qualité des signaux de parole diminue alors considérablement, ce qui affecte directement les performances de reconnaissance vocale.

Le système d'audition d'un robot doit être robuste autant aux bruits ambiants qu'à l'égo-bruit. L'apprentissage profond a récemment permis de concevoir des méthodes de reconnaissance vocale robustes au bruit ambiant. Cette approche est possible grâce à la disponibilité d'immenses ensembles de données contenant ce type de bruit jumelé avec de la parole. Cependant, les segments audios contenant le bruit interne de robots sont beaucoup plus rares. Créer un tel ensemble de données est une tâche ardue et demande beaucoup de temps, car les robots mobiles sont constamment en mouvement et naviguent dans des environnements connus ou inconnus. Les caractéristiques acoustiques des environnements peuvent différer. De plus, le matériel utilisé dans la conception du robot peut avoir un impact sur le bruit généré. Par exemple, le contact du caoutchouc des pneus sur une surface lisse ou sur du béton ne produit pas le même son. Il est aussi possible que des modifications mécaniques soient apportées aux robots au fil du temps, modifiant ou générant différents bruits. Par conséquent, le modèle de réduction doit être rapide et facile à calibrer pour s'adapter aux différents robots et aux différents environnements.

En robotique, l'audio peut permettre au robot de reconnaitre des évènements et lui permettre d'agir rapidement [29]. En plus de la reconnaissance vocale, d'autres techniques audios comme la classification d'évènements sonores sont importantes. De ce besoin découle l'importance de supprimer seulement l'égo-bruit du robot afin de rehausser le reste du signal. Ce modèle de réduction de l'égo-bruit est donc placé au début de la chaine de traitement audio des robots. Pour que son utilisation soit viable, il doit être fonctionnel en temps réel en ajoutant peu de latence aux signaux audios.

L'objectif principal de ce projet est de concevoir un modèle de réduction d'égo-bruit utilisant plusieurs microphones et pouvant être calibré rapidement sur un robot sans l'ajout de capteurs externes. La question de recherche liée à cet objectif est :

Est-il possible de concevoir un modèle de réduction de l'égo-bruit multicanal robuste et rapide à calibrer sur un robot mobile ?

Ce mémoire présente une méthode de réduction de l'égo-bruit combinant l'apprentissage de matrices de covariance du bruit à un algorithme de formation de faisceau de réponses à variance minimum sans distorsion (*Minimum Variance Distortionless Response* – MVDR). Cette méthode utilise le signal multi-canal provenant d'une matrice de microphones. Le système se calibre en moins de 90 secondes, fonctionne en temps réel et n'ajoute pas de distorsion au signal filtré, ce qui permet de l'utiliser avec des modèles de reconnaissance de la parole et de détection d'évènements sonores préentrainés.

Le chapitre 2 montre les différentes techniques utilisées dans le domaine de réduction de l'égo-bruit. Ensuite, le chapitre 2.3.6 présente le développement du système et les résultats obtenus sous la forme d'un article soumis à la conférence *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Par la suite, le chapitre 4 introduit une analyse des trois principaux paramètres du système. Puis, le chapitre 5 décrit l'implémentation détaillée de la solution et les améliorations futures. Finalement, le chapitre 6 fait un retour sur les points importants du projet.

CHAPITRE 2

TECHNIQUES D'EXTRACTION DE CARACTÉRISTIQUES AUDIOS EN PRÉSENCE DE L'ÉGO-BRUIT DE ROBOTS

Ce chapitre a pour objectif de situer le projet de recherche par rapport aux travaux publiés dans le domaine de la réduction de l'égo-bruit des robots. La première section introduit des travaux où les données des moteurs sont directement incluses dans les modèles de reconnaissance vocale (*Automatic Speech Recognition* – ASR) et de localisation. La deuxième section décrit les différentes techniques de rehaussement de la parole. La troisième section présente les techniques de réduction de l'égo-bruit.

2.1 Reconnaissance vocale

L'égo-bruit peut être inclus directement dans l'entrainement des techniques de reconnaissance vocale et de localisation. Lee et Chang [24] proposent une technique qui inclut l'état (actif/inactif) des moteurs dans un réseau de neurones de reconnaissance vocale. Ils proposent aussi une technique qui extrait les caractéristiques du mouvement du robot [25]. Pour y arriver, ils entrainent un premier réseau de neurones à prédire l'état des moteurs à partir des cœfficients cepstraux de la fréquence de Mel (*Mel-Frequency-Cepstral-Cœfficients* – MFCC). Ils extraient le vecteur de l'une des couches du réseau, appelé goulot d'étranglement. Ils entrainent ensuite un modèle ASR avec ce vecteur en entrée et les MFCC. Au niveau de la localisation, Furukawa et coll. [9] propose un système où un processus gaussien estime par régression la matrice de corrélation de l'égo-bruit à partir de l'état des capteurs d'un drone. Ils utilisent ensuite l'algorithme de décomposition des valeurs propres généralisé avec multiple classification de signal (*Generalized Eigenvalue Decomposition-based Multiple Signal Classification*) pour estimer la direction d'arrivée (*Direction of Arrival* – DOA) du signal dans un spectre temps-direction. La limitation de ces techniques est qu'elles sont liées à l'ASR. Toutes les techniques doivent être en mesure de fonctionner avec l'égo-bruit, ce qui les rend plus complexes.

2.2 Rehaussement de la parole

Il existe des techniques permettant de rehausser le signal de la parole dans un contexte robotique où l'égo-bruit réduit considérablement la qualité des signaux audio.

2.2.1 Filtre simple

Rascon et coll. [34] proposent trois techniques couramment utilisées pour filtrer le bruit en traitement audio : la séparation non linéaire de Berouti, l'estimation adaptative basée sur les quantiles et le contrôle récursif minime moyen amélioré (*Improved Minimal Controlled Recursive Averaging* – IMCRA). L'utilisation d'un filtre non linéaire, comme celui de Berouti, ajoute de la distorsion au signal de sortie, ce qui dégrade considérablement la qualité audio. Le IMCRA demande de l'ajustement manuel de paramètres, qui peuvent changer d'un robot à l'autre. L'avantage des petits filtres comme ceux implémentés dans cet article est qu'ils demandent peu de ressources computationnelles. Ils sont donc fonctionnels sur la majorité des plateformes robotiques. Cependant, les performances ne sont pas satisfaisantes lorsque le Ratio Signal-Bruit (*Signal-to-Noise Ratio* – SNR) est plus petit que 0 dB.

2.2.2 Séparation à l'aveugle

La technique de séparation des sources à l'aveugle (*Blind Source Separation* – BSS) permet de séparer plusieurs sources sonores à l'aide d'un ou plusieurs microphones. Wang et Cavallaro [48] utilisent cette technique pour séparer la parole de l'égo-bruit. Plus précisément, ils évaluent le nombre de microphones nécessaires pour réduire complètement l'égo-bruit. Dans leurs travaux, ils définissent que la méthode nécessite plus de microphones que le nombre de sources de bruit entourant le drone. Puisque les moteurs ne produisent pas exactement le même bruit, il faut un microphone pour chaque moteur. Il faut aussi un microphone pour le bruit ambiant et un microphone par interlocuteur. Cette approche est fonctionnelle pour des robots avec peu d'actionneurs. Cependant, certains robots humanoïdes contiennent plus de 30 moteurs. Par exemple, Sophia [1] contient 32 moteurs seulement pour l'expression faciale de son visage et 86 au total. Selon les résultats de l'article, la performance du BSS diminue rapidement en situation dynamique, et en robotique, l'interlocuteur ou le robot sont souvent en mouvement. De plus, les auteurs mentionnent que la réverbération est assumée comme nulle, car les drones fonctionnent majoritairement à l'extérieur. Donc, les résultats peuvent différer considérablement pour un robot intérieur. Une deuxième limitation survient lorsqu'il y a plus de microphones que de sources à séparer. L'algorithme peut sur-déterminer le nombre de sources, ce qui apporte des problèmes de séparation. Effectivement, la parole d'une seule personne peut

1. https ://www.hansonrobotics.com/sophia-2020/

se retrouver sur deux canaux différents, ce qui détériore la qualité du signal en sortie. De plus, l'algorithme souffre d'ambiguïté de permutation. Cette problématique consiste à la permutation des canaux de sortie entre deux segments différents. La parole peut se retrouver sur le canal 1 lors du premier segment, puis sur le canal 2 lors du deuxième segment. Il faut donc classifier les canaux de sortie pour s'assurer d'obtenir la parole lors du traitement en temps réel sur plusieurs segments.

Encinas et coll. [5] utilisent un algorithme similaire nommée analyse à spectre singulier (*Singular Spectrum Analysis*). Cet algorithme sépare les sources à l'aide d'un seul microphone et il élimine le problème du BSS lorsqu'il y a plus de sources que de microphones. Il doit cependant connaitre le nombre de sources à séparer. Ils utilisent deux sources, une pour le bruit du drone et l'autre pour la parole. Dans un cas réel, il n'est pas toujours possible de connaitre en avance le nombre de sources à séparer. Tous leurs tests sont effectués dans une chambre anéchoïque, donc sans réverbération. Comme mentionné par Loellman et coll. [26], la réverbération est un défi important pour la communication humain-robot. Les résultats de l'article ne représentent donc pas des situations réelles où la réverbération est présente.

Bando et coll. [1] proposent un algorithme d'analyse des matrices indépendantes à bas rang (*Independent Low-Rank Matrix Analysis* – ILRMA) qui est aussi une BSS. Ils combinent l'algorithme avec un filtre d'erreur quadratique moyenne à faible durée sur l'amplitude spectrale (*Minimum Mean-Square-Error Short-Time Spectral Amplitude*). Cet algorithme ne peut pas fonctionner en temps réel sur des ordinateurs à plus petite capacité de calcul. Dans leur article, ils mentionnent que traiter 5 secondes audios prends 20 secondes. Leur robot est un robot de secours et les signaux audios sont écoutés par des opérateurs lors de sinistres. Les opérateurs doivent être en mesure d'écouter le flux audio en temps réel pour repérer la voix humaine. Les auteurs ont donc implémenté un autre algorithme pour l'écoute en temps réel basé sur l'analyse robuste des composantes principales. Cet algorithme n'est pas linéaire, ce qui ajoute de la distorsion aux signaux audios. Il permet cependant aux opérateurs du robot d'écouter le flux audio en temps réel malgré la faible qualité du son. Lorsqu'il semble y avoir la parole humaine, ils enregistrent les signaux audios et les traitent sur un autre système avec l'algorithme ILRMA pour les écouter avec une meilleure qualité. Ce système est intéressant pour ce type de robots. Cependant, la latence engendrée par ce système peut dégrader la qualité de l'interaction humain-robot. Il est préférable d'avoir un système qui fonctionne en temps réel et qui offre directement une qualité audio acceptable.

2.2.3 Filtrage temps-fréquence

Wang et Cavallaro [49] comparent le BSS au filtrage temps-fréquence (*Time-Frequency Spatial Filtering* – TFS). La particularité de cette technique est qu'elle nécessite la DOA de la source cible. La technique estime la DOA du signal pour chaque catégorie temps-fréquence du spectrogramme, puis la compare avec la DOA de la source cible pour créer une matrice de probabilité. Plus la direction est similaire, plus la catégorie a une grande probabilité d'appartenir à la source cible. Ensuite, cette matrice est utilisée comme un masque sur la matrice de corrélation globale pour calculer la matrice de corrélation du signal cible. Finalement, le signal filtré est calculé avec un filtre de Wiener. Comparativement au BSS, cette méthode offre de meilleures performances sur des signaux avec un faible SNR. Elle a cependant besoin de la direction d'arrivée de la source. Dans un article subséquent [51], les auteurs ont amélioré la méthode de filtrage temps-fréquence en ajoutant une estimation du DOA. Ils estiment l'angle d'arrivée avec une fonction de vraisemblance spatiale en combinant dans un histogramme les angles de chacune des catégories temps-fréquence et un calcul statistique de Kurtosis. Les performances des deux méthodes diminuent rapidement lorsque les directions d'arrivée de la parole et du bruit sont similaires. Cette limitation a un impact important lorsque ce système est utilisé sur des drones, car la partie centrale du drone est entourée par les rotors. La figure 2.1 illustre la position des microphones sur leur drone. Cette position double la hauteur du drone, ce qui peut gêner ses déplacements.

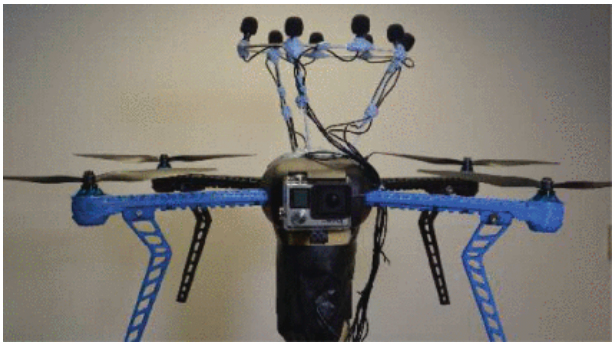

FIGURE 2.1 Emplacement des microphones sur le drone de Wang et Cavallaro [49]

2.2.4 Emplacement des microphones

Dans certains systèmes, l'emplacement des microphones affecte la performance du rehaussement de la parole et limite la possibilité de l'ajouter sur tous les robots. Il est préférable de concevoir un modèle qui s'adapte à la géométrie de microphones plutôt que l'inverse. Plusieurs travaux [58, 59, 8, 20] présentent des exemples de modèles de réduction d'égo-bruit qui nécessitent la modification du robot. Ils placent des capteurs piézoélectriques

ou des microphones proches des hélices et des moteurs et se servent de ces signaux pour estimer l'égo-bruit. Ils assument que ces capteurs/microphones captent seulement le bruit des moteurs en raison de leur proximité. Ils filtrent ensuite le signal du microphone central à l'aide du signal des références. Nakadai et coll. [30] ont proposé une méthode similaire avec un robot humanoïde. Le premier microphone est au niveau des oreilles du robot et le deuxième est dans la coque. Even et coll. [6] combinent cette méthode avec un filtre de Wiener. Mae et coll. [27] combinent cette méthode avec l'ILRMA et Ishimura et coll [18] avec un algorithme BSS basé sur l'analyse de vecteurs indépendants pour estimer l'égo-bruit à partir des microphones de références. Il n'est pas toujours possible d'ajouter des capteurs sur tous les moteurs. Certains robots n'ont pas l'espace ou sont équipés de trop de moteurs. Pour être modulaire, le système de réduction d'égo-bruit ne doit pas nécessiter la modification physique des robots.

De plus, l'égo-bruit ne provient pas uniquement des moteurs. Il peut provenir d'interférence de résidus électromagnétiques. Par exemple, l'un des quatre microphones du robot NAO [2] est situé près du câble WLAN. L'article de Taghia et coll. [44] illustre que l'égo-bruit sur le canal de ce microphone est difficile à réduire. Leur modèle basé sur le rehaussement adaptatif dans le domaine fréquentiel (*Frequency Domain Adaptive Line Enhance*) réussit à réduire considérablement l'égo-bruit sur les autres canaux. Le modèle de réduction d'égo-bruit doit s'adapter aux différents bruits présents sur le robot.

2.2.5 Combinaison audio-vidéo

Comme mentionné à la section 2.2.3, la technique TFS assume que la DOA de la cible est disponible. Pour obtenir cette donnée, Sanchez-Matilla, Wang et Cavallaro [35] ont conçu un système audio-vidéo. Leur système de vision capte la position de personnes dans l'image et fournit la DOA au système de réduction de l'égo-bruit TFS. Dans leur article, ils proposent une méthode de calibration qui permet d'aligner temporellement et géométriquement les deux systèmes. Cette technique est principalement limitée à des robots équipés de système de vision. De plus, il doit être en mesure de détecter physiquement la personne pour permettre au TFS de fonctionner. Dans ce type de technique, si la DOA est erronée, le signal de la source cible est théoriquement coupé du signal de sortie. Pour entrainer et tester les modèles audio-vidéos, Wang et coll. [54] ont bâti une base de données disponible publiquement [3]. Cette base de données utilise une matrice de microphones de huit canaux. Il contient des enregistrements de l'égo-bruit de drone et de conversation entre deux personnes. Les enregistrements de cette base de données sont intéressants pour

2. https ://www.softbankrobotics.com/emea/en/nao
3. http ://cis.eecs.qmul.ac.uk/projects/avq/

la présente recherche, car il permet de faire des tests et comparer les résultats avec leur système.

2.2.6 Combinaison de la séparation à l'aveugle et du filtrage temps-fréquence

Pour pallier au désavantage du BSS et du TFS, Wang et Cavallaro ont publié un autre article [52] où ils intègrent les avantages des deux techniques dans un même système. Pour séparer la parole et les autres sources de bruit, l'analyse des composantes interdépendante (*Independant Component Analysis* – ICA), utilisée originalement dans leur BSS, est plus performante que le TFS. Elle souffre cependant d'ambiguïté de permutation. L'utilisation du TFS permet donc de réaligner l'information de l'ICA dans le but de classifier l'information contenu dans la matrice de probabilité de l'ICA. Ensuite, un filtre de Wiener permet de séparer les différentes sources sur différents canaux tout en gardant le signal cible sur le premier canal. Les auteurs présentent leurs résultats sur leur site web. Les résultats montrent qu'il faut de longs segments audios pour réduire significativement l'ambiguïté de permutation, ce qui n'est pas possible pour être temps réel. Il y a aussi la présence de distorsion dans les signaux filtrés. De plus, l'algorithme doit être en mesure de connaitre précisément la direction de la source cible.

2.2.7 Combinaison d'un réseau de neurones et du filtrage temps-fréquence

Wang et Cavalero [53] comparent trois systèmes de réduction de l'égo-bruit intégrant un réseau de neurones profonds. Le premier système est un réseau de neurones entrainés à prédire un masque sur le spectrogramme d'entrée. Le réseau prédit le pourcentage d'appartenance à la parole de chaque catégorie temps-fréquence. Le deuxième système utilise le même principe, à l'exception que le signal obtenu est ensuite utilisé pour obtenir la matrice de covariance de la parole. Le système utilise un filtre de Wiener pour appliquer la matrice de covariance au signal original et ainsi isoler la parole. Le troisième système est une modification du deuxième en ajoutant le module TFS. Comme mentionné à la section 2.2.3, ce module calcule la DOA de chacune des catégories temps-fréquence du spectrogramme pour ensuite le comparer au DOA de la source cible et créer une matrice de probabilités d'appartenir à la parole. Le masque généré par le réseau de neurones est multiplié à cette matrice de probabilités. Les systèmes de rehaussement audios avec entrainement sont principalement limités à isoler seulement la parole. Tous les autres sons sont filtrés. L'objectif de la recherche proposée est l'inverse de ce type de système. Le but est de reconnaitre les caractéristiques de l'égo-bruit et de le retirer du signal. L'élément intéressant de cet article est au niveau de l'utilisation des masques pour des signaux à multiples canaux. Le

réseau de neurones prédit un masque pour chacun des canaux indépendamment, puis les combine avec la moyenne de chacun des éléments de la matrice. Cette technique n'est pas dépendante du nombre de microphones utilisés lors de l'entrainement du modèle. Elle est donc plus modulaire, ce qui est l'un des objectifs de la recherche proposée.

Les bases de données utilisées pour entrainer les modèles sont créées à partir de données de parole et d'égo-bruit enregistrées à l'intérieur et à l'extérieur. Dans leur article, ils ont testé l'entrainement sur six bases de données différentes, quatre de 60 heures et deux de 120 heures. Pour obtenir ces données, ils ont combiné des segments de parole à des segments d'égo-bruit. Au total, ils ont 6300 signaux de parole de moins de 8 secondes et 7 signaux d'égo-bruit de 40 à 214 secondes. En combinant les signaux, le nombre de données augmente rapidement. L'entrainement d'un modèle avec ce nombre de données demande beaucoup de ressources computationnelles, ce qui n'est pas toujours accessible. De plus, le modèle a accès à maximum 974 secondes de données d'égo-bruit, ce qui représente 0.4 % des données d'entrainement dans la petite base de données. L'entrainement de modèle sur des données de parole n'est pas une solution viable pour entrainer des modèles rapides à calibrer sur de nouveaux robots.

2.2.8 Réseau de neurones convolutifs

En vision par ordinateur, la conception de réseaux de neurones convolutionnels a permis d'obtenir des modèles performants en diminuant considérablement le nombre de paramètres nécessaires [23]. En audio, le spectrogramme ressemble à une image. Les catégories temps-fréquence peuvent avoir un lien avec les autres catégories autour. Tan et coll. [46] utilise cette technique pour entrainer un modèle de rehaussement de la parole sur un drone. Dans leur recherche, ils comparent trois techniques différentes : spectre de magnitude cible, spectre complexe cible et masque complexe à ratio optimal. Dans les deux premières techniques, le réseau de neurones prédit la magnitude du bruit dans le spectrogramme. Dans la troisième technique, la sortie du modèle est deux masques, pour les nombres réels et imaginaires, qui sont appliqués sur le spectrogramme d'entrée pour le filtrer.

Un article publié par Kim et coll. [21] illustre un système à multiples entrées et à multiples sorties intégrant les réseaux de neurones convolutifs. Le système utilise deux encodeurs, le premier pour le microphone de référence et le second pour les autres microphones de la matrice. Cela leur permet d'obtenir les caractéristiques du signal en entrée. L'idée d'utiliser deux encodeurs distincts est de permettre au système de garder la relation temporelle et spatiale entre les microphones. L'architecture du système est présenté à la figure 2.2. Le réseau convolutif du séparateur utilise les caractéristiques des encodeurs pour prédire un masque. Ce masque est multiplié aux caractéristiques des encodeurs pour permettre au

modèle de sélectionner seulement ceux qui sont importants pour la réduction de l'égo-bruit. Ensuite, un décodeur utilise ces nouvelles caractéristiques pour prédire le signal filtré. Cette suite d'actions est effectuée sur chacun des microphones pour obtenir en sortie le même nombre de canaux en entrée. Le principe d'avoir deux encodeurs en entrée est intéressant, car il permet au modèle de commencer l'apprentissage des caractéristiques sans devoir prendre en considération les autres canaux dès le début. Cependant, le deuxième encodeur dépend d'un nombre précis de microphones. Pour bien fonctionner, il faut réentrainer le réseau avec de nouvelles entrées. Comme mentionné à la section 2.2.4, l'emplacement des microphones est important pour obtenir des résultats en situation réelle. La figure 2.3 illustre la position des microphones sur le drone. En situation réelle, le drone serait déstabilisé. Dans la recherche proposée, un soin particulier est donné au choix de la position des microphones pour s'assurer que le modèle soit facilement utilisable sur des robots réels.

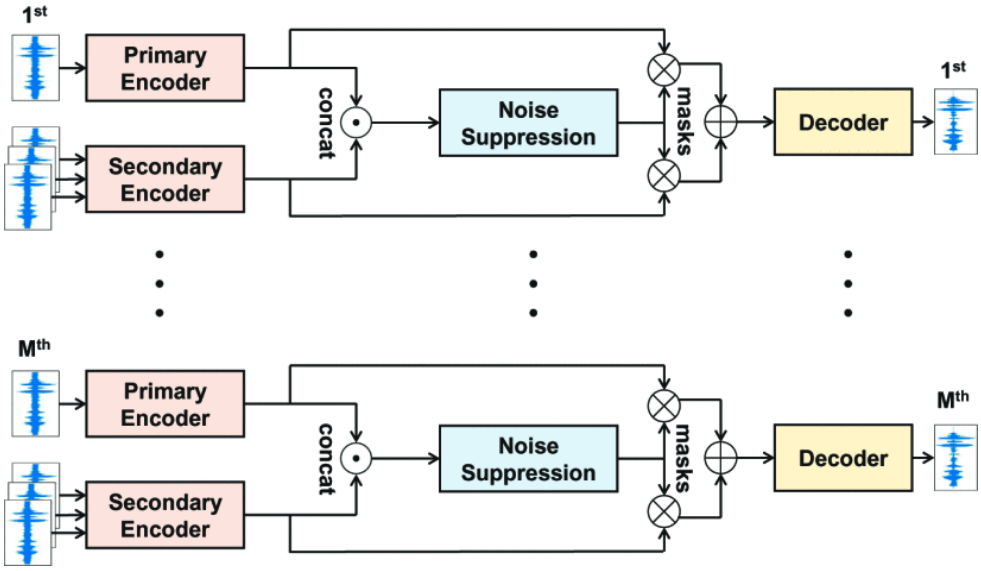

FIGURE 2.2 Architecture du système de Kim et coll. [21]

2.2.9 Réseau de neurones bidirectionnels avec mémoire à court et à long terme

Brieglebet et coll. [2] ont entrainé un réseau de neurones bidirectionnel avec mémoire à court et à long terme à prédire un vecteur de caractéristiques pour chacune des catégories temps/fréquences du spectrogramme. Tous les vecteurs de caractéristiques sont ensuite classifiés par l'algorithme k-moyenne pour connaitre à quelle classe (parole/bruit/silence) la catégorie temps-fréquence appartient. Finalement, si la catégorie appartient au bruit, elle est filtrée à l'aide d'un masque sur le spectrogramme original. Lorsque les données sont dépendantes du temps, les réseaux récurrents ont l'avantage de trouver des signatures qui prennent en considération cette relation. Ce réseau est intéressant pour la modélisation

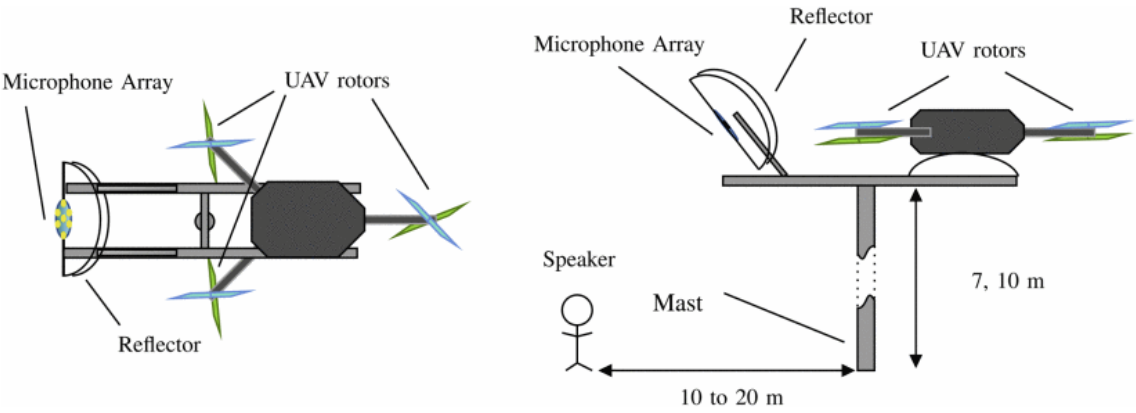

FIGURE 2.3 Emplacement des microphones sur le drone de Kim et coll. [21]

de l'égo-bruit, car les spectrogrammes sont temporels. La limitation de leur méthode est qu'elle classifie chacune des catégories temps-fréquence comme appartenant à une seule classe. Si l'interlocuteur parle à une fréquence très proche de l'égo-bruit, il n'est pas possible de la séparer.

2.2.10 Réseau antagoniste génératif de rehaussement de la parole

Spadini et coll. [42] proposent un système d'entrainement utilisé dans l'entrainement de réseau de neurones dans les jeux vidéos. Aux échecs par exemple, deux réseaux antagonistes génératifs compétitionnent entre eux pour gagner. En rehaussement de la parole, il y a deux réseaux de neurones, le premier (G) rehausse la parole, alors que le deuxième (D) classifie deux signaux pour reconnaitre s'ils proviennent du même signal ou si l'un d'eux provient de la prédiction du réseau G. Pour entrainer ces deux réseaux, il y a trois étapes : 1) Entrainer un réseau D à classifier deux mêmes signaux dans la classe 1 ; 2) Classifier l'entrée et la sortie du réseau G dans la classe 2 ; 3) Entrainer le réseau G à décevoir le réseau D pour qu'il se trompe de classe avec sa sortie. Ces trois étapes sont appliquées en boucle jusqu'à ce que les performances n'augmentent plus. Ensuite, seulement le réseau G est utilisé en temps réel pour rehausser la parole. Cette technique d'entrainement demande beaucoup de ressources computationnelles et nécessite d'être jumelée à un filtre qui estime l'amplitude logarithme spectrale avec une erreur quadratique moyenne minimum (*Minimum Mean-Square Error Log-Spectral Amplitude Estimator*) pour fonctionner à de bas SNR. Puisque le réseau G n'est pas entrainé directement à reconstruire la parole, les risques de surentrainement diminuent. Cependant, il existe d'autres méthodes, comme les couches d'abandons [43], qui peuvent permettre aux réseaux de neurones de ne pas surentrainer. Le premier réseau peut être entrainé à prédire directement le signal filtré.

2.2.11 Estimation de la puissance spectrale

Hioka et coll. [11] proposent un système de réduction de l'égo-bruit intégrant la densité spectrale de puissance (*Power Spectral Density* – (PSD)) à un algorithme MVDR et à un filtre de Wiener. Leur système est dépendant de l'emplacement et du type (omnidirectionnel et unidirectionnel) de chaque microphone sur le drone. Les microphones sont déportés à l'avant du drone et l'algorithme de faisceau est configuré pour ignorer tous les sons venant des rotors et des moteurs. Le PSD permet d'estimer un filtre de Wiener pour isoler le son sur chacun des canaux des microphones. Ensuite, Yen et coll. [55] proposent une extension au système pour ajouter un arbre de régression qui permet d'utiliser les données des moteurs pour améliorer l'estimation du filtre de Wiener avec le PSD. Dans un autre article, Yen et coll. [56] remplacent l'arbre de régression par un réseau de neurones profonds, ce qui améliore les résultats. Ils proposent ensuite une amélioration [57] en gardant les mêmes technologies, mais en ajoutant un deuxième filtre de Wiener à l'architecture. Ce système est dépendant des types de microphones et de leur configuration sur le drone. Ils doivent être à leur emplacement précis pour fonctionner, permettant ainsi à l'égo-bruit des rotors d'arriver dans la même direction. Il ne serait donc pas possible d'utiliser ce système sur d'autres plateformes robotiques. L'algorithme MVDR permet d'isoler un signal cible en ajoutant un minimum de distorsion, ce qui est important pour l'utiliser dans des modèles de reconnaissance vocale ou de détection d'évènements sonores. Cependant, la performance du filtre de Wiener diminue significativement lorsque le bruit est non-stationnaire. En robotique, le robot change de direction et de vitesse, ce qui va diminuer les performances du filtre.

2.3 Techniques de réduction de l'égo-bruit

Cette section présente des travaux permettant de cibler des caractéristiques de l'égo-bruit, puis de le réduire des signaux audios.

2.3.1 Approches basées sur les gabarits

Ince et coll. [13, 15] ont basé leur recherche sur l'utilisation de gabarits d'enregistrement de l'égo-bruit. Un gabarit est un enregistrement réel d'une séquence de données comme l'égo-bruit. Leur modèle commence par créer une base de données de gabarits et les assigne à des mouvements du robot. Durant la phase de réduction de l'égo-bruit, un gabarit est choisi à l'aide de l'algorithme des plus proches voisins sur les données des capteurs. Cela permet de trouver le mouvement le plus similaire dans la base de données. Le gabarit est ensuite soustrait au signal bruité. L'utilisation de paramètres dans la soustraction permet de choisir entre le niveau de réduction de l'égo-bruit et le niveau de distorsion dans le

signal résultant. La méthode est appelée soustraction de gabarit, mais conceptuellement le gabarit trouvé dans la base de données leur permet d'estimer un masque qui est ensuite multiplié point par point au spectrogramme bruité. Le gabarit sélectionné impacte significativement la performance du modèle. Il faut que celui-ci soit le plus proche du signal réel, ce qui demande une base de données de taille importante. Ils ont ensuite développé des techniques pour que leur modèle s'améliore en continu [16, 12]. Durant le fonctionnement du robot, le système génère de nouveaux gabarits s'il détecte seulement la présence d'égo-bruit dans les signaux audios et qu'aucun gabarit similaire est présent dans la base de données. Cela permet aux robots de s'adapter à des environnements inconnus. Malgré les limitations de cette technique, elle permet de prouver que les mêmes mouvements génèrent des signaux équivalents. Ils ont donc les mêmes caractéristiques.

D'autres travaux combinent l'utilisation des gabarits avec d'autres techniques. Ince et coll. [14] combinent les gabarits avec une technique de séparation de source appelée classification à multiple signal. Ensuite, dans un autre article [17], ils combinent les gabarits avec la séparation des sources géométriques et la théorie des caractéristiques manquantes (*Missing Feature Theory* – MFT). Nishumura et coll. [31] utilisent les gabarits en combinaison à la théorie à la MFT pour sélectionner des caractéristiques à utiliser dans un modèle de reconnaissance vocale. Malgré la combinaison à d'autres systèmes, ces systèmes nécessitent tout de même la synchronisation des données des capteurs aux signaux audios, ce qui n'est pas désirable sur tous les robots.

2.3.2 Approches basées sur les dictionnaires

La différence entre les dictionnaires et les gabarits de la section 2.3.1 est au niveau de l'information qu'ils contiennent. Un gabarit est exactement un enregistrement de l'égo-bruit, alors que les dictionnaires sont entrainés à reconnaitre des caractéristiques spécifiques et les distribuer sur ce qu'ils appellent des atomes. À l'aide d'un classificateur, il est possible de sélectionner les atomes qui représentent le mieux le signal d'entrée et de les additionner pour reconstruire l'égo-bruit. Un système de réduction d'égo-bruit utilisant les dictionnaires est proposé par Deleforge et Kellerman [4]. Ils entrainent le dictionnaire à reproduire le spectrogramme de l'égo-bruit à l'aide de l'algorithme de la décomposition de la valeur K singulière optimisée en phase. Durant l'inférence de l'égo-bruit, les atomes sont sélectionnés avec l'algorithme itératif de correspondance orthogonale optimisée en phase. Schmidt et coll. [38] ont ensuite modifié cette approche en utilisant l'algorithme factorisation de matrice non négative (*Non-Negative Matrix Factorization* – NMF) pour l'entrainement du dictionnaire et une machine à vecteur de support (*Support Vector Machine* – SVM) pour la classification des atomes. Le SVM est entrainé sur les données des

moteurs, ce qui permet de réduire la complexité du modèle car l'entrée contient beaucoup moins de données. Leurs résultats les ont encouragés à inclure ces données dans l'entrainement du dictionnaire [37]. Ils s'en servent comme régularisateur pour mieux distribuer les atomes sur les différents mouvements du robot. Cette nouvelle technique réduit le nombre d'atomes requis dans le dictionnaire pour atteindre les mêmes performances. Les techniques précédentes utilisent des équations non-linéaires pour la réduction de l'égo-bruit ce qui ajoute de la distortion dans le signal. Ils ont ensuite publié une nouvelle méthode [40] qui permet l'utilisation de leur technique avec des matrices de microphones et avec un filtre de Wiener. Ce filtre permet de réduire la distorsion non-linéaire dans le signal, mais le résultat peut être impacté en présence de bruit instationnaire. De plus, comme la section précédente, ces systèmes nécessitent les données de capteurs autres que les microphones pour fonctionner.

L'utilisation de dictionnaires fixes ne permet pas au modèle de s'adapter à des environnements inconnus. Les dictionnaires contiennent seulement de l'information par rapport aux environnements où les données d'entrainement ont été enregistrées. Dans le domaine de l'égo-bruit, Fang et coll. [7] ont publié un article présentant un modèle basé sur des dictionnaires partiellement adaptatifs, qui ne nécessite pas les données de capteurs autres que les microphones. La partie fixe du dictionnaire est entrainée sur l'égo-bruit, alors que la partie adaptative s'entraine en temps réel sur les bruits inconnus. Cette méthode permet de ne pas perdre l'information acquise durant l'entrainement, mais de toutefois continuer à évoluer pour mieux performer. Cette approche permet aussi d'utiliser des dictionnaires moins volumineux, car ils n'ont pas besoin d'avoir l'intégralité des exemples durant l'entrainement dû à l'adaptabilité du modèle. Cependant, leur système utilise un filtre de Wiener. Comme mentionné plus haut, ce filtre offre des performances réduites en présence de bruit instationnaire

2.3.3 Détection et suppression des harmoniques

Marmaroli et coll. [28] proposent une méthode de réduction de l'égo-bruit en détectant les harmoniques produits par moteurs. Ils ont développé un algorithme qui analyse l'ordre des harmoniques pour les atténuer. Schmidt et Kellermann [41, 39] ont aussi testé d'emmagasiner la structure harmonique de l'égo-bruit dans les dictionnaires car, selon eux, elle est déterministe. Le signal non bruité est ensuite reconstruit à partir des dictionnaires. Leurs résultats ont montré que cette technique permet au modèle de mieux performer lorsque les données d'entrainement et de tests ne sont pas bien équilibrées. La méthode des harmoniques est limitée à l'utilisation des moteurs. En robotique, ce ne sont pas toutes les sources d'égo-bruit qui produisent des harmoniques.

2.3.4 Factorisation de matrices non négatives

Comme présenté dans la section 2.3.2, certains utilisent le NMF pour entrainer un modèle basé sur les dictionnaires. Cette méthode est aussi utilisée pour d'autres types de systèmes de réduction de l'égo-bruit. Tezuka et coll. [47] utilisent la factorisation de matrices non négatives infinies partiellement à l'aveugle pour extraire une matrice d'activation et une matrice de caractéristiques de l'égo-bruit. Puis, lors de l'inférence, ils utilisent la matrice de caractéristique de l'égo-bruit pour estimer la matrice de caractéristiques du reste du son et deux matrices d'activation, pour l'égo-bruit et pour le reste du son. Les résultats illustrent qu'il n'est pas nécessaire d'extraire beaucoup de caractéristiques de l'égo-bruit pour le réduire. L'avantage de cette technique est qu'elle réduit l'égo-bruit linéairement, ce qui ajoute peu de distorsions dans le signal de sortie. Une limitation de leur résultat est que le SNR de départ n'est pas mentionné. Il est donc difficile d'évaluer la performance du système.

Haubner et coll. [10] utilise un NMF à plusieurs canaux (*Multichannel NMF* – MNMF) qui modélise, dans une première phase d'entrainement, la matrice de covariance de l'égo-bruit, puis, dans une deuxième phase, la matrice de covariance de la parole. Les données d'entrainement de la première phase contiennent seulement de l'égo-bruit pour extraire les caractéristiques de l'égo-bruit du robot. Celles de la deuxième phase contiennent de la parole mixée à l'égo-bruit pour extraire les caractéristiques de la parole en présence d'égo-bruit. Ce système doit donc être entrainé à reconnaitre la cible. Le système ne fonctionnera donc pas sur des sources inconnues.

Takusaki et coll. [45] combinent le MNMF à une contrainte spatiale de rang 1 sur un robot de secours en forme de tuyau pour réduire aveuglement, comme les techniques de BSS de la section 2.2.2, l'égo-bruit du robot. Cette technique a aussi des ambiguïtés de permutation lors de son fonctionnement.

2.3.5 Soustraction spectrale avec un réseau de neurones

Ito et coll. [19] ont conçu deux techniques utilisant la soustraction spectrale pour réduire le bruit dans les signaux. Dans leur premier système, ils assument que le spectre du bruit est stable durant un même mouvement et que le signal de tous les mouvements peut être obtenu durant l'entrainement du modèle. Durant les tests de leur modèle, ils ont constaté que ce ne sont pas tous les mouvements qui sont stables dans le temps. Les mouvements de marche de leur robot humanoïde sont instables et ne sont pas réduits par le modèle. Dans leur deuxième système, ils ont employé un réseau de neurones entrainés à prédire les cœfficients *filterbank* et la puissance du bruit à l'aide des données des moteurs. En assumant que l'égo-bruit est uniforme dans un même canal, ils reconstruisent l'égo-bruit

à partir des cœfficients et de la puissance pour ensuite le soustraire au signal original. Ce type de système est toutefois limité à l'utilisation de données des moteurs, ce qui n'est pas accessible sur tous les robots.

2.3.6 L'effet des données de capteurs sur la réduction de l'égo-bruit

La majorité des articles présentés précédemment utilisent les données de capteurs, par exemple la vitesse et l'accélération des moteurs ou des microphones de référence, pour améliorer les performances de leur système. Malgré l'amélioration des performances, l'ajout de ces données peut causer différents problèmes. Ceux-ci doivent être parfaitement synchronisés avec le signal audio d'entrée. Schillaci et coll. présentent dans leur article [36] l'effet d'utiliser des données incohérentes lors de l'inférence d'un modèle de réduction de l'égo-bruit. Une mauvaise prédiction peut venir détériorer davantage la qualité du signal audio.

CHAPITRE 3

MÉTHODE DE RÉDUCTION DE L'ÉGO-BRUIT DES ROBOTS

AVANTS PROPOS

Auteurs et affiliation :

Pierre-Olivier Lagacé : étudiant à la maîtrise, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique.

François Ferland : professeur, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique.

François Grondin : professeur, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique.

Date de soumission : 28 février 2023

Conférence : *IEEE/RSJ International Conference on Intelligent Robots and Systems*

Titre français :

Réduction de l'égo-bruit d'un robot en utilisant l'apprentissage des matrices de covariance et l'algorithme de réponses à variance minimums sans distorsion.

Contribution au document :

À la suite de la revue de littérature présentée au chapitre 2, un système de réduction de l'égo-bruit fut conçu pour répondre à la question de recherche. Cet article situe le projet de recherche dans la littérature et détaille le système proposé dans ce mémoire. Il propose une méthodologie, ainsi que tous les résultats obtenus pour évaluer les performances de réduction de l'égo-bruit du système.

Résumé français :

La performance des systèmes de reconnaissance de la parole et d'évènements sonores a particulièrement augmenté grâce aux méthodes d'apprentissage profond. Cependant, quelques-unes de ces tâches restent difficiles quand les algorithmes sont déployés sur des robots dus aux bruits mécaniques des actionneurs et des interférences électriques qui sont absentes lors de l'entrainement des réseaux de neurones. Un système de réduction de l'égo-bruit comme étape de prétraitement peut résoudre cette problématique lors de l'utilisation de modèle préentrainé de reconnaissance vocale et d'évènements sonores sur des robots. Dans cet article, nous proposons une nouvelle méthode pour réduire l'égo-bruit utilisant seulement une matrice de microphones et moins de deux minutes d'enregistrement de bruit. En utilisant l'analyse de composante principale (*Principal Component Analysis* – PCA), la meilleure matrice de covariance est sélectionnée parmi un dictionnaire créé en ligne durant la calibration et est utilisée avec l'algorithme de formation de faisceau de réponses à variance minimums sans distorsion (*Minimum Variance Distortionless Response* – MVDR). Les résultats montrent que la méthode proposée fonctionne en temps réel, améliore le ratio signal-distorsion (*Signal-to-Noise Ration* – SDR) jusqu'à 10 dB, réduit le taux d'erreur de mots (*Word Error Rate* – WER) de 55% dans certains cas et améliore la précision moyenne (*Average Precision* – AP) de la détection d'évènements sonore jusqu'à 0.2).

3.1 Abstract

The performance of speech and event recognition systems significantly improved recently thanks to deep learning methods. However, some of these tasks remain challenging when algorithms are deployed on robots due to the unseen mechanical noise and electrical interference generated by their actuators while training the neural networks. Ego-noise reduction, as a preprocessing step, can help solve this issue when using pre-trained speech and event recognition algorithms on robots. In this paper, we propose a new method to reduce ego-noise using only a microphone array and less than two minute of noise recordings. Using Principal Component Analysis (PCA), the best covariance matrix candidate is selected from a dictionary created online during calibration and used with the Minimum Variance Distortionless Response (MVDR) beamformer. Results show that the proposed method runs in real-time, improves the signal-to-distortion ratio (SDR) by up to 10 dB, decreases the word error rate (WER) by 55% in some cases and increases the Average Precision (AP) of event detection by up to 0.2.

3.2 Introduction

In human-machine interaction, the audio modality plays a central role in speech and sound-event recognition. In the last few years, robust speech and event recognition based on transformers have been proposed (e.g., Whisper [33], Google Speech-to-Text [3] and PANNS [22]). Although these systems perform well with ambient noise, their performance drops quickly with unusual noises and low Signal-to-Noise Ratio (SNR). When microphones are mounted on a robot, the audio signal quality decreases considerably due to noise produced by the motors and WiFi/Bluetooth antennas. This makes denoising challenging, as the noise generated by the robot, referred to as ego-noise, is usually non-stationary.

Numerous approaches have been proposed to reduce the ego-noise for Unmanned Ground Vehicles (UGVs) and Unmanned Aerial Vehicles (UAVs). Time-Frequency Spatial filtering (TFS) [50, 51] uses the direction of arrival (DOA) of the target sound source at each time-frequency bin to mask interfering noise. Since it relies on spatial cues, performance decreases when the ego-noise and target sound are close. On the other hand, Blind Source Separation (BSS) [48, 49] extracts each sound source individually using Independent Component Analysis (ICA) and permutation alignment. BSS works efficiently with low Signal-to-Noise Ratios (SNRs), but suffers from the permutation ambiguity. A TFS-BSS approach [52] is proposed to solve the permutation ambiguity, but it relies on long audio segments, which makes it unsuitable for real-time applications.

Another strategy consists of using a reference signal captured by a microphone [20, 58, 59] or a piezoelectric sensor [8] close to the motors. This assumes the target sound is not captured by the reference microphone. However, this depends on the robot configuration, which makes it difficult to generalize to any robots. Other methods rely on a database that contains noise templates and motor inertial data to select the best noise instance that fits the observed actuator profile [12, 13, 14, 16]. However, this requires synchronization between audio and inertial signals. A similar method consists of using a audio only dictionary-based approach. The dictionary is trained to represent the multiple components of the ego-noise, and then used to reconstruct and suppress the ego-noise from the noisy signal. Non-Negative Matrix Factorization (NMF) [39, 37, 40] and phase optimized singular value decomposition (PO-KSVD) [4] can be used to train the dictionary. The motors state are used to reconstruct the ego-noise from the dictionary components. Some NMF approaches do not use motors state in their framework [47, 10], which can introduce non-linear distortion and hinders speech recognition performance. On the other hand, Fang et al. [7] use linear operation that does not introduce distortion, but performs poorly on non-stationary noise. Deep-learning based methods can also be used to separate target speech from ego-noise [19, 2, 21, 46, 53]. These models perform well when trained on large datasets that contain ego-noise and target sounds. However, these trained networks usually perform poorly in a different unseen audio scene, which is common in robotics.

In this paper, a template-based framework using ego-noise spatial covariance matrix (SCM) with minimum variance distortionless response (MVDR) algorithm is presented. The framework improves SDR while being robust to non-stationnary noise and can be quickly calibrate in novel audio domains. This is appealing as it avoids retraining deep neural network, which usually requires hours of audio samples and computing resources not available on a robot. A calibration step is used to generate a dictionary of ego-noise SCMs. Principal Component Analysis (PCA) is used to reduce the number of dimensions of the SCM. This compact representation is used to find the best fit between the SCMs in the database and the measured SCM. The corresponding SCM is used in the MVDR algorithm to separate the target sound source from the ego-noise. Results show that this method increases both the performance of speech and event recognition.

This paper is organized as follows. Section 3.3 presents the overall ego-noise reduction method. Section 3.4 describes the experimental setup, followed by Section 3.5 with the results obtained with real-recorded data and Section V with the conclusion of this paper.

3.3 MVDR using SCM Estimation

Figure 3.1 presents the proposed framework. The multi-channel target sound, ego-noise and audio mixture for microphones $m \in \{1, \dots, M\}$, where $M \in \mathbb{N}$ stands for the number of microphones, are denoted by $\mathbf{s}[n] \in \mathbb{R}^M$, $\mathbf{b}[n] \in \mathbb{R}^M$ and $\mathbf{x}[n] \in \mathbb{R}^M$, respectively, where $n \in \mathbb{N}$ corresponds to the sample index. It is assumed that noise is purely additive, such that :

$$\mathbf{x}[n] = \mathbf{s}[n] + \mathbf{b}[n]. \tag{3.1}$$

These signals can be represented in the frequency domain using a Short-Time Fourier Transform (STFT) with frame size of $N \in \mathbb{N}$ samples and hop size of $\Delta N \in \mathbb{N}$ samples, and are denoted as $\mathbf{X}[k, l] \in \mathbb{C}$, $\mathbf{S}[k, l] \in \mathbb{C}$ and $\mathbf{B}[k, l] \in \mathbb{C}$, where $k \in \{0, 1, \dots, K-1\}$ and $l \in \mathbb{N}$ represent the frequency bin and frame indices, respectively. The spatial covariance matrices of the input signal $\Phi_{\mathbf{XX}}[k]$ is computed over L frames as follows :

$$\Phi_{\mathbf{XX}}[k] = \frac{1}{L} \sum_{l=1}^{L} \mathbf{X}[k, l]\mathbf{X}[k, l]^H, \tag{3.2}$$

where $\{\dots\}^H$ stands for the Hermitian operator.

The calibration stage begins by recording the robot's ego-noise during a few minutes, more specifically 90 sec of ego-noise data with our framework. The ego-noise recording is separated in $J \in \mathbb{N}$ segments of 0.5 sec, and the SCMs are computed for each segment j using (3.2). Since the SCM is hermitian, its upper triangle (including the diagonal) can be flattened to create a vector $\mathbf{v}_j[k] \in \mathbb{C}^{M(M+1)/2}$ for each frequency bin k that holds all relevant elements. These vectors are concatenated in a supervector $\mathbf{V}_j \in \mathbb{C}^{KM(M+1)/2}$ for all frequency bins. Each supervector can be reduced to a dense vector using Principal Component Analysis (PCA), and denoted $\mathbf{D}_j \in \mathbb{C}^I$.

At test time, the same PCA transformation is applied to the supervector obtained from the SCMs for noisy audio segments of 0.5 sec, and the result is denoted as $\hat{\mathbf{D}} \in \mathbb{C}^I$. The l^2-norm is computed $\hat{\mathbf{D}}$ and vectors $\mathbf{D}_j$ from the dictionary. The closest dictionary vector $\mathbf{D}_{j^*}$ is then selected :

$$j^* = \arg\min_j (\|\hat{\mathbf{D}} - \mathbf{D}_j\|_2^2). \tag{3.3}$$

Dimensionality reduction with a PCA serves two purposes : 1) reduction of the memory footprint and number of computations ; 2) projection of the supervector observation on the

noise subspace, which eases comparison between the noisy SCMs (target and noise) and the noise SCMs. PCA decomposition can be computed efficiently on a low-power embedded computer, which makes dictionary generation on a robot fast once the calibration audio signals are recorded. The corresponding noise SCM estimate $\hat{\Phi}_{\mathbf{BB}}[k]$ is used to obtain the MVDR weights :

$$\mathbf{w}[k] = \frac{\hat{\Phi}^{-1}_{\mathbf{BB}}[k]\Phi_{\mathbf{XX}}[k]}{\mathrm{Tr}\{\hat{\Phi}^{-1}_{\mathbf{BB}}[k]\Phi_{\mathbf{XX}}[k]\}}\mathbf{u}, \tag{3.4}$$

where the inverse of each SCM noisy matrix (denoted as $\hat{\Phi}^{-1}_{\mathbf{BB}}[k]$) can be precomputed once during calibration and directly stored in the dictionary to speed up computations. The one-hot reference vector $\mathbf{u} \in \{0,1\}^M$ is chosen to select the microphone with the highest SNR, which can be estimated from the selected noise covariance matrix. Beamforming is finally applied and generates the enhanced signal :

$$Y[k,l] = \mathbf{w}[k]^H \mathbf{X}[k,l]. \tag{3.5}$$

The inverse STFT (iSTFT) can then be used to transform back the signal to the time domain and obtain $y[n]$.

3.4 Experimental Setup

Figure 3.2 shows the small Clearpath Robotics Jackal UGV [4] equipped with a 16SoundsUSB [5] MA used in this experiment. The 16 omnidirectionnal microphones are positioned around the robot frame. Half of them are 3 cm higher than the others, to ensure spatial discrimination in the z-axis. The MA is connected by USB to a laptop computer which runs the framework implemented on Python and ROS [6]. Audio is sampled at 32000 samples/sec and divided in segments of 0.5 sec. The STFT uses a Hann window of $N = 2048$ samples and a hop size of $\Delta N = 256$ samples.

The experiments are conducted in three different rooms : a small office room of 30 m^2, a large room of 150 m^2 and a hallway of 30 m^2. The reverberation time RT_{60} are respectively 200 msec, 600 msec and 160 msec, and estimated from recorded hand claps in each room.

For evaluation purposes, ego-noise and speech are recorded separately. Ego-noise is recorded in each room while the robot moves in the room at different speeds (up to 0.4 m/sec), and in arbitrary directions. Figure 3.2 shows the robot moving in the large room. While

4. https ://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/
5. https ://github.com/introlab/16SoundsUSB
6. https ://github.com/introlab/egonoise/

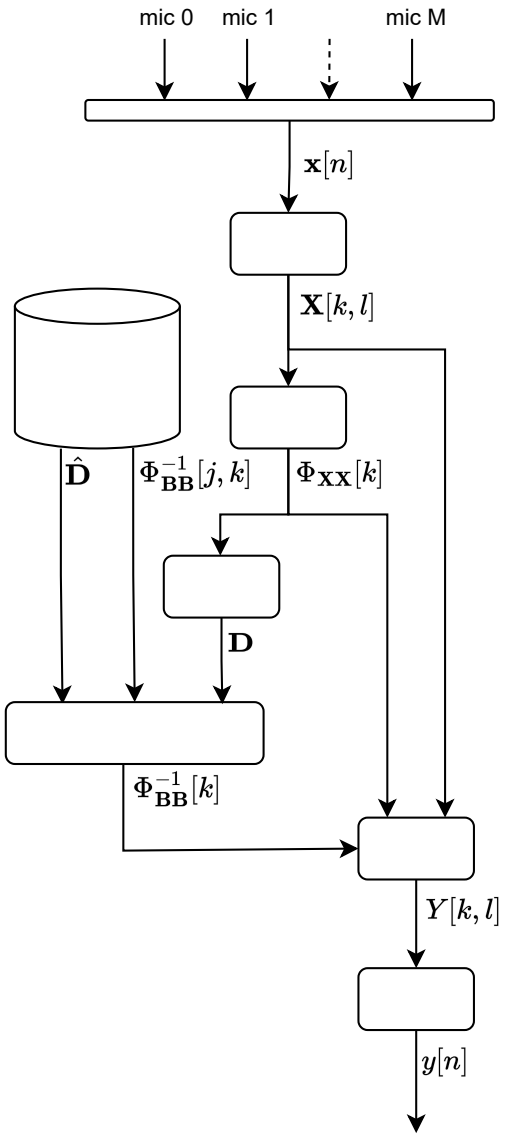

FIGURE 3.1 Block diagram of the proposed framework.

recording ego-noise, no other sound sources are active around the robot except ambient noises. A loudspeaker then plays target sound sources, with the volume adjusted to a casual conversation level. The loudspeaker is moved manually around the robot at a distance ranging from 1 and 3 meters to recreate relative motion between the robot and the target sound source. The target sound sources consist of speech, music, screaming, alarms and door slams. A total of 36 audio files from the LibriSpeech [32] dataset are used, from 3 male and 3 female speakers. These speech clips vary in duration between 6 and 15 sec. The loudspeaker also plays music (5 sec), screaming (5 sec) and alarms sounds (3 sec). Sound from the slamming door comes from the actual door in the room. A total of 90 sec of ego-noise is recorded for calibration, and another 120 sec is recorded and mixed

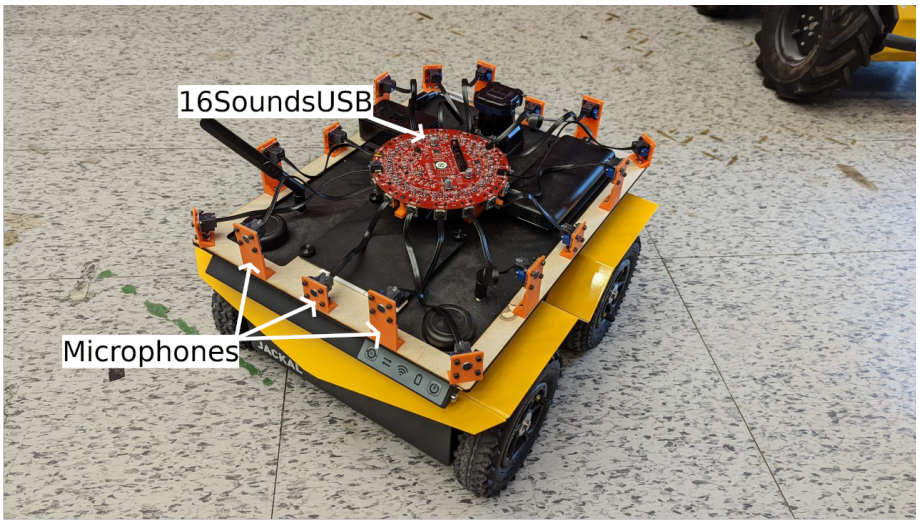

FIGURE 3.2 Clearpaths Jackal Robot equipped with the microphone array.

randomly with target sound source recordings for validation purposes. This provides 288 noisy speech clips and 120 noisy sound events for each room.

3.5 Results and Discussion

The metrics used to analyse the performance of the proposed framework is the Signal-to-Noise Ratio (SNR), the Signal-to-Distortion Ratio (SDR), the Word Rate Error [7] (WER) for speech recognition and the Average Precision [8] (AP) of classification for events detection. For Speech recognition, the small English model from Whisper is used. For sound events classification, the Wavegram Logmel Cnn14 model from PANNS is used. For each metric, the enhanced signal from the framework is compared to the input and voice only signals. The input and voice only signals are selected from one microphone in the MA which shows on average a superior SNR value compared to some other microphones closer to the actuators.

3.5.1 Speech Recognition

Figure 3.4 shows an example of the input, filtered and voice only spectrograms in the large room with an input SNR of -6.09 dB. In the input spectrogram, the speech is almost entirely masked by the ego-noise. The filtered signal using the proposed framework has a SNR of 6.31 dB and numerous speech features now show on the spectrogram. A Dell XPS laptop can process a 0.5 sec audio segment in 0.2 sec in the Python environment using Numpy [9], which confirms that the framework can perform ego-noise reduction in real time.

7. https ://torchmetrics.readthedocs.io
8. https ://scikit-learn.org
9. https ://numpy.org

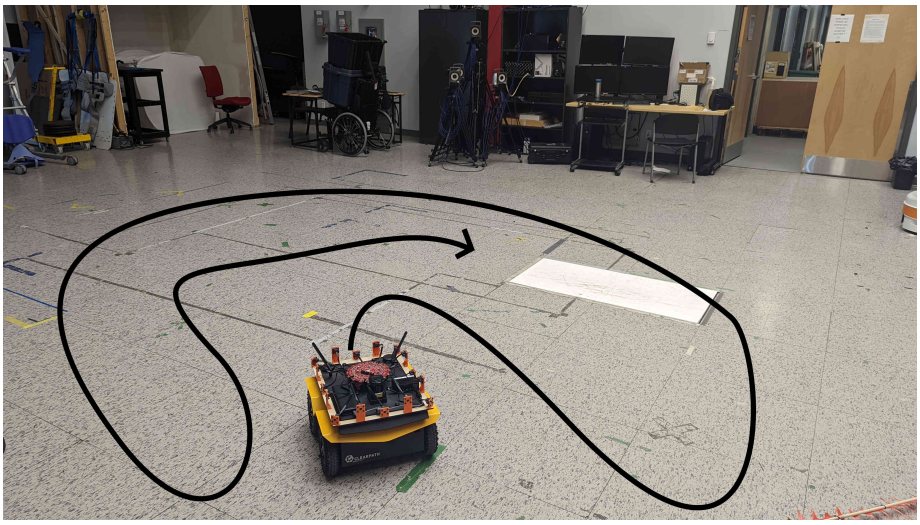

FIGURE 3.3 Robot moving in the large room.

Table 3.1 presents the average SNRs and SDRs for each room for speech segments. For these metrics, only segments containing the target sounds are used and silence segments are discarted.

Figure 3.5 and Figure 3.6 show scatter plots of the input and filtered SNRs and SDRs for all 0.5 sec segments. The results shows that the enhanced signal have a better SNR and SDR compared to the input signal. Results show linear performance, except at low SNRs (around -5 and -10 dB), where the weak power of the target sounds in the audio signal can cause deterioration of the MVDR performance.

Table 3.2 shows the average WER of the input, enhanced and voice only signal. As expected, the WER of the input signal in all three rooms is higher compared of the voice-only signal. The proposed framework reduces considerably the WER by 30% to 55%. Input signals recorded in the small room and the hallway show a higher SNR, resulting in a better WER results with the input signal.

TABLEAU 3.1 Average input and enhanced SNRs and SDRs for each room for speech segments

Room	**SNR (dB)**		**SDR (dB)**	
	Input	**Enhanced**	**Input**	**Enhanced**
Small	1.5	11.29	2.14	11.87
Large	-2.69	8.7	-1.75	9.57
Hallway	1.6	12.06	2.28	12.72

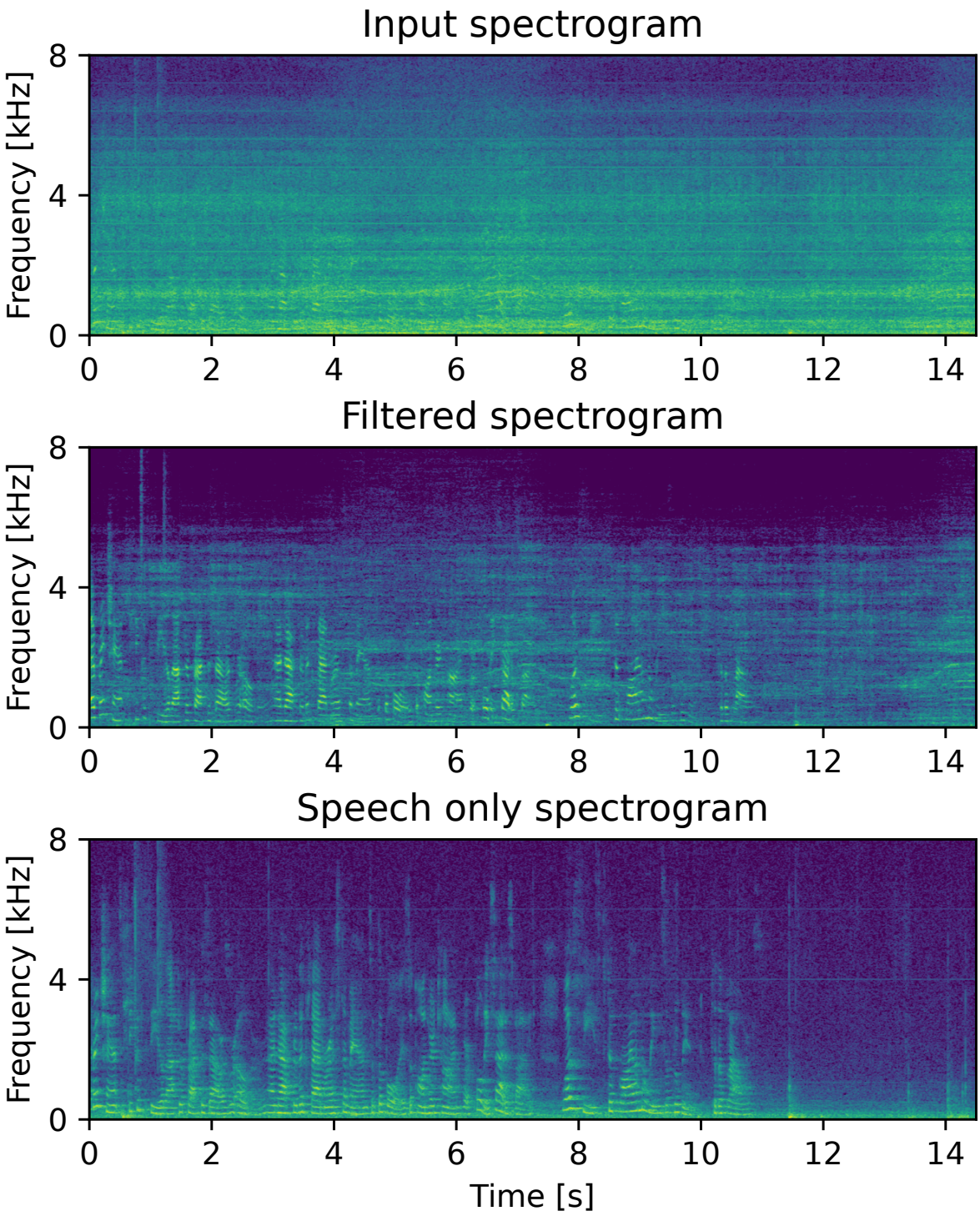

FIGURE 3.4 Spectrograms on the input, filtered and voice only signal. The SNR of the input and filered signal are -6.09 dB and 6.31 dB, respectively, which means an improvement of 12.4 dB.

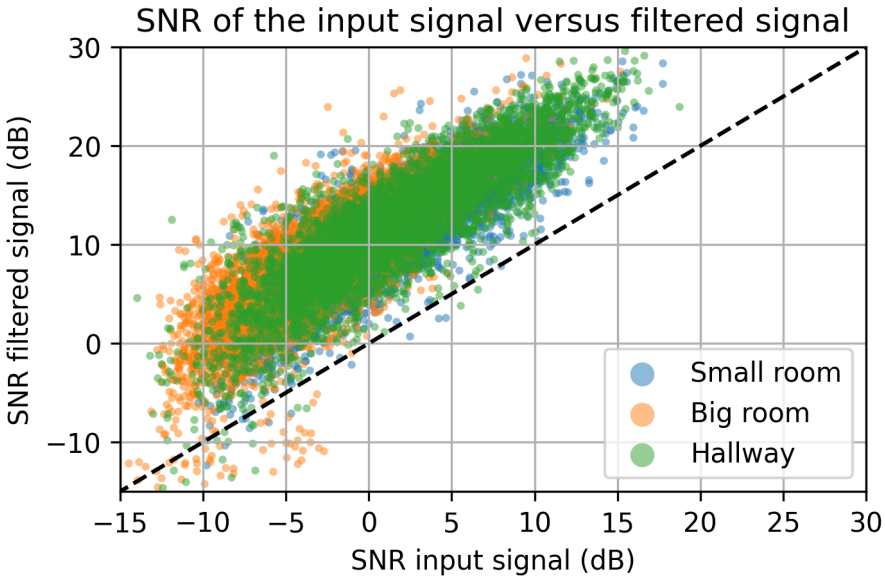

FIGURE 3.5 SNRs before and after filtering on all 0.5 sec segments.

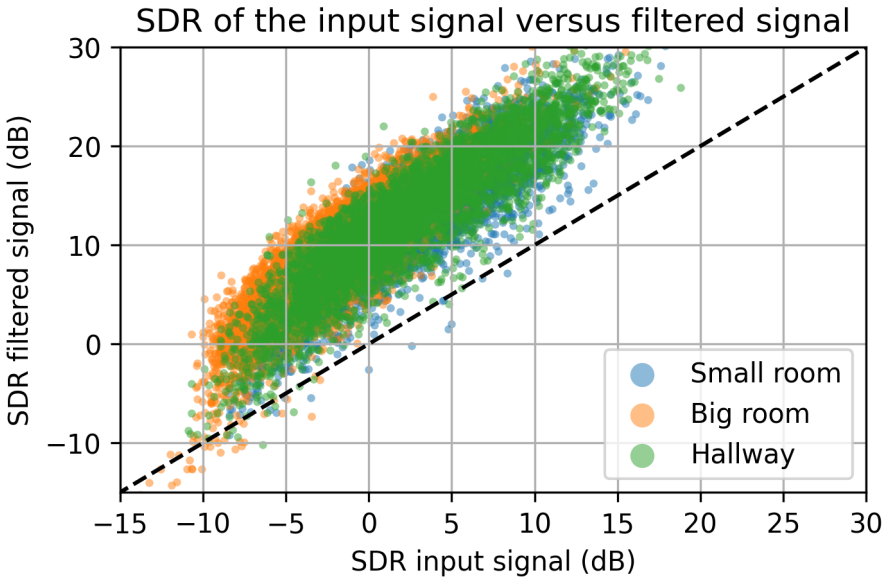

FIGURE 3.6 SDRs before and after filtering on all 0.5 sec segments.

TABLEAU 3.2 Word error rate (WER) with the proposed framework

Room	**Input**	**Enhanced**	**Voice only**
Small	62.6%	30.9%	11.1%
Large	90.3%	35.5%	14.9%
Hallway	62.4%	25.3%	8.52%

3.5.2 Sound Event Detection

Figure 3.7 shows an example of the input, filtered and music-only spectrograms in a large room with SNRs of -1.93 dB for the input signal and 13.13 dB for the filtered signal. While music is mainly masked by ego-noise in the input signal spectrogram, harmonics become visible in the enhanced signal spectrogram.

Figure 3.8 and Figure 3.9 show the input and filtered SNR and SDR for all the events in the large room. Because results are similar in all three rooms, we focus on the large room for clarity. While the difference between each event is mainly the SNR of the input signal, the system improves the sound quality in most scenarios.

The PANNS model is used to classify the sound events using the input, enhanced and event only signals. Results shown in Table 3.3 suggest that the enhanced signal increase the AP on music signal. Some results show a smaller AP in event only signal than noisy signal and even bigger with the enhanced signal. Results are mainly better than observed in [22]. Unlike in Audioset, data recorded in this experiment is clear and isolate from other sound. Mitigated performances are however observed with the door slam sound event. We believe this to be be due to the short and faint sounds of a door closing, which can be more difficult to capture when computing the input signal spatial covariance matrix. While the alarm sound event remains difficult to detect with PANNS, the framework still provide some slight increase for the AP. It is believed that the framework is more beneficial to audio segment with lower SNR values.

3.6 Conclusion

This paper introduce a framework to reduce the ego-noise of a robot using a microphones array. The framework relies on a dictionary of noise SCMs trained during a short calibration period and the MVDR algorithms to enhanced the target sounds sources. For better performance, the SCMs are concatenated in a supervector, which dimensions are reduced offline using the PCA algorithm. Results obtained with a UGV and a 16-microphone array show an improvement of approximately 10 dB on average for the SNRs and SDRs in the enhanced signal using only 90 sec of calibration time. Moreover, once the enhanced speech signal is fed to a automatic recognition system, the word error rate decreases by to 55% compared to the performance obtained using the noisy input signals. This framework also improves average precision of sound event detection performance, by up to 0.2 in some cases. This new framework brings new opportunities to integrate speech recognition and sound sources detection on robots that generate unseen ego-noise at training time for deep neural network models.

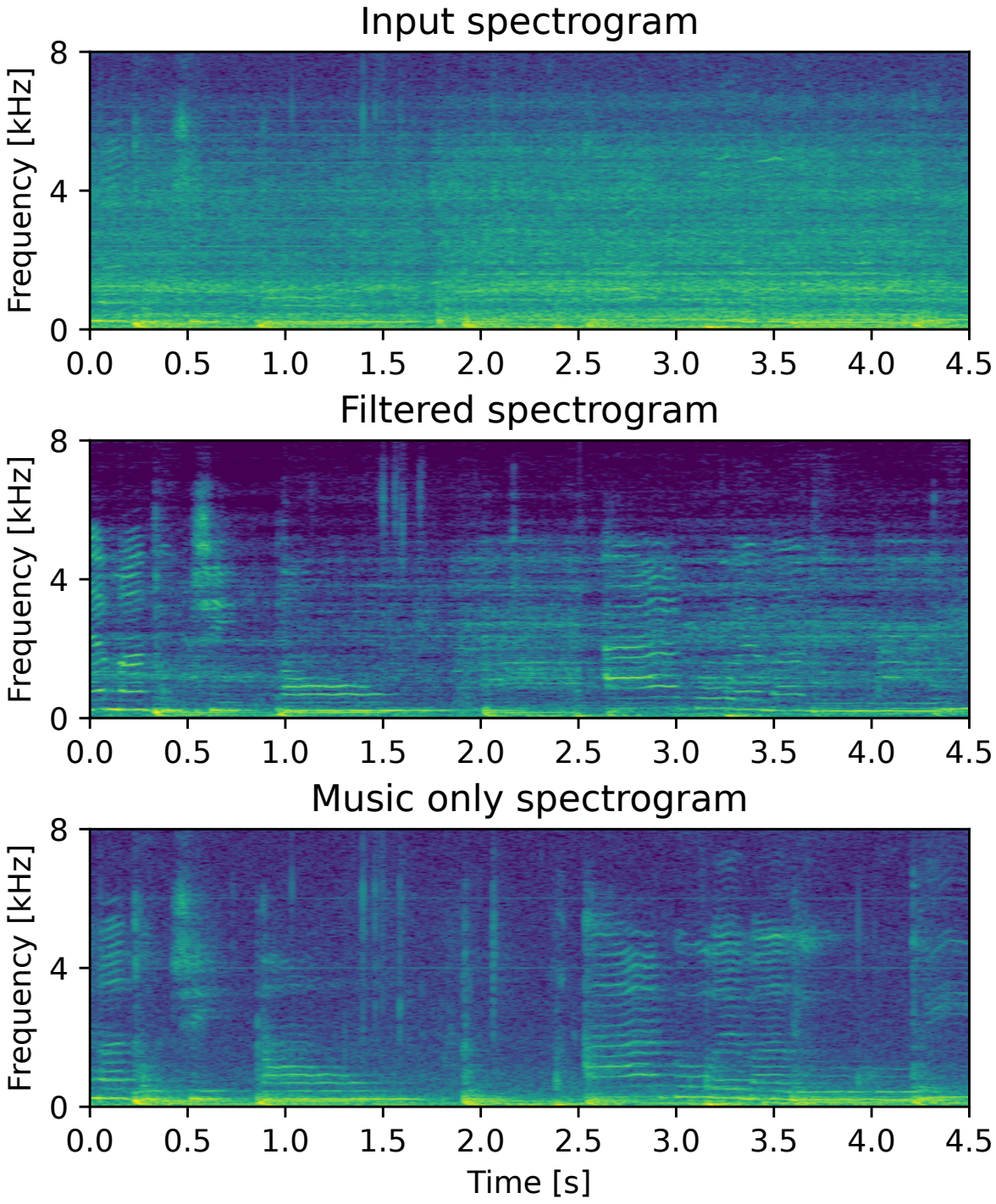

FIGURE 3.7 Spectrograms on the input, filtered and music only signal. The SNR of the input and filered signal are -1.93 dB and 13.13 dB, respectively.

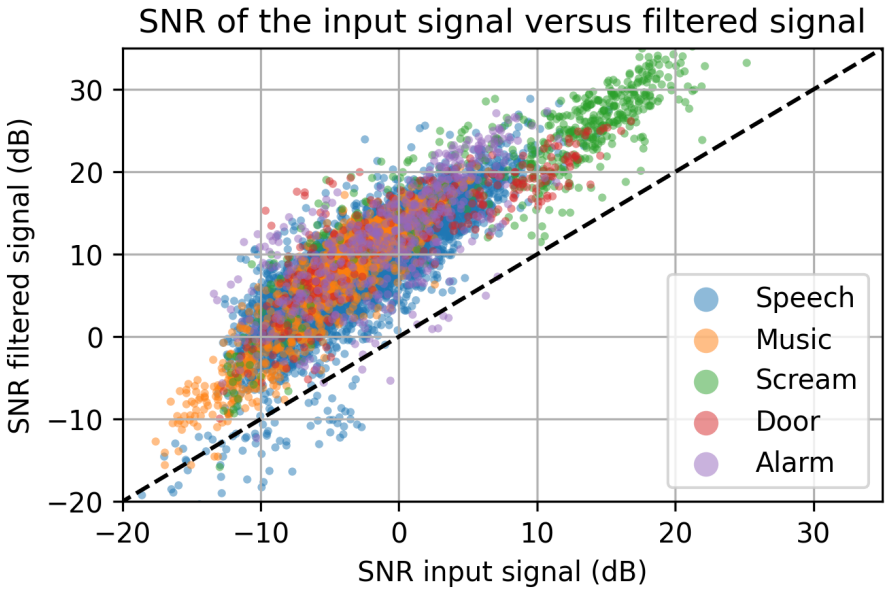

FIGURE 3.8 SNRs before and after filtering for each sound event type.

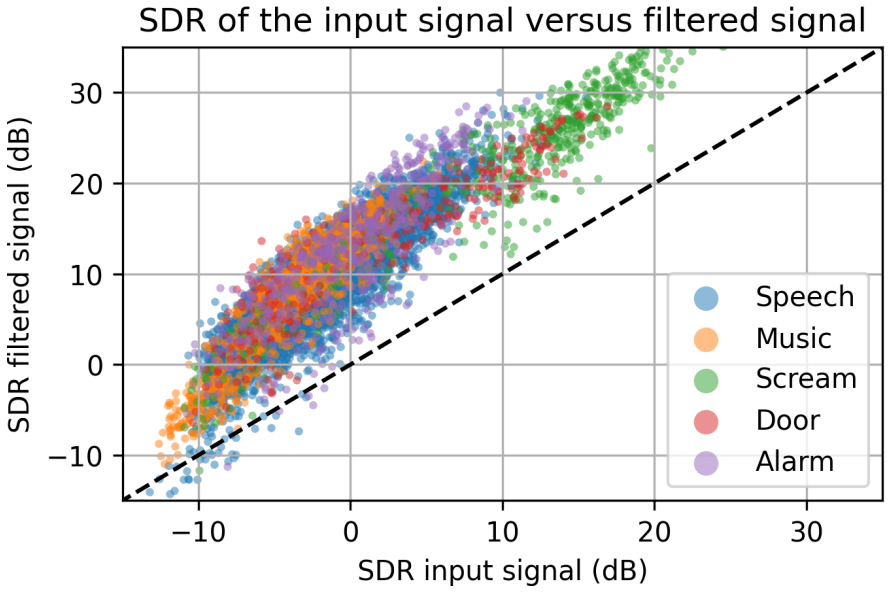

FIGURE 3.9 SDRs before and after filtering for each sound event type.

TABLEAU 3.3 Average precision of event detection with the proposed framework

Event	Room	Input	Enhanced	Event only
Speech	Small	0.999	1.0	1.0
	Large	0.994	1.0	1.0
	Hallway	0.999	1.0	1.0
Music	Small	0.722	0.792	0.914
	Large	0.487	0.685	0.784
	Hallway	0.868	0.953	0.948
Scream	Small	0.993	0.994	0.992
	Large	0.998	0.998	1.0
	Hallway	0.994	1.0	1.0
Door	Small	0.313	0.232	0.104
	Large	0.178	0.109	0.091
	Hallway	0.129	0.100	0.089
Alarm	Small	0.091	0.105	0.108
	Large	0.091	0.093	0.096
	Hallway	0.090	0.098	0.105

Acknowledgment

The authors would like to thank Prof. Alexandre Girard for providing access to a Clearpath Robotics Jackal robot, and Samuel Faucher for the insightful discussions.

CHAPITRE 4

ANALYSE DES PARAMÈTRES

Ce chapitre présente une analyse des trois principaux paramètres du système : la quantité d'exemples dans la base de données, la longueur des segments audios et la taille de la STFT dans le spectrogramme. La figure 4.1 introduit chacun des paramètres. Ceux-ci sont expliqués dans leur section respective. Pour les trois analyses, seul le paramètre ciblé est modifié. Les paramètres de base de l'article sont utilisés, soit 90 secondes de temps de calibration, une longueur de segment de 0.5 seconde et une taille de STFT dans le spectrogramme de 2048. Puisque les résultats présentés au chapitre 2.3.6 sont similaires dans les trois pièces, seulement les données dans la grande pièce sont utilisées. De plus, seulement la parole est utilisée comme signal cible afin de simplifier l'analyse. Les différents paramètres sont comparés à l'aide de l'augmentation SNR et SDR du signal résultant.

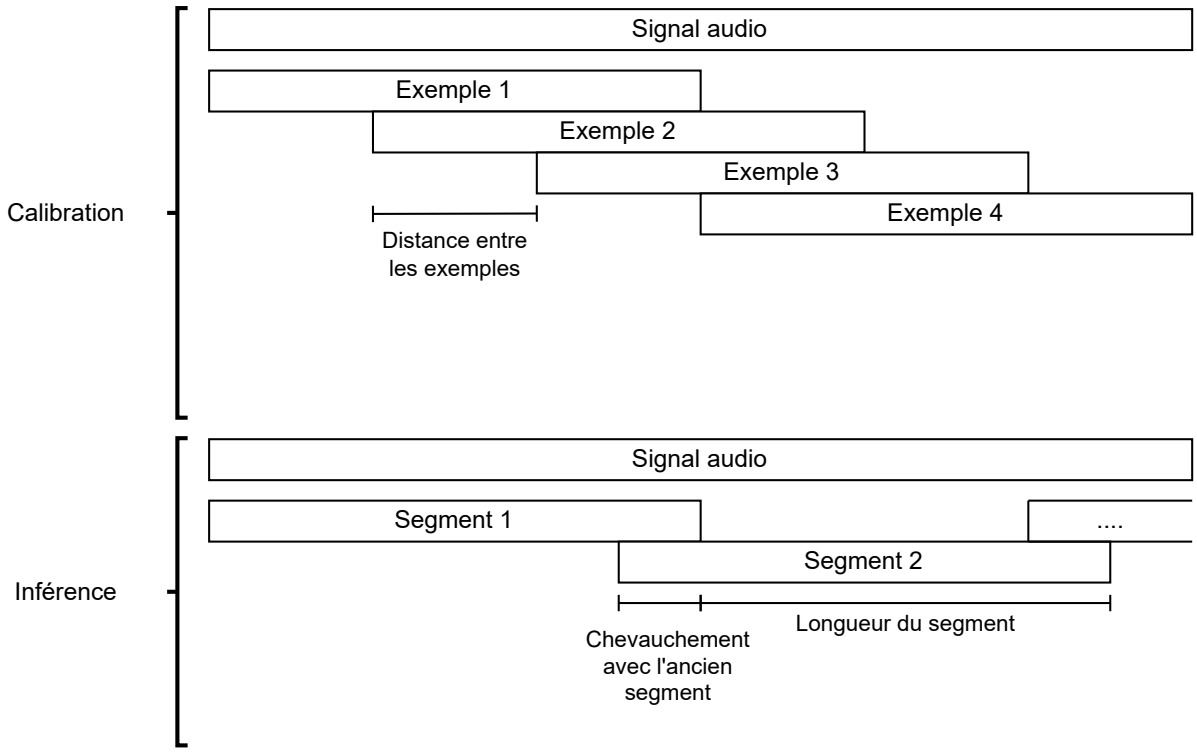

FIGURE 4.1 Représentation graphique des longueurs des segments et du chevauchement durant la calibration et l'inférence du système.

4.1 Quantité d'exemples dans la base de données

La quantité d'exemples dans la base de données dépend de deux éléments : la distance entre deux exemples et le temps de calibration. Pour un même temps de calibration, une petite distance extrait plus d'exemples du signal. Ils sont plus similaires, mais ils représentent aussi un moment plus précis de l'égo-bruit du robot. Quant à lui, si le temps de calibration est trop faible, le robot n'a pas le temps de se déplacer à tous les endroits dans la pièce et à toutes les vitesses. Pour profiter le plus possible du temps de calibration, il est préférable d'utiliser une faible distance et d'augmenter le temps de calibration au besoin. Dans ce mémoire, une distance de 0.0625 seconde est utilisée. Le temps de calibration varie dans le tableau 4.1 pour évaluer l'impact de la quantité d'exemples sur les performances du système. Les résultats montrent que le temps de calibration a un impact positif sur le SNR et sur le SDR. Il est donc utile de calibrer le robot plus longtemps pour obtenir plus d'exemples d'égo-bruit dans la base de données. Puisque la méthode de sélection utilise la réduction des dimensions PCA, la quantité d'exemples dans le dictionnaire n'a pas un impact significatif sur le temps de traitement du système.

Tableau 4.1 SNR et SDR pour différents temps de calibration

Temps de calibration (s)	**Quantité d'exemples**	**Augmentation SNR (dB)**	**Augmentation SDR (dB)**
15	221	10.0	9.8
30	450	10.71	10.5
60	908	11.05	10.83
90	1326	11.4	11.32

4.2 Longueur de segments

La longueur du segment varie la quantité d'information que le système a pour réduire l'égo-bruit. À noter qu'un chevauchement est présent sur tous les segments. Ce chevauchement permet de ne pas perdre d'information lors du calcul de la STFT et de la iSTFT. Il est lié à la taile de la STFT dans le spectrogramme. Donc, il est égale pour chacun des tests dans cette section. Les résultats présentés au tableau 4.2 montrent qu'un segment plus long permet au système de mieux performer. Cependant, la longueur des segments influence aussi la latence de l'audio. Le segment est accumulé à l'entrée du système et traité d'un coup. La vitesse de réaction du robot dépend donc de la longueur des segments. L'intégrateur robotique doit faire un compromis entre la performance du système et la latence

de l'audio en sortie. Finalement, toutes les longueurs de segments testées fonctionnent en temps réel sur l'ordinateur utilisé.

TABLEAU 4.2 SNR et SDR pour différentes longueurs de segments

Longeur de segment (s)	**Temps de traitement (s)**	**Augmentation SNR (dB)**	**Augmentation SDR (dB)**
0.25	0.13	9.32	7.33
0.5	0.2	11.4	11.32
0.75	0.28	12.34	12.09
1.0	0.32	12.48	12.37

4.3 Taille de la Transformée de Fourier rapide à court terme le spectrogramme

La quantité de fréquences dans le spectrogramme a un impact sur la résolution du signal temps-fréquence. Une grande résolution peut aider à séparer les fréquences provenant de l'égo-bruit et de la parole, ce qui rend la réduction de l'égo-bruit plus facile. Cependant, la matrice de covariance devient plus spécifique, ce qui peut demander plus d'exemples dans la base de données pour représenter tous les cas possibles. À noter que le chevauchement est encore une fois gardé constant en fonction taille de la STFT. Un chevauchement plus grand est nécessaire pour une plus grande STFT pour éviter la perte d'information lors de la STFT inverse. La quantité de fréquences impacte la quantité de calculs à toutes les étapes du système, et donc directement le temps de traitement du signal. Les résultats présentés à la figure 4.3 montrent que 1024 fréquences permettent d'obtenir la meilleure augmentation SNR et SDR en plus de réduire le temps de traitement du système. Cependant, la taille de la STFT impacte aussi la qualité audio à la sortie de la iSTFT. Si le modèle de reconnaissance vocale ou de détection d'évènements sonores utilise une taille de STFT plus grande que celle du système de réduction d'égo-bruit, ses performances peuvent diminuer. Il faut donc s'assurer d'utiliser une taille de STFT assez grande.

4.4 Retour sur le choix des paramètres

Les paramètres du chapitre 2.3.6 ont été choisis en fonction de leur impact sur les performances et de l'utilisabilité du système. Premièrement, 90 secondes de temps de calibration offre les meilleures performances parmi les quatres tests tout en restant rapide à calibrer. Deuxièmement, un segment de 0.5 seconde de longueur est un compromis entre la vitesse

TABLEAU 4.3 SNR et SDR pour différentes tailles de STFT dans le spectrogramme

Longeur de segment (s)	**Temps de traitement (s)**	**Augmentation SNR (dB)**	**Augmentation SDR (dB)**
512	0.05	11.69	11.47
1024	0.12	12.03	11.69
2048	0.2	11.4	11.32

de réaction du robot et la performance du système. Finalement, 2048 fréquences dans le spectrogramme ont été utilisées pour minimiser les pertes de performance dans les modèles de reconnaissance vocale et de détection d'évènements sonores.

CHAPITRE 5

IMPLÉMENTATION DÉTAILLÉE DE LA SOLUTION

Ce chapitre présente l'implémentation détaillée de la solution présentée au chapitre 2.3.6. La section 5.1 porte sur l'implémentation logicielle et la section 5.2 sur l'implémentation matérielle.

5.1 Implémentation logicielle

Le système présenté au chapitre précédent est implémenté sur *Robot Operating System* (ROS). La figure 5.1 présente l'architecture du système. La librairie audio_utils [10] (en bleu sur la figure) est utilisée pour sa facilité à intégrer l'audio dans ROS. Premièrement, le nœud *capture_node.cpp* permet de sélectionner une matrice de microphones, de lire en temps réel le signal audio et de l'écrire sur dans un message ROS *AudioFrame.msg*. Avec ce nœud, il est possible de sélectionner la longueur du segment voulue pour chaque message. C'est donc ce dernier qui accumule le segment de 0.5 seconde nécessaire au système de réduction de l'égo-bruit. Cependant, l'architecture est modulaire et peut prendre différentes longueurs de segments. Les signaux des 16 microphones sont contenus dans un même message. Deuxièmement, le nœud *playback.msg* permet de jouer sur un haut-parleur le signal audio provenant de ce même message ROS.

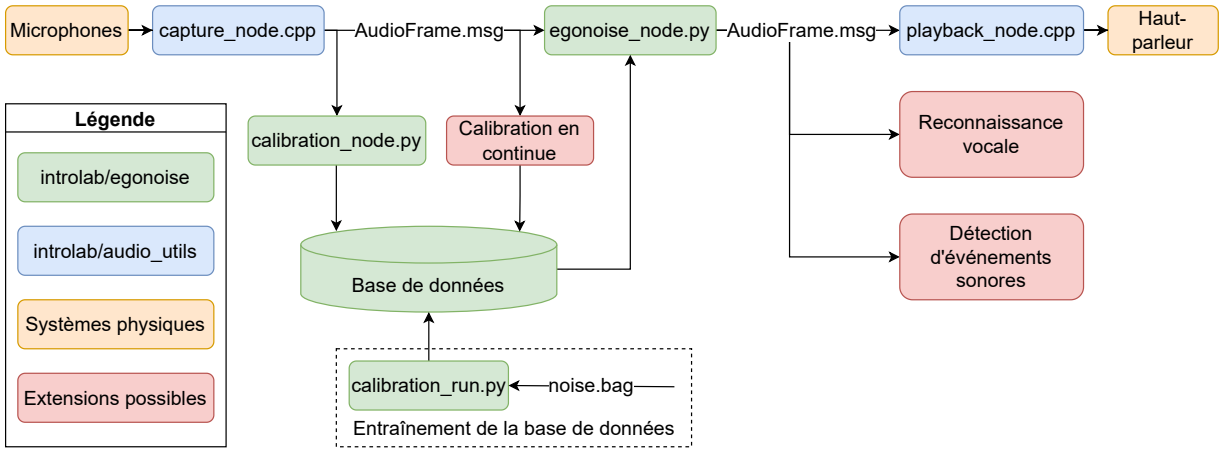

FIGURE 5.1 Architecture du système.

10. https ://github.com/introlab/audio_utils

La méthode présentée dans ce mémoire est représentée par les nœuds en vert dans la figure 5.1. Premièrement, le nœud *calibration_ run.py* permet d'entrainer la base de données à partir d'une liste de fichier rosbags contenant le bruit du robot ciblé. Deuxièmement, le nœud *calibration_ node.py* permet d'entrainer la base de données en ligne à partir du signal lu par le nœud *capture_ node.cpp*. Il prend en entrée la durée de calibration voulue. Troisièmement, le nœud *egonoise_ node.py* permet l'utilisation du système expliqué au chapitre 2.3.6. Tous les nœuds sont implémentés en Python3 et utilisent numpy [11], Sklearn [12] et Kissdsp [13]. La base de données est un dossier contenant tous les supervecteurs, expliqué à la section 3.3, en format Numpy. Le modèle de transformation PCA de Sklearn ainsi que la représentation dense de ces vecteurs sont aussi enregistrés dans ce dossier. Lors de l'initialisation du nœud *egonoise_ node.py*, le modèle PCA et la représentation dense des supervecteurs sont chargés en mémoire. Les supervecteurs sont chargés lorsqu'ils sont sélectionnés par l'algorithme de sélection. Ce processus permet de diminuer l'utilisation de la mémoire vive (RAM) utilisée par le système. Un fois un supervecteur chargé, la iSCM est reconstruite pour être utilisée dans le MVDR. Une fois le signal traité, il est publié en message ROS *AudioFrame.msg*.

Les éléments en rouge sont des fonctionnalités qui pourraient être ajoutées au système dans le futur. Ces éléments sont discutés au chapitre 6.

Ce système est disponible sur GitHub. Cependant, pour mesurer la performance du système, un script a été conçu pour obtenir les résultats présents au chapitre 2.3.6. Le script prend en entrée deux listes de fichiers rosbags : la première de l'égo-bruit et la deuxième du signal cible. Une fois lancé, le script mixte l'égo-bruit au signaux cibles, exécute la réduction de l'égo-bruit de la même manière que le nœud *egonoise_ node.py*, puis calcule les métriques SNR et SDR. D'autres scripts sont ensuite utilisés pour calculer le WER et AP à partir des résultats.

5.2 Implémentation matérielle

La figure 3.2 présente le robot et la matrice de microphones utilisés pour ce projet. La matrice de microphones a été conçue dans le cadre de ce projet pour un robot Jackal de la compagnie Clearpath. Ce robot est un UGV qui fonctionne à l'intérieur et à l'extérieur. Les tests ont été effectués seulement à l'intérieur pour évaluer le système en présence de réverbération. Les microphones ont été positionnés sur le contour du robot à l'aide de support imprimé en 3D en PLA. Ceux-ci sont attachés à un support de bois découpé au laser.

11. https ://numpy.org/
12. https ://scikit-learn.org/stable/
13. https ://github.com/FrancoisGrondin/kissdsp

Certains microphones et câbles sont proches des antennes de communication Bluetooth et Wifi du robot ce qui crée de l'interférence dans les signaux audios. Le système est capable de filtrer l'interférence, mais il serait préférable, pour une prochaine version, d'éloigner les microphones et câbles des antennes pour éviter l'interférence. Le PCB d'acquisition est positionné sur le dessus du robot pour simplifier le projet. Il serait désirable de l'insérer à l'intérieur du robot pour libérer l'espace pour d'autres capteurs. Pour l'acquisition du signal, la matrice de microphones est connectée à un Raspberry Pi. Les équipements sont alimentés par une batterie externe 5V.

Pour les tests, le robot est déplacé à l'aide d'une manette Bluetooth. Les déplacements sont aléatoires avec des vitesses variables. Le robot de base à deux modes : basse et haute vitesse. Seulement le mode basse vitesse est utilisé (maximum 0.4 m/sec), car un robot autonome intérieur se déplace principalement lentement. Ensuite, les enregistrements sont transférés à un ordinateur Dell XPS pour traiter les données et obtenir les résultats présentés au chapitre 2.3.6. L'ordinateur interne du robot pourrait être utilisé pour le traitement en temps réel. L'ordinateur peut avoir un impact sur la vitesse d'exécution du système. Le chapitre 4 explique l'impact de certains paramètres sur la vitesse d'exécution du système. Ils peuvent donc être changés en fonction des performances de l'ordinateur choisi.

CHAPITRE 6

CONCLUSION

La reconnaissance vocale et la détection d'évènements sonores permettent au robot de réagir aux sons de leur environnement. Cependant, le bruit causé par les éléments mécaniques et électriques des composantes du robot diminue la performance des modèles pré-entrainés. Ce mémoire présente une méthode de réduction de l'égo-bruit combinant l'apprentissage de matrice de covariance du bruit à un algorithme de formation de faisceau de réponses à variance minimum sans distorsion.

Le chapitre 2.3.6 présente mathématiquement la technique utilisée et les performances du système sur une plateforme robotique réelle. Les performances sont évaluées par l'amélioration du ratio signal-distorsions et l'amélioration de la performance de reconnaissance vocale et de détection d'évènements sonores. Les résultats obtenus montrent que le système améliore le ratio signal-distorsion en moyenne de 12 dB pour la tous les types de signal cible. La performance de la reconnaissance vocale augmente de 30 % à 55 %. Au niveau de la détection d'évènements sonore, l'amélioration de la performance augmente seulement pour le signal de la musique de 7 % à 20 %. Cependant, aucune augmentation significative n'est mesurée pour les autres évènements.

Le chapitre 4 introduit une analyse des principaux paramètres du système. Les résultats obtenus montrent que le temps de calibration permet d'obtenir plus d'exemples d'égo-bruit dans la base de données, ce qui se traduit en une meilleure performance. La longueur du segment permet aussi d'obtenir de meilleure performance, mais ajoute de la latence au système. Une plus grande taille de la Transformée de Fourier rapide à court terme ne permet pas d'augmenter les performances et nécessite plus de temps de traitement. Cependant, elle permet de garder plus d'information sur les fréquences dans le signal résultant, ce qui peut possiblement être bénéfique pour les modèles de reconnaissance vocale et de détection d'évènement sonore. Le choix des paramètres est dépendant de la performance voulue et du temps de traitement souhaité, ce qui est lié à la quantité de calcul engendré par le système.

Le chapitre 5 décrit l'implémentation détaillée de la solution. Le système est implémenté sur ROS. Il peut être intégré sur différents robots utilisant une matrice de microphones.

Plusieurs améliorations au système sont proposées, dont un mode de calibration en continu qui permettrait au robot de continuer son apprentissage lorsqu'il change d'environnement.

Le présent projet offre un nouveau système de réduction de l'égo-bruit disponible publiquement. Le système devrait pouvoir s'intégrer rapidement à tous les systèmes robotiques utilisant ROS et une matrice de microphones, car il ne nécessite aucun capteur externe et peut être calibré en seulement quelques minutes à l'aide du robot. De plus, il ne dépend pas du signal cible, ce qui permet de l'utiliser sur des signaux inconnus, et utilise un algorithme linéaire qui n'ajoute pas de distorsion au signal filtré.

Quelques améliorations peuvent être apportées au système proposé. Premièrement, un nœud de calibration en continu permettrait au système de continuer de se calibrer lors de son utilisation. Lorsque le robot change d'environnement ou que certains bruits changent, le dictionnaire peut alors évoluer et permettre au système de réduire l'égo-bruit sans diminution des performances. Pour cet ajout, il faut cependant s'assurer que le signal audio contient seulement l'égo-bruit du robot. Deuxièmement, il est possible d'ajouter des nœuds ROS de reconnaissance vocale et de détection d'évènements sonores directement dans le système pour permettre aux intégrateurs robotiques de bénéficier directement de ces données. Troisièmement, au niveau du nœud *egonoise_node*, le spectrogramme est calculé pour chaque segment de 0.5 seconde. Il est possible d'accumuler le spectrogramme et la matrice de covariance en temps réel, ce qui permet de sauver du temps de calcul lors de l'utilisation du système. Quatrièmement, il est possible de diminuer la quantité de calculs du MVDR en fonction du microphone de référence choisi pour diminuer l'empreinte CPU du système.

Les résultats de l'article montrent une amélioration de la reconnaissance vocale. Cependant, les résultats de la détection d'évènements sonores sont médiocres à l'exception de la musique. Plusieurs raisons peuvent expliquer ces résultats. Le SNR de départ peut être plus bas si le robot fonctionne à plus haute vitesse ou si l'évènement est produit moins fort. Cela peut impacter les résultats. Il serait donc intéressant de récolter des données avec des SNR plus variées. De plus, le modèle PANNS est entrainé sur 527 classes provenant d'Audioset. Les évènements audios utilisés dans cet article peuvent différer de ceux provenant d'Audioset. Par exemple, plusieurs classes d'alarmes sont présentes dans PANNS : alarme, alarme de cellulaire, alarme de feu, etc. Dans ce mémoire, seulement la classe alarme est utilisée, ce qui peut avoir un impact sur les résultats. Il peut être intéressant d'entrainer un modèle maison pour valider le système. Avec cette technique, les données utilisées pour valider les résultats se rapprochent plus de celles utilisées pour chacune des classes lors de l'entrainement du modèle.

Finalement, il serait intéressant de tester le système proposé avec d'autres robots. Par exemple, si un bras manipulateur est ajouté sur le robot ClearPath, la source de bruit bouge autour des microphones. Puisque le système fonctionne avec un algorithme de faisceau, les performances peuvent changer. La calibration du système serait plus longue pour permettre au robot de se déplacer à tous les endroits possibles. De plus, si le bras manipulateur est entre la source cible et les microphones, il est possible que la source cible soit complètement enlevée du signal audio.

LISTE DES RÉFÉRENCES

[1] Y. Bando, H. Saruwatari, N. Ono, S. Makino, K. Itoyama, D. Kitamura, M. Ishimura, M. Takakusaki, N. Mae, K. Yamaoka, Y. Matsui, Y. Ambe, M. Konyo, S. Tadokoro, K. Yoshii, et H. G. Okuno, "Low latency and high quality two-stage human-voice-enhancement system for a hose-shaped rescue robot," *Journal of Robotics and Mechatronics*, vol. 29, no. 1, p. 198–212, 2017.

[2] A. Briegleb, A. Schmidt, et W. Kellermann, "Deep clustering for single-channel ego-noise suppression," dans *Proceedings of the International Congress on Acoustics*, 2019, p. 2813–2820.

[3] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," dans *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, p. 4774–4778.

[4] A. Deleforge et W. Kellermann, "Phase-optimized k-SVD for signal extraction from underdetermined multichannel sparse mixtures," dans *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, p. 355–359.

[5] F. G. Encinas, L. A. Silva, A. S. Mendes, G. V. González, V. R. Q. Leithardt, et J. F. De Paz Santana, "Singular spectrum analysis for source separation in drone-based audio recording," *IEEE Access*, vol. 9, p. 43 444–43 457, 2021.

[6] J. Even, H. Sawada, H. Saruwatari, K. Shikano, et T. Takatani, "Semi-blind suppression of internal noise for hands-free robot spoken dialog system," dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, p. 658–663.

[7] H. Fang, G. Carbajal, S. Wermter, et T. Gerkmann, "Joint reduction of ego-noise and environmental noise with a partially-adaptive dictionary," dans *Proceedings of the ITG Conference on Speech Communication*, 2021, p. 114–118.

[8] R. P. Fernandes, E. C. Santos, A. L. L. Ramos, et J. A. Apolinário Jr., "A first approach to signal enhancement for quadcopters using piezoelectric sensors," dans *Proceedings of the International Conference on Transformative Science and Engineering, Business and Social Innovation*, 2015, p. 536–541.

[9] K. Furukawa, K. Okutani, K. Nagira, T. Otsuka, K. Itoyama, K. Nakadai, et H. G. Okuno, "Noise correlation matrix estimation for improving sound source localization by multirotor UAV," dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, p. 3943–3948.

[10] T. Haubner, A. Schmidt, et W. Kellermann, "Multichannel nonnegative matrix factorization for ego-noise suppression," dans *Proceedings of the Speech Communication*, 2018, p. 136–140.

[11] Y. Hioka, M. Kingan, G. Schmid, et K. A. Stol, "Speech enhancement using a microphone array mounted on an unmanned aerial vehicle," dans *Proceedings of the IEEE International Workshop on Acoustic Signal Enhancement*, 2016, p. 1–5.

[12] G. Ince, K. Nakadai, et K. Nakamura, "Online learning for template-based multi-channel ego noise estimation," dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, p. 3282–3287.

[13] G. Ince, K. Nakadai, T. Rodemann, Y. Hasegawa, H. Tsujino, et J. Imura, "Ego noise suppression of a robot using template subtraction," dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, p. 199–204.

[14] G. Ince, K. Nakadai, T. Rodemann, Y. Hasegawa, H. Tsujino, et J. Imura, "A hybrid framework for ego noise cancellation of a robot," dans *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010, p. 3623–3628.

[15] G. Ince, K. Nakadai, T. Rodemann, J. Imura, K. Nakamura, et H. Nakajima, "Assessment of single-channel ego noise estimation methods," dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, p. 106–111.

[16] G. Ince, K. Nakadai, T. Rodemann, J. Imura, K. Nakamura, et H. Nakajima, "Incremental learning for ego noise estimation of a robot," dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, p. 131–136.

[17] G. Ince, K. Nakadai, T. Rodemann, H. Tsujino, et J. Imura, "Multi-talker speech recognition under ego-motion noise using missing feature theory," dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, p. 982–987.

[18] M. Ishimura, S. Makino, T. Yamada, N. Ono, et H. Saruwatari, "Noise reduction using independent vector analysis and noise cancellation for a hose-shaped rescue robot," dans *Proceedings of the IEEE International Workshop on Acoustic Signal Enhancement*, 2016, p. 1–5.

[19] A. Ito, T. Kanayama, M. Suzuki, et S. Makino, "Internal noise suppression for speech recognition by small robots." dans *Proceedings of the European Conference on Speech Communication and Technology*, 2005, p. 2685–2688.

[20] B. Kang, H. Ahn, et H. Choo, "A software platform for noise reduction in sound sensor equipped drones," *IEEE Sensors Journal*, vol. 19, no. 21, p. 10 121–10 130, 2019.

[21] J. Kim, J. Choi, J. Son, G. Kim, J. Park, et J. Chang, "MIMO noise suppression preserving spatial cues for sound source localization in mobile robot," dans *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2021, p. 1–5.

[22] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, et M. D. Plumbley, "PANNS : Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, p. 2880–2894, 2020.

[23] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, et L. Jackel, "Handwritten digit recognition with a back-propagation network," dans *Proceedings of the Advances in Neural Information Processing Systems*, D. Touretzky, édit. Morgan-Kaufmann, 1989.

[24] M. Lee et J. Chang, "DNN-based speech recognition system dealing with motor state as auxiliary information of DNN for head shaking robot," dans *Proceedings*

of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2018, p. 1859–1863.

[25] M. Lee et J. Chang, "Augmented latent features of deep neural network-based automatic speech recognition for motor-driven robots," *Applied Sciences*, vol. 10, no. 13, p. 1–10, 2020.

[26] H. W. Loellmann, H. Barfuss, A. Deleforge, S. Meier, et W. Kellermann, "Challenges in acoustic signal enhancement for human-robot communication," dans *Proceedings of the Speech Communication*, 2014, p. 1–4.

[27] N. Mae, D. Kitamura, M. Ishimura, T. Yamada, et S. Makino, "Ego noise reduction for hose-shaped rescue robot combining independent low-rank matrix analysis and noise cancellation," dans *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2016, p. 1–6.

[28] P. Marmaroli, X. Falourd, et H. Lissek, "A UAV motor denoising technique to improve localization of surrounding noisy aircrafts : Proof of concept for anti-collision systems," dans *Proceedings of the French Congress of Acoustics and Annual IOA Meeting*, 2012, p. 1–6.

[29] J. Maxime, X. Alameda-Pineda, L. Girin, et R. Horaud, "Sound representation and classification benchmark for domestic robots," dans *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2014, p. 6285–6292.

[30] K. Nakadai, T. Lourens, H. G. Okuno, et H. Kitano, "Active audition for humanoid," dans *Proceedings of the National Conference Artificial Intelligence*, 2000, p. 832–839.

[31] Y. Nishimura, M. Ishizuka, K. Nakadai, M. Nakano, et H. Tsujino, "Speech recognition for a humanoid with motor noise utilizing missing feature theory," dans *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2006, p. 26–33.

[32] V. Panayotov, G. Chen, D. Povey, et S. Khudanpur, "Librispeech : An ASR corpus based on public domain audio books," dans *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, p. 5206–5210.

[33] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, et I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv :2212.04356*, 2022.

[34] C. Rascon, O. Ruiz-Espitia, et J. Martinez-Carranza, "On the use of the AIRA-UAS corpus to evaluate audio processing algorithms in unmanned aerial systems," *Sensors*, vol. 19, no. 18, p. 3902–3922, 2019.

[35] R. Sanchez-Matilla, L. Wang, et A. Cavallaro, "Multi-modal localization and enhancement of multiple sound sources from a micro aerial vehicle," dans *Proceedings of the ACM international conference on Multimedia*, 2017, p. 1591–1599.

[36] G. Schillaci, C. N. Ritter, V. V. Hafner, et B. Lara, "Body representations for robot ego-noise modelling and prediction. towards the development of a sense of agency in artificial agents," dans *Proceedings of the International Conference on the Synthesis and Simulation of Living Systems*, 2016, p. 390–397.

[37] A. Schmidt, A. Brendel, T. Haubner, et W. Kellermann, "Motor data-regularized non-negative matrix factorization for ego-noise suppression," *Journal on Audio, Speech, and Music Processing*, vol. 11, p. 1–15, 2020.

[38] A. Schmidt, A. Deleforge, et W. Kellermann, "Ego-noise reduction using a motor data-guided multichannel dictionary," dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, p. 1281–1286.

[39] A. Schmidt et W. Kellermann, "Informed ego-noise suppression using motor data-driven dictionaries," dans *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, p. 116–120.

[40] A. Schmidt et W. Kellermann, "Multichannel nonnegative matrix factorization with motor data-regularized activations for robust ego-noise suppression," dans *Proceedings of the IEEE International Conference on Autonomous Systems*, 2021, p. 1–5.

[41] A. Schmidt, H. Lollmann, et W. Kellermann, "A novel ego-noise suppression algorithm for acoustic signal enhancement in autonomous systems," dans *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, p. 6583–6587.

[42] T. Spadini, G. S. Imai Aldeia, G. Barreto, K. Alves, H. Ferreira, R. Suyama, et K. Nose-Filho, "On the application of SEGAN for the attenuation of the ego-noise in the speech sound source localization problem," dans *Proceedings of the Workshop on Communication Networks and Power Systems*, 2019, p. 1–4.

[43] N. Srivastava, "Improving neural networks with dropout," Thèse de doctorat, Université de Toronto, 2013.

[44] J. Taghia, D. Kolossa, et R. Martin, "ALE for robots ! A single-channel approach to robot self-noise cancellation," dans *Proceedings of the IEEE International Workshop on Acoustic Signal Enhancement*, 2016, p. 1–5.

[45] M. Takakusaki, D. Kitamura, N. Ono, T. Yamada, S. Makino, et H. Saruwatari, "Ego-noise reduction for a hose-shaped rescue robot using determined rank-1 multichannel nonnegative matrix factorization," dans *Proceedings of the IEEE International Workshop on Acoustic Signal Enhancement*, 2016, p. 1–4.

[46] Z. Tan, A. H. T. Nguyen, et A. W. H. Khong, "An efficient dilated convolutional neural network for UAV noise reduction at low input SNR," dans *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2019, p. 1885–1892.

[47] T. Tezuka, T. Yoshida, et K. Nakadai, "Ego-motion noise suppression for robots based on semi-blind infinite non-negative matrix factorization," dans *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014, p. 6293–6298.

[48] L. Wang et A. Cavallaro, "Ear in the sky : Ego-noise reduction for auditory micro aerial vehicles," dans *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2016, p. 152–158.

[49] L. Wang et A. Cavallaro, "Microphone-array ego-noise reduction algorithms for auditory micro aerial vehicles," *IEEE Sensors Journal*, vol. 17, no. 8, p. 2447–2455, 2017.

[50] L. Wang et A. Cavallaro, "Time-frequency processing for sound source localization from a micro aerial vehicle," dans *Proceedings of the IEEE International Conference on acoustics, Speech and Signal Processing*, 2017, p. 496–500.

[51] L. Wang et A. Cavallaro, "Acoustic sensing from a multi-rotor drone," *IEEE Sensors Journal*, vol. 18, no. 11, p. 4570–4582, 2018.

[52] L. Wang et A. Cavallaro, "A blind source separation framework for ego-noise reduction on multi-rotor drones," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, p. 2523–2537, 2020.

[53] L. Wang et A. Cavallaro, "Deep learning assisted time-frequency processing for speech enhancement on drones," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 6, p. 871–881, 2020.

[54] L. Wang, R. Sanchez-Matilla, et A. Cavallaro, "Audio-visual sensing from a quadcopter : Dataset and baselines for source localization and sound enhancement," dans *Proceedings of the International Workshop on Intelligent Robots and Systems*, 2019, p. 5320–5325.

[55] B. Yen, Y. Hioka, et B. Mace, "Estimating power spectral density of unmanned aerial vehicle rotor noise using multisensory information," dans *Proceedings of the European Signal Processing Conference*, 2018, p. 2434–2438.

[56] B. Yen, Y. Hioka, et B. Mace, "Improving power spectral density estimation of unmanned aerial vehicle rotor noise by learning from non-acoustic information," dans *Proceedings of the International Workshop on Acoustic Signal Enhancement*, 2018, p. 545–549.

[57] B. Yen, Y. Hioka, et B. Mace, "Source enhancement for unmanned aerial vehicle recording using multi-sensory information," dans *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2020, p. 850–857.

[58] S. Yoon, S. Park, Y. Eom, et S. Yoo, "Advanced sound capturing method with adaptive noise reduction system for broadcasting multicopters," dans *Proceedings of the IEEE International Conference on Consumer Electronics*, 2015, p. 26–29.

[59] S. Yoon, S. Park, et S. Yoo, "Two-stage adaptive noise reduction system for broadcasting multicopters," dans *Proceedings of the IEEE International Conference on Consumer Electronics*, 2016, p. 219–222.