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

SYSTÈME D'AUDITION ARTIFICIELLE EMBARQUÉ OPTIMISÉ POUR ROBOT MOBILE MUNI D'UNE MATRICE DE MICROPHONES

Thèse de doctorat
Spécialité : génie électrique

François GRONDIN

Sherbrooke (Québec) Canada

Décembre 2017

MEMBRES DU JURY

François MICHAUD

Directeur

Roch LEFEBVRE

Évaluateur

Patrick CARDINAL

Évaluateur

Jean-Marc VALIN

Évaluateur

RÉSUMÉ

Dans un environnement non contrôlé, un robot doit pouvoir interagir avec les personnes d'une façon autonome. Cette autonomie doit également inclure une interaction grâce à la voix humaine. Lorsque l'interaction s'effectue à une distance de quelques mètres, des phénomènes tels que la réverbération et la présence de bruit ambiant doivent être pris en considération pour effectuer efficacement des tâches comme la reconnaissance de la parole ou de locuteur. En ce sens, le robot doit être en mesure de localiser, suivre et séparer les sources sonores présentes dans son environnement.

L'augmentation récente de la puissance de calcul des processeurs et la diminution de leur consommation énergétique permettent dorénavant d'intégrer ces systèmes d'audition articielle sur des systèmes embarqués en temps réel. L'audition robotique est un domaine relativement jeune qui compte deux principales librairies d'audition artificielle : ManyEars et HARK. Jusqu'à présent, le nombre de microphones se limite généralement à huit, en raison de l'augmentation rapide de charge de calculs lorsque des microphones supplémentaires sont ajoutés. De plus, il est parfois difficile d'utiliser ces librairies avec des robots possédant des géométries variées puisqu'il est nécessaire de les calibrer manuellement.

Cette thèse présente la librairie ODAS qui apporte des solutions à ces difficultés. Afin d'effectuer une localisation et une séparation plus robuste aux matrices de microphones fermées, ODAS introduit un modèle de directivité pour chaque microphone. Une recherche hiérarchique dans l'espace permet également de réduire la quantité de calculs nécessaires. De plus, une mesure de l'incertitude du délai d'arrivée du son est introduite pour ajuster automatiquement plusieurs paramètres et ainsi éviter une calibration manuelle du système. ODAS propose également un nouveau module de suivi de sources sonores qui emploie des filtres de Kalman plutôt que des filtres particulaires.

Les résultats démontrent que les méthodes proposées réduisent la quantité de fausses détections durant la localisation, améliorent la robustesse du suivi pour des sources sonores multiples et augmentent la qualité de la séparation de 2.7 dB dans le cas d'un formateur de faisceau à variance minimale. La quantité de calculs requis diminue par un facteur allant jusqu'à 4 pour la localisation et jusqu'à 30 pour le suivi par rapport à la librairie ManyEars. Le module de séparation des sources sonores exploite plus efficacement la géométrie de la matrice de microphones, sans qu'il soit nécessaire de mesurer et calibrer manuellement le système.

Avec les performances observées, la librairie ODAS ouvre aussi la porte à des applications dans le domaine de la détection des drones par le bruit, la localisation de bruits extérieurs pour une navigation plus efficace pour les véhicules autonomes, des assistants main-libre à domicile et l'intégration dans des aides auditives.

Mots-clés : Audition robotique, localisation de sources sonores, suivi de sources sonores, séparation de sources sonores, système embarqué

À mes parents

REMERCIEMENTS

Je tiens d'abord à remercier mon directeur François Michaud pour son soutien et sa disponibilité tout au long de mon doctorat. La confiance qu'il m'a manifestée a été un élément-clé dans la réussite de ce projet. Tout au long de mon parcours, il a su me transmettre sa passion pour le domaine, ce qui a consolidé mon intérêt pour la recherche. C'est d'ailleurs principalement grâce à lui si je désire maintenant poursuivre une carrière académique. Il m'a également transmis son intérêt pour l'engagement social et l'implication au sein de la communauté, notamment par l'entremise de l'initiative FIRST.

Je remercie les membres de mon jury, c'est-à-dire Roch Lefebvre, Patrick Cardinal et Jean-Marc Valin, qui ont accepté de participer à la révision de cette thèse. Durant ce projet, j'ai été soutenu financièrement par le Fonds de Recherche du Québec – Nature et Technologies (FRQNT). Je tiens à remercier cet organisme pour sa contribution significative.

J'aimerais remercier Dominic Létourneau, Vincent-Philippe Rhéaume et Cédric Godin pour leur aide avec le matériel utilisé pour effectuer mes expériences sur plusieurs plateformes robotiques. Je tiens également à remercier tous les membres du laboratoire IntRoLab qui ont participé aux expériences. De plus, je veux souligner la participation spéciale de David Brodeur, Jean-Samuel Lauzon et Sébastien Laniel qui ont intégré le système d'audition ODAS au sein de plusieurs projets du laboratoire.

Tout au long de mes études, j'ai su compter sur l'amitié et le support de mon ami David Rancourt, qui m'a encouragé à persévérer malgré les nombreux défis que j'ai dû relever. Je tiens à le remercier particulièrement pour son optimisme et enthousiasme contagieux qui m'ont inspiré tout au long de mes études.

Finalement j'aimerais remercier de tout coeur ma famille, c'est-à-dire mes parents, Robert et Raymonde, ainsi que ma soeur, Stéphanie, qui m'ont encouragé et soutenu tout au long de mon doctorat, et ce dans les moments de réussite mais également durant les périodes plus difficiles. Leur support et leur amour inconditionnel m'ont permis d'aller de l'avant et mener à terme ce grand projet.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

CHAPITRE 1

INTRODUCTION

Les progrès technologiques des dernières années en ce qui concerne, entre autres, les semiconducteurs, l'informatique, les actionneurs mécaniques, les télécommunications et le stockage d'énergie, ont ouvert la porte à de nouvelles applications robotiques. Des robots mobiles et autonomes sont dorénavant perçus comme des acteurs qui pourront considérablement améliorer la qualité de vie des individus. Plusieurs de ces robots sont déjà employés pour effectuer des tâches domestiques. Par exemple, les robots Scooba et Roomba nettoient et aspirent la poussière au plancher de manière autonome, tandis que le robot Dressman repasse au fer les vêtements [Sung *et coll.*, 2008]. D'autres robots tels que Aibo sont utilisés à des fins de divertissement auprès des enfants en bas âge [Quinlan *et coll.*, 2003]. Certains robots dont Wakamaru et Paro œuvrent également comme compagnons domestiques [Namera *et coll.*, 2008; Saint-Aimé *et coll.*, 2007]. Ces applications révèlent que l'interaction humain-robot est une composante importante en robotique. La vision est la principale modalité utilisée comme moyen d'interaction, mais récemment une attention plus particulière a été accordée à l'audition articielle en robotique.

La conception d'un système d'audition articielle dans un contexte robotique introduit des contraintes. En effet, les robots mobiles se déplacent dans un environnement bruité et dynamique. Le système d'audition doit ainsi être en mesure de localiser, suivre et séparer plusieurs sources sonores dans un milieu au sein duquel les phénomènes de réverbération et de bruit ambiant varient au fil du temps. De plus, les algorithmes doivent pouvoir être utilisés en temps réel sur un robot qui possède une capacité de calcul limitée. Il est alors parfois avantageux de choisir une disposition géométrique symmétrique pour les microphones. Entre autres, les matrices sphériques sont utiles car elles permettent d'exploiter la décomposition des ondes sonores en harmoniques d'ondes planes [Rafaely, 2005]. Cependant, cette configuration géométrique est difficilement réalisable sur un robot qui possède déjà ses propres contraintes physiques en raisons du matériel installé sur celui-ci. Un système d'audition pour un robot mobile doit donc pouvoir composer avec plusieurs configurations géométriques asymmétriques imposées par les caractéristiques physiques du robot.

Plusieurs robots possèdent un système d'audition binaurale (i.e., deux microphones) qui s'inspire de l'être humain. Le robot SIG2 utilise un système d'audition binaurale pour localiser et séparer plusieurs sources sonores [Nakadai *et coll.*, 2000, 2003, 2002]. Par ailleurs, les robots BIRON et Robovie sont également dotés de deux microphones employés pour localiser une source sonore [Fritsch *et coll.*, 2004; Kanda *et coll.*, 2002; Lang *et coll.*, 2003]. Cette technique d'audition utilise couramment la fonction de transfert de la tête du robot pour estimer les propriétés acoustiques reliées à la diffraction et la réflexion des signaux sur la tête du robot [Keyrouz *et coll.*, 2007a, 2006, 2007b; Youssef *et coll.*, 2010]. L'utilisation de deux microphones réduit considérablement le nombre de composantes électroniques nécessaires mais limite les performances, en particulier lorsque plusieurs source sonores sont actives simultanément. En effet, lorsqu'il y a moins de microphones que de sources sonores actives, le système est sous-déterminé et il devient impossible d'effectuer une séparation complète [Syskind *et coll.*, 2007]. D'autre part, à défaut d'employer un système inspiré de l'être humain, il est possible d'améliorer les performances en augmentant le nombre de microphones. Le robot Asimo utilise un ensemble de huit microphones installés autour de la tête du robot [Nakadai *et coll.*, 2009, 2008; Yamamoto *et coll.*, 2006]. Les robots SIG2 et Spartacus ont également été modifiés pour intégrer une matrice de huit microphones sur leur torse [Michaud *et coll.*, 2007; Yamamoto *et coll.*, 2007]. Un nombre élevé de microphones améliore les performances mais nécessite une plus grande quantité de calculs. Le nombre de microphones est également souvent restreint à huit en raison du nombre de canaux limités sur la carte d'acquisition.

Plusieurs travaux ont été réalisés afin de doter les robots de la capacité de localiser et suivre des sources sonores à l'aide d'une matrice de microphones [Antonacci *et coll.*, 2005, 2006; Kim *et coll.*, 2011; Liu *et coll.*, 2007; Nakadai *et coll.*, 2002; Valin *et coll.*, 2004a, 2006]. La localisation de sources sonores permet au robot de diriger son attention dans la direction d'un locuteur. Cette application est particulièrement utile lorsqu'un utilisateur désire interpeller le robot. La distance entre le robot et le locuteur est généralement largement supérieure à l'espacement entre les microphones. L'effet de champ lointain s'applique donc à ce scénario [Naylor et Gaubitch, 2010]. La direction du locuteur par rapport au robot est ainsi obtenue, mais la distance qui les sépare demeure inconnue en raison de l'effet de champ lointain.

La localisation peut d'abord s'effectuer en estimant la différence du temps d'arrivée. Cette opération est possible grâce à une corrélation croisée entre chaque paire de microphones, de façon similaire à ce qui existe pour l'audition binaurale [Youssef *et coll.*, 2012a,b]. L'analyse en composantes indépendantes peut également être employée pour effectuer la localisation [Nesta et Omologo, 2012; Nesta *et coll.*, 2008]. Il est aussi possible de procéder à une inversion du système multi-entrées-multi-sorties dans le domaine temporel.

Cette stratégie permet d'estimer la fonction de transfert entre chaque source et chaque microphone à partir de réponses impulsionnelles finies. Celles-ci sont obtenues suite à une optimisation de fonctions de coût qui exploitent la non-gaussianité, la non-stationnarité et la coloration des sources sonores [Buchner *et coll.*, 2007, 2005b] Le système est robuste au bruit et à la réverbération, mais la charge de calculs est importante, le segment de parole doit durer quelques secondes et la convergence vers une solution optimale n'est pas garantie. Un formateur de faisceau peut être utilisé afin de projeter les observations des microphones sur une base qui permet une interférence constructive dans la direction de la source active. Puisque l'énergie se situe majoritairement dans les basses fréquences, une normalisation du spectre blanchit le signal et améliore ainsi la résolution spatiale, en plus de diminuer les effets de la réverbération [Do et Silverman, 2011]. L'inconvénient de cette technique repose sur le fait que les bandes sont normalisées au même niveau indépendamment du rapport signal sur bruit pour chacune d'entre elles. Pour y remédier, la pondération des bandes à partir d'une estimation du bruit additif et de la réverbération tardive est proposée [Valin *et coll.*, 2004a, 2006, 2007a,b].

Il est également possible d'utiliser plusieurs méthodes inspirées de la technique de classification multiple de signaux (MUSIC) [Schmidt, 1986] MUSIC est robuste au bruit additif et offre une excellente résolution spatiale, mais nécessite une quantité importante de calculs. Il est possible de réduire davantage les effets du bruit additif à l'aide d'une décomposition en valeur propre généralisée, à condition d'être en mesure de caractériser précisément le bruit ambiant [Mizumoto *et coll.*, 2011; Nakadai *et coll.*, 2012; Nakamura *et coll.*, 2011a, 2012, 2011b]. Une approche incrémentale est proposée dans le but d'estimer dynamiquement le bruit additif [Okutani *et coll.*, 2012]. Par contre, une calibration empirique est nécessaire pour définir le taux d'adaptation. Le bruit produit par les actionneurs du robot peut également mener à une détérioration des performances. L'estimation de ce bruit à partir d'un dictionnaire et de la position, la vitesse et l'accélération des actionneurs permet de réduire considérablement ce phénomène [Ince *et coll.*, 2012, 2011b]. Par contre, ce système exige une instrumentation complète des joints mobiles du robot et l'entraînement préalable du dictionnaire. Lorsque la matrice de microphone possède une géométrie symétrique (linéaire, circulaire, sphérique, etc.), la décomposition en harmoniques spatiales permet d'effectuer une projection cohérente. Les observations reliées à chaque bande de fréquence sont projetées vers un seul espace commun, au sein duquel est ensuite effectuée une décomposition en valeurs propres, ce qui réduit significativement la quantité de calculs [Lunati *et coll.*, 2012]. Malheureusement, cette astuce est valide uniquement lorsque la forme de la matrice est symétrique, ce qui est souvent une contrainte importante lorsque les microphones sont fixés sur un robot avec une silouette asymétrique. La projection cohérente permet également d'identifier le nombre de sources sonores actives simultanément à l'aide du critère d'information d'Akaike [Danes et Bonnal, 2010]. Cette approche statistique élimine le problème de détection de sources à partir d'un seuil fixe qui mène souvent à de fausses détections ou à des détections manquées.

Lorsque la position d'une source sonore est connue, il est possible d'utiliser la matrice de microphones pour rehausser la qualité du son provenant de cette direction et ainsi diminuer les interférences indésirables. La séparation de sources sonores pour une matrice de microphones vise à isoler le signal de chaque locuteur, sans nécessairement éliminer complètement le bruit additif et la réverbération. Il est possible d'atténuer les signaux interférents à l'aide d'un formateur de faisceau, sans pour autant les éliminer complètement. Cette technique simple et robuste peut être employée pour initialiser les conditions initiales d'un système de séparation aveugle [Nakadai *et coll.*, 2003, 2009, 2002; Valin *et coll.*, 2004b,c]. Le formateur de faisceau peut être modifié de manière à prendre en considération la fonction de transfert de la tête du robot [Maazaoui *et coll.*, 2012]. Un taux d'adaptation dynamique est également proposé pour accélérer la convergence tout en préservant la stabilité du système [Nakajima *et coll.*, 2008a] D'autre part, l'analyse en composantes indépendantes dans le domaine fréquentiel exploite la distribution non-gaussienne des sources dans le but de les séparer. L'algorithme récursif convolutif permet une convergence rapide malgré la courte durée des signaux observés [Nesta *et coll.*, 2011]. L'utilisation de l'algorithme dans le domaine fréquentiel entraîne cependant des permutations et des gains aléatoires, qui peuvent être corrigés à l'aide de plusieurs mécanismes [Nesta *et coll.*, 2008, 2009]. Finalement, l'inversion du système multi-entrées-multi-sorties dans le domaine temporel peut être employée pour séparer les signaux [Buchner *et coll.*, 2003a,b, 2004a, 2005a]. Cependant, comme c'est le cas pour la localisation, cette technique exige une puissance de calculs élevée, un segment de parole d'une durée importante et sa convergence vers une solution optimale n'est pas garantie. Une fois la séparation complétée, un post-filtrage est effectué afin de mettre l'emphase sur les bandes de fréquences moins bruitées [Valin *et coll.*, 2004b].

L'audition articielle permet également à un utilisateur de prononcer des commandes vocales à l'intention du robot. Les signaux sont d'abord localisés, suivis, séparés et post-filtrés, pour ensuite être utilisés par un système de reconnaissance de la parole. Il est d'ailleurs possible d'identifier les mots prononcés par plusieurs locuteurs qui parlent simultanément. Par exemple, un robot peut reconnaître trois phrases prononcées simultanément [Yamamoto *et coll.*, 2005a]. Ce système de reconnaissance peut être combiné à un système d'audition binaurale [Nakadai *et coll.*, 2004; Takeda *et coll.*, 2006] ou un système équipé d'une matrice de microphones [Yamamoto *et coll.*, 2005b]. La reconnaissance de parole s'effectue traditionnellement à partir de modèles de mélange Gaussien et de modèles de Markov cachés [Young, 1996]. Récemment, l'utilisation de réseau de neurones convolutifs en apprentissage profond a permis d'améliorer la tolérance au bruit pour cette tâche [Hinton *et coll.*, 2012].

Il existe enfin d'autres usages en audition robotique, comme la reconnaissance de locuteur qui vise à identifier le locuteur selon ses caractéristiques vocales, indépendemment des mots prononcés. En général, un locuteur est identifié parmi un groupe de candidats connus. Pour ce faire, il est possible d'effectuer une reconnaissance de locuteur distincte sur chaque microphone séparément [Ji *et coll.*, 2008]. Par le passé, le système WISS a été conçu pour effectuer une reconnaissance de locuteur à partir d'un signal séparé à l'aide d'une matrice de microphones dans un environnement bruité [Grondin et Michaud, 2012]. De manière plus générale, des locuteurs peuvent également être triés selon leur âge et leur sexe [Zhan *et coll.*, 2009].

Il y a aussi la reconnaissance de bruits ambiants vise à identier des bruits de l'environnement qui se distinguent de la voix humaine. Cette perception permet au robot de mieux analyser la scène auditive et ainsi identier certains évènements. Le système extrait les caractéristiques propres à plusieurs sons du quotidien, entraîne plusieurs modèles et effectue par la suite une comparaison avec les sons perçus [Chu *et coll.*, 2008, 2009; Goldhor, 1993].

Pour la mise en œuvre de telles capacités, des librairies d'autition artificielle dédiées aux applications robotiques telles que ManyEars [Grondin *et coll.*, 2013] et HARK [Nakadai *et coll.*, 2010b] ont été réalisées. Bien qu'elles constituent un avancement majeur dans ce domaine, ces librairies demeurent difficiles à intégrer sur des robots en raison des contraintes géométriques de la matrice de microphones, d'une calibration longue et fastidieuse et d'une charge importante de calculs. Cette thèse vise à déterminer s'il est possible d'apporter des améliorations sur ces points pour faciliter le déploiement de telles capacités sur des robots possédant des géometries variées et une puissance de calcul limitée.

La nouvelle librairie d'audition artificielle en robotique développée, appelée ODAS (*Open embeddeD Audition System*), vise à implémenter les caractéristiques suivantes (les caractéristiques 1-5 sont des objectifs algorithmiques, et la caractéristique 6 est un défi d'implémentation) :

1. *Adaptabilité à la géometrie du robot* : Les robots, de par leurs contraintes mécaniques et la multitude de capteurs embarqués, possèdent des géométries différentes. La

librairie d'audition artificielle doit être en mesure de composer avec une disposition spatiale des microphones qui varient d'un robot à l'autre.

2. *Utilisation d'une matrice de microphones fermée* : Il est fréquent que les microphones soient installés autour d'une structure rigide (par exemple, le torse du robot ou sa tête). Dans ce cas précis, les ondes sonores ne se propagent pas en champ libre dans l'air, et ceci peut affecter les performances. Dans ce projet, il est désirable que cette configuration soit également prise en considération.
3. *Calibration rapide de la librairie d'audition* : Il est souhaitable de calibrer la librairie en indiquant manuellement la position des microphones et leur orientation, sans avoir recours à une caractérisation acoustique complète de la matrice de microphones sur le robot.
4. *Minimisation de la charge de calculs* : Puisque la librairie d'audition vise d'abord et avant tout à être déployée sur des plateformes robotiques autonomes, il faut réduire autant que possible la charge de calculs nécessaire. Ceci permet de prolonger la durée de la charge des batteries du robot, employer des ordinateurs peu dispendieux et peu volumineux, et libérer du temps de calcul pour d'autres processus à bord du robot (ex : vision, cartographie, planificiation, contrôle, etc.)
5. *Augmentation du nombre de microphones* : Ajouter un plus grand nombre de microphones vient rehausser la qualité de la localisation et de la séparation [Weinstein *et coll.*, 2004]. Cependant, l'augmentation du nombre de microphones implique une hausse de la charge des calculs. Les matrices de microphones sont normalement composées de huit capteurs [Nakadai *et coll.*, 2010b; Valin *et coll.*, 2004a,b], et il serait intéressant d'en augmenter leur nombre.
6. *Portabilité du système* : Il est souhaitable que la librairie développée puisse fonctionner sur l'ensemble des systèmes d'exploitation (Windows, MacOS et Linux), et également au besoin sur des systèmes embarqués possédant leur propre système d'opération. La librairie devrait également pouvoir être interfacée facilement avec le système d'exploitation ROS [Quigley *et coll.*, 2009], largement utilisé par la communauté scientifique dans le domaine de la robotique.

La figure 1.1 présente la librairie ODAS mise de l'avant dans cette thèse qui se scinde en trois modules : 1) la localisation, 2) le suivi et 3) la séparation. Un système de post-filtrage permet de rehausser la qualité sonore des sources séparées, mais n'est pas détaillé dans cette thèse puisqu'il est identique en tout point à celui proposé par le passé pour la librairie ManyEars [Valin *et coll.*, 2004b]. Cette librairie peut ensuite être connectée à des engins de reconnaissance de la parole, de reconnaissance de locuteur ou de classification de sons.

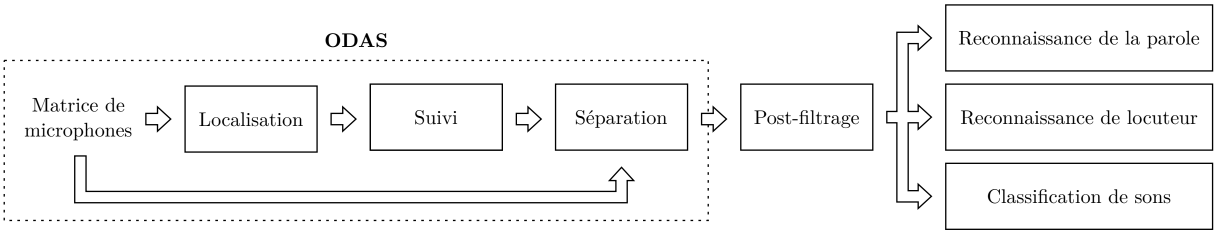


Figure 1.1 Aperçu de la librairie ODAS

Cinq contributions importantes sont présentées dans cette thèse :

1. L'introduction d'un modèle de directivité analytique pour chaque microphone pour la localisation et la séparation de sources sonores.
2. La recherche hiérarchique de sources sonores, et la réduction de l'espace de recherche selon la géométrie de la matrice de microphones pour faciliter la localisation.
3. La modélisation de l'incertitude du délai d'arrivée des sources sonores pour calibrer automatiquement certains paramètres.
4. L'utilisation d'un filtre de Kalman pour le suivi de sources sonores multiples.
5. La réalisation d'un système d'audition artificielle qui fonctionne en temps réel avec 16 microphones.

L'organisation de la thèse correspond aux travaux entrepris pour réaliser la librairie ODAS et atteindre les objectifs mentionnés. Tout d'abord, le chapitre 2 présente une revue de littérature des méthodes mises en place au sein des librairies d'audition artificielle ManyEars et HARK, qui sont les plus utilisées dans le domaine. Ensuite, les trois chapitres suivants présentent sous forme d'articles les trois modules d'ODAS. Le chapitre 3 introduit le module de localisation proposé pour la nouvelle librairie ODAS. Cette méthode modélise les microphones selon un faisceau directif afin d'améliorer les performances de localisation avec une matrice de microphones fermée. Une technique de calibration automatique des paramètres permet également d'optimiser le système de localisation selon la géométrie de la matrice. Enfin, la recherche hiérarchique de sources sonores réduit considérablement la charge de calculs. Le chapitre 4 décrit le module de suivi, qui emploie des filtres de Kalman modifiés pour suivre plusieurs sources simultanément, et réduire la charge de calcul par rapport aux méthodes existantes. Le chapitre 5 présente un module de séparation de sources sonores qui exploite les méthodes les plus couramment utilisées, en incluant un modèle directif pour chaque microphone afin de rehausser les performances dans le cas d'une matrice de microphones fermée. Le chapitre 6 introduit la librairie ODAS et présente un aperçu de son architecture logicielle. Finalement, le chapitre 7 conclut la thèse en soulignant les contributions scientifiques et propose de nouvelles fonctionnalités et des applications concrètes pour la librairie ODAS.

CHAPITRE 2

LIBRAIRIES D'AUDITION ARTIFICIELLE

L'intérêt récent pour l'audition artificielle en robotique a mené à la conception de librairies logicielles permettant d'effectuer la localisation, le suivi et la séparation de sources sonores. Les deux librairies les plus utilisées par la communauté roboticienne et disponibles en code ouvert sont ManyEars[1], conçue à l'Université de Sherbrooke (Québec, Canada) [Grondin *et coll.*, 2013] et HARK[2], développée à l'Institut de Recherche de Honda du Japon en collaboration avec l'Université de Kyoto (Japon) [Nakadai *et coll.*, 2008]. La description de ces deux librairies dans le présent chapitre est sommaire et vise à expliquer les fonctionnalités des librairies. Plus de détails sur les méthodes utilisées par ces librairies sont présentées aux chapitres 3, 4 et 5.

2.1 ManyEars

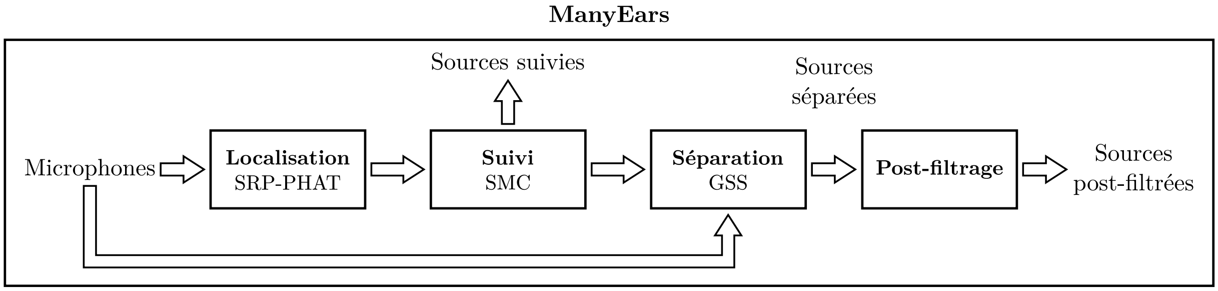


Figure 2.1 Architecture de la librairie ManyEars

La figure 2.1 illustre les modules qui constituent la librairie ManyEars : la localisation, le suivi, la séparation et le post-filtrage. La librairie ManyEars localise les sources sonores à l'aide d'une méthode dérivée de la corrélation croisée généralisée avec transformation de phase [Valin *et coll.*, 2004a]. ManyEars parcourt ensuite l'ensemble des directions dans l'espace en trois dimensions autour de la matrice de microphones. Cette méthode, appellée SRP-PHAT, associe à chaque direction les délais d'arrivée du son obtenus pour chaque paire de microphones grâce au calcul de la corrélation croisée, et retourne plusieurs directions candidates pour l'origine d'une ou plusieurs sources sonores. Une fois les directions d'arrivée potentielles générées par le module de localisation, ManyEars effectue un suivi

[1] http ://manyears.sf.net

[2] http ://hark.jp

des sources sonores à l'aide d'un ou plusieurs filtres particulaires, aussi appelés méthodes de Monte Carlo séquentielles (SMC), dans le but de modéliser le déplacement d'une ou plusieurs personnes, et filtrer les fausses détections [Valin *et coll.*, 2006, 2007a].

La position de chaque source sonore permet ensuite d'effectuer une séparation dans la direction correspondante pour rehausser la qualité du signal et diminuer les interférences provenant des autres sources indésirables. ManyEars emploie une méthode de séparation géométrique des sources (GSS), qui exploite l'indépendance des sources sonores et les directions d'arrivée différentes pour chaque source [Parra et Alvino, 2002; Valin *et coll.*, 2007b]. L'objectif est de minimiser une fonction de coût composée de deux termes, l'un mesurant le degré de corrélation des sources sonores, et l'autre visant à réduire l'écart entre la direction d'arrivée obtenue durant la localisation et celle employée au cours de la séparation. Contrairement à la méthode de GSS proposée par [Parra et Alvino, 2002] qui vise une optimisation globale, ManyEars utilise un algorithme de descente par gradient stochastique qui minimise cette fonction de coût avec un pas fixe, qui doit être suffisamment grand pour permettre une convergence rapide, mais assez petit pour garantir la stabilité du processus. Lors de cette séparation, ManyEars utilise un modèle de propagation en champ libre pour les ondes sonores, malgré le fait que la méthode soit parfois employée avec des matrices de microphones fermée. Une fois la séparation effectuée, il est possible de rehausser davantage la qualité sonore de chaque source avec une dernière étape de post-filtrage. ManyEars estime le bruit stationnaire et les signaux provenant des sources interférentes afin d'améliorer davantage le rapport signal sur bruit [Valin *et coll.*, 2004b]. Cette méthode permet notamment de générer des masques temps-fréquence qui améliore la reconnaissance vocale effectuée à partir des signaux séparés [Yamamoto *et coll.*, 2005b].

ManyEars a d'abord été déployée dans l'environnement FlowDesigner [Létourneau *et coll.*, 2005], pour ensuite être déployée comme librairie autonome [Grondin *et coll.*, 2013] et intégrée à l'environnement ROS [Quigley *et coll.*, 2009], le système d'opération utilisé sur la plupart des plateformes robotiques. ManyEars a également été adaptée pour fonctionner sur un processeur de signaux numériques dédié [Briere *et coll.*, 2008] avec certaines fonctionalités limitées (ex : limite du nombre de sources sonores suivies simultanément), en raison de la puissance de calcul disponible. Par le passé, ManyEars a été intégrée sur plusieurs robots possédant une matrice de huit microphones, dont Spartacus [Michaud *et coll.*, 2007], SIG2 [Yamamoto *et coll.*, 2005a], ASIMO [Yamamoto *et coll.*, 2006], IRL-1 [Grondin *et coll.*, 2013] et beam+ [Laniel *et coll.*, 2017]. La librairie a également été utilisée au sein d'un système de gestion de dialogue [Frechette *et coll.*, 2012], d'identification de personnes [Ouellet *et coll.*, 2014] et pour des applications de reconnaissance des émotions dans la voix humaine [Brodeur *et coll.*, 2016].

2.2 HARK

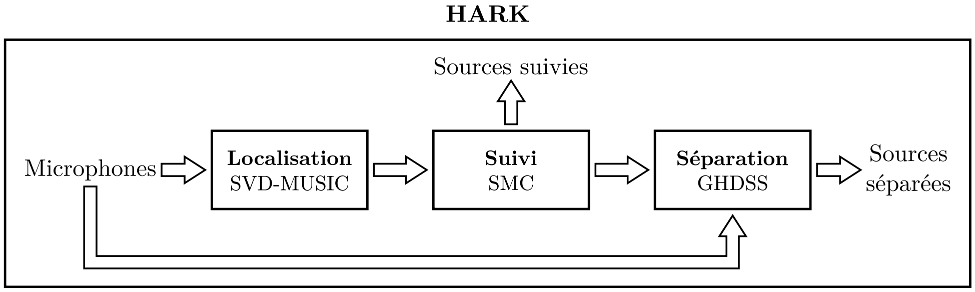


Figure 2.2 Architecture de la librairie HARK

La figure 2.2 démontre que la librairie HARK possède une architecture similaire à celle de ManyEars, à l'exception des méthodes de localisation et de séparation. La librairie HARK emploie plusieurs méthodes de localisation dérivées de la technique de classification de signaux multiples (MUSIC) [Schmidt, 1986]. Cette technique consiste à estimer les matrices de corrélation des signaux des microphones dans le domaine fréquentiel, et les décomposer en deux sous-espaces, l'un contenant du bruit seulement, et l'autre du bruit et du signal de la parole. Une recherche est ensuite effectuée dans les directions d'intérêt pour identifier les directions orthogonales au sous-espace du bruit. Cette méthode est conçue pour un signal à bande étroite, mais est adaptée aux signaux à bande large en effectuant cette recherche sur plusieurs bandes de fréquences qui couvrent le spectre de la parole humaine. La méthode de décomposition standard par valeurs propres (SEVD-MUSIC) adapte MUSIC pour des signaux à large bande, et permet une localisation robuste au bruit à condition que celui-ci soit moins énergétique que les signaux de parole [Nakadai *et coll.*, 2010b]. La méthode de décomposition généralisée par valeurs propres (GEVD-MUSIC) permet de palier à cette contrainte lorsque le bruit est dominant par rapport à la parole [Nakamura *et coll.*, 2011a]. Les méthodes SEVD- et GEVD-MUSIC peuvent réduire la précision de la localisation lorsque les bases du sous-espace du bruit ne sont pas orthonormales. La décomposition généralisée par valeurs singulières (GSVD-MUSIC) est introduite pour rétablir l'orthonormalité et améliorer la précision [Nakamura *et coll.*, 2012]. Bien qu'elles offrent une robustesse au bruit, les méthodes SEVD-, GEVD- et SVD-MUSIC nécessitent une quantité importante de calculs pour effectuer les décompositions par valeurs propres ou singulières et projeter les directions d'intérêt dans le sous-espace du bruit. Pour diminuer la charge de calculs associée à la recherche de directions, l'espace est souvent restreint à deux dimensions seulement, ce qui permet donc d'obtenir l'azimut mais pas l'élévation des directions des sources. Dans le cas d'une matrice de microphone fermée, il est possible d'améliorer les performances de localisation des méthodes SEVD-, GEVD- et SVD-MUSIC en mesurant les fonctions de transfert entre chaque direction d'arrivée du son sur un plan en deux dimensions et chaque microphone. Cette approche nécessiste toutefois une calibration fastidieuse, dont la complexité augmente drastiquement pour une recherche sur un plan en trois dimensions. Lorsque le système HARK estime les matrices de corrélation sur un court interval de temps, les trajectoires des sources sonores sont filtrées avec un filtre particulaire [Nakadai *et coll.*, 2008]. Cependant, lorsque la localisation se fait en deux dimensions sur un interval de temps de l'ordre des secondes, il est possible d'utiliser directement le résultat de la localisation puisque les fausses détections et la sparsité des signaux de parole sont filtrés naturellement lors de l'estimation des matrices de corrélation [Nakadai *et coll.*, 2010b].

HARK utilise également la méthode de séparation géométrique de sources sonores [Parra et Alvino, 2002]. Pour accélérer la convergence tout en garantissant la stabilité, [Nakajima *et coll.*, 2008a] modifie la méthode GSS qui devient GHDSS (séparation géométrique de sources avec une décorrélation d'ordre élevé), et propose un algorithme visant à adapter la taille du pas dynamiquement. Comme dans le cas de la localisation, il est possible d'améliorer les performances de séparation en intégrant les fonctions de transfert mesurées entre chaque direction d'arrivée et chaque microphone, à la condition d'effectuer une caractérisation acoustique complète de la matrice de microphones.

La librairie HARK utilise également FlowDesigner [Létourneau *et coll.*, 2005] afin de connecter et coordonner les modules de traitement qui la composent. De plus, une interface permet d'intégrer HARK au système d'opération ROS [Quigley *et coll.*, 2009]. Cette librairie a été déployée notamment sur les robots Asimo [Yamamoto *et coll.*, 2006] et SIG2 [Yamamoto *et coll.*, 2005a]. Puisque HARK effectue des opérations mathématiques complexes comme la décomposition par valeurs propres, il est difficile de porter cette librairie vers du matériel à faible coût. Par exemple, lorsque HARK est déployée sur des drones avec une capacitié de calculs limitée, le flux audio est transmis par réseau sans-fil vers un ordinateur au sol qui effectue les calculs [Ohata *et coll.*, 2014; Okutani *et coll.*, 2012]. Il s'agit d'une contrainte majeure car ceci réduit l'autonomie du robot et limite sa portée.

CHAPITRE 3

LOCALISATION DE SOURCES SONORES

Avant-propos

Titre : *A Lightweight and Optimized Sound Source Localization Method for Open and Closed Microphone Array Configurations*

Auteurs et affiliations :

François Grondin : étudiant au doctorat, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique

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

Date de soumission : 12 octobre 2017

État : Soumis

Revue : *IEEE Transactions on Robotics*

Titre français : Méthode minimisée et optimisée de localisation de sources sonores pour des matrices ouvertes et fermées de microphones

Contribution au document : Cet article contribue à la thèse en décrivant une nouvelle méthode de localisation de sources sonores qui s'adapte à plusieurs géométries de matrices de microphones et diminue la charge de calculs.

Résumé en français : L'interaction humain-robot dans des conditions normales nécessite un filtrage des sources sonores présentes dans l'environnement. Cette capacité implique normalement l'utilisation d'une matrice de microphones pour localiser, suivre et séparer les sources sonores en temps réel. Les techniques de traitement des signaux avec plusieurs microphones peuvent améliorer la tolérance au bruit mais la charge de calculs associée augmente habituellement en fonction du nombre de microphones, ce qui limite le temps de réponse et la diffusion de cette technologie sur des robots mobiles. Puisque la localisation de sources sonores nécessite une quantité importante de calculs, il est désirable de minimiser celle-ci afin de faciliter le déploiement sur des robots. La forme du robot introduit également des contraintes géométriques pour les matrices de microphones. Cet

article présente une nouvelle méthode de localisation de sources sonores, dénommée SRP-PHAT-HSDA, qui parcourt l'espace de recherche à partir des grilles avec des résolutions grossières et fines, ce qui réduit considérablement le nombre d'accès en mémoire. Un modèle de directivité pour les microphones est également introduit pour diminuer la quantité de directions à parcourir et ignorer certaines paires de microphones. Une méthode de calibration est proposée afin de configurer automatiquement les paramètres qui sont habituellement définis manuellement selon la forme de la matrice de microphones. À l'aide d'une matrice de seize microphones et d'un système embarqué à faible coût, il est démontré que la méthode introduite offre des résultats équivalents à ceux des autres méthodes, mais emploie quatre fois moins de ressources de calcul tout en conservant la même précision.

Abstract

Human-robot interaction in natural settings requires filtering out the different sources of sounds from the environment. Such ability usually involves the use of microphone arrays to localize, track and separate sound sources online. Multi-microphone signal processing techniques can improve robustness to noise but usually involves processing cost increasing with the number of microphones used, limiting response time and widespread use on different types of mobile robots. Since sound source localization methods are the most expensive in terms of computing resources, minimizing the amount of computations required would facilitate their implementation and use on robots. The robot's shape also bring constraints on the microphone array geometry and configurations. This paper presents a novel sound source localization method, called SRP-PHAT-HSDA, that scans space with coarse and fine resolution grids to reduce the number of memory lookups. A microphone directivity model is used to reduce the number of directions to scan and ignore non significant pairs of microphones. A configuration method is also introduced to automatically set parameters that are normally empirically tuned according to the shape of the microphone array. Using a 16-microphone array and low cost hardware, results show that the method performs as well as other sound source localization methods by using up to 4 times less computing resources, while preserving localization accuracy.

3.1 Introduction

Distant Speech Recognition (DSR) occurs when speech is acquired with one or many microphone(s) moved away from the mouth of the speaker, making recognition difficult because of background noise, overlapping speech from other speakers, and reverberation [Kumatari *et coll.*, 2012; Woelfel et McDonough, 2009]. DSR is necessary for enabling verbal interactions without the necessity of using intrusive body- or head-mounted devices. But still, recognizing distant speech robustly remains a challenge [Vacher *et coll.*, 2015]. Microphone arrays make it possible to capture sounds for DSR [Kumatari *et coll.*, 2012] in human-robot interaction (HRI). This requires the installation of multiple microphones on the robot platform, and to process distant speech perceived by filtering out noise from fans and actuators on the robot and non-stationary background in reverberant environments, fast enough to support live interactions. This process usually involves localizing, tracking and separating the perceived sound sources [Grondin *et coll.*, 2013] before performing speech recognition [Brodeur *et coll.*, 2016; Frechette *et coll.*, 2012].

In this process, sound source localization (SSL) is an expensive step in terms of computation. Improved capabilities for SSL can be directly associated with the number of microphones used, which influences processing requirements [Valin *et coll.*, 2007a]. Reducing the amount of computations to perform robust SSL can therefore be a benefit, either to increase the number of microphones in the array, to free on-board computing resources for other processes (e.g., speech processing, vision, planning, navigation), or to facilitate its implementation on various types of embedded computing platforms.

SSL returns the direction of arrival (DoA) of the sound sources, and to do so a variety of SSL algorithms exists. For instance, Rascon et al. [Rascon *et coll.*, 2015] present a lightweight SSL method that uses little memory and CPU resources, but is limited to three microphones and scans the DoA of sound source only in 2D. Stochastic region contraction [Do *et coll.*, 2007] and hierarchical search [Zotkin et Duraiswami, 2004] have also been studied to speed up scanning, but limit the search to a 2D surface. Nesta and Omologo [Nesta et Omologo, 2012] describe a generalized state coherence transform to perform SSL, which is particularly effective when multiple sound sources are present. However, this method relies on independent component analysis (ICA), which takes many seconds to converge. Drude et al. [Drude *et coll.*, 2015] use a kernel function that relies on both phase and level differences, at the cost of increasing the computational load. Loesch and Yang [Loesch et Yang, 2010] also introduce a localization method based on time-frequency sparseness, which remains sensitive to high reverberation levels. Multiple Signal Classification based on Standard Eigenvalue Decomposition (SEVD-MUSIC) makes SSL robust to additive noise [Nakadai *et coll.*, 2010b]. SEVD-MUSIC, initially used for narrowband signal [Schmidt, 1986], has been adapted for broadband sound sources such as speech [Ishi *et coll.*, 2009], and is robust to noise as long as the latter is less powerful than the signals to be localized. Multiple Signal Classification based on Generalized Eigenvalue Decomposition (GEVD-MUSIC) method [Nakamura *et coll.*, 2011a] has been introduced to cope with this issue, but the latter method increases computations. Multiple Signal Classification based on Generalized Singular Value Decomposition (GSVD-MUSIC) reduces computational load of GEVD-MUSIC and improves localization accuracy [Nakamura *et coll.*, 2012], but still relies on eigen decomposition of a matrix. Other methods take advantage of specific array geometries (linear, circular or spherical) to improve robustness and reduce computational load [Danes et Bonnal, 2010; Pavlidi *et coll.*, 2012; Rafaely *et coll.*, 2010]. Even though interesting properties arise from these geometries, these configurations are less practical for a mobile robot due to physical constraints introduced by its specific shape. SSL can also be performed using a Steered Response Power with Phase Transform (SRP-PHAT). The SRP-PHAT is usually computed using weighted Generalized Cross-Correlation with

Phase Transform (GCC-PHAT) at each pair of microphones [Grondin *et coll.*, 2013; Valin *et coll.*, 2007a]. SRP-PHAT requires less computations than MUSIC-based methods, but still requires a significant amount of computations when scanning the 3D-space for a large number of microphones.

To address these issues, this paper introduces a novel SRP-PHAT method referred to as SRP-PHAT-HSDA, for Hierarchical Search with Directivity model and Automatic calibration. SRP-PHAT-HSDA scans the 3D space over a coarse resolution grid, and then refines search over a specific area. It includes a Time Difference of Arrival (TDOA) uncertainty model to optimize the scan accuracy using various grid resolution levels with open and closed microphone array configurations. A microphone directivity model is also used to reduce the number of directions to scan and ignore non significant pairs of microphones.

The chapter is organized as follows. To better understand SSL computation requirements, Section 3.2 characterizes the computing requirements of SRP-PHAT in comparison to SEVD-MUSIC, to justify and situate the improvements brought by SRP-PHAT-HSDA. Section 3.3 describes SRP-PHAT-HSDA. Section 3.4 presents the experimental setup involving 16-microphone circular and closed cubic arrays on a mobile robot, implementing SSL on a Raspberry Pi 3, followed by Section 3.5 with the results. Finally, Section 3.6 concludes this paper with final remarks and future work.

3.2 Computing Requirements of SRP-PHAT versus SEVD-MUSIC

SSL is usually divided in two tasks : 1) estimation of TDOA, and 2) DoA search over the 3D space around the microphone array. The main difference between SRP-PHAT and SEVD-MUSIC lies in Task 1 : SRP-PHAT relies on the Generalized Cross-Correlation with Phase Transform method (GCC-PHAT), while SEVD-MUSIC uses Singular Eigenvalue Decomposition (SEVD). The intend here is to demonstrate which method is the most efficient for Task 1, and then, using this method, how can Task 2 be further improved to reduce computing needs.

Both methods first capture synchronously the acoustic signals x_m from the M microphones in the array. These signals are divided in frames of N samples, spaced by ΔN samples and multiplied by a the sine window $w[n]$:

$$x_m^l[n] = w[n]x_m[n + l\Delta N] \tag{3.1}$$

with l, i and n representing the frame, microphone and sample indexes, respectively. The methods then compute the Short-Time Fourier Transform (STFT) with a N-samples real Fast Fourier Transform (FFT), where the expression $X^l_m[k]$ stands for the spectrum at each frequency bin k, and the constant j is the complex number $\sqrt{-1}$:

$$X^l_m[k] = \sum_{n=0}^{N-1} x^l_m[n] \exp\left(-j2\pi kn/N\right) \tag{3.2}$$

SRP-PHAT relies on the Generalized Cross-Correlation with Phase Transform (GCC-PHAT), which is computed for each pair of microphones p and q (where $p \neq q$). The Inverse Fast Fourier Transform (IFFT) provides an efficient computation of the GCC-PHAT, given the time delay n is an integer :

$$r^l_{pq}[n] = \frac{1}{N} \sum_{k=0}^{N-1} \frac{X^l_p[k] X^l_q[k]^*}{|X^l_p[k]||X^l_q[k]| + \epsilon} \exp\left(j2\pi kn/N\right) \tag{3.3}$$

The IFFT complexity depends on the number of samples per frame N, which is usually a power of 2. The order of complexity for a real IFFT is $\mathcal{O}(N \log N))$. With $M(M-1)/2$ pairs of microphones, SRP-PHAT computing complexity reaches $\mathcal{O}(M^2 N \log N)$.

SEVD-MUSIC relies on singular eigenvalue decomposition of the cross-correlation matrix. The $M \times M$ correlation matrix $\mathbf{R}[k]$ is defined as follows, where $\mathrm{E}\{\ldots\}$ and $\{\ldots\}^H$ stand for the expectation and Hermitian operators, respectively :

$$\mathbf{R}[k] = \mathrm{E}\{\mathbf{X}[k]\mathbf{X}[k]^H\} \tag{3.4}$$

The $M \times 1$ vector $\mathbf{X}^l[k]$ concatenates the spectra of all microphones for each frame l and frequency bin k (where the operator $\{\ldots\}^T$ stands for the transpose) :

$$\mathbf{X}^l[k] = \begin{bmatrix} X^l_1[k] & X^l_2[k] & \ldots & X^l_M[k] \end{bmatrix}^T \tag{3.5}$$

In practice, the correlation matrix is usually computed at each frame l with an estimator that sums vectors over time (a window of L frames) for each frequency bin k :

$$\mathbf{R}^l[k] = \frac{1}{L} \sum_{\Delta L=0}^{L-1} \mathbf{X}^{l+\Delta L}[k] \mathbf{X}^{l+\Delta L}[k]^H \tag{3.6}$$

SEVD-MUSIC complexity depends on the size of the matrix $\mathbf{R}^l[k]$, and is $\mathcal{O}(M^3)$ [Holmes *et coll.*, 2007]. This operation is performed at each frequency bin k, for a total of $N/2$ bins, which leads to an overall complexity of $\mathcal{O}(M^3N)$.

To better express the computing requirements of both methods, Table 3.1 presents simulation results of the time (in sec) required to process one frame l, with various values of N and M, on a Raspberry Pi 3. SRP-PHAT compute $M(M-1)/2$ real N-sample FFTs using the FFTW C library [Frigo et Johnson, 1998], and SVD-MUSIC evaluates $N/2$ SEVD of $M \times M$ matrices using the Eigen C++ library [Guennebaud et Jacob, 2014]. Some methods (e.g., [Nakadai *et coll.*, 2010b; Nakamura *et coll.*, 2011a, 2012]) compute SEVD only in the lower frequency range (where speech is usually observed) to reduce the computational load. However, this discards some useful spectral information in the higher frequencies (in speech fricative sounds for instance), which are considered with the SRP-PHAT method. To ensure a fair comparison, both methods treat the whole spectral range. SRP-PHAT requires from 201 ($M = 8$ and $N = 2048$) to 529 ($M = 32$ and $N = 512$) less computing time that the SEVD-MUSIC method. This suggests that SRP-PHAT is more suitable for online processing, as it performs Task 1 effectively. It is therefore desirable to use SRP-PHAT for Task 1, and optimize Task 2 to get an efficient SSL system.

Tableau 3.1 Processing time in sec/frame for SRP-PHAT and (SEVD-MUSIC)

M	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$
8	1.8E−4 (4.1E−2)	3.3E−4 (8.1E−2)	7.1E−4 (1.6E−1)	1.6E−3 (3.3E−1)
16	7.6E−4 (2.3E−1)	1.4E−3 (4.5E−1)	3.1E−3 (9.0E−1)	6.9E−3 (1.8E+0)
24	1.8E−3 (7.0E−1)	3.3E−3 (1.4E+0)	7.0E−3 (2.8E+0)	1.6E−2 (5.6E+0)
32	3.2E−3 (1.5E+0)	5.9E−3 (3.1E+0)	1.3E−2 (6.2E+0)	2.9E−2 (1.2E+1)

3.3 SRP-PHAT-HSDA Method

To understand how SRP-PHAT-HSDA works, let us start by explaining SRP-PHAT, to then explain the added particularities of SRP-PHAT-HSDA. Figure 3.1 shows the overview of the SRP-PHAT-HSDA method, using the M microphone signals to localize V potential sources. The Microphone Directivity module and the MSW Automatic Calibration module are used at initialization, and provide parameters to perform optimized GCC-PHAT, MSW filtering and Hierarchical Search online.

The underlying mechanism of SRP-PHAT is to search for V potential sources for each frame l over a discrete space [Grondin *et coll.*, 2013]. For each potential source, the com-

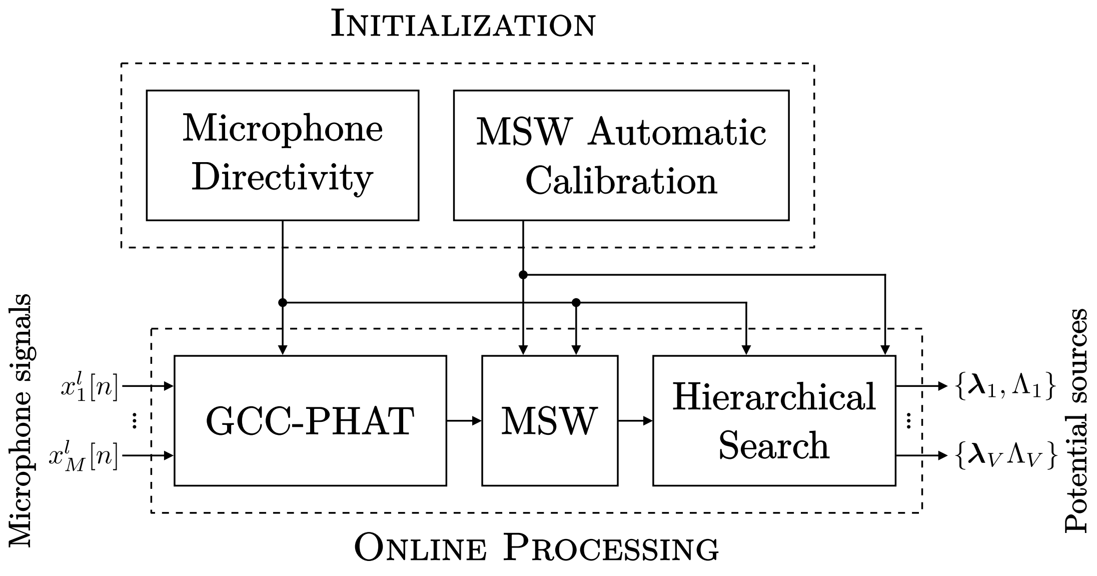


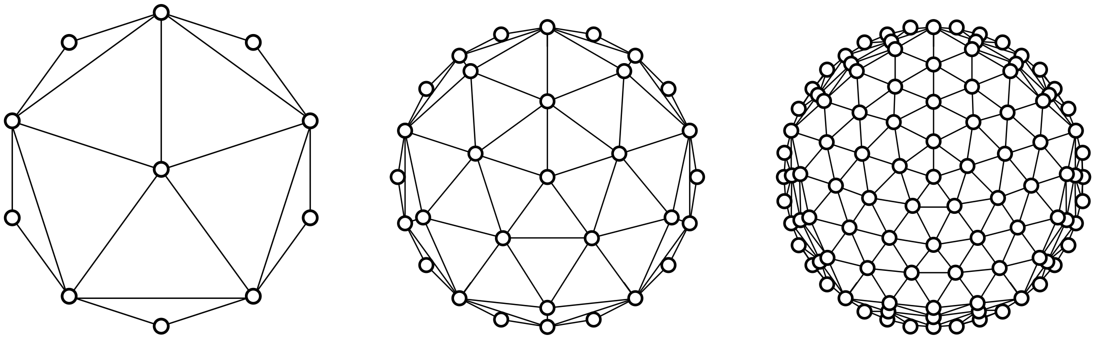
Figure 3.1 Block diagram of SRP-PHAT-HSDA

puted GCC-PHAT frames are filtered using a Maximum Sliding Windows (MSW). The sum of the filtered GCC-PHAT frames for all pairs of microphones provide the acoustic energy for each direction on the discrete space, and the direction with the maximum energy corresponds to a potential source. Once a potential source is obtained, its contribution is removed from the GCC-PHAT frames, and the space is scanned again. This process is repeated V times until the DoAs ($\boldsymbol{\lambda}_v, v = 1, \dots, V$) and energy levels ($\Lambda_v, v = 1, \dots, V$) of all potential sources are generated.

A discrete unit sphere provides potential DoAs for sound sources. As in [Grondin *et coll.*, 2013] and [Valin *et coll.*, 2007a], a regular convex icosahedron made of 12 points defines the initial discrete space, and is refined recursively $\mathcal{L}$ times until the desired space resolution is obtained. Figure 3.2 shows the regular icosahedron ($\mathcal{L} = 0$), and subsequent refining iterations levels ($\mathcal{L} = 1$ and $\mathcal{L} = 2$).

Each point on the discrete sphere corresponds to a unit vector $\mathbf{u}_k$, where k stands for the point index where $k = 1, 2, \dots, K$, and $S = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K\}$ is the set that contains all vectors, where the number of points $K = 10 \times 4^{\mathcal{L}} + 2$ depends on the resolution level $\mathcal{L}$. In the SRP-PHAT method proposed in [Valin *et coll.*, 2007a], the scan space is refined four times ($\mathcal{L} = 4$) to generate 2562 points and obtain a spatial resolution of 3 degrees.

To further reduce SRP-PHAT computations, and maintain a high localization accuracy regardless of the microphone array shape, SRP-PHAT-HSDA adds the following elements :



(a) $\mathcal{L} = 0$, $K = 12$ (b) $\mathcal{L} = 1$, $K = 42$ (c) $\mathcal{L} = 2$, $K = 162$

Figure 3.2 Discrete unit spheres

- *Microphone Directivity (MD)* : When the number of microphone M increases, the computational load also increases by a complexity of $\mathcal{O}(M^2)$. The proposed method assumes microphone have a directivity pattern, and this introduces constraints that reduces the space to be scanned and the number of pairs of microphones to use, which in turn decreases the amount of computations.
- *Maximum Sliding Window Automatic Calibration (MSWAC)* : TDOA estimation is influenced by the uncertainty in the speed of sound and the microphones positions (which may be difficult to measure precisely with microphone arrays of complex geometry), and scan grid discretization, which should be modelled somehow. The MSW size can be tuned manually by hand to maximize localization accuracy, but this remains a time consuming task which has to be repeated for each new microphone array geometry. The TDOA uncertainty model solves this challenge as it automatically tunes the MSW size to maximize localization accuracy.
- *Hierarchical Search (HS)* : Searching for potential sources involves scanning the 3D space according to a grid with a specific resolution. Finer resolution means better precision but higher computation. To reduce computations, a solution is to first do a scan with a grid at coarse resolution to identify a potential sound source, and then do another scan with a grid with a fine resolution using the location found during the first scan to pinpoint a more accurate direction.

3.3.1 Microphone Directivity

In a microphone array, microphones are usually assumed to be omnidirectional, i.e., acquiring signals with equal gain from all directions. In practice however, microphones on a robot platform are often mounted on a rigid body, which may block the direct propagation path between a sound source and a microphone. The attenuation is mostly due to diffraction, and changes as a function of frequency. Since the exact diffraction model is not available, the proposed model relies on simpler assumptions : 1) there is a unit gain for sound sources with a direct propagation path, and 2) the gain is null when the path is blocked by the robot body. As the signal to noise ratio is generally unknown for the blocked microphones, it is safer to assume a low SNR, and setting the gain to zero prevents noise to be injected in the observations. Moreover, the gain is set constant for all frequencies, and a smooth transition band connects the unit and null gain regions. This transition band prevents abrupt changes in gains when the sound source position varies. Figure 3.3 introduces $\theta(\mathbf{u},\mathbf{d})$, as defined by (3.7), the angle between a sound source located at $\mathbf{u}$, and the orientation of the microphone modeled by the unit vector $\mathbf{d}$.

$$\theta(\mathbf{u},\mathbf{d}) = \arccos\left[\frac{\mathbf{u}\cdot\mathbf{d}}{|\mathbf{u}||\mathbf{d}|}\right] \tag{3.7}$$

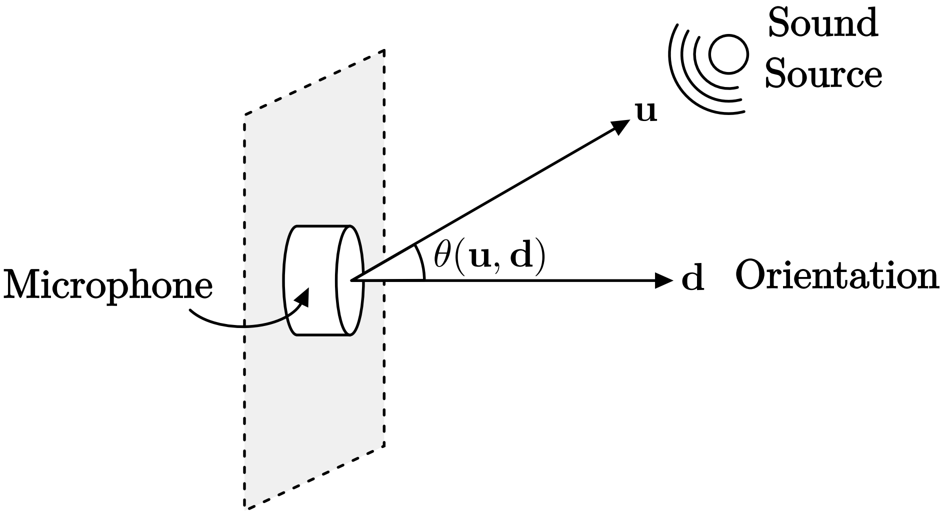


Figure 3.3 Microphone directivity angle θ as a function of microphone orientation and source direction

Figure 3.4 illustrates the logistic function that models the gain $G(\mathbf{u},\mathbf{D})$ as a function of the angle $\theta(\mathbf{u},\mathbf{d})$, given in (3.8). The expression $\mathbf{D}$ is a set that contains the parameters $\{\mathbf{d},\alpha,\beta\}$, where α stands for the angle where the gain is one while β corresponds to the angle at which the gain is null. The region between both angles can be viewed as a transition band.

$$G(\mathbf{u},\mathbf{D}) = \frac{1}{1+\exp\left(\left(\frac{20}{\beta-\alpha}\right)\left(\theta(\mathbf{u},\mathbf{d})-\frac{\alpha+\beta}{2}\right)\right)} \tag{3.8}$$

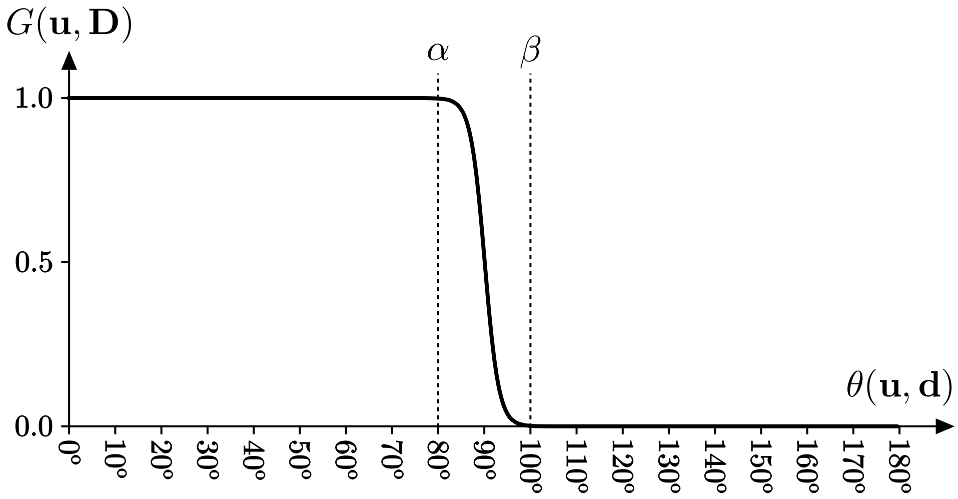


Figure 3.4 Microphone gain response

To make SSL more robust to reverberation, the scan space is restricted to a specific direction. For instance, the scan space is limited to the hemisphere that points to the ceiling to ignore reflections from the floor. The unit vector $\mathbf{d}_0$ stands for the orientation of the scan space.

Since microphone directivity introduces some constraints on the scan space, the spatial gains $G(\mathbf{u}_k, \mathbf{D}_p)$ and $G(\mathbf{u}_k, \mathbf{D}_q)$ need to be large enough for a source located in the direction $\mathbf{u}_k$ to excite both microphones p and q. The gain $G(\mathbf{u}_k, \mathbf{D}_0)$ also needs to be large enough for this direction to be part of the scan space. The mask $\zeta_{pq}(\mathbf{u}_k)$ models this condition, where the constant G_{min} stands for the minimal gain value :

$$\zeta_{pq}(\mathbf{u}_k) = \begin{cases} 1 & G(\mathbf{u}_k, \mathbf{D}_0)G(\mathbf{u}_k, \mathbf{D}_p)G(\mathbf{u}_k, \mathbf{D}_q) \geq G_{min} \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

When the mask $\zeta_{pq}(\mathbf{u}_k)$ is zero, the value of the corresponding sample in the GCC-PHAT frame is negligible and can be ignored. When all pairs of microphones are uncorrelated ($\zeta_{pq}(\mathbf{u}_k) = 0$ for all values of p and q), the direction $\mathbf{u}_k$ can simply be ignored ($\zeta(\mathbf{u}_k) = 0$) :

$$\zeta(\mathbf{u}_k) = \begin{cases} 1 & \sum_{p=1}^{M} \sum_{q=p+1}^{M} \zeta_{pq}(\mathbf{u}_k) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

Similarly, the GCC-PHAT between microphones p and q needs to be computed only when $\zeta_{pq} = 1$, that is when these microphones are excited simultaneously at least once for a given direction $\mathbf{u}_k$:

$$\zeta_{pq} = \begin{cases} 1 & \sum_{k=1}^{K} \zeta_{pq}(\mathbf{u}_k) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

3.3.2 MSW Automatic Calibration

The TDOA between two microphones $\mathbf{m}_p$ and $\mathbf{m}_q$ is given by the expression $\tau_{pq}(\mathbf{u})$. Under the far field assumption, the TDOA is set according to (3.12), where $\mathbf{u}$ represents the normalized direction of the sound source, f_S stands for the sample rate (in samples/sec) and c for the speed of sound (in m/s).

$$\tau_{pq}(\mathbf{u}) = \frac{f_S}{c}\left(\mathbf{m}_p - \mathbf{m}_q\right) \cdot \mathbf{u} \tag{3.12}$$

The speed of sound varies according to air temperature, humidity and pressure. These parameters usually lie within a known range in a room (and even outside), but it remains difficult to calculate the exact speed of sound. In SRP-PHAT-HSDA, the speed of sound is modeled using a random variable $c \sim \mathcal{N}(\mu_c, \sigma_c)$, where μ_c is the mean and σ_c the standard deviation of the normal distribution. The exact position of each microphone is also modeled by a trivariate normal distribution $\mathbf{m}_p \sim \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$, where $\boldsymbol{\mu}_p$ stands for the 1×3 mean vector and $\boldsymbol{\Sigma}_p$ for the 3×3 covariance matrix.

The first step consists in solving for the expression $a = f_S/c$ (in samples/m). To make calculations easier, a normally distributed random variable $\eta \sim \mathcal{N}(0, 1)$ is introduced :

$$a = \frac{f_S}{\mu_c + \sigma_c \eta} \tag{3.13}$$

The previous equation can be linearized given that $\mu_c \gg \sigma_c$. Expanding this function as a Taylor series, the following approximation holds :

$$a \approx \frac{f_S}{\mu_c}\left(1 - \frac{\sigma_c}{\mu_c}\eta\right) \tag{3.14}$$

This results in a being a normally distributed random variable, with mean μ_a and standard deviation σ_a.

The second step consists in solving the projection of the distance between both microphones represented by random variables $\mathbf{m}_p$ and $\mathbf{m}_q$, on the deterministic unit vector $\mathbf{u}$, represented below as $b_{pq}(\mathbf{u})$:

$$b_{pq}(\mathbf{u}) = (\mathbf{m}_p - \mathbf{m}_q) \cdot \mathbf{u} \tag{3.15}$$

The intermediate expression $(\mathbf{m}_p - \mathbf{m}_q)$ is a random variable with a normal distribution $\sim \mathcal{N}(\boldsymbol{\mu}_{pq}, \boldsymbol{\Sigma}_{pq})$, where $\boldsymbol{\mu}_{pq} = \boldsymbol{\mu}_p - \boldsymbol{\mu}_q$ and $\boldsymbol{\Sigma}_{pq} = \boldsymbol{\Sigma}_p + \boldsymbol{\Sigma}_q$. The position uncertainty is usually significantly smaller than the distance between both microphones, such that $\|\boldsymbol{\mu}_{pq}\|^2 \gg \|\boldsymbol{\Sigma}_{pq}\|$, where the expression $\|\dots\|$ stands for the vector and matrix norms. The random variable $b_{pq}(\mathbf{u})$ has a normal distribution :

$$b_{pq}(\mathbf{u}) = \mu_{b,pq}(\mathbf{u}) + \sigma_{b,pq}(\mathbf{u})\eta = \boldsymbol{\mu}_{pq} \cdot \mathbf{u} + \eta\sqrt{\mathbf{u}^T\boldsymbol{\Sigma}_{pq}\mathbf{u}} \tag{3.16}$$

The random variable $\tau_{pq}(\mathbf{u})$ is the product of the normal random variables a and $b_{pq}(\mathbf{u})$, which gives the following expression :

$$\tau_{pq}(\mathbf{u}) = (\mu_a + \sigma_a\eta_a)(\mu_{b,pq}(\mathbf{u}) + \sigma_{b,pq}(\mathbf{u})\eta_b) \tag{3.17}$$

where η_a and η_b are two independent random variables with standard normal distribution. Since $\mu_a \gg \sigma_a$, and $\mu_{b,pq}(\mathbf{u}) \gg \sigma_{b,pq}(\mathbf{u})$ (for all p and q), the following approximation holds :

$$\tau_{pq}(\mathbf{u}) \approx \mu_a\mu_{b,pq}(\mathbf{u}) + \mu_a\sigma_{b,pq}(\mathbf{u})\eta_b + \mu_{b,pq}(\mathbf{u})\sigma_a\eta_a \tag{3.18}$$

The random variable $\tau_{pq}(\mathbf{u})$ therefore exhibits a normal distribution $\sim \mathcal{N}(\mu_{\tau,pq}(\mathbf{u}), \sigma_{\tau,pq}(\mathbf{u}))$ where :

$$\mu_{\tau,pq}(\mathbf{u}) = \left(\frac{f_S}{\mu_c}\right)(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q) \cdot \mathbf{u} \tag{3.19}$$

$$\sigma_{\tau,pq}(\mathbf{u}) = \sqrt{\frac{f_S^2}{\mu_c^2}\mathbf{u}^T(\boldsymbol{\Sigma}_p + \boldsymbol{\Sigma}_q)\mathbf{u} + [(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q) \cdot \mathbf{u}]^2\frac{f_S^2\sigma_c^2}{\mu_c^4}} \tag{3.20}$$

This models the TDOA estimation uncertainty, and is used to configure MSW size. In practice, GCC-PHAT generates frames with discrete indexes, and therefore the estimated TDOA value (denoted by $\hat{\tau}_{pq}(\mathbf{u}_k)$) for each discrete direction $\mathbf{u}_k$ is usually rounded to the closest integer :

$$\hat{\tau}_{pq}(\mathbf{u}_k) = \left\lfloor \left(\frac{f_S}{\mu_c}\right)(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q) \cdot \mathbf{u}_k \right\rceil \tag{3.21}$$

To cope with the disparity between $\hat{\tau}_{pq}(\mathbf{u}_k)$ and the observation of the random variable $\tau_{pq}(\mathbf{u})$, a MSW filters each GCC-PHAT frame for all pairs of microphones, where $\hat{r}^l_{pq}[n]$ stands for the filtered frame. The MSW has a size of $2\Delta_{pq}+1$ samples (the frame index l is omitted here for clarity) :

$$\hat{r}_{pq}[n] = \max\left\{r_{pq}[n-\Delta\tau_{pq}], \ldots, r_{pq}[n+\Delta\tau_{pq}]\right\} \tag{3.22}$$

Figure 3.5 illustrates how the partial area under the probability density function (PDF) of $\tau_{pq}(\mathbf{u})$ stands for the probability the MSW captures the TDOA value.

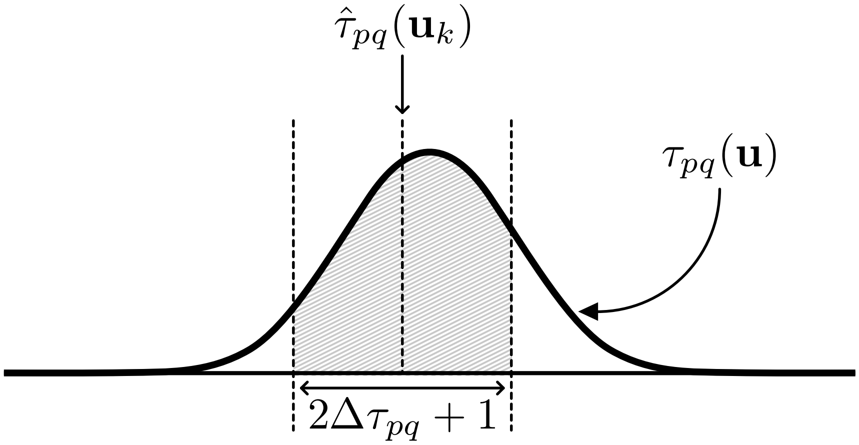


Figure 3.5 MSW and PDF of the TDOA random variable

The area under the curve corresponds to the integral of the PDF that lies within the interval of the MSW, and is defined as the probability $\tau_{pq}(\mathbf{u})$ is observed given $\hat{\tau}_{pq}(\mathbf{u}_k)$ and $\Delta\tau_{pq}$:

$$p(\tau_{pq}(\mathbf{u})|\hat{\tau}_{pq}(\mathbf{u}_k), \Delta\tau_{pq}) = \int_{\hat{\tau}_{pq}(\mathbf{u}_k)-\Delta\tau_{pq}-0.5}^{\hat{\tau}_{pq}(\mathbf{u}_k)+\Delta\tau_{pq}+0.5} f(\tau|\tau_{pq}(\mathbf{u}))d\tau \tag{3.23}$$

where the function $f(\tau|\tau_{pq}(\mathbf{u}))$ is defined as :

$$f(\tau|\tau_{pq}(\mathbf{u})) = \mathcal{N}(\tau|\mu_{\tau,pq}(\mathbf{u}), \sigma_{\tau,pq}(\mathbf{u})) \tag{3.24}$$

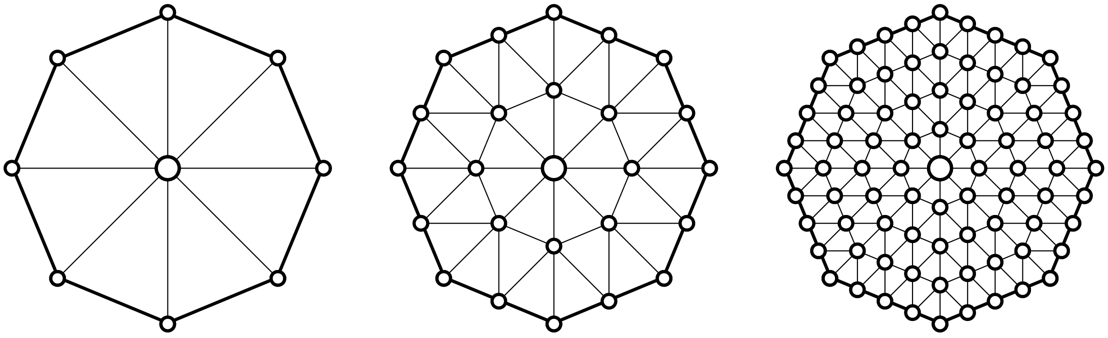
A surface integral around the discrete direction $\mathbf{u}_k$ gives the probability that the MSW captures a source in a neighboring direction :

$$p(\tau_{pq}|\hat{\tau}_{pq}(\mathbf{u}_k), \Delta\tau_{pq}) = \iint_A p(\tau_{pq}(\mathbf{u})|\hat{\tau}_{pq}(\mathbf{u}_k), \Delta\tau_{pq})dA \tag{3.25}$$

Since there is no closed expression for this surface integral, a discrete integration over space is used to estimate the probability probability that the MSW captures a source :

$$p(\tau_{pq}|\hat{\tau}_{pq}(\mathbf{u}_k), \Delta\tau_{pq}) \approx \sum_{e=1}^{E} \frac{p(\tau_{pq}(\mathbf{v}_{k,e})|\hat{\tau}_{pq}(\mathbf{u}_k), \Delta\tau_{pq})}{E} \tag{3.26}$$

An octagon made of $E = 4(2^{\mathcal{D}} + 4^{\mathcal{D}}) + 1$ points estimates the discretized surface, where $\mathcal{D}$ stands for the number of recursive iterations. The radius of the octagon corresponds to the distance between $\mathbf{u}_k$ and its closest neighbor. Figure 3.6 shows octagons for $\mathcal{D} = 0$, 1 and 2 iterations :



(a) $\mathcal{D} = 0$, $E = 9$ (b) $\mathcal{D} = 1$, $E = 25$ (c) $\mathcal{D} = 2$, $E = 81$

Figure 3.6 Discrete octogons

For all directions in the set $S = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K\}$, the probability that the MSW captures sources in neighboring directions is estimated as follows :

$$p(\tau_{pq}|\hat{\tau}_{pq}(S), \Delta\tau_{pq}) \approx \sum_{k=1}^{K} \sum_{e=1}^{E} \frac{p(\tau_{pq}(\mathbf{v}_{k,e})|\hat{\tau}_{pq}(\mathbf{u}_k), \Delta\tau_{pq})}{KE} \tag{3.27}$$

For a given discrete direction $\mathbf{u}_k$, the probability that the discrete point $\mathbf{v}_{k,e}$ is captured for all pairs of microphones is estimated with the following expression :

$$p(\mathbf{v}_{k,e}|\mathbf{u}_k) \approx \sum_{p=1}^{M} \sum_{q=p+1}^{M} \frac{p(\tau_{pq}(\mathbf{v}_{k,e})|\hat{\tau}_{pq}(\mathbf{u}_k), \Delta\tau_{pq})}{M(M-1)} \tag{3.28}$$

The objective is to maximize $p(\mathbf{v}_{k,e}|\mathbf{u}_k)$ for all directions $\mathbf{u}_k$ and discrete points $\mathbf{v}_{k,e}$, while keeping the MSW window size as small as possible to preserve the localization accuracy. To achieve this, Algorithm 1 increments the parameters $\Delta\tau_{pq}$ progressively until the threshold C_{min} is reached. This calibration is performed once at initialization.

Algorithm 1 MSW Automatic Calibration

```
for all pairs $pq$ do
    $\Delta\tau_{pq} \leftarrow 0$
    Compute $p(\tau_{pq}|\hat{\tau}_{pq}(S), \Delta\tau_{pq})$
end for
Compute $p(\mathbf{v}_{k,e}|\mathbf{u}_k)$ for all $k$ and $e$
while $\min_{k,e} \{p(\mathbf{v}_{k,e}|\mathbf{u}_k)\} < C_{min}$ do
    $(pq)^* \leftarrow \arg\min_{pq} \{p(\tau_{pq}|\hat{\tau}_{pq}(S), \Delta\tau_{pq})\}$
    $\Delta\tau_{(pq)^*} \leftarrow \Delta\tau_{(pq)^*} + 1$
    Compute $p(\tau_{(pq)^*}|\hat{\tau}_{(pq)^*}(S), \Delta\tau_{(pq)^*})$
    Compute $p(\mathbf{v}_{k,e}|\mathbf{u}_k)$ for all $k$ and $e$
end while
```

3.3.3 Hierarchical Search

Hierarchical search involves two discrete grids : one with a coarse resolution and the other with a fine resolution. A matching matrix connects the coarse and fine resolution spaces (denoted by S' and S'', respectively). This $K' \times K''$ matrix, denoted by the variable $\mathcal{M}$, connects each direction from the fine resolution grid (composed of K'' directions) to many directions in the coarse resolution grid (made of K' directions).

The similitude between a direction $\mathbf{u}'_c$ in S' and a direction $\mathbf{u}''_f$ in S'' is given by $\delta_{pq}(c, f)$, which is the intersection between subsets $I'_{pq}(\mathbf{u}'_c)$ and $I''_{pq}(\mathbf{u}''_f)$:

$$\delta_{pq}(c, f) = I'_{pq}(\mathbf{u}'_c) \cap I''_{pq}(\mathbf{u}''_f) \tag{3.29}$$

where :

$$I'_{pq}(\mathbf{u}'_c) = [\hat{\tau}_{pq}(\mathbf{u}'_c) - \Delta\tau'_{pq}, \hat{\tau}_{pq}(\mathbf{u}'_c) + \Delta\tau'_{pq}] \tag{3.30}$$

$$I''_{pq}(\mathbf{u}''_f) = [\hat{\tau}_{pq}(\mathbf{u}''_f) - \Delta\tau''_{pq}, \hat{\tau}_{pq}(\mathbf{u}''_f) + \Delta\tau''_{pq}] \tag{3.31}$$

The expressions $\Delta\tau'_{pq}$ and $\Delta\tau''_{pq}$ depend on the window size of the MSWs, computed for the coarse and fine resolutions grids, for each pair of microphones pq. Each direction $\mathbf{u}''_f$ in the fine resolution grid is mapped to the U most similar directions in the coarse resolution grid, derived using Algorithm 2.

The matching matrix provides a mean to connect at initialization the coarse and fine resolution grids, which are then used to perform the hierarchical search. Algorithm 3 first

Algorithm 2 Hierarchical Search Matching Process

```
for $f = 1$ to $K''$ do
    for $c = 1$ to $K'$ do
        $\mathcal{M}(c, f) \leftarrow 0$
        $\mathcal{V}(c) \leftarrow 0$
        for all pairs $pq$ do
            if $\zeta_{pq}(\mathbf{u}'_c) = 1$ and $\zeta_{pq}(\mathbf{u}''_f) = 1$ then
                $\mathcal{V}(c) \leftarrow \mathcal{V}(c) + \delta_{pq}(c, f)$
            end if
        end for
    end for
    for $u = 1$ to $U$ do
        $c^* \leftarrow \arg\max_c \mathcal{V}(c)$
        $\mathcal{M}(c^*, f) \leftarrow 1$
        $\mathcal{V}(c^*) \leftarrow 0$
    end for
end for
```

performs a scan using the coarse resolution grid, and then a second scan over a region of the fine resolution grid to improve accuracy. The expressions $\hat{r}'_{pq}$ and $\hat{r}''_{pq}$ stand for the GCC-PHAT frames at pair pq filtered by the MSW for the coarse and fine resolutions grids, respectively. To consider the microphone directivity in the scanning process, the GCC-PHAT result for each pair pq and directions $\mathbf{u}'_c$ or $\mathbf{u}''_f$ is summed only when the binary masks $\zeta_{pq}(\mathbf{u}'_c)$ or $\zeta_{pq}(\mathbf{u}''_f)$ are set to 1. The energy levels (defined by the expressions $\mathcal{E}'$ and $\mathcal{E}''$) are normalized with the number of active pairs for each direction (expressed by $\mathcal{T}$). The variable ϵ is set to a small value to avoid division by zero. Scanning for the v potential source returns the DoA $\boldsymbol{\lambda}_v = \mathbf{u}''_{f^*}$ and the corresponding energy level $\Lambda_v = \mathcal{E}''(f^*)$. The proposed Hierarchical Search involves $K'+K''U/K'$ directions to scan in average, compared with K'' directions for a fixed grid with the same resolution. For instance, with $\mathcal{L}' = 2$, $\mathcal{L}'' = 4$ and $U = 10$ for SRP-PHAT-HSDA, and $\mathcal{L} = 4$ for SRP-PHAT, there are in average 320 directions to scan, instead of 2562.

3.4 Experimental Setup

Experiments involve two 16-microphone array configurations installed on a mobile robot : 1) an opened microphone array (OMA) with microphones placed on a circular plane, and 2) a close microphone array (CMA) with microphones placed on a cubic structure. Figure 3.7 shows these two configurations. The microphones in green are used for experiments that involve only 8 microphones, while both green and orange microphones are considered

Algorithm 3 Hierarchical Search Scanning Process

```
for $c = 1$ to $K'$ do
    $\mathcal{E}'(c) \leftarrow 0$, $\mathcal{T} \leftarrow 0$
    for all pairs $pq$ do
        if $\zeta_{pq}(\mathbf{u}'_c) = 1$ then
            $\mathcal{E}'(c) \leftarrow \mathcal{E}'(c) + \hat{r}'_{pq}[\hat{\tau}_{pq}(\mathbf{u}'_c)]$
            $\mathcal{T} \leftarrow \mathcal{T} + 1$
        end if
    end for
    $\mathcal{E}'(c) \leftarrow \mathcal{E}'(c)/(\mathcal{T} + \epsilon)$
end for
$c^* \leftarrow \arg\max_c \mathcal{E}'(c)$
for $f = 1$ to $K''$ do
    $\mathcal{E}''(f) \leftarrow 0$, $\mathcal{T} \leftarrow 0$
    if $\mathcal{M}(c^*, f) = 1$ then
        for all pairs $pq$ do
            if $\zeta_{pq}(\mathbf{u}''_f) = 1$ then
                $\mathcal{E}''(f) \leftarrow \mathcal{E}''(f) + \hat{r}''_{pq}[\hat{\tau}_{pq}(\mathbf{u}''_f)]$
                $\mathcal{T} \leftarrow \mathcal{T} + 1$
            end if
        end for
        $\mathcal{E}''(f) \leftarrow \mathcal{E}''(f)/(\mathcal{T} + \epsilon)$
    end if
end for
$f^* \leftarrow \arg\max_f \mathcal{E}''(f)$
return $\{\mathbf{u}''_{f^*}, \mathcal{E}''(f^*)\}$
```

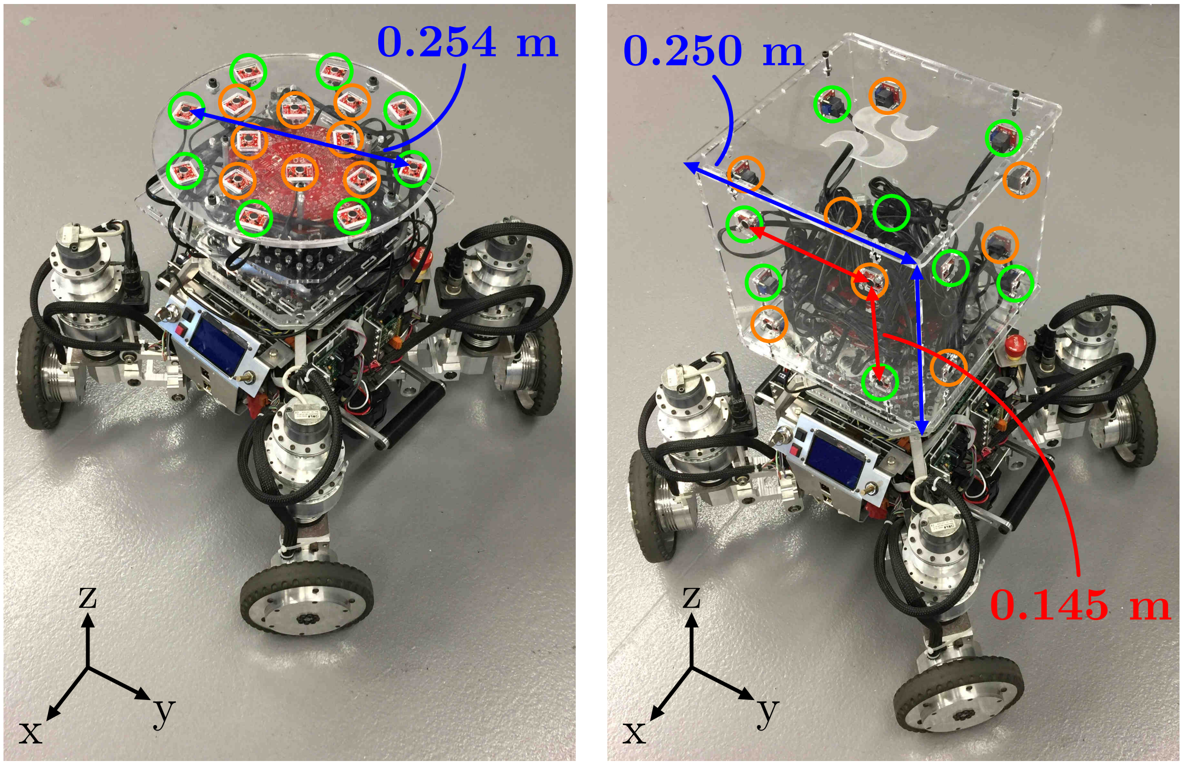
for experiments with 16 microphones. The OMA configuration consists of a circular surface with a diameter of 0.254 m, while the CMA configuration involves a cubic structure with 0.250 m edges, where microphones form many squares with 0.145 m edges. With the OMA configuration, all microphones have a direct path with the sound sources around the robot, while with the CMA sounds may be blocked by the cubic structure.

A diagonal covariance matrix models the uncertainty of microphone positions :

$$\mathbf{\Sigma}_p = \begin{bmatrix} (\sigma_{xx})_p & 0 & 0 \\ 0 & (\sigma_{yy})_p & 0 \\ 0 & 0 & (\sigma_{zz})_p \end{bmatrix} \tag{3.32}$$

and is set according to the microphone array configuration :

- For OMA, the variance $(\sigma_{zz})_p$ in the z-direction is set to zero, while the variances $(\sigma_{xx})_p$ and $(\sigma_{yy})_p$ in dimensions x and y are equal to σ^2_{mic}. All microphones point



(a) OMA

(b) CMA

Figure 3.7 16-microphone array configurations

to the ceiling, and therefore the direction unit vector $\mathbf{d}_p$ is oriented in the positive z-axis for all microphones.

- For CMA, all microphones point outwards the cube, and therefore the direction unit vector $\mathbf{d}_p$ is oriented in the positive or negative x and y-axes. The variances $(\sigma_{xx})_p$, $(\sigma_{yy})_p$ and $(\sigma_{zz})_p$ are set respectively to σ^2_{mic} if the microphone lies on a face in the plane that spans the corresponding x-, y- or z-axis, or are set to 0 otherwise.

Table 3.2 lists SRP-PHAT-HSDA parameters used for the experiments. The frame sizes N correspond to a duration of 16 msec (N/f_S) to match speech stationarity. The hop size ΔN is set to provide a 50% overlap between frames. The refining level is set to $\mathcal{D} = 1$, which ensures a reliable integration and maintains the memory usage and execution time as low as possible during initialization. We set $V = 4$ to detect up to four simultaneously active sound sources. The parameter σ^2_{mic} is chosen to model the uncertainty introduced by the membrane area of all microphones. The minimum gain G_{min} gets a value close to zero to generate the appropriate masks to limit the search space. The mean and standard deviation μ_c and σ_c are set to model the speed of sound in typical indoor and outdoor conditions. The minimum probability threshold C_{min} is chosen to ensure a good coverage of the random distribution of the TDOAs, while keeping the resolution high. The scan space directivity $\mathbf{d}_0$ points to the ceiling to remove reflections from the floor, which corresponding values of α_0 and β_0 to keep only the top hemisphere. The number of links U between the directions in the fine and coarse resolution grids is chosen to ensure an effective mapping while minimizing the number of scanned directions. The resolution levels of the coarse ($\mathcal{L}'$) and fine ($\mathcal{L}''$) grids are set to minimize the number of lookups while maintaining a good resolution. The parameters α_p and β_p are chosen to ensure a smooth directivity response for all microphones that neglects signals coming from behind the microphones.

Tableau 3.2 SRP-PHAT-HSDA parameters

Parameter	Value	Parameter	Value
f_S (samples/sec)	16000	μ_c (m/s)	343.0
N (samples)	256	σ_c	5.0
ΔN (samples)	128	C_{min}	0.3
$\mathcal{D}$	1	$\mathbf{d}_0$	[0 0 1]
U	10	$\mathcal{L}'$	2
V	4	$\mathcal{L}''$	4
α_p ($p > 0$)	80°	α_0	80°
β_p ($p > 0$)	100°	β_0	90°
G_{min}	0.1	σ^2_{mic}	1E−6

3.5 Results

Localization accuracy is computed with the proposed method to validate the MSW Automatic Calibration method. The CPU usage for a single core between SRP-PHAT and the proposed SRP-PHAT-HSDA is measured to compare computational load on a Raspberry Pi 3 (equipped with a ARM Cortex-A53 Quad-Core processor clocked at 1.2GHz). Performance with multiple speech sources around the robot is also evaluated. Both methods are implemented with the ODAS framework (Open embeddeD Audition System) in C language (without Neon/SSE optimization), which is available online as open source [1].

3.5.1 Localization Accuracy

To evaluate localization accuracy, the robot is installed in the middle of a large room and a loudspeaker is positioned $r = 3$ m away at an height of $h = 1.15$ m referenced to origin of the microphone array, at azimuths of $\phi = 0°, 10°, \ldots, 350°$, for a total of 36 positions. For each position, a white noise signal plays in the loudspeaker for 2 sec. The recorded signals

[1] http ://github.com/introlab/odas

are then mixed to generate two active sources at different azimuths. Figure 3.8 illustrates an example with one source recorded at $\phi_1 = 30°$, and the other recorded at $\phi_2 = 120°$. All permutations of $\{\phi_1, \phi_2\}$, where $\phi_1 \neq \phi_2$, $(\{0°, 10°\}, \{0°, 20°\}, \dots, \{350°, 340°\})$ are investigated, for a total of 1260 permutations.

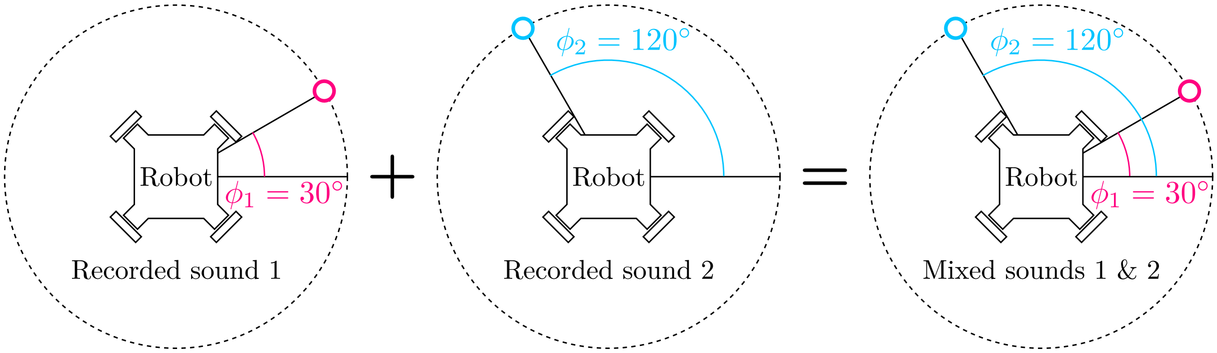


Figure 3.8 Experimental setup example with $\phi_1 = 30°$ and $\phi_2 = 120°$

The expression $\boldsymbol{\gamma}(\phi)$ stands for the DoA in Cartesian coordinates that corresponds to azimuth ϕ :

$$\boldsymbol{\gamma}(\phi) = \frac{1}{\sqrt{r^2 + h^2}} \begin{bmatrix} r\cos(\phi) & r\sin(\phi) & h \end{bmatrix} \tag{3.33}$$

The Root Mean Square Error (RMSE) corresponds to the smallest distance between potential source v and both theoretical DoAs at angles ϕ_1 and ϕ_2 :

$$\text{RMSE}_v = \min\{\|\boldsymbol{\lambda}_v - \boldsymbol{\gamma}(\phi_1)\|, \|\boldsymbol{\lambda}_v - \boldsymbol{\gamma}(\phi_2)\|\} \tag{3.34}$$

The average RMSE for both potential sources $v = 1$ and $v = 2$ provide insight regarding localization accuracy :

$$\text{RMSE} = \frac{\text{RMSE}_1 + \text{RMSE}_2}{2} \tag{3.35}$$

Table 3.3 presents the RMSE results with 16 microphones. The SRP-PHAT method for a single grid with refining level $\mathcal{L} = 1, 2, 3, 4$ performs localization with fixed values for the size of MSW ($\Delta\tau_{pq} = 0, 1, 2, 3$), and is compared with the SRP-PHAT-HSDA method which automatically calibrates $\Delta\tau_{pq}$. Moreover, Hierarchical Search using two grids with different refining levels $\mathcal{L} = \{2, 4\}$ (which means $\mathcal{L}' = 2$ and $\mathcal{L}'' = 4$) is also examined, for fixed and automatically selected values of $\Delta\tau_{pq}$. The RMSE values in bold stand for the smallest values across $\Delta\tau_{pq} = 0, 1, 2, 3$ and HSDA, for a given refining level.

Results suggest that when set to a constant, the ideal value of $\Delta\tau_{pq}$ changes according to the grid resolution level $\mathcal{L}$. For the OMA configuration, setting automatically the value

Tableau 3.3 SSL RMSE with two active sound sources

Config	$\Delta\tau_{pq}$	$\mathcal{L}$				
		1	2	3	4	$\{2,4\}$
OMA	0	0.279	**0.140**	**0.081**	**0.064**	**0.064**
	1	**0.231**	0.186	0.208	0.222	0.222
	2	0.311	0.349	0.375	0.384	0.385
	3	0.501	0.559	0.584	0.596	0.596
	HSDA	**0.231**	0.147	**0.081**	**0.064**	**0.064**
CMA	0	0.528	**0.282**	**0.153**	**0.128**	**0.165**
	1	**0.414**	0.286	0.272	0.273	0.285
	2	0.477	0.448	0.454	0.456	0.454
	3	0.599	0.624	0.649	0.657	0.649
	HSDA	**0.306**	**0.197**	**0.117**	**0.094**	**0.105**

of $\Delta\tau_{pq}$ leads to the same accuracy as when the best constant value of $\Delta\tau_{pq}$ is chosen except for the case when $\mathcal{L} = 2$, where the RMSE is slightly higher, yet almost equal. This occurs when the size of the MSW is overestimated. For the CMA configuration, the automatic calibration leads to better accuracy compared to constant values chosen empirically. Finally, Hierarchical Search ($\mathcal{L} = \{2, 4\}$) provides the same accuracy as the high resolution ($\mathcal{L} = 4$) grid with OMA, and increases the RMSE marginally with CMA. It is possible to improve the accuracy of Hierarchical Search such that it matches the fixed grid with $\mathcal{L} = 4$ by increasing the parameters U, at the cost of increasing slightly the computational load.

3.5.2 Computational Load

Figure 3.9 shows the CPU usage for a single core on the Raspberry Pi 3. Results demonstrate that SRP-PHAT-HSDA reduces considerably computational load. CPU usage reduces by a factor of four for the CMA configuration with 16 microphones. The SRP-PHAT-HSDA uses less computations with CMA than OMA, which is due to the microphone directivity model that disregards the non significant pairs of microphones. SRP-PHAT-HSDA is capable of online performance with 16 microphones, while the previous SRP-PHAT method can no longer run online past 12 microphones, as the CPU usage exceeds 100% (106% and 105% with 13 microphones for CMA and OMA, respectively).

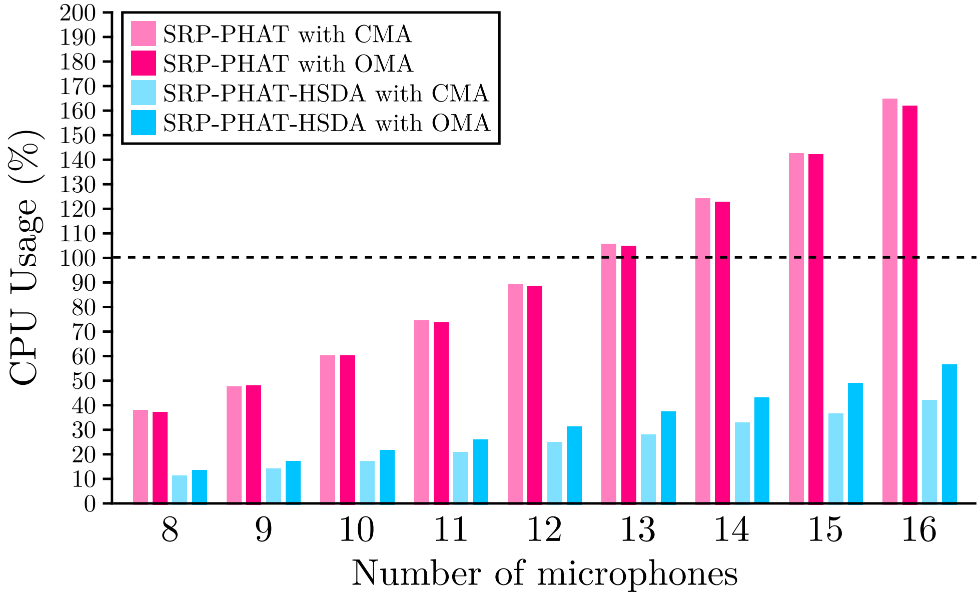
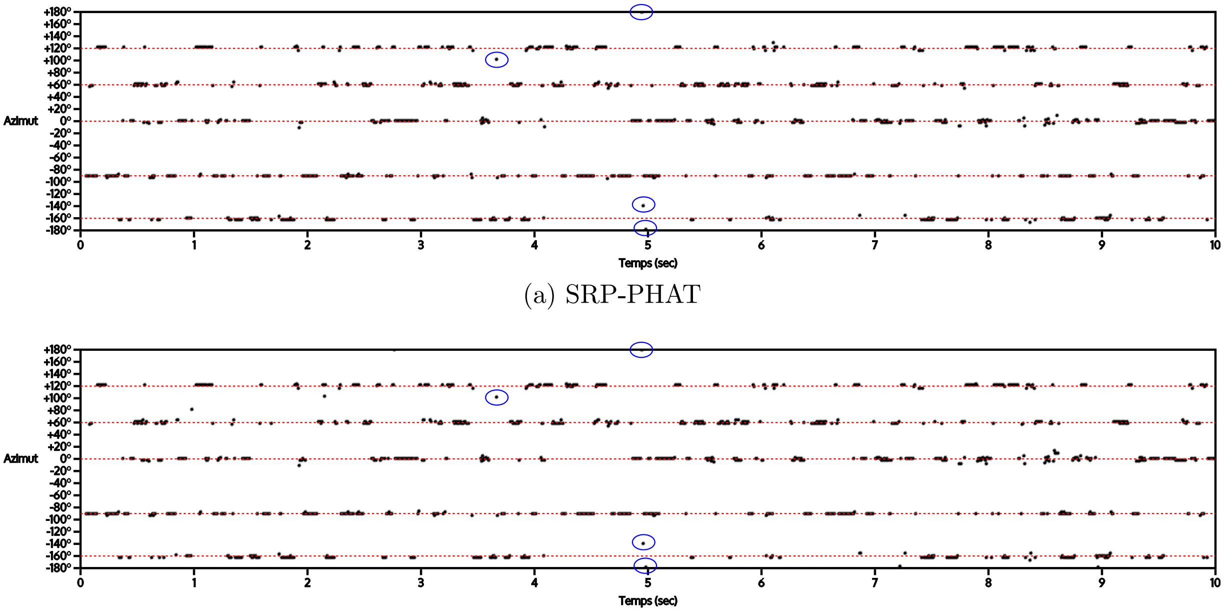


Figure 3.9 CPU Usage on a single core on a Raspberry Pi 3

3.5.3 Localization of Multiple Speech Sources

Five speech sources are played in loudspeakers located at azimuths $\phi_1 = 0°$, $\phi_2 = 60°$, $\phi_3 = 120°$, $\phi_4 = 200°$ and $\phi_5 = 270°$. Sources 1, 3 and 5 are male speakers while sources 2 and 4 are female speakers. They all speak continuously during 10 seconds. Since potential sources are significant when the energy level Λ_v is high, only the most energetic potential sources (25% of all potential sources which have the highest values of Λ_v) are plotted in Fig. 3.10 and Fig. 3.11, for the OMA configuration with 8 and 16 microphones, respectively, and in Fig. 3.12 and Fig. 3.13, for the CMA configuration with 8 and 16 microphones, respectively.

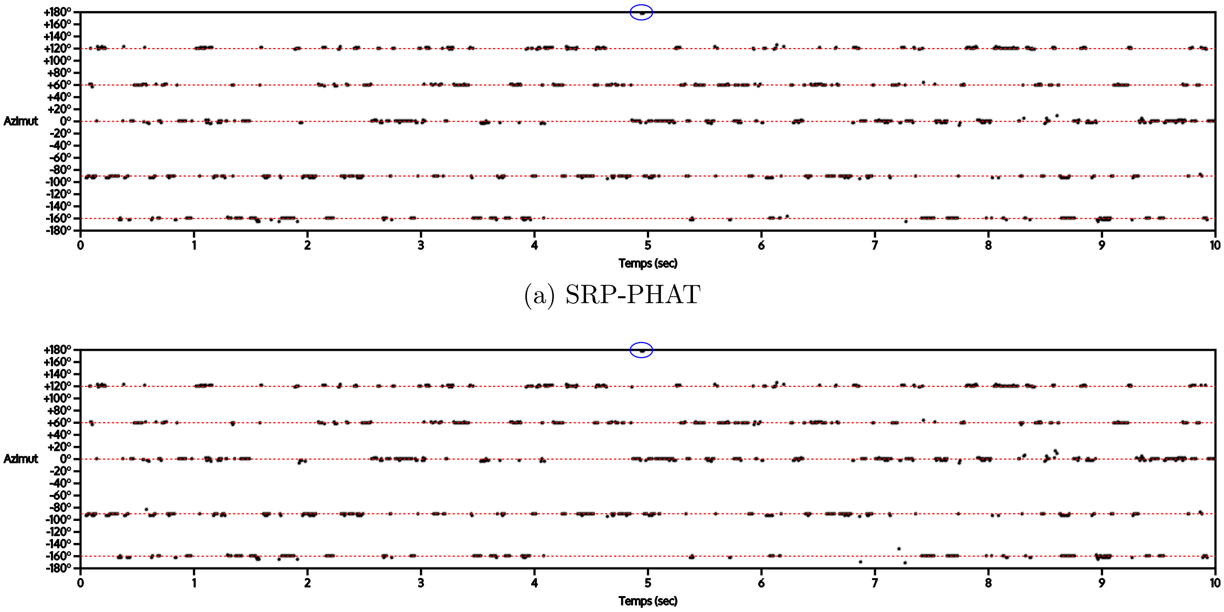
Results show that for the OMA configuration, both SRP-PHAT and SRP-PHAT-HSDA perform similarly : with both methods, there are four false detections with 8 microphones between 3 sec and 5 sec, and one false detection with 16 microphones at 5 sec. On the other hand, the SRP-PHAT-HSDA outperforms the SRP-PHAT method with the CMA configuration : there are numerous false detections with SRP-PHAT for 8 microphones, while there are less with SRP-PHAT-HSDA, and there are many false detections with SRP-PHAT with 16 microphones, while there are none with the proposed SRP-PHAT-HSDA. This robustness is due to the Microphone Directivity model that exploits the direct path of sound propagation with closed microphone array shapes. This suggests that the HSDA method should be used with CMA configurations to reduce false detections.



(a) SRP-PHAT

(b) SRP-PHAT-HSDA

Figure 3.10 Azimuths obtained with the OMA configuration for five speech sources and 8 microphones (true azimuths are plotted in red, and false detections are circled in blue)



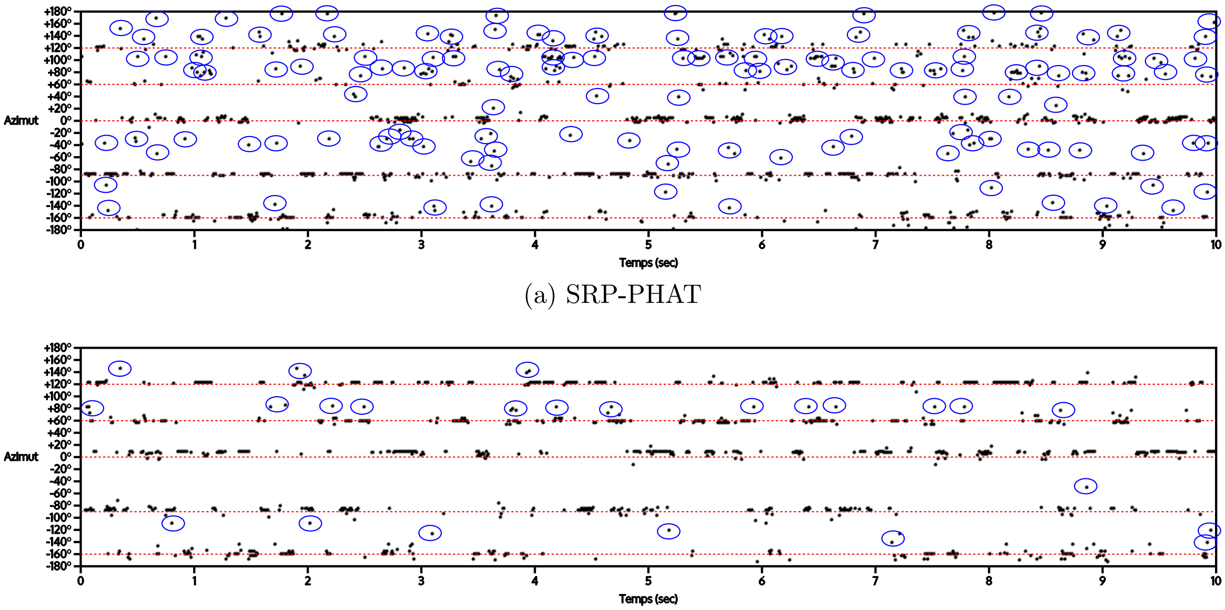
(a) SRP-PHAT

(b) SRP-PHAT-HSDA

Figure 3.11 Azimuths obtained with the OMA configuration for five speech sources and 16 microphones (true azimuths are plotted in red, and false detections are circled in blue)

3.6 Conclusion

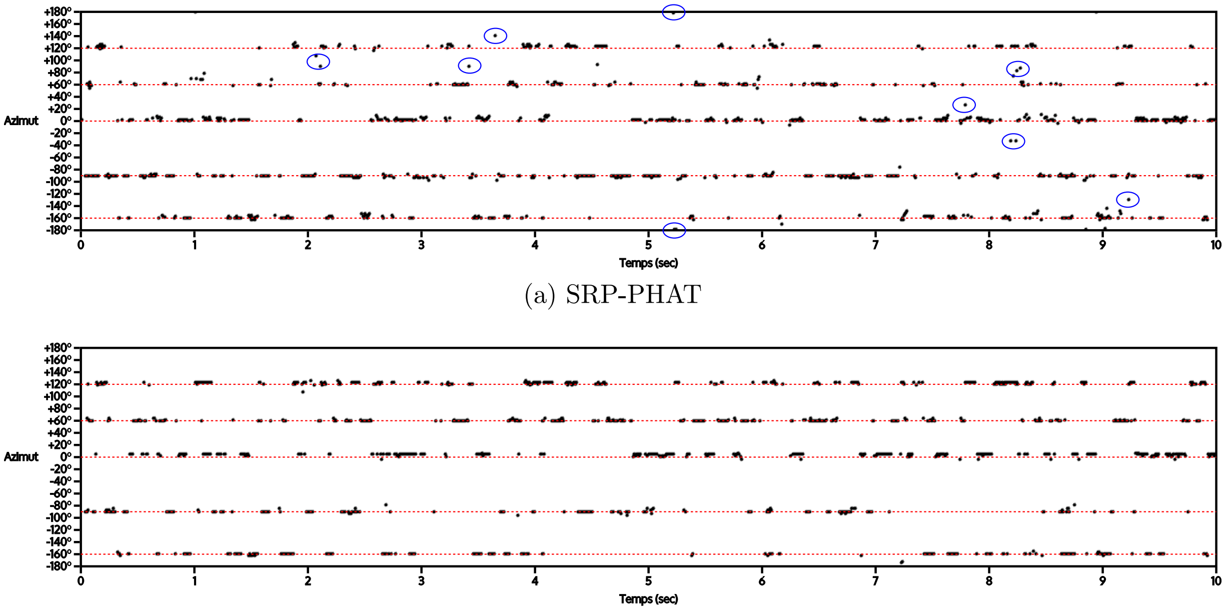
This paper introduces a novel SSL method optimizing SRP-PHAT to minimize computations and ensure a high localization accuracy without relying on manual tuning of parame-



(a) SRP-PHAT

(b) SRP-PHAT-HSDA

Figure 3.12 Azimuths obtained with the CMA configuration for five speech sources and 8 microphones (true azimuths are plotted in red, and false detections are circled in blue)



(a) SRP-PHAT

(b) SRP-PHAT-HSDA

Figure 3.13 Azimuths obtained with the CMA configuration for five speech sources and 8 microphones (true azimuths are plotted in red, and false detections are circled in blue)

ters. SRP-PHAT-HSDA scans the 3D space more efficiently using two grids of coarse and fine resolution. A microphone directivity model also reduces the amount of computations and reduce false detections with Closed Microphone Array (CMA) configurations. The TDOA uncertainty model optimizes the MSW sizes according to the array geometry and the uncertainties in the speed of sound and the positions of the microphones. Using SRP-PHAT-HSDA should provide clear benefit when combined with sound source tracking and separation methods to provide a complete approach for distant speech recognition and processing.

In future work, the SRP-PHAT-HSDA method could include a model that optimize the Maximum Sliding Window size for each individual point to scan and pair of microphones (instead of only each pair of microphones as it is currently the case). The next step is to include sound source tracking and separation to implement a complete pre-filtering system for distant speech recognition.

Acknowledgment

The authors would like to thank Dominic Létourneau, Cédric Godin and Vincent-Philippe Rhéaume for their help in the experimental setup.

CHAPITRE 4

SUIVI DE SOURCES SONORES

Avant-propos

Titre : *Lightweight Simultaneous 3D Sound Source Tracking Using Kalman Filters for Mobile Robots*

Auteurs et affiliations :

François Grondin : étudiant au doctorat, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique

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

Date de soumission : 12 octobre 2017

État : Soumis

Revue : *IEEE Transactions on Robotics*

Titre français : Suivi de sources sonores en 3D à l'aide de filtres de Kalman pour robots mobiles

Contribution au document : Cet article contribue à la thèse en décrivant une nouvelle méthode de suivi de sources sonores multiples en 3D qui introduit des filtres de Kalman afin de minimiser la charge de calculs.

Résumé en français : L'audition en robotique doit composer avec les phénomènes qui caractérisent les ondes sonores, qui sont omnidirectionnelles, s'additionnent et sont parcimonieuses, ce qui rend le traitement plus difficile dans un environnement bruité. La localisation des sources sonores est nécessaire pour permettre à un robot mobile d'interagir avec son environnement, et le suivi permet de composer avec le mouvement du robot et des sources sonores. La localisation de sources sonores génère habituellement des résultats bruités, qui doivent être adoucis et filtrés par un système de suivi de sources. Cet article introduit une méthode inspirée du filtre de Kalman qui permet de suivre simultanément la direction 3D de plusieurs sources sonores. Cette méthode offre des performances de suivi

identiques ou supérieures au suivi avec filtre particulaire, et fonctionne jusqu'à 30 fois plus rapidement, ce qui permet une implémentation sur du matériel à faible coût.

Abstract

Robot audition requires to deal with the intrinsic nature of sound signals, which are omnidirectional, additive, instantaneous, sparse and sporadic, making processing difficult in noisy environments. Localizing sound sources is an important capability for mobile robots operating in real life settings, and the motion of the robot or the sound sources make tracking necessary to generate audio streams associated with sound sources without having to categorize them continuously. Sound source localization methods usually return noisy features that need to be smoothed and filtered by tracking the sound sources. This paper presents a Kalman-based method capable of simultaneously tracking in 3D the directions of sound sources. This method shows similar or better tracking performance compare to particle-based tracking and runs up to 30 times faster, which makes it ideal for implementation on low-cost embedded hardware.

4.1 Introduction

Sound Source Localization (SSL) provides direction of arrival (DoA) of one or many active sound sources around a microphone array [Grondin *et coll.*, 2013]. There are two main categories of SSL methods commonly used in robot audition : 1) Multiple Signal Classification (MUSIC), based on Standard Eigenvalue Decomposition (SEVD-MUSIC) [Nakadai *et coll.*, 2010b], Generalized Eigenvalue Decomposition (GEVD-MUSIC) [Nakamura *et coll.*, 2011a], or Generalized Singular Value Decomposition (GSVD-MUSIC) [Nakamura *et coll.*, 2012] ; and 2) Steered Response Power with Phase Transform (SRP-PHAT) [Grondin *et coll.*, 2013]. These SSL methods provide noisy observations of the DoAs of sound sources, caused by the sporadic activities of sound sources (e.g., the sparsity of speech), combined with the presence of multiple competing sound sources. Sound Source Tracking (SST) methods are thus used to filter out this noise and provide a smooth trajectory of the sound sources, which can be used to derive distinct audio streams for each source (a process known as sound source separation (SSS) [Grondin *et coll.*, 2013]).

SST methods can be categorized into three types :

- Viterbi search. Anguera et al. [Anguera *et coll.*, 2007] propose a post-processing Viterbi method to track a sound source over time. This method introduces a significant latency when used online, making it appropriate only for offline processing. Tracking is also performed on discrete states, which restrains the direction of the tracked source to a fixed grid.

- Sequential Monte Carlo (SMC) filtering. The SMC method, also called particle filtering, performs low latency tracking for a single sound source [Vermaak et Blake, 2001; Ward *et coll.*, 2003; Williamson et Ward, 2002]. Valin et al. [Grondin *et coll.*, 2013; Valin *et coll.*, 2007a] adapt the SMC method to track multiple sound sources. This method consists in sampling the space with finite particles to model the non-Gaussian state distribution. SMC provides tracking with continuous trajectories, but requires a significant amount of computations, and is undeterministic because it uses randomly generated particles.
- Kalman filtering. Rascon et al. [Rascon *et coll.*, 2015] propose a lightweight method that relies on Kalman filters for tracking with continuous trajectories, and it reduces considerably the amount of computations. This method is however limited to DoAs in spherical coordinates, using elevation and azimuth, which generates distortion as the azimuth resolution changes with elevation. It also introduces azimuth wrapping. Marković et al. [Marković *et coll.*, 2016] present an extended Kalman filter on Lie groups (LG-EKF) to perform directional tracking with an 8-microphone array. LG-EKF solves the azimuth wrapping phenomenon, but limits the tracking to a 2D circle, and is therefore unsuitable for tracking sources on a 3D spherical surface.

This chapter proposes a simultaneous SST method based on a modified 3D Kalman filter (M3K) in Cartesian coordinates. This method replaces the SMC filters used in Valin et al. [Valin *et coll.*, 2007b] by Kalman filters, and introduce three new concepts : 1) normalization of the states to restrict the space to a unit sphere ; 2) derivation of a closed-form expression for the likelihood of a coherent source to speed up computations ; and 3) weighted update of the Kalman mean vector and covariance matrix for simultaneous tracking of sound sources. These modifications provide efficient tracking of multiple sound sources, makes the method convenient for low-cost embedded hardware as it requires less computations than the SMC method, and solves the distortion and wrapping introduced by Kalman filtering with spherical coordinates. The method is also deterministic, i.e., it generates the same output every time for a given set of noisy observations.

The chapter is organized as follows. Section 4.2 presents the M3K method. Section 4.3 describes the experimental setup involving a mobile robot, and Section 4.4 presents the results obtained from experiments, comparing M3K with SMC in terms of computational load, tracking of static and moving sound sources.

4.2 Modified 3D Kalman Filters for SST

Figure 4.1 illustrates the block diagram of a typical SSL method with SST. For each frame l, SSL uses the captured signals from the M-microphone array $\mathbf{X}^l = \{\mathbf{X}_1^l, \mathbf{X}_2^l, \ldots, \mathbf{X}_M^l\}$ and generates V potential sources $\boldsymbol{\Psi}^l = \{\boldsymbol{\psi}_1^l, \boldsymbol{\psi}_2^l, \ldots, \boldsymbol{\psi}_V^l\}$. SST then uses the potential sources and returns I tracked sources $\boldsymbol{\Phi}^l = \{\phi_1^l, \phi_2^l, \ldots, \phi_I^l\}$. Each potential source $\boldsymbol{\psi}_v^l$ consists of a direction $\boldsymbol{\lambda}_v^l$ in Cartesian coordinates, and the steered beamformer energy level Λ_v^l, where v stands for the potential source index :

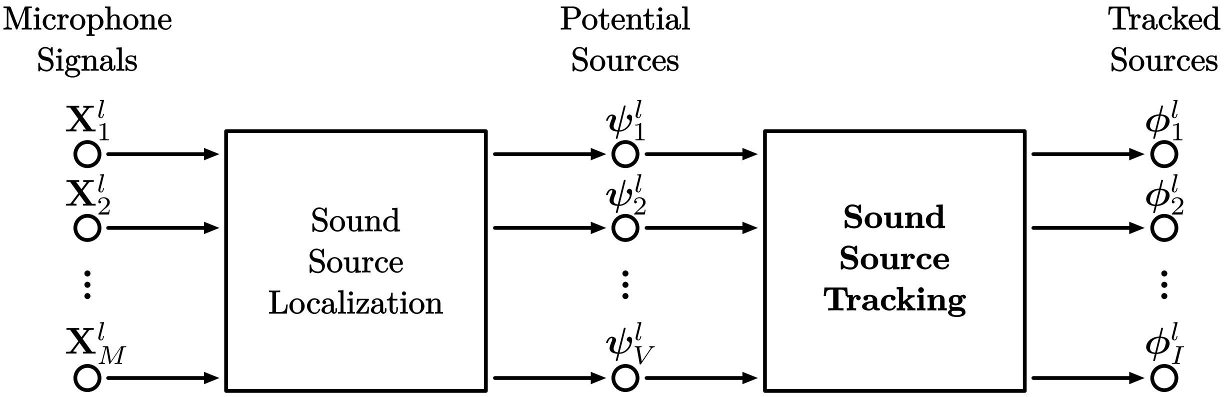


Figure 4.1 Block diagram of sound source localization and tracking

$$\boldsymbol{\psi}_v^l = \{\boldsymbol{\lambda}_v^l, \Lambda_v^l\} \tag{4.1}$$

where

$$\boldsymbol{\lambda}_v^l = \begin{bmatrix} (\lambda_x)_v^l & (\lambda_y)_v^l & (\lambda_z)_v^l \end{bmatrix}^T \tag{4.2}$$

SST then generates I tracked source trajectories, where $\boldsymbol{\phi}_i^l$ stands for the estimated trajectory of the tracked source at index i :

$$\boldsymbol{\phi}_i^l = \begin{bmatrix} (\phi_x)_i^l & (\phi_y)_i^l & (\phi_z)_i^l \end{bmatrix}^T \tag{4.3}$$

and the set $\boldsymbol{\Phi}^l$ contains all the tracked sources at frame l :

$$\boldsymbol{\Phi}^l = \{\phi_1^l, \phi_2^l, \ldots, \phi_I^l\} \tag{4.4}$$

Figure 4.2 illustrates the nine steps of the M3K method, where Normalization (step B), Likelihood (Step D) and Update (Step H) are introduced to include Kalman filtering and replace particle filtering in the multiple sources tracking method presented by Valin et al. [Valin *et coll.*, 2007a]. First, the new states of each tracked source are predicted (step A) and normalized (step B) in relation to the search space. Second, the potential sources are then assigned (step C) to either a source currently tracked, a new source of a false detection (steps D, E, F). Third, the method then adds (step G) new sources to be tracked if needed, and removes inactive sources previously tracked. Fourth, the states of each tracked source are updated (step H) with the relevant observations, and the direction of each tracked source is finally estimated (step I) from the Gaussian distributions.

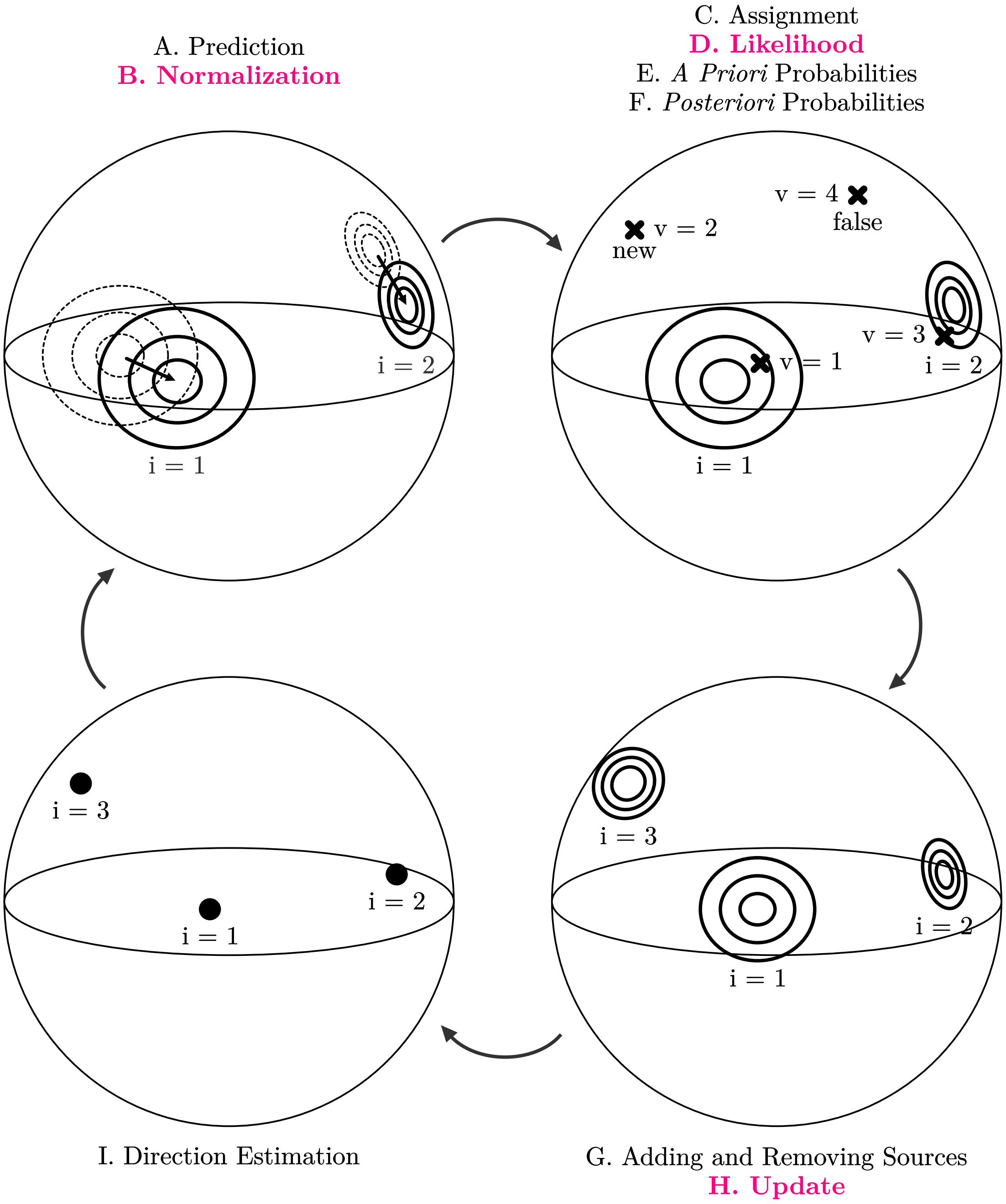


Figure 4.2 Tracking simultaneous sound sources using M3K. Tracked sources are labeled $i = 1, 2, 3$ and potential sources are labeled $q = 1, 2, 3, 4$.

Before presenting in more details these nine steps in the following subsections, let us first define the Kalman filter model used in M3K. A Kalman filter estimates recursively the state of each source and provides the estimated source direction. Normally, distributed random variables model the 3D-direction ($(d_x)^l_i$, $(d_y)^l_i$ and $(d_z)^l_i$) and 3D-velocity ($(s_x)^l_i$, $(s_y)^l_i$ and $(s_z)^l_i$), where $\{\ldots\}^T$ stands the transpose operator :

$$\mathbf{d}^l_i = \left[\begin{array}{ccc} (d_x)^l_i & (d_y)^l_i & (d_z)^l_i \end{array} \right]^T \tag{4.5}$$

$$\mathbf{s}^l_i = \left[\begin{array}{ccc} (s_x)^l_i & (s_y)^l_i & (s_z)^l_i \end{array} \right]^T \tag{4.6}$$

The 6×1 random vector $\mathbf{x}^l_i$ concatenates these positions and velocities :

$$\mathbf{x}^l_i = \left[\begin{array}{c} \mathbf{d}^l_i \\ \mathbf{s}^l_i \end{array} \right] \tag{4.7}$$

The Kalman model assumes the state evolves over time according to the following linear model :

$$\mathbf{x}^l_i = \mathbf{F}\mathbf{x}^{l-1}_i + \mathbf{B}\mathbf{u}^l_i + \mathbf{w} \tag{4.8}$$

where the matrix $\mathbf{F}$ stands for the state transition model, $\mathbf{B}$ represents the control-input model, $\mathbf{u}^i_l$ is the control vector and $\mathbf{w}$ models the process noise. In the 6×6 matrix $\mathbf{F}$, the expression $\Delta T = \Delta N / f_S$ denotes the time interval (in second) between two successive frames, with ΔN being the hop size in samples between two frames, and f_S the sample rate in samples per second :

$$\mathbf{F} = \left[\begin{array}{cccccc} 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \tag{4.9}$$

With M3K, there is no control input, and therefore the expression ($\mathbf{B}\mathbf{u}^i_l$) in (4.8) is ignored. The process noise $\mathbf{w}$ exhibits a multivariate normal distribution, where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$. In M3K, the process noise lies in the velocity state and is parametrized with the variance σ^2_Q :

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_Q^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_Q^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_Q^2 \end{bmatrix} \tag{4.10}$$

The observations $\mathbf{z}_l^i$ are represented by random variables in the x-, y- and z-directions, obtained from the states :

$$\mathbf{z}_l^i = \mathbf{H}\mathbf{x}_l^i + \mathbf{v} \tag{4.11}$$

where

$$\mathbf{z}_l^i = \begin{bmatrix} (z_x)_i^l & (z_y)_i^l & (z_z)_i^l \end{bmatrix}^T \tag{4.12}$$

The 3×6 matrix $\mathbf{H}$ stands for the observation model :

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.13}$$

Expression $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ models the observation noise, where the 3×3 diagonal covariance matrix $\mathbf{R}$ is defined as :

$$\mathbf{R} = \begin{bmatrix} \sigma_R^2 & 0 & 0 \\ 0 & \sigma_R^2 & 0 \\ 0 & 0 & \sigma_R^2 \end{bmatrix} \tag{4.14}$$

M3K therefore requires only two manually-tuned parameters, σ_Q^2 and σ_R^2, which influence the tracking sensitivity (a large value of σ_Q^2 increases observation uncertainty) and inertia (a small value of σ_R^2 reduces velocity uncertainty) of each tracked source.

4.2.1 Prediction (Step A)

The vector $\hat{\mathbf{x}}_i^{l|l}$ represents the mean value of the *posteriori* states estimate, where the 6×6 matrix $\mathbf{P}_i^{l|l}$ stands for the *posteriori* state error covariance matrix of each tracked source i. The tracking method predicts new states (also referred to as *a priori* states) for each sound source i. Predicted mean vector and covariance matrix are obtained as follows :

$$\hat{\mathbf{x}}_i^{l|l-1} = \mathbf{F}\hat{\mathbf{x}}_i^{l-1|l-1} \tag{4.15}$$

$$\mathbf{P}_i^{l|l-1} = \mathbf{F}\mathbf{P}_i^{l-1|l-1}\mathbf{F}^T + \mathbf{Q} \tag{4.16}$$

Each prediction step increases the states uncertainty, which is then reduced in the Update step (step H) when the tracked source is associated to a relevant observation.

4.2.2 Normalization (Step B)

The observations $\boldsymbol{\lambda}_v^l$ stands for sound direction of the potential source v, which constitutes a unit vector. A normalization constraint is therefore introduced and generates a new state mean vector $(\hat{\mathbf{x}}')_i^{l|l-1}$, for which direction $(\hat{\mathbf{d}}')_i^{l|l-1}$ lies on a unitary sphere and velocity $(\hat{\mathbf{s}}')_i^{l|l-1}$ is tangential to the sphere surface :

$$(\hat{\mathbf{x}}')_i^{l|l-1} = \begin{bmatrix} (\hat{\mathbf{d}}')_i^{l|l-1} \\ (\hat{\mathbf{s}}')_i^{l|l-1} \end{bmatrix} \tag{4.17}$$

where

$$(\hat{\mathbf{d}}')_i^{l|l-1} = \frac{\hat{\mathbf{d}}_i^{l|l-1}}{\|\hat{\mathbf{d}}_i^{l|l-1}\|} \tag{4.18}$$

and

$$(\hat{\mathbf{s}}')_i^{l|l-1} = \hat{\mathbf{s}}_i^{l|l-1} - \hat{\mathbf{d}}_i^{l|l-1}\left(\frac{\hat{\mathbf{s}}_i^{l|l-1} \cdot \hat{\mathbf{d}}_i^{l|l-1}}{\|\hat{\mathbf{d}}_i^{l|l-1}\|^2}\right) \tag{4.19}$$

This manipulation violates the Kalman filter assumptions, which state that all processes are Gaussian and that the system is linear [Julier et Uhlmann, 1997]. In practice however, Kalman filtering remains efficient as this normalization only involves a slight perturbation in direction and velocity, which makes the nonlinearity negligible.

During the normalization process, the covariance matrix remains unchanged. The Update step (step H) ensures that the radial component of the matrix $\mathbf{P}_i^{l|l-1}$ stays as small as possible such that the PDF (Probability Density Function) lies mostly on the unit sphere surface.

4.2.3 Assignment (Step C)

Assuming that I sources are tracked, the function $f_g(v)$ assigns the potential source at index v to either a false detection (-2), a new source (-1), or a previously tracked source (from 1 to I) :

$$f_g(v) \in \{-2, -1, 1, 2, \ldots, I\} \tag{4.20}$$

There are V potential sources that can be assigned to $I + 2$ values, which leads to $G = (I+2)^V$ possible permutations. The assignation vector for the permutation g concatenates the V assignment functions for the potential sources :

$$\mathbf{f}_g = \left[\begin{array}{cccc} f_g(1) & f_g(2) & \ldots & f_g(V) \end{array} \right] \tag{4.21}$$

4.2.4 Likelihood (Step D)

The energy level Λ_v^l gives significant information regarding sound source activity. When a sound source is active ($\mathcal{A}$), which means the source emits a sound, a Gaussian distribution models the energy level and the probability is given by :

$$p(\Lambda_v^l|\mathcal{A}) = \mathcal{N}(\Lambda_v^l|\mu_{\mathcal{A}}, \sigma_{\mathcal{A}}) \tag{4.22}$$

where $\mu_{\mathcal{A}}$ and $\sigma_{\mathcal{A}}$ stand for the mean and standard deviation of the normal distribution, respectively.

When the source is inactive, the energy level is modeled with the same distribution but different parameters :

$$p(\Lambda_v^l|\mathcal{I}) = \mathcal{N}(\Lambda_v^l|\mu_{\mathcal{I}}, \sigma_{\mathcal{I}}) \tag{4.23}$$

where $\mu_{\mathcal{I}}$ and $\sigma_{\mathcal{I}}$ also represent the mean and standard deviation of the normal distribution, respectively. Typically the standard deviation $\sigma_{\mathcal{I}}$ is similar to $\sigma_{\mathcal{A}}$, but the mean $\mu_{\mathcal{I}}$ is smaller than $\mu_{\mathcal{A}}$. Moreover, the probability that the potential source $\boldsymbol{\lambda}_v^l$ is generated by the tracked source i is obtained with the following volume integral :

$$p(\boldsymbol{\lambda}_v^l|\mathcal{C}_i) = \iiint p(\boldsymbol{\lambda}_v^l|(\mathbf{d}')_i^{l|l-1})p((\mathbf{d}')_i^{l|l-1})\,dx\,dy\,dz \tag{4.24}$$

The symbol $\mathcal{C}_i$ stands for a coherent source, which describes a sound source located at a specific direction in space. As modeled by the Kalman filter, the probability $p((\mathbf{d}')_i^{l|l-1})$ follows this normal distribution :

$$(\mathbf{d}')_i^{l|l-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_i^l, \boldsymbol{\Sigma}_i^l\right) \tag{4.25}$$

where

$$\boldsymbol{\mu}_i^l = \mathbf{H}(\hat{\mathbf{x}}')_i^{l|l-1} \tag{4.26}$$

and

$$\boldsymbol{\Sigma}_i^l = \mathbf{H}\mathbf{P}_i^{l|l-1}\mathbf{H}^T \tag{4.27}$$

Given the normalized tracked source direction $(\mathbf{d}')_i^{l|l-1}$, the following expression represents the probability that the potential source $\boldsymbol{\lambda}_v^l$ is observed :

$$\boldsymbol{\lambda}_v^l|(\mathbf{d}')_i^{l|l-1} \sim \mathcal{N}\left((\mathbf{d}')_i^{l|l-1}, \mathbf{R}\right) \tag{4.28}$$

Note that swapping the mean and the random variable leads to the same PDF (the expression $|\ldots|$ stands for the matrix determinant) :

$$p(\boldsymbol{\lambda}_v^l|(\mathbf{d}')_i^{l|l-1}) = \frac{(2\pi)^{2/3}}{|\mathbf{R}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{\lambda}_v^l - (\mathbf{d}')_i^{l|l-1})^T \mathbf{R}^{-1} (\boldsymbol{\lambda}_v^l - (\mathbf{d}')_i^{l|l-1})} \tag{4.29}$$

$$p((\mathbf{d}')_i^{l|l-1}|\boldsymbol{\lambda}_v^l) = \frac{(2\pi)^{2/3}}{|\mathbf{R}|^{1/2}} e^{-\frac{1}{2}((\mathbf{d}')_i^{l|l-1} - \boldsymbol{\lambda}_v^l)^T \mathbf{R}^{-1} ((\mathbf{d}')_i^{l|l-1} - \boldsymbol{\lambda}_i^l)} \tag{4.30}$$

The following random variable is therefore defined :

$$(\mathbf{d}')_{l|l-1}^i|\boldsymbol{\lambda}_v^l \sim \mathcal{N}\left(\boldsymbol{\mu}_v^l, \boldsymbol{\Sigma}_v^l\right) \tag{4.31}$$

where

$$\boldsymbol{\mu}_v^l = \boldsymbol{\lambda}_v^l \tag{4.32}$$

and

$$\boldsymbol{\Sigma}_v^l = \mathbf{R} \tag{4.33}$$

The expression $p(\boldsymbol{\lambda}_v^l|(\mathbf{d}')_i^{l|l-1})p((\mathbf{d}')_i^{l|l-1})$ is equivalent to $p((\mathbf{d}')_i^{l|l-1}|\boldsymbol{\lambda}_v^l)p((\mathbf{d}')_i^{l|l-1})$, which results in the product of two Gaussian distributions. According to [Bromiley, 2003], this results in a new Gaussian distribution, scaled by the factor ω_{iv}^l :

$$p(\boldsymbol{\lambda}_v^l|(\mathbf{d}')_i^{l|l-1})p((\mathbf{d}')_i^{l|l-1}) = \omega_{iv}^l \mathcal{N}((\mathbf{d}')_i^{l|l-1}|\boldsymbol{\mu}_{iv}^l, \boldsymbol{\Sigma}_{iv}^l) \tag{4.34}$$

As derived in [Bromiley, 2003], the mean vector $\boldsymbol{\mu}^l_{iv}$ and covariance matrix $\boldsymbol{\Sigma}^l_{iv}$ of the resulting distribution are equal to :

$$\boldsymbol{\mu}^l_{iv} = \boldsymbol{\Sigma}^l_{iv}((\boldsymbol{\Sigma}^l_i)^{-1}\boldsymbol{\mu}^l_i + (\boldsymbol{\Sigma}^l_v)^{-1}\boldsymbol{\mu}^l_v) \tag{4.35}$$

$$\boldsymbol{\Sigma}^l_{iv} = ((\boldsymbol{\Sigma}^l_i)^{-1} + (\boldsymbol{\Sigma}^l_v)^{-1})^{-1} \tag{4.36}$$

The scaling factor equals to :

$$\omega^l_{iv} = e^{\left(\frac{1}{2}\left[(C_1)^l_{iv}+(C_2)^l_{iv}-(C_3)^l_{iv}-(C_4)^l_{iv}\right]\right)} \tag{4.37}$$

where

$$(C_1)^l_{iv} = \log|\boldsymbol{\Sigma}^l_{iv}| - \log\left(8\pi^3|\boldsymbol{\Sigma}^l_i||\boldsymbol{\Sigma}^l_v|\right) \tag{4.38}$$

$$(C_2)^l_{iv} = (\boldsymbol{\mu}^l_{iv})^T(\boldsymbol{\Sigma}^l_{iv})^{-1}\boldsymbol{\mu}^l_{iv} \tag{4.39}$$

$$(C_3)^l_{iv} = (\boldsymbol{\mu}^l_i)^T(\boldsymbol{\Sigma}^l_i)^{-1}\boldsymbol{\mu}^l_i \tag{4.40}$$

$$(C_4)^l_{iv} = (\boldsymbol{\mu}^l_v)^T(\boldsymbol{\Sigma}^l_v)^{-1}\boldsymbol{\mu}^l_v \tag{4.41}$$

The new Gaussian distribution is substituted in (4.24), and since the volume integral over a trivariate normal PDF is equal to 1, the probability is simply equal to the scaling factor computed in (4.37) :

$$p(\boldsymbol{\lambda}^l_v|\mathcal{C}_i) = \omega^l_{iv}\iiint \mathcal{N}((\mathbf{d}')^{l|l-1}_i|\boldsymbol{\mu}^l_{iv}, \boldsymbol{\Sigma}^l_{iv})\,dx\,dy\,dz = \omega^l_{iv} \tag{4.42}$$

This provides a direct way to compute the probability $p(\boldsymbol{\lambda}^l_v|\mathcal{C}_i)$, which is far more efficient than SMC where the probability is estimated by sampling the distribution. Figure 4.3 illustrates the analytic simplification of how M3K simplifies the computation of the triple integral introduced in (4.24).

Each state probability (in the blue area) is multiplied by the probability that $\boldsymbol{\lambda}^l_v$ is observed (in pink) by this state. This involves a significant amount of computations to sample the state space : this is in fact what the particle filter does, and why the computational load is important. It is therefore more efficient to compute the closed expression that represents the overlap (in green) between state PDF (in blue) and the PDF obtained in (4.31) from swapping variables (in yellow).

When a new source appears or a false detection occurs, the observation lies anywhere on the scanned space. This is denoted by the symbol $\mathcal{D}$ for diffused signal. The SSL

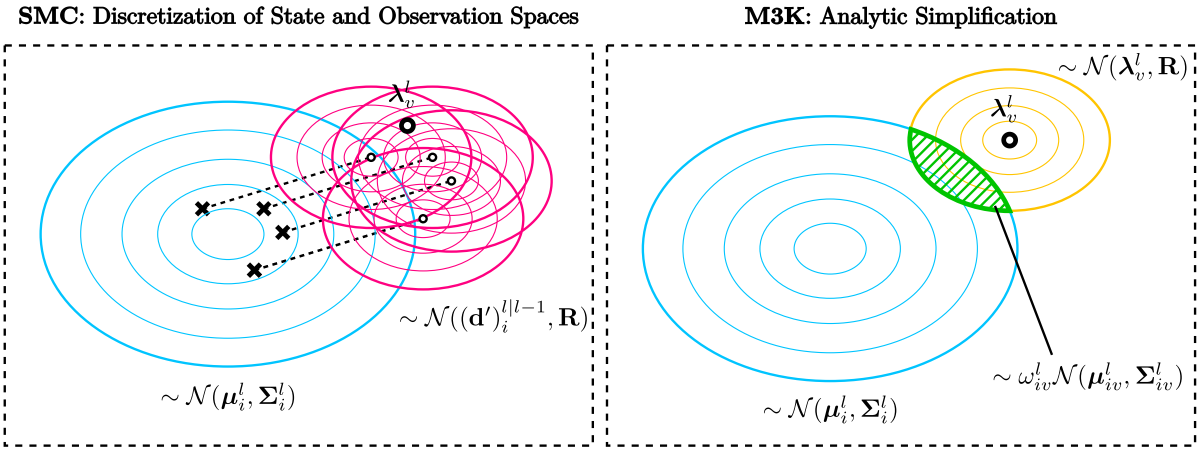

SMC: Discretization of State and Observation Spaces
M3K: Analytic Simplification
$\boldsymbol{\lambda}_v^l$
$\sim \mathcal{N}((\mathbf{d}')_i^{l|l-1}, \mathbf{R})$
$\sim \mathcal{N}(\boldsymbol{\mu}_i^l, \boldsymbol{\Sigma}_i^l)$
$\sim \mathcal{N}(\boldsymbol{\lambda}_v^l, \mathbf{R})$
$\boldsymbol{\lambda}_v^l$
$\sim \omega_{iv}^l \mathcal{N}(\boldsymbol{\mu}_{iv}^l, \boldsymbol{\Sigma}_{iv}^l)$
$\sim \mathcal{N}(\boldsymbol{\mu}_i^l, \boldsymbol{\Sigma}_i^l)$

Figure 4.3 Analytic simplification of M3K compared to SMC. The probability that the observation $\boldsymbol{\lambda}_v^l$ occurs corresponds to the sum of the product of the probability of each state (in blue) with the probability this state generates the observation $\boldsymbol{\lambda}_v^l$ (in pink).

module generates DoAs from the scanned space around the microphone array. This space is modeled by a unit sphere around the array, but the search often partially covers the complete area due to blind spots introduced by the microphone array geometry and other constraints [Grondin et Michaud, 2017]. A uniform distribution therefore models the PDF, where $\hat{K}$ denotes the number of points scanned, and K the total number of points needed to discretize the entire sphere :

$$p(\boldsymbol{\lambda}_v^l|\mathcal{D}) = \frac{\hat{K}}{K}\left(\frac{1}{4\pi}\right) = \frac{\hat{K}}{4\pi K} \tag{4.43}$$

where $1/4\pi$ stands for the uniform distribution over a complete sphere.

The overall likelihood for each possible potential-tracked source assignation results in the combination of the energy level Λ_v^l and potential source positions $\boldsymbol{\lambda}_v^l$ observations, concatenated in the vector $\boldsymbol{\psi}_v^l$. Figure 4.4 illustrates the three types of assignment :

1. False detection : the perceived signal is diffused ($\mathcal{D}$) and the source is inactive ($\mathcal{I}$).
2. New source : the perceived signal is diffused ($\mathcal{D}$) and the source is active ($\mathcal{A}$).
3. Tracked source i : the potential source direction is coherent with the tracked source i ($\mathcal{C}_i$) and the source is active ($\mathcal{A}$).

In Fig. 4.4, it is assumed that the potential sources are generated only in the top hemisphere, which motivates the use of a uniform distribution in this region only for new sources and false detection assignments.

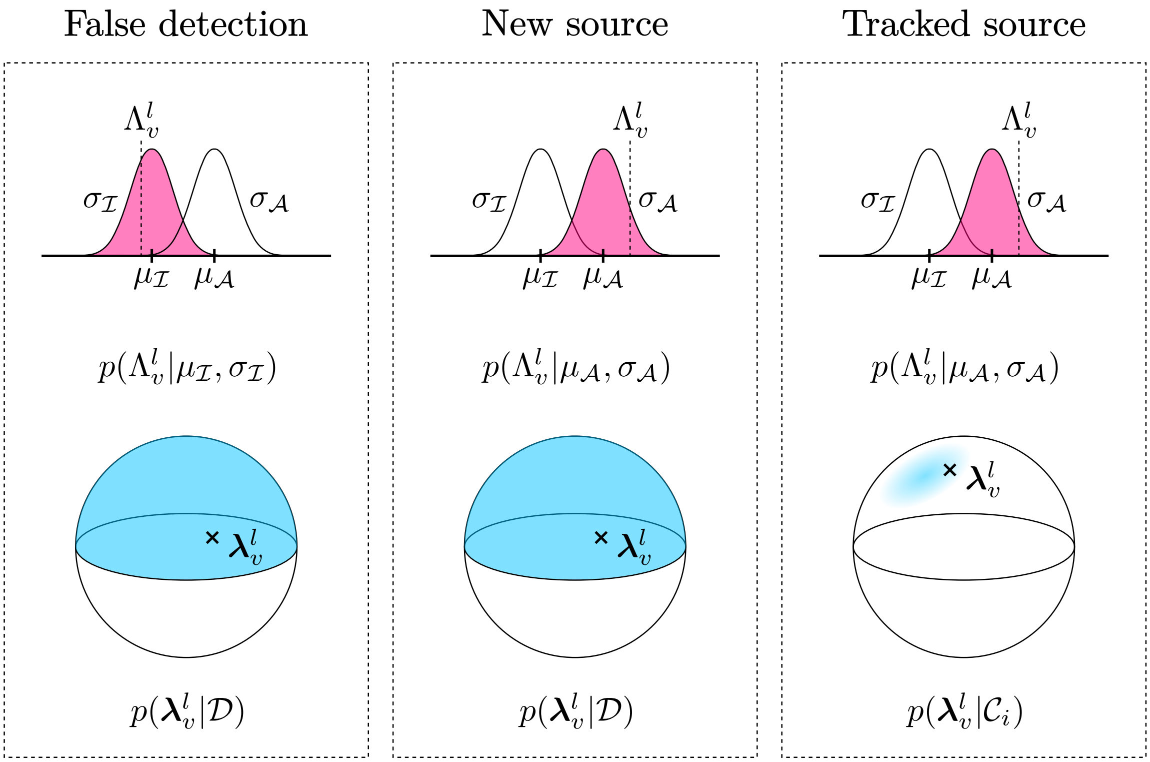

False detection
New source
Tracked source
Λ_v^l
$\sigma_{\mathcal{I}}$
$\sigma_{\mathcal{A}}$
$\mu_{\mathcal{I}}$
$\mu_{\mathcal{A}}$
$p(\Lambda_v^l|\mu_{\mathcal{I}},\sigma_{\mathcal{I}})$
$p(\Lambda_v^l|\mu_{\mathcal{A}},\sigma_{\mathcal{A}})$
$p(\Lambda_v^l|\mu_{\mathcal{A}},\sigma_{\mathcal{A}})$
$\boldsymbol{\lambda}_v^l$
$p(\boldsymbol{\lambda}_v^l|\mathcal{D})$
$p(\boldsymbol{\lambda}_v^l|\mathcal{D})$
$p(\boldsymbol{\lambda}_v^l|\mathcal{C}_i)$

Figure 4.4 Types of assignments for each potential source

The likelihood probability $p(\boldsymbol{\psi}_l^q|f_g(q))$ is therefore computed as follows :

$$p(\boldsymbol{\psi}_v^l|f_g(v)) = \begin{cases} p(\Lambda_v^l|\mathcal{I})p(\boldsymbol{\lambda}_v^l|\mathcal{D}) & f_g(v) = -2 \\ p(\Lambda_v^l|\mathcal{A})p(\boldsymbol{\lambda}_v^l|\mathcal{D}) & f_g(v) = -1 \\ p(\Lambda_v^l|\mathcal{A})p(\boldsymbol{\lambda}_v^l|\mathcal{C}_{f_g(v)}) & f_g(v) \geq 1 \end{cases} \tag{4.44}$$

The product of the individual likelihood probabilities generates the likelihood probability for each permutation of assignments :

$$p(\boldsymbol{\Psi}^l|\mathbf{f}_g) = \prod_{v=1}^{V} p(\boldsymbol{\psi}_v^l|f_g(v)) \tag{4.45}$$

4.2.5 A Priori Probabilities (Step E)

The *a priori* probabilities that a false detection, a new source or a tracked source occur are simply defined with the constant parameters P_{false}, P_{new}, and P_{track}, respectively :

$$p(f_g(v)) = \begin{cases} P_{false} & f_g(v) = -2 \\ P_{new} & f_g(v) = -1 \\ P_{track} & f_g(v) \geq 1 \end{cases} \tag{4.46}$$

These parameters are set empirically but it is observed they have little impact on the performance of tracking. The *a priori* probability for a given permutation corresponds to the product of each individual assignment :

$$p(\mathbf{f}_g) = \prod_{v=1}^{V} p(f_g(v)) \tag{4.47}$$

4.2.6 Posteriori Probabilities (Step F)

Bayes' theorem provides a method to obtain the *posteriori* probability for each permutation $\mathbf{f}_g$:

$$p(\mathbf{f}_g|\mathbf{\Psi}^l) = \frac{p(\mathbf{\Psi}^l|\mathbf{f}_g)p(\mathbf{f}_g)}{\sum_{g=1}^{G} p(\mathbf{\Psi}^l|\mathbf{f}_g)p(\mathbf{f}_g)} \tag{4.48}$$

To calculate the probability that a specific assignment is observed, the discrete Kronecker delta $\delta[n]$ is introduced :

$$\delta[n] = \begin{cases} 0 & n \neq 0 \\ 1 & n = 1 \end{cases} \tag{4.49}$$

The probability that the tracked source i generates the potential source v is therefore computed as follows :

$$p(i|\boldsymbol{\psi}_v^l) = \sum_{g=1}^{G} p(\mathbf{f}_g|\mathbf{\Psi}^l)\delta[f_g(v) - i] \tag{4.50}$$

The probability that a new source is observed is computed similarly :

$$p(\text{new}|\boldsymbol{\psi}_v^l) = \sum_{g=1}^{G} p(\mathbf{f}_g|\mathbf{\Psi}^l)\delta[f_g(v) + 1] \tag{4.51}$$

Finally, the probability that a tracked source is observed by any potential sources is computed using the combinations where there is an assignment between at least one potential source and the tracked source :

$$p(i|\mathbf{\Psi}^l) = \sum_{g=1}^{G} p(\mathbf{f}_g|\mathbf{\Psi}^l) \left(1 - \prod_{v=1}^{V} (1 - \delta[f_g(v) - i])\right) \tag{4.52}$$

4.2.7 Adding and Removing Sources (Step G)

Sound sources may appear and disappear dynamically as a new sound source starts or a tracked source stops being active. When a new source is detected ($p(\text{new}|\boldsymbol{\psi}_v^l) > \theta_{new}$), this step waits N_{prob} frames to confirm this is really a valid source and not just a sporadic detection. During this probation interval, the observation noise variance is set to the parameter $(\sigma_R^2)_{prob}$ to take a small value, as it is assumed the observations should lie close to each others during this time interval. The average of the probability $p(i|\boldsymbol{\psi}_v^l)$ of the newly tracked sound source is evaluated, and the source is kept only if the average exceeds the threshold θ_{prob}.

Once the existence of a source is confirmed, it is tracked until the source becomes inactive ($p(i|\mathbf{\Psi}^l) < \theta_{dead}$) for at least N_{dead} frames. During this active state, the observation noise variance is increased to the value of $(\sigma_R^2)_{active}$, to deal with noisy observations and possible motion of the sources. When the source no longer exists, it is deleted and tracking of this source stops.

4.2.8 Update (Step H)

For each tracked source, the Kalman gain is computed as follows :

$$\mathbf{K}_i^{l|l-1} = \mathbf{P}_i^{l-1|l}\mathbf{H}^T(\mathbf{H}\mathbf{P}_i^{l|l-1}\mathbf{H}^T + \mathbf{R})^{-1} \tag{4.53}$$

The expression $\hat{v}(i)$ stands for the index of the potential source that maximizes the probability $p(i|\mathbf{\Psi}^l))$:

$$\hat{v}(i) = \arg\max_v \left\{p(i|\mathbf{\Psi}^l)\right\} \tag{4.54}$$

This process is similar to gating, excepts that the probability $p(i|\mathbf{\Psi}^l)$ is used instead of the Mahalanobis distance between the observation and the tracked source position [Leonard et Durrant-Whyte, 1991]. The weighting factor $p(i|\mathbf{\Psi}^l)$ modulates the update rate of the mean vector and covariance matrix :

$$\hat{\mathbf{x}}_i^{l|l} = (\hat{\mathbf{x}}')_i^{l|l-1} + p(i|\mathbf{\Psi}^l)\mathbf{K}_i^{l|l-1}(\boldsymbol{\lambda}_{\hat{v}(i)}^l - \mathbf{H}(\hat{\mathbf{x}}')_i^{l|l-1}) \tag{4.55}$$

$$\mathbf{P}_i^{l|l} = \mathbf{P}_i^{l|l-1} - p(i|\mathbf{\Psi}^l)\mathbf{K}_i^{l|l-1}\mathbf{H}\mathbf{P}_i^{l|l-1} \tag{4.56}$$

When no potential source is clearly associated to the tracked source i, the probability $p(i|\mathbf{\Psi}^l)$ gets close to zero. The mean of the updated state $\hat{\mathbf{x}}_i^{l|l}$ is then similar to the mean of the predicted state $\hat{\mathbf{x}}_i^{l|l-1}$. Similarly, the updated covariance matrix $\mathbf{P}_i^{l|l}$ is similar to the predicted matrix $\mathbf{P}_i^{l|l-1}$, which grows after each prediction steps. In other words, when the observations do not provide useful information, the tracked source moves according to its inertia while the exact position uncertainty grows.

4.2.9 Direction Estimation (Step I)

The updated states provide an estimation for each sound source direction. This estimated direction $\boldsymbol{\phi}_i^l$ corresponds to the first moment of the *posteriori* random variable $(\mathbf{d}')_i^{l|l}$:

$$\boldsymbol{\phi}_i^l = \iiint \mathcal{N}((\mathbf{d}')_i^{l|l}|\mathbf{H}\hat{\mathbf{x}}_i^{l|l}, \mathbf{H}\mathbf{P}_i^{l|l}\mathbf{H}^T)(\mathbf{d}')_i^{l|l}\, dx\, dy\, dz \tag{4.57}$$

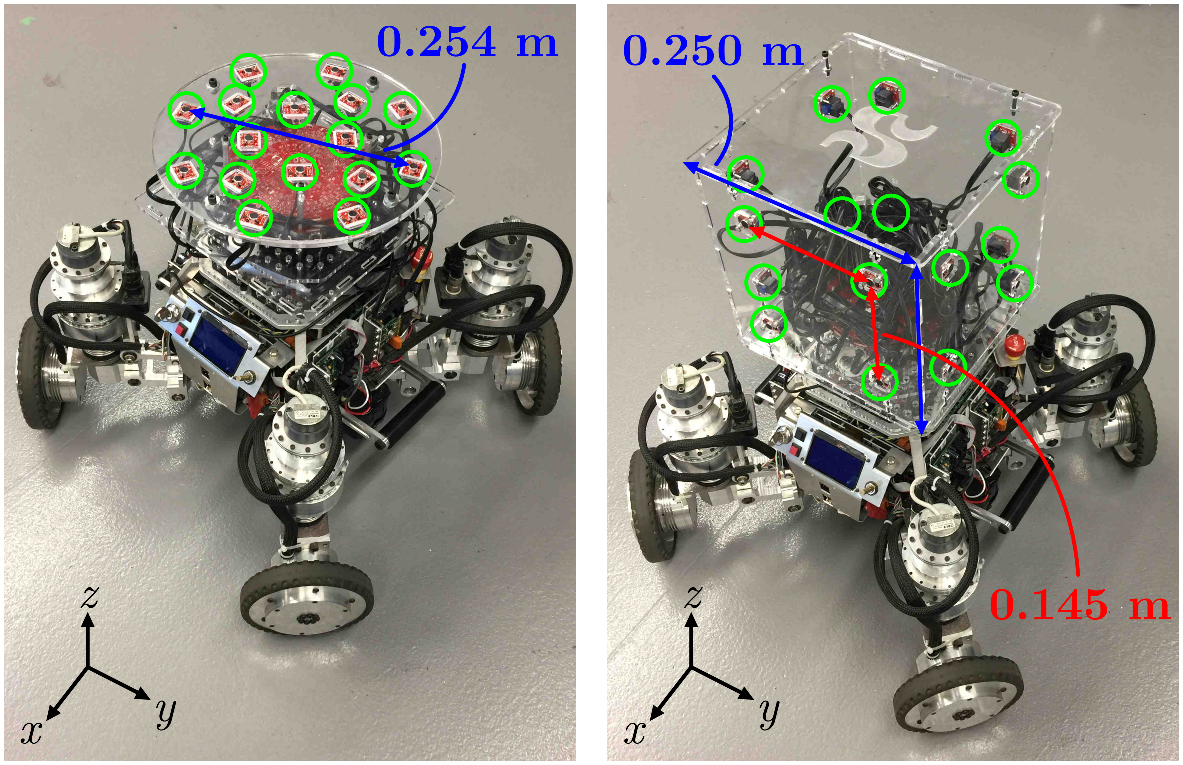
which simplifies to

$$\boldsymbol{\phi}_i^l = \mathbf{H}\hat{\mathbf{x}}_i^{l|l} \tag{4.58}$$

4.3 Experimental Setup

Experiments involve two 16-microphone array configurations installed on a mobile robot : 1) an opened microphone array (OMA) on a circular plane, and 2) a close microphone array (CMA) with microphones placed on a cubic structure. Figure 4.5 shows these two configurations. In the OMA configuration, the circular plane has a diameter of 0.254 m. For the CMA configuration, each edge is 0.250 m long, and microphones lie on a square shape with edges of 0.145 m.

Table 4.1 lists the M3K parameters used for the experiments. There are $M = 16$ microphones, and the SSL module [Grondin et Michaud, 2017] returns $V = 4$ potential sources per frame (it is observed that searching for more potential sources mostly return noisy observations, which are neglected by SST). In these experiments, M3K can track up to $I_{max} = 10$ sources simultaneously. The number of tracked sources can exceed the number of potential sources provided by SSL because the sources are active at different frames.



(a) OMA

(b) CMA

Figure 4.5 16-microphone array configurations

The energy level for active and inactive sound sources follows a Gaussian distribution with means $\mu_{\mathcal{A}}$ and $\mu_{\mathcal{I}}$, and variances $\sigma^2_{\mathcal{A}}$ and $\sigma^2_{\mathcal{I}}$. The Bayesian Extension method has been used to automatically tune these parameters [Otsuka *et coll.*, 2011]. However, for these experiments, setting these parameters empirically leads to good detection rates and tracking accuracy. The parameters $(\sigma^2_R)_{prob}$ and $(\sigma^2_R)_{active}$ match standard deviations of approximately 3° and 8° on the grid, respectively. The expression $(\sigma^2_R)_{prob}$ is smaller than $(\sigma^2_R)_{active}$ since the observations lie close to each other during the short probation interval. The variance of the process noise σ^2_Q is set to a value high enough to follow a source that changes direction, but low enough to preserve the source inertia. Parameters P_{false}, P_{new} and P_{track} are chosen empirically : they have little impact on trakcing performance as long as P_{track} is greater than P_{false} and P_{new}. New source detection requires θ_{new} to be close to a probability of 1, but small enough to detect new sources, and is therefore set empirically to 0.7. Probation corresponds to the time interval while a source is tracked but not displayed. It is defined as $N_{prob}\Delta N/f_S$ sec, which lies within the duration range of a single phoneme (i.e., 40 msec) [Anastasakos *et coll.*, 1995]. The number of inactive frames $N_{dead}\Delta N/f_S$ is set to match a duration of 1.2 sec, which is a reasonable silence period to consider a source is no longer active. The sample rate f_S and hop size ΔN correspond to the parameters used by the SSL method that generates the potential sources.

Tableau 4.1 SST Module Parameters

Parameters	Values	Parameters	Values
M	16	P_{false}	0.1
V	4	P_{new}	0.1
I_{max}	10	P_{track}	0.8
$\mu_{\mathcal{A}}$	0.20	θ_{new}	0.7
$\sigma^2_{\mathcal{A}}$	0.0025	N_{prob}	5
$\mu_{\mathcal{I}}$	0.10	θ_{prob}	0.8
$\sigma^2_{\mathcal{I}}$	0.0025	N_{dead}	150
$(\sigma^2_R)_{prob}$	0.0015	θ_{dead}	0.9
$(\sigma^2_R)_{active}$	0.0030	f_S	16000
σ^2_Q	0.000009	ΔN	128

4.4 Results

The proposed M3K method is tested in a real environment on a mobile robot and compared with the SMC method. The computational load of M3K is first measured on a low-cost embedded hardware, and compared to the load with SMC. Tracking experiments with static and moving sound sources are then examined. Male and female speakers talking in English are used as sound sources. The reverberation level in the room reaches $RT_{60} = 600$ msec, and there is no background noise. Only the tracking of the azimut is presented, because for both methods elevation matched the height of the sound sources for all trials.

4.4.1 Computational Load

A Raspberry Pi 3 (which processor is a ARM Cortex-A53 Quad-Core clocked at 1.2GHz) is used to compare CPU usage of M3K and SMC for a single core with C code, which is not optimized with Neon/SSE instructions. To assess the performance in terms of the number of tracked sources, the maximum number of simultaneously tracked sources I_{max} is set from 1 to 6, while there are 10 active sound sources located at the following azimuths : 0°, 40°, 70°, 100°, 140°, 180°, 220°, 260°, 300° and 330°. Figure 4.6 shows the CPU usage with both methods. The SMC method allows online processing for up to four tracked sources, and then CPU usage exceeds 100% (the usage increases to 127% with five tracked sources). The M3K reduces significantly the amount of computations, providing online processing with eight tracked sources (the usage is slightly above 100% (and rounded to 100%) with nine tracked sources). When a single source is tracked, M3K uses 0.8% of the CPU (rounded to 1% on Fig. 4.6), while SMC reaches a CPU usage of 24%. The M3K method is therefore up to 30 times more effective in terms of computational load. As the number of tracked sources increases, the number of permutations $(I+2)^V$ rises exponentially, which explains the high CPU usage for large values of I.

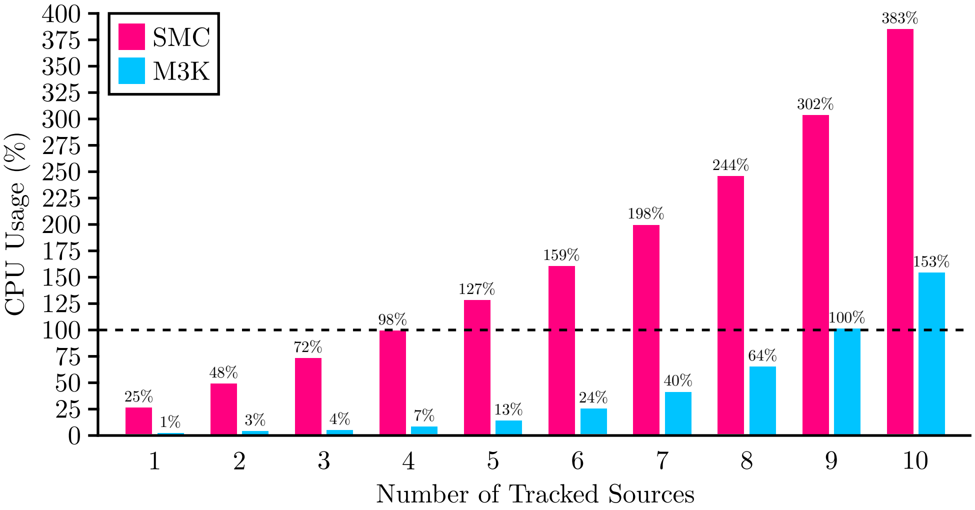


Figure 4.6 CPU Usage of SMC and M3K on a Raspberry Pi 3

4.4.2 Static Sound Sources

The first expriment conducted involves four static sources, to reproduce the test conditions of [Valin *et coll.*, 2007a]. In this experiment, a loudspeaker is positioned $r = 3$ m away from the robot, at azimuths of 10°, 100°, 190°, and 280°, and a height of 1.2 m related to the robot microphone array origin. Figure 4.7 illustrates how these signals for each position are recorded individually, and then mixed together.

Figure 4.8 and Fig. 4.9 show the potential sources generated by the SSL module and the corresponding tracked sources using SMC and M3K for the OMA and CMA configurations, respectively. Tracked sources trajectories illustrate that M3K performs as well as SMC for both OMA and CMA configurations.

We then increased the number of tracked sound sources to nine, reaching the limit of acceptable tracking performance. These nine static speech sources, at azimuths of 10°, 50°, 90°, 130°, 170°, 210°, 250°, 290° and 330°. Figure 4.10 and Figure 4.11 show the potential sources and the tracking results. The high number of sources makes detection and tracking more challenging for two reasons : 1) sources are closer to each other, which makes

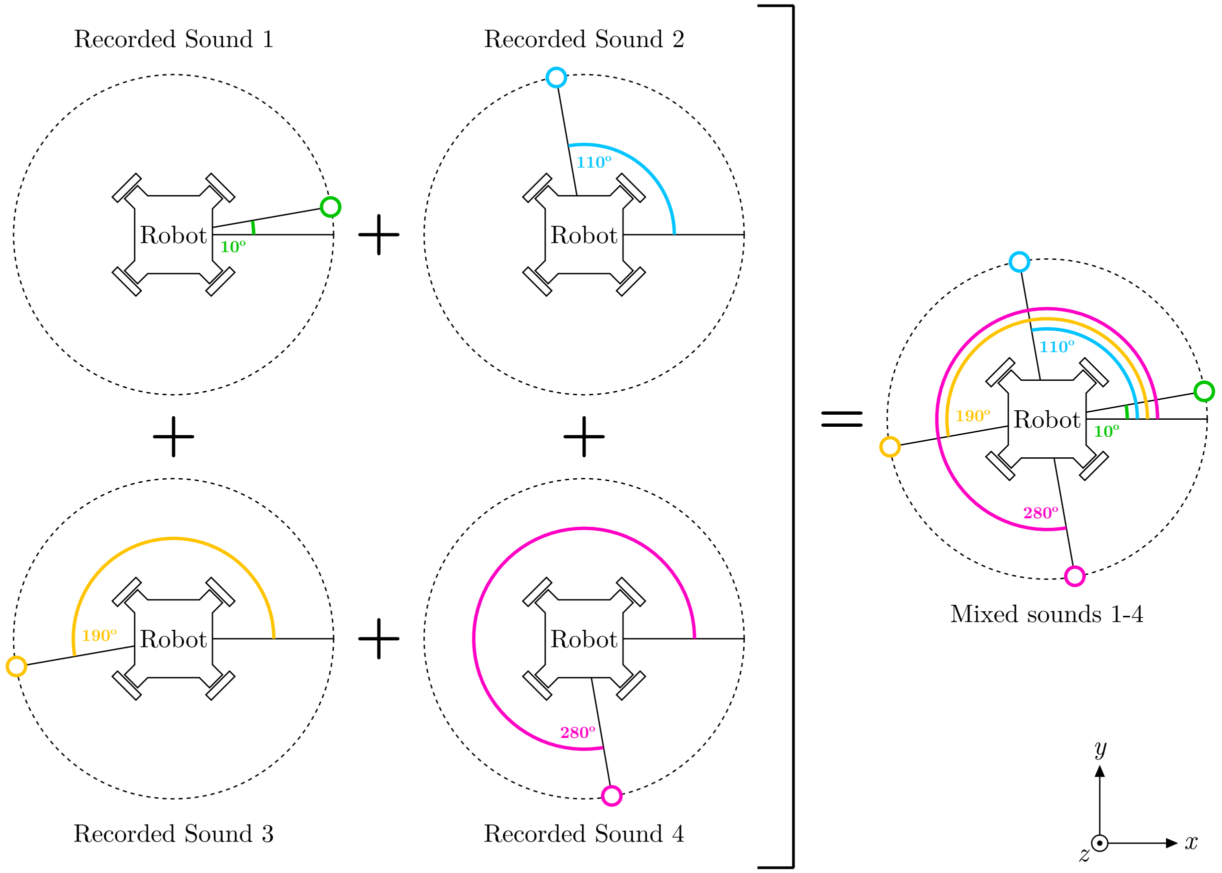
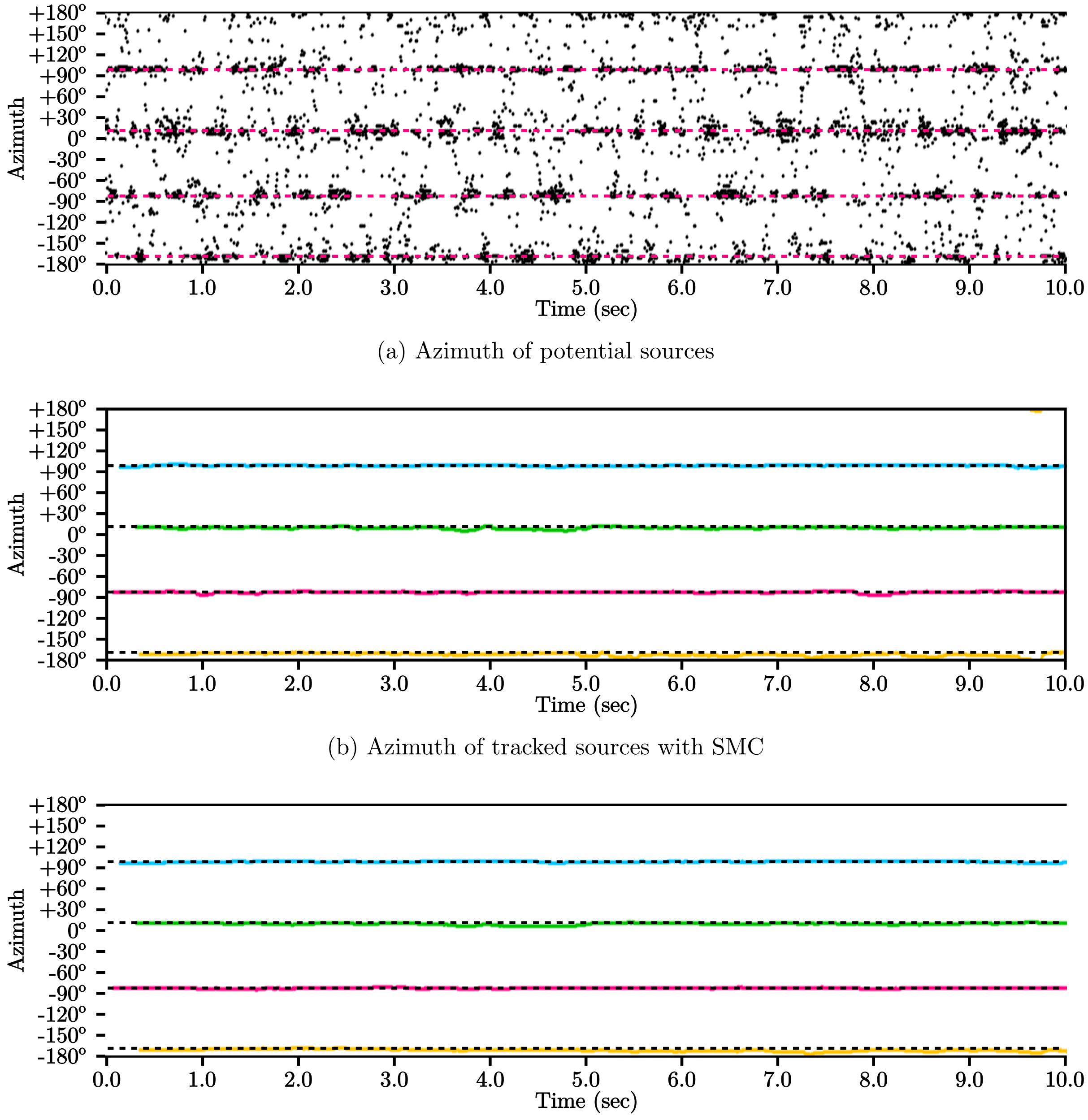


Figure 4.7 Mixing of four static sources

differentiation difficult between two static sources ; and 2) observations sparsity increases, which means each sound source gets assigned fewer potential sources from the SSL module as they are distributed over all active sources. SMC performs tracking accurately with the CMA configuration, but there is an error in tracking with the OMA that starts at 5 sec. The source drifts, and another source is created to track the source at 290°. M3K, which assumes all sources have a null or constant velocity, models more accurately the static sound source dynamics with both configurations. With nine sources, both M3K and SMC also take more time to detect all sources (up to 1 sec of latency) due to observations sparsity.

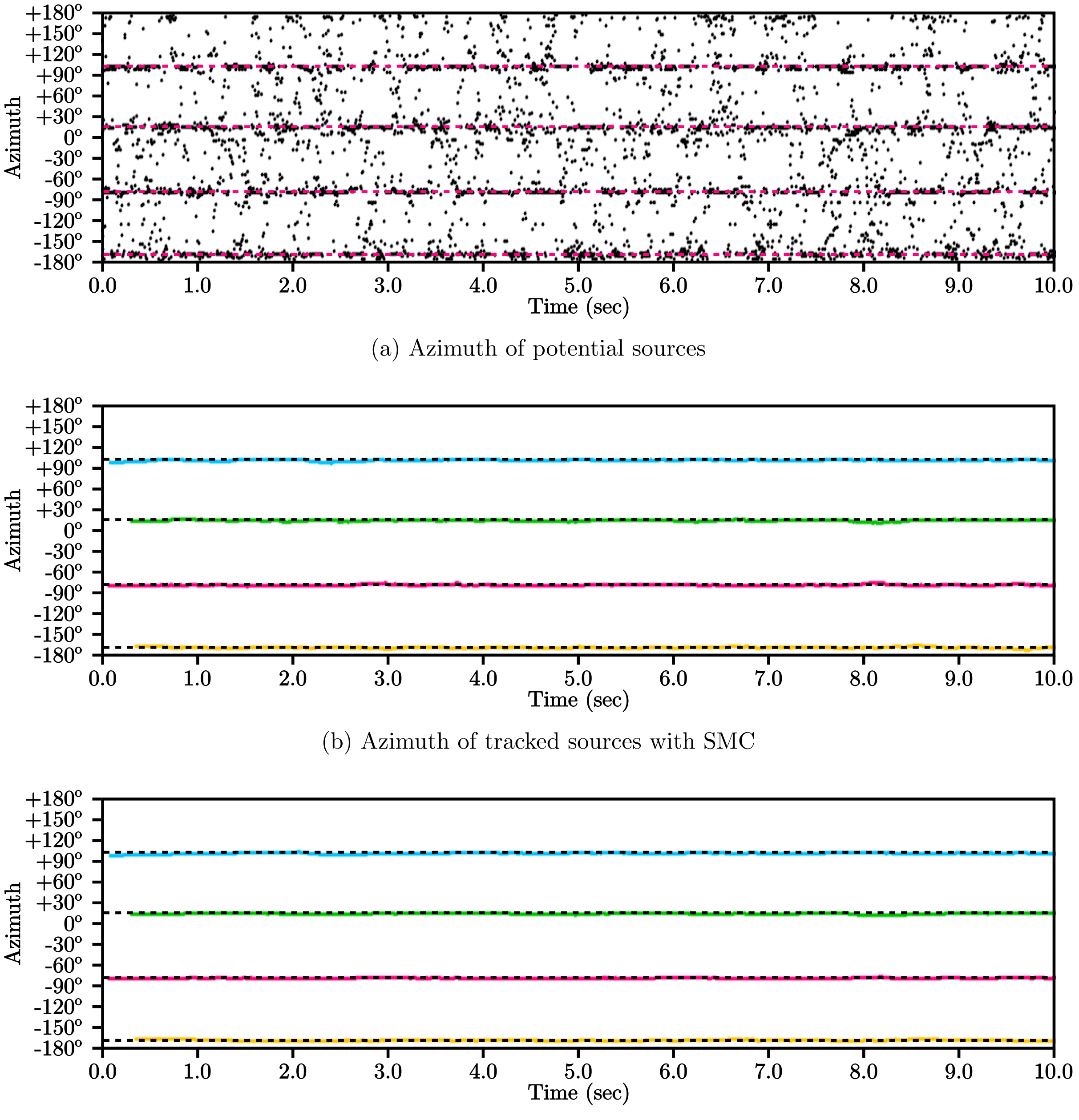


(a) Azimuth of potential sources

(b) Azimuth of tracked sources with SMC

(c) Azimuth of tracked sources with M3K

Figure 4.8 Four static speech sources with OMA configuration

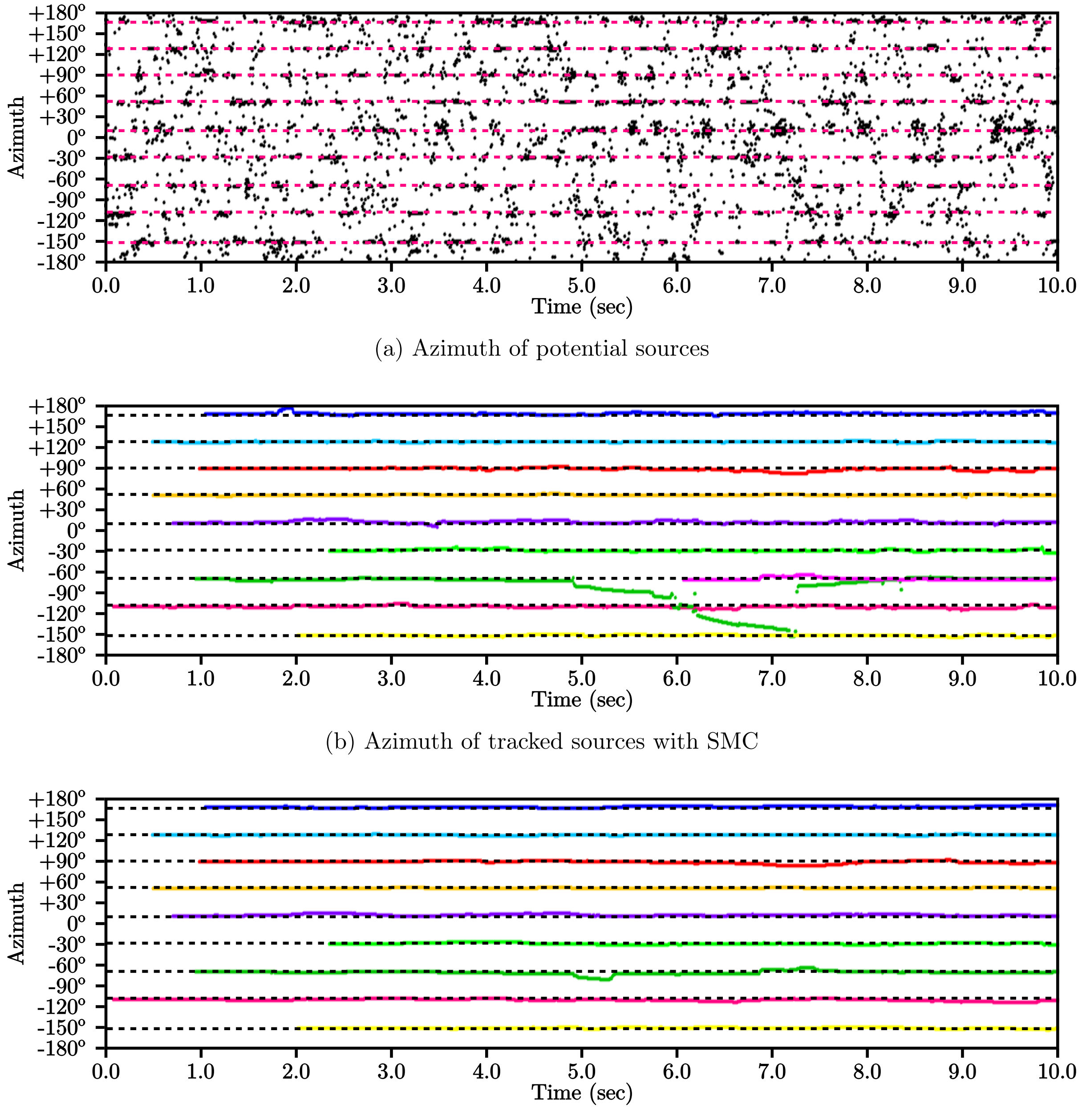


(a) Azimuth of potential sources

(b) Azimuth of tracked sources with SMC

(c) Azimuth of tracked sources with M3K

Figure 4.9 Four static sources with CMA configuration

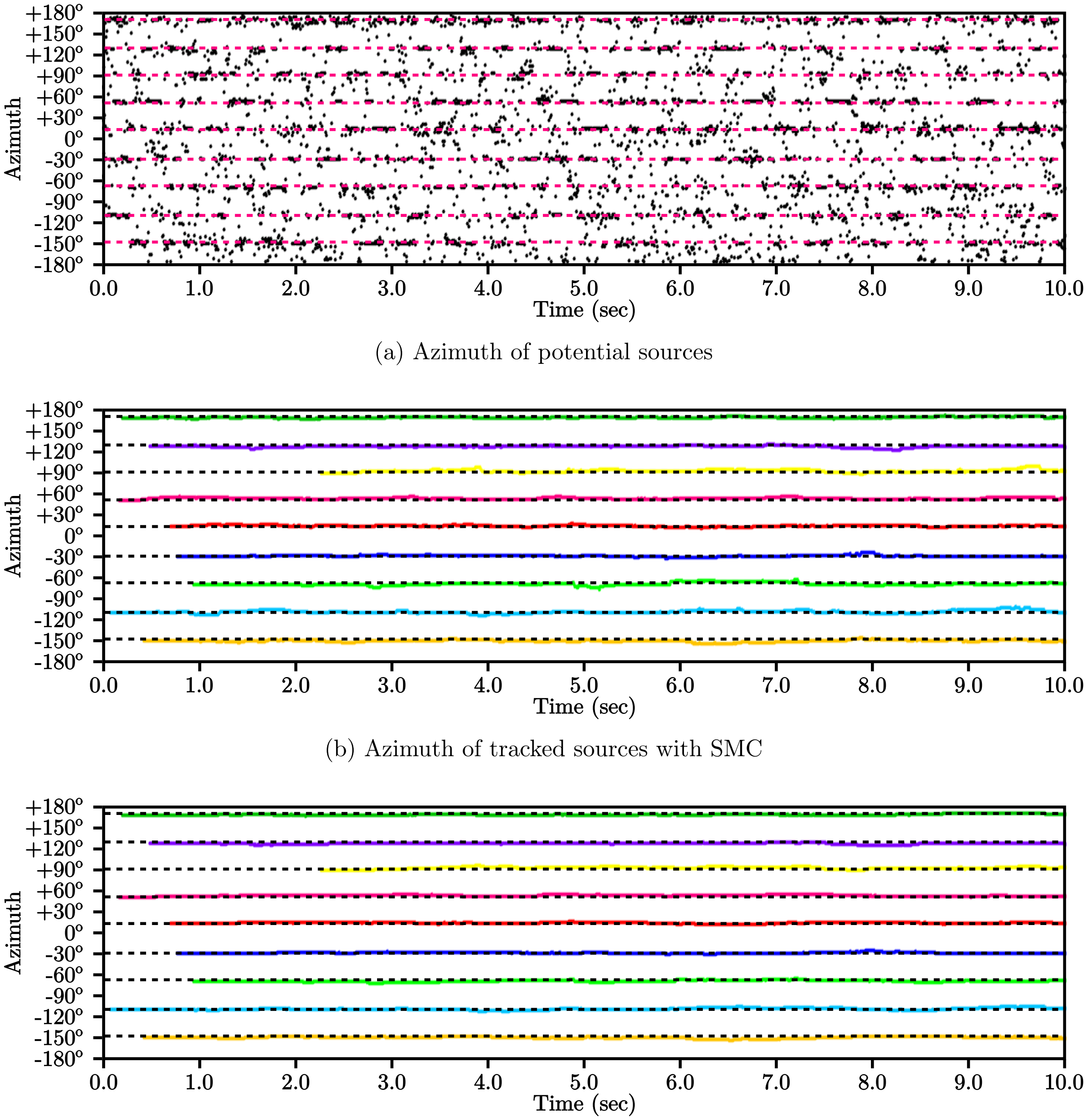


(a) Azimuth of potential sources

(b) Azimuth of tracked sources with SMC

(c) Azimuth of tracked sources with M3K

Figure 4.10 Nine static sources with OMA configuration



(a) Azimuth of potential sources

(b) Azimuth of tracked sources with SMC

(c) Azimuth of tracked sources with M3K

Figure 4.11 Nine static sources with CMA configuration

4.4.3 Moving Sound Sources

Two tests conditions are examined. The first involves four moving sources crossing, tested with OMA and CMA. For this experiment, a male speaker performs four trajectories at $r = 3$ m away from the robot, starting from different positions as illustrated by Fig. 4.12. These recordings are combined to generate the case of four simultaenous moving sources that cross each other.

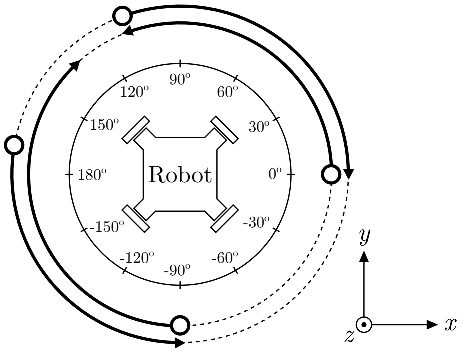
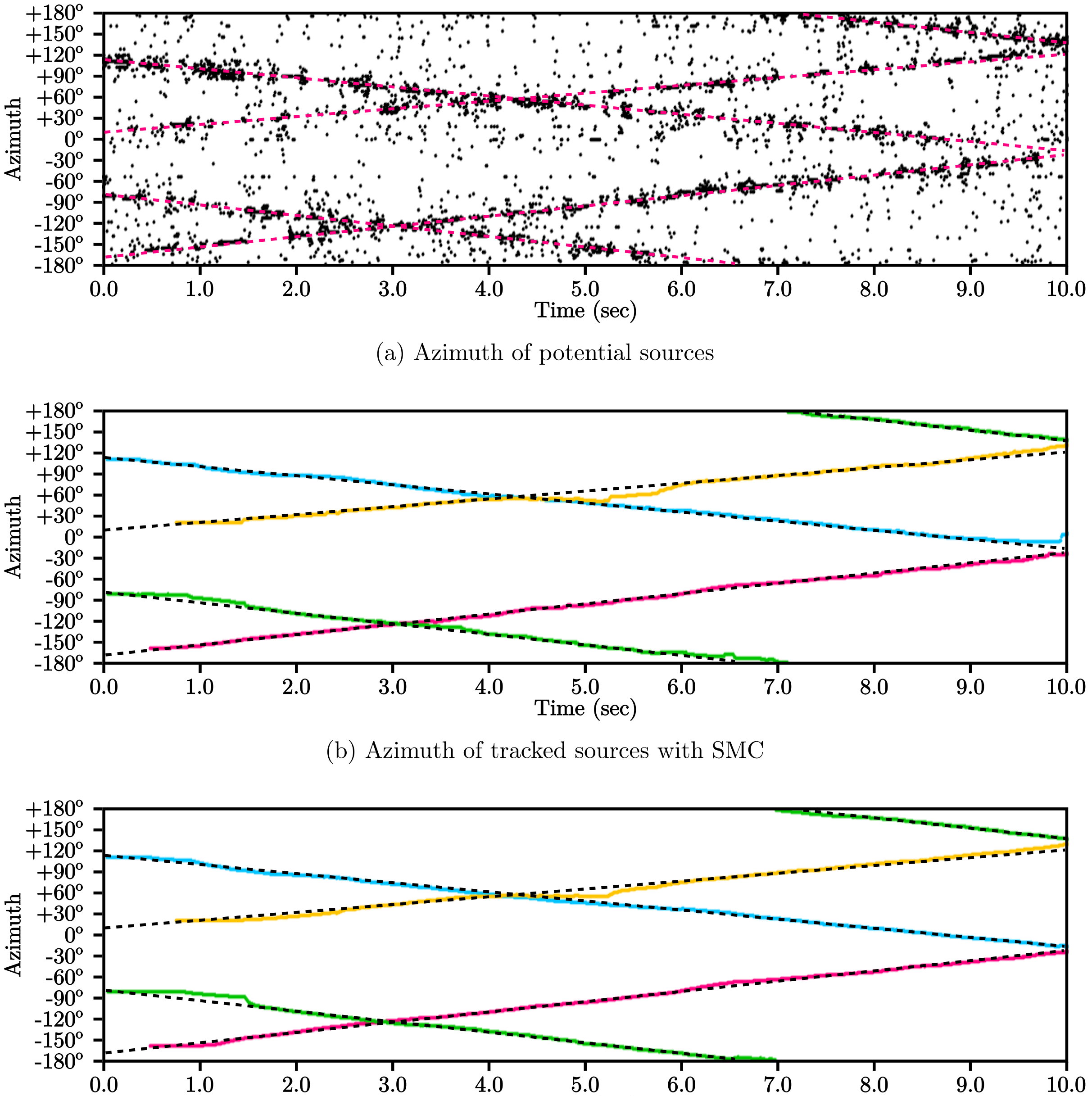


Figure 4.12 Trajectories of four sources crossing

Figure 4.13 and Fig. 4.14 present tracking results using both methods, with the OMA and CMA configurations. Results demonstrate that M3K performs as well as SMC with OMA, and M3K performs better than SMC with CMA. In fact, sources crossing at 5 sec are permuted with SMC, while they keep their respective trajectories with M3K. This is caused by the model dynamics that provides more inertia with M3K than with SMC. There is also a false detection at 6 sec, with the same azimuth and a different elevation, which is due to reverberation from the floor.

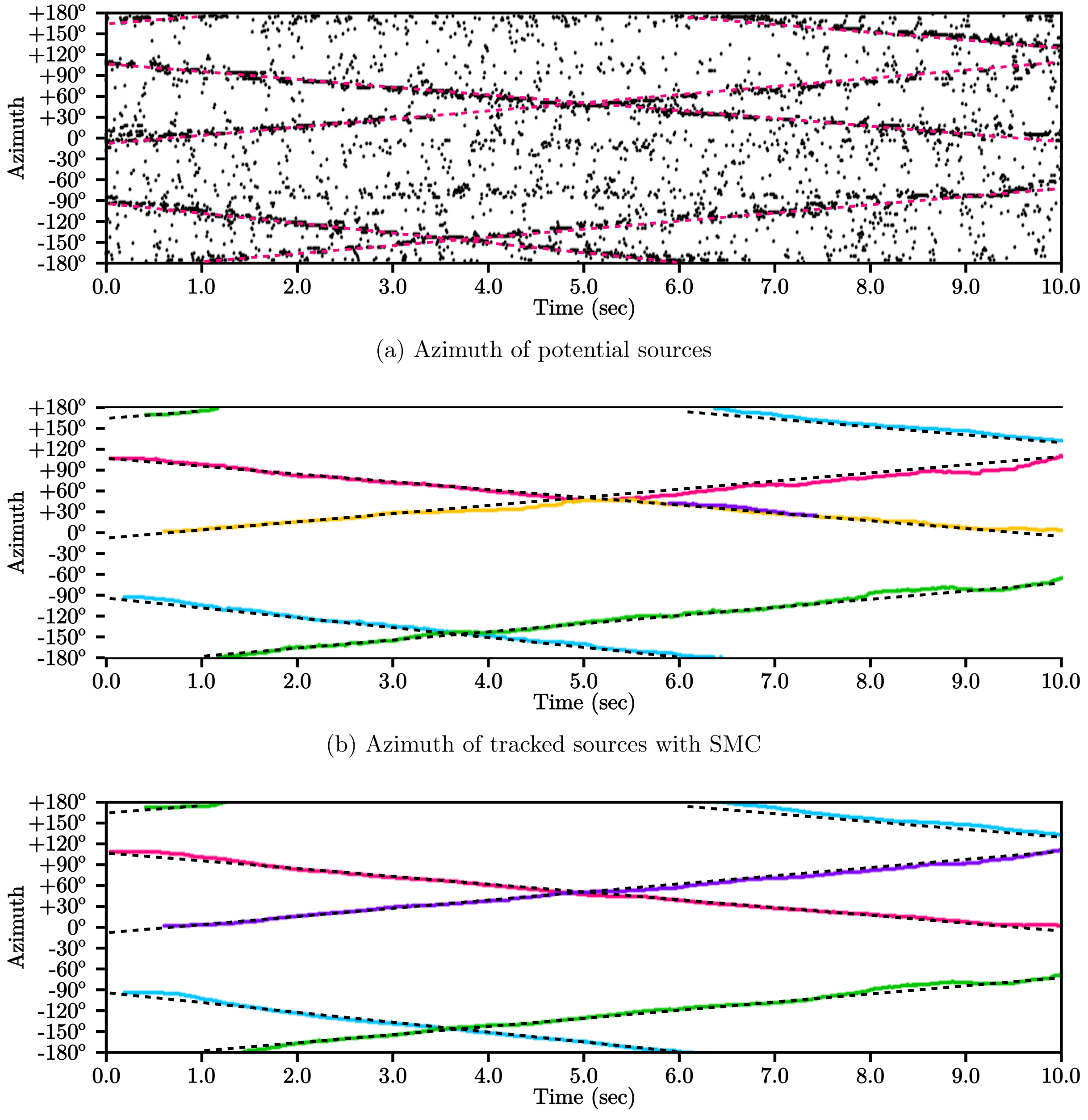


(a) Azimuth of potential sources

(b) Azimuth of tracked sources with SMC

(c) Azimuth of tracked sources with M3K

Figure 4.13 Four crossing sources with OMA configuration



(a) Azimuth of potential sources

(b) Azimuth of tracked sources with SMC

(c) Azimuth of tracked sources with M3K

Figure 4.14 Four crossing sources with CMA configuration

The second test conditions involves moving sound sources following each other. A male speaker performed four trajectories at r = 3 m away from the robot, which are mixed together such that sources are following each other, as shown in Fig. 4.15.

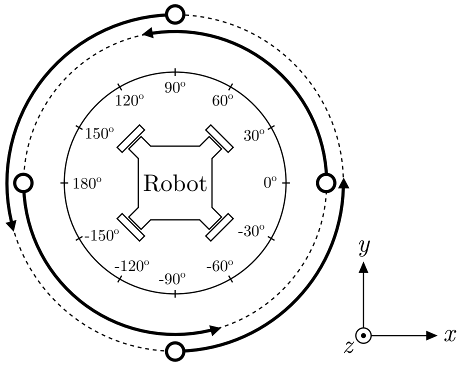
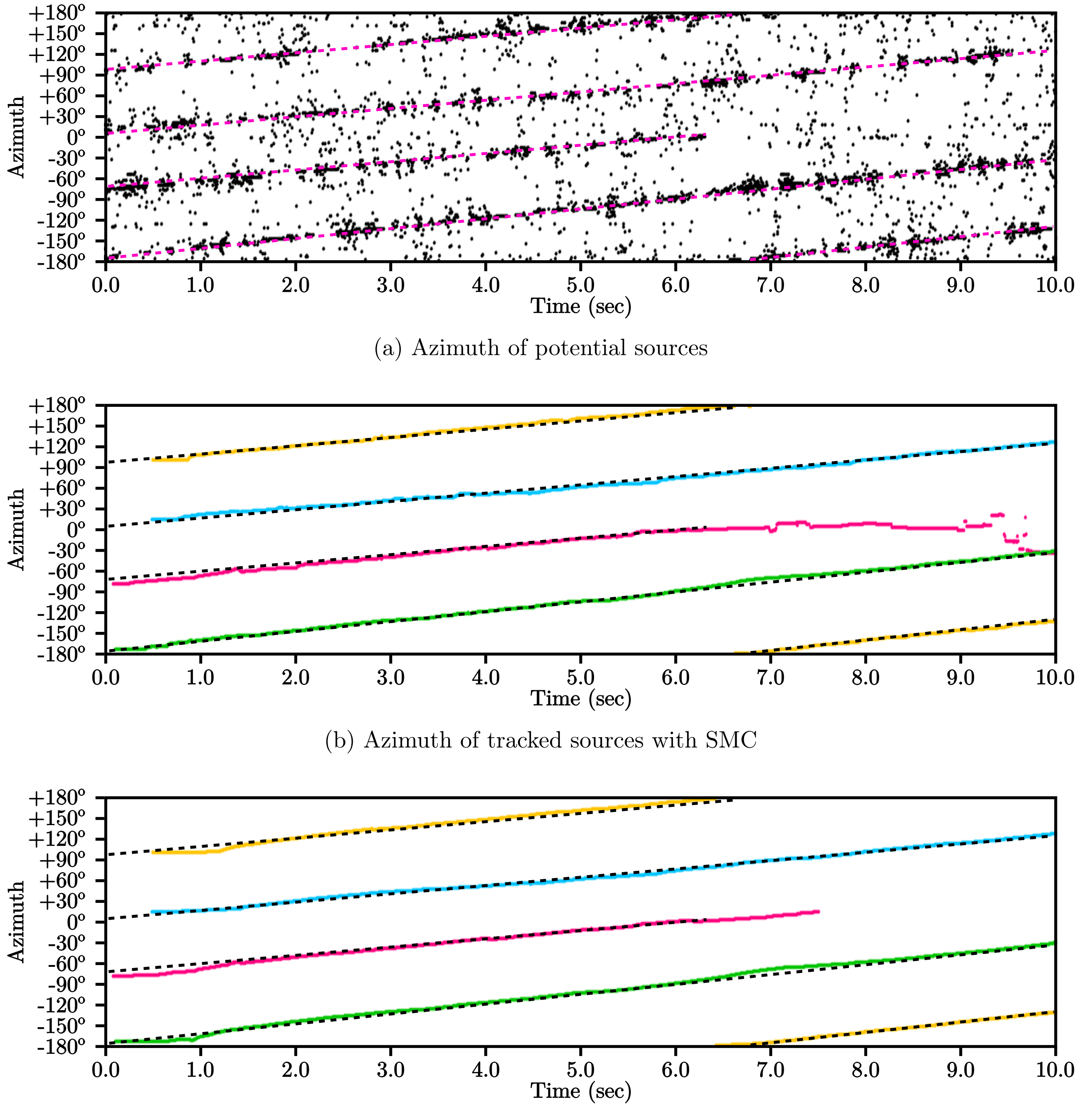


Figure 4.15 Trajectories of four sources following each other

Figure 4.16 and Fig. 4.17 show the potential sources and tracking results with the OMA and CMA configurations. With OMA, both tracking systems perform well, except when one source becomes inactive around 6 sec : M3K removes the source in Fig. 4.16c as required, but SMC keeps tracking it and eventually diverges in the wrong direction, as the particle filter with the parameters proposed in [Valin *et coll.*, 2007a] is more sensitive to noisy observations. Similarly, with CMA, M3K and SMC track the sources accurately, except that both keep tracking the source that becomes inactive around 7 sec. To remove inactive source quickly, the parameter $\mu_{\mathcal{A}}$ defined in Table 4.1 could be increased, at the cost of detecting new sources with more latency.

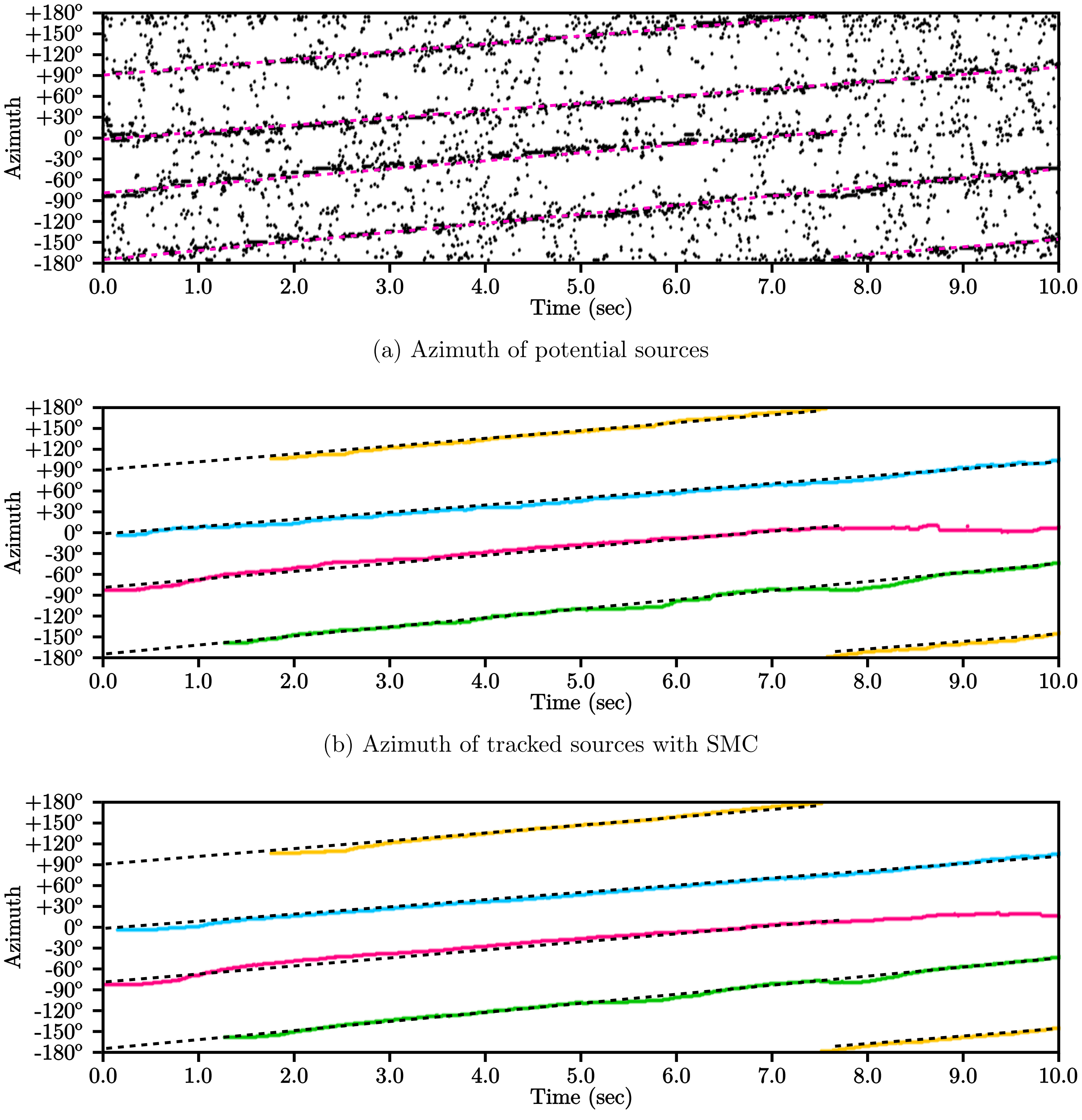


(a) Azimuth of potential sources

(b) Azimuth of tracked sources with SMC

(c) Azimuth of tracked sources with M3K

Figure 4.16 Four speech sources following each other with OMA configuration



(a) Azimuth of potential sources

(b) Azimuth of tracked sources with SMC

(c) Azimuth of tracked sources with M3K

Figure 4.17 Four speech sources following each other with CMA configuration

4.5 Conclusion

This paper describes M3K, a simultaneous sound source 3D tracking method based on Kalman filters. This method provides efficient tracking of sound sources in various conditions (simultaneous static sources, simultaneous moving sources crossing and simultaneous moving sources following each other), with accuracy comparable or better compared to SMC, and reduces by up to 30 times the amount of computations. This efficiency makes the method more appropriate for implementing SST on low-cost embedded hardware.

Currently, M3K relies on a single dynamic model with a constant velocity for the sound sources. Particle filtering provides multiple dynamic models (accelerating sources, sources with constant velocity, and stationary sources [Valin *et coll.*, 2007a]), which may improve tracking performance. As future work, it would be interesting to replace the single Kalman filter with multiple Kalman filters that obey different dynamic models.

Acknowledgment

The authors would like to thank Dominic Létourneau, Cédric Godin and Vincent-Philippe Rhéaume for their help in the experimental setup.

CHAPITRE 5
SÉPARATION DE SOURCES SONORES

Titre : *Adding a Microphone Directivity Model to Improve Sound Source Separation for Mobile Robots*

Auteurs et affiliations :

François Grondin : étudiant au doctorat, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique

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

Date de soumission : 12 octobre 2017

État : Soumis

Revue : *IEEE Transactions on Robotics*

Titre français : Ajout d'un modèle de directivité pour microphone afin d'améliorer la séparation de source sonore pour robots mobiles

Contribution au document : Cet article contribue à la thèse en introduisant un modèle de directivité des microphones aux méthodes existantes de séparation de sources sonores, ce qui augmente le rapport signal sur bruit jusqu'à 2.7 dB lorsqu'un robot utilise une matrice de microphones fermée.

Résumé en français : Dans le domaine de l'audition robotique, la séparation de sources sonores consiste à séparer des sources les unes des autres à partir des signaux d'une matrice de microphones, ce qui améliore la qualité sonore pour effectuer plus efficacement l'identification de locuteur, la reconnaissance vocale et d'autres tâches. L'installation de microphones sur un robot mobile implique parfois des contraintes au niveau de la géométrie de la matrice, et, dans certains cas, le chemin direct de propagation du son entre les sources sonores et les microphones est obstrué par le torse du robot. Dans cet article, un modèle de directivité pour les microphones est proposé afin d'améliorer la qualité de séparation lorsque cette situation est observée. Ce modèle est appliqué aux méthodes de formateur de faisceau délais et somme, de séparation géométrique de source sonore et de formateur de faisceau avec une réponse sans distortion et avec variance minimale. Des

expériences sont effecutées sur un robot muni d'une matrice de 16 microphones, et il est démontré que le modèle de directivité proposé améliore le rapport signal sur bruit durant la séparation jusqu'à 2.7 dB pour une matrice fermée.

Abstract

In robot audition, sound source separation consists of separating sound sources from each other using signals captured by multiple microphones, enhancing source source audio streams for improved processing for identification, recognition or other processing. Placing such microphones on a mobile robot often introduces constraints on the microphone array geometry, and in some configurations, the direct path between the microphones and the sound sources can be obstructed by the robot's body. In this paper, a microphone directivity model is introduced to improve separation performance in such conditions. This model is applied to the existing Delay-and-Sum beamformer, Geometric Source Separation method, and the Minimum Variance Distortionless Response beamformer. Experiments are performed using a robot equipped with a 16-microphone array, and results demonstrate that the microphone directivity model improves the separation Signal-to-Noise Ratios by up to 2.7 dB for a closed compared to an open microphone array configuration.

5.1 Introduction

The cocktail party effect (CPE) is the ability to focus on a specific conversation while other conversations and noise are present in the background [Arons, 1992]. From a signal processing perspective, the CPE consists of multiple independent sound sources convolved with Room Impulse Responses (RIRs), mixed together to generate observations captured by one or many microphones. Sound source separation (SSS) implies recovering each individual input source with no interference from the other input sources, at the cost of introducing some artifacts in the spectral envelope of the source. Perfect separation, i.e., having no interference from other sources, requires the robot audition system to have more microphones than sources [Furuya, 2001], which implicitly calls for installing microphone arrays on robots. However, in practice, reverberation and additive noise make perfect separation difficult to achieve. The objective therefore consists in maximizing the Signal-to-Noise Ratio (SNR) to get as close as possible to a perfect separation.

Figure 5.1 illustrates how SSS usually consists of a separation step that uses the Directions of Arrival (DoAs) of sound sources, followed by a post-filtering step to reduce further the amount of noise and interference based on the targeted type of signals (e.g., speech). Typical separation methods are Delay-and-Sum (DS) beamformer [Yardibi *et coll.*, 2010], Geometric Sound Separation (GSS) beamformer [Parra et Alvino, 2002] and Minimum Variance Distortionless Response (MVDR) beamformer [Vorobyov, 2013]. They all assume that all microphones capture signals from all sound sources. However, this hypothesis is invalid when microphones are placed on the robot's body. In fact, the quality of the separated signals for the targeted sources decreases when separation includes microphone signals dominated by interference sources. In this case, it is desirable to ignore these noisy observations during the separation process.

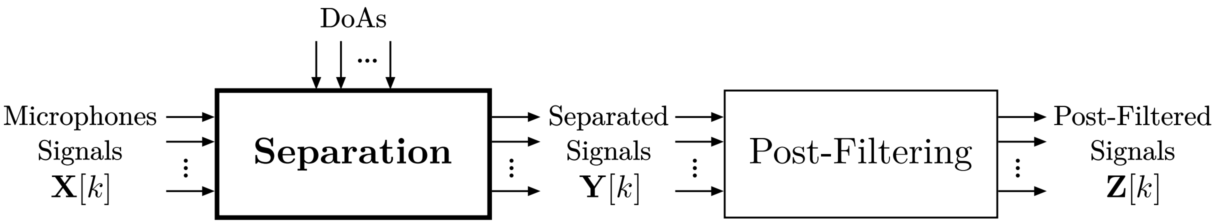


Figure 5.1 Block diagram of SSS made of separation and post-filtering steps in cascade

Therefore, this paper introduces the use of a directivity model that modulates the gain of each microphone based on its orientation with respect to the DoAs of the sound sources. This chapter is organized as follows. Section 5.2 presents a brief review of separation and post-filtering methods used to achieve SSS. Section 5.3 explains how the directivity model can be applied to DS, GSS and MVDR beamforming separation methods. Section 5.4 describes the experimental setup using a real robot, followed by Section 5.5 with the observed results.

5.2 Brief Review of SSS Methods

For many separation methods, the exact number of active sound sources needs to be known. For instance, the TRINICON framework aims to perform separation in the time domain for a known number of sources [Buchner *et coll.*, 2004b]. This method relies on non-gaussianity, non-stationnarity and coloration of speech signals. TRINICON depends on an optimization process that requires many seconds of speech to converge, which introduces undesirable latency. Moreover, as the reverberation level gets higher, the RIR involves more coefficients over time and this increases the number of computations, which makes the method less suitable for online applications. TRINICON can also be adapted to perform separation in the frequency domain [Kellermann *et coll.*, 2006]. Independent Component Analysis (ICA) also performs separation in the frequency domain. This method assumes the source signals exhibit a non-gaussian distribution and the number of microphones matches the number of sources [Hyvärinen et Oja, 2000]. When dealing with an overdetermined system, Principal Component Analysis (PCA) can be used to reduce the number of observations such that it matches the number of sources [Abdi et Williams, 2010]. However, methods in the frequency domain experience permutation and scaling errors across the frequency spectrum, which can be coped using DoA estimation and other repair mechanisms.

There are however methods which alleviate the requirement of knowing the exact number of active sound sources, allowing the robot audition system to be underdetermined (i.e., have fewer microphones than the number of sound sources). The Delay-and-Sum (DS) beamforming method [Yardibi *et coll.*, 2010] relies on the DoA of sound sources and performs computation in the time or frequency domains. The time-domain implementation involves fewer computations, which makes it appealing for real-time applications on low-cost embedded systems. Geometric Source Separation (GSS) exploits the independence of the sound sources and the geometric constraint related to the DoA of each source [Parra et Alvino, 2002]. It aims to provide a unit gain in the direction of the target source and a zero gain in the direction of interfering sources, while ensuring separated signals are independent. Valin et al. [Valin *et coll.*, 2004b] adapted the method to regularize the demixing terms and accelerate the cross-correlation matrix estimation. Moreover, Nakajima et al. [Nakajima *et coll.*, 2008a,b] present a method to adapt continuously the step size to ensure convergence of the optimization process. The geometric constraint in GSS relies on either the sound direct path propagation model or the robot Head-Related Transfer Functions (HRTF) [Pedersen *et coll.*, 2004]. When the sound source moves, Nakadai et al. [Nakadai *et coll.*, 2010a] propose to load dynamically the corresponding HRTFs and adapt smoothly the separation over time. Finally, MVDR beamforming method aims to improve the performance of DS beamforming by using the DoA and minimizing the power of noise and interference at the output of the beamformer [Vorobyov, 2013]. But this method requires the inversion of the observation correlation matrix for each frequency bin, which can become challenging in terms of computations when the number of microphones in the array increases. Regularization also solves the singularity issued that may occur due to matrix inversion [Vorobyov, 2013]. MVDR leads to better performance when HRTF between the source and the microphones are used instead of the free propagation model used with the DS beamformer [Barfuss *et coll.*, 2017]. This method however requires manual measurements of the transfer function for each direction around the robot, which makes it difficult to quickly adapt the method to different robots.

In SSS, post-filtering is used to attenuate further interference and noise in the separated signals and improve the SNR. The post-filter proposed by Zelinski [Zelinski, 1988] uses a Wiener filter that relies on auto- and cross-correlation. Zelinski's method assumes zero correlation between noise between the microphones, though in practice this is often not the case. To include this condition, McCowan and Bourlard [McCowan et Bourlard, 2003] present a new post-filter that considers the complex coherence of the noise field. These techniques rely on the estimation of the cross-correlation matrix for the M microphones and K frequency bins, that leads to a complexity order of $\mathcal{O}(M^2K)$, which increases significantly the number of computations for large microphone arrays. Valin et al. [Valin *et coll.*, 2004b] also proposes a post-filtering method that relies on the SNR estimated from the interfering sources and stationary background noise. The interfering sources spectra is obtained through separation and the background noise is estimated via the Minima-Controlled Recursive Averaging (MCRA) [Cohen, 2003]. However this post-filtering method requires to track all active sound sources, which may not be possible in some noisy environments, and ignores non-stationary diffuse noise, such as the sound emitted by the robot actuators (referred to as ego noise). To deal with ego noise, Ince et al. [Ince *et coll.*, 2009, 2010] propose template subtraction and post-filtering methods that predict the actuators' noise spectral content based on the motion sequences. Since the model trained offline may differ from the online observations in various environments, an incremental learning method updates its database to estimate ego noise more accurately [Ince *et coll.*, 2011a]. However, these methods based on ego noise estimation need to have access to the motion sequences, which are not always available on some robots. Sparse coding based on Non-Negative Matrix Factorization (NMF) is an interesting solution as it can estimate the actuators' noise independently of motion sequences [Wilson *et coll.*, 2008]. Deleforge and Kellermann [Deleforge et Kellermann, 2015] also propose a Phase-Optimized K-SVD method to recover speech corrupted by actuator noises, trained with the ego noise generated by the robot. Instead of training with ego noise, Wood and Rouat [Wood et Rouat, 2016, 2017] introduce the GCC-NMF method that trains the database with speech signals, and post-filter the separated signal to enhance the speech spectral components. But overall, regardless of the method used, post-filtering performance depends on the quality of the separation step, which we aim to improve with the use of a microphone directivity model.

5.3 Microphone Directivity Model Applied to DS, GSS and MVDR

Before introducing the microphone directivity model, let us provide and define the concepts related to beamforming. Consider an array of M-microphones generates M synchronous signals. These signals x_m are divided in frames of N samples, spaced by ΔN samples and multiplied by a the square-root Hann window $w[n]$:

$$x_m^l[n] = w[n]x_m[n + l\Delta N] \tag{5.1}$$

with l, m and n representing the frame, microphone and sample indexes, respectively. For each signal $x_m^l[n]$, the Short-Time Fourier Transform (STFT) is computed using a N-samples real Fast Fourier Transform (FFT), where the expression $X_m^l[k]$ stands for the spectrum at each frequency bin k :

$$X_m^l[k] = \sum_{n=0}^{N-1} x_m^l[n] \exp\left(\frac{-j2\pi kn}{N}\right) \tag{5.2}$$

The vector $\mathbf{X}[k]$ represents the observations in the frequency domain for all M microphones :

$$\mathbf{X}^l[k] = \begin{bmatrix} X_1^l[k] & X_2^l[k] & \dots & X_M^l[k] \end{bmatrix}^T \tag{5.3}$$

where $(\dots)^T$ stands for the transpose operator.

The observed signals $X_m^l[k]$ at each microphone m usually depends on the active sound sources $S_i^l[k]$, the transfer function $A_{mi}^l[k]$ between each source i and each microphone m, and some additive noise $B_m^l[k]$ observed at each microphone m. These elements therefore lead to the following model :

$$X_m^l[k] = \sum_{i=1}^{I} A_{mi}^l[k] S_i^l[k] + B_m^l[k] \tag{5.4}$$

This can also be expressed in the following matrix form :

$$\mathbf{X}^l[k] = \mathbf{A}^l[k]\mathbf{S}^l[k] + \mathbf{B}^l[k] \tag{5.5}$$

where the $M \times I$ transfer functions matrix corresponds to :

$$\mathbf{A}^l[k] = \begin{bmatrix} \mathbf{A}_1^l[k] & \mathbf{A}_2^l[k] & \dots & \mathbf{A}_I^l[k] \end{bmatrix} \tag{5.6}$$

where each transfer function vector for source i, also referred to as steering vector, is :

$$\mathbf{A}_i^l[k] = \begin{bmatrix} A_{1i}^l[k] & A_{2i}^l[k] & \dots & A_{Mi}^l[k] \end{bmatrix}^T \tag{5.7}$$

The separated signals $\mathbf{Y}^l[k]$ are obtained from the demixing matrix $\mathbf{W}^l[k]$ and the microphone signals $\mathbf{X}^l[k]$:

$$\mathbf{Y}^l[k] = \mathbf{W}^l[k]\mathbf{X}^l[k] \tag{5.8}$$

where

$$\mathbf{Y}^l[k] = \left[\begin{array}{cccc} Y_1^l[k] & Y_2^l[k] & \dots & Y_I^l[k] \end{array} \right]^T \tag{5.9}$$

and

$$\mathbf{W}^l[k] = \left[\begin{array}{cccc} (\mathbf{W}_1^l)^T & (\mathbf{W}_2^l)^T & \dots & (\mathbf{W}_I^l)^T \end{array} \right]^T \tag{5.10}$$

which holds all the demixing elements W_{im}^l between source i and microphone m :

$$\mathbf{W}_i^l[k] = \left[\begin{array}{cccc} W_{i1}^l[k] & W_{i2}^l[k] & \dots & W_{iM}^l[k] \end{array} \right]^T \tag{5.11}$$

The purpose of the microphone directivity model is to provide a more accurate model for $\mathbf{A}^l[k]$, which in turn generates a more effective demixing matrix $\mathbf{W}^l[k]$. Given the unit vector $\mathbf{u}_i$ that points in the direction of the source i, and the microphone m located at $\mathbf{m}_m$, the expected propagation delay (in samples) referenced to the origin at the center of mass of the array corresponds to :

$$\tau_m(\mathbf{u}_i^l) = \left(\frac{f_S}{c}\right) \mathbf{m}_m \cdot \mathbf{u}_i^l \tag{5.12}$$

where f_S and c stand for the sample rate (in samples/sec) and the speed of sound (in m/sec), respectively. In the microphone directivity model, the RIR has a gain that depends on the microphone orientation in relation to the tracked sound source direction, as shown in Fig. 5.2. The angle θ corresponds to the following expression :

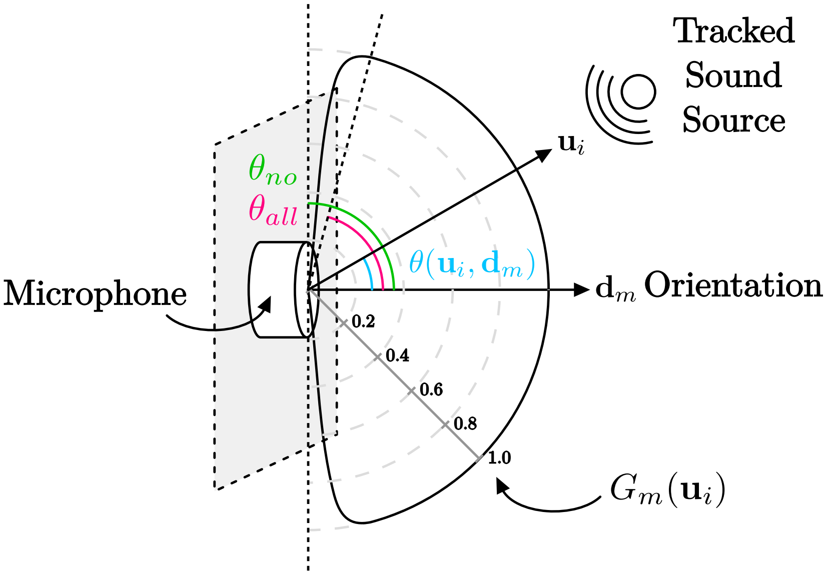


Figure 5.2 Angle between the tracked source and the microphone orientation

$$\theta(\mathbf{u}_i^l, \mathbf{d}_m) = \arccos\left(\frac{\mathbf{u}_i^l \cdot \mathbf{d}_m}{|\mathbf{u}_i^l||\mathbf{d}_m|}\right) \tag{5.13}$$

and the gain response $G_m(\mathbf{u}_i)$ of microphone m is a function of the angle $\theta(\mathbf{u}_i, \mathbf{d}_m)$:

$$G_m(\mathbf{u}_i^l) = \frac{1}{1 + \exp(\alpha_m(\theta(\mathbf{u}_i^l, \mathbf{d}_m) - \beta_m))} \tag{5.14}$$

$$\alpha_m = 20/((\theta_{no})_m - (\theta_{all})_m) \tag{5.15}$$

$$\beta_m = ((\theta_{no})_m + (\theta_{all})_m)/2 \tag{5.16}$$

The true beampattern of the microphone may differ from this function, as it usually depends on the type of microphone and the complex shape of the robot's body. In practice however, this function models with sufficient precision the microphone beampattern when installed on a rigid body. The parameters α_m and β_m stand for the steepness and the midpoint value of the curve, as defined by (5.15) and (5.16). The expression $(\theta_{all})_m$ stands for the angle where the gain is one, while $(\theta_{no})_m$ corresponds to the angle at which the gain is null. The region between both angles can be viewed as a transition band.

The steering vector $\mathbf{A}_i^l[k]$ introduced in (5.6) thus includes the gain $G_m(\mathbf{u}_i^l)$ and propagation delay τ_{im}^l.

$$A_{mi}^l[k] = G_m(\mathbf{u}_i^l) \exp\left(\frac{-j2\pi k \tau_m(\mathbf{u}_i^l)}{N}\right) \tag{5.17}$$

The following subsections explain how a microphone directivity model can be added to DS, GSS and MVDR separation methods to influence G_m based on the the contribution of the signals coming only from the microphones that can enhance the separated signal. These variants are referred to as Directional-DS (D-DS), Directional GSS (D-GSS) and Directional MVDR (D-MVDR), and their block diagrams are shown in Fig. 5.3a, Fig. 5.3b and Fig. 5.3c, respectively. When all gains G_m equal 1, these variants are the equivalent of the standard methods.

5.3.1 Directional DS (D-DS)

Adding the microphone directivity model to a DS beamformer aims to enhance the signal in the direction of the target source with constructive interference. Given $\mathbf{A}_{i'}^l[k]$, the demixing matrix elements $W_{i'm}^l$ for the D-DS beamformer are chosen to ensure a unitary gain in the

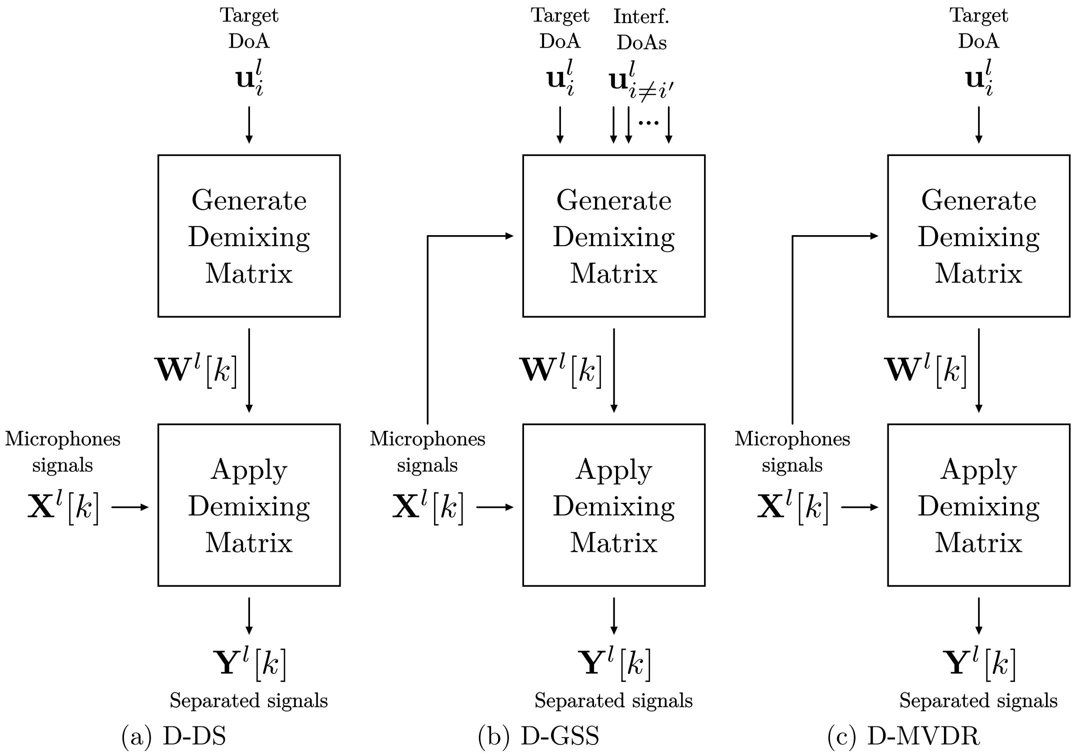


Figure 5.3 Block diagrams of D-DS, D-GSS and D-MVDR separation methods

direction of the target source :

$$\mathbf{W}_{i'}^l[k]\mathbf{A}_{i'}^l[k] = \sum_{m=1}^{M} W_{i'm}^l[k]A_{mi'}^l[k] = 1 \tag{5.18}$$

The following expression satisfies the previous condition :

$$W_{i'm}^l[k] = \frac{G_m(\mathbf{u}_{i'}^l)}{\sum_{m=1}^{M}(G_m(\mathbf{u}_{i'}^l))^2}\exp\left(\frac{j2\pi k\tau_m(\mathbf{u}_{i'}^l)}{N}\right) \tag{5.19}$$

Note that the DS beamforming method is a specific case of D-DS when all gains G_m are set to 1, and the scalar value simplifies to $1/M$. The DS and D-DS beamforming methods are appealing for real-time applications on low-cost hardware as they can be implemented in the time domain with interpolation for fractional delays :

$$y_{i'}[n] = \sum_{m=1}^{M}\left(\frac{G_m(\mathbf{u}_{i'}^l)}{\sum_{m=1}^{M}(G_m(\mathbf{u}_{i'}^l))^2}\right)x[n-\tau_m(\mathbf{u}_{i'}^n)] \tag{5.20}$$

where the DoA $\mathbf{u}_{i'}^{n}$ corresponds to the average of DoAs $\mathbf{u}_{i'}^{l}$ of frames l in which sample n falls. Moreover, when the gain G_m reached zero, the microphone signal x_m can be ignored, which further reduces the amount of computations.

5.3.2 Directional GSS (D-GSS)

D-GSS uses the demixing elements of the D-DS beamformer at initialization, and then optimizes them over time to minimize two cost functions :

$$J_1 = \|\mathbf{R}_{yy}^{l}[k] - \mathrm{diag}[\mathbf{R}_{yy}^{l}[k]]\|^2 \tag{5.21}$$

$$J_2 = \|\mathbf{W}^{l}[k]\mathbf{A}^{l}[k] - \mathbf{I}\|^2 \tag{5.22}$$

where the matrix norm is defined as $\|\mathbf{C}\|^2 = \mathrm{trace}[\mathbf{C}\mathbf{C}^H]$, and the expression $\{\ldots\}^H$ stands for the Hermitian operator.

The cost function J_1 aims to maximize the independence between separated sources. Since speech sources are non-stationary, setting their cross-correlation to zero guarantees independence [Valin *et coll.*, 2004b]. The function J_2 ensures unity gain in the direction of the target source and null gains in the direction of interfering sources. The transfer functions in the vector $\mathbf{A}^{l}[k]$ in the cost function J_2 include the gains G_m, which account for the microphone directivity. The gradient descents minimize the cost functions J_1 and J_2 :

$$\frac{\partial J_1(\mathbf{W}^{l}[k])}{\partial \mathbf{W}^{*}[k]} = 4\left[\mathbf{E}[k]\mathbf{W}[k]\mathbf{X}^{l}[k]\right](\mathbf{X}^{l}[k])^H \tag{5.23}$$

$$\frac{\partial J_2(\mathbf{W}^{l}[k])}{\partial \mathbf{W}^{*}[k]} = 2\left[\mathbf{W}[k]\mathbf{A}^{l}[k] - \mathbf{I}\right](\mathbf{A}^{l}[k])^H \tag{5.24}$$

where $\mathbf{E}[k] = \mathbf{R}_{yy}^{l}[k] - \mathrm{diag}[\mathbf{R}_{yy}^{l}[k]]$.

The demixing matrix is updated according to the adaptation rate μ, and the regularization factor λ :

$$\mathbf{W}^{l+1}[k] = (1-\lambda\mu)\mathbf{W}^{l}[k] - \mu\frac{\partial J(\mathbf{W}^{l}[k])}{\partial \mathbf{W}^{*}[k]} \tag{5.25}$$

where

$$\frac{\partial J(\mathbf{W}^{l}[k])}{\partial \mathbf{W}^{*}[k]} = \alpha[k]\frac{\partial J_1(\mathbf{W}^{l}[k])}{\partial \mathbf{W}^{*}[k]} + \frac{\partial J_2(\mathbf{W}^{l}[k])}{\partial \mathbf{W}^{*}[k]} \tag{5.26}$$

and the expression $\alpha[k]$ stands for the energy normalization factor :

$$\alpha[k] = \left\|\mathbf{X}^{l}[k](\mathbf{X}^{l}[k])^H\right\|^{-2} \tag{5.27}$$

The parameter μ must be chosen carefully : it needs to be large enough to provide quick adaptation, but small enough to ensure convergence during the optimization process.

5.3.3 Directional MVDR (D-MVDR)

D-MVDR generates the demixing vector $W_i^l[k]$ that maximizes the Signal-to-Interference-plus-Noise Ratio (SINR). The goal consists in maximizing the power in the target source direction, while minimizing the power from all other directions. D-MVDR relies on the estimation of the correlation matrix $R_{xx}^l[k]$, that can be obtained recursively as follows :

$$\mathbf{R}_{xx}^l[k] = (1-\alpha)\mathbf{R}_{xx}^{l-1}[k] + \alpha(\mathbf{X}^l[k](\mathbf{X}^l[k])^H) \tag{5.28}$$

where the parameter α stands for the adaptation rate.

Regularization prevents singularities when inverting the correlation matrix, where the parameter $\sigma > 0$ weights the contribution of the correlation matrix in relation to the $M \times M$ identity matrix $\mathbf{I}$:

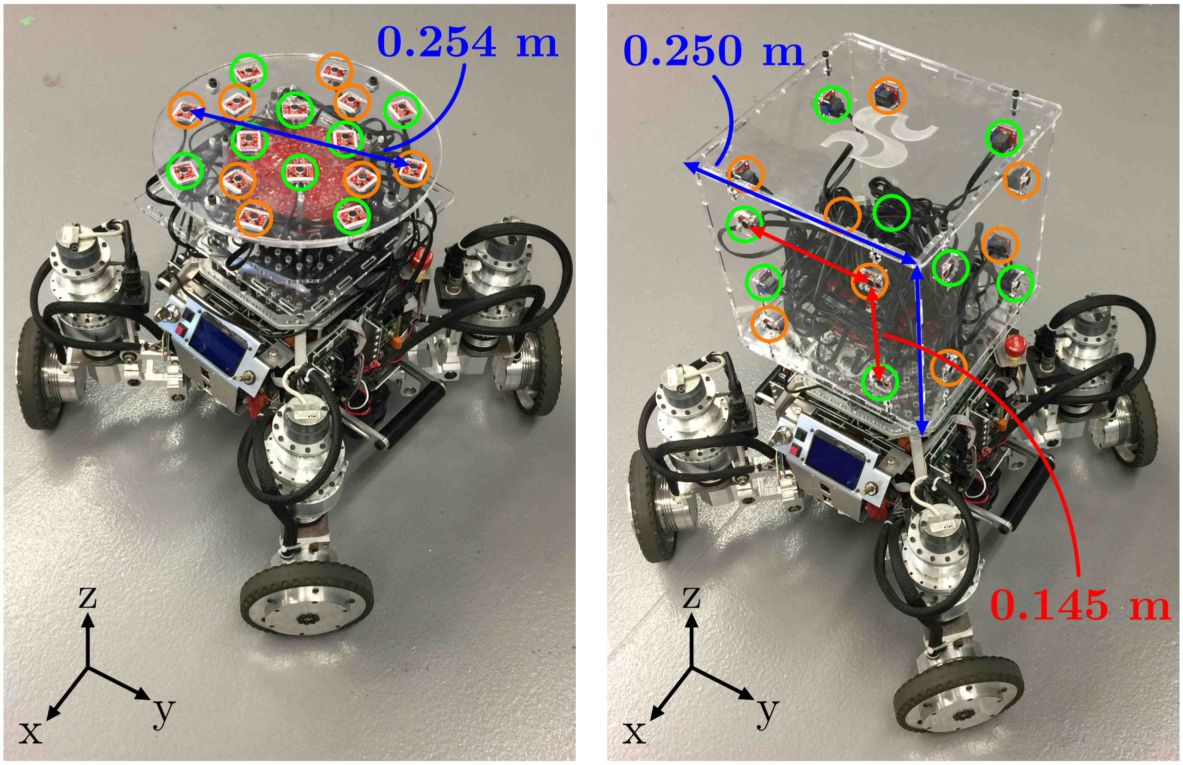
$$\hat{\mathbf{R}}_{xx}^l[k] = (1-\sigma)\mathbf{R}_{xx}^l[k] + \sigma\mathbf{I} \tag{5.29}$$

$$\mathbf{W}_i^l[k] = \frac{(\hat{\mathbf{R}}_{xx}^l[k])^{-1}(\mathbf{A}_i^l[k])^*}{(\mathbf{A}_i^l[k])^T(\hat{\mathbf{R}}_{xx}^l[k])^{-1}(\mathbf{A}_i^l[k])^*} \tag{5.30}$$

Note that the D-MVDR is a generalization of the MVDR ($\sigma \neq 1$, $G_m = 1$), the D-DS ($\sigma = 1$, $G_m \neq 1$) and DS ($\sigma = 1$, $G_m = 1$) methods.

5.4 Experimental Setup

Experiments are performed with two configurations of microphone arrays installed on a robot. In this first configuration, named Opened Microphone Array (OMA), all microphones lie on the same circular surface with a diameter of 0.254 m and face the ceiling, as shown in Fig. 5.4a. The second configuration, referred to as Closed Microphone Array (CMA), consists of a cubic shape with four microphones installed on each lateral face, as shown in Fig. 5.4b. The four microphones on each face make a square with 0.145 m edges, and are mounted on a square surface with 0.250 m edges. The green microphones are used for experiments performed with only 8 microphones, while all microphones (both green and orange) are used for experiments with 16 microphones.



(a) OMA

(b) CMA

Figure 5.4 Microphone array configurations on a mobile robot

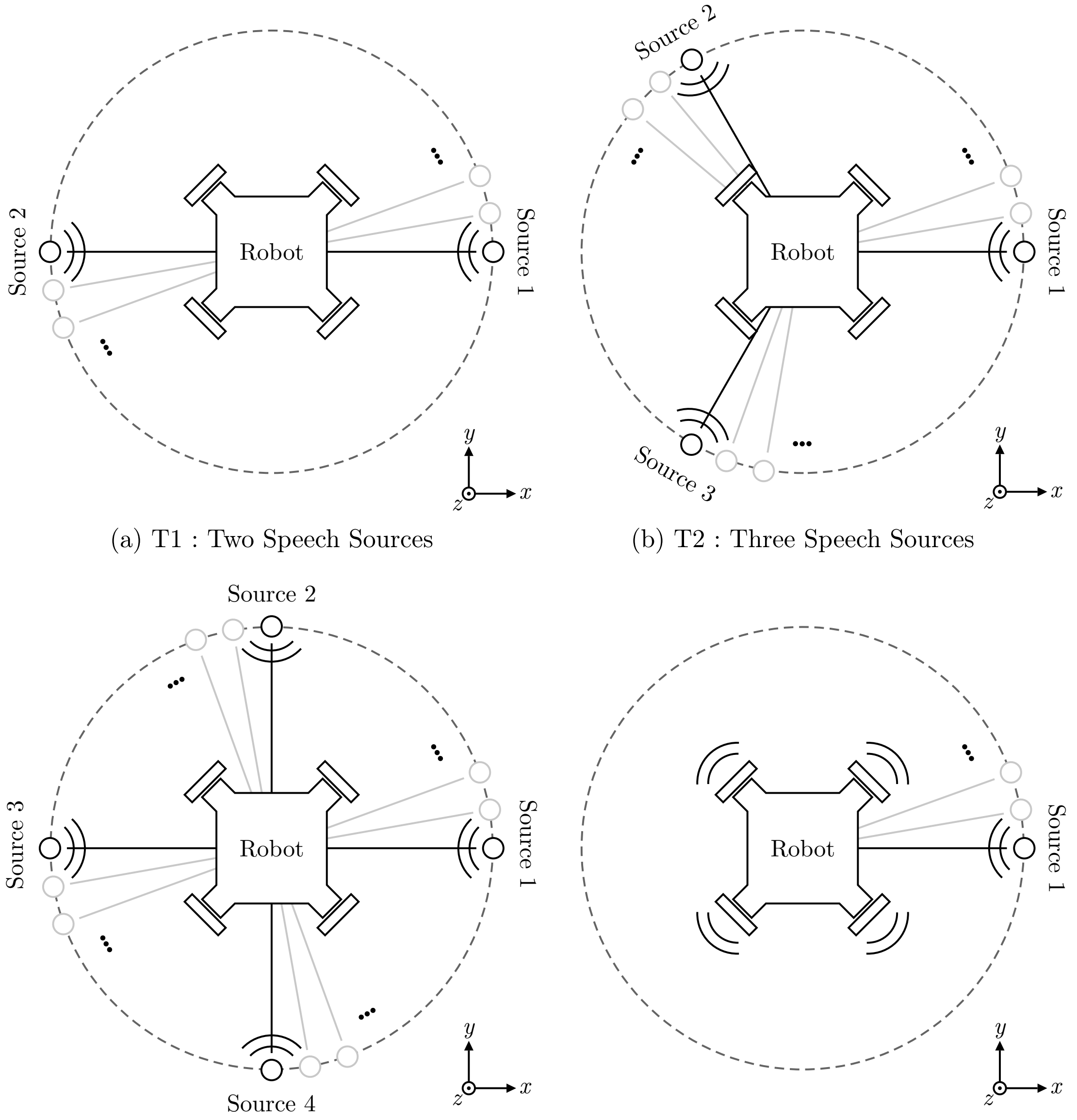
Four test conditions are investigated to measure the performances of each separation method :

T1 *Two Speech Sources* : Both sources are positioned around the robot, and spaced by a azimuth angle of 180°. Experiments are performed for each 10° angle interval around the robot, e.g., Source 1 at 0° and Source 2 at 180°, Source 1 at 10° and Source 2 at 190°, and so on. Figure 5.5a illustrates this test condition.

T2 *Three Speech Sources* : This is the same as T1, but there are three speech sources spaced evenly by 120°, as shown in Figure 5.5b.

T3 *Four Speech Sources* : Same as T1 and T2, but with four speech sources spaced evenly by 90°, as in Figure 5.5c.

T4 *One Speech Source and Robot Motors* : The robot generates noise when it moves due to its actuators, and the characteristics of this noise depend on the motion produced by the robot. In this test condition, noise is recorded when the robot performs various motions, and then mixed with a speech sound source positioned around the robot, as shown in Figure 5.5d.



(a) T1 : Two Speech Sources

(b) T2 : Three Speech Sources

(c) T3 : Four Speech Sources

(d) T4 : One Speech Sources and Motors

Figure 5.5 Illustration of the test conditions

Table 5.1 presents the parameters used in the experiments. The audio signals are processed at the fixed sample rate f_S. The parameter c corresponds to the expected speed of sound at room temperature. The frame size N corresponds to speech stationarity duration, and the hop size ensures an overlap of 50%. The microphone directivity parameters are chosen to provide a beampattern with the shape of a hemisphere with a transition band of 20°. The D-GSS and GSS parameters μ and λ are identical to those proposed by Valin et al. [Valin *et coll.*, 2004b]. The parameters α and σ are chosen to adapt fast enough to non-stationary noise, and to avoid singularity when inverting correlation matrices.

Tableau 5.1 Parameters used in the experiments

Methods	Parameters	Values
DS, D-DS, GSS, D-GSS, MVDR & D-MVDR	M	8 or 16
	f_S	16000 samples/sec
	c	343.0 m/sec
	N	256
	ΔN	128
D-DS, D-GSS & D-MVDR	$(\theta_{no})_m$	80°
	$(\theta_{all})_m$	100°
GSS & D-GSS	μ	0.01
	λ	0.5
MVDR & D-MVDR	α	0.1
	σ	0.01

5.5 Results

Male and female speakers who pronounce different utterances are used as sound sources. The room has a reverberation level of $RT_{60} = 600$ msecs and no background noise. The performance of each method is evaluated from individual source recordings performed in a large room, which are then mixed together to create various test conditions with multiple sound sources. The target source $S^l_{i'}[k]$ first plays in the loudspeaker in the room and the microphone array captures these observations, referred to as $(X_{target})^l_m[k]$:

$$(X_{target})^l_m[k] = A^l_{mi'}[k]S^l_{i'}[k] \tag{5.31}$$

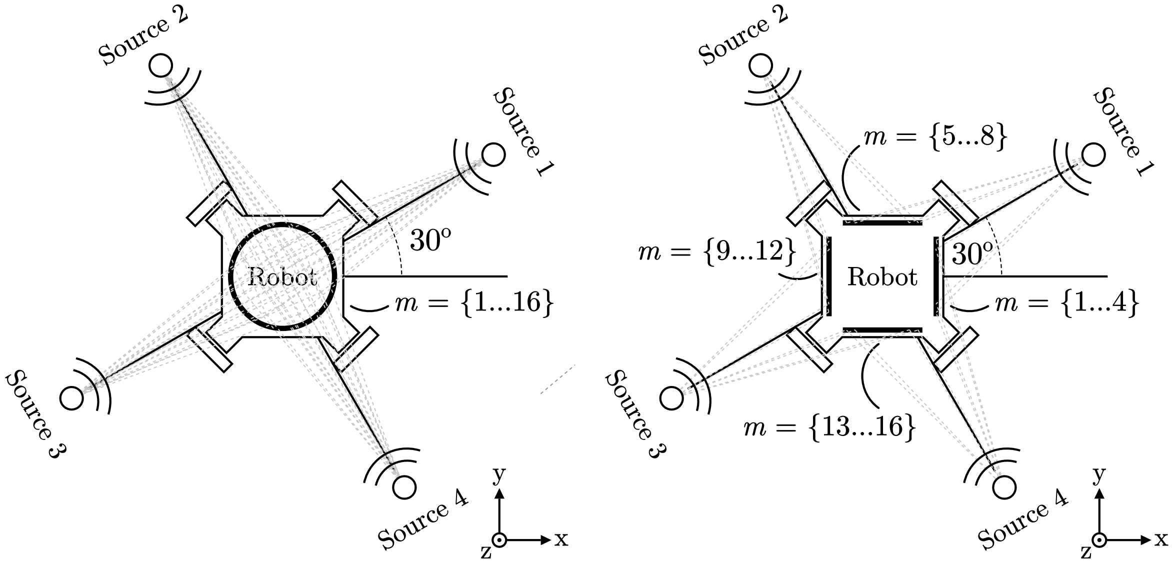
In the same way, the coherent noise signal results from the sum of each competing speech source different from the target source ($i \neq i'$) captured by the microphone array. The robot actuators' noise when it moves ($B^l_m[k]$) is also recorded individually. The overall noise consists of all these noisy observations mixed together :

$$(X_{noise})^l_m[k] = \sum_{i=1,i\neq i'}^{I} A^l_{mi}[k]S^l_i[k] + B^l_m[k] \tag{5.32}$$

The SNR observed by the microphone array before separation corresponds to the target source power in relation to the noise power :

$$SNR_{mic}(i', m) = \frac{\sum_l \sum_k |(X_{target})^l_m[k]|^2}{\sum_l \sum_k |(X_{noise})^l_m[k]|^2} \tag{5.33}$$

To illustrate the motivation behind the use of a microphone directivity model, Fig. 5.6 shows the specific setup for T3 with four sources positioned at 30°, 120°, 210° and 300°. For the OMA configuration, there are direct paths between all sources and microphones. With the CMA configuration, there are direct paths for sound between Source 1 and microphones $1; \ldots, 8$, Source 2 and microphones $5, \ldots, 12$; Source 3 and microphones $9, \ldots, 16$; and finally Source 4 and microphones $13, \ldots, 16, 1, \ldots, 4$.



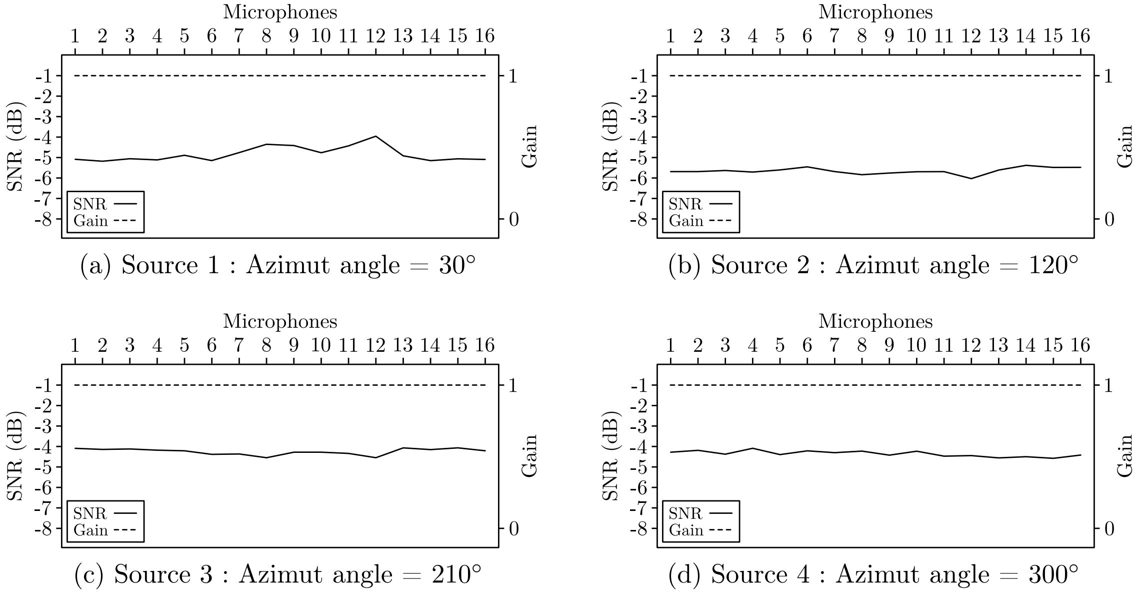
(a) OMA (b) CMA

Figure 5.6 Specific case of T3 with an initial offset of 30°

Performance evaluation is done by examining the SNR of each microphone to validate the relevance of the directionality model for microphones. The spectrograms of the sources before and after separation are also provided to illustrate on a time-frequency scale the enhancement provided by the separation methods. Finally, the SNRs of separated signals are examined for all methods in the four test conditions, to observe their influence in relation to the two array configurations.

5.5.1 Microphones SNRs

The SNR of each microphone is computed to demonstrate the impact of the OMA and CMA configurations on the separation step. SNRs of all microphones for the OMA and CMA configurations are shown in Fig. 5.7 and Fig. 5.8 to demonstrate the use of the directivity model. The gain $G_m(\mathbf{u}_i^l)$ applied to each microphone according to the source position and microphone directivity is also plotted. For the OMA configuration, the SNRs are similar for all microphones. This is expected as all microphones have a direct path with the speech sources. With this configuration, all gains get a value of 1, which makes sense as more observations provide a better separation when all SNRs are similar. For the CMA configuration, the gains tend to reach a value of 1 for the microphones which SNRs are maximized (which is because there is a direct path between the microphone and the source). This confirms the principle that the scalar gain in the microphone directivity model gives more weight to the most reliable microphones. Moreover, with the CMA configuration, some gains reach zero for microphones with low SNR values, which implies that it is preferable to perform separation with less observations when some of them are too noisy.



(a) Source 1 : Azimut angle = 30°

(b) Source 2 : Azimut angle = 120°

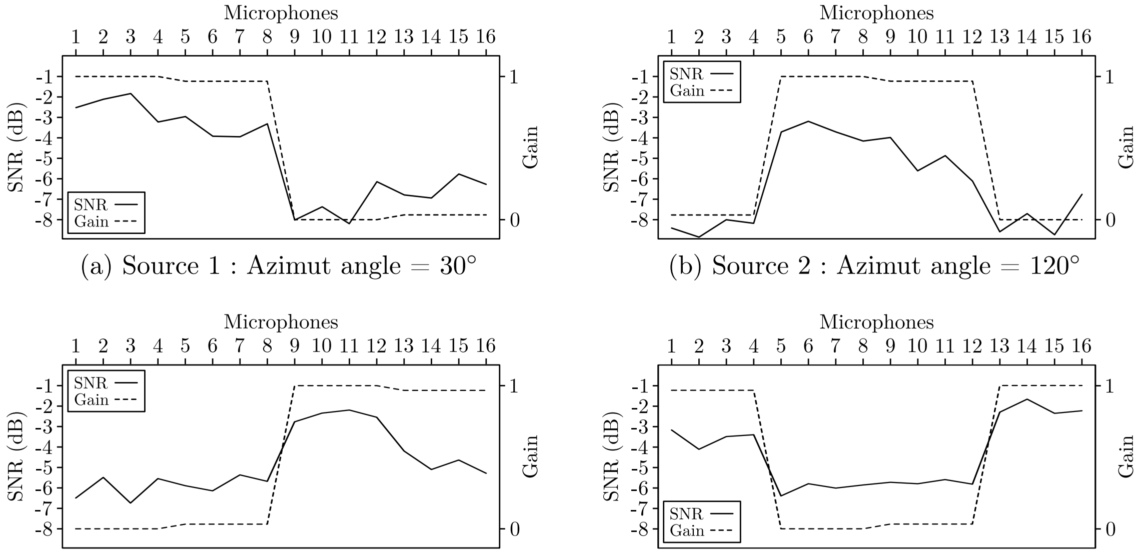
(c) Source 3 : Azimut angle = 210°

(d) Source 4 : Azimut angle = 300°

Figure 5.7 Microphones SNRs vs Gains with the OMA configuration

5.5.2 Spectrograms

Spectrograms are generated with microphones and separated signals to visualize the effect of the separation methods. Spectrograms of both speech sources located at 0° and 180° in T1 illustrate the impact of the cubic structure on separation with the CMA configuration.



(a) Source 1 : Azimut angle = 30°

(b) Source 2 : Azimut angle = 120°

(c) Source 3 : Azimut angle = 210°

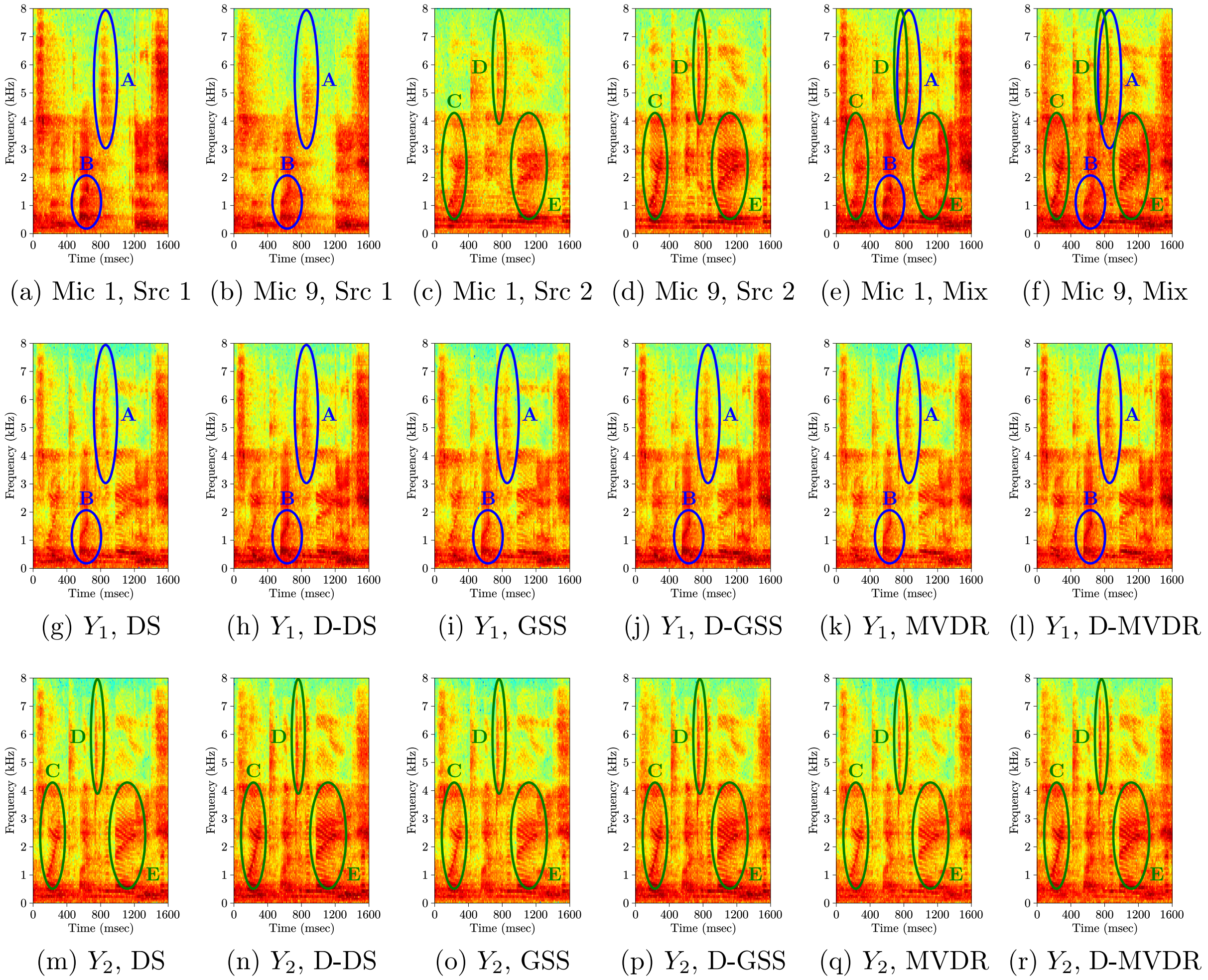
(d) Source 4 : Azimut angle = 300°

Figure 5.8 Microphones SNRs vs Gains with the CMA configuration

Figure 5.9a and Fig. 5.9b illustrate the spectrograms of Source 1, captured by microphones 1 and 9. Microphone 1 faces Source 1, while microphone 9 is hidden by the cubic structure. Zones A and B clearly show that the spectrogram is more energetic for microphone 1, which confirms that the SNR is higher for this microphone that faces the source. Similarly, Fig. 5.9c and Fig. 5.9d show the spectrograms of Source 2, also captured by microphones 1 and 9. In this case, the source faces microphone 9, and not microphone 1. Time-frequency zones C, D, and E have more energy for microphone 9 when compared to microphone 1. Moreover, the energy onset in zone D exhibits a sharper transition for microphone 9, which indicates that this microphone captures more efficiently the direct path portion of the signal. These sources are mixed together to generate the observations shown in Fig. 5.9e and Fig. 5.9f. These spectrograms demonstrate that the relevant zones A and B of Source 1 are more visible for microphone 1, while zones C, D and E stand out more for microphone 9.

Separation of Source 1 with methods D-DS, D-GSS and D-MVDR generate cleaner time-frequency features (their intensity increases) when compared to DS, GSS and MVDR, as shown in Fig. 5.9g-5.9l. Similarly, Fig. 5.9m-5.9r demonstrate that zones C-E are also preserved more accurately with D-DS, D-GSS and D-MVDR for Source 2.

Spectrograms of one speech source located at 0° and the motors' noise in T4 illustrate how adding directivity handles non-stationary diffuse noise. Figure 5.10a and Fig. 5.10b illustrate the spectrograms of Source 1, captured by microphones 1 and 9. Microphone 1 faces Source 1, while microphone 9 is hidden by the cubic structure. The time-frequency



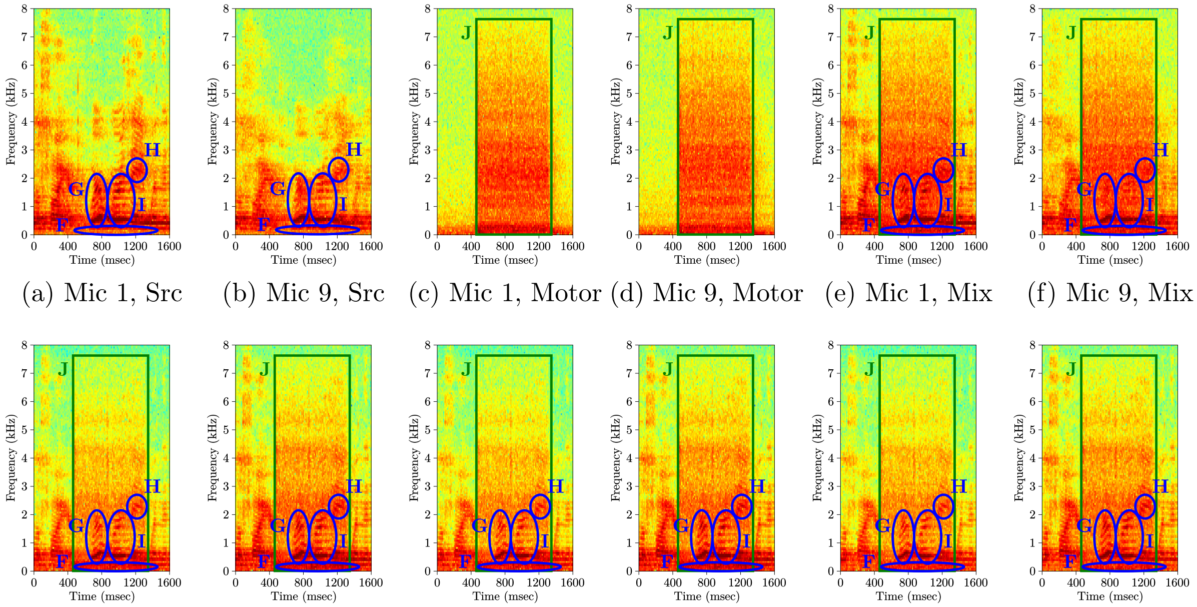
(a) Mic 1, Src 1 (b) Mic 9, Src 1 (c) Mic 1, Src 2 (d) Mic 9, Src 2 (e) Mic 1, Mix (f) Mic 9, Mix

(g) Y_1, DS (h) Y_1, D-DS (i) Y_1, GSS (j) Y_1, D-GSS (k) Y_1, MVDR (l) Y_1, D-MVDR

(m) Y_2, DS (n) Y_2, D-DS (o) Y_2, GSS (p) Y_2, D-GSS (q) Y_2, MVDR (r) Y_2, D-MVDR

Figure 5.9 Spectrograms obtained using different separation methods applied to two speech sound sources mixed together, with the CMA configuration

regions F, G, H, I are more energetic for microphone 1 than for microphone 9. Similarly, Fig. 5.10c and Fig. 5.10d show the spectrograms of the robot's motors' noise, also captured by microphones 1 and 9. In this case, the energy is comparable because noise is diffused evenly in all directions. Zone J represents the noise generated by the motors, while the rest of the spectrogram consists of fan noise. Speech and motor noise are mixed together to generate the observations shown in Fig. 5.10e and Fig. 5.10f. Figures 5.10g-5.10l show the enhancement of the speech signal in zones F-I, while the motor noise in zone J is attenuated. Note that MVDR/D-MVDR reduce significantly the amplitude of noise in low frequencies in zone F, as opposed to DS/D-DS and GSS/D-GSS.



(a) Mic 1, Src (b) Mic 9, Src (c) Mic 1, Motor (d) Mic 9, Motor (e) Mic 1, Mix (f) Mic 9, Mix

(g) Y_1, DS (h) Y_1, D-DS (i) Y_1, GSS (j) Y_1, D-GSS (k) Y_1, MVDR (l) Y_1, D-MVDR

Figure 5.10 Spectrograms obtained using different separation methods applied to one speech sound source mixed with motor noise, with the CMA configuration

5.5.3 Separated Signals SNRs

The SNRs obtained for each separation method are introduced in this section to measure the improvements caused by the introduction of the directivity model. The overall SNR for a given source i' positioned at $\mathbf{u}_i^l$ corresponds to the sum of the target energy over the sum of the interference energy over all microphones :

$$SNR_{mics}(i') = \frac{\sum_m \sum_l \sum_k |(X_{target})_m^l[k]|^2}{\sum_m \sum_l \sum_k |(X_{noise})_m^l[k]|^2} \tag{5.34}$$

The demixing elements $W_{im}^l[k]$, obtained for all microphones $m = 1, \ldots, M$ and the target source i', are applied to the target and noise spectra in (5.35) and (5.36), to determine the impact of target source enhancement and noise attenuation.

$$(Y_{target})_{i'}^l[k] = \sum_{m=1}^{M} W_{i'm}^l[k](X_{target})_m^l[k] \tag{5.35}$$

$$(Y_{noise})_{i'}^l[k] = \sum_{m=1}^{M} W_{i'm}^l[k](X_{noise})_m^l[k] \tag{5.36}$$

The separated signal SNR corresponds to the following equation :

$$SNR_{sep}(i') = \frac{\sum_l \sum_k |(Y_{target})^l_{i'}[k]|^2}{\sum_l \sum_k |(Y_{noise})^l_{i'}[k]|^2} \tag{5.37}$$

The average signal enhancement is given by :

$$\Delta SNR = \frac{1}{I} \sum_{i=1}^{I} SNR_{sep}(i) - SNR_{mics}(i) \tag{5.38}$$

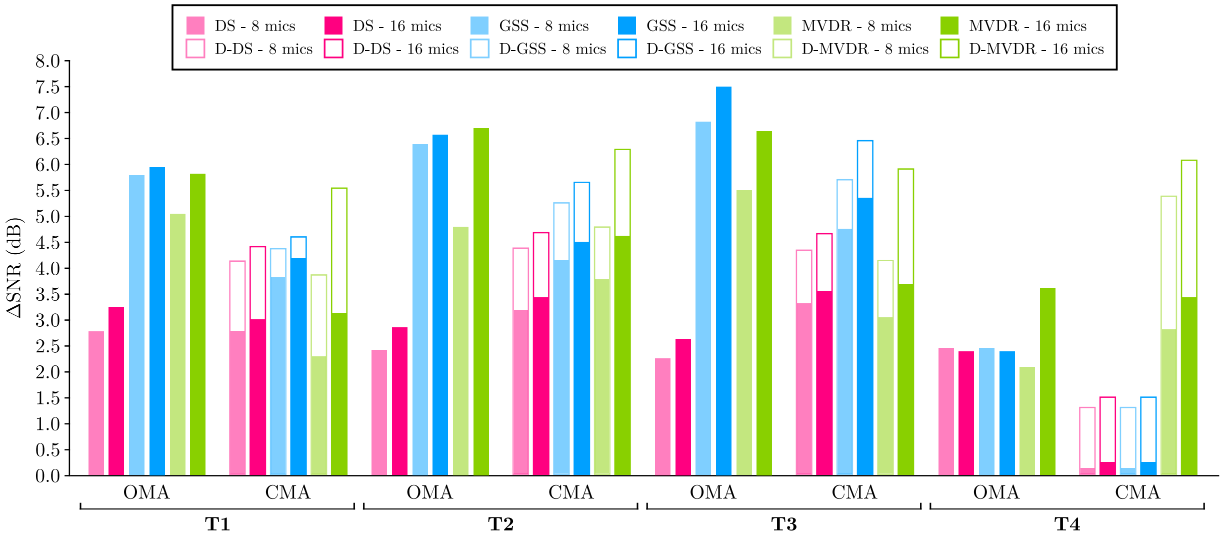


Figure 5.11 ΔSNRs with CMA configuration

The results are compiled and shown in Fig. 5.11 for both CMA and OMA robot configurations, and $M = 8$ and $M = 16$ microphones. ΔSNRs demonstrate that for T1 to T3, both for CMA and OMA configurations, the GSS/D-GSS performs better than DS/D-DS and MVDR/D-MVDR. This is expected as the interfering speech sound sources are directional, and the GSS/D-GSS methods minimize the gain in the directions of these sources. In T4, the MVDR/D-MVDR methods perform better, as the motors' noise is diffuse. When only one source is separated, the GSS/D-GSS gets the same demixing matrix as the DS/D-DS method, which explains why their SNRs are identical in T4. For all scenarios with the CMA configuration, D-DS, D-GSS and D-MVDR improve significantly the SNR of the separated sources when compared to DS (improvements by 1.4 dB, 1.3 dB, 1.1 dB and 1.3 dB for S1 to S4, respectively), GSS (improvements by 0.4 dB, 1.2 dB, 1.1 dB and 1.3 dB) and MVDR (improvements by 2.4 dB, 1.7 dB, 2.2 dB and 2.7 dB) methods. The experiments performed with the OMA configuration generate the same SNRs for the DS/D-DS, GSS/D-GSS and MVDR/D-MVDR methods. This is expected as all microphones have a direct path with the sound sources, and therefore all gains G_m get a value of 1. The GSS and MVDR methods perform better than with the CMA configuration, which is due to the fact that the free sound propagation hypothesis, on which GSS and MVDR rely, models with more accuracy sound propagation for OMA. Note that the D-DS, D-GSS and D-MVDR methods with CMA offer performance similar to DS, GSS and MVDR with OMA. D-MVDR with CMA outperforms MVDR with OMA in T4, as the closed microphone array structure mechanically filters some of the noise coming from motors on the otherside of the cube. Finally, separation performs better with more microphones (16 instead of 8), except for T4 with the OMA configuration, as the 8 selected microphones may be less noisy on average than all the 16 microphones due to the spatial distribution of the motor noise.

5.6 Conclusion

For microphone arrays placed on a robot's body, with microphones being blocked by the robot's shape, using a microphone directivity model to adapt the gain applied to each microphone based on the microphone's directivity, provides clear benefit for separating sound sources. Results suggest that adding such microphone directivity model can improve the separation SNRs by up to 2.7 dB for CMA, and generates the same SNRs for OMA configuration. Such model is appealing for a robotic solution as it exploits the directivity of the microphones to provide optimal results with various shapes of microphone arrays, and it avoids having to conduct expensive calibration of microphone arrays which would involve HRTFs. Moreover, with the microphone directivity model and the CMA configuration, some microphones have a zero gain and they can be discarded during the separation process, which reduces the amount of computations with D-DS. In future work, the gain could be adjusted according to the estimated SNR at each microphone, for better performance when noise is distributed unevenly across microphones. Moreover, when the estimated gain reaches zero for a microphone, the signals from this microphone could be ignored in the separation process to further reduce the amount of computations.

Acknowledgment

The authors would like to thank Dominic Létourneau, Cédric Godin and Vincent-Philippe Rhéaume for their help in the experimental setup.

CHAPITRE 6

ARCHITECTURE DE LA LIBRAIRIE ODAS

La librairie ODAS est réalisée en language C et distribuée sous forme de logiciel libre [1]. Ce chapitre présente un aperçu de l'architecture de cette librairie.

6.1 Modèles d'objets

Des objets effectuent les calculs sur plusieurs fils d'exécution et font cheminer les données dans la chaîne de traitement. Ces objets sont construits à partir de cinq modèles : *Amessage*, *Aconnector*, *Amodule*, *Asource* et *Asink*.

La composante *Amessage* (abbréviée par l'expression amsg) permet l'échange de données entre deux fils d'exécution asynchrones. Deux tampons FIFO (premier entré, premier sorti) assurent une mise en mémoire des messages vides et remplis pour palier aux délais sporadiques entre le fil d'exécution de remplissage et celui de vidage des messages, tel qu'illustré à la figure 6.1.

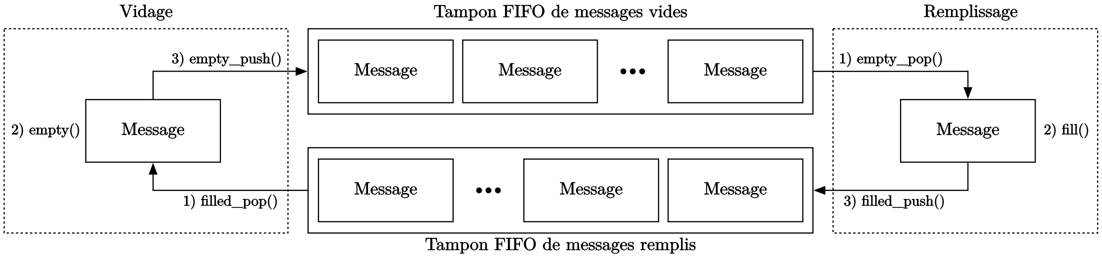


Figure 6.1 Structure d'une composante amsg

L'élément *Aconnector* (ou acon) permet de copier un message à l'entrée vers plusieurs destinations différentes. Ce mécanisme est nécessaire lorsqu'un message doit être dupliqué et envoyé vers plusieurs objets. La figure 6.2 illustre le mécanisme qui copie le contenu du message initial vers plusieurs messages destinataires, en utilisant la structure amsg pour permettre l'utilisation de fils d'exécution asynchrones.

La figure 6.3 illustre le fonctionnement du *Amodule* (amod). Le contenu d'un ou plusieurs messages est copié vers le module, qui effectue un traitement et copie à son tour un nouveau contenu vers les messages de sortie.

[1] http ://github.com/introlab/odas

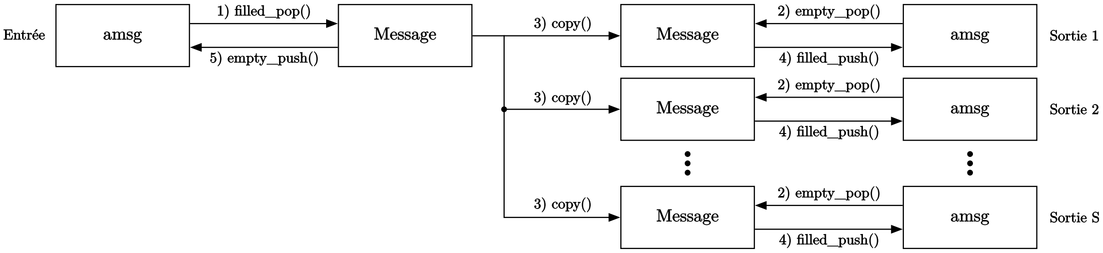


Figure 6.2 Structure d'une composante acon

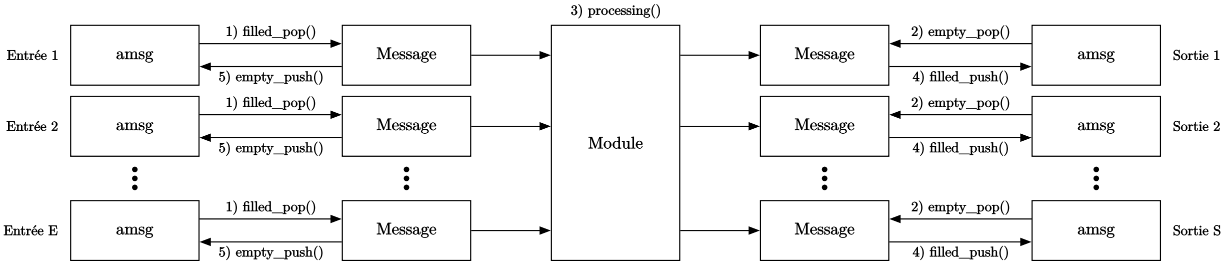


Figure 6.3 Structure d'une composante amodule

Une source permet de générer un message (à partir d'un fichier de données ou d'une carte de son par exemple) qui sera traité par les modules de la librairie ODAS, tel qu'illustré à la figure 6.4.

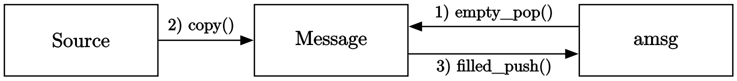


Figure 6.4 Structure d'une composante asrc

La figure 6.5 illustre la composante *Asink* (asnk) qui utilise le contenu d'un message dans le but de l'exporter vers une interface (fichier, socket, etc.)

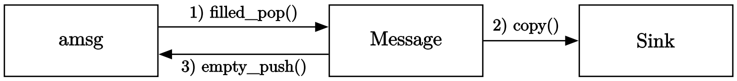


Figure 6.5 Structure d'une composante asnk

6.2 Objets

Des objets sont créés à partir des modèles *AMessage*, *AConnector*, *ASource* et *ASink* pour les types de données suivants :

- *Hops* : Des échantillons qui forment une trame dans le domaine temporel pour plusieurs canaux.
- *Spectra* : L'enveloppe spectrale (sous forme de nombres complexes) de trames dans le domaine fréquentiel pour plusieurs canaux.

- *Pots* : Directions des sources potentielles et l'énergie associée à chacune d'elle.
- *Tracks* : Directions des sources suivies et un identifiant unique pour les distinguer.

Les objets créés à partir du modèle *AModule* accomplissent différentes tâches :

- *STFT* : Calcule la transformée de Fourier de courte durée à partir d'un message *Hops* et génère un message *Spectra*.
- *iSTFT* : Calcule la transformée de Fourier inverse de courte durée à partir d'un message *Spectra* et génère un message *Hops*.
- *Mapping* : Copie certains canaux d'un message *Hops* vers un autre message *Hops*.
- *Resample* : Rééchantillonne un signal à partir d'un message *Hops* et génére un nouveau message *Hops*.
- *SSL* : Localise et génère des sources potentielles *Pots* à partir des trames *Spectra*.
- *SST* : Effectue la tâche de suivi de sources sonores et produit des sources suivies *Tracks* à partir des sources potentielles *Pots*.
- *SSS* : Sépare les sources sonores et applique un post-filtrage (identique à celui proposé dans la librairie ManyEars [Grondin *et coll.*, 2013]) à partir des sources suivies *Tracks* et les trames *Spectra* des microphones et génère des trames *Spectra* séparées et post-filtrées.

La figure 6.6 illustre les interconnexions entre ces objets qui composent la librairie ODAS.

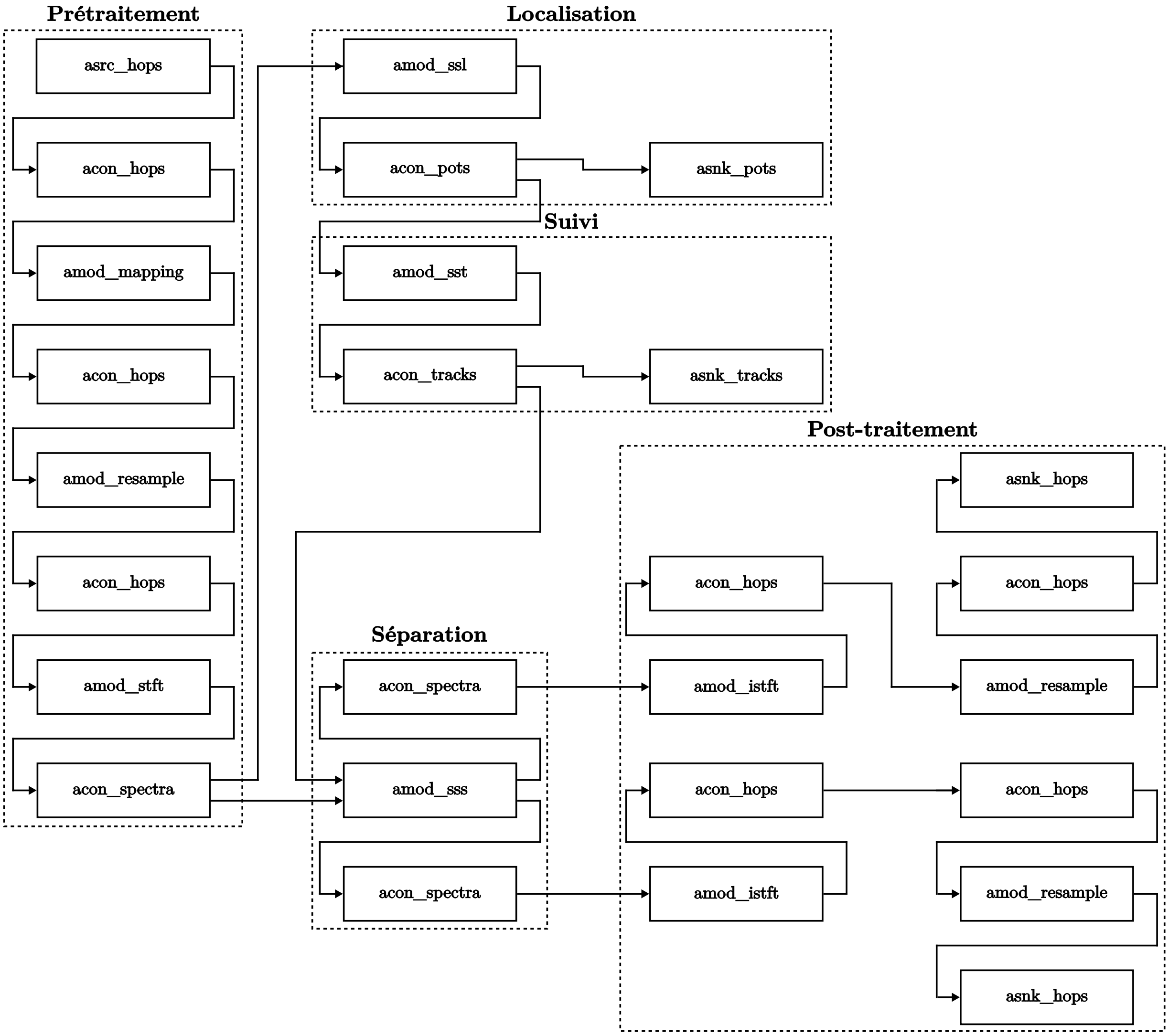


Figure 6.6 Vue d'ensemble du flux de données dans ODAS

CHAPITRE 7
CONCLUSION

Cette thèse présente une nouvelle librairie d'audition artificielle, dénommée ODAS, qui vise à améliorer les performances de localisation, suivi et séparation de sources sonores sur un robot mobile. Voici comment les objectifs présentés au chapitre 1 sont rencontrés avec ODAS :

1. *Adaptabilité à la géometrie du robot* : Il est démontré que les modules de localisation, suivi et séparation présentés peuvent s'adapter à des configurations différentes au niveau de la géométrie de la matrice de microphones.
2. *Utilisation d'une matrice de microphones fermée* : Le module de localisation proposé s'adapte à une configuration avec une matrice de microphones fermée, et réduit considérablement les fausses détections normalement présentes avec une méthode de localisation traditionnelle. De plus, l'ajout d'un modèle qui tient compte de la directivité des microphones permet au module de séparation de rehausser jusqu'à 2.7 dB de plus le rapport signal sur bruit avec une matrice fermée pour un formateur de faisceau à variance minimale.
3. *Calibration rapide du système d'audition* : Le module de localisation ajuste automatiquement certains paramètres pour conserver une précision dans la direction d'arrivée des sources sonores, indépendamment de la géométrie de la matrice. L'ajout d'un modèle analytique simple pour la directivité des microphones permet de garantir de bonnes performances durant la localisation et la séparation, sans effectuer de caractérisations acoustiques de la matrice de microphones.
4. *Minimisation de la charge de calculs* : Les modules de localisation et de suivi des sources sonores requièrent jusqu'à 4 et 30 fois moins de calculs que la librairie ManyEars, respectivement, tout en garantissant des performances équivalentes ou supérieures aux méthodes traditionnelles.
5. *Augmentation du nombre de microphones* : Il est démontré qu'il est possible de faire fonctionner en temps réel les modules de localisation, suivi et séparation de sources sonores avec une matrice de 16 microphones, alors que leur nombre était habituellement limité à 8 microphones avec les architectures ManyEars [Grondin *et coll.*, 2013] et HARK [Nakadai *et coll.*, 2010b].

6. *Portabilité du système* : La librarie ODAS est réalisée en language C et intègre les méthodes de localisation, suivi et séparation présentées dans cette thèse. Cette librarie est disponible gratuitement sous forme de code ouvert [1] afin de faciliter la diffusion au sein de la communauté scientifique. L'utilisation du language C et le peu de dépendances vers des librairies externes facilitent la portabilité d'ODAS sur la plupart des systèmes d'exploitation existants (Windows, MacOS et Linux). De plus, une librairie dénommée ODAS-ROS permet d'encapsuler chaque module pour une intégration directe dans l'environnement ROS [2], dans le but d'encourager les roboticiens à intégrer cette technologie.

D'autres fonctionnalités ont aussi été considérées pour ODAS :

- Des travaux complémentaires ont été réalisés dans le but d'améliorer la robustesse de la localisation de sources sonores face au bruit stationnaire. L'utilisation de masques temps-fréquence binaires [Grondin et Michaud, 2015, 2016a] peuvent améliorer la robustesse au bruit, et peuvent être ajoutés à la librairie ODAS.
- Une fois qu'une ou plusieurs sources sonores sont séparées, il est utile de pouvoir classifier cette source comme étant de la parole humaine ou un bruit du quotidien. Une technique utilisant la tonalité vocale permet d'effectuer une classification plus robuste à la réverbération et au bruit de fond ambiant, par rapport à l'utilisation des coefficients cepstraux avec les fréquences Mel [Grondin et Michaud, 2016b]. Cette classification des sources sonores peut également être ajoutée à la librairie ODAS.

Enfin, la librairie ODAS a été et peut également être utilisée dans plusieurs applications concrètes :

- Les intrusions par drones au-dessus d'une zone sécurisée (notamment les cours extérieures des centres correctionnels) constitue un problème qui prend de l'ampleur depuis les dernières années, et les autorités recherchent activement des solutions pour détecter ces intrusions. Des expériences ont démontré qu'il est possible de localiser le son émis par les drones à voilure tournante [Lauzon *et coll.*, 2017]. Pour ce faire, plusieurs balises équipées de huit microphones sont installées à l'extérieur, et le système ODAS localise les sources sonores perçues en direction du ciel par rapport à chaque balise. Les calculs sont effectués en temps réel sur un Raspberry Pi 3, et les directions captées par l'ensemble des balises sont envoyées par le réseau local à

[1] http ://github.com/introlab/odas
[2] http ://github.com/introlab/odas_ros

un serveur qui effectue une triangulation dans le but d'obtenir la position 3D du drone. Cette approche est prometteuse, car, contrairement aux systèmes utilisant des caméras et/ou des capteurs d'ondes radio, le système de détection sonore peut fonctionner la nuit et lorsque le drone fonctionne en mode autonome sans émettre en direction d'un opérateur à distance.

- Récemment, plusieurs assistants vocaux, tels que Amazon Echo, Google Home et Apple HomePod, ont fait leur apparition sur le marché. Ces plateformes permettent une interaction main libre dans un environnement bruité à l'aide de plusieurs microphones. Ces interfaces limitent l'interaction à une seule source sonore, et cette source est détectée suite à la prononciation d'un mot-clef. Le son enregistré est ensuite téléversé vers le nuage, où s'effectue la reconnaissance de la parole. Cette nécessité d'envoyer les données vers le nuage représente cependant une brèche de sécurité dans la vie privée [Chung *et coll.*, 2017]. La mise en place d'un système de localisation, suivi et séparation de sources sonores, combiné avec un engin de reconnaissance vocale hors-ligne, permettrait de résoudre cette problématique. La librairie ODAS, de par son efficacité à fonctionner sur du matériel embarqué à faible coût et sa capacité à s'adapter à plusieurs géométries de matrices de microphones, est un élément de la solution.
- La conception de véhicules autonomes en milieu urbain représente un défi technique important [Buehler *et coll.*, 2009]. Des caméras [Guo *et coll.*, 2016] et lasers [Rasmussen, 2002] sont déployés sur le véhicule pour permettre à celui-ci de naviguer efficacement dans un environnement dynamique. Il est démontré que les événements sonores ayant lieu près du véhicule (ex : klaxon, sirène d'urgence, etc.) peuvent avoir un impact sur la prise de décision d'un conducteur [Parkin *et coll.*, 2016]. Il serait donc pertinent d'utiliser ODAS pour localiser, suivre et séparer ces sons de l'environnement, pour ensuite les classifier et modifier au besoin le trajet du véhicule.
- Les aides auditives viennent rehausser la qualité de vie des personnes malentendantes. Récemment, des matrice de microphones ont été introduites pour permettre aux utilisateurs d'écouter le son provenant d'une direction précise [Brandstein et Ward, 2013]. La librairie ODAS pourrait être utilisée par ces aides auditives pour effectuer la localisation, le suivi et la séparation sur un système embarqué à faible coût. De plus, dans le cas d'une personne sourde qui utilise le langage des signes et lit sur les lèvres, il serait intéressant de signaler un événement sonore qui se produit à l'extérieur de son champ de vision (à l'aide d'une ceinture équipée de vibreurs par exemple). Il pourrait s'agir d'un bruit relié à un danger (klaxon de voiture, cri, etc.) ou encore un individu qui prononce le nom de l'utilisateur à voix haute. Le système

ODAS pourrait être employé pour effectuer cette tâche de localisation à l'aide d'une matrice de microphone, également installée au niveau de la ceinture de l'utilisateur.

En conclusion, la librairie ODAS intègre de nouvelles fonctionnalités rendues possibles grâce aux améliorations apportées aux méthodes de localisation, suivi et séparation présentées dans cette thèse. Par sa robustesse, son adaptabilité, sa calibration rapide, sa faible charge de calculs et sa portabilité, nous souhaitons qu'ODAS devienne une librairie profitable au domaine de l'audition en robotique et ouvre la voie pour de multiples applications.

LISTE DES RÉFÉRENCES

Abdi, H. et Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews : Computational Statistics*, volume 2, numéro 4, p. 433–459.

Anastasakos, A., Schwartz, R. et Shu, H. (1995). Duration modeling in large vocabulary speech recognition. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1. p. 628–631.

Anguera, X., Wooters, C. et Hernando, J. (2007). Acoustic beamforming for speaker diarization of meetings. *IEEE Transactions on Audio, Speech, and Language Processing*, volume 15, numéro 7, p. 2011–2022.

Antonacci, F., Lonoce, D., Motta, M., Sarti, A. et Tubaro, S. (2005). Efficient source localization and tracking in reverberant environments using microphone arrays. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4. p. 1061–1064.

Antonacci, F., Saiu, D., Russo, P., Sarti, A., Tagliasacchi, M. et Tubaro, S. (2006). Experimental evaluation of a localization algorithm for multiple acoustic sources in reverberating environments. Dans *Proceedings of the European Signal Processing Conference*. p. 1–4.

Arons, B. (1992). A review of the cocktail party effect. *American Voice I/O Society*, volume 12, numéro 7, p. 35–50.

Barfuss, H., Buerger, M., Podschus, J. et Kellermann, W. (2017). HRTF-based two-dimensional robust least-squares frequency-invariant beamformer design for robot audition. Dans *Proceedings of the IEEE Hands-Free Speech Communication and Microphone Array*. p. 56–60.

Brandstein, M. et Ward, D. (2013). *Microphone Arrays : Signal Processing Techniques and Applications*. Springer Science & Business Media.

Briere, S., Valin, J.-M., Michaud, F. et Létourneau, D. (2008). Embedded auditory system for small mobile robots. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*. p. 3463–3468.

Brodeur, D., Grondin, F., Attabi, Y., Dumouchel, P. et Michaud, F. (2016). Integration framework for speech processing with live visualization interfaces. Dans *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*.

Bromiley, P. (2003). *Products and Convolutions of Gaussian Probability Density Functions* (Rapport technique). University of Manchester.

Buchner, H., Aichner, R. et Kellermann, W. (2003a). Blind source separation for convolutive mixtures exploiting nongaussianity, nonwhiteness, and nonstationarity. Dans *Proceedings of the International Workshop on Acoustic Echo and Noise Cancellation*.

Buchner, H., Aichner, R. et Kellermann, W. (2003b). A generalization of a class of blind source separation algorithms for convolutive mixtures. Dans *Proceedings of the International Symposium on Independent Component Analysis and Blind Signal Separation*. p. 945–950.

Buchner, H., Aichner, R. et Kellermann, W. (2004a). Blind source separation for convolutive mixtures : A unified treatment. *Audio Signal Processing for Next-Generation Multimedia Communication Systems*, p. 255–293.

Buchner, H., Aichner, R. et Kellermann, W. (2004b). TRINICON : A versatile framework for multichannel blind signal processing. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3. p. 889–892.

Buchner, H., Aichner, R. et Kellermann, W. (2005a). A generalization of blind source separation algorithms for convolutive mixtures based on second-order statistics. *IEEE Transactions on Speech and Audio Processing*, volume 13, numéro 1, p. 120–134.

Buchner, H., Aichner, R. et Kellermann, W. (2007). TRINICON-based blind system identification with application to multiple-source localization and separation. *Blind Speech Separation*, p. 101–147.

Buchner, H., Aichner, R., Stenglein, J., Teutsch, H. et Kellennann, W. (2005b). Simultaneous localization of multiple sound sources using blind adaptive MIMO filtering. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3. p. 97–100.

Buehler, M., Iagnemma, K. et Singh, S. (2009). *The DARPA Urban Challenge : Autonomous Vehicles in City Traffic*, volume 56. Springer.

Chu, S., Narayanan, S. et Kuo, C.-C. J. (2008). Environmental sound recognition using MP-based features. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. p. 1–4.

Chu, S., Narayanan, S. et Kuo, C.-C. J. (2009). Environmental sound recognition with time–frequency audio features. *IEEE Transactions on Audio, Speech, and Language Processing*, volume 17, numéro 6, p. 1142–1158.

Chung, H., Iorga, M., Voas, J. et Lee, S. (2017). Alexa, can I trust you ? *Computer*, volume 50, numéro 9, p. 100–104.

Cohen, I. (2003). Noise spectrum estimation in adverse environments : Improved minima controlled recursive averaging. *IEEE Transactions on Speech and Audio Processing*, volume 11, numéro 5, p. 466–475.

Danes, P. et Bonnal, J. (2010). Information-theoretic detection of broadband sources in a coherent beamspace MUSIC scheme. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 1976–1981.

Deleforge, A. et Kellermann, W. (2015). 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*. p. 355–359.

Do, H. et Silverman, H. F. (2011). A robust sound-source separation algorithm for an adverse environment that combines MVDR-PHAT with the CASA framework. Dans *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. p. 273–276.

Do, H., Silverman, H. F. et Yu, Y. (2007). A real-time SRP-PHAT source location implementation using stochastic region contraction (SRC) on a large-aperture microphone array. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1. p. 121–124.

Drude, L., Jacob, F. et Haeb-Umbach, R. (2015). DOA-estimation based on a complex Watson kernel method. Dans *Proceedings of the European Signal Processing Conference*. p. 255–259.

Frechette, M., Létourneau, D., Valin, J.-M. et Michaud, F. (2012). Integration of sound source localization and separation to improve dialogue management on a robot. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 2358–2363.

Frigo, M. et Johnson, S. G. (1998). FFTW : An adaptive software architecture for the FFT. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3. p. 1381–1384.

Fritsch, J., Kleinehagenbrock, M., Lang, S., Fink, G. A. et Sagerer, G. (2004). Audiovisual person tracking with a mobile robot. Dans *Proceedings of the International Conference on Intelligent Autonomous Systems*. p. 898–906.

Furuya, K. (2001). Noise reduction and dereverberation using correlation matrix based on the multiple-input/output inverse-filtering theorem (MINT). Dans *Proceedings of the International Workshop on Hand-Free Speech Communication*. p. 59–62.

Goldhor, R. S. (1993). Recognition of environmental sounds. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1. p. 149–152.

Grondin, F., Létourneau, D., Ferland, F., Rousseau, V. et Michaud, F. (2013). The ManyEars open framework. *Autonomous Robots*, volume 34, numéro 3, p. 217–232.

Grondin, F. et Michaud, F. (2012). WISS, a speaker identification system for mobile robots. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*. p. 1817–1822.

Grondin, F. et Michaud, F. (2015). Time difference of arrival estimation based on binary frequency mask for sound source localization on mobile robots. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 6149–6154.

Grondin, F. et Michaud, F. (2016a). Noise mask for TDOA sound source localization of speech on mobile robots in noisy environments. Dans *Proceedings of the IEEE International Conference on Robotics and Automation.* p. 4530–4535.

Grondin, F. et Michaud, F. (2016b). Robust speech/non-speech discrimination based on pitch estimation for mobile robots. Dans *Proceedings of the IEEE International Conference on Robotics and Automation.* p. 1650–1655.

Grondin, F. et Michaud, F. (2017). A lightweight and optimized sound source localization method for open and closed microphone array configurations. *Submitted to IEEE Trans. Robot.*

Guennebaud, G. et Jacob, B. (2014). Eigen C++ Template Library for Linear Algebra. *http://eigen.tuxfamily.org.*

Guo, J., Hu, P. et Wang, R. (2016). Nonlinear coordinated steering and braking control of vision-based autonomous vehicles in emergency obstacle avoidance. *IEEE Transactions on Intelligent Transportation Systems*, volume 17, numéro 11, p. 3230–3240.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N. et Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition : The shared views of four research groups. *IEEE Signal Processing Magazine*, volume 29, numéro 6, p. 82–97.

Holmes, M., Gray, A. et Isbell, C. (2007). Fast SVD for large-scale matrices. Dans *Proceedings of the Workshop on Efficient Machine Learning at Neural Information Processing Systems*, volume 58. p. 249–252.

Hyvärinen, A. et Oja, E. (2000). Independent component analysis : Algorithms and applications. *Neural Network*, volume 13, numéro 4, p. 411–430.

Ince, G., Nakadai, K. et Nakamura, K. (2012). Online learning for template-based multichannel ego noise estimation. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems.* p. 3282–3287.

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

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

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

Ince, G., Nakamura, K., Asano, F., Nakajima, H. et Nakadai, K. (2011b). Assessment of general applicability of ego noise estimation. Dans *Proceedings of the IEEE International Conference on Robotics and Automation.* p. 3517–3522.

Ishi, C. T., Chatot, O., Ishiguro, H. et Hagita, N. (2009). Evaluation of a MUSIC-based real-time sound localization of multiple sound sources in real noisy environments. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 2027–2032.

Ji, M., Kim, S., Kim, H. et Yoon, H.-S. (2008). Text-independent speaker identification using soft channel selection in home robot environments. *IEEE Transactions on Consumer Electronics*, volume 54, numéro 1, p. 140–144.

Julier, S. J. et Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. Dans *Proceedings of the International Symposium on Aerospace/Defence, Sensing, Simulation and Controls*, volume 3. p. 182–193.

Kanda, T., Ishiguro, H., Ono, T., Imai, M. et Nakatsu, R. (2002). Development and evaluation of an interactive humanoid robot "Robovie". Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2. p. 1848–1855.

Kellermann, W., Buchner, H. et Aichner, R. (2006). Separating convolutive mixtures with TRINICON. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5. p. 961–964.

Keyrouz, F., Diepold, K. et Keyrouz, S. (2007a). Humanoid binaural sound tracking using Kalman filtering and HRTFs. *Robot Motion and Control*, p. 329–340.

Keyrouz, F., Maier, W. et Diepold, K. (2006). A novel humanoid binaural 3D sound localization and separation algorithm. Dans *Proceedings of the IEEE RAS International Conference on Humanoid Robots*. p. 296–301.

Keyrouz, F., Maier, W. et Diepold, K. (2007b). Robotic binaural localization and separation of more than two concurrent sound sources. Dans *Proceedings of the International Symposium on Signal Processing and Its Applications*. p. 1–4.

Kim, U.-H., Mizumoto, T., Ogata, T. et Okuno, H. G. (2011). Improvement of speaker localization by considering multipath interference of sound wave for binaural robot audition. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 2910–2915.

Kumatari, K., McDonough, J. et Raj, B. (2012). Microphone array processing for distant speech recognition : From close-talking microphones to far-field sensors. *IEEE Signal Processing Magazine*, p. 127–140.

Lang, S., Kleinehagenbrock, M., Hohenner, S., Fritsch, J., Fink, G. A. et Sagerer, G. (2003). Providing the basis for human-robot-interaction : A multi-modal attention system for a mobile robot. Dans *Proceedings of the ACM International Conference on Multimodal Interaction*. p. 28–35.

Laniel, S., Létourneau, D., Labbé, M., Grondin, F. et Michaud, F. (2017). Enhancing a beam+ telepresence robot for remote home care applications. Dans *Proceedings of the IEEE International Conference on Virtual Rehabilitation*. p. 1–2.

Lauzon, J.-S., Grondin, F., Létourneau, D., Lussier-Desbiens, A. et Michaud, F. (2017). Localization of RW-UAVs using particle filtering over distributed microphone arrays. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems.*

Leonard, J. J. et Durrant-Whyte, H. F. (1991). Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics*, volume 7, numéro 3, p. 376–382.

Létourneau, D., Valin, J., Côté, C. et Michaud, F. (2005). Flow Designer : The free data-flow oriented development environment. *Software 2.0*, volume 3.

Liu, Z., Zhang, Z., He, L.-W. et Chou, P. (2007). Energy-based sound source localization and gain normalization for ad hoc microphone arrays. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2. p. 761–764.

Loesch, B. et Yang, B. (2010). Blind source separation based on time-frequency sparseness in the presence of spatial aliasing. Dans *Proceedings of the International Conference on Latent Variables Analysis and Signal Separation.* p. 1–8.

Lunati, V., Manhes, J. et Danès, P. (2012). A versatile system-on-a-programmable-chip for array processing and binaural robot audition. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems.* p. 998–1003.

Maazaoui, M., Abed-Meraim, K. et Grenier, Y. (2012). Adaptive blind source separation with HRTFs beamforming preprocessing. Dans *Proceedings of the IEEE Sensor Array and Multichannel Signal Processing Workshop.* p. 269–272.

Marković, I., Ćesić, J. et Petrović, I. (2016). On wrapping the Kalman filter and estimating with the SO (2) group. Dans *Proceedings of the International Conference on Information Fusion.* p. 2245–2250.

McCowan, I. A. et Bourlard, H. (2003). Microphone array post-filter based on noise field coherence. *IEEE Transactions on Speech and Audio Processing*, volume 11, numéro 6, p. 709–716.

Michaud, F., Côté, C., Létourneau, D., Brosseau, Y., Valin, J.-M., Beaudry, É., Raïevsky, C., Ponchon, A., Moisan, P., Lepage, P. *et coll.* (2007). Spartacus attending the 2005 AAAI conference. *Autonomous Robots*, volume 22, numéro 4, p. 369–383.

Mizumoto, T., Nakadai, K., Yoshida, T., Takeda, R., Otsuka, T., Takahashi, T. et Okuno, H. G. (2011). Design and implementation of selectable sound separation on the Texai telepresence system using HARK. Dans *Proceedings of the IEEE International Conference on Robotics and Automation.* p. 2130–2137.

Nakadai, K., Ince, G., Nakamura, K. et Nakajima, H. (2012). Robot audition for dynamic environments. Dans *Proceedings of the IEEE International Conference on Signal Processing, Communication and Computing.* p. 125–130.

Nakadai, K., Lourens, T., Okuno, H. G. et Kitano, H. (2000). Active audition for humanoid. Dans *Proceedings of the AAAI National Conference*. p. 832–839.

Nakadai, K., Matsuura, D., Okuno, H. G. et Kitano, H. (2003). Applying scattering theory to robot audition system : Robust sound source localization and extraction. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2. p. 1147–1152.

Nakadai, K., Matsuura, D., Okuno, H. G. et Tsujino, H. (2004). Improvement of recognition of simultaneous speech signals using AV integration and scattering theory for humanoid robots. *Speech Communication*, volume 44, numéro 1, p. 97–112.

Nakadai, K., Nakajima, H., Hasegawa, Y. et Tsujino, H. (2009). Sound source separation of moving speakers for robot audition. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. p. 3685–3688.

Nakadai, K., Nakajima, H., Ince, G. et Hasegawa, Y. (2010a). Sound source separation and automatic speech recognition for moving sources. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 976–981.

Nakadai, K., Okuno, H. G., Kitano, H. *et coll.* (2002). Real-time sound source localization and separation for robot audition. Dans *Proceedings of the IEEE International Conference on Spoken Language Processing* . p. 193–196.

Nakadai, K., Okuno, H. G., Nakajima, H., Hasegawa, Y. et Tsujino, H. (2008). An open source software system for robot audition HARK and its evaluation. Dans *Proceedings of the IEEE RAS International Conference on Humanoid Robots*. p. 561–566.

Nakadai, K., Takahashi, T., Okuno, H. G., Nakajima, H., Hasegawa, Y. et Tsujino, H. (2010b). Design and implementation of robot audition system 'HARK' – Open source software for listening to three simultaneous speakers. *Advanced Robotics*, volume 24, numéro 5-6, p. 739–761.

Nakajima, H., Nakadai, K., Hasegawa, Y. et Tsujino, H. (2008a). Adaptive step-size parameter control for real-world blind source separation. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*. p. 149–152.

Nakajima, H., Nakadai, K., Hasegawa, Y. et Tsujino, H. (2008b). High performance sound source separation adaptable to environmental changes for robot audition. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 2165–2171.

Nakamura, K., Nakadai, K., Asano, F. et Ince, G. (2011a). Intelligent sound source localization and its application to multimodal human tracking. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 143–148.

Nakamura, K., Nakadai, K. et Ince, G. (2012). Real-time super-resolution sound source localization for robots. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 694–699.

Nakamura, K., Nakadai, K., Nakajima, H. et Ince, G. (2011b). Correlation matrix interpolation in sound source localization for a robot. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. p. 4324–4327.

Namera, K., Takasugi, S., Takano, K., Yamamoto, T. et Miyake, Y. (2008). Timing control of utterance and body motion in human-robot interaction. Dans *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*. p. 119–123.

Naylor, P. et Gaubitch, N. D. (2010). *Speech Dereverberation*. Springer Science and Business Media.

Nesta, F. et Omologo, M. (2012). Generalized state coherence transform for multidimensional TDOA estimation of multiple sources. *IEEE Transactions on Audio, Speech, and Language Processing*, volume 20, numéro 1, p. 246–260.

Nesta, F., Svaizer, P. et Omologo, M. (2008). A BSS method for short utterances by a recursive solution to the permutation problem. Dans *Proceedings of the IEEE Sensor Array and Multichannel Signal Processing Workshop*. p. 357–360.

Nesta, F., Svaizer, P. et Omologo, M. (2011). Convolutive BSS of short mixtures by ICA recursively regularized across frequencies. *IEEE Transactions on Audio, Speech, and Language Processing*, volume 19, numéro 3, p. 624–639.

Nesta, F., Wada, T. S. et Juang, B.-H. (2009). Coherent spectral estimation for a robust solution of the permutation problem. Dans *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. p. 105–108.

Ohata, T., Nakamura, K., Mizumoto, T., Taiki, T. et Nakadai, K. (2014). Improvement in outdoor sound source detection using a quadrotor-embedded microphone array. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 1902–1907.

Okutani, K., Yoshida, T., Nakamura, K. et Nakadai, K. (2012). Outdoor auditory scene analysis using a moving microphone array embedded in a quadrocopter. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 3288–3293.

Otsuka, T., Nakadai, K., Ogata, T. et Okuno, H. G. (2011). Bayesian extension of MUSIC for sound source localization and tracking. Dans *Proceedings of INTERSPEECH*. p. 3109–3112.

Ouellet, S., Grondin, F., Leconte, F. et Michaud, F. (2014). Multimodal biometric identification system for mobile robots combining human metrology to face recognition and speaker identification. Dans *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*. p. 323–328.

Parkin, J., Clark, B., Clayton, W., Ricci, M. et Parkhurst, G. (2016). *Understanding Interactions between Autonomous Vehicles and Other Road Users : A Literature Review* (Rapport technique). University of the West of England.

Parra, L. C. et Alvino, C. V. (2002). Geometric source separation : Merging convolutive source separation with geometric beamforming. *IEEE Transactions on Speech and Audio Processing*, volume 10, numéro 6, p. 352–362.

Pavlidi, D., Puigt, M., Griffin, A. et Mouchtaris, A. (2012). Real-time multiple sound source localization using a circular microphone array based on single-source confidence measures. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing.* p. 2625–2628.

Pedersen, M. S., Kjems, U., Rasmussen, K. B. et Hansen, L. K. (2004). Semi-blind source separation using head-related transfer functions [speech signal separation]. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5. p. 713–716.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. et Ng, A. Y. (2009). ROS : an open-source robot operating system. Dans *Proceedings of the IEEE International Conference on Robotics and Automation Workshop on Open Source Software*, volume 3. p. 5.

Quinlan, M. J., Chalup, S. K. et Middleton, R. H. (2003). Techniques for improving vision and locomotion on the Sony AIBO robot. Dans *Proceedings of the Australasian Conference on Robotics and Automation.*

Rafaely, B. (2005). Analysis and design of spherical microphone arrays. *IEEE Transactions on Speech and Audio Processing*, volume 13, numéro 1, p. 135–143.

Rafaely, B., Peled, Y., Agmon, M., Khaykin, D. et Fisher, E. (2010). *Spherical Microphone Array Beamforming.* Springer, 281–305 p.

Rascon, C., Fuentes, G. et Meza, I. (2015). Lightweight multi-DOA tracking of mobile speech sources. *EURASIP Journal on Audio, Speech, and Music Processing*, volume 2015, numéro 1, p. 1–16.

Rasmussen, C. (2002). Combining laser range, color, and texture cues for autonomous road following. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4. p. 4320–4325.

Saint-Aimé, S., Le-Pevedic, B., Duhaut, D. et Shibata, T. (2007). Emotirob : companion robot project. Dans *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication.* p. 919–924.

Schmidt, R. O. (1986). Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, volume 34, numéro 3, p. 276–280.

Sung, J.-Y., Grinter, R. E., Christensen, H. I. et Guo, L. (2008). Housewives or technophiles ? : understanding domestic robot owners. Dans *ACM/IEEE International Conference on Human Robot Interaction.* p. 129–136.

Syskind, M. P., Larsen, J., Kjems, U. et Parra, L. C. (2007). A survey of convolutive blind source separation methods. *Springer Handbook on Speech Processing and Speech Communication.*

Takeda, R., Yamamoto, S., Komatani, K., Ogata, T. et Okuno, H. G. (2006). Missing-feature based speech recognition for two simultaneous speech signals separated by ICA with a pair of humanoid ears. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 878–885.

Vacher, M., Lecouteux, B. et Portet, F. (2015). *On Distant Speech Recognition for Home Automation*. Springer, 161-188 p.

Valin, J.-M., Michaud, F., Hadjou, B. et Rouat, J. (2004a). Localization of simultaneous moving sound sources for mobile robot using a frequency-domain steered beamformer approach. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1. p. 1033–1038.

Valin, J.-M., Michaud, F. et Rouat, J. (2006). Robust 3D localization and tracking of sound sources using beamforming and particle filtering. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4. p. 841–844.

Valin, J.-M., Michaud, F. et Rouat, J. (2007a). Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering. *Robotics and Autonomous Systems*, volume 55, numéro 3, p. 216–228.

Valin, J.-M., Rouat, J. et Michaud, F. (2004b). Enhanced robot audition based on microphone array sources separation with post-filter. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3. p. 2123–2128.

Valin, J.-M., Rouat, J. et Michaud, F. (2004c). Microphone array post-filter for separation of simultaneous non-stationary sources. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1. p. 221–224.

Valin, J.-M., Yamamoto, S., Rouat, J., Michaud, F., Nakadai, K. et Okuno, H. (2007b). Robust recognition of simultaneous speech by a mobile robot. *IEEE Transactions on Robotics*, volume 23, numéro 4, p. 742–752.

Vermaak, J. et Blake, A. (2001). Nonlinear filtering for speaker tracking in noisy and reverberant environments. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5. p. 3021–3024.

Vorobyov, S. A. (2013). Principles of minimum variance robust adaptive beamforming design. *Signal Processing*, volume 93, numéro 12, p. 3264–3277.

Ward, D. B., Lehmann, E. A. et Williamson, R. C. (2003). Particle filtering algorithms for tracking an acoustic source in a reverberant environment. *IEEE Transactions on Speech and Audio Processing*, volume 11, numéro 6, p. 826–836.

Weinstein, E., Steele, K., Agarwal, A. et Glass, J. (2004). *LOUD : A 1020-node Modular Microphone Array and Beamformer for Intelligent Computing Spaces* (Rapport technique). Massachusetts Institute of Technology.

Williamson, R. et Ward, D. B. (2002). Particle filtering beamforming for acoustic source localization in a reverberant environment. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. p. 1777–1780.

Wilson, K. W., Raj, B. et Smaragdis, P. (2008). Regularized non-negative matrix factorization with temporal dependencies for speech denoising. Dans *Proceedings of INTERSPEECH*. p. 411–414.

Woelfel, M. et McDonough, J. (2009). *Distant Speech Recognition*. Wiley.

Wood, S. U. et Rouat, J. (2016). Blind speech separation with GCC-NMF. Dans *IEEE/ACM Transactions on Audio, Speech and Language Processing*. p. 3329–3333.

Wood, S. U. et Rouat, J. (2017). Real-time speech enhancement with GCC-NMF. *Proceedings of INTERSPEECH*, p. 2665–2669.

Yamamoto, S., Nakadai, K., Nakano, M., Tsujino, H., Valin, J.-M., Komatani, K., Ogata, T. et Okuno, H. G. (2006). Real-time robot audition system that recognizes simultaneous speech in the real world. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 5333–5338.

Yamamoto, S., Nakadai, K., Nakano, M., Tsujino, H., Valin, J.-M., Komatani, K., Ogata, T. et Okuno, H. G. (2007). Design and implementation of a robot audition system for automatic speech recognition of simultaneous speech. Dans *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*. p. 111–116.

Yamamoto, S., Nakadai, K., Valin, J.-M., Rouat, J., Michaud, F., Komatani, K., Ogata, T. et Okuno, H. G. (2005a). Making a robot recognize three simultaneous sentences in real-time. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 4040–4045.

Yamamoto, S., Valin, J.-M., Nakadai, K., Rouat, J., Michaud, F., Ogata, T. et Okuno, H. G. (2005b). Enhanced robot speech recognition based on microphone array source separation and missing feature theory. Dans *Proceedings of the IEEE International Conference on Robotics and Automation*. p. 1477–1482.

Yardibi, T., Bahr, C., Zawodny, N., Liu, F., Cattafesta III, L. et Li, J. (2010). Uncertainty analysis of the standard delay-and-sum beamformer and array calibration. *Journal of Sound and Vibration*, volume 329, numéro 13, p. 2654–2682.

Young, S. (1996). A review of large-vocabulary continuous-speech. *IEEE Signal Processing Magazine*, volume 13, numéro 5, p. 45.

Youssef, K., Argentieri, S. et Zarader, J.-L. (2010). From monaural to binaural speaker recognition for humanoid robots. Dans *Proceedings of the IEEE RAS International Conference on Humanoid Robots*. p. 580–586.

Youssef, K., Argentieri, S. et Zarader, J.-L. (2012a). A binaural sound source localization method using auditive cues and vision. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. p. 217–220.

Youssef, K., Argentieri, S. et Zarader, J.-L. (2012b). Towards a systematic study of binaural cues. Dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 1004–1009.

Zelinski, R. (1988). A microphone array with adaptive post-filtering for noise reduction in reverberant rooms. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. p. 2578–2581.

Zhan, Y., Leung, H., Kwak, K.-C. et Yoon, H. (2009). Automated speaker recognition for home service robots using genetic algorithm and Dempster–Shafer fusion technique. *IEEE Transactions on Instrumentation and Measurement*, volume 58, numéro 9, p. 3058–3068.

Zotkin, D. N. et Duraiswami, R. (2004). Accelerated speech source localization via a hierarchical search of steered response power. *IEEE Transactions on Audio, Speech, and Language Processing*, volume 12, numéro 5, p. 499–508.