UNIVERSITE DE NICE-SOPHIA ANTIPOLIS - UFR SCIENCES
Ecole Doctorale des Sciences et
Technologies de l'Information et de la Communication

THESE

pour obtenir le titre de
Docteur en Sciences
de l'Université de Nice-Sophia Antipolis
Discipline: Automatique, Traitement du Signal et des Images

présentée et soutenue par
LUCA GIULIO BRAYDA

Multi-hypotheses Feedback for Robust Speech Recognition Using a Microphone Array Input

Thèse dirigée par CHRISTIAN WELLEKENS
soutenue le 24-04-2007 avec mention:

Très Honorable

Jury

M. Javier Hernando Pericas - Professeur Rapporteur
M. Pietro Laface - Professeur Rapporteur
M. Christian Wellekens - Professeur Directeur de thèse
M. Maurizio Omologo - Examinateur

Dedicated to my grandmother Ida

Abstract

The aim of this work is improving robustness of speech recognizers using microphone arrays. Performing Automatic Speech Recognition (ASR) in real environments is as much difficult as the amount of noise increases and the speaker is far from the microphone. Recent studies showed that speech quality in terms of Signal to Noise Ratio (SNR) can be increased using microphone arrays. By exploiting the spatial correlation among multi-channel signals, one can steer the array toward the speaker (beamforming). This can be done by simply exploiting inter-channel destructive interference of noise with a delay-and-sum technique, where inter-sensor delays are estimated and applied to each channel signal. Alternatively, per-channel filters (filter-and-sum) can be implemented: these filters can be fixed or adapted on a per-channel or per-frame basis, depending on the chosen criterion. In this work we address the problem that increasing the SNR does not imply increasing recognition performance to the same extent. Seltzer [2004] proposes to apply an adaptive filter-and-sum beamformer (Limabeam) based on a Maximum Likelihood (ML) criterion rather than on the SNR. In this method, a set of Finite Impulse Response (FIR) filters of a filter-and-sum beamformer is adapted in an unsupervised way using clean speech models which best align noisy speech features. Then the recognizer uses the sum of the filtered signals to generate a final transcription. In this thesis we show that considering in parallel N-best hypotheses instead of the best one, prior to optimization, can increase recognition performance close and beyond to that of a supervised algorithm: indeed, we exploit the acoustical confusability of the first N-best hypothesized transcriptions to establish a set of competing hypotheses. Acoustic features of each hypothesis are then optimized in parallel, producing concurrent adaptive FIR filter sets which are strongly related to their related hypothesis. We show that transcriptions maximizing the Word Recognition Rate (which is the optimal criterion for ASR), together with similar acoustically confusable transcriptions, are able to be better optimized than the simple first output of a Viterbi decoder. The initial N-best list of hypotheses is consequently automatically re-ranked, under a ML criterion, after a second recognition step is performed: the main result is that the new ML hypothesis is closer to

the right transcription and thus performances are improved. The proposed algorithm is able to efficiently recover errors initially made by the recognizer in the first recognition step. The framework of the *N-best Limabeam* was tested in one simulated and two real different challenging environments: a very low SNR environment with pure additive noise, a low reverberation insulated room with many surrounding additive noise sources and a large meeting room with a high reverberation time. Being the latter environment the more realistic scenario, a database which mimics different talker positions and head orientations is used. We explored the potential of delay-and-sum beamforming and of the proposed algorithm when varying the pointing direction of the microphone array and we derive a Recognition Directivity Pattern (RDP), which is very useful to understand the impact of reflections affecting automatically recognized speech signals. Finally, we exploit additional information related to the environment, such as a set of measured room impulse responses: we propose a Training-set based version of the Calibrated Limabeam and we derive an upper bound for performances when coupling Matched Filtering with the proposed N-best UL. The proposed methods provide a form of room equalization via Maximum Likelihood, recognition-oriented filters.

Thesis Outline

This work is organized as follows:
In Chapter 1 we review the current approaches for Speech Enhancement and Speech Recognition when a microphone array input is chosen. We stress on the fact that they represent two different, but possibly complementary problems. In Chapter 2 we outline the Limabeam algorithm, which this work is based upon, together with the first results obtained in a simulated environment affected by additive noise. In Chapter 3 we introduce the proposed N-best Unsupervised Limabeam, detailing motivation, principles and providing insight on its performances in both simulated and real environments, primarily affected by additive noise. In Chapter 4 we face the reverberation problem, by confirming the effectiveness of the proposed technique, evaluating recognition performances in function of the relative position and direction of both speaker and microphone array, and proposing further ways to compensate for room reflections. In Chapter 5 we outline and detail the analysis and interventions at the hardware level, on the microphone array used to collect most of the database used in this work. Finally, in Chapter, 6 we provide concluding remarks and possible future improvements.

Acknowledgments

"Imagination, Courage, Perseverance" (Ken Pope)

The first and most important person I intend to acknowledge is my PhD advisor, Christian Wellekens, who believed in me since I proposed myself as a PhD applicant. Both from a scientific and a human point of view he has been a light in the darkness of the undiscovered paths of science and strategy. I consider myself lucky, because he let me choose my road, with almost no restriction in time and space, trusted and supported me in good and bad moments.

The SHINE Team of Trento comes next: I would like to thank Maurizio Omologo for his great sense of collaboration and kindness, which revealed to be fundamental for my achievements. Thanks to Marco Matassoni, Piergiorgio Svaizer, Luca Cristoforetti, Alessio Brutti, Christian Zieger and Alessandro Giacomini for their skilled support in every moment of need. Thanks also to Gianni Lazzari. Special thanks for the colleagues and friends of the office 532: because of you, coming to work has always been fun as well.

Thanks to the people who encouraged me in taking the right decision almost four years ago, when I had no idea of what a thesis was: Luca Rigazio, Javier Hernando, Patrick Nguyen, Xavi Anguera.

A particular acknowledgment goes to Claudio Bertotti, whose skills impressed me and whose sympathy, cleverness and passion demonstrate the scientific irrelevance of human hierarchies. Science should have (and listen to) more people like Claudio.

I would like to acknowledge the people who helped me extricate in the jungle of French and Italian bureaucracy: Sonia Bernabè, Christine Mangiapan, Gwenaelle Le-Stir. Thanks also to the skilled system administrator Patrick Petitmengin.

I learned after a while that my family's open mindedness is much bigger that what I imagined: my father Giorgio, my mother Flavia and my brother Francesco constantly supported me and sought to understand my state of mind.

My girlfriend Chiara has always stood by me, showing exemplar empathy and patience. Her suggestions taught me to turn the lead into gold, helping me to transform some apparent defeats into opportunities for self-consciousness and personal evolution.

Thanks to all the people who shared even just one coffee break with me: very frequently I found in these moments comfort and relief. Thanks also to the several doctoral students of my department, whom made my lunch pauses stimulating and funny.

A special acknowledgment goes to the mountains of Trentino and to the sea of the French Riviera, which stand and lays to remind me that my contribution is just a little brick in the giant foundations of human knowledge and that watching above me can help clarifying the things inside me. Now it's time to say "Berghail!" from the top of this three-years mountain, which I ended up to climb.

Thanks to the wind, which allows me to sail against it, if necessary, and to have fun in the regatta of my life.

Résumé de la thèse

Objectif de la thèse

Le but de ce travail est d'accroître la robustesse des reconnaisseurs de parole en utilisant des réseaux de microphones. Reconnaître la parole (ASR) à partir de microphones distants en environnements réels est un tâche ardue parce que le signal désiré étant éloigné du microphone, il est distordu par la réponse impulsionnelle de la pièce. En outre, le reconnaisseur doit faire la distinction entre ce signal et d'autres sources sonores outre le bruit additif. Finalement, le locuteur peut se déplacer continûment par rapport au microphone. Ceci constitue l'effet bien connu sous le nom de "cocktail party" qui est une des causes majeures d'erreurs en reconnaissance de la parole.

Des études récentes montrent qu'en présence de bruits additifs ou convolutionnels, la qualité de la parole en termes de rapport signal/bruit (SNR) peut être améliorée par l'utilisation de réseaux de microphones parce qu'ils ajoutent la connaissance spatiale au signal multi-canaux. Accroître le SNR n'implique pas automatiquement accroître les performances du reconnaisseur : c'est ce problème que nous traitons dans cette thèse.

Seltzer [Seltzer, 2004] propose d'appliquer une formation de faisceau de type filtrage-sommation adaptative basée sur un critère du maximum de vraisemblance (ML) plutôt que sur le SNR. Dans cette méthode,le banc de filtres à réponse impulsionnelle finie (FIR) de la formation de faisceau filtrage/sommation est adapte de façon non-supervisée en utilisant des modèles de parole propre qui réalisent l'alignement optimal avec les traits de parole bruitée. Ensuite, le reconnaisseur utilise la somme des signaux filtrés pour générer une transcription finale du signal. Dans cette thèse, nous montrons que prendre en considération en parallèle les N meilleures hypothèses finales au lieu d'une seule avant l'optimisation, permet d'approcher le taux de reconnaissance obtenu par un reconnaisseur supervisé : en effet, nous exploitons la confusion acoustique des premières N-meilleures transcriptions hypothétiques pour établir un ensemble d'hypothèses compétitives. Les traits acoustiques de chaque hypothèse sont alors optimisés en parallèle et produisent des ensembles de filtres

FIR adaptatifs compétitifs qui sont étroitement liés à leur hypothèse. Nous montrons que les transcriptions minimisant le taux de reconnaissance de mots (WRR : qui est le critère optimal pour l'ASR), en association avec des transcriptions similaires acoustiquement proches, conduisent à des meilleures optimisations que l'unique meilleur résultat du décodeur Viterbi. La liste initiale des N-meilleures hypothèses est ensuite automatiquement réordonnée selon le critère ML après qu'un second pas de reconnaissance ait été réalisé : le principal résultat est que la nouvelle hypothèse ML est plus proche de la transcription correcte et donc que le taux de reconnaissance est amélioré. L'algorithme proposé peut efficacement corriger des erreurs faites initialement par le reconnaisseur dans le pas initial.

Amélioration du signal de parole et reconnaissance

L'objectif de ce chapitre 1 est de décrire et de différencier les concepts d'amélioration et de reconnaissance de la parole. Le but de l'amélioration de la parole est d'accroître le rapport signal/bruit de la parole bruitée. Grâce au réseau de microphones, le domaine spatial peut être pris en compte en même temps que le domaine temps-fréquence habituel dans le traitement des signaux audio. Avec les réseaux de microphones, on peut tirer parti de la corrélation spatiale entre les signaux multi-canaux et diriger le réseau vers le locuteur (*formation de faisceau*). La formation de faisceau est une technique dérivée de la théorie des antennes [Venn and Buckley, 1988]. Son objectif est de former un *faisceau* vers la source d'intérêt. Les sources de bruit inopportunes sont de ce fait atténuées. La formation de faisceau s'adresse principalement au bruit additif et vise à maximiser *le gain de réseau*, c'est à dire le rapport entre le SNR du réseau avec celui du simple microphone. Cependant, comparé aux usages en radar et sonar, application de la formation de faisceau en parole est différente pour les raisons suivantes [Compernolle, 1990] :

1. La position de la source (le locuteur) est rarement fixe par rapport au réseau.
2. Le log SNR est généralement positif (pour des valeurs négatives, l'effet Lombard est observé [J-C. Junqua, 1996]), sauf pour la réverbération, où la puissance du signal diffusé est même plus elevée que celle du signal non-réfléchi.
3. La parole n'est pas un signal à bande étroite et les perturbations peuvent avoir un spectre de même contenu que la source utile.
4. Le contenu spectral de la parole change continuellement , parfois rapidement et alterne avec des périodes de silence.

Toutes ces différences doivent être prises en compte lors du développement d'un formateur de faisceau pour le traitement de la parole. Différentes techniques existent et certaines sont plus adaptées pour des champs de bruit spécifiques (non-cohérent, cohérent, diffus). Par exemple, on peut simplement exploiter l'interférence destructive du bruit entre canaux par la technique retards-et-sommation, où les retards entre senseurs sont estimés et appliqués au signal de chaque canal. Une autre technique consiste à développer un filtre par canal (technique filtrage-et-sommation) : ces filtres peuvent être fixes ou adaptatifs par canal ou par trame, selon le critère choisi. Intuitivement, accroître la qualité de la parole devrait tout naturellement accroître le taux de reconnaissance. Cependant, lorsqu'on améliore la parole, on tente de corriger les caractéristiques spectrales pour que l'intelligibilité et le confort auditif mesurés qualitativement par un auditeur humain ou quantitativement par le SNR (ou autres mesures similaires) soient accrus.

Tous les phénomènes mis en oeuvre pour qu'un humain puisse effectivement distinguer et transcrire un quelconque contenu lexical sont considérés comme naturels. Ces phénomènes peuvent être

- Une robustesse naturelle aux bruits non-stationnaires et à l'annulation du signal et sa distorsion
- L'effet Lombard (parce que nous essayons constamment de dépasser la puissance du bruit lorsque le SNR tombe sous 0dB, nous prêtons attention au signal le plus puissant)
- L'effet de masquage (le bruit en des fréquences particulièrement proches d'harmoniques très visées perturbe faiblement l'intelligibilité), le degré de séparation aveugle de sources dont les humains sont capables (qui résoud partiellement le problème de plusieurs locuteurs parlant simultanément).
- L'extraordinaire capacité d'adaptation rapide à des environnements complètement nouveaux
- D'autres techniques de compensation psychoacoustique dont nous avons à peine connaissance.

La reconnaissance de parole est sensiblement différente : dans la plupart des applications, le système n'a d'autres connaissances qu'un ensemble de modèles statistiques entraînés par exemple sur une grande base de données de parole propre et phonétiquement équilibrée prononcée par plusieurs locuteurs. L'espace mathématique dans lequel la parole est représentée avec ses traits discriminants fondamentaux (ou l'objectif est de discriminer entre des unités lexicales, qu'elles soient des mots ou des phonèmes) est robuste à un certain niveau et à une certaine nature de bruit mais certainement pas dans tout scénario et spécialement en présence de réverbération modérée ou forte.

Comme mentionné dans [Brandstein and Ward, 2001], il n'y a pas de relation directe entre le SNR qui est une mesure évaluée perceptuellement et la performance d'un reconnaisseur de parole.

Dans les environnements bruités, un énorme besoin de robustesse du reconnaisseur de parole est requis et la robustesse en reconnaissance dépend directement *de la façon* dont le signal multi-canaux a été optimisé. Une intense activité est consacrée mondialement à l'évaluation des performances des reconnaisseurs de parole utilisant un réseau de microphones, en particulier dans les communautés liées aux projets européens AMI et CHIL : NIST a récemment organisé des campagnes de tests de référence NIST [2004] qui ont montré que le taux d'erreur obtenu avec un réseau de 64 microphones est environ le double de celui obtenu par un microphone de proximité pour une tâche de reconnaissance de parole spontanée de grand vocabulaire. Un tel scénario impose de considérer l'ensemble de paramètres suivants :

1. le type et le nombre de sources de bruit entourant locuteur
2. le nombre de microphones
3. le type et le nombre de traits utilisés pour la représentation de la parole à reconnaître
4. la taille du vocabulaire.

Dans ce travail, nous analysons les performances de la reconnaissance de parole sur une tâche relativement simple avec un réseau de 8 microphones. Nous faisons varier les degrés de liberté concernant les sources de bruit et les traits : ainsi, nous pouvons étudier un scénario assez réaliste et de complexité acceptable La plus grande partie de la littérature concernant la reconnaissance de parole avec des réseaux de microphones rapporte des techniques dont les principes sont hérités de la robuste ASR mono-canal classique. Plus spécifiquement, elles utilisent trois blocs fondamentaux en cascade :

1. le bloc d'amélioration de la parole produisant un signal plus intelligible
2. le bloc d'extraction des traits, qui est responsable de la transformation du signal en une suite temporelle de vecteurs de traits appropriés (robustes)
3. le bloc de décodage (par exemple, un reconnaisseur base sur les HMM (modèles de Markov cachés) qui compare la séquence de traits extraits avec les modèles de parole préalablement entrainés et fournit la transcription

Les deux premiers blocs sont désignés comme le Front-End tandis que le troisième forme le Back-End. Lorsqu'on utilise un réseau de microphones, le bloc d'amélioration de la parole peut être un système MISO (entres multiples/sortie unique) ou le signal multi-canaux est traité pour produire un signal unique de SNR plus élevé.

Cependant, la recherche actuelle sur ce sujet a montré que la quantité d'information qui peut être échangée entre le Front-End et le Back-End n'est pas nécessairement représentée par la sortie

d'un système MISO et que le problème est plus complexe. Les possibilités de la reconnaissance avec réseaux de microphones sont importantes car le signal multi-canaux enregistré apporte plus d'information au système : pour cette raison, les trois blocs fondamentaux peuvent être organisés différemment, peuvent même être fusionnés ou de nouveaux chemins d'information peuvent être établis entre eux. Outre *l'accroissement* de la quantité d'information échangée, il est intéressant de reconsidérer le *critère* selon lequel les traits sont optimisés : lorsque le problème de l'amélioration du signal de parole est applique dans le cadre de la reconnaissance, certaines implications considérées comme évidentes du point de vue perceptuel (RSR, SNR, direction du rayonnement, désaccord des microphones) ne sont plus valides (nous prouverons cela dans cette thèse) : tout critère fondé sur ces mesures peut faillir dans certaines conditions expérimentales. En définitive, ce n'est pas sur des critères perceptuels que les résultats de la reconnaissance sont évalués : la métrique cumulative en décodage Viterbi utilise la vraisemblance des traits justifiant un certain modèle. Cette importante considération montre que se focaliser sur des critères basés sur la vraisemblance est la meilleure approche dans un cadre d'une telle complexité.

Limabeam et la reconnaissance avec rétroaction

Au chapitre 2 nous expliquons en détail une technique [Seltzer, 2004] pour améliorer les reconnaisseurs de parole basés des réseaux de microphones. Elle consiste à introduire une boucle de rétroaction entre le reconnaisseur et le bloc d'amélioration de la parole. Dans cette technique nommée l'algorithme Limabeam une formation de faisceau par filtrage-et-sommation est controlée par la sortie d'un premier pas de reconnaissance. Les coefficients des filtres sont choisis afin de maximiser la vraisemblance des traits bruit/'es justifiant le modèle. Un second pas de reconnaissance fournit une performance accrue.

Le formateur de faisceau (BEAMformer) est contrôlé par le critère de vraisemblance (MAximum LIkelihood).La technique est attrayante car l'optimisation est menée dans un domaine beaucoup plus proche du reconnaisseur que du bloc d'amélioration du seul SNR.

Plus spécifiquement, l'algorithme peut être appliqué de trois différentes façons : le Limabeam Oracle(OL) qui suppose correct le premier pas de reconnaissance (il est utile d'observer la relation entre la transcription Oracle et le formateur de faisceau optimisé dérivé) ; le Limabeam Calibré (CL) adapte les filtres selon une seule transcription Oracle, gèle ensuite l'ensemble des filtres et traite l'ensemble test restant en utilisant ces filtres. L'avantage de cette technique est que la complexité de calcul se limite à la phase de calibration Le désavantage évident est que la calibration doit être

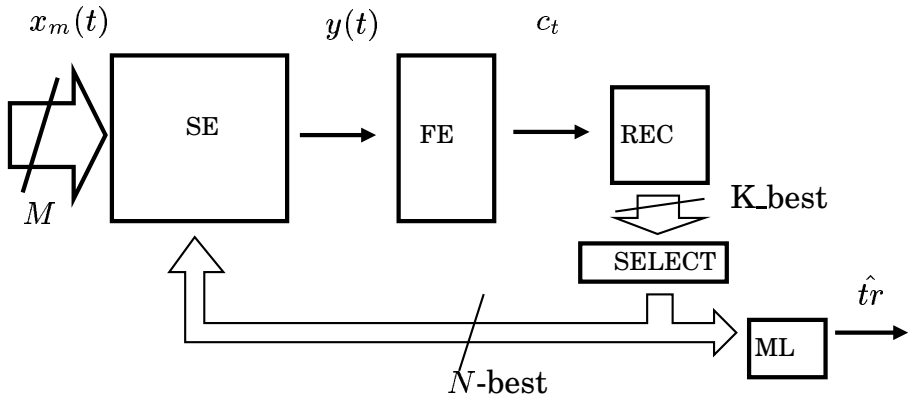


FIG. 1. Système de reconnaissance propose a réseaux de microphones et a rétroaction a hypothèses multiples

faite pour chaque variation de locuteur et de position et de la direction de la source de bruit dans l'environnement : cette variation n'est généralement pas connue a priori.

Finalement, si les données de calibration ne sont pas disponibles, les filtres sont optimisés directement à partir du résultat du premier pas de reconnaissance qui peut être aussi correct que faux. Dans ce cas, nous parlons de *Limabeam non-supervisé (UL)* [Seltzer, 2002]. Tout au long de cette thèse, nous traiterons des 3 versions de Limabeam.

Limabeam non-supervisé avec critère sur les N-meilleures transcriptions (N-best UL)

L'algorithme Limabeam est une des techniques permettant au reconnaisseur d'échanger de l'information avec le formateur de faisceau.

Cependant, le volume d'information dans la rétroaction peut être augmenté afin d'obtenir de meilleures performances. Au chapitre 3 nous proposons un schéma alternatif pour la Reconnaissance de Parole. La technique proposée [Brayda, 2006c] se base sur l'application de Limabeam et peut être considerée comme une généralisation de l'algorithme. Nous montrons et exploitons le fait que contrôler le formateur de faisceau à l'aide de filtres estimés à partir d'une première hypothèse de transcription n'est pas nécessairement la meilleure solution. Au contraire, nous essayons d'estimer les meilleurs filtres à partir de plusieurs transcriptions *concurrentes* et ensuite nous choisissons la meilleure transcription selon un critère de vraisemblance maximale. Les transcriptions concurrentes sont extraites des N-meilleures réponses des reconnaisseurs.

Un macro-bloc du schéma proposé est représente en figure 1.

Le système fonctionne partiellement comme une chaîne classique formée de l'amélioration de

parole (SE), l'extraction de traits(FE) et la reconnaissance (REC). L' élément de rétroaction est comme dans le cas du Limabean, constitué de transcriptions. Cependant, à partir du bloc REC nous étendons la rétroaction d'une seule hypothèse de transcription aux K-meilleures premières transcriptions. Dans le bloc $SELECT$ nous réduisons le nombre de telles hypothèses en un ensemble plus petit où chaque transcription est la plus différente possible des autres. Un tel ensemble contient N-meilleures transcriptions (et N $\ll$ K), qui sont transmises aux optimiseurs.

A partir de ce nouvel ensemble, N-meilleures optimisations *parallèles* sont effectuées dans le bloc SE et le signal multi-canaux est reformé en faisceau comme cela se fait dans la phase d'optimisation du Limabeam. La différence est que nous augmentons le nombre d'optimisations aux N-meilleures : notre intention est de mettre toutes les transcriptions choisies en compétition et d'obtenir une nouvelle transcription de vraisemblance maximale. Une fois que l'optimisation a convergé, de nouveaux traits sont extraits et la reconnaissance effectuée. Alors qu'avec le Limabeam conventionnel, nous n'avons à ce point qu'une seule hypothèse supposée correcte mais pouvant être erronée, notre méthode choisit parmi les N-meilleures nouvelles hypothèses selon un critère ML. La transcription finale $\hat{tr}$ est donc trouvée. Nous montrerons que l'approche proposée que nous appelons *Limabeam N-meilleures non-supervise* (N-best UL), est capable d'améliorer significativement les performances de reconnaissance par rapport au système retards-et-sommation, Limabeam non-supervisé et Limabeam Oracle.

Notre algorithme est capable de reclasser automatiquement un ensemble de conjectures initiales permettant ainsi de *corriger des erreurs* faites après le premier pas de reconnaissance mieux que le simple Limabeam non-supervisé : en fait, la chance est plus èlevée de sélectionner dans un plus grand ensemble une transcription plus proche de la transcription correcte que celle de première hypothèse. Notre algorithme est même efficace lorsqu'il est appliqué à une liste où la transcription correcte *n'est pas* la premier choix, ce qui est toujours le cas lors d'une erreur de reconnaissance. En effet, l'optimisation des traits en parallèle réduit la contrainte sur l'ordre des vraisemblances produites après une première passe de reconnaissance. Chaque vraisemblance de chaque vraisemblance concurrente croit durant l'optimisation et croit une transcription *différemment*. Nous observons que l'accroissement de la vraisemblance par la technique proposée est logarithmique en fonction du nombre de N-meilleures hypothèses considérées.

En conséquence, des hypothèses de taux d'erreurs minimum peuvent être promues dans la nouvelle liste des N-meilleures et finalement arriver au premier rang. Nos expériences montrent que le rang de la transcription correcte est toujours amélioré.

Outre ce reclassement automatique, il est utile de remarquer que notre approche exploite effi-

cacement la *confusabilité acoustique* entre transcriptions : plus spécifiquement, la présence de la transcription correcte dans la liste des N-meilleures transcriptions n'est pas indispensable. Comme la performance du Limabeam Oracle est limitée pour notre tâche, nous observons qu'une transcription aisément confondue peut optimiser les filtres mieux que la transcription correcte parce qu'elle agit comme "attracteur".

Environnement et tâches

Un réseau linéaire équidistant est choisi pour sa géometrie idéale semblable à celui qui est utilisé dans la plupart des applications en environnement réel. Les chiffres anglais de la base de données TI-digits [Leonard, 1984] sont choisis comme tâche pour l'ensemble de cette thèse dans laquelle nous évaluons l'algorithme N-best UL dans des conditions de bruit difficiles : d'une part nous devons réduire la complexité du formateur de faisceau et donc nous essayons de limiter la complexité du Back-End et donc la taille du vocabulaire. D'autre part, TI digits est un corpus très largement répandu et ainsi les résultats de ce travail pourront entre comparés par d'autres chercheurs. La base de données TI digits contient des prononciations de séquences de chiffres anglais enchaînés enregistrés à l'aide d'un microphone de proximité par des locuteurs américains de sexes, ages et accents différents. Comme modèles de mots, nous utilisons des HMM gauche-droite a 18 états. Les distributions de sortie sont des densités mono-Gaussiennes. Dans le Front-End, on extrait 12 coefficients Mel-cepstraux plus la log-énergie et leurs derivées premières et secondes soit un total de 39 traits. Ces traits sont calcules toutes les 10ms en utilisant une fenêtre de Hamming glissante de 25ms. La gamme de fréquences couverte par le banc de filtres de l'échelle Mel est limitée à 700-7500 Hz pour éviter les gammes de fréquence ne contenant pas de signal utile. La normalisation de la moyenne cepstrale est appliquée. L'algorithme N-best UL est évalué dans quatre environnements classés par difficulté de reconnaissance croissante

1. Un environnement simulé, où un réseau enregistre de la parole distante affectée d'un bruit blanc émis latéralement d'une direction collinéaire au réseau.
2. Un environée par un bruit de ventilateur émis latéralement d'une direction collinéaire au réseau. Le bruit du ventilateur est réel.
3. Un environnement réel, où dans une chambre silencieuse un réseau enregistre une source unique de parole entourée de plusieurs sources de bruit important.
4. Un environnement réel et très agressif, où a l'intérieur d'une pièce fortement réeverbérante un réseau enregistre de la parole émise en dix différentes places pour simuler les différentes

positions du locuteur et ses orientations de tête.

Le premier environnement est utile pour comprendre les limites théoriques d'un formateur de faisceau à vraisemblance maximale : spécifiquement,nous montrons que le retards-et-sommation maximise à la fois le SNR et le taux de reconnaissance avec du bruit blanc.

Le N-best UL est utile lorsque le bruit additif devient plus cohérent, c'est à dire réaliste : dans le second environnement, une amélioration significative est obtenue (près de 19% d'amélioration relative par rapport à UL).

Dans le troisième scénario, le champ de bruit correspond à un log SNR inférieur a zéro, le type de bruit est plutôt diffus, donc le gain de N-best UL est plus faible (4.6%). Ceci arrive aussi parce que dans ce scénario le locuteur est non-contraint et que les retards inter-microphoniques sont calculés en temps réel en utilisant une technique basée sur la phase de l'inter-spectre de puissance (CSP). Une estimation du retard basée sur la CSP trouve les directions de cohérence maximale où on peut trouver la position du locuteur. Nous nous concentrons sur cet environnement réel [Brayda, 2006b], où il est intéressant de présenter une analyse plus profonde du signal de parole optimisé dans les domaines temps, fréquence, spectre et cepstre : dans tous les cas, l'utilité de la méthode proposée est mise en évidence.

Nous discutons aussi les relations possibles entre les taux plus élevés de reconnaissance, la réponse fréquentielle en amplitude et phase des filtres de vraisemblance maximale.

Des expériences utilisant un domaine d'optimisation différent sont aussi presentées et discutées. Le grand meeting est le quatrième scénario, plus réaliste que nous étudions au Chapitre 4. Dans cette pièce, les phénomènes suivants sont présents simultanément :

- La parole est affectée par les réflexions des murs et il n'est plus possible de faire l'hypothèse de l'indépendance statistique entre la parole et les perturbations.
- La parole est fréquemment prononcée loin du réseau de microphones qui est généralement installe sur les tables ou sur les murs. Ceci accroît l'écho dans le signal de parole.
- Le locuteur peut se déplacer et tourner la tête pendant qu'il parle : plus précisement lorsque le locuteur ne s'oriente pas vers le réseau de microphones, la parole capturée par chaque microphone du réseau sera principalement caractérisée par des contributions de réflexions. Ceci affecte significativement les performances du reconnaisseur.

Le potentiel de la formation de faisceau retards-et-sommation et du N-best UL est exploré [Brayda, 2006a] pour des variations de direction de cible ou pour des retards-et-sommation *contrôlé par* θ. De cette analyse, nous obtenons un diagramme de directivité de reconnaissance (RDP) qui est très utile pour comprendre l'impact des réflexions sur les résultats de la reconnaissance. Nous

trouvons une corrélation entre les pics de RDP et ceux de la transformée de Fourier inverse de la CSP mentionnée plus haut : ceci suggère que les directions de reconnaissance maximale sont liées-canaux maximales , quoi que pas nécessairement avec la même amplitude relative.

Alors que nous obtenons des améliorations avec des filtres très courts lorsque le bruit est principalement additif, dans de tels environnements, le problème à résoudre est très différent. C'est pourquoi, pour l'optimisation nous décidons d'exploiter une information additionnelle à cet environnement comme un ensemble de réponses impulsionnelles mesurées : à ce point de vue, une nouvelle version du Limabeam appelée Limabeam calibré sur l'ensemble d'entraînement (TCL) est proposée et experimentée. Les filtres à vraisemblance maximale dérivés du TCL sont très courts comparés aux différentes réponses impulsionnelles. En conséquence, ils ne peuvent pas physiquement prendre en compte et compenser les réflexions précoces et tardives. Une façon de prendre en compte toutes les réflexions est le filtrage adapté (Matched Filtering (MF)). Nous montrons que la connaissance exacte de la reponse impulsionnelle est une contrainte sevère et qu'en négligeant ne fut ce que l'orientation de la tête décroit les performances de façon catastrophique. Nous étudions les performances de reconnaissance avec le filtrage adaptatif. Finalement, nous combinons le filtrage adaptatif, qui est , à notre connaissance, la methode la plus efficace pour accroitre la reconnaissance en environnements réverbérants, avec notre N-best UL bien adapté au bruit additif et nous montrons que cette association conduit à̀ de nouvelles améliorations.

En considérant que dans ce dernier scénario, la distance moyenne entre le locuteur et les microphones est d'environ 3,5m (et le locuteur ne fait fréquemment pas face au réseau), une performance de 76.4% absolue est relativement elevée. Les méthodes proposées fournissent une sorte d'égalisation de la pièce au moyen de filtres dédiés à la reconnaissance selon le critère de la vraisemblance maximale.

Le réseau de microphones MARKIII modifié

Le chapitre 5 est consacré à l'équipement utilisé pour acquérir les bases de données utilisées dans toutes les expériences de ce travail. Cet équipement est une version modifiée du réseau de microphones NIST MarkIII [Rochet, 2004], un système capable d'acquérir 64 signaux audio synchrones à 44.1 kHz, initialement pour la reconnaissance de parole à distance, la localisation du locuteur et en général pour l'acquisition et l'amélioration de messages vocaux pour des applications mains-libres. Les premières expériences utilisant le MarkIII original [Brayda, 2005b] avaient

montré que la cohérence entre une paire générique de signaux était affectée par un biais dû au bruit électrique de mode commun. Une modification technique fut realisée pour éliminer chaque source interne de bruit des modules analogiques. Le réseau ainsi modifié [Brayda, 2005a] fournit une qualité de signaux d'entrée attendus par la théorie. Sans ce re-design partiel, toute tentative d'estimation fiable des retards entre canaux (utilise dans le retards-et-sommation controlé par le CSP et le N-best UL) et des réponses impulsionnelles de pièce (nécessaires pour TCL et MF) auraient conduit à des résultats non-réalistes et imprédictibles. L'analyse et le re-design du réseau décrits dans ce chapitre ont été transférés aux Universités de Karlsruhe (Allemagne), Barcelone (Espagne) et ITC-irst (Trente, Italie) où le MarkIII modifié est à présent utilisé.

Contributions de cette thèse et conclusion

Dans ce travail, nous améliorons les performances d'un reconnaisseur de parole dans divers scénarios où les conditions environnementales présentent généralement un sérieux problème. L'environnement introduit un important divorce entre les conditions d'entraînement sous lesquelles la représentation statistique de la parole est dérivée (parole propre, microphone de proximité) et les conditions de test où le bruit additif et/ou la réverbération modifie substantiellement les caractéristiques du signal. La dégradation des performances observées dans ces circonstances et en pratique dans toute applications du monde réel, peut être partiellement compensée grâce à l'usage de réseaux de microphones. Les réseaux de microphones ont été utilisés extensivement pour améliorer la qualité auditive du signal et sont devenus des outils essentiels lorsque la parole est capturée en environnement bruyant ou réverbérant ou imposer aux locuteurs le port de micros-cravate ou de revers est indésirable ou inconfortable. Cependant, puisque les reconnaisseurs de parole ne fonctionnent pas selon les mêmes principes que l'oreille humaine, l'amélioration du signal de parole n'accroît pas proportionnellement les performances d'un reconnaisseur distant du locuteur. Pour gérer cette limitation, nous considérons l'algorithme Limabeam proposé par Seltzer en 2003 avec l'objectif d'améliorer la parole venant d'un réseau de microphones en utilisant le même critère que celui de la reconnaissance et non simplement pour atteindre une meilleure qualité auditive. Ceci est réalisé en insérant une boucle de rétroaction entre le reconnaisseur et le système d'amélioration de la parole à savoir un formateur de faisceau filtrage-et-sommation : la rétroaction est constituée d'une hypothèse de transcription à partir de laquelle un ensemble de filtres FIR est adapté. Cependant, un reconnaisseur classique considère non pas une seule transcription mais bien un ensemble d'hypothèses avant de proposer une transcription finale : avec cette même phi-

losophie, nous généralisons le Limabeam en accroissant le nombre de rétroactions et en construisant un ensemble d'optimisations parallèles. Le nombre d'hypothèses considérées est fonction des premières N-meilleures sorties du reconnaisseur. Donc, N-meilleures optimisations concourrent à la génération d'autant d'ensembles de filtres, chacun ayant une fonction objectif différente, chacune étroitement liée aux hypothèses de l'entrée.

Après l'optimisation, une seconde reconnaissance est menée sur le groupe de traits optimisés et on choisit la meilleure parmi les hypothèses en compétition sous un critère de vraisemblance maximale. Avec cette technique, nous pouvons surpasser à la fois le formateur de faisceau classique retards-et-sommation et l'algorithme Limabeam original.

Considèrer plus d'hypothèses jusqu'à la fin de l'optimisation est la clef de l'accroissement des performances. La présence de la transcription correcte dans la liste des N-meilleures hypothèses s'avère ne pas être indispensable puisque les filtres à vraisemblance maximale peuvent dériver de transcriptions faciles à confondre acoustiquement avec la transcription correcte.

Nous mettons en évidence que l'algorithme proposé est capable de corriger les erreurs faites par le reconnaisseur à la première étape. En outre, l'approche des N-meilleures hypothèses proposée a été testée dans des environnements applicatifs : un milieu très bruité et une salle de réunion très réverbérante. Dans les deux cas, la technique s'est montrée fructueuse. Dans ce scénario, nous observons que la performance de reconnaissance dépend principalement de deux facteurs : la direction vers laquelle pointe le réseau et celle dans laquelle parle le locuteur. Nous montrons que contrôler le premier facteur est essentiel et que les meilleures directions vers lesquelles pointer le réseau dans des applications de parole peuvent être à la fois la position du locuteur et les murs de la pièce qui agissent comme des sources auxiliaires. Nous montrons aussi que, à notre meilleure connaissance, les performances les plus elevées dans un tel scénario peuvent être obtenues en intégrant la connaissance *a priori* de la réponse impulsionnelle de la pièce par le filtrage adaptatif et le N-best UL.

Contents

List of Figures

List of Tables

Chapter 1

State of the Art

1.1 Introduction

During the last three decades [Huang, 2001], performance of speech recognizers significantly increased even for large vocabulary tasks. Results in research are driving the attention of the Information and Communication Technology community toward speech technology, because integrating speech in a Human-Computer Interface (HCI) certainly leads to a better usability and efficiency. However, robustness of recognizers to environmental noise has always been an issue which many researchers are currently addressing: the upper-bound performances of recognizers are generally achieved when a close-talk microphone is recording the voice signal, i.e. when no competing speaker or noise sources and no reverberation affects the original, clean speech signal. Many desired applications, such as automatic transcription of meetings, command recognition in a car environment, voice-driven agenda on a cellphone, can require the speaker to be either far from the microphones, or to be surrounded by one or many noise sources, or both. In these situations speech recognizers dramatically fail to reach the minimal threshold of performance that the usability is requiring, even with a short vocabulary size. The use of more sensors has shown significant advantages in the field of antennas [Venn and Buckley, 1988], because at least two receivers can have a spatial knowledge of the desired transmitter location. The spatial dimension is exactly what makes microphone array useful in order to improve the quality of the speech signal prior to recognition. The term "quality" is generally associated to the extent the Signal-to-Noise Ratio (SNR) is increased. However, there is no direct relationship between a higher intelligibility due to SNR improvements and higher speech recognition score [Omologo, 2001]. This motivates us in considering the multi-channel signal as a

set of data which has to be combined so the output best matches the clean speech models, without necessarily making this output sound better to our ears.

1.2 Architectural constraints to environmental robustness

A speech recognizer is as much efficient as it is capable of compensating speech variabilities. These variabilities can be intrinsic (accents and style, speech rate, different speakers and gender, degree of spontaneity) or extrinsic (additive noise, channel). A recognizer being able to perform such compensations (ideally: all of them) is said to be *robust*. If recognition is to be done *hands-free*, then the effect of latter variabilities is as big as the user is far from the microphone. This work focuses on robustness to the extrinsic variabilities, which are caused by the environment where speech is uttered, for hands-free applications.
Specifically, the environments in which the hands-free speech signal is recorded can be roughly classified in three classes:

- Small enclosures: car compartments, elevators, cockpits.
- Medium-large environments: from small office, meeting rooms, to conference halls and worship places.
- Open air: urban and extra-urban spaces.

The first and third classes imply speech being only partially affected by reverberation (generally caused by wall and window reflections), while significant disturbances may come from additive noise contributions located around the source of interest (surrounding speakers, cars passing by, open window, talking radio). In these scenarios, the speaker is generally not very far from the microphone and noise can be estimated and compensated in very different ways, among which we cite Spectral Subtraction [Boll, 1979], aiming at estimating and subtracting (with thresholds) the noise power from the noisy speech; Wiener Filtering [Lim and Oppenheim, 1979], which applies a linear filter to the noisy speech so that the clean speech mean square estimation error is minimized; the Minimum Mean Square Error (MMSE) [Ephraim and Malah, 1984] estimation of the clean speech power spectrum, which takes advantage from a non linear averaging procedure . The noise estimation becomes difficult when the noise is non stationary, which is the case in most applications. Note that, in these methods, the noise sources are generally assumed independent from speech. Only part of the reverberation effects of these environments can be compensated using Cepstral Mean or Variance Normalization (CMN, CVN) if they are time-invariant or by RASTA

processing if they are slowly time-variant [Hermansky and Morgan, 1994]. On the other hand, speech recorded in environments of the second class can be affected by additive noise at various SNRs, but the major cause of degradation is the channel effect, i.e. the convolutional distortion which is function of the enclosure dimensions, form, material, and of the mutual position of both the speaker and the microphones inside it. In this second class of scenarios, the speaker can be meters away from the microphone and the reflections, which generate an echo effect, are generally seen by the recognizer as highly correlated undesired sources. The high correlation between echo and direct speaker-to-microphone path is what causes failure of additive noise compensation techniques in such environments. Dereverberation can be achieved by inverting a room impulse response [Miyoshi and Kaneda, 1988], or by modifying the LP residual [Yegnanarayana and Murthy, 2000], but more problems arise when one has to separate the two different contributions of the response duration and of the speech duration [Nakatani, 2006]. Furthermore, the channel estimation is very difficult when the speaker is changing position and, mostly, head orientation. The algorithms designed to achieve environmental robustness have no knowledge of the space surrounding the only one microphone used to capture the signal. Regions of space with undesired sources can be physically isolated when using a directional microphone, with the evident drawback of attenuating even the interested sources if the microphone is in a fixed position (for example attached to a wall of a meeting room) and these sources are in the "shadow" regions [Acero, 1993]. Microphone arrays can overcome this drawback, because they intuitively have knowledge of the surrounding space: the human binaural sound capture system is after all a two directional microphone array with a 24 cm inter-microphone distance. Because we are not (yet) aware of the complex algorithms implemented in the powerful parallel machine which is our brain, it is sufficient to note the spatial selectivity of a microphone array is increased by increasing the number of sensors. It is known that for an isotropic noise field, i.e. where disturbances are propagating in every direction, the SNR degradation is proportional to the square of the speaker-to-microphone distance. Being δ_{ct} such distance in close-talk conditions (i.e. from 0.02 to 0.05 meters), if a speaker is located at $N\delta_{ct}$ meters from the microphones the gain loss would be:

$$G = 10log_{10}\left(\frac{\delta_{ct}}{N\delta_{ct}}\right)^2 \tag{1.1}$$

Then, it is known that for a microphone array the ideal gain in function of the number of microphones M is [Elko, 2000]

$$G' = 20log_{10}M \tag{1.2}$$

Thus, solving the equation $G + G' = 0$ it comes out that we need a $M = N$ microphone array size to recover the gain loss and get the same speech quality of a close-talk microphone. Note that from already two meters we would need 100 microphones, which is impractical and costly for many applications. For this reason, research in the last ten years focused on improving the SNR of a microphone array output. Currently, microphones can be placed in very different number and geometries [Flanagan, 1991; H. F. Silverman, 1998]. For in-car applications, few microphones, not necessarily equally spaced, are desired [Grenier, 1992; Yapanel, 2002] while in large rooms one could place from few [McCowan, 2002] to many microphones on a single wall or several distributed microphones [Shimizu, 2000] in the middle of the room or on the walls [Omologo, 2006]. Note that, as we will explain, also the way the voice wave spreads has to be considered when the interested source is near or far from the microphones. In this work we will address the room space because we believe an office, a conference or an auditory room are work environments where speech recognition can significantly help computer-human and human-human interaction and because this space addresses more complex noise robustness issues. As capture device, we will use an array of 8-microphones, which are enough to observe some spatial selectivity, but not too much to be used in practical applications.

1.3 Microphone arrays

The use of a microphone array for distant-talking interaction is based on the potentiality of obtaining a signal of improved quality, compared to the one recorded by a single far microphone [Flanagan, 1991; Omologo, 1998; Brandstein and Ward, 2001]. A microphone array system allows the talker message to be enhanced as well as noise and reverberation components to be mitigated, so that it can be used to achieve a hands-free human-machine voice interaction.

A microphone array consists of a set of acoustic sensors placed at different locations to spatially sample a sound pressure field. Using a microphone array it is possible to selectively pick-up a speech message, while avoiding the undesirable effects due to distance, background noise, room reverberation and competitive sound sources. This objective can be accomplished by means of a spatio-temporal filtering approach.

The directivity of a microphone array can be electronically controlled, without changing the sensor positions or requiring the talker to speak close to the microphones. Moreover, detection, location, tracking, and selective acquisition of an active talker can be performed automatically to improve the intelligibility and quality of a selected speech message in applications such as teleconferencing

and hands-free communication (e.g. car telephony).

1.3.1 Aperture function and Directivity pattern

Voice is a mechanical wave, propagating from the speaker to infinity in every direction. The sound propagation is almost that of spherical waves for sounds, at mostly low frequency, originated by the chest vibration, while it is that of planar waves for the full spectrum sounds coming from mouth and nose [Ziomek, 1995]. When speech is played by an acoustic transducer, such as a loudspeaker, instead, the propagation is more spherical. In this work we assume spherical propagation for the speech signal, which is both a function of time and space. If more microphones are arranged to form a network, regardless of its geometry this network can be considered a sampled version of a continuous transducer: the speech signal $s(t, \mathbf{v})$, which travels across time and space, is received by a sensor located in $\mathbf{v} = [x_v, y_v, z_v]^T$ at the time instant t. The continuous sensor, a.k.a *aperture*, acts like a linear filter [Ziomek, 1995], and the relation between the transmitted and the received signal is:

$$y(t, \mathbf{v}) = a(t, \mathbf{v}) * x(t, \mathbf{v}) \tag{1.3}$$

where a(t,**v**) is the impulse response of the linear filter and y(t,**v**) is the received signal. In the frequency domain (1.3) becomes:

$$Y(f, \mathbf{v}) = A(f, \mathbf{v})X(f, \mathbf{v}) \tag{1.4}$$

where $A(f, \mathbf{v})$ is a quantity known in antenna theory [Gilbert and Morgan, 1955; Walach, 1984] and called *aperture function*: it represents the response of any point of the receiver, located in position **v**. We now need to model the received signal $x(t, \mathbf{v})$. We recall that the mechanical wave observed by the aperture obeys to the Helmholtz equation:

$$\nabla^2 x(t, \mathbf{v}) - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}x(t, \mathbf{v}) = 0 \tag{1.5}$$

where c is the speed of sound, assumed equal to 340 m/s. Under the two assumptions that the signal $x(t, \mathbf{v})$ can be represented with the product (separability) $x(t)x(\mathbf{v})$ and that $x(t)$ is a sinusoid, it can be shown that (1.5) is satisfied by:

$$x(t, \mathbf{v}) = e^{j2\pi f\left(t - \frac{\mathbf{v}}{c}\right)} = x(t)e^{j\frac{2\pi f}{c}\mathbf{v}} \tag{1.6}$$

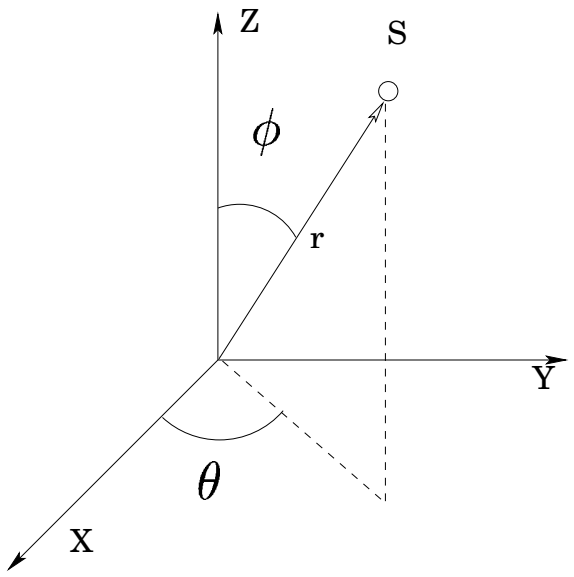


Figure 1.1. Spherical coordinates of the speech source S.

within a scale factor. Equation 1.6 states that a time domain sinusoid traveling in space is observed differently depending on its speed and frequency and on the position of a given point along the aperture. The dependency on both frequency and speed of sound can be compacted in one variable, the *wavenumber*:

$$k = \frac{2\pi f}{c} \tag{1.7}$$

for simplicity of notation and also because c is constant if the environment has a constant temperature [Cramer, 1993]. In the frequency domain, the (1.6) becomes:

$$\mathscr{F}\{x(t, \mathbf{v})\} = X(f)e^{j2\pi\alpha\mathbf{v}} \tag{1.8}$$

where $\mathscr{F}$ denotes Fourier Transform and α is the *normalized wavenumber vector*, so that $\alpha = \frac{k}{2\pi}$. So far we did not model the direction of the sound source with respect to the aperture: let us consider the case in which the geometry of the aperture is linear, i.e. the microphones are put in a row of arbitrary length. A linear aperture is by definition an opening of the acoustic transducer which restricts the size of the bundle of acoustic waves incident to its surface. Thus we need to model the fact that the amount of signal seen by the receiver is as much bigger as the source of interest is in front of the aperture, which can be done by considering the signal in spherical coordinates as shown in Figure 1.1.

The source S is in this way uniquely determined by its distance from a point of the aperture, r, and by its zenith or elevation ϕ and the azimuth θ. The link between the frequency of the incident sound wave frequency, the sound speed and the direction is expressed by the *wavenumber vector* , with $\mathbf{k}$ given by:

$$\mathbf{k} = 2\pi[\alpha_x, \alpha_y, \alpha_z] = \frac{2\pi}{\lambda}[\sin\phi\sin\theta \quad \sin\phi\cos\theta \quad \cos\phi] \tag{1.9}$$

Note that (1.7) is a particular case of (1.9) where $\phi = \theta = \frac{\pi}{2}$. By substituting (1.6) in (1.4), and considering spherical coordinates, the received signal becomes:

$$Y(f, \mathbf{v}) = A(f, \mathbf{v})X(f)e^{j2\pi\boldsymbol{\alpha}\cdot\mathbf{v}} \tag{1.10}$$

The *spatial response* of the aperture is the response to a time-domain impulse signal ($X(f) = 1$). It is obtained by summing all the contribution from the points of the linear aperture, assumed of infinite length. It is thus defined as:

$$S(f, \boldsymbol{\alpha}) = F\{A(f, \mathbf{v})\} = \int_{-\infty}^{+\infty} A(f, \mathbf{v})e^{j2\pi\boldsymbol{\alpha}\cdot\mathbf{v}}d\mathbf{v} \tag{1.11}$$

where $F\{\}$ is the Spatial Fourier Transform (the spatial frequency being represented by $\boldsymbol{\alpha}$). Thus, (1.11) shows the response of the aperture in the spatio-temporal frequency as a function of the direction. Notice that we deal with two transform domains, i.e. the usual time-frequency pair $t \leftrightarrow f$ plus $\mathbf{v} \leftrightarrow \boldsymbol{\alpha}$. Also the exponential term in (1.11) implicitly depends on frequency through λ, since $\lambda = \frac{c}{f}$. Note also that the variables f and $\boldsymbol{\alpha}$ are not independent, as they are linked through the speed of sound. The squared magnitude of the spatial response is the *directivity pattern* or *beampattern*, which can be defined as:

$$B(f, \phi, \theta) = |S(f, \phi, \theta)|^2 \tag{1.12}$$

where the independent variables have been explicited. This quantity is used to evaluate the attenuation in dB at certain angles of the spatial response. Every array-processing method tends to modify the directivity pattern depending on the desired application.

1.3.2 Discrete, linear arrays

Equation 1.11 is valid for a sensor array conceived as a continuous entity of infinite length: this is clearly not true in reality. The array is composed of M *discrete* entities, each having its own

aperture function in the frequency domain $A_m(f, \mathbf{v})$: thus the superposition principle applies and (1.12) becomes, for a discrete array:

$$S_D(f, \boldsymbol{\alpha}) = \sum_{m=0}^{M-1} w_m(f) A_m(f, \mathbf{v}) e^{j2\pi \boldsymbol{\alpha} \mathbf{v_m}} \tag{1.13}$$

where we introduced in (1.13) the complex weights $w_m(f)$, associated with the effect of microphone m. Note the dependency of $\mathbf{v}$ on m, while for the moment $\boldsymbol{\alpha}$ is assumed constant for each microphone: this is because we assume the distance between the speech source and the array much bigger than the array length, so that the angles θ and ϕ do not vary too much from microphone to microphone. While many configurations of microphones can be chosen, in this work we deal with linear arrays, so (1.13) is simplified as done in [McCowan, 2001]:

$$S_{DL}(f, \alpha_x) = \sum_{m=0}^{M-1} w_m(f) A_m(f, x_m) e^{j2\pi \alpha_x x_m} \tag{1.14}$$

where the array spans the x axis. Equation 1.14 can be further simplified as follows:

1. $A_m(f, x_m)$ can be included in the weights if microphones have the same transfer function. This assumption is plausible, since we are using electret (calibrated) microphones [Brayda, 2005a].

2. $x_m = md$ for the m-th microphone if the linear array is equispaced, i.e. d is the inter-microphone distance.

3. $\alpha_x = \frac{cos\theta}{\lambda}$ if the propagation is cylindrically rather than spherically approximated (i.e. the effect of ϕ is negligible).

This results in:

$$S_{DLE}(f, m, d, \theta) = \sum_{m=0}^{M-1} w_m(f) e^{j2\pi \frac{fmd \sin\theta}{c}} \tag{1.15}$$

Equation 1.15 states that the angular response of a discrete, linear and equispaced microphone array is determined by specifying the number of sensors, the microphone spacing and the frequency. The only degree of freedom remains $w_m(f)$: by properly setting the complex weights, the shape of the directivity pattern can be changed. As can be seen in the figures, the beampattern is as narrow as d or f increase: these implies that a very large array is highly selective and spatial aliasing constraints dictate the choice of d in function of the maximum bandwidth of interest.

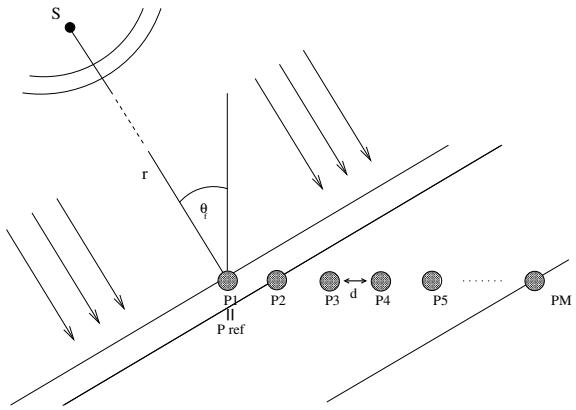


Figure 1.2. Far-field speech propagation

1.3.3 Clean speech multi-channel model

In section 1.3.2 we defined the array as a discrete receiver: the set of transmitted signals reaching the microphones can be thus defined as:

$$\begin{aligned} s_0(t) &= a_0 s(t-\tau_0) \\ s_1(t) &= a_1 s(t-\tau_1) \\ &\vdots \\ s_{M-1}(t) &= a_{M-1} s(t-\tau_{M-1}) \end{aligned} \tag{1.16}$$

where a_m are the attenuations (the amplitude of the signal decreases proportionally to the speaker-microphone distance) and τ_m are the delays that occur at each microphone (unless a plane wave impacts all the microphones in the same time instant, which is rarely the case, each microphone observes a differently delayed version of the original speech signal). The way the acoustic wave impacts the array is important to correctly quantify the delays τ_m: intuitively, if the source is very far from the array, the spherical wave originating from the speaker can be approximated with a plane wave when received by the microphones. In this case the delays can be quantified with the *far field assumption*. Conversely, when the speaker is close, the acoustic wave is still spherical and the *near field assumption* has to be used.

1.3.4 The far-field assumption

A typical configuration where the far-field assumption can be used is depicted in Figure 1.2: the acoustic wave coming from the source S spans the distance r and arrive, almost as a plane wave, at the first microphone *P1*, which is taken as the reference microphone *Pref*. Then it reaches the

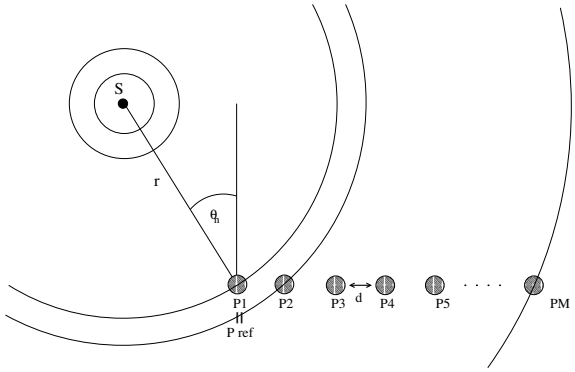


Figure 1.3. Near-field speech propagation

other microphones with a certain delay. Note that the θ_f angle can be considered constant for each microphone. For a linear array of length $L = (M-1)d$, the plain wave signal is considered "far" if the following condition holds [Steinberg, 1976]:

$$\|\mathbf{S} - \mathbf{Pm}\| > \frac{2L^2}{\lambda} \tag{1.17}$$

where $\|\mathbf{S} - \mathbf{Pm}\|$ is the distance between the source and any microphone. We can also assume the attenuation coefficients of equation (1.16 roughly do not change or, more practically, that $a_m = 1$. The delays τ_m in the far field condition as thus computed as [Bitzer and Simmer, 2001]:

$$\tau_m^{far} = \frac{\|\mathbf{S} - \mathbf{Pref}\| - \|\mathbf{S} - \mathbf{Pm}\|}{c} = \frac{(m-1)d\sin\theta_f}{c} \tag{1.18}$$

where $\|\mathbf{S} - \mathbf{Pref}\|$ is the distance between the source and the reference microphone. Note that the far-field condition varies with frequency. A far field assumption is suitable for hands-free speech recognition or speaker localization in a very large room.

1.3.5 The near-field assumption

It is often more realistic not to consider plane waves, but to deal with the true spherical nature of sound propagation. A spherical wave assumption is consistent if the speaker is not far away from the array, i.e.:

$$\|\mathbf{S} - \mathbf{Pm}\| \leq \frac{2L^2}{\lambda} \tag{1.19}$$

Being the source much closer to the sensors, as depicted in 1.3,

the a_m coefficients must be considered as:

$$a_m = \frac{\|\mathbf{S} - \mathbf{Pref}\|}{\|\mathbf{S} - \mathbf{Pm}\|} \tag{1.20}$$

while the delays are evaluated by trigonometry as:

$$\tau_m^{near} = \frac{\|\mathbf{S} - \mathbf{Pref}\| - \|\mathbf{S} - \mathbf{Pm}\|}{c} = \frac{r - \sqrt{r^2 - 2(m-1)rd\sin\theta_n + (m-1)^2 d^2}}{c} \tag{1.21}$$

Alternatively, (1.21) tells us that we can recover the talker direction (not the position) once the delay and the microphone position are known.

Equation 1.18 has got one degree of freedom less than (1.21) but when the task becomes speaker localization, it is intuitive that a speaker far from the array can be located much less easily than a close one.

1.3.6 Is array geometry an issue in Speech Recognition?

Typically the performance of any array processing algorithm increase linearly or asymptotically with M, i.e. the number of microphones. One could be tempted to study the system behavior for $M \to \infty$, but in practice the number of microphones is determined by

- The application
- Hardware complexity
- Cost

As mentioned, in-car experiments [Yapanel, 2002; Grenier, 1992] cannot make use of more than 5-8 microphones because of limited room. For the same reasons, the inter-microphone distance cannot be high. These constraints do not apply in large rooms. If the in-car array has then to provide a noise cancellation for a hands-free GSM cellphone, then the real-time constraints impose low hardware complexity, the same constraints which stand during an audio-conference in a room. For off-line applications, as using Speech Recognition to store the entire text of a meeting, this constraint does not hold. Last but not least, the high quality, expensive microphones used to test algorithms in research are not necessarily the ones used for commercial purposes, where all processing is being pushed into DSPs and where the only analogue part is left to cheaper microphones. For Speech Recognition purposes, then, array processing may be of little use: though [Giuliani, 1997] shows that linear and harmonic nested arrays are generally useful, still in [Brandstein and Ward, 2001] it is emphasized that benefits coming from a specifically designed configuration of microphone arrays

can be canceled out. This fact comes from artificial signals generated during the simulations, from the several approximations which occur in the front-end, and finally from the adaptation to acoustic modeling. We conclude that a little effort has to be put in array geometry for ASR, except to avoid the *spatial aliasing*.

1.3.7 Spatial aliasing

The Nyquist theorem fixes the minimum sampling frequency to avoid aliasing in the time domain to

$$f_s \geq 2f_{max} \tag{1.22}$$

where f_{max} is the maximum frequency of the signal to be sampled at frequency f_s. This is also valid in the space domain. Each microphone samples the speech signal and, in general, each array of antennas has to respect the spatial sampling theorem:

$$f_x \geq 2f_{x,max} \tag{1.23}$$

where $f_{x,max}$is the maximum spatial frequency in samples per meter of the signal spatially sampled at $f_x = \frac{1}{d}$. If the signal is represented in spherical coordinates, then the direction of propagation of the signal is described by the wavenumber vector of Equation 1.9. If we assume the array to be linear, in the x direction, then the wavenumber vector reduces to the scalar:

$$\alpha_x = \frac{\sin\theta \sin\phi}{\lambda} \tag{1.24}$$

This number represents the spacial frequency of the signal, whose maximum is $\frac{1}{\lambda}$. By recalling that $f_x = \frac{1}{d}$, Equation 1.23 becomes:

$$d \leq \frac{\lambda}{2} \tag{1.25}$$

Violating this condition will result in generating grating lobes in the directivity pattern. This can be intuitively seen in Figure 1.4 where two situations in which $d = \frac{\lambda}{2}$ and $d = \lambda$ are depicted: in figure a) two waves from broadside (i.e. along the y axis) arrive at the two microphones after two periods, i.e. after having spanned twice their wavelengths in space. The reception of the signal is correct. Then, from *endfire* (i.e. along the x axis) another signal spans two wavelengths before being captured by $m1$ and almost two before being captured by $m2$ (the values detected are indicated by the two dashed arrows). The reception is correct. Conversely, in figure b) the inter-microphone

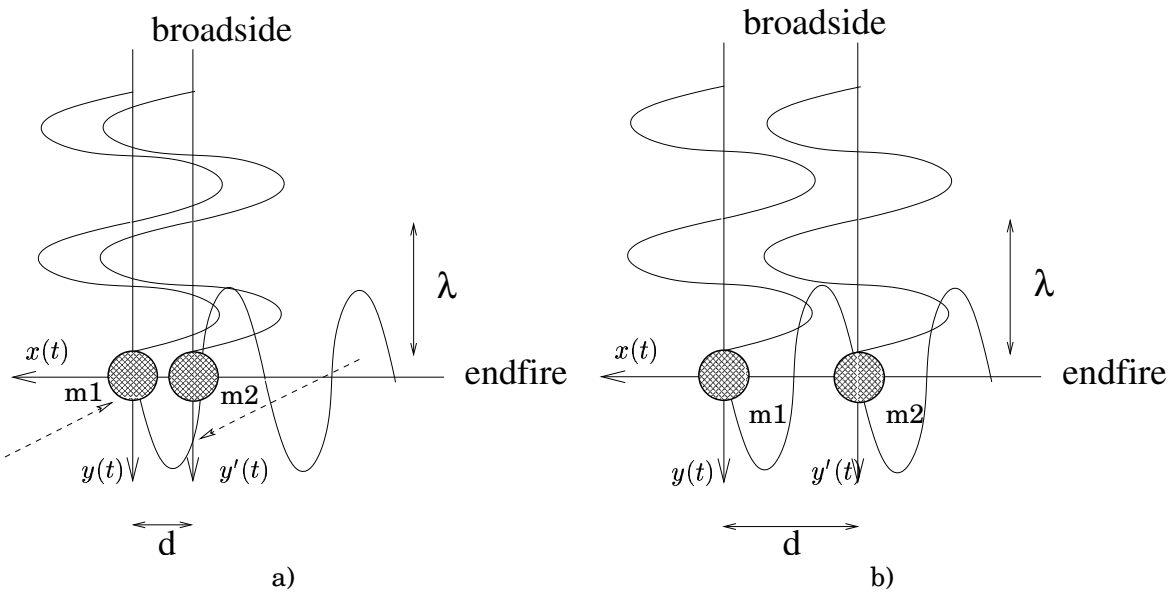


Figure 1.4. Example of spatial non-aliasing ($d = \frac{\lambda}{2}$) in a) and aliasing ($d = \lambda$) in b). In both configurations the wave coming from broadside is correctly recognized, but the one coming from the end-fire direction is correctly seen as in a) (its values are indicated by dashed arrows), while it is indistinguishable from the one coming from the end-fire position in b).

distance equals the signal wavelength. The signals coming from broadside are correctly received, but the microphones cannot detect the difference between a sinusoid and an all-zero signal for the source located in endfire position. This implies that, by fixing d and λ, spatial aliasing tends to be heavier for increasing θ angles: the more the source is lateral with respect to the array, the more the signal risks to be wrongly reconstructed. For speech purposes, where we can say that the maximum frequency of interest is around 8 kHz, then d should be less than $\frac{c}{2f} = 2,12cm$. In practice array with a larger d are used, because the aliasing condition concerns only sources in end-fire position, while in many applications some *a priori* knowledge of the source of interest position is available (for example it is roughly in front of the array).

1.4 Noise fields

Depending on the environment, the noises $n_m(t)$ are differently characterized. One way of categorizing noise fields is to measure the inter-microphone correlation, which is given by the *coherence function* [Carter, 1993]:

$$\Gamma_{n_1,n_2}(f) = \frac{P_{n_1,n_2}(f)}{\sqrt{P_{n_1,n_1}(f)P_{n_2,n_2}(f)}} \tag{1.26}$$

where $P_{n_1,n_2}(f)$ is the cross-spectrum between noise signals measured at microphones m_1 and m_2. This quantity is ≤ 1 for the Schwartz inequality and indicates the degree of correlation of two microphone signals.

1.4.1 Non coherent noise

In a *non coherent* noise field, noises measured in different points are uncorrelated. Microphone electrical noises are non-coherent, because they are randomly distributed across microphones. It is the rarest form of real noise which can be measured in a real environment and it is the easiest to compensate, because the destructive interference between uncorrelated signals can be exploited. Thus for these noises $\Gamma_{n_1,n_2}(f) \approx 0$ for every microphone pair.

1.4.2 Coherent noise

Coherent noise fields measured at different microphones are strongly correlated (i.e. $\Gamma_{n_1,n_2}(f) \approx 1$ for every microphone pair). Coherent noise corresponds to a competitive speaker situation or to computer, fan or air-conditioning noise. By definition coherent noises are not reflected by any surface, because this would increase the amount of signal scattering due to multipath, leading the coherence function to decrease. However, in a room this scattering does exist. Since coherent noises generally come from a specific direction, a microphone array can be robust to them if the weights $w_m(f)$ of (1.15) are modified in order to attenuate any signal coming from that specific direction.

1.4.3 Diffuse noise

The *diffuse* noise field (also known as homogeneous or isotropic) is the most encountered in a room or in a car: noise is propagating in any direction and cross-correlation is generally lower for far microphones and higher for near microphones. The coherence function of diffuse noise fields can be modeled as:

$$\Gamma_{n_1,n_2}(f)|_{diffuse} = \frac{sin(2\pi f d_{12}/c)}{2\pi f d_{12}/c} \tag{1.27}$$

where d_{12} is the distance between m_1 and m_2. Thus the Γ function approaches the non-coherent behavior when d_{12} is increasing and the coherent behavior when decreasing. Interestingly, a diffuse noise reaches each microphone with equal energy.

1.5 Room impulse responses (IRs)

In Equation 1.16 we modeled the audio waveform in such a way that the far-field and near-field scenarios can be easily described. However, especially in enclosures such as meeting rooms, the situation is much more complex: the propagating wave is reflected by walls, thus many delayed and attenuated versions of the same original wave are captured by a microphone. Because additive noise and reflections are two different phenomena, we consider them separately, and assuming no additive noise is present, we can rewrite (1.16) as:

$$\begin{aligned} x_0(t) &= s(t) * h_0(t) \\ x_1(t) &= s(t) * h_1(t) \\ &\vdots \\ x_{M-1}(t) &= s(t) * h_{M-1} \end{aligned} \tag{1.28}$$

where $h_m(t)$ is the impulse response from the speaker to microphone m. Several observations follow:

- The convolution operator with a filter h_m models by construction the sum of the multipaths. Reverberation is roughly a Linear Time Invariant (LTI) system and it is thus characterized by its Impulse Response (IR).
- There is one single source of speech, but there are M impulse responses, one for each microphone. This is because the observed multipath is different for each microphone, and the further apart the microphones are, the more different the responses.
- The effects of attenuation a_m and delay τ_m are now included in each h_m.

Figure 1.5 depicts a typical highly reverberant room impulse response. From left to right, we can note that the shape of the response is almost causal with respect to the the main peak, which corresponds to the direct path. A very strong peak is observable just after it and a lower one is present after 60 ms. The three peaks, called *early reflections*, are likely to be perceived as one by a human listener, because of the *precedence effect* [Haaso, 1972]. However, smaller peaks and the long tail, which is the proper *reverberation*, is clearly acoustically distinguishable. The depicted impulse response has been measured by means of a “chirp-like” (swept sinusoid) signal [Matassoni, 2002], designed so that its autocorrelation is a quasi perfect Dirac impulse function. Thus, by playing the chirp signal with a speaker in the desired location and recording it with a microphone, one can cross-correlate the acquired signal with the original input and get the speaker-to-microphone

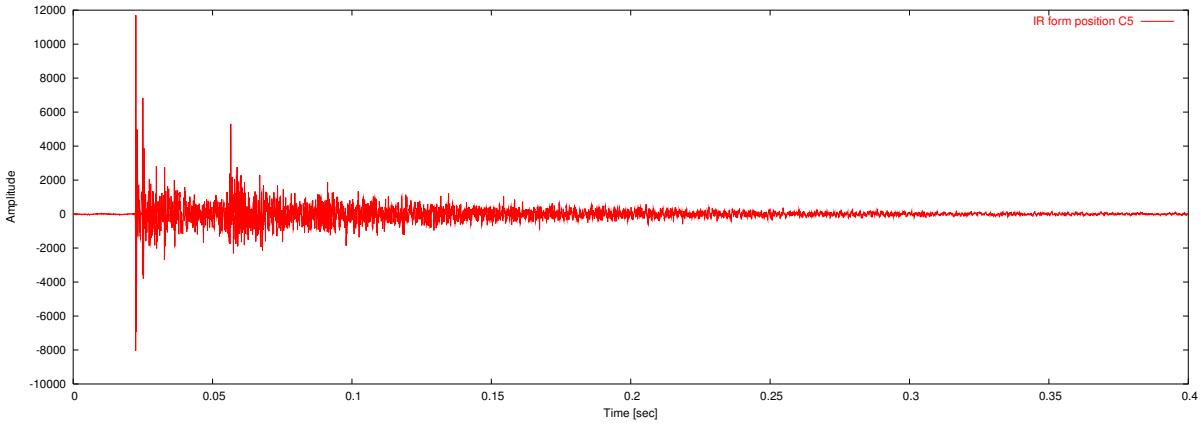


Figure 1.5. Room impulse response of a very reverberant room (used in this work). The main peak corresponds to the the direct path, while secondary peaks are the main, early reflections. A long tail represents the reverberation, generally the most audible phenomenon

impulse response. Because room IRs have exponential decay, there is a threshold after which the signal is considered negligible: the *reverberation time* or T60 is the time lag between the main peak and the instant the signal energy decays of more than 60 dB below the energy of the main peak. The depicted IR, for example, has T60=700 ms, which is very high. Room impulse responses can also be automatically generated with the image method [Allen and Berkley, 1979] by simulating a room, the source and the receiver position, and the T60. Compensating for reverberation is more than an issue for speech enhancement and recognition: IRs are in general non-minimum phase [Neely and Allen, 1979], thus very rarely one can find an inverse filter. Several solutions are present in the literature for speech dereverberation, but speech recognition is particularly sensitive to the way speech is deconvolved. CMN is able to compensate for early reflections (which are caused by thr main IR peaks), but the main problem for speech recognition is the late reverberation (coming from the convolution with the almost scattered IR tail). This work will also address the reverberation problem in Chapter 4.

1.6 Modeling reverberation and multiple noise sources

In real world applications each microphone receives a modified version of the speech signal, due to reverberation effects, plus an additive noise component. Equation 1.28 can thus be updated as:

$$\begin{aligned} x_0(t) &= s(t) * h_0(t) + n_0(t) \\ x_1(t) &= s(t) * h_1(t) + n_1(t) \\ &\vdots \\ x_{M-1}(t) &= s(t) * h_{M-1}(t) + n_{M-1}(t) \end{aligned} \tag{1.29}$$

where $n_m(t)$ is the noise source received by microphone m. More rigorously, we should define each $h_m(t)$ as:

$$h_m(t) = h_m^R(t) * h_m^P(t) \tag{1.30}$$

to emphasize that the channel effect is modeled by two parts: $h_m^R(t)$ is the room IR, while $h_m^P(t)$ is the impulse response proper to the microphone, regardless of the environment. In practice it is very difficult to separate the two effects, so we consider them as a whole. As the environmental conditions becomes more severe, the impact of $h_m^P(t)$ becomes lower and lower in performances. Note also that the M additive noise components can come from one to many noise sources. In the case of one source of noise, what is observed at each microphone still varies from microphone to microphone, because noise is affected by reverberation as well. In the case of multiple noise sources (NS), each microphone records the sum of the effects of all these sources. The m-th microphone would then record:

$$x_m(t) = s(t) * h_m(t) + \sum_{j=0}^{NS-1} n_j(t) * h_{jm}(t) \tag{1.31}$$

The more realistic problem would be even worse, because for a moving speaker the $h_m(t)$ impulse response would change over time. Furthermore, the application may be interested in more than one speech source, thus $s(t)$ would be just one of the possible component of a very complicated signal. In this work we recognize speech from a single source of interest and we consider several scenarios, with both additive noise (represented by one and more surrounding noise sources) and reverberation effects, so we use the model described by Equation 1.29.

1.7 Array-based Speech Enhancement

A microphone array can be used to recover the original clean speech signal $s(t)$ by properly processing the multi-channel signal of Equation 1.29. We can compactly represent this signal in the frequency domain as:

$$\mathbf{X}(f) = \mathbf{H}(f)S(f)\mathbf{d} + \mathbf{N}(f) \tag{1.32}$$

where $\mathbf{X}(f)$ is the vector of the Fourier Transform of the different microphone inputs, $\mathbf{N}(f)$ is an additive noise vector field, $\mathbf{H}(f)$ is the diagonal matrix of the different speaker-to-microphones frequency responses and $\mathbf{d}$ is the propagation vector, which includes information about the microphones delays as defined in (1.18) and the attenuation coefficients:

$$\mathbf{d} = [a_0 e^{-j2\pi f \tau_0},\ a_1 e^{-j2\pi f \tau_1}, \cdots,\ a_{M-1} e^{-j2\pi f \tau_{M-1}}]^T \tag{1.33}$$

The purpose of Speech Enhancement is to increase the SNR of a noisy speech. Intuitively, this increases the recognition rate, but, as pointed out in [Brandstein and Ward, 2001], there is no direct relationship between the SNR, which is an objectively-evaluated measure, and the performance of a speech recognizer. However, most of the literature concerning speech recognition with microphone arrays considers performing speech enhancement on the multi-channel signal first and feeding then the recognizer with a single-channel enhanced signal. This is the reason why the main speech enhancement methods are worth to be explored.

1.7.1 Beamforming

Beamforming is a technique derived from the antenna theory [Venn and Buckley, 1988]. It aims at forming a *beam* toward the source of interest, using the microphone array. Undesired noise sources are in this way attenuated. Beamforming principally addresses additive noise. Different techniques exist, and some are more suitable for specific noise fields (non-coherent, coherent, diffuse). However, with the respect to the radar or sonar counterparts, application of beamforming in speech is different for the following reasons [Compernolle, 1990]:

1. The speech source position relative to the array is rarely fixed.
2. The SNR is generally positive (for negative values the Lombard effect occurs [J-C. Junqua, 1996]), except for reverberation, where the power of the scattered signal is even higher than power of the non-reflected signal.
3. Speech is not narrow-band and disturbances can have the same spectral content as the interesting source.
4. The speech spectral content is changing over time, sometimes frequently, and silence periods occur.

All these issues have to be considered when designing a beamformer for speech processing. Nevertheless, the constraints for optimal beamformer design can be more severe or relaxed if speech enhancement only has to be performed rather than recognition: this topic will be discussed in Section 1.9, but the main point is that human listeners are more robust to non-stationary noises and signal distortion or cancellation compared to the best automatic speech recognizer. Regardless of the application, the general model of a beamformer is:

$$Y(f) = \mathbf{W}(f)^H \mathbf{X}(f). \tag{1.34}$$

where $\mathbf{W}(f)$ is the vector of the frequency-domain microphone weights defined in (1.13). If reverberation effects are neglected, which implies $\mathbf{H}(f) = \mathbf{1}$, then (1.34) can be expanded as:

$$Y(f) = \mathbf{W}(f)^H \mathbf{d}(f) S(f) + \mathbf{W}(f)^H \mathbf{N}(f). \tag{1.35}$$

Beamformers tend to maximize the so-called *array gain*, which is the ratio of the SNR achieved by the array with respect to the SNR achieved by a single microphone:

$$G_a(f) = \frac{SNR_{array}(f)}{SNR_{microphone}(f)} \tag{1.36}$$

If both speech and noise are stationary, the denominator of (1.36) is:

$$SNR_{microphone}(f) = \frac{P_{SS}(f)}{P_{\bar{N}\bar{N}}(f)} \tag{1.37}$$

where P_{SS} is the clean speech power spectrum and $P_{\bar{N}\bar{N}}$ is the average noise power spectrum. The average spectrum is taken because the noise power from a single microphone can be assumed to be represented as the average noise power along the microphone array. The numerator in (1.36) can be given by the ratio of the array output power spectrum when clean speech only is present ($\mathbf{N}(f)$ set to zero) and the array output power spectrum when noise only is present ($S(f)$ set to zero):

$$SNR_{array}(f) = \frac{P_{YY}|_{speechonly}}{P_{YY}|_{noiseonly}} = \frac{|\mathbf{W^H d}|^2 P_{SS}}{\mathbf{W^H P_{NN} W}} \tag{1.38}$$

where the dependency on frequency has been neglected for practical notation. Because we would like to express (1.38) in terms of (1.37), which would make the array gain dimensionless, we can factorize $\mathbf{P_{NN}}$ by carrying out the average noise power $P_{\bar{N}\bar{N}}$. This leads to:

$$SNR_{array}(f) = \frac{|\mathbf{W^H d}|^2 P_{SS}}{\mathbf{W^H P_{N'N'} W} P_{\tilde{N}\tilde{N}}} \tag{1.39}$$

and the array gain becomes:

$$G = \frac{|\mathbf{W^H d}|^2}{\mathbf{W^H P_{N'N'} W}} \tag{1.40}$$

Array based speech Enhancement aims at finding:

$$\tilde{\mathbf{W}} = \arg\max_{\mathbf{W}} G(\mathbf{W}, \mathbf{N}, \theta) \tag{1.41}$$

The different ways of fixing the $\mathbf{W}$ weights makes the distinction between *delay and sum*, *filter and sum* and *super-directive* beamformers.

Alternatively, the array gain can be defined as [Cray and Nuttall, 2001; McCowan, 2001]:

$$G_a[f, \theta_0, \phi_0] = \frac{D[f, \theta_\mathbf{0}, \phi_\mathbf{0}]}{\frac{1}{4\pi}\int_0^{2\pi}\int_0^{\pi} D[f, \theta, \phi] \sin\theta d\theta d\phi} \tag{1.42}$$

This definition applies if the noise field is isotropic (see (1.4.1)) and measures the ability of the array of suppressing unwanted noise, knowing an observation direction (θ_0, ϕ_0) along which the desired source is located.

Delay and Sum Beamforming

Delay and Sum (D&S) beamforming aims at recovering in Equation 1.35 the clean speech signal $S(f)$ by exploiting the destructive interference of the M components of noise $\mathbf{N}(f)$. At the same time, one wants the original clean speech signal $S(f)$ to be undistorted. This is accomplished by imposing the constraint:

$$\mathbf{W^H d} = 1. \tag{1.43}$$

which prevents the weight to modify the clean speech signal. Thus, a solution to the linearly constrained system is:

$$\mathbf{W} = \frac{1}{M}\mathbf{d}. \tag{1.44}$$

In this way the weights act only on the second term of (1.35), which related to noise only. We recall that for a discrete, linear equispaced microphone array the spatial response is (1.15)

$$S_{DLE}(f,m,d,\theta) = \sum_{m=0}^{M-1} w_m(f) e^{j2\pi \frac{fmd\sin\theta}{c}} \tag{1.45}$$

If the delays in (1.44) are expanded as:

$$\tau_m = \frac{md\sin\theta'}{c} \tag{1.46}$$

then the expression of the linear constraint given by (1.44) and (1.33) is substituted in (1.45) and the spatial response becomes:

$$S_{D\&S}(f,m,d,\theta) = \sum_{m=0}^{M-1} \frac{1}{M} e^{j2\pi \frac{fmd(\sin\theta - \sin\theta')}{c}} \tag{1.47}$$

We recall that the directivity pattern derived from (1.45), i.e. its square modulus, has a main lobe around its maximum, which is $\theta = \frac{\pi}{2}$. (1.47) states that if a set of delays τ_m is applied to the multi-channel signal, then the main lobe of the directivity pattern moves to $\theta = \theta'$. This means that we can "steer" the array with a given set of delays, which can be linked to a desired "look direction" θ'. The look direction, a.k.a Direction Of Arrival (DOA), which corresponds to the speaker position, is generally unknown and several techniques exist to estimate it [Omologo, 2006] via Time Delay Estimation (TDE). The technique used in this work is based on the Cross-Power Spectrum Phase (CSP), a.k.a Generalized Cross Correlation Phase Transform (GCC-PHAT), and it is detailed in Appendix A. If the noise field is diffuse, we can express the array gain in function of the noise coherence function defined in (1.26) as [Cox, 1987]:

$$G = \frac{|\mathbf{W^H d}|^2}{\mathbf{W^H \Gamma_{N,N} W}} \tag{1.48}$$

where $\mathbf{\Gamma_{N,N}}$ is the correlation matrix, the elements of which are the coherence functions $\Gamma_{N_{n_i},N_{n_j}}$ measured at microphones i and j. If noises at the microphones are totally uncorrelated, the in-diagonal elements are coherent noise, while off-diagonal elements are non-coherent noise, thus leading to $\Gamma = I$. In this case the array gain is also known as *White Noise Gain* (WNG) and is given by:

$$G_{a,WNG} = \frac{|\mathbf{W^H d}|^2}{\mathbf{W^H W}} \tag{1.49}$$

The Delay and Sum defined in (1.44) are the optimal solution for $G_{a,WNG}$ to be maximized: substituting 1.44 into (1.49) leads the gain equal to M. This complies with (1.2 when measuring in dB.

D&S is not designed to compensate for reverberation effects, which would mean including the $\mathbf{H}(f)$ vector in the derivation of the optimal weights. Intuitively, only the part of the reverberation effect which is uncorrelated from microphone to microphone can be compensated, because destructive interference would occur when summing up reverberation effects, exactly as it happens for the additive noise case [Allen, 1977]. Because the array gain D&S is directly proportional to the number of microphones, one simple solution is to use very large arrays, as it is done in [Flanagan, 1985]. This can be suitable for applications in large rooms, but not desirable in small offices, for desktop applications or in the car.

Filter and Sum Beamforming

The weights $\mathbf{W}(\mathbf{f})$ of a D&S beamformer are a particular solution of a more general form, in which every component of the weight vector can be represented as:

$$w_n(f) = b(f)e^{j\phi(f)} \tag{1.50}$$

where both the real and the imaginary part depend on frequency. In the D&S case $b = \frac{1}{N}$ does not depend on frequency and $\phi(f) = \frac{-2\pi m df \sin\theta'}{c}$ is linear with frequency. In *Filter and Sum* beamformers the weights can have for example shape of IIR or FIR filters. In this work we consider FIR filters because they can model more easily a room IR or its approximate inverse filter.

Super-directive Beamforming

If the noise field is diffuse rather than non coherent, the array gain can be maximized by solving (1.41) and finding the optimal weights. The constraint is to get an undistorted signal of interest in the look direction, so the D&S constraint of (1.43) still holds and the solution can be calculated using Lagrange multipliers [Cox, 1987]:

$$\tilde{\mathbf{W}} = \alpha\Gamma_{NN}^{-1}d \tag{1.51}$$

If α is chosen to fulfill the linear constraint of Equation 1.43 then we get what is called the Minimum Variance Distortionless Response (MVDR) [Frost, 1972]:

$$\tilde{\mathbf{W}}_{\mathbf{MVDR}} = \frac{{\mathbf{\Gamma_{NN}}}^{-1}\mathbf{d}}{\mathbf{d}^H{\mathbf{\Gamma_{NN}}}^{-1}\mathbf{d}} \tag{1.52}$$

where the coherence function Γ_{NN} can substitute the normalized noise power spectrum $P_{N'N'}$.

In this case the array gain is:

$$G_{MVDR} = \mathbf{d}^H \mathbf{\Gamma_{NN}}^{-1} \mathbf{d} \tag{1.53}$$

The computed weights in (1.52) are able to efficiently cancel an interferer coming from a specific direction: the beamformer is thus called *Superdirective* (SB). Note that the array gain reduces to the WNG if $\mathbf{\Gamma} = \mathbf{I}$, as it should be. Thus, super-gain tends to the D&S performance if noise are uncorrelated, i.e. D&S is the optimal solution for non-coherent noise fields, while SB is the optimal solution for more diffuse noise fields (in a Minimum Variance sense). However, for this method to properly work, a number of conditions have do be satisfied. As stated in [Bitzer, 1999], SB requires infinite precision of the sensors, i.e. microphones must have the same frequency response and no internal noise is allowed, which is rarely the case for real applications. The authors propose to take into account microphone noise variances in the coherence function computation. They also assume the direction of a main interferer is known. This requires exact interferer localization; small errors in direction estimation can make the direction of the source of interest fall outside the main lobe of the optimized beampattern. Apart from the noise field to be diffuse, which is not always the case in real environments, the SB requires the speech source of interest to be in endfire position as much as possible: this requirement is not practical for horizontally wall-mounted microphone arrays, as the one used in this work, which have to be able to beam toward speakers spread all over the π space in front of the array. SB is effective for canceling the direct path of a main interferer whose direction is known and when no or low reverberation is present, because the reflections remain unattenuated. Recent work [Bitzer and Simmer, 2001] showed that SB has the drawback of boosting low frequency uncorrelated noise: this may cause problems for a speech recognizer processing the beamformed signal, especially if its front-end has a high resolution in the low speech bands (this is the case with Mel-Filterbank based front-ends). Concerning the coherence function, even if authors do not agree in general about its usefulness [Bitzer, 1999; Chu, 1997; Doerbecker, 1997], it has been shown [Knapp and Carter, 1976; Omologo and Svaizer, 1997] that it can be at least very useful to derive, in a modified form, the delays τ_m of the vector d which are used to time-align the multi-channel signal. In reverberant environments, the filters can be set to model the room acoustic. Because a high reverberation time implies room impulse response are many taps long, a filter compensating for this effect can be as much long. This would imply issues in computational complexity for a beamformer adaptive filter, as it will be explained in Chapter 4.

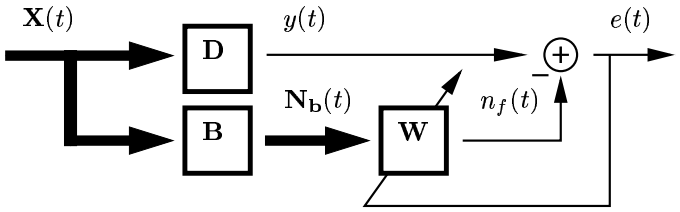


Figure 1.6. Generalized Sidelobe Canceler

1.7.2 Generalized Sidelobe Canceler (GSC)

A method that combines noise compensation with array speech enhancement is the Generalized Sidelobe Canceler, proposed in [Griffith and Jim, 1982] and depicted in figure 1.6.

The multi-channel signal $\mathbf{X}(t)$ is given to a beamformer $\mathbf{D}$, which produces a single enhanced channel $y(t)$. If the beamformer is the conventional D&S, we have $\mathbf{D} = \frac{1}{M}\mathbf{d}$ as stated by (1.44) In parallel, $\mathbf{X}(t)$ is given to a blocking matrix $\mathbf{B}$, designed to filter out the desired speech signal. The blocked, noise-only channels $\mathbf{N_b}(t)$ are then filtered by means of a set of filters $\mathbf{W}$ which replicates and sums the multi-channel noise. Finally the noise sum $n_f(t)$ is subtracted from the enhanced speech and the output $e(t)$ can be used to adaptively drive the $\mathbf{W}$ filters to minimize the output power. The system derives its name from the fact that it effectively cancels out everything which does not come from the look direction determined by the $\mathbf{D}$ block and can be considered as an adaptive noise canceler with multiple reference signals [Widrow and Stearns, 1985]. The $\mathbf{B}$ matrix is generally designed to simply generate the differences between adjacent channels. A similar algorithm exist in the time domain, proposed in [Frost, 1972], which indeed minimizes a constrained LMS problem. Some authors [Kaneda and Ohga, 1986] judge the Frost constraint too rigid and propose the AMNOR algorithm, where a more relaxed constraint for the filter adaptation is used. However, the looking direction is assumed to be very precise: it has been shown that GSC is very sensitive to steering errors [Cox, 1987; Walach, 1984]. Furthermore, as stated in [Omologo, 2001; Bitzer, 1999], a GSC can provide limited improvements in reverberant environments when its output is given to a speech recognizer: the $\mathbf{W}$ block generates replicas of signals which have the same statistics of the residual noise signal. Whatever is not in the look direction is assumed to be noise. However, the multipath caused by reverberations makes desired speech fall outside the look direction. Thus, it is automatically considered as noise, it is not blocked by the $\mathbf{B}$ and finally it is subtracted from the output of the D&S, inevitably causing signal distortion. This distortion can still be tolerable from a speech enhancer, but may be not acceptable for a speech recognizer. For speech enhancement only, it has been proposed [Nordholm, 1993] to inject some artificial noise before the adaptive filtering stage, which would mask the desired signal up to a certain threshold and

would prevent the desired signal distortion.

1.7.3 Post filtering

The MVDR solution of (1.52) maximizes the array gain, which is shown in [Cox, 1987] to be equivalent to minimizing the array output power. The MVDR solution determines optimal weights for a diffuse noise field. However, for other kind of noise fields, this is not necessarily the optimal solution. Instead of minimizing the power, the Minimum Mean Square Error (MMSE) measure can be minimized. If the multi-channel input $\mathbf{X}$ is defined as in Equation 1.29 the MMSE criterion consists in finding the weights:

$$\tilde{\mathbf{W}}_{\mathbf{opt}} = \arg\min_{\mathbf{W}} E[\{s - \mathbf{W}^H\mathbf{X}\}\{s^* - \mathbf{X}^H\mathbf{W}\}] \tag{1.54}$$

and its general optimal solution is given by the Wiener-Hopf equations and constitutes the *multi-channel Wiener Filter*:

$$\tilde{\mathbf{W}}_{opt} = \mathbf{\Phi}_{xx}^{-1}\boldsymbol{\phi}_{xs} \tag{1.55}$$

where $\mathbf{\Phi}_{xx}$ is the correlation matrix between the microphone inputs and $\boldsymbol{\phi}_{xs}$ is the cross-correlation vector between the microphone inputs and the desired clean speech. If we now consider just one single speech source and we neglect the reverberation effects, as it is generally done when studying the MVDR, then we can substitute (1.32) in (1.55) and get, as explained in [Simmer, 2001]:

$$\tilde{\mathbf{W}}_{opt} = \frac{P_{YY}|_{speech}}{P_{YY}|_{speech} + P_{YY}|_{noise}} \frac{\mathbf{\Phi_{NN}^{-1}}\mathbf{d}}{\mathbf{d^H}\mathbf{\Phi_{NN}^{-1}}\mathbf{d}} \tag{1.56}$$

where $P_{YY}|_{speech}$ and $P_{YY}|_{noise}$ were already used in (1.7.1) and are the array output when speech only or noise only respectively are present. Equation (1.56) states that the weights minimizing the MMSE criterion are the product the MVDR weights times a single-channel (scalar) Wiener post-filter which is a function of the SNR measurable at the beamformer output. A number of techniques have been proposed to correctly estimate the Wiener filter, based on the array input and output, which are well described in [Marro, 1998]. In general, post-filters perform as good as the multi-channel noises are uncorrelated and the time delay compensation performed by a D&S is correct. However, some authors report that they introduce artifacts in reverberant environments [Omologo, 2001], which can decrease performances of speech recognizers.

1.7.4 Matched Filtering

The main drawback of a D&S beamformer is that it does not address reverberation. By steering the array to the interested source, even with perfect TDE, the constructive interference is done on the main, direct path. All the main reflections are considered as disturbances. For an array with M microphones, installed in a room where a signal has one direct path and K reflections, the Signal-to-Reflection-Ratio (SRR) of a D&S system can be represented as [Flanagan, 1993]:

$$SRR_{D\&S} = \frac{M^2}{M(K-1)} = \frac{M}{K-1} \tag{1.57}$$

This is because the time-aligned direct path signals add in amplitude, while time-spread reflections add in power. Clearly, though the SRR is M times superior with respect to the one microphone case, it decreases monotonically as K increases, which is the case in large, very reflecting rooms. Moreover, whether the beamformer be conventional or superdirective, still there are some reflections which enters in the main beam, so that the echo effect can't be canceled.
Matched Filtering, a well-known method in antenna theory [Cook and Bernfeld, 1993] was proposed to be applied for spatially selective speech acquisition and beamforming in [Flanagan, 1991]. It consists in beamforming on the direct path and on the main reflections by constructively cumulating the effect of the reflections (without hoping to suppress them, as done via simple beamforming). Matched Filtering can be seen as a Filter-and-Sum beamformer, where, each filter, applied to each microphone, is the time-reversal of the impulse response from the source of interest to that microphone. For a given speaker-to-microphone impulse response $h_m(t)$, then the matched filter is:

$$w_m(t) = h_m(-t) = F^{-1}(H_m^*(f)) \tag{1.58}$$

where conjugation forms a Fourier pair with time reversal because $h_m(t)$ is real. Considering (1.28), for an additive noise free environment the resulting beamformer can be represented as:

$$y(t) = \sum_{m=0}^{M-1} s(t) * h_m(t) * h_m(-t) \tag{1.59}$$

The effect of filtering with the time reversal of the impulse response implies that the quantity $h_m(t) * h_m(-t)$ tends to be very similar to a delta in the time domain, centered in the $L-th$ sample, where L is the response length. Because all the reflections have contributed, through convolution, to the formation of this delta, there will be as much M contributions as there are reflections, thus the delta will be KM high. On the other hand, outside the big delta, the residual reflections will be

$K(K-1)$ for each microphone, thus $MK(K-1)$ in total. The resulting $SRR_{matched}$ is:

$$SRR_{matched} = \frac{(KM)^2}{MK(K-1)} = \frac{KM}{K-1} \tag{1.60}$$

By comparing Equations (1.60) to the D&S case 1.57, when $K \to \infty$ the SRR tends to M and this is independent on the amount of reflections. The use of the time reversal is justified by some considerations about the invertibility of a room impulse response, even for one microphone: room impulse responses can be modeled by FIR filters, but, as outlined in Section 1.5, they are in general non-minimum phase. This means that some zeros of their z-transform may be outside the unit circle (the region of the Argan-Gauss plane where $|z| = 1$). In order to invert such response a non-causal signal is needed [Oppenheim and Shafer, 1999]: in fact the zeros of the direct system would become poles of the inverse system; the zeros which are outside the unit circle will be poles outside the unit circle; since for the inverse system to be stable, the Region Of Convergence (ROC) must include the unit circle, then the ROC will not be outside the outermost pole, which is the condition for causality. Thus the inverse system will not be causal. Of course in practice the inverse system will be causal with a proper delay. Matched filtering was tested in a real environment as well [E. E. Jan and Flanagan, 1995]: in this case room impulse responses were measured with pseudo-random sequences [F. J. MacWilliams, 1980] (however we will use a more efficient technique in this work) from which filters were created. The experiments are interesting in the sense that a remarkable enhancement is obtained for a speaker-to-microphones distance of 3 meters. However a high sensitivity to the focal point of the beamformer was experienced, which means that if the source moves from the position which led to the impulse response measurement, enhancement is not guaranteed and becomes unpredictable. Other researchers [Affes and Grenier, 1997] combined successfully MF with GSC for Speech Enhancement (in this case a constrained identification of the room IR was also possible), though this has not been tried for speech recognition purposes. Matched filtering has been employed successfully for Speaker Identification Problems [Jan and Flanagan, 1995; Lin, 1994]. To our best knowledge, there is no method which performs better than Matched Filtering for speech enhancement (and fixed speech source) purposes.

1.7.5 Adaptive Sub-space Filtering

GSC and Matched Filtering address two different problems: the first technique provides additive noise reduction, but does not deal with reverberation, which entails non-blocked speech; the second compensates the multipath without any *a priori* knowledge of the speech or noise frequency content.

It has been proposed to fuse these technique in a Subspace Tracking algorithm [Affes and Grenier, 1997], which should take advantage from both of them. The technique works as outlined below:

1. An estimate of the room IRs via LMS-like algorithm, whose inputs are the noisy multi-channel signal, is obtained. The initial configuration of the IRs are the TDE delays. Then Matched Filtering is applied using the estimated IRs.

2. In parallel, the GSC blocks the desired speech, and subtracts a filtered version of the multi-channel noise. The filters are estimated via the LMS algorithm (independent from the previous step)

3. The two previous steps are interlaced and iterated: the IRs are tracked using a multi-channel input less and less noisy, while the GSC blocks less and less desired speech, as less reverberation is present after Matched Filter is applied.

The algorithm is also capable of tracking small displacement (not head rotations) of the speaker in front of the microphone array. The underlying idea of the algorithm is interesting. However, the exact lengths of the IRs have to be known, in order to perform perfect channel identification, reconstruction and tracking, as explained in [Haykin, 2002] and detailed in domains other than speech processing, when ill-defined delay spreads are experienced [Veen, 1997]. Furthermore, some *a-priori* information about the total energy of the IRs has to be provided: from one side this is an interesting information, because it is plausible for an algorithm to consider that the IR energy inside a room is roughly constant across different IRs, but from another side also IR length is not always accessible, as it entails room calibration prior to utilization. In practice only a qualitative measure of the IR length, i.e. the decaying time of the impulse response, can be accessed, as it will be discussed in Chapter 4. The algorithm was tested in a large room, but the speaker-to-array distance was less than 1 meter, which implies very little late reverberation is present in the real IRs: it is not clear whether the algorithm is able to compensate these effects.

1.7.6 Cepstral Processing

We saw that Matched Filtering provides a way to face the reverberation problem with a multi-channel signal. However, the most important assumption is that the transfer functions between the source and all the microphones are known. This information is not available in practice. The problem of dereverberation is thus inherently linked to that of *channel estimation*, which is in general a hard problem for several reasons:

- It is difficult to separate a room transfer function from the microphone transfer function.
- Room transfer functions are in general non-minimum phase [Oppenheim and Shafer, 1999].
- In real applications the enclosure transfer functions vary with time, sometimes very fast due to head movements.

However, an attempt to identify the enclosure transfer functions with two microphones was proposed in [Petropulu and Subramaniam, 1994]. The authors try to recover the speech signal from the difference of cepstra of the two microphone signals. The clever idea behind the algorithm is to consider a combination of the two transfer function as a new unique signal, whose minimum and maximum phase parts h_{min} and h_{max} have to be reconstructed. The algorithm is here outlined:

- Compute minimum and maximum phase cepstral *differences* between the two channels, exploiting the fact that a minimum (maximum) phase cepstrum is causal (anti-causal).
- From these differences or, alternatively, form its phase [Hayes, 1990; Petropulu and Nikias, 1992], the two parts h_{min} and h_{max} can be reconstructed.
- The minimum and maximum phase cepstra of the two transfer functions are computed from h_{min} and h_{max}.
- The speech signal is obtained via inverse cepstrum operation.

A modified version of the algorithm, called Bicepstrum Iterative Reconstruction Algorithm (BIRA) has been conceived to work when additive Gaussian noise is present [Petropulu and Nikias, 1991, 1992, 1993]. The attractiveness of the algorithm is however smoothed by some underlying assumption, for example that one of the two transfer functions must not have zeros on the unit circle, that there must not be zero-pole cancellations between the speech signal and the enclosure transfer functions, and the two transfer functions must have no zeros in common. The latter requirement may not be the case in reality, mostly because when microphones are close the two functions tend to be similar, so microphones should be apart from each other. However, the authors say that this constraint could be relaxed in presence of more than two microphones. Concerning the zero-pole cancellation, the high variability of speech cannot ensure the fulfillment of this requirement. More details can be found in [Petropulu and Subramaniam, 1996].

1.7.7 Explicit Speech Modeling

The methods described up to this point have a common denominator, i.e. they cast the robustness problem as an attempt to modify the spatial response of the microphone array so that the signal has a higher SNR or SRR. This is done by manipulating physically quantities which have a rather clear physical meaning, such as coherence function, steering vectors, noise blocking matrix and room impulse responses. However, the optimization can be cast in the speech model domain. This attempt showed significant improvement for single microphone algorithms such as stochastic matching [C.H.Lee, 1998] and Parallel Model Combination [Gales, 1995; Gales and Young, 1996], and an analysis of sensitivity to noise estimation errors of some signal and model-based methods was proposed by us in [Brayda, 2004]. However, little work has been done to adapt single-microphone methods to a multi-channel environments. One way to perform this integration is to explicitly express the speech model in a multi-channel framework: in [Brandstein, 1998], for instance, the authors use the Dual Excitation Speech Model, which was proposed in [Hardwick, 1993] for just one microphone, to minimize the same error criterion but averaged on all the microphones. More specifically, the Dual Excitation model is:

$$S(f) = V(f) + U(f) \tag{1.61}$$

where $V(f)$ and $U(f)$ are the Fourier transform of the voiced and unvoiced signals by which speech is assumed to be composed. The voiced part is assumed a weighted sum of N harmonics, the amplitudes and fundamental frequency of which have to be estimated:

$$V(f) = \sum_{n=-N}^{N} A_n W(f - nf_0) \tag{1.62}$$

where A_n is the amplitude for the $n-th$ harmonic. The following relations also hold:

$$f_0 = \frac{f_{pitch}}{f_{sampling}} \quad N = \lfloor \tfrac{1}{2f_0} \rfloor \tag{1.63}$$

which means that the fundamental frequency is normalized by the sampling frequency and that the number of harmonics is uniquely determined by the (estimated) fundamental frequency. The MMSE to minimize is modeled as:

$$\epsilon_{SISO} = \int_{-\infty}^{\infty} \left| S(f) - \sum_{n=-N}^{N} A_n W(f - nf_0) \right|^2 df \tag{1.64}$$

Once the amplitudes and fundamental frequency are estimated, the voiced signal is formed and the unvoiced part is derived by simple subtraction. The expansion to a multi-channel framework is done by including in the error function expressing many $S_m(f)$ as much the microphones are, each of them filtered by a G_m function:

$$\epsilon_{MISO} = \int_{-\infty}^{\infty} \left| \frac{1}{M} \sum_{m=0}^{M-1} G_m(f) X_m(f) - \sum_{n=-N}^{N} A_n W(f - nf_0) \right|^2 df \tag{1.65}$$

Where the voiced part is a Single Output derived by a Multiple Input system (SIMO). The G_m can be the D&S weights of (1.44) or the filters of a filter-and-sum beamformer, or the flipped impulse response of a matched filter as seen in (1.58), a filter-and-sum beamformer, or the MVDR weights of Equation 1.52. The subtraction leading to the unvoiced signal evolves, in the multi-channel context to:

$$\hat{U}_{MISO}(f) = \frac{1}{M} \sum_{m=0}^{M-1} H_m(f)(G_m(f)X_m(f) - \hat{V}_{MISO}(f)) \tag{1.66}$$

where $H_m(f)$ essentially filters noise plus unvoiced parts and can represent the adaptive filters matrix $\mathbf{W}$ seen in a GSC, or can be the input for the generation of a post-filter. The explicit speech modeling approach can be used with other single-channel noise compensation methods. For example in [Brandstein, 1999], the Multi-Pulse Linear Predictive Coding (MPLC) proposed in [Atal and Remde, 1982] is extended with the same averaging procedure across channels. More details and further extensions of the explicit speech modeling can be found in [Brandstein and Griebel, 2000]: in all these multi-channel adaptation, the reverberation effect was always simulated through the image method [Allen and Berkley, 1979] and additive noise was always white Gaussian: such a setup can lead to results quite different from a real office environment. According to us the most important feature to retain of these approach is that the optimization involves many parameters (the A_m, f_0, G_m, H_m just mentioned) which have, each, a precise physical meaning. However, once all of them are "mixed" in a highly non-linear error criterion to minimize, their mutual relation is unknown: the global optimal solution could be found for values that are not locally optimal. For example, if we extract just the pre-filters G_m from the global optimal set of amplitudes, fundamental frequency, pre- and post- filters, they could perform worse then, say, the MVDR solution alone. This will be important in the following.

1.8 Array-based Speech Recognition

An intensive activity of evaluating performances of microphone array based speech recognizers is being conducted world-wide, in particular in the communities related to the EC AMI and CHIL projects: NIST has recently organized benchmarking campaigns NIST [2004] which showed that the error rate provided by a 64-microphone array based recognizer is about twice the error obtained on the corresponding close-talking microphone signal, given a large vocabulary spontaneous speech recognition task. Generally speaking, if microphone arrays are effective to improve the intelligibility of speech signal, it is intuitive that they can also be beneficial for speech recognition purposes. However, ongoing research in this topic showed that the amount of information that can be exchanged with the speech recognizer is not necessarily represented by the output of a SIMO system, i.e. the single-channel, time-domain enhanced speech signal and it is indeed a more complex problem. In noisy environments a tremendous need of robustness of the speech recognition system is needed, and robustness in recognition directly depends on *how* the multi-channel signal has been optimized. Speech Enhancement (SE) generally produces a more intelligible signal. Speech recognition generally process a single-channel signal via a Feature Extraction block (FE), which is responsible of generating appropriate (robust) features across time. These features are finally given to a HMM-based recognizer (REC). However, array-based speech recognition can organize these three fundamental blocks differently, occasionally fuse them, or creating new paths of information between them. We divide state-of-the-art, array-based speech recognition algorithms in three categories, depending on the way they treat the interconnections between the SE,FE and REC blocks.

1.8.1 In-chain Enhancement - Front-end - Recognition

Algorithms belonging to the first group have speech enhancement applied for speech recognition, without any modification to the front-end structure, nor to the recognition decoding phase. This means that the front-end computes feature vectors from a single channel; this channel has the best possible SNR, depending on the speech enhancement algorithm adopted. This framework is depicted in Figure 1.7:

where the SE module have both input $x_m(t)$ and output $y(t)$ in the time domain, the FE module projects the signal $y(t)$ in the frame domain and generates $y[t]$ (t now indicates the frame number), and finally the REC module processes the sequence of feature vectors, generating the text string tr_y. Any of the techniques outlined in Section 1.7 can be used: for example in [Grenier, 1992] the authors

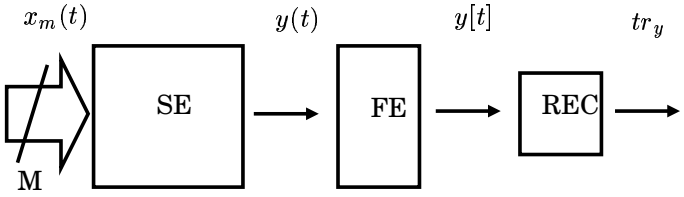


Figure 1.7. In-chain Enhancement - Front-end - Recognition

compare the GSC in the frequency domain and the Frost algorithm in time-domain. Although the task is small and DTW is used in the decoding phase, results in a real car environment show that GSC is superior to Frost method; however, an eigen-constraint on the beamformer is needed because a high sensitivity of the recognizer to steering errors is experienced. The purpose of the eigen-constraint is to widen the main lobe of the directivity pattern and it is computed by extending the delay $\mathbf{d}$ vector to a matrix, where each column represents a fictitious source point around the true point of interest, supposed known. By widening the desired locations, the main lobe widens. In a similar car environment [Oh, 1992] GSC is shown superior to D&S: however they use few microphones (we know the usefulness of D&S is proportional to the number of microphones) and the adaptive filters of the GSC are modified in silence periods only, when the user stops pressing a push-to-talk button: this is clearly done to avoid the problem of spectral leakage (with consequent signal cancellation) the GSC suffers. It was confirmed even in speech enhancement experiments [Compernolle, 1990] that so far that is the only way not to make GSC being detrimental with reverberation. Other ways of optimizing adaptive filter coefficients are proposed in [Gillespie and Atlas, 2002, 2003], based on *correlation shaping*: in the first work the authors point out that the perfect knowledge of the room impulse response is not enough for Matched Filtering to compensate for reverberation, so they propose a Least Square minimization (still being the impulse responses known), which lead to better filters. Though in this thesis we will show that this is not true (it is not clear how they measure neither the inter-channel delays prior to D&S, nor the impulse responses prior to MF) in a very reverberant room, they interestingly point out that the more microphones are used, the few taps per microphone are needed to optimize their filters. In a further work [Gillespie and Atlas, 2003] they indicate the LP residual as the feature that carries most of the information about the reverberation effects, so the filters are optimized on LP residuals and then applied to the time-domain multi-channel noisy speech signal. However, no evidence of this method in real environment is given. Note that, to the best of our knowledge, there is no method that makes use of the phase of the Fourier Transform of each channel, apart from the estimation of the delays in D&S, for which clearer insight will be given in Chapter 4, which could be intuitively useful in this

framework. An attempt to include it in the estimation of per-channel filters was proposed in [Lai and Aarabi, 2004] in a very controlled environment (i.e. little noise, with a low reverberation time). For this reason their claim of the superiority of their method over a Super-directive beamformer is still to be verified in a more noisy room, but the result that even with low reverberation a super-directive approach fails, has to be considered. It is finally worth mentioning the work proposed in [T. Yamada, 2002], where the D&S is compared to the AMNOR noise reduction techniques and a modified Viterbi algorithm is proposed, where an additional dimension representing the speaker direction is added to the usual frame-based HMMs 2D search space of the decoding algorithm. The 3D-Viterbi does not show particular improvement when the speaker is in a fixed position, but it does improve recognition results with a moving speaker and its benefits are additive with an adaptive filter-and-sum beamformer.

1.8.2 Enhancement connected with Front-end - Recognition

Another way to perform array-based speech recognition is to relax the constraint of the single-channel entering the FE block. Broadly speaking, the speech enhancement is most of the time performed in some transform domain, such as frequency or quefrency; the feature extraction process passes through the same domain as well, so it is intuitive that the multi-channel features could "survive" as such and directly be given to the FE block, without being synthesized to a time-domain mono signal. The general strategy is depicted in figure 1.8:

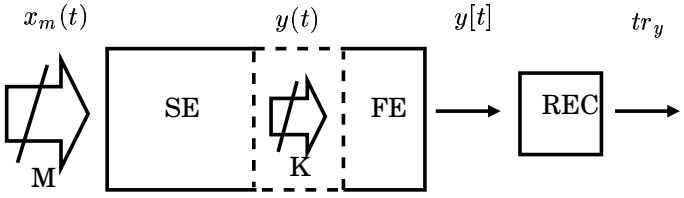


Figure 1.8. Enhancement connected with the Front-end - Recognition

where a set of K features is directly fed to FE. One such a system is proposed in [Sullivan and Stern, 1993], where the M microphone signals are each divided into C sub-bands, then C cross-correlation coefficients are derived (all the microphones contribute to a single correlation coefficient) and normalized, cepstrum is computed and fed to the recognizer. Clearly a single-channel enhanced signal was not generated in this process. Though the method is interesting, because the correlation between microphones should be a relevant information for the recognizer. Nevertheless performance are relatively good in simulations (the array is simulated, white noise is added and no reverberation is present), while it fails with real data. In simulations the cross-correlation-based

algorithm improves both with respect to (single-channel) LPC processing and when recording with a microphone different from the one used at training time, but surprisingly in real environments LPC is better and testing with matched microphones lead to worse results than with unmatched microphones. Considerations about these phenomena will be discussed in Section 1.9. In [Shimizu, 2000] a *space diversity* approach is used: the microphone network is not a linear array. Specifically, microphones from a distributed network in the room, with no *a-priori* knowledge of their geometry. Thus, beamforming is impossible for spatial aliasing constraints. The time-domain signal can't be summed, but the authors propose to sum and average the multi-channel signals in the *feature* domain. Performances are low because microphones far from the speaker (the worst from a recognition point of view) tend to decrease the overall performance. Another approach was proposed in [McCowan, 2002], where the FE block is fed with the output of a speech enhancement phase (D&S + post filtering as explained in (1.7.3)) and the output from a *mask estimation* phase. The mask evaluates how much a specific band is affected by noise and tells the recognizer to discard that band if the noise is thought to be beyond a certain threshold, thus making recognition with a *missing data* technique. The algorithm is tested in a simulated environment and relies both on noise estimation and parameter tuning for thresholding, which can be unpractical, but interestingly it shows that a beamformer can be used, apart from generating an enhanced speech, also to derive other environmental useful information.

1.8.3 Enhancement - Recognition feedforward/feedback

We classify in the third and last group those algorithms aiming to exchange information between the enhancement and the recognition blocks. As can be seen in Figure 1.9, this exchange can be

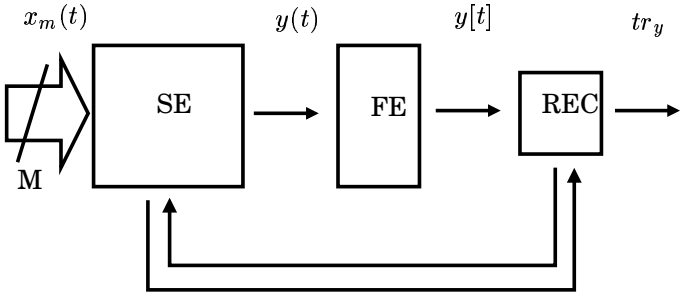


Figure 1.9. Enhancement - Recognition feedforward/feedback

done in two ways: either the multi-channel processor adds some constraints to the recognition step (in this case a *feedforward* recognition is done), or the recognizer output constrains the speech enhancement phase (a *feedback* path is created). The first idea was proposed in [McCowan and

Sridharan, 2001], where the actual multi-channel signal is composed by $M * C$ streams, where C is the number of subbands. Each band is beamformed and recognized, so at the end of the chain the recognizer has to fuse the (possibly different) scores. This is done by weighting each sub-band acoustic score with a function that depends on an estimated per-band SNR, which is computed by the enhancement block. This method showed improvements over beamforming methods when just additive white noise is corrupting only one band at a time for each utterance. The last method we present is based on the work proposed in [Seltzer, 2003], with a feedback scheme: a filter-and-sum beamformer is driven by the output of a first recognition step. The coefficients of the filters are estimated so that the following criterion is satisfied:

$$\hat{\mathbf{h}} = \arg\min_{\mathbf{h}} \sum_{\mathbf{t=1}}^{\mathbf{T}} \|\mathbf{z_t} - \mu_{\mathbf{s_t}}\|^{\mathbf{2}} \tag{1.67}$$

where (h) are the filter coefficients, $\mathbf{z}_t$ is the feature vector at frame t and μ_{s_t} is the Gaussian mean vector of the model associated to frame t after a Viterbi alignment is performed. More details about this method will be given in Chapter 2, but by now it is sufficient to note that this method attempts to reduce the distance between the features in the cepstral or log-Filterbank-energies domain, and the clean speech models, of course in the same domain. Because the beamformer is driven with a maximum likelihood criterion, it is called Limabeam. The technique is appealing because the optimization is done in a domain much closer to the recognizer rather than in way that attempts to increase the SNR only. Other authors [Raab, 2004] attempted to cast the optimization in the cepstral domain, but no apparent improvement was obtained.

1.9 Speech Enhancement vs Speech Recognition

We have seen that the geometry of a microphone array has consequences on its selectivity, which is mathematically the width of the main beam in the directivity pattern (see (1.3.1)). If a very precise algorithm can locate the speech source, then more microphones and/or a larger inter-microphone distance can be used. However, if the source is moving and the new position can't be quickly tracked, the desired signal would fall outside the main beam, with disastrous consequences on enhancement and recognition performances. In practice, research has clearly shown that excessive precision can limit robustness, and a wider main lobe is the best solution to adopt. Furthermore, because of reverberation effects, restricting the beam to one direction only may imply neglecting the other paths, which can be a precious source of information (the Matched Filtering, see Section 1.7.4, demon-

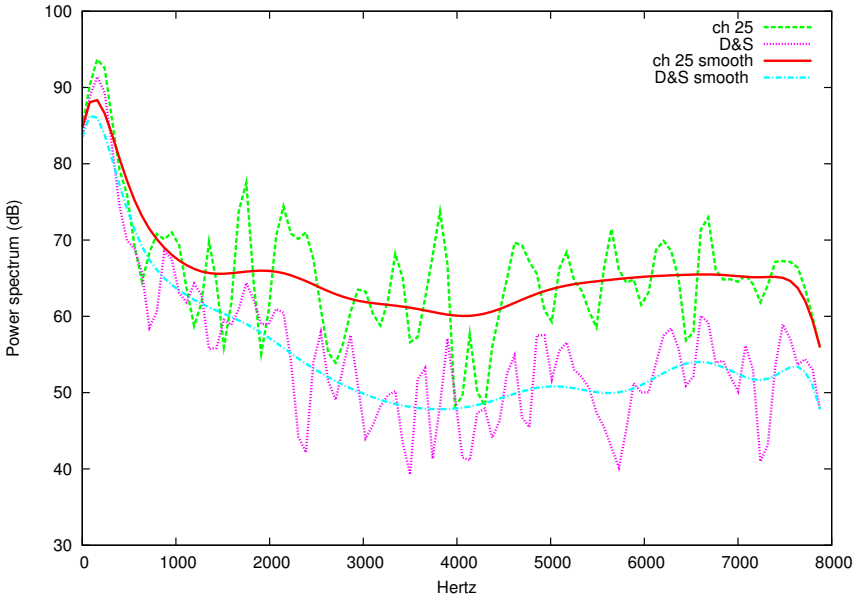


Figure 1.10. Low pass-effect of the D&S beamforming on speech acquired in a real noisy environment.

strates that), which has anyway to be handled. This is why in this work we chose to keep the number of microphone not high (eight), which is also the case in some more practical applications such as PC-desktop (on screen) and car environments. We briefly analyze in the following the advantages and drawback of speech enhancement algorithm when applied to speech recognition.
It is well known that D&S beamforming has a low-pass effect: in practice, the destructive interference is stronger at higher frequencies. This can be seen in Figure 1.10, where spectra and smoothed spectra are plotted for a single channel and for a signal beamformed with 8 microphones in a real environment with a 0dB additive noise.

In general, MVDR-based methods (superdirective beamformers, constrained or unconstrained), provide the optimal solution for a given (stationary) sound field only and for a narrowband signal only [Monzingo and Miller, 1980], while we know speech is a broadband signal. Furthermore a MVDR solution is not optimum in a Minimum Mean Squared Error (MMSE) sense [Simmer, 2001], which we would like to be the case if we want to apply a Maximum Likelihood criterion as we do in ASR. This non-optimality motivated the study of post-filters, which nevertheless can introduce artifacts in the speech signal prior to recognition [Omologo, 2001]. Furthermore super-directive beamformers need a precise estimation of the noise field and fail in reverberant environments, simply because they are not designed to face them. A GSC attempts to perform noise compensation, effectively increases the SNR if no reverberation is present and a perfect look direction is pre-determined. Some efforts were done to limit the signal cancellation [Hoshuyama, 1999; Com-

pernolle, 1990], but it was shown that GSC (together with post-filtering techniques) can produce limited improvement [Omologo, 2001; Seltzer, 2003] over D&S when the output is fed to a speech recognizer.

Matched Filtering and Correlation Shaping are interesting cases where already maximizing, respectively, the SRR and the SNR entails a perceptual degradation [Rabinkin, 1998]: this topic will be covered in depth in Chapter 4, but in synthesis it has been discovered that using the full length Matched Filtered, some pre-echo are artificially created in the waveform to be optimized. Though the SRR (and possibly the SNR) are maximized, the intelligibility of speech is disturbed. It is intuitive that this would degrade recognition performance as well, thus implying that maximizing the SNR cannot be the objective function for a dereverberation technique applied to speech recognition.

The Subspace Tracking algorithm could be useful for recognition, but *a priori* knowledge of both energy and length of the room IRs is needed, which can be unpractical. Though, it drives the attention on the fact that early reverberation has the effect of canceling speech when beamforming is performed, and to the necessity of tracking IRs once they are given or estimated. The Cepstral deconvolution outlined in (1.7.6) has not been tested yet for speech recognition purposes. The BIRA algorithm makes use of the speech phase, which is generally discarded for recognition, even though recent work for single microphone showed it can be included [LiDeng, 2004]. For it to be applicable, the recorded signal must have no zeros on the unit circle, which is equivalent of demanding no vanishing cepstra and can be avoided when pre-emphasis is applied. However the microphones have to be far apart from each other and this can be in contrast with the geometry of a microphone array. The method is however an attempt to move the channel compensation problem in the cepstral domain, which is successfully achieved by CMN, at least for early reverberations, for single channel ASR. To the best of our knowledge, methods addressing late reverberation effects compensation with microphone arrays in the feature domain are far to be effective in real environments.

Instead of compensating features, model contamination [Matassoni, 2002] has been proposed and moves the problem to a completely different perspective, where the clean speech signal is no more the desired output, but one rather wishes a better matching between the noisy features and a set of model as close to the real conditions as possible.

The explicit speech modeling hints that, when optimizing a set of physically different parameters all together, the solution found may not be necessarily optimal for a reduced set of these parameters: this means that if we are looking for parameters that maximize the likelihood of a certain feature set given a model set, the optimal set of parameters may not maximize the SNR. For two different objective functions it is likely that the optimization criterion leads to two different optimal

sets.

The correlation-based approach outlined in 1.8.2 shows an important result: it confirms that when the complexity of experimental conditions grows (each microphone records a different room transfer function, furthermore coupled with an unknown device transfer function, different for each microphone), the distance between simulations and real experiments grows as well. The fact that an algorithm works by simply adding a specific kind of noise (most authors start adding white noise) does not at all imply that in real conditions the performance will be maintained.

Furthermore, if recognition in a meeting room only is addressed, recent work [Ferras, 2005] showed that D&S beamforming gives high performance if the inter-microphone-delays are very carefully computed, for example exploiting the cross-correlation between the channels [Omologo and Svaizer, 1997]. Surprisingly, in this case a "well done" D&S is superior to the time-varying filters proposed by Aarabi (see (1.8.1)), to the multi-channel feature averaging technique of Shimizu and to the cross-correlation-based technique proposed by Sullivan, both outlined in (1.8.2). Support to this work comes from the fact that the recognition rate was averaged on four different meeting room databases, coming from CMU (Carnegie Mellon University, USA), ICSI (International Computer Science Institute, USA) [Janin, 2004], LDC (Language Data Consortium, USA) and NIST (National Institute of Standard and Technologies, USA) [NIST, 2004]. For this reason along this work results will be compared to the best version of D&S we can attain.

What it comes out from the analysis of array-based speech enhancement and speech recognition, is that sometimes the best algorithms in simulations are the worst with real data. Also decreasing the microphone mismatch or increasing the length of the filters does not necessarily imply increasing recognition performances. In addition, when the problem of enhancing speech is cast in a recognition framework, some implications which are thought obvious from a perceptual point of view (SRR, SNR, look direction, microphone mismatch) are not valid anymore. It is indeed from this important consideration that focusing on likelihood-based criteria may be the best thing to do in such a complex framework.

Chapter 2

The Limabeam algorithm: principles, implementation and results

2.1 Theoretical background

2.1.1 Front-end

We briefly introduced the Limabeam algorithm in Section 1.8.3. In this chapter we look closer to its theoretical background, we then detail how we implemented it and we present experiments on a small database. The Limabeam algorithm is an adaptive filter-and-sum beamformer.

As most enhancement algorithms do, the multi-channel signal is first time-aligned:

$$x'_m(t) = x_m(t - \tau_m) \tag{2.1}$$

where the delays τ_m are computed with respect to a reference microphone, which can usually lies in the center of the array. The purpose of this operation is to form a "beam" in the direction where the interested source is supposed to be. Once the signals are aligned, they are filtered: by recalling Equation (1.34), the general time-domain form of a filter-and-sum beamformer, applied to a discrete M-microphone array, is:

$$y(t) = \sum_{m=0}^{M-1} w_m(t) * x'_m(t) \tag{2.2}$$

The filter $w_m(t)$ has the form announced in Equation 1.50 and can be represented as a FIR filter. In this algorithm a FIR filter of L taps is used for each channel. In the discrete time domain [1] (2.2) becomes:

$$y[k] = \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} w_m[l] x'_m[k-l] \tag{2.3}$$

where k is the time domain index and where the convolution operation was explicited. Clearly MxL filter parameters are involved in (2.3). The total set of such parameters can be represented as a large vector $\mathbf{w}$:

$$\mathbf{w} = [w_0[0], w_0[1], ..., w_0[L-1], w_1[0], ...w_1[L-1], \quad ... \quad , w_{M-1}[L-1]] \tag{2.4}$$

and thus the vector containing all the samples of the beamformed signal $\mathbf{y}[\mathbf{w}]$ depends on this large vector.

Feature extraction is then performed on the beamformed signal. In Limabeam, Log Filter Bank Energies (LFBE) are computed on the Mel scale for each frame:

$$\mathbf{e}_t[\mathbf{w}] = \log_{10}\left(A|\mathscr{F}(\mathbf{y}_t[\mathbf{w}])|^2\right) \qquad \forall t : 0 \to T-1 \tag{2.5}$$

where $\mathbf{e}_t[\mathbf{w}]$ is the vector of LFBEs (E being its dimension, i.e. the number of triangle-shaped frequency bands in the FilterBank plus the log-energy of the total signal) computed from framed beamformed signal $\mathbf{y}_t[\mathbf{w}]$, $\mathscr{F}$ denotes Fast Fourier Transform (FFT) and A is the ExF Mel matrix (F being the number of FFT points). From these, Mel Frequency Cepstral Coefficients (MFCC) are computed:

$$\mathbf{c}_t[\mathbf{w}] = \mathbf{\Phi}\mathbf{e}_t[\mathbf{w}] \tag{2.6}$$

where $\mathbf{c}_t[\mathbf{w}]$ is the vector of MFCCs (C being its dimension) and $\mathbf{\Phi}$ is a Discrete Cosine Transform (DCT) which provides rotation and quasi-decorrelation. Together, the T feature vectors are given to an HMM-based recognizer, which generates the most likely transcription.

[1] note that we express (2.1) and (2.2) in the continuous time domain because the delays can have a precision of less than a sample

2.1.2 Back-end and feedback

The output of a general speech recognition system, based on optimal Bayesian classification, is the estimated transcription $\hat{tr}$ which maximizes the joint probability of the observed features and a possible transcription tr, i.e:

$$\hat{tr} = \arg\max_{tr} P(\mathbf{c}[\mathbf{w}], tr) = \arg\max_{tr} P(\mathbf{c}[\mathbf{w}]|tr)P(tr) \tag{2.7}$$

where the features are expressed in the cepstral domain. The *a priori* probability $P(tr)$ can be excluded from the maximization if we assume that $P(tr)$ is a constant, i.e. all sentences are equiprobable. Thus, the maximization can concern the sole term $P(\mathbf{c}[\mathbf{w}]|tr)$, which is the *likelihood* that a given transcription tr justifies the $\mathbf{c}[\mathbf{w}]$ feature vectors.
In Limabeam, the optimal weights are searched for a given hypothesized transcription which maximizes the likelihood in the LFBE domain, instead of the MFCC domain: this is done because the MFCC are known to lead to high performance in ASR, while the LFBE are chosen for optimization because they give equal chance to all speech bands to be optimized. This issue will be further investigated in Chapter 3. Finding optimal weights consists in finding the set $\mathbf{w}_{opt}$ of filter coefficients which satisfies:

$$\mathbf{w}_{opt} = \arg\max_{\mathbf{w}} P(\mathbf{e}[\mathbf{w}]|\hat{tr}) \tag{2.8}$$

It is important to note that the transcription used in (2.8), $\hat{tr}$, is the one used in (2.7). Equation (2.8) states that, given a transcription (from Equation 2.7), a set of filters $\mathbf{w}_{opt}$ can be found that maximizes the likelihood of the features (in the LFBE domain). In other words, the recognizer is used a first time to process the features in the cepstral domain and generate a possible transcription, then a system tries to increase the probability that this transcription generates the observed features in the LFBE domain. Limabeam is in fact a two step recognition algorithm, where in the second step the optimization of the filter coefficients is indeed an adaptation phase. This algorithm is similar to the Viterbi training algorithm, but differs by the adaptation of filter parameters instead of HMM parameters. If a HMM-based speech recognizer is used, then the maximization of the likelihood in (2.8) has to be performed across all possible paths which connect HMM states:

$$\mathbf{w}_{opt} = \arg\max_{\mathbf{w}} \sum_{s \in S} \prod_{t=0}^{T-1} P(\mathbf{e}_t[\mathbf{w}]|s_t)P(s_t|s_{t-1}, \hat{tr}) \tag{2.9}$$

where S represents all the possible state sequences. For each state s_t, the likelihood is decom-

posed in the product of two conditional probabilities: the first one (the so-called *output probability*), expresses the likelihood for the given state s_t to generate the features $\mathbf{e}_t[\mathbf{w}]$, while the second (the so-called *transition probability*) expresses the probability of visiting the current state at frame t given a state history restricted to the previous state (Markov assumption). It is furthermore assumed that, among the S set, only the state sequence which gives the maximum likelihood is retained (which replaces the sum term in (2.9) with a *max* operator on the S set). This is because the contribution of the maximum likelihood path is large compared to lower likelihood ones (Viterbi assumption). By taking log probabilities (the monotonicity of which does not alter the optimization), Equation (2.9) can be rewritten as:

$$\mathbf{w}_{opt} = \arg\max_{\mathbf{w}} \sum_{t=0}^{T-1} [\log P(\mathbf{e}_t[\mathbf{w}]|s_t) + \log P(s_t|s_{t-1}, \hat{tr})] \tag{2.10}$$

According to this equation, filter coefficients have to be optimized by jointly optimizing the two terms in the summation. In Limabeam this is done iteratively by first optimizing the state sequence, then finding optimal weights given the optimal state sequence. The procedure can be iterated until convergence of the likelihood function. Specifically, the optimal state sequence in ((2.10)) can be evaluated with a Viterbi *forced alignment*. The inputs of the forced alignment are the hypothesized transcription, the features and the HMM parameters, the output is a frame-to-state matching, which allows the Log LikeliHood:

$$LLH[\mathbf{w}] = \sum_{t=0}^{T-1} \log P(\mathbf{e}_t[\mathbf{w}]|s_t) \tag{2.11}$$

to be evaluated and maximized. We expand equation (2.11) in the case the output probabilities are modeled by multivariate Gaussian distributions:

$$LLH[\mathbf{w}] = \sum_{t=0}^{T-1} -\frac{1}{2} \left[(\mathbf{e}_t[\mathbf{w}] - \mu_{\mathbf{e}_t})^T \Sigma^{-1}_{\mathbf{e}_t[\mathbf{w}]} (\mathbf{e}_t - \mu_{\mathbf{e}_t}) + Elog(2\pi) + log\,|\Sigma_{\mathbf{e}_t}| \right] \tag{2.12}$$

where $\mu_{\mathbf{e}_t}$ and $\Sigma_{\mathbf{e}_t}$ are respectively the E-dimensional mean and the ExE covariance matrix of the Gaussian to which the vector $\mathbf{e}_t$ is aligned to at frame t and $|\cdot|$ is the determinant operator. Most frequently, mixtures of Gaussians are used as output probabilities. However, we restrict the study to the mono-Gaussian case, for the sake of simplicity. It can be noticed that in (2.12) the dependence of the mean and covariance matrix on the parameters $\mathbf{w}$ is not explicited, because when the LLH function is maximized the state sequence is fixed, thus changing parameters does not affect the models. We recall that models are trained with clean speech. Maximization of (2.11)

implies computing the filter coefficients which cancels the gradient:

$$\nabla_{\mathbf{w}} LLH[\mathbf{w}] = \sum_{t=1}^{T} J[\mathbf{w}]\Sigma_{\mathbf{e}_t}^{-1}(\mathbf{e}_t - \mu_{\mathbf{e}_t}) \tag{2.13}$$

where $J[\mathbf{w}]$ is the Jacobian matrix (which has ML rows and E columns), formed by the partial derivatives of each of the E features with respect to each of the ML filter taps. Because the LLH function in (2.12) and its gradient in (2.13) depend both non-linearly on the set of filters $\mathbf{w}$, non-linear optimization is performed via Conjugate Gradient Ascent [Press, 1988]. Once the set of filters $\mathbf{w}_{opt}$ is obtained, then it is used to re-beamform the multi-channel signal and a second recognition step is performed. Seltzer showed that on average this technique leads to improvement in speech recognition accuracy.
The system is depicted in Figure 2.1.

Each channel $x_m(t)$ contributes to the calculation of the delays τ_m, which are applied in the TDC block according to Equation (2.1), then signals are filtered and summed according to (2.3), where the filters are initialized to:

$$\begin{cases} w_m[0] = \frac{1}{M} \\ w_m[i] = 0 \end{cases} \quad \forall i : 1 \rightarrow L-1, \forall m : 0 \rightarrow M-1 \tag{2.14}$$

Then, for each frame, feature vectors $\mathbf{c}[\mathbf{w}]$ and $\mathbf{e}[\mathbf{w}]$ are computed in parallel according to (2.6) and (2.5). Recognition is performed by the REC block (step 1, as in Figure 2.1) and the hypothesized transcription $\hat{tr}$ is generated as in (2.7). Models (relative to the words of the utterance $\hat{tr}$) and features in the cepstral domain are then given to the ALIGN block (step 2): a state s_t, and consequently a Gaussian distribution is assigned to each frame. After that, HMMs , features $\mathbf{e}[\mathbf{w}]$ (both in the LFBE domain) and the estimated state sequence are then given to the OPT block (step 3) for optimization: the Conjugate Gradient iteratively computes equations (2.12) and (2.13) up to convergence. After convergence a $\mathbf{w}_{opt}$ large vector is found, all the $x_m(t)$ are re-filtered and summed. This last operation is the actual *feedback* which is given to the beamformer from the back-end, as it can be observed in Figure 2.1. From this new beamformed signal, features are re-computed and the second and last recognition (step 4) is performed. For the sake of clarity, Figure 2.1 induces to think that the feedback element is the filter set $\mathbf{w}_{opt}$: this is true if we consider the optimization process being part of the Back End (the recognizer). Alternatively, as it is done in [Seltzer, 2003], the feedback element is the hypothesized transcription, and the optimization stage is thus part of the Speech Enhancement and Front-End. The two points of view are equivalent, as the optimiza-

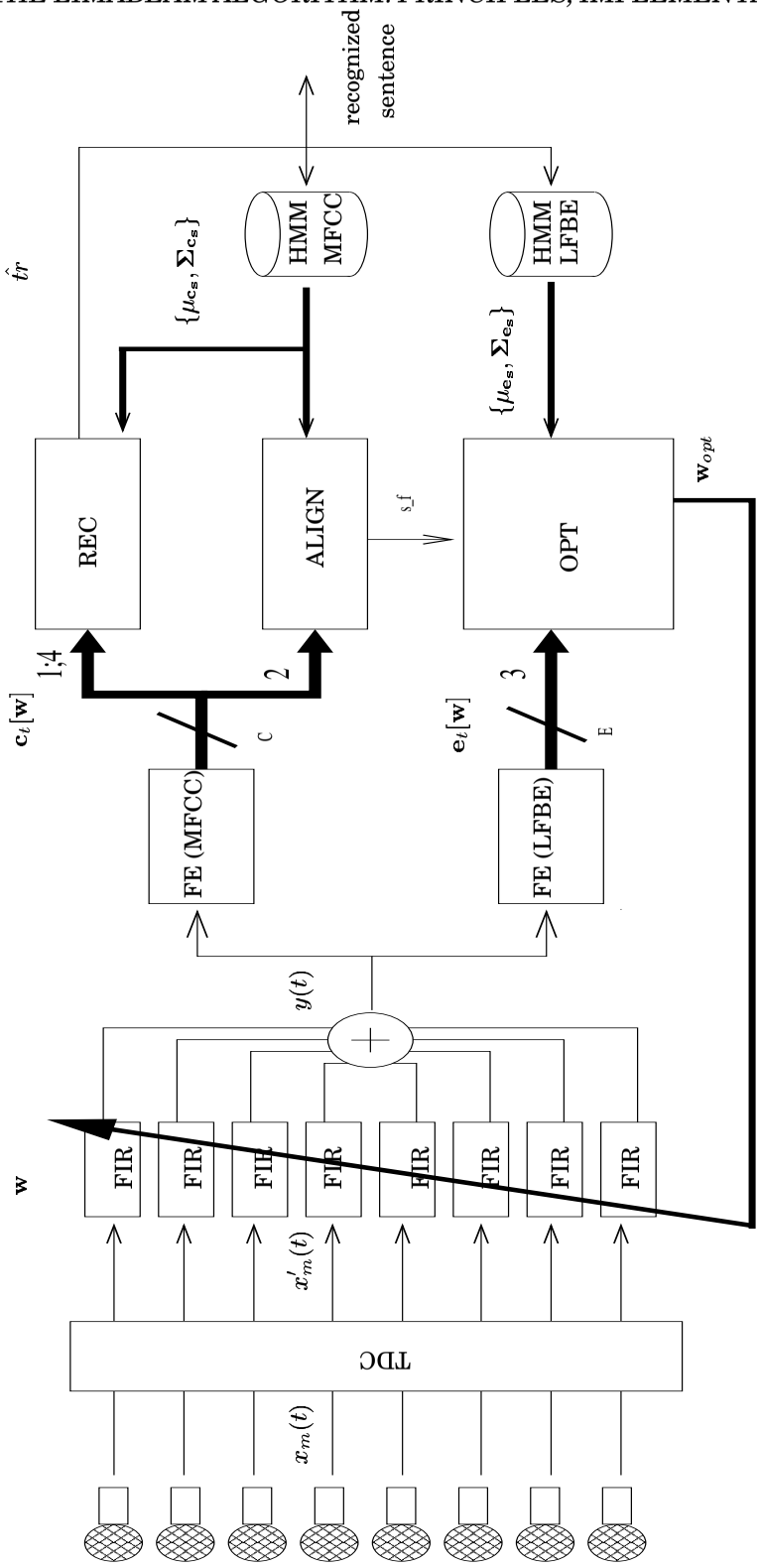


Figure 2.1. Architecture of the Limabeam algorithm

tion exchanges data with both the Front-End and the Back-End (see Figure 1.9). We will provide descriptions in either way in the following.

2.1.3 Oracle, Calibrated and Unsupervised Limabeam

We have seen that Limabeam optimizes a set of filters for each utterance in a test set, trying to reduce the distance between the features set and the clean speech model set to which they are aligned. Because Limabeam is an adaptation algorithm, its performances are limited by the amount of correctly recognized data (words or phonemes, depending on the type of vocabulary) during the first recognition step. Intuitively, if adaptation is performed on badly recognized data, the filters derived should lead to worse performances in the second recognition step. The algorithm can accept an amount of supervision. Specifically, three ways of supervising this algorithm were proposed in the original work:

- Oracle Limabeam (OL)
- Calibrated Limabeam (CL)
- Unsupervised Limabeam (UL)

If the filters are optimized for every utterance given the correct tr (and not an estimated $\hat{tr}$), then *Oracle* Limabeam [Seltzer and Raj, 2003] is performed. It is interesting to study this case, because it gives the upper bound of performances if the Viterbi alignment is the best possible. It should furthermore be clarified that for "Oracle" we intend a system providing the knowledge about the correct transcription to the Limabeam algorithm. However, one can give an even more precise information to the optimization process: instead of providing the correct transcription only, the alignment with respect to the close talk signal is given. In this way the Viterbi alignment should not be disturbed by environmental noise. Furthermore, the amount of supervision can be varied in another way: instead of providing for each utterance the correct transcription, one can do that for only a part of the test set, then "freeze" the filters estimated and use them to process the rest of the test set. In this case *Calibrated* Limabeam [Seltzer and Raj, 2001] is performed. The advantage of this technique is that the computational complexity lays only in the calibration phase. The evident drawback is that calibration should be done for each variation of speaker and noise source position and direction in the environment, variation which in general is not known a priori. Finally, if calibration data is not available, the filters are optimized directly from the output of the first recognition step, which can be right or wrong. In this case we are talking about *Unsupervised* Limabeam [Seltzer, 2002]. Throughout this work we will deal with the three versions.

2.1.4 Subband vs Time domain implementations

Seltzer proposes two implementations of Limabeam: in the first [Seltzer, 2004], FIR filters act in the time domain, just as any conventional filter-and-sum beamformer. In the second [Seltzer and Stern, 2003, 2006], called *Subband* Limabeam, FIR filters act on the DFT coefficients (each DFT bin is considered a subband) of each microphone signal. In the frequency domain this can be represented as:

$$Y_t[b] = \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} W_m^l[b] X_m^{'(t-l)}[b] \tag{2.15}$$

where $Y_t[b]$ is b-th bin of the DFT of frame t, $W_m^l[b]$ is the l-th tap of the FIR filter relative to the b-th DFT bin and microphone m, and $X_m^{'(t-l)}[b]$ is the b-th bin of the m-th microphone signal on the $(t-l)$-th frame, i.e. when the Fourier Transform was made l frames back the current frame t. Equation (2.15) can be compared with (2.3) to see that the filter have the same structure, but while the first has taps acting on time samples, the second acts on subband version of the fullband speech signal. In both cases the function to maximize is globally the same (Equation (2.12)), i.e. the optimization is carried in the LFBE domain. In the subband case the derivation of the Gradient expression differs from Equation (2.13). From the performances point of view, in [Seltzer, 2003] it is said that the difference is theoretically only in the computational complexity: it is shown that the subband implementation requires much less coefficients to estimate. Specifically, a bank of L taps subband filter would be equivalent to a $N + (L-1) \cdot R$ time domain filter, where N and R are the window size and the frame rate expressed in samples. Since subbands are not independent, the optimization cannot be done independently for each bin. On the other hand, by processing jointly all the bins, the complexity would even increase with respect to the time domain counterpart. The solution contemplated is to process independently groups of bins which own to different Mel triangles. In this way the highest complexity is driven by the number of DFT bins in the largest Mel triangle.

In this work we will not consider the alternative implementation in the subband domain of Limabeam, for three reasons:

1. The performance are theoretically the same: the S-Limabeam is just a way of estimating parameters more efficiently.

2. The time domain implementation allows to observe easily how filters evolve in the optimization when compensating for reverberation. In fact shapes of FIR impulse responses are well

known in time domain, and some possible ways of inverse filters, e.g the Matched Filtering, have a time domain FIR representation, which is much more comfortable.

3. We would like to explore the case in which the optimization is done in the MFCC instead that in the LFBE domain. This has an important consequence on a possible sub-band implementation, because each cepstral coefficients would be function of all the Mel triangles. The complexity would be in this case even increased with respect to the time domain implementation.

2.2 Implementation

In this section we give some insights about the implementation of the Limabeam algorithm, mainly focusing on the computation of the features, on the different feature domains involved in the optimization step, and the choices made about the way the adaptive filters are optimized.

2.2.1 Parametrization

The Limabeam algorithm was implemented by Seltzer in 2003 with the use of the SphinxIII recognition system [CMU, 2003] for handling speech models, and the FFTW [FFTW, 1999] libraries for signal processing routines. We decided to use the Hidden Markov Model Toolkit (HTK). In this thesis the version 3.2.1 is used [Young, 2003]) for both front-end and back-end processing. HTK has the advantage of being freely used, but the drawback of being very hard to modify. This is because the signal processing routines always pass through real-time targeted bufferization, which besides makes any intervention on the code rather difficult. We decided to create modified modules inspired from HTK and to couple them with the core of Limabeam, which we implemented (except the gradient ascent algorithm part). The need of such independent modules rises from the fact that each iteration of Limabeam (see 2.1.2) involves the operations of Filter and Sum Beamforming, computation of Spectra, Log Mel Filter Bank Energies and Mel Frequency Cepstral Coefficients. The modules created are listed in Table 2.1.

Once the modules were created, their output was compared to the corresponding HTK modules output, to check their equivalence. This is done because the HTK modules, which are faster, are used whenever possible: they are in fact used in the Feature Extraction, the ALIGN and REC blocks of Figure 2.1. The created modules are instead used in the filter-and-sum beamforming step and in the OPT block.

Modules for speech processing
Channel filtering (for filter-and-sum)
Hamming window and Pre-Emphasis
Mel Matrix Computation
Computation of LFBE and MFCC
Filter ans Sum beamformer

Table 2.1. Signal processing Modules created and used in the front-end and optimization steps of Figure 2.1:

2.2.2 Model handling

In [Seltzer, 2002] both Viterbi alignment, prior to the FIR optimization routine, and recognition are performed with MFCCs together with the log energy and their first and second derivatives, while the Likelihood maximization is done in the LFBE domain. Two sets of models were trained, one for each domain, from the same training set. The set of 1 Gaussian LFBE models used in [Seltzer, 2002] are estimated via Single Pass Retraining on the MFCC models, while in this work the sets are trained in parallel because HTK allows this kind of retraining only when both models have exactly the same dimensions of speech vectors. We use 24 LFBE for the optimization models.

2.2.3 Conjugate Gradient.

The Conjugate Gradient algorithm is the heaviest part of Limabeam in terms of computational complexity. This happens because an ML x E matrix has to be computed for each frame of a test utterance (we recall that M is the number of microphones, L is the number of taps for a single-channel FIR filter and E is the dimension of the Filter-Bank + Energy vector). This Jacobian matrix represents the effect that a variation of a single filter tap has on one specific energy of the filter bank. After being computed, the Likelihood Gradient vector (whose dimensions are ML x 1) is computed according to Equation (2.13). The Gradient indicates how close to a global maximum the Log-LikeliHood is. To check its increase, the LLH function is also continuously computed according to (2.12). Both the LLH and its Gradient together with a set of FIR coefficients are fed to the routine which implements the non-linear Conjugate Gradient algorithm. Many Conjugate Gradient implementations exist in the literature.

Finding conjugate directions

In [Shewchuk, 1994] two main methods are described to find conjugate directions: the Fletcher-Reeves method and the Polak-Ribiere method. The Fletcher-Reeves method may fail in finding conjugate directions if the initialization point is not sufficiently close to the final solution (whose

Modules for Conjugate Gradient
Loader of HTK LFBE models
Loader of Forced Alignment output
Log Likelihood Computation
Jacobian Matrix Computation
Gradient Computation

Table 2.2. Modules created around Limabeam: Conjugate Gradient

location in the likelihood landscape [1] is of course unknown), while the Polak-Ribiere method is faster but may fall in infinite loops. These loops can be avoided by periodically re-starting the steepest ascent search routine.

Finding zeroes along conjugate directions

Furthermore, there are two methods for finding maxima of the function in the chosen conjugate direction: the Secant method is preferable to the Newton-Raphson method because it only requires the calculation of the first derivative of the function being minimized in two points, while with Newton-Raphson the Hessian should be computed. We chose to use the Polak-Ribiere method coupled with the Secant method. We found an implementation in [Press, 1988], which computes the FIR coefficients which maximize the LLH, following its Gradient. Finally, two versions of Conjugate Gradient exist in [Press, 1988]. The first, which Seltzer used [Seltzer, 2004] and was used in part of this work as well, iteratively moves to the gradient direction, then evaluates the LLH as much times until a minimum point is reached (about 20 times per iteration), then the gradient is re-computed. Convergence is reached in function of a tunable threshold. Experiments show that very few iterations are needed (about 4 or 5) to find a minimum. The second makes a much more frequent use of the Gradient and, while it can give better performance, it is much (about 20 times) slower. This happens because most of the time the algorithm computes Jacobian matrices [2]. The modules written to handle the Conjugate Gradient input/output are listed in Table 2.2:

[1] for "likelihood landscape" we intend the likelihood expressed in function of one or more FIR coefficients, regardless of the domain (MFCC, LFBE) considered

[2] While monitoring the LLH evaluations through the iterations, we noticed that to avoid infinite loops in the Polak-Ribiere method, the steepest descent direction search routine is re-started after a certain number of iterations: much more points than what is necessary are evaluated and the algorithm of Numerical Recipes wastes time in the LLH calculations. Furthermore, the step-size is non-adaptive: this means that when being nearby a local minimum, still the LLH of very far points in the solution space are considered. Because LLH and gradient evaluations are the more costly part of the algorithm, we found a more suitable implementation in [MacKay, 2004]. Performance were almost exactly the same, but with a three times lower computational cost. Further speed was reached by calculating Fourier Transform with the FFT-Mayer routines [Mayer, 2001].

2.3 First experiments

The purpose of the following sections is to test the performance of Limabeam with a relatively simple task. The first experiments on the Limabeam algorithm are intended to be conducted in a controlled environment (position of the speaker known, no reverberation). This is done because it would be difficult to understand immediately the causes of recognition errors when speech is uttered in a very reverberant environment, is surrounded by many noise sources, and the position of the speaker is arbitrary.

2.3.1 Environmental setup and task

Because we intend to first match Seltzer's results, no convolutional distortion, i.e. no reverberation is desired for this set of experiments.

Two simulated environments: white and fan noise

Consequently pure, non reverberated noise is synthetically added. Specifically, 8 virtual channels are created by adding 8 noise signals to the clean speech. Speech is synchronous across microphones (far field assumption), which results in simulating a distant speaker right in front of the virtual array. This implies that perfect array steering is assumed, which is reasonable because in this phase we want to avoid possible steering errors. Two scenarios are considered for noise robustness. In the first scenario, a segment of white uncorrelated noise is run-time randomly selected from a long stationary white noise file and is added channel by channel to the clean speech signal. This simulates one or more noise sources, generating a perfectly non-coherent noise field (this type of noise field was previously described in 1.4.1). The power of the white noise is the same for all the channels. In the second scenario, which is slightly more realistic, several seconds of real fan noise from a computer were recorded in a 3,5 $\times$ 3,5 $\times$ 4m office, using a Sennheiser microphone very close to the noise source. Because the microphone was very close to the noise source (about 10 cm from the fan), the reverberation effect of the room is negligible and complies with the specifications mentioned before. From this fan noise signal, 8 noise signals are extracted: they are chunks of the same noise file shifted by 0.12 ms (i.e. the second microphone records the same noise 0.12 ms after the first microphone, the third microphone records the same noise 0.12 ms after the second and so on). Then the noises are added to the same clean speech signal. This simulates a source of speech in broad-side position (i.e. in front of the virtual array), as in the first scenario, and a source of fan noise in end-fire position, with an inter-microphone distance of 4 cm ($4 \times 10^{-2} m \simeq 340 \frac{m}{s} \times 0.12 \times$

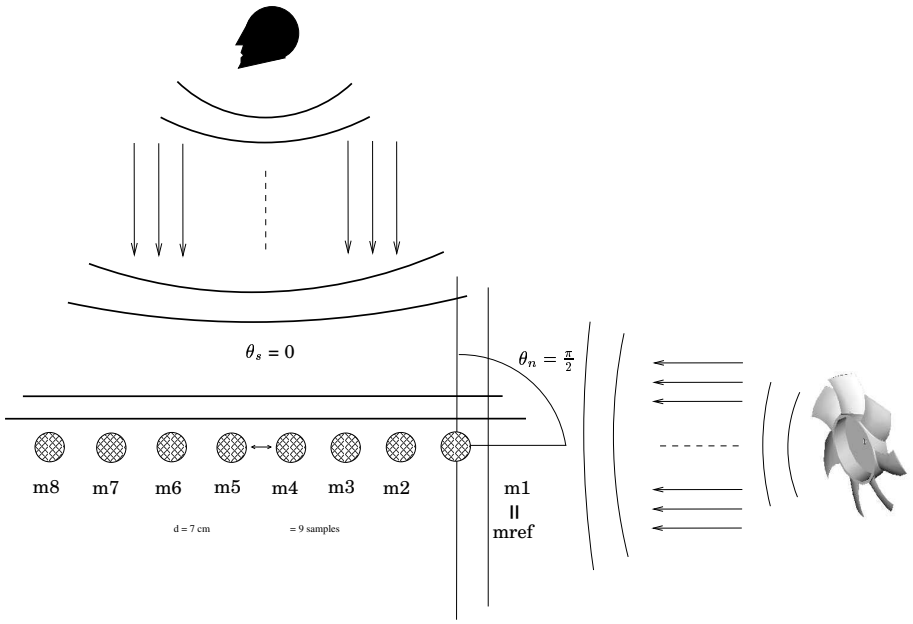


Figure 2.2. Second simulated scenario in which the clean speech source is broad-side to the array, while the recorded, synthetically added fan noise is end-fire to the array. The spherical propagation is approximated with a planar wave, because a far field is assumed.

$10^{-3}s$). The inter-microphone delay of 0.12 ms corresponds to the quantity τ_m introduced in 1.3.3. This scenario is depicted in figure 2.2.

Databases

English digits from the TI-digits [Leonard, 1984] database are chosen as a task through all this work, in which we intend to perform tests in severe noise conditions: from one side we have to deal with the complexity of the beamformer, thus we try to limit the complexity of the back end by limiting the size of the vocabulary. Furthermore, TI digits are a widespread corpus so that results of this thesis can be more easily checked by other researchers. The TI digits database contains English connected-digits utterances recorded with a close-talk microphone from US speakers with different genders, ages and accents. Specifically, both data from men (21-70 years old) and woman (17-59 years old) and data from children (6-15 years old) are available, but in this work only the adult part is used. Speech models are trained on a set of 8624 sentences, uttered by 55 men and 57 women, while the original (clean) test set contains 8700 sentences, uttered by 56 men and 57 women. The original TI database is sampled at 20 kHz. However, all the experiments in this chapter are made with signals sampled at 44.1 kHz: this is done because we want the adaptive filters involved in the algorithms to have the maximal resolution in the time domain. Consequently,

	Men	Women	#sentences	#digits
Train	55	57	8624	28336
WHOLETEST	52	52	1001	3271
SUBTEST	52	52	104	301

Table 2.3. Sets of the TI-digits database used to train and test the speech recognition system

a poly-phase filter is designed to upsample the TI database from from 20kHz to 44.1 kHz. Of this test set, we retained a selection of sentences, corresponding to the well-known set "A N1" of the Aurora2 [Hirsch and Pearce, 2000] corpus (from now on named "WHOLETEST"). The Aurora2 corpus derives from the TI corpus and was conceived to test speech recognition algorithms over the phone line: it consists of a subsampled version of TI (8 kHz) with real noise synthetically added. In this work we used the same files of the set A, but the noise inoculation is performed by us. Through all the experiments in this chapter, the bandwidth of noises was limited to [0-10kHz] prior to addition with the clean speech. This is done because the SNR measure is as much fair as noise affects only the frequency range where the signal affected by noise exists. Thus, white and fan noise were first generated and then low-passed (noise keeps on being white in the speech band). In these first experiments we use two test sets. The first is WHOLETEST, which contains 1001 utterances (a total of 3271 digits). The second (named "SUBTEST") is a subset of 104 utterances (a total of 301 digits): there is one utterance for each of the 52 male and female speakers and the amount of digits is roughly one order of magnitude less than the WHOLETEST. In conclusion, the TI-digits test set was scaled down twice: the SUBTEST intends to give relatively quick results with sufficient speaker variability, while the WHOLETEST can generalize the SUBTEST results on a well-known set of utterances. The kind of test sets used to generate preliminary results on the Limabeam algorithm are summarized in Table 2.3. For each phrase of the test set and for each virtual channel acquiring the phrase, the specific per-channel noise is added. Then beamforming, front-end parametrization and recognition are performed.

Front-end and Back-end

We use word models represented by 18 state left-to-right HMMs, with no possible skips. Output distributions are defined by 1-Gaussian probability density functions. The feature extraction in the front-end of the speech recognizer involves 12 Mel Frequency Cepstral Coefficients and the log-Energy together with their first and second derivatives, for a total of 39 coefficients. Features were computed every 10 ms, using a 25 ms sliding Hamming window. The frequency range spanned by the Mel-Scale FilterBank was limited to 100-7500 Hz to avoid frequency regions with no useful

signal energy. Cepstral Mean Normalization is applied.

2.3.2 Outperforming Delay and Sum with Oracle Limabeam

In the Oracle (i.e. supervised) version of this algorithm the Viterbi algorithm generates the optimal state sequence by aligning the beamformed output to the correct, known utterance. This implies that FIR filters generated by the optimization which uses the correct transcription should be optimal. By showing both the performance coming from the D&S and the Oracle Limabeam (called from now on OL), one can observe the amount of improvement that the latter technique can have on the former. We recall that there is a link between the two techniques: the OL starts from a D&S configuration, as stated by Equation 2.14. Because in the optimization process of Limabeam a likelihood function is maximized according to a gradient ascent criterion, the likelihood necessarily increases across the iterations. If the incremental improvement falls below a certain threshold, the algorithm stops and the best filters obtained up to the last iteration are used. In the next subsections we describe our first results, the purpose of which is to obtain performance from Oracle Limabeam superior to that of D&S:

1. The OL is initially tested in the first simulated scenario, i.e. with pure white noise: the OL shows no improvements both on single and multi-channel signals.

2. An attempt to constrain the FIR filter gain and another constraint, which “freezes” optimization in the silence periods is implemented, and few but inconsistent improvements are observed with a single channel.

3. If Power Spectrum instead of Magnitude Spectrum is used to compute the LFBE used to optimize the filters, the previous two arrangements become consistent and provide a slight improvement on a single channel.

4. Using a simulated array, the gain entailed by the D&S is large enough to increase recognition performance, and the OL cannot provide any improvement.

5. With pure white noise this turns to be exactly the result complying with the theory, as D&S beamforming maximizes the White Noise Gain (however, this condition very rarely happens in real environments).

6. Finally, by changing the noise with a more realistic fan noise (second simulated scenario), we are able to overcome D&S with OL, because we diverge from the optimal conditions for D&S

beamforming. We are able to test Unsupervised Limabeam as well, which we expect to give performance higher than D&S but lower than OL.

First results with White additive noise

We start testing both D&S and OL on the SUBTEST set, described in 2.3.1. In this scenario 1 or 8 microphones are simulated with white noise added to each channel at 5 and 10 dBs. We recall that the noise is white in the speech band of the original TI-digits database: its spectrum is depicted in Figure 2.3. In Table 2.4 results in *unit accuracy* are reported for the baseline, i.e. when the close-talk signal is available, and three other different configurations. The unit accuracy is computed as follows:

$$\frac{\#digits - (\#insertions + \#substitutions + \#deletions)}{\#digits} \cdot 100 \tag{2.16}$$

The baseline is the upper bound for performance, while degradation is observed as the amount of noise is increasing. Clearly results are not as expected: with one microphone only we apply OL and observe no improvements. With 8 microphones D&S beamforming dramatically improves performance, but OL, which we recall is initialized as the D&S configuration, cannot bring any improvement. We verified that most of the filters are not optimized, so a worse performance of OL comes from the few filters which are optimized in the LFBE domain. However, results show that this does not imply a better word recognition rate.

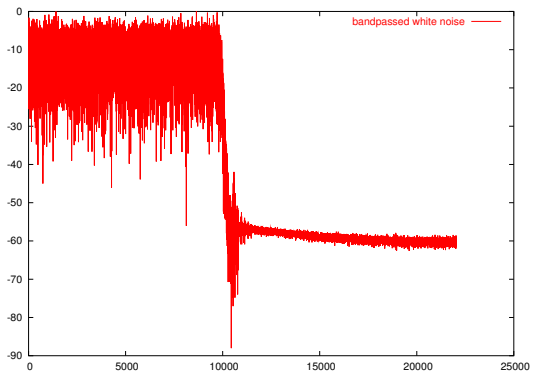


Figure 2.3. White noise band-passed to 10kHz

configuration	sum	OL
1 mic, 100 taps, 10dB	82.72	82.72
8 mics, 100 taps, 10dB	95.68	92.36
1 mic, 100 taps, 5dB	56.81	56.81
8 mics, 100 taps, 5dB	92.70	87.71
close-talk	99.11	

Table 2.4. First results with Oracle Limabeam, when speech is affected by white noise. The baseline (99.11%) is compared with a single noisy channel and a simulated 8-microphone array. Results in unit accuracy.

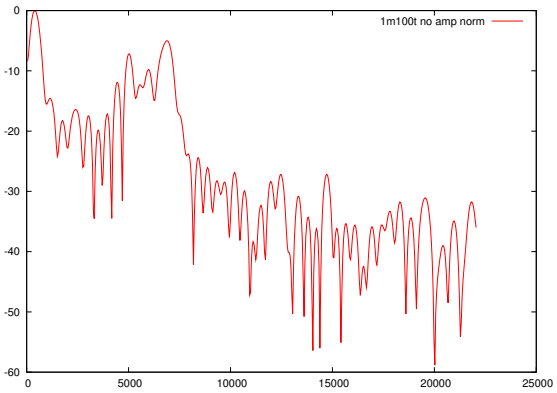


Figure 2.4. Oracle 100 taps FIR for 1 microphone only, no amplitude normalization.

Filter gain normalization and silence mismatch for single-channel

In order to understand why performances are so limited, we observe the frequency response of the FIR filters. Figure 2.4 shows a 100-taps FIR applied to one microphone only (for the sake of understanding the problem it was not necessary to test more than one microphone). The figure clearly shows that the FIR effect is to filter out every frequency except the band 200-600 Hz and the band around 7 kHz, which leads to an undesired very strong low-pass effect and to lower recognition performances. Furthermore, this behaviour was more evident when allowing the Conjugate Gradient algorithm to iterate with a threshold on the Likelihood of 0.01 (i.e. the algorithm stopped when the LLH function has increments lower than 1%), while we normally leave this threshold to 0.1.

An expected behaviour would be an improved recognition score with a lower threshold. Our conclusion is that the LLH is converging toward a poor local maximum. The algorithm should not cancel the speech frequency content in the useful bands, which roughly span the (100Hz-4000Hz) frequency range. A possible solution is to limit the attenuation of the filters at each step of the gradient ascent algorithm, by constraining the impulse response power of the FIR filter associated to microphone m to be normalized to 1:

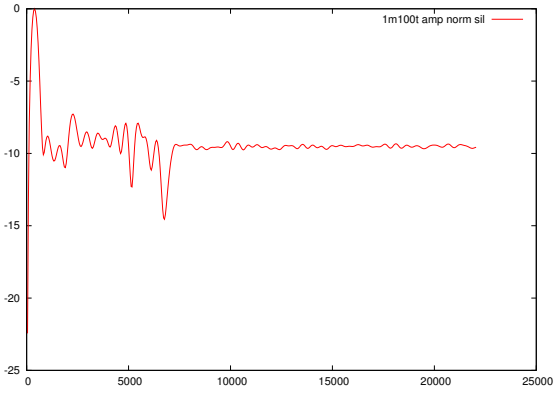


Figure 2.5. Oracle 100 taps FIR for 1 microphone only, gain normalization on the whole utterance.

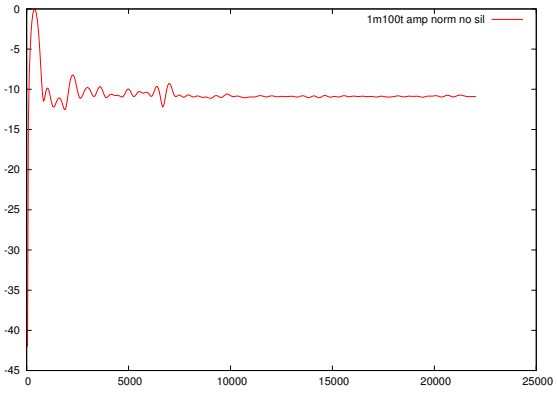


Figure 2.6. Oracle 100 taps FIR for 1 microphone only, gain normalization with Gradient computed on the speech part only.

$$\sum_{l=0}^{L-1} w_m[l]^2 = 1 \forall m : 0 \rightarrow M-1 \tag{2.17}$$

In practice, this prevents the gradient ascent routine to reach a point where the FIR coefficients are very close to zero, which would almost cancel the signal. A FIR filter obtained in the same environmental conditions, but using this constraint, is depicted in Figure 2.5: the figure shows that the heavy passband effect is still present, though only the band 200-600 Hz prevails.
We hypothesize that further improvements can be obtained if the mismatch between white noise and silence model is reduced. Indeed, we know that the gradient ascent tries to match the noisy features with the clean models, as stated by (2.12): however, non-speech frames contain silence at training time, but full-band noise at test-time. If there are more noise-only frames than noisy speech frames in an utterance (which is frequently the case in the TI-digits database), then the algorithm will tend to cancel even the bands with useful speech content. We decide to compute the Gradient in the speech part only: we have knowledge of the segmentation thanks to the Oracle state sequence/labels [3]. The effect is shown in Figure 2.6: the strong low-pass effect is still present and recognition scores are also decreased. Results are reported in Table 2.5: the behaviour when a front-end mismatch is present (i.e. in the "Magnitude" column) is not consistent with the fact that optimizing filters in the speech part should be beneficial. However, a slight improvement when normalizing the FIR gain is observed. However it will be seen that this contributes to the improvement, if joint with alternative definition of log-spectral features, as seen below.

So far the parameters used for recognition are Mel Frequency Cepstral Coefficients (MFCC) derived from Log-FilterBank Energies (LFBE) computed on a *Magnitude* Spectrum of the speech frame:

$$\mathbf{e}_t[\mathbf{w}] = \log_{10}\left(A|\mathscr{F}(\mathbf{y}_t[\mathbf{w}])|\right) \qquad \forall t : 0 \rightarrow T-1 \tag{2.18}$$

which is similar to Equation 2.5 except for the square modulus, which here is a simple modulus. This implies different features than the ones derived via Equation 2.5, where Log-FilterBank Energies are computed on the *Power* Spectrum: in fact the k-th entry of the feature vector corresponds to the k-th Mel Filterbank triangle and it is obtained as:

$$e_{t,k}[\mathbf{w}] = \log_{10} \sum_{b \in k} a_k[b]|Y_t[b,\mathbf{w}]| \qquad \forall t : 0 \rightarrow T-1 \tag{2.19}$$

[3] This would be possible in a real application only by using a Voice Activity Detector (VAD)

1 mics, 100 taps , 10dB White	Magnitude	Power
single channel	82.72%	90.70%
no gain norm	82.72%	90.70%
gain norm	83.39%	91.69%
gain norm, Gradient in speech part	81.73%	92.03%

Table 2.5. Unit accuracies with Oracle Limabeam when speech is affected by white noise. Results on SUBTEST for three different methods, using 1 microphone only and 100 taps for the FIR filter. Speech parameters are computed either with Magnitude or Power of the per-frame FFT.

which is different from:

$$e_{t,k}[\mathbf{w}] = \log_{10} \sum_{b \in k} a_k[b] |Y_t[b, \mathbf{w}]|^2 \qquad \forall t : 0 \rightarrow T-1 \tag{2.20}$$

In Limabeam the terms in the Jacobian matrix (see Equation 2.13) are computed by using Power Spectral components. Hence there is a mismatch between optimization and recognition features. It is worth noticing that this mismatch exists because of the sum term in both 2.19 and 2.20: without this weighted sum, the cepstrum derived without the use of FilterBank would be identical to within an additive term (coming from the square) and no difference in performance would be noticed. Table 2.5 shows that when no front-end mismatch is present (i.e. in the "Power" column), both the adjustments are effective, and we are finally able to outperform the D&S performance.

We kept the last two configurations as the best so far (with the Power for parametrization) and we extended the study to 8 microphones. Results are reported in Table 2.6: it shows that, while with 1 microphone the effect of the FIR filter was to remove some noise, with 8 microphone this task is already performed by the beamformer and the OL could not give any further gain. The reason for that stands in the efficiency of a microphone array: so far we have been looking for a filter able to improve recognition accuracies using one microphone only and without the need of any noise estimation technique nor a VAD, at least in OL. By properly adjusting the Limabeam algorithm one is able to do so, but the gain in word recognition rate is small compared to the spatial selectivity that a microphone array can provide. This is more evident in such a simulated framework, where the clean speech is synchronous across channels.

Clearly, while a high degree of noise reduction is achieved (53.5% of relative improvement with respect to single channel performance), the OL is not adding any additional benefit. We observed that the filter optimization algorithm is not able to move it apart from the initial guess (D&S configuration): the negative likelihood, expressed in function of the filter parameters, has already reached in this case a local maximum. Thus, around this position, the likelihood is always worse.

8 mics, 100 taps , 10dB White	Power
single channel	90.70%
Delay and Sum	95.68%
gain norm, Gradient in speech part	95.68%

Table 2.6. Unit accuracies with Oracle Limabeam when speech is affected by white noise. Results on SUBTEST for three different methods , using 8 mics and 100 taps per FIR filter. Speech parameters are computed either with the Magnitude or Power of the per-frame FFT.

While this behaviour may appear surprising at a first sight, it is confirmed by theory explained in 1.7.1 for a Delay and Sum beamformer. In fact the FIR filters are initialized according to Equation 2.14, or, equivalently, to Equation 1.44. When white noise is present, the noise field is purely non-coherent (see 1.4.1) and the weights of a microphone array already are in the optimal solution, which maximize the White Noise Gain of Equation 1.49. However, we recall that this theoretical aspect is based on a maximal SNR criterion: here we say that a SNR-based way of weighting filters is largely enough to compensate for white uncorrelated noise for recognition, and that a likelihood-based optimization of these filters is not beneficial in this specific case. Since in reality we are never in such a condition, this motivates us to seek performance in a more realistic scenario, which can be obtained by changing the type and direction of the noise.

Changing noise kind : directional fan noise

We have seen that Delay and Sum is effective when the additive noise field is diffuse or even non-coherent (depending on the amount of correlation between the channels). The noise reduction is obtained because the noise parts simply cancel each-other. This is due to inter-channel destructive interference. The OL tries to optimize filters which already give a very high likelihood, or, at least, which give a local maximum of the likelihood. One could argue that there is still a margin of improvement, since the baseline (i.e. clean speech recognized by clean speech) is 99.11% (see Table 2.4), but the mismatch can be simply due to the noise which is in the speech band and which cannot be removed by a simple FIR filter. In this Section, real fan noise is recorded and synthetically added to the clean speech channels, as described in Section 2.3.1. We recall that the fan noise was low-passed to the frequency range [0-10kHz], in order for the SNR to be consistent with the bandwidth of the TI-digits database (see Figure 2.7). However, the recorded fan noise already presents a relatively strong coloration, and is similar to a traditional "brown" noise.

With fan noise, channels are no more uncorrelated. We simulate a source of noise left to the array (when facing the array) in end-fire position. The inter-channel delay is 5 samples ($0.11 \cdot 10^{-3}s \cdot 44100\frac{samples}{s} \simeq 5samples$). For these experiments, no amplitude normalization nor Gradient

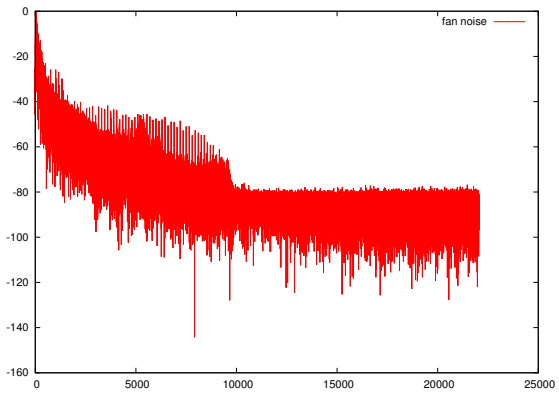


Figure 2.7. Fan noise band-passed to 10kHz

"freezing" was used, because we discovered them to be useful with a single microphone only. The optimized FIRs are shown in Figure 2.8 when 10 taps are optimized, and in Figure 2.9 when 100 taps are optimized. In order to be comparable, the FIRs in this section refer all to the same utterance, which is "six four six", because filters computed using different transcriptions are different. We can notice that the frequency response depicted in Figure 2.9 has changed, compared to Figure 2.6. Low frequencies, where the directional fan noise is present most, are canceled out, some sub-bands of the speech band are enhanced and a constant -7dB gain is applied outside the useful speech bandwidth. This is intuitively correct, because the filter should be neutral where no information is available. Notice that the figure depicts only one filter, relative to one channel, because all the filters are very similar. However, if less taps are optimized, as in Figure 2.8, the shape is of course smoother, the theoretically beneficial (due to the type of noise spectrum) high pass effect is present, but the filters are still slightly different in shape across channels. The more taps are used, the more the per-channel frequency responses of the filters obtained tend to be alike. The reason for that can be twofold: first, the channels are virtually close, and observe the same kind of noise to within a single phase term (the inter-channel delay); second, no reverberation is present in this simulation, so FIR cannot compensate for additional delays, potentially different for each channel, which would determine a different shape.

Based on this kind of filters, we fix the filter length (10 taps) and compare results on the SUBTEST set for the two different noise conditions investigated so far (fan and white noise) and for different SNRs. Noise Signals are properly scaled in amplitude to match the required SNR. Results are shown in Table 2.7.

The accuracies report and confirm that when speech is affected by a non-coherent noise (right column), the OL is not beneficial. Conversely, when speech is affected by a more coherent noise (left

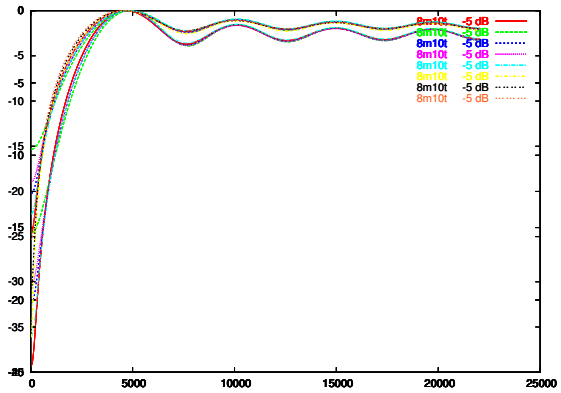


Figure 2.8. Oracle 10 taps FIR for 8 microphones when directional fan noise is at -5 dB. The 8 FIRs have the same behavior. This is probably due to the lack of reverberation.

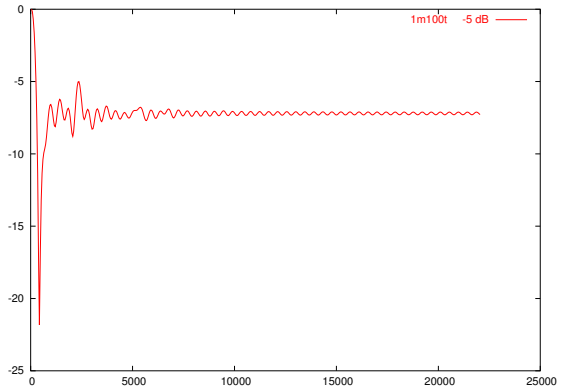


Figure 2.9. Oracle 100 taps FIR for 8 microphones when directional fan noise is at -5 dB (just one filter is shown). The 8 FIRs have the same behavior.This is probably due to the lack of reverberation.

8mics 10 taps, SUBTEST	fan, 10 kHz	white, 10 kHz
10 dB, D&S	97.67%	95.68%
10 dB, Oracle	97.67%	95.68%
5 dB, D&S	95.35%	92.70%
5 dB, Oracle	95.68%	92.70%
0 dB, D&S	78.41%	73.21%
0 dB, Oracle	85.71%	73.21%
-5 dB, D&S	55.48%	54.55%
-5 dB, Oracle	68.44%	54.55%

Table 2.7. Comparison between fan noise and white noise added to clean speech of the SUBTEST set and processed with Oracle Limabeam and Delay and Sum beamforming, in the band 0-10 kHz. 8 mics and 10-taps FIR are used. Environmental conditions vary from 10 to -5 dB.

column), the OL is able to achieve a certain amount of improvement over the D&S configuration. This improvement is function of the SNR and, unlike other techniques aiming to compensate for environmental effects, it is larger at lower SNR (29.1% and 33.8% at -5[4] and 0 dB respectively). We observed that filters generated at low SNR conditions are much more high-pass than the ones generated at higher SNR conditions. The high-pass effect is particularly beneficial after a D&S (we recall that here a simple sum is done, since the clean speech signal is synchronous across channels), because D&S always introduces a low-pass effect. The low-pass effect is not disturbing any human listener up to a certain degree, but affects speech recognizer performances. Regardless of what happens at higher frequencies, the more the low pass effect is compensated by the FIR filter, the higher the word recognition rate will be.

2.4 Testing Unsupervised Limabeam

We have seen that OL significantly improves recognition performance over D&S with a non-white noise. We recall that the Oracle version of Limabeam aligns the features of the beamformed signal with the correct transcription. However, in a real application this transcription is not available. As mentioned in 2.1.3 the Unsupervised Limabeam (UL) differs from the Oracle Limabeam because the UL uses the transcription generated, after a first recognition step, as an initial guess. This guessed transcription, possibly not the correct one, is fed to the Viterbi alignment. Of course, if the initial guess turns to be the correct transcription, then the OL and the UL are equivalent and the FIR filters sets generated are the same. Conversely, for other alternative transcriptions, different alignments are produced and, as a consequence, the filters sets change. We tested UL when the SNR is -5 dB for two reasons: first, because it is a very hard condition to match the OL filter gain dynamics; second, the margin between the D&S and the OL is the highest experienced so far (from 55.48% to 68.44% on SUBTEST) and it would be interesting to see what is the improvement of UL with respect to D&S and what is still the gap between UL and OL, this gap leaving space to future improvements. Results, reported in Table 2.10, are shown for the SUBTEST in the first, upper table: the Unsupervised Limabeam adds about 26% relative to the D&S, while the Oracle counterpart adds another 6%. As the details about the type of errors (insertions, substitutions, deletions) show, this improvement is concentrated mostly in insertions. Thus, a more consistent validation must be carried on a larger test set. The lower table confirms that on WHOLETEST there is a consistent improvement coming from UL. Furthermore, we analyze the shape of the FIR

[4]This very low SNR is measured between 0 and 10 kHz. A more realistic measure can be done in the frequency range spanned by the Mel Filterbanks

8 mics, 10 taps	SUBTEST		
-5dB fan	accuracy	errors (I,S,D)	RI
delay and sum	55.48%	(82, 46, 8)	-
Unsupervised	65.78%	(62, 35, 6)	23.1%
Oracle	68.44%	(51, 38, 6)	29.1%

8 mics, 10 taps	WHOLETEST		
-5dB fan	accuracy	errors (I,S,D)	RI
delay and sum	60.49%	(760, 472, 55)	-
Unsupervised	70.92%	(508,384,55)	26.3%
Oracle	73.26%	(452, 351, 68)	32.3%

Table 2.10. Unit accuracies with Unsupervised Limabeam on both SUBTEST and WHOLETEST, using 8 mics and 10 taps per FIR filter.

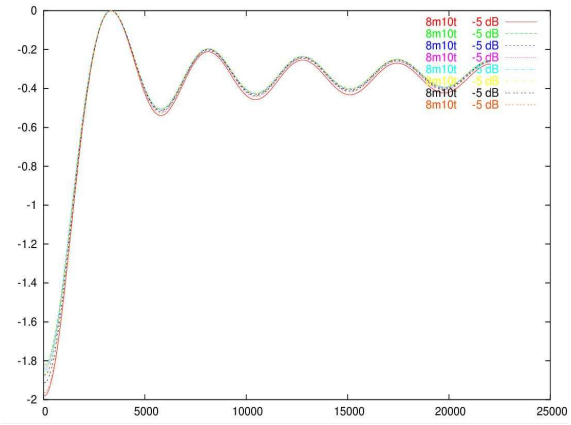


Figure 2.10. Unsupervised 10 taps FIR for 8 microphones when directional fan noise is at -5 dB. The 8 FIRs have the same behavior.

filters, as we did for the OL case. By looking at the optimized FIRs of Figure 2.10, we can notice that the behavior across the bands for UL and OL is similar. The only difference is that in the unsupervised case the filters seem to be "stretched down": in fact, by comparing Figure 2.10 with Figure 2.8, the attenuation of low bands, where the fan noise is mostly concentrated, is 15dBs more efficient for the OL filters. This is also observable in Figure 2.11, where the Unsupervised FIR was plotted rescaled to match the dynamics of OL filter gain. In the case 100 taps are used, the same phenomenon is present, as can be observed by comparing Figure 2.12 with Figure 2.9.

2.4.1 Conclusions about Oracle and Unsupervised Limabeam

We have seen that the supervised version of the Limabeam algorithm, called Oracle, is able to significantly improve recognition performance at very low SNR when a controlled environment is

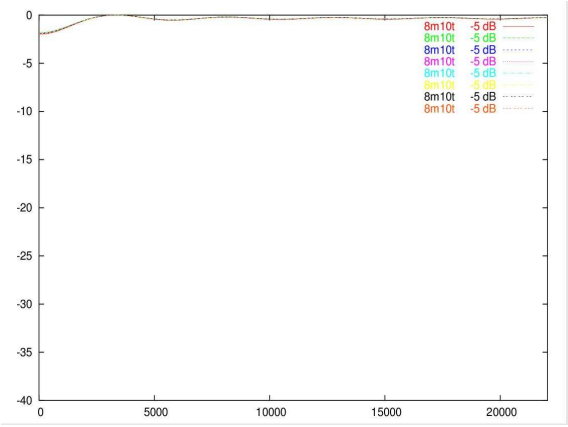


Figure 2.11. Unsupervised 10 taps FIR for 8 microphones when directional fan noise is at -5 dB. (Rescaled).

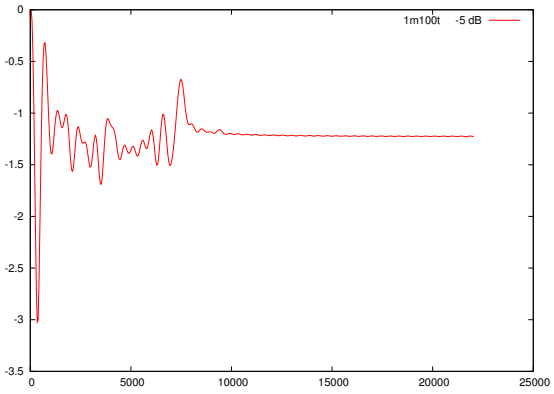


Figure 2.12. Unsupervised 100 taps FIR for 8 microphones when directional fan noise is at -5 dB.

simulated. This improvement is not observable when using pure additive white noise, because the D&S seems to be the best method even from a recognition point of view. However, real-time data will contain inter-channel delays, which are caused by the fact that the speaker is not necessarily in front of the array. As we will see, these delays can be compensated using a speaker localization technique. Alternatively, they could be automatically computed by a gradient-ascent procedure. We also show that Unsupervised Limabeam can get better performances over Delay and Sum Beamforming. Nevertheless, a major effort can be done to get performances close to the Oracle Limabeam. One key point is that the Unsupervised Limabeam is *de facto* an adaptation algorithm which optimizes FIRs on the *hypothesized* transcription. A gain over the Delay and Sum is only possible if the accuracy is, broadly speaking, higher than 50%, because there is more data better recognized, on which FIRs can be adapted, than data badly recognized. In the next chapter we propose a N-best approach to the FIR filter set optimization. We show that it is possible to overcome both Unsupervised Limabeam and Oracle Limabeam, thus significantly improve recognition performance.

Chapter 3

N-best Unsupervised Limabeam

In this Chapter we outline and detail the proposed algorithm, the N-best Unsupervised Limabeam [Brayda, 2006c]. It will be shown how the likelihood of a hypothesized transcription can be optimized and we will give two very representative examples. We will also explain why the proposed approach overcomes both Unsupervised and Oracle Limabeam. Particular emphasis will be given to two phenomena: the acoustical confusability between transcriptions, which is exploited by the proposed algorithm to automatically re-rank a set of initial guesses produced by the decoder, and the capability of our algorithm of recovering errors made at a first recognition step. The N-best UL is first tested in the simulated environment of the previous chapter, where fan noise was used, then its performance is evaluated in a real environment: in both cases improvements are shown.

3.1 Principles of N-best approach to the Limabeam algorithm

The Limabeam algorithm, presented in Chapter 2, is one of the techniques allowing the recognizer to exchange information with the beamformer. The scheme of such exchange has been described in Chapter 1: now that the algorithm is understood, we know that the feedback depicted in Figure 1.9 is a FIR filter set, which drives the beamformer toward a possibly maximally likely solution. However, the amount of information in the feedback can be increased in order to reach higher performance. In this work we propose an alternative scheme of feedback Speech Recognition. The proposed technique relies on the application of Limabeam and can be considered a generalization of this algorithm. We exploit the fact that driving the beamformer with filters estimated on a first hypothesized transcription is not necessarily the best solution. Instead, we try to estimate the best filters from many competing transcriptions, then we select the best transcription according to a ML

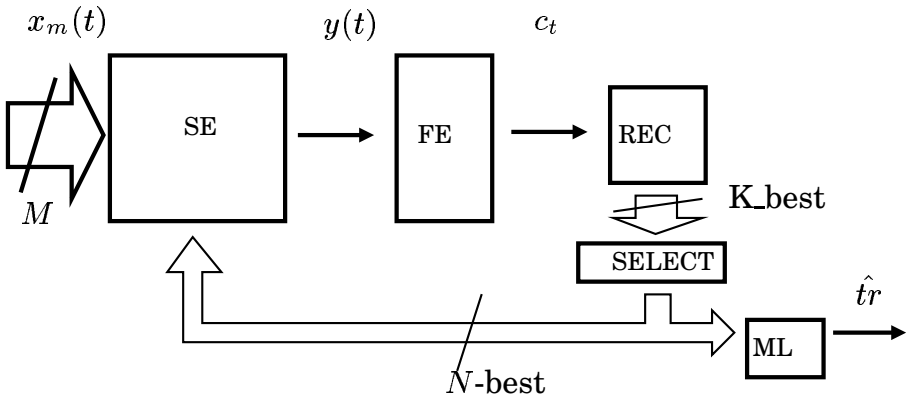


Figure 3.1. Proposed Multi-Hypotheses Feedback Array-based Speech Recognition system

criterion. A macro-block of the proposed scheme is depicted in Figure 3.1.

The system works partially as a classical chain formed by Speech Enhancement (SE), Feature Extraction (FE) and Recognition (REC). The feedback element is, as in the case of Limabeam formed by transcriptions. However, from the REC block we extend the feedback from just one hypothesized transcription to the first K_best hypothesized transcriptions. In the $SELECT$ block we reduce the number of such hypotheses to a smaller set where each transcription is as much different as possible from the others. This will be clarified in Section 3.1.4. Such set contains N-best transcriptions, which are fed to the optimizers. Based on this new set, N-best parallel optimizations are performed in the SE block and the multi-channel signal is re-beamformed, as it happens in the Limabeam optimization phase. The difference is that we increased the number of optimization to N-best: our intention is to put all the transcriptions in competition and to get a new maximum likelihood transcription. Once the optimization has converged, features are re-extracted and recognition is performed. While with conventional Limabeam at this point we have just one hypothesis, which is assumed to be the best one, in our case we have to chose between N-best new hypotheses. The ML block is responsible to chose the best one, which for us can be the one having the maximum likelihood and the final transcription $\hat{tr}$ is generated. We will show that the proposed approach, which we call N-best Unsupervised Limabeam, is able to improve significantly the recognition performance over D&S, Unsupervised Limabeam and Oracle Limabeam.

3.1.1 N-best speech recognition

N-best recognition is known to be a useful approach when using progressive knowledge sources in multi-pass search strategies [Huang, 2001]. When performing ASR in noisy environments, it is well known that the correct transcription could be found in a very low position in an N-best list. One

can expect that the use of a microphone array can raise the position of the correct transcription in the N-best list. Let us assume to have a system that outputs the N-best hypotheses for known sentences and thus is able to select the correct one. Figure 3.2 depicts the performance of the system when one or eight microphones (in this case D&S is applied) are used in our experimental conditions. It is evident that the use of the array increases the amount of correct sentences among the N-best, i.e. the chances of picking up the correct transcription. This happens because all the candidates are more intelligible, less noisy, or better matching the models, depending on the array processing method adopted. At this point one would like the correct transcription be the first choice rather than the n-th. One way to do that is to re-process each hypothesis until it better matches the models against which it will be compared.

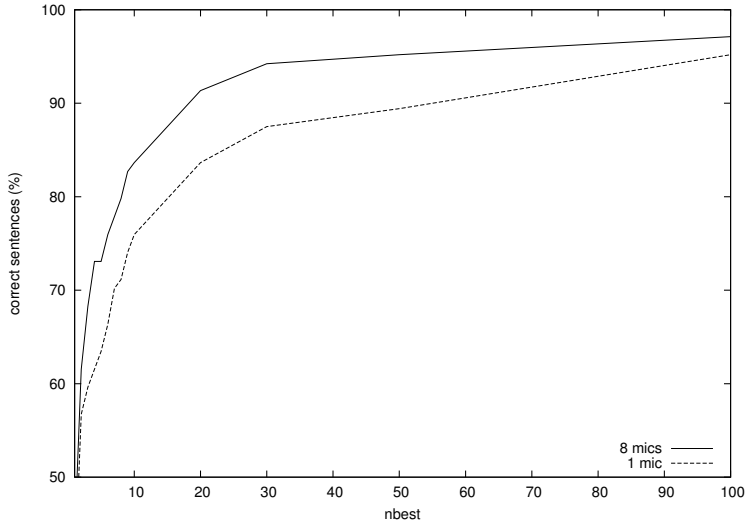


Figure 3.2. Percentage of correct sentences found by a system that recognizes the N-best hypotheses over transcribed sentences in a noisy environment. With more microphones the correct sentence is "pushed up" in the first alternatives. This result represents an upper-bound for system performance reported in the following.

3.1.2 N-best applied to Limabeam

In this section we detail the different steps of our original N-best approach applied to Limabeam. The equations of the main steps will be tagged with the $STEP$ label. We have seen that applying Unsupervised Limabeam improves recognition accuracy: this algorithm is effective because a filter set is optimized even when the optimization is performed on a *wrong* transcription. The algorithm has been already described in Chapter 2, but we recall here the main equations for the sake of clarity. First the features of the beamformed (via D&S) input are recognized and the most likely transcription is output:

$$\hat{tr}_1 = \arg\max_{tr} P(\mathbf{c}[\mathbf{w}_{init}]|tr) \tag{3.1}$$

Equation 3.1 is equivalent to (2.7), but here we neglected the *a priori* probability $P(tr)$, which does not change through all the process, we put $\mathbf{w} = \mathbf{w}_{init}$ and $\hat{tr} = \hat{tr}_1$ to emphasize that filters start from an initial configuration (in this case the coefficients of a D&S beamformer) and that this is a first recognition step, respectively. The optimized filter set is then computed via:

$$\mathbf{w}_{opt} = \arg\max_{\mathbf{w}} P(\mathbf{e}[\mathbf{w}]|\hat{tr}_1) \tag{3.2}$$

Finally, features are re-beamformed with the new filter set $\mathbf{w}_{opt}$ and a second recognition step is performed:

$$\hat{tr}_2 = \arg\max_{tr} P(\mathbf{c}[\mathbf{w}_{opt}]|tr) \tag{3.3}$$

If we define a similarity measure $\mathscr{S}(x, y)$ between transcriptions x and y, then we can represent the improvement of Limabeam as the average similarity on a test set:

$$\overline{\mathscr{S}(\hat{tr}_2, tr_c)} > \overline{\mathscr{S}(\hat{tr}_1, tr_c)} \tag{3.4}$$

where tr_c is the correct transcription and $\hat{tr}_1$,$\hat{tr}_2$ are the hypothesized transcriptions at the first and second recognition step (optimization is made between the two steps). $\mathscr{S}$ can be the Digit Accuracy, the Percent Correct, or any measure related to the well known Levenshtein distance between strings x and y. Results show that, on average on the test set, the optimization process leads, to a better result (in the Oracle Limabeam $tr_1 = tr_c$). If a possibly wrong transcription is given (Unsupervised Limabeam, $tr_1 \neq tr_c$ in general), the pertinency of the result depends on how much the wrong hypothesized transcription is dissimilar (in terms of $\mathscr{S}$) from the correct one. This is because Limabeam is essentially *adaptation*. The adaptation was proved to be beneficial only with one step, i.e:

$$\overline{\mathscr{S}(\hat{tr}_3, tr_c)} < \overline{\mathscr{S}(\hat{tr}_2, tr_c)} \tag{3.5}$$

In the following we will not iterate optimizations as well. The Unsupervised Limabeam seeks to maximize the likelihood of some features, given a hypothesized transcription. However, we focus our attention on the two likelihoods:

$$P(\mathbf{c}[\mathbf{w}_{opt}]|\hat{tr}_1) \tag{3.6}$$

$$P(\mathbf{c}[\mathbf{w}_{opt}]|tr_c) \tag{3.7}$$

Equation 3.6 is the likelihood of the optimized features justifying the new transcription after a first recognition step, while (3.7) uses the same features, but it is evaluated on the correct transcription (in practice we can compute the second likelihood only for evaluation, since tr_c is of course unavailable).

Proposition 1 *These two likelihoods can be further increased*

In fact they are not "absolutely maximal": for absolutely maximal we mean that (3.6) is maximal with respect to all possible transcriptions: however, it is by definition only maximal with respect to *one* transcription (tr_2). We point out that the UL works because there are transcriptions maximizing *simultaneously* (3.6) and (3.7). This means that tuning (i.e. optimizing) a set of features with a specific hypothesized transcription ($\hat{tr}_1$) drives the features to the correct transcription (tr_c). However, *this does not mean that the first hypothesized transcription is the best to perform such operation*. Specifically, we seek for:

- a filter set $\mathbf{w}_{opt,x}$ with a transcription $\hat{tr}_{1,x}$ so that

$$P(\mathbf{c}[\mathbf{w}_{opt,x}]|\hat{tr}_{1,x}) > P(\mathbf{c}[\mathbf{w}_{opt,1}]|\hat{tr}_1) \tag{3.8}$$

- a transcription $\hat{tr}_{2,x}$ so that:

$$P(\mathbf{c}[\mathbf{w}_{opt,x}]|\hat{tr}_{2,x}) > P(\mathbf{c}[\mathbf{w}_{opt}]|\hat{tr}_2) \tag{3.9}$$

- for which it is verified that:

$$P(\mathbf{c}[\mathbf{w}_{opt,x}]|tr_c) > P(\mathbf{c}[\mathbf{w}_{opt}]|tr_c) \tag{3.10}$$

where x denotes the alternative index (i.e. the x-best) output from the Viterbi algorithm. The first two points mean that we seek alternative transcriptions and filters, other than the first hypothesized and optimized, for which the likelihood justifying them can be higher than what is found by

simple UL. The third point means that the sought filter set must increase the likelihood to the correct transcription more than what is obtained via UL. Because acoustical confusability positively influenced the optimization in Limabeam, the most natural set of transcriptions where to find the couple $(\mathbf{w}_{opt,x}, \hat{tr}_x)$ is the *list of the K_best transcriptions* output from the first recognition step of Equation 3.2. This is equivalent to finding the set G_{K_best} of K_best transcriptions:

$$STEP1 \qquad G_{K\_best} = \{\hat{tr}_{1,k}\} = \arg\max_{tr} P(\mathbf{c}[\mathbf{w}_{init}]|tr) \qquad \forall k : 1 \rightarrow K\_best \tag{3.11}$$

where the argmax operator was defined (it returns the first K_best maxima of the operand which it refers to) and the index $1, k$ means that we are treating the $k - best$ alternative at the recognition step 1. The likelihoods evaluated on these transcriptions are normally given by the Viterbi algorithm in a decreasing order, which implies that:

$$P(\mathbf{c}[\mathbf{w}_{init}]|\hat{tr}_{1,1}) \geq P(\mathbf{c}[\mathbf{w}_{init}]|\hat{tr}_{1,2}) \geq \cdots \geq P(\mathbf{c}[\mathbf{w}_{init}]|\hat{tr}_{1,K\_best}) \tag{3.12}$$

where $P(\mathbf{c}[\mathbf{w}_{init}]|\hat{tr}_{1,1}) = P(\mathbf{c}[\mathbf{w}_{init}]|\hat{tr}_1)$ would be chosen by the UL, as it is the highest likelihood. However, the K_best list is on this task generally very long, because it includes all possible repetitions of silence units in-between digits. Thus, we reduce its cardinality by extracting a sub-set G_{N-best} of N-best transcriptions from the G_{K_best} set:

$$STEP2 \qquad G_{N-best} = \{\hat{tr}_{1,n}\} \subseteq G_{K\_best} = \{\hat{tr}_{1,k}\} \qquad N - best \ll K\_best \tag{3.13}$$

We then perform, *independently*, N-best optimizations. This is equivalent to finding a set of filters W_{N-best}:

$$STEP3 \qquad W_{N-best} : \{\mathbf{w}_{n,opt}\} = \arg\max_{\mathbf{w}} P(\mathbf{e}[\mathbf{w}_{init}]|\hat{tr}_1, n) \qquad \forall n : 1 \rightarrow N - best \tag{3.14}$$

The purpose of this parallel optimization is to increase all the likelihoods relative to each n hypothesized transcription. This implies that:

$$\begin{gathered} P(\mathbf{c}[\mathbf{w}_{opt,1}]|\hat{tr}_{1,1}) \geq P(\mathbf{c}[\mathbf{w}_{init}]|\hat{tr}_{1,1}) \\ \vdots \\ P(\mathbf{c}[\mathbf{w}_{opt,N-best}]|\hat{tr}_{1,N-best}) \geq P(\mathbf{c}[\mathbf{w}_{init}]|\hat{tr}_{1,N-best}) \end{gathered} \tag{3.15}$$

We can notice that the equation above implies that there exist a set G_{Q-best} of $Q - best$ tran-

scriptions:

$$G_{Q-best} = \{\mathbf{w}_{opt,q}\} \subseteq G_{N-best} \qquad Q - best \leq N\_best \tag{3.16}$$

for which the following inequality holds:

$$P(\mathbf{c}[\mathbf{w}_{opt,q}]|\hat{tr}_{1,q}) \geq P(\mathbf{c}[\mathbf{w}_{opt,1}]|\hat{tr}_{1,1}) \tag{3.17}$$

where $P(\mathbf{c}[\mathbf{w}_{opt,1}]|\hat{tr}_{1,1})$ would be obtained by applying UL. This implies that, after optimization, the strict decreasing order of the likelihoods, is no more a constraint, and that a new maximum value of the likelihood can be found, which demonstrates Proposition 1. This in turn implies that the first hypothesized transcription does not necessarily generate the highest likelihood after optimization and constitutes the limit of the Unsupervised Limabeam. Taking advantage of the last equation, we recall that all that we wish is getting closer to the models of the *correct* transcription. In fact, the LHS of Equation 3.17 is a maximum with respect to $\hat{tr}_{1,q}$, not with respect to tr_c. Performance will be higher than UL if we find, among the N-best transcriptions (not the $Q - best$, but this will be clarified later), a final set of filters $\mathbf{w}_{opt,final}$ such that:

$$P(\mathbf{c}[\mathbf{w}_{opt,final}]|\hat{tr}_c) \geq P(\mathbf{c}[\mathbf{w}_{opt,1}]|\hat{tr}_c) \tag{3.18}$$

Of course both the quantities are not available in practice. In order to get a valid approximation, the N-best optimized features are recognized:

$$STEP4 \qquad G_{2,N-best} = \{\hat{tr}_{2,n}\} = \arg\max_{tr} P(\mathbf{c}[\mathbf{w}_{opt}]|tr) \qquad \forall n : 1 \rightarrow N - best \tag{3.19}$$

At this point, the N-best list has changed: new transcriptions are output, together with their likelihoods $P(\mathbf{c}[\mathbf{w}_{n,opt}]|\hat{tr}_{2,n})$. The next step (STEP5) finds the solution f^*. In the case our algorithm is successful we can expect:

$$P(\mathbf{c}[\mathbf{w}_{opt,final}]|tr_c) \approx P(\mathbf{c}[\mathbf{w}_{opt,f^*}]|\hat{tr}_{f^*}) \tag{3.20}$$

where f^* is the best "elected" transcription, which can be found with a ML evaluation:

$$STEP5 \qquad \hat{tr}_{f^*} = \hat{tr}_{2,f^*} = \arg\max_{n \in G_{2,N-best}} P(\mathbf{c}[\mathbf{w}_{n,opt}]|\hat{tr}_{2,n}) \tag{3.21}$$

which in practice means selecting, among the transcriptions of the $G_{2,N-best}$, the one with the

highest value. Experimentally we show that on average:

$$\overline{\mathscr{S}(\hat{tr}_{2,f^*}, tr_c)} > \overline{\mathscr{S}(\hat{tr}_{2,1}, tr_c)} \tag{3.22}$$

meaning that $\hat{tr}_{2,f^*}$ is closer to the correct transcription than $\hat{tr}_{2,1}$, which is equivalent of saying that performance are higher than UL. The system we propose is depicted in Figure 3.3. The signal coming from a microphone array is processed via conventional D&S, then Feature Extraction (FE) and a first recognition step is performed (REC). The HMM recognizer generates N-best hypotheses. For each hypothesis and in parallel, the Limabeam algorithm is applied: first a Viterbi alignment is performed (switch to 1: ALIGN) and fixed, then FIR coefficients are adaptively optimized via Conjugate Gradient (switch to 2: OPT). After convergence, the N-best features are recognized (switch to 3: REC) and another set of new transcriptions is produced. Finally, the last block compares the new N-best Log-LikeliHoods (ML) choosing the highest and the recognized sentence is produced.

3.1.3 Automatic re-ranking

In the previous section we showed that it is possible to find a transcription f^* which is closer to the correct one than the first-hypothesized transcription. This algorithm is even efficient when applied to a list where the correct transcription is *not* the first one, which is always the case when a recognizer fails. Indeed, the effect of optimizing features in parallel relaxes the constraint on the order of the likelihoods produced after a first recognition step. The likelihood of each single competing likelihood increases during optimization. The main point is that

Proposition 2 *The relative improvement in likelihood of competing hypotheses is not the same.*

Thus, the parallel optimization automatically *re-ranks* the N-best list. This is very useful when maximizing the likelihood of *maximum recognition rate hypotheses*: in fact, this gives the chance to tr_c to be "pushed up" in the new N-best list and eventually to become the best, f^*. Our experiments will show that the rank of the correct transcription always improves, but that it does not always reach the first place.

3.1.4 Selection of the N-best transcriptions: the silence models

In Section 3.1.2 K_best hypotheses are kept. Several of them contain many silence models either in front or at the end or even in the middle of the sentence. We will discard several hypotheses that differs by a number of inserted silences. The K_best hypotheses output from the Viterbi algorithm

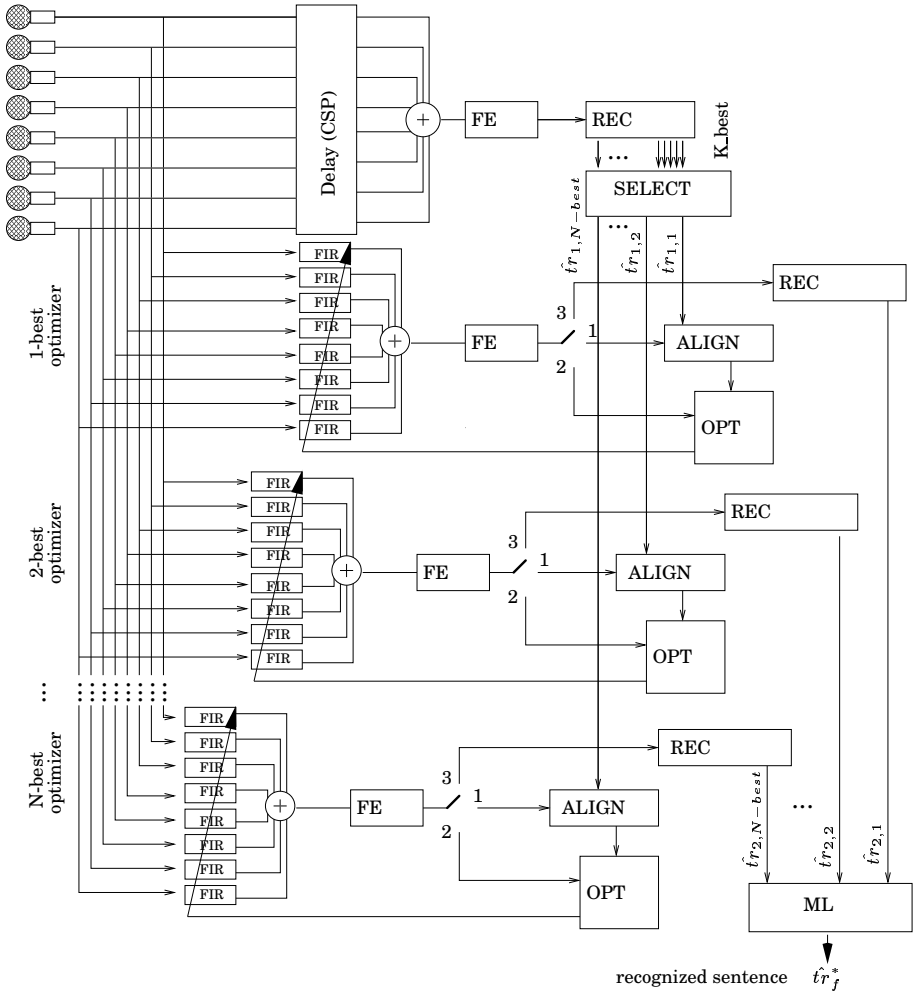


Figure 3.3. Block diagram of the N-best Unsupervised Limabeam.

cannot be fed as they are for the optimization: in our experiments the HMM "sil" (silence) model is formed by three emitting states, and the "sp" model (the short pause) is formed by one emitting state. This state is identical to the middle state of "sil" and it is thus a short version of the silence model. If we had to consider, for each transcription made of D digits, all the possible positions of the "sil" (which are four: head, tail, head and tail or simply absent) and of "sp" (which are $D-1$), then the complexity, initially linear on the number of hypotheses $N-best$, would become:

$$T(N-best) = O(4N-best) \tag{3.23}$$

Finally, if also the "sp" word is included, the complexity explodes to:

$$T(N-best) = T(K\_best) = O(2^{D+1}N - best) \tag{3.24}$$

where the exponent collects all the $D-1+2$ combinations. Clearly the problem is not tractable in the second case for D sufficiently high. [1] Because a policy is needed, we decide to take possible silences out, then to append two "sil" in head and tail of each sentence. This is done for two reasons: first, to comply with the Aurora labels, because the clean test data (associated with a correct transcription) do have a silence period in head or tail. In the following we want to compare the N-best unsupervised Limabeam performance with the Oracle Limabeam performance. If the correct sentence is in the N-best list, then it will be equal to the Oracle transcription. Second, by forcing silences at the beginning or at the end of the utterance, no more than three frames will be assigned by the Viterbi alignment (45 ms with the current window size and frame rate). This amount of time is much smaller than the average silence duration before and after speech and this procedure is thus fair. We will see that it is also beneficial for the N-best UL performances. Finally, it is evident that the reduction generated many identical hypotheses; however, their likelihood is in general different: in this case it is reasonable to assume that each competing transcription is represented by its maximum value of likelihood. In synthesis, the steps accomplished in the $SELECT$ block of Figure 3.3 are:

1. Take the K_best outputs of the Viterbi algorithm as they are.
2. Sort the transcription by decreasing likelihood.
3. Cut any occurrence of "sil" and "sp".
4. If there are multiple occurrences of the same transcription, take the one with highest likelihood (best representative basis).
5. Append sil to the beginning and to the end each transcription of the reduce a N-best set.

3.2 Experimental results

The experiments described in the following rely on the same setup as outlined in Chapter 2. First, we show an example where the correct transcription is present in the N-best list: its likelihood is maximized and the correct transcription becomes the chosen f^*. We then measure performances of the N-best UL at -5 dB and study their behaviour as a function of the length of the N-best

[1] However, it would be tractable for recognition tasks with very few words per sentence or isolated ASR.

list. After that, we point out that aligning an Oracle transcription on the close talk or on the beamformed output does not give significant differences in results: the same cannot be guaranteed in more reverberant environments. Experiments at all dBs are then presented and the mutual relation between the WRR and likelihood relative improvements is studied. Another example of auto-ranking is given, where this time an acoustically confusable transcription is the f^*, while the correct transcription, though pushed up in the list, generates worse recognition results. An analysis of ML data in the time, spectral and cepstral domain is provided, together with an insight on the magnitude and phase response of ML filters. The evolution of the accuracies across three methods is investigated, and the amount of confusability in the best optimizing transcription is measured via Levenshtein distance. Performances in a real environments, with surrounding noise sources and a SNR of 0dB, are then measured. We conclude this Chapter by deriving a new performance upper bound and by discussing both the most suitable Front-end domain and dimension of feature vectors which are optimal for obtaining the best performances as a function of the overall system complexity.

3.2.1 Oracle transcription "pushed up"

In this Section we give an insight of how the likelihoods evolve in the proposed system when the number of competing hypotheses is limited to four as an example. The purpose is also to describe equations announced in Section 3.1.2. We show that the correct transcription tr_c is "pushed up" in the N-best list after N-best UL is performed. We consider the specific case of the utterance "eight" (file FLP_8B according to the Aurora2 notation), which is recorded when the fan noise is at -5 dB, then beamformed. Table 3.1 shows the evolution of the likelihood values through all the steps of the proposed algorithm. The values shown here are taken directly from the output of the HVite routine of the HTK toolkit, and represent the average likelihood per frame computed on the cepstral models used for recognition. The beamformed features of the utterance "eight" are recognized a first time (Equation 3.14); the generated transcriptions are shown in column i. Their likelihood justifying the recognized transcriptions are shown in column ii: they correspond to all the RHS of Equation 3.15 and are of course ranked in decreasing order (column iii). After N-best alternatives are extracted, silence is appended to their beginning and end (column iv). Notice that we directly depicted the first N-best transcriptions, not the first K_best, many of which have been eliminated. With these modified transcriptions, a Viterbi alignment is performed: the current likelihoods are visible in column v, in blue and bold. One could argue that they are slightly worse, but this is not affecting negatively the steps after. Optimization is done on each competing transcription, and the

blue and bold values of column vii are reached (they are the LHS of Equation 3.14) In all cases the optimization is working properly, since the likelihoods are always increased. Furthermore, the relative improvements are not constant at all, which validates Proposition 2, and we have found a set G_{Q-best} of transcriptions ("s8zs", "s80s" and "s8s") the likelihoods of which (-5.71,-5.13 and -3.78) are higher than -6.15, which was the starting likelihood of "s8zs". It is important to notice that we increased the likelihood justifying the *hypotheses*; but what about the value of the likelihoods justifying the *correct* transcription? This information (unavailable in practice) is shown in columns v,vi,vii in plain. We can clearly notice that the conclusion announced in (3.20) holds. Such values are of course all equal in column v, because all the features are justifying tr_c. A second step of recognition is then performed and the $G_{2,N-best}$ set is obtained (Equation 3.19) and shown in column viii, while their likelihoods are shown in column ix. If we had performed only Unsupervised Limabeam, we would not have got the G_{Q-best} set, thus we would have just optimized "s8zs" and we would have obtained again the same transcription (with consequently no improvement), because the optimization was not able to provide enough compensation: this is true because in column vii its likelihood justifying the correct transcription is -5.76, which is far lower than -4.07 and -3.78. The final step of the proposed technique is to select the maximal value of column ix and "elect" the related transcription ("s8s", in green bold) as the best, which in this case is correct. Few considerations follow:

1. The presence of the correct hypothesis in the N-best list is important. For this reason enhancing speech via a microphone array is very useful.

2. Even if the hypothesized transcription is not correct, the optimization lead to create potential candidates to the best transcription: the final 2-best hypothesis is correct!

3. With the Oracle Limabeam, only the fourth row is considered, and the result is correct. In practice we achieved the performance of this supervised algorithm in an unsupervised way. Keeping in mind the previous observation, we will see that performance can be even higher.

The behavior of the proposed technique can also be checked in Figure 3.4, where we show that its effect is to automatically re-rank the N-best list, thus to create a new 1-best transcription: the plotted value are relative to columns ii and ix and thus show the very first and the very last likelihood values in the process.

Rec 1		Select		Align	Opt		Rec2		ML
i	ii	iii	iv	v	vi	vii	viii	ix	x
s8zs	**-6.15**	**1**	**s8zs**	-6.40	10%	-5.76	s8zs	-5.71	3
				-6.15	**7.2%**	**-5.71**			
s80	-6.28	2	s80s	-6.40	36.4%	-4.07	s8s	-4.07	2
				-6.52	**21.3%**	**-5.13**			
08zs	-6.35	3	s08zs	-6.40	5.3%	-6.06	s8zs	-5.93	4
				-6.73	**1%**	**-6.66**			
s8s	-6.40	4	s8s	-6.40	41%	-3.78	**s8s**	**-3.78**	**1**
				-6.40	**41%**	**-3.78**			

Table 3.1. Example of correct transcription automatically "pushed up" in the N-best list. The table shows the evolution of likelihood values across optimization. The likelihood computed on the current $n - best$ transcription is a good approximation of the likelihood computed on the correct transcription (shown in blue, bold font), which is not available in practice. Thus the correct transcription, initially ranked 4th, becomes the first, right choice after N-best UL.

3.2.2 Results at -5 dB

We check the performance of the proposed N-best Unsupervised Limabeam on the SUBTEST set with a challenging SNR of -5dB: in this condition the margins of improvement are more evident. In Figure 3.5 its performance is depicted when the N-best list is given unmodified to the Viterbi alignment (magenta, dotted line), like in column i of Table 3.1 and in the silence-constrained case (red, solid line). Seltzer's Unsupervised Limabeam is reported as a green, full circle. Clearly the use of competing parallel optimizations leads to dramatic improvements, especially in the silence-constrained case. We recall that when the length of the N-best list is one, the algorithm is equivalent to UL (65.78% unit accuracy), which is not affected by the constraint about silence. This constraint starts being active when at least three transcriptions are competing. Then the curves tend to have a log-like behavior: as the length of the N-best list increases, also the accuracy increases and reaches an asymptote when this length is around 20. Here, the highest asymptote is 72.76%, which means that we reached a relative improvement of 19.6% with respect to the Unsupervised Limabeam. Adding silence is efficient, but it is worth noticing that the shape of the curves is very similar, which is a sign that the mutual order of the N-best transcriptions is not significantly affected. We point out that each time that the accuracy increases with the length of the N-best list, say from length X_1 to X_2, with $X_2 > X_1$, it means that, after optimization, the new 1-best transcription of a X_2-best list has a Word Recognition Rate higher than the 1-best transcription of a X_1-best list. In other words, for each increment of WRR, the X_2-th element has played a key-role: specifically, it has been chosen to be the new 1-best. This is true because the optimizations are independent from each other. Furthermore, the non strict monotonicity of the curves (from 11 to 13-best in the higher curve, from 14 to 15-best in the lower) can also be explained. This phenomenon happens when the X_2-th element has played the same key-role, but in a bad sense: its likelihood is so high and so

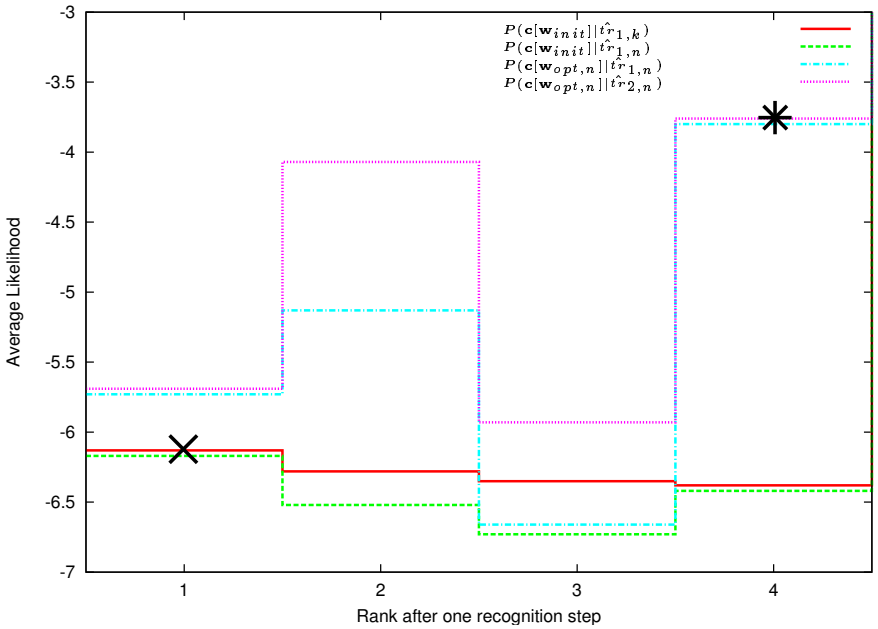


Figure 3.4. Example of normalized likelihood of 4 best hypotheses of a single sentence. Before optimization, transcriptions are ranked by likelihood. After, all likelihoods are increased and the 4th hypothesis, which has a lower WER than the 1st, is now the new maximally likely.

close to a wrong transcription that it is "elected" new 1-best and the algorithm has a worse WRR. This phenomenon can be seen in Figure 3.6, where the Digit Accuracy curve is plotted together with the trend of the Likelihood. The "Likelihood" here is the sum of all the acoustic scores of all the sentences of the SUBTEST set computed by the recognizer, divided by the number of sentences. The Likelihood curve must be monotonic, because only a new competing transcription in the N-best list can determine a new maximum of the likelihood. The highest performance is simply ensured by increasing the length of the N-best list up to a new length X_3, where a higher WRR transcription will also be the ML.

3.2.3 Considerations about the "Oracle" term and the clean speech alignment

As mentioned in Section 2.1.3, two kinds of correct information can be provided to the optimization stage of the Limabeam algorithm:

- Either the correct transcription is given, and the Viterbi algorithm provides an alignment from the beamformed output. This alignment can be theoretically different from the one obtained on the close-talk signal.

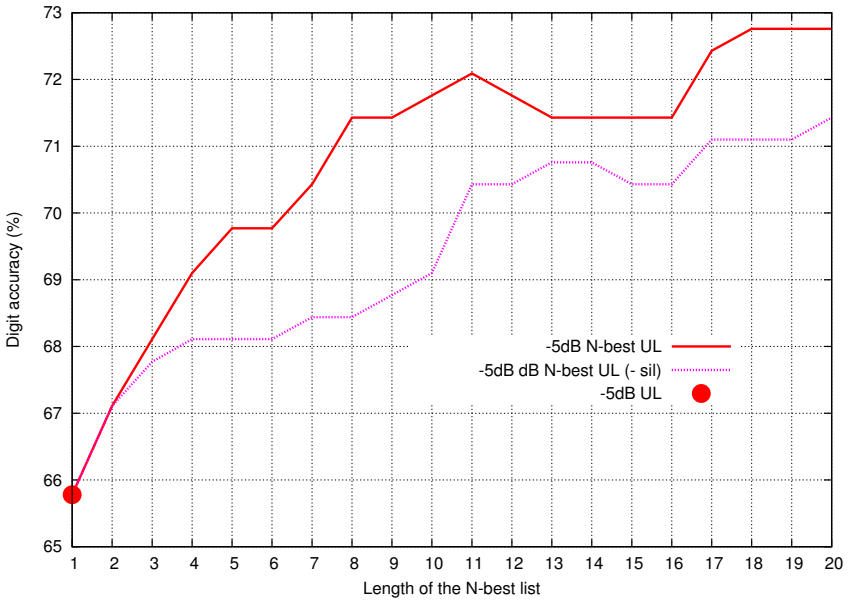


Figure 3.5. Performances on the SUBTEST set of Unsupervised Limabeam (green circle), N-best Unsupervised Limabeam without forcing silence boundaries (magenta, dotted line) and with silence boundaries (red, solid line). The N-best UL gives a dramatic improvement with respect to UL in either of the two forms, which also have the same behavior. The non-monotonicity is due to the mismatch between the Maximum Word Recognition Rate and the Maximum Likelihood criteria

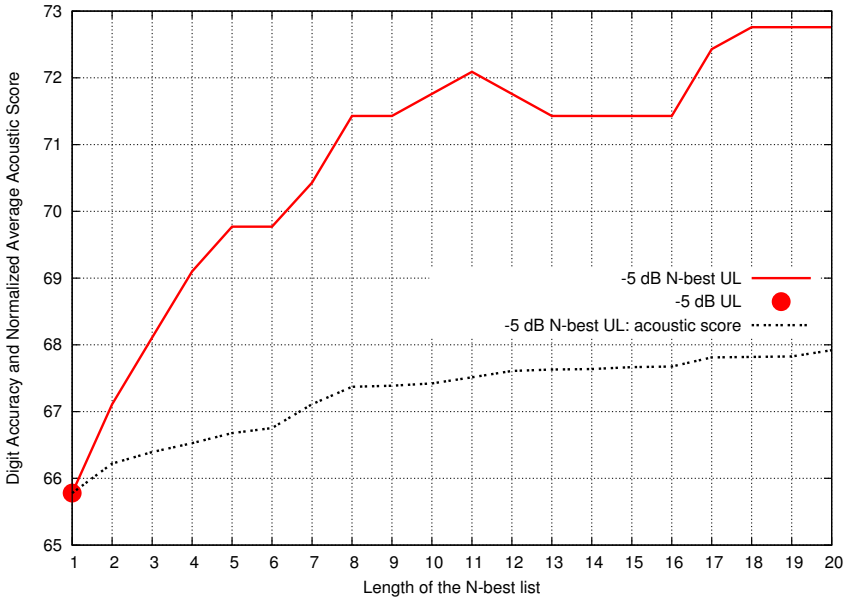


Figure 3.6. Trends of the N-best UL and of the Likelihood curves. The behaviour of the two curves is similar. While the Likelihood must be monotonic for increasing length of the N-best list, the Digit Accuracy has no such constraint. The relative improvement of the Likelihood is limited compared to the relative improvement of the Digit Accuracy. Little variations of Likelihoods imply bigger variations of Accuracy

- Or the Viterbi algorithm provides an alignment directly from the close-talk.

Thus, the second solution is not necessarily better than the first, because noise still affecting the beamformed signal can mislead the forced alignment and give word boundaries far from the real one. We compared two different alignments : as an example, we compare the two solutions (see Figure 3.7). The state sequence of alignment derived from the close talk visits all the three states of the silence model "sil", while in the other case the central (and more representative) state is skipped. This happens because the likelihood of the noise is very far from the silence model. It can also be noted that, when the alignment is obtained from the beamformed signal, the beginning frame of, for example, "zero" comes later than the real beginning frame in the close-talk case. At this point one could argue that the differences in these alignments may result in general different performances: however, we verified that only marginal fluctuations in the accuracies occur. It should also be specified that the noisy signals from which the alignments of Figure 3.7 come from the HIWIRE database. This database was recorded in the IRST SILENT room so that the noisy utterances are synchronous to the close-talk ones: we were thus able to measure performance of N-best UL with and without the alignment on the close-talk. However, the utterance of the database collected in the IRST CHIL room (see Chapter 4) are not synchronous with the close-talk, and we cannot compare the alignments. Finally, matching reverberant signals to close-talk alignments would imply setting most of the echoes of the digits to the silence model, which would be unrealistic. For these reasons, the "Oracle" term will refer in this work to a system which provides the correct transcription only, thus reducing to the first aforementioned case.

3.3 Experiments at different SNRs

Here we analyze how the N-best UL scales to higher dBs: we have seen (Section 2.3.2) that applying the Oracle Limabeam at SNRs equal to 5 and 10 dB led to almost no improvement. Thus we can expect the margin of improvement of the N-best UL to be small as well.

Figures 3.8(a) and 3.8(b) show that the behaviour of D&S, OL and N-best UL at -5 and 0 dB is similar. We observe the amount of improvement achieved by the proposed method with respect to D&S beamforming and depicted as a green dashed line (38.8% relative at -5 dB, 43.1 % at 0 dB). Of course the D&S is a straight line, because no multiple hypotheses are considered nor optimization is performed. At higher SNRs, the margins are indeed smaller and we prefer to report them in a table. All results concerning the SUBTEST are summarized in Table 3.2 for varying methods and noise conditions, together with the relative improvements. At low SNRs the N-best UL is very

```
1 46 sil 2
47 47 sil 4
48 48 zero 2
49 49 zero 3
50 50 zero 4
51 51 zero 5
52 52 zero 6
53 54 zero 7
55 55 zero 8
56 57 zero 9
58 59 zero 10
60 60 zero 11
61 61 zero 12
62 63 zero 13
64 66 zero 14
67 72 zero 15
73 76 zero 16
77 84 zero 17
85 85 one 2
86 86 one 3
87 87 one 4
88 88 one 5
89 89 one 6
90 91 one 7
92 97 one 8
98 101 one 9
102 106 one 10
107 109 one 11
110 110 one 12
111 111 one 13
112 118 one 14
119 119 one 15
120 120 one 16
121 124 one 17
125 140 sil 2
141 141 sil 4
```

```
1 1 sil 2
2 15 sil 3
16 25 sil 4
26 46 zero 2
47 47 zero 3
48 48 zero 4
49 49 zero 5
50 51 zero 6
52 53 zero 7
54 57 zero 8
58 58 zero 9
59 60 zero 10
61 61 zero 11
62 62 zero 12
63 63 zero 13
64 66 zero 14
67 72 zero 15
73 77 zero 16
78 79 zero 17
80 80 one 2
81 81 one 3
82 82 one 4
83 83 one 5
84 84 one 6
85 86 one 7
87 96 one 8
97 101 one 9
102 106 one 10
107 109 one 11
110 110 one 12
111 112 one 13
113 113 one 14
114 115 one 15
116 118 one 16
119 122 one 17
123 124 sil 2
125 140 sil 3
141 141 sil 4
```

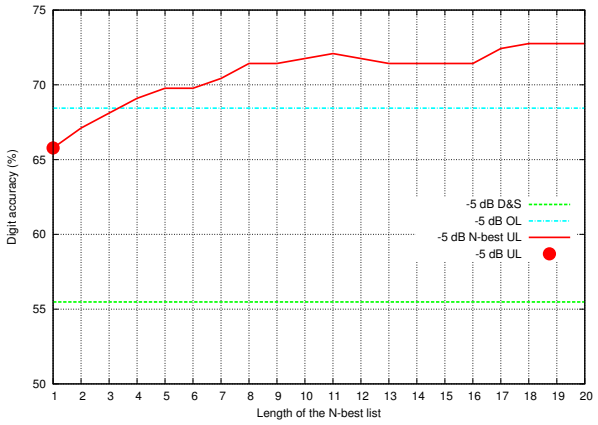
Figure 3.7. Alignment obtained when inputs are the correct transcription and the beamformed signal (left) and alignment obtained when inputs are the correct transcription and the close-talk signal (right).

effective (72.76% absolute at -5 dB and 87.71% at 0 dB), while for SNRs above zero its contribution is more limited (96.01% absolute at 5 dB and 98.01% at 10 dB) and the relative improvement over D&S is constantly around 14%.

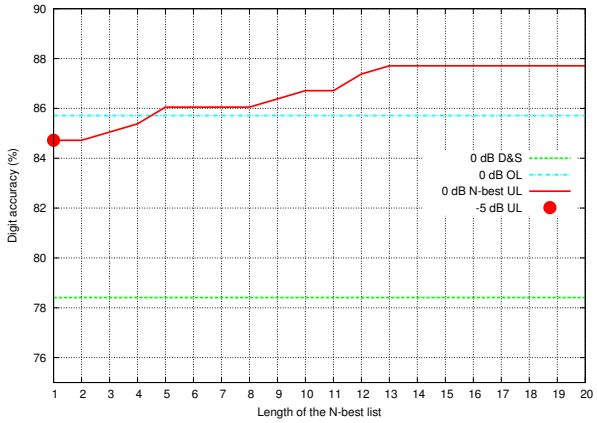
The average relative improvement is 37.7% over D&S and 18.9% over Unsupervised Limabeam. The latter result represents the maximum improvement brought to the state of the art by means of the proposed N-best UL. The most surprising result is the fact that the N-best UL goes over the OL for a sufficiently long N-best list: this happens at all SNR and it is evident in subfigures 3.8(a) and 3.8(b), where the OL is depicted as a blue dashed-dotted straight line. The reason is not intuitive and deserves further clarification, given in the following sections.

3.4 Explain the Oracle problem

The improvement of the Oracle Limabeam over the simple D&S is relatively good. However (on TI-digits), clearly the N-best Unsupervised Limabeam superseeds the OL when the number of hypotheses optimized in parallel is greater than three (at -5 dB) or four (at 0 dB). Indeed, we justify



(a) -5 dB (Digit Accuracy)



(b) 0 dB

Figure 3.8. Comparison between D&S, OL and N-best UL at -5 and 0 dB. The behaviour of the curves is similar.Results in digit accuracy.

2

8 mics, 10 taps, SUBTEST	fan noise level (dB)				
method	-5	0	5	10	average
D&S	55.48	78.41	95.35	96.67	81.72
UL	65.78	84.72	95.68	97.67	85.96
OL	68.44	85.71	95.68	97.67	86.88
N-best UL	72.76 (19)	87.71 (13)	96.01 (16)	98.01 (2)	88.62
RI	38.8	43.1	14.2	14.6	37.7

Table 3.2. Comparison between D&S, OL and N-best UL in the (-5,10) dB range. The major improvements over D&S are at very low SNR, while for relatively high SNR the relative gain in accuracy is constant.The number in parenthesis is the minimal length of the N-best list to achieve the correspondent result. Results in digit accuracy.

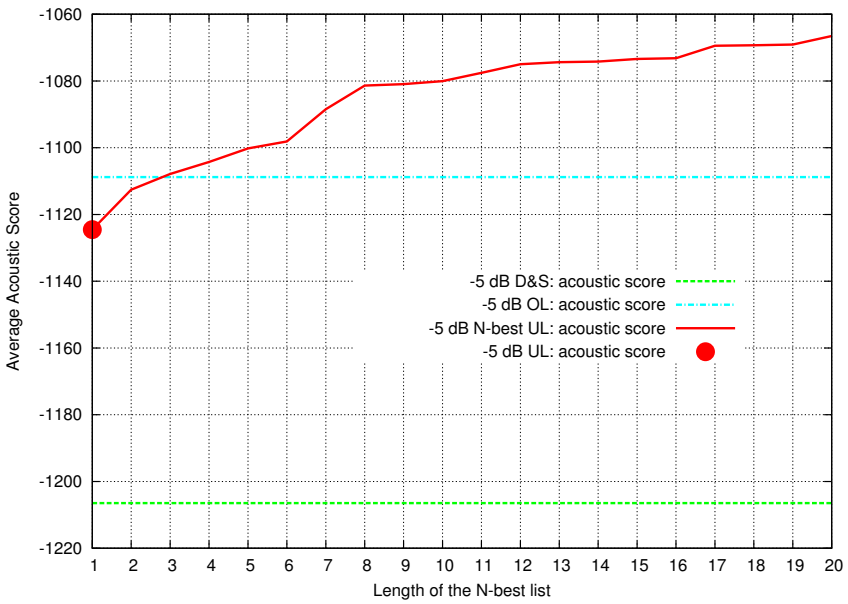


Figure 3.9. Insight of the behavior of the Likelihood: its values are given for all methods: the same distances in performances are preserved.

below why the OL is not an upper bound to our performance. First, we start comparing the Likelihood curve of the N-best UL with respect to the same value computed for D&S and OL: this is showed in Figure 3.9. The trends are very similar. It is important to notice that there is in this case consistency between higher recognition rates and higher likelihoods. However, though related, recognition relative improvements do not correspond with equivalent likelihood relative improvements. This is useful because it shows how much sensitive are recognition results to likelihood variations. Figure 3.9 shows that the Average Likelihood passed from -1124 to -1066 with 20 competing sentences. Knowing that the close-talk speech likelihood is -198, assuming this as the best possible reachable value, the relative likelihood improvement is 6.3%. In a corresponding way, the WRR increases of a 20.4% relative (from 65.78% to 72.76%) The RI of the OL (-1108 absolute) with respect to UL is even smaller: 1.7% and the related WRR relative improvement is 7.7.% (from 65.78 to 68.44%) WRR is then strongly sensitive to likelihood. This is evident in Table3.3. If we consider that the likelihood and the recognition score evolve in a similar way, we cannot expect a high recognition rate for Oracle tests, but observing the evolution in terms of the N-best list of the N-best UL, we can expect improved results with this technique *by exploiting one or more acoustically confusable transcriptions*, which are available in the N-best list.

Rec 1		Select		Align	Opt		Rec2		ML
i	ii	iii	iv	v	vi	vii	viii	ix	x
0841s	-9,12	1	s0841s	-9.34	24.7%	-7.04	s7241s	-7.04	2
				-9.35	**16%**	**-7.83**			
07241s	-9.17	2	s07241s	-9.34	7.4%	-8.69	s7241s	-8.69	5
				-9.39	**5.3%**	**-8.89**			
07841s	-9.22	3	s07841s	-9.34	8.9%	-8.60	s7241s	-8.60	4
				-9.44	**4.7%**	**-8.99**			
08410s	-9.28	4	s08410s	-9.34	30%	-6.63	**s7241s**	**-6.63**	**1**
				-9.35	**9.8%**	**-8.56**			
s7241s	-9.34	**9**	s7241s	-9.34	8.7%	-8.52	07241s	-8.51	**3**
				-9.34	**8.7%**	**-8.52**			

Table 3.3. Example of correct transcription which is automatically "pushed up" in the N-best list (from 9th to 3rd position), but not enough to generate a correct transcription. Instead, the first four parallel competitors were well maximized and the initial 4-best becomes the 1-best. The likelihood computed on the current $n-best$ transcription is a good approximation of the likelihood computed on the correct transcription (shown in blue, bold font), which is not available in practice. Thus the correct transcription, initially ranked 4th, becomes the first, right choice after N-best UL.

3.4.1 Exploiting acoustic confusability

We have seen in the example described in Section 3.2.1 that a correct transcription, initially down in the N-best list, can be ranked first by our proposed technique after optimization. We have also noticed that an incorrect, acoustically confusable transcription allowed the optimization to produce features with a very high likelihood. The likelihood was high enough that at a second recognition step the transcription output was correct. In this section we give an example where we show that *the presence of* tr_c *in the N-best list is not necessary*. Some confusable transcription may optimize the filters better than tr_c. We chose an example to show to which extent this is true: Table 3.3 is similar to Table 3.1, in the sense that the correct transcription is down in the N-best list. After optimization, it is automatically re-ranked and pushed up to the third position: however, the optimization was too limited (the likelihood improvement with respect to tr_c is only 8.7%, see column vi). In parallel, the first four competitors were improved much more and were well recognized. This shows that applying the Oracle Limabeam, which would be equivalent to considering the fifth row only, is not optimal for recognition performance. Finally, the likelihood to "07241s" is -8.51, while the likelihood to "s7241s" is -8.52.

We have seen that the margin of improvement of the proposed technique is logarithmic to the amount of parallel N-best hypotheses considered. However, in the last example we measured the likelihood that *clean features* justify the correct transcription: its value is -2.05. This means that there is still a lot of likelihood improvement to achieve.

One could argue that increasing the number of N-best competitors could lead the features to be optimized too close to a bad transcription. However, the N-best UL reached an asymptote and does

not move from it, thus this risk is inexistent. We motivate this fact by observing that, as previously mentioned, the FIR filters have the capability of smoothly changing the spectrum of the speech signal. They are not in general able to deeply change the spectral shape, including the formants, because they are only ten taps long.

3.4.2 Noise reduction; time, spectrum and cepstrum of an optimized sentence

In this section we analyze the effect of the optimal filters on the pure recorded noise, and on an utterance in time, frequency and quefrency domain. In the next section we will analyze the nature of the ML filters themselves. The purpose is to verify how close the signals in different domains are to the clean speech signal. All the pictures are related to the same utterance, which is "five six seven" (file FEJ_257A), which is perfectly recognized by the N-best UL.

Noise suppression

Figure 3.10 depicts the amount of noise reduction achieved by the filters derived via N-best UL on the spectrum of the recorded fan noise. Because we are interested in finding a relation with recognition results, we are analyzing the frequency bands which are significant for recognition only, i.e. from 100 Hz to 7500 Hz. These are the cut-off frequencies of the first and last Mel Filterbank triangles. The pure noise is depicted as a green, dashed line. Already a fair amount of noise suppression is achieved via OL, depicted with a blue, dashed-dotted line, but it is evident that N-best UL is able to further suppress noise before 1000 Hz and after 5000 Hz. This is interesting because, if we use for example 24 such Filterbanks, in this range 10 filters will fall before 1000 Hz and three after 5000 Hz. Thus, as we showed that N-best UL performs better than OL, the recognition improvement is linked to the noise attenuation in these specific bands, at least with this kind of noise.

Time domain

Figure 3.11 reports on time domain signals: they are obtained by applying D&S beamforming, OL, N-best UL and the fourth is the clean speech signal. It is in general difficult to observe the amount of improvement directly on the time domain signal, but in this case the shape of the utterance optimized with N-best UL most resembles the clean speech one. As it was also evident from the noise spectrum, the low frequencies are well compensated.

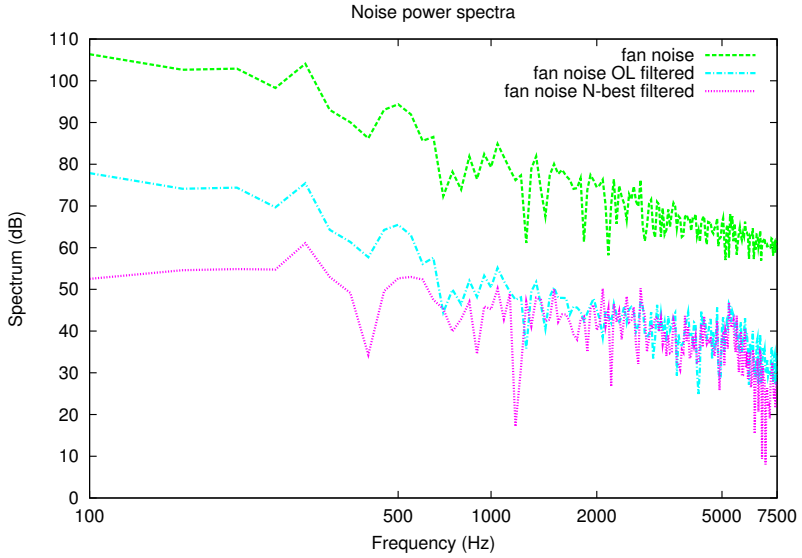


Figure 3.10. Effect on the long-term power spectrum of pure fan noise recorded by one channel (in green, dashed line) of a 10 tap filter optimized with OL (blue, dashed-dotted line) or with the N-best UL (magenta, dotted line). The spectrum is plotted in a log-scale, in the frequency band spanned by the Mel Filterbank. N-best UL provides a major noise reduction, especially for lower and higher frequencies of interest.

Frequency domain

Figure 3.12 shows the long-term spectrum of the whole utterance, to which D&S, OL, and N-best UL are applied. The spectrum signal appears to be excessively smoothed by the use of the first two techniques, while the proposed one follows the clean speech spectrum (in red, solid line).

Cepstral domain

Finally, we analyze the domain closest to recognition performance (Figure 3.13): in the cepstral domain, restricted to the static coefficients [c2-c24], clearly the proposed method generates a shape very similar to the clean speech cepstrum. A bias separates the two curves from c2 to c9. Beyond c9 the shape are almost equal.

3.4.3 Analysis of Maximum Likelihood filters

We wonder what are the characteristics of ML filters generated by the N-best UL. Figure 3.14 shows the frequency response of three such filter sets. Each filter set spans one row of the picture: the Power Spectrum, in dB, is plotted on the left and the Phase response, in radians, is plotted on the right. The Power Spectrum is in a log-scale and frequencies below 100 Hz are neglected, which is what the Mel Filterbank actually does in the Front-End computation. The first two sets

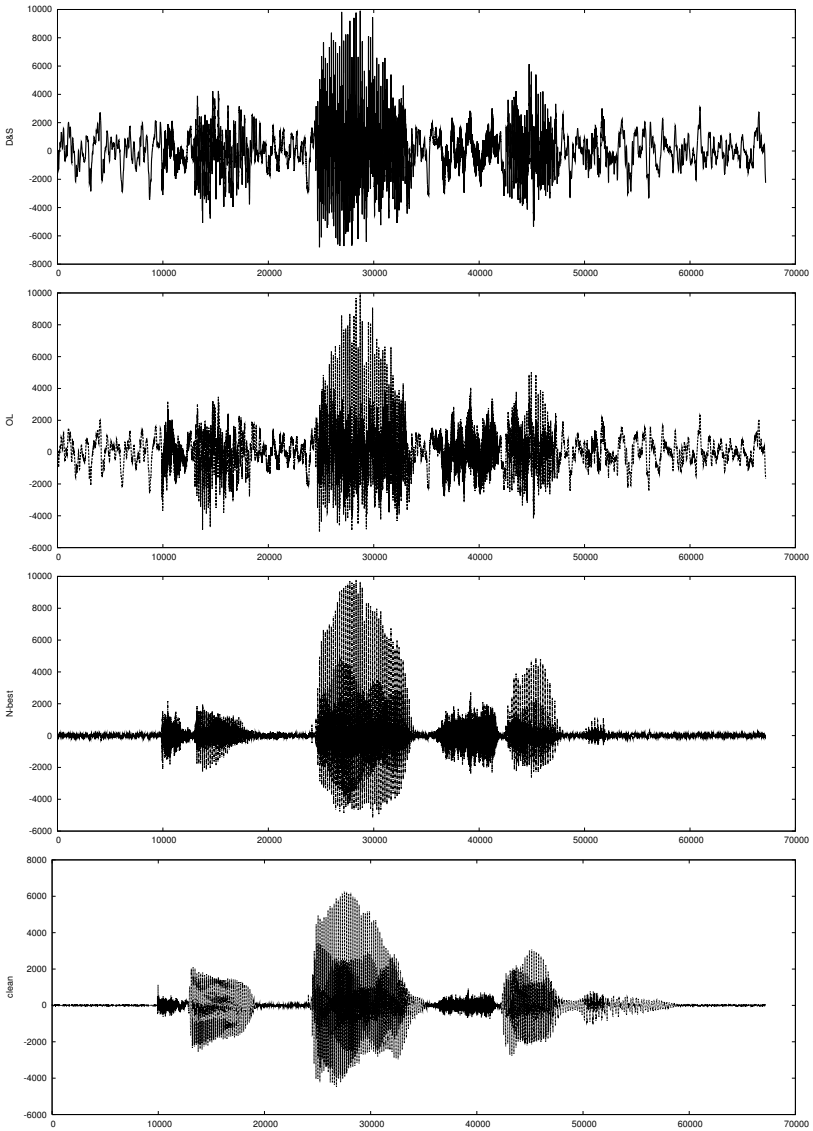


Figure 3.11. Time domain signal of the utterance "two five seven", after D&S beamforming (first, uppermost), OL (second), N-best UL (third), and original clean (fourth). The N-best UL gets the time domain signal closer to the clean one.

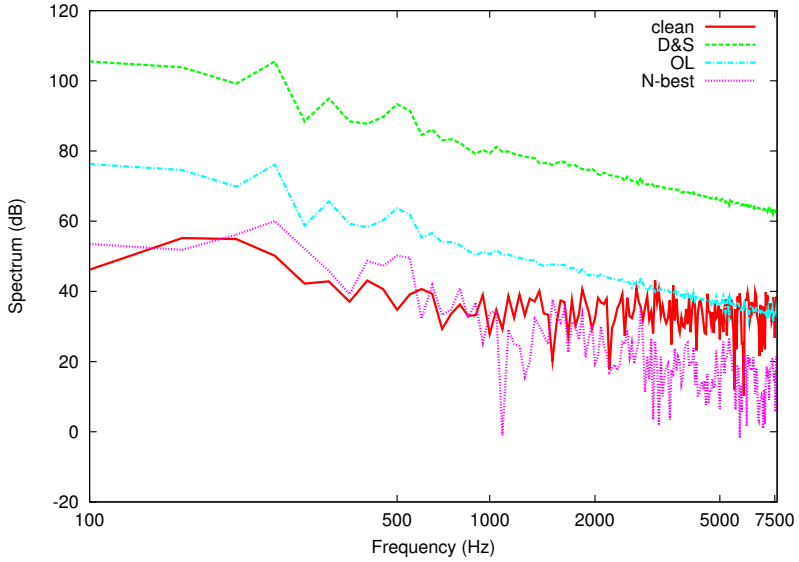


Figure 3.12. Effect on the power spectrum of a filter-and-sum beamformed utterance, using D&S (green, dashed line), using filters optimized with OL (blue, dashed-dotted line) or with the N-best UL (magenta, dotted line). Clean speech power spectrum in red, solid line. The spectrum is plotted in a log-scale, in the frequency band spanned by the Mel Filterbank. N-best UL causes the log-spectrum to be much more similar to the clean counterpart.

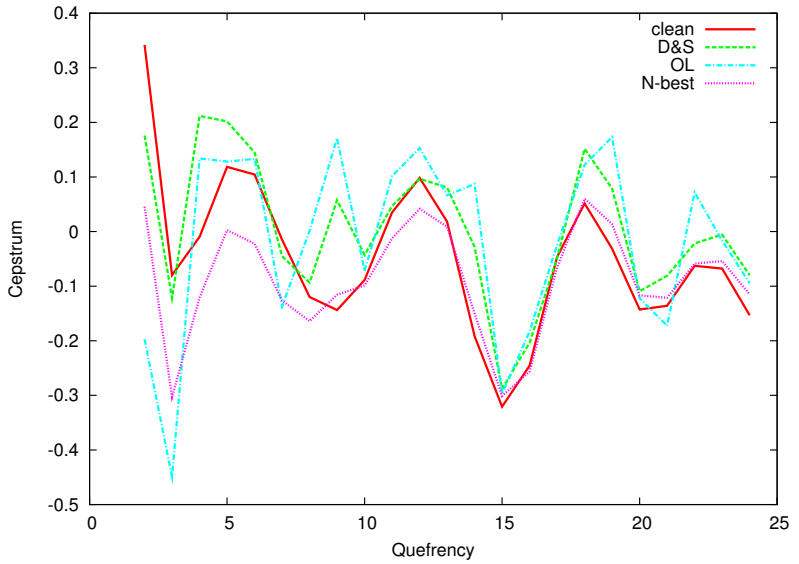


Figure 3.13. Effect on the static cepstrum (from c2 to c24) of a filter-and-sum beamformed utterance, using D&S (green, dashed line), OL or with the N-best UL. Clean speech cepstrum in red, solid line. The N-best UL cepstrum well follows the clean ones (higher cepstrum) or it differs by a bias (lower cepstrum).

are optimized on the transcriptions “two two nine” (of the female speaker tagged as FN) and “zero” (of the male speaker tagged as BC), while the third set is optimized on “seven nine” (of the female speaker tagged as DW). The difference lies in the fact that in the first two sets the N-best UL gives a higher accuracy then the OL, while in the third set the OL is better. For each subfigure, the OL filter set is depicted with a magenta, dotted line (there is one curve per microphone), while the N-best UL is depicted with a red, solid line.

Considerations on Power Spectra

Observing the left side of Figure 3.14 several aspects can be noticed:

- In a) the power Spectrum of the filter set optimized with N-best UL is much more high-pass then its OL counterpart. The same behaviour can be observed in subfigure c), where a valley around 5 kHz is also present.
- The high pass effect is always beneficial (see Section 2.3.2), because it compensates the naturally low pass effect of a D&S beamformer. On this task it is also beneficial because the energy of the fan-noise is mostly concentrated in the lower frequency range.
- The link between high-pass effect and high recognition rate is cross-validated by the Subfigure e), where the OL has a higher recognition rate and its filter set has a more evident high pass effect.
- The Power Spectrum of a filter set is linked to the transcription used to optimize it (compare subfigures a), c) and d)). Specifically, around 5 kHz in a) we have a peak for the OL (the transcription used for optimization is by definition “sil two two nine sil”), but a valley for the N-best UL (the transcription is the 17th-best in the list and it is “sil eight two two nine oh sil”), in b) there are valleys only and in c) peaks only.
- Some microphones are much more high passed and thus attenuated than others. This is mostly evident in e). However, attenuated microphones are rarely the same from utterance to utterance: we verified that the average per-channel energy is approximately constant.
- There should be no relation, at least on this task, between the attenuated microphones and the position of the speaker or of the noise source with respect to the microphone array: in fact we recall that the energy received at each simulated channel is the same and the only difference lays in the noise delay. The algorithm is simply choosing to attenuate one channel rather than another if this increases the likelihood function.

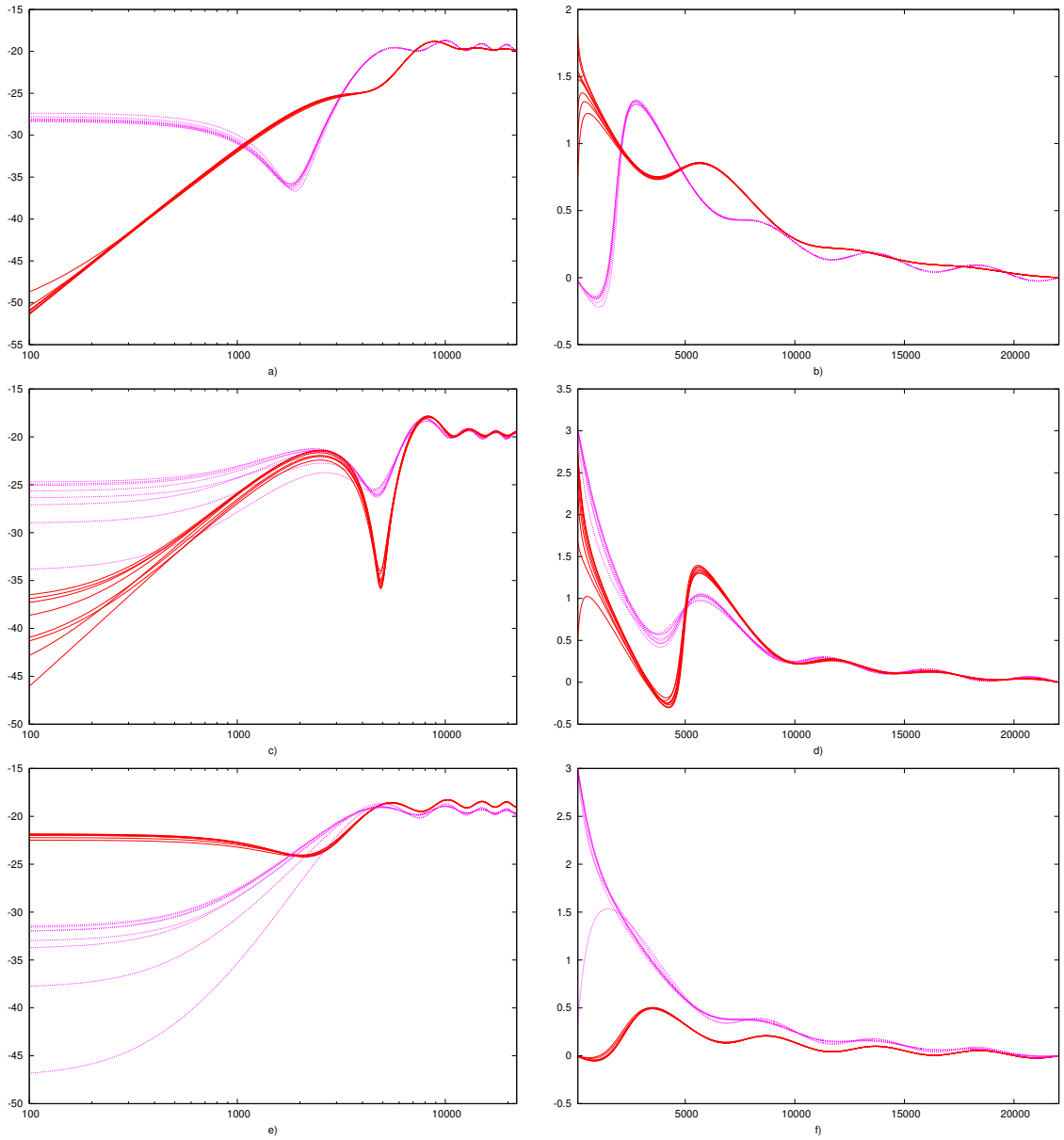


Figure 3.14. Comparison of OL (magenta, dotted line) and N-best UL (red, solid line): frequency response of three filters, each optimized on a different transcription. The power spectrum is plotted on the right on a log scale, the phase on the left. The best recognized sentences are related to solid lines in the first two rows and with dotted lines in the last row of pictures. High pass effect and phase linearity are two key requirements for filters to generate maximum WRR utterances.

Considerations on Phases

In Figure 3.14 we observe the behaviour of the Phase response: for each picture we are aware of the FIR filter set generating the maximal recognition utterances. These utterances share a common characteristic: the Phase is approximately linear before 4 kHz. This happens if N-best UL is used (red solid line) both in b) and d) and if OL is used in f). If the Phase is not linear at certain frequencies, the corresponding harmonics in the time-domain are shifted by a certain number of samples, proportional to the derivative of the Phase with respect to the frequency. Specifically, we noticed that the better recognized the data are, the more constant the phase is. This should not have an impact if one channel only is used, because our ASR system gets rid of the Phase information, but it does if more channels are affected by Phase distortion before being summed: in fact the amount of destructive interference provided by the beamformer can change significantly. This is evident in the simple example below:

$$\frac{1}{M}\sum_{m=0}^{M-1} cos(2\pi f^* t + \phi_m) = \begin{cases} cos(2\pi f^* t) & \text{if } \phi_m = 2m\pi; \\ 0 & \text{if } \phi_m = (2m-1)\pi. \end{cases} \qquad \forall M, even \tag{3.25}$$

where at the fixed frequency f^* the cosines sum again to a cosine or to zero, depending on the phase shift. It is worth noticing that any non-linearity of the phase has no effect in the bands smaller than 7.5 kHz, which we plotted anyway, because frequencies above this threshold are filtered out by the Mel Filterbank. Because we observed on most of the filters that *high pass filtering and phase linearity imply a higher recognition rate, this can constitute a sufficient condition*. Because the absence of both these effects causes the recognition rate to be worse, this can constitute a necessary condition and we can conclude that there is a tight relation between high pass filtering, phase linearity and higher recognition rates.

3.4.4 Accuracy evolution across the methods

We have seen that by optimizing in parallel 20 sentences and then selecting the ML transcription provides a maximum improvement of 13.6% relative to OL. However, this is only an average value: we would like to better understand how many and which sentences are involved in the optimization. In other words: is the N-best approach better optimizing the filters of the sentences which the OL was not able to optimize or is there no such a relationship? The answers to this question provide information about the relevance of optimizing with an acoustically confusable transcription and the pertinency of optimizing with an oracle transcription respectively.

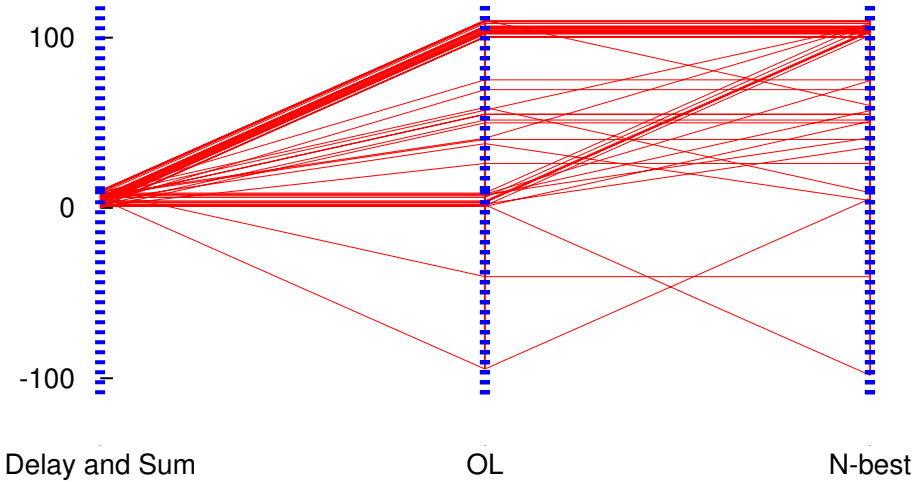


Figure 3.15. Accuracy Evolution

Figure 3.15 shows in a trellis style how the filters evolve through three different methods: D&S, OL and N-best UL. The environmental conditions are -5 dB and the parallel optimization involved 20 sentences. For each sentence of the SUBTEST set we compute the relative improvement of the OL and of the N-best UL with respect to the D&S technique (for which, as a consequence, all the accuracies are set to zero). Then we put the computed value on a vertical anchor line, which spans from -100% to +100% accuracy (in general, the improvement/regression in accuracy can be larger for each sentence, but this does not happen on this set). If the accuracy of a certain sentence is improving with the use of the N-best UL with respect to OL or with the use of OL with respect to D&S, then the two values on the anchor lines are connected with a segment of positive slope. Notice that only the sentences for which the accuracies change across the three methods are depicted. The pictures shows that the majority of segments concentrates in a kind of parallelogram: clearly, from the D&S configuration, a change in accuracy is never negative except for two cases. The improvements are polarized toward the 100% value (left segment of the parallelogram), while several sentences (though optimized with the right transcription), did not report any improvement (lower basis of the parallelogram). However, the most interesting part of the picture is the right part. We can notice that most of the sentences are already well optimized (see the upper basis of the parallelogram) which means that basically the N-best is not generally “canceling” the improvement provided by the OL. Furthermore, several segments rise, mostly from the 0% level in the OL line towards a higher value in the N-best line (right segment of the parallelogram): these are the afore-

mentioned data for which the correct transcription did not provide enough improvement to move from the accuracy of the D&S performance, but for which the N-best UL was instead beneficial. And here lies the reason why the N-best UL is globally superior to the OL. Finally, we depicted only the accuracies of the sentences which change their status, which are 43% of the set. Thus, from the picture we excluded 21% of data did not need to be optimized, because their accuracy was already 100%. Of course, when we apply the algorithm we don't know this in advance, so we conclude that applying the N-best approach to already well-recognized data (after a D&S) do not significantly alter the already achieved pertinency, from a recognition point of view, of such data. This is important because it confirms that optimizing well recognized data with acoustically confusable transcriptions is not harmful for this task. The remaining 36% of data is recognized with the same accuracy for all the three methods, but this accuracy is below 100%. This means that there is still a margin of improvement on this task: using more than 8 microphones is likely to reduce this percentage.

3.4.5 Amount of confusability in the best optimizing transcription

In this section we analyze how much the best transcription differs from the Oracle transcription. Figure 3.4 shows ten optimal transcriptions, which cause the recognition score after N-best UL to be strictly higher than the score obtained with OL. In the first column we show the filename, reported in the Aurora2 format, as it is more compact than the TI DIGITS format: the filename contains information about the gender and ID of the speaker, together with the correct transcription. Data are optimized with the transcription showed in the second column. Silence is present at the boundaries of every transcription as stated in 3.1.4. The rank in the N-best list output after the first recognition step is shown in the third column. We can notice that the best optimizing transcription is not much different from the correct one, and that its rank is quite scattered in the N-best list. Because a quantitative study better clarifies this result, we measure the Levenshtein distance between every correct transcription and the corresponding N-best optimizing transcription, which in turns gives the best accuracy using the N-best UL. This is done for each transcription of the WHOLETEST set. Table 3.5 shows for each digit the number of errors (columns "sub", "ins" and "del") out of the total number of appearances "num", the number of well matched digits "corr" ="num"-("sub"+"del"), and the percent accuracy (see Equation 2.16) and percent correct "%corr"="corr"/"num". Furthermore, we extracted from the confusion matrix the most matched digit ("most") together with the second and third most confused one ("sub1" and "sub2"). Finally, we report in the last row the summary for all the 1001 transcriptions of the set. By observing this table, we can note several facts:

1. The most frequent phenomenon for a N-best optimal transcription to be different from the

file	n-best optimization transcription ($n \leq 20$)			n
FEJ_257A	sil	oh two nine seven	sil	7
FFN_229A	sil	nine eight two nine oh	sil	17
FGN_7241A	sil	oh eight four one oh	sil	4
FHF_5349A	sil	oh nine three four nine	sil	5
FLJ_1432A	sil	one one three two	sil	7
FMJ_68OOA	sil	oh six seven eight oh oh oh	sil	8
MBC_ZA	sil	eight zero	sil	17
MED_876ZA	sil	eight zero six zero oh	sil	8
MIB_924A	sil	oh nine eight four	sil	10
MLE_O6OA	sil	oh zero oh	sil	4

Table 3.4. Transcriptions used for optimization giving a (strictly) higher accuracy than the Oracle transcriptions. The transcription whose Levenshtein distance is high from the correct transcription correspond to accuracies close to zero before the optimization starts.

correct one is an insertion. Insertions cover 52% of the errors. Furthermore, more than half of them are caused by the "oh" digit (which explains its negative accuracy).

2. The less frequent phenomenon is deletion: they are 0.05% of the errors. Deletions affect mostly "two", which is the shortest digit.

3. The String Recognition Rate corresponds to the amount of correct transcriptions that also give the highest recognition rate after the N-best UL.

4. Regardless of insertions, the optimal transcriptions are quite similar to the correct ones (Percent Correct at 74.82%), and their dissimilarity are mostly due to confusions of "five", "four" and "two".

Points 1) and 2) state in practice that the N-best optimizing transcription is an augmented version of the correct one: the errors are mostly an "oh" at the beginning or at the end of the sentence, possibly with on average one or two substitutions (this was observed on error histograms). It is important to note that the quasi total absence of deletions is an important proof of pertinency, because it implies that the spectral content of the original transcription is not canceled and information loss for adaptation is prevented. Point 3) states that in 10.89% of the cases the OL is the best solution: however, we recall that we are considering only 20 transcriptions in parallel for the optimization, and that the correct transcription can appear well down in the N-best list.

3.4.6 Distribution of ML sentences

The digit accuracy was shown to increase with the length of the N-best list. However, it would be interesting to see what is the influence of each n-best hypothesis on the last stage of our ap-

	sub	ins	del	num	corr	%acc	%corr	most	sub1	sub2
!NO!	910	0	0	910	0	0.0	0.0	oh 510	nine 115	one 84
eight	45	51	17	282	220	59.9	78.0	eight 220	oh 19	!NO! 17
five	153	7	12	301	136	42.9	45.2	five 136	nine 100	one 20
four	163	0	13	289	113	39.1	39.1	four 113	one 98	oh 36
nine	15	115	2	313	296	57.8	94.6	nine 296	oh 10	!NO! 2
oh	44	510	7	305	254	-83.9	83.3	oh 254	nine 16	zero 11
one	9	84	2	312	301	69.6	96.5	one 301	nine 5	!NO! 2
seven	68	16	7	289	214	68.5	74.0	seven 214	zero 28	oh 18
six	61	54	6	291	224	58.4	77.0	six 224	zero 27	oh 19
three	22	5	3	295	270	89.8	91.5	three 270	oh 8	eight 6
two	145	17	21	294	128	37.8	43.5	two 128	zero 56	eight 30
zero	4	51	1	286	281	80.4	98.3	zero 281	oh 3	!NO! 1
String Recognition Rate: 10.89 %										
Unit Accuracy: 46.88 %, Percent Correct: 74.82 %										
Units: 3257, Correct: 2437, Errors: 1730, Del: 91, Ins: 910, Sub: 729										

Table 3.5. Levenshtein distances measured on WHOLETEST between the correct transcriptions (used in OL) and the N-best optimizing transcriptions (used in N-best UL). The latter are on average an augmented version of the former, with very little loss of information for the optimization/adaptation phase.

proach. In other words: is the choice of the best hypothesis, made by our algorithm, biased around a certain rank n? We expect the answer to this question to be no, because the transcription which best maximizes the likelihood may be in first, second, twentieth position or even further. Figure 3.16 shows the distribution of the choice of ML transcription, while the length of the N-best list is increasing. The distributions were measured on the WHOLETEST set: because it is composed by 1001 sentences, the amount of chosen sentences indicates percentage as well. The upper, shorter line, for example, depicts which transcription is chosen between the only two available competitors. We can notice that each time a set of new competitors is available (i.e. each time the length of the list increases by 1), it becomes immediately discriminative. Furthermore, the fact that for increasing lengths the curves globally decrease (for every n) means that the new, maximally likely transcription chosen from the last available value of n becomes first at detriment of all the others competitors. These are proofs that after parallel optimization all the likelihoods are in the same range of the previous competitors and confirms that measures based on likelihood are very sensitive.

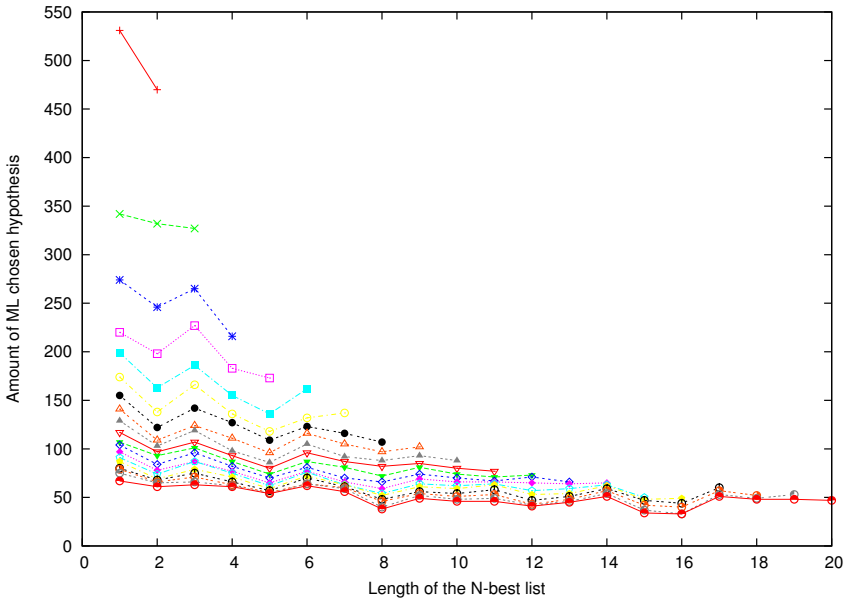


Figure 3.16. Distribution of the choice of the ML transcription across increasing lengths of the N-best list. The histogram shows that the hypotheses are all equi-probable, with a slight preference for the first. When 20 hypotheses are considered the distribution is quasi-uniform.

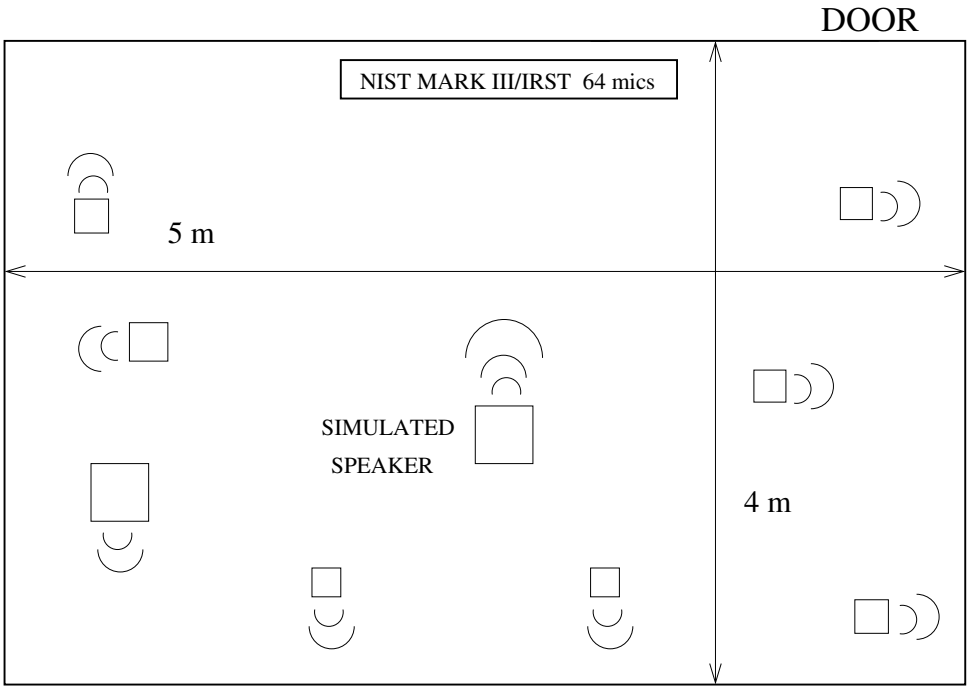


Figure 3.17. Data acquisition room: clean speech is played by the central speaker, noise is continuously played by 8 speakers around the central one. SNR measured at source-level is 0dB.

3.5 Real data: HI-WIRE

3.5.1 Motivation for treating real data

The N-best approach to Unsupervised Limabeam is successful because it compensates the major limitation, i.e. the ignorance of acoustically confusable alternatives to the output of the recognizer. However, the analysis is not enough valid if we do not test in a real environment. We recall that the noise used in simulations, was real, but it was synthetically added. There are so many more variables one cannot control in a real environment: for example, so far we assumed perfect steering of the microphone array and a strongly coherent noise field. In real environments the scenario is more complex, for two reasons: first, steering a microphone array is a delicate phase which requires itself algorithms to be properly performed. From the state of the art we understood that array-based speech recognition algorithms are very sensitive to steering errors. Second, the coherence between noise signals recorded at different microphones is not constant in space. Furthermore, and most important, that source is convolved with a specific room impulse response, which is a function of both the noise and the microphone source position. The number of parameters in such scenario is thus very high. We would like to test our approach in a more realistic scenario [Brayda, 2006b]. Indeed, artificially added noise led to the following conclusions:

- The preliminary experiments on Limabeam (see Chapter 2) showed little improvement if the noise field is perfectly non-coherent, though synthetically added.
- The experiments with fan noise dramatically improved performances.

Can we expect higher performance than D&S in a real noise field? We remind that N-best UL can hardly supersede D&S in highly diffuse simulated noise. Also, we are interested in isolating the additive noise problem from the reverberation problem, which will be dealt with later in this work. Indeed the effect on the algorithm of the two kinds of interference can be totally different. It is essential to organize test identifying these effects separately. We consider that using part of the database of IST EU FP6 HIWIRE project meets this requirement.

3.5.2 Environmental setup

The scenario considered is a quasi-anechoic room, in which clean speech and several sources of noise are played simultaneously. The room, located in the ITC-IRST laboratory, is depicted in Figure 3.17. It is 5 x 4 meters and has a relatively short reverberation time (143 ms). Acquisitions in such room put into evidence more the effects of additive noise than convolutional distortions. Clean

speech is played by a high quality speaker (Tannoy 600A Nearfield Monitor): the data played are part of the TI-DIGITS, database, and correspond to the Aurora test set A [3]. Signals are recorded by the NIST MarkIII/IRST [Brayda, 2005a], placed at 1.3 meters from a Tannoy speaker, located in the center of the room. This device is a linear 64-microphone array, with 2 cm sensor spacing. The improvement of the device performance in acquiring audio data is part of this work and will be discussed later in Chapter 5. While clean speech is played by the Tannoy speaker, noise is simultaneously played by 8 sources, scattered in the room at different locations. The average SNR is 0dB. The interfering signals resemble a typical noise encountered in a cockpit. Notice that the SNR is measured at source-level: the true SNR varies depending on speakers and microphone location. For example, for the two noise sources pointing to the wall and opposite to the array, some of the energy is absorbed and the microphone array will record mainly the reflected paths; however, this fact will contribute to create a more diffuse noise field. It turns out that the SNR can be for some microphones considerably lower than 0 dB. The audio data was captured by the MarkIII at 44.1 kHz and subsampled at 16 kHz. For our task we choose to use 8 microphones, 16 cm spaced from each other: this configuration represents a trade-off among the high performance which depends on an increasing number of sensors, spatial aliasing requirements and the need of a reasonable complexity and time response of the system (for filters optimization). Considering performance, in such a noise field we check that the gain when using 64 microphones (inter-microphone distance = 2cm) instead of 8 microphones (distance = 16 cm) is low: a high spatial selectivity is already reached by the second configuration. Considering spatial aliasing, there is no speech source in end-fire position, thus we can consider the constraint met. Considering complexity, we recall that the computed Gradient vector (see 2.1.2 in Chapter 2) has dimensions $M \times L$, where M is the number of microphones and L is the number of taps per filter (consequently, the current configuration computes 8x10 coefficients), and thus complexity is linear with an increasing number of microphones: since increasing the number of microphones is not that discriminant, we restrict them to reduce the amount of computations.

3.5.3 Results and new *a posteriori* upper bound

By modifying the ML block, which selects the maximally likely transcription among the N-best optimized competitors, we can reach a new upper-bound. Instead of computing Equation 3.21, we

[3] The speakers are not the same than the ones in WHOLETEST (but still the number of sentences is 1001): simulation data matches the filenames of the Aurora set labeled "A N1", while the part of HIWIRE data available for our experiments matches the "A N2" group. However, it is well known that the speech content is well balanced for both sets, and performance can be compared

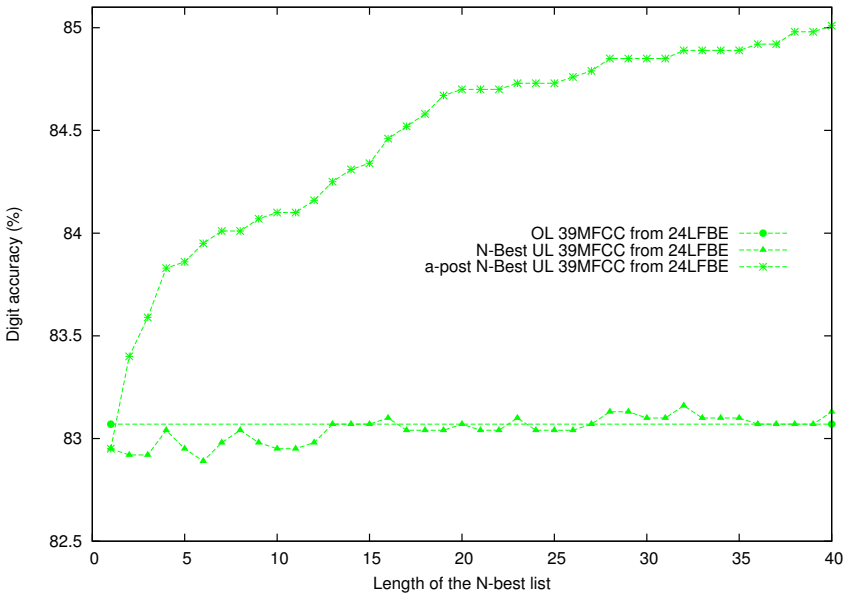


Figure 3.18. Performance of OL, N-best UL and *a posteriori* N-best UL on the HIWIRE test set. The positive slope of our approach is almost invisible: the N-best UL goes definitely over OL only after 27 transcriptions are considered in parallel. The relative improvement over UL is low (1 %)

compute:

$$STEP5 \qquad \hat{tr}_{f^*} = \hat{tr}_{2,f^*} = \underset{n \in G_{2,N-best}}{\arg\max} \; \mathscr{S}(\hat{tr}_{2,n}, tr_c) \tag{3.26}$$

which means that we simply select the transcription with the minimal Levenshtein distance from the correct one. Of course this distance is not available in practice, and can be determined only *a posteriori*. In the following, we will refer to this upper bound with label *a posteriori* N-best Unsupervised Limabeam. Clearly, by substituting the ML criterion with a maximum WRR criterion, it will be evident how much choices based on the value of the likelihood are beneficial for increasing recognition results. Figure 3.18 shows performances when the length of the N-best list was pushed to 40 elements: clearly the N-best UL, (depicted as a green, dashed line with triangles), supersedes the OL (green, dashed line with circles) only after 27 transcriptions. The general slope is positive, but it is smaller than what we observed in the case of a coherent noise field (see Figure 3.8(b)). In addition the *a posteriori* N-best UL (green, dashed line with stars), which has a tan^{-1} behaviour, shows that there is still a high potential for improvements.

	39from24	39from16
OL vs D&S	8.6%	14.8%
N-best UL vs UL	1.1%	4.6%
N-best UL vs D&S	9.0%	18.8%

Table 3.6. Relative improvement D&S, OL, and N-best UL, with two different pairs of Front-End. All the improvement grow from left to right, even if their absolute don't. It's interesting to note that the higher the OL is in absolute, the higher will be the gain of the N-best UL over UL and OL, meaning that an optimal Front-End really boosts the optimization.

3.5.4 Maximum Likelihood - Maximum WRR mismatch and complexity

The optimization is fruitful, though improvements are smaller than what observed when the noise field was coherent: the D&S beamforming, scores 81.47%, while OL and N-best UL are respectively 83.07% and 83.13%. Is that just a simple matter of environmental conditions? Or is that related with the optimization stage? We show that the reason lies in the optimization stage. So far we used 24 LFBEs for the optimization stage and 39 MFCCs for alignment and recognition. In Limabeam, optimization features are expressed as a function of the FIR filter taps by means of a Jacobian matrix (see Equation 2.13), the dimensions of which are $M \times L$ for rows and E for columns (E being the number of LFBE used). We consider the hypothesis that the Gradient descent algorithm is dealing with too many parameters in order to find a suitable global maximum, but getting stuck in local maxima degrades the performance. This is a crucial problem, where the noise conditions make the objective function more difficult. Also the high dimensionality of the optimization is another difficulty. Since we cannot modify our objective function, we try to improve convergence by reducing the number of parameters. If we decreased M, we would lose much of the gain determined by the microphone array, particularly in this kind of field. If we decreased L, we would negatively affect performances: 10 is already a low amount of taps. Thus, we can try to act on the Front-End. We are encouraged in doing so by the fact that, according to some researchers [Gillespie and Atlas, 2002], Speech Recognition results with Microphone Arrays could be increased in noisy environments when varying Front-End parameters such as the window size, the same parameters which are well known to be optimal when only a single-channel is used. We decrease the size of the LFBE vector to 16. Figure 3.19 shows the digit accuracy for the two pairs of Front-Ends (the pair is formed by an optimization and a recognition domain). Different pair are denoted by different colors and line type, while each technique is identified by a specific dot. We decided to show the *a posteriori* N-best UL out of the picture, in Figure 3.20.

Because we would also like to see the maximum achievable improvement, in Figure 3.20(a) we show how the N-best UL and the *a posteriori* counterpart follow each other. The maximum with an unsupervised method is reached at 84.92% absolute, while we could reach up to 86.22% if we found

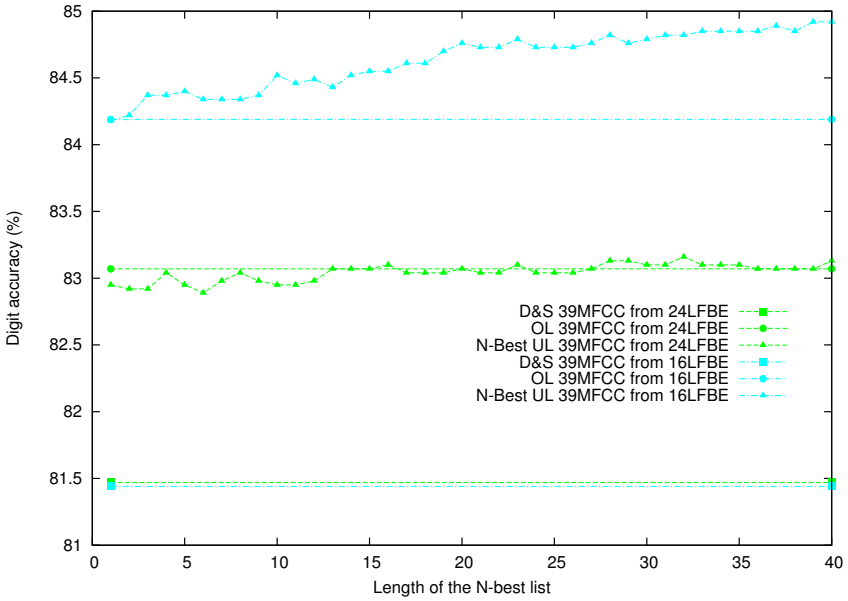
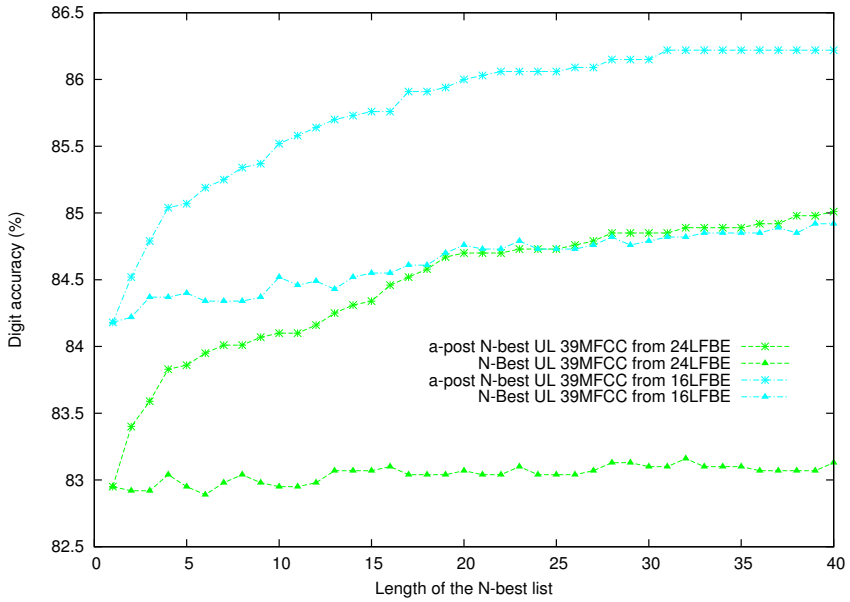


Figure 3.19. Performance of D&S, OL, and N-best UL, with two pairs of Front-End. The highest performance and the minimal WRR-likelihood distance is reached when 16 LFBEs are used for optimization and 39 MFCCs are used for alignment and recognition. The size of the LFBE feature vector has an influence on the effectiveness of the optimization stage.

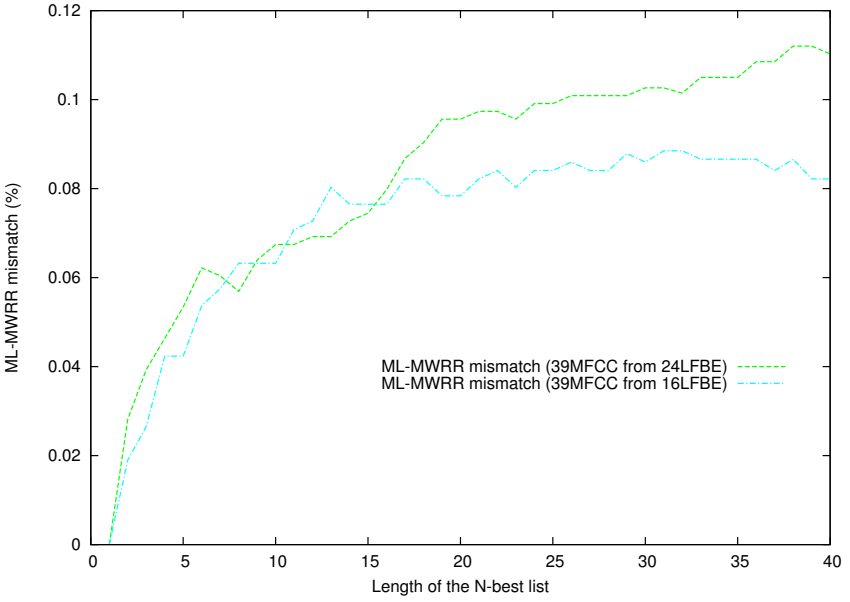
a more suitable criterion than ML to choose the best competitor. It indicates that there is still a margin of improvement for further research. Figure 3.20(b) simply plots the relative supervised-unsupervised improvement computed on the left figure: it shows the mismatch between the ML and the Maximum WRR criteria, which is the optimal. A negative slope indicates convergence between the two criteria. We can notice that the desired behaviour happens after 27 and 32 hypotheses when using 39 MFCCs with 16 LFBE respectively. This is a proof that a "bad" transcription, too different from the correct one, never becomes first on average , i.e. that increasing the length of the N-best list is always beneficial. We could vary other Front-End parameters to get higher performances, but this is behind the scope of this work. Here we just care to positively influence a Gradient descent-based search of maxima.

3.6 LFBE vs MFCC

The optimization has been kept so far in the LFBE domain. This has been described in Section 2.1.2. The author of Limabeam says [Seltzer, 2004] that LFBE are used because a Gradient descent-based algorithm would not benefit from the dominant magnitude of the first MFCCs. Instead, the LFBE have aproximately the same magnitude: it is desired that every energy Filterbank Energy band gets equal chances to contribute to the maximization of the likelihood.



(a)



(b)

Figure 3.20. Performance of N-best UL and *a posteriori* N-best UL, with two pairs of Front-End. The highest performance in a) is reached when 16 LFBEs are used for optimization and 39 MFCCs are used for alignment and recognition. The size of the LFBE feature vector has an influence on the effectiveness of the optimization stage. In b) we see the relative distance between the graphics in a): a negative slope indicated that the ML criterion is getting close to the Maximum WRR. This happens only when 16 LFBEs are used.

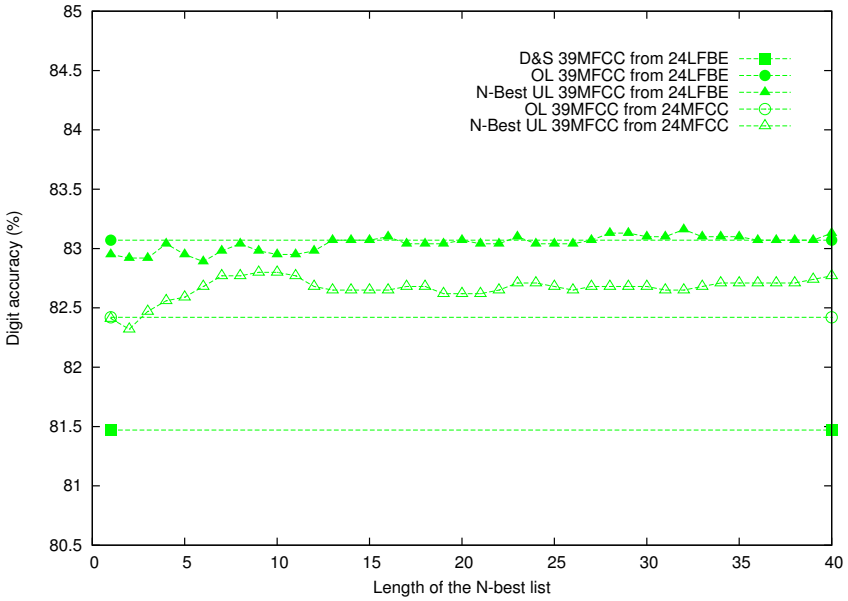


Figure 3.21. Performance of D&S, OL and N-best UL for two different optimization domains: optimizing with 24 LFBEs gives higher performances than optimizing with 24 MFCCs. Recognition is always performed using 39 MFCCs.

Thus we analyze the performance of the N-best UL and compare the two pairs of Front-Ends considered so far with the related counterparts, where the optimization is carried in the MFCC domain. This will solve any doubt concerning the best domain where to perform optimization. Figures 3.21 and 3.22 show the comparison between the two optimization domains. We kept the same accuracy range in order for them to be easily cross-compared by eye. Several aspects can be observed:

1. The OL with MFCC-based optimization is always inferior to the LFBE-based one. The D&S line is of course the same, as no optimization is done.

2. The shape of the curves is similar even if the two domains are completely different: in both domains the N-best approach is useful.

3. The MFCC-based optimization forces the curves to reach their asymptote earlier than the LFBE counterpart.

4. In both cases there is a positive bias when using LFBEs.

These results are in contrast with the work published in [Raab, 2004], where a version of Limabeam (it is not clear whether it is UL or OL, but this is not relevant) the optimization in the cepstral domain is shown to be more beneficial than the one carried in the energy domain. However, in the same work Limabeam performs worse than a standard D&S in a real environment, while we

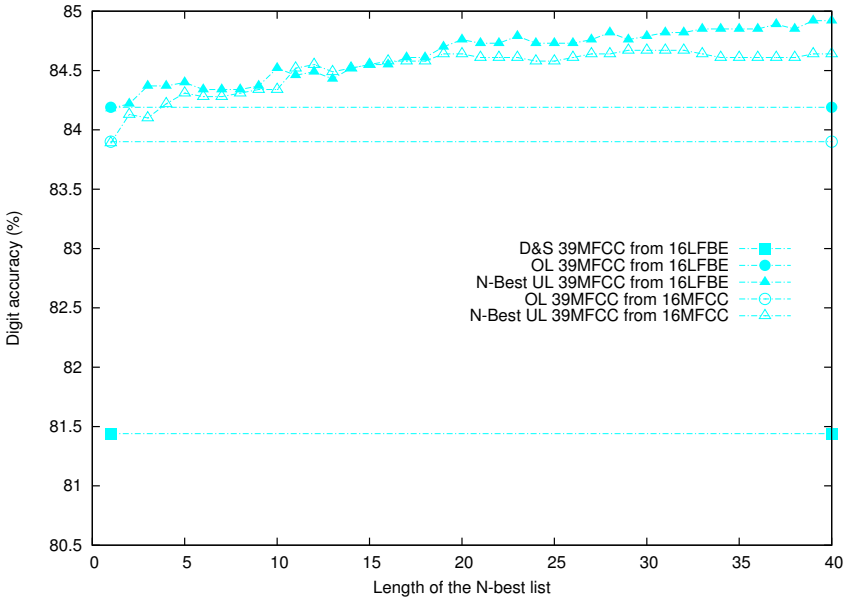


Figure 3.22. Performance of D&S, OL and N-best UL for two different optimization domains: optimizing with 16 LFBEs gives higher performances than optimizing with 16 MFCCs. Recognition is always performed using 39 MFCCs.

show that an improvement is always possible. We have seen that, independently on the domain used for optimization, the choice of the number of parameters used for optimizing the FIR filters of the N-best UL is crucial. We have found that decreasing the number of LFBEs leads to comparable performances when simple D&S is performed. In contrast, after N-best UL is applied the relative improvement with fewer (16) with respect to more (24) parameters is more than doubled (18.8% and 9.0% respectively over D&S). Results also show that a good way of proceeding to reach the highest performance is to find the right amount of parameters for optimization. In the next chapter the most challenging environment is tested: a highly reverberant meeting room. Among other aspects, we will see that the proposed N-best UL is able to improve recognition performance also in this scenario, especially if filters are initialized with Matched Filters.

Chapter 4

Optimal beamformers in reverberant environments

4.1 Introduction

The proposed N-best unsupervised Limabeam technique has shown significant improvements in noisy environments where the interfering signal is mostly additive. Results with both a coherent noise field in a simulated environment and a diffuse noise field in a real environment indicate that it is possible to find short ML filters which increase recognition results even with very low SNRs. However, the most challenging scenario is a reverberant room, where the following phenomena are present simultaneously:

- Speech is affected by the wall reflections and it is no longer possible to assume statistical independence between speech and disturbance.
- Speech is frequently uttered far from the microphone array, which is generally located on tables or walls. This increases the amount of echo in the speech signal.
- The speaker may vary position and head orientation while speaking: specifically, when the speaker is not aiming toward the array, speech captured by each microphone of the array will be mostly characterized by contributions due to reflections. This significantly affects recognition performance.

Because we intend to cover all such conditions, we construct a new environmental setup, described in Section 4.2. In the experiments conducted so far in this work, the array steering was

assumed perfect (in the case of non-coherent field) or were automatically computed via Cross-power Spectrum Phase (CSP, see Appendix A) (in the case of diffuse noise field). A CSP-based Time Delay Estimation finds the maximum coherence directions, where the speaker position could be found. However, it could even happen, as we experimentally show, that the best recognition is not obtained in the speaker direction estimated by the CSP, especially if sound sources are not facing the microphone array. It is thus interesting to study [Brayda, 2006a] how performance varies in function of different pointing directions: the analysis provided in Section 4.3 puts into evidence the primary and secondary directions where recognition scores are higher. We also discover that they correlate well with the peaks of the aforementioned CSP-based coherence measure. The problem to be solved is very different from the additive noise case, and in the following Sections we will show that:

- The proposed N-best UL shows improvements over conventional D&S that is significant but smaller than the additive noise case (section 4.3).
- Higher improvement is possible if, rather than estimating ML filters on line, one seeks to equalize the room via an off-line procedure. A new version of the Calibrated Limabeam, called Training-set Calibrated Limabeam (TCL) is proposed and tested in Section 4.4. ML filters derived from TCL are very short compared with the different room impulse responses. Consequently, they cannot physically take into account and compensate for early and late reflections.
- We investigate recognition performance with Matched Filtering (in Section 4.6) which accounts for these reflections, but requires prior knowledge of the impulse response. We show that this knowledge must be exact, and that neglecting even just the head orientation can dramatically decrease performance.
- Finally, we combine Matched Filtering, which is, to the best of our knowledge, the most effective way of increasing recognition in reverberant environments, with our proposed N-best Unsupervised Limabeam, well suited to additive noise, and we show that this coupling leads to further improvements.

4.2 Environmental setup and task

The experimental setup consists of the same 1001 sentences of the test set “A N1”, used with fan noise in Chapter 3: the TI-DIGITS signals have been reproduced by a high-quality Tannoy loudspeaker in the 6 $\times$ 4,8 m CHIL room available at ITC-irst and acquired at a sampling frequency

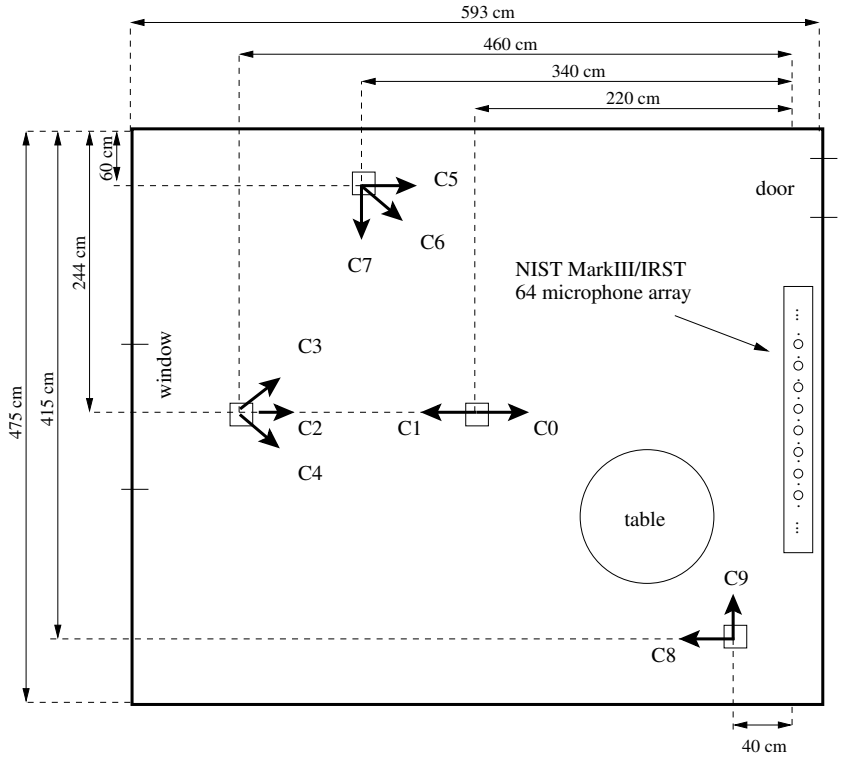


Figure 4.1. Map of the ITC-irst CHIL room (6m $\times$ 5m), reporting on positions of array and acoustic sources.

of 44.1kHz by means of the Mark III board. This test set, referred to in the following as CHILREVERB, has been evenly divided in subsets, varying position and orientation of the loudspeaker with respect to the array for a total number of 10 different configurations. Figure 4.1 identifies in the room map the 10 subsets, indexed by C0 to C9. The database mimics three main speaker positions, and different head orientations for each positions. The distances from each speaker positions to the walls are also reported. As a result the SNR, evaluated at one microphone of the array, varies from 10 to 25dB, depending on position, orientation and energy of the original signal. The T60, varying in function of the position and orientation, is approximately 700 ms. This high reverberation time is mostly originated by the walls: reflections from the ceiling and the floor can be neglected because of the materials they are made of.

4.3 Delay-and-sum and angle-driven beamforming

In this section we study the role and impact of reflections on recognition performance. First, we analyze how performance varies in function of the “look” direction, through an angle-driven beam-

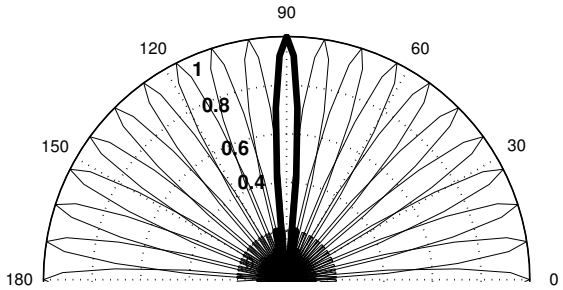


Figure 4.2. Amount of the π-space (see Figure 4.2) spanned by a microphone array with M=8, d=0.04m, f_{max}=7500 Hz and steered for 19 different angles. The main lobe of a single beampattern appears in bold. Sidelobes, not plotted, appear only close to 0° and 180° for high frequencies.

former. This kind of beamformer is not providing the maximal performance, but indeed it is useful to locate regions in the rooms where speech is better recognized. We also apply N-best UL to this beamformer. Second, a more efficient beamformer, driven by the CSP, is tested. We find a relation between the two beamformers. Third, we couple CSP-driven beamforming and N-best UL and report on performance in all positions.

The aim of the angle-driven beamformer is to set the delays $\tau_m = \frac{m d \cos\theta}{c}$ for each microphone. Being the purpose to form a beam at a specific direction, given θ, a different set of delays can be computed for each desired angle. In this work we focus on spanning the π-space in front of the microphone array and look at equi-angled directions. For each direction we "steer" the array to a specific angle θ, then beamforming (theta-D&S) and recognition are performed: this results in getting a Recognition Directivity Pattern (RDP), the main lobes of which will "point" to regions where signals are better recognized. In order to cover all the space in front of the array, while limiting aliasing, we propose to set the number of beams R to:

$$R = \frac{\pi}{\arg_\theta D_{-3dB,l}(f_{max},\theta) - \arg_\theta D_{-3dB,r}(f_{max},\theta)} \tag{4.1}$$

where f_{max} is the maximum frequency of interest and the denominator is the main lobe width when the lobe attenuation is -3 dB, which is the distance in radiants between the point to the left $D_{-3dB,l}$ and to the right $D_{-3dB,r}$ of the main lobe peak at -3 dB. Thus, steering the array results in beamforming as depicted in Figure 4.2, where we considered a sub-array of 8 microphones, with 4 cm inter-microphone distance d = 4cm and a maximum frequency of 7500 Hz. This setting ensures aliasing to be almost negligible in the band spanned by a Mel Filterbank (actually grating lobes appear when steering the array to endfire positions, but only for the highest examined frequencies) and also limits the system complexity (the more the microphones, the higher the number R of beams

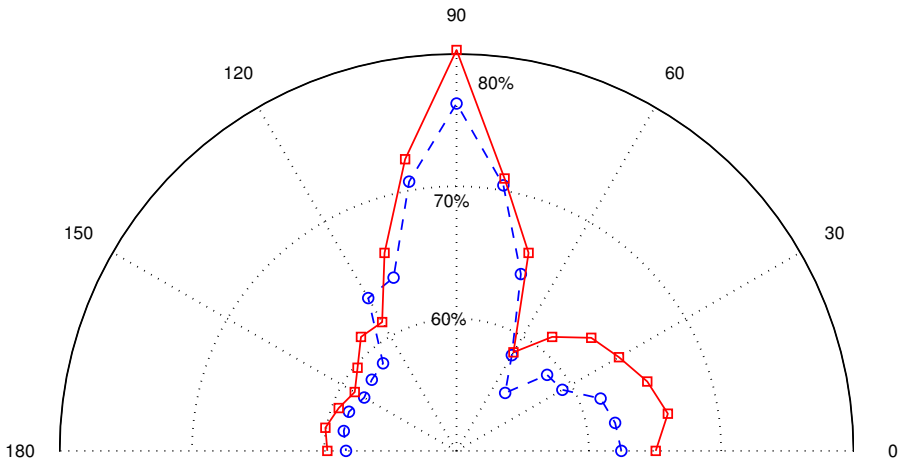


Figure 4.3. Polar Recognition Directivity Pattern when speaker is in configuration C0: the array points with a very narrow beam toward the speaker. Performance of CSP-D&S performance is comparable to the highest value of the main beam. The pattern magnitude is measured in WRR, starting from 50%.

of Equation (4.1); the higher the distance d, the higher the grating lobes).

The RDPs of all the ten configurations are commented in the following. Figures from 4.3 to 4.12 can be compared to Figure 4.1 to understand the relation between the RDP peaks, the location of the speaker and of the main wall reflections. The plots, singularly considered, clearly show the regions in the π-space where it is convenient to point the microphone array. A theta-driven D&S gives performance depicted as a dashed, blue line. Furthermore, applying N-best UL for each direction leads to the performance depicted as a solid, red line.

Peaks of the RDP

More globally, in this room two opposite behaviors of the RDP can be distinguished:

- The highest performance of the theta-D&S, which is the top value of the main RDP peak, is obtained by pointing in the direction of the speaker. This can be noticed in C0,C2,C5,C6,C7,C8 and C9.
- The highest performance is obtained, instead, by pointing to the main reflections. This happens in C3,C4 and partially in C1.

The first behavior can be further distinguished in two sub-cases:

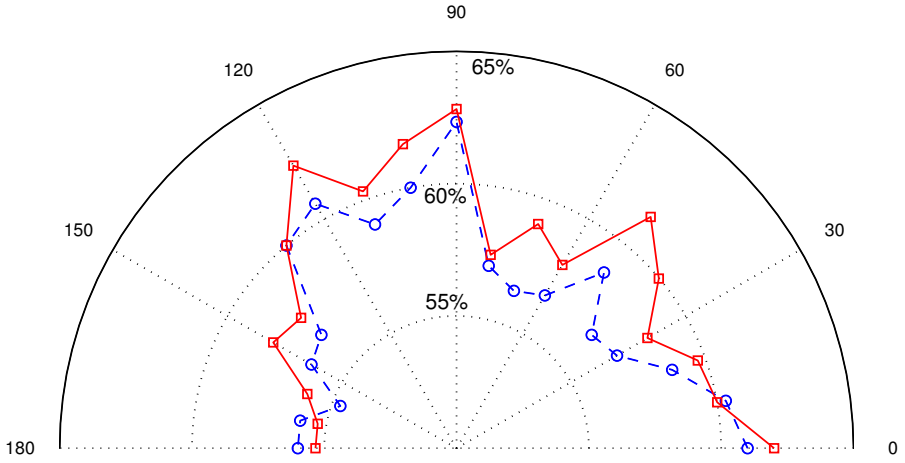


Figure 4.4. Polar Recognition Directivity Pattern when speaker is in configuration C1: because the speaker is pointing to the window, reflections are scattered, and are not clearly distinguishable from the RDP. There is no preferential direction for recognition. However it is still convenient to point the array toward the speaker: here the RDP has the highest magnitude.

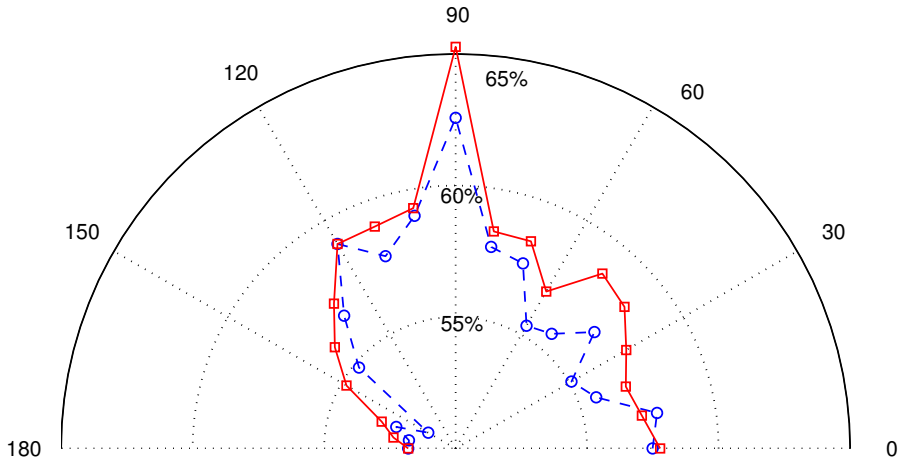


Figure 4.5. Polar RDP when speaker is in configuration C2. The array points with a very narrow beam toward the speaker, while smaller sidelobes between 0° and 60° collect minor reflections.

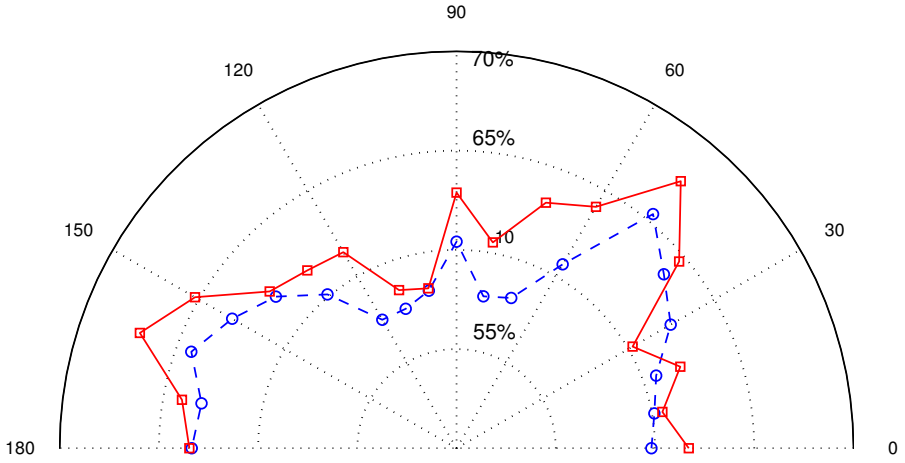


Figure 4.6. Polar RDP when speaker is in configuration C3. Surprisingly, the highest recognition rate correspond to the two main reflections, detectable when pointing the array to 50° and 160°. The CSP-D&S has the same performace than the highest 50° peak.

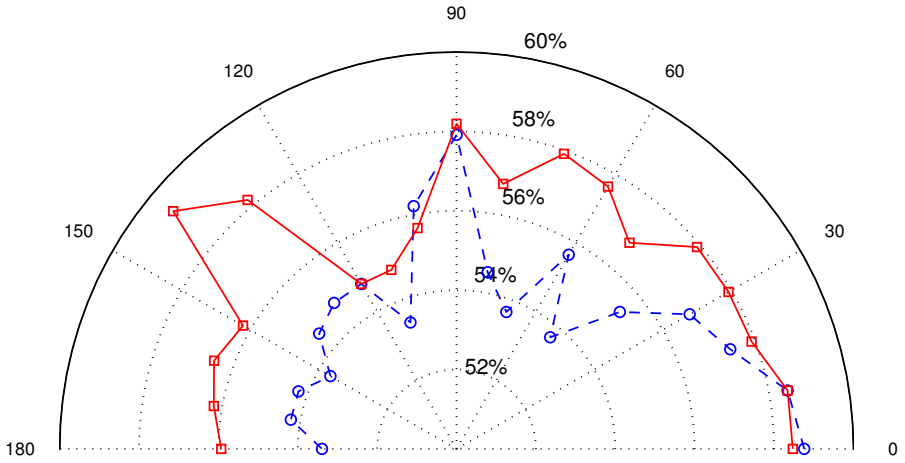


Figure 4.7. Polar RDP when speaker is in configuration C4. Similarly as C3, the highest recognition rate correspond to the main reflection, detectable when pointing the array to 140°. The second-highest peak correponds to 0°, in the direction of the speaker. As in C3, the CSP-D&S has the same performace than the highest 50° peak. Applying N-best UL is crucial to detect the main reflection: note that a simple theta D&S can't detect it.

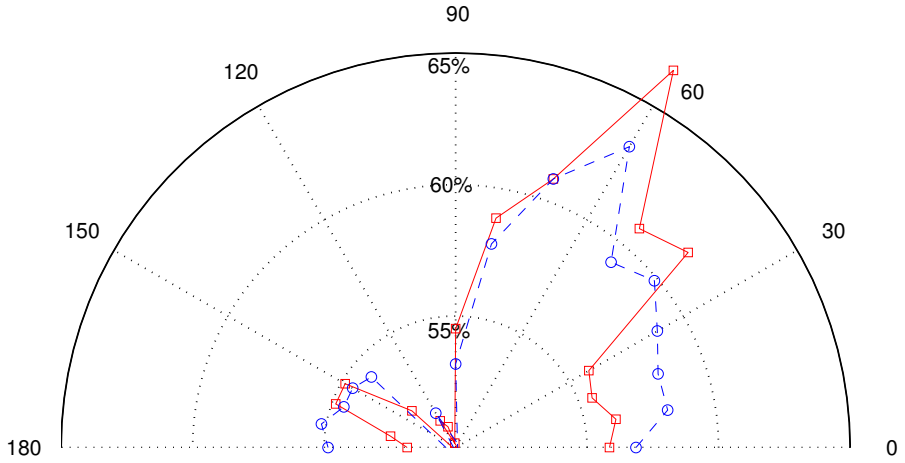


Figure 4.8. Polar RDP when speaker is in configuration C5. The RDP definitely points toward the speaker, which in turns faces the door. Early reflections on the closer side wall are beneficial between 30° and 60°. N-best UL is very effective in the most relevant direction.

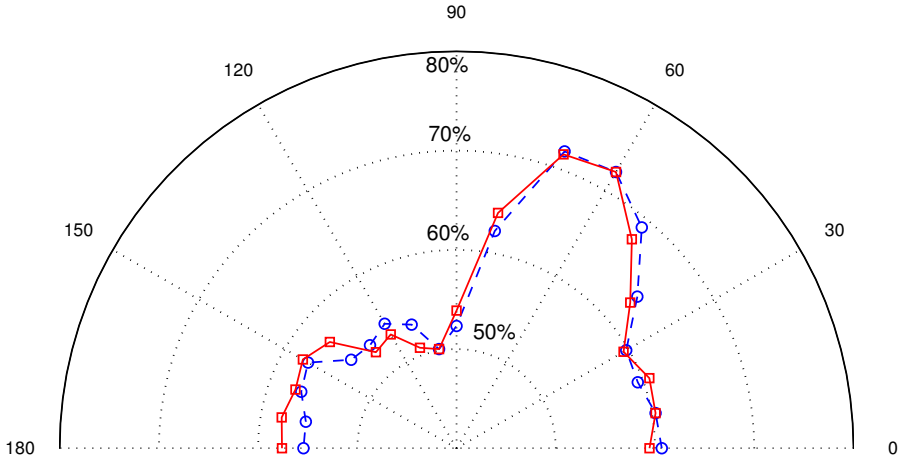


Figure 4.9. Polar RDP when speaker is in configuration C6. The RDP points with a large beam toward the speaker. Very small sidelobes between 120° and 180° collect minor reflections.

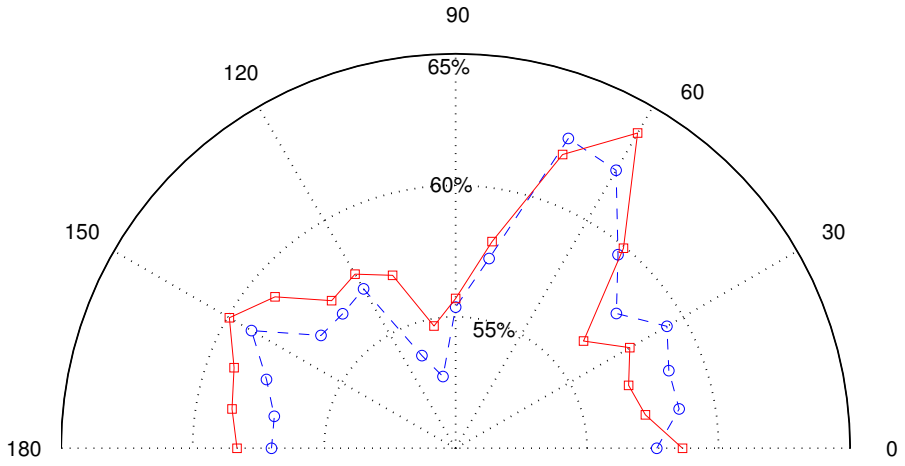


Figure 4.10. Polar RDP when speaker is in configuration C7. The RDP points toward the source, located at 60°, but a large lobe 'seeks' the main reflection at 150°. In this configuration the CSP-D&S points to the latter recognition lobe, which is related to a CSP peak with more coherence but less impact on recognition performance.

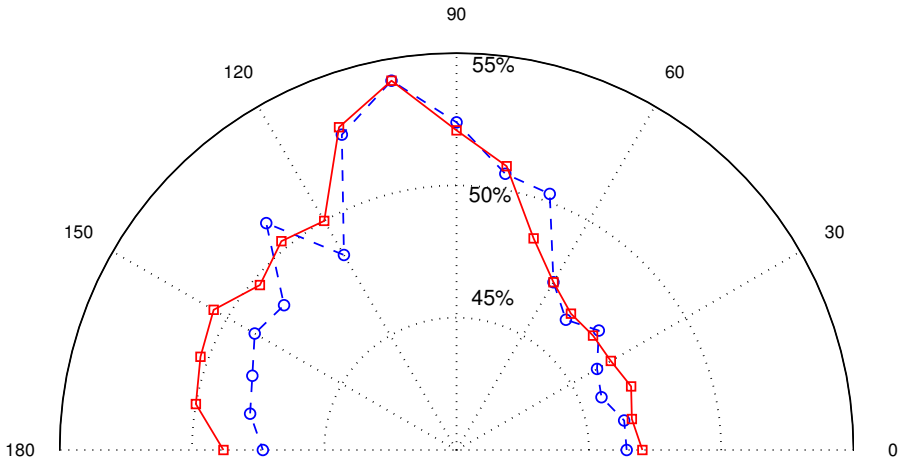


Figure 4.11. Polar RDP when speaker is in configuration C8. The RDP definitely points to the main reflection, far on the opposite wall, though the speaker is very close to the array.

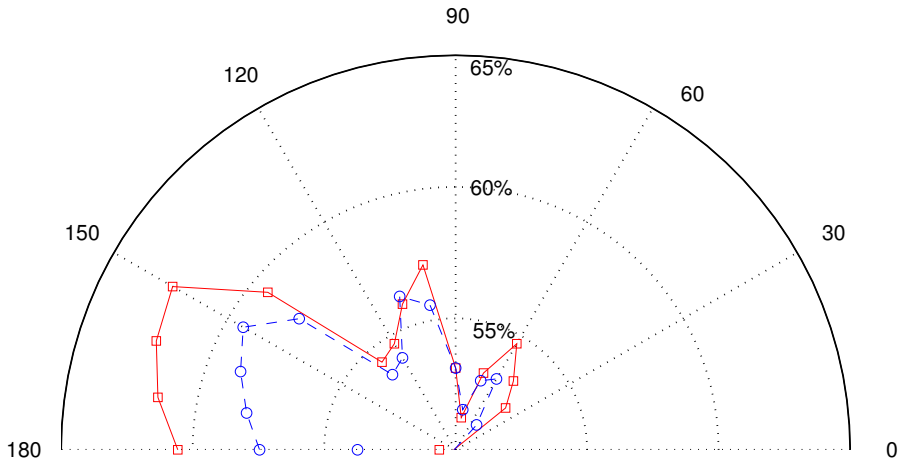


Figure 4.12. Polar RDP when speaker is in configuration C9. The RDP points at the speaker (in this case N-best UL is effective), but two lobes collect the contribution of the correspondent main reflections.

- The speaker is pointing to the array: the situation is so favorable that the obvious solution is to point the array towards the speaker (C0,C2,C5,C6).
- The speaker is pointing completely away from the array: the reflections contain so much echo that the corresponding recognition performance is low (C7,C8,C9).

The second behavior, is found when the first *reflection* from the speaker is directed to the array. This reflection contains less reverberation than any other multi-path and causes the highest recognition rates.

In all the scenarios depicted the N-best UL is effective, and the best relative improvements over theta-D&S are obtained on the direction with maximum recognition rate.

Efficiency of a CSP-D&S

It is desirable to let the array automatically point to the RDP peaks. This is generally achieved by using the CSP-D&S. Using a theta-driven beamforming allows to easily plot the RDP, but it does not provide explicitly the speaker location. Thus, the best starting point for optimization is generally a CSP-driven D&S, for several reasons:

- Estimation (and possibly tracking) of the speaker location in possible.

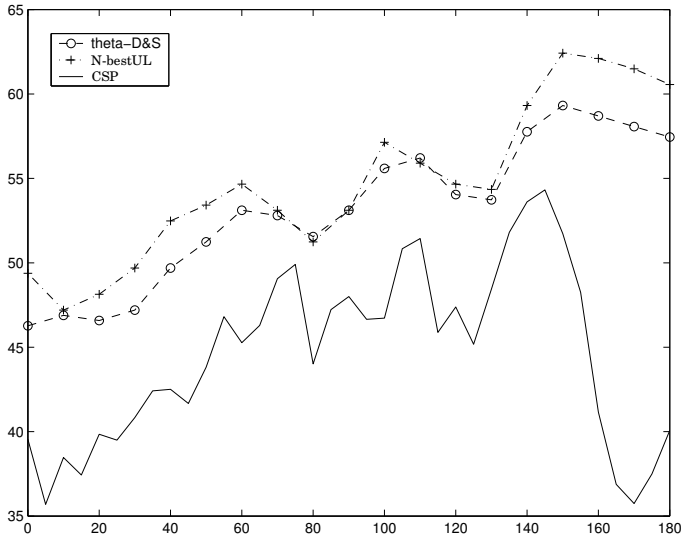


Figure 4.13. RDP in Cartesian Coordinates for configuration C9: the RDP peaks are well related to the main CSP peaks. CSP peak heights were normalized for plotting purposes only.

- Establishing *a priori* the pointing direction does not take into account the coherence among multi-channel signals
- Exact delays with sub-sample precision are extracted from the CSP, while with theta-driven D&S we used integer delays
- delays extracted via CSP can possibly account for the near field, while using just one pointing direction implies a far field assumption

Relation between theta-D&S and CSP-D&S

Apart from C7, if recognition is performed after a CSP-D&S, we get at least the maximum values of the RDP pattern main lobes: very frequently, across the ten positions, CSP-D&S recognition score coincides with the maximum value of the main RDP lobe. This establishes a relation between the two beamformers, i.e. the CSP-D&S is a specific theta-D&S (generally) pointing to its maximum. The relation between the shape of the RDP and the reflections detected by the CSP coherence measure, is definitely confirmed when analyzing C9 in Figure 4.13, where we plot both of them, properly rescaled, in Cartesian Coordinates. Angles where we find peaks of the coherence measure correspond to angles where we find peaks of high recognition accuracies. However, the same cannot be said of the relative magnitude of such peaks. In fact in C7 a theta-D&S performs better than a CSP-D&S, because the former directs the beamformer to the weak-coherence path, which is more relevant from a recognition oriented perspective than the strong, main reflection. Because in this

particular configuration the information of the CSP main peak was partially useful, we tried to automatically select the two main peaks, sentence by sentence (this was done simply by finding the maxima of the CSP function with linear regression and zero crossing of the first CSP discrete derivative) and we achieved the single-channel performance, which is roughly the average of the two main peaks performance.

Being scores higher for CSP-D&S, N-best UL can be performed by initializing it with the computed delays. Figure 4.14 shows the WRR in function of the 10 test positions, while Table 4.1 reports details and relative improvements. Several aspects can be noticed:

1. CSP-D&S is in general beneficial, but eight microphones only increase performance from 59.32% to 63.67%. With additive noise the gain would have been far higher. This underlines the sub-optimality of D&S in such environment.

2. N-best UL is able to overcome the CSP-D&S in all cases except in C1, C3, where the speaker is not facing the array

3. While single-channel performance is comparable for both C3 and C4, after beamforming there is a high difference: the RDPs of these two positions are similar in shape, but different in magnitude. Very different scattering of the reflections, for example on the table, can be the only possible explanations for such a behavior.

4. The highest relative improvements of N-best UL are obtained in C2,C5,C6, while in C0 much of the improvement is easily obtained by the CSP-D&S. This method is able to increase recognition accuracy even when the speaker is very far from the array and can recover D&S errors.

We conclude that when the speaker is not facing the array, the maximum WRR directions are frequently corresponding to main reflections. This hints that the amount of well recognized information carried by the reflections is very high.

4.4 Training-set Calibrated Limabeam

In the previous section the proposed N-best UL was shown to better compensate for reflections than a D&S beamformer. However, we intend to study how much a likelihood-based system is able to "learn" the environmental conditions. Such a problem cannot be solved by means of algorithms like the N-best UL, which optimize filters on a sentence-by sentence basis. We show that optimization of the filters can be efficient, especially in a reverberant environment, if more data is observed. In

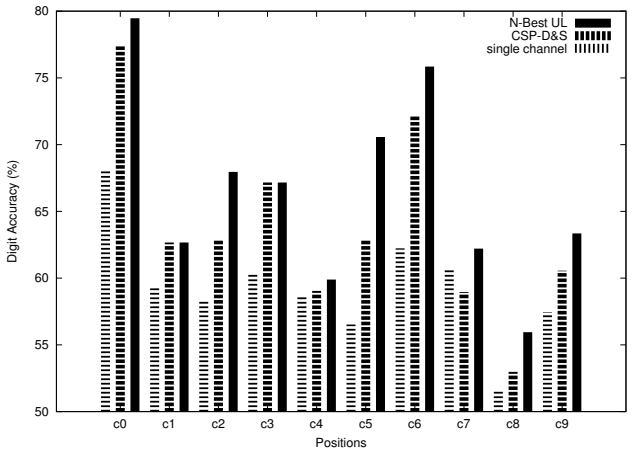


Figure 4.14. Baseline results: Digit Accuracy (%) in the 10 test position using single-channel, CSP-D&S and the proposed N-best UL.

config	single mic.	CSP-D&S	CSP-D&S+N-best UL	RI
C0	68.04	77.42	79.47	9.1
C1	59.33	62.67	62.67	0
C2	58.38	62.87	67.96	13.7
C3	60.41	67.16	67.16	0
C4	58.76	59.04	59.89	2.1
C5	56.52	62.88	70.57	20.7
C6	62.23	72.14	75.85	13.3
C7	60.59	58.96	62.21	7.9
C8	51.49	52.98	55.95	6.3
C9	57.45	60.56	63.35	7.1
Average	**59.32**	**63.67**	**66.51**	**7.8**

Table 4.1. Table reporting WRR and Relative Improvements (of N-best UL over CSP-D&S, in %) for the 10 test configurations; Average is the overall WRR over the 10 configurations .

fact in [Seltzer, 2004] the Calibrated Limabeam (CL) shows improvement comparable to the UL in real environment with limited reverberation (T60 = 240 ms) and with a very small speaker-to-microphone distance (about 0.4 m). In CL, filters are optimized only once on several seconds of speech, then the FIR filter set is kept fixed for the rest of the task. We propose to extend this method by using pure reverberated data for optimization: instead of calibrating the set of filters on a sentence extracted from the test set, we use some seconds of speech from the Training set (the sentence length varies with the sentence subset, having one subset for each direction) and convolve them with a set of room impulse responses which do not match the test conditions (for example, filters are trained with impulse responses of configurations C0, but tested with data spoken in C5). The objective is to derive a set of filters which improves performance independently on the position of the speaker. We find that, for sufficiently short filters, the recognition performance is independent on the set of room impulse responses used for performing the proposed Training-set

Calibrated Limabeam (TCL). The filter length has been limited to 10 taps, as we did with the additive noise case. Figure 4.15 reports the WRR of a TCL as a function of the number of taps.

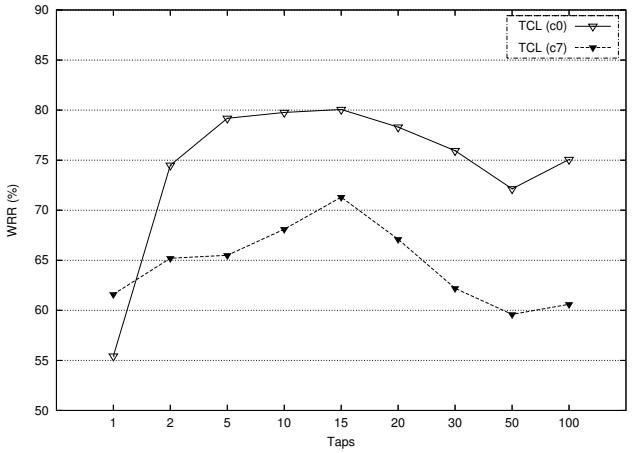


Figure 4.15. WRR (%) for TCL technique as a function of the filter's length.

The filters of this experiment were trained on 10 seconds of speech, balanced across gender and content [1]. Positions C0 and C7 were chosen because, as seen from previous sections, they behave very differently. Results indicate that 10-15 taps are most suitable to the offline calibration, as we have seen in the case of additive noise: though that is a completely different environment to solve, the constraint on the filter lengths still holds and constitutes a necessary condition to obtain significant improvement. Limiting the amount of parameters to optimize remains an unavoidable constraint.

Since we want to be independent from the speaker location, the room impulse response used for the training will be randomly chosen among all impulse responses (averaging). However, we could expect better results if training and test are made with the same location. Figure 4.16 shows the opposite. There is almost no dependency on the training position. A diagonal maximal ridge cannot be obtained, but it would correspond to an attempt of compensation of the room reflections with short FIR filters. However CL and TCL provide better results than N-best UL.

This also means that fixing the filter set and then changing position or just orientation results in poor recognition performance. In case of moving speakers, N-best UL better addresses this issue. Furthermore, we compare TCL and CL in Figure 4.17.

While for TCL we use the 10 seconds of speech previously mentioned to estimate and freeze the

[1]The 10-seconds is the concatenation of sentences MBD_5Z68A MCF_9123A MNW_7944A FBH_5734A FCG_8125A FDC_6409A.

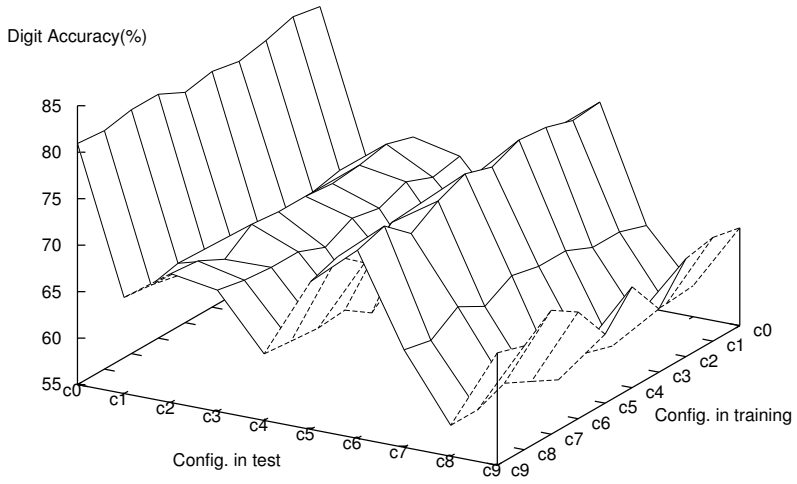


Figure 4.16. WRR for filters trained for 1 second in one position (rows) and tested in another (columns). Results in Digit Accuracy.

FIR filter set, for CL we use the first sentence of each mini-test set (the first of C0, the first of C1 and so on), as done in [Seltzer, 2004], the length of which is variable. Globally using TCL is slightly more beneficial (68.36%) than CL (67.31%). Using pure reverberated speech made the optimization on average more efficient. TCL holds the best performance achievable in this room without prior knowledge of the precise location of the speaker. However, across position TCL is preferable if the speaker points to the wall where the array is mounted on (C0,C2,C5), while CL is better in less favorable cases.

From this analysis it is clear that any method aiming at maximizing the likelihood cannot directly face the reverberation problem. This is due to the constraint on the filter tap set, which prevents to adequately compensate the room impulse response. Clearly we need a method providing longer filters for this purpose. We have seen, though, that a method like N-best UL is able to achieve high improvement if the array is correctly pointing to the speaker or the main reflections. This hints that the problem of compensating for reverberation may be solved *before* any kind of ML optimization. In order to do that, the initialization point of the optimization should be the most favorable possible: thus we cannot relay on a D&S beamforming, because of its limited improvements. In the next section we show that the optimal filters to initialize the N-best UL in a reverberant environment are the Matched Filters.

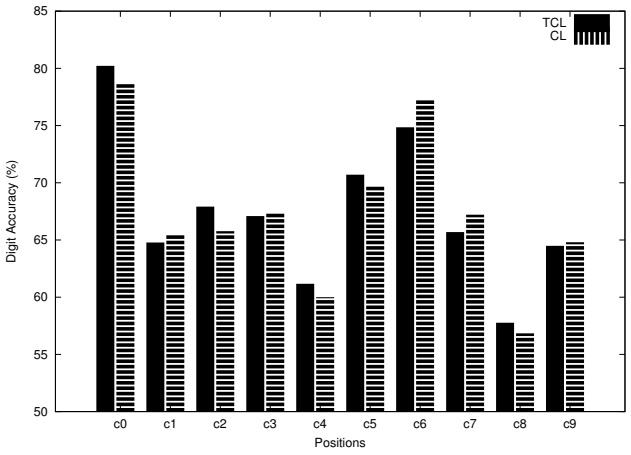


Figure 4.17. WRR (%) in the 10 test positions for TCL and CL. Results of TCL exclude conditions of perfect match between training and test impulse response.

4.5 Matched Filtering

The techniques presented so far assume the beamformed signal being of the form of Equation 2.2, here reported for the sake of clarity:

$$y(t) = \sum_{m=0}^{M-1} w_m(t) * x'_m(t) \tag{4.2}$$

where we recall $x'_m(t) = x_m(t - \tau_m)$ is the signal received at microphone m, the delay τ_m of which has been compensated, for example, via CSP, and $x_m(t) = s(t) * h_m(t)$ is the reverberated speech. We recall that Matched Filtering (see Section 1.7.4), instead, finds:

$$y(t) = \sum_{m=0}^{M-1} h_m(-t) * x_m(t) \tag{4.3}$$

where, by construction, $x_m(t)$ does not need TDC, because convolution with time flips of impulse responses automatically provides it. MF realigns not only the primary delay (usually associated to the direct path) but also the secondary delays. For speech recognition purposes, it has been shown [Flanagan, 1993] that optimal filters come from a truncated version of $g_m(t) = h_m(K - t)$, in order to avoid anti-causal echo effects after convolution.

We verify that by finding the filter length K, in samples, maximizing the WRR: Figure 4.18 shows the WRR in function of the MF length: depending on the position, the peak in accuracy is reached for different lengths. It is worth noticing that this length may well correlate with the relative T60: the MF is effective once it includes the direct path and the main reflections. In c0 these reflections

are 1000 taps away from the direct path and the accuracy curve slowly lowers down, while for c1 the optimal length is around 3000 and for c8 8000, which means there are useful (from a recognition point of view) reflections at about 70 and 180 ms respectively from the direct path.

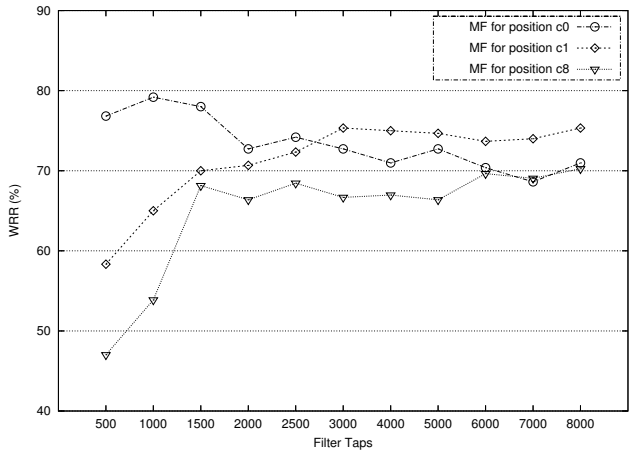


Figure 4.18. Performance of Matched Filtering as a function of number of taps.

A MF length of 1500 taps is thus chosen, as it guarantees medium-high recognition score in all the configurations. We propose to couple Matched Filtering and N-best UL by initializing the filters to:

$$g'_m(t) = g_m(t) * w_m(t) \tag{4.4}$$

where $w_m(t)$ is the filter estimated with N-best UL at the m-th microphone and $g_m(t)$ is the truncated, time reversed impulse response for that microphone. Because optimization can handle only a limited amount of parameters, at each step only $w_m(t)$ is modified and the signal re-beamformed using the updated $g'_m(t)$. The short filter has thus the purpose of modifying the truncated impulse response according to a ML criterion. Performance of MF and MF coupled with N-best UL are shown in Figure 4.19:

Clearly the use of MF prior to N-best UL dramatically improves performance with respect to any kind of D&S beamforming, particularly in non-favorable positions such as C1,C4,C7,C8, where the speaker never points to the array. This is a remarkable result. Furthermore, comparing relative improvements, we can see that the contribution of the N-best UL strongly depends on the initialization configuration: in Table 4.2 the behaviour of N-best is almost specular in the two cases: if the improvement is below its average (7.8%) without MF, it will be above once its average (12.6%) MF is performed. This is the strongest proof that MF, once it mostly compensates for reverberation

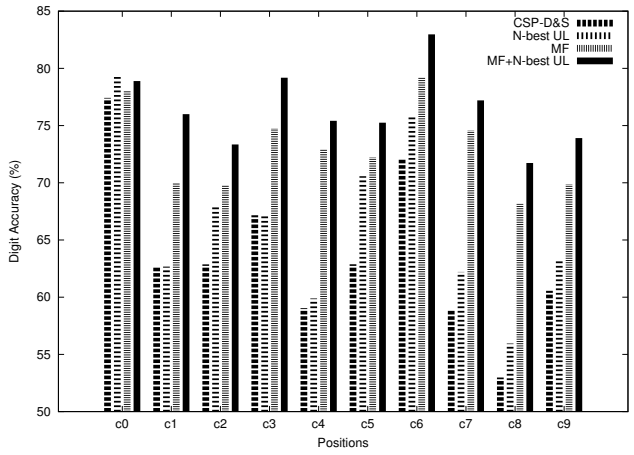


Figure 4.19. Performance of CSP D&S, N-best UL, Matched Filtering and the combination of Matched Filtering and N-best UL.

effects, lets the N-best UL find a more suitable set of filters for speech recognition

beamformer	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	**avg**
RI N-best wrt CSP-D&S	9.1	0	13.7	0	2.1	20.7	13.3	7.9	6.3	7.1	**7.8**
RI MF+N-best wrt MF	4.0	20.0	11.8	17.4	9.3	10.8	17.8	10.2	11.2	13.3	**12.6**

Table 4.2. Comparison of relative improvements (%), showing the contributions of the N-best UL when starting from a CSP-D&S configuration and when starting from a MF configuration. Speaker orientation is the most influencing factor.

4.6 Matched and unmatched filtering

MF requires however perfect knowledge of the speaker-to-microphone impulse response set. It is indeed an upper bound for all performance measured in the CHIL meeting room. It is interesting to find up to which point it is possible to switch the $g_m(t)$ filters from one position to another, as we did with TCL.

Figure 4.20 and Table 4.3 show that performance drops down dramatically in off-diagonal couples, i.e. when filters not matching the same conditions are used. Instead, they are very high on the diagonal, where the filters are Matched. What we actually do is to create an "unmatched filter". This indicates that the knowledge of the speaker location is not enough to compensate for reverberation, but knowledge about the head orientation is needed: results for C2,C3 and C4, which are in the same location but 45 degrees apart from each other, are high only in matched condition.

	$h_m(t)$ from										
$g_m(t)$	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	aver
c0	**78.01**	54.67	60.48	54.84	57.34	47.49	51.39	51.47	45.54	44.10	54.53
c1	59.82	**70.00**	50.00	46.33	50.56	44.82	45.51	48.86	43.45	45.96	50.53
c2	65.69	50.67	**69.76**	54.84	56.50	49.16	55.42	52.44	42.56	43.48	54.05
c3	63.34	51.67	50.90	**74.78**	55.65	50.84	54.80	55.05	43.75	45.65	54.64
c4	70.09	52.33	60.78	58.06	**72.88**	47.83	54.18	52.77	43.45	45.96	55.83
c5	57.48	47.67	46.41	53.67	48.02	**72.24**	69.66	60.26	38.99	44.10	53.85
c6	57.77	49.67	48.50	50.73	48.59	63.88	**79.26**	64.17	42.56	41.61	54.67
c7	58.36	48.00	51.20	51.32	49.72	56.52	70.59	**74.59**	39.88	45.96	54.61
c8	58.94	47.33	44.61	48.97	46.89	39.13	45.20	45.60	**68.15**	52.17	49.70
c9	52.49	44.67	42.81	45.16	48.59	41.81	50.15	51.79	49.40	**69.88**	49.67
aver	62.20	51.67	52.55	53.87	53.47	51.37	57.62	55.70	45.77	47.89	53.21

Table 4.3. Accuracy for unmatched filtering for varying tap length with IR in one position (rows) and tested in another position (columns). Matched Filtering is on the diagonal, where train and test position are the same. It is evident that testing with another IR is not beneficial.

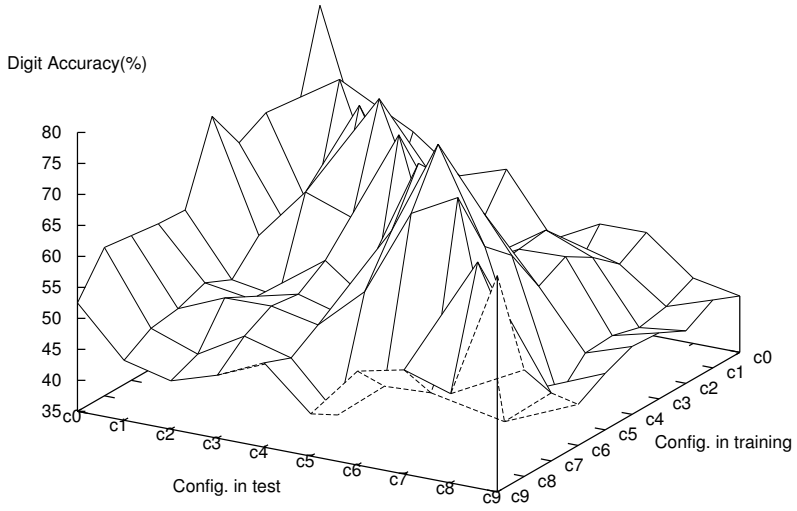


Figure 4.20. Matched and unmatched filter

beamformer	*taps*	*seconds*	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	aver
CH33	-	-	68.04	59.33	58.38	60.41	58.76	56.52	62.23	60.59	51.49	57.45	59.32
CSPD&S	-	-	77.42	62.67	62.87	67.16	59.04	62.88	72.14	58.96	52.98	60.56	63.67
UL	10	variable	78.89	61.67	65.87	67.45	60.45	67.89	73.37	61.89	55.36	62.73	65.56
N-best UL(20)	10	variable	79.47	62.67	67.96	67.16	59.89	70.57	75.85	62.21	55.95	63.35	66.51
RIwrtUL	-	-	2.75	2.61	6.12	-0.89	-1.42	8.35	9.31	0.84	1.32	1.66	2.76
RIwrtD&S	-	-	9.1	0	13.7	0	2.1	20.7	13.3	7.9	6.3	7.1	7.8
CL	10	variable	78.74	65.42	65.77	67.35	60.00	69.73	77.26	67.21	56.84	64.80	67.31
TCLunm	10	10	80.22	64.78	67.92	67.09	61.17	70.71	74.85	65.69	57.77	64.49	67.47
MF	1500	-	78.01	70.00	69.76	74.78	72.88	72.24	79.26	74.59	68.15	69.88	72.96
RIwrtD&S	-	-	2.61	19.64	18.56	23.20	33.79	25.22	25.56	38.08	32.26	23.63	25.57
MF+N-best(20)	1500+10	-	78.89	76.00	73.35	79.18	75.42	75.25	82.97	77.20	71.73	73.91	**76.39**
RIwrtMF	-	-	4.0	20.0	11.8	17.4	9.3	10.8	17.8	10.2	11.2	13.3	12.6
RIwrtD&S	-	-	6.5	35.7	28.2	36.6	40.0	33.3	38.9	44.4	39.9	33.8	35.0

Table 4.4. Summary table. Accuracy for all positions. TCL unm is "leaving one out", i.e. it does not consider matching training and test conditions

4.7 Discussion

In this chapter we have investigated the use of microphone array processing with real data in a very reverberant room, analyzing the impact of different beamforming techniques on recognition performance. Several beamforming techniques based on inter-channel delay handling (theta-D&S, CSP-D&S) and on a likelihood-based filter-and-sum beamformer (CL, TCL, N-Best UL) have been tested: the overall performance is shown in Table 4.4. The way we attempt to face the reverberation problem suggests the following conclusions:

- Critical aspects are the correct estimation of the inter-channel delay and the initialization of the filters.
- Performance is relatively high when the speech source is directed to the sensors as well as the array is steered toward the source, but in this case it is very sensitive to steering errors. To cope with these errors, a CSP-driven beamformer can automatically locate the useful wavefront.
- On the other hand, when sources and microphones are not faced to each other, which mimic, for example, speakers with different head orientations, there is a direct correspondence between the peaks of the CSP and RDP figures.
- High performance obtained by coupling MF with the proposed likelihood-based beamformer indicate that the margin of improvement is still high for techniques aiming at finding a filter optimum from the recognition point of view.

Finally, two important features of the reverberation phenomenon revealed to be crucial for recognition performance: the length of the filters and its *a priori* knowledge. If no knowledge of the

impulse responses is provided, only short filters can improve recognition scores under a ML criterion (computing longer filters leads to convergence problems). They can be trained randomly on a set of impulse responses: TCL obtains 67.47%, against the 63.67% of the best kind of D&S beamformer. The offered improvement is a 10.5% relative. Unfortunately, the few number of taps used do not allow to consider the main reflections at the current sampling rate. On the other hand, if this knowledge is provided, long filters can be used to compensate for reverberation, then complementary short ML filters can be applied effectively to possibly remove additive noise and reach an average WRR of 76.39% (35% relative to D&S). This represents an upper bound for performance, but it is also a very high result considering the reverberation time of the room and the fact that speakers are on average 3 m distant from, and most of the times not facing, the microphone array.

4.8 Future research

Future work can be directed to establish a criterion for selecting higher recognition lobes independently from speaker location and orientation.

Because Matched Filters can't be switched, a method which can automatically select, based on different confidence measures, the correct Matched Filter, sentence by sentence, would be desirable. Also given such method, one should cope with the available amount of impulse responses: their diversity (especially concerning head orientation) does not easily allow to represent a room with few of them.

Chapter 5

Modifications on NIST MarkIII array

This Chapter focuses on the device used to acquire the HIWIRE and the CHILREVERB databases. The audio data has been used for all the experiments of Chapters 3 and 4. The device is a modified version of the NIST Microphone Array MarkIII [Rochet, 2004], a system able to acquire 64 synchronous audio signals at 44.1 kHz, primarily conceived for far-field automatic speech recognition, speaker localization and in general for hands-free voice message acquisition and enhancement. Preliminary experiments conducted on the original array [Brayda, 2005b] had showed that coherence among a generic pair of signals was affected by a bias due to common mode electrical noise. A hardware intervention was realized to remove each internal noise source from analog modules of the device. The modified array [Brayda, 2005a] provides a quality of input signals that fits results expected by theory. Without this necessary partial re-design, every attempt to reliably estimate inter-channel delays (used by CSP-D&S and N-best UL) and room impulse responses (necessary to perform TCL and MF) would have led to non-realistic, unpredictable results. The analysis and the designed hardware layout described in this chapter are present in the microphone arrays currently used in the Universities of Karlsruhe, Barcelona, Athens and at the ITC-irst (Italy) and IBM (USA) laboratories.

5.1 Summary of modifications

In order to pick-up speech from a distance of 5-6 meters as well as to apply effectively enhancement techniques based on filter-and-add beamforming, the NIST MarkIII array was considered as the best device available for research purposes. This array consists of eight microboards, each having eight microphone inputs and related amplification and A/D conversion stages. The whole digital stream is eventually made available to the user through a very effective interface to Ethernet. In general, using a microphone array and an accurate time delay estimation technique, as that based on Generalized Cross-Correlation (GCC) PHAse Transform (PHAT), a.k.a. Cross-power Spectrum Phase (CSP), described in A, one can solve the speaker localization problem and provide enhanced speech in a very effective way. However, system performance can highly depend on the quality of the input signals. One of the key points to derive excellent results from the above mentioned techniques is that input channels be independent each other. For instance, if a synchronous common-mode noise occurs in two microphones, a time delay estimation technique will reveal an artificial coherence at zero sample delay. The latter fact is equivalent to have an active noise source in front of the array, which actually does not exist. We start from a preliminary observation that a 50 Hz interference was evident in all the input channels of the MarkIII array. Once eliminated that source of noise in the easiest way, i.e. by replacing in-house alimentation with rechargeable batteries, a consistent synchronous interference was still present in the input signals. Although this interference had a rather small dynamics, the coherence between two signals was still biased at zero samples. To remove completely or to deviate the given electrical interference, the hardware of the device was changed, based on some substitutions of electrical components (e.g. polarized capacitors, tension regulators, etc.) as well as on modifications of the power supply ground stage in order to feed each microboard and each microphone circuit with an independent power supply. The modification process was conducted in several steps, each revealing an objectively quantifiable improvement with respect to the previous one. In the next sections, a detailed description of the array device will be given. In particular, the computation of the coherence between microphone signals will be described with a technical discussion and related figures. Secondly, the basic hardware changes will be discussed with suitable details for a possible intervention on the circuitry in order to fix a similar platform. More details about this activity can be found in [Bertotti, 2004].

5.2 The MARKIII Microphone Array

The NIST Microphone Array MarkIII [Rochet, 2004] is an array of microphones composed by 64 elements, specifically developed for voice recognition and audio processing. It records synchronous data at a sample rate of 44.1 kHz or 22.05 kHz with a precision of 24 bits.

The particularities of this array are the modularity, the digitalization stage and the data transmission via an Ethernet channel using the TCP/IP protocol.

The array uses 64 electret microphones installed in a modular environment. Two main components constitute the system: a set of microboards for recording the signals and a single motherboard to transmit the digital data over the network. There are eight microboards in the array, and every microboard is connected to eight microphones. The first step done by the microboard is the polarization of the microphones and the amplification of the signals. Electret microphones need a phantom power to work properly and provide a low voltage signal. So the microboard adapts the signals to be converted in the digital format. The digitalization of the audio signals is done on each microboard, using four dedicated stereo analog to digital converters. The choice of putting the A/D converters as close as possible to the microphones reduces the possibility of having the analog signal disturbed by electrical interferences.

The task of the motherboard is to collect all the digital signals from the single microboards, multiplex them and pack all the data in a format suitable for being sent over the network. The motherboard uses an Ethernet channel to transmit the digital signals: it gets an IP address via a DHCP service and sends broadcast data on the network. If a PC needs audio signals from the array, it has just to contact the array using a certain protocol and read the data from the network card. Due to the huge amount of material (64 ch $\times$ 44100 samples/sec $\times$ 3 bytes = 8.07 MB/sec), it has been chosen to use the UDP protocol. This allows to transfer a big quantity of data, but lacks of integrity checks. If the receiving computer is momentarily not fast enough to read all the packets, some packets are simply lost and the recorded signal will contain discontinuities. A software protocol to resend the lost packets has been implemented but is not encouraged for the high chances to lose data again.

The weak part of the chain is the storing of the data on the computer. In theory it could be possible to connect the MarkIII array to a switch and then listen to the data from a generic computer on the network. But since the transmission volume is very high, a computer with a single network interface card is not able to get all the data and loses packets. This is a crucial aspect since missing samples in the signal lead to worse performance of any of the above mentioned technologies. The solution is to install a dedicated network card on a PC and connect the array directly to that ma-

chine. This leads to the loss of flexibility guaranteed by the Ethernet protocol, but at least allows to record seamless data. However, there is the necessity to tune the operating system for receiving a lot of UDP packets. This tuning could not be done for Microsoft Windows machines, forcing the array to be used only with UNIX/LINUX operating systems. The machine connected to the array has to be in any case pretty fast, able to store data without losing incoming packets. This leads to the necessity to have a dedicated machine only for data recording, while real-time processing seems not feasible at the moment.

5.3 THE MARKIII/IRST based on batteries

In this section we describe the problems we encountered with the array, originally designed at NIST [Rochet, 2004], and how we solved them. It is worth noticing that, for project constraints, the underlying purpose of this initial improvement was to obtain a performant device in a short time, no matter how complex, costly or reproducible the solution would be. This improvement led us to obtain a first new prototype of MarkIII, from now on called "MARKIII/IRST". Each of the following subsections describes how the disturbances were eliminated one by one. In some cases the final solution was obtained after many subsequent trials, fully described in [Bertotti, 2004]. Of all the problems solved in [Bertotti, 2004], only a subset related to the quality of the speech signals acquired is presented here.

5.3.1 Early saturation effect of microphones

It was observed that when a speaker was near the array the microphone signals immediately saturated. One could guess that the Panasonic microphones were too sensitive or the OPAMPs were pushed to the limit. In any case, the device did not allow to control input levels. Moreover, it is worth noting that some microphones were more sensitive than others. The biggest ratio from the most sensitive (ch 35 and ch 8, respectively, in the array available at ITC-irst) was of 2:1, i.e. 6 dB in amplitude. Since no trimmer or other regulations of the input level were available, we eventually decided to physically bypass the first amplification stage as described in the following and shown in Figure 5.1. For comparison purposes, one can find in [Rochet, 2004] the original layout of the NIST device.

The two capacitors C1 and C6, placed at the very beginning and at the very end of the amplification stage, were 1 μF polarized capacitors in the original design. They were substituted with two polyester 0.47 μF capacitors, which generate much less noise. The first amplification stage was

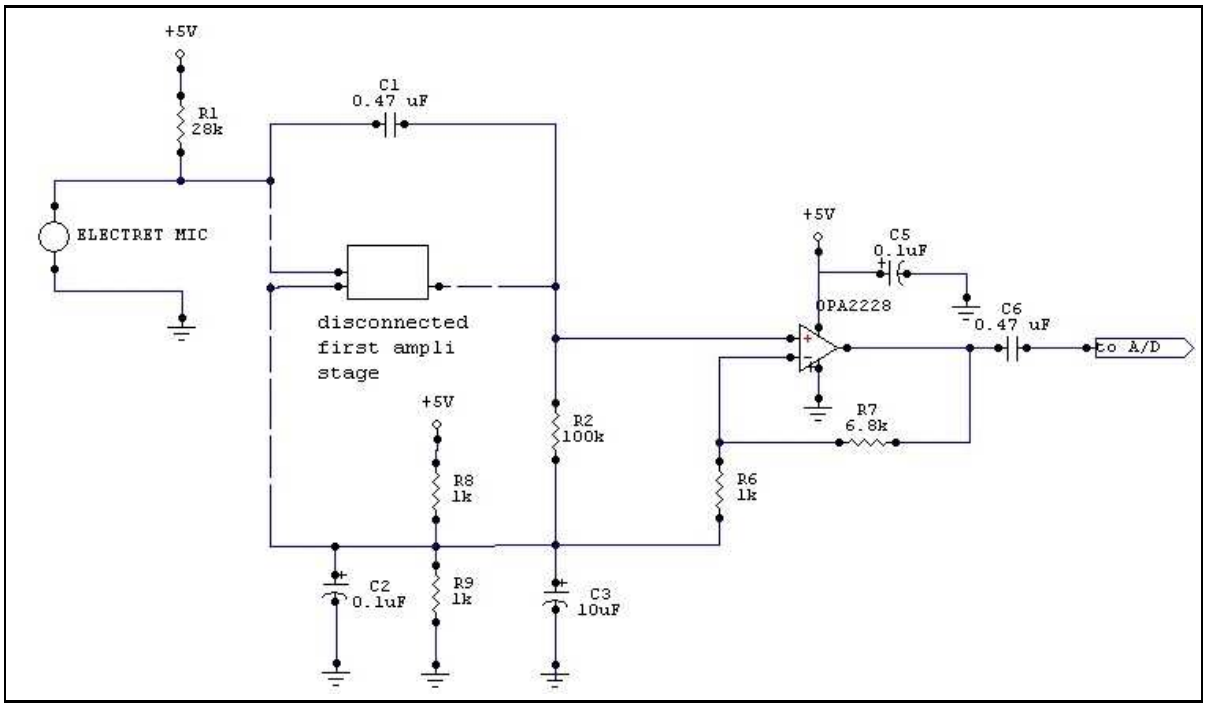


Figure 5.1. Modifications of the amplification stage in the first prototype (MarkIII/IRST).

then bypassed via a 0.47 μF capacitor, keeping the second stage polarization to the phantom GND with a 100 kΩ resistor. As a result, the original gain of 68 was reduced to 6.8, which is suitable to avoid any signal clipping.

5.3.2 50 Hz disturbance

In our preliminary recordings (done in the insulated room used to collect part of the HIWIRE database) we observed the presence of a perceivable 50 Hz interference. We realized the disturbance was due to the power supply: this problem was solved by substituting the 220V-AC to 9V-DC power adaptor, provided with the array, with a Pb rechargeable battery. This was not the final solution, as in a second step we solved the device noise problem (see Section 5.3.3) by switching to a battery power supply for the whole analog part. It is worth noting that, even with the best battery-based power supply available, still a light 50Hz disturbance persisted: it was much lower than the one coming from the AC current and it was totally due to environmental electromagnetic fields. By consequence it was definitely eliminated by surrounding the MarkIII with a Faraday cage.

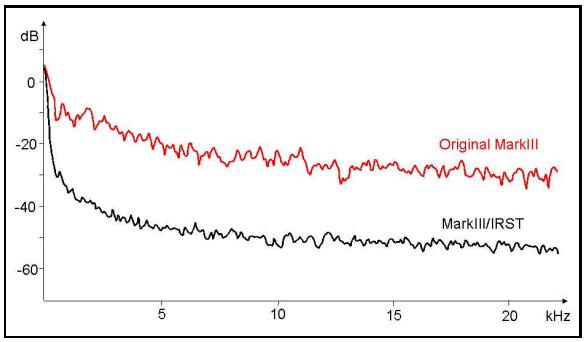


Figure 5.2. Spectra corresponding to a 600 ms of background noise. The red, upper line hints at the signal quality of the original MarkIII, while the black, lower one hints at the signal quality of the MarkIII/IRST. A reduction of 20 dB is evident at most of the frequencies.

5.3.3 Device noise

The device noise represents the major obstacle to the use of the MarkIII for speaker localization and beamforming purposes. It is also subtle to detect, as this problem is neither perceivable in normally reverberant rooms nor evident through waveform or spectral analysis of a single channel.

The device noise problem was evident once eliminated the 50 Hz interference (see Section 5.3.2). In other words, the following experiments regard the use of the MarkIII array powered by a rechargeable battery and installed in a very quiet insulated room. The room is characterized by less than 30 dBA background noise level (that is very close to the acoustics of an anechoic chamber) and a reverberation time lower than 100 ms. Recordings were done at 44.1 kHz. As discussed below, the electrical problem can be revealed both at single channel level (perceptually evident through listening tests) and at inter-channel correlation level (through inter-channel coherence measurements) analysis.

Single channel analysis

The device noise can be perceptually detected only in recordings taken in a very silent room, because in this condition it can be distinguished from real background noise. Alternatively it can be detected, without the need of an anechoic chamber, by manually detaching the microphones from the boards: the signals acquired from the array is then only pure noise coming from the devices. The effect of the device noise can be observed in Figure 5.2, where two average spectra of 600 ms of silence sequence are provided. The red, upper line is relative to a single channel of the original MarkIII array. The black, lower line is relative to a silence sequence of the same length recorded with the MarkIII/IRST. The environmental conditions were approximately the same, but clearly

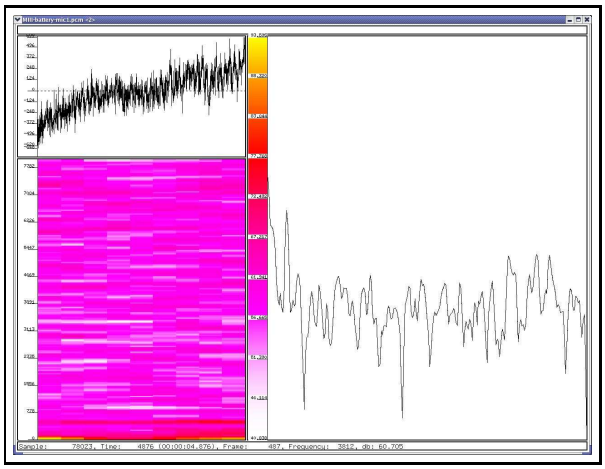


Figure 5.3. Analysis of a background noise sequence of 32ms length. The lower left part of the figure reports the spectrogram. The log power spectrum is given in the right part. The device noise is here more evident both in its dynamics and in its spectral characteristics. Note that the slope of the signal is due to a 2.5 Hz interference characterizing the given recordings.

the device noise affects the whole spectrum. According to the given figures, more than 20 dB noise reduction was obtained at almost all the frequencies. Another very detailed analysis was done by shortcutting each microphone input in order to measure only the board circuitry noise and, also in this case, a noise reduction of about 15-20 dB was observed. To better understand the entity of the noise, Figure 5.3 is related to some silence collected in the ITC-irst insulated room. From Figure 5.3, one can observe that the noise dynamics (between -300 and +300) involves about 9 bits. It was clear that losing 9 bits out of the first 16 most significant ones was a heavy limitation to the potential of this array.

Cross-channel analysis

An analysis of the CSP (described in Section A) between pairs of channels put into evidence other problems related to the so called “device noise”. This noise component, which can be observed in all the channels, is neither acoustic noise nor transduction noise of the microphones. It dominates over acoustic background noise of a relatively quiet environment. It exhibits a “common mode” within the 8 channels of each array microboard. Different modules (e.g. from channel 1 to channel 8, and from channel 9 to channel 16) have different and uncorrelated noise components. This is evident on the basis of a CSP analysis.

Figures 5.4 and 5.5 show the noise coherence between channels 1 and 8, which was derived from the analysis of a chirp-like signal reproduced through a hi-fi loudspeaker placed at the left side of the array: in this case, a strong coherence is evident between the (mainly electrical) noise

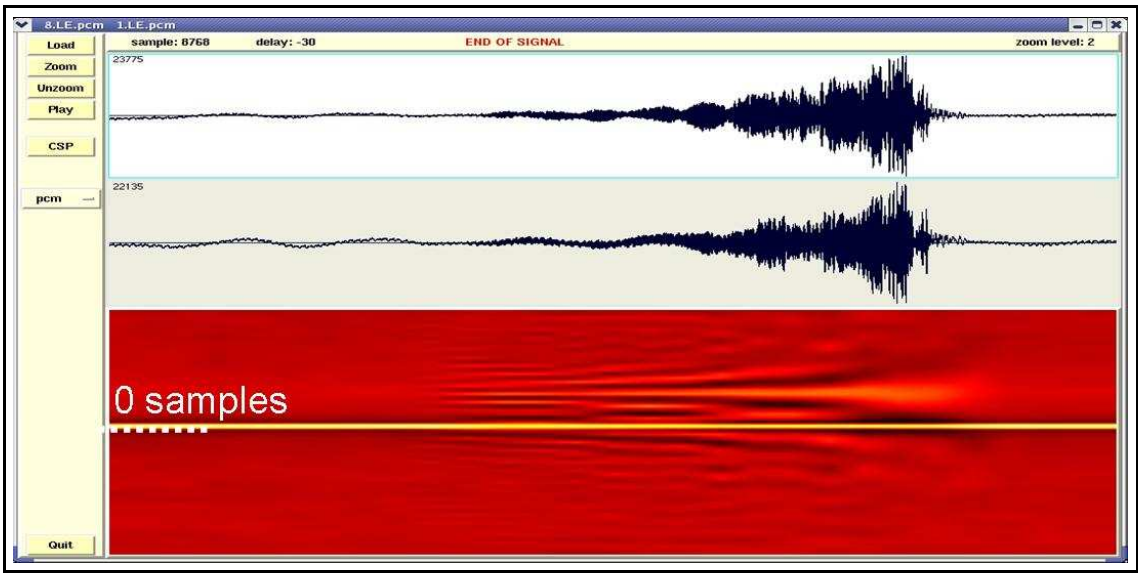


Figure 5.4. Chirp signals acquired in an insulated room before the intervention on the device. As the two channels belonged to the same microboard, there is a high peak of CSP function at 0 samples inter-microphone delay, which masks the true peak: this means a strong coherence between the device noise sequences.

sequences. A strong coherence at 0 samples is equivalent, for any localization algorithm, to determine a direction of arrival from an acoustic source right frontal to the array. In practice, the device noise takes all the energy of the CSP and concentrates it where no sources actually exist. Figure 5.5 specifically shows how the artificial peak, at 0 samples, dominates the secondary, true, peak located at +5 sample delay.

On the other hand, the same analysis repeated on channels 1 and 9, which are on two different microboards and therefore have no common mode noise, demonstrates the absence of any coherence at any particular delay.

Device noise removal

The single and cross-channel analysis clearly show the effect of the device noise. We describe in the following how we detected its origin and how we eliminated it. It is worth noting that a better solution was found with the next prototype, the MarkIII/IRST-Light. The device noise was caused from the tension regulator LM2940 (see technical documentation of Mark III in [Rochet, 2004]). There is one such a regulator for each of the 8 microboards. This tension regulator provides the operation voltage to 8 Panasonic microphones, to 4 A/D converters and to 8 OPAMPs. As mentioned in Section 5.3.3, the device noise has a common mode within the 8 channels of each array microboard.

In order to keep the original device layout, the problem was solved by physically removing such

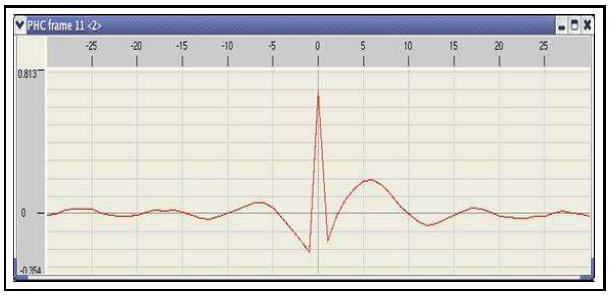


Figure 5.5. A slice of the CSP-gram in a fixed instant shows the artificial peak of the CSP, which masks the true one, located at a 5 samples delay.

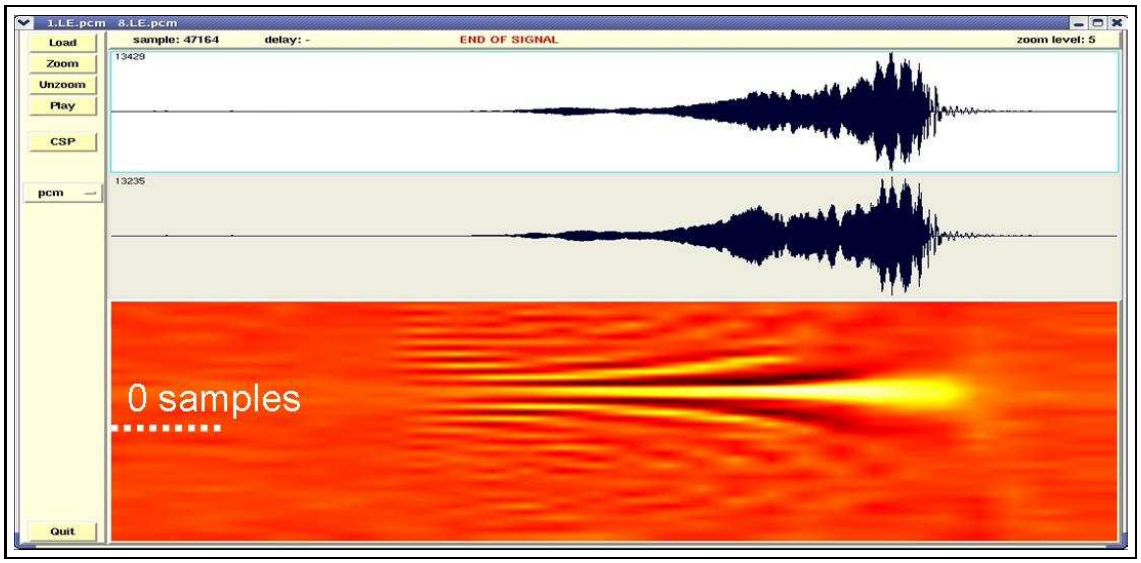


Figure 5.6. Signals extracted from Channel 1 and Channel 8 after our intervention. The peak of the CSP function reported in the lower part of the figure shows a strong coherence only when the chirp is played.

regulators and feeding the analogue part of every board directly with a circuit of battery designed ad hoc, while the digital part remained fed with a new transformer stabilized and filtered ad hoc. It is worth noting that part of the device-noise is caused by the LM2940 and part by the surface-mounted polarized capacitors, which should theoretically remove the regulator noise. These capacitors have an inner leakage current which creates the necessary oxide between the armors, thus generating a disturbance. Hence, they were substituted with polyester capacitors, which are bigger but generate much less noise. An effective solution was to feed the analogue part of each microboard with 4 $\times$ 1.2V, 5Ah batteries (a total of 32 batteries), so to guarantee the galvanic de-coupling of each power supply source. Nevertheless the best solution (see Section 5.4) turned out to be rising the power supply ground of each microphone with respect to the real ground. The use of external batteries

required an analysis of power consumptions prior to any decision about the components to buy. This analysis, together with a history of our several trials and the corrected layouts of the circuitry around the removed tension regulators, is detailed in [Bertotti, 2004].

After our intervention the device noise disappeared: Figures 5.6 and 5.7 have to be compared with Figures 5.4 and 5.5 respectively. The delay in samples at which the chirps arrive at the two microphones is clearly detected. In fact, by comparing the CSP coherence measures of Figures 5.4 and 5.6, it is evident that the constant yellow stripe at zero samples, caused by the device noise, has disappeared completely. With the new device, the coherence representation is now highlighting the true interchannel delay (i.e. +5 samples). For a single frame, this fact is evident in the main peak depicted in Figure 5.7.

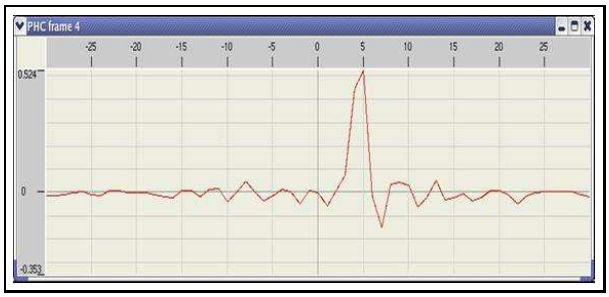


Figure 5.7. A slice from the CSP-gram in a fixed instant reveals now the true peak at a 5 samples delay. The device noise is totally absent.

5.3.4 8 kHz and 16 kHz common ground noise

A further problem was observed by analyzing the spectrogram of some utterances. This problem became evident once both the 50 Hz and the device noise problems were solved. Two disturbances at about 8 kHz and 16 kHz appeared in the spectrogram, as shown by Figure 5.8: two relatively strong stripes appear in red and violet in the spectrogram on the left part of the picture, which correspond to the two peaks evident in the right part.

Though the disturbance was present at frequencies not closely related to the speech signal, it was verified that it did not come from the environment and it was then worth to investigate, as it represented another common mode noise component across different channels preventing a clean data collection. We discovered it was due to the coupling between the digital and the analog ground. This coupling was made around the A/D converter PCM1802: the device was originally provided with two separate pins for the two grounds. In the original project of the MarkIII the two pins were connected via a short circuit. This makes the analog ground, which the audio signal relies upon,

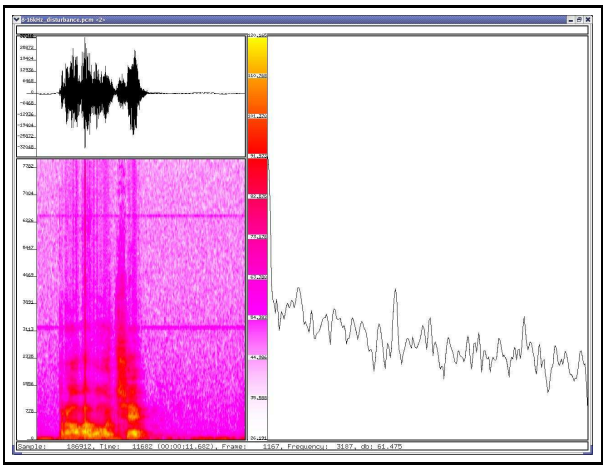


Figure 5.8. The 8 kHz and 16 kHz disturbance peaks are evident in the right part of the picture, where the spectrum of a silence segment is taken after the utterance depicted in the left part. Notice the absence of the device noise, removed as described in Section 5.3.3

coincident with the digital ground, which collects the noise coming from the various integrated devices, such as the A/D converter and the two tension regulators. The final solution consists in avoiding the common ground by feeding each microboard separately with an independent group of batteries, thus obtaining 8 groups of 4 x 1.2V, 5Ah batteries. Figure 5.9 shows the battery box entirely built at ITC-irst. More pictures and details are available in [Bertotti, 2004].

5.4 THE MARKIII/IRST-LIGHT

This section reports on further improvements of the MarkIII/IRST. For the purpose of making the modifications we did in the MarkIII/IRST easily reproducible by an expert in electronics in every laboratory of the CHIL project [CHIL, 2004], we were motivated to find another solution. The new prototype, from now on called “MARKIII/IRST-Light”, solves the same problems reported in section 5.3 in a very efficient, cheap and replicable way. It even performs better than the MARKIII/IRST in terms of SNR and coherence measures: see details in [Bertotti, 2005]. The multichannel corpus being collected at University of Karlsruhe [CHIL, 2004], is based on the use of this improved release of the device.

5.4.1 Manual gain correction

In order to better exploit the acquired signal dynamic range, in the new layout (Figure 5.10) we chose to keep both the amplifiers while reducing the total gain and making it tunable: the poten-

Figure 5.9. Inside of the power supply box: from the 8 groups of 4 batteries the power supply passes through the red and violet cables, placed on purpose in those positions. The transformer, which provides power supply for the digital part (in acquisition state) and recharges the batteries (in recharging state), appears in the center of the box.

tiometer R11 allows the total gain to be in the range 12÷16.7 (R11's nominal value gives a total gain of 15), which is both a compromise between good amplification and clipping avoidance, and a way to cope with the different sensitivity of the electret Panasonic microphones. Notice that R11 must be of high quality (possibly of plastic-film type).

5.4.2 High impedance microphone power supply

The main purpose of the MarkIII/IRST-Light is to reduce complexity and cost while keeping, and possibly improving, performance. It was realized that performance could even be improved with a different approach, taking into account that noises, circulating both on the analog and on the digital ground, could be deviated instead of suppressed. In order to feed microphones with a very clean supply, a high impedance path was designed for the DC coming from the batteries and each microphone power supply ground level was rised with respect to the real ground. A typical π RC cell scheme was built via R1, R2, R3 and C1. This is feasible because the electret microphones power consumption is very low. Notice that:

- C1 and CB are preferably of Tantalium type;
- C2 and C7 are preferably of Polyester type;
- there is one CB every 8 microphones, i.e. one per microboard.

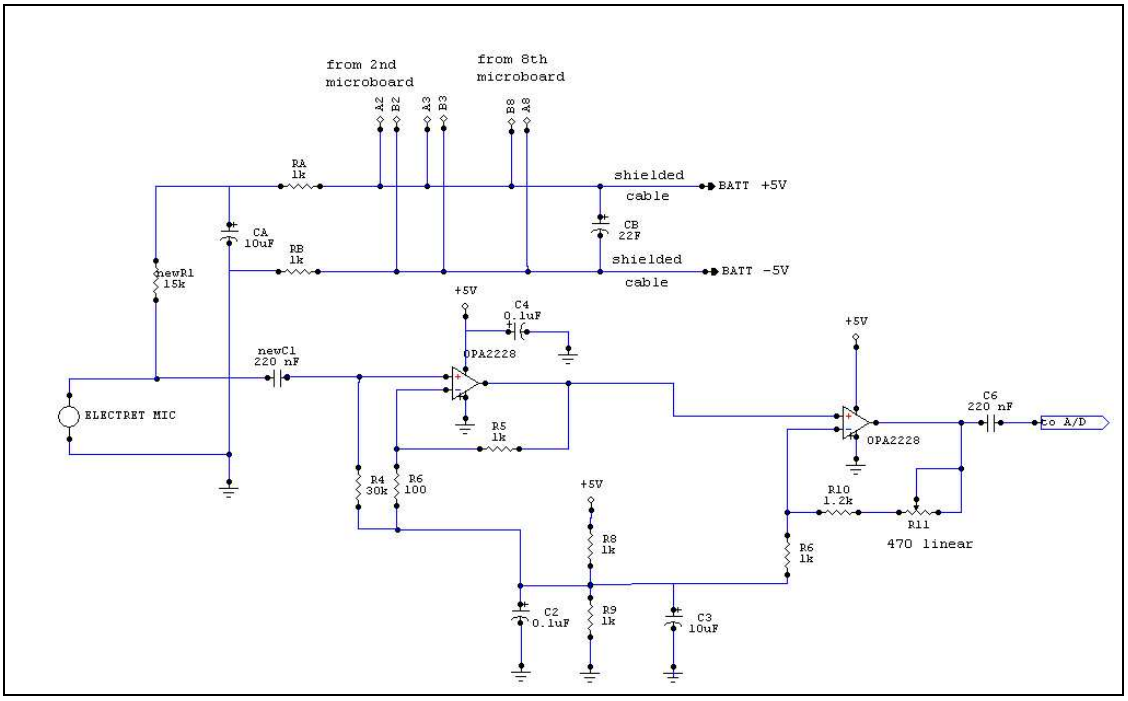


Figure 5.10. Modifications of the amplification stage in the MarkIII/IRST-Light. Notice the high impedance power supply stage, which connects each group of 8 microphones on the same microboard to a dual positive-negative power supply.

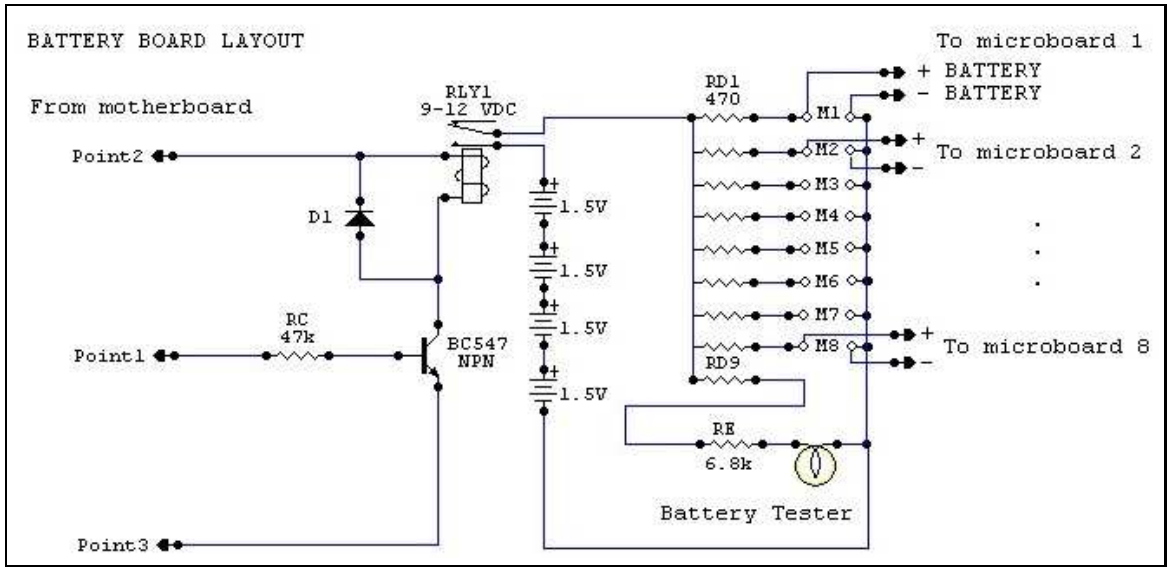


Figure 5.11. Battery saver microboard layout.

5.4.3 Battery saver microboard

We built and inserted into the Faraday cage a further microboard (Figure 5.12) to power the microphones only when the MarkIII is acquiring signals: this is simply done by letting this microboard be driven by the Capture Led ([Rochet, 2004], page 40). The purpose is to let batteries last as long as possible: we placed a series of 4 Alcaline batteries. The new microboard amplification stage layout is depicted in Figure 5.11. We estimated the MarkIII/IRST-Light can continuously acquire for 150 hours with this configuration, but one could freely make the series voltage be in the range [4,5 - 9V] or different combinations series-parallel to increase the duration. The battery saver microboard needs three signals from the motherboard: "Point 1" is the signal coming from the Capture Led, "Point 2" is the motherboard power supply for the relais, "Point3" is the motherboard GND. A small battery tester was added to check the batteries state.

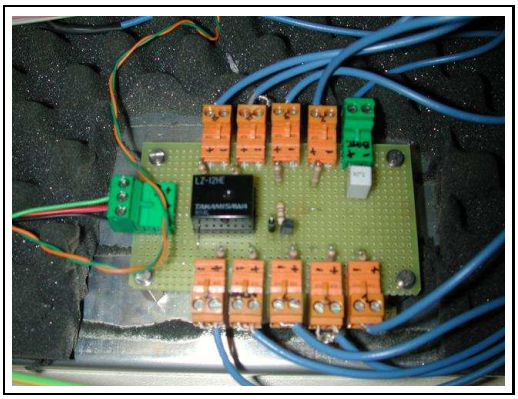


Figure 5.12. Battery saver microboard, inserted in the Faraday cage of the array.

5.5 Conclusions

This Chapter reported on a recent activity conducted at ITC-irst laboratories which allowed us to realize a new release of the NIST MarkIII microphone array.

The current prototype is able to provide clean signals that are suitable for speech enhancement as well as for automatic speaker localization purposes, thanks to improved characteristics in terms of coherence among different channel signals. The MarkIII/IRST Light prototype is presently used at the Universität Karlsruhe (TH), Germany, to record a large corpus of seminars and meetings for benchmarking of various speech and acoustic related technologies under study inside the Integrated European CHIL project [CHIL, 2004]. Similar versions of this array are used also at the

Univeversitat Politècnica de Catalunya, Barcelona, Spain, and at ITC-irst. Moreover, the new hardware layout is being used at NIST to produce a new generation of MarkIII arrays, on the basis of the above described interventions. The resulting analogue circuitry, together with a very effective digital section formerly designed and realized by NIST, makes the new device a very useful tool for future research and prototyping in the field of microphone arrays and distant-talking interaction.

Chapter 6

Conclusion

In this Chapter we summarize the contributions of this thesis, and outline for possible future improvements. In this work we improve the performance of a speech recognizer in scenarios where the environmental conditions generally represent a serious problem. The environment introduces a large mismatch between training conditions, in which the statistical representation of speech is derived by clean, close talk signal, and the test conditions, where additive noise and/or reverberation significantly change the signal characteristics. The dramatic drop in performances generally experienced when this phenomena occur, in practice almost all the time for real world applications, can be partially recovered thanks to the use of microphone arrays. Microphone arrays have been extensively used to enhance the quality of speech signals and are becoming a very useful tool when speech must be captured in a noisy or reverberant environment, where forcing each speaker to wear a close-talk microphone is undesirable or unpractical. However, because speech recognizers do not act as human listeners, the enhancement of the speech signal does not proportionally improve distant-talking speech recognition performances. To cope with this limit, we considered the Limabeam algorithm, which was proposed by Seltzer in 2003 with the objective of enhancing speech coming from a microphone array with the same criterion used by the speech recognizer, and not simply on the basis of a better audible quality. This is done by inserting a feedback loop between the recognizer and the speech enhancer, which is a filter-and-sum beamformer: the feedback is constituted by an hypothesized transcription on which a FIR filter set is consequently adapted. However, a conventional speech recognizer considers many hypotheses before producing a single text sentence: with the same philosophy, we generalize the Limabeam algorithm by increasing the number of feedbacks and constructing a set of parallel optimizations. The number of hypotheses

considered is a function of the first N-best outputs of the recognizer. Thus, N-best optimizations concur to the generation of as many FIR filter set, each one having a different objective function, each one strongly dependent on the hypotheses in input. After optimization, a second recognition step is performed on the optimized N-best group of features and we "elect" the best among the competing hypothesis under a ML criterion. With this technique we are able to overcome both a conventional Delay and Sum beamformer and the original Limabeam algorithm. Considering more hypotheses until the end of the optimization is the key aspect that greatly improves recognition performances. The presence of the correct transcription in the N-best list also reveals not to be a must, since ML filters can be derived from transcription which are acoustically confusable with the correct one. We emphasize that the proposed algorithm is able to recover errors made by the recognizer at the first recognition step. Furthermore, the proposed N-best approach was tested in environments where speech-oriented applications are desired: a very noisy enclosure and a very reverberant meeting room. In both cases the technique showed improvements. In this scenario we observe that recognition performance highly depends on two factors: the direction where the microphone array is pointing and the direction toward the speaker is speaking. We show that controlling the first factor is essential and that the best pointing directions, for speech recognition purposes, can be both the speaker position and the walls of the room, which act as "shadow" sources. We also show that, to the best of our knowledge, the highest recognition performances in such scenario can obtained when integrating *a priori* knowledge of the room impulse response, via Matched Filtering, with the proposed N-best Unsupervised Limabeam.

Future research

We believe that the exchange of information between the recognizer and the beamformer can be part of a more general framework, where Speech Enhancement and Speech Recognition are no more independent modules. This framework already has reached two improvement steps: in the first phase, Limabeam has shown that the beamformer can be "opened", in the sense that the information used to maximize a likelihood function is not only the single-channel beamformed output, but also the multi-channel signals, which all concur to the optimization process. In the second phase, this work shows that the recognizer itself can be "opened", in the sense that more likelihood functions can be optimized if not only the single transcription from the Viterbi algorithm is used, but also the first N-best hypotheses. The positive results obtained in both phases (which are additive, since the N-best UL works even with a single channel) hint that a more thorough fusion

between the beamformer and the decoder would lead to even higher performance improvement, possibly with less amount of computation: competing acoustically confusable transcription share after all much of their spectral content. Another aspect which deserves attention is the compensation domain. In this work the adaptive FIR filters modify the multi-channel signal and consequently its features, thus all the processing is done in the feature domain, while the statistical models used for recognition are untouched. Because recognition performance with a single channel can be greatly improved thanks to model compensation algorithms, such as PMC [Gales and Young, 1996], one can iteratively compensate features and models. Great attention should be given in this case to convergence issues and to avoidance of signal cancellation problems. Thus some constraints should be introduced when optimizing the FIR filter sets. Last but not least, through all this work the criterion used for optimization is the Maximum Likelihood, which is the same used in the most recent speech recognizers: we observed that recognition results are very sensitive to even small likelihood improvements. Thus, a deeper insight should be given to establish which are the parameters the whole beamforming-feature extraction-recognition chain (and feedback) is more sensitive to.

Appendix A

The Cross-Power Spectrum Phase Technique

A microphone array performs a spatial sampling of the acoustic wavefronts propagating inside an enclosure. It is often of interest the capability of comparing the signals captured by different microphones in order to calculate a degree of similarity between them as a function of their mutual delay. Given two microphones and their related signals s_i and s_j, it is possible to define a Coherence Measure (CM) function $C_{ij}(t,\tau)$ that expresses for each delay τ, the similarity between segments (centered at time instant t) extracted from the two signals. While the two microphones are receiving the wavefronts generated by an active acoustic source, this function is expected to have a prominent peak at the delay corresponding to the direction of wavefront arrival (e.g. positive if the source is on the left and negative if it is on the right). For each microphone pair a bi-dimensional representation of the CM function can be conceived. In this representation horizontal axis is referred to time, vertical axis is referred to delay and the coherence magnitude is represented by means of a "heat" palette (the higher the correlation at a particular time lag, the brighter the color or grey level). Figure A.1 represents this.

A particularly convenient CM function can be obtained starting from a Cross-power Spectrum Phase (CSP) analysis [Omologo and Svaizer, 1994, 1997], also known as PHAT transform, a particular case of Generalized Cross Correlation [Knapp and Carter, 1976]. The procedure for estimating a CSP-based Coherence Measure (CSP-CM) starts from the computation of the spectra $X_i(t,f)$ and $X_j(t,f)$ through Fourier transforms applied to windowed segments of signals x_i and x_j centered around time instant t. Then these spectra are used to estimate the normalized Cross-power Spec-

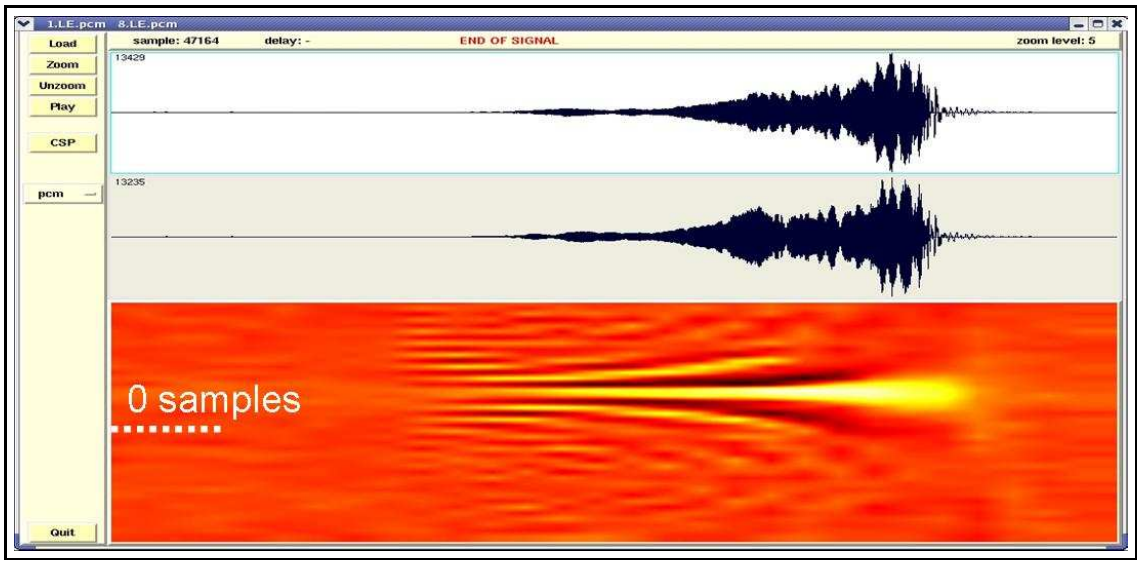


Figure A.1. Signals extracted from Channel 1 and Channel 8, when a chirp signal is played. The peak of the CSP function reported in the lower part of the figure shows a strong coherence when the chirp is played.

trum:

$$\Psi_{ij}(t,f) = \frac{X_i(t,f) \cdot X_j^*(t,f)}{\|X_i(t,f)\| \cdot \|X_j(t,f)\|} \tag{A.1}$$

that preserves only information about phase difference between x_i and x_j. Finally the inverse Fourier transform of $\Psi_{ij}(t,f)$ is computed:

$$C_{ij}(t,\tau) = \int_{-\infty}^{\infty} \Psi_{ij}(t,f) e^{j2\pi f\tau} df \tag{A.2}$$

The resulting function (considered as dependent of the lag τ) is the transform of an all-pass function and has a constant energy, mainly concentrated on the mutual delays at which there is high correlation between the two channels.

Indeed, if $\bar{\tau}$ is the true inter-channel delay between x_i and x_j, then (A.2) presents a delta pulse centered on the delay $\bar{\tau}$. The delay estimate is derived from:

$$\hat{\bar{\tau}}_{ij}(t) = \arg\max_{\tau} C_{ij}(t,\tau) \tag{A.3}$$

Thus, the information in the CSP peaks, where the inter-channel coherence is higher, can provide the following informations:

- It locates the delays, and indirectly the acoustic source position via trigonometry: the CSP can

drive a D&S beamformer (CSP-D&S) toward the maximum coherence directions. In this case, TDE and compensation is done in frequency domain to allow sub-sample precision. This is to limit the steering errors, to which adaptive beamforming is particularly sensitive [Widrow, 1982]

- It allows to analyze the multipath propagation inside a room, as delays associated to direct wavefront and to principal reflection are easily detectable.
- It can be used to analyze the mutual "independence" between the acquisition channels of an array. In the fact problems as cross-talk or common mode noise components generated within the acquisition device are clearly put into evidence by the appearing of graphical patterns (i.e. lines) in the CM that otherwise, in a quiet environment, should be rather uniform along the τ coordinate.

Limits to the effectiveness of the CM are given by the inter-microphone distance: the higher this distance, the lower the inter-channel coherence, and the lower the magnitude of the CSP peaks. A rule of thumb in reverberant environments (such as the CHIL room), which are the most difficult scenarios where to perform such measure, indicates that this distance should be never higher than 0.6m.

Bibliography

Acero, A. (1993). *Acoustical and environmental robustness in automatic spech recognition*. Boston, MA:Kluwer Academic Publishers.

Affes, S. and Grenier, Y. (1997). A signal subspace tracking algorithm for microphone array processing of speech. *Trans. Speech. Audio Proc.*

Allen, J., Berkley, D., and Blauert, J. (1977). Multimicrophone signal-processing technique to remove room reverberation from speech signals. *Journal of the Acoustic Society of America*, 62:902–915.

Allen, J. B. and Berkley, D. (1979). Image method for efficiently simulating small-room acoustics. *Journal of Acoust. Soc. Am.*

Atal, B. S. and Remde, J. R. (1982). A new model of lpc excitation for producing natural-sounding speech at low bit rates. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing*, volume 1.

Bertotti, C., Brayda, L., Cristoforetti, L., Omologo, M., and Svaizer, P. (2004). Url: http://www.eurecom.fr/∼brayda/markiii-irst.pdf.

Bertotti, C., Brayda, L., Cristoforetti, L., Omologo, M., and Svaizer, P. (2005). Url: http://www.eurecom.fr/∼brayda/markiii-irst-light.pdf.

Bitzer, J. and Simmer, U. (2001). *Microphone Arrays*, chapter Superdirective Microphone Arrays. New York: Springer-Verlag.

Bitzer, J., Simmer, U., and Kammeyer, K. (1999). Multi-microphone noise reduction techniques for hands-free speech recognition - a comparative study. In *Workshop*, Tampere, Finland.

Boll, S. F. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoustic, Speech and Signal Processing*, pages 208–211.

Brandstein, M. (1998). On the use of explicit speech modeling in microphone array applications. In *Proc of ICASSP*.

Brandstein, M. (1999). An event-based method for microphone array speech enhancement. In *Proc of ICASSP*.

Brandstein, M. and Griebel, S. (2000). *Theory and Applications of Acoustic ignal Processing for Telecommunications*, chapter 1: Nonlinear,Model-Based Microphone Array Speech Enhancement. S. L. Gay and J. Benesty.

Brandstein, M. and Ward, D. (2001). *Microphone arrays - signal processing techniques and applications*. New York: Springer-Verlag.

Brayda, L., Bertotti, C., Cristoforetti, L., Omologo, M., and Svaizer, P. (2005a). Modifications on NIST MarkIII array to improve coherence properties among input signals. In *AES, 118th Audio Engineering Society Convention, Barcelona, Spain*.

Brayda, L., Bertotti, C., Cristoforetti, L., Omologo, M., and Svaizer, P. (2005b). On calibration and coherence signal analysis of the CHIL microphone network at IRST. In *Joint Workshop on Hands-Free Speech Communication and Microphone Arrays, March 17-18, 2005, Piscataway, USA*.

Brayda, L., Rigazio, L., Boman, R., and Junqua, J.-C. (2004). Sensitivity analysis of noise robustness methods. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*.

Brayda, L., Wellekens, C., Matassoni, M., and Omologo, M. (2006a). Speech recognition in reverberant environments using remote microphones. In *ISM 2006, 8th IEEE International Symposium on Multimedia, December 11-13, 2006, San Diego, USA*.

Brayda, L., Wellekens, C., and Omologo, M. (2006b). Improving robustness of a likelihood-based beamformer in a real environment for automatic specch recognition. In *Proceedings of Specom*, St.Petersbourg, Russia.

Brayda, L., Wellekens, C., and Omologo, M. (2006c). N-best parallel maximum likelihood beamformers for robust speech recognition. In *Proceedings of EUSIPCO*, Florence, Italy.

Carter, G. (1993). *Coherence and time delay estimation*. IEEE Press.

CHIL (2004). Url: http://chil.server.de.

C.H.Lee (1998). On stochastic feature and model compensation approaches to robust speech recognition. *Speech Communication*, 25:29–47.

Chu, P. L. (1997). Superdirective microphone array for a set-top videoconferencing system. In *Proc. IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*.

CMU (2003). The cmu sphinx group open source speech recognition engines. URL: http://cmusphinx.sourceforge.net/html/cmusphinx.php.

Compernolle, D. V. (1990). Switching adaptive filters for enhancing noisy and reverberant speech from microphone array recordings. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing*.

Cook, C. E. and Bernfeld, M. (1993). *Radar signals - An introduction to theory and application*. Artech House, Norwood, MA.

Cox, H., Zeskind, R. M., and Owen, M. M. (1987). Robust adaptive beamforming. *IEEE Transactions on Acoustics Speech and Signal Processing*, 35:1365–1375.

Cramer, O. (1993). The variation of the specific heat ratio and the speed of sound in air with temperature, pressure, humidity, and co2 concentration. *Journal of the Acoustic Society of America*, 93(5):2510–2516.

Cray, B. A. and Nuttall, A. H. (2001). Directivity factors for linear arrays of velocity sensors. *The Journal of the Acoustical Society of America*, 110(1):324–331.

Doerbecker, M. (1997). Speech enhancement using small microphone arrays with optimized directivity. In *Proc. Int. Workshop on Acoustic Echo and Noise Control*, pages 100–103.

E. E. Jan, P. S. and Flanagan, J. L. (1995). Matched-filter processing of microphone array for spatial volume selectivity. In *Proc of ISCAS*.

Elko, G. (2000). *Superdirectional microphone arrays*, chapter 10, pages 181–237. Kluwer Academic Publishers, S. L. Gay and J. Benesty eds.

Ephraim, Y. and Malah, D. (1984). Speech enhancement using aminimum mean-square error short-time spectral amplitude estimator. *IEEE Trans. Acoustic, Speech and Signal Processing*, ASSP-32(6):1109–1121.

F. J. MacWilliams, N. J. S. (1980). Pseudo-random sequances and arrays. *Proc. of IEEE*, pages 593–619.

Ferras, M. (2005). Multi-microphone signal processing for automatic speech recogntiion in meeting rooms. Master's thesis, ICSI, Berkeley, CA, USA.

FFTW (1999). Url: http://www.fftw.org.

Flanagan, J., Berkley, D., Elko, G., West, J., and Sondhi, M. (1991). Autodirective microphone systems. *Acustica*, 75:58–71.

Flanagan, J., Johnston, J., Zahn, R., and Elko, G. (1985). Computer-steered microphone arrays for sound transduction in large rooms. *Journal of the Acoustical Society of America*, 78:1508–1518.

Flanagan, J., Surendran, A., and Jan, E. E. (1993). Spatially selective sound capture for speech and audio processing. *Trans. on Speech Communication*.

Frost, O. (1972). An algorithm for linearly constrained adaptive array processing. In *Proceedings of the IEEE*, volume 60, pages 926–935.

Gales, M. J. F. (1995). *Model-based techniques for for noise robust speech recognition*. PhD thesis, Cambridge University, Cambridge, England.

Gales, M. J. F. and Young, S. J. (1996). Robust speech recognition using parallel model combination. *IEEE Trans. on Speech and Audio Processing*, 4(5):352–359.

Gilbert, E. and Morgan, S. (1955). Optimum design of directive antenna arrays subject to random variations. *Bell System Technical Journal*, pages 637–663.

Gillespie, B. W. and Atlas, L. E. (2002). Acoustic diversity for improved speech recognition on reverberant environments. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*.

Gillespie, B. W. and Atlas, L. E. (2003). Strategies for improving audible quality and speech recognition accuracy of reverberant speech. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*.

Giuliani, D., Matassoni, M., Omologo, M., and Svaizer, P. (1997). Use of different microphone array configurations for hands-free speech recognition in noisy and reverberant environment. In *Proc. Eurospeech*.

Grenier, Y. (1992). A microphone array for car environments. In *Proc. of ICASSP*.

Griffith, L. and Jim, C. (1982). An alternative approach to linearly constrained adaptive beamforming. In *IEEE Trans. on Antennas and Propagation*, volume AP-30, pages 27–34.

H. F. Silverman, W. R. Patterson III, J. L. F. (1998). The huge microphone array. *IEEE Concurrency: Parallel, Distributed and Mobile Computing*, 06(4):36–46.

Haaso, H. (1972). The influence of a single echo on the audibility of speech. *Journal of the Audio Engineering Society*, 20.

Hardwick, J., Yoo, C., and Lim, J. (1993). Speech enhancement using the dual excitation speech model. In *Proc of ICASSP*.

Hayes, M. H., Lim, J. S., and Oppenheim, A. V. (1990). Signal reconstruction from phase or magnitude. *Trans. on ASSP*.

Haykin, S. (2002). *Adaptive Filter Theory*. Prentice Hall.

Hermansky, H. and Morgan, N. (1994). Rasta processing of speech. *IEEE Trans. Acoustic, Speech and Signal Processing*, 2(4):578–589.

Hirsch, H. G. and Pearce, D. (2000). The aurora experimental framework for the performance evaluations of speech recognition systems under noisy condidions. In *ISCA ITRW ASR2000 Workshop on Automatic Speech Recognition: Challenges for the Next Millennium*, Paris, France.

Hoshuyama, O., Sugiyama, A., and Hirano, A. (1999). A robust adaptive beamofmer with a blocking matrix using coefficient constrained adaptive filters. *Trans. IEICE*, E82-A(4):640–647.

Huang, X., Acero, A., and Hon, H. (2001). *Spoken Language Processing*. Carnegie Mellon University.

J-C. Junqua, J. P. H. (1996). *Robustness in automatic speech recognition*. Kluwer.

Jan, E. E. and Flanagan, J. (1995). Microphone arrays for speech processing. In *Proc. of URSI*.

Janin, A., Ang, J., Bhagat, S., R.Dhillon, J.Edwards, Macias-Guarasa, J., Morgan, N., Peskin, B., Shriberg, E., Stolcke, A., Wooters, C., , and Wrede, B. (2004). The icsi meeting corpus: Resources and research. Meeting Recognition Workshop (Montreal, Canada).

Kaneda, Y. and Ohga, J. (1986). Adaptive microphone-array system for noise reduction. *IEEE Transactions On Acoustics, Speech, And Signal Processing*, 34(6):1391–1400.

Knapp, C. H. and Carter, G. C. (1976). The generalized correlation method for estimation of time delay. In *IEEE Transactions on Acoustics, Speech and Signal Processing*, volume 24(4), pages 320–327.

Lai, C. Y.-K. and Aarabi, P. (2004). Multiple-microphone time-varying filters for robust speech recogntiion. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*.

Leonard, R. G. (1984). A database for speaker-indipendent digit recogntiion. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*, pages 111–114.

LiDeng, Droppo, J., and Acero, A. (2004). Enhancement of log mel power spectra of speech using a phase-sensitive model of the acustic environment and sequential estimation of the corrupting noise. *IWEEE Transactions on SPeech and Audio Processing*, 12(1).

Lim, J. S. and Oppenheim, A. V. (1979). Enhancement and bandwidth compression of noisy speech. *Proc. of the IEEE*, 67(12):1586–1604.

Lin, Q., Jan, E. E., and Flanagan, J. L. (1994). Microphone arrays and speaker identification. *Trans on Speech and Audio Proc*.

MacKay, D. J. C. (2004). Macopt optimizer. URL: http://www.inference.phy.cam.ac.uk/mackay/c/macopt.html.

Marro, C., Mahieux, Y., and Simmer, K. U. (1998). Analysis of noise reduction and dereverberation techniques based on microphone arrays with postfiltering. *IEEE Trans. Speech and Audio processing*, 6(3):240–259.

Matassoni, M., Omologo, M., Giuliani, D., and Svaizer, P. (2002). Hmm training with contaminated speech material for distant-talking speech recognition. In *Computer Speech and Language*, volume 16(2), pages 205–223.

Mayer, R., Buneman, Hartley, Gauss, and Euler (2001). Fast fft routine. URL: http://home.iae.nl/users/mhx/fft_c.frt.

McCowan, I., A.Morris, and H.Bourlard (2002). Improving speech recogniton performance of small microphone arrays using missing data techniques. In *Proc. ICSLP*.

McCowan, I. and Sridharan, S. (2001). Microphone array sub-band speech recognition. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*.

McCowan, I. A. (2001). *Robust Speech Recognition using Microphone Arrays*. PhD thesis, Queensland University of Technology, Australia.

Miyoshi, M. and Kaneda, Y. (1988). Inverse filtering of room acoustics. *IEEE Transactions on Speech and Audio Processing*.

Monzingo, R. A. and Miller, T. W. (1980). *Introduction to Adaptive Arrays*. John Wiley and Sons.

Nakatani, T., Kinoshita, K., and Miyoshi, M. (2006). Harmonicity-based blind dereverberation for single-channel speech signals. *accepted to IEEE Trans. on Audio, Speech and Language Processing*.

Neely, S. T. and Allen, J. B. (1979). Invertibility of a room impulse response. *Journal of the Acoustical Society of America*, 66:165–169.

NIST (2004). The nist meeting room project. http://www.nist.gov/speech/test_beds/mr_proj.

Nordholm, S., Claesson, I., and Bengtsson, B. (1993). Adaptive array noise suppression of handsfree speaker input in cars. *IEEE Transactions on Vehicular Technology*, 42(4):514–518.

Oh, S., Viswanathan, V., and Papamichalis, P. (1992). Hands-free voice communication in an automobile with a microphone array. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*.

Omologo, M., Brutti, A., Svaizer, P., and Cristoforetti, L. (2006). Speaker localization in chil lectures: Evaluation criteria and results. In edited by Steve Renals and Bengio, S., editors, *MLMI 2005: Revised selected papers*, pages pp. 476–487. Springer Berlin/Heidelberg.

Omologo, M., m. Matassoni, and Svaizer, P. (2001). *Microphone Arrays*, chapter Speech Recognition with Microphone Arrays. Springer.

Omologo, M. and Svaizer, P. (1994). Acoustic event localization using a cross-power spectrum phase based technique. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*.

Omologo, M. and Svaizer, P. (1997). Use of the cross-power-spectrum phase in acoustic event location. *IEEE Trans. on Speech and Audio Processing*, 5, n. 3:288–292.

Omologo, M., Svaizer, P., and DeMori, R. (1998). *Spoken Dialogues with Computers*, chapter Acoustic Transduction. Academic Press, London, UK.

Oppenheim, A. V. and Shafer, R. W. (1999). *Discrete Time Signal Processing*, chapter 5. Prentice Hall,Englewood Cliffs, NJ.

Petropulu, A. and Nikias, L. (1993). Blind deconvolution using signal reconstruction from partial higher order cepstra information. In *Trans. On Sig. Proc.*

Petropulu, A. and Subramaniam, S. (1994). Cepstrum based deconvolution for speech dereverberation. In *Proc. ICASSP*.

Petropulu, A. P. and Nikias, L. (1991). Blind deconvolution based on signal reconstruction from partial information using higher-order spectra. In *Proc. Of ICASSP*.

Petropulu, A. P. and Nikias, L. (1992). Signal reconstruction from the phase of the bispectrum. In *Trans. On Sig. Proc.*

Petropulu, A. P. and Subramaniam, S. (1996). Cepstrum based deconvolution for speech dereverberation. In *Trans.Speech Audio Proc.*, volume 4(5), pages 392–396.

Press, W., Flannery, B. P., Teukolsky, S., and Vetterling, W. (1988). *Numerical Recipes in C: the Art of Scientific Computing*. Cambridge Univ. Press.

Raab, D., McDonough, J., and Wolfel, M. (2004). A cepstral domain maximum likelihood beamformer for speech recognition. In *Proceedings of Interspeech"*, Jeju Island, Korea.

Rabinkin, D., Renomeron, R., and Flanagan, J. (1998). Optimal truncation time for matched filter array processing. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*.

Rochet, C. (2004). Url: http://www.nist.gov/smartspace/toolchest/cmaiii/userg/microphone_array_mark_iii.pdf.

Seltzer, M. (2003). *Microphone array processing for robust speech recognition*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA.

Seltzer, M. and Raj, B. (2003). Speech recognizer-based filter optimization for microphone array processing. In *IEEE Signal Processing Letters*, volume 10(3), pages 69–71.

Seltzer, M., Raj, B., and Stern, R. M. (2004). Likelihood-maximizing beamforming for robust hands-free speech recognition. In *IEEE Trans. on Speech and Audio Procesing*, volume 12(5), pages 489–498.

Seltzer, M. and Stern, R. M. (Nov. 2006). Subband likelihood-maximizing beamforming for speech recognition in reverberant environments. *IEEE Transactions on Audio, Speech and Language Processing*, 14(6):2109–2021.

Seltzer, M. L. (2004). personal correspondence.

Seltzer, M. L. and Raj, B. (2001). Calibration of microphoe arrays for improved speech recognition", proc. of eurospeech 2001. In *Proc of Eurospeech*.

Seltzer, M. L., Raj, B., and Stern, R. M. (2002). Speech recognizer-based microphone array processing for robust hands free speech recognition. In *Proc of ICASSP*.

Seltzer, M. L. and Stern, R. M. (2003). Subband parameter optimization of microphone arrays for speech recognition in reverberant environments. In *Proc of ICASSP*.

Shewchuk, J. R. (1994). *An Introdcution to the Conjugate Gradient Method Withouth the Agonizing Pain*. CMU, http://www.cs.cmu.edu/ quake-papers/painless-conjugate-gradient.pdf.

Shimizu, Y., Kajita, S., Takeda, K., and Itakura, F. (2000). Speech recognition based on space-diversity using distributed multi-microphone. In *Proc of ICASSP*.

Simmer, K. U., Bitzer, J., and Marro, C. (2001). *Microphone Arrays*, chapter Post-Filtering Techniques. Springer.

Steinberg, B. D. (1976). *Principles of Aperture and Array System Design*. John Wiley and Sons.

Sullivan, M. and Stern, R. M. (1993). Multi-microphone correlation-based processing for robust speech recognition. In *Proc. of ASA*.

T. Yamada, S. Nakamura, K. S. (2002). Distant-talking speech recognition based on a 3-d viterbi search using a microphone array. *IEEE Trans. on Speech and Audio Processing*, 10(2):48–56.

Veen, A. V. D., Talwar, S., and Paulraj, A. (1997). A subspace approach to blind space-time signal processing for wireless communication systems. *IEEE Trans. on Signal Processing*, 45(1):173–190.

Venn, B. V. and Buckley, K. (1988). Beamforming: a versatile approach to spatial filtering. In *IEEE ASSP Magazine*.

Walach, E. (1984). On superresolution effects in maximum likelihood adaptive antenna arrays. *IEEE Transactions on Antennas and Propagation (ISSN 0018-926X)*, AP-32:259–263.

Widrow, B., Duvall, K., and Newman, R. G. W. (1982). Signal cancellation phenomena in adaptive antennas: Causes and cures. *IEEE Trans. on Antennas and Propagation*, 30(3):469– 478.

Widrow, B. and Stearns, S. D. (1985). *Adaptive Signal Processing*. Englewood Cliffs.

Yapanel, U., Zhang, X., and Hansen, J. (2002). High performance digit recognition in real car environments. In *Proc of ICSLP*.

Yegnanarayana, B. and Murthy, P. (2000). Enhancement of reverberant speech using lp residual signal. *IEEE Transactions on Speech and Audio Processing*, 8(3):267–281.

Young, S. (2003). Htk speech recognition toolkit. http://htk.eng.cam.ac.uk.

Ziomek, L. J. (1995). *Fundamentals of Acoustic Field Theory and Space-Time Signal Processing*. CRC Press.