Université du Québec
Institut National de la Recherche Scientifique
(Énergie, Matériaux et Télécommunications)

A Soft Computing Approach for On-Line Automatic Speech Recognition in Highly Non-Stationary Acoustic Environments

by
Md Foezur Rahman Chowdhury

Dissertation submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy (Ph.D.) in Telecommunications

April 27, 2012

Evaluation Jury

President of Jury and Internal Examiner	Prof. Tiago Falk (INRS-ÉMT, Montréal, QC, Canada)
External Examiner	Dr. Michael Picheny (IBM Thomas J Watson Research Center Yorktown Heights, NY USA)
External Examiner	Prof. Stephen Zahorian (ECE Dept., SUNY-Binghamton University, NY USA)
Research Director	Prof. Douglas O'Shaughnessy (INRS-ÉMT, Montréal, QC, Canada)
Research Co-Director	Prof. Sid-Ahmed Selouani (Université de Moncton, Shippagon, NB, Canada)

INRS
Université d'avant-garde

To

My beloved wife Farhana.

My lovely son Farhan, and daughter Fatiha.

My dear dad Abdur Rob Chowdhury.

In loving memory of my dear mom Shahida Begum.

Résumé

Ce travail de recherche aborde les problèmes de la conception d'un système de reconnaissance automatique de la parole ASR (Automatic Speech Recognition) en ligne (on-line) robuste au bruit, à savoir, la reconnaissance de la parole en ligne auto-adaptable à environnement-détectable similaire au processus humain et son exécution dans les environnements acoustiques réels hautement non-stationnaires. Commençant par une étude de l'état d'art des technologies ASR en différé (off-line), on présente, en premier, les approches courantes utilisées dans la littérature de l'ASR afin de formuler un système de reconnaissance continue en ligne de la parole basé sur la technique HMM. Dans cette approche, on examine la technique biaisée dynamique de suppression de trame (frame dynamic bias removal technique) pour l'ASR en ligne, qui a une très bonne performance d'ASR pour de la parole propre (non corrompue par du bruit). Nous introduisons alors une nouvelle technique pour un ASR en ligne typique basé sur la technique en ligne bayésienne d'inférence. Dans ce cas-ci, nous étudions la performance de la technique de la moyenne récursive commandée par des minimum MCRA (minima controlled recursive averaging) pour le détection et la compensation de bruit de canal simple en réalisant les essais en ligne d'ASR pour le signal de parole dans des environnements acoustiques hautement non-stationnaires et comparer alors leurs résultats avec la parole bruitée au discours bruyant correspondant pour l'ASR

en différé. Finalement, nous présentons une architecture d'ASR en ligne basée sur une technique non-linéaire et un modèle non-gaussienne pour modéliser des scénarios acoustiques réels. Dans cette approche nous proposons la technique de l'optimisation d'essaim de particules PSO (particle swarm optimization) pour dépister et estimer le bruit, et nous avons montré par des expériences que la technique d'optimisation PSO améliore la performance du système en ligne de reconnaissance de la parole de manière significative dans les environnements acoustiques hautement non-stationnaires.

Introduction & principales contributions

La reconnaissance de la parole est un processus de conversion des expressions parlées de la parole en mots ou textes. Ces textes peuvent être la sortie finale ou l'entrée au traitement de langage naturel. En raison de caractéristiques naturelles et efficaces du signal de parole dans l'échange d'information, il est devenu la manière la plus rapide que l'être humain peut utiliser pour communiquer avec des machines. Avec l'arrivée des technologies de calcul modernes, les interfaces entre les hommes et les machines deviennent plus réalistes pour l'accès et la gestion de l'information lorsque (i) l'espace de l'information est large et complexe, (ii) les utilisateurs sont techniquement naïfs, et (iii) seule les téléphones sont disponibles. Les interfaces de communication, basées sur la parole, le plus généralement utilisées entre les hommes et les machines sont: (a) l'identification de la parole simple comme, la commande et le contrôle, saisie de données par téléphone, dictée, transcriptions: légal, médical, TV, et (b) conversations interactives et machines intelligentes tel que, les kiosques d'informations, traitement transactionnel, et agents intelligents, parcourir la musique, navigation sur le web,

contrôle de voitures et la navigation GPS, etc. La reconnaissance automatique de la parole (RAP) est un champ de recherche très intéressant pour la conception de l'interface homme-machine. L'ASR est un champ de grande fascination et également fascinant à aborder. C'est un champ de recherche de grandes frustrations également lorsque les résultats sont moins fascinants. On pense que les aspects suivants sont les contributions significatives de ce travail de recherche à l'avancement des connaissances dans le domaine de l'ASR en ligne robuste au bruit.

- **Contribution 1:** Nous avons conçu et implémenté un système ASR en ligne robuste au bruit, avec l'extraction de caractéristiques, détection en ligne et détection de brusque changements dans les environnements acoustiques, addition conjoint dynamique des trames et compensation des distorsions du canal JAC (channel distortions compensation), et les fonctionnalités de reconnaissance de la parole dans un environnement de calcul multi-fileté multithread.
- **Contribution 2:** Nous avons proposé la technique d'inférence en ligne bayésienne pour les détections de point de changement (BOCPD) et l'adaptation rapide pour la détection du bruit basée par MCRA et la technique d'estimation dans des environnements hautement non-stationnaire et à changements abruptes. Nous avons développé un modèle BOCPD basé-par-trames pour l'adaptation rapide du MCRA aux conditions acoustiques réelles non-stationnaires. Nous avons montré par des expériences que le BOCPD peut réduire le retard de la fenêtre de mise à jour dans le MCRA en quantité significative dans les plus mauvais scénarios lorsque le SNR change rapidement des conditions les plus elevées vers les plus basses.
- **Contribution 3:** Nous avons implémenté le modèle BOCPD par-trames RAP

proposé pour l'adaptation rapide de la technique de MCRA pour la technique ASR en ligne utilisant la base de données de parole Aurora 2 [1]. Une compensation de distorsion de canal dynamique de trame est implémentée dans un processus de reconnaissance à deux étages utilisant un outil de simulation de calcul en temps réel ATK [2]. Les résultats expérimentaux montrent l'amélioration significative dans l'exactitude de travail comparée à l'ASR basique dans le mode Batch.

- **Contribution 4:** Nous avons implémenté un filtre de bande critiques perceptuel et une technique de seuil masquant pour réduire le bruit musical dans la détection du bruits basé par MCRA et la technique de compensation dans notre approche proposée pour l'ASR en ligne basé par JAC. Les résultats expérimentaux montrent une réduction significative du bruit musical en termes de PESQ et SNR segmental (SegSNR) comparés à la technique de rehaussement de la parole standard.
- **Contribution 5:** Nous avons proposé un filtre (PF) de particules basé par Monte Carlo séquentiel pour l'exploitation de la parole en ligne dans les environnements acoustiques hautement non-stationnaires et à changement abrupte. Nous avons développé cette approche basée sur le modèle non-stationnaire et non Gaussien pour le signal de parole dans les conditions acoustiques réelles. Nous avons montré par les expériences que le filtre PF peut aider à concevoir et à améliorer la performance de reconnaissance du système d'ASR en ligne.
- **Contribution 6:** Nous avons également proposé le filtre d'optimisation d'essaim de particules (PSO) pour la détection du bruit non-stationnaire de canal simple dynamique des trames et l'approche de compensation pour l'ASR en ligne

proposé. PSO est un genre émergeant de technique de filtrage basé sur les phénomènes des oiseaux volants (bird flocking) et le mouvement des poissons (fish schooling), qui est tout à fait intensivement utilisé comme une forme alternative et efficace d'algorithmes Génétiques. On le prouve par nos expériences que le filtrage basé sur le PSO est une technique appropriée pour concevoir un ASR en ligne utilisable dans les conditions acoustiques hautement non-stationnaires.

- **Contribution 7:** Nous avons testé une approche commune de PF et de PSO pour l'auto-adaptabilité du système d'ASR en ligne pour changer des bruits acoustiques non-stationnaires fortement changeant. PSO est utilisé pour assurer la localisation des particules dans la partie la plus probable du secteur de la densité de probabilité prédictive à posteriori. Les résultats expérimentaux d'ASR en ligne proposé pour la parole test bruitée montrent une amélioration significative dans le réduction du taux d'erreur de mot comparé à celui basé sur le filtre PF simple et le filtre PSO respectivement.

Contexte

L'état de l'art ASR courant a trouvé ses applications commerciales réussies pour l'utilisation des interfaces homme-machine de tous les jours. L'ASR a atteint sa position actuelle en raison des efforts de recherche continue de beaucoup de scientifiques, ingénieurs, et linguistes de la parole pendant les trois dernières décennies dans le développement des technologies très innovateur pour la reconnaissance de la parole basée sur des techniques du modèle de Markov caché (HMM) statistique. Fondamentalement, c'est dans la dernière décennie où les technologies de reconnaissance de

la parole ont émergé des plateformes dépendantes de locuteur aux plateformes indépendantes de locuteur et du petit vocabulaire au grand vocabulaire continue et la forme spontanée de reconnaissance de la parole. Malgré ces grands développements, la performance de l'ASR est encore loin de la compétence humaine de perception de la parole.

L'état actuel d'ASR fonctionne particulièrement bien sous les environnements acoustiques contrôlés connus. Le problème fondamental d'ASR est que sa performance se dégrade rapidement lorsque les environnements d'apprentissage et de test ne correspondent pas, c'est-à-dire, sa performance est insuffisante dans les environnements de test inconnus. Ces disparités sont dues aux variabilités intra-locuteurs et inter-locuteurs, aux environnements acoustiques de fond, aux microphones, et aux variabilités du canal. Beaucoup de techniques très innovatrices ont été développées pour réduire au minimum ces variabilités, et la plupart d'entre elles sont performante dans un environnement acoustique cohérent dépendant d'un contexte spécifique. Ces techniques sont basées sur des suppositions au sujet du bruit ou les différences dans la collecte et l'apprentissage sur une condition de bruit spécifique. D'ailleurs, ces techniques ont besoin que l'ASR fonctionne dans le mode batch, c'est-à-dire, l'ASR décode les expressions entières de la parole dans un groupe d'ocourences.

Dans les scenarios du monde-réel, les environnements acoustiques sont très complexes. Afin de rendre l'ASR robuste au bruit, il est exigé de surveiller et de détecter les environnements environnants et d'explorer la nature des bruits plutôt que de faire de simples suppositions d'un type spécifique de bruit. Malheureusement, l'ASR actuelle manque des techniques très innovatrices qui depistent les environnements acoustiques de fond et analysent la nature des bruits avant le décodage de la parole test.

Les mécanismes humains de perception de la parole à l'intérieur du cerveau sont encore beaucoup moins compris et restent comme une boîte noire pour les chercheurs du domaine de la parole. Cependant, les chercheurs ont observé cela pendant la conversation humain-à-humaine, les gens surveillent le locuteur aussi bien que l'environnement acoustique environnant sans interruption dans les conditions défavorables et ils ont la capacité de s'adapter rapidement aux environnements acoustiques changeants. Les sources les plus communes des variabilités dans les environnements acoustiques sont: (i) variabilité inter-locuteur - due à la longueur du conduit vocal et aux variations des caractéristiques, (ii) variabilité intra-locuteur: un locuteur ne peut pas répéter le même discours exactement de la même manière, et (iii) variabilité environnementale, connue sous le nom de problème extrinsèque, tel que: (a) environnement acoustique - comme la parole de fond, musique, bruit de rue, bruit de voiture, réverbération de pièce, bruit gaussien additif etc. ; (b) canal de communication - telle que des capteurs, codeurs de parole, distorsion de convolution, effets de canal non linéaire, annulation d'effacement d'écho etc. Dans les environnements acoustiques du monde-réel, plusieurs de ces variabilités se chevauchent et les mécanismes humains de perception de la parole peuvent traiter ces variabilités complexes avec succès. Cependant, dans l'ASR, on suppose que ces variabilités sont mutuellement exclusives et distinctes et ne se chevauchent pas afin de réduire ses la complexité des modèles acoustiques.

Dans les études de perception de la parole humaine, on a constaté que les êtres humains utilisent de multiples caractéristiques pour la perception de la parole dans des conditions bruyantes pour prévoir le signal de parole à partir de la source. Dans des conditions défavorables, les personnes dépistent les environnements environnants,

détectent tous changements brusques des milieux, extraient l'information similaire au bruit, les caractéristiques du locuteur, des conditions de l'environnement acoustique spécifique, et des conditions du canal et les analysent. Les mécanismes humains de perception de la parole utilisent l'information du bruit extraite pour s'adapter aux environnements en cours d'évolution et puis décoder le signal de parole pour essayer de le comprendre, c'est-à-dire, faire une certaine hypothèse sur les scores de sorties, appelés des scores de confiance basées sur certains schémas de mesure, et puis envoyer un signal de retour au mécanisme de perception de la parole si la parole décodée n'est pas intelligible. Selon des observations, on a constaté que l'adaptation et la rétroaction sont des processus continus jusqu'à ce qu'un signal de parole ne soit pas compréhensible aux auditeurs. Les personnes essaient d'utiliser des gestes du corps et le contexte des communications s'il est connu à priori. Lorsque l'on considère les communications humain-à-ordinateur, comme l'ASR et les systèmes de dialogue de machine, il est essentiel de surveiller les flux audio des locuteurs sources, les environnements acoustiques de fond, et les changements de canal puisqu'ils représentent des défis significatifs dans le maintien de la performance du système ASR. Néanmoins, il est difficile de concevoir un tel système de reconnaissance intelligent de la parole détectant l'environnement comme celui du système humain, qui explorera la nature des bruits. Actuellement, il n'y a aucun algorithme innovateur qui peut être utilisé pour surveiller les environnements acoustiques et pour analyser les bruits pour adapter les modèles acoustiques du système ASR aux conditions changeantes.

Les approches statistiques basées sur le HMM courantes comme montrées dans la Fig. 1.1 ne peuvent pas traiter la dégradation de performance des systèmes ASR lorsque le rapport signal-sur-bruit (SNR) diminue. L'estimation appropriée des pa-

ramètres HMM comprenant les probabilités de transition, les densités de sortie des états de Markov, et les statistiques de bruit échoue dans les conditions acoustiques défavorables. Un autre problème de la RAP basé sur les HMMs est lié à l'utilisation des schémas de traitement de signal HMM différé pour estimer les paramètres HMM. Les HMMs utilisent l'algorithme Baum-Welch de maximisation-d'espérance (EM), qui utilise un schéma rétroprogréssif à intervalle fixe pour estimer les meilleurs séquences d'état possibles des mots correspondant aux expressions inconnues. L'estimation des paramètres du modèle HMM exige également de grandes mémoires, et le processus d'optimisation converge lentement. Les schémas HMM différés (off-line) ne peuvent pas estimer les paramètres des modèles qui varient lentement avec le temps ou subissent des changements rapides peu fréquents en raison de l'environnement bruyant défavorable. Pour dépister dynamiquement l'environnement acoustique, mettre à jour l'information de bruit, et s'adapter au nouvel environnement, une approche alternative comme le mécanisme de perception de l'être humain doit être développée. L'utilisation des schémas HMM en ligne est susceptible d'être l'une des solutions possibles.

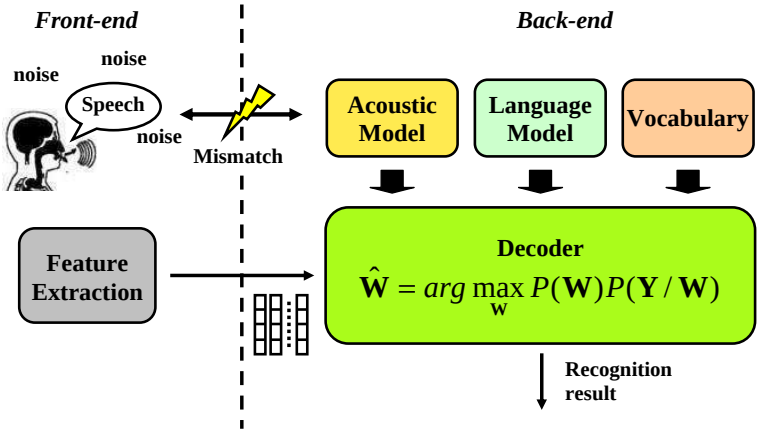


FIGURE R-1 – Système de reconnaissance automatique de la parole basé sur les HMMs.

Beaucoup d'algorithmes innovateurs ont été développés avec l'avancement de la recherche dans les domaines de l'intelligence artificielle de identification de modèle, la théorie de l'information, et aussi dans le traitement du signal et la reconnaissance. Ces algorithmes ont également trouvé des applications commerciales réussies. Dans le domaine de traitement de la parole, les différents algorithmes innovateurs sont disponibles et peuvent estimer les bruits très efficacement même pour de très faibles rapports signal-sur-bruit (SNR).

Dans le domaine de traitement du signal statistique, le filtre de particules (PF) basé sur l'algorithme Monte Carlo séquentiel (SMC) est une technique stochastique qui peut être utilisé pour estimer les signaux inclus dans des bruits hautement non-stationnaires. La prédiction et la mise à jour séquentielles du signal dans des bruits non-stationnaires sont largement utilisés pour l'estimation des paramètres des modèles en ligne dans des séries chronologiques en temps réel, telles que le marché des actions, la plate-forme pétrolière de forage, les finances etc. Dans les domaines d'optimisation, des algorithmes de recherche stochastiques sont largement utilisés pour une recherche efficace des paramètres du modèle optimal dans un espace de recherche complexe de grande dimension. Actuellement, les algorithmes d'optimisation de groupe de particules stochastique évolutionnaire PSO (evolutionary stochastic particle swarm optimization) sont devenus populaires pour résoudre le problème d'optimisation de certaines fonctions objectives dans des problèmes réels. Dans le domaine de classification de modèle, l'inférence en ligne Bayésienne pour la segmentation et le groupage ont attiré plus d'attention. Il peut être utilisé avec succès pour détecter les changements soudains des locuteurs, des conditions environnementales, et des conditions du canal. Former le champ de traitement de la parole, des méthodes statistiques pour

l'estimation du bruit, par exemple, MCRA peut être utilisé avec la technique d'inférence en ligne bayésienne pour détecter les changements lents ou rapides dans les conditions acoustiques hautement non-stationnaires. Cette approche peut être utilisée pour concevoir un système ASR en ligne à environnement détectable.

Récemment, le déploiement réussi des communications sans fil multimédia à bande large 3G/4G ont mis des demandes sur les systèmes ASR à environnement détectable pour beaucoup d'activités basées sur la voix comprenant le parcours du web basé sur la voix, recherche de musique, composition téléphonique, dictée de documents etc. Puisque les téléphones mobiles travaillent en conditions acoustiques très incertaines, ce qui s'appellent commercialement les environnements impulsifs, un système ASR différé basé sur les HMMs conventionnels ne peut pas fournir de bons services aux clients.

Afin d'améliorer la robustesse du système ASR sous les environnements acoustiques impulsifs, il est essentiel de développer des nouvelles techniques très innovatrices qui peuvent rendre le système ASR averti des conditions acoustiques de fond et s'adapte rapidement aux nouveaux changements d'environnements en temps réel.

Dans ce travail de thèsis, nous proposons une architecture d'un système ASR en ligne à détection d'environnement basé sur des technologies existantes. Nous développons de nouvelles techniques en se focalisant sur des cas spécifiques, tels que les bruits de fond non-stationnaires changeant rapidement. Les utilisations des techniques de d'exploration d'environnement pour dépister et détecter les bruits de fond à variations lentes ou soudaines et l'extraction d'information sont propose dans ce travail de thèsis. Des idées innovatrices basées sur des techniques d'inférence bayésiennes en ligne utilisant un PF et des techniques d'optimisation stochastiques basées sur les

algorithmes PSO stochastiques sont étudiées ici pour la reconnaissance simultanée et la compensation de modèle acoustique afin de s'adapter dynamiquement à de nouveaux bruits acoustiques extrêmement variables. On propose un nouveau cadre basé sur l'intégration des algorithmes de détection de bruit dynamique et la compensation de modèle simultanée en utilisant le PF et le PSO stochastique dans un système basé sur les HMMs. Cette approche mène au développement d'ASR en ligne robuste au bruit dans un environnement bruité non-stationnaire du monde réel.

Vision de recherche

Notre vision consiste â démontrer comment le problème de robustesse des systèmes actuels de la RAP dans des environnements inconnus et bruités peut être résolu en comprenant comment les êtres humains répondent aux environnements bruyants inconnus et s'adaptent rapidement à de nouvelles conditions changeantes le système des environnements ; l'incorporation de la connaissance acquise dans la RAP courant nous aiderait à contribuer à la résolution de la question de la robustesse de la prochaine génération des système de la RAP (ASR, en ligne).

Objectifs & méthodologie

- **Objectif Global:** Ce travail de thèsis aborde les questions de conception et de la robustesse au bruit des systèmes ASR en ligne, plus spécifiquement, de la reconnaissance de la parole en ligne et de sa performance dans les environnements bruyants hautement non-stationnaires pour les dispositifs mobiles portatifs.

- **Premier Objectif:** Motivé par le fait que les systèmes ASR en différé courants sont tous vulnérables dans les conditions bruyantes, notre travail est focalisé sur l'ajout d'auto-adaptabilité et la détection de l'environnement aux systèmes ASR. Les actuels approches adoptées mènent au développement d'ASR en ligne pour les environnements bruyants du monde réel.
- **Deuxième Objectif:** la détection dynamique des changements environnementaux acoustiques, l'extraction de l'information du bruit plutôt que de simples suppositions sur les bruits et la compensation du modèle dans l'espace des paramètres basé sur l'information extraite de bruit sont les premières étapes vers le développement d'un ASR à environnement détectable comme le processus humain. Par conséquent, une intégration de la connaissance de l'environnement dans l'ASR par une meilleure exploitation de notre connaissance de la production humaine de la parole et de la perception mènent â une ASR en ligne à environnement détectable robuste au bruit. Ceci facilite la conception d'un système ASR à environnement détectable avec une réduction plus importante des taux d'erreur de mot (WER) avec un coût de calcul raisonnable sur un éventail de conditions de corruption acoustiques hautement non-stationnaires.
- **Troisième Objectif:** Les sous objectifs sont: (i) étudier le mécanisme humain de décodage de la parole dans les scénarios en temps réel à différents environnements acoustiques, et (ii) développer un algorithme innovateur qui préparera le terrain pour surmonter les limitations courantes de la technologie ASR. À l'égard de ces objectifs, une compensation conjointe du bruit acoustique hautement non-stationnaire et des distorsions dans l'espace des paramètres d'une manière récursive par trames avec un lissage temporel pondéré des paramètres

est proposé afin de tenir compte les changements brusques dans les environnements.

Dans les applications en temps réel, le décodeur ASR reçoit un flux de trames des expressions parlées en direct. Le décodeur ne connaît pas à l'avance les frontières de la phrase du flux entrant des signaux de parole. Par conséquent, le décodeur ASR fonctionne sur chaque flux entrant trame-par-trame et estime le meilleur score de confiance pour chaque trame. Lorsque une frontière de mot est détectée partant d'un silence ou d'une pause, le décodeur produit le meilleur mot présumé comme les sorties de tous les scores de confiance de toutes les trames comme le mot identifié à la volée. Cependant, le décodeur ASR peut détecter une telle frontière de mot en conditions non bruitées. La frontière de mot devient floue â cahse de l'ajout des bruits de fond et des distorsions de canal. Ces bruits et distorsions biaisent également les scores de confiance de chaque trame. Par conséquent, la performance de l'ASR en temps réel se dégrade rapidement dans les conditions bruités. La performance de l'ASR en temps réel peut être améliorée en minimisant le biais (bias) des distorsions pour chaque trame. La technique de suppression du biais dynamique de trame peut être utilisée pour minimiser la distorsion de chaque trame par la soustraction de la moyenne des caractéristiques dans le domaine cepstral.

La technique de compensation en ligne dynamique par trame adaptative des distorsions du canal non stationnaire est montrée à la Fig. R-2. L'algorithme complet pour la compensation de suppression du biais dans les environnements en temps réel est basé sur une approche conjointe à canal simple pour le bruit additif hautement non-stationnaire et la compensation des distorsions du canal dans l'espace des caractéristiques dans la Fig. R-3.

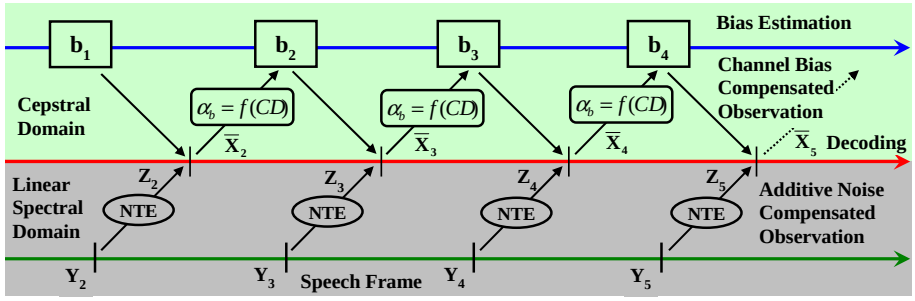


FIGURE R-2 – La technique de compensation biais conjointe dynamique adaptatif de trames (JAC) avec un paramètre de lissage temporel. NTES signifie l'estimation du bruit et la soustraction dans le domaine spectral linéaire. Elle est une fonction de la non-stationnarité des environnements.

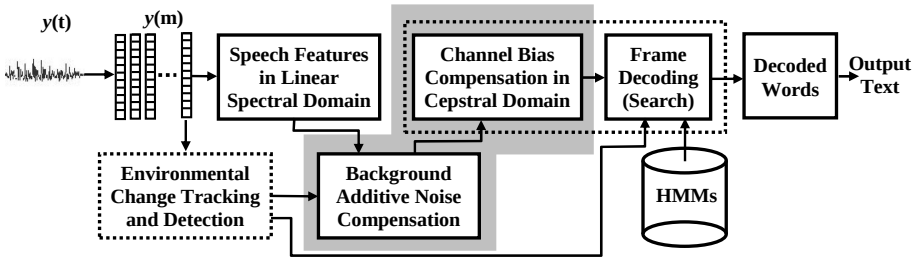


FIGURE R-3 – L'architecture proposée pour la reconnaissance automatique de la parole en ligne (ASR).

Les infrastructures ASR en ligne sont implémentés en utilisant l'util ATK [2] pour concevoir les systèmes de reconnaissance de parole qui fonctionnent en mode temps-réel dans des bruits non-stationnaires. L'ATK [2] fournit des plateformes multi-filetées (multi-threaded) afin de concevoir un ASR en ligne pour des applications du monde réel. L'ATK fournit des APIs en temps réel pour HTK. Il se compose d'une variété de composants à relier ensemble pour implémenter différents architectures et applications en temps-réel. L'architecture de base de l'ATK, comme montrée à la Fig. R-4, comprend les trois fonctionnalités principales suivantes [2]:

– **Paquet**: C'est un ensemble d'information. Les paquets sont utilisés pour transmettre une variété d'information entre des composants exécutés en asynchrones.

En particulier, les paquets sont utilisés pour transporter diverses formes d'entrée utilisateur et de signaux de sortie (la parole, marqueurs d'événement tels que les cliques de la souris, etc..). Dans ces cas, chaque paquet a un témoin de temps pour définir le segment temporel auquel il est relié. Les types de données qu'un paquet peut porter incluent des chaines de textes, des fragments de forme d'onde, des vecteurs de caractéristiques codés, des étiquettes de mot et des étiquettes sémantiques.

- **Buffer**: C'est une file d'attente de paquet FIFO. Les buffers fournissent le canal pour passer des paquets d'un composant à l'autre. Les buffers peuvent être de taille fixe ou de taille illimitée. Les composants souhaitant accéder à un buffer, peuvent tester pour voir si l'opération de buffer serait bloquée avant d'exécuter l'opération.
- **Composant**: C'est un élément de traitement. Chaque composant est exécuté dans son propre thread individuel. Les composants communiquent en passant des paquets par l'intermédiaire des buffers. En outre, les composants ont une interface de commande qui peut être utilisée pour mettre à jour les paramètres de commande lors du fonctionnement et modifier de ce fait le comportement d'exécution du composant.

Dans l'ATK, les trois ressources les plus requises, par exemple, i) le dictionnaire, ii) la grammaire (comme montré à la Fig. R-5), et iii) les HMMs de la parole non bruitée, doivent être préparés en mode batch en utilisant HTK. Pour la phase test, l'ATK fournit en ligne le décodage en utilisant HVite au lieu de HDecode dans HTK. Chacune des trois ressources exigées peut être définie comme entrées dans un fichier de configuration qui est chargé au démarrage. Un tel fichier contiendra également ty-

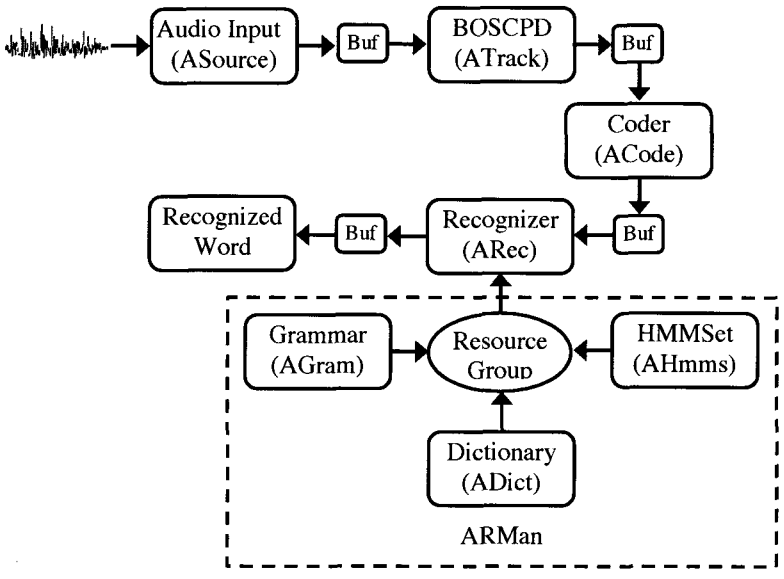


FIGURE R-4 – Architecture de base pour un système de reconnaissance en ligne [2].

piquement les spécifications des paramètres de codage. Le composant ARec dans ATK comme montré dans la Fig. R-4 fournit une fonctionnalité similaire au décodeur standard Viterbi du HTK. Il fournit également le support du modèle du langage tri-gram qui n'est pas disponible dans HTK. ARec est fourni avec un groupe de ressources contenant les ensembles HMM requis, le dictionnaire, la grammaire, et optionnellement un modèle de langage n-gram. Il décode alors les vecteurs caractéristiques entrants en conséquence.

En fonction, le système de reconnaissance en ligne ATK demeure toujours dans un des cinq états possibles comme indiqué par le diagramme d'état montré dans la Fig. R-6. Le système de reconnaissance change d'état, selon les configurations des modes de fonctionnement. L'afficheur d'ARec affiche le mode courant comme une séquence de 4 caractères: représentant les configurations pour CYCLE (1=oneshot, C=continuous), FLUSH (I=immed, M=tomark, S=tospeech), STOP (I=immed, M=tomark, S=tosilence),

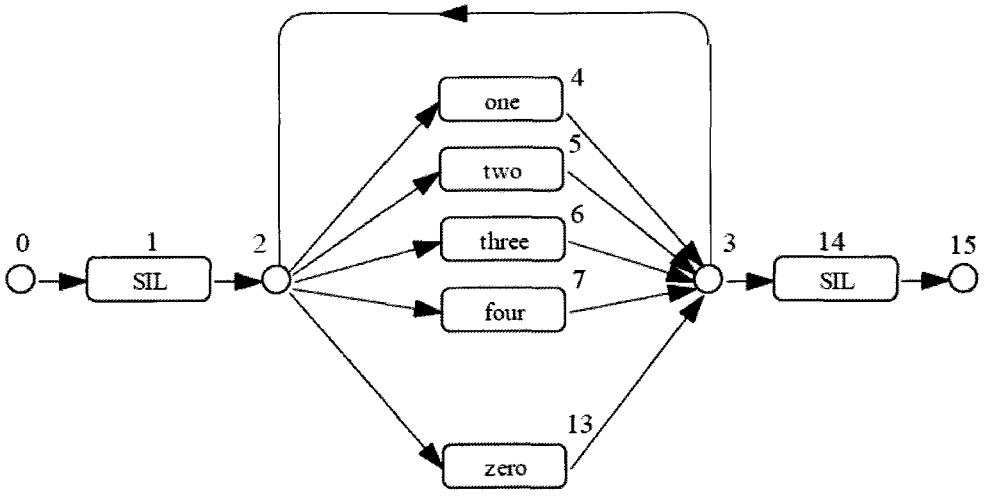


FIGURE R-5 – Reconnaissance d'un Simple chiffre Grammaire/Réseau [2].

et RESULTS (I=immed, A=asap, E=atend, X=all).

Lors de sa création, un objet ARec est placé dans l'état WAIT. Lorsqu'il est dans l'état WAIT, le système de reconnaissance attend qu'une commande Start() soit publié par l'intermédiaire de son interface de commande. Lorsque cette commande Start() est reçue, le système de reconnaissance se déplace vers l'état PRIME dans lequel il charge les ressources de reconnaissance indiquées par le groupe courant de ressource. Il se déplace alors immédiatement à l'état FLUSH d'où il prend des paquets de son buffer d'entrée et les jette jusqu'à ce qu'il soit prêt à commencer la reconnaissance comme déterminé par la configuration du mode flush. Ceci peut se produire soit immédiatement, lorsque le marqueur START est reçu, ou dès que le paquet entrant d'observation aura une armature marquée comme parole. Une fois, dans l'état RUN, le système de reconnaissance effectue la reconnaissance des paquets entrants jusqu'à soit un marqueur STOP est reçu, une trame de parole marquée comme un

silence est reçue ou une commande Stop() est publiée. Dans l'état ANS, le système de reconnaissance nettoie le traitement de reconnaissance et revient aux états WAIT ou PRIME selon la configuration du mode CYCLE. Une description plus détaillée de ce processus de reconnaissance peut être trouvé dans [2].

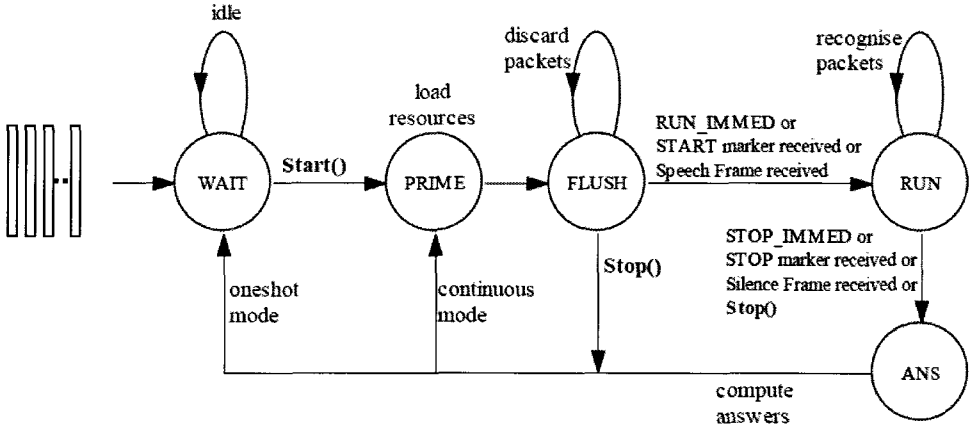


FIGURE R-6 – Diagramme de transition d'état du système de reconnaissance ATK [2].

Principaux résultats

Pour la détection et l'estimation en ligne du bruit, le délai dans la mise à jour de l'estimation du bruit tout de suite après les changements rapides des conditions acoustiques affecte sérieusement la performance de l'extraction du bruit de la parole, en particulier, dans les régions de transitions. Lorsqu'un changement se produit, la fenêtre de recherche est initialisée aux nouvelles conditions. Le bruit de fond cause un changement dans la moyenne et la variance des caractéristiques spectrales de la

parole. Pour les systèmes de détection en temps réel du bruit dans un environnement qui a pu avoir abruptement changé, l'algorithme de détection doit pouvoir dépister le bruit en se basant sur les trames précédentes de la parole. La technique d'adaptation rapide proposée basée sur BOCPD et MRCA est capable de détecter les points de changement spectral abrupt et s'adapte aux environnements acoustiques hautement non-stationnaires changeants rapidement. L'algorithme proposé est donné au tableau 1.

Le défi pour l'algorithme MCRA de détection de bruit est de mettre à jour sa fenêtre de recherche de minimum dès que les changements se produiront en condition acoustique pendant son temps de changement hautement non-stationnaire. Le BOCPD effectue la détection de changement rapide en ligne et adapte la fenêtre de recherche de minimum de l'algorithme MCRA, ce qui mène au délai minimum dans la mise à jour de la fenêtre de recherche de minimum. Un autre défi pour le détection d'environnement basé par BOCPD-MCRA et pour l'algorithme d'adaptation rapide est l'initialisation appropriée des paramètres de détection α, α_s, α_p, β, γ, et α_δ, qui est la clef pour une performance de détection réussi du bruit non stationnaire. Le ASR de base dans [1] est conçu pour évaluer la performance de la tâche Aurora 2 standard de reconnaissance d'une suite de chiffres dans le bruit et l'environnement distordu du canal. Ce système ASR de ligne de base fonctionne dans le mode batch (en différé) et utilise la configuration MFCC_E_D_A pour les caractéristiques MFCC. Il utilise les modèles HMM du mot entier avec 18 états par mot comprenant 2 états factices au début et à la fin. Ces modèles HMM sont des modèles de gauche-à-droite sans sauter-au-dessus des états. Il utilise une mixture de 6 Gaussiennes par état. Cette ASR de base est testé en utilisant HTK comme système de reconnaissance de référence.

TABLE 1 – proposition d'une adaptation rapide basée par BOCPD de d'algorithme de détection de bruit.

1:Initialization: Set
$P(r_0 = 0) \leftarrow 1$ initial run length r_0, and $P(r_0) \leftarrow \tilde{S}(r)$ run length r
$\nu_1^{(0)} \leftarrow \nu_{prior}$, $\chi_1^{(0)} \leftarrow \chi_{prior}$, $\lambda \leftarrow 250$ constant Hazard function
$cpFlag \leftarrow \{0, 1\}$ for CPD, $k \leftarrow$ frequency@kth bin in Hz
$R_m \leftarrow 0$ holds maximum run length information
2:**for each speech frame m do**:
3: Observe new DFT coefficient Y_m at kth bin of mth frame
4: Evaluate predictive probability using student t distribution
$\pi_m^r = P(Y_m|\nu_m^{(r)}, \chi_m^{(r)})$
5:Evaluate the Hazard function $H(r_m)$
6: Calculate the growth probabilities
$P(r_m = 0, Y_{1:m}) = \sum P(r_{m-1}, Y_{1:m-1})\pi_m^{(r)} H(r_m)$
7: Calculate change point probabilities
$P(r_m = 0, Y_{1:m}) = \sum P(r_{m-1}, Y_{1:m-1})\pi_m^{(r)} H(r_m)$
8: Calculate the evidence $P(Y_{1:m}) = \sum P(r_m, Y_{1:m})$
9: Determine the run length distribution
$P(r_m|Y_{1:m}) = P(r_m, Y_{1:m})/P(Y_{1:m})$
10: Update sufficient statistics. Posterior update depends on UPM
$\nu_{m+1}^{(0)} \leftarrow \nu_{prior}$, $\chi_{m+1}^{(0)} \leftarrow \chi_{prior}$
$\nu_{m+1}^{(r+1)} \leftarrow \nu_m^r + 1$, $\chi_{m+1}^{(r+1)} \leftarrow \chi_m^r + \mu(Y_m)$
11: Perform prediction
$P(Y_{m+1}|Y_{1:m}) = \sum P(Y_{m+1}|Y_m^{(r)})P(r_m|Y_{1:m})$
12: Update R_m for each run length r
13: Search the changepoint in R_m and if changepoint detects set
$cpFlag \leftarrow 1$
14: Start noise tracking algorithm
15: Search minimum psd $P_{min}(m, k)$ over D-frame window
16: Update the minimum whenever
$V_{count} ==$ length of V $(V < D)$ $\|$ cpFlag $== 1$
Reset subwindow (U) counter Ucount if cpFlag $== 1$
Set $cpFlag \leftarrow 0$
17: compute and update the noise PSD
18:**while on-coming speech frame buffer not empty**

Le tableau 2 montre les performances de la reconnaissance des chiffres de Aurora 2 en diffère (en mode batch) pour le modèle d'apprentissage propre [1]. La tâche de Aurora 2 standard de reconnaissance d'une suite de chiffres dans le bruit et les environnements distordus de canal était testée en différé pour deux modes des données d'apprentissage: (i) apprentissage en non-bruité, et (ii) apprentissage multi-condition. La performance du DSR de Aurora 2 pour des données d'apprentissage non-bruité est très faible comparée à celle avec multi-condition. Cependant, la même performance est obtenue pour des données test non-bruité dans les deux cas d'apprentissage. Nous avons décidé d'utiliser le modèle d'apprentissage propre-seulement pour notre test. L'objectif de notre ASR en ligne est de compenser les expressions test pour les environnements bruités, c'est-à-dire, pour ramener le signal de parole test proche du modèle propre. Cette tactique maintient le modèle d'apprentissage inchangé.

Nous avons examiné la tâche de Aurora 2 standard de reconnaissance d'une suite de chiffre dans du bruit non-stationnaire et l'environnement distordu du canal dans de condition en temps-réel en utilisant les mêmes paramètres de configuration utilisés dans le mode différés en utilisant ATK toolkit, l'API pour HTK en temps-réel multi-fileté (multi-threaded) [1].

Le tableau 3 et le tableau 4 montrent les performances de reconnaissance des chiffres de Aurora 2 en différé (mode batch) et en temps-réel sans compensations biaisées, respectivement.

Dans l'environnement en temps-réel, la connaissance a priori des conditions acoustiques n'est pas connue. Par conséquent, l'apprentissage de l'ASR devrait être fait sur des données propres d'abord et puis, il doit détecter les changements environnementaux pour une auto-adaptation aux nouvelles conditions. Pour des données test

TABLE 2 – Performance du système de base avec apprentissage sans bruit pour la reconnaissance des chiffres de Aurora 2 [1].

Clean training - Results														
	A					B					C			Average
	Subway	Babble	Car	Expo Hall	Average	Restaurant	Street	Airport	Station	Average	Subway (MIR)	Street (MIR)	Average	
clean	98.83	98.97	98.81	99.14	98.94	98.83	98.97	98.81	99.14	98.94	99.02	98.97	98.99	98.96
20 dB	96.96	89.96	96.84	96.20	94.99	89.19	95.77	90.07	94.38	92.35	94.47	95.19	94.83	94.06
15 dB	92.91	73.43	89.53	91.85	86.93	74.39	88.27	76.89	83.62	80.79	87.63	89.69	88.66	85.46
10 dB	78.72	49.06	66.24	75.10	67.28	52.72	66.75	53.15	59.61	58.06	75.19	75.27	75.23	66.86
5 dB	53.39	27.03	33.49	43.51	39.36	29.57	38.15	30.69	29.74	32.04	52.84	48.85	50.84	40.75
0 dB	27.00	11.73	13.27	15.98	17.00	11.70	18.68	15.84	12.25	14.62	26.01	21.64	23.83	18.48
-5 dB	12.62	4.96	8.35	7.65	8.40	5.0	10.07	8.11	8.49	7.92	12.10	10.70	11.40	9.24
Average	65.78	50.73	58.08	61.35	58.99	51.63	59.52	53.37	55.32	54.96	63.89	62.90	63.40	59.12

propres, l'Aurora 2 obtient les mémes performances dans les deux cas. Cependant, ses performances se dégradent rapidement en temps-réel comparé à ses performances en mode batch. La technique de suppression biaisée de trame-adaptative dans le domaine cepstral avec les configurations de caractéristiques MFCC MFCC_0_D_A_Z est utilisée pour améliorer la performance de la reconnaissance des chiffres reliés de l'Aurora 2. Pour la compensation biaisée cepstral dynamique de trame en temps réel, l'ASR a besoin des coefficients MFCC statiques (C_0-C_{12}). La performance de l'Aurora 2 dans l'environnement en temps-réel est montrée dans le tableau 4. Cependant une comparaison graphique des performances des expériences préliminaires sur la tâche de l'Aurora 2 de reconnaissance des chaines de chiffres dans les deux modes en ligne vs. différé (en mode batch) dans des bruits hautement non-stationnaires avec et sans compensations biaisée est montrés dans la Fig. R-7 pour des données test ensemble

TABLE 3 – Performance du système de base sans aucune compensation biaisée pour la tâche de Aurora 2 de reconnaissance des chaines de chiffre.

	Clean training - Results													
	A					B					C			Average
	Subway	Babble	Car	Expo Hall	Average	Restaurant	Street	Airport	Station	Average	Subway (MIR)	Street (MIR)	Average	
clean	97.97	98.31	98.12	98.12	98.13	97.97	98.31	98.12	98.12	98.13	97.27	97.25	97.26	97.84
20 dB	49.80	37.06	42.20	47.92	44.25	37.64	47.04	38.32	36.35	39.84	59.23	58.25	58.74	47.61
15 dB	38.78	27.45	32.30	34.46	33.25	28.95	33.89	29.44	28.05	30.08	47.26	44.38	45.82	36.38
10 dB	31.35	18.53	23.51	24.38	24.44	19.40	25.21	21.29	21.38	21.82	32.69	30.06	31.38	25.88
5 dB	22.94	12.33	16.61	17.03	17.23	10.01	17.29	14.32	12.58	13.55	22.66	22.44	22.55	17.76
0 dB	13.6	8.40	11.01	11.77	11.20	6.71	11.85	8.65	6.48	8.42	14.70	14.90	14.80	11.47
-5 dB	8.54	5.91	8.53	8.55	7.90	3.88	8.53	6.26	6.57	6.31	9.97	12.42	11.20	8.47
Average	37.57	29.71	33.18	34.60	33.77	29.22	34.59	30.91	29.93	31.16	40.54	39.96	40.25	35.06

a, la Fig. R-8 pour les données test ensemble *b*, et la Fig. R-9 pour les données test ensemble *c*, respectivement.

Les résultats expérimentaux montrent que la technique de suppression biaisée dynamique de trame-récursive dans le domaine cepstral améliore la performance de l'Aurora 2 sensiblement comparée à ses résultats obtenus par le système de base. Puisque les données test pour Aurora 2 sont des phrases pré-enregistrées, le décodeur ASR dans ATK lit une phrase à chaque fois de la liste de données test et imite les expressions parlées en temps-réel en envoyant un flux de trames au décodeur. L'installation de l'expérience de base suit le standard fourni par ETSI [3].

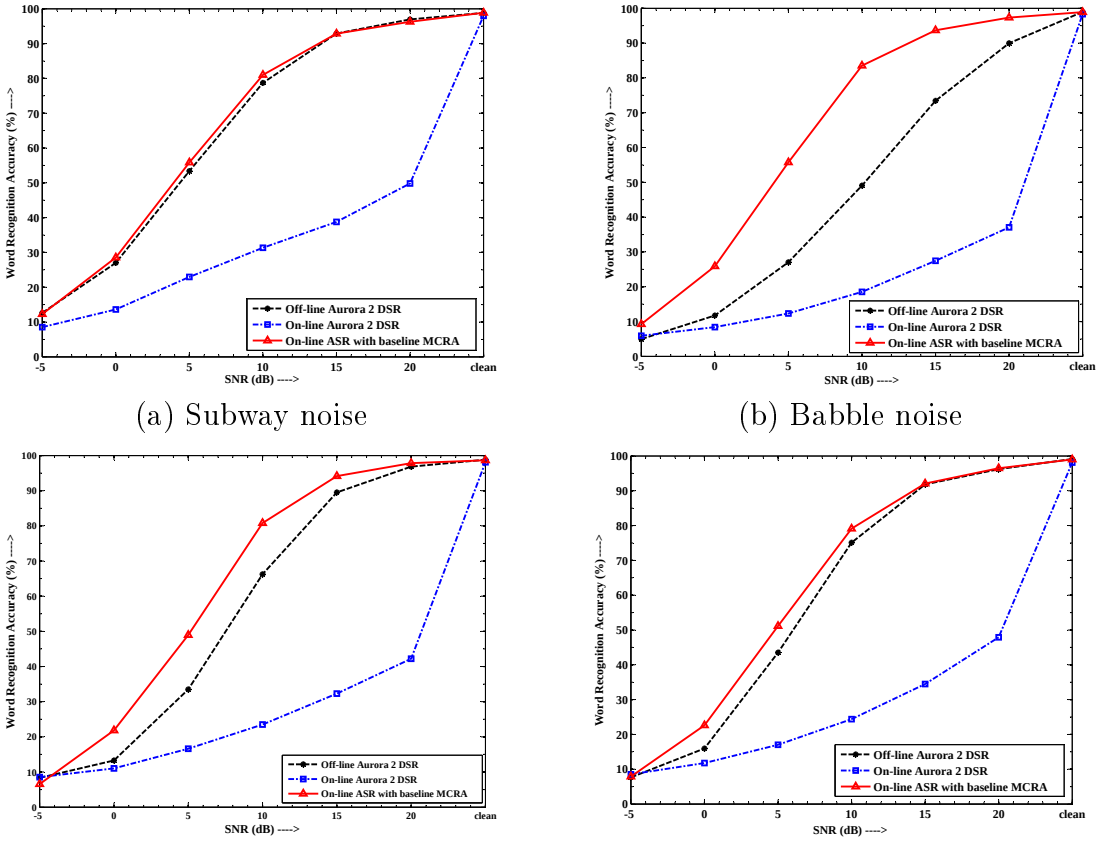
Pour confirmer la validité de la technique d'adaptation spectrale rapide proposée, nous avons comparé la performance de cet algorithme à la technique de base la plus populaire MCRA [4] pour dépister et estimer les bruits non-stationnaires. À partir des résultats de simulation, on peut voir que notre méthode proposée performe avec

TABLE 4 – Performance du système de reconnaissance pour la technique de suppression biaisée récursive de trame de la tâche de l'Aurora 2 de reconnaissance des chaines de chiffre.

	Clean training - Results													
	A					B					C			Average
	Subway	Babble	Car	Expo Hall	Average	Restaurant	Street	Airport	Station	Average	Subway (MIR)	Street (MIR)	Average	
clean	98.86	98.88	98.66	99.01	98.85	98.86	98.88	98.66	99.01	98.85	98.89	98.82	98.86	98.86
20 dB	96.28	97.34	97.85	96.48	96.99	97.45	96.92	97.35	97.99	97.43	95.98	96.70	96.34	96.92
15 dB	92.85	93.68	94.15	92.07	93.19	94.47	92.41	94.66	95.25	94.20	91.68	93.08	92.38	93.26
10 dB	80.99	83.52	80.79	79.11	81.10	85.02	80.05	88.04	83.86	84.24	79.40	78.63	79.02	81.46
5 dB	55.85	55.75	48.97	51.13	52.93	62.65	53.93	65.45	57.33	59.84	54.10	52.77	53.44	55.41
0 dB	28.53	25.89	21.83	22.63	24.72	30.37	24.43	35.95	26.53	29.32	28.62	23.03	25.83	26.63
-5 dB	12.27	9.26	6.56	7.88	8.99	11.07	8.36	11.72	9.00	10.04	11.84	8.87	10.36	9.80
Average	66.52	66.33	64.12	64.04	65.25	68.56	65.00	70.26	67.00	67.71	65.79	64.56	65.18	66.05

excellence pour les plus mauvais scénarios où les conditions acoustiques changent rapidement de conditions de SNR très élevé a un SNR très faible. Les résultats de ces simulations sont montrés dans les Figs. R-10, R-11, R-12, et R-13, respectivement.

L'ASR en ligne proposé est testé en utilisant des données test ensemble du corpus de parole Aurora 2 dans deux environnements bruyants - 1) Metro, et 2) Discussion. Pour confirmer la validité de l'ASR proposé utilisant BOCPD-MCRA, nous avons comparé sa performance à la technique MCRA de base [4]. Dans le système de base, une technique de normalisation de la moyenne cepstral (CMN) est utilisée pour supprimer le biais du canal du signal de parole test. Pour notre technique proposée, nous avons utilisé la technique de suppression du biais du canal dynamique de



(a) Subway noise (b) Babble noise

(c) Car noise (d) Exhibition hall noise

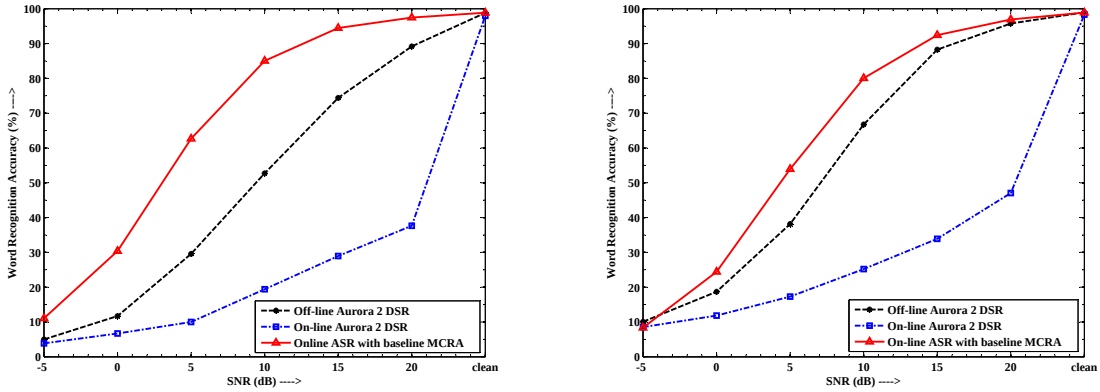
FIGURE R-7 – Performance de reconnaissance de Aurora 2 (Test ensemble A: filtré par ITU-T G.712).

trame-récursive comme décrit dans Eqs. 1 et 2:

$$\bar{\mathbf{b}}_t = \alpha_{w_t} \bar{\mathbf{b}}_{t-1} + (1 - \alpha_{w_t}) \tilde{\mathbf{y}}_t \tag{1}$$

$$\bar{\mathbf{x}}_t \approx \tilde{\mathbf{y}}_t - \bar{\mathbf{b}}_t \tag{2}$$

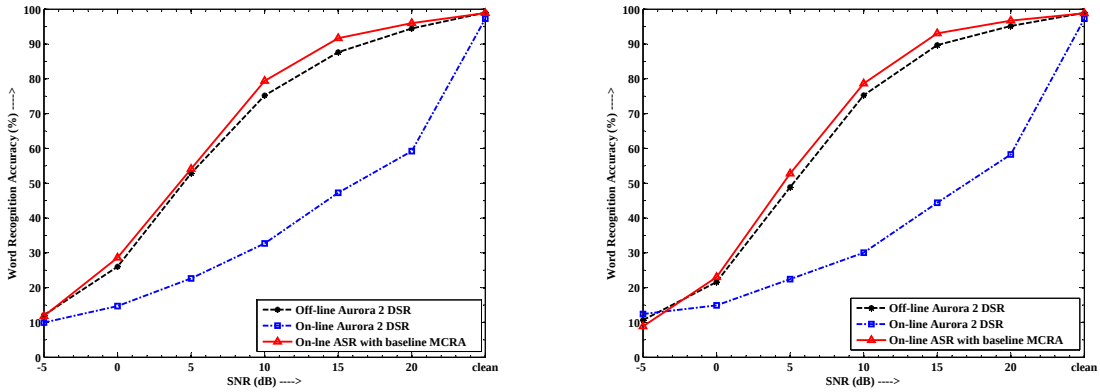
À partir des résultats de simulation, on peut voir que notre méthode proposée



(a) Restaurant noise (b) Street noise

(c) Airport noise (d) Train noise

FIGURE R-8 – Performance de reconnaissance de Aurora 2 (Test ensemble B: filtré par ITU-T G.712).



(a) Subway noise (b) Street noise

FIGURE R-9 – Performance de reconnaissance de Aurora 2 (Test ensemble C: filtré par MIR).

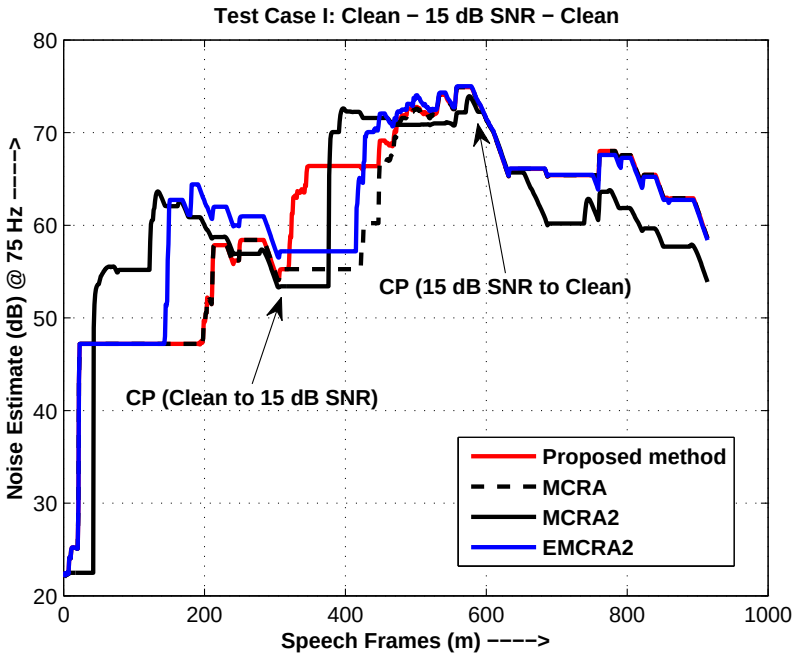


FIGURE R-10 – Performance de la technique d'adaptation rapide proposée contre des techniques de détection de bruit basée sur MCRA pour le cas de test I.

se comporte bien compare à la méthode de base. La méthode proposée améliore la performance de l'ASR en ligne dans les deux cas des environnements bruités non-stationnaires comme montré dans la Fig. R-14 et la Fig. R-15 pour les bruits de métro et de discussion, respectivement.

L'algorithme pour cette technique de compensation de bruit de canal simple basée par PSO (Particle Swarm Optimization) est décrit dans les algorithmes 0.1, 0.2, et 0.3

Conclusion

Cette thèsis a abordé un certain nombre de questions non résolues liés â la non-stationnarité des environnements acoustiques pour la reconnaissance automatique

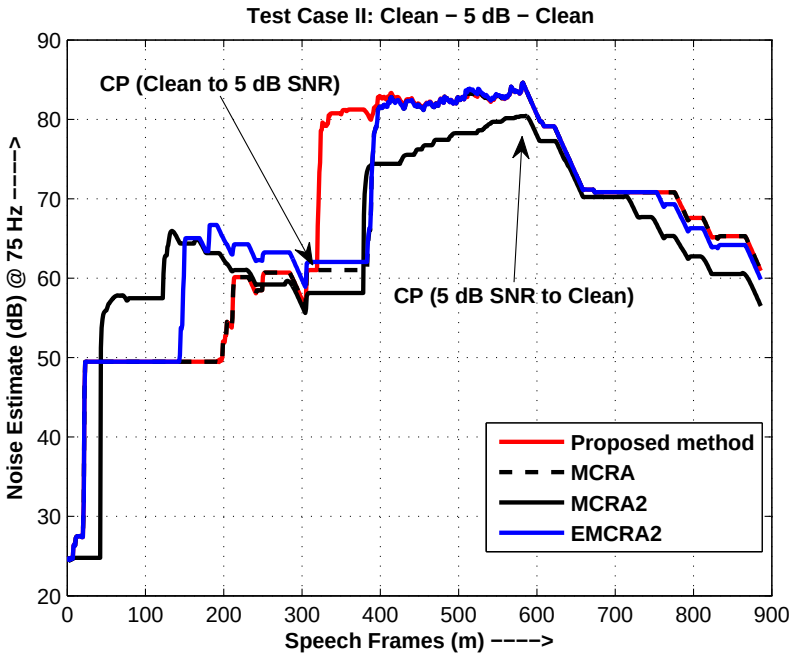


FIGURE R-11 – Performance de la technique d'adaptation rapide proposée contre des techniques de détection de bruit basée sur MCRA pour le cas de test II.

robuste de la parole. Les questions principales sur lesquelles nous nous sommes focalisées dans ce traval sont dans les domaines de traitement et de reconnaissance de la parole, tels que détecter les changements environnementaux brusques dans des conditions en temps réel. Plus spécifiquement, nous nous intéresssons â la robustesse d'ASR en adoptant une compensation des distorsions de canal et de bruit additif conjoint en configuration dynamique de trame en ligne JAC (an on-line frame dynamic joint additive and channel distortions compensation) dans les environnements acoustiques hautement non-stationnaires. L'ASR obtenue en ligne a une performance supérieure dans des conditions bruyantes non-stationnaires par rapport à l'état actuel de l'ASR qui fonctionne dans le mode batch (en différé).

Nous avons entamé notre travail par une étude étendue des questions de la robus-

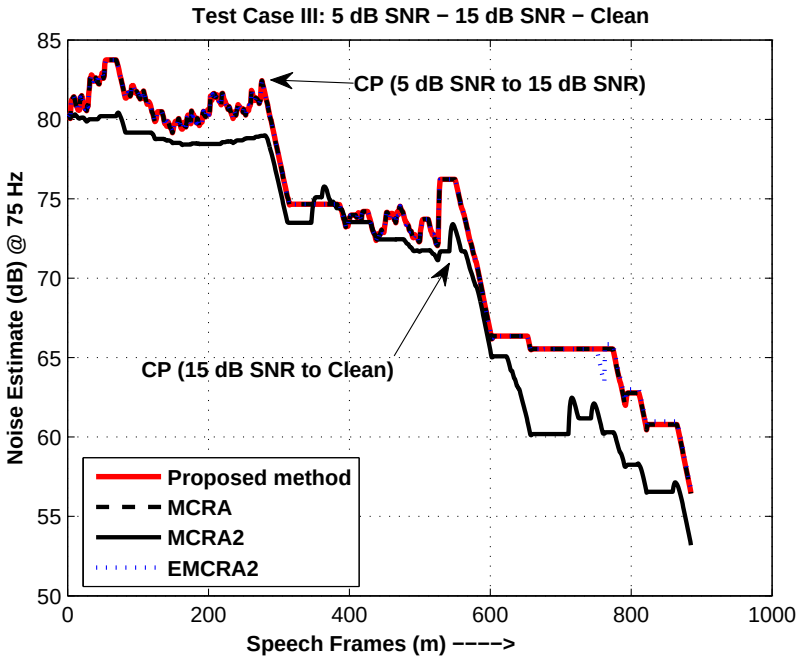


FIGURE R-12 – Performance de la technique d'adaptation rapide proposée contre des techniques de détection de bruit basée sur MCRA pour le cas de test III.

tesse des technologies de l'état de l'art d'ASR. Nous avons développé des technologies pour concevoir et analyser la performance d'un ASR en ligne robuste au bruit, avec une détection de bruit hautement non-stationnaire, la détection de changement soudain dans les environnements acoustiques, et la compensation conjointe de signal de parole observé dans des fonctionnalités de conditions bruitées pour la reconnaissance de la parole. Nous avons ajouté ces fonctionnalités dans le traitement bout-en-bout (front-end) d'ASR et des étapes de décodage afin de simuler l'ASR en ligne. Nous avons discuté la technique de réduction de bruit musical pour minimiser la distorsion de la parole basée sur le filtrage perceptuel et la modélisation du masquage du bruit de la parole, qui peuvent être optionnellement déployés pour un pré-traitement de l'ASR.

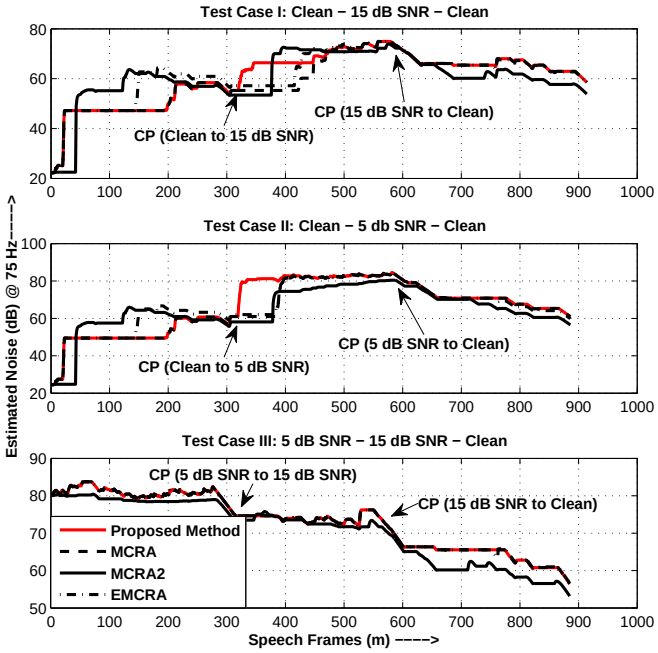


FIGURE R-13 – Performance de la technique d'adaptation rapide proposée contre des techniques détection de bruit basée sur MCRA pour les cas de test I, II, et III.

Nous avons développé une compensation par trame synchrone séquentielle du biais du bruit (frame synchronous sequential noise bias compensation) et la reconnaissance de la parole dans des conditions bruitées basées sur la technique d'inférence en ligne bayésienne. La technique de détection de point de changement en ligne bayésienne en association avec les algorithmes MCRA classiques ont été implémentées celes-ci peuvent être vues comme des technique de calcul dans le pré-traitement en arrière plan du système ASR pour fonctionner dans les environnements en temps réel. Le système ASR en ligne est indépendant du locuteur et indépendant de la tâche et l'apprentissage a été effectue sur un vocabulaire de 8440 mots. Il utilise des HMMs de 18 états avec 3 mixtures de distribution gaussienne continue. La topologie du modèle du système ASR présenté dans cette thèsis semble être optimale et donne le meilleur

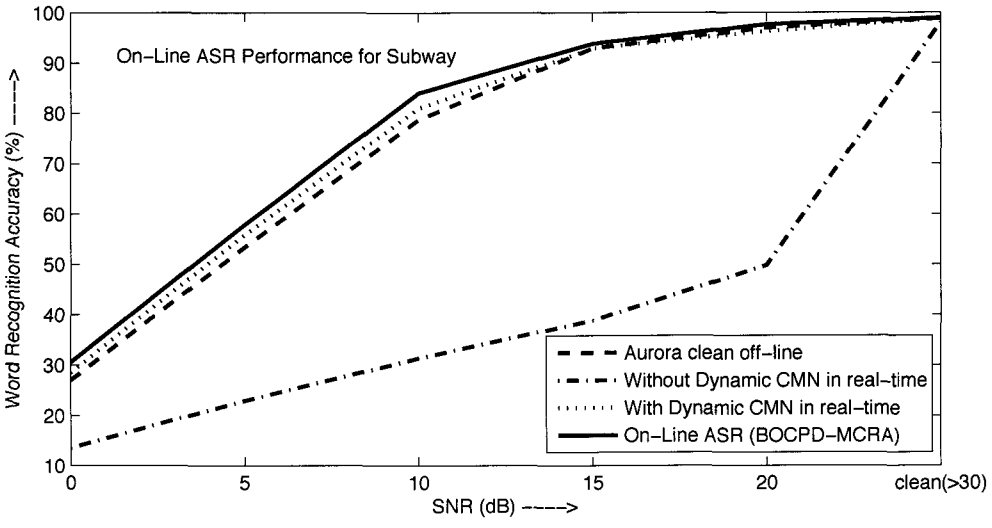


FIGURE R-14 – Performance de BOCPD-MCRA proposé pour ASR en ligne basé par JAC dans l'environnement de Metro.

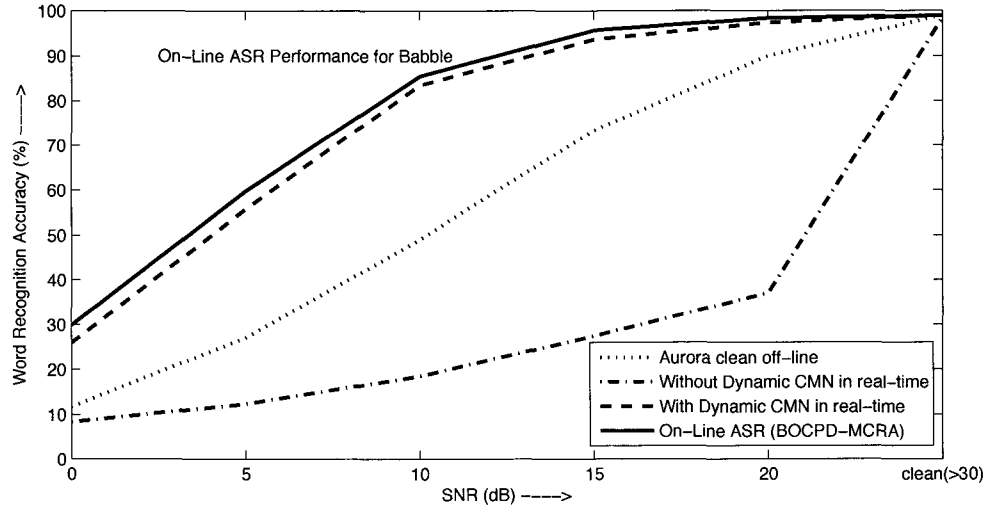


FIGURE R-15 – Performance du BOCPD-MCRA proposé pour ASR en ligne basé par JAC dans un environnement de discussion.

résultat de l'ASR.

Dans le cas de la technique de dé-bruitage synchrone de trame de parole basé sur

Algorithm 0.1 Pseudo Code for Particle Swarm Optimization (PSO) Algorithm

Input: $N \leftarrow$ No. of Particles; $S \leftarrow$ Swarm; $P \leftarrow$ Best position.
Set:
  $t \leftarrow 0$
  $S \leftarrow$ initial value
  $P \equiv S$
Evaluate S and P, and define index g of the best position
while termination criterion not met **do**
  Update S using Eqs. 4.1 and 4.2, respectively
  Evaluate S
  Update P and redefine index g
  Set: $t \leftarrow t + 1$
end while
Record best position

JAC, nous avons étudié plusieurs algorithmes populaires, et avons également proposé nos propres méthodes. Nous avons proposé deux techniques: (i) Inférence en ligne bayésienne en même temps que des algorithmes de détection de bruit MCRA avec un filtre de bande critique perceptuel, et (ii) la prédiction séquentielle et l'adaptation du signal de parole basées sur une approche de modélisation non-Gaussienne et non-stationnaire en utilisant la technique d'optimisation par essaim de particules (PSO) pour la reconnaissance de la parole en ligne dans des conditions acoustiques réelles. Nous avons prouvé par des expériences que ces deux approches peuvent aider à améliorer la performance de reconnaissance de la parole d'un système ASR pour des applications en temps réel. Par l'intégration des algorithmes de filtrage PSO, la performance de l'ASR peut être améliorée dans différentes conditions bruyantes, comparées avec l'utilisation des modèles autonomes de régression linéaire classique. Les filtres de particules sont donc les candidats prometteurs pour d'autres études de reconnaissance automatiques de la parole et pour des applications pratiques, par exemple, les applications ASR robustes aux bruits dans les environnements mobiles.

Algorithm 0.2 Non-Stationary Noise Compensation Using DMS-PSO [5]

Set: $m \leftarrow$ Each swarm's population size
Set: $n \leftarrow$ Swarm's number
Set: $R \leftarrow$ Regrouping period
Set: $L \leftarrow$ Local refining period
Set: $L_FEs \leftarrow$ Max fittest evaluations (FE) using in the local search
Set: $Max_FEs \leftarrow$ Max fitness evaluations, stop criterion
Set: $m \times n \leftarrow$ Particles initialization for position and velocity
Set: $m \times n \leftarrow$ Particles initialization for position and velocity
Set: $FEs \leftarrow 0$
Set: $gen \leftarrow 0$
while $FEs < 0.95 \times Max_FEs$ **do**
  $gen = gen + 1$
  for $i = 1 \quad to \quad (m \times n)$ **do**
    $Find \quad lbest_i$
    for $d = 1 \quad to \quad D$ **do**
      if $rand < 0.5$ **then**
        $V_i^d = w \times V_i^d + c_1 \times rand1_i^d \times (pbest_i^d - X_i^d) + c_2 \times rand2_i^d \times (lbest_i^d - x_i^d)$
        $V_i^d = min(max(V_i^d - V_{max}^d), V_{max}^d)$
        $X_i^d = X_i^d + V_i^d$
      else
        $X_i^d = pbest_i^d$
      end if
    end for
    if $X_i \in [X_{min}, X_{max}]^D$ **then**
      Calculate the fitness value
      $FEs = FEs + 1$
      Update *pbest*
    end if
  end for
  if $mod(gen, L) == 0$ **then**
    Sort *lbest* according to their fitness value and refine the first $\lceil 0.25n \rceil$ best *lbest* using Quasi-Newton method
    $FEs = FEs + \lceil 0.25n \rceil \times L_FEs$
    Update the corresponding *pbest*
  end if
  if mod(gen,R)==0 **then**
    Regroup the swarms randomly
  end if
end while

Algorithm 0.3 Non-Stationary Noise Compensation Using DMS-PSO

Set: Frame index $m \leftarrow 0$
while Frame of speech not end **do**
  $r(m) \leftarrow d(m)$
  Find best particle from DMS-PSO Algorithm 3.2
  Evaluate $W(z)$ using best particles as the filter coefficients
  $y(m) \leftarrow W(z) \otimes r(m)$
  $e(m) \leftarrow (d(m) - y(m))$
  $\hat{x}(m) \leftarrow e(m)$
  $m \leftarrow m + 1$
end while

Cet these a proposé une architecture d'ASR en ligne en utilisant un algorithme rapide d'adaptation basé par BOCPD-MCRA pour des bruits inconnus et non-stationnaires variant rapidement. L'architecture ASR en ligne proposée a exploité les avantages de l'inférence en ligne bayésienne basée sur un calcul flexible pour la technique de détection de point de changement (BOCPD).

Nous avons intégré cet algorithme dans les algorithmes bien connus de détection de bruit basés par MCRA utilisant les données de parole de Aurora 2 qui démontrent les ensembles de données simulés du monde-réel. Le cadre de travail ASR en ligne proposé qui est basé sur l'adaptation rapide d'une fenêtre glissante à de nouvelles conditions acoustiques fournit la délinéation (delineation) commode de l'implémentation de l'algorithme de point de changement dans l'architecture des algorithmes de détection de bruit basés par recherche du densité spectrale de puissance (psd) minimum courant. D'après les résultats expérimentaux, nous avons constaté que non du nouvel algorithme ASR en ligne nous permet de décoder la trame des expressions de la parole test dynamiquement à différentes conditions SNR pour des environnements hautement non-stationnaires. Cependant, il a besoin davantage d'amélioration pour atteindre une précision de reconnaissance plus élevée à des conditions de SNR plus

faibles comparé au système de base.

Nous travaillons sur la technique de prédiction séquentielle en ligne bayésienne et d'estimation pour améliorer plus encore la performance de l'ASR en ligne proposé pour le déploiement réel dans les environnements acoustiques non-stationnaires.

Comme futur travail de recherche, les points suivants peuvent être considérés:

- La performance de la reconnaissance de la parole dans les environnements acoustiques réels pourrait être améliorée en prolongeant l'approche de recherche courante pour tenir compte d'un modèle acoustique plus réaliste basé sur une modélisation non-linéaire et non-Gaussienne pour aborder le problème largement discuté de non-stationnarité pour l'ASR.
- La prédiction et l'adaptation séquentielles bayésiennes avec le l'amélioration fine de HMMs courant avec de multiples flux, des paramètres normalisés, augmentation des données d'apprentissage, et une large taille de vocabulaire peuvent être de nouvelle direction de recherche pour un ASR à environnement détectable comme il est fait par le processus humain.
- L'inclusion des modèles de comportement inspirés par les systèmes biologique peut faire qu'il soit possible de contribuer â résoudre les problèmes de performance de l'ASR courant dans des conditions acoustiques réelles. Le PSO est une telle approche qui pourrait ouvrir de nouvelle direction de recherches pour l'ASR robuste au bruit.
- Les paramètres du JAC proposé et de l'ASR en ligne basé par PSO peuvent être encore optimisés expérimentalement, afin de minimiser les distorsions de la parole et les artefacts.

Abstract

This dissertation develops new techniques to improve the noise robustness of automatic speech recognition (ASR) in highly non-stationary real-world on-line acoustic environments. It examines human-like soft attributes, namely, environment-awareness and self-adaptability. One of the main shortcomings of current ASR to maintain high performance consistently in diverse test environments is its poor ability in handling the non-stationarity of unknown test conditions, specifically in an on-line mode. Currently, researchers are trying to learn from nature to solve complex problems. The soft computing technique is an outcome from this study. It is a biological problem solving model inspired by nature. In this thesis, a novel soft computing approach using Bayesian on-line inference is proposed that detects rapid changes in the test acoustic conditions followed by updating joint background noise and channel distortion compensation (SJAC) for on-line ASR. In contrast to conventional hard computing techniques to compensate the non-stationary background noises and distortions for current ASR in batch mode (off-line), the bio-inspired soft modeling techniques prove to provide more flexible noise processing and handling capability to track the changes in acoustic environments with time and adapt ASR in on-line mode to these previously unseen real-life ambiguous conditions.

In this dissertation, we carefully studied the Bayesian online change point detec-

tion (BOCPD) technique in the context of tracking and detecting the rapidly changing non-stationary acoustic conditions. Based on the study results, we develop a novel soft computing model to detect the rapid variations in the acoustic environments by monitoring the statistical properties of the noisy speech signal. In real-life situations, the non-stationarity of the environments introduces variations in the statistical properties of the speech signal. Both the mean and the variance of the power spectral density of speech signals remain unknown and subject to changes with the changes in the acoustic environments. In our proposed soft model, we incorporate the BOCPD technique into the well known minima controlled recursive averaging (MCRA) noise tracking algorithm to track and compensate environmental distortions in the feature space and adapt the ASR to new conditions in on-line mode with minimum delay in response to the rapid environmental variations. The proposed soft technique, called the Bayesian on-line spectral change point detection (BOSCPD), achieves significant improvement in recognition performance of on-line ASR compared to the baseline MCRA-based technique when evaluated over the Aurora 2 speech database.

Towards applying soft computing to improve the noise robustness of ASR in on-line conditions, we introduce the well known evolutionary particle swarm optimization (PSO)-based soft computing technique in the second part of this dissertation. PSO has previously been applied to optimization of highly non-linear multi-modal objective functions. We use a PSO-based soft technique for adaptive modeling of the real-world acoustic environments that are non-linear and non-Gaussian. We implement a dynamic multi-swarm PSO technique, called DMS-PSO, to track and compensate the non-stationary noises by adaptively tracking the rapid variations in the real-life unseen ambiguous environments. From the experimental results, we find that the

PSO-based soft computing technique improves the performance of the on-line ASR significantly compared to the proposed BOSCPD technique in highly non-stationary acoustic environments.

The soft computing techniques explored in this dissertation prove to add human-like attributes to current stat-of-the-art on-line ASR to compensate the non-stationary distortions and improve the recognition performance in the unknown test conditions. The experimental results show that the soft modeling technique may be an alternative approach to improve the noise robustness of current ASR.

Acknowledgments

I would like to express my deep gratitude to my research director, Prof. Douglas O'Shaughnessy, and co-research director, Prof. Sid-Ahmed Selouani, for their invaluable guidance and constructive comments throughout the work of this PhD dissertation. They were my mentor, role model, and friend during this PhD research and study in INRS-EMT. Successful completion of this dissertation would not have been possible without their advice and support.

I am expressing my especial thanks to Prof. Douglas O'Shaughnessy for giving me the opportunity to work in the most fascinating field of speech recognition at the well-known *Institut National de la Recherche Scientifique* (INRS) of the University of Quebec, and ushering me into a new perspective in Telecommunications Engineering. Working with Prof. Douglas O'Shaughnessy has been a wonderful experience because of his astounding theoretical insight, endless energy and enthusiasm, honesty, and sense of humor.

The best part of this dissertation has been working with Prof. Sid-Ahmed Selouani for his enthusiasm and willingness to share knowledge and experiences. This dissertation has been truly collaborated, and without his expertise it never would have finished. I have developed a deep appreciation for his technical ability to assist me with accuracy effectiveness since the very beginning of experimental path, but

more important I have enjoyed his very comfortable accompany.

I am also indebted to office staffs of INRS-EMT at Montreal, especially Madam Hélène Sabourin, Madam Nathalie Aguiar, and Sylvain Fauvel for their tireless administrative and technical support during my PhD research in INRS-EMT. I will never forget their contribution for finishing this dissertation.

Of course, I also would like to thank the reviewers of my dissertation, Dr. Tiago Falk, Dr. Michael Picheny, and Prof. Stephen Zahorian, for their careful reading and valuable points in revising my dissertation.

I would like to express my gratitude to Dr. Mouloud Djamah, a former PhD researcher in speech communication group of INRS-EMT, for his valuable time and support to translate the resume of this dissertation from English to French language.

I would also like to express my thanks to Prof. Ponnuthurai Nagaratnam Suganthan, Associate Professor of *Nanyang Technological University*, Singapore for his documents and source codes on Dynamic Multi-Swarm Particle Swarm Optimization (DMS-PSO) technique to validate my proposed PSO-based SJAC ideas for on-line ASR.

Finally, I am grateful to my wife, my son, and my daughter for their patience and love. Without them this work would never have come into existence.

Md Foezur Rahman Chowdhury

Montréal, April 2012

Contents

List of Figures

List of Notations

$\Delta\Delta$	Acceleration coefficients of MFCC feature 114
$\eta(n)$	Additive noise for dual-channel PSO model 100
ι	Iteration counter in PSO 94
$\lambda \equiv 1/\sigma^2$	Precision matrix 172
$\tilde{\mathbf{y}}_m$	Additive noise compensated observed speech signal 84
$\mathbf{B}$	Parameter for model adaptation function 49
$\mathbf{b}$	Channel bias in the cepstral domain 26
$\mathbf{n}$	Additive background noise in the cepstral domain 26
$\mathbf{r}(\mathbf{x}, \mathbf{n}, \mathbf{b})$	A correction vector for speech model in the cepstral domain ... 155
$\mathbf{s}(\mathbf{x}, \mathbf{n}, \mathbf{b})$	A correction vector for speech model in the cepstral domain ... 156
$\mathbf{X}$	Feature vectors of the clean speech signal 46
$\mathbf{x}$	Clean speech signal in the cepstral domain 26
$\mathbf{Y}$	Feature vectors of the noisy speech signal 46
$\mathbf{y}$	Observed speech signal in the cepstral domain 26
$\mathcal{N}(\mu, \Sigma)$	Normal distribution with mean μ and covariance Σ 46
μ_x	Mean of the clean speech signal 46
ν	Parameter of the inverse function in feature space 47
ω	Angular velocity in rad/sec for the DTFT transformation 152
$\Omega(n)$	Output noise of the adaptive filter $W(z)$ in DMS-PSO 106
ω_k	kth spectral band in the DFT domain 152
Π_i	Objective function for ith particle in DMS-PSO model 106
Σ_x	Covariance of the clean speech signal 46
τ	$\in [1, ..., (t-1)]$ 74
θ	Hyper parameter for UPM model in BOCPD 74
θ_ν	Predictive model parameter for UPM in BOCPD 74
$\theta_{h_{const}}$	Parameter for constant hazard function 78
θ_h	Parameter for hazard function in BOCPD 74
$\tilde{\alpha}_n(m, k)$	Time varying smoothing parameter 60
$A \subset R^{n_d}$	n_d dimensional search space in PSO 93

x_{scs}	Scaled raw confidence score for word 162
$Y(\omega)$	DTFT of the observed speech signal $y(t)$ 152
$Y(m,k)$	STFT of the noisy speech signal 58
$y(t)$	Observed noisy speech signal in the time domain 43
$\sigma_n^2(m,k)$	Noise variance for the kth frequency bin in the mth frame 60

List of Tables

List of Acronyms

ADSP	Adaptive Digital Speech Processing
AGA	Adaptive Gaussian Attenuation
AM	Amplitude Magnitude
ASR	Automatic Speech Recognition
ATK	Real Time API for HTK
BIC	Bayesian Information Criterion
BOCPD	Bayesian On-Line Change Point Detection
BOSCPD	Bayesian On-Line Spectral Change Point Detection
CD	Change Detection
CMLLR	Constrained Maximum Likelihood Linear Regression
CMN	Cepstral Mean Normalization
CMNVS	Cepstral Mean Normalization for Variance Scaling
CVC	Consonant Vowel Consonant
dB	decibel
DFT	Discrete Fourier Transform
DL	Difference Limen
DMS-PSO	Dynamic Multi-Swarm Particle Swarm Optimization
DSP	Digital Signal Processing
DSR	Distributed Speech Recognition
DTFT	Discrete Time Fourier Transform
DTW	Dynamic Time Warping
EM	Expectation Maximization
EMCRA	Enhanced MCRA
ETSI	European Telecommunications Standards Institute
FBSM	Frame-Based Stochastic Matching
FFT	Fast Fourier Transform
FIFO	First-In First-Out
fMLLR	Feature Space MLLR
FSSM	Frame Synchronous Stochastic Matching
FST	Finite State Transducer
GMM	Gaussian Mixture Model
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GSNR	Global Signal-to-Noise Ratio
HLDA	Heteroscedastic Linear Discriminant Analysis
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
IFFT	Inverse Fast Fourier Transform
IMCRA	Improved Minima Controlled Recursive Averaging
ITU-T	International Telecommunication Union - Telecommunications
It-Sa	Itakura-Saito Distortion
JAC	Joint additive and channel distortions compensation
kHz	kilo Hertz
KL	Kullback-Leibler
LDA	Linear Discriminant Analysis
LAFS	Lexical Access From Spectra
LeSF	Least Square Filtering
LMS	Least Mean Square
LPC	Linear Predictive Coding
LTI	Linear Time Invariant
LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Maximum *a posteriori*
MCRA	Minima Controlled Recursive Averaging
MFCC	Mel Frequency Cepstral Coefficient
MIRS	Modified Intermediate Reference System
ML	Maximum Likelihood
MLF	Master Label File
MLLR	Maximum Likelihood Linear Regression
MLP	Multi-Layer Perceptron
MMSE	Minimum Mean Square Error
MOS	Mean Opinion Score
NLM	Noise Language Model
NLMS	Normalized Least Mean Square
MSS	Magnitude Spectral Subtraction
NSF	Noise Shaping Filter
NSS	Nonlinear Spectral Subtraction
PE	Pitch Estimation
PESQ	Perceptual Evaluation of Speech Quality
PLP	Perceptual Linear Prediction
PMC	Parallel Model Combination
PSD	Power Spectral Density
PSO	Particle Swarm Optimization
PSS	Power Spectral Subtraction
PSTN	Public Switched Telephone Network
PWF	Perceptual Weighting Filter
RASTA	Relative Spectra
RLS	Recursive Least Square
SE	Speech Enhancement
SegSNR	Segmental SNR
SC	Soft Computing
SLF	Standard Lattice Format
SM	Stochastic Matching
SNR	Signal-to-Noise Ratio
SS	Spectral Subtraction
SSM	Sequential Stochastic Matching
STFT	Short Time Fourier Transform
STSA	Short Time Spectral Amplitude
SVF	Spectral Variation Function
TI	Texas Instruments
UV	Unvoiced
VTS	Vector Taylor Series
V	Voiced
VAD	Voice Activity Detector
VC	Vowel Consonant
VCV	Vowel Consonant Vowel
VTLN	Vocal Tract Length Normalization
WSS	Weighted Spectral Slope
WER	Word Error Rate
WF	Wiener Filter

Chapter 1

Introduction

Speech recognition is a process of converting spoken speech utterances to words or text. This text can be the final output or the input to natural language processing. Due to the natural and efficient characteristics of speech in exchanging information, it becomes the fastest way human beings can communicate with machines.

With the advent of modern fast computing and broadband telecommunications technologies, the interfaces between men and machines become more realistic for information access and management when (i) the information space is broad and complex, (ii) the users are technically naive, and (iii) only telephones are available. Most commonly used speech-based communication interfaces between men and machines are: (a) simple speech input recognition, such as command and control, data entry over the telephone, dictation, transcriptions: legal, medical, TV, and (b) interactive conversation and machine understanding, such as information kiosks, transactional processing, and intelligent agents, music browsing, web browsing, car control and navigation, Global Positioning System (GPS) navigation etc.

1.1 Background & Motivation

Current state-of-the-art ASR has found its successful commercial applications for everyday usage of man-machine interfaces. ASR reached its current position as a result of continuous research efforts of many speech scientists, engineers, and linguists during the last three decades in developing cutting-edge technologies for speech recognition based on data-driven hidden Markov model (HMM) techniques. It is only in the last decade that speech recognition technologies emerged from speaker-dependent platforms to speaker-independent platforms and from small vocabulary to large vocabulary continuous and spontaneous speech recognition forms. In spite of these developments, the performance of ASR is still far from human speech perception performance.

ASR performs exceptionally well under known controlled acoustic environments. A fundamental problem is that ASR performance degrades quickly when the training and testing environments do not match, i.e., performance is unsatisfactory in unknown test environments [6], [7]. These mismatches are due to inter- and intra-speaker variabilities, acoustic background environments, microphones, and channel variabilities. Many cutting edge techniques have been developed to minimize these variabilities, and most of them perform successfully within a specific context-dependent consistent acoustic environment [8], [9]. These techniques are based on assumptions concerning the noise or the differences in collecting and training on a specific noise condition. Most of these techniques need ASR to work in a batch mode, i.e., ASR decodes a whole speech utterance as a unit [2], [7].

Human speech perception mechanisms inside the brain are still poorly understood

and remain as a black box to speech researchers. However, researchers have observed that during human-to-human conversation, people monitor the speaker as well as the surrounding acoustic environment continuously under adverse conditions and they have the ability to quickly adapt to changing acoustic environments [10], [11]. The most common sources of variability in the acoustic environments are (i) inter-speaker variability - due to vocal tract length and characteristic variations, (ii) intra-speaker variability - a speaker cannot repeat the same speech exactly in the same way, and (iii) environmental variability, known as the extrinsic problem, such as (a) the acoustic environment - like background speech, music, street noise, car noise, room reverberation, additive noise, and (b) the communication channel - such as transducers, speech coders, convolutional distortion, non-linear channel effects, fading echo cancelation etc. In real-world acoustic environments, many of these variabilities overlap each other and human speech perception mechanisms can deal with these complex variabilities successfully. However, in ASR, it is assumed that these variabilities are mutually exclusive and distinct and do not overlap each other in order to reduce acoustic model complexity [12].

In human speech perception studies, it was found that human beings use multiple cues/traits for speech perception under noisy conditions to predict the speech signal from the source. In adverse conditions, people track surrounding environments, detect any abrupt changes in the backgrounds, extract information of noise-like speaker traits, specific acoustic environment conditions and channel conditions, and analyze them [13]. Human speech perception mechanisms use the extracted noise information to adapt to the changing environments and then decode the speech signal, try to understand it, i.e., make some hypothesis on the output score, called a confidence score

based on some measurement schemes, and then send a feedback signal to the speech perception mechanism if the decoded speech is not intelligible [10]. From observations, it was found that adaptation and feedback are continuous processes until a speech signal is not understandable to listeners. People even try to use body gestures and the context of communications if it is known *a priori*.

When we consider human-computer communications, like ASR and machine dialog systems, it is essential to monitor the audio streams of source speakers, background acoustic environments, and channel changes since they represent significant challenges in maintaining the performance of the ASR system. Nevertheless, it is hard to design such a human-like environment-aware intelligent speech recognizer that explores the nature of noise [12]. Hardly few algorithms in the current literature have been shown to monitor and track acoustic environments properly and analyze their noises on-line, so as to adapt the acoustic models of an ASR system to its changing conditions.

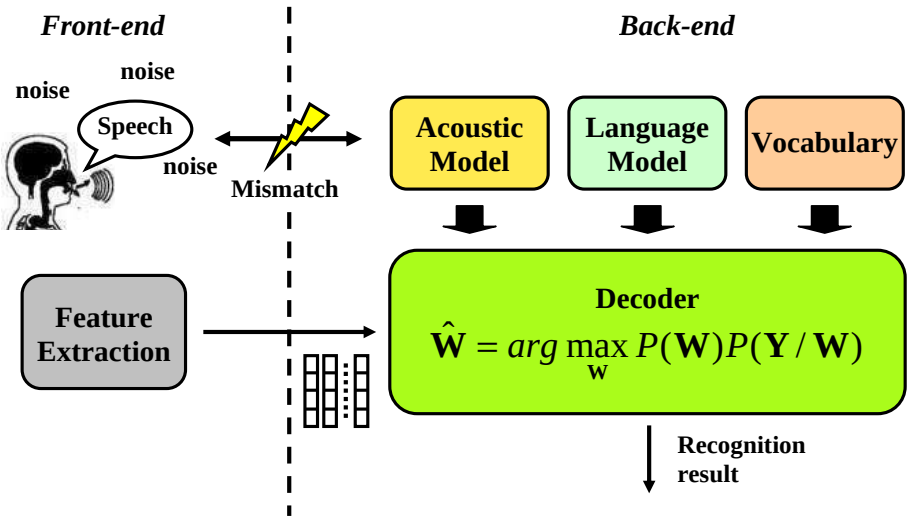


FIGURE 1.1 – HMM-based automatic speech recognition (ASR).

Current classical HMM-based statistical approaches as shown in Fig. 1.1 face difficulty to deal with the performance degradation of the ASR systems when the signal-to-noise ratio (SNR) decreases. The proper estimation of HMM parameters including transition probabilities, Markov state output densities, and noise statistics fails in adverse acoustic conditions. Another problem of HMM-based ASR is that it uses off-line HMM signal processing schemes to estimate the HMM parameters. HMMs use an off-line expectation-maximization (EM) Baum-Welch algorithm, which uses the fixed-interval *forward-backward* scheme to estimate the best possible state sequences of words corresponding to the unknown test utterances [7], [9], [14], [15]. The estimation of HMM model parameters increases computational cost and complexity, and the optimization process converges slowly. Off-line HMM schemes are not able to estimate the model parameters that undergo infrequent rapid changes as a result of adverse noisy environments.

On-line signal processing techniques have been reported in the literature for HMMs to deal with dynamic environments where the statistics of the observed data are changing with time. These on-line techniques are based on sequential expectation-maximization (EM) technique [16], [17]. Though these algorithms have shown success in dealing with context-dependent dynamic environments, they need more improvements to work in real-world acoustic conditions. For dynamically tracking the acoustic environment, updating the noise information, and adapting to the new environment, an alternative approach like the perception mechanism of human beings needs to be developed. The use of on-line frame-dynamic noise tracking and compensation schemes is likely to be one of the possible solutions.

Recently, the successful deployment of 3G/4G broadband multimedia wireless

communications put demands on environment-aware ASR systems for much voice-based activities including voice based web-browsing, music searching, phone dialing, E-mail and document dictation etc. Since mobile phones work in very uncertain acoustic conditions, which are commercially called 'impulsive environments', a conventional HMM based off-line ASR system is not able to deliver good services to the customers. In order to improve the robustness of an ASR system under impulsive acoustic environments, it is essential to develop new cutting-edge techniques that can make the ASR system aware of the background acoustic conditions and quickly adapt to new environmental changes in a real-time manner [18].

Many innovative algorithms have been developed with the advancement of research in the fields of artificial machine intelligence, pattern recognition, information theory, and as well as speech signal processing and recognition. These algorithms have found successful commercial applications as well. In the speech processing field, different innovative algorithms are available that can estimate noises very efficiently even under a very low SNR [19].

In the statistical signal processing field, the sequential prediction and updates of a signal in non-stationary noises are widely used for the estimation of on-line model parameters in real-time time series, such as the stock market, drilling oil rigs, and finance [20]. In optimization fields, stochastic search algorithms are widely used for efficiently searching the optimum model parameters in a high–dimensional complex search space. Currently, evolutionary stochastic particle swarm optimization (PSO) [21], [22] algorithms are becoming popular to solve the optimization problem of certain objective functions in real-life problems.

In the pattern classification field, Bayesian on-line inference for segmentation and

clustering has attracted more attention [20], [23]. It can be successfully used to detect sudden changes in speakers, environmental conditions, and channel conditions. For the speech processing field, statistical methods for noise tracking and estimation, e.g., minima controlled recursive averaging (MCRA) [4], can be used with the Bayesian on-line inference technique to track and detect slow or fast changes in highly non-stationary acoustic conditions. This approach can be used to design an environment-aware on-line ASR.

Speech is a very complex phenomenon involving biological information processing systems that enables humans to accomplish very sophisticated communication tasks. These tasks use both logical and intuitive processing. Conventional 'hard computing' approaches have achieved prodigious progress, but their capabilities are still far behind that of human beings, particularly when called upon to cope with unexpected changes encountered in the real world [24].

With the advent of advanced technologies for processing statistical and cognitive information, a new computing technology, called soft computing [25], [26], based on sound biological understanding has been evolving. It is a new paradigm of the computational intelligence and the role model for soft computing is the human mind.

According to the definition given by Professor L. Zadeh "The guiding principle of soft computing is to exploit the tolerance for imprecision, uncertainty, partial truth, and approximation to achieve tractability, robustness and low solution cost and better rapport with reality" [26], [27] [28]. At present the main constituents of soft computing are: i) Fuzzy Logic, ii) Neural Computing, iii) Evolutionary Computation, iv) Particle Swarm Optimization, v) Machine Learning, and vi) Probabilistic Reasoning (PR), with the latter subsuming belief networks, chaos theory and parts of learning theory

[29], [30], [24].

The acoustic modeling technique of current ASR relies on an off-line learning strategy. In this off-line learning approach, the statistical HMM models learn from historical data. This modeling technique is appropriate if the underlying dynamics of the test acoustic environments do not change over time. In reality, it is not the case as the characteristics of the acoustic conditions vary and evolve over time. In this case, the off-line learning approach fails to take into account these variabilities unless the model is re-learned or compensated. With the evolving of the soft computing-based intelligent computing techniques, it is possible to incorporate a current ASR system with sequential or on-line learning attributes [31].

In this dissertation, we explore a soft computing technique for a system architecture of environment-aware on-line ASR based on statistical and probabilistic reasoning technologies. We develop a novel soft computing model giving focus to very specific cases, such as rapidly changing non-stationary background noises. Usages of environment tracking techniques to detect slowly or suddenly varying background noises and extract noise information are proposed in this dissertation work. Innovative ideas based on Bayesian on-line inference techniques and evolutionary stochastic PSO techniques are investigated here for simultaneous recognition and acoustic model compensation in order to adapt dynamically to new rapidly varying acoustic noises. A new framework based on the integration of dynamic noise tracking algorithms and simultaneous feature compensation using the soft computing approach into an HMM-based system is proposed. This approach leads to the development of an on-line ASR to be more noise robust in real-world acoustic environments.

1.2 Objectives

The main objective of this dissertation is to investigate the robustness problems of current HMM-based ASR in real-life non-stationary acoustic environments, and to develop solutions to mitigate these deficiencies. In particular, this dissertation intends to develop a soft computing model to improve the robustness of current ASR in unknown acoustic environments. In the course of this thesis, we pursue the following goals:

- **Primary Objective:** Motivated by the fact that current HMM-based off-line ASR systems are all vulnerable in unknown test conditions, this dissertation focuses on developing a new soft computing technique using Bayesian on-line inference to deal with dynamic environments in previously unseen conditions.
- **Secondary Objective:** An integration of soft knowledge into ASR by the better exploitation of our knowledge of human speech production and perception mechanisms to the development of noise robust on-line ASR. This facilitates designing an environment-aware ASR system with larger and more consistent reduction in word error rates (WER) at reasonable computational cost over a wider range of corrupting highly non-stationary noises.
- **Tertiary Objective:** This work develops a cutting edge algorithm based on a newly evolving PSO soft computing technique to improve the noise robustness of ASR. Toward this goal, we propose a soft adaptive filtering technique to track highly non-stationary acoustic distortions in feature space for front-end processing of current HMM-based ASR.

1.3 Contributions

The following conference papers, journal article publications, posters and book chapters are significant contributions of the current dissertation towards the advancement of soft knowledge for on-line noise robust ASR. These works will lead to develop new applications for mobile multimedia devices.

1. Md Foezur Rahman Chowdhury, Sid-Ahmed Selouani, and Douglas O'Shaughnessy. *A Soft Joint Additive and Channel Distortions Computing Approach to Improve The Robustness of On-Line ASR in Non-Stationary Environments.* n Proc. 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'12), Montreal, Quebec, Canada, April 29-May 2, 2012.

2. Md Foezur Rahman Chowdhury, and Sid-Ahmed Selouani. *Identification et vérification du locuteur distribuées dans les communications mobiles.* In book chapter 7 to the book title: Traitement du signal et de l'image en biométrie, Hermes (Europe) editions, 2012.

3. Md Foezur Rahman Chowdhury, Sid-Ahmed Selouani, and Douglas O'Shaughnessy. *Bayesian On-Line Spectral Change Point Detection: A Soft Computing Approach for On-Line ASR.* In International Journal of Speech Technology, Springer, vol. 14, Online First (http://www.springerlink.com/content/1381-2416), 11 October 2011.

4. Md Foezur Rahman Chowdhury, Sid-Ahmed Selouani, and Douglas O'Shaughnessy. *A Highly Non-Stationary Noise Tracking and Compensation Algorithm, with Applications to Speech Enhancement and On-Line ASR.* In Proc. IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'12), March 25 - 30, 2012, Kyoto, Japan.

5. Md Foezur Rahman Chowdhury, Sid-Ahmed Selouani, and Douglas O'Shaughnessy. *A Rapid Adaptation Algorithm for Tracking Highly Non-Stationary Noises Based on Bayesian Inference for On-Line Spectral Change Point Detection.* In Proc. INTERSPEECH 2011, pp. 1205-1208, August 28-31, Florence, Italy.

6. Md Foezur Rahman Chowdhury, Sid-Ahmed Selouani, and Douglas O'Shaughnessy. *Real-Time Bayesian Inference: A Soft Computing Approach to Environmental Learning for On-Line Robust Automatic Speech Recognition.* In Proc. of 6th International Conference on Soft Computing Models in Industrial and Environmental Applications SOCO 2011 (vol. 87/2011, Advances in Intelligent and Soft Computing: pp. 445-452), Salamanca, Spain, April 6-8, 2011.

7. Md Foezur Rahman Chowdhury, Sid-Ahmed Selouani, and Douglas O'Shaughnessy. *Text-independent distributed speaker identification and verification using GMM-UBM*

speaker models for mobile communications. In Proc. 10th International Conference on Information Sciences Signal Processing and their Applications (ISSPA'10), Kuala Lumpur, Malaysia, pp. 57-60, May, 2010.

8. Md Foezur Rahman Chowdhury, Sid-Ahmed Selouani, and Douglas O'Shaughnessy. *Frame recursive dynamic mean bias removal technique for robust environment-aware speech recognition in real world applications.* In Proc. 23rd IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'10), Calgary, Alberta, Canada, pp. 1-5, May, 2010.

9. Md Foezur Rahman Chowdhury, Sid-Ahmed Selouani, and Douglas O'Shaughnessy. *A Study on Bias-Based Speech Signal Conditioning Techniques for Improving The Robustness of Automatic Speech Recognition.* In Proc. 22nd IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'09), St John's, Newfoundland and Labrador, Canada, pp. 664-669, May 3-6, 2009.

10. Md Foezur Rahman Chowdhury, Sid-Ahmed Selouani, and Douglas O'Shaughnessy. *Distributed automatic text-independent speaker identification using GMM-UBM speaker models.* In Proc. 22nd IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'09), St John's, Newfoundland and Labrador, Canada, pp. 372 - 375, May 3-6, 2009.

11. Md Foezur Rahman Chowdhury, Sid-Ahmed Selouani, and Douglas O'Shaughnessy. *Towards Human-Like Environment-Aware Automatic Speech Recognition.* In Poster SYTACom Research Workshop, École de Technologie Supérieure, Montreal, Quebec, Canada, Tuesday, April 28, 2009.

Moreover, the following conference paper has accepted for conference presentation and this will further enhance significantly the contributions of the current dissertation towards the advancement of soft knowledge for on-line noise robust ASR.

1. Md Foezur Rahman Chowdhury, Sid-Ahmed Selouani, and Douglas O'Shaughnessy. *A Soft Computing Approach to Improve The Robustness of On-Line ASR in Previously Unseen Highly Non-Stationary Acoustic Environments.* Accepted for presentation to the 11th International Conference on Information Sciences Signal Processing and their Applications (ISSPA'12), Montreal, Quebec, Canada, July 3 -5, 2012.

Finally, we summarize the significant contributions of the current dissertation work to the advancement of soft-computing knowledge in the field of on-line noise robust ASR as follows:

- **Contribution 1:** We design and implement a soft computing model for noise robust on-line ASR, with feature extraction, on-line tracking and detection of abrupt changes in acoustic environments, frame dynamic joint additive and channel distortion compensation (JAC), and speech recognition functionalities in a multi-threaded computing environment.
- **Contribution 2:** We propose a soft computing model using Bayesian on-line inference for tracking abrupt spectral change detection and rapid adaptation in highly non-stationary acoustic environments. We develop a frame-based Bayesian on-line spectral change point detection (BOSCPD) model for rapid adaptation of MCRA to non-stationary acoustic conditions. We show through experiments that the proposed BOSCPD can reduce significantly the delay in updating the minima search window in MCRA-based algorithms in the worst scenarios when the noise floor changes rapidly from low to high.

  We implement the proposed BOSCPD-based soft tracking model for on-line ASR using the Aurora 2 speech database [1]. A frame dynamic channel distortion compensation is implemented in a two-pass recognition process using the real-time computing simulation tool ATK [2]. The experimental results show significant improvement in word accuracy compared to the baseline MCRA technique.
- **Contribution 3:** We also propose a particle swarm optimization (PSO)-based soft adaptive filter for frame-dynamic non-stationary noise tracking and compensation. PSO is an emerging kind of filtering technique based on bird flocking and fish schooling phenomenon, which is quite extensively used for an alternative and efficient form of genetic algorithms and gradient-based search techniques.

PSO techniques are suitable for modeling non-Gaussian and non-linear systems. It is shown through our experiments that PSO-based soft adaptive filtering is a candidate technique for front-end processing of on-line ASR in unknown and highly non-stationary acoustic conditions.

1.4 Organization of Thesis

This dissertation has been organized into six chapters as follows:

- In Chapter 1, we discuss the shortcomings of the current HMM-based state-of-the-art ASR and the motivation of the dissertation work. Then we give our dissertation objectives, summarize the contributions of our work and finally the outline of our dissertation.
- In Chapter 2, we present issues regarding noise robustness of current ASR, followed by the basic model of a speech communication system, state-of-the-art signal processing techniques for ASR, and joint additive and channel distortion (JAC) compensation techniques to achieve robust performance of ASR in noisy conditions. We also discuss the MCRA-based classical speech tracking algorithms in highly non-stationary acoustic environments.
- In Chapter 3, we describe the main idea of Bayesian on-line inference for change point detection (BOCPD) technique, followed by the proposed mathematical model, called Bayesian on-line spectral change point detection (BOSCPD), for noise tracking and rapid adaptation in highly non-stationary acoustic environments. We also discuss the soft frameworks for on-line ASR in real-time applications.

- In Chapter 4, we introduce the biological population-based stochastic search algorithm, called particle swarm optimization (PSO), for adaptive front-end processing of the ASR in rapidly changing acoustic environments for real-world applications. We implement a dynamic multi-swarm particle swarm optimization technique (DMS-PSO) technique for soft on-line ASR.
- In Chapter 5, we present our experimental setups and results of the proposed soft computing model for on-line speech recognition based on the algorithms introduced in the previous chapters. We also provide a description of the speech database that has been used to evaluate the performance of the proposed soft computing technique-based on-line ASR.
- Finally, Chapter 6 contains conclusions as well as suggestions for future work.

Chapter 2

Review of Noise Robustness in Automatic Speech Recognition

2.1 Introduction

Most speech recognition systems deal with audio streams that typically contain a single speaker over a single channel under constrained acoustic environmental conditions. In [10], Pols found from his investigation on flexible human speech recognition that ASR systems are trying to imitate underlying human speech recognition and understanding techniques under controlled and consistent environmental conditions. However, human beings have developed their own language understanding techniques from context independent and uncontrolled acoustic environments. They can recognize and understand speech under extreme noisy conditions, and have a very high degree of robustness to different variabilities of acoustic environments.

Currently, in ASR, we are trying to gain performance to equal or perhaps even

outrank human performance. Nevertheless, current ASR technologies are not designed to emphasize the flexibility, robustness, and efficiency of human performance. ASR follows a shorter way of mimicking human language understanding techniques to serve commercial needs. A real design of ASR to behave like human beings would be very complex. At present we do not have enough information on how human beings decode speech in the brain to build such a system, nor is such a thing on the immediate horizon.

In an earlier study in [6], Lippmann found that the performance of ASR was one or more orders of magnitude worse than human performance on similar tasks. However, there have been enormous advances in improving the robustness of ASR over the last several years, there is still a large gap between human and machine performance. In order to improve this inferior performance of machine recognition system, that is, robustness with acoustic noises and channel as well as speaker variabilities, it needs more research on improving low-level acoustic phonetic modeling, especially for modeling spontaneous speech recognition systems [6], [32].

Human listeners use various acoustic features and cues with a high degree of flexibility [33]. However, ASR is not that flexible. In addition, human listeners have amazing power to adapt quickly to new acoustic environments, like a variable speaking rate, telephone quality speech, or somebody having a cold, using pipe speech (e.g., speech in intra-ship communication), or having a heavy accent, or surrounding acoustic noises like car noise, street noise, babble noise, wide-band noise, narrow-band noise, non-stationary noise etc. This clearly indicates that human speech recognition and understanding capacities are highly adaptive as they can predict and track rapidly varying acoustic environments and adapt to current situations quickly.

For the last three decades speech researchers have developed many approaches to make ASR more human-like robust but with limited success [32], [13]. In the soft computing domain, researchers brand these approaches as "hard computing" techniques compared to the bio-inspired soft computing model [24]. Even though soft researchers are interested to categorize the conventional techniques as hard computing techniques, some of these systems are also capable of making soft decisions. These efforts to improve the robustness of ASR results can be classified into two categories - one is human speech recognition (HSR) and the other one is automatic speech recognition (ASR). While many of these algorithms work quite well for specific context dependent situations, in general, they do not perform well in previously unseen non-stationary acoustic environments.

In this chapter, we review the human speech perception and recognition mechanism (HSR) in section 2.2. This is followed by a review of the prominent hard noise robust techniques for ASR that have been developed in the past in section 2.3. An overview of the classical noise compensation techniques for ASR is presented in section 2.4. Section 2.5 briefly describes the basic architecture of an environment-aware ASR recently proposed in the ASR literature. Finally, we summarize our review results in section 2.6.

2.2 Human Speech Recognition

Human listeners can show awareness of different speech perception phenomena and their underlying mechanisms, as mentioned below, to achieve surprisingly flexible, robust, and efficient performance for speech recognition and understanding. Speech

scientists and engineers might see good opportunities to implement certain elements for improving the performance of their speech recognition systems [10] [34].

2.2.1 Background Noise and Room Reverberation

A speech recognizer trained in clean condition (SNR $>$ 30 dB) degrades in performance as the background signal-to-noise ratio (SNR) in testing environments decreases, and substantially at SNR of +10 dB or less. However, human listeners perform well even under such low SNR. In addition, aspects of human speech perception depend on the size of the vocabulary and the native languages of the speaker and of the listener. Experiments have shown that, at about -10 dB SNR, all speech becomes unintelligible even for very limited vocabularies, such as digits or spelling alphabets [10], [35].

Speech sounds, especially consonantal sounds, lose their intelligibility and create confusion under various conditions of noises, e.g., speech-like noise spectrum, pink noise (low-pass filtered), non-stationary noise (such as door slams, car passing, etc.) having SNR from +15 to -6 dB, and also under room reverberation [36], [10]. In order to improve the performance of ASR, we need to explore the relationship between the effects of noise, reverberation, and speech intelligibility.

2.2.2 Spectral Distortions

Current ASR is based on statistical pattern recognition techniques under controlled environments, where it is trained to learn phoneme, triphone, or word templates. Recognition is then performed by measuring the shortest distance or greatest simila-

rity between testing and reference templates. Before applying pattern-matching techniques, sometimes speaker adaptation is applied. On the other hand, human beings learn in diverse conditions and their templates seem to be much more flexible and adaptable. High-pitched small-headed youngsters seem to have little difficulty to understand their low-pitched big-headed parents. People can easily understand telephone quality speech (300-3400 Hz). Substantial variability in speaking rate does not seem to bother people a lot [37], [10].

Present state-of-the-art ASR is very sensitive to spectral distortions. Therefore, it is essential for ASR to be insensitive to spectral variations to achieve human-like noise robust performance.

2.2.3 Auditory Modeling

Neuro-mechanical signal processing in the peripheral auditory system is so complex that we may not understand it well enough to imitate that process in ASR front-end modeling, apart from its functionality [10]. However, current-state-of-the-art ASR systems use perceptual-based features such as MFCC and PLP [38], which perform well under clean conditions. Though these features are extracted from the speech signal based on critical band filtering, to follow human speech perception, they do not represent the optimal features to show robustness under adverse conditions that human beings demonstrate even in severely degraded environments. Optimal features need careful selection of the spectro-temporal characteristics of the speech signal to be robust to diverse environments.

2.2.4 Multiple Features

Human listeners use a flexible multi-feature approach for speech perception [39], [13]. On the other hand, current speech recognition systems are based on a pattern matching technique using fixed and limited perceptual features like MFCC or PLP, which is, in fact, the main limitation of ASR [39]. Human beings use multiple acoustic cues to recognize words in efficient and flexible ways [10]. However, implementation of such flexible acoustic cues is still very hard for ASR, as the decoding mechanism of these cues by a human brain seems extremely complex and mostly unknown, like a black box.

2.2.5 Adaptation and Speaker Normalization

Human beings can adapt to different speakers, speaking styles, speaking rates, and speaking under emotion or stress, almost instantly. However, current so-called adaptive ASR uses adaptive techniques that are far from human performance. This adaptive ASR needs chunks of speech to adapt [40]. If such ASR is not adapted, it performs poorer for new speakers and new acoustic environments.

Adaptation to a new environment, whether it is background noise, another speaker, or a different speaking style, should not require new training, except just a quick adaptation of all models. This idea needs optimal use of parameter dependency. Current researchers are giving more attention to this optimal use of parameter dependency. Speech researchers are also thinking of a tree-based multi-scale dependency model-based approach for improving the adaptation capacity of adaptive speech recognizers [41], [10].

2.2.6 Predictability

Human beings have amazing predictability power. They are better than an n-gram language model in predicting what might come next in the speech stream. From this viewpoint, we can say that the perplexity of language for human listeners is always much lower than for ASRs. Most recognizers use the left-to-right processing for decoding and prefer to parse a whole sentence. The performance of state-of-the-art ASR systems is largely hindered by out-of-vocabulary words, which unavoidably set the upper limit of the word error rate (WER) of the ASRs. On the contrary, human beings have little difficulty to understand, interpret, and remember unknown words or new word compounds [42], [10].

2.2.7 Co-Articulation and Reduction

In conventional ASR we pay much attention to the dynamic spectro-temporal events, i.e., formant transitions in a higher resolution of the speech spectrum, which, in turn, give us better understanding of the human sensitivity to vocalic transitions [43]. However, the Difference Limen (DL) in endpoint frequency for 40 ms tone glides is as low as 30 Hz; it is more than 200 Hz for vowel-consonant (VC)-like stimuli with a short (20 ms) formant transition. This may be another indication that high spectral resolution is not always required and that unique spectral targets are quite useless [10]. It was also found by observing the formant transitions that formant undershoot hardly occurs in fast rate speech compared to normal speech for comparable consonant-vowel-consonant (CVC)-segments [44].

In fast speech, a speaker adapts his speaking style, also called articulation speed,

so that the vowel target can be easily reached for a specific context [13]. Contextual and prosodic conditions cause a lot of variations in the vowel midpoint formant position reached, while higher speaking rate and shorter duration add very little to that variability. On the other hand, changing the speaking style from read to spontaneous speech causes vowel reduction, and more specifically, a centralization of mainly first formant F1 [45]. It would be preferable to model such systematic phenomena as specific knowledge in ASR, rather than treat these phenomena just as variability in training data.

In ASR, triphones or HMM states are used in modeling the contextual information to take into account the co-articulation effects [9], [15]. However, they could not distinguish n levels of vowel reduction due to change in speaking rate [46]. In [45], it was found that consonant reduction is also as important as vowel reduction. Several acoustic measurements such as segmental duration, spectral center of gravity, intervocalic sound energy difference, intervocalic F2 slope difference, and the amount of vowel reduction in the syllable kernel, were used in [45] to find consonant identification errors for vowel-consonant-vowel (VCV) syllables extracted from spontaneous vs. read speech for both stressed and unstressed syllables. All these acoustic measures appear to be indicators of both vowel and consonant reduction and are all correlated to changes in speaking style and syllable stress.

2.2.8 Pronunciation Variation

Speech recognizers use standard pronunciation rules from a word lexicon irrespective of the speaker and the context in which the words occur, which is a huge

oversimplification. The human lexicon certainly does not work like that. In [47], the authors use cues from vowel and consonant reduction, word boundary effects like deletion and stress clash, and allophone variation, for efficient modeling of word lexicon search. Therefore, it is clear that for improving the robustness of ASR, we have to search for new algorithms so that pronunciation can be better represented.

2.2.9 Speech Perception Models

Speech scientists and engineers have developed many speech perception and word recognition models based on the analysis of human psychoacoustic behaviors, such as motor theory [48], analysis-by-synthesis [49], quantal theory [50], logogen model [51], cohort model [52], lexical access from spectra (LAFS) [53], first order context sensitive coding (ERIS) [54], autonomous search [55], dual coding [56], interactive activation TRACE model [57], short-list [58], adaptive learning [59], etc.

Though HMM-based ASR dominates current ASR approaches, without incorporation of explicit adaptation mechanisms it performs poorly in noisy environments. In order to improve the performance of current speech recognizers, it is essential to include in the models more specific knowledge to be extracted easily from the speech streams. The knowledge may be obtained dynamically by tracking the surrounding environmental characteristics, speaker specific variabilities, e.g., speaking style, speaking rate, word stress, reduction, co-articulation etc., and by quickly adapting the model parameters. This dynamic modeling will certainly enhance the performance of current ASR systems [40], [60].

2.2.10 Multi-Modal Speech Recognition

Current speech recognizers are limited to acoustic modeling of speech signals. However, speech communication is not necessarily limited to the auditory mode only. In practice, human listeners use different modes sub-consciously like body gestures, facial expressions, lip reading, and eye blinks. Now audio-visual feature-based speech recognizers called multi-modal ASRs are becoming more popular as the performance of such ASR systems is being improved over conventional acoustic model-based ASR systems [61].

In this section, we present some of the important approaches for human speech recognition. From this review study, we find that human speech recognition (HSR) is far superior to speech recognition mechanisms developed by human. Speech researchers and engineers are trying to develop new cutting-edge technologies very similar to human understanding and recognition performance. This led to the development of robust automatic speech recognition. In the following section, we briefly present our review result on robust automatic speech recognition, which will help to get a closer look on the status of current ASR systems.

2.2.11 Discussion

In this section, we present some of the important aspects of the human speech perception and recognition mechanisms. The objective of this review is to get an insight into the HSR systems. Since the soft computing concept is derived from nature, it is essential to understand how human beings use their natural processing techniques for speech recognition and understanding. Soft computing researchers might be able to

develop new ideas by incorporating certain knowledge, e.g., speaker and environment adaptation, multi-modal recognition, auditory modeling, speech perception etc., of the HSR systems.

2.3 Robust Automatic Speech Recognition

The performance of ASR systems degrades quickly when there is a mismatch between testing and training environments [62]. There has been much interest to develop various techniques to improve the robustness of ASR systems under various adverse conditions. Many of these algorithms perform well for specific tasks or environments in general, but they are not generalized to all the situations responsible for acoustic degradation. At present, researchers are trying to develop generalized techniques so that ASR systems can dynamically track the surrounding acoustic conditions and quickly adapt to the new environments. However, before going to a detailed analysis of these techniques to improve the noise robustness of ASR, we need to know the fundamental building blocks of the speech communication model, which are discussed in the next subsection.

2.3.1 Speech Communication Model

Current state-of-the-art noise robust ASR is based on two models [9], [14]: (i) the acoustic model (AM), and (ii) the language model (LM), as shown in Fig. 2.1. The AM model of ASR is based on the human speech communication model [11], [63] as shown in Fig. 2.2. This initial model is used for non-linear channels and background models. Later it is simplified based on the assumptions of a linear channel and additive noise

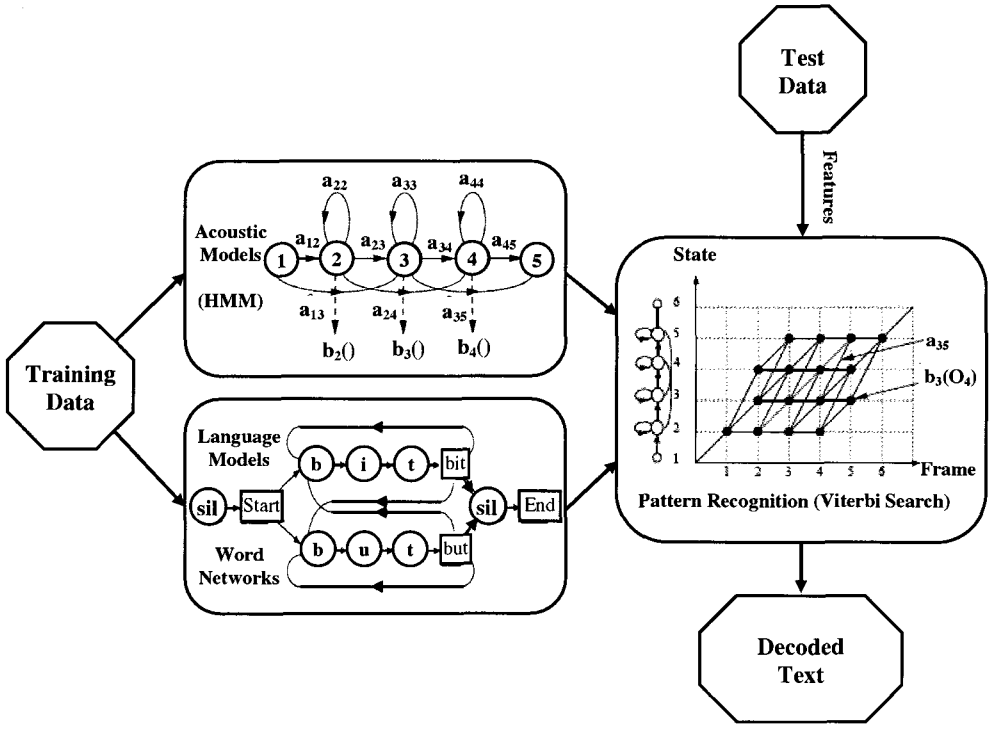


FIGURE 2.1 – Main building blocks of a state-of-the-art HMM-based (ASR).

degradation. A mathematical form of the speech communication model [11], which is formulated in detail as described in Appendix A, is

$$\begin{aligned} \mathbf{y} &= \mathbf{x} + \mathbf{b} + IDFT\{\ln(1 + e^{DFT[\mathbf{n}-\mathbf{b}-\mathbf{x}]})\} \\ &= \mathbf{x} + \mathbf{b} + \mathbf{r}(\mathbf{x}, \mathbf{n}, \mathbf{b}), \end{aligned} \tag{2.1}$$

where $\mathbf{y}$ is the observed noisy speech signal, $\mathbf{x}$ is the uncorrupted clean speech signal, $\mathbf{b}$ is the channel bias, $\mathbf{n}$ is the additive background noise, and $\mathbf{r}(\mathbf{x}, \mathbf{n}, \mathbf{b})$ is a correction vector in the cepstral domain [11].

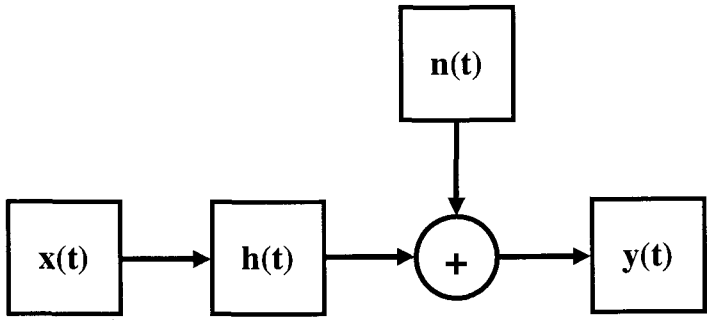


FIGURE 2.2 – Speech communication model.

2.3.2 Techniques to Improve the Robustness of ASR

Current techniques to improve the noise robustness of automatic speech recognizers can be categorized into the following two approaches [7]:

- model adaptation-based techniques, and
- feature compensation-based techniques.

The basic idea of model adaptation-based techniques is that the feature vectors are degraded due to external noises and attempt to handle this noise corruption by adapting the acoustic model of the speech signals, whereas the feature compensation techniques are based on the principle to make the feature vectors more insensitive to noise by compensating them without adapting the acoustic model.

Many successful techniques for both the approaches have been reported in the literature. A brief description of these techniques to improve the noise robustness of ASR is discussed next.

2.3.3 Model-Based Approaches

A model-based technique takes into account the effects of background additive noises in two ways. Firstly it adapts the statistical acoustic model to match the new acoustic environment by estimating the noise distribution or through the estimation of the perturbations in the speech distributions caused by the noise [9], [14]. Secondly, it helps the acoustic model to discard the unreliable part of the feature vector. However, model compensation-based techniques are computationally expensive. Some of these techniques need large chunks of transcribed data for the adaptation during the recognition process. Some of the most widely used model-based adaptation techniques are listed in the following subsections.

2.3.3.1 Multi-condition Training

Multi-condition training is a simple and direct model adaptation technique. It includes all the possible testing noise conditions in training the acoustic model by which the statistical model can take into account all the possible variabilities in the acoustic features due to background noise [64], [32]. Although this approach achieves a certain degree of robustness, it cannot always be applied in training the acoustic model depending upon the predictability of the range of noises that may be encountered. However, such a multi-condition training has the risk of making statistical models diffuse, i.e., models with a large variance. Therefore, there is no guarantee that a multi-condition model will produce good results in an specific acoustic condition. There is also a problem of generalization for such trained models to new testing conditions not seen in training [65].

2.3.3.2 Signal Decomposition

In this method, the signal is decomposed into different components and then a set of HMMs is used for each decomposed signal component for recognizing them simultaneously [66], [32]. The recognition is carried out by searching through the combined state space of the constituent models. A major disadvantage of the signal decomposition method is that it needs to train the acoustic model in a large number of noisy conditions, which is hardly possible in practice. Another disadvantage is that the computational cost increases exponentially as the number of signal components increases. Even for a single component noise, it needs a search through three-dimensional spaces, which is computationally expensive. Since it is not possible to know *a priori* the number of noise components present in a speech signal, it is difficult for a signal decomposition method to improve the noise robustness of ASR systems beyond a certain point.

2.3.3.3 Maximum Likelihood Linear Regression

The maximum likelihood linear regression (MLLR) model was developed basically for speaker recognition ; later it was used successfully for adaptation of acoustic models of speech signals under noisy conditions [67]. In this model, the means and variances of a Gaussian mixture model (GMM) are adapted to new noise conditions using data from the new environment. A linear transformation technique is used to transform the mean vectors and variances for model adaptation based on parameter estimation using a maximum likelihood method. However, the estimation of parameters requires a chunk of data of the new conditions for adaptation during the recognition, which is

considered as a major drawback of the MLLR method for previously unseen acoustic conditions. The transformation of parameters to adapt to new environments may not be linear. In such cases, a non-linear transformation is suitable for parameter estimation by a mixture of linear regression classes, which requires a large amount of adaptation data.

2.3.3.4 Maximum *a Posteriori* (MAP) Adaptation

Model adaptation can also be accomplished using the maximum *a posteriori* (MAP) approach. This adaptation process is sometimes referred to as Bayesian adaptation. The MAP estimation framework provides a way of incorporating prior information in the training. Hence, if we know what the parameters of the model are likely to be (before observing any adaptation data) using the prior knowledge, we might well be able to make good use of the limited adaptation data, to obtain a decent MAP estimate [15]. This type of prior is often termed as informative prior. Note that if the prior distribution indicates no preference as to what the model parameters are likely to be (a non-informative prior), then the MAP estimate obtained will be identical to that obtained using a maximum likelihood approach [68]. However, MAP adaptation adapts the mean and variance of each Gaussian model on a component-by-component basis, so more adaptation data is needed relative to MLLR.

2.3.3.5 Multi-Band Processing

In multi-band processing, also known as sub-band processing, the speech signal is passed through several narrow-band filters centered at different frequencies [32]. The output of each filter is processed separately to extract the feature vectors. These

feature vectors are processed with different HMMs for each sub-band to extract sub-band likelihoods or probabilities. These probabilities are combined together to get an effective likelihood that can be used for recognition. The major merit of this technique is that if a particular sub-band is corrupted by noise, it has less significant effect on the combined likelihood or probabilities, which in turn, contributes to improve the recognition performance. Therefore, the recognition performance depends on the reliability of each sub-band. This method is fruitful for speech contaminated by colored noise. Since each sub-band is processed independently, this method degrades the performance of speech recognition for clean speech, as it distorts the clean speech. To avoid this problem, some researchers [32] proposed a full-decomposition method, where all possible combinations of sub-bands including full-band, which does not treat different bands independently, are considered while computing the final likelihood to use in decoding.

2.3.3.6 Multi-Streaming Processing

In this approach, different feature vectors are extracted from the same speech signal. Different processing techniques used for extracting different features weight different aspects of the signal, which may be complementary in nature. Essentially, the ideas of this method came from observing how human beings use multiple acoustic cues to identify words in noisy environments for speech communications [69].

From investigation on how a human being performs robust speech recognition under severe adverse conditions, it was found that there are multiple representations of speech signals at different stages of human auditory processing and their integration to get robustness in noisy environments. Multi-stream feature-based techniques involve

different processing techniques to extract different feature vectors from the same speech stream, providing much complementary information useful for robust decoding of the speech signal under severe adverse conditions. Different feature streams are combined together based on the following two approaches:

1. Feature combination: In this approach, different feature vectors are concatenated together and then using different feature dimension reduction techniques like linear discriminant analysis (LDA) or Heteroscedastic linear discriminant analysis (HLDA) [19], the overall dimensions of combined feature vectors are reduced to a lower level in order to minimize the computational cost.
2. Posterior combination: In this method [69], probability outputs of the acoustic models of different streams are computed and then combined together in order to get the total probability to train the acoustic model.

2.3.3.7 Missing Data Approach

In this method [70], it is assumed that noise dominates some of the spectro-temporal regions in the speech spectrogram and these noise-corrupted regions are considered to be missing or unreliable data for improving the noise robustness of ASR. The missing data method uses the following two techniques to improve the robustness of ASR: (i) Marginalization technique, where the local emission probability is estimated as just the emission probability of the reliable part of the speech signal, and (ii) Data imputation, where values corresponding to the unreliable parts of the speech signal are estimated and then used to compute the local emission probability. However, the missing data method needs robust algorithms to identify the reliable

regions in the spectrum. Simple noise estimation techniques are used as a basis for the identification task.

2.3.3.8 Tandem Modeling

In the tandem approach a Multi-layer perceptron (MLP) classifier is first trained to estimate the context-independent phoneme posterior probabilities. The probability vectors are further processed to de-correlate and to optionally reduce the dimensionality and used as the acoustic features that model the output of conventional data-driven HMM and GMM models (HMM/GMM). A MLP nonlinearly transforms the input phoneme posterior probabilities data to a higher dimensional space defined by the output of hidden units and performs LDA analysis on the hidden unit outputs. The output of LDA is called Tandem features, which represent the *a posterior* probabilities of the phonemes with greater discrimination between phoneme classes. Tandem modeling outperforms conventional HMM systems under noisy conditions, but does not perform as well compared to HMM based systems for clean speech [71].

2.3.4 Vector Taylor Series

In the Vector Taylor series (VTS) technique, the speech model in Eq. 2.1 is expanded by its Vector Taylor series approximation. In this technique, the non-linearity in the speech model in Eq. 2.1 is approximated as a feature preprocessor with a Gaussian in the spectral domain. Several variations of VTS, e.g., 1st order VTS, 2nd order VTS, higher order VTS, truncated VTS etc., find successful application for model adaptation to improve the noise robustness of ASR in off-line mode [72], [73].

2.3.5 Feature-Based Approaches

The model-based adaptation techniques as we discuss in subsection 2.3.3 are computationally very complex and expensive [32], [13]. Feature based approaches are alternatives to model based adaptation techniques in order to minimize the computational complexities by generating acoustic features invariant to the noise. Such techniques use some knowledge from external sources about the effect of noise on the features and use auditory-like transformations of the features to remove the noise-prone aspects of the features. Different techniques are adopted to improve the noise robustness of speech recognizers based on a feature-based adaptation approach. These are described in the following sections.

2.3.5.1 Psychoacoustic and Neuro-Physical Knowledge

The most widely used speech feature vectors, MFCC and PLP coefficients, incorporate human auditory perception knowledge successfully into the feature extraction process. The performance of speech recognizers increases substantially by including some speech processing characteristics of the human auditory system. MFCC features are based on the Mel scale filter bank, also known as critical bands, which approximates the power law of hearing by mapping acoustical frequency to perceptual frequency approximately linearly up to 1 kHz and logarithmically at higher frequencies [13].

Both MFCC and PLP feature vectors have been shown to improve the robustness of ASR systems [9]. PLP is based on linear predictive (LP) analysis with a number of prior transformations, including critical band integration on the bark scale, equal

loudness pre-emphasis, and cubic root compression to account for the power law of hearing. The auditory spectrum obtained in this way is used to extract the LP coefficients, and then the equivalent PLP spectrum, and finally the PLP cepstrum. Both MFCC and PLP reduce undesirable variabilities as a result of the incorporation of various auditory-like transformations [32].

2.3.5.2 Spectral Subtraction

In spectral subtraction (SS), an estimate of the clean speech spectrum is obtained by subtracting the additive noise spectral estimation from the noisy speech spectrum. However, the success of this method depends on the reliable estimation of the noise power spectrum. In this method, the noise is estimated from the non-speech part of the speech signal using a voice activity detector (VAD). However, this technique suffers from a number of difficulties as mentioned below:

- When the speech to noise ratio is very low, i.e., when noise dominates the speech signal, it becomes a very difficult task to find a reliable estimate of the speech signal from this noisy signal.
- Furthermore, this technique is only suitable for stationary noises. For non-stationary noises, it is not possible even by using a very accurate VAD detector to accurately follow the noise spectral statistics, as they change quite rapidly. Therefore, it usually results in removal of a significant part of the speech information.
- The method performs worse if the subtraction of the noise power results in negative values when the estimated noise exceeds the actual noise magnitude. A threshold is used to partially solve this problem, resulting in a new residual

noise called musical noise.

Several new techniques have already been developed to solve this problem. One is to use non-linear spectral subtraction (NSS), which combines spectral subtraction with a noise masking technique. A more prominent technique, called the relative spectra (RASTA) technique, has been shown to be quite successful for improving the word error rate of speech recognizers. The underlying technique of RASTA is to suppress the noise components whose temporal properties are quite different from that of the speech in the spectral domain.

2.3.5.3 Wiener Filtering

In the signal processing domain, Wiener filtering is an optimal filtering technique in the mean square sense with some prior assumptions as follows:

- Both the speech signal and noise are statistically independent of each other
- Noise is assumed to be stationary or at least wide sense stationary

Wiener filtering needs *a priori* knowledge of a noise in its reference channel. However, in real-time acoustic environments the reference noise is not available. This limitation of a Wiener filter results in poor estimation of the noise spectral density in unknown test conditions. Despite this weakness, the Wiener filter is widely used as a speech enhancement technique for spectral subtraction based noise compensation in known test conditions to improve the noise robustness of ASR, e.g., the ETSI Front End 2.0 for distributed speech recognition [1], [3].

2.3.5.4 Noise Masking

Noise masking is a psychological phenomenon observed in humans. Acoustic stimuli lower than a certain threshold, fixed adaptively based on the noise level, cannot be perceived as a result of the masking effect. Based on our knowledge of perception, this involves reduction of the contribution of the lower energy regions of the spectrum during the recognition process. Employing this idea in the ASR system, a simple noise flooring and its extension in the HMM framework were shown to provide improved noise robustness. Noise masking in the logarithmic spectral domain and the cepstral domain have also been tried [13], [32].

2.3.5.5 Linear Discriminant Analysis

Reducing the feature dimension is a sensible approach towards improving the performance of a speech recognition system that uses auditory features. Dimensionality reduction is used in practical pattern classification applications where the ultimate objective is to design a system that classifies the vector of features in different classes by partitioning the feature space. In a typical classification problem the system designer chooses a number of features. The system designer believes that each of these features helps in some discrimination. Nevertheless, it is difficult to ensure that the information contained in each feature is extra to what is already available through the remaining features. If the parameters of the statistical model were known a priori, adding new features would not degrade the performance of a pattern recognition system. At most, if the new features do not contain any new information they will be ignored.

Feature dimension reduction not only improves the recognizer performance but can also speed up the pattern classification process. Fisher introduced a technique of dimension reduction to a one-dimensional linear subspace for the problem of two-class classification. Other researchers later extended this technique to handle multiple classes, known as linear discriminant analysis (LDA). LDA, also called Fisher discriminant analysis or multiple discriminant analysis in the literature, is a widely used technique for reducing the feature dimension. LDA and its generalization heteroscedastic linear discriminant analysis (HLDA) have wide applications to speech recognition [74].

2.3.5.6 Constrained Maximum Likelihood Linear Regression

Constrained maximum likelihood linear regression (CMLLR), also known as feature space MLLR (fMLLR), is a feature adaptation technique [75]. It estimates a set of linear transformations for the features. The effect of these transformations is to shift the feature vector in the initial system so that each state in the HMM system is more likely to generate the adaptation data [7]. For extremely resource constrained systems, the time required to perform the sufficient statistics accumulation for adaptation shows big challenge for this technique [76].

2.3.5.7 Histogram Normalization

Histogram Normalization, a non-linear non-parametric normalization technique, aims to match the cumulative density of the test features with the one collected during the training. This approach has successfully been used in both spectral and cepstral domains to deal with stationary noise [7].

2.3.5.8 Stochastic Matching Compensation

In the Stochastic Matching (SM) technique [77], [62], the compensation transformation is based on the mismatches between the training and the testing data and is done off-line. An affine transformation function is used for this transformation in the cepstral domain. The advantage of the SM technique is that it does not make any prior assumptions about the nature of the degradation [7]. It is an important criterion to compensate the distortions in unknown environments. A MLLR-based modeling approach for SM is reported in [77].

2.3.5.9 Sequential Compensation

In this technique, the compensation parameters are estimated in sequential manner. The authors in [78] used this technique to track the noise distortion of a signal and to adjust the compensation function using partial recognition state sequences within a small window. The authors used the Kullback-Leibler (KL) information measure as the optimization criterion.

In [79], the author proposed a sequential Stochastic Matching (SSM) compensation technique based on minimization of the recursive prediction error. It is an approximation of the technique developed in [78] and it is based on optimization of the recognition sequence. In supervised compensation, this recognition sequence is assumed to be known. For unsupervised compensation, the sequence is obtained from the first pass of ASR on the sentence [7].

Li Deng and et al. in [80] developed a sequential estimation of non-stationary noise within the speech feature enhancement framework of noise-normalized SPLICE

for robust speech recognition. In this technique, the authors used a 1^{st} order vector Taylor series (VTS) decomposition technique of the non-linear model of the acoustic environment in the cepstral domain. The noise mean vectors were estimated frame-by-frame and variance was ignored in this method.

In [81], the authors developed a sequential noise estimation method in the non-stationary acoustic conditions within the framework of 1st order VTS for robust speech recognition. They used the sequential Expectation-Maximum (EM) technique to develop this noise compensation model. In this model, only the additive noise is considered and both the mean and variance of the noise were updated frame-by-frame. The effect of channel distortion was ignored.

The authors in [7] developed an on-line SM compensation technique that could perform the compensation in parallel with recognition. In this technique, the authors estimated the noise compensation parameters frame-by-frame using the forward-only probability during Baum-Welch forward-backward search in HMM-based ASR. In this technique, an environment change monitoring system was used to take into account the rapid changes in acoustic environment. This technique does not need any knowledge about the test environments. However, it only considered additive noise and ignored channel distortions.

2.3.6 Discussion

In section 2.2, we discuss how human beings can successfully recognize speech signals using their amazing speech perception and understanding mechanisms. How they decode speech signal in noisy conditions is still a hot research topic. How the

human brain works for robust HSR still remains unknown and mysterious to speech scientists and researchers. In section 2.3, we present the issues regarding the robustness of ASR developed by human beings. The robustness issue of ASR mainly works in two ways: model adaptation and feature extraction. These adaptation techniques can adapt the model parameters or compensate the feature parameters for the changes in test conditions. One important change that ASR sees happen is acoustic noise. In the following section, we discuss various mathematical techniques currently employed for acoustic noise compensation in ASR.

2.4 Acoustic Noise Compensation

Most of the techniques developed to improve the robustness of HMM-based ASR are based on the speech communication model of Eq. 2.1. This acoustic model contains highly non-linear and complex exponential functions. It can be further simplified using mathematical manipulation as described in Eq. A.13 and it can be written in the following form:

$$\begin{aligned}\mathbf{y}&=\mathbf{n}+IDFT\{\ln(1+e^{DFT[\mathbf{x}+\mathbf{b}-\mathbf{n}]})\\&=\mathbf{n}+\mathbf{s}(\mathbf{x},\mathbf{n},\mathbf{b}),\end{aligned} \tag{2.2}$$

where $\mathbf{n}$ is the noise, and $\mathbf{s}(\mathbf{x},\mathbf{n},\mathbf{b})$ is a correction vector. The phase information is omitted in Eq. 2.1 and Eq. 2.2 assuming that ASR performance does not depend on phase information. Mathematical formulation of this model is explained in Appendix A and details can be found in [11].

For high SNR, the channel distortion is the main dominant effect and Eq. 2.1 reduces to the following form:

$$\mathbf{y} \approx \mathbf{x} + \mathbf{b}. \tag{2.3}$$

Similarly, at very low SNR, the additive noise is a more dominant factor for distortion of speech and Eq. 2.2 reduces to the following form:

$$\mathbf{y} \approx \mathbf{n}. \tag{2.4}$$

In real-life environments, test acoustic environments are highly non-stationary and complex, and, in most cases, consist of overlapped different acoustic conditions. The distortion in Eq. 2.1 becomes a joint function of both the additive noise and channel distortions. Moreover, there is no *a priori* information about these acoustic conditions and the SNR changes very rapidly. However, the mathematical solution of this complex model for highly non-stationary acoustic environments is very complex and there is no closed-form solution.

Most of the current approaches developed to improve the robustness of ASR use simple assumptions that the acoustic environments are stationary, and the SNR is high. Some techniques consider the non-linearity to a certain degree based on a linear or piece-wise linear approximation of the non-linear function [7], [82], [11], [63]. In all cases, it is also assumed that information of test conditions is known *a priori*. These approaches used to solve the robustness problem of ASR can be classified basically into two categories:

1. additive noise removal techniques, and

2. channel bias removal techniques.

These techniques increase the robustness of ASR either in feature space or in a model domain. Their performance depends on the extent to which the model of degradation used in the compensation process accurately describes the true nature of the distortion to which a speech signal has been subjected. However, the computational complexities of these algorithms increase with the degree of accuracy of the approximation of the non-linearity in the distortions.

2.4.1 Additive Noise Compensation

For additive noise bias compensation techniques, it is assumed that speech and additive noise are uncorrelated and stationary. Under this assumption, the acoustic model in Eq. A.2 as described in Appendix A can be redefined as follows:

$$y(t) = x(t) + n(t), \tag{2.5}$$

where $y(t)$ is the observed noisy signal, $x(t)$ and $n(t)$ the uncorrelated speech and noise signals respectively. This noise changes the Gaussian power distribution of the original speech signal into a bimodal or non-Gaussian form. The typical effects of the additive noise over the clean speech distribution are given in detail in [83].
In the cepstral domain, Eq. 2.5 takes the form as shown below:

$$\mathbf{y} = \mathbf{x} + IDFT\{\ln(1 + e^{DFT[\mathbf{n}-\mathbf{x}]})\}. \tag{2.6}$$

Here it is clear that at high SNR, the complex term in Eq. 2.6 becomes negligible and the observed speech signal will be very close to the original speech signal as

$$\mathbf{y} \approx \mathbf{x}. \tag{2.7}$$

Based on this idea, several speech processing algorithms such as (*i*) spectral subtraction techniques, (*ii*) statistical model-based algorithms, (*iii*) subspace algorithms, and (*iv*) SNR-based polynomial regression techniques were developed in order to get an estimate of the speech signal based on a more accurate model of the additive noise. The target of these speech enhancement techniques is to increase the SNR to get the estimated speech signal as shown in Eq. 2.7. These are the simplest speech enhancement algorithms that mostly work in the DFT domain and are based on the basic principle that the noise is additive. It is also assumed that noise can be estimated during the speech absence period, i.e., the silences or pauses of the speech signal, and can be subtracted from the speech signal during speech present periods. However, the main disadvantage of these speech enhancement techniques is that they produce annoying speech distortions known as musical noise.

2.4.2 Channel Bias Compensation

Speech distortions for additive noise and channel effects are highly non-linear functions in both the log-spectral domain and cepstral domain. The modeling of these non-linear functions is very complex and computationally expensive. The techniques that have been developed to mitigate this non-linearity problem can be broadly categorized as follows:

1. bias removal techniques,
2. affine transformation,
3. linear regression modeling techniques, and
4. vector Taylor series expansion

These techniques are based on a linear or piece-wise linear approximation of the non-linear function. They are also based on the assumption that SNR is high. They can be used to increase the robustness of ASR either in feature space or in a model domain. The performance of these methods depends on the extent to which the model of degradation used in the compensation process accurately describes the true nature of the distortion to which a speech signal has been subjected. However, the computational complexities of these algorithms increase with the degree of accuracy of the approximation of the non-linearity in the distortions.

In this chapter, we briefly describe the bias removal techniques since the proposed soft computing for joint additive and channel distortions compensation (JAC) technique is based on these techniques. The acoustic model in Eq. 2.1 can be rewritten into the following simplified form in the cepstral domain:

$$\mathbf{y}_m \approx \mathbf{x}_m + \mathbf{b}_m. \tag{2.8}$$

where $\mathbf{x}_m$ is the cepstrum of the clean speech signal for the mth frame, and $\mathbf{b}_m$ is a non-stationary additive bias in the cepstral domain for the mth frame. This model is valid when there is little additive noise present, i.e., at very high SNR. At low SNR, from Eq. 2.2 it is found that $\mathbf{y}_m \approx \mathbf{n}_m$. At other SNRs, one must estimate two biases:

one for the channel distortion and the other for the additive noises.

Let $\mathbf{Y} = \{y_1, y_2, y_3, ..., y_D\}$ be the feature vector for each frame of the observed speech signal, and Λ_x be the training HMMs for the clean speech $\mathbf{X} = \{x_1, x_2, x_3, ..., x_D\}$ with probability distribution function (pdf) $\sim \mathcal{N}(x; \mu_x, \Sigma_x)$ as shown below:

$$p(x) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_x|}} e^{-\frac{1}{2}(x-\mu_x)' \Sigma_x^{-1} (x-\mu_x)}, \tag{2.9}$$

where μ_x is the mean and Σ_x is the covariance of the clean speech signal. D is the dimension of the feature vectors for each speech frame. Under these conditions, bias can be estimated and removed from the noisy speech signal using transformation techniques, as shown in Fig. 2.3. These transformations can be done either in a feature space or in a model domain as described in the following section.

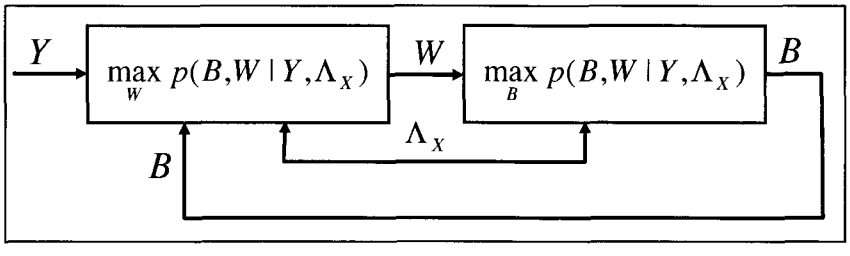


FIGURE 2.3 – Joint maximization of Eq. 2.8. Here $\mathbf{B}$ is the parameter of the transformation function, and is called a bias vector. $\mathbf{W}$ is the word or phone sequence to be decoded, and $\mathbf{\Lambda}_X$ is the acoustic model for the clean speech signal.

2.4.2.1 A Feature-Based Transformation

A feature-based approach uses some knowledge from external sources about the effect of noise on the features and uses auditory-like transformations of the features

to remove the noise-prone aspects of the features.

In the feature-based technique, a bias vector $\bar{\mathbf{b}} = \{b_1, b_2, ..., b_{M_f}\}$ is estimated and then the transformation function $F_\nu(.)$ in Fig. 2.4 with $\nu = \bar{\mathbf{b}}$ translates to removing an offset from the received signal as

$$\bar{\mathbf{x}}_m = F_\nu(\mathbf{y}_m) = \mathbf{y}_m - \bar{\mathbf{b}}_m, \tag{2.10}$$

where m $\{m = 1, ..., M_f\}$ is the index of the speech frames, ν is the parameter of the inverse function in feature space, and $\bar{\mathbf{b}}_m$ is the bias vector estimated from each speech frame.

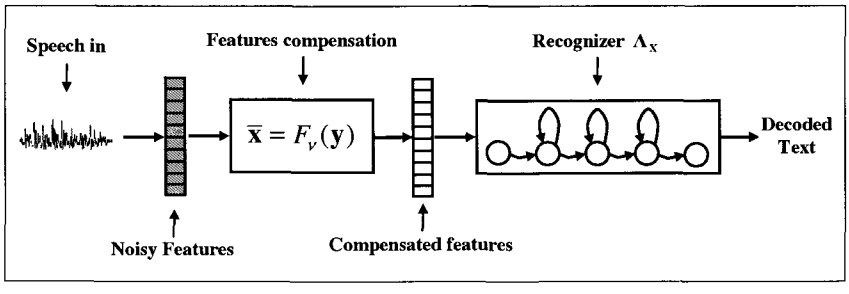


FIGURE 2.4 – Acoustic noise compensation in feature space. Here $\mathbf{\Lambda_X}$ is the HMM model for clean speech signal $\mathbf{x}$, F_ν is the feature transformation function and ν represents the bias $\bar{\mathbf{b}}$ estimate to be removed from the feature.

2.4.2.2 A Model-Based Transformation or Adaptation

This technique takes into account the effects of background additive noises in two ways. Firstly it adapts the statistical acoustic model to match the new acoustic environment by estimating the noise distribution or through the estimation of the

perturbations in the speech distributions caused by the noise [9], [14]. Secondly, it helps the acoustic model to discard the unreliable part of the feature vector. However, model compensation-based techniques are computationally expensive. Some of these techniques need lots of transcribed data for the adaptation during the recognition process.

In a model-based approach, the transformation function $M_{\bar{B}}(\Lambda_X)$ as shown in Fig. 2.5 translates to removing an offset from the model mean and variance as follows:

$$\begin{aligned} \bar{\mu}_Y &= \mu_X + \bar{\mathbf{B}}_\mu, \\ \bar{\mathbf{\Sigma}}_Y &= \mathbf{\Sigma}_X + \bar{\mathbf{B}}_\Sigma, \end{aligned} \tag{2.11}$$

where $\bar{\mathbf{B}}$ defines the set of biases that are shared across states, and model units. Thus, the problem of robust speech recognition is reduced to that of estimating a set of biases $\bar{\mathbf{B}}$.

The model-based adaptation technique to improve the robustness of speech recognizers is computationally very complex and expensive. The feature-based transformation technique is developed as an alternative to model-based adaptation techniques in order to minimize the computational complexities by generating acoustic features invariant to the noise.

When $\bar{\mathbf{B}}$ is reduced to a global (i.e., one vector for each utterance), then both $M_{\bar{\mathbf{B}}}(.)$ and it become equivalent [84]. However, several bias estimation techniques have been developed for speech recognition. The prior requirement for the bias removal is to get a good estimate of channel bias. The channel bias can be estimated by maximum

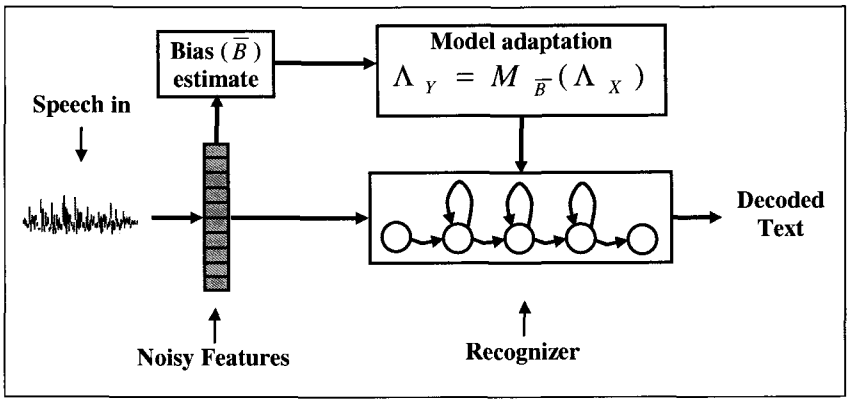


FIGURE 2.5 – Acoustic model adaptation in model domain. Here $\mathbf{\Lambda_Y}$ is the transformed HMM model for the observed noisy speech signal $\mathbf{y}$, and $\mathbf{\Lambda_X}$ is the HMM model for clean speech signal $\mathbf{x}$.

a posteriori (MAP) adaptation that maximizes the transformation function as

$$(\bar{\mathbf{B}}, \bar{W}) = \arg\max_{\mathbf{B},W} p(\mathbf{B}, W|Y, \mathbf{\Lambda}_X), \tag{2.12}$$

where W is the word (or phone) sequence. This MAP adaptation holds $\mathbf{B}$ fixed and solves for optimal $\bar{W}$, then holds $\bar{W}$ fixed and solves for the optimal $\bar{\mathbf{B}}$. In this case, finding $\bar{W}$ is the standard recognition problem while finding $\bar{\mathbf{B}}$ may be approached via the EM algorithm. Accordingly, an iterative two-pass recognition is necessary in which $\bar{W}$ is computed in the first pass, $\bar{\mathbf{B}}$ is estimated, and then a second pass recognition is performed to obtain an improved estimate of the word sequence given $\bar{\mathbf{B}}$ as shown in Fig. 2.3 [84].

2.4.3 Joint Additive and Channel Distortions Compensation

For real-world highly non-stationary environments the distortion as described in Eq. 2.1 becomes a joint function of both the additive background noise and channel distortions. Current state-of-the-art techniques to remove distortions either for additive noise or for channel distortions in feature space or the model domain fail to work for joint compensation of both the additive noise and channel distortions. Only a few algorithms have been developed to compensate the noisy speech signal jointly for both the additive noise and the channel distortions within certain context-dependent constraints. In practice for on-line ASR, it is desirable to treat both additive and convolutive bias simultaneously and jointly without any *a priori* information, while at the same time using only one set of clean speech models for real-world applications.

The concept of joint normalization of noise and filter effects was first introduced in [11] for current HMM-based ASR in batch mode. The author developed two algorithms: (*i*) a SNR-dependent cepstral normalization (SDCN) algorithm, and (*ii*) a codeword-dependent cepstral normalization (CDCN) algorithm. Both algorithms exhibit higher accuracy than the algorithms that perform independent compensation for noise and channel distortions. However, the success of these algorithms depends heavily on the *a priori* availability of the stereo database of the test environment to train the correction vectors. Since in real world situations such a stereo database is not available, the algorithms do not work for context-independent real-time environments.

In his Ph.D. work [63], Gales developed a model-based joint compensation technique called parallel model combination (PMC) as shown in Fig. 2.6. The PMC technique adapts the speech model to the noisy environment in two steps. First, it esti-

mates the noise statistics during a pause or silence of the speech signal and transforms the clean HMMs from the cepstral domain to the DFT domain to adapt them for additive noise. In a second step, it adapts the HMMs in the cepstral domain for channel distortions based on *a priori* information of the test conditions. However, PMC is based on some assumptions that additive noise and channel distortions are stationary and noise can be estimated during the silence or non-speech portion of the utterances, which is, however, not possible in real cases at low SNR. Therefore, PMC works well to improve the robustness of ASR only for context-dependent acoustic environments.

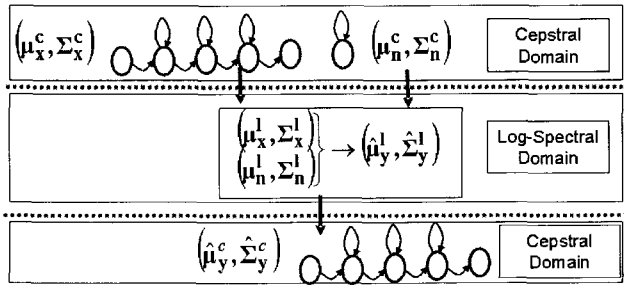


FIGURE 2.6 – Parallel model combination for JAC compensation. The suffix c indicates cepstral domain and l indicates log-spectral domain.

In [85], Afify developed a unified additive and channel bias compensation approach to improve the robustness of noisy *Lombard* speech based on expectation maximization (EM) algorithms in conjunction with PMC. In this joint bias compensation algorithm, linear spectral bias is estimated from the speech pauses and used to compensate the clean speech models. Then the additive bias-compensated speech signal is adapted in the cepstral domain. However, this algorithm also assumes that

acoustic environments are stationary, which is not true in practical cases. Moreover, at low SNR, it is difficult to detect the pauses or non-speech periods of the speech signal.

A unified approach to compensate for the additive noise and the *Lombard* effect in adverse environments was developed in [86]. This joint approach consists of (*i*) a spectral addition algorithm to compensate for the additive noise during the training phase, and (*ii*) an HMM state labeling algorithm to compensate for the *Lombard* effect. This approach demonstrated great improvement of ASR robustness in severe adverse environments. The main advantage of this algorithm is that the compensation for noise and the *Lombard* effects was made in the training phase. Therefore, it greatly reduced the computational complexity in the recognition phase. However, this algorithm is dependent on the *a priori* information of the acoustic environment, which is not available for real world highly non-stationary acoustic environments.

In [87], Li et al. developed a joint additive noise and channel distortion compensation (JAC) technique to compensate for both the additive noise and channel distortions in a unified approach. In this JAC approach, the authors used a first-order vector Taylor series approximation to adapt the HMM model parameters in a dynamic fashion. The adaptation process needs to estimate the model parameters for adaptation from the non-speech periods (beginning and end) of the speech signal. However, such a JAC approach is time consuming since it needs to adapt all the HMMs and is not suitable for real-time applications.

In [88], the authors introduced a joint scheme to compensate the channel and noise distortions in order to improve the robustness of ASR in adverse test conditions. In their work, the authors developed an adaptive Gaussian attenuation (AGA) algorithm

to compensate the noisy speech signal for additive noise over a wide range of noise conditions. A variance-scaling technique for cepstral mean normalization (CMNVS) was used for feature normalization for acoustic distortions. A joint process of these two algorithms improved the performance of ASR significantly. However, these algorithms were developed for stationary noise conditions with a mandatory requirement of 3 seconds of stationary noisy speech to obtain maximum compensation effects. However, this JAC in [88] might be a milestone for the JAC algorithm for compensating for highly non-stationary noise in real-time applications.

The authors in [89] proposed an algorithm for joint evaluation of multiple speech patterns. In this work, the authors exploited human speech perception to develop an iterative training of the HMMs using multiple speech patterns in bursty acoustic conditions, called virtual pattern evaluation, based on a hybrid approach of dynamic time warping (DTW) and the HMM framework. The proposed technique requires a repetition of exactly the same spoken word whose confidence score is low, which is very difficult to implement in real world applications like dialogue systems. A user cannot be expected to repeat a sentence in exactly the same manner as in previous utterances.

The authors in [84] proposed a codebook-based stochastic matching (CBSM) framework for integrated bias removal both at the feature level and at the model level. In this work, the authors integrated a hierarchical signal bias removal technique with their proposed CBSM algorithm and further extended it to account for n-best candidates for HMM-based ASR. The CBSM technique is based on a maximum likelihood (EM) technique. It was tested for the cellular telephone network and got significant improvement in recognition accuracy. However, CBSM is based on the assumption

that the utterance boundaries are known *a priori.*

In [7] an on-line frame-synchronous Stochastic Matching framework was implemented to compensate for abruptly varying noise. The basic idea of the proposed method is to perform the compensation and the recognition steps at the same time. The environment changes were identified using monitoring algorithms. The Stochastic Matching compensation method makes little *a priori* hypothesis on the acoustic conditions. However, it uses an affine compensation function for the test data, and its parameter needs to be computed off-line.

In all these JAC approaches, the authors used context based information to compensate for the background and channel distortions. For the JAC approach to work successfully on-line for practical applications, it should have the capability to simultaneously track and adapt to the surrounding unknown and non-stationary test environments. With the evolving of the soft computing-based intelligent computing techniques, it is possible to incorporate current ASR system with sequential or on-line feature compensation or model learning attributes [31]. However, the on-line learning of HMM models is computationally expensive. An on-line joint background additive and channel distortions compensation (JAC) technique [11], [90] in feature space is a convenient approach to avoid computationally expensive HMMs adaptation for rapidly varying test environments.

2.4.4 Simultaneous Noise Tracking and Estimation

Additive noise estimation algorithms are well known for their performance to estimate the noise spectrum. A proper estimate of the noise spectrum is crucial for

the quality of the enhanced speech signal. However, these speech enhancement algorithms suffer from several weaknesses, which make them inappropriate to estimate a noise spectrum under highly non-stationary acoustic environments. First, if the noise estimate is low, annoying musical noise will be audible, and if the noise estimate is too high, speech will be distorted, possibly resulting in loss of intelligibility. Secondly, these algorithms use a voice activity detector (VAD) to estimate and update the noise spectrum during non-speech periods of the observed noisy signal as the reference of the source of noise and use this noise estimated during speech presence periods. These approaches might work satisfactorily in stationary acoustic environments. However, VAD does not work at low SNR since, in this case, it is difficult to find speech/non-speech boundaries in noisy speech with very low SNR conditions [19].

In real-world acoustic environments, the spectral characteristics of the noise change vary rapidly and frequently. Therefore, it is difficult to get a proper estimate of the noise in a highly non-stationary environment by simply using the speech enhancement algorithms [19]. A more realistic approach is to estimate the noise spectrum continuously, even during speech activity. More precisely, the noise estimation algorithms might be able to track the noise spectrum continuously, compensate noisy speech frame-by-frame for estimated noise, and detect the abrupt changes in the noise spectrum. Recently, noise estimation and tracking have been getting more attention from researchers in the field of speech enhancement. These ideas are getting momentum for tracking suddenly changing non-stationary noises.

Several state-of-the-art noise estimation algorithms have been reported in the speech processing literature. These algorithms are known as single channel noise tracking algorithms, which estimate the noise spectrum continuously frame-by-frame

even during speech-present periods. These noise tracking algorithms can be classified into two main categories [19]:

- **Minimal-Tracking Algorithms**: Minimal-tracking algorithms are based on the assumption that the power of the noisy speech signal in individual frequency bands often decays to the power level of the noise, even during speech activity. Hence, by tracking the minimum of the noisy speech power in each frequency band, it is possible to get a rough estimate of the noise spectrum in that band. Two different approaches have evolved for tracking the minimum of noise level in each frequency band. The first algorithm is known as a minimum statistics algorithm that tracks the minimum of the noisy speech power spectrum within a finite window. The second algorithm tracks the minimum of the noisy speech spectrum continuously without requiring a window [91], [4], [92].
- **Time-Recursive Averaging Algorithms**: The time-recursive averaging algorithms are based on the fact that the noise spectrum does not affect the spectrum of a speech signal uniformly. Some regions of the speech spectrum are affected by noise more than others. Each spectral component will typically have a different effective SNR. Therefore, it is possible to estimate and update the noise spectrum of a particular frequency band that has extremely low SNR, leading to determining the probability of speech being present at a particular frequency band. These observations led to the development of a recursive-averaging of past noise estimates and the present noisy speech spectrum. The weights change adaptively depending either on the effective SNR of each frequency bin or on the speech-presence probability [19], [93], [92].

Several time-recursive averaging algorithms have been reported in the speech pro-

cessing literature, such as:

- SNR-dependent recursive averaging,
- weighted spectral averaging, and
- recursive averaging algorithms based on signal-presence probability.

The recursive averaging algorithms based on signal-presence probability have several forms of implementation such as:

- Minima-controlled recursive averaging (MCRA) [4],
- Improved minima-controlled recursive averaging (IMCRA) [92],
- MCRA-2 algorithm [94], and
- enhanced MCRA algorithm [95].

2.4.4.1 MCRA for Single Channel Non-Stationary Noise Tracking

Recently, minimum statistics-based single-channel noise-tracking algorithms, e.g., MCRA [4], have been getting the attention of speech researchers and engineers in tracking and estimating non-stationary noises. These algorithms assume that the power of the noisy speech signal in individual frequency bands often decays to the power level of the noise, even during speech activity. Hence, by tracking the minimum of the noisy speech power in each frequency band, it is possible to get a rough estimate of the noise spectrum in that band [4].

MCRA algorithms track the minimum of the noisy speech power spectrum within a finite search window. They do not need any voice activity detector (VAD) for pause or silence detection and can even track the noise during the active speech periods [19]. These features make the MCRA algorithm an ideal candidate for JAC-based on-line ASR for real-world applications.

In MCRA, once the noise estimate is obtained for each frame of the speech signal, a standard speech subtraction-based enhancement technique can be used to denoise the speech signal, as shown in Fig. 2.7. MCRA-based noise tracking algorithms do not need any prior noise information and VAD for noise psd estimates [19]. These features make MCRA-based noise tracking techniques suitable for on-line single channel JAC distortion compensation for real-time ASR.

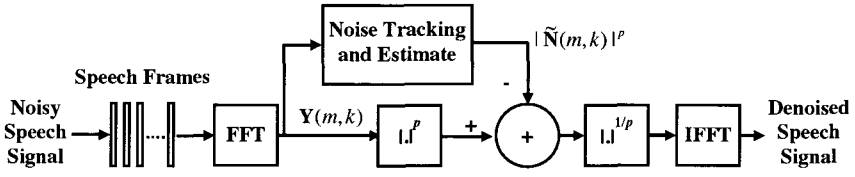


FIGURE 2.7 – Non-stationary noise tracking, estimation, and subtraction. For the MCRA noise tracking algorithm, the power suffix p is set to 2.

The mathematical formulation of the MCRA-based non-stationary noise tracking algorithms can be described briefly as follows:

The observed speech signal as mentioned in Eq. 2.5 is divided into overlapping frames by applying the Hamming window function $w(n)$ and then each windowed and overlapped frame is transformed in the frequency domain by computing the FFT as follows:

$$Y(m,k) = \sum_{\lambda=0}^{N_w-1} y(\lambda + mM)w(\lambda)e^{-j\frac{2\pi}{N_w}\lambda k}, \tag{2.13}$$

where $Y(m,k)$ is the STFT of the observed noisy speech signal $y(n)$, m is the time frame index, M is the frame update step in time, and $k\{k = 0, ..., N_w - 1\}$ is the

frequency bin index, and N_w is the size of the analysis window $w(n)$.

Since speech and noise are assumed to be uncorrelated, it is possible to estimate the noise power spectrum by tracking the minimum of the periodogram $P(m,k)$ of the noisy speech signal $Y(m,k)$ over a fixed window long enough to bridge the broadest peak in the speech signal [96], [94]. The periodogram $P(m,k)$ varies abruptly over time in rapidly changing acoustic environments. Under this condition, it is preferable to use a first-order recursive averaging of the periodogram $P(m,k)$ of $Y(m,k)$ [19] as follows:

$$P(m,k) = \alpha(m,k)P(m-1,k) + (1-\alpha(m,k))|Y(m,k)|^2, \tag{2.14}$$

where $\alpha(m,k)$ is a smoothing parameter. $\alpha(m,k)$ is time-frequency dependent to avoid over-smoothing problems. This smoothing factor is calculated based on the signal presence probability in each frequency bin separately. This probability can be calculated using the ratio of the noisy speech power spectrum $P(m,k)$ to its local minimum $P_{min}(m,k)$ calculated over a finite window [4], [19].

In MCRA, the noise power spectral density (psd) estimate based on the signal presence probability is derived using the following two hypotheses:

$$\begin{aligned} H_0(m,k) &: Y(m,k)=N(m,k), \\ H_1(m,k) &: Y(m,k)=X(m,k)+N(m,k), \end{aligned} \tag{2.15}$$

where $X(m,k)$ and $N(m,k)$ are the STFT of the clean speech and noise respectively, $H_0(m,k)$ and $H_1(m,k)$ represent the absence or presence of speech hypotheses res-

pectively. The noise variance for the kth band is defined as $\sigma_n^2(m,k) = E\left[|N(m,k)|^2\right]$.

The noise psd estimate is updated based on the following hypotheses:

$$H_0' : \hat{\sigma}_n^2(m,k) = \alpha_n \hat{\sigma}_n^2(m-1,k) + (1-\alpha_n)|Y(m,k)|^2,$$
$$H_1' : \hat{\sigma}_n^2(m,k) = \hat{\sigma}_n^2(m-1,k), \tag{2.16}$$

where α_n $(0 \leq \alpha_n \leq 1)$ is a smoothing parameter. $\sigma_n^2(m,k)$ denotes the variance of the noise in the kth frequency bin [4].

In Eq. 2.16, the noise estimate is updated whenever speech is absent, otherwise it is kept constant. The estimate of the noise PSD can be estimated in the mean-square sense as follows:

$$\hat{\sigma}_n^2(m,k) = \tilde{\alpha}_n(m,k)\hat{\sigma}_n^2(m-1,k) + [(1-\tilde{\alpha}_n(m,k))]|Y(m,k)|^2, \tag{2.17}$$

where $\tilde{\alpha}_n(m,k) = \alpha_n + (1-\alpha_n)p(m,k)$ is a time-varying smoothing parameter and it varies within the range $\alpha_n \leq \tilde{\alpha}_n(m,k) \leq 1$.

Accordingly, the noise spectrum can be estimated by averaging past spectral power values, using a smoothing parameter that is adjusted by the signal presence probability as follows:

$$\hat{p}(m,k) = \alpha_p \hat{p}(m-1,k) + (1-\alpha_p)I(m,k), \tag{2.18}$$

where α_p $(0 < \alpha_p < 1)$ is a smoothing parameter, and $I(m,k)$ is an indicator function for Eq. 2.18. $I(m,k) = 1$ if $S_r(m,k) > \delta$ and $I(m,k) = 0$ otherwise. Here $S_r(m,k) =$

$P(m,k)/P_{min}(m,k)$ is the ratio of the local energy of the noisy speech and its derived minimum over a search window of length L . δ is a threshold for speech presence determined as follows:

$$P(m,k) = \alpha_s P(m-1,k) + (1-\alpha_s)|Y(m,k)|^2, \tag{2.19}$$

where α_s $(0 < \alpha_s < 1)$ is a smoothing parameter. $P_{min}(m,k)$ is defined as follows:

$$P_{min}(m,k) = \min\{P(j,k)\};\ \text{for } m-2L < j < m, \tag{2.20}$$

which is calculated [95] as

$$P_{min}(m,k) = \begin{cases} P(0,k) & \text{if } m = 0, \\ \min\{P_{min}(m-1,k), P(m,k)\} & \text{if } m\%L \neq 0, \\ \min\{P_{tmp}(m-1,k), P(m,k)\} & \text{otherwise,} \end{cases} \tag{2.21}$$

$$P_{tmp}(m,k) = \begin{cases} P(0,k) & \text{if } m = 0, \\ \min\{P_{tmp}(m-1,k), P(m,k)\} & \text{if } m\%L \neq 0, \\ P(m,k) & \text{otherwise,} \end{cases} \tag{2.22}$$

where % sign is used to indicate modulus after division [95].

The parameter L determines the resolution of the local minima search. The local minimum is based on a window of at least L frames, but not more than $2L$ frames. The length of the window controls the bias upwards during continuous speech and the bias downwards when the noise level increases [96], [4].

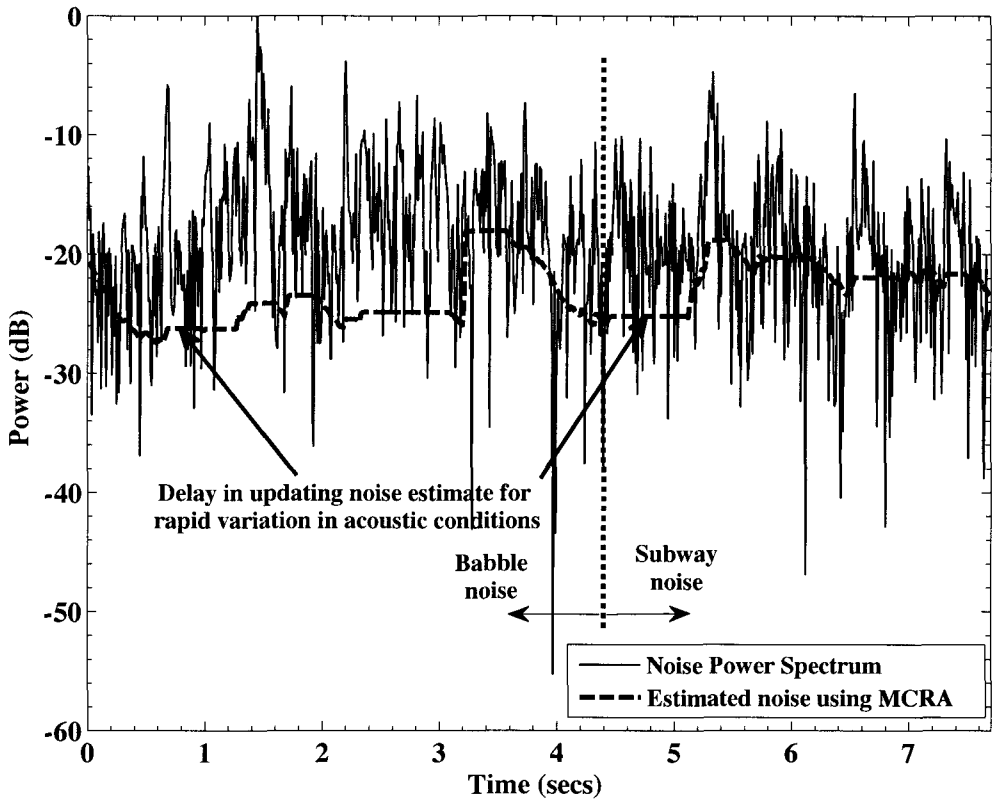


FIGURE 2.8 – The delay in MCRA to update the noise estimate in response to the rapid changes in acoustic environments at $f=$ 1500 Hz. The test speech signal was sampled at 8 kHz and it is degraded by babble noise and subway noise at 5 dB SNR. A frame duration of 25 millisecond (ms) with 60% overlapped is used in this test example.

MCRA-based algorithms cannot update the noise power spectral density (PSD) estimate right away for the drastic changes in spectral properties of the non-stationary noises as shown in Fig. 2.8. The estimated noise PSD lags behind the true noise PSD by two times the length L of the minimum search window in worst case scenarios, e.g., from high SNR to very low SNR conditions. It is a serious shortcoming of MCRA-based algorithms for tracking rapidly changing noises in real world acoustic regimes. Several derivatives of MCRA have been developed to reduce the delay in updating

the noise PSD after sudden changes in noise [92], [94], [95]. However, these algorithms fail to sufficiently reduce the adaptation delay.

2.4.4.2 Speech Enhancement

The noise over a frame duration can be assumed to be stationary or quasi-stationary. An estimate of the noise spectrum for each frame using MCRA noise tracking algorithms and its subtraction from the respective frame as shown in Fig. 2.7 will be an alternative implementation of the classical spectral subtraction algorithm formulated. The implementation of this spectral subtraction algorithm is formulated as follows:

$$|\hat{X}(m,k)|^2 = \begin{cases} |Y(m,k)|^2 - \alpha_{os}|\hat{N}(m,k)|^2, & \text{if} \quad |Y(m,k)|^2 > (\alpha_{os}+\beta_{sf})|\hat{N}(m,k)|^2 \\ \beta_{sf}|\hat{N}(n,k)|^2, & \text{otherwise,} \end{cases} , \tag{2.23}$$

where α_{os} is a oversubtraction factor ($\alpha_{os} \geq 1$), and $\beta_{sf}(0 < \beta \ll 1)$ is the spectral floor parameter.

This spectral subtraction algorithm is known as an oversubtraction algorithm [97]. The advantage of this algorithm is that it removes an overestimate of the noise power spectrum while preventing the resultant spectral components from going below a pre-set minimum value called the spectral floor. The two parameters α_{os} and β_{sf} control, with great flexibility, the overall performance of this oversubtraction algorithm. The spectral floor parameter β_{sf} controls the amount of the remaining residual noise and the amount of perceived musical noise. If β_{sf} is too large, residual noise will be audible

but the musical noise will not be perceivable. On the other hand, if β_{sf} is too small, the musical noise will be annoying but the residual noise will be greatly reduced. The oversubtraction factor α_{os} controls the speech spectral distortion in Eq. 2.23. For very large α_{os}, the resulting speech spectrum in Eq. 2.23 will be distorted to a great extent to the point where the intelligibility suffers greatly. For best noise elimination with the least amount of speech distortion, i.e., minimum musical noise, α_{os} should be small for speech frames with high SNR, and vice versa. Based on this observation, the authors in [97] made α_{os} a function of frame SNR as follows:

$$\alpha_{\varphi} = \alpha_0 - \tfrac{3}{20} SNR_m \qquad -5dB \leq SNR \leq 20\ dB\ , \tag{2.24}$$

where α_{φ} is the oversubtraction factor and it is a function of the SNR for each frame, α_0 is the desired value of α_{os} for the mth frame at 0 dB SNR, and SNR is the short-time SNR estimated for the mth frame.

In the noise tracking algorithm, it is not possible to get the clean speech signal for the measurement of the SNR. Therefore, an *a posteriori* estimate of the SNR is to be computed from the ratio of the noisy speech power to the estimated noise power. A plot of α as a function of *a posteriori* SNR is shown in Fig. 2.9.

2.4.5 Discussion

Though speech researchers have addressed the robustness problem and developed many cutting-edge algorithms, as we saw in sections 2.2 and 2.3 to improve the performance degradation of ASR, they are still far behind a human performance. These algorithms work well within controlled environments, but seldom work well

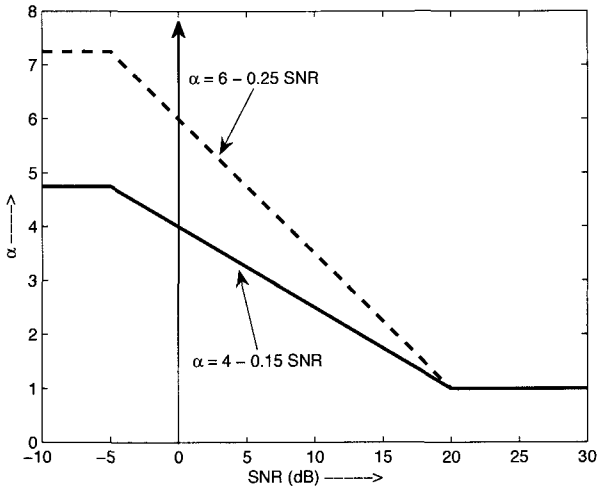


FIGURE 2.9 – Plot of the oversubtraction factor α_φ as a function of the SNR [19]. The factor α in this figure represents α_φ in Eq. 2.24.

in unknown environments. For example, in impulsive environments, like for portable devices such as cell phones, where usually no information exists about the occurring time, the level and the nature of the sudden noise available, these algorithms fail to work. In the following section, we briefly present our review results on new evolving techniques that eventually will lead to design self-adaptable ASR.

2.5 Environment-Aware ASR

Recently, speech researchers and engineers are getting more interested about how to address the issue of noise robust environment-aware ASR. Speech researchers are now focusing on a common framework to address the problems of environment awareness of the ASR. Some important approaches towards developing a general architecture of combating the robustness problems of ASR in adverse environmental

conditions have just started to appear in the ASR literature. However, research in this field is very new and little information is available in the literature. Some of the approaches are discussed below.

In a recent article [12], the authors proposed a general architecture called "environment sniffing" to detect, classify, and track acoustic environmental conditions, as shown in Fig. 2.10. Here the environmental conditions include speaker, noise, channel, and signals. The goal of their framework is to seek out detailed information about the environmental characteristics instead of just detecting the environmental changes. This is the first time that an idea of the architecture of environment sniffing systems was published in the literature. The authors claimed that this sniffing system could be used for many applications, such as ASR, speech coding, speaker ID, speech enhancement, language ID, noise transcription etc. The authors tested this architecture for ASR under a wide range of variations of car noise and found improved accuracy over conventional ASR systems.

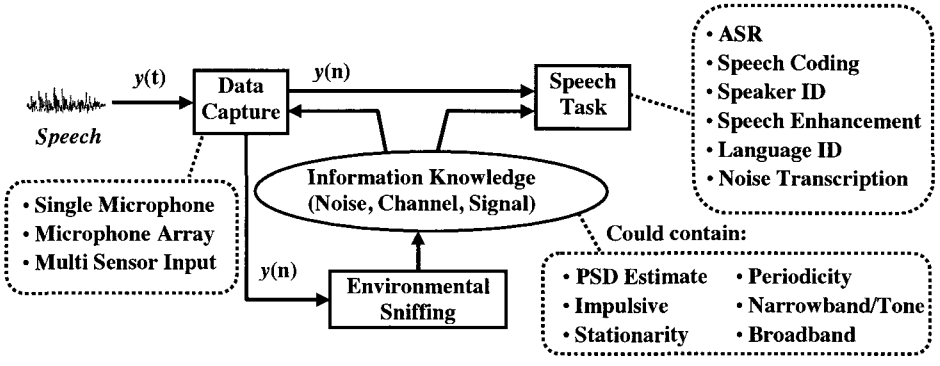


FIGURE 2.10 – Environment sniffing architecture in [12]. Here $y(t)$ is the noisy input analog speech signal, and $y(n)$ is the digitized noisy speech signal.

In [12], the authors focused on mainly three areas of current state-of-the-art ASR

to develop an environment sniffing system that can successfully monitor the surrounding acoustic environment and adapt the model or parameters: (i) Input sensor - single microphone, or multiple microphones, (ii) Extraction of acoustic environment information as a function of the input signal, and (iii) Adaptation of the acoustic model and/or usage of best features based on environment information. However, the main complexities of this sniffing system lie in the area of extracting environment information and model adaptation and/or best feature selection. Environment information may consist of noise information, speaker identity and speaking style, channel information etc. For model adaption, they suggested to use one of several adaptation schemes, for example, Jacobian adaptation [98], maximum likelihood linear regression (MLLR) [67], parallel model combination [90] etc.

The general framework of sniffing the environmental conditions and its successful application for in-vehicle dialog systems [12] created new changes in the direction of research in the field of environment-aware ASR systems. Nevertheless, while such an environment sniffing system could provide significant knowledge to help direct and improve the subsequent speech processing tasks and thereby increase robust speech system performance, it is very complicated and requires huge noise databases of long hours including all possible noise occurrences. It also works off-line and still poses some limitations to mimic human speech recognition characteristics under adverse environmental conditions. It does not include the simultaneous recognition and model adaptation process, a real-time process that human beings try to do in adverse acoustical conditions. Model adaptation techniques such as MLLR, PMC, Jacobian etc., all work off-line and they require a section of data of the new environment, which is difficult to obtain during an on-line model adaptation process.

2.6 Summary

In this chapter, we give a comprehensive review of the prominent noise robustness techniques of human speech recognition (HSR) and automatic speech recognition (ASR) systems. We also discuss the basic model of speech communication, which is used as the building block of the acoustic model of the current state of ASR. The cutting-edge technologies that have been developed to compensate for both additive and channel distortions in order to improve the robustness of speech recognition systems in noisy environments are also briefly discussed in this chapter.

In order to develop new cutting-edge technologies to solve the non-stationarity problem for ASR in real-world environments, the contribution of this chapter can be summarized as follows:

- facilitating the understanding of past and present trends of speech technology and underlying constraints in solving noise robustness of ASR;
- helping to understand the basic acoustic model of ASR and technologies to improve the noise robustness of ASR in noisy conditions;
- pointing to the comparative features of different methods, their merits and demerits to improve the noise robustness of speech recognition;
- finally, facilitating the understanding of the basic idea of MCRA-based single channel noise tracking and compensation techniques to improve the noise robustness of ASR in noisy conditions.

We present the contribution of this thesis in the next two chapters. The first chapter is devoted toward improving the ASR performance in on-line condition using a soft Bayesian on-line spectral inference technique in a non-stationary acoustic speech

environment. In the second chapter, we present the bio-inspired evolutionary PSO-based soft adaptive filtering technique to improve the robustness of on-line ASR in previously unseen non-stationary noises. This is followed by the chapter containing the experimental setup, results, and performance evaluation of the contributed algorithms.

Chapter 3

Proposition 1: Bayesian On-Line Spectral Inference - A Soft Computing Approach to Improve the Robustness of On-Line ASR

3.1 Introduction

We present the review results on cutting-edge techniques that are currently used for robust automatic speech recognition in the previous chapter (Chapter 2). Though these techniques work well to improve the noise robustness of ASR in a context dependent environment, they lag far behind the human performance for self-adaptability or environment awareness in unknown test conditions. Speech researchers and scientists are trying to develop new innovative techniques to add human like self-adaptability

features to current ASR. Currently, they are trying to learn from nature and to apply this knowledge to current ASR systems. This leads to the development of a new kind of approach called soft computing technique.

Bio-inspired soft computing (SC) is a set of methodologies that combines different well-known artificial intelligent methods that work synergistically and provides, in one form or another, flexible information processing capability for handling real life ambiguous situations [29]. The Bayesian belief network or inference (BI) technique is one of the constituent technologies of the SC methods. Bayesian inference provides probabilistic reasoning for a learning mechanism to update a system affected by randomness or probabilistic uncertainty.

In recent years SC methods for treating uncertainties and variabilities have reached the speech processing and speech recognition fields. Since human speech is a biological signal and soft computing techniques are generally inspired by biological processes, soft computing techniques are better suited for tackling many of the challenging problems of speech processing and speech recognition [24]. Soft computing techniques have the potential to extract information from time-varying complex acoustic environments and can be used in improving the noise robustness of current ASR in the real-life scenarios. Bayesian inference of the spectral variation over time could be used as a soft learning technique for ASR to compensate for non-stationary noises in feature space.

Motivated by the ability of the soft computing techniques over the conventional hard techniques to handle complex natural processes, we develop a novel soft model to track and compensate the previously unseen non-stationarity of the acoustic environments to improve the noise robustness of ASR in on-line mode. The proposed soft

model is based on Bayesian on-line belief or inference for simultaneous recognition and acoustic model compensation in feature space in order to adapt ASR dynamically to new rapidly varying acoustic conditions. This approach leads to the development of a new framework of on-line ASR to be noise robust in real-world acoustic environments.

In this chapter, the proposed framework of the soft computing model to improve the noise robustness of ASR in on-line condition is presented as follows. An overview of a soft computing model using Bayesian on-line belief based on the mathematical formulation of a Gaussian process in Appendix C is presented in section 3.2. Section 3.3 describes the proposed architecture of the Bayesian on-line spectral change point detection (BOSCPD) algorithm. The soft BOSCPD technique for tracking and compensating background additive non-stationary noises is described in section 3.4. We describe the soft JAC (SJAC) technique for on-line ASR in section 3.5. Section 3.6 briefly describes about the simulation setups. Finally, we summarize the chapter in section 3.7.

3.2 Soft-Computing: Bayesian Approach

Real world acoustic environments are very complex in nature and vary rapidly over time. For current ASR, off-line learning limits its ability to accurately capture the dynamics of acoustic environments. It can only model the events that were encountered during the learning process. Therefore, current ASR should be allowed to track the changes over time and to adapt to these previously unseen conditions. The MCRA algorithm is a good candidate to incorporate into the front-end of such a smart self-adaptable ASR. However, the main disadvantage of MCRA to perform this job

is that it fails to detect the abrupt changes when noise floor jumps from low to high values [19]. MCRA can detect changes with large delay. For real-time ASR, we need to detect the changes in environment with high precision, i.e., with minimum detection delay time. Under theses circumstances, an on-line Bayesian prediction model that learns the acoustic environments with high uncertainty and fuzziness could be used to update the ASR to new conditions with time [31].

Bayesian inference (BI)-based probabilistic modeling has long been used off-line (batch mode) for analyzing and tracking high non-stationarity and environmental change detection in systems [20]. With the advancement of computing power, the Bayesian inference-based soft computing modeling technique finds its application for tracking unknown non-stationary systems having a high degree of uncertainties.

3.2.1 Bayesian Off-Line Inference

Bayesian inference (BI) or belief has long been used for off-line (batch mode) change point detection in time series. The Bayesian change point detection (CPD) technique uses a change point model of the parameters and integrates out the uncertainty in the parameters rather than using a point estimate. Bayesian approaches to CPD have been retrospective, where the central aim is to infer change point locations in batch mode [20], [99], [23]. These methods work fine for off-line time-series data sets. They are not designed for on-line prediction systems that need to adapt predictions in light of incoming parameter changes.

Recently, an application of BI, called Bayesian on-line change point detection (BOCPD), in real-word time series data sets, e.g., finance, oil drilling, robotics, and

satellite tracking, has been reported in the literature [20], [23]. One appealing feature of BOCPD is that it allows one to express uncertainty about the number and location of change points. For a noisy speech signal, BOCPD can be used as a frame-based causal predictive filter, i.e., can generate an accurate predictive distribution of the next unseen spectral data of the speech frame, given only the spectral properties of the already observed speech frames.

The inability of MCRA-based noise tracking algorithms to react right away to abrupt changes in non-stationary noises has a detrimental impact on their performances. Bayesian on-line inference for change point detection (BOCPD) can be used to reduce this performance degradation by recognizing speech spectral properties' change events and adapting the MCRA model appropriately.

3.2.2 Bayesian On-Line Inference for CPD

The Bayesian on-line change point detection (BOCPD) algorithm mainly focuses on the time since the last change point, called the run length r . It uses an underlying predictive model (UPM) of the time series that changes at each change point. It also uses a hazard function $H_h(r|\theta_h)$ that describes how likely a change point is given the run length r. The UPM is used to compute the posterior predictive $p(x_t|x_{(t-\tau)}, \theta_\nu)$ for any $\tau \in [1, ..., (t-1)]$, at time t . The parameters $\theta = \{\theta_m, \theta_h\}$ form the set of hyper-parameters for the model, and are assumed to be fixed and known.

The posterior run length $p(r_t|x_{1:t})$ at time t is estimated sequentially to predict the on-line changes by marginalizing the run length variable as follows:

$$p(x_{t+1}|x_{1:t}) = \sum_{r_t} p(x_{t+1}|x_{1:t}, r_t) p(r_t|x_{1:t})$$
$$= \sum_{r_t} p(x_{t+1}|x_t^{(r)}) p(r_t|x_{1:t}), \tag{3.1}$$

where $x_t^{(r)}$ refers to the last r_t observations of x, and $p(x_{t+1}|x_t^{(r)})$ is computed using the UPM. The run length posterior can be found by normalizing the joint likelihood:

$$p(r_t|x_{1:t}) = \frac{p(r_t, x_{1:t})}{\sum_{r_t} p(r_t, x_{1:t})}. \tag{3.2}$$

The joint likelihood can be updated on-line using a recursive message passing scheme

$$\gamma_t := p(r_t, x_{1:t})$$
$$= \sum_{r_{t-1}} p(r_t, r_{t-1}, x_{1:t}))$$
$$= \sum_{r_{t-1}} \underbrace{p(r_t|r_{t-1})}_{hazard} \underbrace{p(x_t|r_{t-1}, x_t^{(r)})}_{UPM} \underbrace{p(r_{t-1}, x_{1:t-1})}_{\gamma_{t-1}}. \tag{3.3}$$

This defines a forward message passing scheme to recursively calculate γ_t from γ_{t-1}. The conditional can be restated in terms of messages as $p(r_t|x_{1:t}) \propto \gamma_t$. All the distributions mentioned so far are implicitly conditioned on the set of hyper-parameters $\theta = \{\theta_m, \theta_h\}$ [23].

3.3 Bayesian On-Line Spectral Change Point Detection (BOSCPD)

In real-world acoustic environments, both the background additive noise and the channel distortions are highly non-stationary in nature and are not known *a priori*. The non-stationarity in the acoustic conditions causes a rapid change in either mean or variance or both mean and variance of the noise power spectrum density (psd) with time. Under these circumstances, the actual model of the speech signal is highly non-linear and non-Gaussian, as shown in Eq. A.12.

The changes in real-world acoustic conditions can easily be monitored by tracking the changes in the statistical properties of the psd for each frame of the observed speech signal. In this dissertation, we apply the UPM model to detect rapid changes in the noise floor by tracking and monitoring the second order statistic of the noise psd for each noisy speech frame. The UPM is modeled with an independent and identically distributed (iid) Gaussian observation with changing mean and precision of the kth DFT bin. Under this condition, the posterior distribution $p(\mu, \lambda)$ is a normal-gamma or Gaussian-gamma distribution, as is shown in Eq. C.18. If precision is replaced with the corresponding variance, the distribution is called a normal-inverse gamma or Gaussian-inverse gamma distribution. Now the UPM for the kth frequency bin of |Y(m,k)| can be set to the predictive distribution (e.g., for a Student-t predictive) and it can be implemented using Eq. C.18 as follows:

$$|Y(m,k)| \sim \mathcal{N}(\mu, \lambda^{-1}), \tag{3.4}$$

where the mean μ is a Gaussian distribution as follows

$$\mu \sim \mathcal{N}\left(\mu_0, (\lambda\kappa)^{-1}\right), \tag{3.5}$$

κ is a model hyperparameter, and the precision λ is a gamma distribution as follows

$$\lambda \sim \text{Gamm}\left(\alpha_\gamma, \beta_\gamma\right). \tag{3.6}$$

In terms of the variance, Eq. 3.5 and Eq. 3.6 can be written as

$$\mu \sim \mathcal{N}\left(\mu_0, \sigma/\kappa\right), \tag{3.7}$$

$$\sigma^{-2} \sim \text{Gamm}\left(\alpha_\gamma, \beta_\gamma\right), \tag{3.8}$$

where α_γ is the scale parameter and β_γ is the shape parameter of the gamma distribution. The standard conjugate prior of a normal-inverse-gamma distribution for variance is computationally advantageous. Here, the model parameters for the change point detection of the kth bin of the mth frame are $\theta_m \leftarrow \{\mu_0, \alpha_\gamma, \beta_\gamma, \kappa\}$.

In this proposed noise psd tracking model, called Bayesian on-line spectral change detection (BOSCPD), we have replaced the product partition model used for time series with speech frames based on the assumption that the arrival of each frame is independent of other frames. A Hamming window is used for windowing the speech signal and the temporal correlation effects between overlapped adjacent speech frames are neglected in order to make the UPM model simple. A constant hazard function,

$H_h(r|\theta_h) := \theta_{h_{const}}$ similar to [20], is used in this paper. A constant hazard function means $p(r_t = 0|r_{t-1}, \theta_h)$ is independent of r_{t-1} and gives rise to geometric inter-arrival times for change points. Under these conditions, the model hyper-parameters are:

$$\theta = \{\theta_m, \theta_{h_{constant}}\}. \tag{3.9}$$

The detailed description of these model hyper-parameters for the BOCPD-based model can be found in [23] and [20].

3.4 Soft BOSCPD for Additive Noise Compensation

For MCRA-based noise tracking and estimation, any delay in updating noise estimation right after rapid changes in the acoustic conditions may seriously affect the speech denoising performance, especially in transitional regions. In this dissertation, the window update mechanism in MCRA is made as a function of the output results of the proposed BOSCPD algorithm. When a change happens, the search window is reset to new conditions.

The result of the BOSCPD algorithm for each noisy speech frame is a decision whether there is an abrupt change in the noise psd or not. If there is a change point detected in a noisy speech frame, the algorithm raises a flag and the noise tracking algorithm uses this decision to update its noise estimation process as follows:

$$f(C_{m,k}) = \begin{cases} 1 & \text{if } \textit{change point is detected,} \\ 0 & \text{otherwise,} \end{cases} \tag{3.10}$$

where $f(C_{m,k})$ is a function of change point C detected by the BOSCPD algorithm for the kth frequency bin of the mth frame of the noisy speech signal. Finally, the noise estimation in Eqs. 2.21 and 2.22 can be updated in response to abrupt environmental change detection as shown in Algorithm 3.1.

Algorithm 3.1 Updating noise estimation based on the proposed BOSCPD algorithm

if $\text{mod}(m/L) = 0 \;||\; f(C_{m,k}) == 1$ **then**
  $P_{min}(m,k) \leftarrow \min\{P_{tmp}(m-1,k), P(m,k)\}$;
  $P_{tmp}(m,k) \leftarrow P(m,k)$;
else
  $P_{min}(m,k) \leftarrow \min\{P_{min}(m-1,k), P(m,k)\}$;
  $P_{tmp}(m,k) \leftarrow \min\{P_{tmp}(m-1,k), P(m,k)\}$;
end if

For on-line noise tracking and adaptation systems in an environment that may have abruptly changed, the tracking algorithm must be able to track noise based on past speech frames. The proposed BOSCPD for rapid adaptation is based on BOCPD and MRCA. It is able to detect abrupt spectral change points and adapt to the rapidly changing highly non-stationary acoustic environments. The proposed algorithm is summarized in Algorithm 3.2.

The challenging problem for the MCRA noise tracking algorithm is to update its minima search window as soon as the changes occur in the acoustic condition during its highly non-stationary changing time. BOSCPD performs the on-line rapid change detection and adapts the minima search window of the MCRA algorithm, which leads to minimum delay in updating the minima search window.

The target of the proposed BOSCPD noise tracking algorithm is to minimize speech distortion in the spectral domain to improve the SNR. For non-stationary

Algorithm 3.2 Proposed BOSCPD algorithm for non-stationary noise tracking and rapid adaptation in unknown test conditions.

1: Initialization

Set: $F_{buffer} \leftarrow 1$ on-coming speech frame buffer is full

Set: $P(r_0 = 0) \leftarrow 1$ for initial run length r_0, or $P(r_0 = 0) \leftarrow \tilde{S}(r)$ for run length r

Set: $\nu_1^{(0)} \leftarrow \nu_{prior}$, $\chi_1^{(0)} \leftarrow \chi_{prior}$, $\lambda \leftarrow 250$ constant Hazard function;

Set: $CPD \leftarrow false$ initial change point state;

Set: $cpFlag \leftarrow 0$ for initial CPD;

Set: $R_m \leftarrow 0$ holds maximum run length information;

Set: $Y_m \leftarrow 0$ initial magnitude value for kth DFT bin of the m speech frame;

while F_{buffer} is not empty **do**

2: DFT Coefficient Tracking

$Y_m \leftarrow |Y(m,k)|$; //Magnitude for kth DFT bin of the mth speech frame

3: Evaluation of the Predictive Probability

$\pi_m^{(r)} \leftarrow P(Y_m|\nu_m^{(r)}, X_m^{(r)})$; //Predictive probability using student t-distribution

4: Evaluation of Hazard Function

$H_m \leftarrow H_h(r_m)$; //Hazard function

5: Calculate the Growth probabilities

$P(r_m = r_{m-1} + 1, Y_{1:m}) \leftarrow P(r_{m-1}, Y_{1:m-1})\pi_m^{(r)}(1 - H_m)$; //Growth probabilities

6: Calculate the change point probabilities

$P(r_m = 0, Y_{1:m}) \leftarrow \sum P(r_{m-1}, Y_{1:m-1})\pi_m^{(r)}(1 - H_m)$; //Change point probabilities

7: Calculate the evidence

$P(Y_{1:m}) \leftarrow \sum_{r_m} P(r_m, Y_{1:m})$; // Evidence

8: Determine the run length distribution

$P(r_m|Y_{1:m}) \leftarrow P(r_m, Y_{1:m})/P(Y_{1:m})$; // Run length distribution

9: Update sufficient statistics. Posterior updates depend on UPM

$\nu_{m+1}^{(0)} \leftarrow \nu_{prior}$; $\chi_{m+1}^{(0)} \leftarrow \chi_{prior}$;//Sufficient statistics

$\nu_{m+1}^{(r+1)} \leftarrow \nu_m^r + 1$; $\chi_{m+1}^{(r+1)} \leftarrow \chi_m^r + \mu(Y_m)$;

10: Perform prediction

$P(Y_{m+1}|Y_{1:m}) = \sum_{r_m} P(Y_{m+1}|Y_m^{(r)}, r_m)P(r_m|Y_{1:m})$; //Prediction

11: Update run length

$R_m \leftarrow R_m(r_m)$; // Update runlength

12: Change point detection

Search for change point CPD in R_m; //Change point detection (CPD)

13: Update function f in Eq. 3.10

14: Run MCRA algorithm

15: Update noise estimate using Algorithm 3.1

end while

noise, the frame-wise denoising process will increase the segmental SNR of the incoming noisy signal.

3.5 Soft JAC for On-Line ASR

For highly non-stationary acoustic environments, the long-term averaged channel bias is not constant. Instead, it is essential to estimate the channel bias for each speech frame over which the channel bias can be considered as stationary or quasi-stationary. A first-order recursive filter with a time smoothing constant can be used to estimate the channel bias by exploiting the correlation with the previous frame. Such an approach is very suitable for real-time applications where the end of a speech utterance is not known *a priori* and the background environment is highly changing in nature.

3.5.1 Single Channel Soft JAC Model

The background additive noise in a stationary environment shifts the average speech distribution. It tends to mask the speech distribution with low amplitude. The noise masking does not affect the portion of the speech signal with high amplitude energy [13]. However, the overall effect of the additive noise is the elimination of the spectral valleys, which asymmetrically decreases the dynamic range of the power or magnitude channel values. The decrease in the dynamic variation is propagated later to the cepstral features and differential features (delta and delta-delta cepstral features), which are linear combinations of the log power or magnitude channel values. The average of the cepstral features is also shifted. Some of the asymmetrical

masking effects were also translated in the cepstral domain. For example, the shape of cepstral distributions for the coefficients $C0$ and $C1$ becomes very asymmetric in noisy conditions [88].

Simultaneous noise tracking and estimation for each frame with the classical noise subtraction technique in Eq. 2.23, as shown in Fig. 2.7, would reduce the effects of the spectral valley distortions, which in turn increases the dynamic range of the power or magnitude channel values. Then a frame recursive dynamic bias estimation and removal normalizes each frame for channel variations.

In real-world applications, the ASR decodes a stream of frames of live spoken utterances. The decoder does not know the sentence boundaries of the incoming stream of speech signals in advance. Therefore, the ASR decoder works on each incoming stream frame-by-frame and estimates the best confidence score for each frame. A frame adaptive bias-removal technique could be used to minimize the distortion of each frame by subtracting the mean of the features in the cepstral domain. For frame-recursive dynamic bias removal in the cepstral domain, Eq. 2.8 can be written using a first-order recursive filter as

$$\bar{\mathbf{x}}_m = \mathbf{y}_m - \bar{\mathbf{b}}_{m-1}, \tag{3.11}$$

$$\bar{\mathbf{b}}_m = \alpha_b \bar{\mathbf{b}}_{m-1} + (1 - \alpha_b)\mathbf{y}_m, \tag{3.12}$$

where $\mathbf{y}_m$ is the observation cepstrum for current frame, $\bar{\mathbf{x}}_m$ is the bias compensated cepstral feature, $\bar{\mathbf{b}}_m$ is the bias estimate in the cepstral domain from the current observation using a first-order recursive filter, and α_b is a time smoothing constant.

α_b provides a smooth estimation of the bias from frame to frame and its value is 0.995 [2].

To prevent $\bar{\mathbf{x}}_m$ being estimated from a small amount of data, it can be updated after a number of successive frames. An initial value for $\bar{\mathbf{x}}_{m=0}$ can be obtained from the global mean value of the trained HMMs. The schematic diagram of the proposed frame-recursive dynamic joint bias compensation for on-line ASR is shown in Fig. 3.1 and Fig. 3.2.

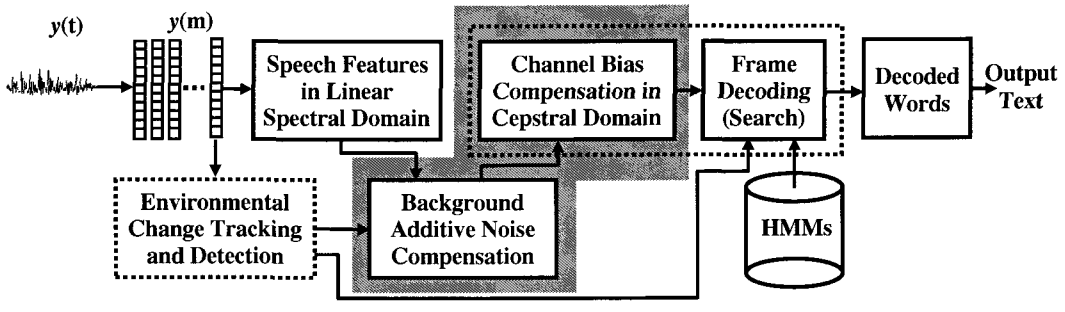


FIGURE 3.1 – Schematic diagram showing the soft architecture of the proposed on-line automatic speech recognition (ASR).The dotted and gray shaded blocks are contributed blocks for the on-line automatic speech recognition in real-life ambiguous unknown acoustic test conditions. Gray shaded region represents proposed JAC compensation.

3.5.2 Soft Channel Distortions Compensation

A first-order recursive filter with a weighted time smoothing parameter for the channel bias compensation is suitable to account for the rapid changes in the acoustic environment due to high non-stationarity in the background test conditions. Mathematically, this approach can be described as follows:

For the feature-based transformation described in Section 2.4.2.1, the estimated speech frame is

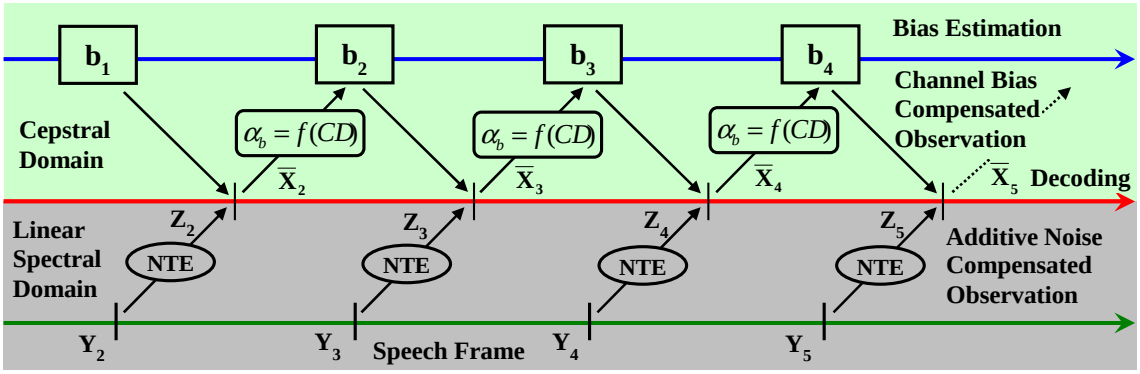


FIGURE 3.2 – Frame adaptive dynamic joint bias compensation (JAC) technique with time smoothing parameter $\alpha_b = 0.995$ [2]. **NTE** stands for noise tracking, estimation and subtraction in a linear spectral domain. **CD** stands for change detection. $\mathbf{Y}$ is the observed speech frame in spectral domain. $\mathbf{Z}$ is the additive noise compensated features in linear spectral domain. $\bar{\mathbf{X}}$ is the channel bias compensated features based on Eq. (3.11) in the cepstral domain. $\mathbf{b}_m$ is the channel bias estimated for the mth frame in a frame-recursive manner and its estimation is a function of change detection **CD**. In the decoding stage, the decoder estimates the bias, which will be used for the next frame, and decodes the best hypothesis for each frame.

$$\mathbf{x}_m = f_v(\mathbf{y}_m) \approx \tilde{\mathbf{y}}_m - \bar{\mathbf{b}}_m, \tag{3.13}$$

where $\tilde{\mathbf{y}}_m$ is the additive noise compensated observed speech signal for the mth frame, and $\bar{\mathbf{b}}_m$ is the estimated bias for the current frame in the cepstral domain.

The bias $\bar{\mathbf{b}}_m$ is updated recursively for the current frame in the cepstral domain using the first-order recursion technique in Eq. 3.11 and, finally, it has to be subtracted from the next frame as shown in Eq. 3.12. However, in a highly non-stationary environment, the channel bias changes quickly during the abrupt transition from one acoustic condition to another. The frame recursive bias compensation technique based on the first-order recursive filter with a constant time-smoothing parameter fails to compensate for the changes of the channel. A static time-smoothing parameter

can affect the bias estimate for the fast changing non-stationary distortions. If it is chosen too large (close to one, such as 0.995) or too low, then the bias estimate might lead to over-estimation or under-estimation of the non-stationary channel bias. Ideally, we would like the smoothing parameter to be small only during a transition of acoustic condition from low to high distortion conditions for better estimating the non-stationarity of the channel bias. Hence, there is a need to make the smoothing factor change dependent, taking into account the abrupt changes of SNR of the speech signal.

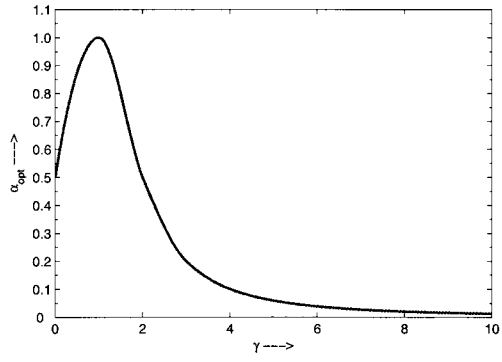
In this thesis, we propose a frame dynamic first-order recursive filter with a weighted time smoothing parameter $\alpha_{b(wt)}$ described in the following:

$$\bar{\mathbf{x}}_m \approx \tilde{\mathbf{y}}_m - \bar{\mathbf{b}}_{m-1}, \tag{3.14}$$

$$\bar{\mathbf{b}}_m = \alpha_{b(wt)} \bar{\mathbf{b}}_{m-1} + (1 - \alpha_{b(wt)}) \tilde{\mathbf{y}}_m, \tag{3.15}$$

where $\bar{\mathbf{b}}_m$ is the updated bias for the current frame, $\alpha_{b(wt)}$ is the weighted (as a function of change detection) smoothing parameter (0.7 to 0.995), $\tilde{\mathbf{y}}_m$ is a noise-compensated current-observed cepstral feature, and $\bar{\mathbf{x}}_m$ is a bias-compensated final observation feature in the cepstral domain. It may be noted here that $\bar{\mathbf{b}}$ needs an initial value, which is supplied from the global mean value of the 13 static MFCC coefficients used for the HMM modeling technique.

The weighted smoothing parameter $\alpha_{b(wt)}$ for channel bias estimation and compensation is a function of smooth *a posteriori* SNR $\bar{\gamma}$, as shown in Fig. 3.3. The optimum weighted smoothing factor $\alpha_{b(wt)}^{opt}$ for each frame is calculated as follows [19]:


1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
α_{opt} --->
0
2
4
6
8
10
γ--->
FIGURE 3.3 – Weighted smoothing parameter $\alpha_{b(w)t}$ for channel bias estimation as a function of the smooth *a posteriori* SNR $\bar{\gamma}$ [19].

$$\alpha_{b(wt)}^{opt} = \frac{\alpha_{b(max)}\alpha_c(m)}{1 + (P(m-1,k)/\hat{\sigma}_d^2(m-1,k) - 1)^2}, \tag{3.16}$$

where $\alpha_{b(max)}$ is set to $\alpha_{b(max)} = 0.995$ to avoid deadlock *a posteriori* SNR $\bar{\gamma}$ becomes 1. $\alpha_c(m)$ is a correction factor that is smoothed over time, and the minimum of the smoothing parameter is set to $\alpha_{min} = 0.7$.

3.6 Simulation

We simulate the proposed soft BOSCPD algorithm to track and compensate the highly non-stationary noises in previously unseen acoustic environments and compare the results with the popular baseline MCRA [4], and two of the most recent derivatives of MCRA, e.g., MCRA2 [94] and EMCRA [95] for noisy speech enhancement. We also simulate the proposed SJAC algorithm for on-line ASR and compare its performance with the baseline MCRA-based JAC for on-line ASR. The performance of

the proposed SJAC is also compared with on-line ASR using the MCRA2, and EM-CRA, and off-line Aurora 2 DSR. The detailed experimental methodologies, speech corpus used in this simulation, and evaluation of the results are presented in Chapter 5.

3.7 Summary

Adaptation to the environmental variabilities and artifacts remains one of the most challenging problems for speech recognition. A robust speech recognition is required to maintain satisfactory recognition performance in previously unseen adverse conditions, which is a tough challenging task for speech scientists. In this chapter, we develop a Bayesian on-line belief-based soft computing approach to compensate for the noise in the feature space in order to improve the noise robust performance of ASR in previously unseen non-stationary acoustic environments. The proposed soft computing technique for noise robust on-line ASR consists of two algorithms: i) the BOSCPD technique, and ii) the soft JAC (SJAC) compensation technique. These algorithms pave the way to develop on-line ASR for real-world applications. Finally, the contributions of this chapter can be summarized as follows:

- helping to develop new cutting-edge soft computing technologies based on the Bayesian on-line belief to solve the non-stationarity problem for ASR in previously unseen real-world test conditions;
- discussing the techniques how to integrate the new technologies into the current ASR in order to develop simultaneous recognition and acoustic model compensation (SJAC) techniques for on-line ASR;

- finally motivating us to advance the soft computing for speech recognition to be noise robust and readily deployable through mobile devices for 3G/4G broadband wireless communications.

In the next chapter, we discuss the basic idea of an on-line ASR based on our proposed bio-inspired soft adaptive filter using the soft evolutionary computing technique, called particle swarm optimization technique.

Chapter 4

Proposition 2: PSO - A Soft Adaptive Filter to Improve the Robustness of On-Line ASR

4.1 Introduction

The bio-inspired soft computing (SC) model appears to be a promising technique to handling real life complex non-stationary acoustic environments. It is becoming an alternative solution to conventional hard computing techniques for speech processing and speech recognition. In recent years SC models for treating uncertainties and variabilities in ambiguous conditions have reached the speech processing and speech recognition fields [24].

Particle swarm optimization (PSO) is an evolutionary algorithm. It is one of the constituent technologies of soft computing techniques. Since human speech is a bio-

logical signal and PSO-based soft computing techniques are generally inspired by biological processes, PSO is better suited for tackling many of the challenging problems of speech processing and speech recognition [24]. This is confirmed by the recent study reports from speech processing literature which show that PSO exhibits flexibility in speech processing with large degree of uncertainty and variability [100], [101]. PSO proves to be superior to the current gradient search-based adaptive filtering techniques for speech enhancement and noise cancellation.

PSO-based SC techniques prove to have the potential to estimate with great precision the coefficients of an adaptive filter to model the unknown time-varying acoustic environments. This biologically inspired precision adaptive filter can be used in improving the noise robustness of current ASR in real-life scenarios. Motivated by the ability of the PSO-based evolutionary SC model over conventional hard adaptive techniques to track the complex non-stationary noisy environments, we propose a dynamic multi-swarm PSO-based soft adaptive filter model to track and compensate the previously unseen non-stationarity of the acoustic environments to improve the noise robustness of ASR in on-line mode.

In this chapter, we develop an on-line soft JAC (SJAC) model to improve the robustness of ASR in rapidly changing non-stationary acoustic environments. Usages of the soft adaptive filter based on dynamic multi-swarm PSO (DMS-PSO) techniques to frame sequentially compensate the non-stationary background additive noises in the front-end, and channel distortion compensation in the back-end of ASR are proposed in this chapter. This approach facilitates opening a new approach to the development of on-line ASR to be noise robust in real-world acoustic environments.

The organization of this chapter is as follows. An overview of particle swarm opti-

mization (PSO) is presented in section 4.2 followed by the mathematical description of the PSO algorithm in section 3.3. Section 4.4 describes the PSO-based additive background noise compensation schemes. An improved version of the PSO, called dynamic multi-swarm PSO (DMS-PSO), is presented in section 3.5. We describe the DMS-PSO-based soft JAC (SJAC) technique for on-line ASR in section 4.6. Section 4.7 briefly describes the simulation setups. Finally, we summarize our achievement in this chapter in section 4.8.

4.2 Particle Swarm Optimization

Particle swarm optimization is a biological population-based stochastic search algorithm. It was originally proposed as a stochastic optimization algorithm in 1995 by Eberhart and Kennedy [21], inspired by the social behavior of bird flocks as shown in Fig. 4.1 or fish schools as shown in Fig. 4.2. Their original intent was to graphically simulate the choreography of a bird flock or fish school. However, it was found that the particle swarm model can be used as an optimizer [21].

PSO has constructive cooperation between particles since particles in the swarm share information. Compared with other optimization algorithms, PSO has many merits. This algorithm is simple, easy to realize, search space is fast, and it demonstrates a wide scope to model noisy time-varying environments [102]. Its efficiency, simplicity as well as adaptability to different problems has rendered PSO as a very attractive approach for solving numerical optimization problems. It results in global solutions even in high-dimensional and multimodal spaces [102], as shown in Fig. 4.3 [103].

PSO is different from other evolutionary algorithms, e.g., genetic algorithms (GAs).

FIGURE 4.1 – Bird Flocking.

FIGURE 4.2 – Fish Schooling.

Indeed, in PSO, the population dynamics simulates a *bird flock's* behavior where social sharing of information takes place and individuals can gain profit from the discoveries and previous experience of all other companions during the search for food. Thus, each companion, called a particle, in the population, which is now called a swarm, is assumed to *fly* over the search space in order to find promising regions of the landscape. For example, in the minimization case, such regions possess lower function values

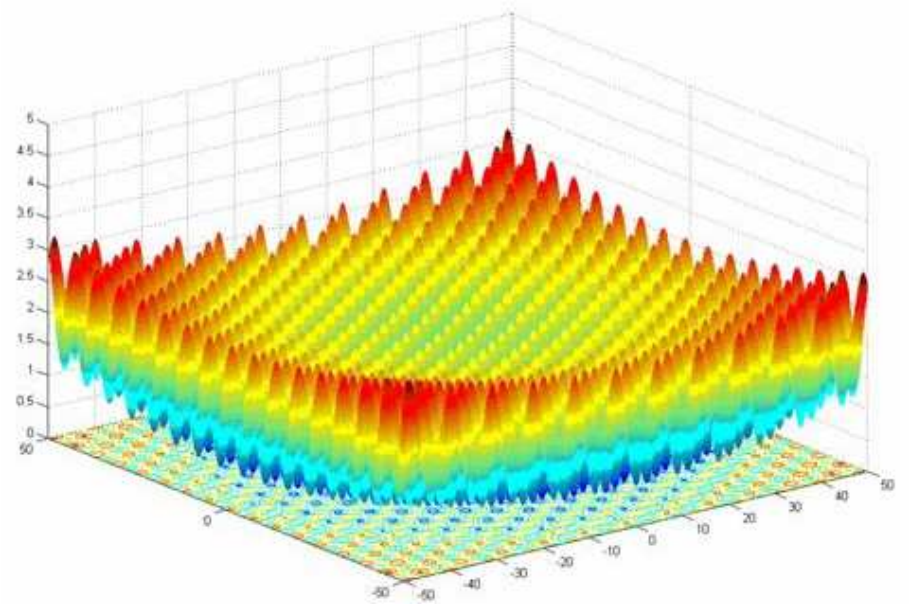


FIGURE 4.3 – Multi-modal Griewank Function (F7).

than others visited previously. In this context, each particle is treated as a point in a D-dimensional space, which adjusts its own *flying* according to its flying experience as well as the flying experience of other particles (companions) [21], [102], [5]. A detailed mathematical formulation of PSO algorithm is presented next.

4.3 Mathematical Framework of PSO

Let $A \subset R^{n_d}$ be the search space, and $f : A \to Y \subseteq R$ be the objective function. In order to keep descriptions as simple as possible, we also assume that A is also the feasible space of the problem at hand, i.e., there are no further explicit constraints posed on the candidate solutions. Also, note that no additional assumptions are required regarding the form of the objective function and search space. As mentioned in the previous chapter, PSO is a population-based algorithm, i.e., it exploits a popu-

lation of potential solutions to probe the search space concurrently. The population is called the *swarm* and its individuals are called the *particles* - a notation retained by nomenclature used for similar models in social sciences and particle physics [22]. The swarm is defined as a set:

$S_p = \{x_1, x_2, x_3, ..., x_{N_p}\}$

of N_p particles (candidate solutions), defined as:

$x_i = \{x_{i1}, x_{i2}, x_{i3}, ..., x_{in_d}\}' \in A,$

where $i = 1, 2, 3, ..., N_p$.

Indices are arbitrarily assigned to particles, while N_p is a user-defined parameter of the algorithm. The objective function, $f(x)$, is assumed to be available for all points in A. Thus, each particle has a unique function value, $f_i = f(x_i) \in Y$.

The particles are assumed to move within the search space, A, iteratively as shown in Fig. 4.4. This is possible by adjusting their 'position' using a proper position shift, called 'velocity', and denoted as:

$v_i = \{v_{i1}, v_{i2}, v_{i3}, ..., v_{in_d}\}',$

where $i = 1, 2, 3, ..., N_p$.

Velocity is also adapted iteratively to render particles capable of potentially visiting any region of A. If ι denotes the iteration counter, then the current position of the ith particle and its velocity will be henceforth denoted as $x_i(\iota)$, and $v_i(\iota)$, respectively.

Velocity is updated based on information obtained in previous steps of the algorithm. This is implemented in terms of a memory, where each particle can store the best position it has ever visited during its search. For this purpose, besides the swarm, S, which contains the current positions of the particles, PSO maintains also

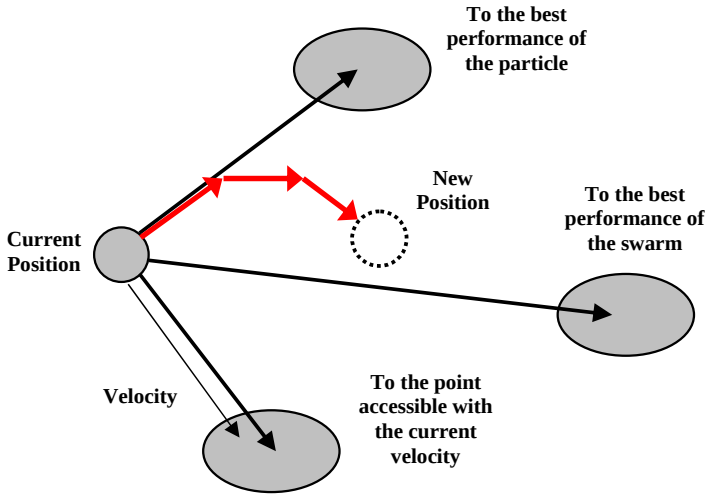


FIGURE 4.4 – Strategy of particle displacement in a PSO technique.

a memory set:

$P_s = \left\{p_1, p_2, p_3, ..., p_{N_p}\right\}^{'}$,

which contains the best position:

$$p_i = \left\{p_{i1}, p_{i2}, px_{i3}, ..., p_{in_d}\right\}^{'} \in A \qquad i = 1, 2, 3, ..., N_p,$$

ever visited by each particle. These positions are defined as:

$$p_i(\iota) = \arg\min_{\iota} \ \mathrm{f}_i(\iota)$$

where ι stands for the iteration counter.

Particle Swarm Optimization (PSO) is based on simulation models of social behavior ; thus, an information exchange mechanism exists to allow particles to mutually communicate their experience. The algorithm approximates the global minimizer with the best position ever visited by all particles. Therefore, it is a reasonable choice to share this crucial information. Let g_p be the index of the best position with the lowest function value in P_s at a given iteration ι, i.e.,

$p_{g_p}(\iota) = \arg\min_{\mathbf{i}} f(p_i(\iota))$.

Then, PSO is defined by the following equations [21]:

$$v_{ij}(\iota+1) = \underbrace{w_p v_{ij}(\iota)}_{momentum} + \underbrace{c_1 R_1 (p_{ij}(\iota) - x_{ij}(\iota))}_{local\ \ information} + \underbrace{c_2 R_2 (p_{g_p j}(\iota) - x_{ij}(\iota))}_{global\ \ information}, \tag{4.1}$$

$$x_{ij}(\iota+1) = x_{ij}(\iota) + v_{ij}(\iota+1), \tag{4.2}$$

where $i = 1, 2, 3, ..., N_p$, $j = 1, 2, 3, ..., n_d$, ι denotes the iteration counter, w_p is the inertia weight chosen in the interval $[0, 1]$, R_1 and R_2 are random variables uniformly distributed within $[0, 1]$, and c_1 and c_2 are weighting factors, also called the 'cognitive' and 'social' parameters, respectively. In the first version of PSO, a single weight, $c = c_1 = c_2$, called the acceleration constant, was used instead of the two distinct weights in Eq. 4.1. However, the latter offered better control on the algorithm, leading to its predominance over the first version.

At each iteration, after the update and evaluation of particles, best positions (memory) are also updated. Thus, the new best position of x_i at iteration $\iota + 1$ is defined as follows:

$$p_i(\iota+1) = \begin{cases} x_i(\iota+1) & \text{if } f(x_i(\iota+1)) < f(p_i(\iota)), \\ p_i(\iota) & \text{otherwise.} \end{cases} \tag{4.3}$$

The new determination of index g_p for the updated best positions completes an iteration of PSO. The operation of PSO is provided in pseudocode in Algorithm 4.1. Particles are usually initialized randomly, following a uniform distribution over the

search space, A. This choice treats each region of A equivalently ; therefore it is mostly preferable in cases where there is no information on the form of the search space or the objective function, requiring a different initialization scheme. Additionally, it is implemented fairly easily, as all modern computer systems can be equipped with a uniform random number generator.

Algorithm 4.1 Pseudo Code for Particle Swarm Optimization (PSO) Algorithm

Input: $N_p \leftarrow$ No. of Particles; $S_p \leftarrow$ Swarm; $P_s \leftarrow$ Best position.
Set:
  $\iota \leftarrow 0$
  $S_p \leftarrow$ initial value
  $P_s \equiv S_p$
Evaluate S_p and P_s, and define index g_p of the best position
while termination criterion not met **do**
  Update S_p using Eqs. 4.1 and 4.2, respectively
  Evaluate S_p
  Update P_s and redefine index g_p
  Set: $\iota \leftarrow \iota + 1$
end while
Record best position

The previous velocity term, $v_{ij}(\iota)$, in the right-hand side of Eq. 4.1, offers a means of inertial movement to the particle by taking its previous position shift into consideration. This property can prevent it from trapping in local minima if suboptimal information is carried by both (e.g., if they both lie in the vicinity of a local minimizer). Furthermore, the previous velocity term serves as a perturbation for the global best particle, x_{g_p}. Indeed, if a particle, x_i, discovers a new position with lower function value than the best one, then it becomes the global best (i.e., $g_p \leftarrow i$) and its best position, p_i, will coincide with p_{g_p} and x_i in the next iteration. Thus, the two stochastic terms in Eq. 4.1 will vanish. If there was no previous velocity term in Eq. 4.1, then

the aforementioned particle would stay at the same position for several iterations, until a new best position is detected by another particle. Otherwise, the velocity term allows this particle to continue its search, following its previous position shift. The values of c_1 and c_2 can affect the search ability of PSO by biasing the sampled new positions of a particle, x_i, towards the best positions, p_i and p_{g_p}, respectively, as well as by changing the magnitude of the search [22].

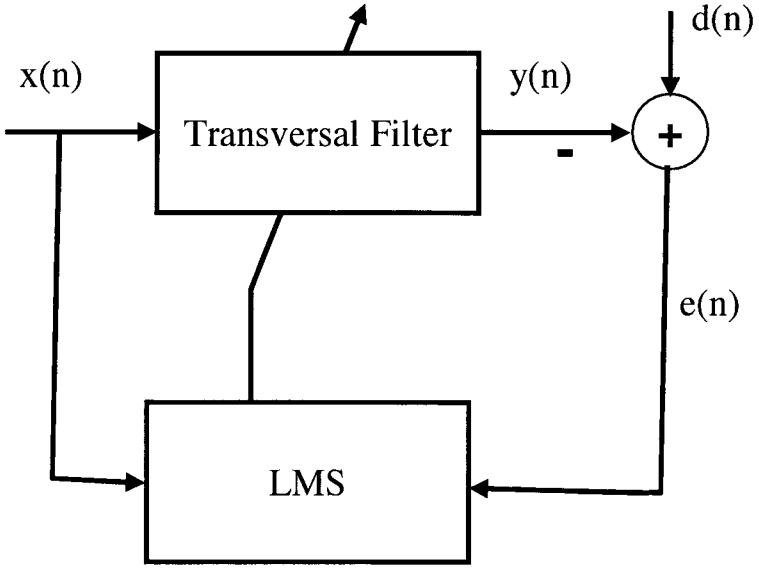


FIGURE 4.5 – LMS algorithm for speech enhancement.

4.4 Additive Noise Compensation Using PSO

Speech denoising in non-stationary acoustic environments is an optimization problem to compute the globally optimal estimate of the speech signal. High non-stationarity in acoustic environments leads to non-linear and non-Gaussian models, and solutions

of these complex non-linear acoustic models have multiple local and global optima. For optimal estimation of speech signals in noisy conditions, it is essential to design an optimal estimator that will provide a global optimal estimation of the speech signal. There have been enormous advances in gradient-search-based optimization techniques over the last several years. However, these gradient-based algorithms cannot solve the multimodal and high dimensional nonlinear objective functions as, in most cases, they converge to local optima.

Recently, PSO [21], [22] has found application in estimating a speech signal in additive noise, as it provides global solutions even in high-dimensional and multimodal spaces. However, it is not tested well enough against multiplicative noises in dynamic environments, which is an interesting research topic at present. Not many results are available about PSO's behavior for speech processing in the presence of non-stationary noise. In other words, the performance of the PSO is not known when noise is inserted into the function values and/or the environment is continuously changing [102].

4.4.1 Dual-Channel Speech Denoising Using PSO

Gradient search-based adaptive optimization algorithms are widely used in estimation theory for the speech enhancement process. For example, the Least Mean Square (LMS) algorithm, as shown in Fig. 4.5, is one of the common adaptive algorithms widely used in Adaptive Digital Speech Processing (ADSP) for denoising noisy speech [104]. The normalized version of the LMS, called NLMS, outperforms the LMS algorithm in the sense of SNR improvement. There are several other adaptive algorithms such as Recursive-Least-Squares (RLS) algorithms, which also have wide

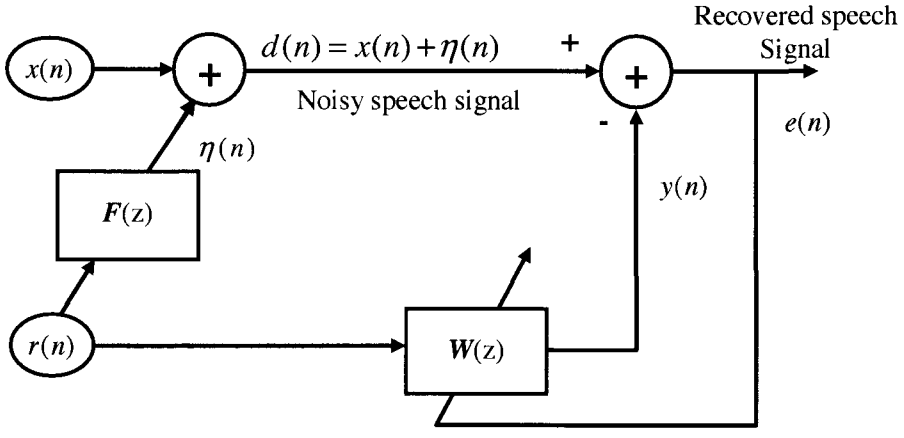


FIGURE 4.6 – Dual-channel speech enhancement using the PSO technique [100]. Here $x(n)$ is the clean speech signal, $d(n)$ is the noisy speech signal, $\eta(n)$ is the background noise added to the speech signal, $y(n)$ is the output of the adaptive filter $W(z)$, $r(n)$ is the source of background additive noises, and $e(n)$ is the error signal, which represents the recovered speech signal.

application for the speech enhancement process in noisy conditions. These adaptive algorithms work based on the principle of the classical Wiener filter. A Wiener filter works in dual-channel mode, i.e., it needs a reference noise information of the noise in order to denoise the noisy speech signal [104].

In [100], the authors used an improved version of the PSO technique for a dual-channel speech enhancement process and found very encouraging results in denoising the speech signal even in very low SNR conditions. The fundamental design process of this dual-channel speech enhancement process is shown in Fig. 4.6.

In a PSO-based dual-channel speech denoising process, the position of each particle in the swarm represents a candidate for the coefficients of the adaptive filter $W(z)$ in Fig. 4.6 [100], [101]. After a predetermined number of iterations, the coefficients of the optimal adaptive filter $W(z)$ are determined according to the position vector of the best particle in the swarm (g_{best}) . Then, $y(n)$ is obtained by modifying the noise

reference by the adaptive filter $W(z)e(n)$. Finally, the enhanced frame is obtained by subtracting $y(n)$ from $d(n)$.

4.4.2 Adaptive Noise Compensation Using PSO for On-Line ASR

In PSO, each potential solution is regarded as a particle. All particles have fitness values and velocities. The particles fly through the D-dimensional problem space by learning from the historical information of all the particles. Using the useful information collected in the search process, the particles have a tendency to fly towards a better search area over the course of the search process. Though PSO works well compared to gradient-based optimization techniques, it has a problem of early convergence. It still is in an early stage of development. To avoid this problem, two main variants of the PSO search process have been developed - i) global PSO, and ii) local PSO. In the local version of PSO, each particle's velocity is adjusted according to its personal best and the best performance achieved so far within its neighborhood instead of learning from the personal best and the best position achieved so far by the whole population in the global version. In focusing on improving the local version of PSO, different neighborhood structures are proposed and discussed [5].

Eberhart and Kennedy, who first developed and introduced the PSO optimization technique [21], claimed that PSO with a large neighborhood would perform better for simple problems and PSO with a small neighborhood might perform better on complex problems [5]. Estimating adaptive filter coefficients based on the minimization of the objective function in Eq. (4.4) in a frame-dynamic fashion in highly

non-stationary acoustic environments is a unimodal complex problem. Within this complex modeling constraints in Eq. (4.4) for adaptive speech denoising, we propose to use PSO with small local neighborhoods.

Currently, several versions of a local PSO algorithm are available in the literature. In [5], the authors proposed a dynamic multi-swarm particle swarm optimization with a local search technique called DMS-PSO. The authors successfully implemented this DMS-PSO for optimizing 25 complex functions and found results superior to other forms of local PSO. Among these complex functions, five were unimodal and the remaining functions were multi-modal. In this dissertation, we choose to use DMS-PSO to optimize the soft adaptive filter coefficients. A perfectly optimized filter provides a better solution to predict the dynamics of the non-stationary acoustic environments.

4.4.3 Discussion

Within the limited scope and time of this dissertation, we decide to implement a DMS-PSO algorithm to design a soft adaptive filter to be suitable for tracking noisy speech signal in non-stationary acoustic environments. We do not perform a detailed analysis of DSM-PSO. A full-scale mathematical analysis of DMS-PSO will be provided in future work. In the next section, we provide a brief description of the form of DMS-PSO.

4.5 Dynamic Multi-Swarm PSO

In DMS-PSO, small neighborhoods are used to reduce the population's convergence velocity and to increase diversity and achieve better results on complex search problems. In this case, the population is divided into small swarms. Each swarm consists of two to three particles and it uses its own members to search for better areas in the search space.

Small swarms search based on their own best historical information; there are risks for these particles to converge to a local optimum because of PSO's convergence property. In order to avoid this premature convergence, an exchange of information among the swarms is allowed in DMS-PSO. In this information exchange, more information including the good ones and the less good ones is kept in the record to add to the varieties of the particles and achieve larger diversity. So a randomized grouping schedule is used to make the particles have a dynamic changing neighborhood structure.

An example of DMS-PSO with local search for three swarms with three particles in each swarm is graphically shown in Fig. 4.7. In this example, each of the randomly partitioned three swarms uses its own particles to search for better solutions. In this case, each solution may converge to near a local optimum. Then the whole swarm population is grouped into new swarms. The new swarms begin their search. These grouping processes are continued until a stop criterion is satisfied. With the random grouping schedule, particles from different swarms are grouped in a new configuration so that each small swarms search space is enlarged and better solutions are possible to be found by the new small swarms [5].

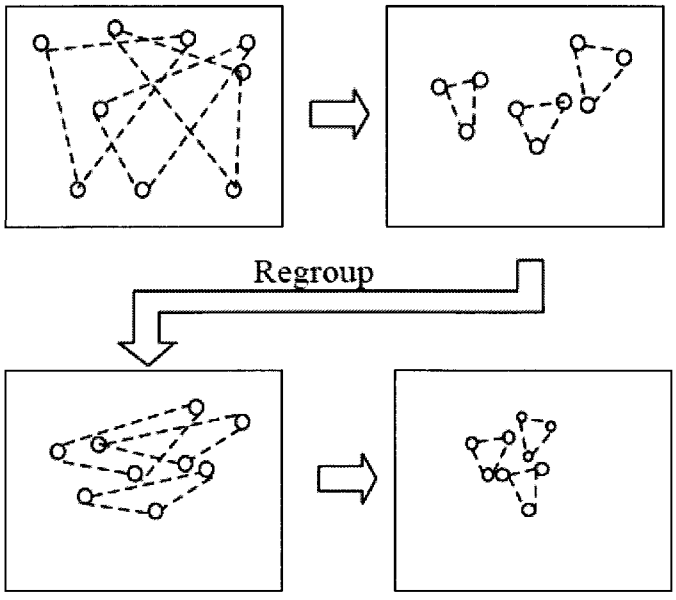


FIGURE 4.7 – DMS-PSO search [5].

In DMS-PSO with local search [5] as presented in Algorithm 3.2, for every R_G generations, the population is grouped randomly and starts searching using a new configuration of small swarms. Here R_G is called the grouping period. In this way, the information obtained by each swarm is exchanged among the swarms. Simultaneously the diversity of the population is increased. The new neighborhood structure has more freedom when compared with the classical neighborhood structure [5]. In DMS-PSO with local search, when updating the positions of the particles, half of the dimensions are kept the same as its best historical position, *pbest*, to make better use of the particles' historical information to improve its global search ability.

In PSO optimization, a larger diversity and a faster convergence velocity are always

Algorithm 4.2 Non-Stationary Noise Compensation Using DMS-PSO [5]

Set: $m_{sp} \leftarrow$ Each swarm's population size
Set: $n_{sn} \leftarrow$ Swarm's number
Set: $n_d \leftarrow$ Search space dimension for each particle
Set: $R_G \leftarrow$ Grouping period
Set: $L_l \leftarrow$ Local refining period
Set: $L_{FEs} \leftarrow$ Max fittest evaluations (FE) using in the local search
Set: $Max_{FEs} \leftarrow$ Max fitness evaluations, stop criterion
Set: $m_{sp} \times n_{sn} \leftarrow$ Particles initialization for position and velocity
Set: $FEs \leftarrow 0$
Set: $gen \leftarrow 0$
while $FEs < 0.95 \times Max_{FEs}$ **do**
  $gen = gen + 1$
  for $i = 1 \quad to \quad (m_{sp} \times n_{sn})$ **do**
    $Find \quad lbest_i$
    for $\iota = 1 \quad to \quad n_d$ **do**
      if $rand < 0.5$ **then**
        $v_i^\iota = w \times v_i^\iota + c_1 \times R_{1i}^\iota \times (pbest_i^\iota - x_i^\iota) + c_2 \times R_{2i}^\iota \times (lbest_i^\iota - x_i^\iota)$
        $v_i^\iota = min(max(v_i^\iota - v_{max}^\iota), v_{max}^\iota)$
        $x_i^\iota = x_i^\iota + v_i^\iota$
      else
        $x_i^\iota = pbest_i^\iota$
      end if
    end for
    if $x_i \in [x_{min}, x_{max}]^{n_d}$ **then**
      Calculate the fitness value
      $FEs = FEs + 1$
      Update *pbest*
    end if
  end for
  if $mod(gen, L_l) == 0$ **then**
    Sort *lbest* according to their fitness value and refine the first $\lceil 0.25n \rceil$ best *lbest* using Quasi-Newton method
    $FEs = FEs + \lceil 0.25n \rceil \times L_F Es$
    Update the corresponding *pbest*
  end if
  if $mod(gen, R_G) == 0$ **then**
    Group the swarms randomly
  end if
end while

a trade-off problem. Since DMS-PSO provides a larger diversity, at the same time, the algorithm loses its fast convergence velocity. This problem is alleviated to give a better search in the better local areas by adding a local search to DMS-PSO. The DMS-PSO algorithm is described in Algorithm 3.2. Nevertheless, a detailed description of DMS-PSO for optimizing multi-modal objective functions is available in [5], [105], [106].

4.6 Soft JAC Compensation Using DMS-PSO

In this dissertation, we implement a DMS-PSO optimization technique for frame-dynamic non-stationary distortion compensation, as shown in Fig. 4.8. In this noise compensation technique, we adopt the dual-channel adaptive noise cancellation technique to minimize the cost function. For on-line speech recognition, the observed input signal $d(n)$ is processed in frames. For speech denoising using DMS-PSO, we follow similar steps in [100] to define the cost function to evaluate the fitness of each particle. The cost function is derived based on the average error between the noisy speech signal, $d(n)$, and the estimated noise signal, $\Omega(n)$, in each frame and it is based on the principle that the fittest particles will have lower cost function values. The cost function is defined as follows:

$$\Pi_i = \frac{1}{N_w} \sum_{k=0}^{N_w-1} \left(d(k) - \Omega_i(k)\right)^2, \tag{4.4}$$

where N_w is the length of each frame, and $\Omega(n)$ is the output of $W(z)$ designed by the algorithm. When Π_i is minimum, the parameters of $W(z)$ provide the best possible representation of the unknown non-stationary acoustic environments.

Algorithm 4.3 Non-Stationary Noise Compensation Using DMS-PSO

Set: Frame index $m \leftarrow 1$
while Frame of speech not end **do**
  $r(m) \leftarrow d(m)$
  Find best particle from DMS-PSO Algorithm 4.2
  Evaluate $W(z)$ using best particles as the filter coefficients
  $\Omega(m) \leftarrow W(z) \otimes r_2(m)$
  $e(m) \leftarrow (d(m) - y(m))$
  $\hat{s}(m) \leftarrow e(m)$
  $m \leftarrow m + 1$
end while

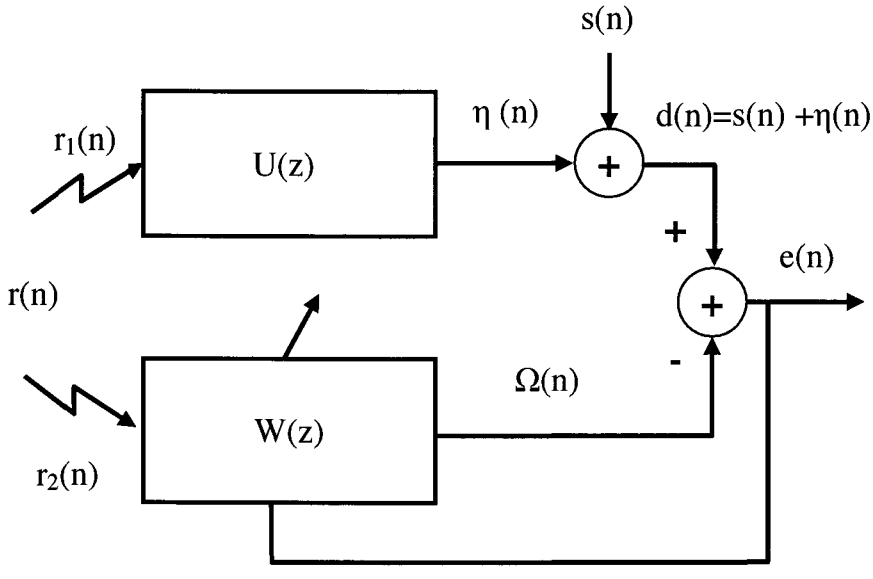


FIGURE 4.8 – Proposed noise compensation using the DMS-PSO technique.

This adaptive filter predicts the unknown environments represented by the filter $U(z)$. In this model in Fig. 4.8, the speech signal $d(n)$ is the clean speech signal $s(n)$ corrupted by the background noise $\eta(n)$. The adaptive filter $W(z)$ tries to model the rapidly changing non-stationary acoustic environments. In this experiment, the reference noise is picked by a microphone. We model the path the reference noise travels from its source to the second microphone by a moving average filter. The

output of this moving average filter is the input to unknown adaptive filter $W(z)$. The filter output $\Omega(n)$ is an estimate of the noise present in the current frame. The error signal in the output of the structure, $e(n)$, becomes an estimate of the clean signal $s(n)$.

$$W(z^{-}) = \frac{p_i^1 + p_i^2 z^{-1} + p_i^3 z^{-2} + p_i^4 z^{-3} + p_i^5 z^{-4}}{1 + p_i^6 z^{-1} + p_i^7 z^{-2} + p_i^8 z^{-3} + p_i^9 z^{-4}} \tag{4.5}$$

We adapt an IIR filter with an order of 4 as shown in Eq. 4.5. In this case, the filter coefficients are used as the dimension of each particle in the swarm. After getting the best fitting particle, the dimensions of this best particle are used as filter coefficients and the output of this filter $\Omega(n)$ is directly subtracted from the observed speech frame, which results in the best estimate of the actual speech signal. The algorithm for this DMS-PSO-based noise compensation technique is described in Algorithm 3.3.

4.7 Simulation

We simulate the bio-inspired DMS-PSO-based soft adaptive filter algorithm to track and compensate the highly non-stationary noises in previously unseen acoustic environments for dual channel noisy speech enhancement. We also simulate the DMS-PSO-based SJAC algorithm for on-line ASR and compare its performance with the baseline BOSCPD-based SJAC for on-line ASR. The detailed experimental methodologies, speech corpus used in this simulation, and evaluation of the results are presented in the next chapter.

4.8 Summary

In this chapter, we discuss and develop evolutionary stochastic-based optimization techniques, especially the DMS-PSO technique for frame-dynamic adaptive noise tracking and estimation. This new algorithm shows the way to develop on-line ASR for real-world applications.

In order to develop on-line ASR technologies to meet the demand of current hand-held mobile devices for real-world applications, the contribution of this chapter is helping to develop PSO-based front-end processing of automatic speech recognition to solve the non-stationarity problem for mobile applications in unknown test conditions.

In the next chapter, we present the detailed experimental methodologies, simulation setups, and results and performance evaluation of the soft computing techniques to improve the robustness of on-line ASR that we propose in this thesis.

Chapter 5

Experiments and Results

In order to demonstrate the performances of our on-line ASR compared to off-Line ASR, we performed experiments in the following chronological order:

- A comprehensive study of current approaches to improve the noise robustness of ASR in noisy acoustic environments, especially in non-stationary noisy conditions,
- Simulation of on-line ASR using Aurora 2 speech data in non-stationary environments using bias-removal techniques as a first step towards developing noise-robust ASR for real-time applications,
- Development of a soft joint additive and channel distortion compensation (SJAC) technique for on-line noise tracking and rapid change detection as an essential criterion for on-line ASR to work in real-world non-stationary acoustic conditions, and
- Development and simulation of an advanced SJAC model using stochastic optimization techniques, called particle swarm optimization (PSO), to implement

the proposed on-line ASR.

We develop our proposed on-line ASR algorithms using the Aurora 2 speech database from the ETSI [1] distributed speech recognition (DSR) specification. Before proposing the on-line ASR, we did an extensive study on the noise robustness problems of current ASR modeling techniques. In this chapter we present our study as well as simulation results for the proposed frame-dynamic on-line ASR algorithms.

In Section 5.1, we briefly describe speech corpora that we use to validate our proposed algorithms for on-line ASR. Section 5.2 presents the simulation results of an on-line ASR using a frame dynamic bias removal technique in non-stationary noisy conditions. We present the result of the proposed soft BOSCPD technique for SJAC-based non-stationary noise tracking and compensation for the on-line ASR in Section 5.3. Section 5.4 discusses the results of on-line ASR using the SJAC based on the particle swarm optimization (PSO) technique. Finally, we summarize the results of this work in Section 5.5.

5.1 Aurora 2 Speech Database

This section describes the Aurora 2 speech corpus developed for distributed speech recognition (DSR) in simulated non-stationary acoustic environments [1], [107]. In the Aurora 2 Distributed Speech Recognition (DSR), speech feature processing is done in the telecommunication terminal (front-end) and the recognition is carried out at a remote central location (back-end) in the telecom network based on the ETSI (European Telecommunications Standards Institute) [3] standards for DSR. In this DSR architecture, the front-end is located at the client side and connected

over the data channel of telecommunication networks to remote back-end recognition servers, as shown in Fig. 5.1. The DSR architecture includes two-stage Mel-warped Wiener filtering for noise enhancement, Mel frequency cepstral coefficient (MFCC) feature extraction and compression as well as bit-streaming, formatting and decoding at the front-end.

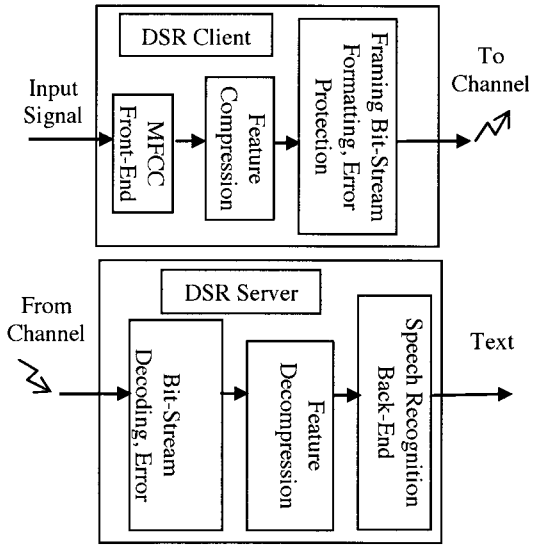


FIGURE 5.1 – Schematic diagram of the Aurora 2 DSR architecture [1], [3].

The source speech for the Aurora 2 database is the TI Digits, consisting of a connected digits task spoken by American English talkers and down-sampled to 8 kHz. It contains speech of isolated digit sequences of up to 7 digits from 110 speakers of US-American adults among whom 55 speakers are male and 55 speakers are female. The original 20 kHz data of TI Digits were recorded in a single sitting session. In Aurora 2, these data have been down-sampled to 8 kHz with a precision of 16 bits with an "ideal" low-pass filter extracting the spectrum between 0 and 4 kHz.

In Aurora 2, the training data set contains 8440 utterances of 110 speakers. Each speaker utters 75 to 77 sentences. These training data are equally split into 20 subsets with 422 short utterances in each subset in multi-condition training mode. The 20 subsets represent four different artificially added noise scenarios Subway, Babble, Car, and Exhibition Hall at 20 dB, 15 dB, 10 dB, 5 dB and $>$ 30 dB signal-to-noise ratios (SNRs).

The Aurora 2 speech database has also three sets of test data (set 'A', set 'B', and set 'C') from 104 speakers (52 male and 52 female). In each category, each speaker utters about 9-10 utterances of digits ranging from a single digit to a maximum of 7 digits. These test data were corrupted by artificially added 8 different real-world noises (e.g., Subway, Babble, Car, Exhibition Hall, Restaurant, Street, Airport, and Train) at 20 dB, 15 dB, 10 dB, 5 dB, 0 dB and -5 dB signal-to-noise ratios (SNRs).

In Aurora 2, speech and noises are added artificially and filtered with the G.712 and MIRS (modified intermediate reference system) characteristic before adding in order to artificially simulate the actual non-linear acoustic distortions [3]. This filtering is done to consider the realistic frequency characteristics of terminals and equipment in the telecommunication area. The G.712 characteristic is defined for the frequency range of the usual telephone bandwidth up to 4 kHz and has a flat characteristic in the range between 300 and 3400 Hz. MIRS shows a rising characteristic with an attenuation of lower frequencies that simulates the behavior of a telecommunication terminal, which meets the official requirements for the terminal input frequency response as specified, e.g., for GSM.

The Aurora 2 DSR [1] is designed to evaluate the performance of the standard Aurora 2 task of recognizing digit strings in noise and a channel distorted environ-

TABLE 5.1 – Recognition accuracy of clean-trained model in batch-mode (off-line) for the Aurora 2 DSR [107].

	Clean training - Results													
	A					B					C			Average
	Subway	Babble	Car	Expo Hall	Average	Restaurant	Street	Airport	Station	Average	Subway (MIR)	Street (MIR)	Average	
clean	98.83	98.97	98.81	99.14	98.94	98.83	98.97	98.81	99.14	98.94	99.02	98.97	98.99	98.96
20 dB	96.96	89.96	96.84	96.20	94.99	89.19	95.77	90.07	94.38	92.35	94.47	95.19	94.83	94.06
15 dB	92.91	73.43	89.53	91.85	86.93	74.39	88.27	76.89	83.62	80.79	87.63	89.69	88.66	85.46
10 dB	78.72	49.06	66.24	75.10	67.28	52.72	66.75	53.15	59.61	58.06	75.19	75.27	75.23	66.86
5 dB	53.39	27.03	33.49	43.51	39.36	29.57	38.15	30.69	29.74	32.04	52.84	48.85	50.84	40.75
0 dB	27.00	11.73	13.27	15.98	17.00	11.70	18.68	15.84	12.25	14.62	26.01	21.64	23.83	18.48
-5 dB	12.62	4.96	8.35	7.65	8.40	5.0	10.07	8.11	8.49	7.92	12.10	10.70	11.40	9.24
Average	65.78	50.73	58.08	61.35	58.99	51.63	59.52	53.37	55.32	54.96	63.89	62.90	63.40	59.12

ment. It works in batch-mode (off-line), and uses 39 MFCC coefficients by using 12 static cepstral coefficients $(C_1, C_2, ..., C_{12})$ and the logarithmic frame energy, 13 Δ coefficients and 13 $\Delta\Delta$ acceleration coefficients for HMMs. It uses whole word HMMs with 18 states per word including 2 dummy states at the beginning and the end. These HMMs are left-to-right models without skip-over states. They use a mixture of 3 Gaussians per state. The Aurora 2 DSR is tested using HTK [15] as the reference recognizer. Table 5.1 shows the performances of the Aurora 2 digit recognition for a clean training model [107].

Aurora 2 speech database is one of the most widely used speech corpora for many benchmark research results on DSR-based speech recognition. It demonstrates the simulated real-world highly non-stationary acoustic environments. For this reason, we use this speech database for our proposed on-line ASR for mobile communication

devices.

TABLE 5.2 – Recognition accuracy of the clean-trained model in simulated on-line mode without frame-dynamic noises and distortion compensation for the Aurora 2 DSR.

	Clean training - Results													
	A					B					C			Average
	Subway	Babble	Car	Expo Hall	Average	Restaurant	Street	Airport	Station	Average	Subway (MIR)	Street (MIR)	Average	
clean	97.97	98.31	98.12	98.12	98.13	97.97	98.31	98.12	98.12	98.13	97.27	97.25	97.26	97.84
20 dB	49.80	37.06	42.20	47.92	44.25	37.64	47.04	38.32	36.35	39.84	59.23	58.25	58.74	47.61
15 dB	38.78	27.45	32.30	34.46	33.25	28.95	33.89	29.44	28.05	30.08	47.26	44.38	45.82	36.38
10 dB	31.35	18.53	23.51	24.38	24.44	19.40	25.21	21.29	21.38	21.82	32.69	30.06	31.38	25.88
5 dB	22.94	12.33	16.61	17.03	17.23	10.01	17.29	14.32	12.58	13.55	22.66	22.44	22.55	17.76
0 dB	13.6	8.40	11.01	11.77	11.20	6.71	11.85	8.65	6.48	8.42	14.70	14.90	14.80	11.47
-5 dB	8.54	5.91	8.53	8.55	7.90	3.88	8.53	6.26	6.57	6.31	9.97	12.42	11.20	8.47
Average	37.57	29.71	33.18	34.60	33.77	29.22	34.59	30.91	29.93	31.16	40.54	39.96	40.25	35.06

5.2 Dynamic Bias Removal Technique Results

This section presents the simulation results of the MCRA-based frame dynamic bias removal technique in the cepstral domain. We have tested the Aurora 2 task of recognizing digit strings in non-stationary environments for on-line conditions using the same configuration parameters for HMM models used for the Aurora 2 DSR. We conduct this simulation using the ATK toolkit [2].

The standard Aurora 2 task of recognizing digit strings in noise and channel distorted environments was tested off-line for two modes of training data: (i) clean-

TABLE 5.3 – Recognition accuracy of clean-trained model for MCRA-based on-line frame recursive bias removal technique of the Aurora 2 task of recognizing digit strings.

	Clean training - Results													
	A					B					C			Average
	Subway	Babble	Car	Expo Hall	Average	Restaurant	Street	Airport	Station	Average	Subway (MIR)	Street (MIR)	Average	
clean	98.86	98.88	98.66	99.01	98.85	98.86	98.88	98.66	99.01	98.85	98.89	98.82	98.86	98.86
20 dB	96.28	97.34	97.85	96.48	96.99	97.45	96.92	97.35	97.99	97.43	95.98	96.70	96.34	96.92
15 dB	92.85	93.68	94.15	92.07	93.19	94.47	92.41	94.66	95.25	94.20	91.68	93.08	92.38	93.26
10 dB	80.99	83.52	80.79	79.11	81.10	85.02	80.05	88.04	83.86	84.24	79.40	78.63	79.02	81.46
5 dB	55.85	55.75	48.97	51.13	52.93	62.65	53.93	65.45	57.33	59.84	54.10	52.77	53.44	55.41
0 dB	28.53	25.89	21.83	22.63	24.72	30.37	24.43	35.95	26.53	29.32	28.62	23.03	25.83	26.63
-5 dB	12.27	9.26	6.56	7.88	8.99	11.07	8.36	11.72	9.00	10.04	11.84	8.87	10.36	9.80
Average	66.52	66.33	64.12	64.04	65.25	68.56	65.00	70.26	67.00	67.71	65.79	64.56	65.18	66.05

only training, and (*ii*) multi-condition training. The performance of the Aurora 2 DSR for clean-only training data is poor compared to that of the multi-condition one. However, it performs the same for clean test data in both the training cases. We have decided to use the clean-only training model for our test. The objective of our on-line ASR is to compensate the test utterances for noisy environments, i.e., to bring back the test speech signal close to the clean model. This tactic keeps the training model unchanged.

In a real-time environment, *a priori* knowledge of the acoustic conditions is not known. Therefore, ASR should be trained on clean data first and then, it needs to track the environmental changes to self-adapt to new conditions. For clean test data, Aurora 2 performs the same in both the batch-mode and on-line. However, its

performance degrades quickly in on-line conditions without any bias compensation compared to its batch-mode performances. These simulation results are presented in Table 5.2.

In order to improve the performance of the on-line ASR using a frame-adaptive bias removal technique, we use the MCRA noise tracking algorithm to track the non-stationary noises and compensate it. In this simulation, we also change the configuration parameters used for training the HMM models. We include the coefficient of order 0 in the static MFCC coefficients ($C_0, C_1, C_2, ..., C_{12}$) and train the HMM models using HTK [15]. For frame-adaptive on-line ASR, we use a running average cepstral mean normalization technique using Eqs. 3.11 and 3.12 and reset the running-mean back to the default at the start of every input utterance. The simulation results for this experiment are shown in Table 5.3.

From these simulation results, we have found that the performance of the Aurora 2 DSR in on-line condition without frame-adaptive bias compensation decreases quite rapidly with a decrease in SNR values, though at clean conditions, its performances are quite similar to the batch mode (off-line). The percent reduction in the performance of the Aurora 2 DSR in on-line mode without bias compensation is presented in Table 5.4. The results show that the average drop in the word accuracy for test set 'A' is 42.09%, for test set 'B' is 43.32%, and for test set 'C' is 36.51%. The overall reduction in word recognition accuracy rate for Aurora 2 DSR is 40.64%. These simulation results show that at very low SNR conditions, especially at 0 dB or below, the ASR recognition accuracy is very poor due to the fact that at those SNRs, speech signals are always highly dominated by the background noise.

In the case of on-line simulation of the Aurora 2 DSR in frame-adaptive MCRA-

TABLE 5.4 – Reduction (%) in the recognition accuracy of the off-line Aurora 2 DSR in frame dynamic decoding without bias compensation schemes in the task of recognizing digit strings.

	Clean training - Results													
	A					B					C			Average
	Subway	Babble	Car	Expo Hall	Average	Restaurant	Street	Airport	Station	Average	Subway (MIR)	Street (MIR)	Average	
clean	0.87	0.67	0.70	1.03	0.82	0.87	0.67	0.70	1.03	0.82	1.77	1.74	1.75	1.13
20 dB	48.64	58.80	56.42	50.19	53.51	57.80	50.88	57.46	61.49	56.91	37.30	38.81	38.05	49.49
15 dB	58.26	62.62	63.92	62.48	61.82	61.08	61.61	61.71	66.46	62.71	46.07	50.52	48.29	57.61
10 dB	60.18	62.23	64.51	67.54	63.61	63.20	62.23	59.94	64.13	62.38	56.52	60.06	58.29	61.43
5 dB	57.03	54.38	50.40	60.86	55.67	66.15	54.68	53.34	57.70	57.97	57.12	54.06	55.59	56.41
0 dB	49.63	28.39	17.03	26.35	30.35	42.65	36.56	45.39	47.10	42.93	43.48	31.15	37.31	36.86
-5 dB	32.33	-19.15	-2.40	-11.76	-0.25	22.40	15.29	22.81	22.61	20.78	17.60	-16.07	0.76	7.10
Average	42.89	41.44	40.43	43.60	42.09	42.09	41.89	42.08	45.90	43.32	35.55	36.47	36.51	40.64

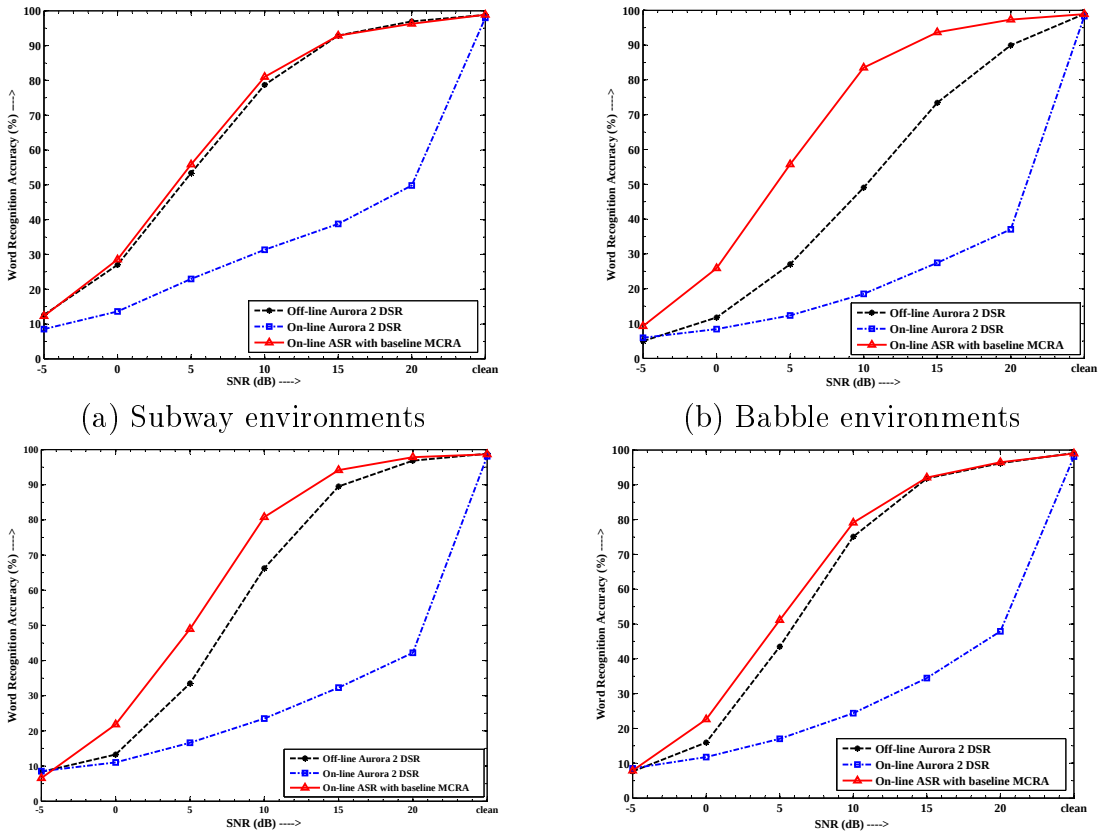
based bias compensation mode, we have found that the joint additive and channel distortion compensation in a frame dynamic fashion increases the word recognition accuracy greatly compared to the off-line recognition accuracy rate, especially for SNRs between 20 dB and 5 dB. This gain in recognition accuracy as shown in Table 5.5, shows that the joint compensation for additive and channel noises works well according to the acoustic model of the speech signal in Eq. A.12. The results show that the average increase in the word accuracy for test set 'A' is 11.67%, for test set 'B' is 23.69%, and for test set 'C' is 2.81%. The overall increase in word recognition accuracy rate for MCRA-based on-line ASR is 12.72% compared to the off-line results in Table 5.1. This simulation result shows that at very low SNR conditions, especially at 0 dB or below, the ASR recognition accuracy is very poor due to the fact that at

those SNRs, speech signals are always highly dominated by the background noise and the observed speech signals become almost noise.

TABLE 5.5 – Improvement (%) of recognition accuracy of the clean-trained model for MCRA-based on-line frame recursive bias removal schemes of the Aurora 2 task of recognizing digit strings.

	Clean training - Results													
	A					B					C			Average
	Subway	Babble	Car	Expo Hall	Average	Restaurant	Street	Airport	Station	Average	Subway (MIR)	Street (MIR)	Average	
clean	0.03	-0.09	-0.15	-0.13	-0.09	0.03	-0.09	-0.15	-0.13	-0.09	-0.13	-0.15	-0.14	-0.10
20 dB	-0.70	8.20	1.04	0.29	2.21	9.26	1.20	8.08	3.82	5.59	1.60	1.59	1.59	3.13
15 dB	-0.06	27.58	5.16	0.24	8.23	26.99	4.69	23.11	13.91	17.18	4.62	3.78	4.20	9.87
10 dB	2.88	70.24	21.97	5.34	25.11	61.27	19.93	65.64	40.68	46.88	5.60	4.46	5.03	25.67
5 dB	4.61	106.25	46.22	17.51	43.65	111.87	41.36	113.26	92.77	89.82	2.38	8.02	5.20	46.22
0 dB	5.67	120.72	64.51	41.61	58.13	160.43	30.78	126.96	116.57	108.68	10.03	6.42	8.23	58.35
-5 dB	-2.77	86.69	-21.44	3.01	16.37	121.40	-16.98	44.51	6.01	38.73	-2.15	-17.10	-9.63	15.16
Average	1.12	30.75	10.40	4.38	11.67	32.79	9.21	31.65	21.11	23.69	2.97	2.64	2.81	12.72

A graphical comparison of the performances of preliminary experiments on the Aurora 2 task of recognizing digit strings in both off-line (batch-mode) vs. MCRA-based on-line modes in highly non-stationary noises is shown in Fig. 5.2 for test data set 'A', Fig. 5.3 for test data set 'B', and Fig. 5.4 for test data set 'C', respectively. These experimental results show that the MCRA-based frame-recursive dynamic acoustic distortions compensation improves the performance of the Aurora 2 significantly compared to its off-line results. Since the test data for the Aurora 2 are pre-recorded sentences, the ASR decoder in ATK reads one sentence each time from



(a) Subway environments

(b) Babble environments

(c) Car environments

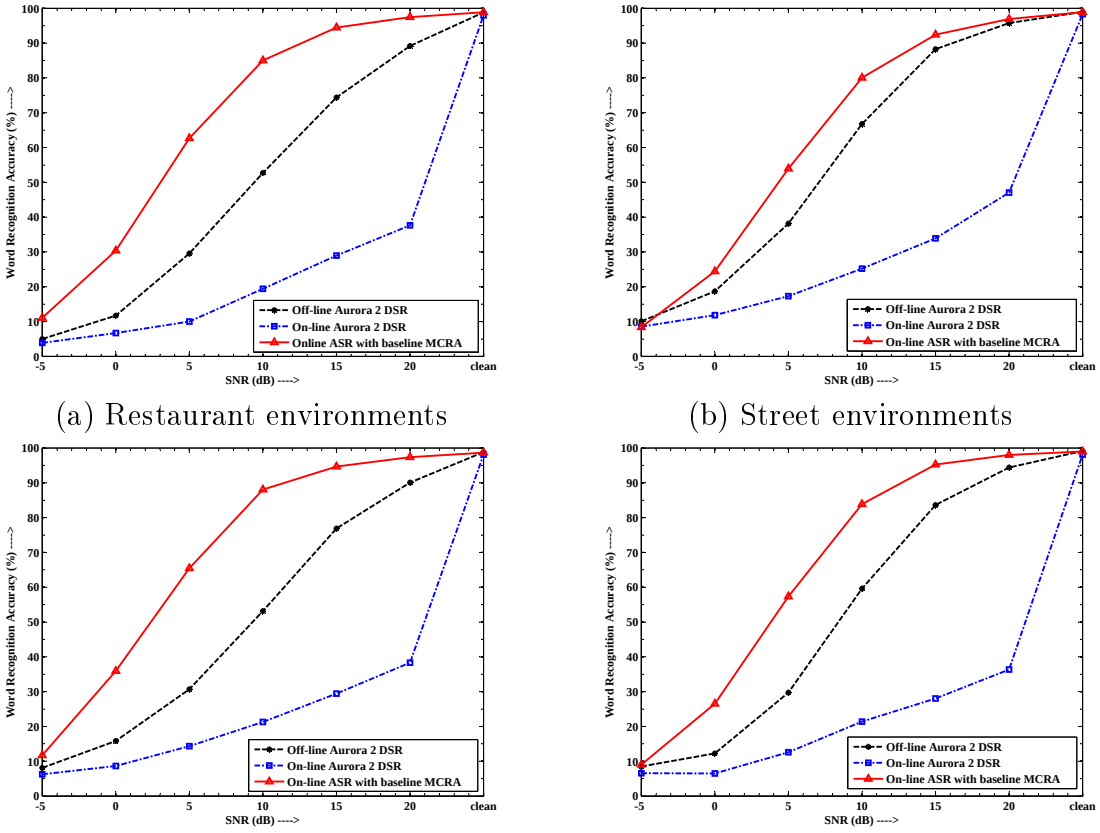
(d) Exhibition hall environments

FIGURE 5.2 – Recognition performances of Aurora 2 DSR in off-line vs. on-line ASR with baseline MCRA for test data set 'A'.

the list of test data and mimics the real-time spoken utterances by sending a stream of frames to the decoder.

5.3 Results for BOSCPD-Based On-Line ASR

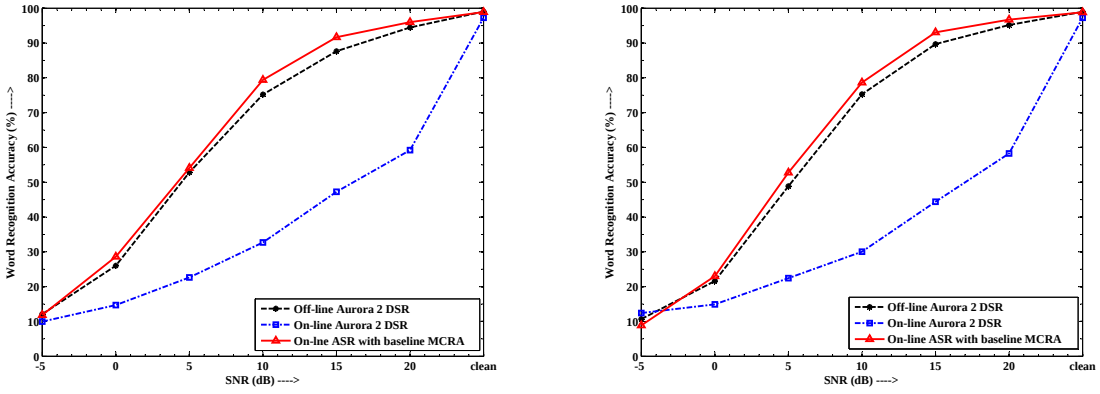
In this section, we present the simulation results for SJAC-based on-line ASR using our proposed BOSCPD technique. The additive non-stationary noise is substantially reduced in the front-end processing using the BOSCPD-based SJAC algorithm. The non-stationary channel distortion bias is removed during the decoding stage as shown



(a) Restaurant environments (b) Street environments

(c) Airport environments (d) Train environments

FIGURE 5.3 – Recognition performances of Aurora 2 DSR in off-line vs. on-line ASR with baseline MCRA for test data set 'B'.



(a) Subway environments (b) Street environments

FIGURE 5.4 – Recognition performances of Aurora 2 DSR in off-line vs. on-line ASR with baseline MCRA for test data set 'C'.

in Fig. 5.5. In these two-step decoding processes, the decoder first estimates the frame hypothesis score based on the bias estimated in the previous frame and then it estimates the bias to be used for the next frames. The experimental setup for this simulation as shown in Fig. 5.5 is described in the following subsections.

5.3.1 Simulation Setup for BOSCPD Algorithm

Before recognition results are presented we first will present the results of some simulations to illustrate the behavior of the proposed BOSCPD algorithm and obtain some non-recognition based performance measures. We measure the performance of the proposed BOSCPD approach using speech samples from the Aurora 2 speech database [1] as described in Section 5.1. In order to validate the proposed BOSCPD algorithm to track fast changing acoustic environments, we devise three simulation environments as follows:

5.3.1.1 Simulation Environment I

For simulation environment I, we examine three test cases of subway noise. In test case I, the test speech sample consists of two acoustic conditions in cascade: the speaker suddenly moves from a clean ($>$ 30 dB) to a 15 dB SNR condition and stays there some time. Then the speaker moves to a clean ($>$ 30 dB) condition again. Similarly, in test case II, the test speech sample consists of two acoustic conditions in cascade: the speaker suddenly moves from a clean ($>$ 30 dB) to a 5 dB SNR condition and then moves to a clean ($>$ 30 dB) condition. In test case III, the test speech sample consists of three acoustic conditions in cascade: the speaker suddenly moves from a 5 dB SNR condition to a 15 dB SNR and then moves to a clean ($>$ 30

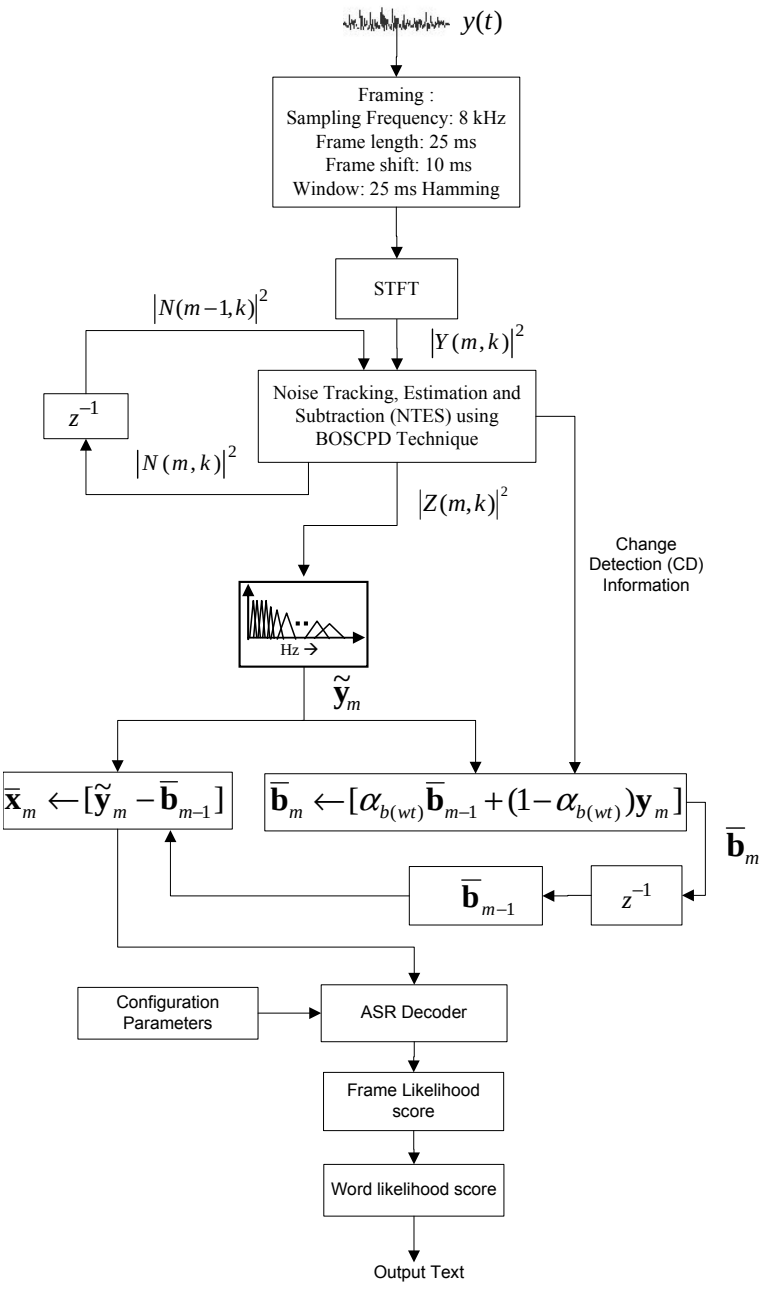


FIGURE 5.5 – Flow diagram of the proposed BOSCPD algorithm for the soft computing model of on-line ASR as described in Fig. 3.1 and Fig. 3.2.

dB) condition. Test cases I and II demonstrate well background acoustic conditions that changed rapidly from very high SNR to low SNR. The first two test conditions

represent worst-case scenarios where MCRA shows maximum delay in adaptation to new conditions.

5.3.1.2 Simulation Environment II

For the simulation environment II, we examine a single test utterance. The test utterance is corrupted by babble noise at 5 dB SNR. This test case demonstrates well background acoustic conditions that change rapidly with time.

5.3.1.3 Simulation Environment III

For the simulation environment III, we examine a test case where the speaker suddenly moves from the babble noise environment at a 5 dB SNR condition to the subway noise environment at a 5 dB SNR condition. This test case demonstrates well background acoustic conditions that change rapidly from one environment to another environment.

These simulation test environments represent a real-time situation for non-stationary noises where both the mean and variance of the speech spectral properties change due to non-stationarity of the acoustic regimes. Both speech signal and noise are assumed to be iid Gaussian. In our experiment, we tracked changes of the magnitude value of a DFT bin of the observed noisy speech signal based on the proposed BOSCPD technique to compensate for the rapidly changing non-stationary noises.

5.3.2 HMM Configurations for On-Line ASR

To confirm the validity of the proposed BOSCPD-based on-line ASR, we compared its performance to the MCRA-based on-line ASR's recognition performance. The on-

line ASR is tested using the same HMM configuration as we discussed in previous subsection 5.2. We use the ATK toolkit to implement it.

5.3.3 Non-stationary Noise Tracking Results

In this subsection, we present the tracking performances of our proposed BOSCPD algorithm compared to the baseline MCRA and some of its derivatives, e.g., MCRA2 [94] and EMCRA [95], in rapidly changing noisy environments. From the simulation results, we find that our proposed BOSCPD algorithm shows an improvement in tracking rapid variations in the spectral properties of the noisy speech signal compared to the MCRA-based noise tracking algorithms. The performances of the BOSCPD algorithm for each test case as we mentioned earlier are discussed next.

5.3.3.1 Test Results for Simulation Environment I

The proposed BOSCPD-based frame dynamic SJAC compensation algorithm for non-stationary noises is validated by comparing its performance to the baseline MCRA-based techniques (e.g., MCRA [4], MCRA2 [94], EMCRA [95]). From the simulation results, it can be seen that our proposed method performs excellently for worst case scenarios where acoustic conditions change rapidly from very high SNR to low SNR conditions in test cases I and II. In test case III, the proposed algorithm follows the MCRA algorithm. Graphical representations of performances of the proposed BOSCPD-based noise tracking, rapid change detection, and adaption algorithms for test cases I, II, and III are shown in Fig. 5.6, 5.7 and Fig. 5.8 respectively.

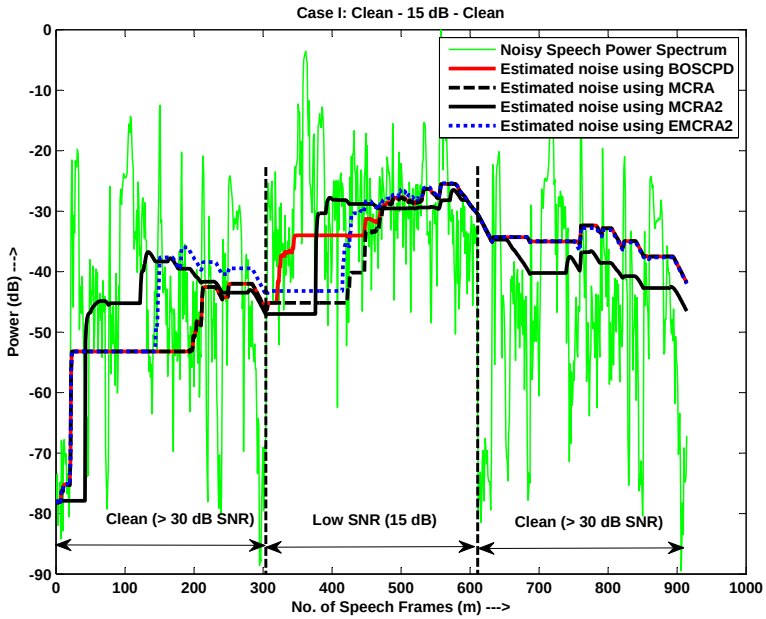


FIGURE 5.6 – Comparison between the noise spectrum (for $f = 750$ Hz) estimated using the proposed BOSCPD algorithm and MCRA [4], MCRA2 [94] and EMCRA [95] algorithms for test case I in the simulation environment I.

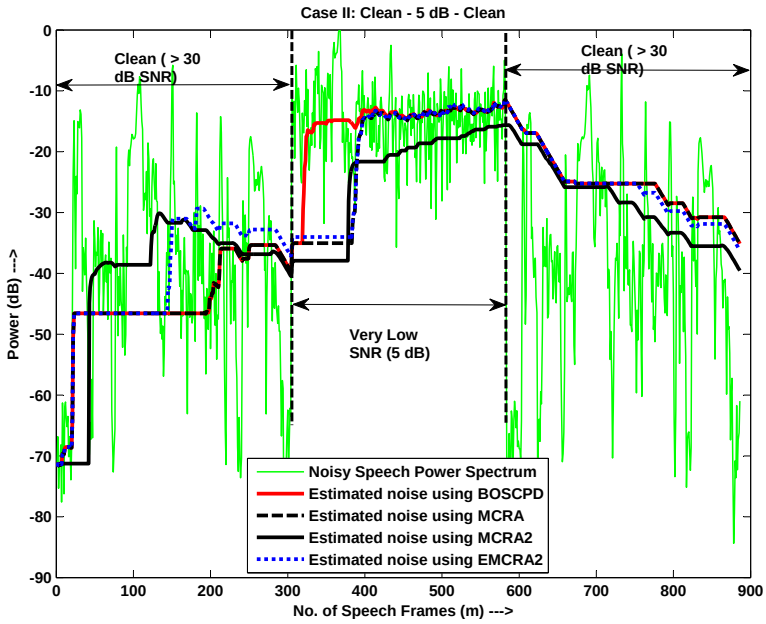


FIGURE 5.7 – Comparison between the noise spectrum (for $f = 750$ Hz) estimated using the proposed BOSCPD algorithm and MCRA [4], MCRA2 [94] and EMCRA [95] algorithms for test case II in the simulation environment I.

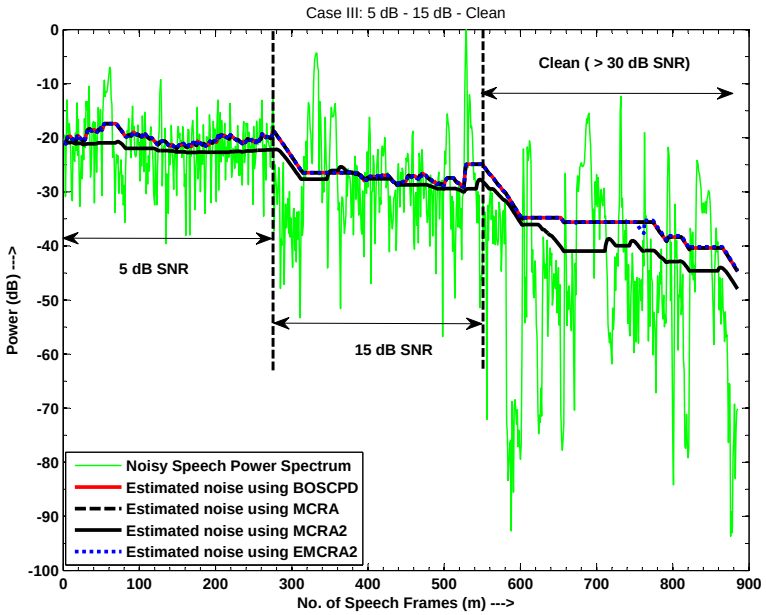


FIGURE 5.8 – Comparison between the noise spectrum (for $f = 750$ Hz) estimated using the proposed BOSCPD algorithm and MCRA [4], MCRA2 [94] and EMCRA [95] algorithms for test case III in the simulation environment I.

5.3.3.2 Test Results for Simulation Environment II

In this case, we evaluate the performance of the proposed BOSCPD technique for noisy speech enhancement. We use several standard objective quality measures such as i) global SNR (GSNR), ii) segmental SNR (segSNR), iii) Itakura-Saito distortion (It-Sa), iv) weighted spectral slope (WSS), and v) perceptual evaluation of speech quality (PESQ). For one particular noisy speech file, results are summarized in Table 5.6.

Figure 5.9 shows an example noise spectrum estimated with our algorithm and with MCRA [4], MCRA2 [94], and EMCRA [95] for a scenario in which the spoken utterance is degraded with highly non-stationary babble noise. Our algorithm is able to track non-stationarity in environments and adapt to the new environment without delay while MCRA-based algorithms required large delay to adapt.

Figure 5.10 compares the performance of the proposed BOSCPD algorithm with MCRA for denoising the noisy speech signal degraded by babble noise. The time window size L is set to 64 frames for both the proposed algorithm and the MCRA algorithm. The proposed algorithm performed better than the original MCRA, which can be easily observed from Figure 5.10(d-e).

TABLE 5.6 – Speech Enhancement Comparison of Different Noise Power Spectrum Estimation Techniques.

	GSNR	SegSNR	It-Sa	WSS	PESQ
Noisy Speech	5.264	-1.545	3.848	93.927	1.987
MCRA	9.304	0.623	3.061	85.681	2.357
MCRA2	8.672	0.166	2.612	88.046	2.316
EMCRA	9.352	0.524	3.507	86.488	2.373
BOSCPD	9.397	0.631	3.050	85.382	2.420

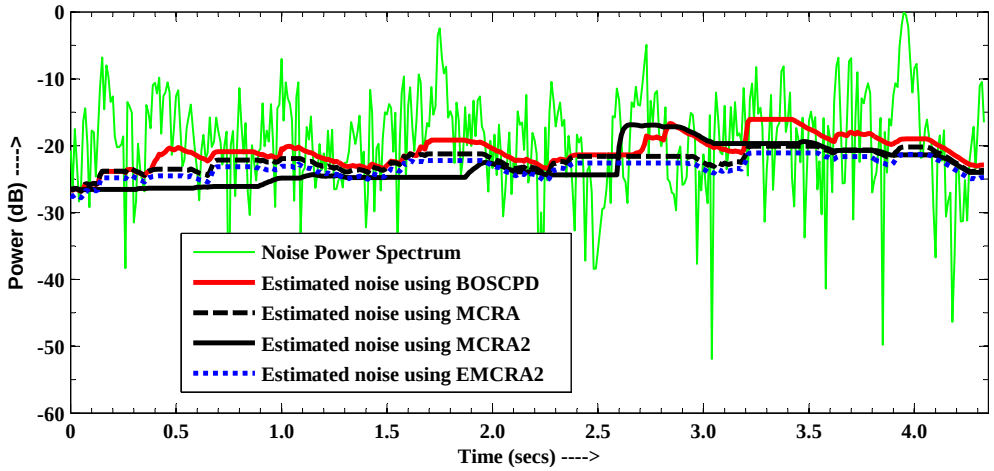


FIGURE 5.9 – Comparison between the noise spectrum (for $f = 1.5$ kHz) estimated using the proposed algorithm and MCRA [4], MCRA2 [94] and EMCRA [95] algorithms for a sentence corrupted by babble noise at 5 dB SNR.

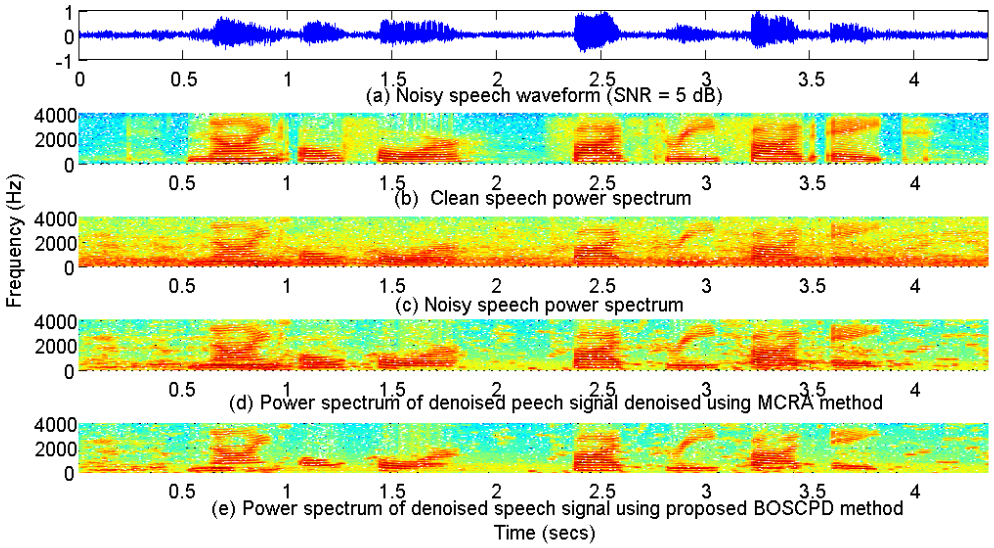


FIGURE 5.10 – Comparison of speech enhancement performances using the proposed algorithm and the baseline MCRA algorithms for the test utterance corrupted by babble noise.

5.3.3.3 Test Results for Simulation Environment III

The noise tracking performance of the proposed BOSCPD algorithm for simulation environment III is shown in Fig. 5.11. This test environment shows an example noise spectrum estimated with our algorithm and with the MCRA [4], MCRA2 [94] and EMCRA [95] algorithms for a scenario in which the noise environment changes suddenly with an increased noise floor. Our algorithm is able to adapt to the new environment within 0.08 sec, while the MCRA and EMCRA algorithms required 1.1 secs, and the MCRA2 algorithm required 1.3 secs to adapt.

In all the tests cases, a standard spectral subtraction-type speech enhancement method has been used to perform the noise removal. Speech signals sampled at 8 kHz are segmented into 25-ms frames using a Hamming window with 60% overlap.

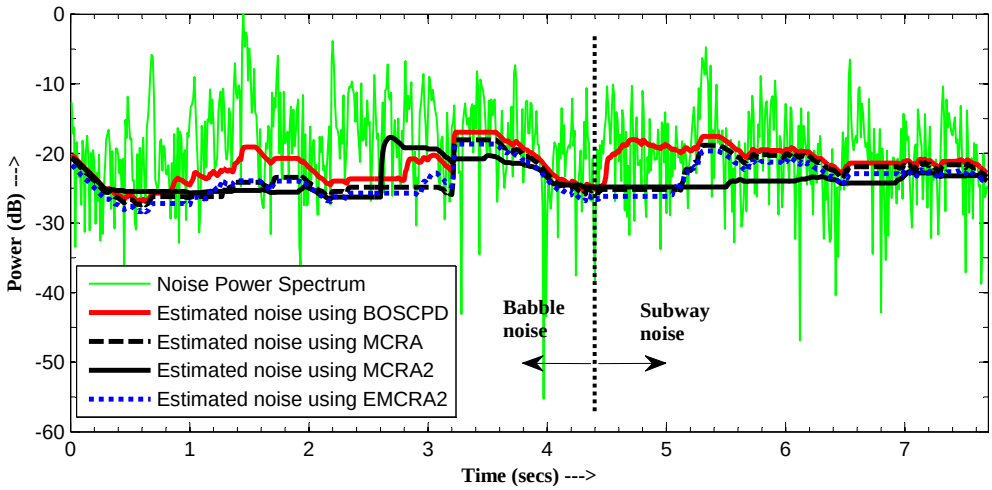


FIGURE 5.11 – Comparison between the noise spectrum (for $f = 1.5$ kHz) estimated using the proposed algorithm and the MCRA [4], MCRA2 [94] and EMCRA [95] algorithms for a sentence corrupted by babble noise ($t < 4.4$ sec) followed by a sentence corrupted by subway noise ($t > 4.4$ sec).

5.3.4 Recognition Performance of the BOSCPD-Based On-Line ASR

The simulation results of the on-line ASR show that the proposed BOSCPD-based SJAC technique increases the word recognition accuracy greatly compared to the baseline MCRA, MCRA2 and EMCRA systems, especially for SNRs between 20 dB and 5 dB. We conduct these recognition simulations for test data set 'A', set 'B', and set 'C' of the Aurora 2 speech database [1], [107]. The recognition performance of the proposed BOSCPD-based on-line ASR is shown in Table 5.7, and percentage gain in word recognition accuracies compared to the baseline MCRA-based on-line ASR is presented in Table 5.11.

The recognition results for test data set 'A' compared to the baseline MCRA, MCRA2 and EMCRA are shown graphically in Fig. 5.12. Test data set 'B' represents restaurant, street, airport, and train station environments, and these results are

TABLE 5.7 – Recognition accuracy of the proposed BOSCPD-based on-line ASR using the clean-trained model for recognizing digit strings.

	Clean training - Results													
	A					B					C			Average
	Subway	Babble	Car	Expo Hall	Average	Restaurant	Street	Airport	Station	Average	Subway (MIR)	Street (MIR)	Average	
clean	98.96	98.90	97.96	98.91	98.68	98.88	98.89	98.86	99.51	99.04	98.98	98.92	98.95	98.89
20 dB	97.65	98.34	98.35	97.38	97.93	97.95	97.02	98.05	98.09	97.78	97.78	97.87	97.83	97.84
15 dB	93.85	95.68	96.55	94.27	95.09	96.37	93.51	95.86	96.45	95.55	92.89	94.58	93.74	94.79
10 dB	83.99	85.52	83.19	82.10	83.70	88.42	84.45	89.54	85.06	86.87	81.69	79.03	80.36	83.64
5 dB	57.85	59.75	49.87	53.83	55.33	66.85	58.83	67.05	59.03	62.94	58.05	55.07	56.56	58.28
0 dB	30.53	29.89	23.03	24.03	26.87	33.47	26.03	38.05	27.03	31.15	30.02	25.63	27.83	28.61
-5 dB	13.50	7.91	7.26	8.18	9.21	12.97	10.06	12.02	9.50	11.14	13.04	9.17	11.11	10.49
Average	68.05	68.00	65.17	65.53	66.69	70.70	66.97	71.35	67.81	69.21	67.49	65.75	66.62	67.51

graphically presented in Fig. 5.13. Similarly, Test data set 'C' represents two simulated test acoustic environments (MIRS filtered), e.g., subway and street, and the recognition results in these noisy conditions are shown in Fig. 5.14.

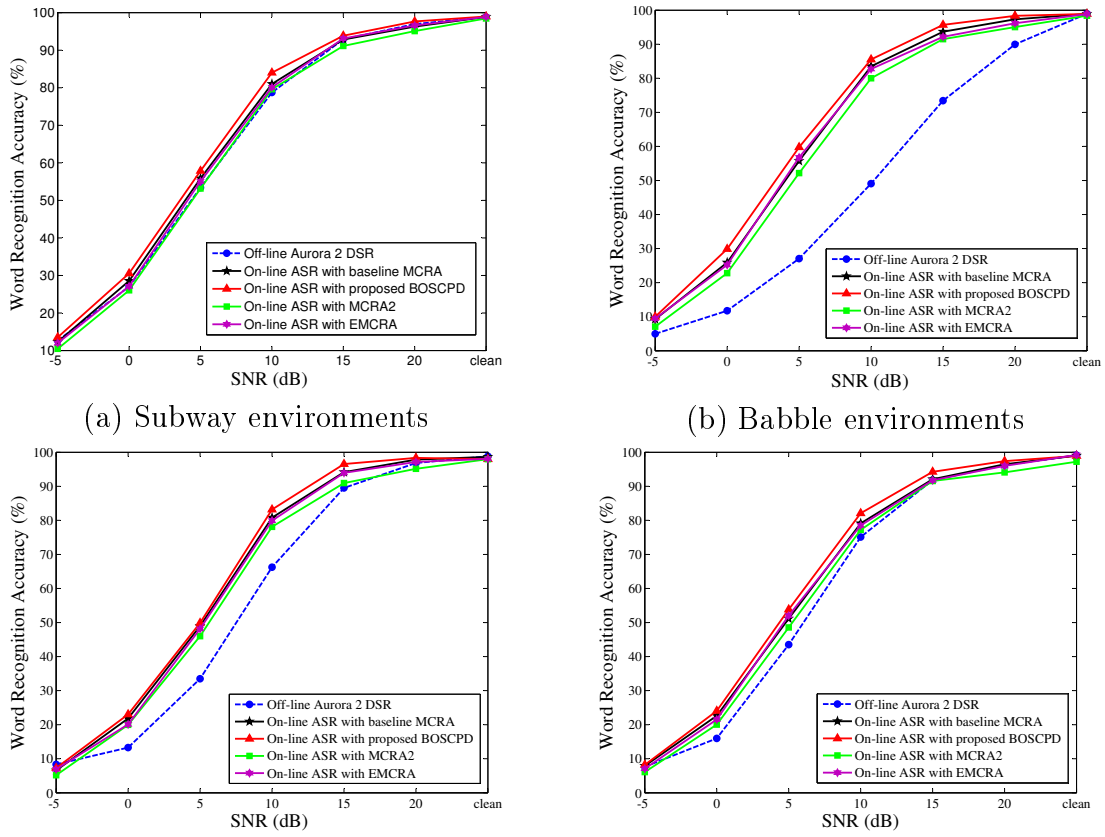
We present the recognition performance of the proposed BOSCPD-based on-line ASR using the clean-trained model in Table 5.7. We also show the percentage improvement in digit recognition accuracies using our proposed BOSCPD algorithm for on-line ASR compared to the baseline MCRA-based on-line ASR in Table 5.11. In this performance evaluation, we include test set data in noisy environments ranging from a clean condition to -5 dB SNR. Our test results show that the proposed BOSCPD-based on-line ASR gains 2.25% overall improvement compared to the baseline MCRA-based on-line ASR. We achieve an improvement of 5.16% in digit recognition accuracy compared to the baseline MCRA system within 20 dB to 0 dB

TABLE 5.8 – Improvement (%) of recognition accuracy of the proposed BOSCPD-based on-line ASR using the clean-trained model for recognizing digit strings compared to the baseline MCRA-based on-line ASR.

	Clean training - Results													
	A					B					C			Average
	Subway	Babble	Car	Expo Hall	Average	Restaurant	Street	Airport	Station	Average	Subway (MIR)	Street (MIR)	Average	
clean	0.10	0.02	-0.71	-0.10	-0.17	0.02	0.01	0.20	0.50	0.18	0.09	0.10	0.10	0.04
20 dB	1.42	1.03	0.51	0.93	0.97	0.51	0.10	0.72	0.10	0.36	1.88	1.21	1.54	0.96
15 dB	1.08	2.13	2.55	2.39	2.04	2.01	1.19	1.27	1.26	1.43	1.32	1.61	1.47	1.65
10 dB	3.70	2.39	2.97	3.78	3.21	4.00	5.50	1.70	1.43	3.16	2.88	0.51	1.70	2.69
5 dB	3.58	7.17	1.84	5.28	4.47	6.70	9.09	2.44	2.97	5.30	7.30	4.36	5.83	5.20
0 dB	7.01	15.45	5.50	6.19	8.54	9.85	6.55	5.84	1.88	6.03	4.89	11.29	8.09	7.55
-5 dB	10.02	7.78	10.67	3.81	8.07	17.16	20.33	2.56	5.56	11.40	10.14	3.38	6.76	8.74
Average	2.30	2.96	1.65	2.32	2.31	3.11	3.04	1.55	1.22	2.23	2.59	1.85	2.22	2.25

SNR ranges.

The word recognition performances of MCRA2- and EMCRA2-based on-line ASRs for test data sets 'A', 'B', and 'C' of Aurora 2 speech corpus as shown in Fig. 5.12, Fig. 5.13, and Fig. 5.14, respectively, are poorer compared to the performance of MCRA-based on-line ASR. Among these two variants of MCRA, MCRA2 performs very poorly compared to EMCRA2 in most cases. This is due to the fact that MCRA2 fails to follow the rapid changes in acoustic environments, as we can see in Fig. 5.9, and it introduces more unwanted distortions especially at higher SNR conditions. However, we include these two variants of MCRA in our performance as they represent the most recent improvements of the MCRA algorithm. The simulation results show that our decision to consider MCRA as our baseline system is right.



(a) Subway environments

(b) Babble environments

(c) Car environments

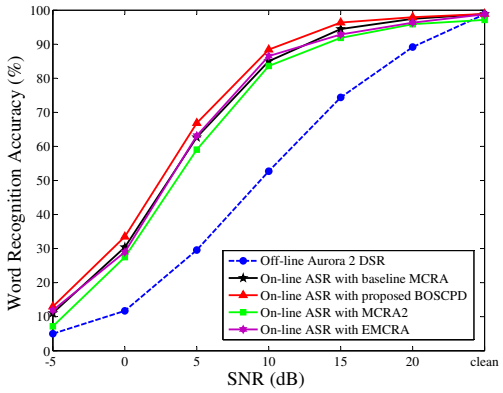
(d) Exhibition hall environments

FIGURE 5.12 – Performance of the proposed on-line ASR for the Aurora 2 test data set 'A'.

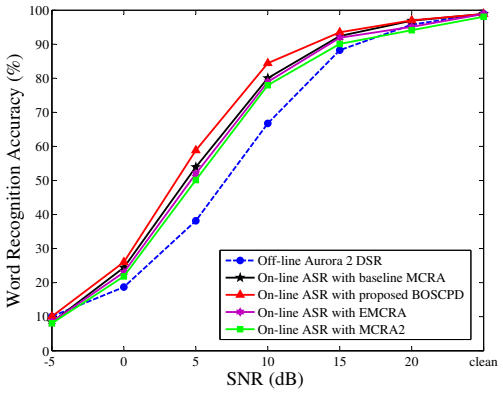
These simulation results show that at very low SNR conditions, especially at 0 dB or below, the ASR recognition accuracy is very poor due to the fact that at those SNRs, speech signals are always highly dominated by the background noise and the observed speech signals become almost noise.

5.3.5 Discussion

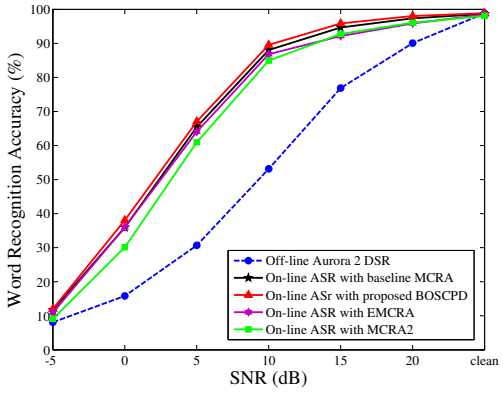
This dissertation presented an architecture of on-line ASR based on the proposed BOSCPD algorithm in rapidly varying non-stationary noises. In this architecture, we



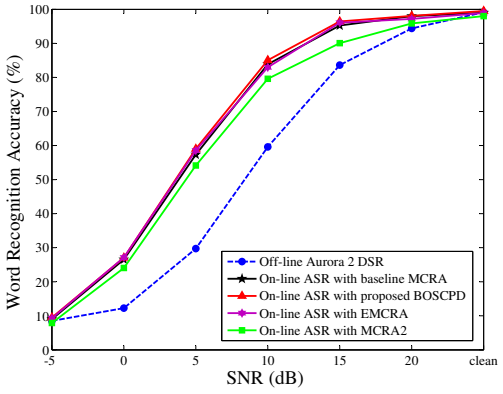
(a) Restaurant environments



(b) Street environments

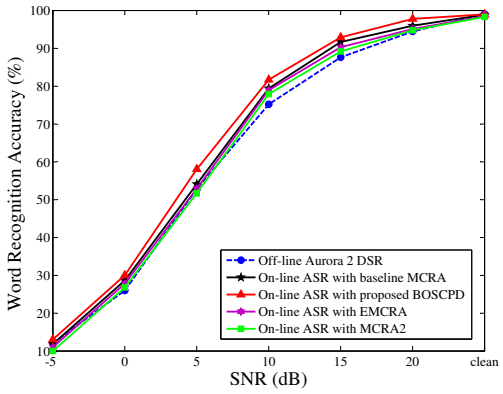


(c) Airport environments

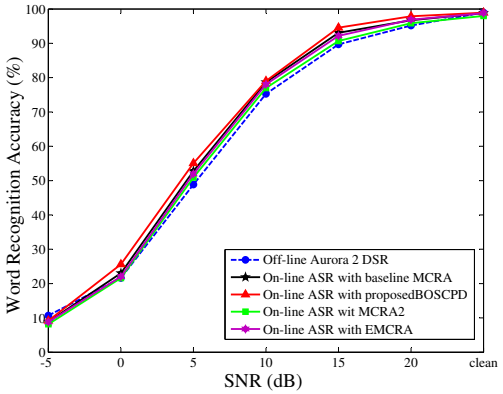


(d) Train environments

FIGURE 5.13 – Performance of the proposed on-line ASR for the Aurora 2 test data set 'B'.



(a) Subway environments



(b) Street environments

FIGURE 5.14 – Performance of the proposed on-line ASR for the Aurora 2 test data set 'C'.

address a number of unsolved issues involved in the non-stationarity of acoustic environments for robust automatic speech recognition. The main issue we have focused on is the frame dynamic non-stationary noise tracking and detecting abrupt environmental changes in real-world conditions. More specifically, we address the robustness of ASR by adopting an on-line frame dynamic joint additive and channel distortion compensation (JAC) in highly non-stationary acoustic environments in particular. From this research work, it is found that on-line ASR performs better in non-stationary noisy conditions compared to the current ASR that works in batch-mode (off-line). The thrust of the research work in this paper is to find JAC distortion compensation approaches and to integrate them for on-line ASR.

Current design criteria require ASR to work in batch-mode (off-line). In batch mode, utterance boundaries are known to the ASR decoder and it normalizes the test utterances by subtracting a global mean cepstral bias from these utterances. Current ASR also needs *a priori* information of the test conditions to improve recognition performances. The context dependency of ASR technologies limits its application in diverse fields. With the advent of fast computing and broadband communications technologies, the application areas of ASR are increasing quickly. In many applications, users of an ASR system move from one place to another randomly (e.g., 3G/4G mobile users). In most cases, these test conditions are unknown and highly non-stationary in nature.

The proposed on-line ASR architecture exploits the advantage of Bayesian on-line inference for the change point detection (BOCPD) technique. We have verified this algorithm using the Aurora 2 speech data, which demonstrated simulated real-world data sets. The proposed on-line ASR framework based on BOSCPD provides conve-

nient delineation of the implementation of the change point algorithm within the architecture of the MCRA noise tracking algorithms. From the experimental results, we found that the new state-of-the-art on-line ASR algorithm enables us to decode the test speech utterances at different SNR conditions in highly non-stationary environments. However, it needs further improvement to attain higher recognition accuracy at low SNR conditions.

5.4 PSO-Based Front-End Processing for On-Line ASR

In this section, we present the simulation results for our proposed on-line ASR based on a dynamic multi-swarm particle swarm optimization (DMS-PSO) technique. We implement the DMS-PSO as noise canceller in the front-end of our proposed on-line ASR. The experimental setup for this simulation is described in the following subsections.

5.4.1 Test Database Preparation for DMS-PSO

In order to simulate the real-time noise cancellation in the front-end of the proposed on-line ASR, we use a dual-channel soft adaptive filter as shown in Fig. 4.6. For a real-time de-noising process using DMS-PSO, the test utterances are prepared by adding non-stationary noises with the clean test utterances of test set A of the Aurora 2 speech corpora. These test utterances are blended with subway, babble, car, and exhibition hall noises from the Aurora 2 noise database. We add these noises at

TABLE 5.9 – Configuration parameters for the DMS-PSO algorithm.

	Parameter	Initial Values
DMS-PSO	Frame length in samples	256
	Frame overlapping	60%
	Swarm numbers	10
	Swarm population size	3
	Total population size	30
	Particle dimension	10
	Acceleration constants c_1,c_2	1.49445
	Upper velocity limit V_{max}	1
	Lower velocity limit V_{min}	-1
	ω_{ini}	0.95
	ω_{end}	0.4
	Grouping period	10
	Local refining period	100

SNR levels 20 dB, 15 dB, 10 dB, 5 dB, 0 dB, and -5 dB. Aurora test data sets have noisy test data at these SNR levels.

In this DMS-PSO-based adaptive filtering process, input noise n_1 is filtered with an ITU-T G.712 filter before adding it to the clean speech signal. The reference noise n_2 is correlated to input noise n_1. However, we assume that noises are uncorrelated with the speech signal.

TABLE 5.10 – Qualitative performance evaluation of the DMS-PSO algorithm for noise cancellation in non-stationary environments.

	Test Case at -5 dB SNR
Mean SegSNR (dB) at noisy conditions	-3.3783
Mean SegSNR (dB) after denoising	**11.0691**
PESQ (MOS) at noisy conditions	1.0669
PESQ (MOS) after denoising	**2.0574**

TABLE 5.11 – Qualitative performance evaluation of the NLMS algorithm for noise cancellation in non-stationary environments.

	Test Case at -5 dB SNR
Mean SegSNR (dB) at noisy conditions	-3.3783
Mean SegSNR (dB) after denoising	**5.8951**
PESQ (MOS) at noisy conditions	1.0669
PESQ (MOS) after denoising	**1.6801**

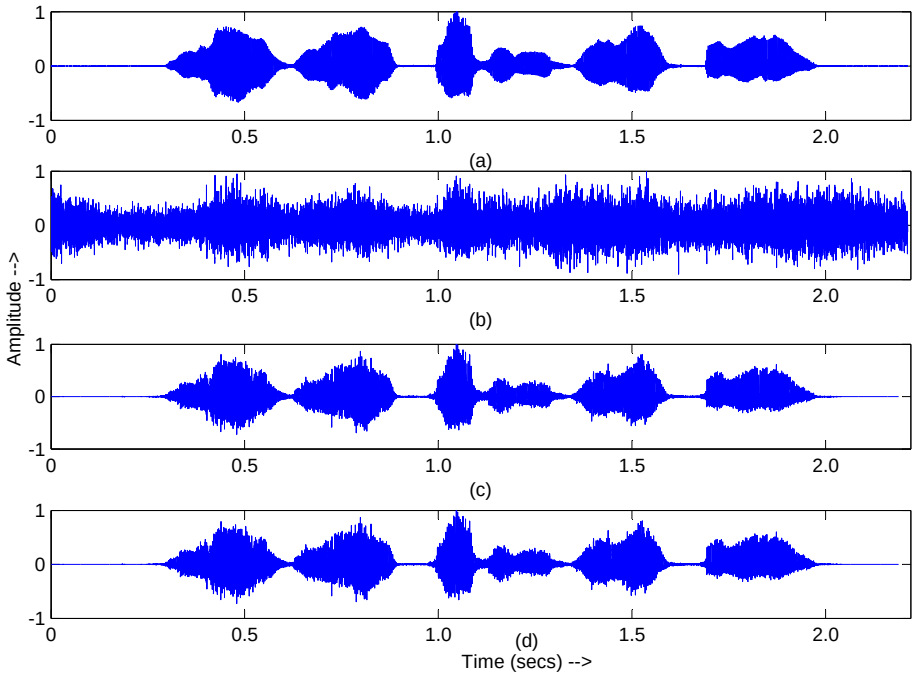


FIGURE 5.15 – From the top, (a) time waveforms of the clean signal, (b) the speech signal corrupted by the non-stationary noise at -5 dB SNR, (c) the denoised speech signal for the DMS-PSO algorithm and (d) the denoised speech signal for the NLMS algorithm.

5.4.2 Setting Configuration Parameters

In DMS-PSO-based noise cancellation in the front-end of our proposed on-line ASR, we process the input speech signal in frames. Each frame contains 256 samples (32 ms for 8 kHz sampling frequency) with 50% overlap between adjacent frames. The experimental conditions for the PSO technique are shown in Table 5.9 [5].

5.4.3 Experimental Results

In the first subsection we compare the performance of the PSO algorithm with standard NLMS [108] algorithm as an adaptive noise canceling technique in the front-end of ASR in non-stationary acoustic environments. In the second subsection, we present the word recognition performance of our proposed on-line ASR using DMS-PSO in the front-end as a dynamic noise canceller. We compare these recognition results with the Wiener-based optimal filter in the Aurora 2 front-end. In the recognition stage, we follow the same procedures as in Section 5.4.

5.4.3.1 DMS-PSO-Based Noise Reduction

The performance of the DMS-PSO algorithm compared to the NLMS [108] algorithm for adaptive non-stationary noise cancellation for a test utterance at -5 dB SNR is shown in Fig. 5.15. From the spectrograms of the test speech signal as shown in Fig. 5.16, it is clear that PSO-based evolutionary algorithms are capable of recovering speech signals even at very low SNR in non-stationary environments compared to the NLMS algorithm. The corresponding improvements in PESQ and Segmental SNR are listed in Table 5.10 and Table 5.11 for DMS-PSO and NLMS algorithms, respectively. DMS-PSO shows big improvement in the Segmental SNR and it doubles the PESQ score. These improvements clearly show that PSO in the front-end processing would play an important role in canceling the non-stationary noises and improving the SNR of the speech signals, which are the main criteria for feature-based noise compensation in current ASR techniques.

The simulation results confirm the validity of the PSO-based adaptive noise com-

pensation algorithm according to the speech communication model in Eq. A.14 and Eq. A.15. The DMS-PSO method works well even at very low SNR as shown in Fig. 5.15. The objective of our proposed DMS-PSO-based method is to increase the SNR values by compensating the additive non-stationary noise frame dynamically in the front-end and then compensating the non-stationary channel distortions in the ASR decoding stage in the back-end.

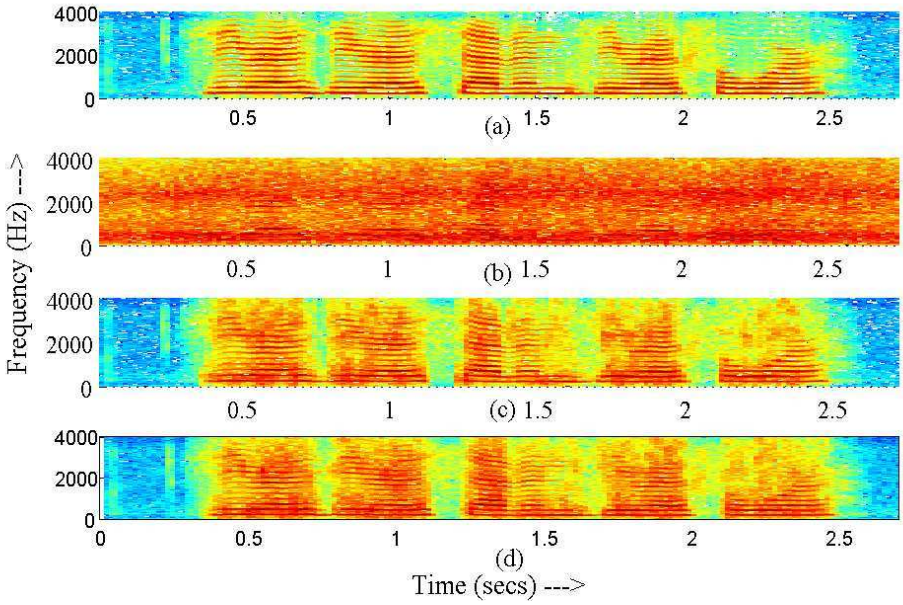


FIGURE 5.16 – Performance of the DMS-PSO for adaptive noise cancelation compared to the NLMS algorithm. From the top, (a) Spectrogram of the clean signal as shown in Figure 5.15(a), (b) spectrogram of the noisy speech signal at -5 dB SNR, (c) spectrogram of the denoised speech signal by the PSO algorithm, and (d) spectrogram of the denoised speech signal by the NLMS algorithm.

5.4.3.2 Recognition Performance of On-Line ASR using DMS-PSO

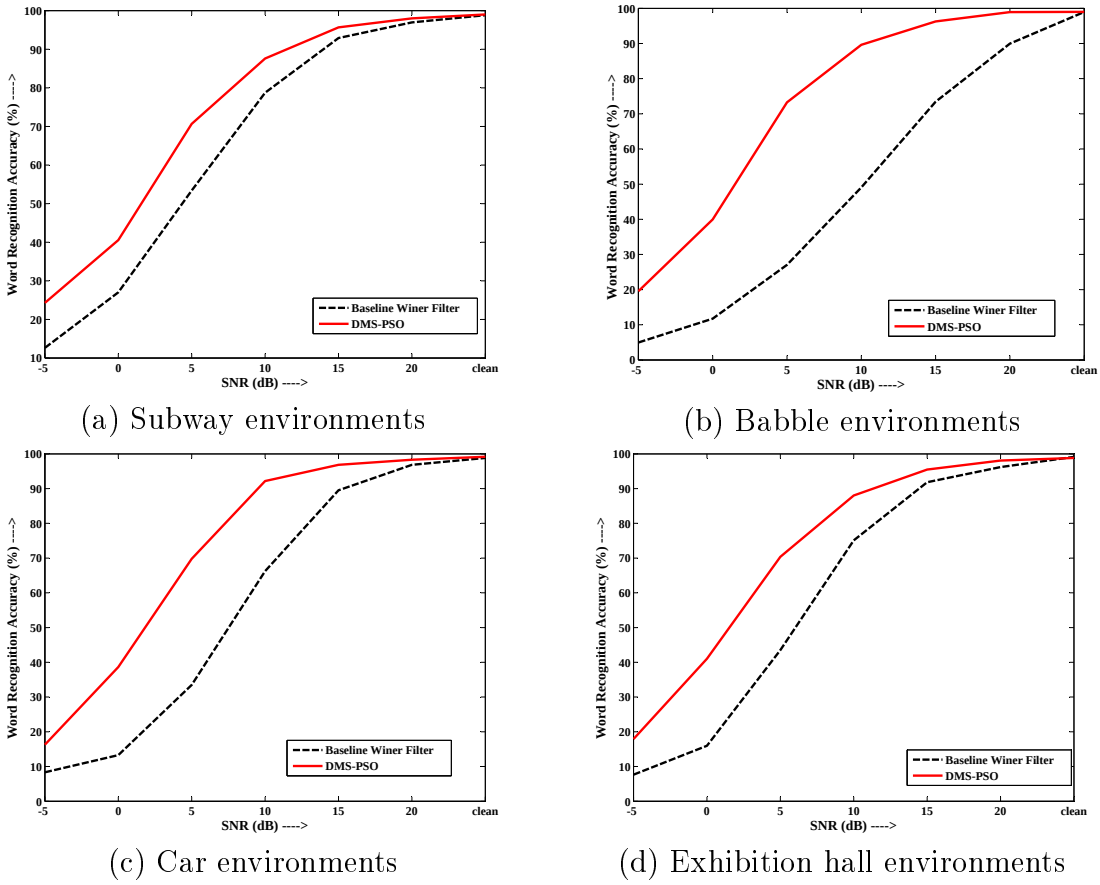
The recognition performance of the on-line ASR using the DMS-PSO optimization technique in dynamic non-stationary noise compensation in non-stationary acoustic

environments is validated by comparing its performance to the baseline Aurora 2 DSR system. The baseline Aurora 2 front-end uses Wiener-based optimal filter for adaptive compensation of acoustic noise. Aurora 2 front-end uses voice activity detector (VAD) to detect speech and non-speech parts of the speech signal. It takes the reference noise during the non-speech period of the noisy speech signal and performs the process enhancement process as a dual channel adaptive noise canceller. Since Aurora 2 front-end's Wiener-based optimal filter acts as a dual channel adaptive filter, in this dissertation, we compare our proposed DSM-PSO-based soft adaptive filter's performance with this Wiener filter.

The simulation results of the on-line ASR show that the DMS-PSO-based frame dynamic noise compensation technique increases the word recognition accuracy greatly compared to the baseline system for SNRs between clean and -5 dB. These simulations are conducted for test data set 'A' of the Aurora 2 speech corpora representing subway, babble, car, and exhibition hall environments, and these are graphically presented in Fig. 5.17. These simulation results show that at very low SNR conditions, especially at 0 dB or below, the ASR recognition accuracy is improved compared to the baseline dual channel Wiener filter. This improvement in performance at low SNR is due to the fact that DMS-PSO optimization technique is capable of modeling the optimal adaptive filter that represents the highly non-stationary acoustic environments.

5.4.4 Discussion

In this section, we propose DMS-PSO for noise reduction in the front-end of our proposed on-line ASR algorithm to improve the robustness of ASR in unknown highly



(a) Subway environments

(b) Babble environments

(c) Car environments

(d) Exhibition hall environments

FIGURE 5.17 – Performance of the DMS-PSO-based on-line ASR compared to the Wiener-based Aurora 2 front-end for the Aurora 2 test data set 'A'.

non-stationary test conditions. The PSO-based optimization technique for tracking and adapting the dynamics of non-stationarity of acoustic environments proves efficient in tracking and compensating noises. We have verified this algorithm for test data set A of the Aurora 2 speech corpora for subway, babble, car, and exhibition hall acoustic conditions. The simulation results show that DMS-PSO-based frame-dynamic adaptive noise tracking and compensation in the front-end of the proposed on-line ASR improves the recognition accuracy greatly even in low SNR conditions, as shown in Table 5.12.

The performance improvement in the DMS-PSO-based SJAC system compared to the baseline Wiener filter based Aurora 2 front-end in presented in Table 5.13. The average increments in recognition accuracy compared to the baseline system are 11.99%, 45.47%, 25.73%, and 18.68% for subway, babble, car, and exhibition hall environments respectively. The average increment in recognition accuracy for the data set 'A' is 25.47% compared to the Baseline Wiener-based Aurora 2 front-end. From our experiments, we find that the proposed DMS-PSO-based frame adaptive SJAC system for on-line ASR works significantly better than the baseline Aurora 2 DSR and needs further improvement to minimize distortions at low SNR conditions. Here DMS-PSO performs comparatively better at moderate SNR conditions, which is supported by the basic idea of the PSO-based evolutionary optimization techniques. However, we achieve these improvements in recognition accuracy at the expense of more computational cost compared to the Wiener-based optimal filter for adaptive noise cancellation.

5.5 Summary

By using the BOSCPD-based noise tracking and rapid adaptation in highly non-stationary acoustic environments and the DMS-PSO-based optimization technique to track dynamics of non-stationarity of the test conditions speech recognition, and a joint background noise and channel distortions compensation technique for on-line ASR introduced in previous chapters, a number of frame dynamic real-time ASR tests are performed for noisy speech signals and their denoised counterparts. From our experimental results, we draw conclusions as follows:

TABLE 5.12 – Recognition performance of clean-trained model for the proposed DMS-PSO-based on-line ASR for recognizing digit strings.

SNR	Subway	Babble	Car	Expo Hall
clean	98.99	98.98	99.16	98.81
20 dB	97.99	98.94	98.31	98.08
15 dB	95.65	96.28	96.85	95.49
10 dB	87.59	89.62	92.19	88.01
5 dB	70.65	73.25	66.79	70.36
0 dB	40.53	39.95	38.62	41.03
-5 dB	24.27	19.56	16.26	17.9
Average	73.67	73.80	72.60	72.81

TABLE 5.13 – Improvement (%) of recognition accuracy of clean-trained model for DMS-PSO-based on-line ASR compared to the proposed BOSCPD-based SJAC system.

SNR	Subway	Babble	Car	Expo Hall	Average
clean	0.16	0.01	0.35	-0.33	0.05
20 dB	1.06	9.98	1.52	1.95	3.63
15 dB	2.95	31.12	8.18	3.96	11.55
10 dB	11.27	82.67	39.18	17.19	37.58
5 dB	32.33	171.00	108.39	61.71	93.36
0 dB	50.11	240.58	191.03	156.76	159.62
-5 dB	92.31	294.35	94.73	133.99	153.99
Average	11.99	45.47	25.73	18.68	25.47

- BOSCPD-based joint non-stationary noise tracking and compensation algorithms for on-line ASR can help to improve ASR performances in real-time applications. Among them, the PSO-based SJAC algorithm gives the best results and can significantly improve the ASR performance of noisy speech at very low SNR conditions.
- In our proposed SJAC-based on-line ASR experiments, we show that significant improvements can be achieved by using an appropriate algorithm to instantly detect and compensate rapid changes in acoustic conditions in unknown non-stationary test environments.
- Experimental results show that our proposed BOSCPD and DMS-PSO approaches for SJAC-based noise compensation in the front-end of on-line ASR applications not only improve the absolute values of ASR accuracies, but also largely increase their dynamic ranges in all conditions. Thus these approaches can be used in real-time ASR applications where people desire high and stable ASR performance even when the surrounding environment is changed abruptly.

Chapter 6

Conclusions and Future Research

6.1 Conclusions

This dissertation has addressed a number of unsolved issues involved in the non-stationarity of acoustic environments for robust automatic speech recognition. The main issues we have focused on in this dissertation are tracking and detecting abrupt acoustic environmental changes in real-world conditions in order to improve the noise robustness of automatic speech recognition. More specifically, it addresses the robustness of ASR by adopting BOSCPD and PSO techniques for an on-line frame dynamic soft joint additive and channel distortion compensation (SJAC) in highly non-stationary acoustic environments. In this dissertation, it is found that BOSCPD-based on-line ASR performs better in non-stationary noisy conditions compared to MCRA-based on-line ASR. The PSO-based dual-channel front end processing for on-line ASR performs better than gradient-search-based dual channel Wiener filter based Aurora 2 front-end-based on-line ASR in non-stationary environments. The thrust of

the research was to develop new ideas which will show ways for developing more robust on-line ASR.

6.2 Review of Achievements

Starting with an extensive study of issues of the robustness of state-of-the-art ASR technologies, we develop ways to design and analyze the performance of noise-robust on-line ASR, with highly non-stationary noise tracking, sudden change detection in acoustic environments, joint compensation of the observed speech signal in noisy conditions. We add these functionalities to ASR front-end processing and decoding stages in order to simulate on-line ASR.

We develop a soft frame-synchronous sequential noise-bias compensation and speech recognition in noisy conditions based on a Bayesian on-line inference technique. The Bayesian on-line change point detection technique in association with classical MCRA algorithm is implemented, which can be as a soft computing technique in the back-end processing of ASR systems to work in real-time environments.

In the case of a SJAC-based speech frame-synchronous denoising technique, we not only study several popular algorithms, but also propose our own methods. We propose two techniques: (i) Bayesian on-line inference in conjunction with an MCRA noise tracking algorithms, and (ii) sequential prediction and adaptation of speech signals based on a non-stationary and non-Gaussian modeling approach using a particle swarm optimization (PSO) technique for on-line speech recognition in real-world acoustic conditions. We have shown through experiments that these two approaches can help improve the speech recognition performance of an ASR system for real-

time applications. By the integration of the PSO soft adaptive filtering algorithms, ASR performance can be improved in different noisy conditions, compared with using classical linear regression models stand-alone. The soft adaptive filters are therefore promising candidates for further automatic speech recognition studies and for practical applications, e.g., noise-robust ASR applications in mobile environments.

6.3 Future Research

In future research work, the following areas could be considered:

- The performance of speech recognition in real-world acoustic environments could be improved by extending the current research approach to take into account a more realistic acoustic model based on a non-linear and non-Gaussian modeling approach to tackle the current widely discussed non-stationarity problem for ASR.
- The Bayesian sequential prediction and adaptation along with fine refinement of current HMMs with multiple streams, normalized parameters, increased training data, and large vocabulary size could be the new direction in research for human-like environment-aware ASR.
- At present, researchers are trying to learn from behavior patterns of species in nature, e.g., birds flocking, fish schooling etc. They are very optimistic about bio-inspired solutions to give us new insights into solving performance issues of current ASR in real-world acoustic conditions. PSO is one such approach that could open a new research direction for noise robust ASR.
- The parameters of the proposed SJAC-based on-line ASR could be further op-

timized experimentally, so as to minimize speech distortions and artifacts.

Appendix A

Mathematical Model of Speech Communication

Speech Communication Model

A general model of the observed speech signal $y(t)$ considering all the noise sources independently can be described [63] as

$$y(t) = [\{([x(t)|_{Lombard}^{Stress}]_{n_1(t)} + n_1(t)) \otimes h_{mike}(t) + n_2(t)\} \otimes h_{chan}(t)] + n_2(t), \quad \text{(A.1)}$$

where $n_1(t)$ is the background noise, $h_{mike}(t)$ is the impulse response of the microphone transducers, $n_2(t)$ and $h_{chan}(t)$ are the additive noise and impulse response, respectively, of the transmission channel, and $n_3(t)$ is the noise present at the receiver. Now Eq. A.1 can be further simplified by combining the various additive noises and channel distortion into composite non-specific sources and ignoring the *Lombard* and stress noises, as shown in Fig. A.1. Under these conditions the acoustic modeling reduces to

$$y(t) = x(t) \otimes h(t) + n(t), \quad \text{(A.2)}$$

where $h(t)$ is the impulse response of a linear-time-invariant (LTI) system, $n(t)$ is additive noise, and $x(t)$ is the clean speech. In this case, the model parameters are assumed to be stationary during the course of observation.

In the discrete time Fourier transform (DTFT) domain, Eq. A.2 can be written into its equivalent form in the spectral domain without phase information as follows:

$$Y(\omega) = X(\omega)H(\omega) + N(\omega). \quad \text{(A.3)}$$

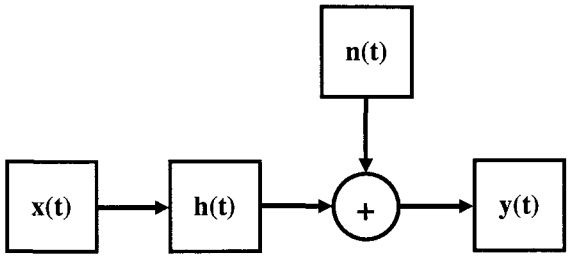


FIGURE A.1 – Speech communication model [11].

where $Y(\omega)$ is the DTFT of the observed noisy speech signal $y(t)$, $X(\omega)$ is the DTFT of the clean speech signal $x(t)$, $H(\omega)$ is the DTFT of the impulse response $h(t)$ of the LTI system, and $N(\omega)$ is the DTFT of the additive noise $n(t)$.

In the DFT domain, Eq. A.3 can be re-written as follows:

$$Y(\omega_k) = X(\omega_k)H(\omega_k) + N(\omega_k), \tag{A.4}$$

where k $\{k = 0, ..., N_w - 1\}$ is the frequency bin index, and N_w is the DFT sampling period.

Now, the magnitude spectrum of the transformed speech signal in Eq. A.4 is

$$|Y(\omega_k)| = |X(\omega_k)||H(\omega_k)| + |N(\omega_k)|. \tag{A.5}$$

In the power spectral domain, Eq. A.5 has the form as follows:

$$
\begin{aligned}
|Y(\omega_k)|^2 &= (|X(\omega_k)||H(\omega_k)| + |N(\omega_k)|)^2 \\
&= |X(\omega_k)|^2|H(\omega_k)|^2 + |N(\omega_k)|^2 \\
&+ 2|X(\omega_k)||H(\omega_k)||N(\omega_k)|\cos\theta_{\omega_k}, \qquad \text{(A.6)}
\end{aligned}
$$

where θ_{ω_k} denotes the random angle between the two complex variables $(|H(\omega_k)||X(\omega_k)|)$ and $|N(\omega_k)|$. Currently there are two approaches for this random angle θ_{ω_k} as follows:

- **case 1**: The phase information is omitted assuming that ASR performance does not depend on phase information. Therefore, the angle term $cos\theta_{\omega_k}$ is set to 0 and uses the power spectrum as the acoustic feature. Under this condition, Eq. A.6 will become

$$
|Y(\omega_k)|^2 = |X(\omega_k)|^2|H(\omega_k)|^2 + |N(\omega_k)|^2. \qquad \text{(A.7)}
$$

However, Eq. A.7 can be rewritten in the following form,

$$
P_Y(\omega_k) = P_X(\omega_k)|H(\omega_k)|^2 + P_N(\omega_k), \qquad \text{(A.8)}
$$

where $P_Y(\omega_k)$ represents the power spectra of the noisy speech observation, $P_X(\omega_k)$ represents the power spectra of the clean speech, $P_N(\omega_k)$ represents the power spectra of the noise, and $|H(\omega_k)|^2$ represents the power spectra of the channel.

– **Case 2**: In this case, $cos\theta_{\omega_k}$ is set to 1 and Eq. A.6 will become

$$|Y(\omega_k)|^2 = |X(\omega_k)|^2|H(\omega_k)|^2 + |N(\omega_k)|^2 + 2|X(\omega_k)||H(\omega_k)||N(\omega_k)|. \quad \text{(A.9)}$$

In this paper, we followed the most commonly used 1st approach, Eq. A.8, as the basis for single channel JAC compensation in feature space for the proposed on-line ASR.

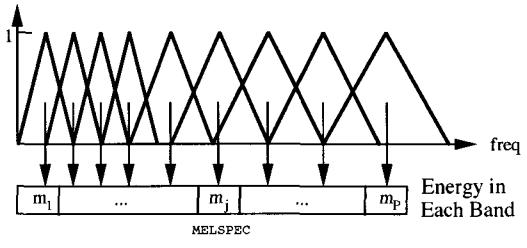


FIGURE A.2 – Mel-cepstral transformation. Here m_i represents ith Mel-spectral band.

After Mel filtering using Mel-filterbank as shown in Fig. A.2 and log transformation of Eq. A.8, we get

$$\ln P_Y(\omega_{m_m}) = \ln[P_X(\omega_{m_m})|H(\omega_{m_m})|^2 + P_N(\omega_{m_m})], \quad \text{(A.10)}$$

where $m_m = 0, ..., M_m - 1$, and M_m is the number of Mel weighting filters, and ω_{m_m} represents a particular Mel-spectral band.

Now Eq. A.10 can be further processed as follows:

$$\begin{aligned}\ln P_Y(\omega_{m_m}) &= \ln\left[P_X(\omega_{m_m})|H(\omega_{m_m})|^2\left(1+\frac{P_N(\omega_{m_m})}{P_X(\omega_{m_m})|H(\omega_{m_m})|^2}\right)\right] \\ &= \ln\left[P_X(\omega_{m_m})|H(\omega_{m_m})|^2\right]+\ln\left[1+\frac{P_N(\omega_{m_m})}{P_X(\omega_{m_m})|H(\omega_{m_m})|^2}\right] \\ &= \ln P_X(\omega_{m_m})+\ln|H(\omega_{m_m})|^2+\ln\left[1+\frac{P_N(\omega_{m_m})}{P_X(\omega_{m_m})|H(\omega_{m_m})|^2}\right] \quad \text{(A.11)}\end{aligned}$$

Taking the IDFT on both sides of Eq. A.10, we can write it in the cepstral domain as

$$\begin{aligned}\mathbf{y} &= \mathbf{x}+\mathbf{b}+IDFT\left\{\ln\left(1+e^{DFT\left[IDFT\left\{ln\left(\frac{P_N(\omega_{m_m})}{P_X(\omega_{m_m})|H(\omega_{m_m})|^2}\right)\right\}\right]}\right)\right\} \\ &= \mathbf{x}+\mathbf{b}+IDFT\left\{\ln\left(1+e^{DFT[\mathbf{n}-\mathbf{b}-\mathbf{x}]}\right)\right\} \\ &= \mathbf{x}+\mathbf{b}+\mathbf{r}(\mathbf{x},\mathbf{n},\mathbf{b}), \quad \text{(A.12)}\end{aligned}$$

where $\mathbf{y} = IDFT\{\ln P_Y(\omega_{m_m})\}$ is the observed speech signal, $\mathbf{x} = IDFT\{\ln P_X(\omega_{m_m})\}$ is the clean speech signal, $\mathbf{b} = IDFT\{\ln|H(\omega_{m_m})|^2\}$ is the channel bias, $\mathbf{n}$ is additive background noise and $\mathbf{r}(\mathbf{x},\mathbf{n},\mathbf{b})$ is a correction vector in the cepstral domain [11]. However Eq. A.10 can be simplified differently as follows:

$$\begin{aligned}\ln P_Y(\omega_{m_m}) &= \ln\left[P_N(\omega_{m_m})\left(1+\frac{P_X(\omega_{m_m})|H(\omega_{m_m})|^2}{P_N(\omega_{m_m})}\right)\right] \\ &= \ln P_N(\omega_{m_m})+\ln\left[1+\frac{P_X(\omega_{m_m})|H(\omega_{m_m})|^2}{P_N(\omega_{m_m})}\right], \quad \text{(A.13)}\end{aligned}$$

Taking the IDFT on both sides of Eq. A.13, we can write it in the cepstral domain as

$$\begin{aligned} \mathbf{y} &= \mathbf{n} + IDFT\left\{\ln(1 + e^{DFT[\mathbf{x}+\mathbf{b}-\mathbf{n}]})\right\} \\ &= \mathbf{n} + \mathbf{s}(\mathbf{x}, \mathbf{n}, \mathbf{b}). \end{aligned} \tag{A.14}$$

where $\mathbf{n} = IDFT\{\ln P_N(\omega_{m_m})\}$ is the additive background noise and $\mathbf{s}(\mathbf{x}, \mathbf{n}, \mathbf{b})$ is a correction vector in the cepstral domain [11].

Since the acoustic model in Eq. A.12 contains a highly non-linear term, the noise compensation algorithms used a simplified form of the acoustic model based on some assumptions as follows:

- for fairly stationary environments and at high SNR
  - $(\mathbf{x} + \mathbf{b}) >> \mathbf{n}$, and
  - $\mathbf{r}(\mathbf{x}, \mathbf{n}, \mathbf{b}) \approx \mathbf{0}$
- as a result Eq. A.12 in the cepstral domain reduces to the following simple form [11]

$$\mathbf{y} \approx \mathbf{x} + \mathbf{b}. \tag{A.15}$$

Appendix B

Implementation of On-Line ASR

B.1 On-Line ASR using ATK

HTK, the speech recognition platform from [15], has long been used in most of the research labs in academic premises around the world for the simulation of off-line ASR. However, for real-time applications, especially for smart-phones-based on-line speech recognition, it is essential to use a speech recognition engine developed based on multi-threaded programming architecture. Multi-threaded programming is used for embedded technology for design and developed real-time systems. ATK, a real-time API for HTK, is developed in [2] to meet these requirements to simulate real-time speech recognition systems. We simulate our proposed on-line ASR algorithms using ATK, which is described briefly in the following subsection.

B.1.1 On-Line ASR Architecture

The proposed on-line ASR infrastructure based on ATK [2] consists of a variety of components connected together as shown in Fig. B.1. The functionalities of the main components are described as follows:

- **Packet**: It is a chunk of information. Packets are used for transmitting a variety of information between asynchronously executing components. In particular, packets are used to convey various forms of user input and output signals (speech, event markers such as mouse clicks, etc). In these cases, each packet has a time stamp to define the temporal span to which it relates. The types of data that a packet can carry include text strings, waveform fragments, coded feature vectors, word labels and semantic tags.
- **Buffer**: This is a first-in first-out (FIFO) packet queue. Buffers provide the

channel for passing packets from one component to another. Buffers can be of fixed size or unlimited size. Components wishing to access a buffer can test to see whether the buffer operation would block before committing to the operation.

- **Component**: It is a processing element. Each component is executed within its own individual thread. Components communicate by passing packets via buffers. In addition, components have a command interface that can be used to update control parameters during operation and thereby modify the run-time behavior of the component.

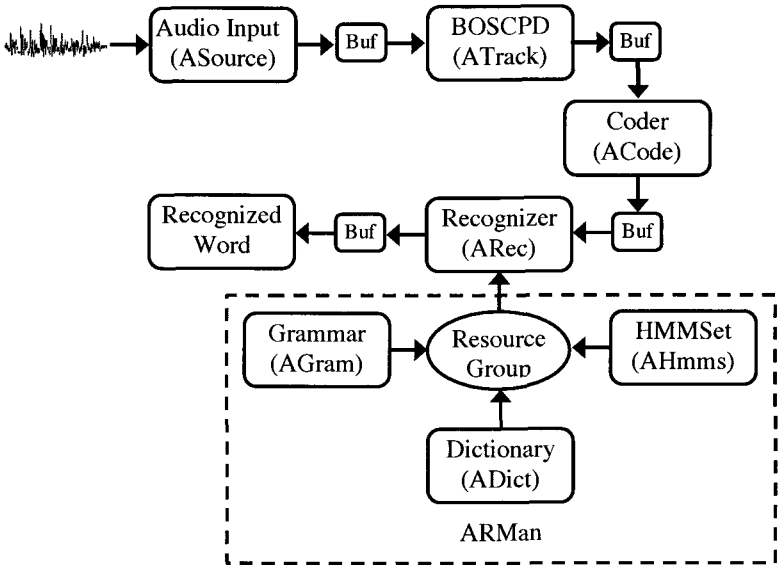


FIGURE B.1 – Basic Architecture for On-Line Recognition System

B.1.2 On-Line Digit Recognition

In ATK, the three most required resources, e.g., i) dictionary, ii) grammar, and iii) HMMs of clean speech, need to be prepared in batch-mode using HTK. For the test

phase, ATK provides on-line decoding using HVite instead of HDecode in HTK. Each of the three required resources can be defined as entries in a configuration file, which is loaded at start-up time. Such a file will also typically contain the specification of the coding parameters.

B.1.3 Frame Dynamic Recognition

The ARec component in ATK as shown in Fig. B.1 provides similar functionality to the standard HTK Viterbi decoder. It also provides tri-gram language model support, which is not available in HTK. ARec is supplied with a resource group containing the required HMM Sets, dictionary, grammar, and optionally an n-gram language model. It then decodes incoming feature vectors accordingly.

In operation, the ATK on-line recognizer always remains in one of five possible states as indicated by the state diagram shown in Fig. B.2. The recognizer changes state, depending on the settings of the operating modes. The ARec display shows the current mode as a sequence of 4 characters: representing the settings for CYCLE (1=oneshot, C=continuous), FLUSH (I=immed, M=tomark, S=tospeech), STOP (I=immed, M=tomark, S=tosilence), and RESULTS (I=immed, A=asap, E=atend, X=all).

On creation, an ARec object is placed in the WAIT state. When in the WAIT state, the recognizer waits for a Start() command to be issued via its command interface. When this Start() command is received, the recognizer moves to the PRIME state in which it loads the recognition resources specified by the current resource group. It then moves immediately to the FLUSH state where it takes packets from its input buffer

and discards them until it is ready to start recognizing as determined by the flush mode setting. This can happen either immediately, when a START marker is received, or as soon as the incoming observation packet has a frame marked as speech. Once in the RUN state, the recognizer recognizes incoming packets until either a STOP marker is received, a speech frame marked as silence is received or a Stop() command is issued. In the ANS state, the recognizer cleans up the recognition processing and returns to either the WAIT or PRIME states depending on the setting of the CYCLE mode. A more detailed description of this recognition process can be found in [2].

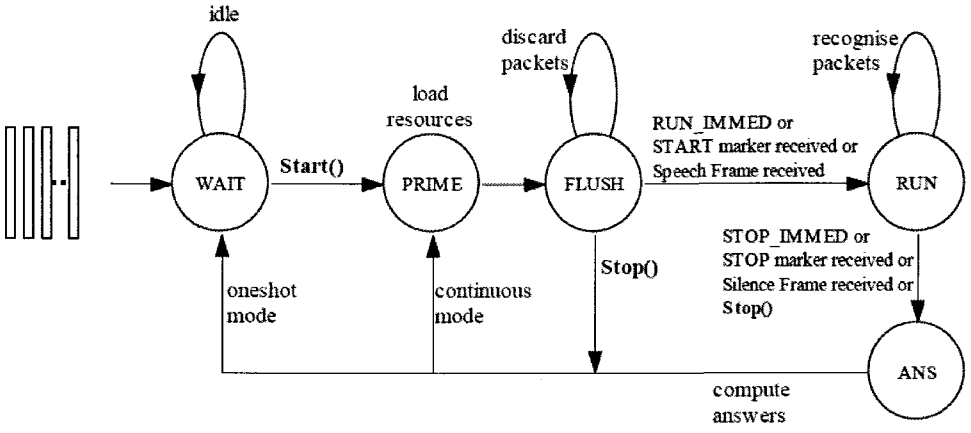


FIGURE B.2 – ATK Recognizer State Transition Diagram [2]

B.1.4 Confidence Scoring

ARec supports a simple method of confidence scoring. Every frame, the acoustic log likelihood (i.e., acoustic score) of the best matching model state and the best matching background model state are saved. When a word is recognized, these *best-*

state and *background state* scores are summed to form a best-possible-acoustic score (bs) and a background score (bg) over the segment of the waveform for which the word is being hypothesized. A raw confidence score in the range -1 to 1 is then computed as

$$rawconf = ac - \frac{(bs + bg)}{2}, \tag{B.1}$$

where ac is the actual acoustic log likelihood of the word. The confidence for that word is then computed as

$$conf = \frac{e^{x_{scs}}}{e^{x_{scs}} + e^{-x_{scs}}}, \tag{B.2}$$

where x_{scs} is the scaled *rawconf* score

$$x_{scs} = \alpha_{scc}.rawconf - \beta_{op}. \tag{B.3}$$

The constant α_{scc} sets the slope of the confidence curve and β_{op} sets the operating point. Their values are set by the configuration parameters CONFSCALE and CONFOFFSET with default values of 0.15 and 0.0, respectively.

The background model is usually stored in a separate HMM definition whose name is determined by setting the configuration variable CONFBGHMM. Once loaded, this HMM is used to compute the background state probability. It might be noted that the transition matrix of the background HMM is completely ignored in this process. Instead, the probability of the current speech vector is computed for each state of the

background HMM, and the maximum log probability is used as the background state score. If no background model is loaded, the average score across all model states is used as a surrogate [2].

B.1.5 Dictionary

The ADict class is derived from the abstract Resource class and an instance of the ADict class is used to store a pronunciation dictionary. Logically a pronunciation dictionary can be viewed as a list of word entries where each word entry contains the orthography for the word and a list of pronunciations. Each pronunciation consists of a list of phones, a probability and an output symbol. The latter is optional but if present, the recognition output will use the output symbol rather than the word itself.

A dictionary can be created empty and then filled via programmed actions, or more commonly, it is loaded from an external file. In either case, a loaded dictionary can be edited by adding/deleting words and changing the pronunciations of existing words.

B.1.6 Configuration Parameters

An example configuration parameters file for ATK-based ASR system is shown below:

```
# Configuration file for Aurora 2-based digit on-line recognizer
#
TARGETKIND      = MFCC_0_D_A_Z
SOURCEFORMAT    = WAV
```

```
HNET:TRACE       = 2

TARGETRATE       = 100000.0

SAVECOMPRESSED   = F

SAVEWITHCRC      = F

WINDOWSIZE       = 250000.0

USEHAMMING       = T

ENORMALISE       = F

ZMEANSOURCE      = F

PREEMCOEF        = 0.97

NUMCHANS         = 26

CEPLIFTER        = 22

NUMCEPS          = 12

SOURCERATE       = 1250

LOFREQ           = 64

HIFREQ           = 4000


HMMSET: HMMLIST = "HMMS.list"

HMMSET: MMF0    = "HMMS.mmf"

ADICT: DICTFILE = "dialer.dct"

AGRAM: GRAMFILE = "dialer.net"


SILFLOOR         = -10.0

USEPOWER         = F


HPARM: CMNDEFAULT      = "./cepmean_ubm"

HPARM: CMNTCONST       = 0.980

HPARM: CMNRESETONSTOP = F

HPARM: CMNMINFRAMES    = 1
```

```
HPARM: TRACE          = 0100

SILDISCARD      = 0
SPEECHTHRESH    = 0
SPCGLCHCOUNT    = 0
SILSEQCOUNT     = 0
SPCSEQCOUNT     =0

ACODE: DISPSHOW = F

HSIGP: TRACE    = 0

HREC:TRACE      = 0
HREC:FORCEOUT   = T
HREC:TRACEDELAY = 0
HREC:CONFSCALE  = 1.0
HREC:CONFOFFSET = 0
HREC:CONFBGHMM  = " "

AIN: TRACE      = 0
AREC: NTOKS     = 0
AREC: LMSCALE   = 5.0 # Grammar scale factor -s in HTK
AREC: NGSCALE   = 0
AREC: WORDPEN   = -17.0
AREC: GENBEAM   = 235.0
AREC: WORDBEAM  = 210.0
AREC: NBEAM     = 235.0
AREC: TRACE     = 0
```

B.1.7 Front End Processing for On-Line ASR

The proposed soft on-line ASR in this work is based on classical context-independent 11-digit HMMs using 16 active states and 6 Gaussian mixtures per state [1]. In the feature analysis, a pre-emphasis coefficient of 0.97, a Hamming window of 25 ms, a frame shift of 10 ms, and 26 Mel-scale filters covering from 64 Hz to 4000 Hz are used in the configuration file, as mentioned in subsection B.1.2. Thirteen cepstral features ($C0$-$C12$) were calculated in combination with 13 Δ and 13 Δ-Δcepstral features. The cepstral mean is subtracted from the trained HMMs during the training period. Fig. B.3 shows the front-end block diagram and the algorithms used in our experiments. In the test experiments, continuous digit accuracy was evaluated based on an ATK real-time ASR simulation process [2].

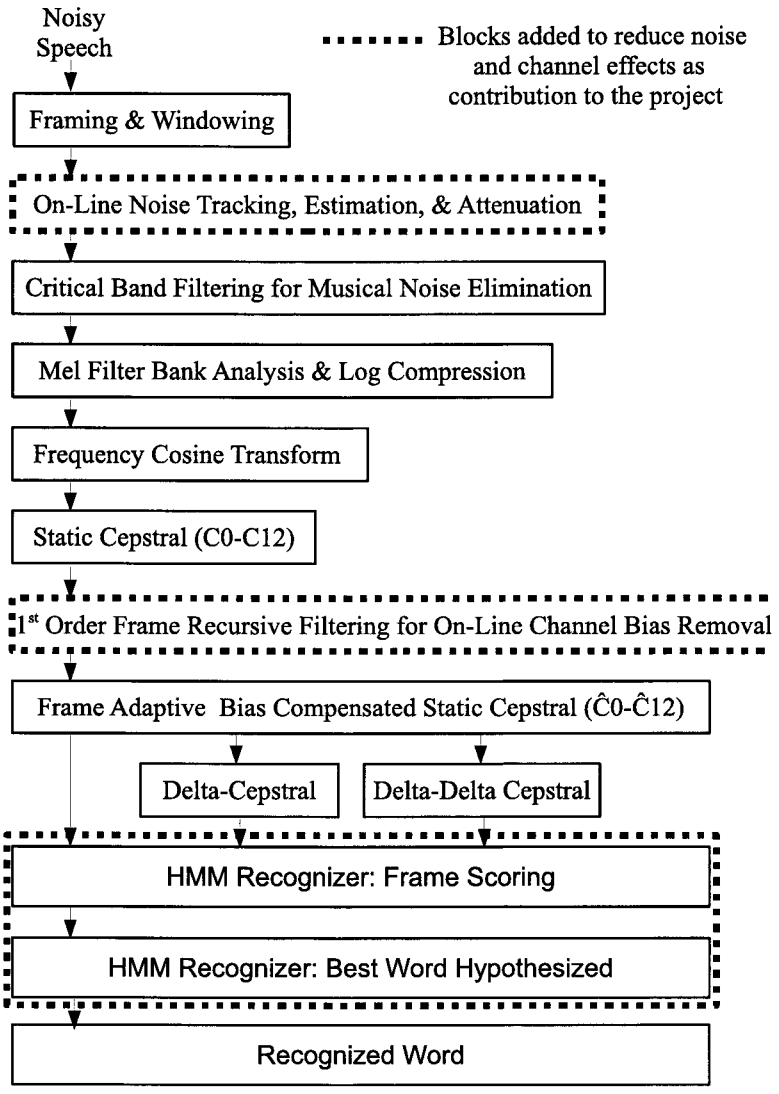


FIGURE B.3 – Front-end for on-line ASR to compensate noise and channel distortions

Appendix C

Bayesian Inference

C.1 Bayesian Inference for the Gaussian Process

The maximum likelihood technique provides point estimates of model parameters, e.g., the mean μ and the variance Σ. The Bayesian inference technique provides a treatment of this problem by introducing prior distributions over these parameters. There are three instances of inferring model parameters, which are discussed for a single Gaussian random variable as follows:

Case I: To infer the mean when the variance is known

For inferring the mean μ given a sequence of N_g observations $\mathbf{X} = \{x_1, x_2, x_3, ..., x_{N_g}\}$ for a single Gaussian random variable x with known variance σ^2, the posterior distribution [109] is

$$p(\mu|X) \propto p(X|\mu)p(\mu), \tag{C.1}$$

where the likelihood function $p(X|\mu)$ can be written as follows [109]:

$$\begin{aligned} p(X|\mu) &= \prod_{n_g=1}^{N_g} p(x_{n_g}|\mu), \\ &= \frac{1}{(2\pi\sigma^2)^{N_g/2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{n_g=1}^{N_g}\left(x_{n_g}-\mu\right)^2\right\}, \end{aligned} \tag{C.2}$$

Now if the prior $p(\mu)$ is given by a Gaussian as follows:

$$p(\mu) = \mathcal{N}\left(\mu|\mu_0, \sigma_0^2\right). \tag{C.3}$$

In this case, it is a conjugate distribution for this likelihood function because the

posterior function $p(\mu|X)$ is a product of two exponentials of quadratic functions of μ and hence it will also be Gaussian.

Under these conditions, the posterior $p(\mu|X)$ is

$$p(\mu|X) = \mathcal{N}\left(\mu|\mu_{N_g}, \sigma^2_{N_g}\right), \tag{C.4}$$

where

$$\mu_{N_g} = \frac{\sigma^2}{N_g\sigma_0^2 + \sigma^2}\mu_0 + \frac{N_g\sigma_0^2}{N_g\sigma_0^2 + \sigma^2}\mu_{ML}, \tag{C.5}$$

$$\frac{1}{\sigma^2_{N_g}} = \frac{1}{\sigma_0^2} + \frac{N_g}{\sigma^2}, \tag{C.6}$$

in which μ_{ML} is the maximum likelihood solution for μ given by the sample mean [109]

$$\mu_{ML} = \frac{1}{N_g}\sum_{n_g=1} N_g x_{n_g}. \tag{C.7}$$

The posterior mean in Eq. C.5 is a compromise between the prior mean μ_0 and the maximum likelihood solution μ_{ML}. For $N_g = 0$, Eq. C.5 reduces to the prior mean, and for $N_g \to \infty$ the posterior mean μ_{N_g} is given by the maximum likelihood solution.

The inverse variance is called the precision. The precision of the posterior in Eq. C.6 is the precision of the prior plus one contribution of the data precision from each of the observed data points. If N_g increases the precision λ_{N_g} steadily increases, corresponding to a posterior distribution with steadily decreasing variance. When

$N_g = 0$, the posterior precision reduces to the prior precision $\lambda_0 \equiv 1/\sigma_0^2$. Similarly, when $N_g \to \infty$ the posterior variance goes to zero and the posterior distribution becomes infinitely peaked around the maximum likelihood solution [109].

Case II: To infer the variance when the mean is known

In this case, it is customary to choose a conjugate form for the prior $p(\lambda)$ for simplicity [109]. Now the likelihood function $p(X|\lambda)$ can be written

$$p(X|\lambda) = \prod_{n_g=1}^{N_g} p(x_{n_g}|\lambda^{-1}) \propto \lambda^{N_g/2} \exp\left\{-\frac{\lambda}{2}\sum_{n_g=1}^{N_g}\left(x_{n_g}-\mu\right)^2\right\}. \tag{C.8}$$

The corresponding conjugate prior should therefore be proportional to the product of a power of λ and exponential of a linear function of λ. This corresponds to a gamma distribution that is defined by

$$\mathrm{Gam}(\lambda|a,b) = \frac{1}{\Gamma(a)} b^a \lambda^{a-1} e^{-b\lambda}. \tag{C.9}$$

Now the prior $p(\lambda)$ is a gamma distribution $\mathrm{Gamm}(\lambda|a_0,b_0)$ and the posterior $p(\lambda|X)$ is

$$p(\lambda|X) \propto \lambda^{a_0-1}\lambda^{N_g/2} \exp\left\{-b_0\lambda - \frac{\lambda}{2}\sum_{n_g=1}^{N_g}\left(x_{n_g}-\mu\right)^2\right\}, \tag{C.10}$$

which shows that the posterior $p(\lambda|X)$ is also a gamma distribution of the form $\mathrm{Gamm}(\lambda|a_{N_G},b_{N_g})$, where

$$a_{N_g}=a_0+\frac{N_g}{2}, \tag{C.11}$$

$$\begin{aligned} b_{N_g}&=b_0+\frac{1}{2}\sum_{n_g=1}^{N_g}\left(x_{n_g}-\mu\right)^2 \\ &=b_0+\frac{N_g}{2}\sigma_{ML}^2, \end{aligned} \tag{C.12}$$

where σ_{ML}^2 is the maximum likelihood estimate of the variance. For the case of variance, the prior distribution will be an inverse gamma distribution.

Case III: To infer the precision and the mean when both of them are unknown

Now the likelihood function $p(X|\mu,\lambda)$ has functional dependence on μ and λ, and it can be written [109] as

$$p(X|\mu,\lambda)=\prod_{n_g=1}^{N_g}\left(\frac{\lambda}{2\pi}\right)^{1/2}\exp\left\{-\frac{1}{2}\left(x_{n_g}-\mu\right)^2\right\}, \tag{C.13}$$

which can be written in a simplified form as follows:

$$p(X|\mu,\lambda)\propto\left[\lambda^{1/2}\exp\left(-\frac{\lambda\mu^2}{2}\right)\right]^{N_g}\exp\left\{\lambda\mu\sum_{n_g}^{N_g}x_{n_g}-\frac{\lambda}{2}\sum_{n_g}^{N_g}x_{n_g}^2\right\}. \tag{C.14}$$

Now the prior $p(\mu,\lambda)$ which must have the same dependence on μ and λ as the likelihood function $p(X|\mu,\lambda)$ can be expressed as

$$p(\mu,\lambda)\propto\left[\lambda^{1/2}\exp\left(-\frac{\lambda\mu^2}{2}\right)\right]^{\beta}\exp\left\{c_g\lambda\mu-d_g\lambda\right\}, \tag{C.15}$$

which can be rewritten as

$$p(\mu,\lambda) \propto \exp\left\{-\frac{\beta\lambda}{2}\left(\mu - c_g/\beta\right)^2\right\}\lambda^{\beta/2}\exp\left\{-\left(d_g - \frac{c_g^2}{2\beta}\right)\lambda\right\}, \tag{C.16}$$

where c_g , d_g , β are constants. The prior $p(\mu,\lambda)$ can be written as

$$p(\mu,\lambda) = p(\mu|\lambda)p(\lambda). \tag{C.17}$$

Comparing Eq. C.16 and Eq. C.17, we find that $p(\mu|\lambda)$ is a Gaussian whose precision is a linear function of λ, and $p(|lambda)$ is a gamma distribution. Now the normalized prior takes the form

$$p(\mu,\lambda) = \mathcal{N}\left(\mu|\mu_0,(\beta\lambda)^{-1}\right)\mathrm{Gamm}\left(\lambda|a,b\right), \tag{C.18}$$

where $\mu_0 = c_g/\beta$, $a = 1 + \beta/2$, and $b = d_g - c_g^2/2\beta$. This distribution is called the normal-gamma or Gaussian-gamma distribution [109].

References

[1] H.-G. Hirsch, and D. Pearce. The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions. In *Proc., 6th International Conference on Spoken Language Processing (ICSLP 2000)*, pages 29–32, Beijing, China, October 2000.

[2] S. Young. *ATK Real-Time API for HTK, ver 1.6.* Machine Intelligence Laboratory, Cambridge University, University of Cambridge, UK, June 2007.

[3] ETSI. Speech processing, transmission and quality aspects (STQ); distributed speech recognition; front-end feature extraction algorithm; compression algorithm; ETSI ES 201 108, v1.1.1(2000-02). Technical report, ETSI, 2000.

[4] I. Cohen. Noise estimation by minima controlled recursive averaging for robust speech enhancement. *IEEE Signal Processing Letters*, 9(1):12–15, January 2002.

[5] J. J. Liang, and P. N. Suganthan. Dynamic multi-swarm particle swarm optimizer with local search. In *IEEE Congress on Evolutionary Computation*, volume 1, pages 522–528, Edinburgh, UK, September 2005.

[6] R. P. Lippmann. Speech recognition by machines and humans. *Speech Communication*, 22(1):1–15, July 1997.

[7] V. Barreaud, I. Illina, and D. Fohr. On-line stochastic matching compensation for non-stationary noise. *Computer Speech and Language*, 22(3):207–229, July 2008.

[8] J. C. Junqua and J. Haton. *Robustness in Automatic Speech Recognition: Fundamentals and Applications.* Kluwer Academic Publishers, 1996.

[9] X. Huang, A. Acero, and H. W. Hon. *Spoken Language Processing - A Guide to Theory, Algorithm, and System Development.* Prentice Hall, 2001.

[10] L. C. W. Pols. Flexible human speech recognition. In *Proc., IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 273–283, Santa Barbara, CA, USA, December 1997.

[11] A. Acero. *Acoustical and Environmental Robustness in Automatic Speech Recognition.* Kluwer Academic Publisher, 1993.

[12] M. Akbacak and J. H. L. Hansen. Environmental sniffing: Noise knowledge estimation for robust speech systems. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(2):465–477, February 2007.

[13] D. O'Shaughnessy. *Speech Communications: Human and Machine.* IEEE Press, 1999.

[14] L. Rabiner, and B. H. Juang. *Fundamentals of Speech Recognition.* Prentice Hall, 1993.

[15] S. Young. *HTK Book, ver 3.4.* Machine Intelligence Laboratory, Cambridge University, University of Cambridge, UK, March 2009.

[16] V. Krishnamurthy, and J. B. Moore. On-line estimation of hidden markov model parameters based on the kullback-leibler information measure. *IEEE Transactions on Signal Processing*, 41(8):2557–2573, August 1993.

[17] G. Mongillo. Online learning with hidden Markov models. *Neural Computation*, 20(7):1706–1716, July 2008.

[18] M. F. R. Chowdhury, S.-A. Selouani, and D. O'Shaughnessy. A study on bias-based speech signal conditioning techniques for improving the robustness of automatic speech recognition. In *Proc., 22nd IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'09)*, pages 664–669, St John's, Newfoundland and Labrador, Canada, May 2009.

[19] P. C. Loizou. *Speech Enhancement: Theory and Practice*. CRC Press, 2007.

[20] R. P. Adams, and D. J. C. MacKay. Bayesian online changepoint detection. Technical report, University of Cambridge, Cambridge, UK, 2007. Report.arXiv:0710.3742v1 [stat.ML].

[21] R. C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proc. 6th Symposium on Micro Machine and Human Science*, pages 39–43, Piscataway, NJ, USA, 1995. IEEE Service Center.

[22] K. E. Parsopoulos, and M. N. Vrahatis. *Particle Swarm Optimization and Intelligence: Advances and Applications*. Information Science Reference (an imprint of IGI Global), Hershey, PA 17033, USA, 2010.

[23] R. Turner. Bayesian change point detection for satellite fault prediction. In *Proc., Interdisciplinary Graduate Conference (IGC)*, pages 213–221, Cambridge, UK, June 2010.

[24] S.-A. Selouani. *Speech Processing and Soft Computing.* Springer, 2011.

[25] L. A. Zadeh. What is soft computing? In *Proc. Soft Computing*, 1997.

[26] L. A. Zadeh. Fuzzy logic, neural networks, and soft computing. *Communications of the ACM*, 37(3):77–84, March 1994.

[27] L. A. Zadeh. The Berkeley Initiative in Soft Computing (BISC. `http://www.cs.berkeley.edu/~zadeh/`.

[28] N. K. Sinha, and M. M. Gupta. *Soft Computing and Intelligent Systems: Theory and Applications.* Academic Press, USA, 2000.

[29] A. Yardimci. Soft computing in medicine. *Applied Soft Computing*, 9:1029–1043, March 2009.

[30] S. J. Ovaska, A. Kamiya, and Y. Chen. Fusion of soft computing and hard computing: Computational structures and characteristic features. *IEEE Transactions on Systems, Man, and Cybernetics. Part C: Applications and Reviews*, 36(3):439–448, May 2006.

[31] N. J. Randon, J. Lawry, and I. D. Cluckie. Online learning for fuzzy Bayesian prediction. In *Proc. International Conference on Soft Methods in Probability and Statistics (SMPS 2006): Advances in Soft Computing*, volume 6, pages 405–412, September 2006.

[32] V. Tyagi. *Novel Speech Processing Techniques for Robust Speech Recognition.* PhD thesis, École Polytechnique Fédérale de Lausanne, France, 2006.

[33] J. B. Allen. How do humans process and recognize speech? *IEEE Transactions on Speech and Audio Processing*, 2(4):567–577, 1994.

[34] R. N. V. Sitaram, and T. Sreenivas. Incorporating phonetic properties in hidden Markov models for speech recognition. *J. Acoust. Soc. Am.*, 102(2):1149–1158, 1997.

[35] H. J. M. Steeneken. *On Measuring and Predicting Speech Intelligibility*. PhD thesis, University of Amsterdam, 1992.

[36] L. C. W. Pols. Three-mode principal component analysis of confusion matrices, based on the identification of Dutch consonants, under various conditions of noise and reverberation. *Speech Communication*, 2(4):275–293, 1983.

[37] J. N. V. Dijkhuizen, P. C. Anema, and R. Plomp. The effect of varying the slope of the amplitude-frequency response on the masked speech-reception threshold of sentences. *J. Acoust. Soc. Am.*, 81(2):465–469, 1987.

[38] H. Hermansky, B. Hanson, and H. Wakita. Perceptually based linear predictive analysis of speech. In *Proc. IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'85)*, pages 509–512, April 1985.

[39] L. Lisker. Rapid vs rabid: A catalogue of acoustic features that may cue the distinction). Technical report, Haskins Labs, 1978. Status Report on Speech Reserach SR-54, pp. 127-132.

[40] D. S. Pallett, J. G. Fiscus, W. M. Fisher, J. S. Garofolo, B. S. Lund, A. Martin, and M. A. Przybocki. 1994 Benchmark tests for the ARPA spoken language program. In *Proc. ARPA Spoken Language System Technology Workshop*, pages 5–36, Austin, TX, USA, 1995.

[41] A. Kannan, and M. Ostendorf. Modeling dependency in adaptation of acoustic models using multiscale tree processes. In *Proc. Eurospeech'97*, volume 4, pages

1863–1866, 1997.

[42] A. G. Samuel. Phonemic restoration: Insights from a new methodology. *Journal of Experimental Psychology*, General 110:474–494, 1981.

[43] A. van Wieringen, and L. C. W. Pols. Discrimination of single and complex consonant-vowel- and vowel-consonant-like formant transitions. *J. Acoust. Soc. Am.*, 98(3):1304–1312, 1995.

[44] L. C. W. Pols, and R. J. J. H. van Son. Acoustics and perception of dynamic vowel segments. *J. Speech Communication*, 13:135–147, 1993.

[45] R. J. J. H. van Son, and L. C. W. Pols. An acoustic profile of consonant reduction. In *Proc. ICSLP'96*, volume 3, pages 1529–1532, Philadelphia, PA, USA, 1996.

[46] D. van Bergem. *Acoustic and Lexical Vowel Reduction*. PhD thesis, University of Amsterdam, 1995.

[47] E. P. Giachin, A. E. Rosenberg, and C.-H. Lee. Word juncture modeling using phonological rules for HMM-based continuous speech recognition. *Computer Speech and Language*, 5:155–168, 1991.

[48] A. M. Liberman, and I. G. Mattingly. The motor theory of speech perception revised. *J. Cognition*, 21(1):1–36, 1985.

[49] K. N. Stevens. Toward a model for speech recognition. *J. Acoust. Soc. Am.*, 32(3):47–55, 1960.

[50] K. N. Stevens. On the quantal nature of speech. *J. of Phonetics*, 17:3–45, 1989.

[51] J. Morton. Interaction of information in word recognition. *J. Psychological Review*, 76:165–178, 1969.

[52] W. D. Marslen-Wilson, and A. Welsh. Processing interactions and lexical access during word recognition in continuous speech. *J. Cognitive Psychology*, 10:29–63, 1978.

[53] D. H. Klatt. Speech perception: A model of acoustic-phonetic analysis and lexical access. *J. Cognitive Psychology*, 7:279–312, 1979.

[54] S. M. Marcus. ERIS - Context sensitive coding in speech perception. *J. of Phonetics*, 9:197–220, 1981.

[55] K. I. Forster. Accessing the mental lexicon. *In: R.J. Wales and E. Walker (Eds.), New approaches to language mechanisms*, pages 257–287, 1976.

[56] D. J. Foss, and M. A. Blank. Identifying the speech codes. *J. Cognitive Psychology*, 12:1–31, 1980.

[57] J. L. McClelland, and J. L. Elman. The trace model of speech perception. *J. Cognitive Psychology*, 18:1–86, 1986.

[58] D. Norris. Shortlist: A connectionist model of continuous speech recognition. *J. Cognition*, 52:189–234, 1994.

[59] S. Grossberg. *Pattern Recognition by Humans and Machines*. Academic Press, Inc., Orlando, USA, 1986. Vol. I, Speech Perception.

[60] X. Wang. *Incorporating knowledge on segmental duration in HMM-based continuous speech recognition.* Institute for Functional Research into Language and Language Use (IFOTT), 1997.

[61] E. Marcheret, V. Libal, and G. Potamianos. Dynamic stream weight modeling for audio-visual speech recognition. In *Proc. IEEE International Conference on*

Acoustic, Speech, and Signal Processing (ICASSP'07), volume 4, pages 945–948, April 2007.

[62] A. Sankar, and C.-H. Lee. A maximum-likelihood approach to stochastic matching for robust speech recognition. *IEEE Transactions on Speech and Audio Processing*, 4(3):190–202, May 1996.

[63] M. J. L. Gales. *Model-Based Techniques for Noise Robust Speech Recognition*. PhD thesis, University of Cambridge, UK, 1995.

[64] S. Furui. Towards robust speech recognition under adverse conditions. In *Proc. ESCA Workshop on Speech Processing in Adverse Conditions*, Cannes, France, 1992.

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

[66] A. P. Varga, and R. K. Moore. Hidden Markov model decomposition of speech and noise. In *Proc. IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP'90)*, volume 2, pages 845–848, Albuquerque, NM, USA, April 1990.

[67] C. J. Leggetter, and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9(2):171–185, April 1995.

[68] J.-L. Gauvain, and C.-H. Lee. Maximum *a posteriori* estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, April 1994.

[69] M. Chengyuan, H. J. Kuo, H. Soltau, and et al. A comparative study on system combination schemes for LVCSR. In *Proc. IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP'10)*, pages 4394 – 4397, Dallas, TX, USA, June 2010.

[70] D. Kolossa, A. Klimas, and R. Orglmeister. Separation and robust recognition of noisy, convolutive speech mixtures using time-frequency masking and missing data techniques. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 82–85, New Paltz, NY, USA, November 2005.

[71] S. Sivadas, and H. Hermansky. Generalized tandem feature extraction. In *Proc., IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'03)*, volume 1, pages 56–59, Hong Kong, China, April 2003.

[72] A. Acero, L. Deng, T. Kristjansson, and J. Zhang. HMM adaptation using Vector Taylor series for noisy speech recognition. In *Proc. ICSLP*, Beijing, China, 2000.

[73] O. Kalinli, M. L. Seltzer, and A. Acero. Noise adaptive training using a Vector Taylor series approach for noise robust automatic speech recognition. In *Proc. IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'09)*, pages 3825–3828, April 2009.

[74] N. K. Goel, and A. G. Andreou. Heteroscedastic discriminant analysis and reduced-rank hmms for improved speech recognition. *Speech Communication*, 26(4):283–297, December 1998.

[75] M. F. J. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. Technical report, University of Cambridge, Cambridge, UK,

1997. CUED/F-INFENG/TR 291.

[76] B. Varadarajan, D. Povey, and S. M. Chu. Quick fMLLR for speaker adaptation in speech recognition. In *Proc. IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'08)*, pages 4297–4300, March 2008.

[77] A. Sankar, and C.-H. Lee. Robust speech recognition based on stochastic matching. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'95)*, 1:121–124, May 1995.

[78] L. Delphin-Poulat, C. Mokbel and J. Idier. Frame synchronous stochastic matching based on Kullback-Leibler information. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'98)*, pages 89–92, 1998.

[79] M. Afify. Sequential bias compensation for robust speech recognition. In *Proc., European Conference on Speech Communication and Technology*, pages 2821–2824, 1999.

[80] L. Deng, J. Droppo, and A. Acero. Recursive estimation of nonstationary noise using iterative stochastic approximation speech recognition. *IEEE Transactions on Speech and Audio Processing*, 11(6):568–580, November 2003.

[81] G.-H. Ding, X. Wang, Y. Cao, F. Ding, and Y. Tang. Sequential noise estimation for noise-robust speech recognition based on 1st-order VTS approximation. In *Proc., IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 363–368, November 2005.

[82] H. Liao, and M. J. F. Gales . Joint uncertainty decoding for robust large vocabulary speech recognition. Technical report, University of Cambridge, UK, 2006.

[83] M. J. L. Gales. Predictive model-based compensation schemes for robust speech recognition. *Speech Communication*, 25(1):49–74, 1998.

[84] C. Lawrence, and M. Rahim. Integrated bias removal techniques for robust speech recognitions. *Computer Speech and Language*, 13(3):283–298, July 1999.

[85] M. Afify, Y. Gong, and J.-P. Haton. A general joint additive and convolutive bias compensation approach applied to noisy Lombard speech recognition. *IEEE Transactions on Speech and Audio Processing*, 6(6):524–538, November 1998.

[86] B. Tian, M. Sun, R. J. Sclabassi, and K. Yi. A unified compensation approach for speech recognition in severely adverse environment. In *Proc., Forth International Symposium on Uncertainty Modeling and Analysis (ISUMA 2003)*, pages 256–261, College Park, MD, USA, September 2003.

[87] J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero. A unified framework of hmm adaptation with joint compensation of additive and convolutive distortions. *Computer Speech and Language*, 23(3):389–405, July 2009.

[88] X. Menéndez-Pidal, R. Chen, D Wu, and M. Tanaka. Compensation of channel and noise distortions combining normalization and speech enhancement techniques. *Speech Communication*, 34(1-2):115–126, April 2001.

[89] N. U. Nair, and T. V. Sreenivas. Joint evaluation of multiple speech patterns for speech recognition and training. *Computer Speech and Language*, 24(2):307–340, April 2010.

[90] M. J. L. Gales, and S. Young. Robust continuous speech recognition using parallel model combination. *IEEE Transactions on Speech and Audio Processing*, 4(5):352–359, September 1996.

[91] R. Martin. Spectral subtraction based on minimum statistics. In *Proc. of the 7th EUSIPCO'94*, volume 1, pages 1182–1185, Edinburgh, U. K., 1994.

[92] I. Cohen. Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging. *IEEE Transactions on Speech and Audio Processing*, 11(5):466–475, September 2003.

[93] G. Doblinger. Computationally efficient speech enhancement by spectral minima tracking in subbands. In *Proc. of the Eurospeech*, volume 2, pages 1513–1516, 1995.

[94] S. Rangachari, and P. C. Loizou. A noise estimation algorithm for highly nonstationary environments. *Speech Communication*, 48(2):220–231, February 2006.

[95] N. Fan, J. Rosca, and R. Balan. Speech noise estimation using enhanced minima controlled recursive averaging. In *Proc., IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'07)*, volume 4, pages 581–584, Honolulu, Hawaii, USA, April 2007.

[96] I. Cohen, and B. Berdugo. Speech enhancement for non-stationary noise environments. *Signal Processing*, 81:2403–2418, 2001.

[97] M. Berouti, M. Schwartz, and J. Makhoul. Enhancement of speech corrupted by acoustic noise. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'79)*, pages 208–211, 1979.

[98] R. Sarikaya and J. H. L. Hansen. Improved Jacobian adaptation for fast acoustic model adaptation for noisy speech recognition. In *Proc., Int. Conf. Spoken Lang. Process. (ICSLP)*, volume 3, pages 702–705, Beijing, China, October 2000.

[99] R. Turner, Y. Saatci, and C. E. Rasmussen. Adaptive sequential Bayesian change point detection. In *Temporal Segmentation Workshop at NIPS 2009*, Whistler, BC, Canada, December 2009.

[100] L. B. Asl, and M. Geravanchizadeh. Speech enhancement using sexual reproduction-based PSO. In *Proc. 10th International Conference on Information Sciences Signal Processing and their Applications (ISSPA'10)*, pages 129–132, Kuala Lampur, Malaysia, May 2010.

[101] M. Geravanchizadeh, and L. B. Asl. Asexual reproduction-based adaptive quantum particle swarm optimization algorithm for dual-channel speech enhancement. In *Proc. 4th International Symposium on Communications, Control and Signal Processing, (ISCCSP 2010)*, pages 1–4, Limassol, cyprus, March 2010.

[102] K. E. Parsopoulos and M. N. Vrahatis. Particle swarm optimizer in noisy and continuously changing environments. In *Artificial Intelligence and Soft Computing, IASTED/ACTA*, pages 289–294. IASTED/ACTA Press, 2001.

[103] Y. Cooren, M. Clerc, and P. Siarry. A parameter-free particle swarm optimization algorithm. In *Proc. 7th EU Meeting on Adaptive, Self-Adaptive, and Multi-Level Metaheuristics*, November 2006.

[104] S. Haykin. *Adaptive Filter Theory.* Prentice Hall, 2001. 4th Edition.

[105] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10(3):1706–1716, June 2006.

[106] S.-Z. Zhao, P.N. Suganthan, and S. Das. Dynamic multi-swarm particle swarm optimizer with sub-regional harmony search. In *IEEE Congress on Evolutionary*

Computation, pages 1–8, Barcelona, Spain, July 2010.

[107] H.-G. Hirsch, and D. Pearce. The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions. In *Proc., ISCA ITRW ASR-2000 Automatic Speech Recognition: Challenges for the Next Millennium*, pages 181–188, Paris, France, September 2000.

[108] S. O. Haykin. *Adaptive Filter Theory, 4th Edition.* Prentice Hall, 2001.

[109] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.