Chapter 4

Robust Distributed Speech Recognition Using Auditory Modelling

Ronan Flynn and Edward Jones

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/49954

1. Introduction

The use of the Internet for accessing information has expanded dramatically over the past few years, while the availability and use of mobile hand-held devices for communication and Internet access has greatly increased in parallel. Industry has reacted to this trend for information access by developing services and applications that can be accessed by users on the move. These trends have highlighted a need for alternatives to the traditional methods of user data input, such as keypad entry, which is difficult on small form-factor mobile devices. One alternative is to make use of automatic speech recognition (ASR) systems that act on speech input from the user. An ASR system has two main elements. The first element is a front-end processor that extracts parameters, or features, that represent the speech signal. These features are processed by a back-end classifier, which makes the decision as to what has been spoken.

In a fully embedded ASR system [1], the feature extraction and the speech classification are carried out on the mobile device. However, due to the computational complexity of high-performance speech recognition systems, such an embedded architecture can be impractical on mobile hand-held terminals due to limitations in processing and memory resources. On the other hand, fully centralised (server-based) ASR systems have fewer computational constraints, can be used to share the computational burden between mobile users, and can also allow for the easy upgrade of speech recognition technologies and services that are provided. However, in a centralised ASR system the recognition accuracy can be compromised as a result of the speech signal being distorted by low bit-rate encoding at the codec and a poor quality transmission channel [2, 3].

A distributed speech recognition (DSR) system is designed to overcome some of the difficulties described above. In DSR, the terminal (the mobile device) includes a local front-end processor that extracts, directly from the speech, the features to be sent to the remote

server (back-end) where recognition is performed. In mobile environments, the speech features can be sent over an error protected data channel rather than a voice channel, making the DSR system more robust to channel errors.

However, DSR systems generally operate in high levels of background noise (particularly in mobile environments). For mobile users in noisy environments (airports, cars, restaurants etc.) the speech recognition accuracy can be reduced dramatically as a consequence of additive background noise. A second source of error in DSR systems is the presence of transmission errors in the form of random packet loss and packet burst loss during transmission of speech features to the classifier. Packet loss can arise in wireless and packet switched (IP) networks, both networks over which a DSR system would normally be expected to operate. Packet loss, in particular packet burst loss, can have a serious impact on recognition performance and needs to be considered in the design of a DSR system.

This chapter addresses the issue of robustness in DSR systems, with particular reference to the problems of background noise and packet loss, which are significant bottlenecks in the commercialisation of speech recognition products, particularly in mobile environments. The layout of the chapter is as follows. Section 2 discusses the DSR architecture and standards in more detail. This is followed by an overview of the auditory model used in this chapter as an alternative front-end to those published in the DSR standards. The Aurora 2 database and a description of the speech recognition system used are also discussed in Section 2. Section 3 addresses the problem of robustness of speech recognition systems in the presence of additive noise, in particular, by examining in detail the use of speech enhancement techniques to reduce the effects of noise on the speech signal. The performance of a DSR system in the presence of both additive background noise and packet loss is examined in Section 4. The feature vectors produced by the auditory model are transmitted over a channel that is subject to packet burst loss and packet loss mitigation to compensate for missing features is investigated. Conclusions are presented in Section 5.

2. Distributed speech recognition systems

2.1. DSR architecture and standards

A DSR system is designed as a compromise between local and centralised recognition, in order to alleviate the issues associated with these approaches [2, 3]. In DSR, the speech recognition task is split between the terminal or client, where the front-end feature extraction is performed, and the network or server, where the back-end recognition is performed. The features that represent the speech are sent by means of an error protected data channel to the classifier for processing. DSR avoids both the speech encoding and decoding stages associated with centralised recognition and so eliminates the degradations that originate from the speech compression algorithms. The bandwidth required to transmit the extracted features to the server is much less than what is required to send the encoded speech signal. DSR systems offer some advantages over other architectures. Recent comparative studies have shown the superior performance of DSR to codec-based ASR [4]. However, in a DSR system, transmission errors in the form of random packet loss and

packet burst loss still need to be taken into consideration. Such transmission errors can have a significant impact on recognition accuracy.

Furthermore, it is well known that the presence of noise severely degrades the performance of speech recognition systems, and much research has been devoted to the development of techniques to alleviate this effect; this is particularly important in the context of DSR where mobile clients are typically used in high-noise environments (though the same problem also exists for local embedded, or centralised architectures in noisy conditions). One approach that can be used to improve the robustness of ASR systems is to enhance the speech signal itself before feature extraction. Speech enhancement can be particularly useful in cases where a significant mismatch exists between training and testing conditions, such as where a recognition system is trained with clean speech and then used in noisy conditions. A significant amount of research has been carried out on speech enhancement, and a number of approaches have been well documented in the literature [5]. There has also been much interest in DSR in recent years, within the research community, and in international standardisation bodies, in particular, the European Telecommunications Standards Institute (ETSI) [6-9], which has developed a number of different recommendations for front-end processors of different levels of complexity.

The ETSI basic front-end [6] was developed for implementation over circuit-switched channels and this implementation is also considered in the other three standards. The advanced front-end [7] produces superior performance to the basic front-end, and was designed to increase robustness in background noise. The implementation of the front-ends over packet-switched Internet Protocol (IP) networks has been specified in two documents published by the Internet Engineering Task Force (IETF). The first of these [10] specifies the real-time transport protocol (RTP) payload format for the basic front-end while the second [11] specifies the RTP payload format for the advanced front-end.

The ETSI basic and advanced front-ends both implement MFCC-based parameterisation of the speech signal. The stages involved in feature extraction based on MFCCs are shown in Figure 1. The speech signal first undergoes pre-emphasis in order to compensate for the unequal sensitivity of human hearing across frequency. Following pre-emphasis, a short-term power spectrum is obtained by applying a fast Fourier transform (FFT) to a frame of Hamming windowed speech. Critical band analysis is carried out using a bank of overlapping, triangular shaped, bandpass filters, whose centre frequencies are equally spaced on the mel scale. The FFT magnitude coefficients are grouped into the appropriate critical bands and then weighted by the triangular filters. The energies in each band are summed, creating a filter bank vector of spectral energies on the mel scale. The size of this vector of spectral energies is equal to the number of triangular filters used. A non-linearity in the form of a logarithm is applied to the energy vector. The final step is the application of a discrete cosine transform (DCT) to generate the MFCCs.

In the ETSI DSR front-ends, speech, sampled at 8 kHz, is blocked into frames of 200 samples with an overlap of 60%. A logarithmic frame energy measure is calculated for each frame before any processing takes place. In the case of the basic front-end, pre-emphasis is carried

out using a filter coefficient equal to 0.97 while the advanced front-end uses a value of 0.9. A Hamming window is used in both the ETSI basic and advanced front-ends prior to taking an FFT. In the ETSI advanced front-end a power spectrum estimate is calculated before performing the filter bank integration. This results in higher noise robustness when compared with using a magnitude spectrum estimate as used in the ETSI basic front-end [12]. The two front-ends both generate a feature vector consisting of 14 coefficients made up of the frame log-energy measure (determined prior to pre-emphasis) and cepstral coefficients C_0 to C_{12}. In order to introduce robustness against channel variations, the ETSI advanced front-end carries out post-processing in the cepstral domain on coefficients C_1 to C_{12} in the form of blind equalisation [13].

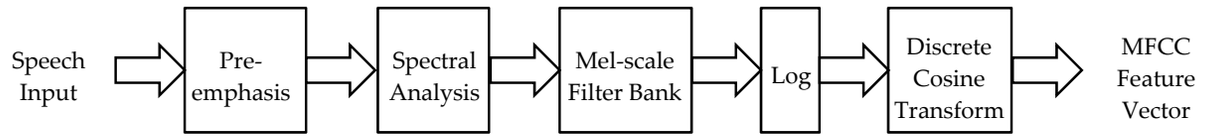

Figure 1. MFCC feature extraction

2.2. Auditory modelling as an alternative front-end

Many computational auditory models have been proposed for use in speech recognition systems, often with excellent results, particularly in the presence of noise. In the work presented here, the auditory model of Li *et al*. [14] is used. The choice of this auditory front-end is motivated by previous work carried out by the authors [15] where a number of auditory front-ends were investigated in a comparative study of robust speech recognition with the widely-used Aurora 2 database [16]. In that study, there was no pre-processing or enhancement of the speech utterances. The front-ends investigated were perceptual linear prediction (PLP) proposed by Hermansky [17], the PEMO algorithm proposed by Tchorz and Kollmeier [18], and the front-end processor proposed by Li *et al.* [14]. For the task of connected digit recognition using the Aurora 2 database, the front-end proposed by Li *et al*. gave the best overall recognition results of all the auditory models examined, and with an overall reduction in recognition error compared to the ETSI basic front-end [6] which was used as a baseline for comparison.

The steps involved in feature extraction in the Li *et al.* auditory model are shown in Figure 2. Speech is sampled at 8 kHz and blocked into frames of 240 samples. Frame overlap is 66.7% and a Hamming window is used prior to taking a FFT. An outer/middle ear transfer function that models pressure gain in the outer and middle ears is applied to the spectrum magnitude. After conversion of the spectrum to the Bark scale, the transfer function output is processed by an auditory filter that is derived from psychophysical measurements of the frequency response of the cochlea. A non-linearity in the form of a logarithm followed by a DCT is applied to the filter outputs to generate the cepstral coefficients. The recognition experiments use vectors that include energy and 12 cepstral coefficients (C_1 to C_{12}) along with velocity and acceleration coefficients. This results in vectors with an overall dimension equal to 39.

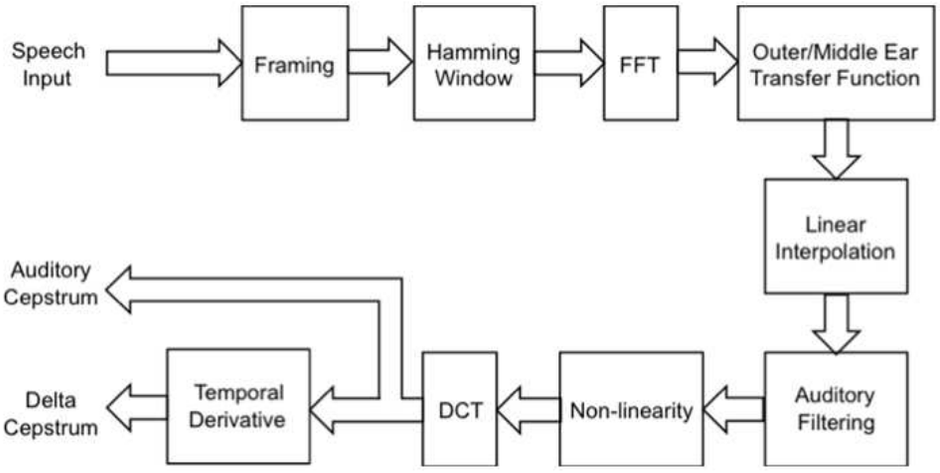

Figure 2. Feature extraction proposed by Li *et al*. [14].

2.3. Aurora 2 database

The recognition problem examined in this work is connected digit recognition using the Aurora 2 database [16]. The motivation behind the creation of the Aurora database was to provide a framework that allowed for the evaluation and comparison of speech recognition algorithms in noisy conditions, thus providing a good basis for comparison between researchers. It has been widely used in the development and evaluation of DSR systems. The speech database is derived from utterances of isolated digits and connected digit sequences spoken by US-American adults originally included in the well-known TIDigits database. The speech in the TIDigits database is sampled at 20 kHz and is down-sampled to 8 kHz in the Aurora database. Some additional filtering is applied to the down-sampled data in order to take into account the frequency characteristics of equipment used in telecommunications systems. The channel characteristics used are G.712 and modified intermediate reference system (MIRS). The down-sampled, filtered speech corresponds to "clean" data in the Aurora database. The Aurora database also contains "noisy" data. This corresponds to clean data with noise artificially added at SNRs of 20 dB, 15 dB, 10 dB, 5 dB, 0 dB and –5 dB. The noise signals added are chosen to reflect environments in which telecommunication terminals are used. In total there are eight different noise types: subway, babble, car, exhibition hall, restaurant, street, airport and train station.

The Aurora framework includes a set of standard test conditions for evaluation of front-end processors. For the purpose of training the speech recogniser, two modes are defined. The first mode is training on clean data and the second mode is multi-condition training on noisy data. The same 8440 utterances, taken from the training part of the TIDigits, are used for both modes. For the multi-condition training, the clean speech signals are used, as well as speech with four different noise types (subway, babble, car and exhibition hall), added at SNRs of 20 dB, 15 dB, 10 dB and 5 dB. There are three different test sets defined for recognition testing, with the test utterances taken from the testing part of the TIDigits database. Test Set A (28028 utterances) employs the same four noises as used for the multi-

condition training. Test Set B uses the same utterances as Test Set A but uses four different noise types (restaurant, street, airport and train station). In both Test Sets A and B, the frequency characteristic used in the filtering of the speech and noise is the same as that used in the training sets, namely G.712. The frequency characteristic of the filter used in Test Set C (14014 utterances) is the MIRS, and is different from that used in the training sets. Subway and street noises are used in Test Set C.

2.4. Speech recognition system

The classifier used for the recognition experiments in the work presented in this chapter is the HMM-based recogniser architecture specified for use with the Aurora 2 database [16], and implemented with the widely-used HTK package [19]. The use of a well-known specification provides a common framework with which to compare different front-ends and feature vectors for the purpose of connected digit recognition. There are eleven whole word HMMs each with 16 states; each state has 3 Gaussian mixtures. The topology of the models is left-to-right without any skips over states. This topology is suitable for modelling the sequential nature of speech and the consecutive states represent the consecutive speech states in a particular utterance. Two pause models, "sil" and "sp", are defined. The "sil" model has 3 states and each state has 6 mixtures. The "sp" model has a single state. The Baum-Welch re-estimation algorithm is applied in the training of the word models. An utterance can be modelled by any sequence of digits with the possibility of a "sil" model at both ends and adjacent digits separated by a "sp" model.

The method used to measure the performance of a speech recognition system is dependent on the type of utterance that is to be recognised, i.e. isolated word or continuous speech. There are three error types associated with the recogniser in a continuous speech recognition system:

Substitutions (S) – A word in the original sentence is recognised as a different word.

Deletions (D) – A word in the original sentence is missed.

Insertions (I) – A new word is inserted between two words of the original sentence.

The performance measure used throughout the work presented here, and also used in [16], is the word accuracy as defined by (1):

$$Word\ accuracy = \frac{N-(S+D+I)}{N} \times 100\% \tag{1}$$

where N is the total number of evaluated words. The word accuracies for each of the Aurora test sets presented throughout this chapter are calculated according to [16], which defines the performance measure for a test set as the word accuracy averaged over all noises and over all SNRs between 0 dB and 20dB. The overall word accuracy for the two training modes, clean training and multi-condition training, is calculated as the average over the three test sets A, B and C.

3. Speech enhancement

Additive noise from interfering noise sources, and convolutional noise arising from transmission channel characteristics both contribute to a degradation of performance in automatic speech recognition systems. This section addresses the problem of robustness of speech recognition systems in the first of these conditions, namely additive noise. As noted previously, speech enhancement is one way in which the effects of noise on the speech signal can be reduced. Enhancement of noisy speech signals is normally used to improve the perception of the speech by human listeners however, it may also have benefits in enhancing robustness in ASR systems. Speech enhancement can be particularly useful in cases where a significant mismatch exists between training and testing conditions, such as where a recognition system is trained with clean speech and then used in noisy conditions, as inclusion of speech enhancement can help to reduce the mismatch. This approach to improving robustness is considered in this section.

In the speech recognition system described here, the input speech is pre-processed using an algorithm for speech enhancement. A number of different methods for the enhancement of speech, combined with the auditory front-end of Li *et al.* [14], are evaluated for the purpose of robust connected digit recognition. The ETSI basic [6] and advanced [7] front-ends proposed for distributed speech recognition are used as a baseline for comparison.

3.1. Speech enhancement overview

The enhancement of noisy speech can be described as an estimation problem in which the original clean signal is estimated from a degraded version of the signal. A significant amount of research has been carried out on speech enhancement, and a number of approaches have been well documented in the literature. A survey of a number of approaches to speech enhancement using a single microphone is presented in [5].

Two measures that can be used to perceptually evaluate speech are its *quality* and its *intelligibility* [5]. Speech quality is a subjective measure and is dependent on the individual preferences of listeners. It is a measure of how comfortable a listener is when listening to the speech under evaluation. The intelligibility of the speech can be regarded as an objective measure, and is calculated based on the number or percentage of words that can be correctly recognised by listeners. The intelligibility and the quality of speech are not correlated [5] and it is well known that improving one of the measures can have a detrimental effect on the other one. Speech enhancement algorithms give a trade-off between noise reduction and signal distortion. A reduction in noise can lead to an improvement in the subjective quality of the speech but a decrease in the measured speech intelligibility [5].

When using speech enhancement in an ASR system, the speech is enhanced before feature extraction and recognition processing. The advantage of this is that there is no impact on the computational complexity of the feature extraction or the recognition processes as the enhancement is independent of both, and the speech enhancement can be implemented as an add-on without significantly affecting existing parts of the system. However, every

speech enhancement process will introduce some form of signal distortion and it is important that the impact of this distortion on the recognition process is minimised.

3.2. Speech enhancement algorithms

In this section, the various speech enhancement algorithms that were examined are briefly described. The algorithms range from well-established algorithms like that of Ephraim and Malah [20], to more recently proposed ones like that of Rangachari and Loizou [21]. Furthermore, the algorithms cover a range of paradigms, including spectral subtraction-based algorithms using the FFT for spectral analysis, as well as methods based on auditory filter banks.

Ephraim and Malah [20] present a minimum mean-square error short-time spectral amplitude (MMSE STSA) estimator. The estimator is based on modelling speech and noise spectral components as statistically independent Gaussian random variables. The enhanced speech is constructed using the MMSE STSA estimator combined with the original phase of the noisy signal. Analysis is carried out in the frequency domain and the signal spectrum is estimated using an FFT.

Westerlund *et al.* [22] present a speech enhancement technique in which the input signal is first divided into a number of sub-bands. The signal in each sub-band is individually multiplied by a gain factor in the time domain based on an estimate of the short term SNR in each sub-band at every time instant. High SNR values indicate the presence of speech and the sub-band signal is amplified. Low SNR values indicate the presence of noise only and the sub-band signal remains unchanged.

Martin [23] presented an algorithm for the enhancement of noisy speech signals by means of spectral subtraction, in particular through a method for estimation of the noise power on a sub-band basis. Martin's noise estimation method is based firstly on the independence of speech and noise, and secondly on the observation that speech energy in an utterance falls to a value close to or equal to zero for brief periods. Such periods of low speech energy occur between words or syllables in an utterance and during speech pauses. The energy of the signal during these periods reflects the noise power level. Martin's minimum statistics noise estimation method tracks the short-term power spectral density estimate of the noisy speech signal in each frequency bin separately. The minimum power within a defined window is used to estimate the noise floor level. The minimum tracking method requires a bias compensation since the minimum power spectral density of the noisy signal is smaller than the average value. In [24], Martin further developed the noise estimation algorithm by using a time- and frequency-dependent smoothing parameter when calculating the smoothed power spectral density. A method to calculate an appropriate time and frequency dependent bias compensation is also described in [24] as part of the algorithm.

Rangachari and Loizou [21] proposed an algorithm for the estimation of noise in highly non-stationary environments. The noisy speech power spectrum is averaged using time and frequency dependent smoothing factors. This new averaged value is then used to update the

noise estimate. Signal-presence probability in individual frequency bins is calculated in order to update the smoothing factors. Signal presence is determined by computing the ratio of the noisy speech power spectrum to its local minimum, which is updated continuously by averaging past values of the noisy speech power spectra with a look-ahead factor. The results in [21] indicate that the local minimum estimation algorithm adapts very quickly to highly non-stationary noise environments.

A technique for the removal of noise from degraded speech using two filtering stages was proposed by Agarwal and Cheng [25]. The first filtering stage coarsely reduces the noise and whitens any residual noise while the second stage attempts to remove the residual noise. Filtering is based on the Wiener filter concept and filter optimisation is carried out in the mel-frequency domain. The algorithm, described as a two-stage mel-warped Wiener filter noise reduction scheme, is a major component of the ETSI advanced front-end standard for DSR [7]. The implementation of noise reduction in the ETSI advanced front-end is summarised in [12].

3.3. Tests and results

This section presents recognition results from tests on the Aurora 2 database [16], using the combination of the speech enhancement algorithms described previously and the auditory model proposed by Li *et al*. (see Section 2.2). In the analysis, two versions of the Li *et al*. front-end are used. The first, referred to as Li *et al*. (I), generates a feature vector consisting of 13 coefficients made up of the frame log-energy measure and the cepstral coefficients C_1 to C_{12}. The second version, referred to as Li *et al*. (II), generates a feature vector that contains the cepstral coefficients C_1 to C_{12} along with a weighted combination of cepstral coefficient C_0 and the frame log-energy measure. The reason for investigating two versions of the Li *et al*. front-end, Li *et al*. (I) and Li *et al*. (II), is to allow for a closer comparison with the ETSI basic front-end [6] and the ETSI advanced front-end [7] respectively. In all cases training was carried out using clean data, so that the effect of the speech enhancement in removing mismatch could be examined. The speech enhancement algorithms were used on both the (clean) training speech as well as the (noisy) test speech. Feature vectors were extracted directly from the enhanced speech with no intermediate processing. The recognition experiments used vectors that include 13 static coefficients along with velocity and acceleration coefficients. This results in vectors with an overall dimension equal to 39. The word accuracies detailed in the tables of results were calculated as previously described in Section 2.4.

In the comparison of Li *et al*. (I) and the ETSI basic front-end, there was no post-processing of the feature vectors carried out. The recognition results using the Aurora 2 database for Li *et al*. (I), for each speech enhancement algorithm, are given in Table 1 and the corresponding results for the ETSI basic front-end (the baseline for this test) are given in Table 2.

The performance of Li *et al*. (II) was compared with the performance of the ETSI advanced front-end. The ETSI advanced front-end includes a SNR-dependent waveform processing block that is applied after noise reduction and before feature extraction. The purpose of this

	Absolute word accuracy %			
Enhancement	Set A	Set B	Set C	Overall
None	62.16	64.31	57.76	62.14
Ephraim & Malah	78.85	79.38	74.78	78.25
Westerlund *et al.*	75.87	76.32	70.45	74.97
Martin	72.47	71.96	70.21	71.81
Rangachari & Loizou	74.50	73.16	74.29	73.92
Agarwal & Cheng	86.33	84.87	81.86	84.85

Table 1. Recognition results for the Li *et al*. (I) front-end.

	Absolute word accuracy %			
Enhancement	Set A	Set B	Set C	Overall
None	61.34	55.75	66.14	60.06
Ephraim & Malah	76.34	75.91	73.71	75.64
Westerlund *et al.*	76.04	72.54	72.36	73.90
Martin	67.98	67.57	68.24	67.87
Rangachari & Loizou	63.58	61.57	67.82	63.62
Agarwal & Cheng	84.39	82.75	78.72	82.60

Table 2. Recognition results for the ETSI basic front-end.

	Absolute word accuracy %			
Enhancement	Set A	Set B	Set C	Overall
None	67.34	69.18	63.44	67.30
Ephraim & Malah	80.36	81.03	79.34	80.42
Westerlund *et al.*	78.70	80.02	78.44	79.18
Martin	73.07	72.93	72.17	72.83
Rangachari & Loizou	76.08	76.16	75.94	76.08
Agarwal & Cheng	87.03	86.85	84.58	86.47

Table 3. Recognition results for the Li *et al*. (II) front-end.

block is to improve the noise robustness in the front-end of an ASR system by enhancing the high SNR period portion and attenuating the low SNR period portion in the waveform time domain, thus increasing the overall SNR of noisy speech [26]. However, the evaluation here is looking primarily at the effect of speech enhancement or noise reduction alone on the connected digit recognition accuracy. Therefore, the waveform processing block in the ETSI advanced front-end was disabled. In addition, the ETSI advanced front-end carries out post-

processing in the cepstral domain in the form of blind equalisation as described in [13]. To ensure a closer match with the ETSI advanced front-end, the feature vectors produced by Li *et al*. (II) undergo post-processing in the cepstral domain by means of cepstral mean subtraction (CMS). The recognition results for Li *et al*. (II), for each speech enhancement algorithm, are detailed in Table 3 and the recognition results for the ETSI advanced front-end are detailed in Table 4.

	Absolute word accuracy %			
Enhancement	Set A	Set B	Set C	Overall
None	65.92	65.48	70.07	66.57
Ephraim & Malah	77.92	77.61	78.64	77.94
Westerlund *et al.*	79.09	79.13	79.70	79.23
Martin	71.26	72.91	72.71	72.21
Rangachari & Loizou	73.77	73.35	78.85	74.62
Agarwal & Cheng	85.92	85.66	83.89	85.41

Table 4. Recognition results for the ETSI advanced front-end.

Table 5 provides an overall view of the relative performance of the different speech enhancement algorithms for each of the four front-end versions considered.

Rank	Li *et al.* (I) FE	ETSI basic FE	Li *et al.* (II) FE	ETSI advanced FE
1	Agarwal & Cheng	Agarwal & Cheng	Agarwal & Cheng	Agarwal & Cheng
2	Ephraim & Malah	Ephraim & Malah	Ephraim & Malah	Westerlund *et al.*
3	Westerlund *et al.*	Westerlund *et al.*	Westerlund *et al.*	Ephraim & Malah
4	Rangachari & Loizou	Martin	Rangachari & Loizou	Rangachari & Loizou
5	Martin	Rangachari & Loizou	Martin	Martin

Table 5. Performance ranking of enhancement algorithms.

3.4. Discussion

Ignoring speech enhancement, and comparing Tables 1 and 2, the performance of Li *et al*. (I) exceeds the baseline ETSI front-end [6] by 2.08% overall. From Table 3 and Table 4, again without speech enhancement applied, there is a difference in recognition accuracy of 0.73% in favour of Li *et al.* (II) when compared with the ETSI advanced front-end [7].

The other results in Tables 1 to 4 show that enhancement of the speech prior to feature extraction significantly improves the overall recognition performance. This improvement in recognition accuracy is observed for both the ETSI basic [6] and advanced [7] front-ends and the front-end proposed by Li *et al.* [14]. A comparison of Table 1 with Table 2 shows that Li *et al.* (I) outperforms the ETSI basic front-end for all of the speech enhancement techniques evaluated. Furthermore, from Tables 3 and 4, it is seen that Li *et al.* (II) again outperforms the ETSI advanced front-end for all speech enhancement methods except Westerlund *et al.* [22], for which the overall recognition results are quite close.

For Li *et al.* (I), Li *et al.* (II), the ETSI basic front-end and the ETSI advanced front-end, the best overall recognition accuracy is obtained for speech enhancement using the algorithm proposed by Agarwal and Cheng [25]. The combination of auditory front-end and the two-stage, mel-warped, Wiener filter noise reduction scheme results in an overall recognition accuracy that is approximately 6% better overall compared with the next ranked front-end and speech enhancement combination. After Agarwal and Cheng [25], the next best performance across the board is obtained using Ephraim and Malah [20], and Westerlund *et al*. [22]. This suggests that the choice of speech enhancement algorithm for best speech recognition performance is somewhat independent of the choice of front-end (though this would have to be validated by further testing with other front ends).

4. Robustness to noise and packet loss

In the previous section, the benefit of speech enhancement prior to feature extraction in a speech recognition system was demonstrated. However, in a DSR system, transmission errors can still have a significant impact on recognition performance. Such transmission errors in the form of bit errors, random packet loss and packet burst loss need to be taken into consideration. This is particularly important in the context of increasing use of packet-based networks for transmission of speech and data in mobile environments. This section examines the performance of a DSR system in the presence of both background noise and packet loss.

4.1. Channel models and loss compensation

A DSR client and server may be interconnected over either a circuit-switched channel or a packet-switched channel. Approaches used in the literature to simulate different channel types fall into two categories. The first makes use of physical layer models that simulate transmission phenomena that occur on the physical channel. The second category involves the use of statistical models that model unconditional packet loss probability and conditional packet loss burst lengths. This is the approach used in this chapter.

To simulate packet loss and error bursts, the 2-state Gilbert model is widely used. In [27-29], a voice over IP (VoIP) channel is simulated using such a model. References [30, 31] simulate IP channels and use a 2-state Gilbert model to simulate burst type packet loss on the channel. Statistical models have also been used to simulate the physical properties of the

communication channel. The Gilbert model was found in [3] to be inadequate for simulating a GSM channel and instead a two-fold stochastic model is used in which there are two processes, namely shadowing and Rayleigh fading. This same model was used by [32], again to model a GSM network. Reference [33] compares three models of packet loss and examines their effectiveness at simulating different packet loss conditions. The models are a 2-state Markov chain, the Gilbert-Elliot model and a 3-state Markov chain. The 2-state Markov chain in [33] uses State 1 to model a correctly received packet and State 2 to model a lost packet. While the Gilbert-Elliot model is itself a 2-state Markov model, there is only a probability of packet loss when in State 2. The three models in [33] are all validated for GSM and wireless local area network (WLAN) channels. Results indicate that the 3-state Markov model gives the best results overall and this model is used in the work described here; the model is described in more detail later in this chapter.

There are a number of techniques documented that are used within DSR systems for the purpose of reducing transmission error degradation and so increasing the robustness of the speech recognition. Error-robustness techniques are categorised in [4] under the headings client-based error recovery, and server-based error concealment. Client-based techniques include retransmission, interleaving and forward error correction (FEC). While retransmission and FEC may result in recovering a large amount of transmission errors, they have the disadvantage of requiring additional bandwidth and introducing additional delays and computational overhead. Server-based methods include feature reconstruction, by means of repetition or interpolation, and error correction in the ASR-decoding stage. Reference [4] provides a survey of robustness issues related to network degradations and presents a number of analyses and experiments with a focus on transmission error robustness.

The work described in [34-38] is focused on burst-like packet loss and how to improve speech recognition in the context of DSR. The importance of reducing the average burst length of lost feature vectors rather than reducing the overall packet loss rate is central to the work in these papers. By minimising the average burst length, the estimation of lost feature vectors is more effective. Reference [34] compared three different interleaving mechanisms (block, convolutional and decorrelated) and found that increasing the degree of interleaving increases the speech recognition performance but that this comes with the cost of a higher delay. It is further suggested in [38] that, for a DSR application, it is more beneficial to trade delay for accuracy rather than trading bit-rate for accuracy as in forward error correction schemes. Reference [35] combines block interleaving to reduce burst lengths on the client side with packet loss compensation at the server side. Two compensation mechanisms are examined: feature reconstruction by means of nearest neighbour repetition, interpolation and maximum a-posteriori (MAP) estimation; and a decoder-based strategy using missing feature theory. The results suggest that for packet loss compensation, the decoder-based strategy is best. This is especially true in the presence of large bursts of losses as the accuracy of reconstruction methods falls off rapidly as burst length increases. Interleaving, feature estimation and decoder based strategies are combined in [36] in order to improve the recognition performance in the presence of packet loss in DSR.

In this section, the 3-state model proposed in [33] is used to simulate packet loss and loss bursts. To compensate for missing packets, two error-concealment methods are examined, namely nearest neighbour repetition and interpolation. Error mitigation using interleaving is also considered.

4.2. Packet loss framework

4.2.1. Packet loss model

The packet loss model used in this work is the 3-state Markov chain proposed by [33]. This 3-state model was found to be more effective at simulating different packet loss conditions in comparison with a 2-state Markov chain and the Gilbert-Elliot model. The model is detailed in Figure 3, showing the three states and the transition probabilities. Occupancy of states 1 and 3 indicate no packet loss while occupancy of state 2 indicates packet loss. In Figure 3, Q, q and Q' are the self-loop probabilities of states 1, 2 and 3 respectively. The model parameters are designed so that state 1 models long duration periods of no loss and state 3 models short periods of no loss, which occur in between packet loss in burst-like conditions. The following four parameters define the model, and from these parameters the transition probabilities of the 3-state model can be determined:

α = overall probability of a packet being lost

β = average packet loss burst length

N_1 = average length, in packets, of loss-free periods

N_3 = average length, in packets, of no-loss periods inside loss periods

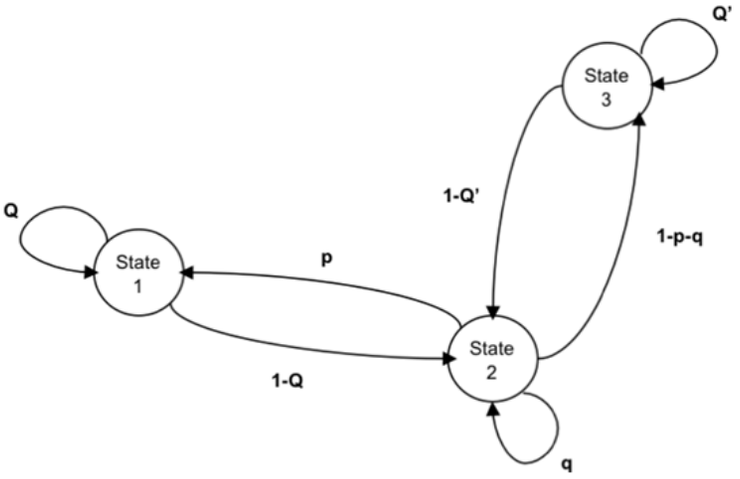

Figure 3. Packet Loss Model [33].

The transition probabilities are calculated from the following equations:

$$q = 1 - \frac{1}{\beta} \tag{2}$$

$$Q = 1 - \frac{1}{N_1} \tag{3}$$

$$Q' = 1 - \frac{1}{N_3} \tag{4}$$

$$p = \frac{1-Q}{Q-Q'}\left[\frac{1-Q'}{\alpha} + q + Q' - 2\right] \tag{5}$$

The authors in [33] suggest that an alternative to performing speech recognition tests using simulated channels is to define a set of packet loss characteristics, thus enabling recognition performance to be analysed across a range of different packet loss conditions. References [34-37] define four channels with different characteristics in order to simulate packet loss. These same four channels are used here to determine the effect of packet loss on speech recognition performance. The parameter values for the four channels are detailed in Table 6. These parameters result from work in [33] on IP and wireless networks. These are network environments over which a DSR system would typically operate. In an IP network, packet loss arises primarily due to congestion at the routers within the network, due to high levels of IP traffic. The nature of IP traffic is that it can be described as being 'bursty' in nature with the result that packet loss occurs in bursts. Signal fading, where the signal strength at a receiving device is attenuated significantly, is also a contributing factor to packet loss in a wireless network. Long periods of fading in a wireless network can result in bursts of packet loss. The authors in [33] measured the characteristics of an IP network and a WLAN, and the results showed the packet loss rate (α) and the burst length (β) to be highly variable. At one point or another, most channel conditions occurred, although not necessarily for long. Based on the experimental measurements, a set of packet loss characteristics was defined in [33] and these are used to analyse recognition performance for different network conditions. The parameters in Table 6 are taken from this defined set of packet loss characteristics.

	α	β	N_1	N_3
Channel A	10%	4	37	1
Channel B	10%	20	181	1
Channel C	50%	4	5	1
Channel D	50%	20	21	1

Table 6. Packet loss parameters.

4.2.2. Packet loss mitigation

Two error concealment methods are examined, namely nearest neighbour repetition and interpolation. These methods attempt to reconstruct the feature vector stream when packet loss is detected. Missing feature vectors are estimated solely from correctly received feature vectors. In a DSR system, nearest neighbour repetition and interpolation would both be

implemented on the server side. Additionally, interleaving, a technique used to reduce feature vector loss burst lengths (with a penalty of additional delay), is also briefly discussed. Interleaving is carried out on the client side of a DSR system, with de-interleaving on the server side.

4.2.2.1. Nearest neighbour repetition

The ETSI advanced front-end [7] specifies that where missing feature vectors occur due to transmission errors, they should be substituted with the nearest correctly received feature vector in the receiver. If there are $2B$ consecutive missing feature vectors, the first B speech vectors are substituted by a copy of the last good speech vector before the error, and the last B speech vectors are substituted by a copy of the first good speech vector received after the error. The speech vector includes the 12 static cepstral coefficients C_1-C_{12}, the zeroth cepstral coefficient C_0 and the log energy term, and all are replaced together. A disadvantage of this method is that if B is large then long stationary periods can arise.

4.2.2.2. Interpolation

The disadvantage of stationary periods that arise with nearest neighbour repetition can be alleviated somewhat by polynomial interpolation between the correctly received feature vectors either side of a loss burst. Reference [34] found that non-linear interpolation using cubic Hermite polynomials gives the best estimates for missing feature vectors. Equation (6) is used to calculate the n^{th} missing feature vector in a loss burst of length β packets, which is equivalent to a loss burst length of 2β feature vectors if each packet contains two feature vectors as defined by the ETSI advanced front-end [7]. The parameter n in (6) is the missing feature vector index.

$$\hat{x}_{b+n} = a_0 + a_1\left(\frac{n}{\beta+1}\right) + a_2\left(\frac{n}{\beta+1}\right)^2 + a_3\left(\frac{n}{\beta+1}\right)^3 \quad 1 \le n \le 2\beta \tag{6}$$

The coefficients a_0, a_1, a_2 and a_3 in (6) are determined from the two correctly received feature vectors either side of the loss burst, x_b and x_{b+n+1}, and their first derivatives, x'_b and x'_{b+n+1}. Equation (6) can be rewritten as

$$\hat{x}_{b+n} = x_b\left(1-3t^2+2t^3\right) + x_{b+\beta+1}\left(3t^2-2t^3\right) + x'_b\left(t-2t^2+t^3\right) + x'_{b+\beta+1}\left(t^3-t^2\right) \quad 1 \le n \le 2\beta \tag{7}$$

where $t = n/(\beta+1)$. It was found in [34] that performance was better when the derivative components in (7) are set to zero. These components are also set to zero for the work presented in this chapter.

4.2.2.3. Interleaving

Research has shown that by minimising the average burst length of lost vectors the estimation of lost feature vectors is more effective [34]. The aim of interleaving is to break a long loss burst into smaller loss bursts by distributing them over time and so making it appear that the errors are more randomly distributed.

In a DSR system, the interleaver on the client side takes a feature vector sequence X_i, where i is the order index, and changes the order in which the vectors are transmitted over the channel. The result is to generate a new vector sequence Y_i that is related to X_i by

$$Y_i = X_{\pi(i)} \tag{8}$$

where $\pi(i)$ is the permutation function. On the server side, the operation is reversed by de-interleaving the received vector sequence as follows:

$$X_i = Y_{\pi^{-1}(i)} \tag{9}$$

where $\pi(\pi^{-1}(i))=i$.

In order for the interleaver to carry out the reordering of the feature vectors, it is necessary to buffer the vectors, which introduces a delay. On the server side, in order to carry out the de-interleaving, buffering of the incoming feature vectors takes place and a second delay is introduced. The sum of these two delays is known as the latency of the interleaving/de-interleaving process.

The spread S of an interleaver is a metric that indicates how good an interleaver is at breaking up error bursts. A burst of length L vectors will be broken into bursts of length 1 if $S \geq L$. For $S < L$ the full distribution of the burst cannot be guaranteed and some sets of consecutive feature vectors may be lost.

For the work in this chapter, block interleaving is implemented. A block interleaver of degree d changes the order of transmission of a dxd block of input vectors. An example of a block interleaver of degree $d = 4$ and spread $S = 4$ is given in Figure 4.

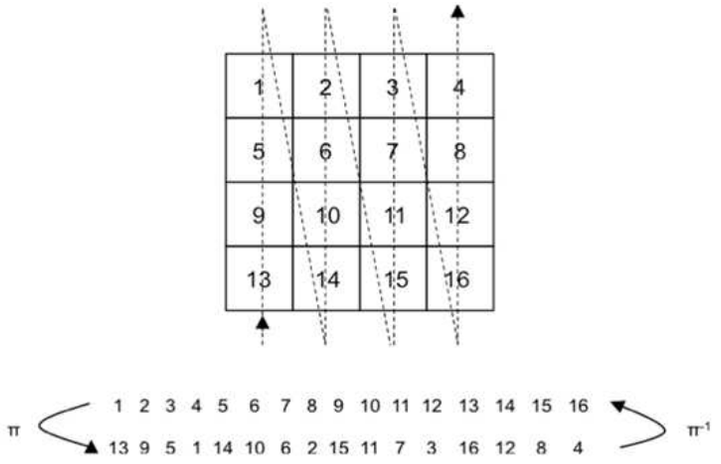

Figure 4. dxd block interleaver where $d = 4$, with permutation function.

4.3. Tests and results

The primary purpose of this section is to investigate the performance of the auditory model proposed by Li *et al*. [14] in combination with speech enhancement in the presence of noise

and packet loss. As a baseline for comparison, results are also presented for the ETSI advanced front-end [7]. In all cases, training was carried out using clean data. The speech enhancement algorithm of Agarwal and Cheng [25] was used on both the (clean) training speech as well as the (noisy) test speech. The ETSI advanced front-end includes a SNR-dependent waveform processing block that is applied after noise reduction and before feature extraction. The waveform processing block in the ETSI advanced front-end is also implemented in the front-end of Li *et al*. in order to ensure a closer match between the two front-ends. Feature vectors are extracted from the output of this waveform processing block. A detailed description of the waveform processing block can be found in [26]. The ETSI advanced front-end carries out post-processing in the cepstral domain in the form of blind equalization as described by [13]. The feature vectors produced by Li *et al*. undergo post-processing in the cepstral domain by means of cepstral mean subtraction (CMS). As defined by the ETSI advanced front-end [7], each packet transmitted over the communication channel carries two feature vectors.

In [7] a distributed speech recognition front-end feature vector compression algorithm is defined. The algorithm makes use of the parameters from the front-end feature extraction algorithm of the ETSI advanced front-end. The purpose of the algorithm is to reduce the number of bits needed to represent each front-end feature vector and so reduce the bit rate required over the communications channel. The feature vector is directly quantized with a split vector quantiser. The 14 coefficients (C_1 to C_{12}, C_0 & ln$E$) are grouped into pairs, and each pair is quantized using its own vector quantisation (VQ) codebook. The resulting set of index values is then used to represent the feature vector. The results documented in this paper are based on feature vectors that have undergone split vector quantisation.

The baseline recognition results for the two front-ends, without vector quantisation and with no packet loss but with noise, are detailed in Table 7. The word accuracies in the following tables are calculated as described in Section 2.4.

	Absolute word accuracy %			
Front-end	Set A	Set B	Set C	Overall
ETSI AFE	87.74	87.09	85.44	87.02
Li *et al*.	88.62	88.09	86.89	88.06

Table 7. Baseline recognition results.

In order to implement split vector quantization it is necessary to design VQ codebooks for each of the seven coefficient pairs. ETSI has made available script files for the ETSI advanced front-end and included with these are the VQ codebooks for the coefficient pairs. The recognition results for the ETSI advanced front-end with feature vector quantization using the ETSI supplied VQ codebooks are given in Table 8.

To allow for close comparison between the ETSI advanced front end and the front-end proposed by Li *et al*., the VQ codebooks for Li *et al*. should be determined in the same

manner as the VQ codebooks for the ETSI advanced front-end. However, this was not possible as the detail of how the ETSI advanced front-end VQ codebooks were calculated is not publicly available at this time. Therefore, an implementation of the Generalized Lloyd Algorithm (GLA), described by [39], was used to design the VQ codebooks for both the ETSI advanced front-end and the front-end of Li *et al*. The recognition results for the two front-ends using the VQ codebooks generated by the GLA implementation are detailed in Table 9. The overall word accuracies in Table 9, with vector quantization, compare well with the baseline accuracies, without vector quantization, in Table 7. There is also close correlation between the recognition results in Table 8 and Table 9 for the ETSI advanced front-end, indicating that the VQ codebooks generated by the GLA implementation used for this work are a good substitute for the VQ codebooks provided by ETSI with the advanced front-end.

	Absolute word accuracy %			
Front-end	Set A	Set B	Set C	Overall
ETSI AFE	87.81	87.11	85.74	87.12

Table 8. Recognition results using ETSI VQ codebooks.

	Absolute word accuracy %			
Front-end	Set A	Set B	Set C	Overall
ETSI AFE	87.73	86.92	85.41	86.94
Li *et al*.	88.22	87.59	86.55	87.63

Table 9. Recognition results with VQ codebooks designed using implementation of the GLA.

Packet loss (where each packet contains two feature vectors) is introduced on the communication channel by using the packet loss model described in Section 4.2.1. The four different packet loss channels investigated are defined in Table 6. Recognition tests, in the presence of packet loss, were carried out for each of the following conditions:

- no speech enhancement, no loss mitigation (Table 10);
- speech enhancement, no loss mitigation (Table 11);
- speech enhancement, nearest neighbour repetition (Table 12);
- speech enhancement, interpolation (Table 13);
- speech enhancement, interleaving, interpolation (Table 14).

Tests were first carried out for packet loss with no steps taken to recover the missing features or to minimise the loss burst length. The test results for both front-ends when no speech enhancement is used are given in Table 10, while recognition results with speech enhancement are given in Table 11. A comparison of Table 10 with Table 11 illustrates the benefit of using speech enhancement in improving recognition performance. Comparing Table 11 with Table 9 (no packet loss) it is seen that packet loss has a significant impact on the recognition results, in particular for channels C and D where the probability of packet loss is 50%.

	ETSI AFE absolute word accuracy %			
Loss parameters	Set A	Set B	Set C	Overall
Channel A	60.56	60.96	64.36	61.48
Channel B	59.31	59.49	63.67	60.25
Channel C	36.45	37.62	37.66	37.16
Channel D	35.38	35.89	37.68	36.04
	Li *et al*. absolute word accuracy %			
Loss parameters	Set A	Set B	Set C	Overall
Channel A	66.66	68.39	63.85	66.79
Channel B	65.59	67.35	62.93	65.76
Channel C	38.27	39.60	36.29	38.41
Channel D	37.78	39.25	36.07	38.03

Table 10. No speech enhancement, no error mitigation.

	ETSI AFE absolute word accuracy %			
Loss parameters	Set A	Set B	Set C	Overall
Channel A	80.17	80.09	77.97	79.70
Channel B	79.86	79.28	77.61	79.18
Channel C	42.50	42.87	40.50	42.25
Channel D	44.07	44.25	42.21	43.77
	Li *et al*. absolute word accuracy %			
Loss parameters	Set A	Set B	Set C	Overall
Channel A	81.08	80.64	79.37	80.56
Channel B	80.33	79.85	78.85	79.84
Channel C	43.35	43.53	41.91	43.13
Channel D	44.46	44.30	43.44	44.19

Table 11. Speech enhancement, no error mitigation.

Two methods, nearest neighbour repetition and Hermite interpolation, are used to reconstruct the feature vector stream as a result of missing features due to packet loss. Table 12 details the recognition results obtained when using nearest neighbour repetition while

Table 13 details the results obtained when Hermite interpolation is implemented (speech enhancement is used in both cases). Both reconstruction methods show improvements in recognition testing over no error mitigation for all four channels. In particular, with feature reconstruction channel C shows improvements in recognition accuracy greater than 55% for both front-ends. Channel D also shows good improvement. Nearest neighbour repetition gives a slightly higher performance compared to Hermite interpolation.

	ETSI AFE absolute word accuracy %			
Loss parameters	Set A	Set B	Set C	Overall
Channel A	84.04	83.34	81.69	83.29
Channel B	80.85	80.37	78.89	80.26
Channel C	68.90	68.25	66.83	68.23
Channel D	50.95	50.82	50.67	50.84
	Li *et al*. absolute word accuracy %			
Loss parameters	Set A	Set B	Set C	Overall
Channel A	84.57	84.00	82.67	83.96
Channel B	81.10	80.89	79.78	80.75
Channel C	68.81	68.60	66.87	68.34
Channel D	50.33	51.10	49.86	50.54

Table 12. Speech enhancement, nearest neighbour repetition.

	ETSI AFE absolute word accuracy %			
Loss parameters	Set A	Set B	Set C	Overall
Channel A	83.87	83.28	82.01	83.26
Channel B	80.81	80.21	78.78	80.17
Channel C	67.62	68.03	66.28	67.52
Channel D	50.33	50.16	48.91	49.98
	Li *et al*. absolute word accuracy %			
Loss parameters	Set A	Set B	Set C	Overall
Channel A	84.23	83.69	82.56	83.68
Channel B	80.87	80.60	79.64	80.51
Channel C	67.05	67.03	65.32	66.70
Channel D	50.33	50.16	48.91	49.98

Table 13. Speech enhancement, Hermite interpolation.

When interleaving is introduced, the receive side perceives that the average loss burst length is reduced [37]. Table 14 shows the recognition results obtained when interleaving, with an interleaving depth of 4, is used in conjunction with Hermite interpolation. Comparing the results in Table 14 with the results in Table 13 it is seen that feature reconstruction is improved when interleaving is employed.

	ETSI AFE absolute word accuracy %			
Loss parameters	Set A	Set B	Set C	Overall
Channel A	86.59	85.66	84.25	85.75
Channel B	82.65	81.89	80.51	81.92
Channel C	78.35	77.92	76.67	77.84
Channel D	58.52	58.77	59.88	58.89
	Li *et al*. absolute word accuracy %			
Loss parameters	Set A	Set B	Set C	Overall
Channel A	86.91	86.29	85.20	86.32
Channel B	82.84	82.15	81.50	82.30
Channel C	78.14	77.92	76.73	77.77
Channel D	57.95	58.13	57.05	57.84

Table 14. Speech enhancement, Hermite interpolation with interleaving ($d = 4$).

4.4. Discussion

The results in Table 7 show that the front-end proposed by Li *et al*. [14], when combined with the speech enhancement algorithm proposed by [25], reduces the overall word error rate of the ETSI advanced front-end [7] by 8%. Looking at Table 9, the vector quantisation has a lesser impact on the overall recognition performance of the ETSI advanced front-end compared with the impact of vector quantisation on the Li *et al*. front-end. The Li *et al*. front-end, combined with speech enhancement, still outperforms the ETSI advanced front-end in the presence of vector quantisation although the improvement in overall word error rate is reduced from 8% (without vector quantisation) to 5.3%. In the presence of packet loss, with no speech enhancement and with no packet loss compensation, a comparison of Table 10 shows that the front-end of Li *et al*. gives better overall recognition results than the ETSI advanced front-end. The benefit of speech enhancement in the presence of packet loss, without any missing feature reconstruction, can be seen by comparing Table 10 and Table 11. With speech enhancement and no packet loss compensation, Table 11 shows that Li *et al*. outperforms the ESTI advanced front-end for all four channels, and for all three test sets. Comparing Table 9 with Table 11, a significant reduction in recognition performance is observed in the presence of packet loss, in particular for channels C and D where the probability of packet loss is 50%. When nearest neighbour repetition is used to reconstruct missing features, Table 12 shows that there is a

significant increase in recognition performance across all channels when compared to the results presented in Table 11. Looking at Table 12, the recognition results for the two front-ends under evaluation are similar across all channels and test sets. The front-end of Li *et al*. performs marginally better overall than the ETSI advanced front-end for channels A, B and C; however, for channel D, the overall recognition performance of the ETSI advanced front-end is better than that of Li *et al*. A comparison of Table 13 with Table 12 shows a slight decrease in recognition performance when Hermite interpolation is used to reconstruct the feature vector stream instead of nearest neighbour repetition. With Hermite interpolation, Table 13 shows that the front-end of Li *et al*. outperforms the ETSI advanced front-end for the packet loss conditions of channels A and B, however, for channel C the reverse is the case. The overall performance of both front-ends is the same for channel D. Interleaving the feature vectors prior to transmission on the channel gives the perception on the receive side that the loss bursts are shorter than they actually are. The advantage of interleaving can be seen by a comparison of Table 14 with Table 13, where overall recognition results are improved for both front-ends when interleaving is introduced. Looking at Table 14 it is seen that Li *et al*. gives the better overall recognition performance for channels A and B while the ETSI advanced front-end gives the better performance for channels C and D. The results indicate that, in the presence of packet loss and environmental noise, the overall recognition performance of the front-end of Li *et al*. is better than that of the ETSI advanced front-end for all channel conditions when there are no packet loss mitigation techniques implemented. For each of the error mitigation techniques used, Li *et al*. outperforms the ETSI advanced front-end for channel conditions when the probability of packet loss is 10%. For packet loss probabilities of 50%, Li *et al*. gives better results than the ETSI advanced front-end for short average burst lengths (4 packets) when nearest neighbour repetition is used. However, the ETSI advanced front-end gives better recognition performance than Li *et al*. for the same channel conditions when Hermite interpolation is used, with and without interleaving. When the average burst length is increased to 20 packets and the probability of packet loss is 50%, the overall recognition performance of the ETSI advanced front-end is better than that of the front-end of Li *et al*.

5. Conclusions

This chapter has examined the speech recognition performance of both a speech enhancement algorithm combined with the auditory model front-end proposed by Li *et al*. [14], and the ETSI advanced front-end [7], in the presence of both environmental noise and packet loss. A number of speech enhancement techniques were first examined, including well-established techniques such as Ephraim and Malah [20] and more recently-proposed techniques such as Rangachari and Loizou [21]. Experiments using the Aurora connected-digit recognition framework [16] found that the best performance was obtained using the method of Agarwal and Chang [25]. The test results also suggest that the choice of speech enhancement algorithm for best speech recognition performance is independent of the choice of front-end.

Packet loss modelling using statistical modelling was also examined, and packet loss mitigation was discussed. Following initial testing with no packet loss compensation, a number of existing packet loss mitigation techniques were investigated, namely nearest

neighbour repetition and interpolation. Results show that the best recognition performance was obtained using nearest neighbour repetition to reconstruct missing features. The advantage of interleaving at the sender's side to minimise the average burst length of lost vectors was also demonstrated.

In summary, the experiments and results outlined in this chapter show the benefit of combining speech enhancement and packet loss mitigation to combat both noise and packet loss. Furthermore, the performance of the auditory model of Li *et al*. was generally shown to be superior to that of the standard ETSI advanced front-end.

Author details

Ronan Flynn
School of Engineering, Athlone Institute of Technology, Athlone, Ireland

Edward Jones
College of Engineering and Informatics, National University of Ireland, Galway, Ireland

6. References

[1] V. Digalakis, L. Neumeyer and M. Perakakis, "Quantization of cepstral parameters for speech communication over the world wide web", *IEEE Journal on Selected Areas in Communications,* vol. 17, pp. 82-90, Jan. 1999.

[2] D. Pearce, "Enabling new speech driven services for mobile devices: an overview of the ETSI standards activities for distributed speech recognition front-ends", in *Proc. AVIOS 2000: The Speech Applications Conference*, San Jose, CA, USA, May 2000.

[3] G. Gallardo-Antolín, C. Peláez-Moreno and F. Díaz-de-María, "Recognizing GSM digital speech", *IEEE Trans. on Speech and Audio Processing,* vol. 13, pp. 1186-1205, Nov. 2005.

[4] Z.H. Tan, P. Dalsgaard and B. Lindberg, "Automatic speech recognition over error-prone wireless networks", *Speech Communication,* vol. 47, pp. 220-242, 2005.

[5] Y. Ephraim and I. Cohen, "Recent advancements in speech enhancement", in *The Electrical Engineering Handbook*, 3rd ed., R.C. Dorf, Ed., Boca Raton, FL: CRC Press, 2006, pp. 15-12 to 15-26.

[6] "Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms", in *ETSI ES 201 108, Ver. 1.1.3*, Sept. 2003.

[7] "Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms", in *ETSI ES 202 050, Ver. 1.1.5*, Jan. 2007.

[8] "Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Extended front-end feature extraction algorithm; Compression algorithms; Back-end speech reconstruction algorithm", in *ETSI ES 202 211, Ver. 1.1.1*, Nov. 2003.

[9] "Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Extended advanced front-end feature extraction algorithm; Compression

algorithms; Back-end speech reconstruction algorithm", in *ETSI ES 202 212, Ver. 1.1.2*, Nov. 2005.

[10] "RTP Payload Format for European Telecommunications Standards Institute (ETSI) European Standard ES 210 108 Distributed Speech Recognition Encoding", in *RFC 3557*, July 2003.

[11] "RTP Payload Formats for European Telecommunications Standards Institute (ETSI) European Standard ES 202 050, ES 202 211, and 202 212 Distributed Speech Recognition Encoding", in *RFC 4060*, May 2005.

[12] D. Macho, L. Mauuary, B. Noe, Y. M. Cheng, D. Ealey, D. Jouvet, H. Kelleher, D. Perace, and F. Saadoun, "Evaluation of a noise-robust DSR front-end on Aurora databases", in *Proceedings of International Conference on Speech and Language Processing*, Denver, Colorado, USA, Sept. 2002, pp. 17-20.

[13] L. Mauuary, "Blind equalization in the cepstral domain for robust telephone based speech recognition", in *Proc. EUSIPCO 98*, Sept. 1998, pp. 359-363.

[14] Q. Li, F. K. Soong and O. Siohan, "A high-performance auditory feature for robust speech recognition", in *Proc. of 6th International Conference on Spoken Language Processing (ICSLP)*, Beijing, China, Oct. 2000, pp. 51-54.

[15] R. Flynn and E. Jones, "A comparative study of auditory-based front-ends for robust speech recognition using the Aurora 2 database", in Proc. IET Irish Signals and Systems Conference, Dublin, Ireland, 28-30 June 2006, pp. 111-116.

[16] H. G. Hirsch and D. Pearce, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions", in *Proc. ISCA ITRW ASR-2000*, Paris, France, Sept. 2000, pp. 181-188.

[17] H. Hermansky, "Perceptual linear prediction (PLP) analysis of speech", *J. Acoust. Soc. Amer.*, vol. 87, pp. 1738-1752, 1990.

[18] J. Tchorz and B. Kollmeier, "A model of auditory perception as front end for automatic speech recognition", *J. Acoust. Soc. Amer.*, vol. 106, pp. 2040-2050, 1999.

[19] *HTK speech recognition toolkit*. Available: http://htk.eng.cam.ac.uk/. Accessed March 2011.

[20] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator", *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 32, pp. 1109-1121, Dec. 1984.

[21] S. Rangachari and P. Loizou, "A noise-estimation algorithm for highly non-stationary environments", *Speech Communication*, vol. 48, pp. 220-231, 2006.

[22] N. Westerlund, M. Dahl and I. Claesson, "Speech enhancement for personal communication using an adaptive gain equalizer", *Speech Communication*, vol. 85, pp. 1089-1101, 2005.

[23] R. Martin, "Spectral subtraction based on minimum statistics", in *Proc. Eur. Signal Processing Conference*, 1994, pp. 1182-1185.

[24] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics", *IEEE Trans. on Speech and Audio Processing*, vol. 9, pp. 504-512, July 2001.

[25] A. Agarwal and Y.M. Cheng, "Two-stage mel-warped wiener filter for robust speech recognition", in *Proceedings of Automatic Speech Recognition and Understanding Workshop*, Keystone, Colorado, USA, 1999, pp. 67-70.

[26] D. Macho and Y. M. Cheng, "SNR-dependent waveform processing for improving the robustness of ASR front-end", in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2001, pp. 305-308.

[27] C. Peláez-Moreno, G. Gallardo-Antolín and F. Díaz-de-María, "Recognizing voice over IP: A robust front-end for speech recognition on the world wide web", *IEEE Trans. on Multimedia,* vol. 3, pp. 209-218, June 2001.

[28] C. Peláez-Moreno, G. Gallardo-Antolín, D.F. Gómez-Cajas and F. Díaz-de-María, "A comparison of front-ends for bitstream-based ASR over IP", *Signal Processing*, vol. 86, pp. 1502-1508, July 2006.

[29] J. Van Sciver, J. Z. Ma, F. Vanpoucke and H. Van Hamme, "Investigation of speech recognition over IP channels", in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2002, pp. 3812-3815.

[30] D. Quercia, L. Docio-Fernandez, C. Garcia-Mateo, L. Farinetti and J. C. DeMartin, "Performance analysis of distributed speech recognition over IP networks on the AURORA database", in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2002, pp. 3820-3823.

[31] P. Mayorga, L. Besacier, R. Lamy and J. F. Serignat, "Audio packet loss over IP and speech recognition", in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Nov. 2003, pp. 607-612.

[32] J. Vicente-Peña, G. Gallardo-Antolín, C. Peláez-Moreno and F. Díaz-de-María, "Band-pass filtering of the time sequences of spectral parameters for robust wireless speech recognition", *Speech Communication,* vol. 48, pp. 1379-1398, Oct. 2006.

[33] B.P. Milner and A.B. James, "An analysis of packet loss models for distributed speech recognition", in *Proc. of 8th International Conference on Spoken Language Processing (ICSLP)*, Jeju Island, Korea, Oct. 2004, pp. 1549-1552.

[34] A. B. James and B. P. Milner, "An analysis of interleavers for robust speech recognition in burst-like packet loss", in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2004, pp. 853-856.

[35] A. B. James and B. P. Milner, "Towards improving the robustness of distributed speech recognition in packet loss", in *Proc. Second COST278 and ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction*, University of East Anglia, U.K., Aug. 2004, p. paper 42.

[36] A. B. James and B. P. Milner, "Combining packet loss compensation methods for robust distributed speech recognition", in *Proc. of Interspeech-2005*, Lisbon, Portugal, Sept. 2005, pp. 2857-2860.

[37] A. B. James and B. P. Milner, "Towards improving the robustness of distributed speech recognition in packet loss", *Speech Communication,* vol. 48, pp. 1402-1421, Nov. 2006.

[38] B. P. Milner and A. B. James, "Robust speech recognition over mobile and IP networks in burst-like packet loss", in *IEEE Trans. on Audio, Speech and Language Processing* vol. 14, ed, Jan. 2006, pp. 223-231.

[39] Y. Linde, A. Buzo and R. Gray, "An algorithm for vector quantizer design", *IEEE Trans. on Communications,* vol. COM-28, pp. 84-95, Jan. 1980.