Estimating Articulatory Parameters from the Acoustic Speech Signal

Korin Richmond

Thesis submitted for the degree of Doctor of Philosophy
University of Edinburgh
2001

Dedication

To my parents, Heather and Graham, whom I admire for a million and one reasons — not least their determination never to stop trying, and for only ever teaching by example.

Acknowledgements

Thanks to all the folk at CSTR for providing a stimulating research environment, for educational and technical support, and for their patience while I dragged this thesis out to the bitter end! In particular, thank you to Paul Taylor for his inimitable style of supervision, which suited me down to the ground, and for his invaluable moral support at the viva. Thanks to my examiners, Martin Russell and Mark Steedman, for their even-handed and constructive criticism, and generally for all the effort to which they went. Thanks to Steve Isard, Alan Wrench, Simon King, Chris Williams, Sam Roweis, Mike Schuster and Jean-Pierre Martens for useful theoretical discussion and other help along the way. Thanks also to Sam Roweis and George Papcun for allowing me to include some of their work here. Big thanks to Mike Schuster for providing a useful base of code. Thank you to John McKenna for his timely disclosure of the secret two point guide to successful PhD writing: 1) Don't get it "right", get it written. 2) Completion by deletion... A huge thank you to all the people I have not mentioned here but still greatly deserve gratitude. I completely acknowledge I could not function without your help, and only hope to be able to reciprocate the favour in future. Finally, immense thanks to Catherine for all her sympathetic support.

Declaration

I declare that, apart from where properly indicated, the work contained in this thesis is entirely the product of my own work.

Abstract

A successful method for inferring articulation from the acoustic speech signal would find many applications: low bit-rate speech coding, visual representation of speech, and the possibility of improved automatic speech recognition to name but a few. It is unsurprising, therefore, that researchers have been investigating the acoustic-to-articulatory inversion mapping for several decades now.

A great variety of approaches and models have been applied to the problem. Unfortunately, the overwhelming majority of these attempts have faced difficulties in satisfactorily assessing performance in terms of genuine human articulation. However, technologies such as electromagnetic articulography (EMA) mean that measurement of human articulation during speech has become increasingly accessible. Crucially, a large corpus of acoustic-articulatory data during phonetically-diverse, continuous speech has recently been recorded at Queen Margaret College, Edinburgh. One of the primary motivations of this thesis is to exploit the availability of this remarkable resource.

Among the data-driven models which have been employed in previous studies, the feedforward multilayer perceptron (MLP) in particular has been used several times with promising results. Researchers have cited advantages in terms of memory requirement and execution speed as a significant factor motivating their use. Furthermore, the MLP is well known as a universal function approximator; an MLP of suitable form can in theory represent any arbitrary mapping function. Therefore, using an MLP in conjunction with the relatively large quantities of acoustic-articulatory data arguably represents a promising and useful first research step for the current thesis, and a significant part of this thesis is occupied with doing this.

Having demonstrated an MLP which performs well enough to provide a reasonable baseline, we go on to critically evaluate the suitability of the MLP for the inversion mapping. The aim is to find ways to improve modelling accuracy further. Considering what model of the target articulatory domain is provided in the MLP is key in this respect. It has been shown that the outputs of an MLP trained with the sum-of-squares error function approximate the mean of the target data points conditioned on the input vector. In many situations, this is an appropriate and sufficient solution. In other cases, however, this conditional mean is an inconveniently limiting model of data in the target domain, particularly for ill-posed problems where the mapping may be multi-valued.

Substantial evidence exists which shows that multiple articulatory configurations are able to produce the same acoustic signal. This means that a system intended to map from a point in acoustic space can be faced with multiple candidate articulatory configurations. Therefore, despite the impressive ability of the MLP to model mapping functions, it may prove inadequate in certain respects for performing the acoustic-to-articulatory inversion mapping.

Mixture density networks (MDN) provide a principled method to model arbitrary probability density functions over the target domain, conditioned on the input vector. In theory, therefore, the MDN offers a superior model of the target domain compared to the MLP. We hypothesise that this advantage will prove beneficial in the case of the acoustic-to-articulatory inversion mapping. Accordingly, this thesis aims to test this hypothesis and directly compare the performance of MDN with MLP on exactly the same acoustic-to-articulatory inversion task.

Contents

List of Figures

List of Tables

Acronyms and terms

ANN	Artificial neural network
ASR	Automatic speech recognition
EMA	Electromagnetic articulography
EPG	Electropalatography
FFT	Fast Fourier Transform
GMM	Gaussian mixture model
HMM	Hidden Markov model
LDM	Linear dynamic model
LPC	Linear prediction coefficients
LSP	Line spectral pairs
MDN	Mixture density network
MFCC	Mel-scale cepstral coefficient
MLP	Multilayer perceptron
MSE	Mean square error
pdf	Probability density function
RMSE/RMS error	Root mean square error
RNN	Recurrent neural network
SCG	Scaled conjugate gradients
XRMB	X-Ray microbeam

Chapter 1

Introduction

1.1 The nature of speech

It is easy to take communication by speech entirely for granted, never stopping to consider its mechanism or complexity. An objective look at the process of speech communication, however, quickly reveals it is a rather complex and even arcane process. Figure 1.1 provides a simplified overview of how humans communicate by speech. We can identify three distinct phases of speech communication: the speech signal itself (outside speaker and listener), the interpretation of the speech signal by the listener and the production of the speech signal by the speaker.

Clearly, the overriding purpose of speech is to convey some message from speaker to listener. Outside the speaker and the listener, the speech message exists and is transmitted as acoustic energy. Acoustic energy propagates through air by the oscillation of air particles in longitudinal waves of compression and rarefaction. These waves radiate out from the speaker and propagate to impinge upon the listeners' ears. The oscillations of the air particles are analysed and transduced by the listeners' ears and turned into electrical impulses, which are in turn conveyed on to the brain by nerves for further analysis. The process of transducing, processing and interpreting patterns of acoustic energy is referred to as "audition". This will not be discussed further. The only aspect of audition relevant here is that in order to extract meaning from the acoustic signal, there must clearly be patterns of oscillation which *mean* things. In other words, the

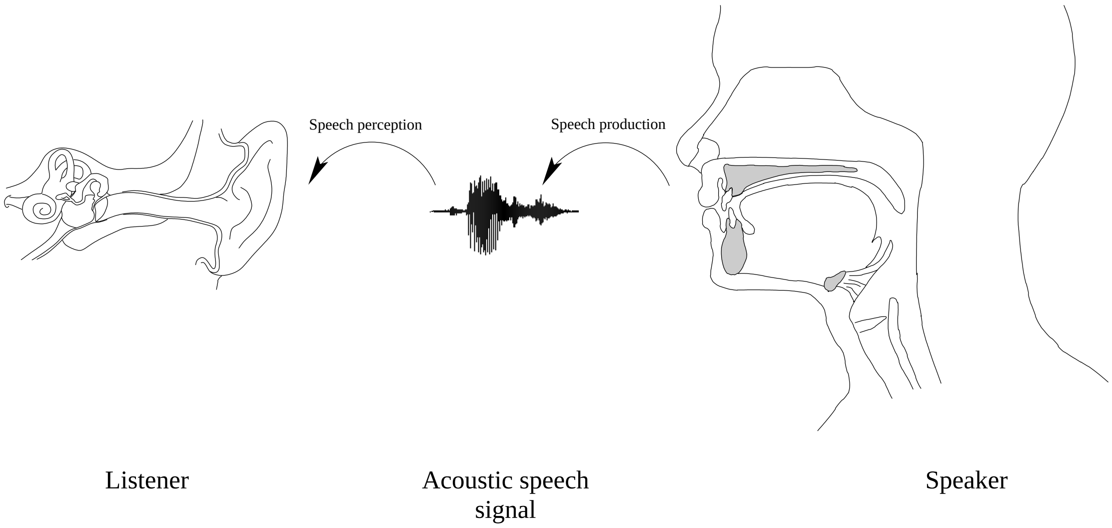


Figure 1.1: The human speech communication process. The speaker moves their articulators in sophisticated ways to produce and manipulate an acoustic speech signal, which propagates through the air. The sound waves impinge upon the listeners' ears, where the oscillations of the air particles are transduced and sent to the brain for further processing.

speaker and listener must agree that certain patterns of acoustic energy are significant, and what their significance is. Thus, in order to produce an intelligible message, the speaker must generate appropriate patterns of acoustic energy.

During "voiced" speech, a major source of acoustic excitation in speech production results from the vibrations of the vocal folds. Another possible source of acoustic energy is the hissing or "frication" caused by forcing air through a constriction in the vocal tract, for example between the tongue and the alveolar ridge (just behind the top incisors) during the production of an [s] phone. However, sources of acoustic energy like vocal cord vibrations and frication are not in themselves capable of producing the full range of variation present in the acoustic speech signal. Further modification and manipulation is necessary to enrich the signal.

When excited by an acoustic source, cavities within the vocal tract act like a filter and modify the spectrum of the source signal; frequency components in some regions of the spectrum will become attenuated for example. Humans have control over the configuration of these oral cavities by moving the tongue, lips, jaw and velum[1]. Consequently, humans can exert amazing control over the modification of an acoustic excitation signal. Different combinations of acoustic excitation and modification by the vocal tract produce distinctive sounds. As mentioned earlier, a speaker can convey information to the listener when they broadly agree on the "rules" or "customs" governing what sounds can go together and what those combinations of sounds mean. These customs form a large part of what we refer to as a language.

1.2 The inversion mapping

Articulation in the human vocal tract can be said to *cause* the acoustic speech signal. This relationship can be summed up as follows:

$$\mathbf{y} = f(\mathbf{x}) \tag{1.1}$$

where $\mathbf{x}$ is a feature vector describing a configuration of the articulators, $\mathbf{y}$ is a feature vector describing the acoustic signal that is produced, and $\mathbf{f}$ is the function relating these two states in the articulatory and acoustic domains. Every possible static configuration of the articulators, $\mathbf{x_a}$, will consistently produce a single acoustic signal $\mathbf{y_a}$. Put another way, the function $\mathbf{f}$ can map from any point in the space of possible articulatory configurations to a single point in the acoustic space.

But since the speech signal measured in the acoustic domain is the consequence of movements occurring in the articulatory domain, one can't help feeling that it must be somehow possible to invert this causative relationship. In other words, given a point in acoustic space $\mathbf{y_a}$, it is natural to ask whether it is possible to invert the function $\mathbf{f}$ and map to the corresponding point in articulatory space which might have produced the acoustic signal. This process is referred to as the acoustic-to-articulatory inversion mapping, or simply the inversion mapping for short.

[1]The velum principally determines whether the nasal and oral cavities are coupled or not.

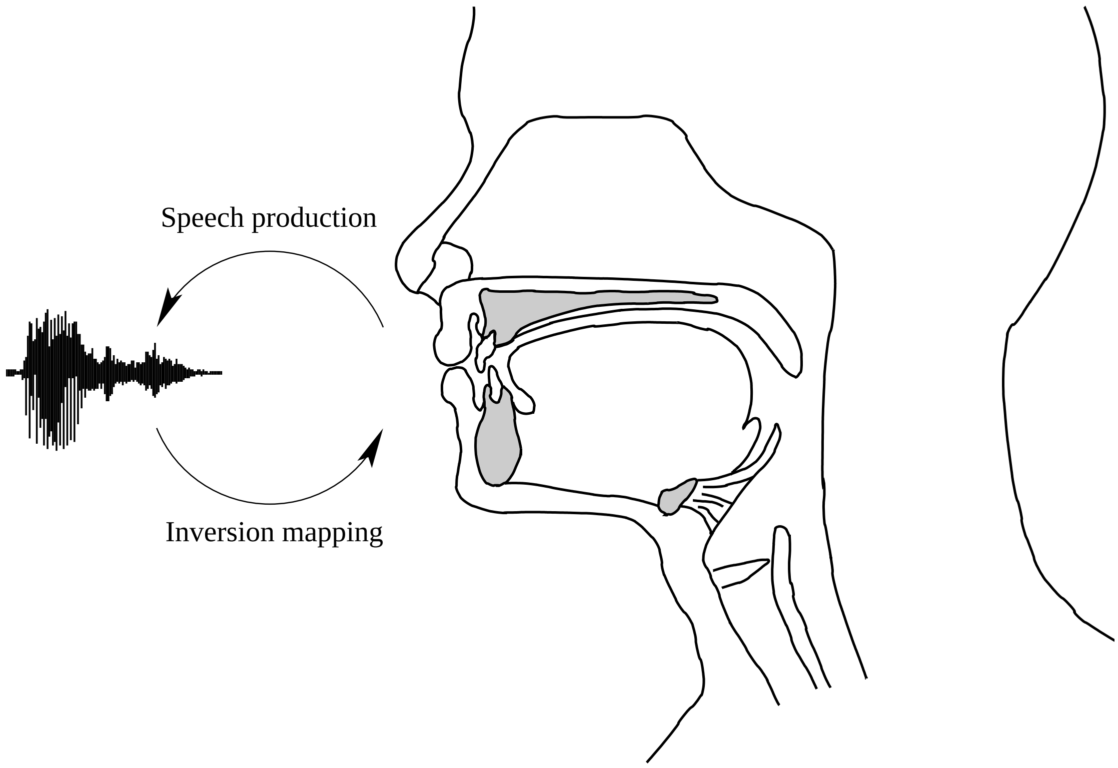


Figure 1.2: Estimating the articulation that might have produced an acoustic speech signal is called acoustic-to-articulatory inversion, or simply the inversion mapping, because in effect we are attempting to invert the normal process of speech production.

1.3 Applications for the inversion mapping

A method for estimating the articulatory trajectories which produced a given acoustic signal is more than just an interesting theoretical question. A successful system for mapping from the acoustic domain to the articulatory domain would find many applications. This section describes several of the most tantalising potential benefits of a successful inversion mapping.

1.3.1 Speech training

A method for inferring articulation from the acoustic signal in real-time could assist individuals undergoing speech training, whether for the treatment of speech disorders or simply to assist in learning the phone inventory and articulation of a foreign language. If a visual display of speech production were available, it would be easier for the user to gain an objective appreciation of their articulation.

The hearing-impaired could stand to benefit especially from visual feedback of their articulation. While it is more straightforward to teach hearing-impaired individuals to recognise and use labial and lingual articulatory gestures, natural proprioceptive appreciation of the velum is rare, and thus appropriate control of the velum during speech is a difficult skill to acquire (Chen 1995). Obviously, a system that provided visual feedback of velar movements would be a very useful tool when training to acquire this skill.

1.3.2 Lip-reading supplement

In addition to the use pointed out in the previous section, it has been suggested that an inversion mapping could provide additional benefit to hearing-impaired individuals. If it were possible to infer articulatory parameters reliably from the acoustic speech signal for any speaker, they could provide a visual representation of speech for hearing-impaired individuals when conversing with non-signing speakers. In lieu of an adequate speech recogniser, a display of an interlocutor's articulation could be used to supplement or replace the traditional external lip-reading cues (Schroeder 1967).

1.3.3 Speech coding

Since the articulators move relatively slowly and consistently, it has been suggested that a more parsimonious representation of speech could be realised in the articulatory domain compared to the acoustic domain (Schroeter & Sondhi 1992). This might prove beneficial anywhere that low bit-rate speech coding and transmission are required (Flanagan, Ishizaka & Shipley (1980),Chennoukh, Sinder, Richard & Flanagan (1997*a*)). For example, in the system of Chennoukh et al. (1997*a*), the input speech signal is first analysed to obtain the first two resonances of the vocal tract. Next, vocal tract shapes are estimated from these and encoded by 3 parameters. These parameters are transmitted over the channel together with the frequency of vocal cord vibration. At the receiver end, the vocal tract shape is restored, the formants calculated and the output speech synthesised by a formant synthesiser. Chennoukh et al. (1997*a*) claim to have obtained an intelligible output speech signal for bit-rates as low as 624 bits/s for voiced sentences (parameter sampling at 48 Hz, with 3 bits for each of 3 vocal tract parameters, and 4 bits for vocal cord vibration frequency). By way of comparison, techniques which aim to efficiently encode the time-domain representation of the speech signal, such as log-PCM (Pulse Code Modulation), adaptive PCM, adaptive delta modulation (ADM) and continuously variable slope delta modulation (CVSD), can typically encode speech signals down to about 16kbits/s. Meanwhile, some of the class of linear predictive coders (LPC) can operate with reasonable quality down to bit-rates of around 2.4kbits/s (Owens 1993).

The possibility of transmitting articulatory speech parameters as a low bandwidth representation of speech could prove useful in a system for speech recognition via mobile phone[2]. This application would suit a client-server approach to speech recognition. The mobile phone handset performs the acoustic analysis, having the benefit of access to the speech signal at full bandwidth. Then, the articulatory parameters are estimated from acoustics, possibly using a user specific mapping. These articulatory parameters vary slowly and therefore can be transmitted with low bandwidth over the phone link to the server, which performs the decoding and search part of recognition. This model

[2]personal communication with Alan Wrench, Queen Margaret University College

has the advantage of being cheap to implement. Instead of implementing a recogniser in every phone and transmitting the words, a single server could be set up to handle multiple calls simultaneously, which would serve thousands of phone users, who presumably don't all need to be using the recognition service all at once. The server is spared the computational burden of acoustic processing and articulatory feature extraction, which are best done in hardware, close to the source of the speech signal. Upgrading the server upgrades all the mobile recognition systems at once, which makes for a cheap path to improvement.

1.3.4 Speech visualisation

In some applications of avatars[3], it would be useful to be able to generate matching mouth movements where only an acoustic speech signal is available. For example, in the computer games industry, an inversion mapping could automatically generate realistic facial movements to accompany snippets of speech for game characters.

1.3.5 Speech recognition

Of course, it has often been suggested that automatic speech recognition (ASR) could capitalise on a method for estimating articulation from the acoustic speech signal. The current state-of-the-art in large vocabulary continuous ASR is due to Hidden Markov Models (HMM) based in the acoustic domain. These models have the advantage of being efficient and flexible. They have indisputably pushed ASR performance far in advance of previous technologies.

However, acoustics-based HMMs have not yet solved the speech recognition problem entirely. Some suggest that the recognition performance that we are ultimately aiming for is beyond the capabilities of acoustic-based HMMs, and that no amount of tweaking or elaboration will achieve our goal. Consequently, it is important to explore different recognition paradigms, even though the initial results may be below that of state-of-the-art HMMs (Bourlard, Hermansky & Morgan 1996).

[3] artificially generated "talking heads"

Compatibility of model and representation

Adequate information about the phonemic content of an utterance is of course generally present in the acoustic waveform. If this were not the case, then human listeners would not be able to understand speech[4]. However, the information in a signal can be more perspicuous in certain representations than in others, and so the choice of representation is of paramount importance. For example, the acoustic waveform itself is not readily analysable in visual form as a sound pressure versus time waveform by humans. This is illustrated in Figure 1.3. It is possible to identify certain gross characteristics of speech patterns in the acoustic waveform, and on that basis perhaps perform some rough segmentation into a string of component phones, but the information that may be gleaned from this time-domain acoustic representation is very limited. By transforming the speech signal into the spectral domain using Fourier analysis, a lot more information becomes readily accessible. In fact, trained linguists can often identify unknown utterances given a spectrogram alone. ASR systems too, like humans, are generally more successful when working with speech signals in the spectral domain. Representation in the spectral domain better suits the pattern matching methods we have at our disposal, and in particular is well suited for use with HMMs in most respects.

Unfortunately, it does not follow that it is easy to instill all the linguists' expertise into an automatic speech recogniser. There remain significant aspects of human speech which make it difficult to build successful speech recognition systems. When a speaker utters a word, we can think of the utterance as a string of phones. However, this string is not a simple concatenation of constant phones in the acoustic domain. The concatenated phone string is produced by a physical system — the articulators. The human speech production system is subject to certain constraints, such that the manifold of speech sounds is restricted to lie within the regions corresponding only to the possible articulator configurations. Moreover, the articulators can only move with limited velocity and acceleration, which introduces artifacts into the transitions between phones. On top of this, further variation is introduced by articulatory "customs" within

[4] Although, the human ability to make inferences about content on the basis of contextual information is noteworthy.

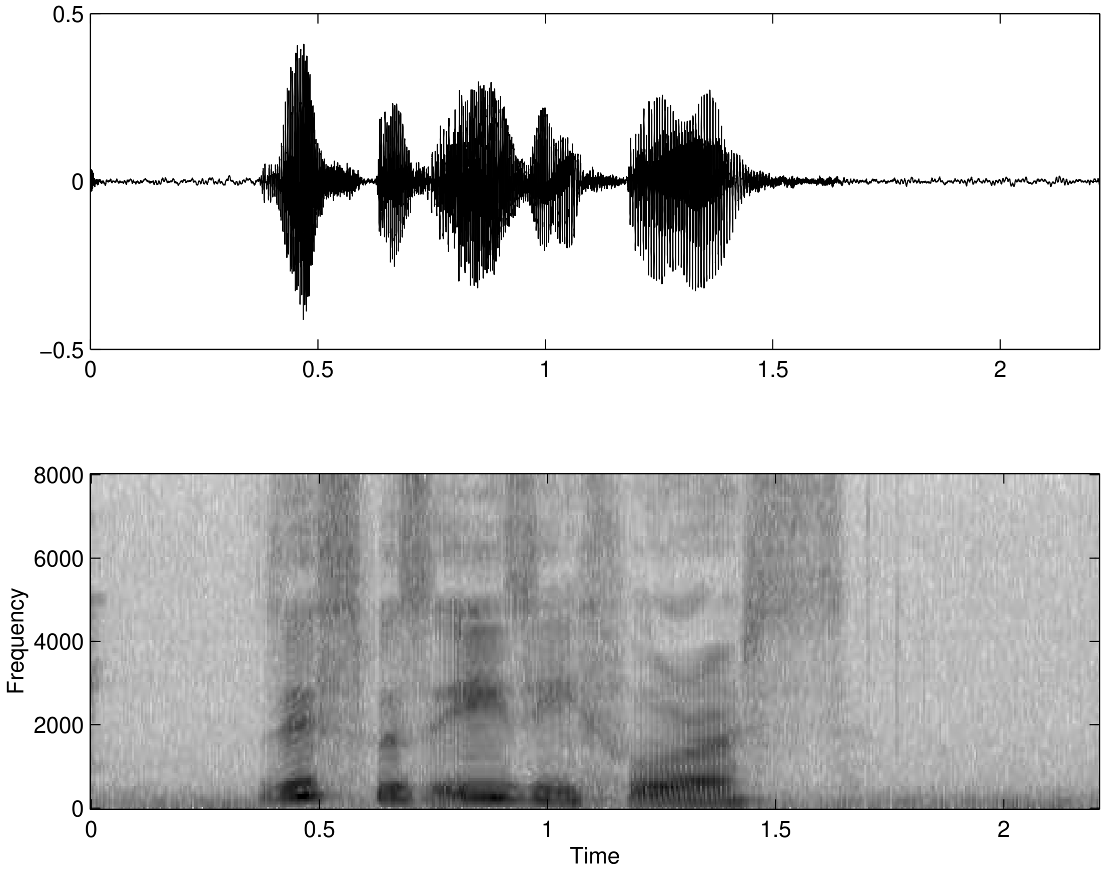


Figure 1.3: Acoustic waveform and spectrogram derived therefrom for the phrase "Is this seesaw safe?", uttered by speaker `fsew0`. The frequency-domain representation of speech in the spectrogram generally makes the boundaries and characteristics of the phonemic content of an utterance more readily identifiable than in the waveform time-domain representation. In fact, experienced linguistics may often guess mystery utterances by examining the spectrogram alone!

a language. Although one way of producing the transition between two sounds may be entirely straightforward physiologically, over time the speakers of a language may come to "agree" to use another pattern[5].

These articulatory constraints and customs, which cause phones in a sequence to become altered by their neighbours, are collectively referred to as coarticulation. Variability due to coarticulation is a major obstacle to continuous speech recognition. One of the potential limitations of the HMM for ASR is the so-called "piecewise stationary assumption", where speech is modelled as a string of independent constant regions. We know this does not correspond to the reality of speech production; in fact, there is nothing unique to speech within the formulation of the HMM for ASR. In order to alleviate the effects of coarticulation in the HMM speech recogniser, triphone models have been introduced, where a separate model is used for each phone in all possible contexts. The disadvantage of this approach is that an explosion in the complexity of the system results. With the increased number of parameters to be trained, data sparsity becomes a real problem.

Meanwhile, just as the HMM is ill-suited to provide an elegant model of coarticulation in the speech process, it may be that coarticulation is not best represented in the acoustic domain. There are reasons to suggest that an articulatory(-like) domain may be better.

Articulatory representation in speech perception and ASR

The exact role articulation plays in the process of decoding the speech signal in humans is highly contentious. For example, Callan, Callan, Kroos & Vatikiotis-Bateson (2000) proposed that speech perception may rely on the use of global mappings between auditory, motor, visual, and orosensory modalities. They recorded and analysed gamma band electrical brain activity with electrodes positioned on the scalp of a human listener. The subject was instructed to listen to various audio signals, which had been purposely degraded with various amounts of noise (multispeaker babble). The subject demon-

[5]For example, the epenthetic 'r', inserted between words ending and beginning with vowels (e.g. "quota allowance"), is common in English speech, but apparently disliked by speakers of other languages, such as Swedish.

strated a higher level of intelligibility of a degraded auditory speech signal when viewing a video of the speaker's facial movements at the same time. Callan et al. (2000) claimed this suggests visual information is influencing auditory speech processing. Furthermore, they reported that areas of the brain known to be associated with motor control of speech were more active when listening to speech degraded by noise. This suggests that these areas may be involved with global processes of speech perception.

McGowan & Faber (1996) introduced a collection of articles which discussed speech recognition and speech perception from an articulatory point of view. For his part in the discussion, Ohala (1996) argued that human speech perception does not rely upon the recovery of articulation, advancing three sources of evidence to support his claim: (a) phonological data from various languages indicate that it is primarily the acoustic-auditory properties of speech sounds which determine their behaviour and form. (b) Infants and nonhuman species, although they are themselves non-articulate, can differentiate between sound contrasts in human speech. (c) Humans can differentiate many complex sounds, such as bird or monkey calls or even music, for which they have little hope of forming a mental model of the underlying sound production mechanism. The strength and relevance of these arguments were contested by O'Shaughnessy (1996).

Meanwhile, some believe that articulation is of paramount importance to human speech perception. Fowler (1996) argued for her own "direct-realist" theory of speech perception. This theory is similar to the "Motor Theory" view of speech perception (Liberman & Mattingly 1985) in that they both maintain that the smallest perceivable "primitives" of speech perception are linguistic "gestures". Browman & Goldstein (1992) likewise viewed speech as being comprised of a system of linguistic gestures. In their view, gestures are the invariant atomic units of speech, which should not be confused with their articulatory or acoustic manifestation.

Finally, there are still others who argue that neither acoustics nor articulation has the upper hand in human speech perception. For example, Lindblom (1996) expounded his view that speech is not composed of invariants "wrapped in noise", but is the product of principled adaptations which respect the needs of the listener (Lindblom 1990).

Regardless of whether humans use any sort of explicit or implicit articulatory information when listening to speech, there are significant reasons to justify interest in in-

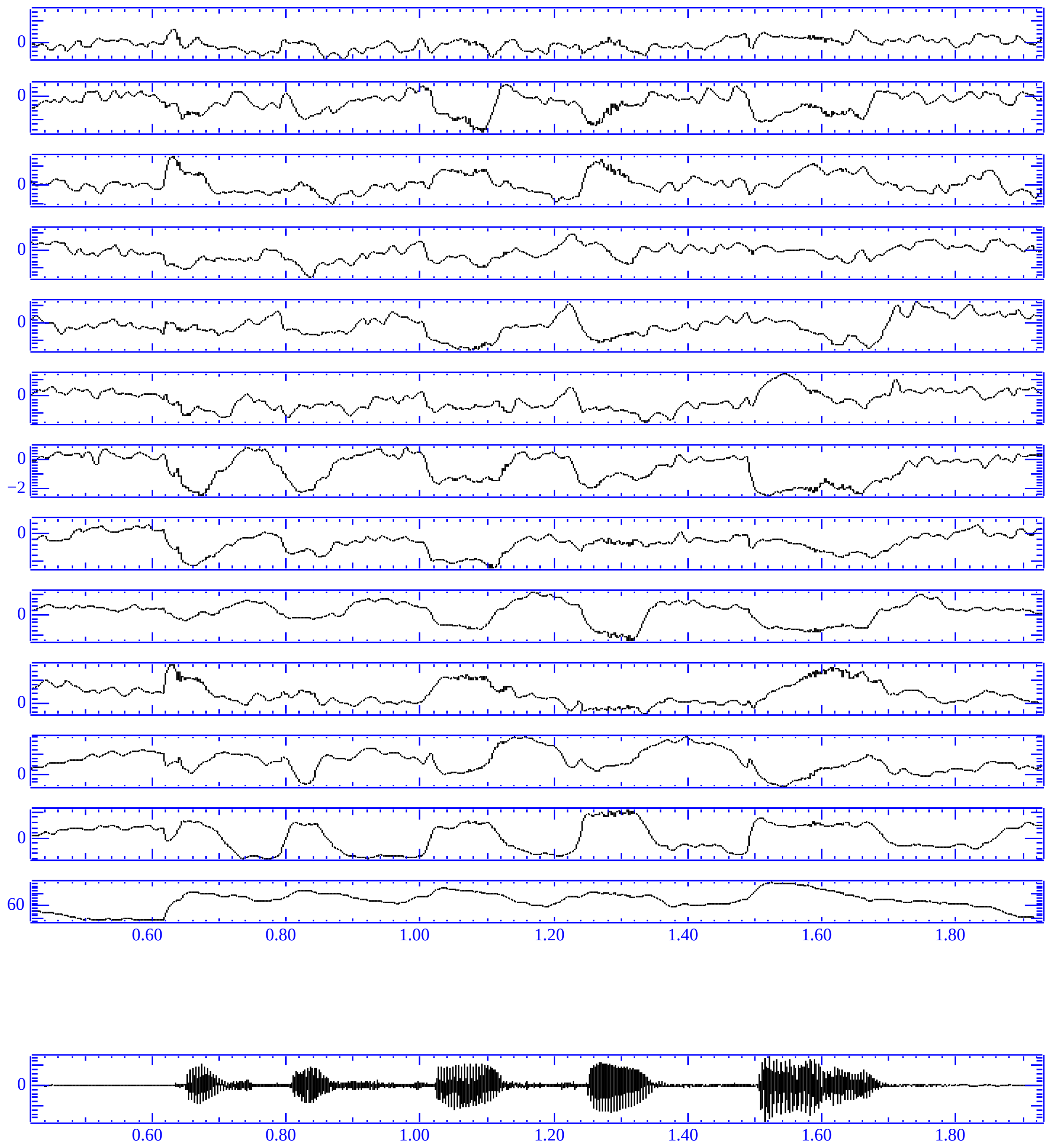


Figure 1.4: Acoustic waveform and derived melcepstra for the phrase "Is this seesaw safe?", uttered by speaker `fsew0`

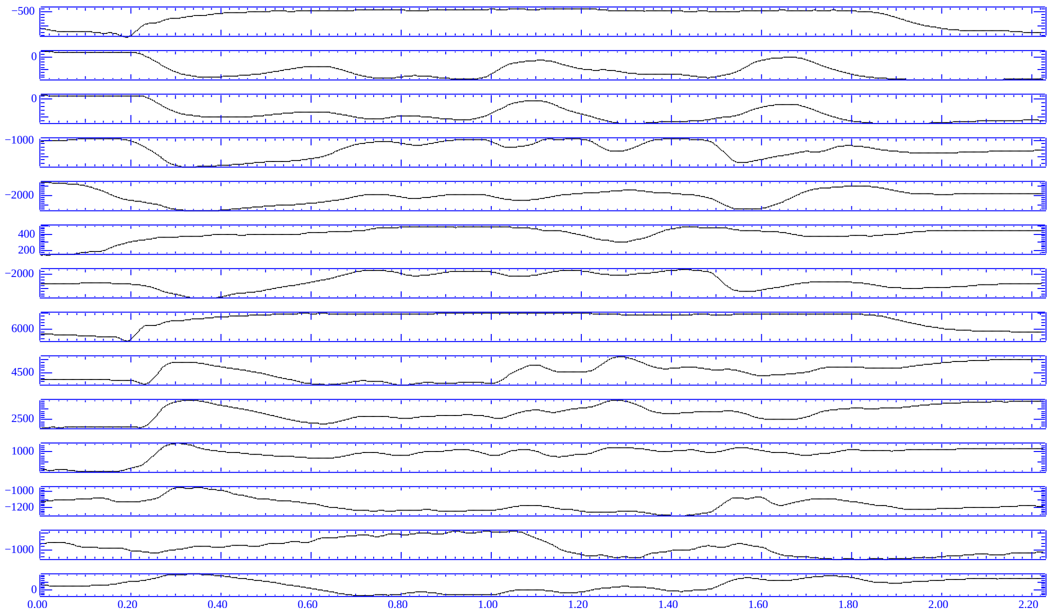


Figure 1.5: Smoothed EMA data for the phrase "Is this seesaw safe?", spoken by speaker `fsew0`. The EMA channel names can be found in Table 3.1. (The 'raw' EMA trajectories have been filtered to remove the noise introduced by measurement error.)

corporating articulatory components in ASR systems. Rose, Schroeter & Sondhi (1996) (and Rose, Schroeter & Sondhi (1994)) covered various aspects of research in this direction, which were further discussed by Moore (1996) and Nearey (1996).

There are certain theoretical advantages to the representation of speech in the articulatory domain. Since coarticulation occurs in the articulatory domain, it may be most straightforward to represent and model coarticulation in the articulatory domain. Generally, articulatory trajectories are smoothly and slowly varying in nature compared with acoustic parameters, which, not being subject to the constraints of the movements of the physical system of the mouth in a linear way, may vary rapidly and abruptly. For example, Figure 1.4 shows the acoustic waveform and the derived cepstral coefficients for the utterance "Is this seesaw safe?" We see that although the lower order coefficients (at the bottom of the figure) may vary slowly, the higher order coefficients become increasingly 'erratic' is nature. Figure 1.5 shows the speaker's articulatory movements recorded for the same utterance by EMA[6] for comparison.

[6] See Chapter 3; Section 3.2.1 in particular

ASR with articulatory features

Several research groups have been working on speech recognition systems which use an articulatory representation of speech either to supplement or to replace acoustic parameters: for example Zlokarnik (1995*b*); Hogden & Valdez (2000); Richard, Lin, Zussa, Sinder, Che & Flanagan (1995); Luettin (1997); Blackburn & Young (1995*c*) and Blackburn & Young (1996); Erler & Freeman (1996); Deng, Ramsay & Sun (1997); Frankel, Richmond, King & Taylor (2000).

Zlokarnik (1995*a*) combined acoustic parameters with articulator positions measured by EMA for two subjects in a speaker-dependent isolated word[7] recognition test by HMM. He reported that an HMM trained and tested with the combined acoustic-articulatory input was capable of reducing the error rate of comparable acoustics-only HMMs by a relative percentage of roughly 60%. When the articulatory parameters during the testing phase were estimated by multilayer perceptron (MLP), the relative error reduction was 18-25%.

Hogden & Valdez (2000) reported on how they have combined MO-MALCOM (described in Section 2.3.4) with standard speech recognition algorithms to build a speech recognition system. The articulatory-like trajectories in their system were inferred from acoustics using a Maximum Likelihood principle, without any explicitly measured human articulatory data. The ability to approximate a mapping between acoustics and articulation without requiring the availability of articulatory data is potentially of crucial significance for speech recognition. Unfortunately, this approach to automatic speech recognition based on MO-MALCOM still has a long way to go, achieving only a 40% recognition rate for the training set during a speaker-dependent isolated word-recognition task.

Work is currently being conducted at the Centre for Speech Technology Research (CSTR), on using Linear Dynamic Models (LDM) in conjunction with measured articulatory trajectories for speech recognition (Frankel & King 2001*a*). The intention behind using LDMs is to do away with the piece-wise stationary assumption in the HMM and provide a continuous model of the speech process. Using articulatory trajectories as observation features suits the LDM model well because they are continuous, smoothly

[7] the words were V_1CV_2 sequences

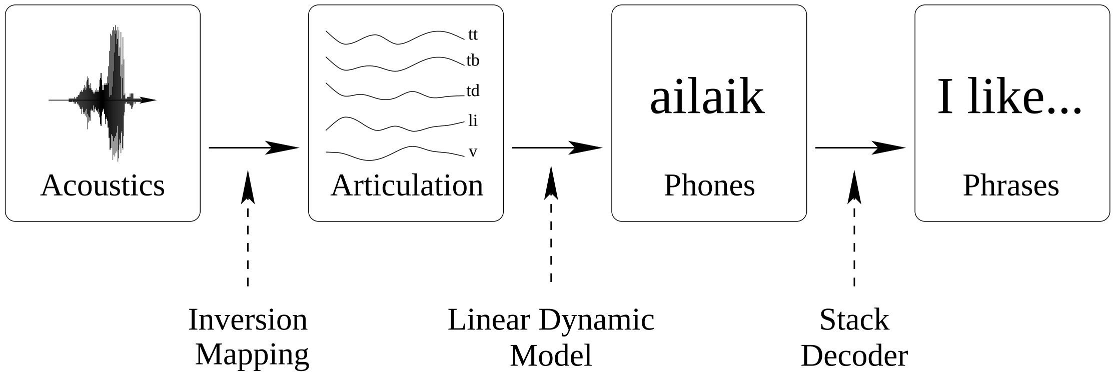


Figure 1.6: An ASR system proposed at CSTR which uses LDMs and articulatory parameters. For an automatic speech recognition system based on an articulatory representation of speech to be of practical use, there must be some reliable method for deriving an articulatory representation of the speech signal from the acoustic waveform. Hence, ASR is a potent motivation for research into acoustic-to-articulatory inversion.

varying parameters (c.f. Figures 1.4 and 1.5), possibly with the addition of some noise. However, although measured articulatory trajectories may be available for training the LDMs, a practical speech recogniser can only take acoustic input. Therefore, a working speech recogniser based in the articulatory domain would require articulatory parameters to be estimated from the acoustic signal alone. Figure 1.6 indicates how the overall system might be envisaged to work.

1.4 Motivation for this thesis

Developments in human articulography techniques mean it has become easier than ever to collect human articulatory data. However, the techniques currently available are still rather invasive, a factor which would prohibit their use outside the laboratory. It may be that some safe, non-invasive articulography technology will one day be developed that can directly measure the movements of the articulators during everyday speech[8]. For the time being at least, it is only convenient to record the acoustic signal.

[8] For example, perhaps a descendent of the technique reported in Holzrichter, Burnett, Ng & Lea (1998), who are working on a system using low power EM-wave sensors which could ultimately provide a totally non-invasive method for measuring speech articulator movements

Notwithstanding the difficulties in obtaining articulatory measurements of speech, several potential applications would benefit greatly from such a representation, as outlined in Section 1.3. Since the speech signal measured in the acoustic domain is caused by movements occurring in the articulatory domain, it is natural to consider attempting to infer articulation from acoustics. It is therefore unsurprising that the task of mapping from the acoustic domain to the articulatory domain has occupied researchers for several decades.

As we shall see in Chapter 2, a broad array of techniques and resources have been applied to the inversion mapping problem. While many promising results have been obtained, a dependable and widely-applicable inversion mapping method has so far proved elusive. One of the principle motivations prompting the work in this thesis is to exploit the the relatively large amount of human parallel acoustic-articulatory speech data[9] which has become available only recently.

Finally, it may turn out that a universal inversion mapping method, capable of recovering the trajectories of all articulators at all times and with a high level of accuracy, is unfeasible. Even so, it may still be possible to estimate articulatory parameters from acoustics well enough to provide benefit in at least some of the potential applications. It may be that the requirements of specific applications of an inversion mapping system will dictate whether there is any benefit to be gained. The application which holds most interest for the author, and which complements other research being carried out at CSTR, is speech recognition. Therefore, it is emphasised that the present research into the inversion mapping is underlyingly motivated by the potential requirements of an ASR system.

1.5 Publications

Some of the material contained in this thesis has appeared in other articles, published at various stages during the period of study for this thesis: Richmond (1999); King, Taylor, Frankel & Richmond (2000); Frankel et al. (2000); Wrench & Richmond (2000); Richmond (2001*a*); Richmond (2001*b*). However, apart from where stated otherwise,

[9]Described in Chapter 3.

none of the material contained in this thesis is the product of collaborative work.

Chapter 2

Previous Work

2.1 Introduction

We can identify two central and closely related questions in the literature on the inversion mapping. They can be summarised as "Is the inversion mapping possible?" on one hand and "How can we perform the inversion mapping?" on the other. This chapter will consider both these questions in turn. First, we will cover some of the evidence and arguments put forward in the literature which indicate that the inversion mapping is a very difficult problem. We will then look at several of the more notable attempts at recovering articulatory parameters from the acoustic speech signal throughout the research into acoustic-to-articulatory inversion conducted over the last three decades.

2.2 Inversion mapping as an ill-posed problem

A well-posed problem may be defined as having three characteristics:

1. The solution can be shown to exist under appropriate conditions
2. The solution is unique
3. The solution varies continuously with the data

Where any of these three conditions does not apply, the problem is said to be *ill-posed* (Borowski & Borwein 1991). Over the years, numerous researchers have highlighted the existence of non-uniqueness in the mapping from the acoustic speech signal to articulation. This suggests that the inversion mapping is an ill-posed problem, which could pose a potentially insurmountable obstacle to recovering articulation from the acoustics. If two or more different articulatory configurations are capable of producing the same acoustic signal, how can an inversion mapping system decide between the alternatives?

The evidence which indicates that the inversion mapping is ill-posed can be categorised as coming from four main sources; theoretical analysis, human experimental evidence, manipulation of articulatory synthesis models, and from analysis of measured human articulatory data.

In theory, the filtering effect of a tube which has two cavities of differing sizes in series along its length is indistinguishable from the equivalent tube where the order of the cavities has been reversed.

Human experimental evidence from as early as the 1920's has demonstrated that a speaker can produce acoustic signals the same as a target sound when the jaw is fixed in position by a bite-block. Lindblom, Lubker & Gay (1979) compared the formant frequencies for four Swedish vowels (/i,u,o,a/) produced by six speakers both when the jaw was free to move and when fixed in a certain position by a bite block either 2.5mm or 22.5mm thick. They found that despite the physiologically unnatural position of the jaw enforced by the bite blocks, the speakers were able to produce vowels with formant patterns well within the range of variation of a set of vowels spoken under normal conditions. Moreover, they comment that to achieve this, the speakers did not require any training time or practice. The ability of a speaker to vary their articulation to produce the desired speech sound has been termed *articulatory compensation.* An anecdotal example of the potential extent of articulatory compensation is the ventriloquist, who can produce an intelligible speech stream while almost completely hiding external signs of articulation.

Work done using articulatory synthesis models has raised several doubts concerning the viability of acoustic-to-articulatory inversion. For example, Atal, Chang, Mathews & Tukey (1978) placed significant emphasis on investigating what they called "fibers"[1] in the articulatory space, and applied their inversion method to studying them (see Section

[1] Atal et al. attribute this term to Husemoller(1975), although do not appear to give a full reference

2.3.2 for a brief description). A fiber is defined as a region in articulatory space within which movement produces little or no change in the corresponding acoustic output. Atal et al. described an iterative method for exploring the extent of such fibers, and gave an example in the case of vowels. They demonstrated, for example, that the mouth opening of their model could vary considerably without affecting the formants characteristic of an /i/ vowel. They went on to show examples for several vowels of vocal tract area functions whose first three formant frequencies are identical.

Roweis (1999) used human articulatory data measured by X-ray microbeam to provide convincing empirical evidence that the instantaneous mapping from acoustics to articulation is ill-posed. The database at his disposal contained approximately 175,000 samples of midsagittal articulatory configurations, which he calls "frames". From this, he compiled a data set of acoustic-articulatory vector pairs. The acoustic feature vectors consisted of line spectral pairs (LSP) calculated over a 23.5ms window (512 samples) centred on each articulatory sample time point. The articulatory feature vector comprised the x- an y-coordinates of eight articulator points.

Roweis then sorted the data on the basis of a distance metric in the articulatory and acoustic domain; for each vector pair, he located the K nearest neighbours in articulatory space, and the K nearest neighbours in acoustic space. In both cases, distance was defined according to a Mahalanobis distance metric, based on the global covariance of the articulatory or acoustic data respectively. Figures 2.1 and 2.2 show the result of this analysis using K=1000 nearest neighbours in scatter plot form. These scatter plots were produced by plotting the points in articulatory space defined by the articulatory feature vector of some frame f and the 1000 nearest neighbours that were found in acoustic space. As can be seen, the points in articulatory space do not fall within a tightly constrained area. Instead, we can see wide spreads (Figure 2.1) and even bimodal distributions (Figure 2.2) of points in articulatory space that correspond to neighbouring points in acoustic space. This indicates the inversion mapping is ill-posed.

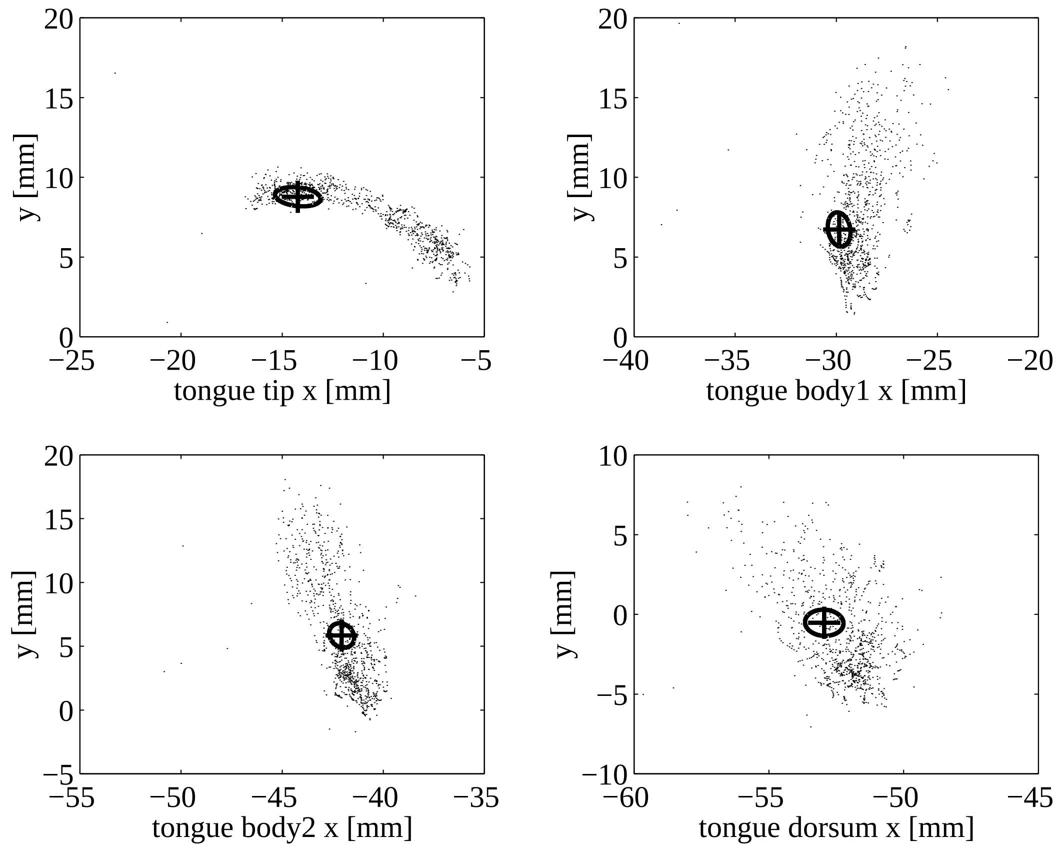


Figure 2.1: Empirical evidence that the inversion mapping is ill-posed from X-ray microbeam data. These plots are produced by taking the point in acoustic space of some reference frame, finding the 1000 nearest neighbours in acoustic space, then plotting the associated points in articulatory space. The '+' symbol indicates the location of the reference frame, while the ellipse shows the two-standard deviation contour of the 1000 nearest frames calculated in articulatory space. In this example, we can see regions spread throughout articulatory space that map to 1000 points in acoustic space which are close neighbours. This indicates that trying to estimate articulation from acoustics is an ill-posed problem. Reproduced with permission from Roweis (1999)

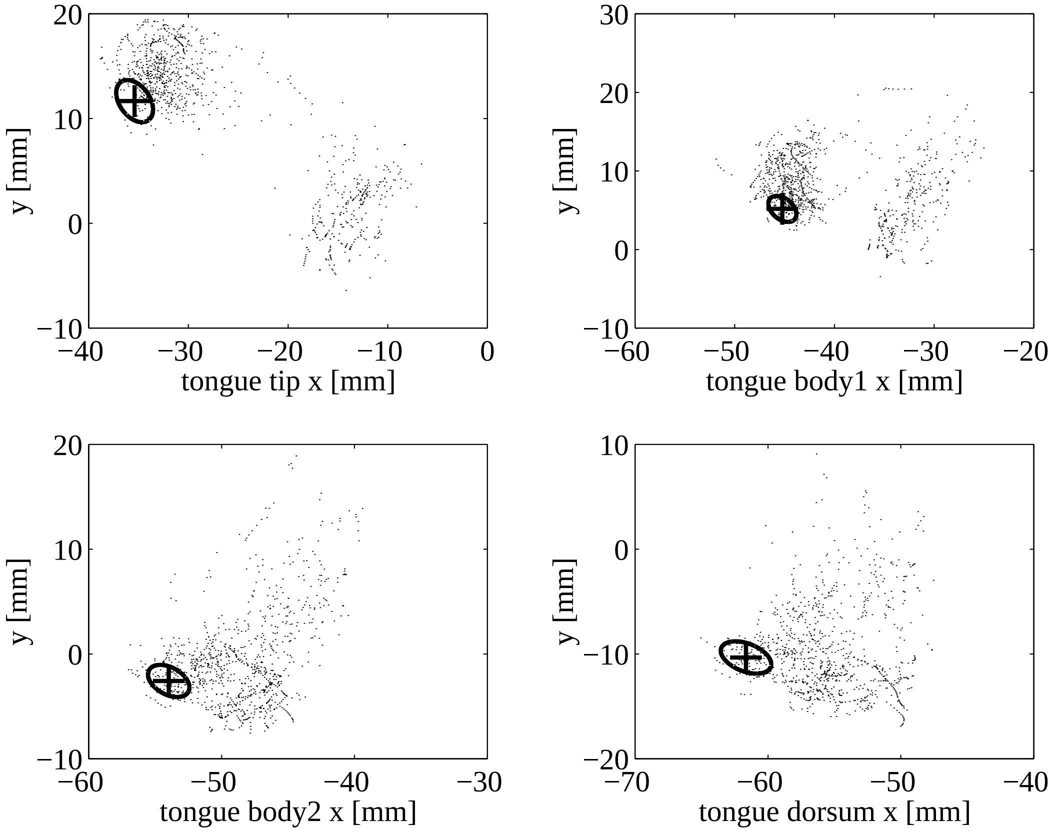


Figure 2.2: Empirical evidence that the inversion mapping is ill-posed from X-ray microbeam data. These plots are produced by taking the point in acoustic space of some reference frame, finding the 1000 nearest neighbours in acoustic space, then plotting the associated points in articulatory space. The '+' symbol indicates the location of the reference frame, while the ellipse shows the two-standard deviation contour of the 1000 nearest frames calculated in articulatory space. In this example, we can see an apparently multimodal distribution of points in articulatory space that map to the 1000 closely neighbouring points in acoustic space. This is characteristic of an ill-posed problem. Reproduced with permission from Roweis (1999)

2.3 Previous attempts at inversion

Evidence such as that covered in Section 2.2 would seem to suggest that the inversion mapping is at least extremely challenging, if not impossible. Nevertheless, a whole sub-field of speech research has devoted itself to tackling the task of recovering articulation from the acoustic speech signal. In this section, we look at the breadth of the approaches to acoustic-to-articulatory inversion that have been previously described.

Much early investigation was based on analytical techniques, such as inverse filtering. Later, articulatory synthesis models became a popular aid to studying the inversion problem. More recently, technologies such as X-ray microbeam cinematography and electromagnetic articulography have made it possible to record actual articulator movements in parallel with speech acoustics in a minimally-invasive way. This, together with increases in computer power, has made it feasible to train data-driven machine learning models on reasonably large quantities of parallel human acoustic-articulatory data.

Because speech researchers have been attempting to perform the inversion mapping for such a long time, a considerable body of material has accrued. This section does not assume the responsibility of providing an exhaustive account of that history. Instead, the aims of this section are twofold. First, it is the intention to provide the reader with an impression of the scope and diversity of the field, highlighting the major research strands. Second, the emphasis is placed very much on previous work which has made use of measured human articulatory data in conjunction with machine learning algorithms. This area is deemed most relevant to the subject matter of the rest of this thesis. In particular, the work described in Papcun, Hochberg, Thomas, Laroche, Zachs & Levy (1992) is covered in the most detail.

2.3.1 Analytical methods

A considerable proportion of the previous work to investigate an acoustic-to-articulatory mapping has been grounded within an analytical approach, where mathematical analysis of an acoustic signal is performed to yield the area function of a tube model that might have generated it.

One technique that has been employed to obtain a readily analysable signal relies on taking measurements of the impulse response at the speaker's lips (Schroeder 1967). An experimental setup is used whereby the speaker's mouth is specially sealed on to the end of a tube. The speaker must then articulate without phonation and with the vocal folds closed. Acoustic pulses within the tube are generated with a bandwidth of about 4kHz and a repetition period of 10ms. The signals of two closely spaced condenser microphones within the impedance tube are sampled and processed by computer.

This technique suffers from several drawbacks. For example, the subject receives little or no acoustic feedback of their speech production. In addition, the procedure and necessary equipment prohibit widespread use with varied speech data.

The acoustic waveform itself, as produced by a speaker normally, has often been used. This has the advantage of being easy to record. An example of this approach is Wakita (1973) (and Wakita (1979)), who attempted to infer the area functions for a model vocal tract by analysis of the acoustic speech waveform for vowels.

Unfortunately, satisfactory analysis is not straightforward. Analytical approaches are afflicted by certain fundamental difficulties. First, it is neither straightforward nor obvious how to assess the accuracy of the inferred vocal tract area function. Second, there is the related difficulty in deciding whether a given estimate of the vocal tract area function is physiologically possible, let alone likely. Third, such techniques are only really suited to a subset of phone types; vowels and voiced consonants. Major difficulties arise for analysis during nasalised sounds, where the velum lowers and the oral and nasal cavities become coupled for example. Voiceless sounds are likewise ill-suited to analysis.

2.3.2 Synthesis models

Researchers have often turned to articulatory synthesis models for help in studying and developing an inversion mapping. A variety of articulatory synthesis models have been exploited in a number of ways:

- A database of acoustic-articulatory parameter pairs (spanning the articulatory domain) can be constructed. The articulatory parameters corresponding to a given acoustic vector may then be estimated by finding the closest matching acoustic vector in the database.

- A database of acoustic-articulatory parameter pairs can be used as training data for various other empirical learning models, such as neural networks.

- The control parameters of the articulatory model can be repeatedly adjusted until the synthesised speech differs minimally with the original acoustic speech signal.

Additionally, the first two techniques can be used to cut the search time for the last. In other words, the code book or empirically trained model is used to provide the initial estimate of the articulatory parameters, which are then further refined in an iterative process. This section will look at a few examples of each of these approaches.

Sampling from the articulatory domain

Arguably the simplest way to use an articulatory synthesis model for research into the inversion mapping is to take a large number of sample points throughout the space defined by the articulatory parameters and synthesise their acoustic counterparts. These articulatory samples can be selected either at regular intervals, randomly, or using some more elaborate sampling scheme. This section covers examples of work which feature all three approaches to sampling, along with how the resulting vector pairs are used for acoustic-to-articulatory inversion.

One ubiquitously cited investigation into the inversion mapping using data produced by a synthesis model is that of Atal et al. (1978). The model they used featured 4 articulatory parameters; length of the vocal tract (varies with lip protrusion), distance from the glottis of the maximum constriction, the cross-sectional area at the point of maximum constriction, and the area at the mouth opening[2]. They sampled at **regular** intervals from the articulatory space defined by these four parameters and generated 30,720 different vocal-tract configurations. The frequencies, bandwidths and amplitudes of the first five formants were then computed for each configuration. These articulatory-acoustic vector pairs were sorted and stored in terms of the acoustic parameters in order

[2]In fact, they explored two models. Their other vocal tract model comprised 20 cross-sectional area functions placed at equally spaced points from the glottis to the lips

to facilitate the process of finding the articulatory vector(s) corresponding to an acoustic vector. Atal et al. termed this method for mapping from acoustics to articulation "inversion by computer-sorting"; a numerical approach which relies on the power of the computer for efficiently handling large volumes of data.

Like Atal et al., Rahim, Kleijn, Schroeter & Goodyear (1991) (and Rahim, Goodyear, Kleijn, Schroeter & Sondhi (1993)) used an articulatory synthesis model to generate a database of articulatory-acoustic vector pairs. However, their database consisted of 75,238 articulatory-acoustic vector pairs, which was generated by sampling **randomly** from the manifold of "reasonable" shapes within the articulatory parameter space of Mermelstein's articulatory model (Mermelstein 1973). These shapes were demonstrated to span the articulatory space of voiced and nasal sounds. The articulatory vector comprised 10 vocal tract areas and a nasalisation parameter. Meanwhile, the acoustic vector contained 18 FFT-derived cepstral coefficients, computed from the vocal tract impulse response after de-emphasis.

Rahim et al. used this data set to train MLPs to map from acoustics to the vocal tract area functions, with application to articulatory analysis/synthesis. The articulatory-acoustic vector pairs were initially clustered into 32 regions in the acoustic domain using a weighted cepstral distance measure. To ensure gradual changes in the articulatory control parameters when moving from cluster to cluster, these regions were enlarged to make them overlap. Next, to address the problem of non-uniqueness in the acoustic-to-articulatory mapping, each of the 32 regions was further divided into 4 separate regions in the articulatory parameter domain, this time using a clustering algorithm with a log-area distance measure.

They trained 128 feed-forward MLPs, each with a single hidden layer of 26 units, on the separate regions identified during the clustering stage, after suitably normalising the data.

In order to recover the articulatory parameter trajectories, Rahim et al. would perform a dynamic programming search through the output of the networks with a cost function both in the articulatory parameter and acoustic domains (Rahim et al. used a Kelly and Lochbaum type synthesiser driven by the system output). The optimum trajectory in vocal tract parameter space was judged by a criterion requiring vocal tract shapes to vary as smoothly as possible, while still maintaining a good spectral match. At each pitch frame, the six "paths" that minimised the sum of acoustic and articulatory parameter cost components accumulated over multiple frames of speech were selected. To ensure smoothly varying trajectories of articulatory parameters, the decision in choosing the optimum sequence of networks was delayed by a dynamic programming length of 15 pitch periods.

Rahim et al. pointed out two fundamental disadvantages with the approach they took to creating a data set of acoustic-articulatory parameters by randomly sampling the parameter space of an articulatory model: a) the validity of the data points is limited by the ability of the model to approximate the mechanism of speech production and b) a sizeable proportion of the vector pairs thus generated, although they may be "physiologically plausible", may be not actually be common in typical speech.

The ramifications of this last point might be two-fold. First, it means that disproportionately large weight is given to relatively inconsequential regions of the acoustic-to-articulatory mapping function. Second, in the drive to keep the total number of vector pairs in the training set down to a manageable size, there is the danger that important, complex regions of the inverse mapping function may be too sparsely sampled to accurately capture the local characteristics of the inverse mapping function. The question of how many samples to take and from which areas of the articulatory domain is a significant problem for the sampling approach.

Interestingly, Rahim et al. went on to suggest a method to work around this difficulty, namely bootstrapping to training on real speech data. The method involves taking a system trained as described above, then presenting vectors of acoustic coefficients computed from real speech. For each acoustic vector presented to the network assembly, the output articulatory parameters are run through the synthesiser. The acoustic output of the synthesiser is then compared to the original acoustic input vector, by means of a specially defined error function. This error signal can be propagated back through the network as part of the usual backpropagation algorithm. And so in this way, the network weights can be modified to minimise the error in the acoustic domain.

Rahim et al. quantified the improvement of the bootstrap system (trained on real speech) over the base system in terms of decibels RMS error in the acoustic domain.

Supplementary training on real speech reduced error by 0.3 dB, which they claimed corresponded to a significant improvement in the perceptual quality of all sentences synthesised. They also provided anecdotal evidence of the nature of the improvement. For example, although the synthesiser was unable to generate a [w] sound adequately using the vocal tract areas generated by Mermelstein's model, the bootstrap training allowed adaptation to this, and subsequently resulted in a more acceptable [w]. However, further training of the network did not improve the quality of synthesis during [l] and [r] segments. This observation exemplifies the way in which limitations in the synthesis model can impinge upon modelling the mapping between acoustics and articulation.

In an effort to address the problem of inadequate sampling from the articulatory space, some researchers have sought to develop **more sophisticated** sampling strategies. One such effort is described by Ouni & Laprie (1999), who aimed to sample from the articulatory domain in such a way that the inversion mapping is locally linearised. To achieve this, they began by supposing the articulatory space is contained within a single hypercube, called the "root" hypercube. Within this hypercube, they tested the linearity of the inversion mapping between any of the vertices by linearly interpolating the acoustic value corresponding to the midpoint between them. They compared this acoustic value with that obtained by synthesising with the corresponding articulatory point. If the difference between these two acoustic values was more than a predefined threshold, then the inversion mapping was deemed to be nonlinear in this region, and the hypercube was divided into sub-hypercubes. This process of subdivision based on linearity tests was repeated recursively until the hypercubes reached a size that was decided as suitable to represent a locally linear acoustic-to-articulatory mapping. The goal of sampling this way is to balance detailed coverage in regions where the inversion mapping is complex with more sparse coverage elsewhere.

Mimic data

Several systems have been described in the literature that may be described as *mimics*, where the aim is to iteratively adjust the control parameters of an articulatory synthesiser until the output is as close as possible to a sample acoustic signal. One example of a mimic system is what Shirai & Kobayashi (1986) call *Model Matching*. This is basically an iterative analysis-by-synthesis algorithm. Real human speech is taken and an iterative procedure is used to try to make the output of the speech synthesiser have minimal distance from the real speech in the spectral domain.

Although the mimic approach in itself represents a way to perform acoustic-to-articulatory inversion, the method is generally inelegant and computationally too expensive for performing wide scale articulatory recovery. Instead, researchers have sought to use this technique to generate enough data to train an empirical learning model with more modest computational requirements. An example of this approach is provided by Kobayashi, Yagyu & Shirai (1991). They took a set of acoustic-articulatory vector pairs generated by the authors' Model Matching method (Shirai & Kobayashi 1986) to use as training and testing data. This data set was constructed from the vowel data in 5200 tokens in the ATR word database. They then used this data to train a single feed-forward MLP with a topology of 12 inputs (cepstral coefficients), two hidden layers of 24 units each, and four output units for four parameters of their articulatory model (two tongue parameters, a jaw parameter and a lip parameter).

The neural network estimate of the articulatory parameters differed by an average of just 3% from the values provided by the Model Matching algorithm. In addition, the neural network estimated articulatory parameters from speech 10 times faster than the Model Matching method. Kobayashi et al. concluded that their neural network provides a fast and stable means for estimating articulatory parameters from the acoustic speech signal.

Model complexity

Among the multitude of different articulatory synthesis models that have been proposed, there is a wide range in complexity and intended physiological accuracy. Some researchers have expressed the hope that using more accurate models of human speech production will pay dividend by improving the reliability and accuracy of an inversion method. Aside from increasing the model complexity, the applications of the models seem to be more or less the same.

Dang & Honda (2000) described an approach which features an elaborate 3-D physiological articulatory model. The purpose of their model, developed on the basis of volumetric MRI data gathered for a single male speaker, is to perform speech synthesis in a way which mimics the human speech production exceedingly closely. To synthesise a sound (5 Japanese vowels in this case), the "muscle activations" necessary to move three control points (on the tongue tip, tongue dorsum and lower incisor) to a template target configuration from the current locations are calculated. As the physiological model is driven by these muscle activation signals towards the target, area functions are periodically calculated from the changing vocal tract dimensions, from which speech sounds are generated.

To apply this articulatory synthesis model to acoustic-to-articulatory inversion, the method is more or less the same as other iterative analysis-by-synthesis models. An input vowel is first classified by comparing its formants to 5 previously defined vowel templates. Working from the template articulation, the vowel is synthesised. By iteratively comparing the "hypothesised" vowel to the real vowel in acoustic space, modifying the hypothesis to reduce the difference, and then resynthesising the vowel, the difference is minimised and a suitable vocal tract shape is derived.

Using highly complex synthesis models obviously carries the disadvantage of increasing the burden of computation. It has also been suggested an additional difficulty could arise when using complex articulatory models to generate parallel acoustic-articulatory data to train data-driven learning models. If a vocal tract model stipulates a larger number of independent parameters to define the vocal tract shape than are extracted during feature analysis of the acoustic signal, then a mismatch could result (Hogden, Lofqvist, Gracco, Zlokarnik, Rubin & Saltzman 1996).

2.3.3 Human articulatory data

A certain amount of human articulatory data has been available to speech researchers for several decades in the form of X-ray cinematography. As well as a myriad of other research areas, this resource has unsurprisingly also been directed at investigation of the inversion mapping.

Ladefoged, Harshman, Goldstein & Rice (1978), used linear regression techniques to estimate the shape of the tongue in the midsagittal plane from the first three formant frequencies during constant vowels. To do this they used tracings taken from cinefluorograms of the heads of five speakers uttering ten vowels within the carrier phrase "say h(vowel)d again". They also compared their recovered tongue shapes with other published sets of X-ray images, and remarked that their method appeared to be generalisable to other speakers.

However, due to the dangers of exposure to X-rays, the use of X-ray cinematography for recording human articulatory movements is limited to allowing only small quantities to be gathered. This has meant that such data has been more suited to theoretical speech production research rather than speech technology research. When using empirical learning models, it is a general rule of thumb that the more data there is available the better.

The development of safer methods for recording human articulatory movements, such as X-ray microbeam cinematography and electromagnetic articulography (see Chapter 3), has given us the ability to record the actual movements of the articulators during speech in a reasonably uninvasive way. Consequently, it has become possible to record much greater quantities of speech, paving the way for large, multispeaker parallel acoustic-articulatory corpora, such as the MOCHA project described in Section 3.3. Such a database holds significant advantages over the parallel acoustic-articulatory parameters furnished by articulatory synthesis models. Not least of these is the fact that the resulting data is not influenced by the limitations and imperfections of the synthesis model generating the data, nor by the sampling distribution choice.

The availability of large enough quantities of human articulatory data means it is possible to train data-driven learning models on real human acoustic-articulatory data. Most of the models that have previously been trained on parallel data from an articulatory synthesiser could equally be trained on real human data. However, although a prototype X-ray microbeam articulography system was first demonstrated as early as the 1960s (Fugimura 1982), curiously little work has actually been reported so far on the use of real articulatory data in deriving an acoustic-to-articulatory mapping. There are a few notable exceptions, some of which we shall cover in this section.

Neural network based inversion

(Papcun et al. 1992) used X-ray microbeam data to train MLPs to estimate the trajectories of three articulator channels for six English stop consonants. They used the y-coordinates for the lower lip, tongue tip, and tongue dorsum. Specifically, gold pellets were attached to the mid-line of the vermilion border of the lower lip, and to the mid-line of the tongue at distances of approximately 10mm and 60mm from the tongue tip. The subjects were three male native English (American) speakers, aged 20, 21 and 25, who were each recorded uttering six nonsense words, or "records". The records were comprised of repeated [-Cə-] syllables, where 'C' was one of the six English oral stop consonants /p,b,t,d,k,g/. Therefore, the total amount of parallel articulatory-acoustic data used in the experiments was 18 records (3 speakers × 6 records each), and the number of examples of each phone was 15 (3 speakers × 1 record × 5 syllable repetitions).

In order to process the raw data into a usable parallel articulatory-acoustic training and testing data set, Papcun et al. described several steps:

- Acoustic waveform was low-pass filtered at 4kHz, then resampled at Codec rate 8012.821Hz. Papcun et al. explain this was to render their research results compatible with the demands of telephone applications.
- The waveform was windowed using a Welch window covering 128 sample points (15.98ms) at 64 point intervals, resulting in a sequence of 50% overlapping frames
- The frames were transformed to the frequency domain by FFT, and the power spectrum converted to a decibel scale and then distributed into 18 standard bark-scale bins. The first two bins (up to 200Hz) were discarded.
- The acoustic coefficients were normalised by setting the highest and lowest 0.1% of the data to 1.0 and 0.0 respectively, and then scaling the intermediate values accordingly.

The articulatory data in each record were synchronised to the frame shift of the acoustic parameters by interpolation on a cubic spline. These traces were then smoothed using a 20 point window. Finally, the articulatory traces were normalised in the same way as for the acoustic coefficients, although using the limits of 0.1 and 0.9 instead[3].

Figure 2.3 shows the network topology employed by Papcun et al. (1992). They claimed to have settled on this topology following a lengthy process of trial-and-error where the aim was to optimise two criteria: minimum network training time and maximum performance. They do not specify what range of architectures and data processing were tested in arriving at this optimum. The network they presented had 400 inputs (25 acoustic context frames with 16 bark bin filterbank coefficients per time frame) and two hidden layers of 8 units each. The acoustic context inputs were constructed from the series of acoustic frames such that the corresponding articulatory frame was 40% of the way into the context frame. Care was taken not to include in the training set any input-output vector pairs from the data that contained less than three frames of non-silence in the acoustic input context window.

The network was trained using standard backpropagation gradient descent with a weight update equation modified by the addition of a momentum term until the root mean square (RMS) error between network output and the target in the training set reduced to 0.08 or less.

Separate networks were trained for each of the three articulator positions studied; the y-coordinate relative to the occlusal plane for the lower lip, tongue tip and tongue dorsum. Taken together as a set, these three networks formed what they termed a 'composite' network.

The two measures used by Papcun et al. (1992) to assess network performance were RMS error and Pearson Product Moment Correlation (PPMC). The former is a measure of the overall distance between an estimated articulatory trajectory and the actual, measured trajectory. Meanwhile, the latter is a measure of the similarity of shape of two trajectories; whether they rise and fall at the same points in time.

Papcun et al. found the trajectories of articulators "critical" for the production of a given consonant demonstrated higher correlation coefficients than for articulators "non-critical" to the production of the consonant. For example, the lips and velum would be

[3] 0.0 and 1.0 are asymptotic (hence unrealisable) limits of the sigmoid activation function used in the network output units.

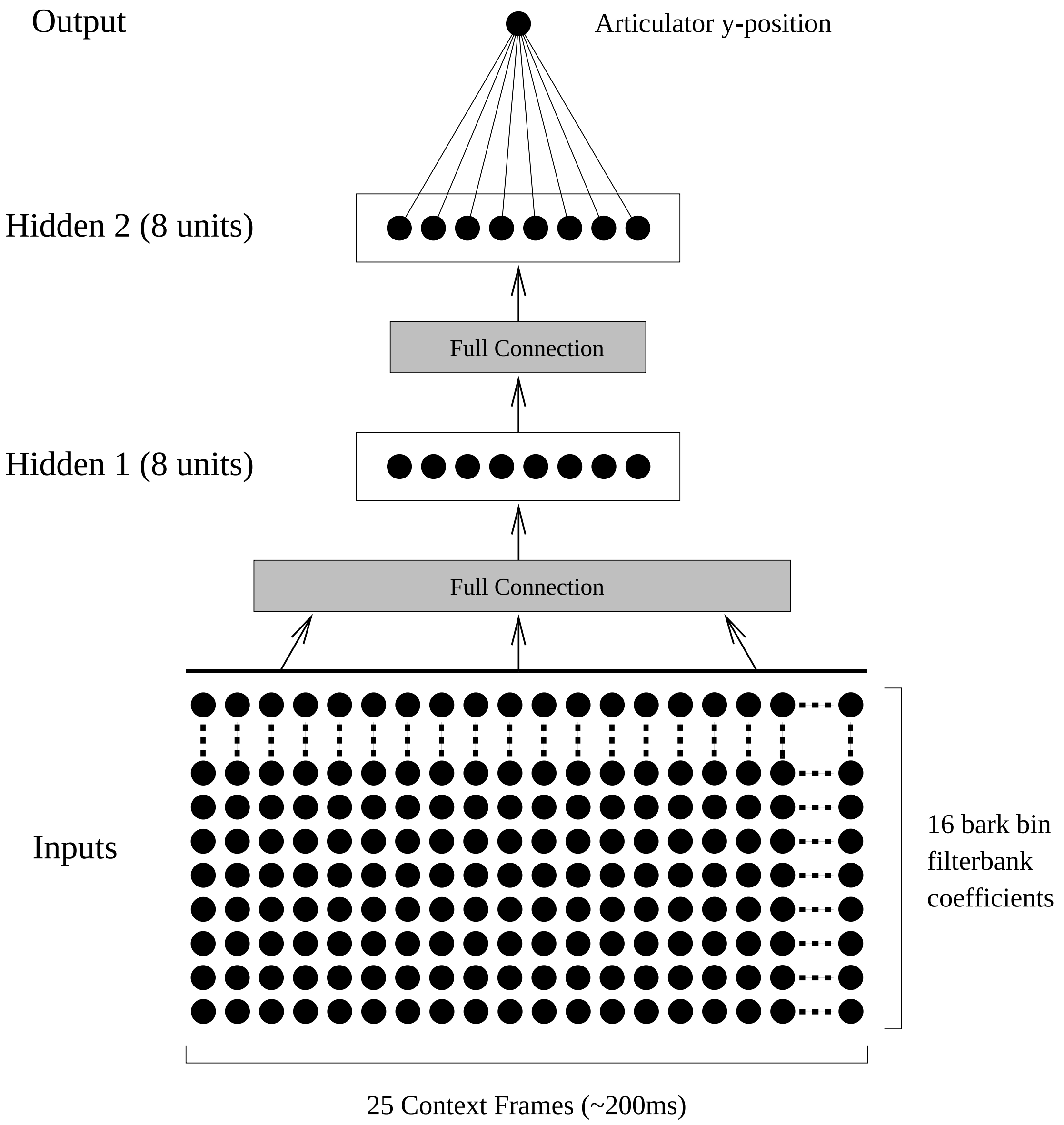


Figure 2.3: The design of feed-forward net used by Papcun et al. (1992). Three such networks were trained separately, one for the y-coordinate of each of 3 articulators studied: lower lip, tongue tip and tongue dorsum. The three networks together were termed the "composite" network. The networks were trained by backpropagation gradient descent, with a momentum term.

critical to the production of [p], whereas the tongue would not be. Interestingly, the average RMS error for critical articulators was *higher*, though, than for articulators which were non-critical to production. Papcun et al. devoted a sizeable portion of their paper to discussing this condition; what they termed the "critical articulator phenomenon".

To explain the disparity in correlation coefficients, Papcun et al. (1992) investigated what they noticed as two common features of critical articulator trajectories: their greater range and regularity. They compared the "peak-to-peak" trajectories of critical and non-critical articulators for each sound type. First, they identified the five peaks of the articulator critical to the production of the phone in a given record (i.e. lower lip for /p,b/, tongue tip for /t,d/ or tongue dorsum for /k,g/). Then, they spliced out the four intervals between the times given by these peaks from each of the three articulatory channels in the record. This was done for all 18 records. Finally, these intervals were normalised in length to 20 time steps, and normalised in amplitude within each record. The result of this procedure was 24 intervals per articulator for each sound type (since 3 speakers, 2 consonants per sound type, which each provided 4 peak-to-peak intervals). Figure 2.4 shows graphs of these intervals overlaid. Papcun et al. (1992) also calculated the mean trajectory for each set of 24 intervals and the corresponding variance, the magnitudes of which are also shown on Figure 2.4.

As evident in Figure 2.4, critical articulator movements were found to be more constrained than those of non-critical articulators. Papcun et al. (1992) debated whether the freer movements of the non-critical articulators could be manifestations of inter-speaker variability, i.e. the three speakers share patterns of critical articulator movement, but have developed idiosyncratic patterns of movement for the less consequential non-critical articulators). Alternatively, it could be explained by intra-speaker variability, i.e. a speaker does not use consistent patterns of movement for non-critical articulators, whether due to some systematic processes or sheer inconsequential randomness. Papcun et al. decided in the end that intra-speaker variability as opposed to inter-speaker variability is the most likely culprit.

Papcun et al. postulated that lower correlation coefficients were observed for non-critical articulators partly because of their lower range of movement. Likewise, they attributed the effect of higher RMSE for network estimates of critical articulator tra-

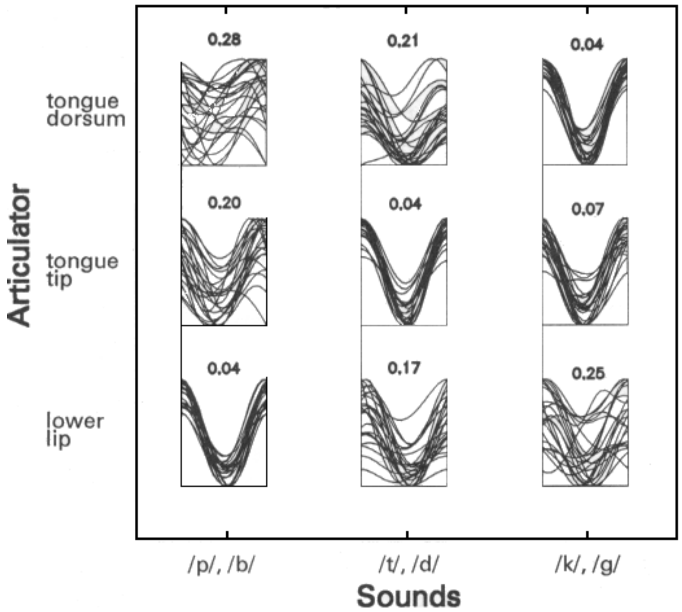


Figure 2.4: Critical versus non-critical articulators. The trajectories of three articulators' y-coordinates during oral stops at three different places of articulation were isolated and superimposed in this plot, after normalising the length and magnitude of each. The trajectories of articulators critical to the production of a given phone demonstrate the lowest variation. (Reproduced with permission from Papcun et al. (1992))

jectories to their greater range of movement. They reasoned that if a target trajectory moves dramatically up and down, and the inferred trajectory is close to this target at each time step, then the shapes of the two curves will be very similar, yielding a high correlation score. However, if the target trajectory only has a small range of movement, the inferred trajectory can be close to this trajectory in terms of RMSE, but without learning a close similarity in shape. Thus, Papcun et al. (1992) concluded their neural network was better able to infer the general shape of bold movements of critical articulators than the lesser movements of non-critical articulators.

The work described by Papcun et al. (1992) was further advanced by Zachs & Thomas (1994). Most of the details of their experiment are very much the same as in Papcun et al. (1992): the data processing, the neural network architecture and so on. Although again using X-ray microbeam data, the data set they used differed from that in Papcun et al. (1992), being comprised of contrastive American English vowels in the "kVd" environment. Specifically, three subjects were instructed to repeat the following words three times: "keyed", "(Kin)caid" (only the end of this word), "cod", "code" and "cooed", giving a total of 45 uttered words in the data set. These words were chosen as they differ principally in tongue position and lip rounding, and are reasonably spread out within the vowel space. Zachs & Thomas (1994) attempted to recover eight articulatory dimensions: x- and y-coordinates for the tongue tip, tongue body, tongue dorsum and lower lip.

The most significant difference between this experiment and the preceding experiment is the introduction of a new error function for training the MLPs. Zachs & Thomas (1994) call this error function "Correlational and scaling error" (COSE[4]). For a given output unit over a set P containing N input-output training pairs, the COSE value is given as

$$COSE = \frac{A}{2}\left(1 - \frac{\sum_{p=1}^{N} o_p d_p}{\sqrt{\sum_{p=1}^{N} o_p^2 \sum_{p=1}^{N} d_p^2}}\right) + \frac{B}{N}\left(\sqrt{\sum_{p=1}^{N} o_p^2} - \sqrt{\sum_{p=1}^{N} d_p^2}\right)^2 \qquad (2.1)$$

where d is the target value, o is the network output value, and A and B are user-defined parameters to adjust the relative weighting of the two terms.

[4] the suggested pronunciation for this is, alas, "cosy"

A closer look at the two terms of Equation 2.1 reveals the aim of this error function. Consider vectors **o** and **d**, the sequence of output values and desired output values respectively, as points in N-dimensional space, where N is equal to the number of points in a given articulatory trace. We see that the second term will have the effect of constraining the point **o** to lie on the same hypersphere as point **d**, since this term will equal zero when vectors **o** and **d** have the same overall length. Meanwhile, the first term will have the effect of constraining vector **o** to point in the same direction as **d**, since minimising this term entails maximising the dot product of **o** and **d** once both have been scaled to unit length (hence minimising the angle between the two vectors).

Zachs & Thomas (1994) evaluated the effect of this error function on network learning and accuracy using a vowel classification task. They first generated templates for the five vowels using the real articulatory data. 8-Dimensional articulatory vectors were taken from the centre of each vowel and discriminant analysis carried out to derive linear discriminant functions that maximally separated the five vowels. Next, a set of 8 neural networks trained on two of the speakers were presented with the acoustic input of the third speaker. The network output was then taken and compared to the vowel templates, by calculating the Euclidean distance from the estimated articulatory configuration to each template. The estimated articulatory configuration was classified as the vowel with the smallest Euclidean distance. For this task, 87% (39/45) of vowels were classified correctly when using the output from a network trained using COSE, as opposed to only 73% (33/45) when using a network trained using the standard squared error function.

Acoustic-articulatory codebook inversion

Hogden et al. (1996) used EMA data to build a codebook of quantised articulatory-acoustic parameter pairs from a corpus containing 90 vowel transitions within the context of two voiced velar oral stops. They used recordings of one Swedish male subject. Receiver coils were affixed to the tongue tip, tongue body, tongue dorsum, tongue rear, lower lip, upper lip and jaw. The articulator positions were sampled at 625 Hz, and then low-pass filtered at 20Hz[5]. Audio was recorded at 20kHz sample rate with 12bit sample

[5]various numbers have been put on the bandwidth of the articulators in the literature. Hogden et al.

size. For each EMA sample point, the vocal tract transfer function (below 5kHz) was calculated from 32 cepstrum coefficients computed from the corresponding 25.6ms wide Hamming window of the acoustic waveform. Rather than use the cepstral coefficients themselves, Hogden et al. (1996) used the smooth spectra, having found better results during pilot studies. Therefore, the acoustic vectors in their experiment were composed of 128 energy measurements, normalised by setting the total energy of each spectrum to unity. Meanwhile, the articulatory vector contained 14 (7 coils $\times$ 2 dimensions) corresponding articulator coordinates.

The utterances contained vowel-to-vowel transitions in the context of /gVVg/. They used the Swedish vowels /i, e, æ, a, o, u/, and front rounded vowels /y, ɯ, ø/ (sic), as well as the English vowel /ɛ/. Thus, with all permutations of transition, 90 "words" were used. Each of these was recorded three times, giving a total of 270 utterances. One of these repetitions was used for the training set, while the other two were used for two separate test sets.

Hogden et al. (1996) employed vector quantisation to create an acoustic codebook, finding that 256 codes gave the best comprise between performance and computational constraints. They used a variant of the "frequency sensitive competitive learning" algorithm. Briefly, a set of 256 reference vectors (R) are initialised with small random numbers to begin with. Each acoustic vector $\mathbf{d}$ in the data set is classified by the reference vector $\mathbf{r}$ that minimises a weighted Euclidean distance metric:

$$argmin_{\{r \in R\}} N_r \sum_i (d_i - r_i)^2 \tag{2.2}$$

where N_r is the number of times a code vector has been used to encode data vectors already. This factor has the effect of encouraging all codes to be used equally often. Next, the reference vectors are set to equal the mean of all the data vectors that have been classified by them. These last two steps are repeated iteratively.

Once a suitable codebook of acoustic codes had been generated, Hogden et al. (1996) created a lookup table of articulatory configurations; for each acoustic code, an articu-

(1996) cite articulatory motions caused by muscle contractions to be typically less than 15Hz ((Muller & McLeod 1982);(Nelson 1977)).

latory vector was stored which was calculated as the average of the articulatory configurations corresponding to all the acoustic vectors classified by that code. The codebook and lookup table taken together formed the method by which Hogden et al. (1996) estimated articulator positions from acoustics. A given acoustic vector would be first coded according to Equation 2.2[6], and then the lookup table would be used to map from this code to the estimated articulatory configuration.

Hogden et al. (1996) reported that RMS error varied from about 0.5mm for the upper lip to 2.3mm for the tongue root y-coordinate. Overall, RMS error averaged about 2mm for the estimated positions of the tongue coils.

Like Papcun et al. (1992), they also found the somewhat paradoxical situation that coil estimates with the lowest RMS errors (jaw x, upper lip x and upper lip y) also demonstrated the poorest correlations, around 50 or 60%. Meanwhile, the correlation of the tongue coils was about 94%. Hogden et al. (1996) attributed this situation to the small range of movement of the upper lip and jaw (in x direction) along with EMA machine measurement error. They postulated that because the range of movement of these articulators is only slightly greater than the size of measurement error, the random measurement error makes up a significant proportion of the variability of the coil position, hence leading to a low correlation.

A further interesting characteristic of the inferred articulatory trajectories noted by Hogden et al. (1996) is an apparent time delay. To investigate this effect, Hogden et al. (1996) compared the performance of systems where the acoustic vector of the acoustic-articulatory vector pairs was delayed more and more in time. In other words, the vector of acoustic coefficients from a Hamming window centred at time $t + \Delta t$ was used to estimate the articulatory configuration at time t. They found that a time delay of 14.4ms produced optimum RMS error and correlation over a whole test set. Hogden et al. (1996) hypothesised this may be because the articulatory configuration at time t has an impulse response extending from that time forward, and therefore a window centred at a point after this time is likely to contain more of the acoustic information about the impulse response. This hypothesis is only partially supported by their observations, though, as the time delays do not appear to be very consistent across

[6] The N_r weight is only used when training the reference vectors, not when coding novel vectors.

all articulators for all utterances.

Similar to Hogden et al. (1996), Okadome, Suzuki & Honda (2000) also described a code-book based approach to acoustic-to-articulatory inversion which makes use of real acoustic-articulatory data, recorded using EMA. However, their data set was considerably larger, consisting of 354 sentences read by three male Japanese speakers. Their code-book entries consisted of pairs of segmented acoustic parameters (30 LPC cepstral coefficients) and the Cartesian coordinates of nine points in the mid-sagittal plane[7]. In fact, Okadome et al. (2000) did not use any vector quantization step, choosing instead to use all 222,894 vector pairs directly. When performing the inversion mapping, they first found the closest candidates (25) in acoustic space, then they used a dynamic programming path search that chose the succession of articulatory configurations which minimised a weighted distance measure of the *spectral distance* and the squared distance between the sequence of articulatory configurations.

Okadome et al. (2000) also presented results where they had augmented the distance measure used to choose a path through the candidate articulatory configurations by making use of the "phonemic information" of an utterance. In addition to the calculation of candidate sequences of articulatory parameters from the code-book, the string of phones known to be present in the utterance was used to form trajectories of articulatory movements "theoretically" present in the utterance using a kinematic triphone model in conjunction with a minimum-acceleration model. Then, at the stage of the dynamic programming search through the lattice of candidate articulatory configurations, the optimum path would minimise an augmented distance measure which contained three weighted terms: the average acoustic spectral distance, the squared distance between successive articulatory points, and the squared distance between the articulatory point from the code-book and the position predicted by the minimum-acceleration model for the sequence of phones.

Okadome et al. (2000) reported an RMS error of around 1.6mm on average for their inversion method using the phonemic information in conjunction with observed timings. They commented that the articulatory features for both vowels and consonants were recovered well. However, performance was somewhat worse when not using observed

[7]lower jaw, upper and lower lips, tongue, velum and Adam's apple

timings, falling to around 1.8mm.

Obviously, the inclusion of knowledge of the sequence of phonemes in the utterance would not benefit a speech recognition system, where the sequence of phones contained in the utterance is not known *a priori*.

Kalman filtering

Another category of approach towards performing the inversion mapping relies on the the use of linear dynamic models, or Kalman Filters (Kalman (1960), Welch & Bishop (1995), Orr (1992)). Typically, the state variables are associated with the articulators, observations are made in acoustic space, and there is some function to map in the forward direction from the articulatory state space to the acoustic observation space. This function is constrained to be linear in the basic Kalman filter, although several extensions have been developed which admit non-linear functions.

Although early applications of this technique to vowels showed significant promise, researchers have found it rather less straightforward to extend the approach to all classes of speech sounds. The difficulty is especially acute for most classes of consonantal sounds.

Dusan (2000) attempted to improve performance of an Extended Kalman Filter based inversion system by imposing what he termed dynamical and "high-level phonological" constraints within the articulatory estimation process. These constraints were introduced by using separate articulatory state-space linear dynamic models, with a dedicated non-linear acoustic observation function, for each *coproduction unit*[8] of speech.

The method described by Dusan (2000) provides an automatic method of segmenting the speech signal and recognising phonological units, based on likelihood computation from Kalman filtering with different models. In short, all the phonological models are run on a section of speech, and the model with the highest likelihood is identified. Then, the final estimate of articulatory trajectories is obtained by way of Kalman smoothing, using the parameters of the sequence of models identified in the first stage.

When developing the system, Dusan (2000) used parallel acoustic-articulatory data produced using an articulatory synthesiser. However, he went on to evaluate the method

[8] Broadly, a coproduction unit consists of the continuous transition between two consecutive phones.

on real speech data recorded both with electromagnetic articulography and an X-ray microbeam system. In these experiments, Dusan (2000) reported average RMS error between actual and estimated articulatory trajectories of about 2mm.

2.3.4 Maximum likelihood approximation

There is a category of approach to acoustic-to-articulatory inversion which does not make use of any articulatory data at all during training. This approach views articulation as some sort of hidden variable in either a continuous or discretized space, which is responsible for producing the acoustic observations. The trajectories for these hidden space variables are optimised using maximum likelihood techniques. Following training, the movements of these variable through the hidden space have been observed to correlate with the trajectories of the real human articulators recorded for the same acoustic test signal. Examples of this approach include (Multiple Observable) Maximum Likelihood Continuity Mapping (Hogden, Nix & Valdez (1998) and Hogden & Valdez (2000)), and the Self Organising Hidden Markov Model of Roweis (1999).

MO-MALCOM

Hogden et al. (1998) explain that their model, Multiple Observable Maximum Likelihood Continuity Mapping (MO-MALCOM), is an attempt to bridge the gap between speech production and speech recognition. The intention is to build stochastic models that make more reasonable assumptions about the mechanisms underlying speech production than do conventional acoustics-based HMMs.

In maximum likelihood continuity mapping (MALCOM)[9], windows of the acoustic waveform are first categorised as one of a finite set of discrete codes using vector quantisation. Each category code has associated with it a probability density function over the location of "articulators" in a latent articulatory space which is called the *continuity map*. A maximum likelihood estimation technique is used to a) find the most likely smooth path through the continuity map given a series of acoustic codes and b) update

[9] MALCOM is the simpler of the two, where the observations are simply sequences of acoustic vector quantization codes. In MO-MALCOM, this observation stream is augmented with the sequence of phone codes.

the parameters of the pdf associated with each of the codes to maximise the likelihood of the sequence. A "smooth path" is defined as one that does not contain Fourier components above a defined cut-off frequency (15Hz in their report). This is in order to obtain plausible articulatory-like trajectories which comply with typical human articulator bandwidths. Apart from this constraint, which may or may not be construed as representing articulatory information, no articulatory data is used during the training of the MALCOM model.

Hogden et al. (1998) claimed that the learned trajectories through the continuity map correspond to articulatory trajectories. They partly justified this claim by reference to statistical theory, which informs us that maximum likelihood estimates of mixture density parameters will approach the actual parameter values of the system generating the data as the amount of training data grows large. They also corroborated their claim with empirical evidence that the estimated mean continuity map positions for the acoustic codes correlated highly with the mean of *measured* articulator positions that produced each code.

SOHMM

Roweis (1999) introduced what he called the Self Organising Hidden Markov Model (SOHMM). By analogy with the relationship of Kohonen's Self Organising Map (SOM) to Vector Quantisation, the SOHMM incorporates the notion of neighbourhood into an HMM. Under the SOHMM model, the states for the underlying Markov chain are packed into a geometric arrangement within what is termed the *topology space*. To achieve this, Roweis explicitly restricted the set of allowable transitions from a given state to include only transitions to states judged to be immediate neighbours under some arbitrary criterion. For example, a SOHMM might consist of a topology space in the form of a three dimensional cube, where only transitions to face-centred neighbours are allowed. Such a topology space is presented in Figure 2.5. The result of imposing a topology space such as this on the Markov chain is that states that are near each other in the topology space are pressured into learning model emissions that tend to occur near each other in time. Furthermore, all possible state sequences in the SOHMM comprise connected trajectories through the topology cube.

Roweis (1999) applied the SOHMM to inferring articulation from acoustics. The acoustic waveform was converted to 12 mel-frequency cepstral coefficients, computed with a window width of 23.5ms at a frame rate of 6.866ms. Vector quantisation was carried out on the resulting cepstral coefficients using 64 codes. This sequence of discrete codes was used to train SOHMMs with topology spaces of various dimensions from 1 to 10. Again, it should be pointed out that no articulatory data has so far been used.

To recover the inferred "articulatory" trajectories from a trained SOHMM and compare them to the real, measured articulatory trajectories, the trajectories in the topology space must be transformed to the articulatory space defined by the 8 gold pellets of X-ray microbeam data. Roweis (1999) calculated the best single linear transform between all the learned state trajectories in the topology space and the actual articulator trajectories. This transform can then be used to convert the state space trajectories to estimated articulatory trajectories.

Unfortunately, while this method proved successful when applied to small amounts of data, it did not apparently scale well to significant amounts of training data. The suspected cause of this is the problem of local minima and overfitting. Consequently, Roweis argued that this purely unsupervised use of the SOHMM will not ultimately be useful for recovering articulation from entire utterances of continuous speech.

Undeterred, Roweis (1999) went on to demonstrate a more sophisticated method of acoustic-to-articulatory inversion based on SOHMMs and mixtures of local linear dynamic models. Real X-ray microbeam articulatory data is used in this method. After performing vector quantization in the articulatory space[10], a set of local linear models is trained to perform the forward, articulatory-to-acoustic mapping. The real articulatory data stream is then transformed into a sequence of codes by determining which of the local models (modes) applies at each time frame. This sequence of codes is used to train a SOHMM, termed the "mode SOHMM". Next, statistics on the acoustic features associated with each of the modes are collected; all the spectra produced by articulatory configurations that were classified as a certain mode are compiled. Another SOHMM, termed the "acoustic SOHMM", is created by copying the topology and transition matrix values of the "mode SOHMM". However, for each state of this SOHMM,

[10] Roweis (1999) used 32 codes in the example he presented

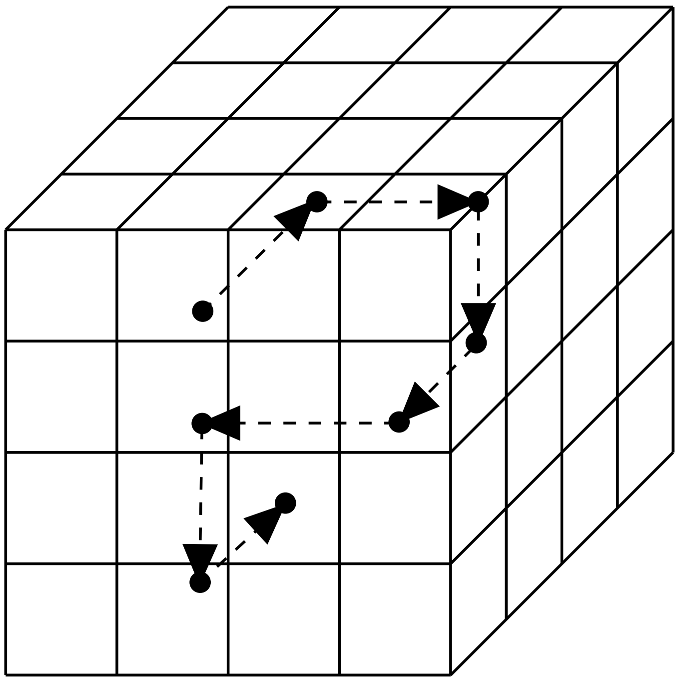


Figure 2.5: An example SOHMM topology space. The states of the Markov chain underlying the SOHMM (represented by the small cells) are packed into a three dimensional topology space (the large cube). Transitions are only allowed between neighbouring states in this space, hence all possible state sequences correspond to some connected trajectory through the topology space. The dashed line shows an example of a state sequence which corresponds to a "city-block" path through the topology space. (Diagram reproduced with permission from Roweis (1999))

the output distribution is set to be a mixture of the output distributions calculated for each "mode" depending on their respective probabilities in the corresponding state in the mode SOHMM.

Recovering articulation from acoustics involves a three stage procedure. First, the incident acoustic feature vector sequence is used for state inference in the acoustic SOHMM. Next, the corresponding state sequence is run through the mode SOHMM. This generates a sequence of probabilities for which local linear model should be active at each time frame. Finally, the most likely sequence of local linear models is used to perform Kalman smoothing (where the hidden, or latent space, variables are the articulators and the observations are the acoustic feature vectors), and yield the articulatory traces given the acoustic observation stream.

Roweis (1999) claimed the results of this hybrid between SOHMMs and linear dynamic models were far superior to both using the SOHMM in a completely unsupervised way and using a single global linear dynamical model. He provided a few estimated articulatory trajectories for comparison with the actual articulator trajectories, which look impressive. However, unfortunately he did not provide any more wide-ranging or quantitative results as to the accuracy of the inferred trajectories, aside from an elementary template-based isolated word-spotting task.

2.4 Discussion

2.4.1 Instantaneous non-uniqueness

Evidence indicating that the instantaneous inversion mapping can be non-unique in certain instances has come from many sources, many of which were covered in Section 2.2. However, it is not clear exactly how much of an obstacle is posed to the development of an inversion mapping system by this. Unfortunately, the data and arguments put forward have not really quantified the extent of non-uniqueness in the inversion mapping. Moreover, critics question the validity of some sources of this evidence. We will briefly consider certain objections here.

Some of the evidence presented in Section 2.2 comes from work done with articulatory synthesis models. Articulatory synthesis models are undoubtedly convenient to use; no expensive, complex and esoteric instruments are required to gather data. However, there are drawbacks. Crucially, the accuracy and scope of the articulatory production model influence the characteristics of the data produced. In particular, it must be admitted that defining exactly how accurate a model of human speech production they represent is somewhat problematic with such models. Hogden et al. (1996) highlight inconsistency between two models to demonstrate this point, comparing a vocal tract represented by a lossless tube that can take on any shape with a vocal tract represented by an acoustic tube with a single energy loss near the glottis. In the first case, very different vocal tract shapes can produce identical transfer functions. Meanwhile, in the second case, the vocal tract shape may in principle be recovered, given adequate information about the acoustic signal.

A lack of physiological accuracy can lead to articulatory synthesis or analytical models which accept articulatory configurations that are not physiologically possible in human speech production. When dealing with such models, it is not straightforward to decide what configurations are plausible, therefore some of the multiple configurations seen in vocal tract models that correspond to a single acoustic signal may not be physiologically possible. More importantly, there is no way of telling how many of the configurations that are physiologically possible are *actually used* in human speech production.

The same objections can be levelled at the evidence for non-uniqueness which comes from bite-block experiments. Although it has been clearly demonstrated that humans *can* produce certain acoustic sounds with different articulatory configurations, it is not clear how much humans use articulatory compensation in normal speech. The extent of this effect has yet to be verified and quantified satisfactorily. It is possible such concerns could evaporate when using large quantities of real, physiologically faithful speech data. Figure 2.6 shows the movements of the tongue tip in the midsagittal plane from the first thirty sentences of the database of speaker `fsew0` overlaid on the same graph. It can be seen that the points are by no means evenly distributed throughout the region of movement, and that many of the movements the tongue tip can make are not present.

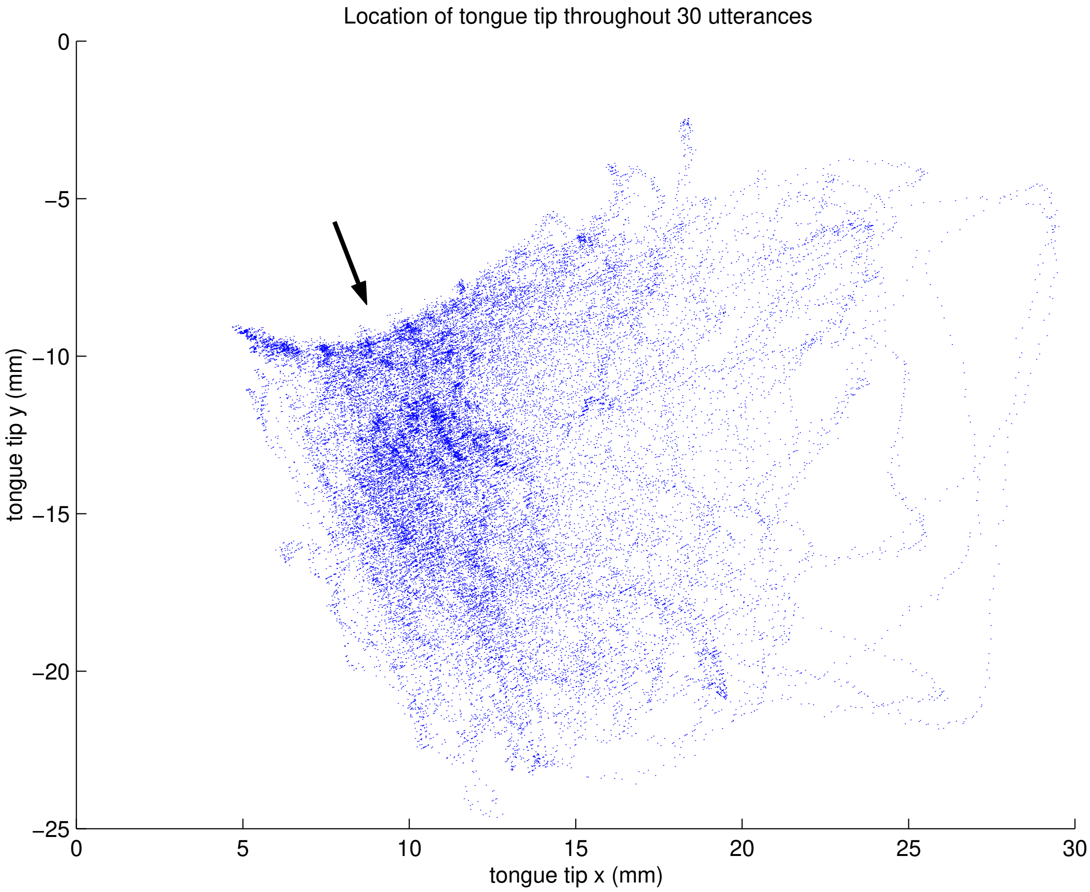


Figure 2.6: A Scatter plot of tongue tip position for the first 30 utterances of speaker `fsew0` database. Approximately 45,000 sample positions are contained within these utterances. Notice that the position samples are not evenly distributed throughout the range of movement of the tongue. The arrow indicates the hard boundary, presumably formed by the alveolar ridge (in MOCHA data, the speaker faces left, as in Figure 3.2). There seems to be a dense region of points extending from the alveolar ridge along the line of the arrow.

Dynamic constraints

Mindful of the one-to-many problem, researchers have sought ways to use the continuous and slow movements of the articulators to finesse instantaneous uncertainty. One can identify a common design philosophy among all the different approaches to performing the inversion mapping:

1. perform an instantaneous mapping from the acoustic domain to the articulatory domain

2. perform some sort of post processing or smoothing of the resulting trajectories on the basis of some articulatory constraints.

The constraints used can vary widely in complexity. Hogden et al. (1996) point out that low pass filtering the articulatory trajectories imposes a simple articulatory constraint when the pass band is set to equal the bandwidth observed for human articulatory movements.

A whole range of more complex articulatory constraints have also been employed. Rahim et al. (1991) used a cost function constraining articulatory trajectories to be as smooth as possible as part of a dynamic programming search through the output of their networks. Schroeter & Sondhi (1989) also used a scheme similar to this. Meanwhile, others have suggested employing the constraint of economy of effort, e.g. Kuc, Tutuer & Vaisnys (1985). Basically, the ideal trajectory under this constraint is one where the articulators move as little as possible. More elaborate techniques have been suggested, such as Chennoukh, Sinder, Richard & Flanagan (1997*c*), whose system features a recurrent algorithm that takes into account the dynamic properties of the articulators. At time t, the position of the articulators at time $t+1$ is forward estimated using the current position together with the velocity and acceleration of the articulatory parameters. Then, at time $t+1$, the candidate positions for the articulators given the acoustics (they use a codebook approach in this case) are compared to the estimate calculated at the previous time step.

Within the common strategy of using articulatory constraints, there is a dichotomy in philosophy. On one hand, some researchers ignore instantaneous non-uniqueness, evidently hoping that errors introduced by doing so will be minimised by the continuity constraints. On the other hand, sometimes researchers take instantaneous non-uniqueness into account, and build into their model some means for dealing with multiple candidate output articulatory configurations for each acoustic input time frame. Here, the intention is that articulatory constraints can chart some optimum course through the series of possibilities; correct information pertaining to articulator positions recovered when the mapping is not ill-posed when taken with constraints on articulatory movements will disambiguate problematic sections of speech where the mapping is non-unique.

The inversion system described by Hogden et al. (1996) is an example of the category that does not set out to explicitly model instantaneous non-uniqueness. Their approach is simply to average all the articulatory positions for a given acoustic code and subsequently use that average as the estimate of articulator positions when mapping from similar acoustic vectors. However, it is not difficult to envision an adaption to their system that *would* incorporate a more flexible description of the articulatory configurations associated with a given acoustic code. For example, instead of calculating and storing the mean of these articulatory configurations, they might have fitted a mixture of Gaussians to them and stored those parameters instead.

An example of a system that does set out to handle non-uniqueness is that of Rahim et al. (1991), whose system was covered in Section 2.3.2. They claimed that having 128 networks assigned to separate regions of the articulatory space accommodated non-unique acoustic-to-articulatory mappings. This method, though, is arguably more suited to handling multiple branches of solutions rather than situations where the solutions form a single region which is spread widely. For example, this multiple regions method would be able to handle the situation where, given a region in acoustic space, there exists a distinct bimodal distribution of associated points in articulatory space; the two regions would most likely be found by the clustering algorithm, the mapping for which would be learned by two separate networks. However, if the points from a small region in acoustic space corresponded to points in articulatory space that form a single cohesive cluster, but one that spans a significant distance along one or more of the articulatory dimensions, the inversion process described by Rahim et al. (1991) could have difficulty accommodating this.

Explicit modelling of non-uniqueness

For all the approaches to acoustic-to-articulatory inversion reviewed, the end goal seems to be to recover articulation as accurately as possible at all times for all articulators. Recognising that the instantaneous mapping is ill-posed, researchers have turned to methods, such as articulatory constraints, to reduce the impact of non-uniqueness as far as possible. It is not clear whether or not following this path, by incorporating more and more constraints, will lead to a perfect method for performing the inversion mapping. In other words, it may be the case that no number of constraints that may be reasonably formulated and applied will be able to disambiguate completely the movements of articulators recovered from the acoustic signal.

Consider the hypothetical example of the production of a /p/ segment. For the production of this segment, the motion of the lips and the velum are critical, and have a large influence on the sound produced. However, the movement of the tongue is not critical to producing the /p/ segment. During the bilabial closure, the tongue could take any number of positions. The exact movements of the tongue are likely to depend heavily on the neighbouring phones, due to coarticulation. It might be possible to make a best guess at how the tongue moves during the time where it is non-critical and has little or no effect on the acoustic signal, but there will arguably always be some degree of uncertainty in this estimate.

If inferred articulation is to provide useful application, it would be useful to know how much confidence to ascribe to the estimate of each articulator position. Consider the case where a range of articulatory configurations (or articulatory "fiber") may produce an acoustic vector, but that within this range, one or more of the articulators may be well defined, while others may vary, as has been demonstrated by Atal et al. (1978). In such an instance, although not *all* the positions of the articulators may be accurately recovered, it seems sensible to ask whether it is possible to recover the positions of the articulators that *are* well defined. It seems that very little of the work that has been carried out so far has been motivated by the hypothesis that some articulators may be inferred with more confidence at certain times than others. Specifically, we have generally not seen attempts at inversion that explicitly model an estimate of the *uncertainty* of inferred articulatory parameters at each point in time.

2.4.2 Evaluation difficulties

Although Papcun et al. (1992), and other researchers, have made use of RMS error and correlation as two measures for comparing articulatory trajectories, their usefulness is limited. First, they represent more a measure of the accuracy of trajectories estimated for individual articulators, rather than an integrated and quantitative measure of the series of estimated configurations of the vocal tract as a whole. Second, RMS error and correlation do not provide a single and absolute assessment of the similarity between an estimated trajectory and the real trajectory. The two measures do provide some means to compare two estimated trajectories relative to each other (against the actual trajectory). However, given only average RMS error and correlation scores for the output of a neural network it would be rather difficult to say whether the network is performing well enough or not. Papcun et al. (1992) point out, for example, that with the number of points contained in an average trajectory even a small correlation is likely to be statistically significant. Using these measures of similarity over utterances of sentence length, containing a rich variety of phones, stretches their credibility even thinner.

2.4.3 Computational cost

It is prudent to consider execution speed and memory usage when developing an inversion mapping, specifically for the purposes of speech recognition. Although computational power is advancing rapidly according to Moore's Law [11], it is obviously advantageous to have a speech recognition system that is as efficient as possible and requires as small a memory footprint as possible. For example, one of the key applications of speech recognition technology in the future could well be to assist in the miniaturisation of "smart" electronic devices, such as the evolution of mobile phones to personal "communicators" without the need for a keypad or screen.

Some of the methods for recovering articulation from acoustics covered in Section 2.3 require significant computational resources. For example, the code book approach has

[11] The observation made by Gordon Moore in 1965 that the number of transistors storable on a given area of silicon (closely related to processor speed) would double every year (revised now to 18 months)

the disadvantage of being relatively expensive computationally. A reasonable performance requires a certain number of codes, which take space to store, and time to access. Although researchers have found ways to try to minimise access times (Chennoukh, Sinder, Richard & Flanagan 1997*b*), codebooks will always require more space than many other methods. Such constraints might prove a decisive factor in many practical applications of an inversion mapping, such as the mobile phone speech recognition application mentioned in Section 1.3.3.

Once trained, a neural network requires only modest computational resources in terms of both memory space and speed of execution relative to other models. Such efficiency is a desirable property, which among other things has spurred interest in neural networks for several researchers working on the acoustic-to-articulatory inversion mapping. For example, as mentioned in Section 2.3, Papcun et al. (1992), Zachs & Thomas (1994), Rahim et al. (1991) and Kobayashi et al. (1991) have described attempts to recover articulation from acoustics using neural networks, and have reported positive results. In particular, we might take the example of Rahim et al. (1991), who cite the efficiency of neural networks as a major motivation for working with them. The total memory requirement of their trained assembly of neural networks was just 4% of that required by the codebook they used for comparison, without any perceived loss of quality. In addition, the network system provided a mapping from acoustics to articulatory parameters 20 times faster than the codebook lookup.

2.5 Conclusion

The instantaneous acoustic-to-articulatory inversion mapping is well recognised as an ill-posed problem. Crucially, it features one-to-many mappings; an acoustic vector at a given moment in time could potentially have been produced by more than one articulatory configuration. This makes modelling the inversion mapping a very difficult task. However, there are ample studies to spur optimism that acoustic-to-articulatory inversion is at least in part possible for *sequences* of speech data, and that there may well be gains to be made should a suitable system be derived for doing so.

Measured human articulatory data is an extremely valuable resource to help develop an inversion mapping method. It is arguably much more useful than data generated by articulatory synthesis models. Synthesised data may contain artifacts resulting from limitations and inaccuracies in the articulatory synthesis model. What is more, by using measured human data we avoid difficulties in deciding how an inversion method is *really* performing; there is no more accurate production model for a speech signal than the vocal tract that actually produced it. We can assess how well an inversion algorithm is performing by comparing the output with how the speaker actually articulated an utterance.

Despite the obvious advantages, surprisingly limited work has been done on the inversion mapping using measured human articulatory data. The few studies that have been done have focused almost entirely on a restricted set of speech sounds only. There is fortunately little overlap between these restricted sets, and therefore there is some scope for optimism that inversion is possible for all speech sounds. However, there is no substitute for actually attempting inversion for all speech sounds at once and for continuous speech. Thus, attempting this and reporting the results would constitute a valuable contribution in itself.

What is more, it would seem prudent to try to model the inverse mapping using a computationally efficient algorithm, such as a neural network, as a first attempt before moving to more complicated methods. The work of Papcun et al. (1992), Zachs & Thomas (1994), Rahim et al. (1991) and Kobayashi et al. (1991) in particular, has provided very useful insight into how we might expect neural networks to perform the acoustic-to-articulatory inversion mapping, and that there is a strong case for attempting a neural network mapping on greater quantities of phonetically diverse speech.

Chapter 3

Articulatory Data

3.1 Introduction

The experiments described later in this thesis rely on a corpus of parallel acoustic-articulatory speech data called the MOCHA database, recorded at Queen Margaret University College. Prior to embarking on describing how this corpus has been used, it will be no doubt prove helpful to introduce the dataset itself. Important considerations include how this dataset compares with other sources of parallel articulatory-acoustic data, how it has been collected, what are the advantages and whether there are limitations of this type of data.

As noted in the discussion in Sections 2.3.2 and 2.3.3, parallel acoustic-articulatory data can come from two main sources: corpora generated by articulatory synthesis models and corpora recorded using human articulography techniques.

A great number of articulatory synthesis models have been proposed and used, with varying degrees of physiological accuracy. The advantage of synthetic models is that within the constraints of a single model, it is relatively cheap and easy to produce copious quantities of acoustic-articulatory data. However, since the data is produced by means of an artificial model, the data will unavoidably embody the characteristics of that model. It is possible for the generating synthesis model to contain inaccuracies, omissions and other flaws. Even where an articulatory synthesis model could be proved to be sufficiently realistic and accurate, it is still possible to misuse the model in a way that does not match the way humans speak. For example, it would be possible to overgenerate from an articulatory synthesis model, spanning the whole of the articulatory domain, whereas it is not thought likely that humans use all the articulatory configurations that are physically at their disposal for producing speech sounds. Unfortunately, the danger that such flaws and complications might be passed on into the database and hamper research efforts based on such a dataset cannot be ignored.

To be sure of the authenticity and reliability of articulatory data, it is arguably much better to use real measurements of human articulation, recorded along with the acoustic waveform. The experiments described in this thesis make use of measured human articulatory data exclusively. Therefore, the focus of this Chapter rests upon human articulography in general, and EMA in particular.

3.2 Human articulography

In the past, X-Ray cinematography has been used to film movies of human articulator movements. Obviously, now we are more wary of the dangers of X-ray exposure, we must rely on techniques which are thought to be more "subject-friendly". Among the more common systems in use for measuring articulatory parameters during speech are the following:

Airflow various systems which measure the airflow/pressure (primarily egressive) from the oral and nasal cavities. Typically, the subject must wear a mask or mouthpieces with a flexible rubber airtight seal.

Electromyography measures the electrical current which is proportional to activity in muscles. Electrodes can either be placed on the overlying skin or needles can be inserted directly into the muscles.

Electropalatography (EPG) The subject wears an artificial hard palate in which an array of electrically conducting contacts is embedded. An additional electrode held in the hand completes a circuit and allows the patterns of tongue-palate contact to be recorded.

EMA Electromagnets are arranged around the speaker's head. Currents are transduced in sensor coils affixed to the articulators, from which their position may be computed. See Section 3.2.1 below for more details.

Electrolaryngography A pair of electrodes are placed on either side of the larynx and held in contact with the skin by a collar. During speech, the electrical impedance across the larynx is measured, from which it is possible to assess the degree of closure of the vocal cords. (Also known as electroglottography)

Nasometer Gives a measure of the presence of nasality during speech production, by measuring the acoustic output from the oral and nasal openings and calculating the ratio. (This is not strictly an instrument for measuring articulation as it measures acoustic energy).

Optopalatography Uses an artificial hard palate covered with fibre optic sensors to dynamically measure tongue-palate contact, distance and pressure. The system uses reflected light distance sensing[1].

Transillumination A light source is located either above or below the glottis (internally or externally respectively), and a light sensor on the other side samples the amount of light radiating through the glottal opening. (also known as Photoglottography).

X-Ray microbeam cinematography Gold pellets are attached to the subject's articulators, using dental adhesive. The subject's head is then positioned between the source of a narrow beam of X-rays (1mm diameter) on one side, and an X-ray detector on the other side. At each sample time, a computer calculates the estimated positions of the pellets, based on the previous known locations, velocities and accelerations. The microbeam performs a small raster-scan of the field where each pellet is expected to be, based on these calculations. The shadow cast on the X-ray detector on the other side of the subject's head indicates the exact location of the pellet. The X-ray microbeam data that has been used in the literature has

[1] The system is still in development by Alan Wrench at Queen Margaret University College, and is not yet publicly available

invariably been provided by the X-ray Microbeam Facility at the Waisman Center of the University of Wisconsin, with the acoustic signal recorded concurrently at 10kHz sample rate

3.2.1 Electromagnetic articulography

There is a very high degree of variation inherent in speech production, observed on both an inter- and intra-speaker basis. This means that preferably very large quantities of articulatory movement data are required for accurate and comprehensive modelling of speech production, especially when using empirical learning techniques, such as ANNs or HMMs. Therefore, to facilitate the collection of a large corpus of synchronous acoustic and articulatory data, the techniques used must ideally be as safe, inexpensive and efficient as possible. The X-ray microbeam technique (Fujimura (1967), Fujimura, Kiritani & Ishida (1973), Nadler, Abbs & Thompson (1985)), fulfilled many of the necessary criteria, but did not provide a totally satisfactory solution, with relatively high cost and access difficulties. Electromagnetic Articulography (EMA) is a minimally invasive technique for transducing the movements of specific points on the active speech articulators, such as the tongue, lips, jaw and velum, which more or less fulfils all criteria of a low cost method for articulography.

The basic principle

Figure 3.1 illustrates the basic principal underlying the EMA recording technique. When a receiver coil is placed within the field of a transmitter electromagnet, alternating at a certain frequency, a current is induced in the coil with the same frequency. The voltage of the induced current is inversely proportional to approximately the cube of the distance between the transmitter and receiver coils. Hence, measuring the voltage of the induced current means this distance can be calculated. By using multiple transmitters, each with a known frequency of alternating current, the position of the coil relative to the transmitters can be calculated.

To record trajectories, the position of the sensor coil is sampled at regular intervals. The rate at which sampling should occur depends on how fast the articulators move.

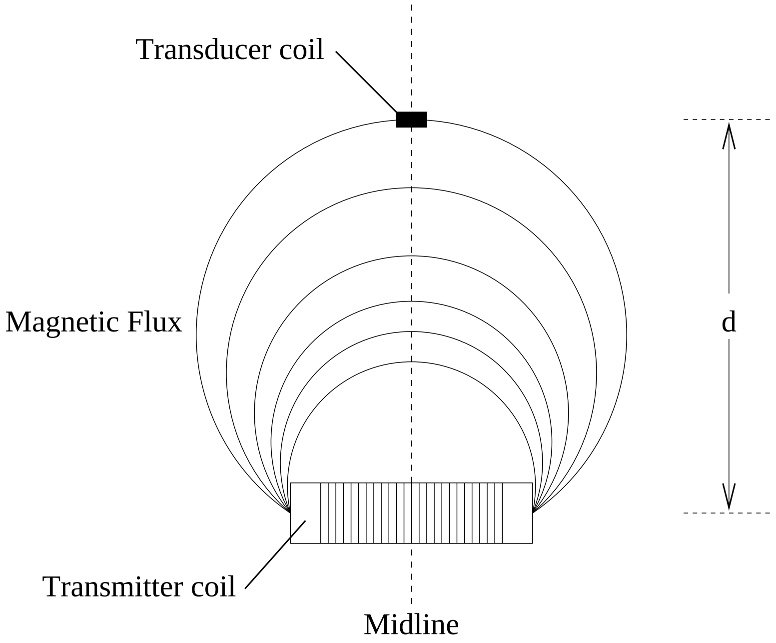


Figure 3.1: A Schematic diagram illustrating the basic principle of EMA. An alternating current is passed through transmitter coil, which consequently sets up an alternating magnetic field (indicated by the flux lines) at the same frequency. When the transducer coil is placed in this field, on the mid-line and with its axis parallel to the axis of the transmitter coil, an alternating current with the same frequency as the transmitter coil is induced in the coil. The voltage of the induced current is inversely proportional to (approximately) the cube of the distance, d, between the transmitter coil and the transducer coil.

Speech gestures that are generated by muscle contraction alone are typically quite slow. The fastest movements of this type are probably those made by the tongue tip, which has been reported to move with velocities in the range of 80cm/s (Perkell, Cohen, Svirsky, Matthies, Garabieta & Jackson 1992). However, articulatory movements which are aerodynamically influenced, for example during alveolar trills and the release of bilabial plosives, can be much more rapid. It is generally accepted that in order to accurately characterise all supraglottal articulatory movements, the bandwidth of the measurement system should go up to around 500Hz.

3.2.2 A practical system

There is an important caveat to the application of the basic principle underlying EMA. Not only does the voltage of the induced current vary as a function of the distance from the transmitter coil, but it is also affected by its orientation within the magnetic flux. In the ideal situation depicted in Figure 3.1, the receiver coil is located on the mid-line of the transmitter coil, oriented such that its axis is parallel to that of the transmitter coil. If the receiver coil were to become rotated or move away from the mid-line, then the induced voltage would decrease. Unfortunately, this change in voltage introduces error into the calculation of the distance from transmitter to receiver.

This complication explains why electromagnetic articulography has long been limited to measurements in one 2-D plane; all the sensor coils must be placed on the plane formed by the mid-line of the transmitter coils. The midsagittal plane (vertical section through the head along the line of symmetry of the face) is the obvious choice for this plane. However, placing the sensor coils on articulators in the midsagittal plane does not avoid introducing error altogether. The problem remains that during speech, the articulators can move in such ways that the coils are moved off the mid-line or rotated within the magnetic field. Sensors attached to the tongue could be particularly susceptible to this.

Perkell et al. (1992) described and evaluated two variations of 2-D electromagnetic midsagittal articulography systems which differ in their approach to dealing with error introduced by sensor coil misalignment with respect to the plane of the transmitter axes. One system uses three transmitter coils and single-axis transducer coil sensors, while the other uses only two (modified) transmitter coils in conjunction with biaxial transducer coil sensors. Perkell et al. (1992) concluded that the three-transmitter design is preferable because it requires a lower magnetic field strength, uses simpler and cheaper sensors and is simpler to calibrate.

The only 2-D electromagnetic articulograph system that is readily available today is manufactured by Carstens Medizinelektronik GmbH[2]. They developed their commercial system, the Carstens AG100 Articulograph in 1988, although initial development began at the Medical School of the University of Göttingen as early as 1982. This Carstens AG100 articulograph is of the type that has three transmitter coils. According to their web site, there are more than forty of their Articulography systems in use worldwide. It was a Carstens AG100 that was used to record the EMA data in the MOCHA database, which is described in Section 3.3.

3.2.3 New developments in EMA

Carstens Medizinelektronik have recently announced the commercial availability of their new system, the AG500. Carstens report development of the AG500 began in 1995 in cooperation with the Phonetics department of the University of Munich under the direction of Prof. Hans G. Tillmann and with the support of NTT Basic Research Labs., Japan. Whereas the AG100 was a strictly 2-D articulography system, the AG500 allows sensor placement anywhere within a 300 mm spherical measurement area and with any orientation of the dipole sensor coil. By using six transmitter coils fixed to a cabin around the speaker's head, it is possible to use the same basic principle of electromagnetic articulography to calculate the XYZ coordinates as well as two angles of rotation which provide a full description of the position and orientation of the sensor coils in the measurement area. The sensor coils are exactly the same as used in the AG100 system. Within the cabin area, the subjects head may move freely. In order to compensate for the movements of the subject's head and transform the measured data into a skull-fixed coordinate system, reference coils are placed on the subjects head. An algorithm then subtracts head-movements from the movements of the sensors

[2]http://www.articulograph.de

under question. This freedom of head movement is an additional advantage over the 2D system, which require a fixed head position, potentially inhibiting full naturalness of speech.

Ziert, Hoole, Honda, Kaburagi & Tillman (2000) evaluated a prototype of this system, and claimed to have measured a spatial resolution of less than 1 mm and a rotation angle detection with an accuracy of approximately 1 degree. Finally, Ziert et al. (2000) also confirmed that the magnetic field strength of their setup is about 2 μT at the centre of the measurement area, which is in accordance with health and safety guidelines.

3.3 The MOCHA database

The **M**ulti**ch**annel **A**rticulatory (MOCHA) database has recently been recorded at Queen Margaret University College, Edinburgh[3]. The MOCHA database is intended to feature up to forty speakers with a variety of regional accents. At the time of carrying out the experiments for this thesis, however, only two speakers had been made available, one male (with a Northern English accent) and one female speaker (with a Southern English accent).

Recording sessions took place in the purpose built sound damped studio at the Edinburgh Speech Production Facility based in the Department of Speech and Language Sciences. This studio has the capability to record four concurrent data streams while the subject speaks:

Acoustic speech waveform recorded at 16kHz sample rate, with 16 bit precision. (audio-technica ATM10a microphone)

Electromagnetic articulograph The positions of ten coils in the midsagittal plane are recorded at 500 Hz with 16 bit precision.

Laryngograph waveform recorded at 16kHz sample rate (Fourcin Laryngograph)

Electropalatography 1 binary value for each of 62 contacts sampled at 200 per second (Reading Electropalatograph)

[3] as part of the Engineering and Physical Sciences Research Council grant number GR/L78680: "Speech Recognition Using Articulatory Data" (Wrench & Hardcastle 2000)

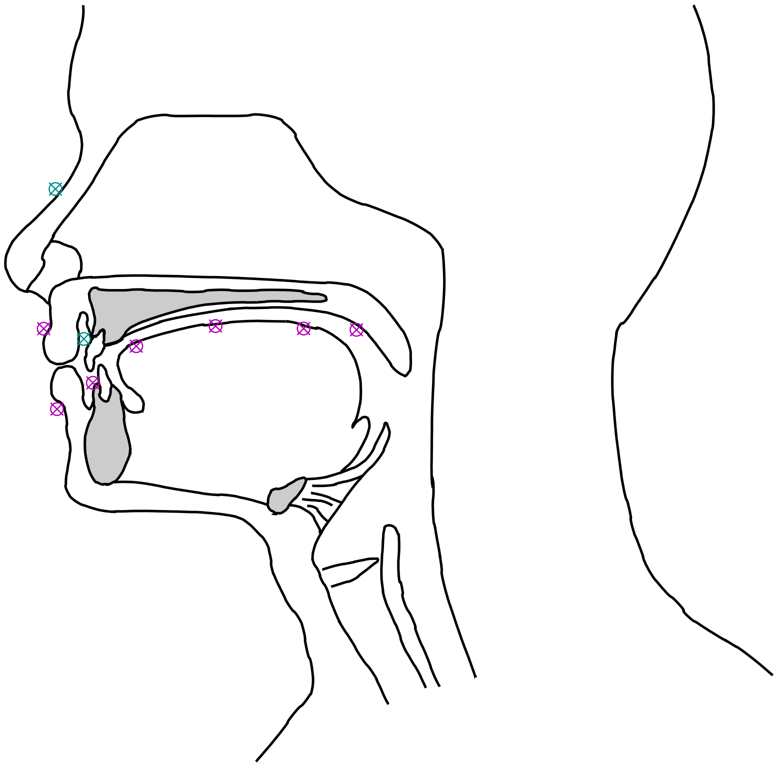


Figure 3.2: The placement of the EMA coils used in the MOCHA database is shown in this diagram of the midsagittal section of a human head. The Magenta coils give a signal which is used as training and testing data. Meanwhile, the two cyan coils shown are used as part of an algorithm to correct for head movement relative to the EMA helmet.

Channel	Articulator
ui_x ui_y	upper incisor (reference coil)
li_x li_y	lower incisor
ul_x ul_y	upper lip
ll_x ll_y	lower lip
tt_x tt_y	tongue tip (5-10mm from extended tip)
tb_x tb_y	tongue body (approx. 2-3cm beyond tt coil)
td_x td_y	tongue dorsum (approx. 2-3cm beyond tb coil)
v_x v_y	velum (approx. 1-2cm beyond hard palate)
bn_x bn_y	bridge of the nose (reference coil)

Table 3.1: EMA channel names. Coils are attached to nine points in the midsagittal plane. The suffix _x corresponds to the x-coordinate of the coil at a given articulator in the midsagittal plane, while _y corresponds to the y-coordinate. Note that two of the coils are affixed to immobile points; the bridge of the nose and the upper incisor. These coils are used by an algorithm which compensates for head movement within the scanner. Therefore, there are seven coils which provide 14 channels of salient articulatory information.

All data are recorded direct to computer and carefully synchronised. In order to facilitate recording these data streams concurrently, the speech is recorded simultaneously onto three computers synchronised using serial port communication.

Table 3.1 shows the names for the EMA channels that are used throughout this thesis. Figure 3.2 is a diagram of the midsagittal section through a human head. The approximate positions of the coils for the recordings in the MOCHA database are shown.

All coordinates are processed to compensate for head movement, and the coordinate system is rotated so that the x-axis lies along the line of a T-bar held between the speaker's teeth, known as the occlusal plane.

Each speaker is recorded reading a set of 460 British TIMIT sentences. These short sentences are designed to provide "phonetically-diverse" material to maximise the usefulness of the data for speech technology and speech science research purposes. It is intended to capture the main connected speech processes in English with good coverage. Details of events such as a coil becoming detached, speaker mistakes, recording glitches etc. are noted down and provided with the data.

Throughout this thesis, the data files are referred to using the same naming convention as used in the MOCHA database. A file base name is made up of two main parts: the speaker identifier and the utterance identifier. The speaker code is made up from the speaker's initials prefixed with either "m" or "f" depending on their sex, and suffixed with a unique digit. The three digit utterance identifier is the index to the TIMIT utterance that is read for that file. For example the base name `msak0_192` would mean "Male speaker with initials S.A.K. reading sentence 192 ('Who authorised the unlimited expense account?')." File extensions appended to these base names identify the type of data where necessary. These may be ".wav", ".ema", ".lar" and ".epg" to indicate waveform, EMA, laryngograph and electropalatography data respectively.

3.4 Data processing

The work presented in this thesis is all based on the recordings of the female speaker `fsew0` from the MOCHA database, described in Section 3.3. The raw data at our disposal consists of the acoustic waveform and the corresponding articulatory trajectories which produced the speech signal, as in the example shown in Figure 3.3 for instance. In general terms, the inversion task involves presenting an acoustic signal, such as that shown in the bottom window of Figure 3.3, as input to some mapping system which aims to estimate the corresponding articulatory trajectories shown above as accurately as possible. However, the data in this raw form is not necessarily the most suitable for use as training data. In order to render this parallel articulatory and acoustic data into a more suitable format, i.e. for use with neural networks, several processing steps were necessary.

First, filterbank analysis was carried out on the acoustic signal, using a Hamming window of 20ms with a shift of 10ms. For each time frame, the acoustic vector consisted of 20 mel-scale filterbank coefficients.

The EMA traces were downsampled to match the 10ms shift rate of these acoustic

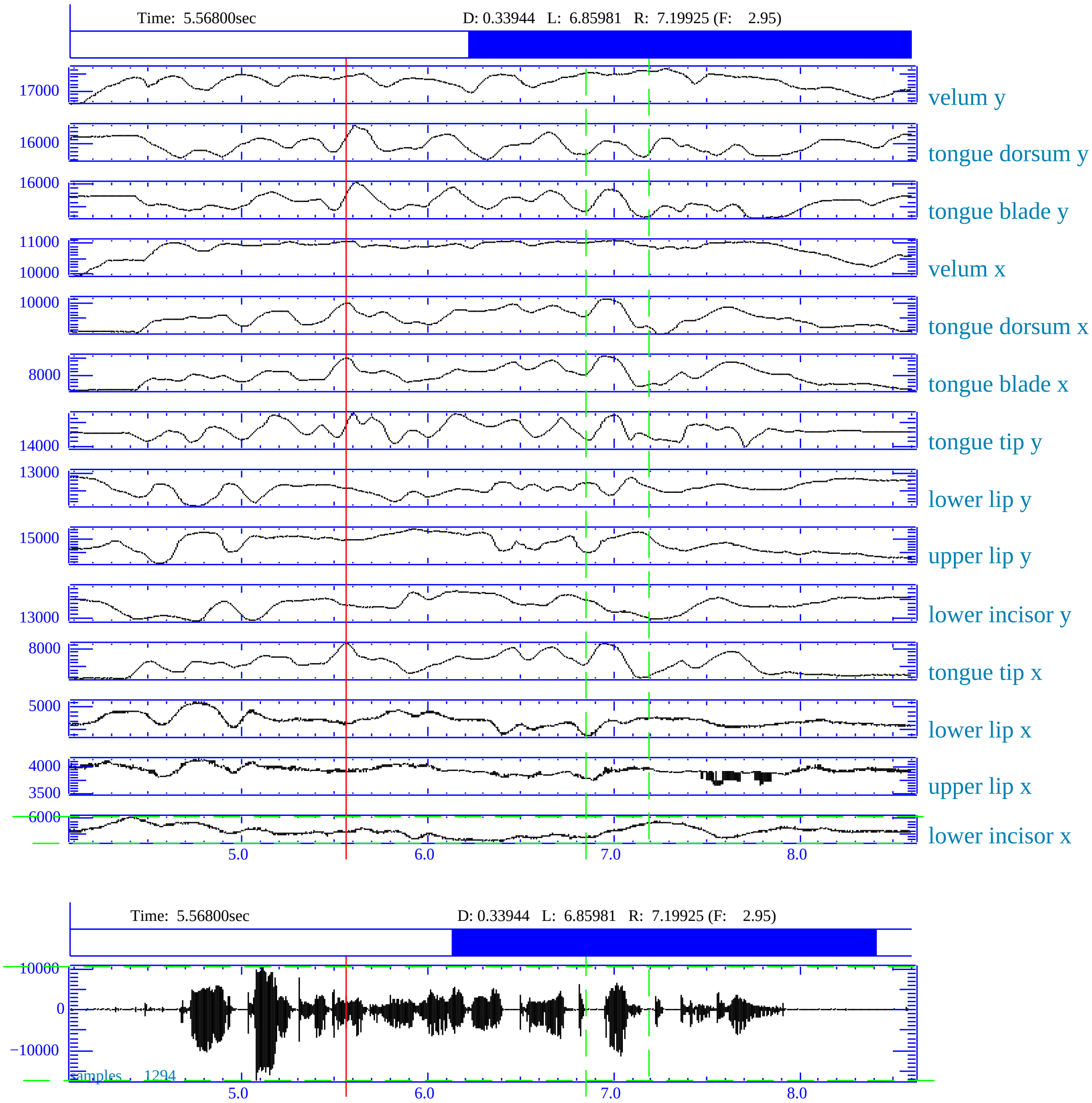


Figure 3.3: Parallel acoustic and articulatory data recorded for 460 TIMIT utterances, from the MOCHA corpus.

feature vectors. An important step in the downsampling process is to lowpass filter the signal to avoid aliasing of higher frequency noise, such as that resulting from EMA measurement error, in the decimated signal. Thus, the signal was first filtered in the forward direction using an FIR filter, then reversed and filtered again, and finally reversed once more. This "double filtering" was carried out in order to counteract phase distortion in the filtered signal.

From the 460 utterances contained in the data set of speaker `fsew0`, 368 files were included in the training set, 46 files in a validation set, while 46 files were put aside for the test set[4].

Within the database of speaker `fsew0`, the files for each utterance contain an average of approximately 1.3 seconds of silence in total before and after the utterance. This compares with an average utterance length of 2.69 seconds. During the silent stretches, the mouth can theoretically take any configuration. This could pose a serious problem to network training, because given an acoustic feature vector representing silence, the network would be attempting to map to a large range of possible articulatory configurations. Therefore, data from silent stretches were omitted from the training set. This was done using the labelling created by HMM forced alignment and provided with the `fsew0` data set. The full training set contained 92,557 pairs of acoustic and articulatory feature vectors.

The ranges of these acoustic input vectors and articulatory output vectors are unsuitable for use with an MLP. In an MLP whose output units have the standard sigmoid non-linear activation function, the output values can lie within the range [0.0, 1.0], where 0.0 and 1.0 are unrealisable asymptotic limits. The acoustic-articulatory feature vector pairs must therefore be normalised to lie within convenient ranges.

3.4.1 Normalisation

A standard technique for data normalisation involves subtracting the mean and dividing by the standard deviation, and finally shifting and scaling the signal for 2 standard deviations (95% of a normal distribution) to lie within the required range.

[4] This was achieved simply by selecting files which had a file number ending in "6" for testing utterances and ones ending in "2" for validation utterances.

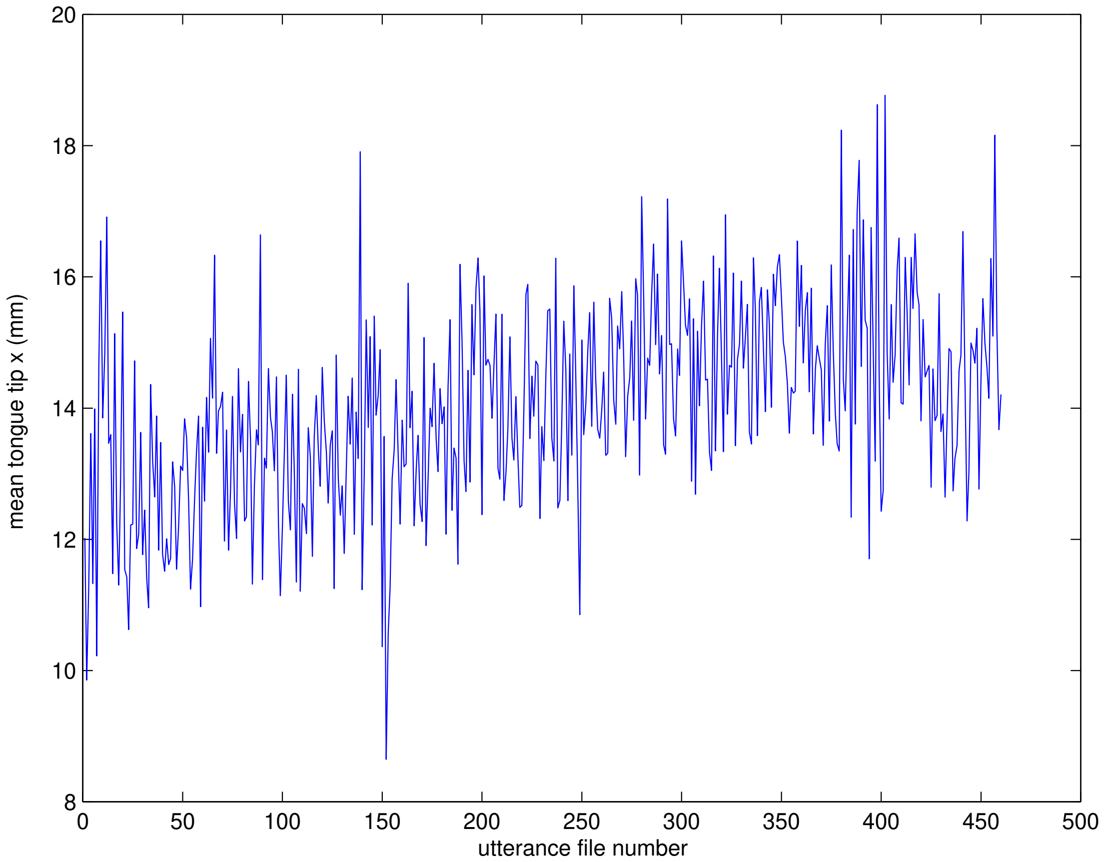


Figure 3.4: A plot of mean tongue tip x-coordinate calculated for each utterance from the database for speaker `fsew0` in the sequence of recording. Notice the slight DC drift apparent as a trend throughout the sequential data files.

Hence, to normalise the EMA data, we could calculate the global means and standard deviations for all articulators across the whole data set and use these for normalisation. However, analysis of the raw data indicates there may be difficulties in doing this. Figures 3.4 and 3.5 illustrate at least two of the considerations which might arise.

Figure 3.4 shows a plot of the mean velum x coordinate calculated on an utterance by utterance basis for speaker `fsew0`. The silent sections at the beginning and end of the files were not included in the calculation of these mean values. If we presume the mean articulator positions vary according to the phonemic content of the utterance, we would expect to see a random variation in the mean articulator values throughout a corpus. In Figure 3.4, we see such random variation. However, this plot also shows a

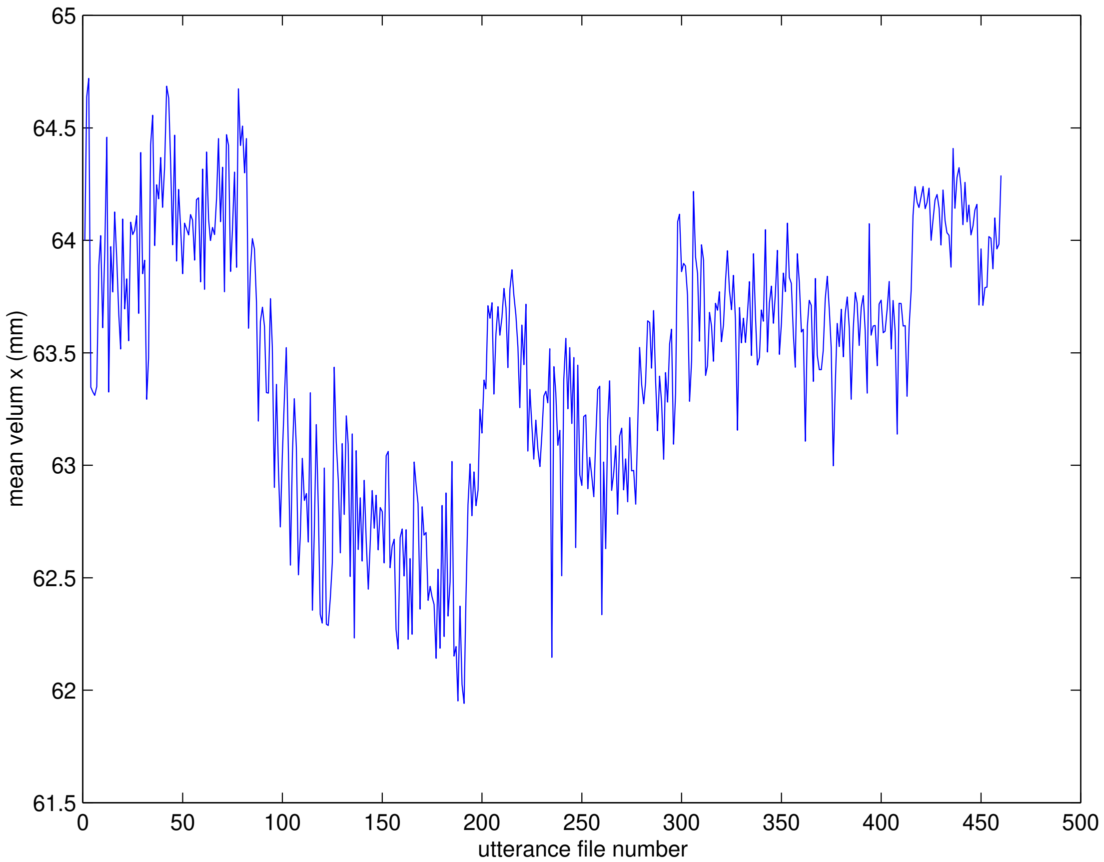


Figure 3.5: A plot of mean velum x coordinate calculated for each utterance from the database for speaker `fsew0` in the sequence of recording. Notice an apparent underlying trend during different sections of the database.

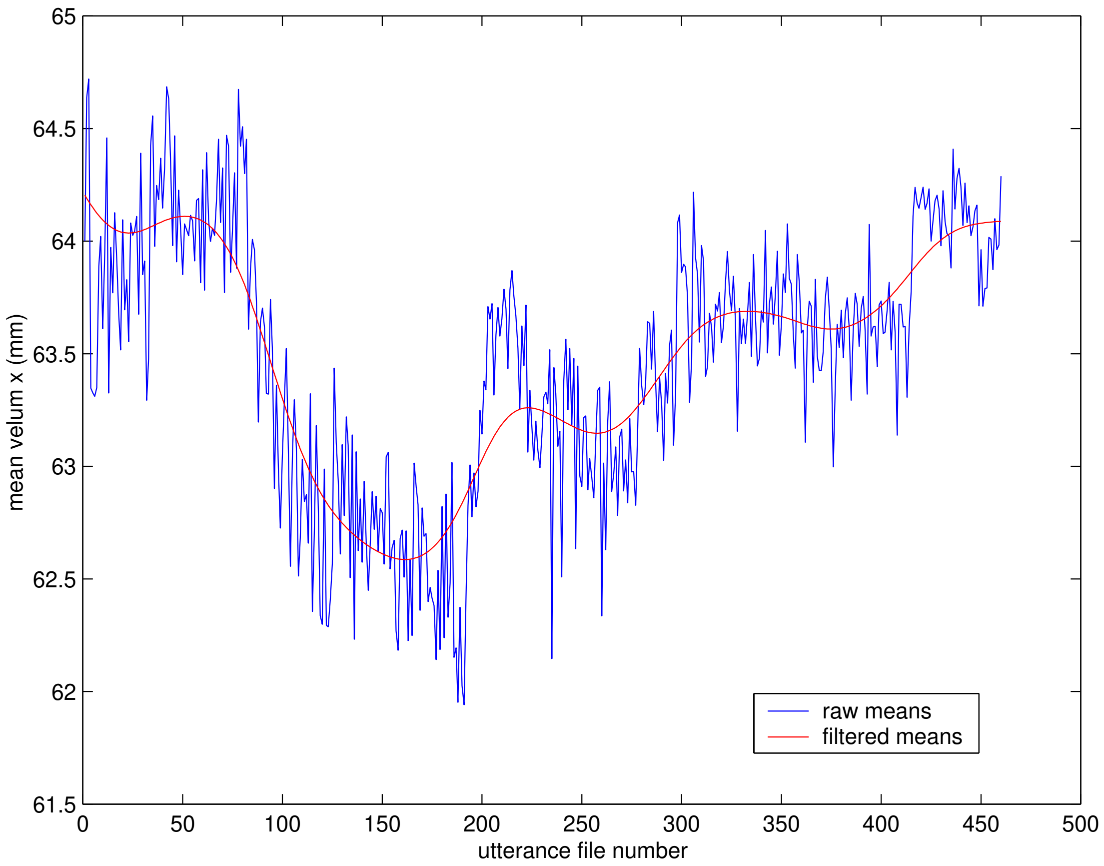


Figure 3.6: A plot of mean velum x coordinate calculated for each utterance from the database for speaker `fsew0` in the sequence of recording. This is the same as shown in Figure 3.5. However, in addition, this diagram also shows the underlying variation, captured by lowpass filtering the sequence of raw means. This trend can be used for the mean values in normalisation. This is similar to using a moving average mean, as opposed to the two extremes of either using the global mean across all utterances, or using the means calculated on an utterance by utterance basis.

DC drift that is apparent through this "noise". Likewise, looking at Figure 3.5, we see a trend apparently underlying random variation within a certain range.

A number of causes for these underlying trends are possible, although the definitive explanation is not yet known. It could be that the phonemic content of the TIMIT utterances is organised in some specific way over the files in recording order. On the hardware side, it has been suggested that temperature changes in the recording equipment could contribute to these observations[5]. It might be suggested that any shifts in the location of the EMA helmet and transmitter coils relative to the subject's head would cause a trend or discontinuity. In principle, the algorithm which corrects for head movement should be capable of correcting for such cases. However, if an EMA sensor coil becomes detached during the recording, it is very difficult to ensure that it is replaced in exactly the same location, and therefore a sudden discontinuity could be present in the recordings for that articulator as a result.

Alternatively, it may be the case that throughout the recording session, the speaker becomes increasingly accustomed to the presence of the articulography equipment, especially the EMA coils and the EPG palate, and consequently their articulation adapts in certain ways[6]. As part of the recording protocol, the subject is requested to wear the EPG palate for a few hours prior to recording in order to become fully accustomed to speaking with it in place. However, it would not really be practical for the speaker to have the EMA receiver coils positioned several hours before the recording session, because they have a tendency to become detached relatively quickly.

To avoid any underlying trends, we might use the means and standard deviations calculated on a file-by-file basis. However, the statistics calculated in this way would be sensitive to the phonemic content of each utterance. For example, during an utterance which contains a large number of nasal segments, such as [m,n], the velum would be lowered for a significantly large amount of time. Therefore, the mean velum height for this utterance would be lower than that for an utterance which contains segments that require the velar port to remain closed. If we were to use means and standard deviations calculated in this way, we should anticipate running the risk of introducing inconsistency

[5]Personal communication, Alan Wrench, Queen Margaret College

[6]This is quite likely, judging from personal experience as a subject

in the articulator values we expect the neural network to learn to output. A velum height of h during an [a] vowel taken from an utterance containing many nasal segments would be lower in the *normalised* space than the same velum height h during an identical [a] vowel, but which is taken from an utterance containing no nasal segments.

If we assume that rapid variation from utterance to utterance results from the differing phonemic content of each utterance, which is something we wish to maintain as far as possible, then a reasonable course of action would seem to be to highpass filter the "signal" of the means in order to remove the trends which are relatively slow moving. Figure 3.6 shows the same plot of means calculated on a file by file basis as Figure 3.5. However, overlaid on this plot is the result of lowpass filtering the signal. Raising this lowpass cutoff value results in incorporating more of the variation observed. This is equivalent to using a smaller moving average window. Now that underlying trends have been identified in the lowpass filtered means, their effect can be diminished in the normalisation process. This can be achieved by using the mean values dictated by the filtered means signal when normalising each file. In other words, to "zero-mean" an articulatory trajectory, we first subtract the file-specific mean returned as a result of lowpass filtering the file means taken in recording order. Next, the trajectory can be scaled to lie within the desired range using the channel's respective standard deviation. Inconsistencies in the recording of absolute articulator position will not affect the range of movement observed in the same way as the mean articulator position. Hence, the standard deviation for use in normalisation can be calculated across the whole corpus, but using the file-specific means identified as part of the "mean filtering" process.

Empirical evidence that using means and variances calculated by the method detailed here yields better results than when using the global means and variances is to be found in Appendix B. On the basis of improved results, unless otherwise stated, all experiments in this thesis use the same training, validation and testing data sets, obtained from the `fsew0` corpus by normalising using the method described here.

To summarise, the filterbank coefficients were normalised to lie within the range [0.0, 1.0]. This was executed using global means and variances, calculated across all 460 utterances in the `fsew0` corpus. Meanwhile, the articulatory traces were normalised to lie within the range of [0.1, 0.9], using the "means-filtering" method described above.

This range differed from that of the acoustic coefficients, because the logistic activation function of the output units in the MLP has the unrealisable asymptotic limits of 0.0 and 1.0. Finally, it is important to note that the silence at the beginning and end of each file was ignored when calculating the means and standard deviations used for normalisation.

3.4.2 Context windows

As already noted, we have used a window of 20ms, centred at time t, to calculate a single frame of filterbank coefficients from the acoustic waveform. However, at several points in this thesis, we will find occasion to use more than 20ms of acoustic data as input to an inversion system. In order to achieve larger acoustic windows, we have chosen simply to concatenate multiple acoustic frames to form a 'context window'. Specifically, to obtain an acoustic window with two frames, we have chosen to adjoin an extra frame to the right of the single acoustic frame at any time t, i.e. centred at time $t + 10$ms. Then, for a context window using 3 frames, we have chosen to append a frame to the left of this context window, i.e. centred at time $t - 10ms$. This process of alternately adding frames to the right then the left was continued to yield context windows of whatever number of frames were required.

Chapter 4

Articulatory Estimation During Continuous Speech by MLP

4.1 Introduction

Common sense dictates favouring an efficient algorithm over computationally more expensive ones, assuming comparable efficacy. A neural network, once trained, requires relatively modest computational resources compared to many other empirical learning models, in terms of both memory space and speed of execution. If a neural network model were capable of performing the inversion mapping with adequate accuracy, it would represent a most satisfactory solution from this point of view. Thus, it is a primary aim of this chapter to exploit the availability of the corpus of phonetically diverse EMA data to extend the work previously reported towards a system with the kind of scope of operation which would be required in practical applications.

The previous work which is probably most similar to the task undertaken in this chapter is that of Papcun et al. (1992). The articulatory-acoustic data at their disposal, provided by the X-ray Microbeam Facility at the Waisman Center of the University of Wisconsin, is very similar in character to EMA data. Their ubiquitously cited paper reported encouraging results on small amounts of speech. As noted in Chapter 2, the data they used consisted of just six utterances of the form [CəCəCəCəCə] produced by each of three male speakers, with C drawn from the set of oral stops /p, b, t, d, k, g/.

The corpus used for the experiments in this thesis is comprised of 460 British TIMIT sentences. This corpus was designed with the intention of incorporating a broad phonetic diversity. Attempting to estimate articulation during acoustics for this more wide-ranging speech presents a greater challenge than the task undertaken by Papcun et al. (1992). In addition, Papcun et al. (1992) only used the y-coordinates of three articulatory points on the tongue, whereas we attempt here to recover both x- and y-coordinates for seven articulatory points from the acoustic signal.

Prior to training an MLP to perform the inversion mapping, it is wise to stipulate how exhaustive a search for the best MLP is reasonable for the present study. Striving to achieve the best possible performance in the MLP-based inversion mapping, we could risk becoming bogged down in the *minutiae* of implementation details. However, as a general principle, it is unwise to optimise on intermediate results. At this "proof-of-concept" stage, it no doubt makes most sense to limit efforts to obtaining a *reasonable* inversion mapping system, for example one that is close to or matches the performance of other systems reported previously. It will then be subsequently be possible to assess how this inversion mapping might be improved. The important principle is to compare like with like as far as possible. Hence, assuming the MLP does demonstrate a reasonable performance, many decisions taken concerning exact implementation details are not critical at this stage. As long as these implementation details, such as choice of acoustic feature extraction algorithm, remain constant as far as possible, we will be able to assess the benefit of modifications to the baseline model.

The rest of this chapter begins by discussing the issue of what MLP architecture might be required for modelling the inversion mapping. Subsequent to this discussion, the results of an MLP are presented and analysed. We go on to look at one way the output of the MLP can be postprocessed to improve the performance of the MLP inversion mapping, before finally discussing certain other aspects of the MLP with respect to modelling the inversion mapping.

Study	# hidden units		Notes
	1st layer	2nd layer	
Papcun et al. (1992)	8	8	articulator specific network
Zachs & Thomas (1994)	8	8	articulator specific network
Rahim et al. (1991)	140	60	global network mapping
Rahim et al. (1991)	26	0	128 local network mappings
Kobayashi et al. (1991)	24	24	global network mapping

Table 4.1: Comparison of hidden layer topologies used in some previous applications of neural networks for recovering articulation from acoustics.

4.2 Network architecture

A notorious problem when using neural networks is how to decide what topology of links and units to use. To help resolve this question with respect to the inversion mapping problem, it seems prudent to refer to what other network architectures have been employed in the past.

In Chapter 2, we looked at several studies involving neural networks applied to the inversion mapping. Table 4.1 summarises the architectures of the networks described.

Unfortunately, none of these studies include any account of arguments justifying the choice of network topology. At most, the authors might mention in passing a lengthy process of trial and error without going into details. More often, however, a neural network architecture is merely presented without discussion. It is hard to identify many similarities between the various network hidden layer topologies previously employed. Nevertheless, it is worth considering the benefits of features exhibited by previous neural networks.

4.2.1 Input context

The networks described by Papcun et al. (1992) (and the follow-up work of Zachs & Thomas (1994)) featured a context window of acoustic input frames. When not using a context window, the network is presented with a single acoustic input vector at each time frame and trained to map to the single corresponding articulatory vector. In contrast, where a context window is used, multiple frames of acoustic coefficients are presented as input. The network may be trained to map to the articulatory feature vector which corresponds to the acoustic frame at the centre or at some other time frame within the context window. Papcun et al. (1992) used a context window of 25 frames, which covered approximately 200ms of the speech signal.

In view of the non-uniqueness in the instantaneous acoustic-to-articulatory mapping discussed in Section 2.2, it seems reasonable to include a context window of acoustic inputs to help disambiguate points of instantaneous one-to-many mappings. In addition, work with linear inversion mappings, presented in Appendix A, indicates the potential benefit of using windows of multiple acoustic frames.

4.2.2 Articulator-specific networks

Papcun et al. (1992) used separate networks for each articulator, referring to the set of networks collectively as a "composite" network, although it is not clear why they did this.

Theory does not seem to suggest a contraindication to training all articulators together as the output of a single network. On the contrary, one of the advantages of training a 'global' network with outputs for all articulators as opposed to training separate networks for each articulatory parameter is the possibility of finding and capitalising on redundancy in the mapping to economise on the overall number of trainable parameters. For example, we might hope to benefit from the highly correlated nature of the movements of the articulators, as demonstrated in Figure 4.1. This diagram shows a scatter plot of multiple samples of velum coil movement in the midsagittal plane. A definite pattern is visible, with the velum-x and velum-y highly correlated. Due to the physical construction of the mouth, there are many other such correlations between the movements of the articulators with varying degrees of complexity. Therefore, we shall use a single MLP with outputs for each of the fourteen articulator channels.

4.2.3 How many layers?

From looking at Table 4.1, possibly the most noticeable feature shared by many of the network topologies described in previous work is that researchers have opted to use two layers of hidden units as opposed to just one.

Unfortunately, the principles surrounding the use of more than one layer of hidden

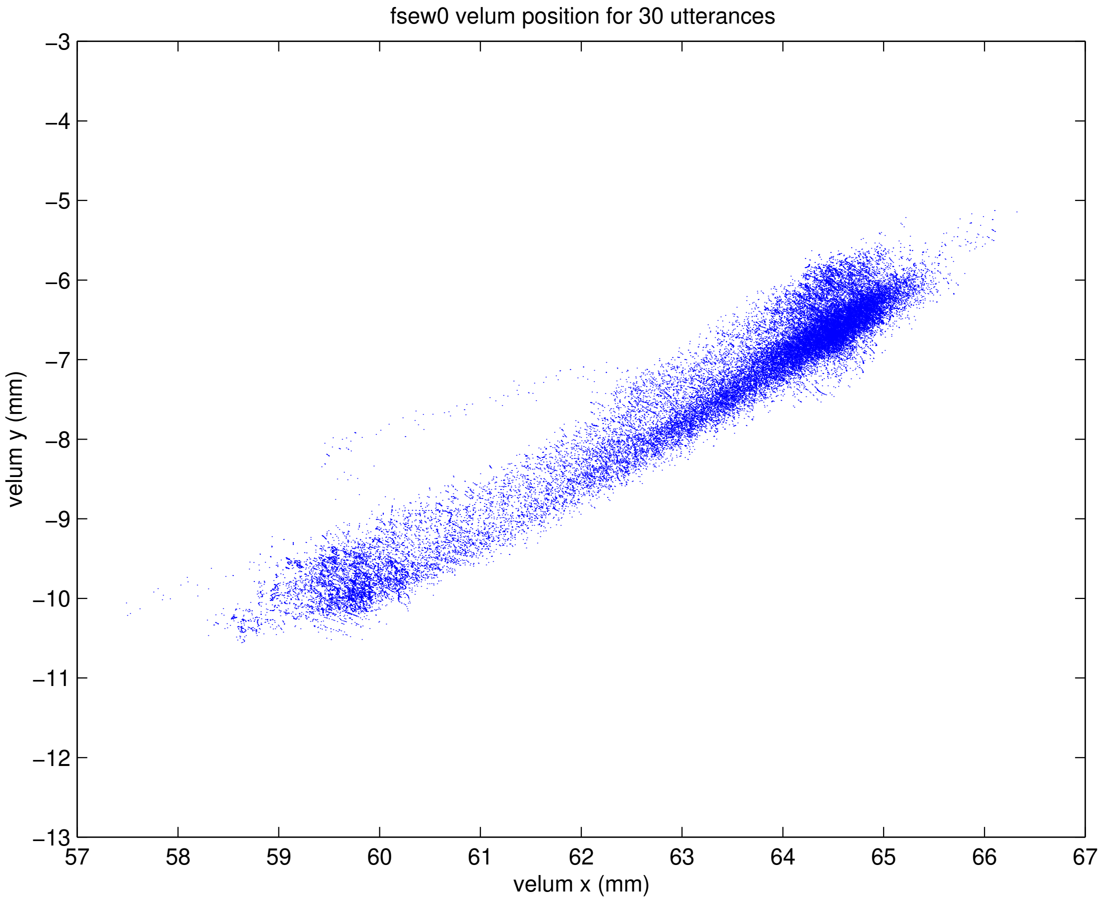


Figure 4.1: Scatter plot of velum position for 30 utterances from speaker fsew0. The movements of the velum in the y-direction correlate highly with movements in the x-direction

units are not yet well understood. One alluring possibility is that a network with multiple hidden layers could represent a more compact, efficient approximation (Bishop 1995). However, as well as proposing some benefits for having multiple hidden layers, some researchers have reported finding that using two hidden layers can exacerbate the problem of local minima. With two hidden layers, it has been shown that local minima can take the form of extreme spikes or blades even when the number of weights is much smaller than the number of training cases.

Moreover, it has been demonstrated that an MLP with a single hidden layer containing a sufficient number of units is able to approximate any mapping function with arbitrary accuracy (Bishop 1995). In view of this, and the fact that the potential advantage of a more compact representation is not relevant to the current investigation, we will limit our attention here to using a single layer of hidden units.

4.2.4 Rules-of-thumb

Numerous "rules of thumb" circulate the neural network field on the subject of how to choose the optimum number of hidden units for a particular application. For example, it has been suggested that the size of the hidden layer should lie somewhere between the sizes of the input and output layers (Blum 1992). Another rule of thumb discounts using any more hidden units than twice the number of input units (Berry & Linoff 1997);(Swingler 1996). Unfortunately, such rules of thumb are overwhelmingly either crude and inadequate approximations at best, or unprincipled at worst.

The question of what network topology should be used to obtain the best performance hinges upon several interdependent factors:

- the dimensionality of the input and output spaces
- complexity of the mapping between the input and output space
- number of input-output pairs in the training set
- amount of noise in the data
- type of activation functions used

- which training algorithm is used
- how many layers are used

However, the exact relationship between these factors is complex. As yet, no strong theoretical framework exists to guide the topology of a neural network applied to a specific task.

4.3 Preliminary MLP

The inconvenient conclusion from the discussion in Section 4.2 is that we are not able to identify with certainty what the most suitable architecture for an MLP applied to the acoustic-to-articulatory inversion mapping problem might be. We have some indications for features which might be helpful, but there is no obvious choice for an MLP architecture.

Deciding on the best neural network architecture rarely seems to be straightforward. Frequently, researchers have to resort to simple trial and error in order to determine the most suitable network topology for a task. Likewise, in this section, we will pick a "best-guess" architecture for a preliminary MLP which draws from some of the factors discussed in Section 4.2, and evaluate the results we obtain on the inversion mapping task. A network with too many hidden units may produce a low training error, but can still have high generalisation error due to overfitting[1] and high variance. Conversely, a network with too few hidden units will result in high error on both the training and testing data sets, due to underfitting and high statistical bias.

4.3.1 Initial MLP architecture

Based in part on the discussion of Section 4.2, the architecture shown in Figure 4.2 was chosen for an initial MLP inversion attempt. This MLP comprises an input layer of 400 units, a single hidden layer of 50 units, and an output layer containing 1 unit for each of the 14 articulatory channels. The 400 unit input layer gives a context window of 20

[1] Overfitting is where trainable parameters within the model are used to accommodate idiosyncrasies in the training set which are not part of the underlying function the network is expected to learn.

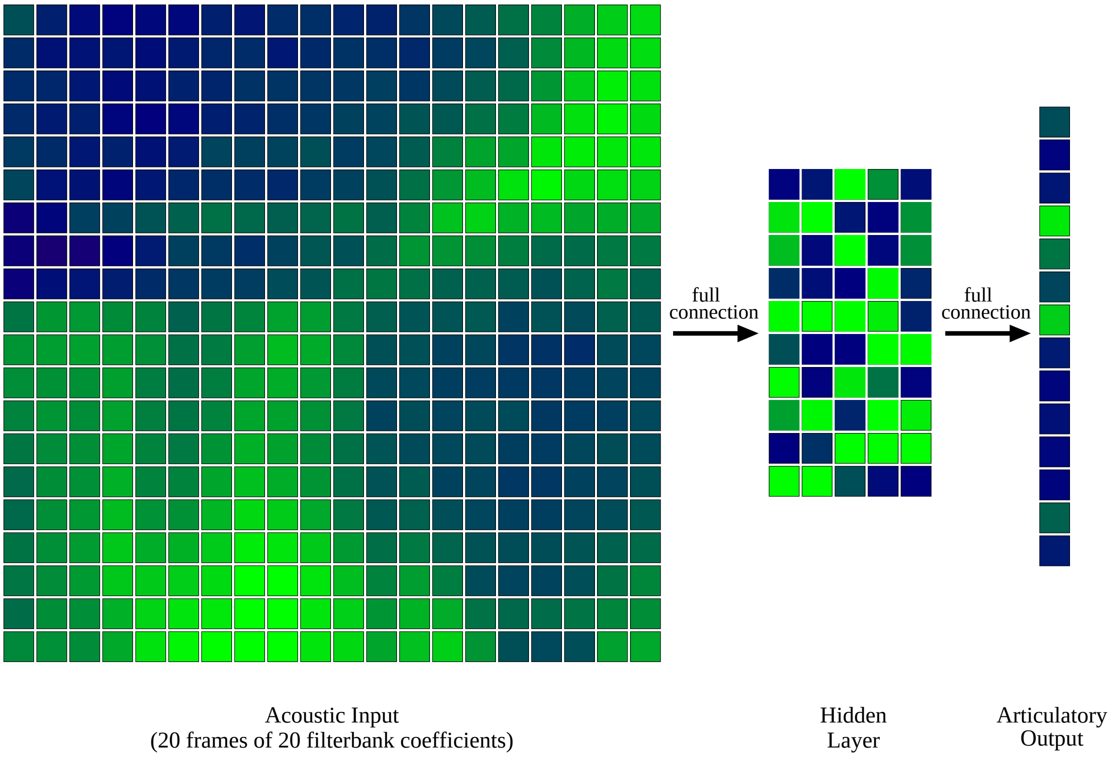


Figure 4.2: The initial feedforward neural network. In the input layer, the total number of units (400) is equalled to the number of acoustic coefficients chosen at each time frame (20) multiplied by the number of time frames included as context (20). There are 50 units in a single hidden layer, and 14 output units, one for each of the x- and y-coordinates of seven articulator points.

frames of 20 filterbank coefficients each. Thus, similar to the networks used by Papcun et al. (1992), this window covers 210ms of the input acoustic speech signal.

4.3.2 Nonlinear optimisation

We can view training an MLP as a two stage process, which is repeated iteratively (see Bishop (1995), pp141). The first stage is to calculate the error of the output units compared with the target output according to some error function, and propagate this back (hence "backpropagation") through the network to find the contribution of each weight to this error. The second stage is to adjust the weights so that the overall error is reduced. Deciding how to adjust the weights to reach the best network performance is a nonlinear optimisation problem.

There are many nonlinear optimisation algorithms available for optimising the network weights. The choice of which algorithm to use depends specifically on the task in hand. Two major categories of optimisation algorithms are first and second order algorithms.

First order optimisation algorithms, as their name implies, only make use of the first derivatives of the error function. When calculating how to adjust the weights at each update, such algorithms only rely on a single calculation of the *gradient* of the error function with respect to the weights. Once this gradient is known, the various algorithms employ different strategies for updating the weights in order ultimately to approach the optimum configuration of the weights, which corresponds to the minimum in the error function. The simplest of these is gradient descent, which is the update rule used in standard "Vanilla Backpropagation", invented independently several times but mostly attributed to Rumelhart, Hinton & Williams (1986). Under this weight update scheme, each weight is adjusted by a step down the direction of the gradient. The step size is governed by a user defined constant η. Neural network textbooks (such as Bishop (1995) or Hertz, Krogh & Palmer (1991)) will invariably cover a slew of more complicated variations on this approach, which feature more elaborate ways of choosing the step to adjust each weight. For example, momentum techniques (e.g. Plaut, Nowlan & Hinton (1986)) include a term which is a proportion of the weight change at the previous weight update to regulate the step size automatically.

More powerful second order optimisation methods make use of additional information about the local form of the error function to estimate the location of the local minimum. Perhaps the purest example is *Newton's method*, where the second derivative Hessian matrix is explicitly used iteratively. Unfortunately, Newton's method is very expensive computationally, because is requires the inversion of an $n \times n$ Hessian matrix at each iteration, taking order n^3 operations, where n is the number of network weights. Moreover, computation of the second derivatives does not fit into the convenient and efficient backpropagation framework. In practice, more efficient approximations are used. For a reasonably small number of weights, various quasi-Newton algorithms are useful, where the idea is to use the Newton rule, but with an *approximation* to avoid computing the inverse Hessian matrix. For larger numbers of weights, various conjugate-gradient algorithms can be very effective, which use line searches along selected directions.

As with standard backpropagation gradient descent, conjugate gradient methods iteratively approach a minimum in the error function. However, whereas standard backpropagation always proceeds in the direction of the gradient of the error function, a conjugate gradient method will proceed in a direction which is conjugate to the direction of previous steps. Thus, the minimisation performed at one step is not partially undone by the next. It is the generally held view that second order techniques find a more direct way to a (local) minimum than first order techniques, although they incur higher computational cost at each cycle of training. The Scaled Conjugate Gradients (SCG) algorithm has been shown to be considerably faster than standard backpropagation and other conjugate gradient methods (Møller 1993).

The SCG optimisation algorithm also has the convenience of not requiring the user to decide values for critical parameters, as all the algorithm's parameters are adaptive and the user only has to supply non-crucial initial values. This contrasts with the simple gradient descent optimisation, which is sensitive to the user supplied learning parameter η. When η is small, the network is very slow to train and may become more easily stuck in local minima. Conversely, when η is large, training can become erratic.

It is worth mentioning in passing an additional consideration in connection with the size of η when using an "online" training algorithm, such as vanilla backpropagation, for training an MLP on inversion mapping data. When training in online mode, it seems to be important to randomise the sequence of training patterns presented to the network. This presumably avoids the network fitting local features of the database during training, and gradually modifying such fitting as training progresses through the training set. This danger could arise because at each time the neural network trainable parameters are updated, such online training algorithms attempt to minimise the error function for the *current* input-output vector pair only. For all but the most trivial learning problems each training pattern taken in isolation is only a noisy approximation of part of the global mapping function to be learned. So, at each update time, the weights may be adjusted to (over)compensate for the mapping represented by the current pattern, as well as extraneous noise. It is possible that due to the sequential nature of the speech training data, where neighbouring patterns represent closely sampled local sections of the overall mapping function, the network will tend not to learn the global mapping. Instead, it will tend to overfit local regions of the mapping function, where the region fitted varies as training proceeds through the data set. This effect may be lessened by using a very small learning parameter (η), so that the weights are only adjusted by an amount too small to be over-sensitive to local continuity effects. However, this would bring with it the disadvantage of decreased speed of convergence.

When training with a batch update algorithm, such as SCG, randomisation of the order of the training set patterns is not necessary. For these algorithms, the statistics necessary for calculating the network weight updates are accumulated across all patterns at once first, and then the weight update which is necessary is calculated. Hence, at each update step, the training algorithm is minimising the error function of the network for *all training patterns at once.*

As is common practice, the network shown in Figure 4.2 was first initialised by randomising the weights to lie in the range [-1.0, 1.0], and then trained using the SCG optimisation algorithm until the RMS error calculated on the separate validation set ceased to reduce.

4.3.3 Results

Two measures that have been used in the past to compare trajectories such as traces of articulatory movements are RMS error and correlation. RMS error is an indication of the overall 'distance' between two trajectories. The correlation score is an indication of similarity of shape and synchrony of two trajectories.

Table 4.2 presents the results for the initial MLP shown in Figure 4.2, when estimating articulation for the unseen test set containing 46 utterances. The first column of Table 4.2 gives the RMS error, calculated as:

$$E_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(o_i - t_i)^2} \tag{4.1}$$

where N is the number of input-output vector pairs, or patterns, in the testing set, o_i is the estimated value for the articulator channel output by the network, and t_i is the

Articulator	RMS error	RMS error(mm)	Correlation
upper lip x	0.176	0.99	0.591
upper lip y	0.145	1.17	0.712
lower lip x	0.160	1.22	0.605
lower lip y	0.138	2.76	0.742
lower incisor x	0.163	0.89	0.561
lower incisor y	0.119	1.19	0.795
tongue tip x	0.130	2.50	0.780
tongue tip y	0.116	2.58	0.836
tongue body x	0.125	2.22	0.803
tongue body y	0.113	2.19	0.823
tongue dorsum x	0.128	2.08	0.784
tongue dorsum y	0.138	2.36	0.695
velum x	0.130	0.43	0.777
velum y	0.130	0.42	0.761

Table 4.2: Performance of the initial MLP containing 50 hidden units, shown in Figure 4.2, when recovering articulation from acoustics for the unseen test set. Performance is assessed by calculating RMS error and correlation for each articulatory channel. The average of the RMS error values given is 1.64mm.

real value.

The second column gives the same RMS errors, but scaled back to the original domain of EMA measurement in millimetres. The rescaling is calculated for each articulatory channel as follows:

$$E_{RMS(mm)} = \frac{4\sigma_a E_{RMS}}{l_u - l_l} \tag{4.2}$$

where, σ_a, l_u and l_l are parameters originally used to normalise the raw EMA data. Specifically, σ_a is the standard deviation for the articulatory channel, and l_u and l_l are the upper and lower limits of the range to which the articulatory data was scaled during normalisation (in this case 0.9 and 0.1 respectively).

Finally, the third column gives a measure of correlation between the actual articulatory trajectory and the network estimated trajectory, calculated by dividing their covariance by the square root of the product of their variances

$$r = \frac{\sum\limits_i (o_i - \bar{o})(t_i - \bar{t})}{\sqrt{\sum\limits_i (o_i - \bar{o})^2 \sum\limits_i (t_i - \bar{t})^2}} \tag{4.3}$$

where $\bar{o}$ and $\bar{t}$ are the mean channel value for the network output and actual articulator position respectively.

Looking at the values for RMS error expressed in millimetres in Table 4.2, it would seem that the network is not able to recover the different articulatory channels with equal accuracy. While this is certainly true, we should be careful to recognise the effect of the differing overall range of movement for each of the articulators. For example, the highest error *as expressed in millimetres* is found in the network estimation of the lower lip y parameter. However, as a percentage of the range of movement of the lower lip, this error is smaller than that for the network estimate of the upper lip x channel. In fact, in terms of the *network output range* the upper lip x channel actually has the highest error. In this respect, although researchers have in the past typically reported RMS error rates expressed in millimetres, it is often more informative to consider the errors for each channel as a percentage of their respective overall range of movement. Judging by the unscaled RMS error (and correlation) values alone, the MLP is best able to infer the y-coordinate of the tongue tip and body points. Conversely, the MLP seems worst at inferring the x-coordinate of the lips and lower incisor.

4.4 Pruning

In Section 4.3, we trained and tested a feedforward MLP with 50 hidden units to perform the inversion mapping. However, two important questions so far remain unanswered. First, it is unclear on the basis of the results in Table 4.2 whether a non-linear mapping, as provided by the MLP, is in fact necessary. To address this question, the reader is referred to Appendix A, where we have evaluated the performance of a linear inversion mapping. In particular, the reader should compare the results in Table 4.2 with those in Table A.2, where a linear mapping has been fitted and tested using the same data sets as the MLP. As can be seen, the performance of our initial MLP surpasses that of the linear model. Hence, this is taken as evidence to justify the use of a non-linear function to model the inversion mapping.

If we accept that a non-linear function is more appropriate for modelling the inversion mapping, then the second obvious question which arises pertains to just how non-linear

	number of context frames					
hidden units	1	3	5	10	15	20
10	0.351	0.332	0.319	0.304	0.298	0.303
20	0.335	0.315	0.301	0.284	0.276	0.278
30	0.329	0.303	0.292	0.273	0.263	0.264
40	0.324	0.303	0.283	0.268	0.257	0.257
50	0.323	0.298	0.284	0.261	0.252	0.251
60	0.324	0.298	0.280	0.259	0.249	0.248
70	0.321	0.296	0.278	0.259	0.251	0.252

Table 4.3: Effect of hidden layer and context window size on the performance of an MLP inversion mapping. 42 MLPs were trained with hidden layer sizes ranging from 10 to 70 units and a range of 6 different input context windows sizes from 1 to 20. Network MSE score calculated on the test set is given as a function of hidden layer and context window sizes. For ease of inspection, the values given in this table are plotted as an error surface in Figure 4.3.

the inversion mapping function is. The non-linear[2] activation functions at given units are key to the MLP's representational power; they act as basis functions from which the overall network mapping function can be composed by appropriate configurations of the MLP weights. The necessary number of component non-linearities depends on the complexity of the non-linear function to be modelled. In other words, the question stands whether 50 hidden units is sufficient for the MLP to perform this inversion mapping task as well as possible. It could be the case that 60 units are needed for example.

We could employ a "guess-and-test" approach to determining an answer to this question. Figure 4.3 and Table 4.3 present the results of such an approach. Different MLPs with a range of 7 hidden layer sizes and 6 acoustic context window sizes have been trained using the SCG optimisation algorithm. As is standard, the separate validation set was used to identify the point at which the MLP reached greatest performance and generalisation. In addition, a 2000 epoch upper limit was placed on the number of

[2]logistic, in this case.

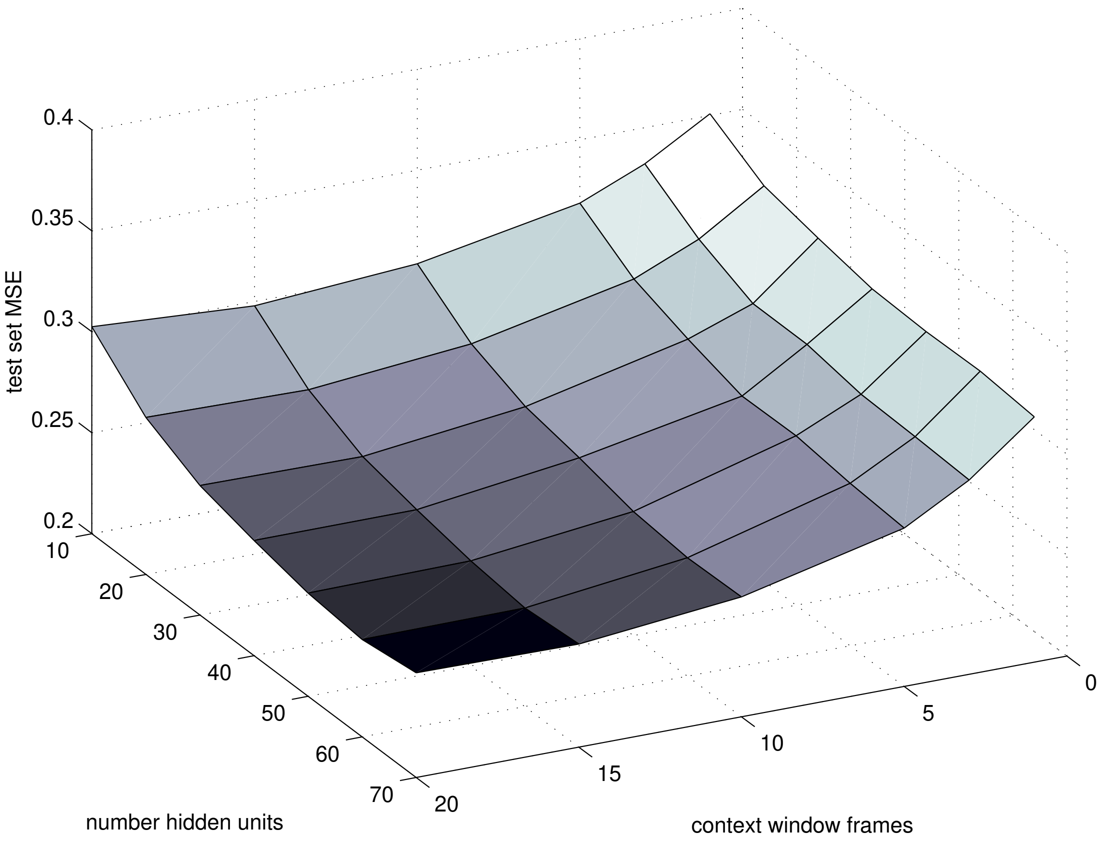


Figure 4.3: Effect of hidden layer and context window size on the performance of an MLP inversion mapping. This plot shows the error surface which characterises the relationship between the number of context frames used as input, the number of units contained in the hidden layer, and MSE calculated for the test set. We see clearly that there is an asymptotic relationship between MLP performance and both hidden layer size and context window size. For the reader's convenience, Table 4.3 gives the values used to plot this error surface.

training cycles for each MLP. Relative performance is presented in terms of the MSE calculated on the unseen test set for each MLP configuration.

Although the error surface is subject to a certain amount of noise, it is evident from Figure 4.3 that the relationship between both number of hidden units and context window size and MLP performance is asymptotic. For example, these results indicate that using any more than 20 context frames (which translates to a window of length 210ms) is unlikely to yield any increase in performance. Likewise, it would appear that using any more than 70 hidden units would also be superfluous.

In fact, it is arguable that somewhat less than 70 hidden units are needed. To arrive at a closer estimate of the necessary number of hidden units, we could increase the "resolution" of our guess-and-test search. In other words, we could train many more MLPs with hidden layers ranging in steps of one between 30 and 50 for example. Alternatively, we might also consider using one of the algorithms available which attempt to tailor the representational power of the MLP model to the task in hand automatically.

There are broadly speaking two classes of "meta" algorithms discernible within the neural network field which have an opposing philosophy for achieving this goal (Bishop 1995). On one hand are various constructivist neural networks, which begin training with a minimal size, and then incrementally grow in complexity as required. On the other hand, are pruning techniques, which attempt to identify and remove those trainable parameters from a neural network which appear to be redundant.

Under the umbrella of pruning algorithms are found two broad categories: penalty-term based and sensitivity (or salience) based algorithms (Reed 1993). For penalty term algorithms, the objective error function used to train the network is modified to include terms which encourage a parsimonious use of the trainable parameters. An example of this algorithm group would be Backpropagation with Weight Decay (see Hinton (1986) or Werbos (1988)), where an additional term is added to the weight update rule which reduces a weight's size by a user definable proportion of the the previous value of the weight. Unless a weight's value is actively reinforced by backpropagation learning, it will be driven towards zero, at which point it is effectively removed from the network.

Under sensitivity pruning algorithms, training and pruning are typically carried out alternately. A neural network of "suitable" size is taken and trained to a minimum in the error function. Then, the *saliency* of the network components is calculated. The component identified as least salient is removed, and then the network is retrained to compensate for the loss. This process can be repeated iteratively until performance falls below some threshold. These algorithms focus on the sensitivity of the error function to the removal of the various network component elements. Different algorithms consider different components (i.e. links and/or units) and use different criteria to judge saliency. For example, amongst the simplest pruning algorithms is Magnitude Based Pruning, where the link with the smallest weight is deemed to be the least salient and so is the prime candidate for removal. This would be an example of a link-based pruning algorithm. Optimal Brain Damage (LeCun, Denker, Solla, Howard & Jackel 1990) and Optimal Brain Surgery (Hassibi & Stork 1993) are two examples of more sophisticated algorithms intended for pruning links between units. Algorithms which prune units from a network include "noncontributing unit pruning" (Sietsma & Dow 1991), where units are removed whose output either does not vary, or is correlated with other units in the same layer, and Skeletonization (Mozer & Smolensky 1989).

The Skeletonization pruning algorithm is based upon the question "what happens to the performance of the network when a given unit is removed?" Thus, the straightforward measurement of relevance would be

$$\rho_i = E_{\text{without unit i}} - E_{\text{with unit i}} \tag{4.4}$$

which can be directly computed. For every unit, we could compare the difference between network error with and without the unit. The unit whose removal has the least harmful effect on network performance is the least significant, and is the unit which should be removed. However, the cost of computing this relevance measure ρ is $O(np)$ presentations of input-output training vector pairs, where p is the number of vector pairs in the training set and n is the number of units under consideration for pruning.

In place of explicitly calculating ρ_i as in Equation 4.4, Mozer & Smolensky (1989) sought a short cut in the form of an approximation to ρ_i. To this end, they introduced a coefficient α_i:

$$o_j = f(\sum_i w_{ji}\alpha_i o_i) \tag{4.5}$$

where o_j is the output of unit j, w_{ji} is the weight of link from unit i to unit j, and f is the activation function for unit j. Clearly, if $\alpha_i = 1$, the unit i will have a normal effect on following units. Conversely, if $\alpha_i = 0$, unit i will have no effect on subsequent units. In short, Equation 4.4 can be rewritten as

$$\rho_i = E_{\alpha_i=0} - E_{\alpha_i=1} \tag{4.6}$$

This ρ_i can be approximated using the derivative of the error with respect to α_i:

$$\hat{\rho}_i = -\frac{\partial E}{\partial \alpha_i} \tag{4.7}$$

This derivative may be computed for each unit by propagating error "signals" back through the network in a way similar to standard MLP training. Using this approximation, we can avoid having to make a complete pass through the training set for all the units eligible for pruning, which would be necessary under the scheme of Equation 4.4.

Pruning algorithms can provide a convenient shortcut for exploring the space of different topologies without training all the possible networks within the range in question. However, there are certain caveats to bear in mind when using them. First, the pruning algorithms lead the way to network topologies of reduced size while still maintaining performance. However, that level of performance depends on the minimum in the error function landscape within which the network was initially located. In other words, if after initial training the network only occupied a local minimum according to the error signal, as opposed to the global minimum, then pruning will obviously not necessarily lead to the network reaching this global minimum through training. Worse, there is perhaps the danger that more units may be pruned than are necessary to achieve the global minimum. This situation may arise where the pruning algorithm only retains enough units to maintain the lower level of performance at the local minimum.

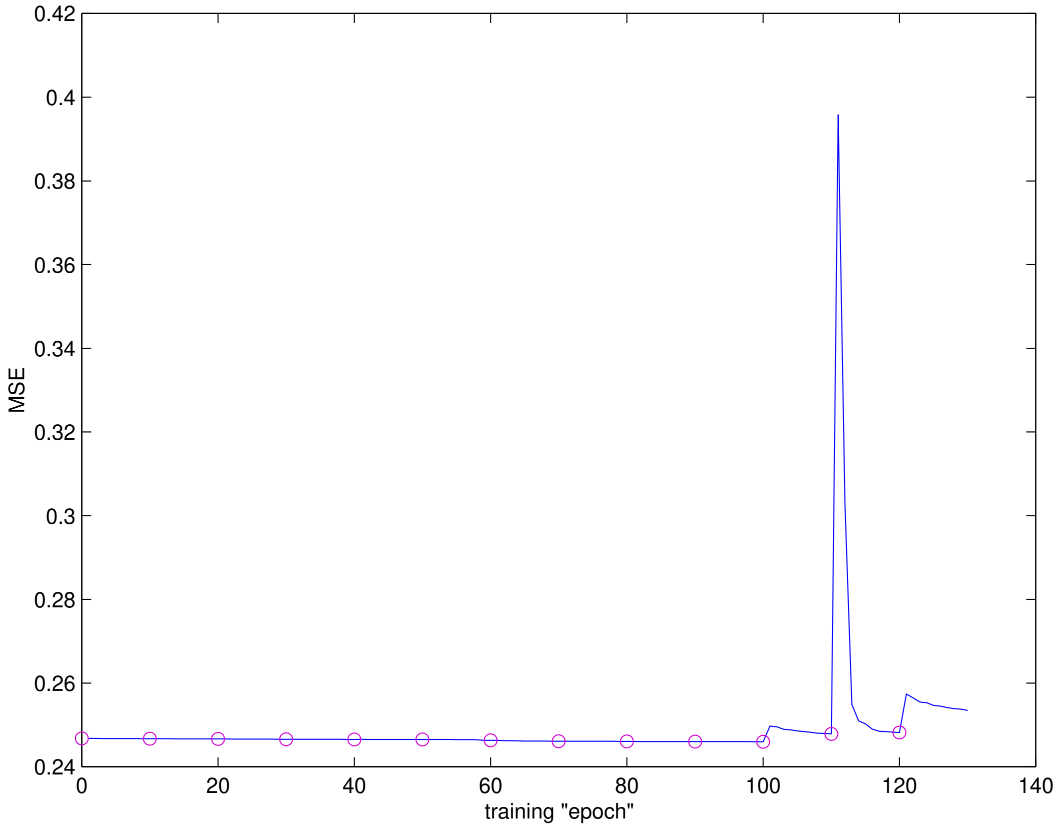


Figure 4.4: A log of the MSE calculated on the training set while pruning the feedforward MLP shown in Figure 4.2. The circles indicate the points at which a unit is removed from the hidden layer. The solid line shows the mean square error calculated in the training set while retraining the network for ten epochs after having removed each unit in turn. The pruning algorithm attempted to remove thirteen units. Following removal of each of the first twelve units, it was possible to retrain the network to within an acceptable error increase margin within 10 epochs (both parameters of the pruning algorithm). Following the removal of the thirteenth unit, the network demonstrated what was (arbitrarily) deemed to be too large a degradation in performance.

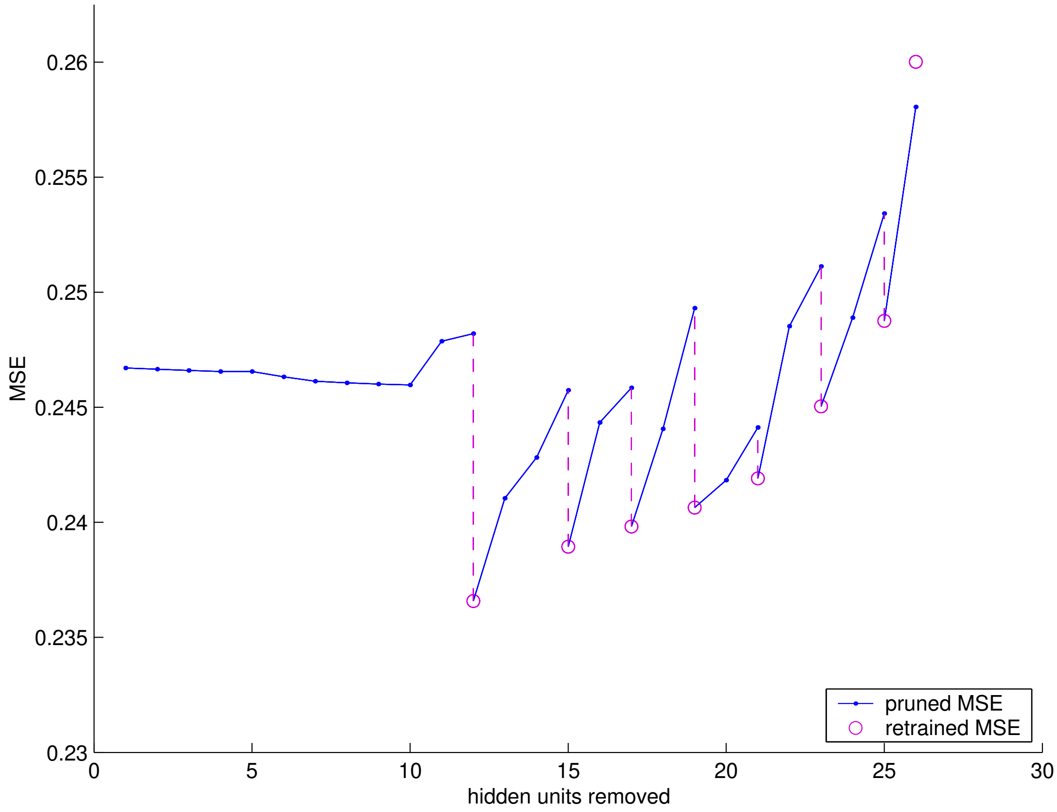


Figure 4.5: An overview of the MSE calculated on the training set while pruning the feedforward MLP shown in Figure 4.2. The solid sections of this plot represent a single pruning "run"; each solid point shows the MSE after having retrained the network for ten epochs following the pruning of one hidden unit at a time. Once the pruning algorithm reaches a point where removing a further unit would lead to unacceptable performance, according to the user defined criteria, the pruning process is terminated. At such a point, the network weights may be randomised and trained again from scratch, which is indicated by the broken line and circles.

4.4.1 Pruning the initial MLP

We can use the Skeletonization pruning algorithm to verify whether the MLP used in Section 4.3 contains either not enough or too many hidden units. If the MLP shown in Figure 4.2 contains too many hidden units, then presumably the pruning process would be able to identify redundant units and remove them. Conversely, we would suspect having selected too small an initial network topology if it were not possible to remove any units at all without increasing the network error on the training set.

Figures 4.4 and 4.5 demonstrate how the hidden units may be pruned from the feedforward MLP shown in Figure 4.2, and that a significant proportion of the network's hidden units may be superfluous to the requirements of modelling the acoustic-to-articulatory mapping function.

Figure 4.4 summarises an application of the Skeletonization algorithm to prune hidden units from the fully trained network. After removing each unit, the algorithm was set to retrain the network for a maximum of 10 epochs[3] using the SCG optimisation algorithm. If after retraining the increase in network mean square error (MSE) compared with that prior to beginning the pruning process on the training set was (still) greater than a user defined threshold, then pruning was stopped and the last unit that had been removed was replaced.

From Figure 4.4 it is clear that it was possible to remove ten units from the hidden layer of the network without causing an increase in error. In fact, the mean square error on the training set decreased marginally from 0.24679 to 0.24597 throughout the process of pruning and retraining. This would seem to indicate that these ten units are entirely superfluous in terms of the representational power required to model the acoustic-to-articulatory mapping (in this case).

Following the removal of each of the next three hidden units, the network demonstrated more substantial jumps in MSE on the training set. Ten epochs of retraining using the SCG optimisation algorithm was sufficient to reduce the error to within the threshold of allowable increase. However, the removal of the thirteenth unit produced

[3] it is of course possible to specify stopping criteria for the retraining phase, but in the present case these are not necessary

a deficit in performance which was not recompensed within the specified ten epochs of retraining, and pruning was ceased.

At this stage, we might tentatively suggest that no more than 38 hidden units are necessary in an MLP for performing the inversion mapping (at least in this case). However, it would arguably be more accurate to say that no more than 38 units are necessary for *this* minimum in the error landscape. It is possible that the minimum to which the network was originally trained prior to pruning is only a local minimum, and moreover, one which requires a certain number of units. Retraining following the removal of each of the hidden units may not have been vigorous enough for the network weight configuration to clear this local minimum and find other minima, which could potentially require less hidden units.

To test this possibility, we can take the pruned network, randomise the weights and then train again from scratch. If it appears possible to train the network down to a level which either matches or exceeds the performance of the network weight configuration which resulted from the pruning stage, then we might continue to attempt to subject this network to further pruning.

Figure 4.5 gives an account of iteratively pruning and then retraining the network from scratch. The plot shows the MSE of the network calculated on the training set throughout the process. The points on the solid lines indicate the MSE after retraining for ten epochs following the removal of each hidden unit. The dashed lines represent the points at which the network has been completely retrained from a randomised weight configuration, with the circles indicating the MSE reached by this training phase. Thus, the separate sections of the solid line represent separate pruning “runs”.

The first twelve points on this graph are the same as shown in Figure 4.4. However, following the removal of the twelfth hidden unit, the network is completely retrained from scratch to try and reach a new minimum, from which it *is* possible to prune further units. As the figure shows (first dashed line and circle point), it was indeed possible to retrain this pruned network to a new minimum.

This network was then subjected to further pruning and retraining, to make a total of eight cycles. Throughout this process, the MSE achieved on the training set by the network trained from scratch following pruning became progressively larger, until, at the eighth iteration of this process, removing a further hidden unit resulted in a substantial increase in MSE, which proved extremely difficult to recover. However, to this reach this point, twenty six of the original fifty hidden units were removed.

Looking at the MSE calculated on the training set can only tell part of the story. To get a more unbiased impression of the generalisation capability of the network, it is necessary to consider the MSE calculated on the separate validation set, which is shown in Figure 4.6.

Looking at Figures 4.5 and 4.6, it is apparent that the optimum number of hidden units for inferring articulation from acoustics for this data set is 38. However, it is interesting to note how removing further units from the network during pruning does not result in a sudden point of 'catastrophic failure'. This is a well known property of neural network models, and is termed "graceful degradation". A trained neural network that is deliberately corrupted in some way (by the addition of noise to certain weights, the removal of links etc.), will not generally undergo a sudden failure; performance is likely to suffer, but not in an "all-or-nothing" manner, which can happen with other models.

4.4.2 Results

In Section 4.4.1, it was found that an MLP with a hidden layer pruned down to just 24 units was able to perform the inversion mapping at a level of performance not drastically worse than the best performance achieved on the training and validation sets. However, the best performance observed was for a network containing 38 units in the hidden layer. Therefore, it is the results of this network on the test set which are presented here.

4.4.3 Quantitative results

Table 4.4 presents the results for the best MLP identified during the pruning process, containing 38 hidden units, when estimating articulation for an unseen test set containing 46 utterances. The average RMS error for this network is 1.62mm. The average RMS error of the MLP with 50 hidden units trained and tested using the same data, given in Table 4.2, was 1.64mm. On the basis of this result, we argue that an MLP with

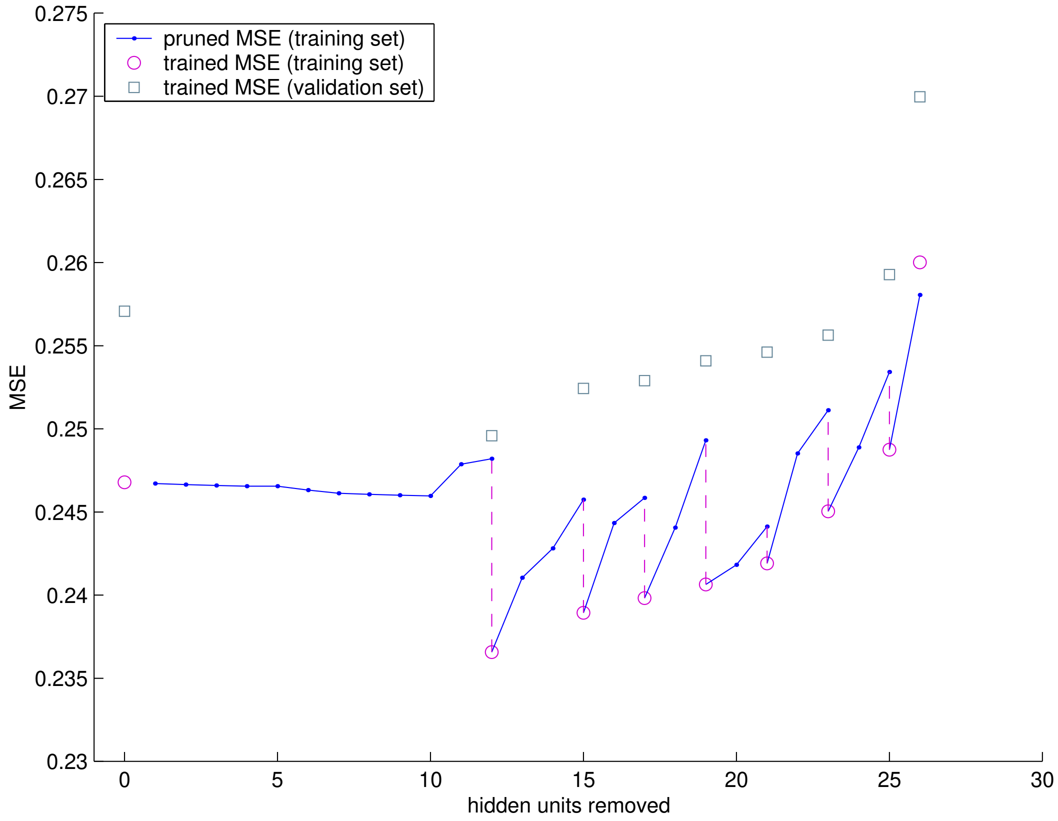


Figure 4.6: As in Figure 4.5, this plot gives an overview of the MSE calculated on the training set while pruning the feedforward MLP. The solid sections of this plot represent a single pruning "run"; each solid point shows the MSE after having retrained the network for ten epochs following the pruning of one hidden unit at a time. Once the pruning algorithm reaches the point where removing further units would lead to unacceptable performance, the pruning process is terminated. The network weights may then be randomised and trained again from scratch, which is indicated by the broken line and circles. This plot also shows the MSE calculated on the *validation* set throughout the pruning/retraining process.

Articulator	RMS error	RMS error(mm)	Correlation
upper lip x	0.177	0.99	0.584
upper lip y	0.143	1.16	0.720
lower lip x	0.159	1.21	0.605
lower lip y	0.137	2.73	0.749
lower incisor x	0.162	0.89	0.562
lower incisor y	0.119	1.19	0.796
tongue tip x	0.126	2.43	0.794
tongue tip y	0.115	2.56	0.840
tongue body x	0.122	2.19	0.810
tongue body y	0.111	2.14	0.832
tongue dorsum x	0.126	2.04	0.792
tongue dorsum y	0.135	2.31	0.711
velum x	0.128	0.42	0.786
velum y	0.129	0.41	0.767

Table 4.4: Performance of pruned MLP with 38 hidden units when recovering articulation from acoustics for the unseen test set. The average of the RMS error values given is 1.62mm. These measures are calculated using the raw network output. Compare these with those shown in Table 4.6, where the network output is first lowpass filtered using channel specific cutoff frequencies.

38 hidden units has sufficient representational power to perform the inversion mapping. Therefore, we further argue that it represents a reasonable baseline against which to compare inversion mapping systems later on in this thesis (assuming the same data sets are used).

4.4.4 Qualitative results

RMS error and correlation scores, such as those in Table 4.4, although limited in several respects, provide some means to objectively and quantitatively compare the performance of two different networks, or indeed different inversion techniques described previously in the literature. However, no doubt the most humanly accessible impression of how the MLP is performing the inversion mapping is provided by visual comparison of the MLP output with the real articulatory trajectories.

Figures 4.7 and 4.8 present the results observed for two of the utterances present in the test set. These are presented in the network output range (i.e. approx. [0.0, 1.0]). Silence at the beginning and end of the files has been omitted. As evident in these figures, the MLP does reasonably well at estimating the positions of some articulators at some times, but less well at other times and for other articulators.

4.4.5 Smoothed MLP output

The physical nature of speech articulators means they are constrained to move relatively slowly and smoothly. However, while continuity is a basic principle of human speech production, an MLP trained with the sum-of-squares error function will not necessarily emulate this. The aim of training with the sum-of-squares error function is to minimise the distance of the MLP output from the target output for each input-output training pattern. In other words, the optimal "instantaneous" mapping that the MLP can provide at a given time frame is the conditional average of the articulator configurations for *all* such input vectors. However, this optimum does not in any way stipulate that the output articulatory configuration at one time frame should depend on the articulatory configuration at any previous or following time frames. Therefore, we rely heavily on the inversion mapping function to yield something resembling articulatory trajectories in the MLP output. Specifically, we make the assumption that acoustic input vectors at time $t-1$ and time $t+1$ map to points in the near vicinity of the articulatory configuration at time t, such that no discontinuities result in the overall sequence of MLP output. This assumption is reasonable, but in practice is liable to be confounded by one-to-many mappings from the acoustic to the articulatory domain.

One way we have sought to decrease discontinuity between adjacent output patterns in the MLPs presented here is by including a context window of acoustic vectors at neighbouring time frames. Up to a point, the output taken directly from the MLP does resemble reasonably continuous trajectories. Nevertheless, comparison of sequences of MLP output and real EMA data, such as in Figures 4.7 and 4.8, reveals that the MLP output is generally not as smooth as genuine articulatory trajectories.

In an effort to make up for inadequacies in the MLP estimated articulatory trajectories, we might turn to various postprocessing techniques. For example, we know that the articulators move relatively slowly. Therefore, one simple example of an articulatory constraint we can impose as a postprocessing step, which utilises this knowledge, is to lowpass filter the MLP output.

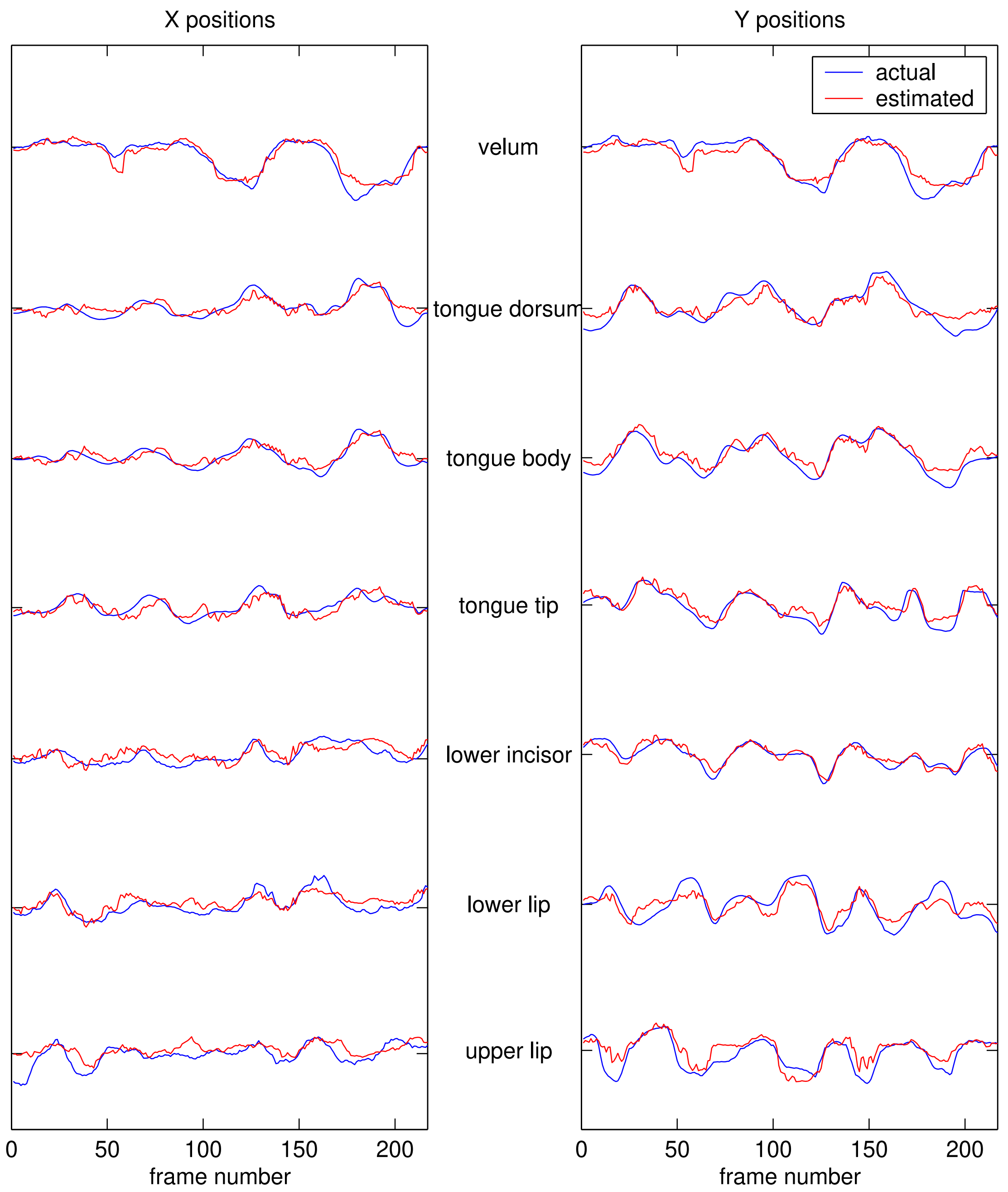


Figure 4.7: A Comparison of the MLP estimated articulatory trajectories with the EMA-measured articulatory trajectories for the unseen test utterance "The speech symposium might begin on Monday."

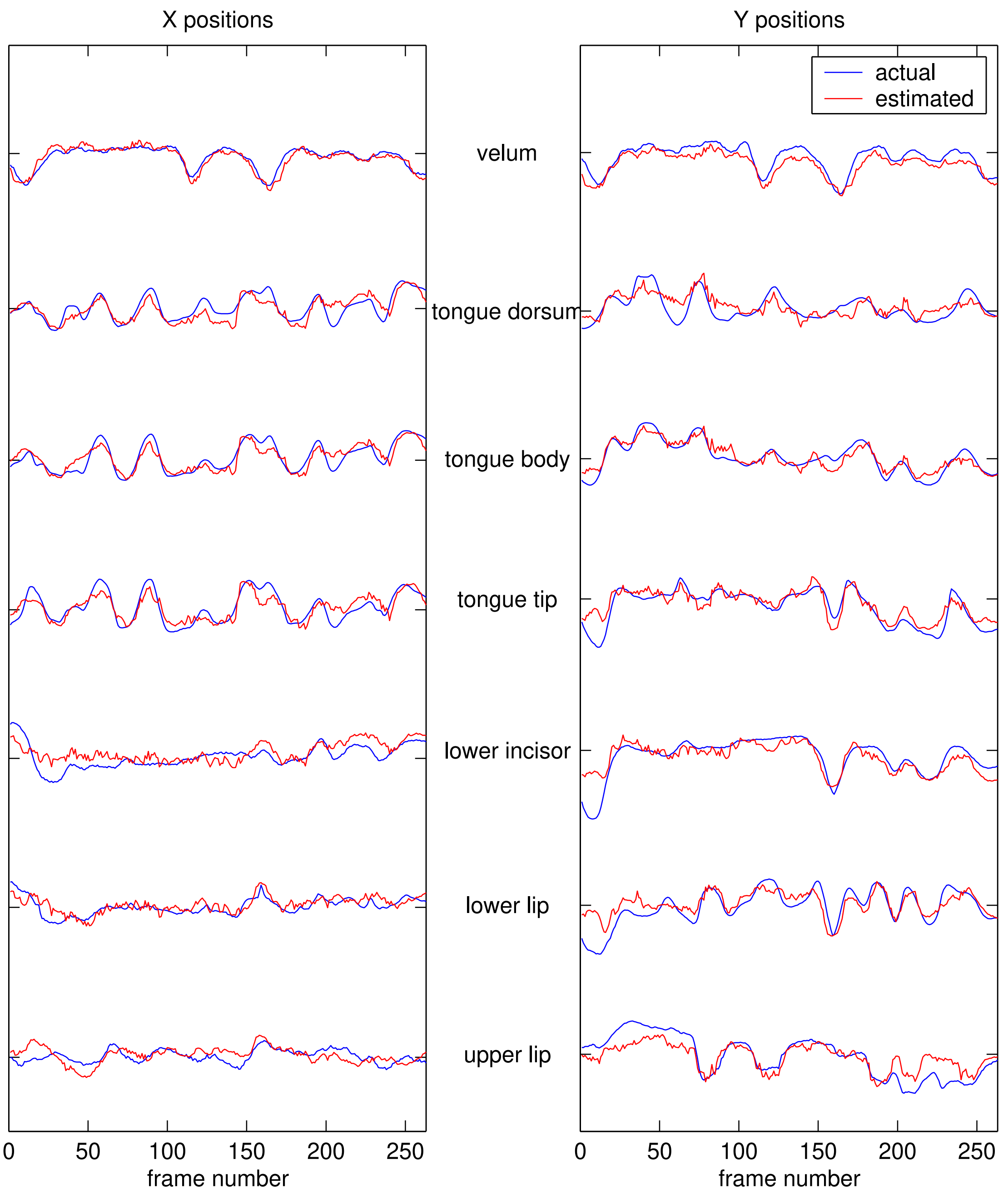


Figure 4.8: A Comparison of the MLP estimated articulatory trajectories with the EMA-measured articulatory trajectories for the unseen test utterance "The high security prison was surrounded by barbed wire."

Lowpass filtering of the MLP output requires that we choose a cutoff frequency. The bandwidth of human articulator movements has been variously claimed in the literature to lie somewhere between 10Hz and 20Hz (for example, Perkell et al. (1992) cite the value of 15Hz). The movements resulting entirely from muscular effort are likely to be below 10Hz, while collision of the articulators against fixed passive articulators, such as the hard palate and teeth, may introduce some higher frequency components[4]. Therefore, we could choose to lowpass filter with a cutoff set on the basis of such estimates. Hogden et al. (1996), for example, used a 20 Hz low-pass filter. However, it is not clear whether all articulator channels should in fact be lowpass filtered with exactly the same cutoff frequency.

As an alternative to using such estimates of articulator bandwidths, we could instead take a more empirical approach. In short, we could take the MLP output for the utterances in the validation set and lowpass filter all the channels separately with a range of different cutoff frequencies. The optimal lowpass filtering cutoff frequency is the one which gives the best RMS error and correlation against the real target articulator trajectories.

Figure 4.9 presents the results of applying this analysis to the validation set for the lower incisor x- and y-coordinate channels. These plots show RMS error and correlation as a function of lowpass filtering cutoff frequency in the range 2-49Hz at 1Hz intervals. A second order Butterworth filter was used to perform the lowpass filtering. Since the frame shift of the processed EMA data and neural network output is 10ms, the 50Hz point is in fact the result obtained by not filtering the MLP output. The optimal cutoff frequencies for these two channels have been chosen as the best compromise between high correlation and low RMS error, and are indicated on these plots by the vertical dotted lines. Figures C.1 to C.7 in Appendix C present similar graphs for all 14 articulatory channels. Meanwhile, the channel-specific lowpass filtering cutoff frequencies chosen as a result of this process are summarised in Table 4.5.

As in Figures 4.7 and 4.8, Figures 4.10 and 4.11 provide a visual comparison of the MLP output with the corresponding real articulatory trajectories. This time, however, the MLP output has been lowpass filtered, using the channel-specific cutoff frequencies

[4]personal communication, Alan Wrench, Queen Margaret University College, Edinburgh.

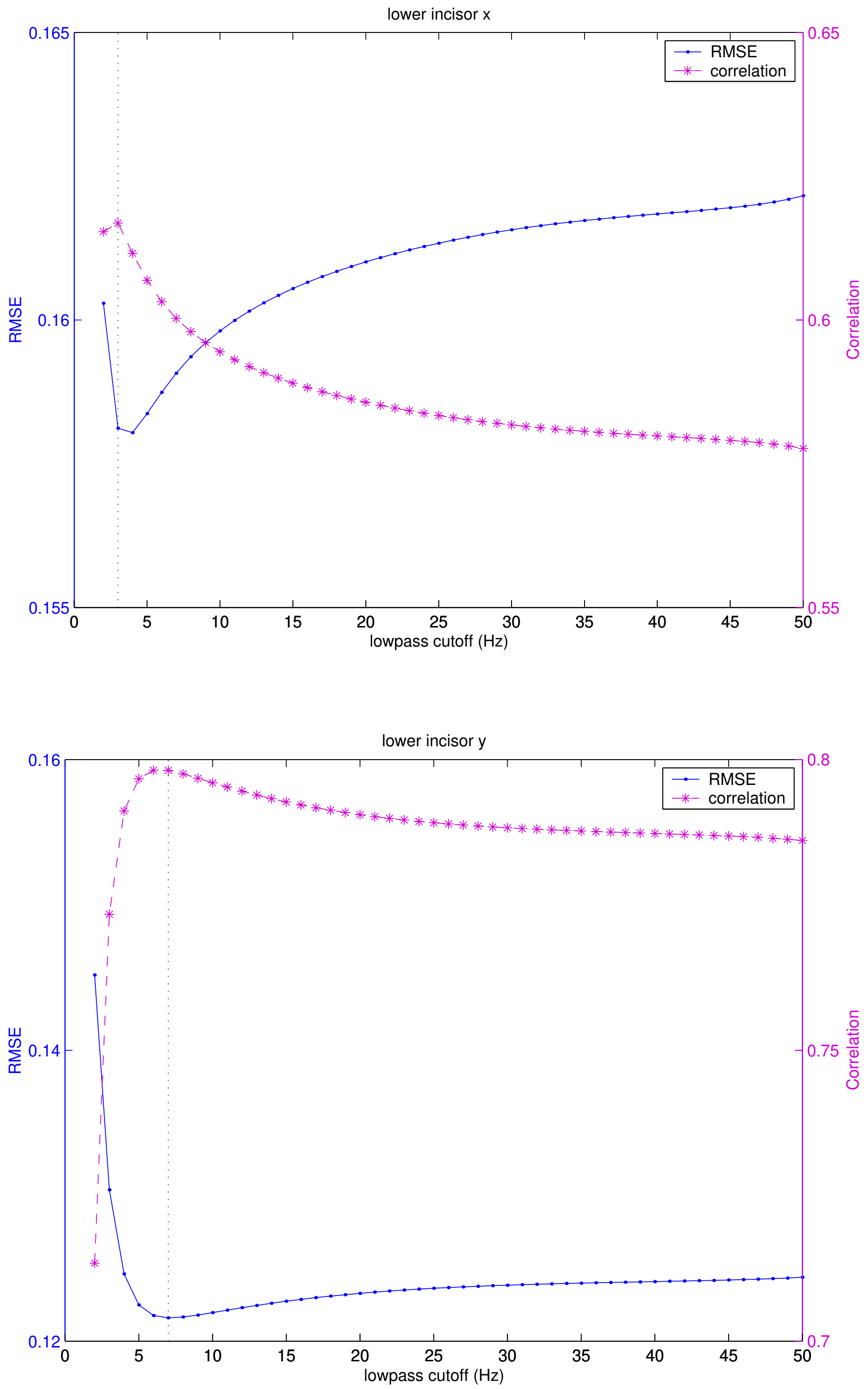


Figure 4.9: Plots of RMS error and correlation as a function of cutoff frequency when lowpass filtering the lower incisor x- and y-coordinate channels of the MLP output. The cutoff frequency chosen as optimal is indicated as the non-continuous vertical line.

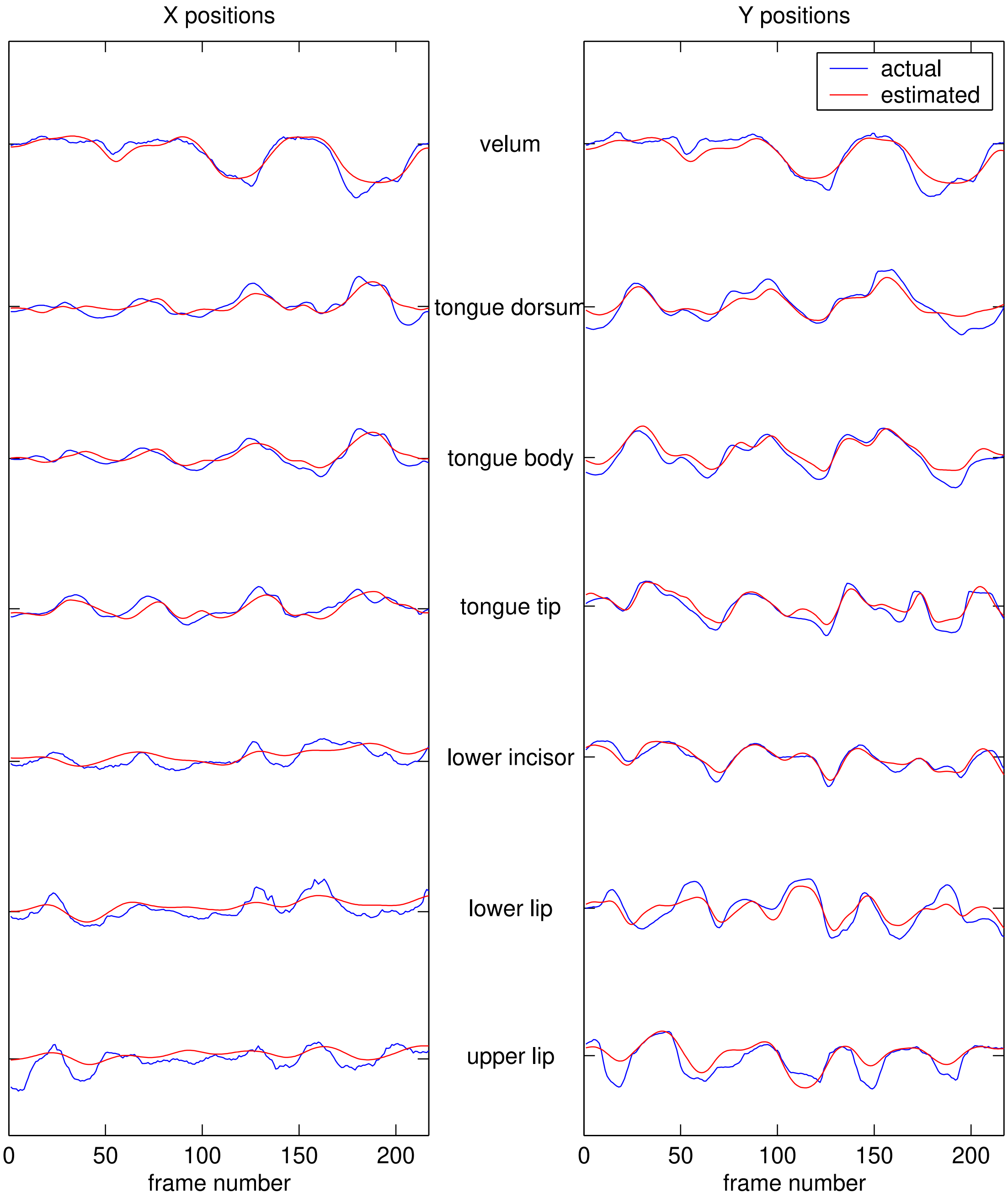


Figure 4.10: A Comparison of the MLP estimated articulatory trajectories with the EMA-measured articulatory trajectories for the unseen test utterance "The speech symposium might begin on Monday." In this case, the MLP output has been postprocessed by lowpass filtering with channel specific cutoff frequencies.

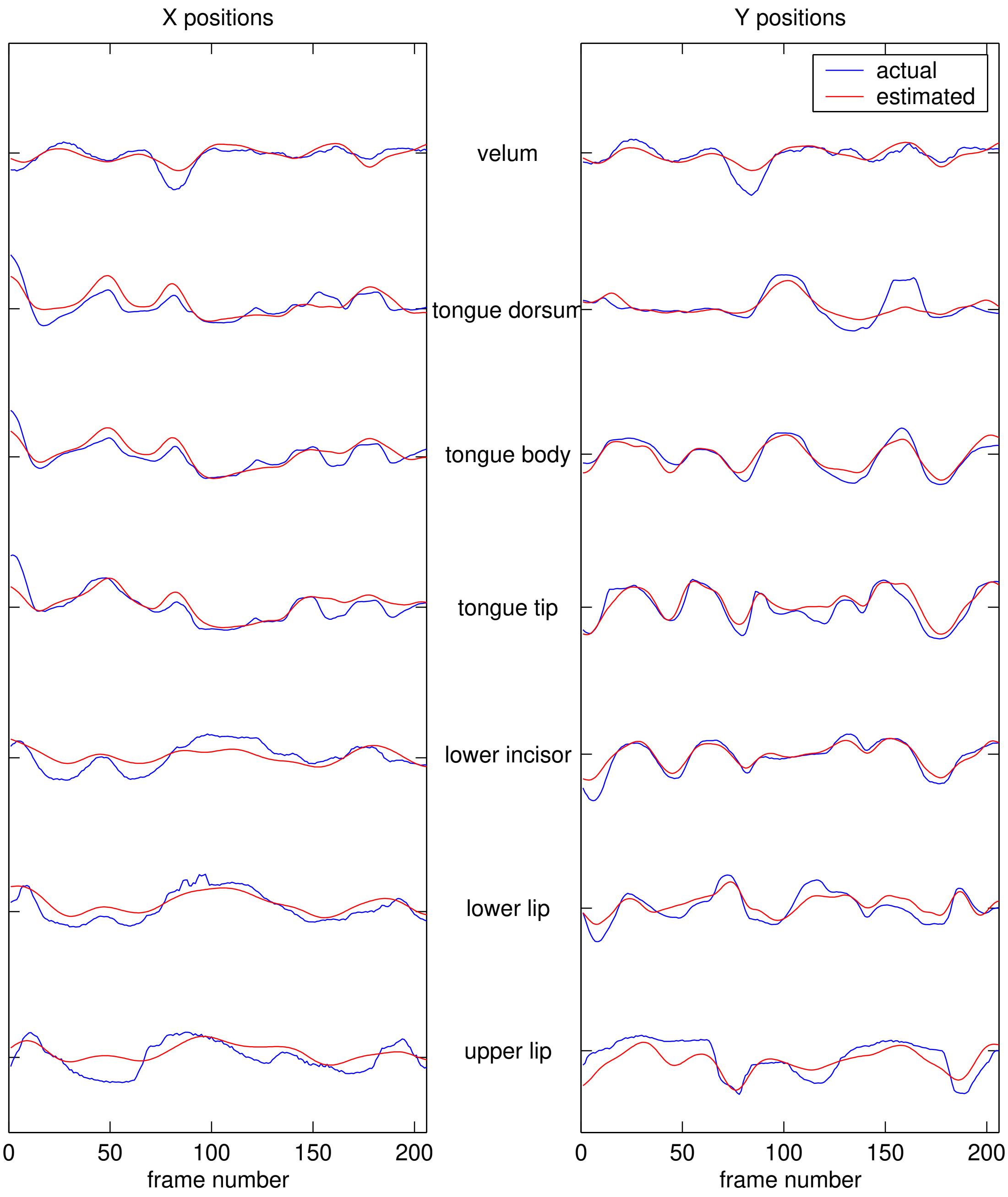


Figure 4.11: A Comparison of the MLP estimated articulatory trajectories with the EMA-measured articulatory trajectories for the unseen test utterance “Why charge money for such garbage?” In this case, the MLP output has been postprocessed by lowpass filtering with channel specific cutoff frequencies.

Articulator	Channel	LP cutoff (Hz)
upper lip	x	3
	y	5
lower lip	x	3
	y	8
lower incisor	x	3
	y	7
tongue tip	x	6
	y	9
tongue body	x	6
	y	7
tongue dorsum	x	7
	y	6
velum	x	5
	y	5

Table 4.5: Channel specific cutoff frequencies used for lowpass filtering the MLP output articulatory trajectories. These values were derived empirically by lowpass filtering MLP output for the validation data set in the range from 2Hz to 50Hz (i.e. unfiltered), and comparing the RMS error and correlation scores obtained when measured against the actual articulatory trajectories (unfiltered). This process and the results are presented in more depth in Appendix C.

given in Table 4.5. The utterance shown in Figure 4.10 is the same as that shown unfiltered in Figure 4.7 in order to provide direct comparison between the filtered and unfiltered MLP output.

As in Table 4.4, Table 4.6 presents the RMS error, RMS error expressed in millimetres and correlation scores for the output of the best MLP when compared with the actual articulation for an unseen test set containing 46 utterances. However, for Table 4.6, the articulatory trajectories output by the neural network have first been lowpass filtered using a 2nd order Butterworth filter with the channel-specific cutoff frequencies given in Table 4.5. Comparing the values in the two tables cell-for-cell, we find that filtering has the beneficial effect of decreasing the RMS error somewhat as well as increasing the correlation measure. The average of the fourteen channel RMS error values for the unfiltered network output is 1.62mm, compared with an average of 1.57mm for the filtered output. The average RMS error for the tongue coils is 2.2mm.

Articulator	RMS error	RMS error (mm)	Correlation
upper lip x	0.173	0.98	0.611
upper lip y	0.138	1.12	0.744
lower lip x	0.155	1.18	0.641
lower lip y	0.135	2.68	0.759
lower incisor x	0.159	0.87	0.590
lower incisor y	0.116	1.17	0.805
tongue tip x	0.122	2.35	0.811
tongue tip y	0.111	2.49	0.852
tongue body x	0.120	2.14	0.821
tongue body y	0.106	2.05	0.848
tongue dorsum x	0.123	2.00	0.804
tongue dorsum y	0.129	2.20	0.741
velum x	0.124	0.41	0.805
velum y	0.125	0.40	0.786

Table 4.6: Performance of MLP with 38 hidden units when recovering articulation from acoustics for the unseen test set, given as RMS error and correlation for each articulatory channel. The average of the RMS error values given is 1.57mm. These measures were calculated after the raw network output had been lowpass filtered using channel specific lowpass cutoff frequencies. Compare these with those shown in Table 4.4, where the network output is used directly without filtering.

4.5 Comparison of results

It is interesting to compare the results presented in this Chapter to those reported previously in the literature. Taking the results given in either Table 4.6 or Table 4.4 for comparison, we find that the performance of the MLP inversion mapping compares favourably with inversion methods previously reported. As noted in Chapter 2, Hogden et al. (1996) reported an error around 2mm for the tongue points; Dusan (2000) reported average RMS error of around 2mm; Okadome et al. (2000) reported RMS error between estimated and actual articulatory trajectories of about 1.8mm on average.

There are certain obstacles which make direct comparison with the results of previous studies difficult. Different studies have used different articulators. As we have seen in the results in this chapter, the accuracy of inferred trajectories depends on the articulator in question. In addition, it is conceivable that differing vocal tract sizes, resulting in different ranges of articulator movement, could exert an effect on RMS errors expressed in millimetres. Another difference which complicates direct comparison is that many previous studies have used only limited quantities of simplistic data. Finally, it must be noted that researchers have not generally made it explicitly clear how exactly they calculate their measures. For example, whereas we have opted here to calculate RMS error separately for each articulatory channel, we could just as easily have chosen our RMS error in terms of the *Euclidean distance* between estimated and actual articulator positions. In other words, at each time frame we could have taken the x- and y-coordinates for one articulator as estimated by the network, and calculated the distance of this point in the midsagittal plane from the actual location of the articulator.

Notwithstanding these difficulties, we assert that the RMS errors of the MLP are of the same order as those of inversion mapping systems previously described in the literature. This fact is taken as further evidence, on top of that provided by the pruning process, to justify our belief that the MLP inversion mapping provides a reasonable baseline against which to compare alternative models later on in this thesis.

4.6 Input pruning

As an interesting exercise, we can apply the Skeletonization pruning algorithm as described in Section 4.4 to the layer of input units. Figure 4.12 shows the results typically observed. This plot shows which of the initial 400 input units remained after the pruning process. The presence of an input unit is indicated by a circle. The total number of input units remaining is 133. With this reduced number of input units, the network performed only 2.2% worse than the MLP prior to pruning in terms of MSE for the training set. At the stage where 137 input units remained, the performance deficit was merely 0.8%. The histograms in Figure 4.13 were produced by summing the numbers of units over either the rows or columns of Figure 4.12.

We might hope that pruning the input layer this way could provide a way of ascertaining which features are important in the acoustic input window. However, it is arguably not a reliable indicator of this. From a practical point of view, the pattern of remaining units simply does not provide any definitive indication of which frequency ranges are most important, or which time frames in the context window are most pertinent. Instead, the units which have been pruned were spread throughout the input window. In addition, there are theoretical objections which discourage the interpre-

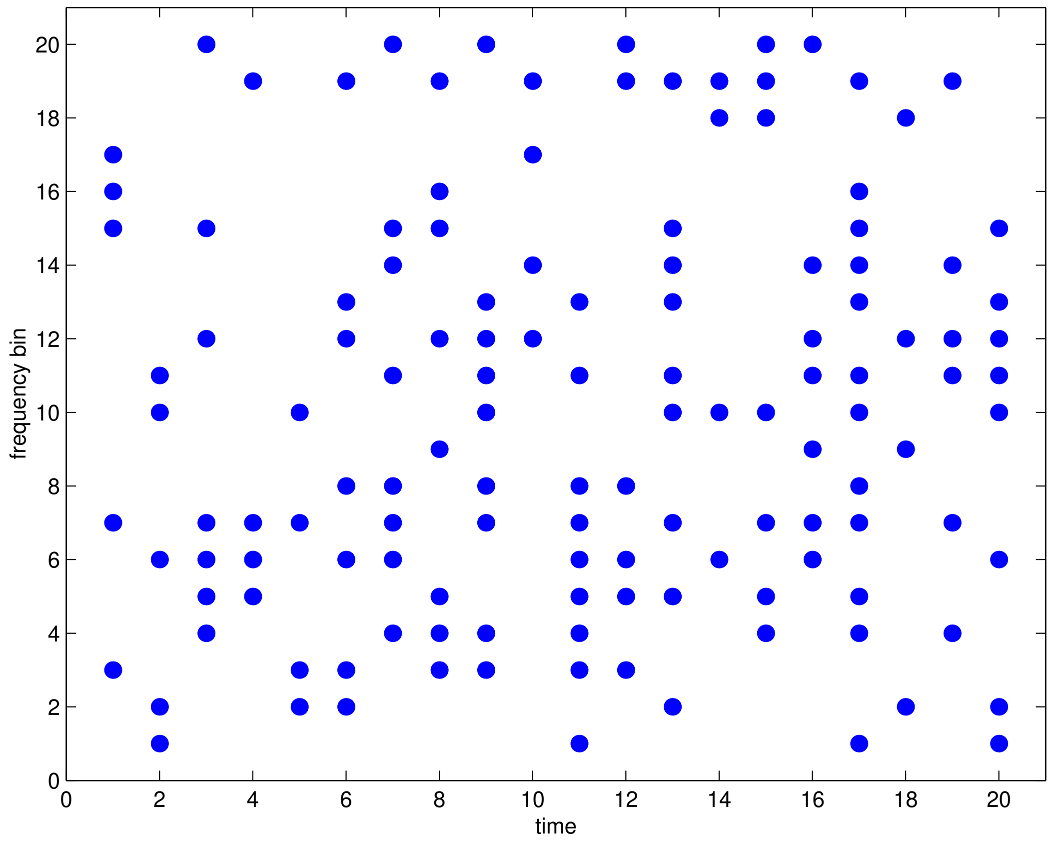


Figure 4.12: The result of pruning input units from the network shown in Figure 4.2, using the same method used to prune the hidden layer units. Of the initial 400 units contained in the input layer, 267 could be removed without resulting in a marked performance deficit. This figure shows the configuration of the remaining input units, with respect to temporal position in the context window (x axis) and filterbank frequency bin (y axis). Both time and the filterbank frequency bins increase along with their respective network unit indices from 1 to 20. The presence of an input unit is indicated by a circle.

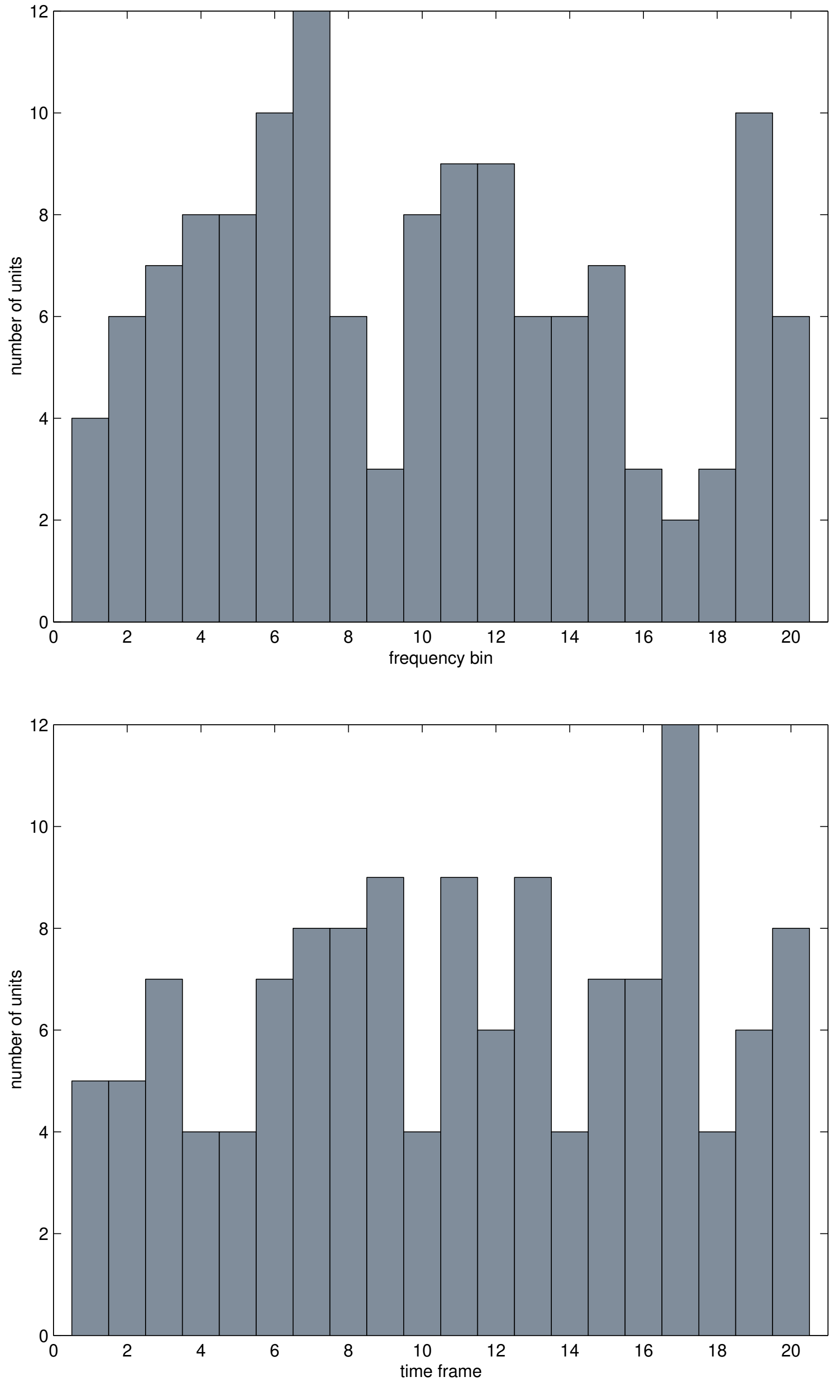


Figure 4.13: Histograms showing the "density" of input units remaining following pruning as a function of frequency bin and position in the context window. For example, there are 8 units in the tenth frequency bin located across all 20 context window time frames.

tation of the pattern of remaining units as an indication of which input features are most salient. The fact that we can remove input units without a loss of performance indicates redundancy in the vector of acoustic input coefficients. However, where two units provide redundant data, *either* could arbitrarily be removed, although we would probably not want to ascribe higher significance to the remaining unit than the one removed. Perhaps the most we can claim is that pruning has demonstrated the *scale* of the redundancy in providing large context windows of acoustic input. In other words, it was possible to remove 67% of the input units without significantly decreasing the accuracy of the network.

4.7 Discussion

For the experiment in this chapter, the speech of only a single speaker has been utilised. There are several additional challenges inherent in multi-speaker inversion. The question of how to normalise the different anatomies and coil placements between two or more speakers rises to the fore.

Studies like Johnson, Ladefoged & Lindau (1993), who compared the articulation of five speakers using XRMB cinematography, do not give cause for optimism that inter-speaker normalisation will be straightforward. The interspeaker variation in articulation patterns led them to conclude that at some level speech production tasks are specified in terms of acoustics rather than spatio-temporal gestures. This would indicate further aggravation of the one-to-many problem in the inversion mapping; speakers with different shape vocal tracts are in addition using different articulation strategies to produce very similar acoustic output.

By working separately with the data from one speaker at a time, such considerations are by and large avoided. Hence, the complications that come with speaker-independent acoustic-to-articulatory inversion are recognised as being beyond the remit of this thesis, and will not be considered further. It is argued this is not an unreasonable comprise to make for the time being. In fact, using data generated via a vocal tract model is effectively the same as working with a single speaker.

4.7.1 Recurrency

In working with both the MLPs in this chapter and the linear mappings in Appendix A, we have found that using a context window of multiple acoustic input frames improves results. Another general method for including the notion of context in an MLP is to augment the structure with recurrent links, where the state of given units at time t is fed as inputs to units at time $t+1$. In this way, the network no longer performs an instantaneous mapping from the input to the output domain which is independent of where the input comes in a sequence of input vectors. In a recurrent neural network, the output given in response to an input vector at time t may also depend in part on the input vectors at previous time frames.

In principle, this characteristic could prove useful when attempting to perform the inversion mapping. Human articulators are incapable of moving from one time frame to the next with more than a certain velocity and acceleration. It might be hoped that recurrent neural networks could model such physical continuity constraints to some degree.

In practice, however, preliminary investigation of various recurrent nets has failed to uncover any clear advantage over the use of straightforward MLPs. If fact, we have not been able so far to train a recurrent network which performs better than the MLPs presented in this chapter. In addition, we have generally found it more time consuming to train recurrent neural networks, in terms of requiring more training cycles and cpu time overall.

Table 4.7 gives the results of an Elman recurrent network which has been trained and tested on the same data sets used throughout this thesis. This network consisted of 20 units in the input layer (for one frame of acoustic coefficients), 38 units in the hidden layer, and 14 output units. In this Elman network, the activation of the hidden units was fed back via recurrent links (with a delay of 1 time step) to serve as additional input to the same layer of hidden units. If we compare the results in Table 4.7 to those in Tables 4.4 and 4.3, we see that, although this Elman network performed better overall than MLPs with only a small number of context input frames (i.e. up to around 3), it did not perform as well as MLPs with larger context windows.

Articulator	RMS error	RMS error (mm)	Correlation
upper lip x	0.185	1.04	0.522
upper lip y	0.158	1.28	0.636
lower lip x	0.169	1.29	0.551
lower lip y	0.140	2.79	0.738
lower incisor x	0.168	0.92	0.548
lower incisor y	0.129	1.30	0.754
tongue tip x	0.143	2.76	0.729
tongue tip y	0.130	2.91	0.783
tongue body x	0.134	2.39	0.770
tongue body y	0.126	2.45	0.775
tongue dorsum x	0.138	2.23	0.747
tongue dorsum y	0.149	2.54	0.653
velum x	0.141	0.46	0.753
velum y	0.136	0.44	0.736

Table 4.7: RMS error and correlation for an Elman RNN with 1 acoustic input frame, calculated for the unseen test set. Comparing these results with those in Table 4.4, we find the Elman network has not performed as well as the MLP with a context window of 20 frames. The average of the RMS error values given is 1.77mm. Overall MSE was 0.30.

For the data sets used throughout this thesis, the target articulatory vector is (approximately) centred in time within the acoustic context input window. This means that when attempting to estimate the location of the articulators at time t, the MLP has the benefit of information from preceding and following frames. In contrast, the recurrent neural network does not have the advantage of access to future time frames. This could be one reason to explain why the recurrent networks do not perform as well as the MLPs in the experiments we have conducted. Another reason could be that the length of time dependencies between acoustic input frames may be too long for the recurrent network to exploit effectively. In an effort to explore the use of recurrency further, we have trained a similar Elman network containing 38 hidden units, but with an input context window of 20 frames. The results of this network are presented in Table 4.8. Unfortunately, this network too was unable to outperform the MLP with 20 context frames. Further investigation will be required in future to fully determine and understand why this might be the case, and ultimately whether the use of recurrent networks is desirable or not.

Articulator	RMS error	RMS error (mm)	Correlation
upper lip x	0.176	0.99	0.593
upper lip y	0.150	1.21	0.688
lower lip x	0.160	1.22	0.620
lower lip y	0.137	2.73	0.754
lower incisor x	0.165	0.90	0.568
lower incisor y	0.119	1.19	0.798
tongue tip x	0.129	2.47	0.786
tongue tip y	0.117	2.61	0.832
tongue body x	0.122	2.18	0.811
tongue body y	0.113	2.18	0.825
tongue dorsum x	0.128	2.07	0.788
tongue dorsum y	0.138	2.35	0.700
velum x	0.126	0.41	0.797
velum y	0.129	0.41	0.768

Table 4.8: RMS error and correlation for an Elman RNN with 20 acoustic input context frames, calculated for the unseen test set. Comparing these results with those in Tables 4.7 and 4.4, we find this Elman network has performed better than the Elman network with a single input frame, but still not as well as the MLP with a context window of 20 frames. The average of the RMS error values given is 1.64mm. Overall MSE was 0.26

4.7.2 Classification experiment

In terms of RMS error, the results of the MLP inversion mapping presented in this chapter compare reasonably well with previous systems reported in the literature. However, as noted in Section 2.4.2, RMS error and correlation provide only a limited measure of inversion mapping performance. It would be more informative to assess how well the articulatory trajectories have been inferred as part of a concrete application.

In Chapter 1, we mentioned that research is currently being conducted at CSTR into Automatic Speech Recognition systems using Linear Dynamic Models (LDM) (Frankel & King 2001*a*), (Frankel & King 2001*b*) and (Frankel et al. 2000). One phone-specific LDM is trained on the all instances of each phone in the training set (70% of the MOCHA TIMIT sentences for a speaker). Various combinations of acoustic and articulatory features have been used as the observed space. Acoustics-only LDMs have been trained on mel-frequency cepstral coefficients (MFCC). Articulatory-only LDMs have been trained on articulatory trajectories recorded by EMA. Meanwhile, mixed LDMs have been trained on the combination of these two feature sets. As a precursor to performing full speech recognition, phone classification experiments have been conducted, using the phones in the remaining 30% of the data set.

The phone classification tests using LDMs represent a suitable "real-world" task for assessing how well the MLP is able to infer articulatory trajectories from the acoustic signal. It is straightforward to substitute the articulation inferred by the MLP for the real articulatory trajectories in the LDM training and testing, and compare the results between the two. The LDM system trained on real EMA data alone achieved a phone classification accuracy of 57%. This compares with an accuracy of 55% when training and testing with articulatory trajectories inferred by the best MLP presented in this chapter. It would seem from this result that the MLP is indeed doing reasonably well.

However, the LDM systems trained with the combination of real EMA and MFCC features demonstrated a classification accuracy of 76%. The equivalent score when using inferred articulatory trajectories was just 69%. This compared with the MFCC-only LDM system performance of 70%. In other words, whereas combining real EMA features with acoustic features improved performance by a significant amount, adding inferred articulation to acoustics meant the results became marginally worse. Clearly, the trajectories estimated by the MLP must differ from the real articulatory trajectories in some significant and non-beneficial way. In the next chapter, we consider exactly what answer the MLP does give in response to an input vector, and whether this model of the articulatory configurations corresponding to the input vector can be improved.

Chapter 5

Modelling Conditional Distributions

This chapter begins by examining what answer the MLP produces as output for a given input vector from a theoretical standpoint. Next, we look at a possible solution to the shortfall of the sum-of-squares trained MLP by considering the Mixture Density Network (MDN) (Bishop 1995). Since these networks have not been widely used in the speech technology field, the major aspects of how MDNs function are explained. Finally, the theoretical advantages of the MDN over the MLP are tested in an inversion mapping experiment using exactly the same data as used in Chapter 4.

5.1 One-to-many mappings and the MLP

5.1.1 Sum-of-squares training

It is well understood that the output of an MLP trained by minimising the sum-of-squares error function approximates the conditional average of the target values in the training data. This has been shown by taking the sum-of-squares error in the limit as the size N of the training set goes to infinity

$$E = \lim_{N \to \infty} \frac{1}{2N} \sum_{n=1}^{N} \sum_{k} \{y_k(\mathbf{x}^n; \mathbf{w}) - t_k^n\}^2 \tag{5.1}$$

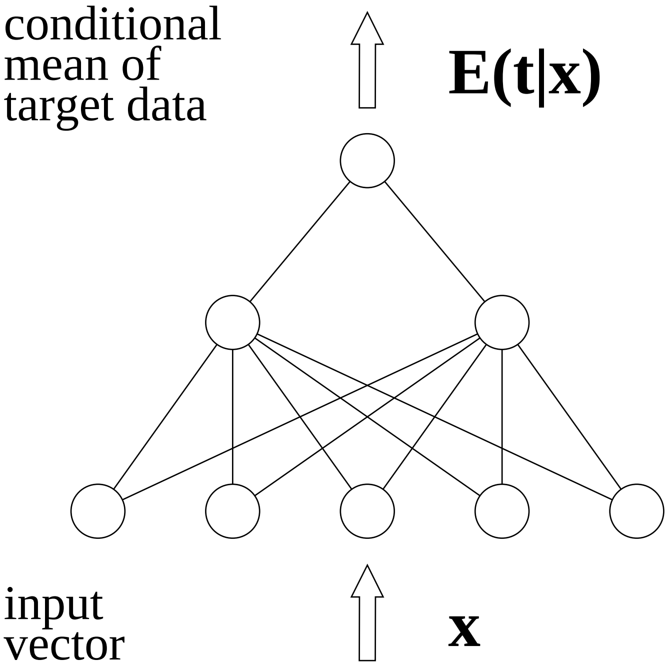


Figure 5.1: It has been demonstrated that the output of an MLP optimised according to the sum-of-squares error function is given by the average of the target data points conditioned on the input vector $\mathbf{x}$.

$$= \frac{1}{2}\sum_k \int\int \{y_k(\mathbf{x};\mathbf{w}) - t_k\}^2 p(t_k, \mathbf{x})\, dt_k\, d\mathbf{x} \tag{5.2}$$

and rewriting this error as follows (see Bishop (1995), pp. 201–202, for the full description of the intermediate steps):

$$E = \frac{1}{2}\sum_k \int \{y_k(\mathbf{x};\mathbf{w}) - \langle t_k|\mathbf{x}\rangle\}^2 p(\mathbf{x})\, d\mathbf{x} + \frac{1}{2}\sum_k \int \{\langle t_k^2|\mathbf{x}\rangle - \langle t_k|\mathbf{x}\rangle^2\} p(\mathbf{x})\, d\mathbf{x} \tag{5.3}$$

where $y_k(\mathbf{x};\mathbf{w})$ is the kth output of the mapping function under input vector $\mathbf{x}$ and trainable parameter vector $\mathbf{w}$, and $\langle t_k|\mathbf{x}\rangle$ is the conditional average of the kth target output. The first term in Equation 5.3 accounts for the error attributable to the model (i.e. the trainable parameters $\mathbf{w}$) compared against the *expected* target value given the input vector. The second term, which is independent of the mapping $y_k(\mathbf{x};\mathbf{w})$ and thus independent of the network parameters $\mathbf{w}$, is attributable to the variance of the target data around the conditional average of the target values ($\langle t_k|\mathbf{x}\rangle$).

Since the second term of the error function is clearly independent from the network mapping $y_k(\mathbf{x};\mathbf{w})$, modification of the network weight vector $\mathbf{w}$ during training will not affect it. Therefore, for the purpose of minimising the sum-of-squares error function during training, the second variance term may be ignored. Since the first term is always positive, it is clear that the minimum of the error function occurs when the first term equals zero, which happens when

$$y_k(\mathbf{x};\mathbf{w}^*) = \langle t_k|\mathbf{x}\rangle \tag{5.4}$$

where $\mathbf{w}^*$ is the optimal configuration of network weights, which corresponds to the minimum in the error function. Equation 5.4 tells us the optimal network output is given by the average of the target data conditioned on the input vector. In other words, under the sum-of-squares error function, the best answer that the network can given is the conditional average of the target data in the training set. There are two important caveats to bear in mind related to this. First, the network must actually have sufficient representational power (i.e. units and links) to achieve this mapping given a suitable optimisation process. Second, the training data set should contain enough data to sufficiently approximate the infinite data set, and hence provide an adequate description of the mapping function.

Let us return to look at the second term of Equation 5.3. It is clear now that if the optimal network configuration $\mathbf{w}^*$ is found by the optimisation process, such that the first term vanishes, then any residual error left over is attributable to the second term. Furthermore, as mentioned, the residual error is a measure of the average variance of the target data points around their mean, conditioned on the input $\mathbf{x}$. In the case of a finite number c of input-output vector pairs in the training data set, the optimal variance (biased estimate) provided by the residual second term in Equation 5.3 can be calculated as

$$\sigma_k^2 = \frac{1}{N} \sum_{n=1}^{N} \{y_k(\mathbf{x}^n; \mathbf{w}^*) - t_k^n\}^2 \tag{5.5}$$

Hence, as Figure 5.2 illustrates, the result of sum-of-squares training is essentially unimodal Gaussian regression on the target data with an $\mathbf{x}$-dependent mean and a global average variance. This result holds whether or not the target data is normally distributed or not; if the target data is not a Gaussian distribution, the network trained with the sum-of-squares error function will nevertheless treat it as such.

5.1.2 Sum-of-squares training and non-uniqueness

Section 5.1.1 discussed how the answer provided by an MLP can approximate the conditional mean of the target data. For classification problems, we can use an "orthogonal" coding scheme, where a target activation of 1.0 for a given output unit signifies the presence of that class or feature, and conversely, an activation of 0.0 signifies the absence of the class or feature. In this case, these averages represent the posterior probabilities of class membership, or presence of the feature. This is beneficial in many respects, and has been put to good use, for example in hybrid ANN/HMM speech recognisers for providing local posterior phone probability estimates (Robinson, Cook, Ellis, Fosler-Lussier, Renals & Williams 2000).

However, for problems involving the prediction of *continuous* variables, the conditional averages provide only a very limited description of the properties of the target

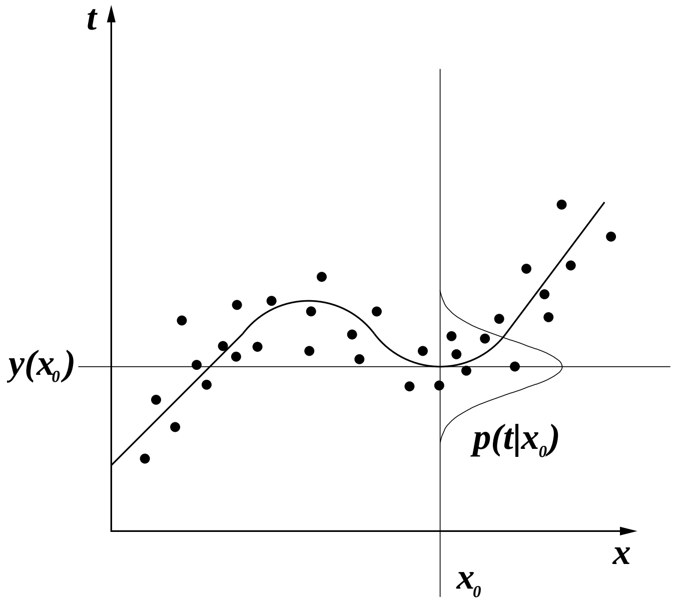


Figure 5.2: It is well known that minimising the sum-of-squares error function during MLP training is equivalent to unimodal regression where the variance is uniform across all input-output vector pairs in the training set. In other words, for any input vector $\mathbf{x}_0$, the best output the network can give under the sum-of-squares error function is the average of the target points $\mathbf{t}$ given $\mathbf{x}_0$. Assuming the network is capable of providing this answer, any residual error is attributable to the variance of all the target data points around their mean, which is again conditioned on $\mathbf{x}_0$.

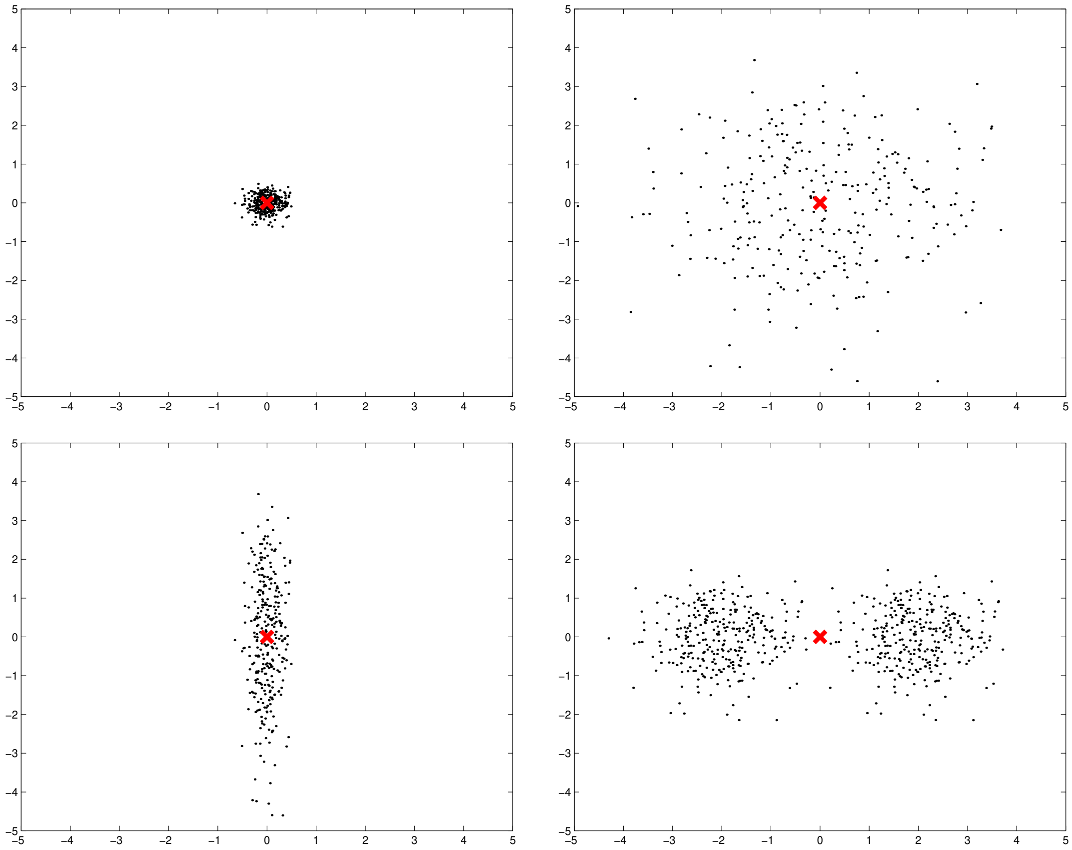


Figure 5.3: Some limitations of the output of a neural network trained using the sum-of-squares error function. It is well know that the output of a neural network trained in this way approximates the conditional mean of the target data points, conditioned on the corresponding input vector. In each of the four plots, the mean of the distribution of data points is indicated by the cross. For the first plot, located in the top left of this figure, we would probably agree that the point estimate given as the answer by the neural network is a reasonable and representative answer. However, in each of the three other plots, the point estimate of the mean by itself does not seem to be a satisfactory answer.

variables. This is particularly true for problems in which the mapping to be learned is multi-valued. We strongly suspect that there are one-to-many mappings inherent to acoustic-to-articulatory inversion, so that a given acoustic input vector might map to a whole range of articulatory output vectors. Thus, it is important to consider what the ramifications of this are when training such a neural network to perform the acoustic-to-articulatory inversion mapping.

Figure 5.3 is a theoretical demonstration of the limitations of the answer that a neural network trained using the sum-of-squares error function provides.

To understand how these examples are intended to represent the inversion mapping, the reader should imagine three conditions. First, that these plots represent some 2-D projection from the 14 dimensional articulatory space, and that each of the 300 points shown in each plot is a point in this articulatory domain. For example, `tt_x` could be plotted against `ul_y`[1]. The second condition that the reader is asked to imagine is that with each of these articulatory data points is associated an acoustic vector. This corresponds to the acoustic signal that would be produced by the articulatory configuration at each data point. Finally, the reader should consider that the 300 acoustic vectors associated with each of articulatory points are the **same** point, or at least very close to each other. In short, therefore, the aim here is to consider what answer our MLP trained using the sum-of-squares error function will give us when presented with a training set containing different distributions of target articulatory configuration that have potentially all produced a single given acoustic input vector.

Looking first at the upper left plot, we see that the articulatory configurations that produce the hypothetical acoustic input vector are confined to a reasonably constrained area within the articulatory space. The mean of these data points is represented by the red cross. From the discussion in Section 5.1.1, we know this mean is theoretically the best answer that the MLP trained on these 300 data points using the sum-of-squares error function would return. In this case, it seems that this mean value is a reasonable answer.

In contrast, we would no doubt have reservations about confirming the red cross

[1] In fact, the data points shown in these plots are not articulatory coordinates, but merely points drawn from a two dimensional Gaussian with certain transformations performed on them.

in the upper right plot as a "reasonable" answer. It is off-putting that, compared with the previous plot, the articulatory configurations are spread widely throughout the articulatory domain. In other words, the distribution of possible target articulatory configurations has high variance.

The plot located in the bottom left of the figure is a refinement of this problem. Here, the range of potential articulator positions is larger in the y-axis than it is in the x-axis. However, the answer that the neural network would give after training on such a data set, again represented by the red cross, does not convey this information. Such a situation might arise in the case of a bilabial stop. For example, the bottom lip is relatively crucial to the production of an [m] segment, and so the variance on the position of this articulator will be low across multiple instances of [m] segments. Conversely, the position of the tongue does not have a critical effect on the production of an [m] segment, and so the variance on the position of the tongue dorsum might be relatively high. Therefore, we might expect to observe this kind of distribution in a projection from the 14D articulatory space when plotting `ll_y` against `td_y`.

Finally, the forth plot illustrates what happens when the neural network is faced with the task of mapping to a multi-modal distribution of points in the target articulatory domain. Because the the network is constrained to learning the conditional mean of the target points, the answer it provides when presented with our acoustic vector could potentially be particularly poor. Not only does the answer in this example lack any information about the variance around the point estimate, but the point estimate itself may be non-representative; the conditional mean does not really belong to either cluster, but is located at some compromise between the two.

This theoretical demonstration highlights at least two pitfalls inherent to using a sum-of-squares trained MLP for modelling the inversion mapping. First, if the articulatory target data features instances of multimodal distributions (as Roweis (1999) has shown; see Figures 2.1 and 2.2), the average of those points according to the sum-of-squares error solution may not actually be a reasonable articulatory configuration at all. As we have seen, the MLP does not have the power to model distributions of target data points any more complex than a unimodal Gaussian. Second, the MLP only provides the mean value of the articulatory target for a given acoustic input. We do not receive any indication as to the variance around that mean. For example, the MLP is not able to distinguish between the case where the target values corresponding to a given input are clustered tightly around their mean and the case where they are spread throughout a large region about the mean. In short, there is unfortunately no way of knowing when the MLP output is likely to be accurate and when to believe it less. In addition, because the MLP gives only the point estimate of the target articulatory data, the MLP may introduce inappropriate consistency to the detriment of stochastic learning models trained on the estimated articulation. In effect, the variation possible in articulatory parameters during a given phone is potentially lost. Consequently, this could have an adverse effect on the way that probabilistic models perform with the recovered articulatory trajectories compared with real articulatory trajectories measured by EMA. This could explain the poor results reported in the LDM-based phone classification tests at the end of Chapter 4.

In the next section, a model is described which allows us to capture both the multimodal aspects of the inversion mapping, as well as giving the variance around the estimated articulatory positions. Specifically, we look at applying a type of neural network called the Mixture Density Network (MDN) to the acoustic-to-articulatory mapping problem. In contrast to the simple MLP, the MDN is capable of modelling a full conditional probability density over the target domain for each input vector. In terms of mapping from acoustics to articulation, the MDN will output a full probability density function for the positions of the articulators over the whole articulatory domain in response to an acoustic input vector.

Since mixture density networks are not commonplace in the speech field, for the reader's convenience we shall first briefly introduce the theory underpinning the model. For a complete description, the reader is directed to Bishop (1994) (or Bishop (1995)). After describing the model, we go on to look at how we have applied this type of neural network to the acoustic-to-articulatory inversion task.

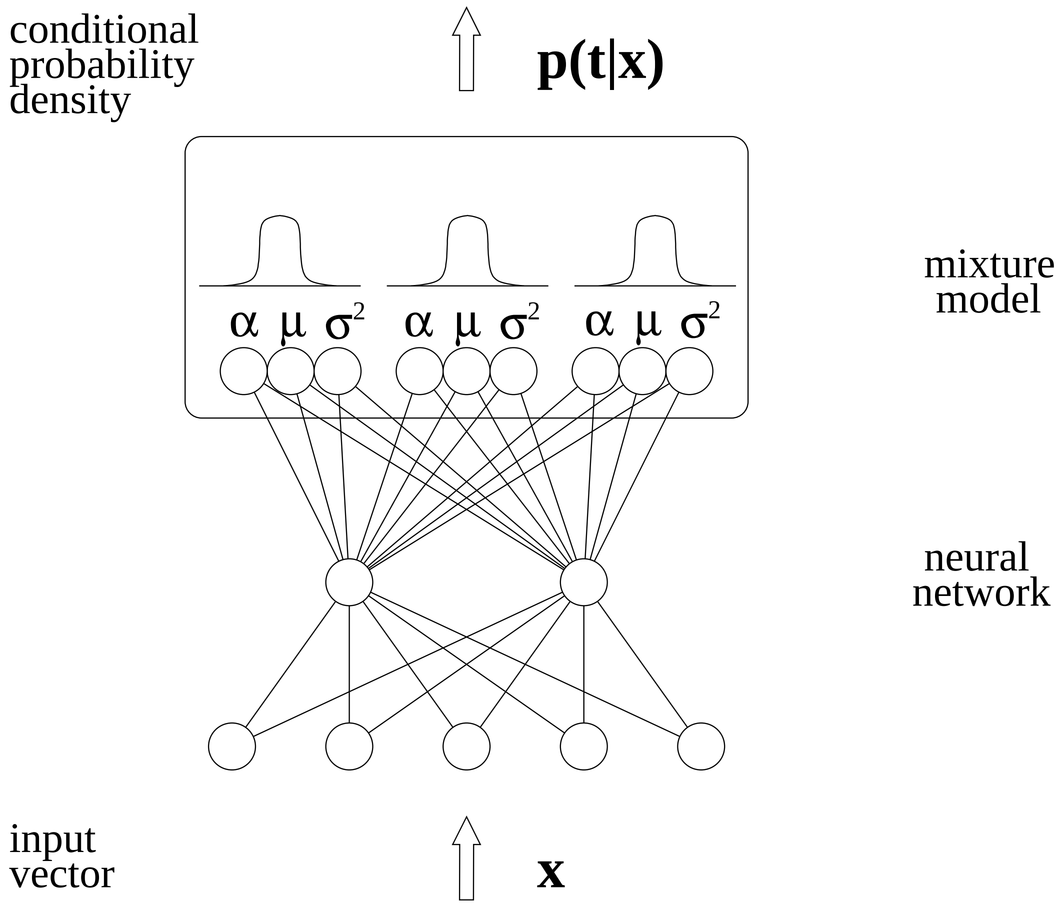


Figure 5.4: The mixture density network is a combination of a mixture model and a neural network. In a trained MDN, the neural network maps from the input vector $\mathbf{x}$ to the control parameters of the mixture model, which in this case uses Gaussian components (priors α, means μ and variances σ^2) but in theory could be any number of kernel functions. The mixture model gives a full pdf description ($p(\mathbf{t}|\mathbf{x})$) of the target domain conditioned on the input vector, whereas in Figure 5.1 the MLP could only provide the conditional average of the target data.

5.2 Introduction to Mixture Density Networks

A mixture density network can be thought of as the combination of a conventional neural network with a mixture density model. An example MDN is shown in Figure 5.4. In this example, the MDN takes an input vector $\mathbf{x}$ of dimensionality 5 and gives the conditional probability density of a vector $\mathbf{t}$ of dimensionality 1 in the target domain. This is in contrast to the example MLP in Figure 5.1, which was only capable of outputting the conditional average of the target data. We shall first consider each of the two components of the MDN separately, and then go on to consider how the mixture density network is trained.

5.2.1 Mixture model component of the MDN

The density function in the example MDN shown in Figure 5.4 is modelled by a Gaussian mixture model with 3 components, so that it is given by

$$p(\mathbf{t}|\mathbf{x}) = \sum_{j=1}^{M} \alpha_j(\mathbf{x})\phi_j(\mathbf{t}|\mathbf{x}) \tag{5.6}$$

where M is the number of mixture components (in this example, three), $\phi_j(\mathbf{t}|\mathbf{x})$ is the conditional probability density given by the jth kernel, and $\alpha_j(\mathbf{x})$ is the mixing coefficient for the jth kernel. The mixing coefficients can be thought of as the *prior* probability that a target vector $\mathbf{t}$ has been generated by the jth kernel, and must sum to one. Note that any of a number of different kernel functions may be used in the mixture model, but only Gaussian kernel functions are considered here.

The target domain in the toy example shown in Figure 5.4 has a dimensionality of just one, and therefore there is naturally only one variance parameter for each component kernel in the mixture model. If the dimensionality were greater than one, we could still use a single variance parameter for each of the kernels. Such a Gaussian is said to have "spherical" covariance, as it is constrained to share the same variance in all dimensions, and thus can be envisioned as a hypersphere in an n-dimensional space.

In principle, the MDN is not limited to using spherical covariance, and so we could also use kernels with either diagonal or full covariance matrices. However, complicating the mixture density network in this way is not necessarily beneficial, because a mixture

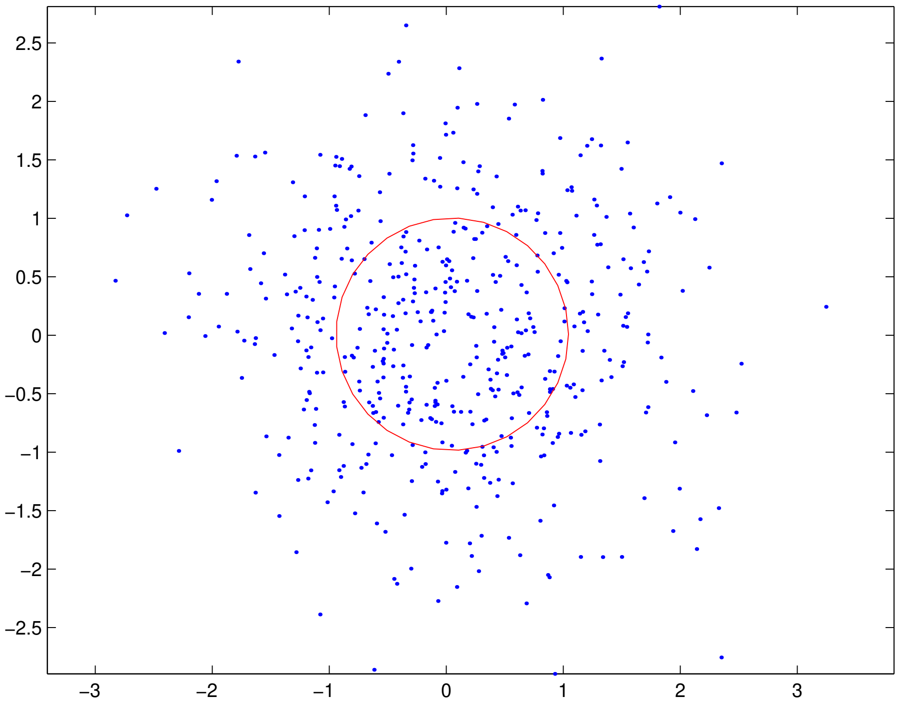


Figure 5.5: A scatter plot of a data set D, comprising 500 points in 2-D space. The set D was generated by sampling from a normal distribution with zero mean and unit variance in both dimensions. The red circle represents a Gaussian fitted to the data D, and has its centre at the mean and radius equal to one standard deviation.

of Gaussians is theoretically able to model any distribution function with arbitrary accuracy, provided enough components are available (Bishop 1995). This is illustrated graphically in Figures 5.5 to 5.12.

Figure 5.5 shows a scatter plot of a data set D of 500 points in 2-D space. This set D was generated by sampling from a normal distribution with zero mean and unit variance in both dimensions. Also shown is the single Gaussian having spherical covariance with mean and variance calculated from set D. This is represented by the red circle, which has its centre at the mean, or at the point (0.0513, 0.0092). The radius of the circle is set to equal one standard deviation, in this case 0.9872. Notice that these are very close to the parameters of the generating distribution, and that this Gaussian with spherical covariance models the distribution adequately.

However, when the variance of the Gaussian is restricted to be the same in both dimensions, a single Gaussian with spherical covariance does not have the flexibility to describe any more complicated distributions than the example shown in Figure 5.5. For example, the distribution in Figure 5.6 does not have equal variance in all dimensions. For this diagram, the same data set D shown in Figure 5.5 has been scaled by 0.5 in the y-dimension only. Using this transformation of set D is equivalent to drawing 500 samples from a Gaussian distribution with zero mean and variances 1.0 and 0.25 in the x- and y-dimensions respectively. The diagram shows how the single Gaussian with spherical variance is not able to accurately model the distribution of points, over-estimating the variance in one direction and under-estimating the variance in the other. Specifically, the variance of this Gaussian (0.6253), is not close to the variance in either the x- or the y- dimensions.

Figure 5.7 shows the result of fitting a Gaussian with 'diagonal' covariance to the same data set as used in Figure 5.6. A Gaussian with diagonal covariance has a co-variance matrix of size $n \times n$, where n is the dimensionality of the space. There is one variance parameter on the diagonal of the matrix for each dimension of the space. All other entries are zero. This means that a Gaussian with diagonal covariance is able to model data sets with independent variance in the different dimensions, as demonstrated in Figure 5.7.

However, when the distribution of the data points in one dimension is *not* independent of the distribution of the points in another dimension, we find that the Gaussian with diagonal covariance provides an inadequate model (see Figure 5.8). To simulate a correlated distribution, where the points' x-coordinates are correlated with the points' y-coordinates, the scaled data set D shown in Figure 5.7 has been rotated anti-clockwise around the mean by 30 degrees. The Gaussian with diagonal covariance is unable to accurately model a set of data points that has an underlying distribution featuring correlation between dimensions[2]. A full covariance matrix is necessary to model arbitrary scaling and rotation in a Gaussian distribution model, as shown in Figure 5.9.

Although a single Gaussian with spherical covariance provides only limited modelling power, a *mixture* of multiple spherical Gaussians is much more flexible. In fact, given

[2] This is one reason why melcepstra are suited for use in typical HMM systems; they are "minimally correlated", and so may be modelled by Gaussians with diagonal covariance matrices.

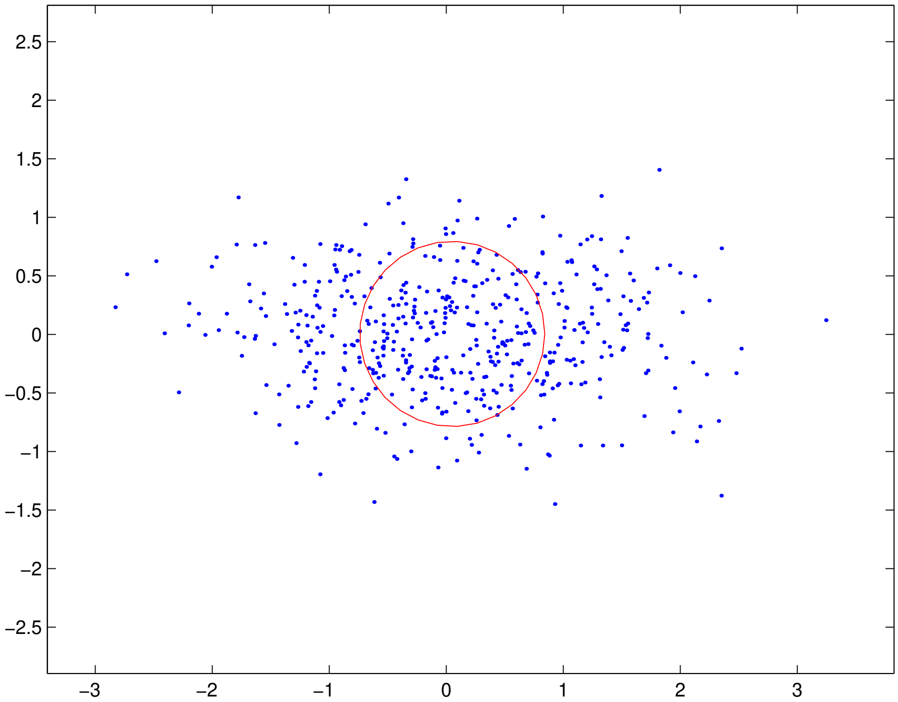


Figure 5.6: A scatter plot of 500 points in 2-D space, generated by sampling from a normal distribution with zero mean and unit variance in the x-dimension and variance 0.25 in the y-dimension. The red circle represents a single Gaussian with spherical covariance fitted to the data set, having its centre at the mean and radius equal to one standard deviation. Notice that a single Gaussian with spherical covariance is not suited to modelling such a data set where the variance is not the same in the different dimensions. Compare this with Figure 5.7

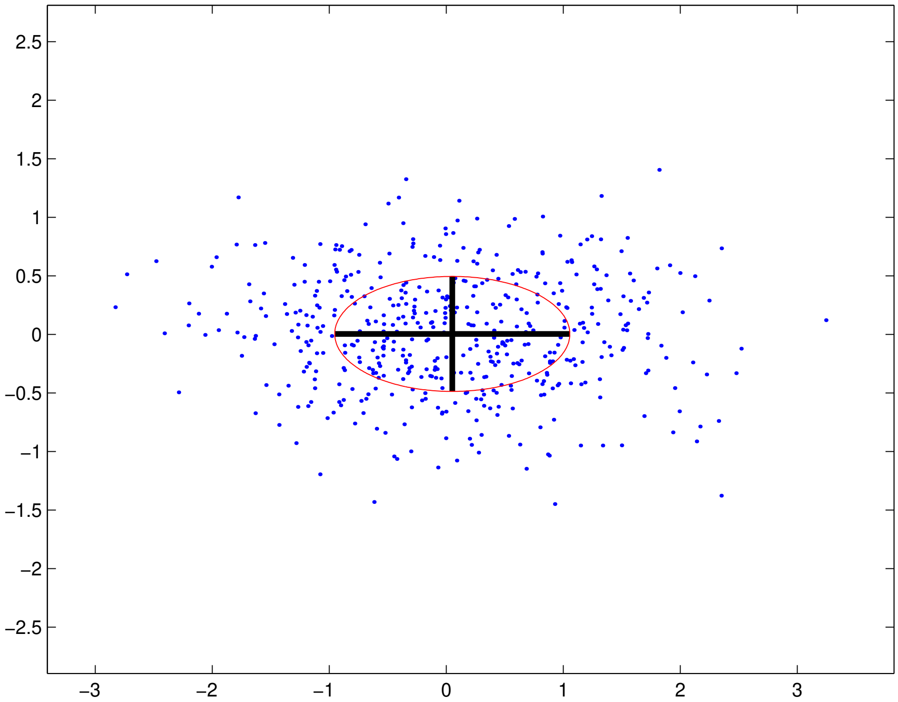


Figure 5.7: A scatter plot of 500 points in 2-D space, generated by sampling from a normal distribution with zero mean and unit variance in the x-dimension and variance 0.25 in the y-dimension. The red ellipse represents a single Gaussian with diagonal covariance fitted to the data set. Compare this model of the data set to that shown in Figure 5.6. Not surprisingly, the Gaussian with diagonal covariance, which has a separate variance term for each dimension, is able to model the data set distribution appropriately.

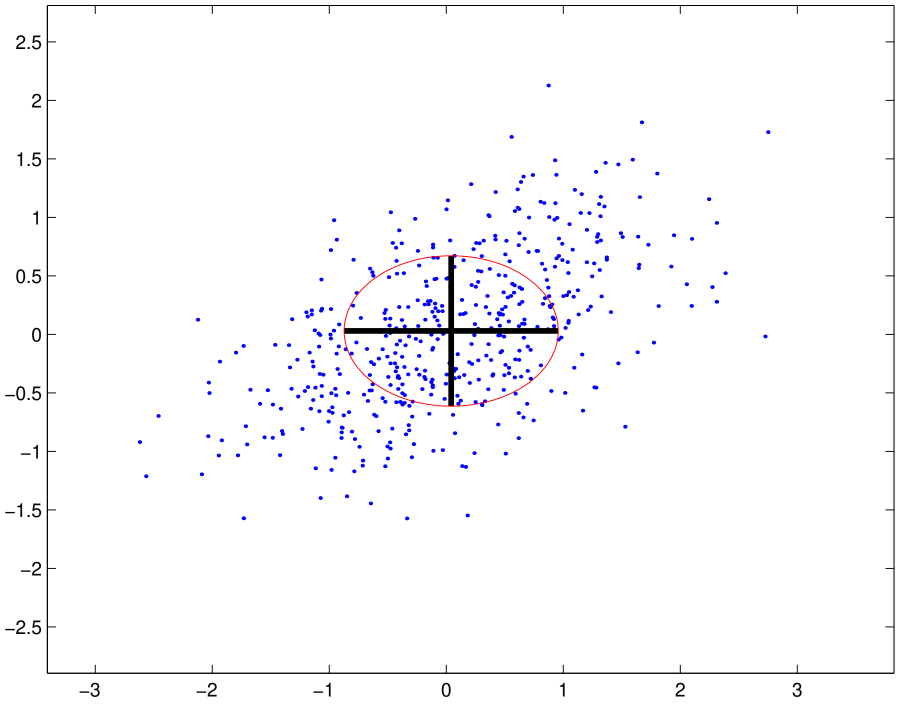


Figure 5.8: A scatter plot of 500 points in 2-D space, generated by sampling from a normal distribution with zero mean and unit variance in the x-dimension and variance 0.25 in the y-dimension. The data set D was then rotated 30 degrees around point (0,0). The red ellipse represents a single Gaussian with diagonal covariance fitted to the data set. Note that the diagonal covariance matrix is unable to provide any rotation in the Gaussian, so cannot cope with correlations between dimensions.

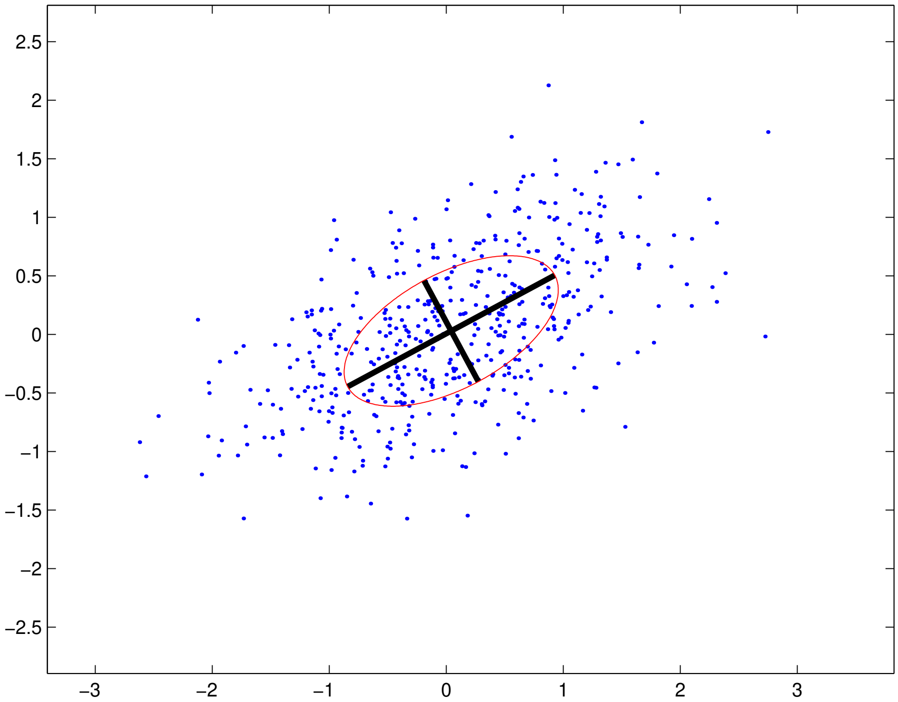


Figure 5.9: A scatter plot of 500 points in 2-D space, generated by sampling from a normal distribution with zero mean and unit variance in the x-dimension and variance 0.25 in the y-dimension. The data set was subsequently rotated anti-clockwise around point (0,0) by 30 degrees to simulate a correlation between x- and y-dimensions. The red ellipse represents a single Gaussian with a full covariance matrix fitted to the data set. This Gaussian with full covariance has the power to model the data distribution, which features correlation, unlike the Gaussian with diagonal covariance shown in Figure 5.8.

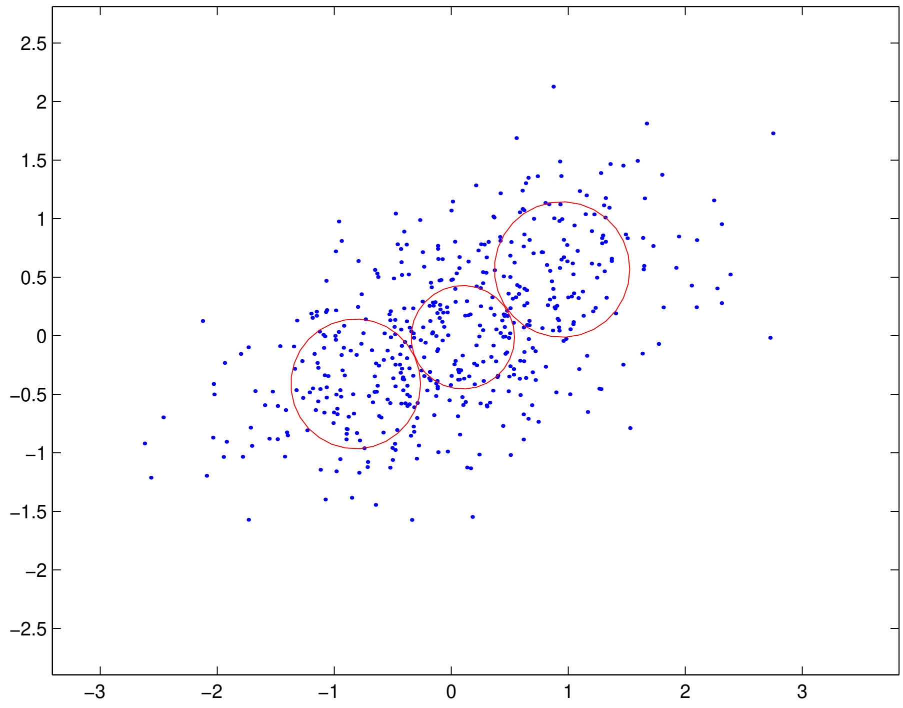


Figure 5.10: A scatter plot of the same 500 points in 2-D space as used in Figure 5.9. However, this time a mixture of three Gaussians with spherical covariance has been fitted to the data set. The individual Gaussian components of the mixture model are represented by the red circles, with their centre located at their respective mean, and their radius equal to one standard deviation from the mean. The combined effect of these three components is shown in the probability density contour plot in Figure 5.11.

enough components, a mixture of Gaussians with spherical covariance can in principle approximate any distribution to arbitrary accuracy. This includes non-Gaussian distributions too. Figure 5.10 shows a mixture of three Gaussians with spherical covariance fitted to the same distribution of points as in Figure 5.8. Finally, the probability density contour plots in Figures 5.11 and 5.12 provide a comparison of the density function for the mixture model and for the Gaussian with a full covariance matrix.

5.2.2 Network component of MDN

The task of the neural network component of the MDN is to provide the parameters of the probability density model in response to the acoustic input vector. To simplify

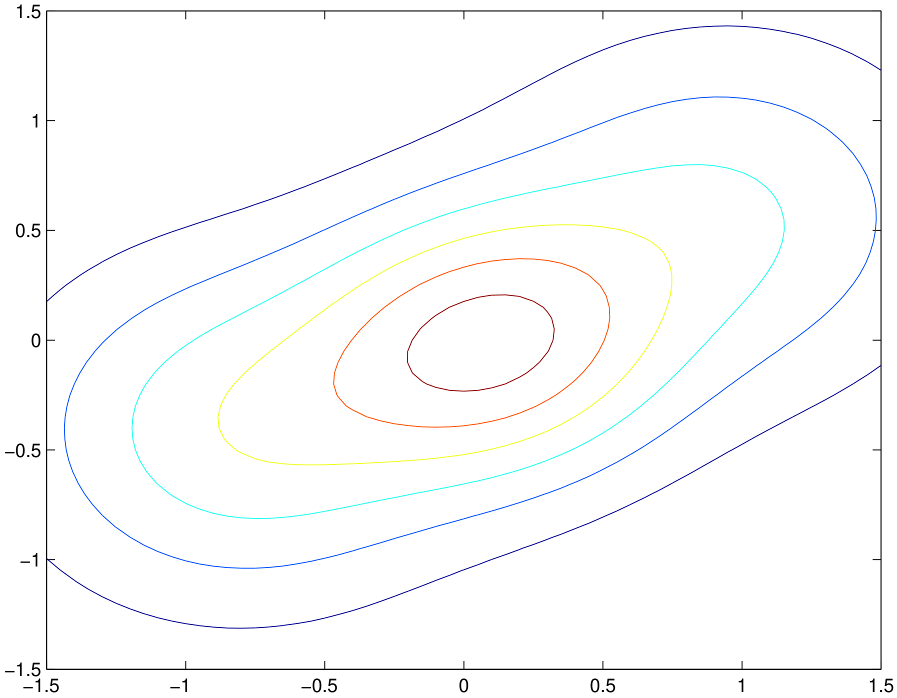


Figure 5.11: A contour plot of probability density for the Gaussian mixture model shown in Figure 5.10. Compare the probability density function represented by this GMM to the density function modelled by the unimodal Gaussian which has a full covariance matrix, shown in Figure 5.12.

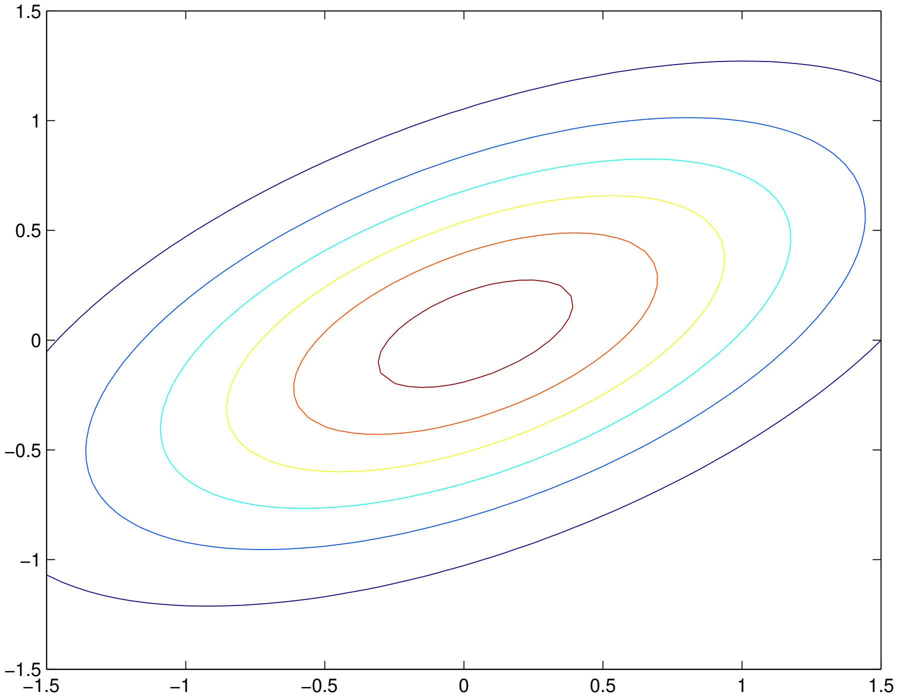


Figure 5.12: A contour plot of probability density for the Gaussian with a full covariance matrix, shown in Figure 5.9. Compare the probability density function represented by this model to the density function modelled by the mixture of three Gaussians with spherical covariance, shown in Figure 5.11.

the discussion, we will assume at this stage that the neural network is somehow already trained to output the correct mixture model parameters in response to a given acoustic input vector. In Section 5.2.3, the question of how to train MDNs is addressed.

In theory, any neural network with universal approximation capabilities can be used to map from the input vector to the mixture model parameters. In the example in Figure 5.4, we see a feedforward MLP with 5 input units, a hidden layer of 2 units with sigmoidal activation and 9 linear output units for the mixture parameters. In general, the total number of network outputs necessary for a mixture of Gaussians with spherical covariance is $(c+2) \times M$, where c is the dimensionality of the target domain, and M is the number of mixture components. Each mixture component has 1 unit for its prior, 1 unit for its variance and c units for the mean of the component in the target space. If Gaussians with diagonal covariance were used, then the network would need enough additional output units for c variance parameters for each component. Using Gaussians with full covariance matrices would obviously necessitate $c \times c$ variance output units for each mixture component.

The output units of the MLP have a linear activation function. However, to make sense as mixture model parameters, certain constraints on their output value are necessary, depending on whether the output unit represents a prior, mean or variance parameter. Clearly, the mixing coefficients (component prior) must lie in the range $0 \leq \alpha_j(\mathbf{x}) \leq 1$ and sum to one in order to represent probabilities. This is achieved by using the *softmax* function (Bridle 1990) to relate the mixing coefficients of the mixture model to the output of the corresponding units in the neural network:

$$\alpha_j = \frac{\exp(z_j^{\alpha})}{\sum_{l=1}^{M} \exp(z_l^{\alpha})} \tag{5.7}$$

where z_j^{α} is the output of the neural network corresponding to the mixture coefficient for the jth mixture component.

The variances of the mixture model are related to the corresponding outputs of the neural network according to the following function:

$$\sigma_j = \exp(z_j^{\sigma}) \tag{5.8}$$

where z_j^σ is the output of the neural network corresponding to the variance for the jth mixture component. This has the convenience of avoiding the variance becoming less than or equal to zero. Finally, the mean parameters for the mixture model are represented directly by the corresponding outputs of the neural network:

$$\mu_{jk} = z_{jk}^{\mu} \tag{5.9}$$

where z_{jk}^{μ} is the value of the output unit corresponding to the kth dimension of the mean vector for the jth mixture component.

To summarise, the process of obtaining a probability density function over variables in the target domain given an input vector is straightforward. The trained neural network is presented with the input vector, the network activation is propagated forward to give an output vector. This output vector is transformed using Equations 5.7, 5.8 and 5.9 to yield the prior, variance and mean mixture model parameters respectively. The next section covers how the mixture density network is trained.

5.2.3 Training the MDN

The objective of training the MDN is to minimise the negative log likelihood of the observed target data points in the training set given the mixture model parameters. This error function can be expressed as

$$E = -\sum_n \ln \left\{ \sum_{j=1}^{M} \alpha_j(\mathbf{x}^n) \phi_j(\mathbf{t}^n|\mathbf{x}^n) \right\} \tag{5.10}$$

with $\phi_j(\mathbf{t}|\mathbf{x})$, because we are using Gaussian kernel functions, given by

$$\phi_j(\mathbf{t}|\mathbf{x}) = \frac{1}{(2\pi)^{\frac{c}{2}} \sigma_j^c(\mathbf{x})} \exp\left\{ -\frac{\|\mathbf{t} - \boldsymbol{\mu}_j(\mathbf{x})\|^2}{2\sigma_j^2(\mathbf{x})} \right\} \tag{5.11}$$

Since it is the neural network which provides the parameters for the mixture model for each input-output vector training pair, the error function given in Equation 5.10 must be minimised *with respect to the network weights.* Fortunately, the derivatives of the error at the network output units corresponding separately to the priors, means and variances of the mixture model may be calculated as follows:

$$\frac{\partial E^n}{\partial z_j^\alpha} = \alpha_j - \pi_j \tag{5.12}$$

$$\frac{\partial E^n}{\partial z_j^\sigma} = -\pi_j \left\{ \frac{\|\mathbf{t} - \boldsymbol{\mu}_j\|^2}{\sigma_j^2} - c \right\} \tag{5.13}$$

$$\frac{\partial E^n}{\partial z_{jk}^\mu} = \pi_j \left\{ \frac{\mu_{jk} - t_k}{\sigma_j^2} \right\} \tag{5.14}$$

where the posterior probabilities π_j are obtained from the prior probabilities α_j and the conditional density functions ϕ_j using Bayes' Theorem,

$$\pi_j(\mathbf{x}, \mathbf{t}) = \frac{\alpha_j \phi_j}{\sum_{l=1}^{M} \alpha_l \phi_l} \tag{5.15}$$

The derivation details of these expressions for calculating the derivatives of the error function with respect to the neural network weights can be found in Bishop (1994) (or Bishop (1995)). The error 'signals' given in Equations 5.12, 5.13 and 5.14 can then be propagated back through the network as normal in network training to find the derivatives of the error with respect to each of the network weights. The derivatives of the error with respect to the network weights indicate how to modify each of the weights in order to reduce the overall error. Thus, training is a non-linear optimisation problem to which standard non-linear optimisation algorithms can be applied in order to find a minimum in the error landscape. The SCG optimisation algorithm was used to train the MDNs described in this chapter.

5.3 Articulatory estimation by MDN

For each of the fourteen articulatory channels separately, mixture density networks were trained containing a combination of either 5 or 10 hidden units in a single hidden layer and either 1 or 2 Gaussian kernels in the density function. Thus, 56 different mixture density networks were trained in total.

Using a single Gaussian in the mixture model is similar to the unimodal regression observed when using an MLP to estimate articulation from acoustics (as discussed in Section 5.1.1). However, in the case of the mixture density network with one Gaussian output, the variance of the Gaussian as well as the mean is conditioned on the acoustic input vector. This yields greater modelling flexibility than with MLP training by the sum-of-squares error rule, where the unimodal regression effectively is bound to have a single, global variance.

Using a mixture of two Gaussians in the mixture density network pdf introduces the additional flexibility of allowing potentially non-Gaussian and bimodal distributions in the target articulatory domain to be modelled more closely. Meanwhile, although trying only 5 and 10 units in the hidden layer does not represent an exhaustive search through the range of hidden layer sizes, it does give some small scope for indicating whether more or less than these numbers are necessary for the optimal generalisation and estimation performance.

Exactly the same training, validation and test data sets as were used with the MLPs described in Chapter 4 were used for training the mixture density networks. Thus the mixture density networks in this chapter have 400 acoustic input units too, which comprise a context window of 20 frames of 20 filterbank coefficients.

Prior to training, the MDNs were first initialised by a k-means based initialisation algorithm. For this initialisation algorithm, the weights for the network are first randomised by sampling from a Gaussian. Then, a Gaussian mixture model of the same form as the MDN output is used to model the **unconditional** density of the target data. The k-means algorithm is used to determine the component centres. The priors are computed from the proportion of the target data belonging to each component, and the variances are calculated as the sample variance of the target data points belonging to each component from the associated mean. Finally, the network biases are adjusted so that the net will output the values in the Gaussian mixture model. For the examples presented here, 10 iterations of the k-means algorithm were used.

The MDNs were trained using the Scaled Conjugate Gradients optimisation algorithm. As is common practice with empirical learning models, training was continued until the network ceased to show an improvement in performance on the separate validation set. Figure 5.13 illustrates this point, using the MDN trained for the `li_y` channel, with 5 units in the hidden layer and a single Gaussian output pdf. The plot compares the MDN error on the separate training and validation sets over the course of 1500 training epochs. The error at each epoch is calculated as the negative log likelihood of all the target data in the two sets given the probability density functions output by the MDN. Since the training set contained almost five times as many acoustic-articulatory vector pairs as the validation set (46367 compared to 10840), the plot shows the average negative log likelihood, calculated by dividing by the respective number of vector pairs in the two sets, in order to ease comparison. From this graph, we see that the likelihood of the target data in the training set increases throughout the 1500 training epochs, rapidly at first and then more slowly. To begin with, as with the training data set, the likelihood of the validation set articulatory data increases rapidly. However, this learning rate begins to tail off after around 125 epochs. Further training does increase the likelihood of the validation set, but not as much as for the training set. As training progresses further, the improvement of the network on the validation set becomes less and less, while the MDN overfits the training set more and more. Overfitting can become pronounced enough to harm the performance of the network on novel data; after around 1000 epochs, the performance of the network on the validation set decreases slightly. The best performance achieved by this network was at 939 epochs, which is indicated by the black cross.

An upper bound of 2000 training epochs was placed on the network training. In other words, training of all networks was stopped after 2000 epochs regardless of whether it appeared that a minimum in the error function on the validation set had been reached. Of the 56 networks trained, only three networks appeared not to have satisfied this criterion within 2000 epochs, and could potentially have improved further if allowed more training cycles. These were the `tt_y` MDN with 5 hidden units and 2 Gaussians, the `v_x` MDN with 5 hidden units and 1 Gaussian, and the `v_y` MDN with 5 hidden units and 2 Gaussians.

5.4 Results

The 56 networks described in Section 5.3 were trained with the Scaled Conjugate Gradients optimisation algorithm until the best possible score on the validation set was observed. These trained networks were then all tested on the unseen test set (46 utter-

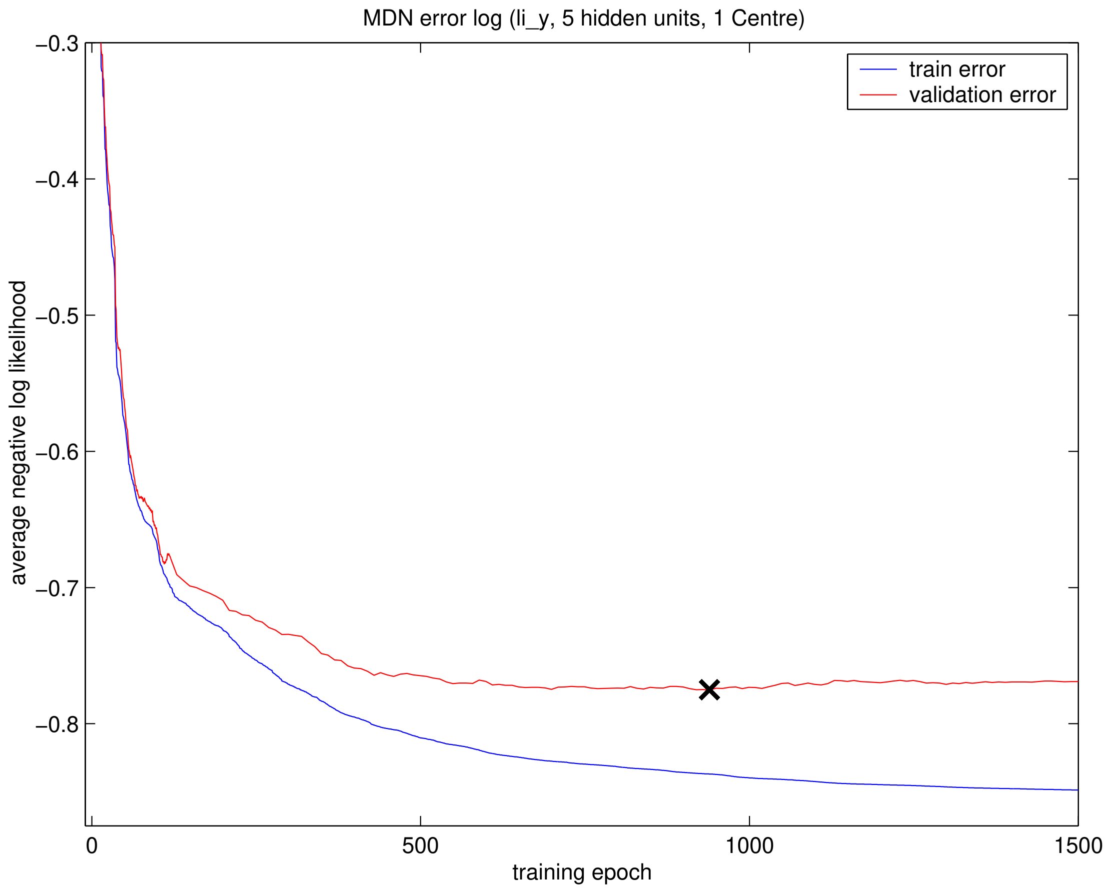


Figure 5.13: Comparison of training and validation set error while training a MDN with 400 acoustic inputs, 5 hidden units and a single Gaussian pdf in the target articulatory domain of the `li_y` channel. The error measure used is the average negative log likelihood, calculated by dividing the negative log likelihood for the training and validation sets as a whole by their respective number of acoustic-articulatory vector pairs contained. As is normal practice with empirical learning methods, the validation set is used to identify when the network has reached optimal generalisation, and thus when to stop training. Although the negative log likelihood on the training set continues to increase with more training (at least up to 1500 epochs), the negative log likelihood on the validation set can cease to improve, and even become worse. The black cross indicates the best score achieved on the validation set by this network (at epoch 939), after which point further training lead to degradation in performance.

ances). The results of this testing are presented in Figures 5.14 to 5.20.

Taking Figure 5.16 as an example, the bar charts show the negative log likelihood for the test set which was obtained for the best of each mixture density network combination trained for the x- and y-channels for one articulator, in this case those for the lower incisor.

5.4.1 Number of Gaussian kernels

A striking trend observed for all articulatory channels is that using two Gaussian kernels in the output mixture density model of the MDNs improves modelling accuracy compared to using just one Gaussian. This is apparent by comparing the pairs of networks containing a fixed number of hidden units, either 5 or 10, for each articulatory channel. The only case where this is not true is for the pair of `ll_y` channel MDNs containing 10 hidden units. However, it is possible this anomaly is just an artifact of network training. The MDN with a 2-Gaussian mixture and 10 hidden units may simply have become stuck in a local minimum, and never realised its full potential by reaching the global minimum in the error landscape. Training all the networks for multiple runs with different random starting conditions and comparing aggregate results could help to determine whether or not this is the case.

5.4.2 Hidden layer size

The results presented in Figures 5.14 to 5.20 seem to suggest that using 10 units in the hidden layer improves modelling accuracy. This is apparent by a pairwise comparison of networks trained for each articulator containing a fixed number of Gaussians in the mixture density model, with either 5 or 10 units in the hidden layer. This observation applies to 27 out of 28 of the pairwise comparisons. The exception is the MDN with 1 Gaussian and 10 hidden units trained on the `ul_y` articulatory channel, which is marginally worse than the MDN with 1 Gaussian and 5 hidden units. As mentioned in Section 5.4.1, this could be due to a training "glitch"; this would at least have to be ruled out before ascribing any significance to this fact.

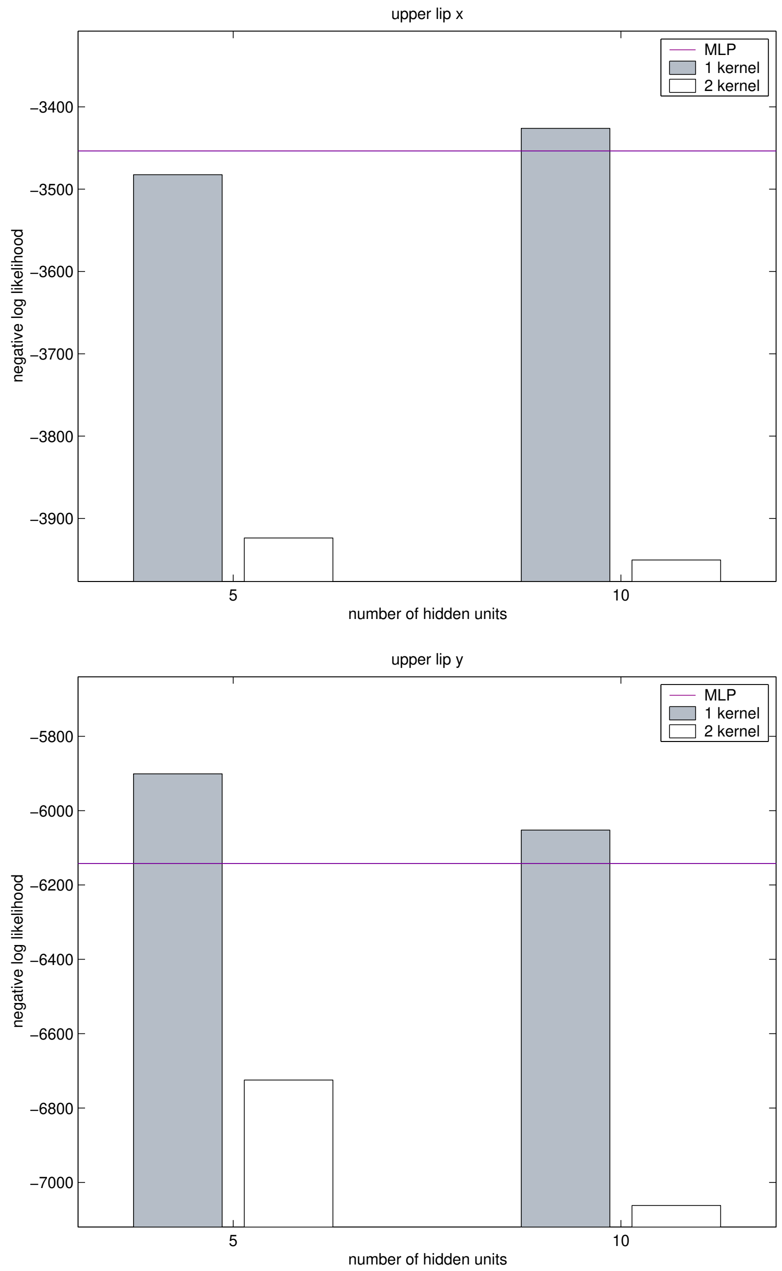


Figure 5.14: Comparison of MDN test set error for the **upper lip** channels. MDNs were trained with hidden layers containing 5 and 10 hidden units, and output probability density functions comprised of either a single Gaussian or a mixture of 2 Gaussians. A separate validation set was used during the training stage to identify when the network had reached an optimum. The negative log likelihood error of the target data in the unseen test set was then calculated for the optimal example of each network architecture combination. For comparison, the plots also show the negative log likelihood of the same test data set as calculated from the output of the best MLP from Chapter 4, represented as a horizontal line across the plot.

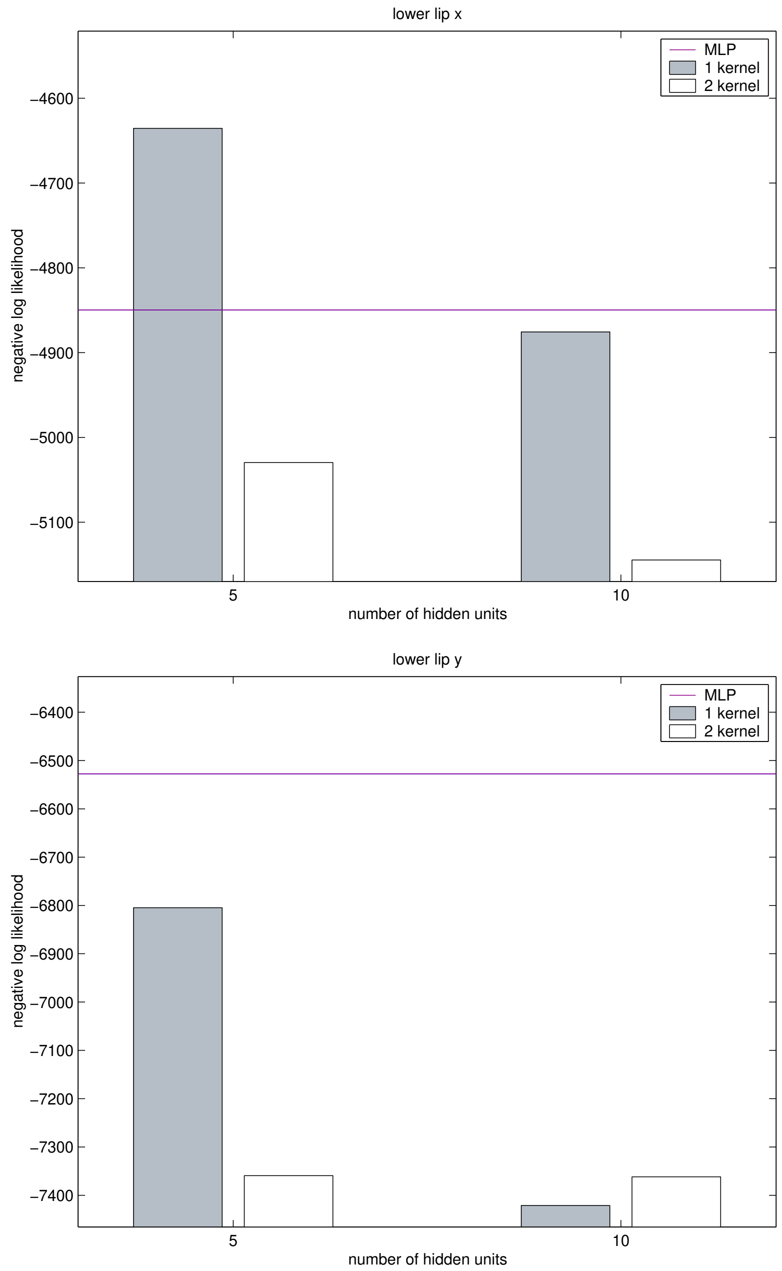


Figure 5.15: Comparison of MDN test set error for **lower lip** channels. MDNs were trained with hidden layers containing 5 and 10 hidden units, and output probability density functions comprised of either a single Gaussian or a mixture of 2 Gaussians. A separate validation set was used during the training stage to identify when the network had reached an optimum. The negative log likelihood error of the target data in the unseen test set was then calculated for the optimal example of each network architecture combination. For comparison, the plots also show the negative log likelihood of the same test data set as calculated from the output of the best MLP from Chapter 4, represented as a horizontal line across the plot.

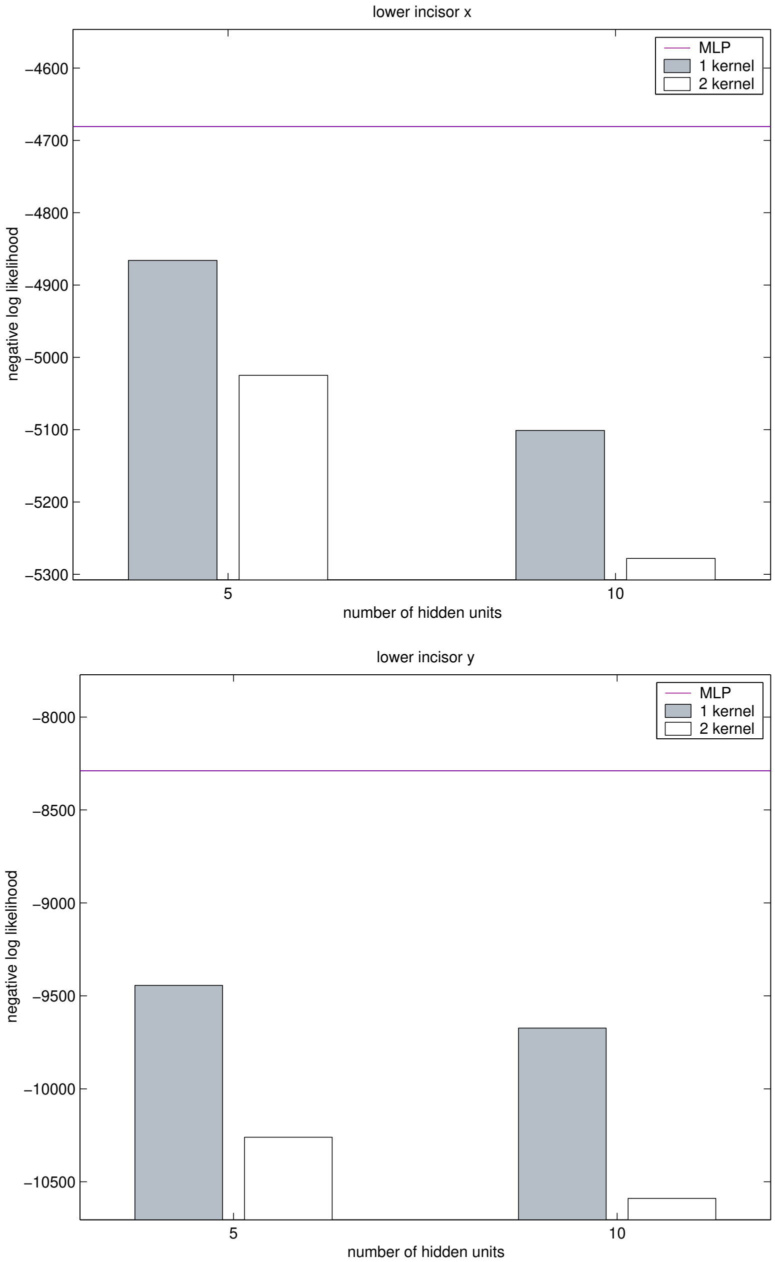


Figure 5.16: Comparison of MDN test set error for **lower incisor** channels. MDNs were trained with hidden layers containing 5 and 10 hidden units, and output probability density functions comprised of either a single Gaussian or a mixture of 2 Gaussians. A separate validation set was used during the training stage to identify when the network had reached an optimum. The negative log likelihood error of the target data in the unseen test set was then calculated for the optimal example of each network architecture combination. For comparison, the plots also show the negative log likelihood of the same test data set as calculated from the output of the best MLP from Chapter 4, represented as a horizontal line across the plot.

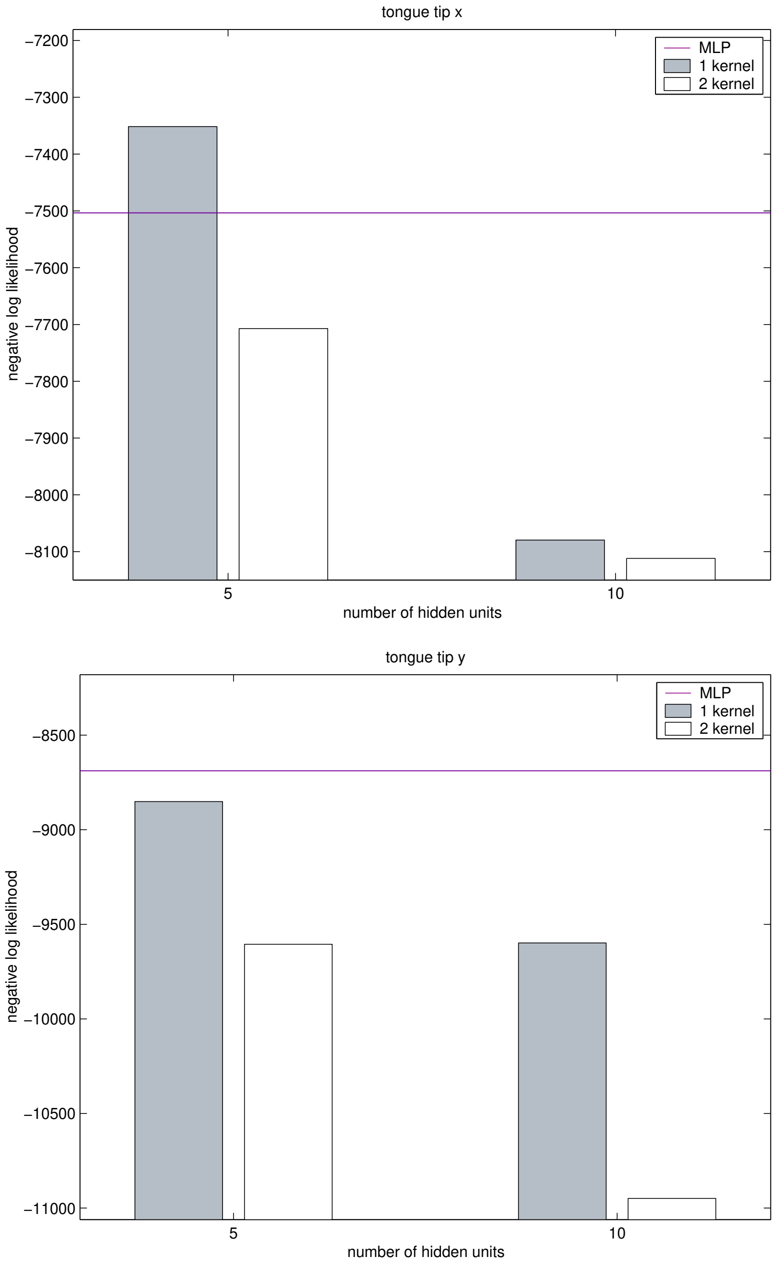


Figure 5.17: Comparison of MDN test set error for **tongue tip** channels. MDNs were trained with hidden layers containing 5 and 10 hidden units, and output probability density functions comprised of either a single Gaussian or a mixture of 2 Gaussians. A separate validation set was used during the training stage to identify when the network had reached an optimum. The negative log likelihood error of the target data in the unseen test set was then calculated for the optimal example of each network architecture combination. For comparison, the plots also show the negative log likelihood of the same test data set as calculated from the output of the best MLP from Chapter 4, represented as a horizontal line across the plot.

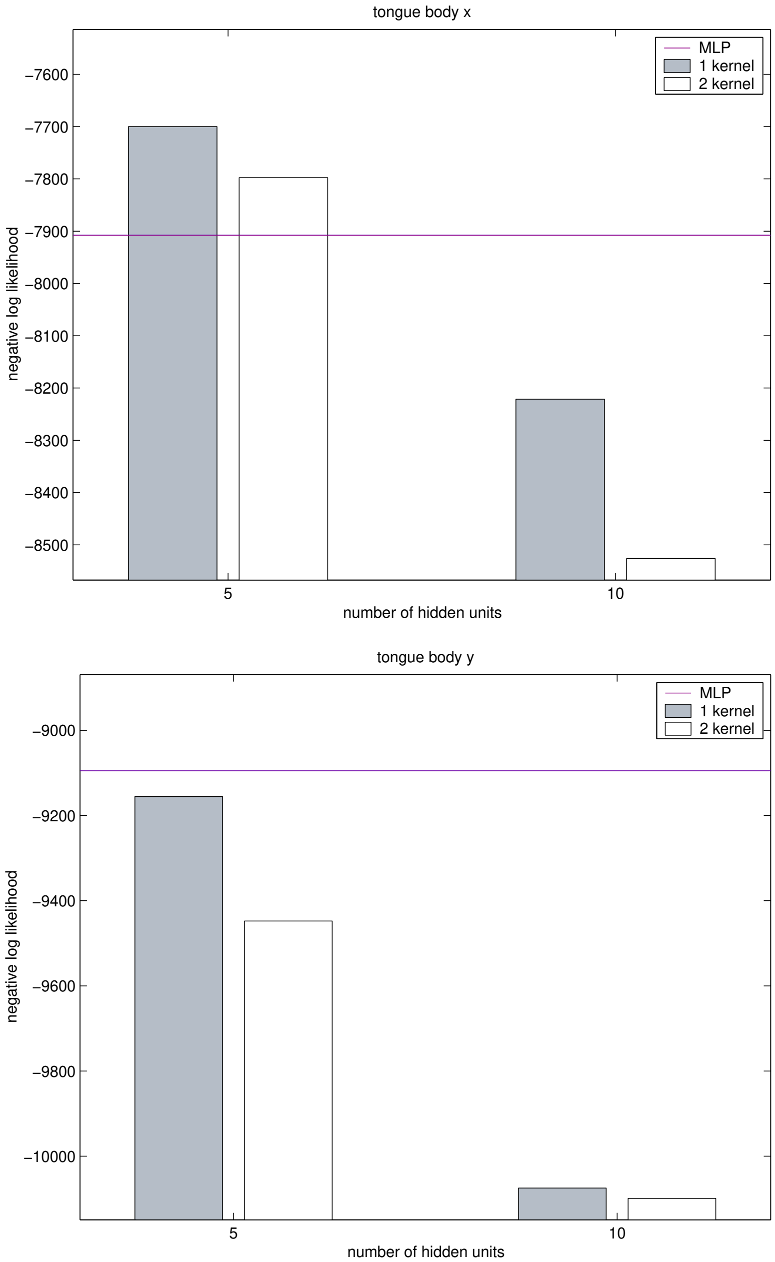
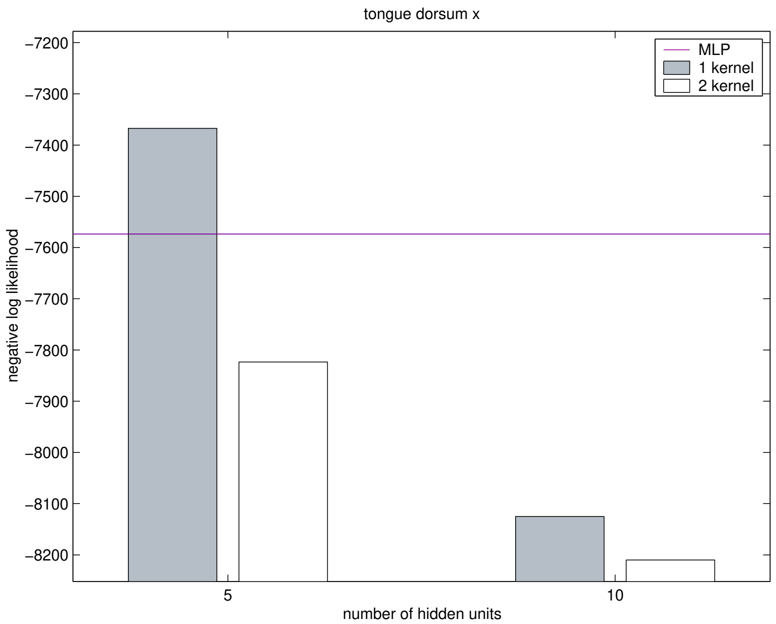


Figure 5.18: Comparison of MDN test set error for **tongue body** channels. MDNs were trained with hidden layers containing 5 and 10 hidden units, and output probability density functions comprised of either a single Gaussian or a mixture of 2 Gaussians. A separate validation set was used during the training stage to identify when the network had reached an optimum. The negative log likelihood error of the target data in the unseen test set was then calculated for the optimal example of each network architecture combination. For comparison, the plots also show the negative log likelihood of the same test data set as calculated from the output of the best MLP from Chapter 4, represented as a horizontal line across the plot.



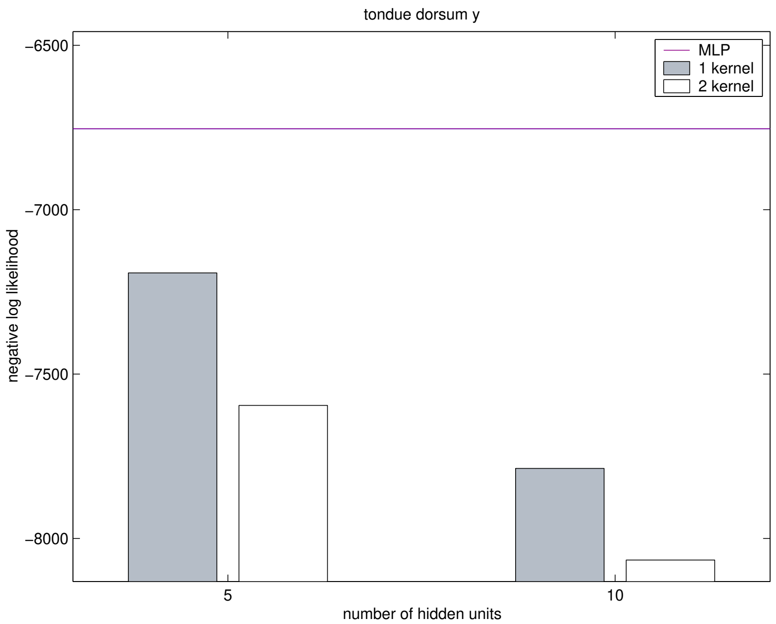


Figure 5.19: Comparison of MDN test set error for **tongue dorsum** channels. MDNs were trained with hidden layers containing 5 and 10 hidden units, and output probability density functions comprised of either a single Gaussian or a mixture of 2 Gaussians. A separate validation set was used during the training stage to identify when the network had reached an optimum. The negative log likelihood error of the target data in the unseen test set was then calculated for the optimal example of each network architecture combination. For comparison, the plots also show the negative log likelihood of the same test data set as calculated from the output of the best MLP from Chapter 4, represented as a horizontal line across the plot.

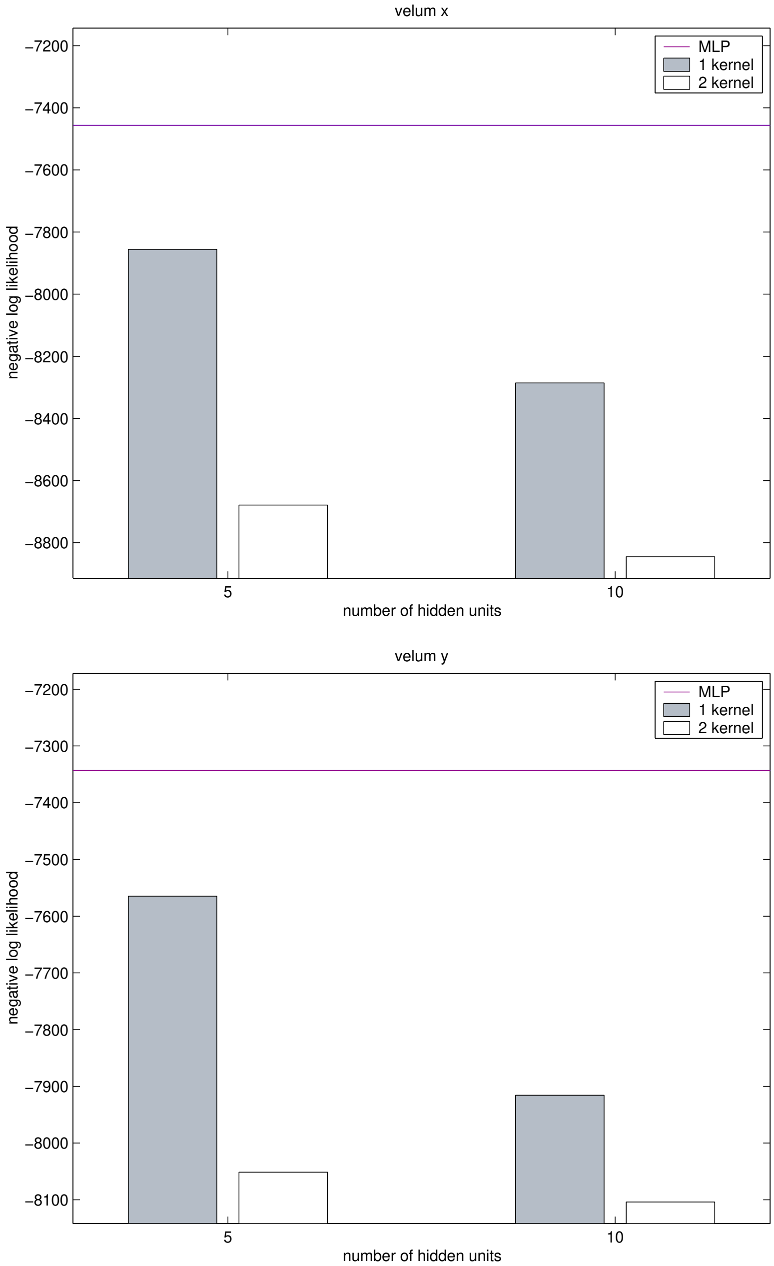


Figure 5.20: Comparison of MDN test set error for **velum** channels. MDNs were trained with hidden layers containing 5 and 10 hidden units, and output probability density functions comprised of either a single Gaussian or a mixture of 2 Gaussians. A separate validation set was used during the training stage to identify when the network had reached an optimum. The negative log likelihood error of the target data in the unseen test set was then calculated for the optimal example of each network architecture combination. For comparison, the plots also show the negative log likelihood of the same test data set as calculated from the output of the best MLP from Chapter 4, represented as a horizontal line across the plot.

5.4.3 Comparison with MLP performance

In order to gauge whether MDNs allow closer modelling of the acoustic-to-articulatory inversion mapping than when using MLPs, we require an error measure to directly compare like with like. The error function for the MDNs is the negative log likelihood of the target articulatory data given the probability density functions estimated on the basis of the acoustic input. In Section 5.1.1, we saw that training an MLP by minimising the sum-of-squares error function is equivalent to unimodal regression, where the mean is conditioned on the input vector, and with a single global variance. Assuming that the network has sufficient representational power and has been trained well enough to closely approximate the conditional mean of the target data, the remaining error is attributable to the variance of the target data itself around its (conditional) mean[3].

The global variance for each articulatory channel (σ_k^2) may be calculated using Equation 5.5, repeated here for the reader's convenience:

$$\sigma_k^2 = \frac{1}{N} \sum_{n=1}^{N} \{y_k(\mathbf{x}^n; \mathbf{w}^*) - t_k^n\}^2 \tag{5.16}$$

where N is the number of input-output vector pairs in the training set, $y_k(\mathbf{x}^n; \mathbf{w}^*)$ is the output of the trained MLP with the optimal weight configuration $\mathbf{w}^*$ in response to the input vector $\mathbf{x}$, and t is the target value.

Hence, at testing time, we can reinterpret the MLP output for each time frame as a single Gaussian probability density function, whose mean and variance are given by the output of the MLP and the pre-computed global variance. Given these probability density functions at each time frame, we can calculate the likelihood of the articulatory target data for the whole test set. This likelihood can be directly compared with the equivalent likelihood calculated according to the probability density functions provided by the MDNs.

Since training an MLP by attempting to minimise the sum-of-squares error function **is** in effect unimodal regression with a constant global variance[4], by directly comparing

[3] cf. the second term of Equation 5.3. The reader is referred to the discussion of Section 5.1.1 for more details

[4] as described in Section 5.1.1.

MLP likelihood results with those of the MDNs we may address two questions. First and foremost, we are effectively asking whether having the variance conditioned on the input (rather than constant, as in the MLP) can increase the likelihood of the target data, and so provide a more accurate model thereof. With this in mind, we should in particular compare the likelihood results of the MLP with those of the MDNs with a single Gaussian density function. Second, we may also obtain further indication as to whether having more than a unimodal density function over the target domain is beneficial in the case of the inversion mapping. This is similar to, and supplements, the comparison of likelihood scores from MDNs with single Gaussian and multimodal density functions.

The likelihood of the test set according to the output of the best MLP described in Chapter 4 has been calculated and is shown in Figures 5.14 to 5.20, represented as the horizontal line across the plot for each articulatory channel.

The Figures show that 7 of the 56 MDNs which were trained demonstrated a lower likelihood score for the test data set than the equivalent MLP estimated trajectory. However, 5 of these MDNs only contained 5 units in their hidden layer, which, as we postulated in Section 5.4.2, does not appear to be a sufficient number of units. In theory, an MDN of appropriate size and with a single Gaussian density function should do at least as well as the equivalent MLP. Hence, it is possible that the two networks from this batch which, despite containing 10 hidden units, performed worse that the MLP may not have reached the global minimum in the error surface. It is also possible that the MDNs need more than 10 hidden units to adequately model the articulatory channels in question.

For the remaining 9 out of the 14 articulator channels, the MDN with 5 hidden units and a single Gaussian probability density function performs better than the equivalent MLP. In all these cases, using a mixture of two Gaussians for the density function improves results further. It is the MDNs with two mixture density components and 10 hidden units though which lead to the most consistent and convincing gains in likelihood.

In order to assess the magnitude of this increase more clearly, Table 5.1 compares in detail the test set likelihood scores for the MLP with those for all the MDNs containing 10 hidden units and a mixture of 2 Gaussians. Columns 2 and 4 of Table 5.1 give the

Articulator channel	MLP		MDN		Increase in
	$-log(\mathcal{L})$	Av.$\mathcal{L}$	$-log(\mathcal{L})$	Av.$\mathcal{L}$	Av.$\mathcal{L}$(%)
ul_x	-3453	1.345	-3950	1.403	4.4
ul_y	-6142	1.693	-7062	1.832	8.2
ll_x	-4850	1.516	-5145	1.555	2.6
ll_y	-6528	1.750	-7362	1.880	7.4
li_x	-4681	1.494	-5278	1.572	5.3
li_y	-8290	2.036	-10589	2.480	21.8
tt_x	-7503	1.903	-8112	2.005	5.4
tt_y	-8689	2.107	-10949	2.557	21.4
tb_x	-7908	1.970	-8526	2.078	5.4
tb_y	-9095	2.182	-10099	2.378	9.0
td_x	-7574	1.915	-8210	2.022	5.6
td_y	-6754	1.785	-8065	1.997	11.9
v_x	-7456	1.895	-8845	2.135	12.7
v_y	-7343	1.877	-8104	2.004	6.7

Table 5.1: A comparison of the likelihood of the test set target data given the framewise probability density functions provided by MLP and MDN. Here, 14 channel specific MDNs have been used with 10 units in one hidden layer and an output mixture density model comprising two Gaussian kernels.

overall negative log likelihood of the articulatory target data in the test set, given the framewise probability density functions computed from the MLP and MDN networks respectively. Columns 3 and 5 give this error expressed as the 'average' likelihood for each frame of speech in the test set[5]. This is calculated by dividing the total log likelihood by the number of input-output pairs in the test set, then transferring back out of the log domain. Using the results from columns 3 and 5, we can calculate the relative improvement of the MDN model over the MLP model as a percentage. This increase in modelling accuracy is shown in column 6 of Table 5.1.

As the last column of Table 5.1 makes clear, using the MDN to model the acoustic-to-articulation inversion mapping yields an improvement for all articulatory channels. The size of this improvement ranges from between 2.6% to 21.8%. Apart from for the velum, it would appear that the best improvement is observed for the y-coordinates of the articulators. Excluding the velum channels, the average improvement for the articulator y-coordinates is about 13.3%, whereas the average for the x-coordinates is only approximately 4.8%. In future work, it would be interesting to investigate what

[5]There are 11660 acoustic-articulatory feature vector pairs in the test set

might be the reason for this disparity. For example, it may be this is merely coincidence, or an idiosyncrasy of this particular speaker. On the other hand, it could be a more general observation. Experiments using data from multiple speakers would be useful to investigate this.

5.5 Discussion

So far in this chapter, we have established that MDNs improve modelling accuracy of the acoustic-to-articulatory inversion mapping in terms of an error measure based on the likelihood of the target articulatory data. However, little attention has been paid to what form the "answer" that the MDN gives us takes. In the last two sections of this chapter, we present the output of the MDN for visual inspection, and then suggest how one of the immediate intuitions with respect to MDN output might find useful application.

5.5.1 MDN probabilitygram

Figures 5.21 to 5.27 provide a visual representation of the output of the MDNs. These plots, which might loosely be termed "articulatory probabilitygrams", show the probability density for the location of an articulator within its range of movement as a function of time. Such plots are produced by taking the probability density function output by the MDN at each time frame (x-axis) and calculating the probability density at certain intervals encompassing the range of movement of the given variable (y-axis)[6]. The probability density for a particular articulator location at a given time frame is represented by the greyscale intensity. Intense black indicates high probability density, whereas white indicates low probability density. For all the Figures 5.21 to 5.27, the greyscale is fixed and equal, in order to make it easier to directly compare the relative probability densities between different plots. In other words, full black on one plot represents the same probability density as full black on all the other plots.

As well as showing the output of the MDNs, the real, measured EMA trajectories are shown for comparison. Phonetic segmentation is also indicated in these plots. This

[6]Figures 5.21 to 5.27 use the range from -0.2 to 1.2 in steps of 0.01

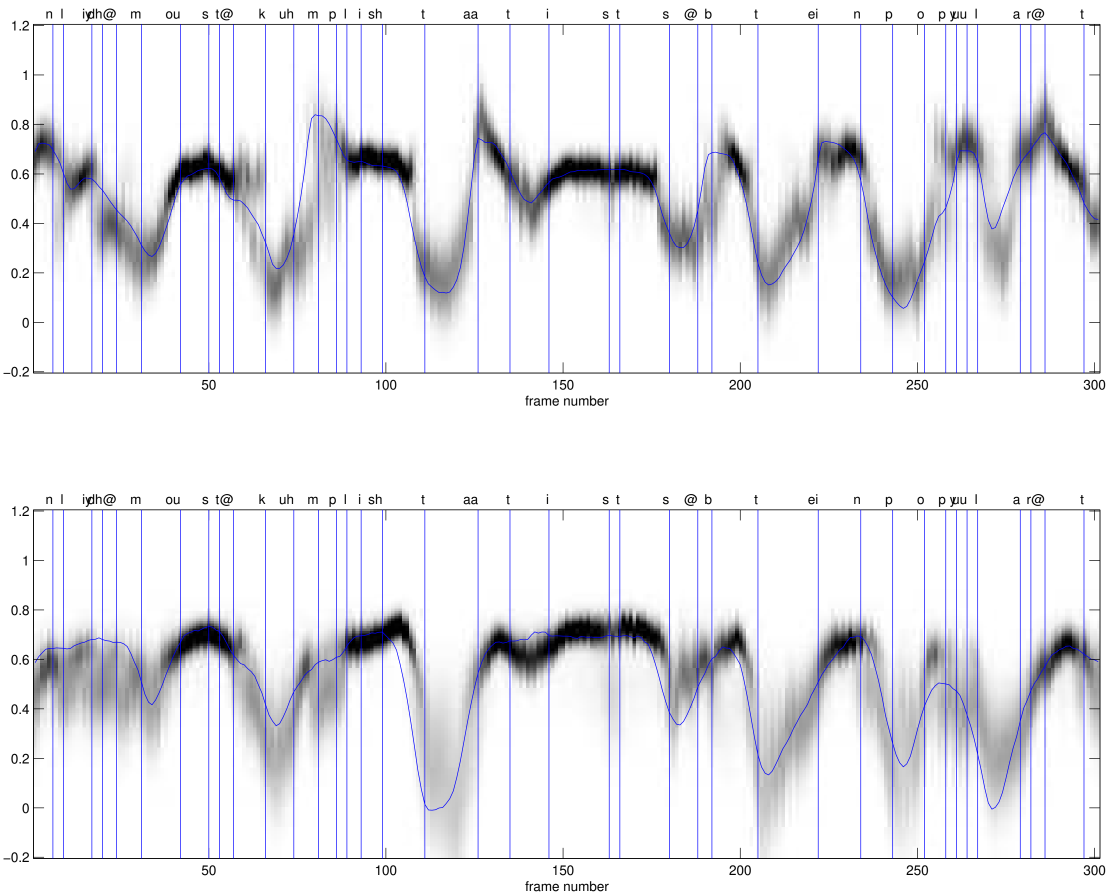


Figure 5.21: Comparison of MDN output with the actual, measured articulator trajectories (continuous line) for the unseen test set utterance "Only the most accomplished artists obtain popularity". The plots represent probability density over the range of an articulator's movement as a function of time using greyscale intensity, where intense black indicates high probability density. The phone labels mark the respective phone end times. The top plot shows the output for the `tt_y` channel, while the bottom plot shows the output for the `li_y` channel.

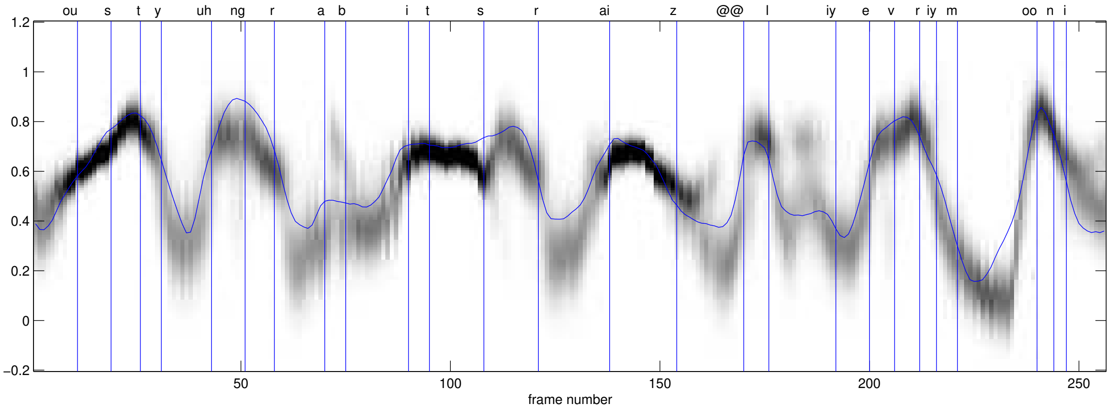


Figure 5.22: Comparison of MDN output with the actual, measured articulator trajectory (continuous line) for the unseen test set utterance "Most young rabbits rise early every morning". The `tt_y` articulatory channel is shown in this example. Probability density over the range of the articulator's movement is shown as a function of time using greyscale intensity, where intense black indicates high probability density. The phone labels mark the respective phone end times.

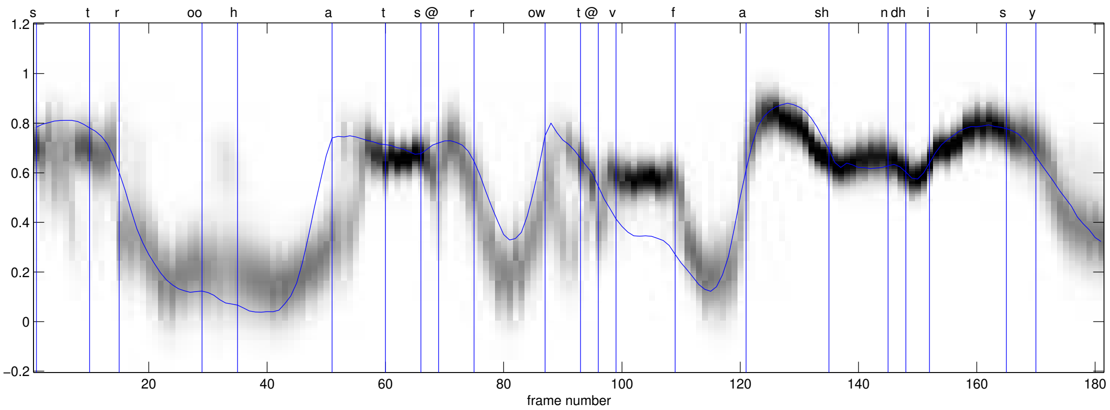


Figure 5.23: Comparison of MDN output with the actual, measured articulator trajectory (continuous line) for the unseen test utterance "Straw hats are out of fashion this year". The `tt_y` articulatory channel is shown in this example. Probability density over the range of the articulator's movement is shown as a function of time using greyscale intensity, where intense black indicates high probability density. The phone labels mark the respective phone end times.

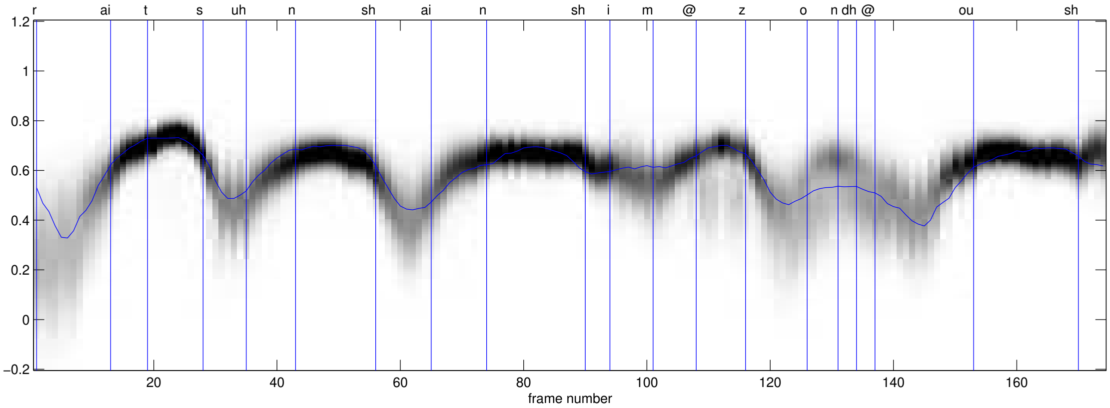


Figure 5.24: Comparison of MDN output with the actual, measured articulator trajectory (continuous line) for the unseen test utterance "Bright sunshine shimmers on the ocean". The `li_y` articulatory channel is shown in this example. Probability density over the range of the articulator's movement is shown as a function of time using greyscale intensity, where intense black indicates high probability density. The phone labels mark the respective phone end times.

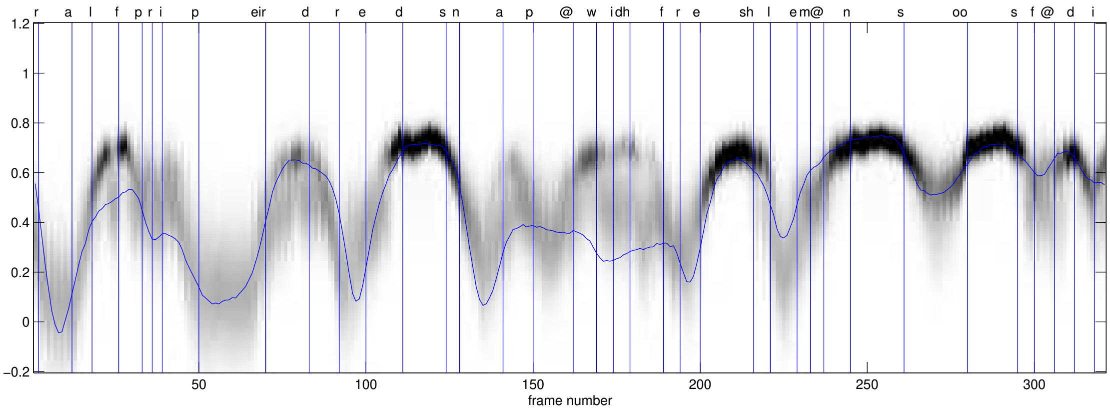


Figure 5.25: Comparison of MDN output with the actual, measured articulator trajectory (continuous line) for the unseen test utterance "Ralph prepared red snapper with fresh lemon sauce for dinner". The `li_y` articulatory channel is shown in this example. Probability density over the range of the articulator's movement is shown as a function of time using greyscale intensity, where intense black indicates high probability density. The phone labels mark the respective phone end times.

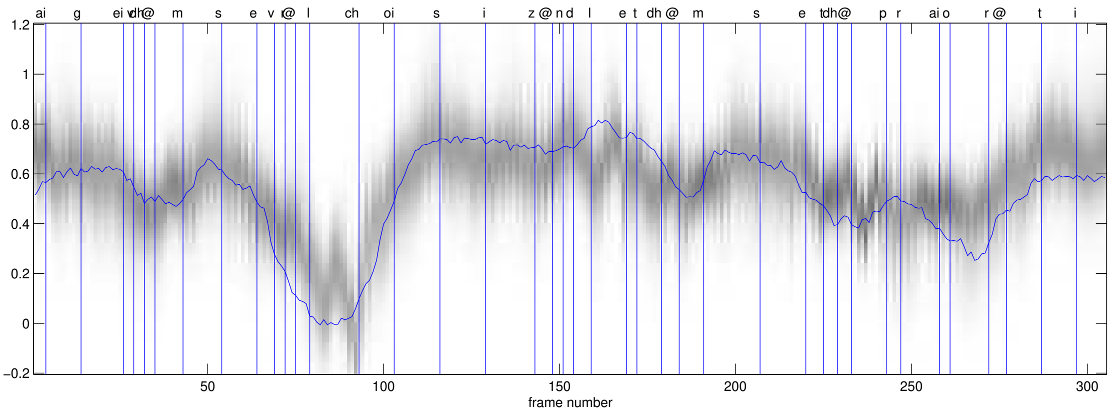


Figure 5.26: Comparison of MDN output with the actual, measured articulator trajectory (continuous line) for the unseen test utterance “I gave them several choices and let them set the priorities”. The `ul_x` articulatory channel is shown in this example. Probability density over the range of the articulator’s movement is shown as a function of time using greyscale intensity, where intense black indicates high probability density. The phone labels mark the respective phone end times.

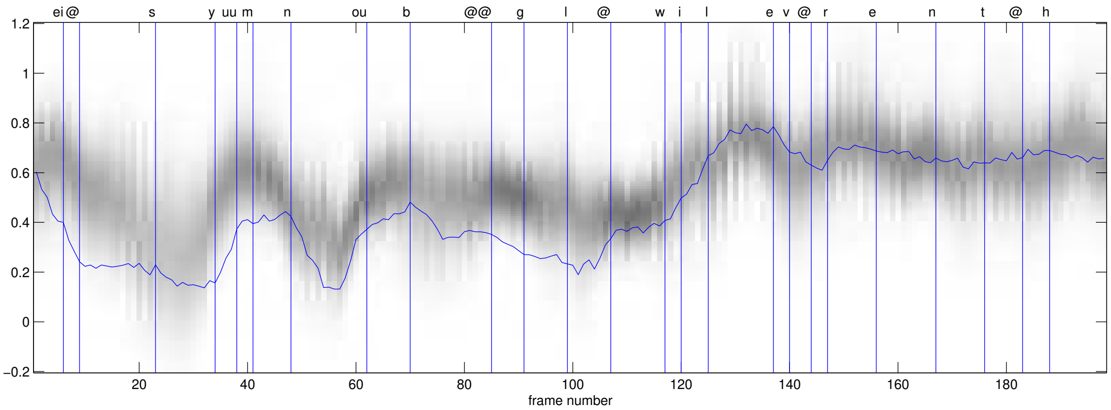


Figure 5.27: Comparison of MDN output with the actual, measured articulator trajectory (continuous line) for the unseen test utterance “They assume no burglar will ever enter here”. The `ul_x` articulatory channel is shown in this example. Probability density over the range of the articulator’s movement is shown as a function of time using greyscale intensity, where intense black indicates high probability density. The phone labels mark the respective phone end times.

labelling is provided as part of the standard MOCHA database, and was produced using an HMM forced alignment. A vertical line indicates the end time boundary of the associated phone label.

As made clear in Table 5.1, the MDN is able to model the `tt_y` channel with the highest likelihood, closely followed by the `li_y` channel. Figure 5.21 shows MDN output for these two articulatory channels for the same utterance. It is interesting to compare these two channels together. Generally, the tongue tip y and lower incisor y movements are highly similar. This is to be expected, as the tongue is attached to the lower jaw, so unless the tongue tip is itself moving relative to the lower jaw, any movement by the jaw results in the same pattern of movement in the tongue tip. However, there are differences. For example, for the first few phones of this utterance, the lower jaw does not move much in the y-axis. This contrasts with the tongue tip in the y-direction, which seems to be "active" in articulation.

It seems to be the case that for both these channels, the MDNs estimate the position of the articulators with a lower variance when they are near their upper bound of movement, which is of course enforced by the upper, passive articulators. Thus, variance tends to be lower during phones such as [n, t, d, s, sh] and so on, where the tongue tip is critical to the production of the segment. This impression is corroborated in the additional MDN output samples for these articulator channels shown in Figures 5.22 to 5.25. However, although estimates for the locations of both articulators demonstrate higher variance when they are further away from from their upper bound, there are differences within this common trend. For example, during the [o] in the word "popularity", the position of the tongue tip in the y-axis is estimated with substantially lower variance than the position of the lower incisor in the same axis.

Figures 5.22 and 5.23 show evidence of multimodality in the inverse mapping from acoustics to articulation. This is potentially a most significant observation, supporting the theoretical arguments and empirical evidence detailed in Chapter 2 that the inverse mapping is ill-posed. In Figure 5.23, we observe bimodality for the [h] phone of the word "hats". In Figure 5.22, a well-defined instance of bimodality occurs in the [b] phone of the word "rabbits", as well as during the [iy] phone of the word "early". In order to demonstrate the existence of bimodality in the MDN output as clearly as possible, Figure 5.28 presents the probability density function at frame 183 for utterance `fsew0_026` shown in Figure 5.22. In Section 5.4.1, it was noted that using a mixture of two Gaussians in the MDNs improved the likelihood scores. Finding instances of multimodality such as these in the MDN output both in part explains this and justifies the use of mixtures of multiple Gaussians in modelling the inversion mapping.

Finally, Figures 5.26 and 5.27 provide sample output during unseen test set utterances for the `ul_x` channel . This channel proved to be the worst in terms of likelihood for both MLP and MDN alike. This fact is evident in Figures 5.26 and 5.27 as a high variance for most frames. Nevertheless, the MDN was still able to increase the likelihood score of the target data over the whole test set by over 4% compared to the MLP.

5.5.2 Correlation of accuracy and variance

Examining probabilitygrams of MDN output, such as those in Figures 5.21 to 5.27, we notice a correlation: when the MDN output variance is low, the accuracy of the estimated location of the articulator is typically higher. This correlation is unsurprising, since the variance of the probability density functions output by the MDN is conditioned on the acoustic input so as to minimise the negative log likelihood error function.

It is natural to wonder whether we might be able to interpret this variance as some sort of confidence measure pertaining to the estimate of the articulator locations. For example, in a MDN with a single Gaussian, the mean gives the expected location of the articulator, while the variance around that mean gives a measure of the confidence we might choose to ascribe to that estimate. If the variance around the estimated mean were low, then we might view this as signifying we can place high confidence in this estimate. Conversely, if the variance around the mean were high, then we would presumably choose not to place as much confidence in the estimate.

How this confidence measure might be treated depends on the specific application. However, it must be noted that we should probably prefer to view the variance parameter as an indication of the MDN's confidence in its estimation of the location of an articulator, rather than directly interpret it as our confidence in that estimate. The MDN is sometimes quite wrong in interesting ways. For example, in Figure 5.23 the

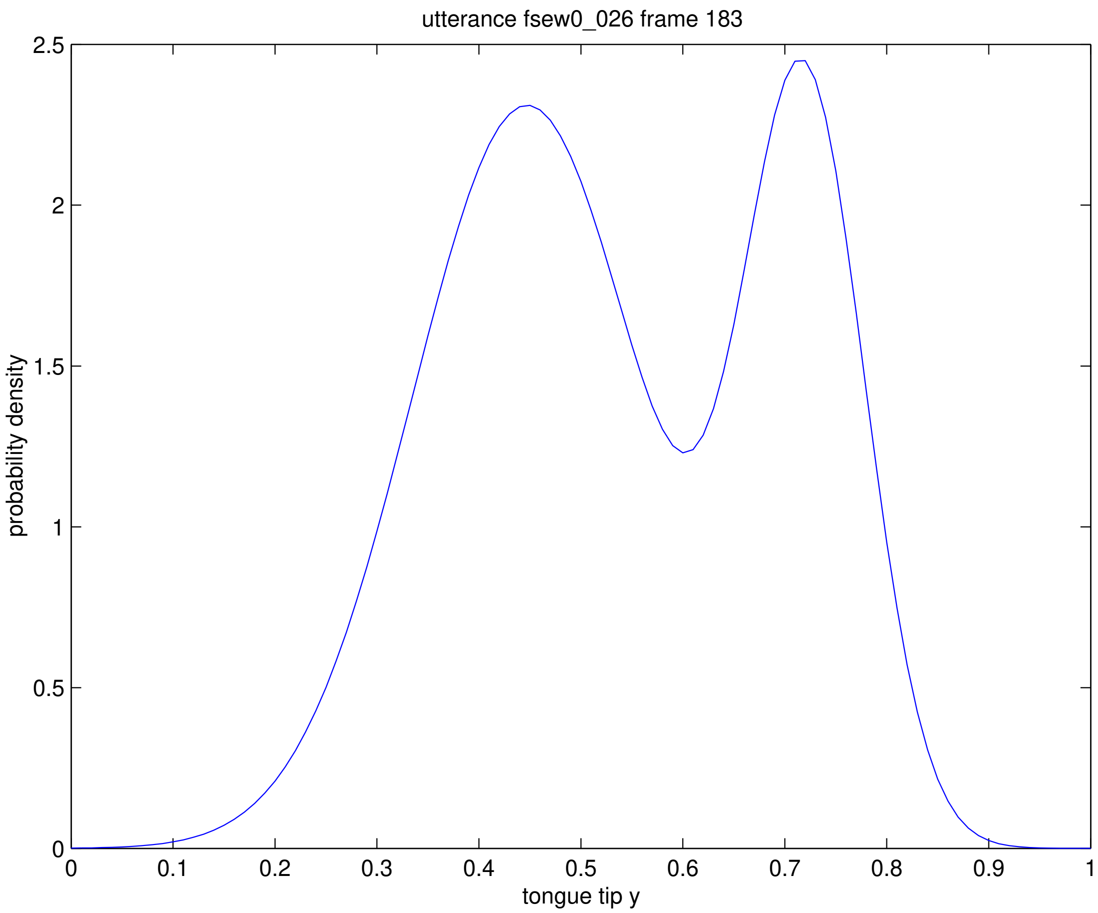


Figure 5.28: The probability density function output by the MDN (containing 10 hidden units and a mixture density model of two Gaussian components) for channel `tt_y` at frame 183 of utterance `fsew0_026` (as shown in Figure 5.22). This frame occurs as part of the [iy] phone of the word "early". The bimodality of the density function output by the network at this frame is pronounced.

MDN estimates a low variance around its estimate of the tongue tip position in the y-axis during the [f] phone within the word “fashion”. However, the actual location of the tongue tip during this phone is much lower in the space. It looks suspiciously as though the MDN has misinterpreted the acoustic energy of the [f] phone for that of a dental fricative; a mistake which is easily made by humans.

Chapter 6

Conclusion

6.1 Contributions

One of the primary aims of the work undertaken for this thesis was to extend the work previously carried out by others to develop a practical method for estimating articulatory parameters from acoustics alone. Attempting to model the inversion mapping using measured human articulatory data in significantly larger quantities than has been available and used before represents a major extension. This dataset was designed to incorporate as much phonetic diversity as possible. In addition, the inversion mapping has been attempted for a more comprehensive set of target articulators than in the majority of efforts previously reported.

Using this data set, we have demonstrated how it is possible to train an MLP to perform the inversion mapping for continuous, phonetically rich speech. In addition to benefit of certain advantages of the MLP over other inversion methods which we have mentioned, the accuracy of the estimated articulatory trajectories in terms of RMS error has been shown to compare favourably to other models previously described in the literature. We have also shown that the accuracy of these MLP-estimated articulatory trajectories can be moderately improved by a lowpass filtering method, which imposes a kind of rudimentary articulatory constraint on the MLP output.

In order to look for ways to make the neural network inversion mapping more accurate and useful, we then undertook a critical examination of what exactly the MLP is capable of and how the characteristics of the MLP manifest themselves when applied to the acoustic-to-articulatory inversion mapping. Specifically, we considered the effect of non-uniqueness in the inversion mapping. Although the prevalence of one-to-many mappings has not been well quantified, we suspected this could be a prime source for error in the MLP inversion mapping. Theory indicates that the MLP is incapable of dealing adequately with such instances of non-uniqueness. Therefore, we applied the mixture density network to the acoustic-to-articulatory inversion mapping, as in theory it is better placed to model one-to-many mappings. In essence, the mixture density network provides a principled method for modelling a full conditional probability density of the target data for each input vector. A mixture density network with adequate and appropriate architecture can represent arbitrary conditional probability distributions in the same way that a conventional neural network can represent arbitrary functions.

In due course it was shown that the MDN does indeed allow closer, more accurate modelling of target articulatory domain than is possible with the MLP. This was demonstrated by reinterpreting the error of the MLP-estimated trajectories in terms of the likelihood of the actual articulatory trajectories, and comparing this likelihood with that obtained using the probability density functions output by the MDNs. The difference in average likelihood between the MLP and MDN ranged from 2.6% to 21.8%, depending on the articulatory channel in question.

6.2 Future work

While this thesis has shown that the MDN provides a more accurate model of the target articulatory domain when performing the acoustic-to-articulatory inversion mapping, we have not considered how the output of the MDN may be used in applications. Therefore, one major direction for future research is to evaluate how best to use the MDN output. Although the details may ultimately depend upon a specific application, this section discusses several suggestions for how the MDN output might be exploited. We then consider other aspects of future work which are envisaged to be useful or interesting.

6.2.1 The MDN "answer"

In many senses, MDNs occupy some ideal middle ground between supervised and unsupervised empirical learning methods. They are trained in a way typical of supervised methods. However, at test time, there are similarities with unsupervised techniques. At each time frame, the MDN gives a whole description of the target domain in the form of a probability density function. The probability density functions output by the MDN may be viewed as equivalent to directly modelling the "fibers" explored by Atal et al. (1978), or regions in the articulatory space which map to a single point in acoustic space. However, the MDN provides the additional benefit of doing this in a probabilistic way. Thus, whereas Atal et al. (1978) explored only the *extent* of a fiber in articulatory space, the MDN gives information pertaining to how likely it is that each part of a fiber will be used to produce a given acoustic output. This density function may in turn be used to compute a number of "answers", depending on the requirements of the application. From this point of view, the "answer" is postponed until testing time, and as such the MDN offers a powerful modelling advantage over other supervised learning models.

In Section 5.5, it was suggested it could be possible to use the variance parameters output by a MDN as a measure of confidence in the estimate of articulator location given by the mean parameters. In the following sections, we elaborate on a few more ways in which we might use the output from a mixture density network trained to estimate articulatory parameters from the acoustic speech signal. These suggestions could be described as falling into two distinct categories of motivating philosophy. On one hand, we might be seeking to decide upon a single "best" trajectory given the MDN output. On the other, we might aim to use the probability density function over the complete target domain as a whole.

Conditional mean trajectory

Given a conditional density function in the form of a Gaussian mixture model, it is straightforward to calculate the conditional mean of the target data. In this way, the MDN answer can be made to approximate the output of a standard least-squares trained MLP as a special case. The conditional mean is calculated as follows:

$$\langle \mathbf{t}|\mathbf{x}\rangle = \sum_j \alpha_j(\mathbf{x})\boldsymbol{\mu}_j(\mathbf{x}) \tag{6.1}$$

where $\alpha_j(\mathbf{x})$ is the mixing coefficient, or prior, for the j'th mixture component, and $\boldsymbol{\mu}_j(\mathbf{x})$ is the vector of means (of length equal to the dimensionality of the target domain) given by the j'th mixture component.

In the least-squares trained MLP, the variance around this conditional mean is fixed across the training set. In contrast, for the probability density functions output by the MDN, the variance too is conditioned on the input $\mathbf{x}$. The fact that the variance can vary according to the input is an improvement over the capability of the MLP. At each time frame, the variance of the density function around the conditional mean can be calculated as follows:

$$\sum_j \alpha_j(\mathbf{x}) \left\{ \sigma_j(\mathbf{x})^2 + \left\| \boldsymbol{\mu}_j(\mathbf{x}) - \sum_l \alpha_l(\mathbf{x})\boldsymbol{\mu}_l(\mathbf{x}) \right\|^2 \right\} \tag{6.2}$$

where both j and l range from 1 to M mixture components present in the mixture model.

Mode trajectory

Section 5.5 presented evidence for the existence of multimodal distributions in the articulatory target data for a given acoustic input. The single conditional mean calculated according to Equation 6.1 is unlikely to provide a satisfactory answer in the event of multimodality. This conditional mean may not actually be located at the centre of a region of high probability mass, no doubt accompanied by a high variance around the conditional mean. Probably the simplest way of assuring that the trajectory avoids this problem and always passes through areas of high probability density it to use the mean of the mixture component at each time frame with the highest prior. This can be termed the "mode" trajectory, and approximates the mixture-of-experts model of Jacobs, Jordan, Nowlan & Hinton (1991) as another special case.

MDN output decoding

If we take the mixture density components to represent potentially different branches of the solution of the inverse mapping function, then we might contemplate using a more sophisticated method than merely choosing the mean of the component with the highest prior at each time frame. It may prove beneficial to choose between the solutions on the basis of some continuity constraints. For example, it would presumably be better to choose the mean point of a mixture density component which does not have the highest prior at time t if it is closer to neighbouring points at time $t-1$ and $t+1$ than all other mixture components. In other words, given a sequence of probability density functions output by the MDN at each time frame, we would be aiming to "decode" the most likely sequence on the basis of additional continuity constraints, motivated by prior knowledge of human articulation.

Such decoding would exhibit close parallels to the methods discussed in Section 2.4.1. For example, Rahim et al. (1991), who used a dynamic programming search through the output of 128 networks trained on separate regions of the vocal tract parameter space, reported good results. They used a cost function in both the acoustic[1] and articulatory domains, whereby the optimum trajectory in vocal tract parameter space was judged by a criterion requiring vocal tract shapes to vary as smoothly as possible.

Given the output of a mixture density network, a dynamic search with at least the level of constraint embodied in the articulatory cost function used by Rahim et al. (1991) would be straightforward. There are additional advantages to be had though when using mixture density networks in this situation. For example, in the mixture density network, the division of the mapping space between different kernel components, or local "experts", is completely automatic; the priors for the mixture components are derived along with the other mixture component parameters by the optimisation process. Rahim et al. (1991), on the other hand, had to divide the mapping space up in a process separate from the network training, and one which could be suboptimal.

Another potential advantage of using mixture density networks arises from the fact

[1]In their application of acoustic-to-articulatory inversion, the output of their assembly of networks was used as control parameters for an articulatory synthesiser.

that not only do they provide point estimates for the articulator positions, they also give a variance around those points. In essence, the mixture density network can model the spread of points around a mean value in the target domain. It is not difficult to imagine how such information might be exploited within the dynamic programming search for the smoothest, most "plausible" trajectory in articulatory space. For example, consider the situation where the best mode point is more than a reasonable distance from its neighbouring point, such that it would require the given articulator to move with uncharacteristic velocity or acceleration. If the variance around this point were high, we might choose to "attenuate" the movement of the articulator towards that point in accordance with our knowledge of human articulator movements.

Probability density function distance

Where it is not necessary to have a single "best-guess" trajectory through the articulatory domain, it could be possible to use the density function as a whole. There are several measures of distance between two probability density functions, such as Hellinger distance, quadratic distance, variation distance, but the Kullback-Leibler distance metric (Kullback & Leibler 1951) is probably the most well-known, given as:

$$KL(p;\tilde{p}) = -\int p(\mathbf{x}) \ln \frac{\tilde{p}(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x} \tag{6.3}$$

where $p(x)$ is the known, true density function and $\tilde{p}(x)$ is the second density function which we want to compare with the first[2].

A simple experiment which would be interesting to try is a phone classification test. Phone-specific Gaussian mixture models could be fitted to the EMA feature vector at the centre of all instances of each phone in the training set. These mixture models would constitute rudimentary phone templates expressed in the articulatory domain. The acoustics from the centre of a known phone in the test set could be presented to the mixture density network to obtain a "mystery" probability density function over the location of the articulators in the articulatory domain. Using the density function distance metric, it would be possible to rank the phone templates, and hence identify

[2] This relationship is important, as the Kullback-Leibler distance is an asymmetric distance measure

that template to which the mystery density function is closest. In this way, it would be possible to test the overall phone classification accuracy observed for all phones in the test set. It would be interesting to compare the phone classification accuracy using this method with that observed using just a point estimate mystery articulatory configuration, based on the highest likelihood given each phone template in turn. Furthermore, it would also be interesting to compare these results with those observed using the same technique with single points in the acoustic domain.

For this experiment to be practical, we require a reasonable way to calculate the Kullback-Leibler distance. An analytical solution exists for calculating Kullback-Leibler divergence between two multivariate Gaussians, but things become more awkward with mixture models[3].

One solution which fits into the requirements of the proposed experiment, would be to use a method that produces a ranking of the templates which approximates that obtained using the Kullback-Leibler divergence, but which does not require explicit calculation of the distance metric. This method entails taking n samples from the mystery mixture model, and evaluating the average log likelihood of the sample under each of the template density functions. Picking the template with the highest log likelihood implies the minimum approximate Kullback-Leibler divergence. This gives the right result as the sample size $n \to \infty$.

The justification for this approximation comes from the fact that Kullback-Leibler divergence is actually based on the expected negative log likelihood of infinite data samples drawn from the true distribution $p(\mathbf{x})$ under the model distribution $\tilde{p}(\mathbf{x})$ (Bishop (1995), pp. 59):

$$
\begin{aligned}
E[-\ln \mathcal{L}] &= -\lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} \ln \tilde{p}(\mathbf{x}^n) &\quad (6.4)\\
&= -\int p(\mathbf{x}) \ln \tilde{p}(\mathbf{x})\, d\mathbf{x} &\quad (6.5)
\end{aligned}
$$

When $p(\mathbf{x})$ and $\tilde{p}(\mathbf{x})$ are the same distribution, then the expected negative log likelihood given in Equation 6.5 has a residual value given as

[3] This, and the solution, personal communication with Chris Williams, Edinburgh University

$$-\int p(\mathbf{x}) \ln p(\mathbf{x})\, d\mathbf{x} \tag{6.6}$$

which is actually the entropy of the density function $p(\mathbf{x})$. In the Kullback-Leibler distance metric, this residual is conveniently subtracted from the measure given in Equation 6.5, so that $KL \geq 0$. However, if $p(\mathbf{x})$ is the mystery pdf output by the MDN, for the purpose of ranking the template density functions, the entropy term is fixed and so may be ignored. Therefore, the maximum expected log likelihood minimises KL divergence.

6.2.2 System optimisation

MDN architecture

While reasonable care was taken with the MLP to find a suitable network topology by means of a pruning algorithm, the same cannot be said for the experiment with the MDNs. At this preliminary stage, the emphasis has rested on showing that the MDN offers clear modelling advantages over the MLP, rather than trying to push the MDN performance as far as possible.

However, on the basis of the two trends identified in Sections 5.4.1 and 5.4.2, it does seem promising that using MDNs with more than 10 units in the hidden layer and more than 2 Gaussians in the mixture density model could lead to even better performance. Therefore, it would certainly be a useful task in the future to perform a more exhaustive search for the best MDN architecture for estimating articulatory parameters from acoustics, whether by using some pruning or constructive meta algorithm, or by simply training and comparing large numbers of networks.

Bidirectional recurrent mixture density networks

Schuster (1999) has described an extension of the mixture density network where the neural network which provides the mixture density parameters is augmented to use information from states of the hidden layer at preceding and following time steps. These are termed bidirectional recurrent mixture density networks (BDRMDN). In principle, using information from neighbouring time frames in this way has the potential to disambiguate instantaneous non-uniqueness in the acoustic-to-articulatory inversion mapping to some degree. However, at this stage, it is unclear whether this will ultimately result in better performance. On one hand, pilot studies carried out during the period of study for this PhD have indicated that the probability density plots produced from the output of a BDRMDN are "smoother" than the equivalent output of an MDN. On the other hand, as noted in Section 4.7.1, the benefit of adding recurrency in the case of the MLP has not yet been clearly demonstrated. For the current thesis, the emphasis has been placed on evaluating what benefit results from using a more sophisticated model of the target articulatory domain when performing the inversion mapping. In future work, however, it will be useful to examine and quantify what additional modelling advantage might be obtained by using networks with links to the states of units at other points in time.

Input coding

In this thesis, the acoustic speech signal has been coded for input to the neural networks using filterbank analysis, with a context window of approximately 200ms. This is similar to the acoustic input representation used by Papcun et al. (1992). However, other researchers have used different acoustic feature extraction methods. For example, Hogden et al. (1996) used cepstrum-smoothed spectra. In addition, the choice of context window has not been universal; the length of context window has varied, and in fact some researchers have not used a context window at all. It is not clear what effect the choice of acoustic feature representation and context has on the accuracy of estimating articulatory parameters from the acoustic speech signal.

For the purposes of this thesis, the aim was to compare MLP and MDN on exactly the same task, and so searching for the best acoustic input format was irrelevant. In future work, however, it will be useful to ascertain what is the most suitable acoustic feature extraction algorithm when attempting to estimate articulation from the speech waveform, and what sort of context window gives the best results.

6.2.3 Multispeaker inversion mapping

In order to isolate one manageable subsection from the overall problem of estimating articulatory parameters from acoustics, this thesis has deliberately postponed working with speech from multiple speakers. Many additional complications to performing the inversion mapping for multiple speakers are envisaged. Not least of these is the question of normalisation of the EMA data between speakers. However, for an inversion system to be of practical benefit, these questions must at some stage be tackled.

An interesting aspect of multiple speaker acoustic-to-articulatory inversion to research would be how to handle speakers for whom there is no articulatory training data. With purely acoustics-based empirical learning systems, such as a speech recogniser, it is possible to take a generic pre-trained system and adapt it to an individual by further training on data the user provides. However, in the case of using articulatory parameters, it is not really practical for the end user to record EMA data of their speech for further training of an inversion mapping system. It would be interesting to investigate whether there are nevertheless any ways in which a generic inversion mapping could be tailored to an individual without the use of further EMA data.

In some applications of an inversion mapping it may not be necessary to explicitly deal with normalisation issues between speakers. In fact, it may not be necessary to deal ultimately with explicit and faithful articulatory trajectories at all. This could be true in a potential extension of an LDM-based speech recognition system (Frankel & King 2001*a*), where we might choose to abstract away from estimating real articulatory trajectories in favour of estimating some idealised "pseudo-articulation" on the basis of Maximum Likelihood principles. In order for the network to estimate such pseudo-articulatory trajectories would require embedded training within the recogniser. After initialising the network by training on real articulatory data, further embedded training could be carried out using the error from the LDM observation space compared to the network output in a Maximum Likelihood way. The network would then just become an efficient and well matched non-linear "feature extraction" interface to the acoustic domain for the LDM. If successful, the articulatory normalisation issue will have been circumvented, since the articulation will have effectively become a hidden variable, estimated by training using Maximum Likelihood instead of articulation from different speakers in a supervised way.

In some respects, this would be similar to the MO-MALCOM system of Hogden & Valdez (2000). They claim that MO-MALCOM paths through the continuity map correlate highly with measured articulatory trajectories. Even more interestingly, they claim finding that the maximum likelihood inferred trajectories through the continuity map are even better discriminators between phones than the measured articulatory trajectories.

Finally, it is worth noting the MDN could be an especially useful and interesting tool for exploring the inversion mapping for multiple speakers. The fact that the MDN outputs a full conditional probability density function at each time frame allows more insight into the form of the inversion mapping for multiple speakers, revealing how the network handles different articulatory targets from different speakers.

6.2.4 Articulatory parameterisation

Throughout this thesis, the raw articulatory measurements provided by the EMA recording have been used as the description of the vocal tract, forming a 14-Dimensional vector. However, there is no reason to believe this is constitutes either a suitable or even an adequate description of the vocal tract. For example, certain EMA channels seem quite highly correlated, and it would certainly not be surprising if a more parsimonious description of the vocal tract would suffice.

Roweis (1999) investigated the underlying embedded space, or manifold, which the articulators actually occupy in articulatory space. Such a manifold exists as a result of two types of constraint on the location of the articulators. First, certain articulatory configurations are physiologically impossible, and therefore data points are never found in those parts of the articulatory space. Second, certain articulatory configurations are rarely, if ever, used. Such regions in the articulatory space will be to a greater or lesser extent sparsely populated. Using a corpus of X-ray microbeam data comprised of the x- and y-coordinates of 8 gold pellets placed on the articulators, Roweis (1999) found that the manifold of articulation observed therein could be transformed and adequately described (i.e. reconstruction error back to the 16-dimensional articulatory-bead space is negligible) in a space having a dimensionality of between four and six. Linear and non-linear principle component analysis was used to derive this estimate.

Meanwhile, even though the EMA feature representation might exhibit redundancy, it does not rule out the possibility that it also does not fully represent the articulatory configurations. It may prove useful to consider whether there are any benefits in employing different articulatory measurements (such as EPG), or encodings of the articulatory data. Such undertakings are likely to be motivated by the specific application of the inversion mapping system.

6.2.5 Speech recognition

One of the main motivations for attempting to estimate articulatory parameters from the acoustic speech signal is the potential for improving speech recognition performance. We saw at the end of Chapter 4 that the performance of the LDM-based phone classifier did not show the same increase when using articulatory parameters estimated from acoustics as when using directly measured articulatory parameters. Therefore, an important direction for future research would be to try to change this situation.

In this thesis, it has been demonstrated that an MDN is able to provide a description of the articulatory configurations corresponding to an acoustic input vector that is more flexible than when using an MLP or other equivalent models of the articulatory domain. Consequently, a promising direction for future research would be to investigate how the output of an MDN could best be used in place of the MLP within speech recognition systems in particular.

From the discussion in Section 6.2.1, at least two ways to use the MDN output in a speech recognition system are suggested, and it would be interesting to explore both further. The first idea would be to use the variance output by the MDN directly. For example, in a standard HMM-based recogniser, the states learn an emission pdf over the observation space, which could equally be couched in the articulatory domain as in the acoustic domain. Perhaps it would be possible to modify the HMM paradigm to incorporate the comparison of two density functions instead of just using the likelihood of a single observation as happens currently, along the lines of the discussion in Section 6.2.1. In the case of the LDM-based system (Frankel & King 2001*a*), it would be possible to "inject" the variance parameters directly into the functioning of the model. The variance output of the MDN can be interpreted as an estimate of the "measurement error" at each time frame and added to the noise term in the mapping from the hidden space to the observation space.

The second approach, described in Section 6.2.1, would be to "decode" the most likely path through the sequence of density functions output by the MDN, using some sort of cost function based on prior articulatory knowledge. This knowledge would comprise a "mouth model" of sorts. In such a mouth model, the state of the articulators at time t would depend heavily on the state of the mouth at time $t-1$ and $t+1$. The underlying assumption is that the movements of the mouth are continuous, relatively smooth and quite slowly-varying. Perhaps it would also be of benefit to incorporate this in a probabilistic model, for example incorporating the prior probability for each articulatory configuration.

Finding the most likely articulatory trajectories through sequences of probability density functions output by an MDN could be performed using a Kalman smoothing approach. The mean parameters output by the MDN would provide the observation at each time step. As above, the observation error could be provided directly by the variance parameters. Articulatory constraints of varying degrees of complexity could be employed. Perhaps the simplest would be to constrain the articulators to move slowly from one time frame to the next, which could be achieved by setting the state transformation matrix to identity for example. Conversely, it might be possible to instill more sophisticated articulatory constraints by following the approach taken by (Dusan 2000), who used trained Kalman filter parameters specific to the transition between all phones in the phone inventory.

A common argument against the hope of finding any improvement in speech recognition performance using articulation estimated from the acoustic signal is that no extra information is made available. In other words, the estimated articulatory parameters are merely a transformation of information that is already present in the acoustic signal, and thus already available to the recogniser. However, using prior knowledge about human articulation in the form of a mouth model to settle upon the best, most likely, trajectory arguably *is* tantamount to incorporating more information into the speech recognition process. The MDN gives us a complete description in the form of a probability density function of where the articulators are likely to be found at each time frame. The next task is to use supplemental information, knowledge about articulation, to capitalise on this advantage.

Appendix A

Global Linear Inversion Mapping

As a simple baseline, it is interesting to calculate how well a global linear mapping from acoustics to articulation performs on the same processed data set which is used throughout this thesis[1]. In short, we aim to model the relationship between the acoustic and articulatory domains using the familiar linear regression model with the form

$$\mathbf{x} = \mathbf{A}\mathbf{y} + \mathbf{b} \tag{A.1}$$

where, following the notation introduced in Section 1.2, $\mathbf{x}$ is a vector describing the articulatory configuration and $\mathbf{y}$ is the corresponding acoustic vector. Mapping to a point in the articulatory domain is achieved as a weighted sum of the components of $\mathbf{y}$ with the addition of a constant, or intercept, $\mathbf{b}$.

In order to help interpret this linear mapping, we can consider the case of mapping from a 20-dimensional acoustic vector to one articulator channel given a training set of N such acoustic-articulatory pairs. If we imagine the acoustic data "plotted against" the corresponding articulatory data, we find a cloud of N points in the 21-D space. The linear mapping function is simply a 20-D hyperplane through this space. For any given acoustic vector, we want the mapping function output to be as close as possible to the articulator value. Therefore, the best orientation and position of the hyperplane will be such that it passes through the cloud of N points and minimises the distance from each

[1] The processing details for this data set are to be found in Section 3.4

Articulator	RMS error	RMS error (mm)	Correlation
upper lip x	0.201	1.13	0.386
upper lip y	0.182	1.47	0.472
lower lip x	0.187	1.43	0.360
lower lip y	0.166	3.30	0.598
lower incisor x	0.180	0.98	0.399
lower incisor y	0.143	1.44	0.683
tongue tip x	0.176	3.37	0.535
tongue tip y	0.162	3.61	0.639
tongue body x	0.164	2.93	0.618
tongue body y	0.146	2.83	0.680
tongue dorsum x	0.165	2.67	0.606
tongue dorsum y	0.168	2.86	0.490
velum x	0.169	0.55	0.573
velum y	0.164	0.52	0.578

Table A.1: Performance of the global linear mapping when recovering articulation from acoustics for the unseen test set. The average of the RMS error values given is 2.08mm. Here, each articulatory vector is estimated on the basis of a single acoustic frame. Compare these results with those given in Table A.2, where 20 acoustic frames are used to map to each articulatory vector.

point in terms of the given articulatory dimension. Since the orientation and position[2] of the hyperplane are determined by $\mathbf{A}$ and $\mathbf{b}$ respectively, the most suitable function coefficients therein may be computed by performing a least-squares fit as the solution of the system of N simultaneous linear equations. This process is referred to as linear regression.

Table A.1 gives the results observed when using a global linear mapping to estimate articulation from acoustics. The optimum coefficients in matrix $\mathbf{A}$ and vector $\mathbf{b}$ from equation A.1 were first determined for the training set. The mapping was then tested on the unseen test set.

For the linear mapping whose results are given in Table A.1, only a single acoustic frame of 20 filterbank coefficients was used as "input". However, it is of further interest to consider the effect of using larger windows of acoustic parameters. Figures A.1 to A.8 demonstrate this effect.

Figure A.1 shows overall that incorporating multiple frames of acoustic input when performing acoustic-articulatory inversion using a global linear mapping is beneficial.

[2]i.e. point where it crosses the axis of the articulator dimension in the 21-D space

This effect, however, would appear to be asymptotic. In other words, including more and more frames of acoustic input yields a smaller and smaller improvement.

Whereas Figure A.1 gives a summary of the relationship between acoustic input size and inversion performance, Figures A.2 to A.8 present the results on a per-channel basis. Again, the same basic relationship of asymptotic improvement is observed for all channels in terms of RMS error and correlation.

Several of these figures appear to exhibit an 'oscillation' in the improvement which results from the addition of each acoustic frame. It is possible this effect results from how the context window is enlarged by one frame each turn. The linear mapping using the smallest acoustic window has just a single acoustic frame of 20 ms centred at time t. This is used to map to the single EMA vector at the same time t. To achieve the acoustic window with two frames, we chose to adjoin an extra frame on the right of the acoustic window, i.e. centred at time $t + 10\text{ms}$[3]. Then, for the acoustic window using 3 frames, we appended a frame to the left of this window, i.e. centred at time $t - 10\text{ms}$. This process of alternately adding frames to the front then the back was continued to yield context windows of whatever number of frames were required (in this case from 1 to 20).

Looking at Figures A.2 to A.8, we find the `ul_x`, `tt_y` and `ll_x` channels to a greater or lesser degree demonstrate a larger improvement with the addition of each frame to the right hand side of the context window. Meanwhile, the other channels all show a larger improvement each time a time frame is added to the left of the acoustic context window, again each to a greater or lesser degree. Why this should be so is not immediately clear, but it is undoubtedly an interesting observation, and worthy of further investigation in the future.

Finally, Table A.2 provides the results observed for the unseen test set when using 20 acoustic frames for the linear inversion mapping. On one hand, these results are included to provide the reader with a direct comparison against the results in Table A.1 for the linear mapping from a single acoustic frame to the articulatory domain. On the other hand, it is also possible to compare these results for the linear mapping with those

[3]The acoustic coefficients were calculated using 20ms windows with 10ms shift, so that they are 50% overlapping. Exact details of the data processing used can be found in Section 3.4

Articulator	RMS error	RMS error (mm)	Correlation
upper lip x	0.184	1.04	0.530
upper lip y	0.167	1.35	0.587
lower lip x	0.174	1.33	0.495
lower lip y	0.154	3.08	0.666
lower incisor x	0.171	0.94	0.486
lower incisor y	0.131	1.31	0.747
tongue tip x	0.159	3.06	0.643
tongue tip y	0.145	3.24	0.725
tongue body x	0.147	2.62	0.711
tongue body y	0.128	2.47	0.767
tongue dorsum x	0.149	2.42	0.693
tongue dorsum y	0.154	2.63	0.597
velum x	0.150	0.49	0.687
velum y	0.147	0.47	0.683

Table A.2: Performance of the global linear mapping when recovering articulation from acoustics for the unseen test set. The average of the RMS error values given is 1.89mm. Here, each articulatory vector is estimated on the basis of a context window of 20 acoustic frames. Compare these results with those given in Table A.1, where a single acoustic frame is used to map to each articulatory vector.

for the nonlinear MLP mapping given in Tables 4.2 and 4.4. The inferior performance of the linear mapping indicates that the acoustic-articulatory mapping is itself nonlinear to some degree. Therefore, on the basis of these results, we argue that the use of a nonlinear function for modelling the inversion mapping is justified.

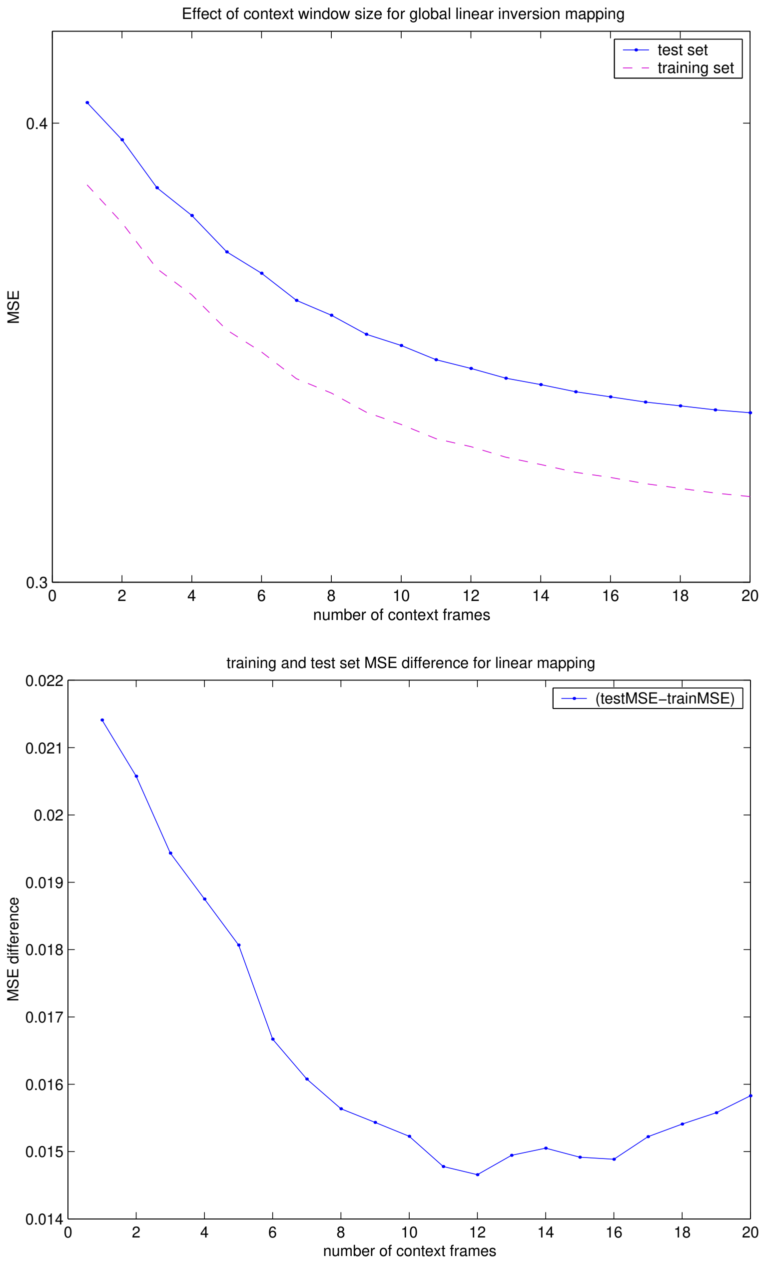


Figure A.1: Mean square error (MSE) as a function of context window size for the global linear inversion mapping from acoustics. The top plot shows MSE for the test set as the number of acoustic frames increases from 1 to 20. The training set MSE is also shown for comparison. To aid comparison, the bottom plot shows the difference between the test set and training set MSE at each acoustic window size. Overall, we see that including context frames reduces error in the linear mapping significantly, but that this is an apparently asymptotic relationship.

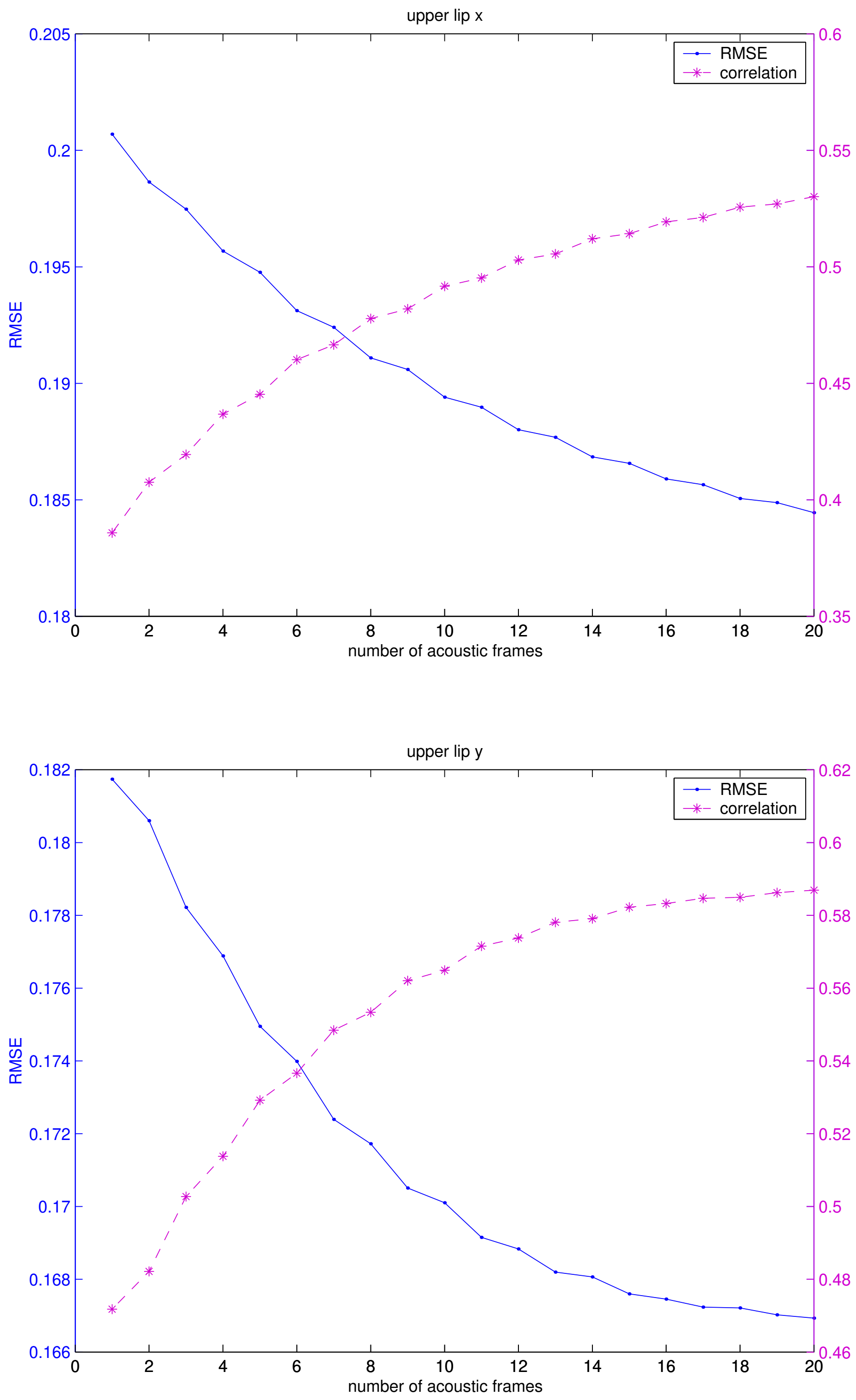
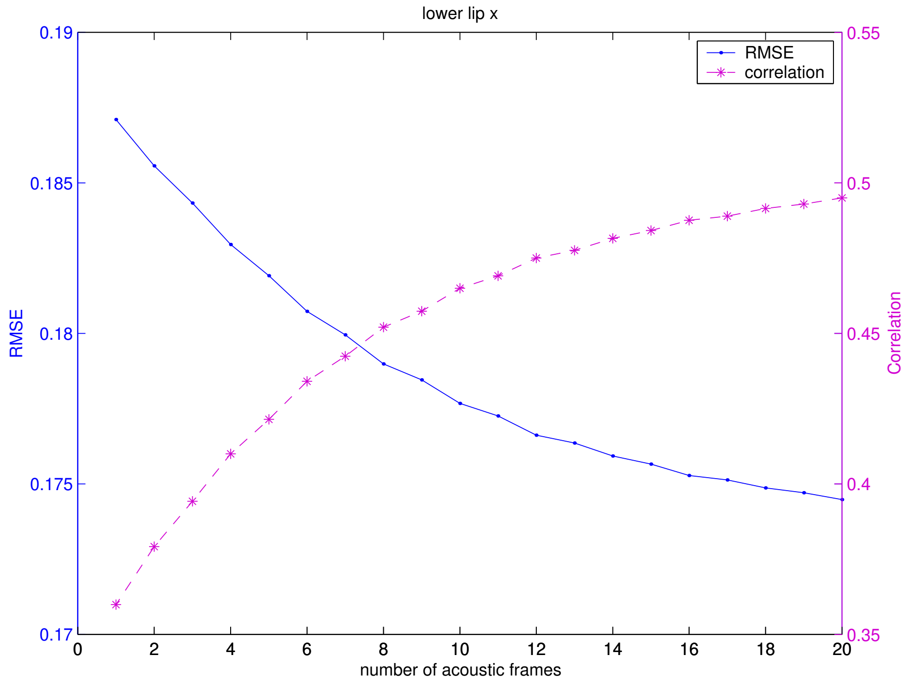


Figure A.2: RMS error and correlation as a function of context window size for a global linear inversion mapping from acoustics for the **upper lip** articulatory channels.



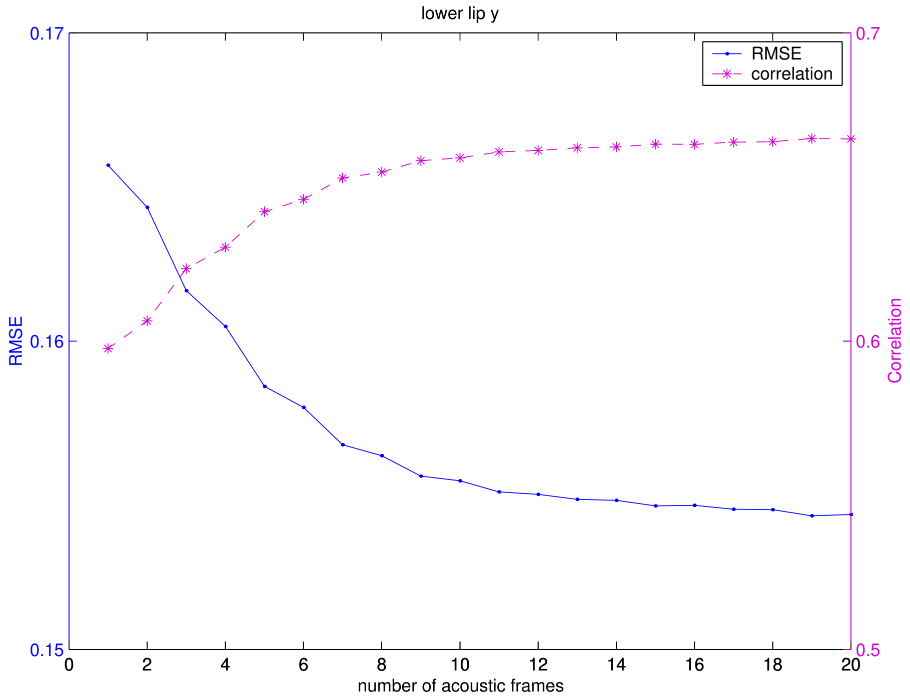


Figure A.3: RMS error and correlation as a function of context window size for a global linear inversion mapping from acoustics for the **lower lip** articulatory channels.

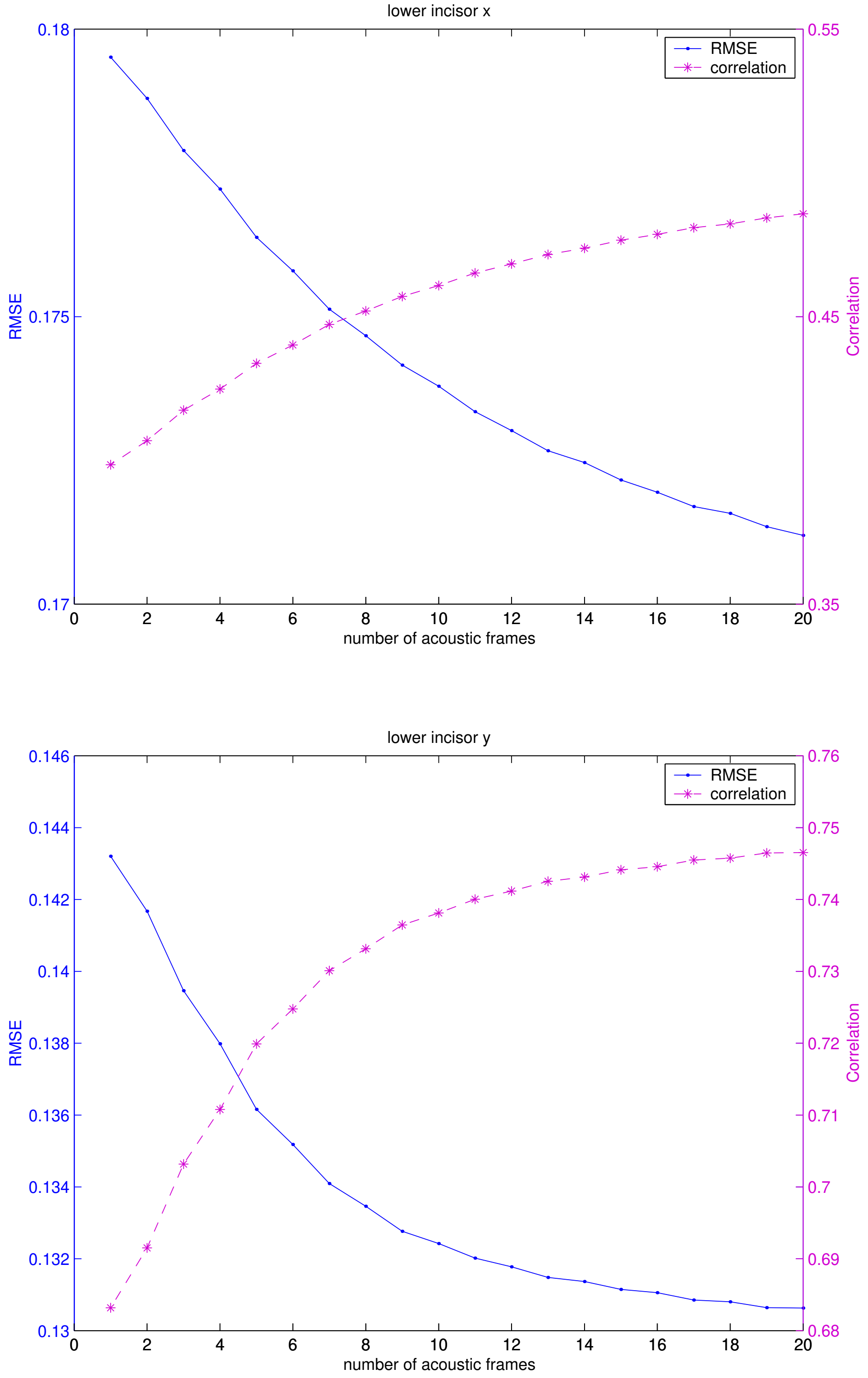
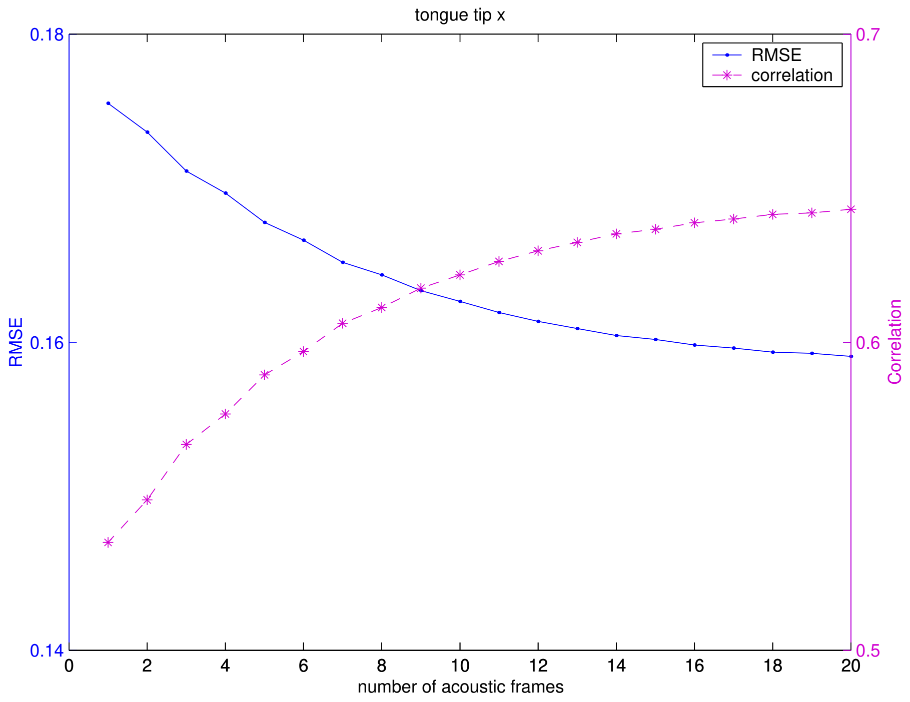


Figure A.4: RMS error and correlation as a function of context window size for a global linear inversion mapping from acoustics for the **lower incisor** articulatory channels.



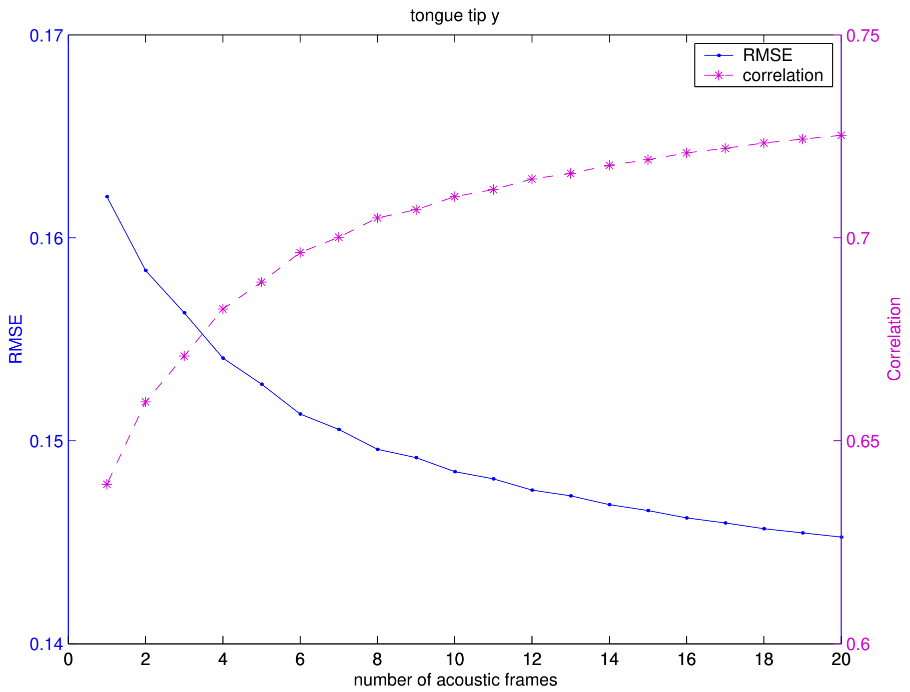


Figure A.5: RMS error and correlation as a function of context window size for a global linear inversion mapping from acoustics for the **tongue tip** articulatory channels.

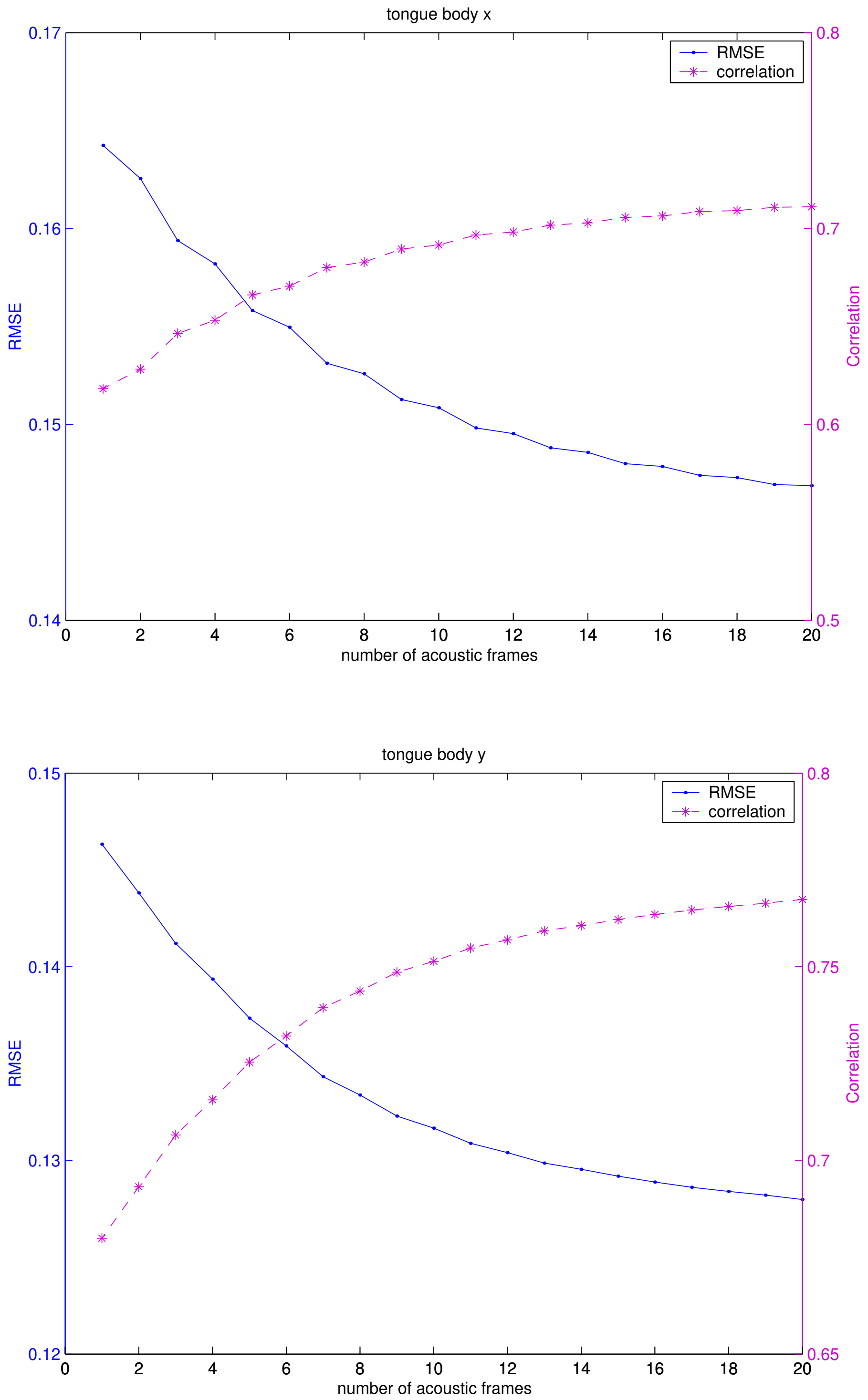
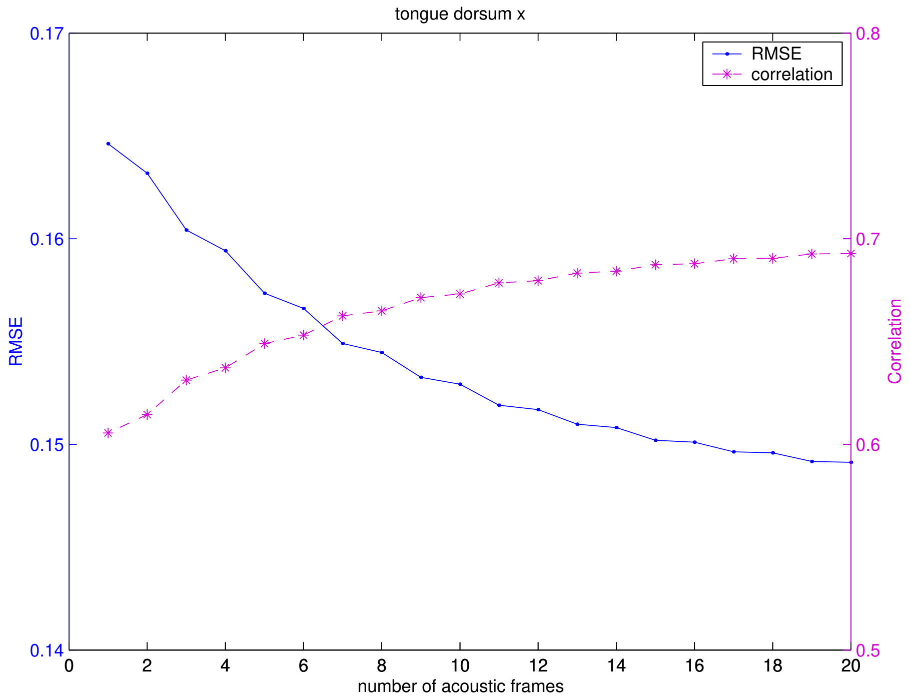


Figure A.6: RMS error and correlation as a function of context window size for a global linear inversion mapping from acoustics for the **tongue body** articulatory channels.



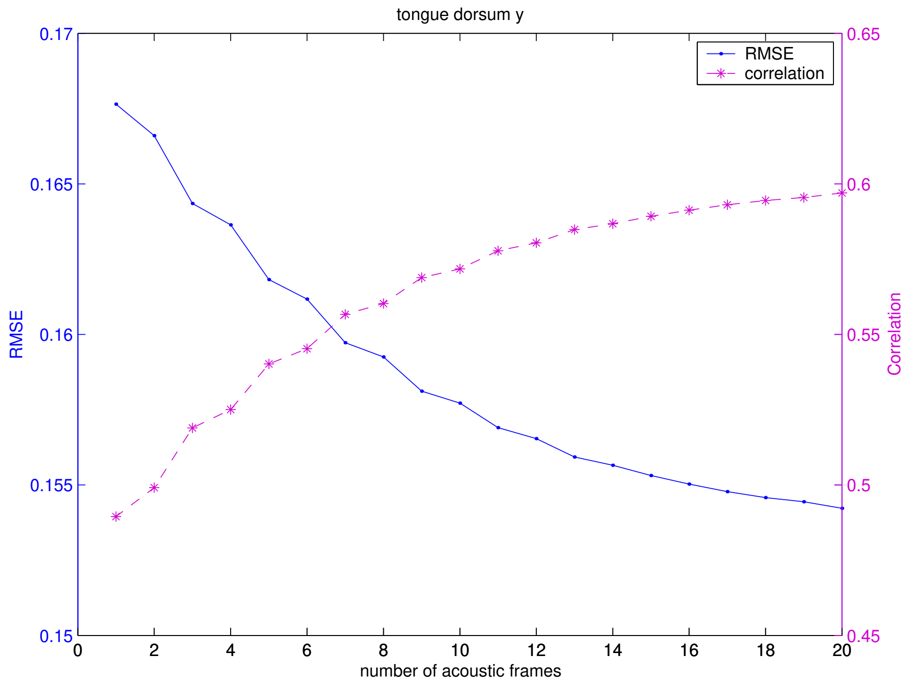
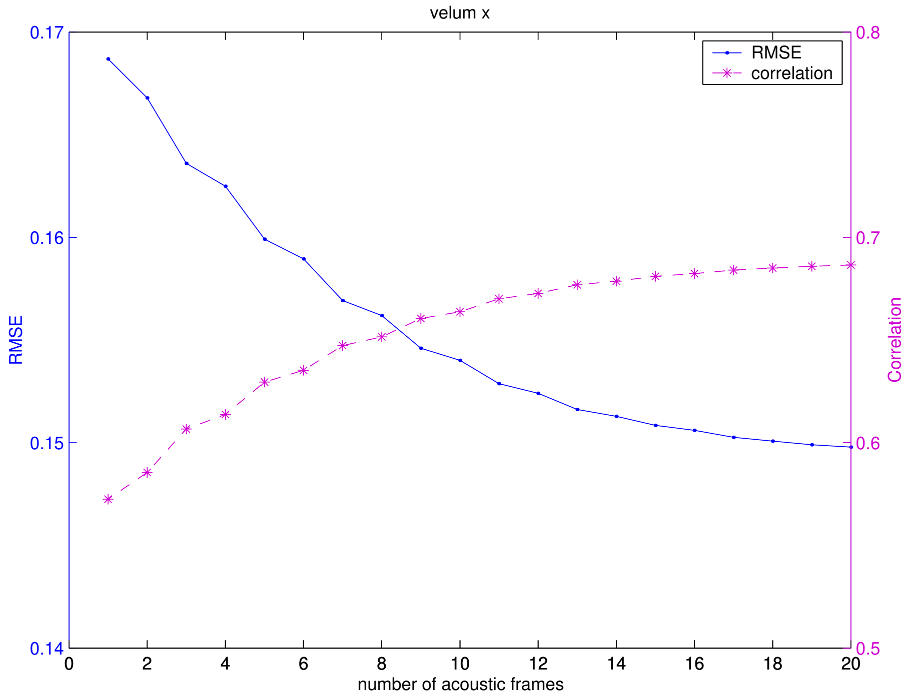


Figure A.7: RMS error and correlation as a function of context window size for a global linear inversion mapping from acoustics for the **tongue dorsum** articulatory channels.



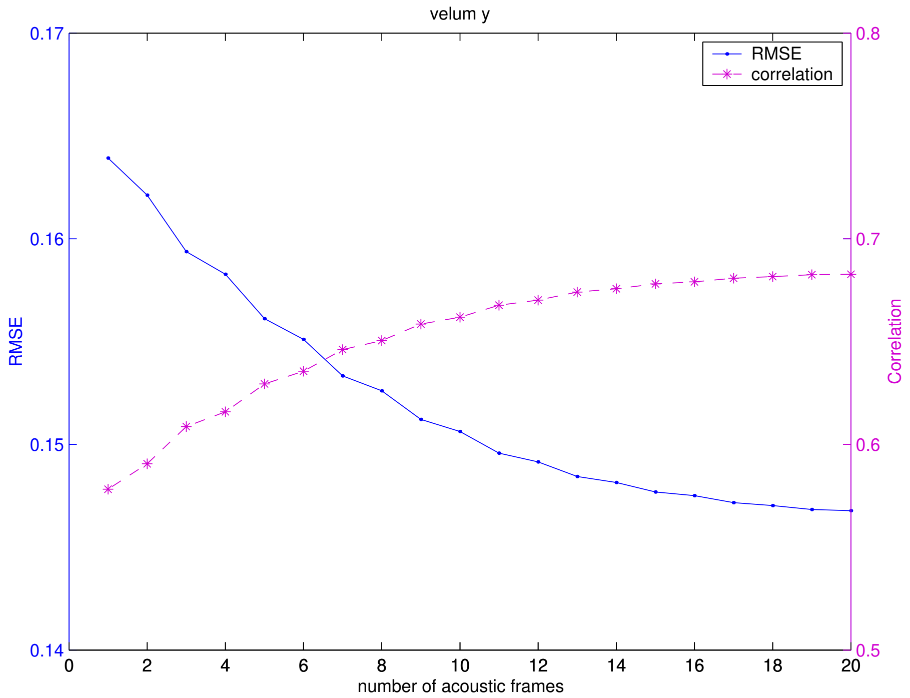


Figure A.8: RMS error and correlation as a function of context window size for a global linear inversion mapping from acoustics for the **velum** articulatory channels.

Appendix B

EMA Trajectory Normalisation

Section 3.4.1 presents a method for removing underlying trends which have been observed in plots of the means of EMA files taken in their sequential recording order. In order to assess whether this method provides any benefit, we can train the same MLP as shown in Figure 4.2 using data normalised with global means and variances and compare results with the MLP which was trained on data normalised using the method detailed in Section 3.4.1. Such results are provided in Table B.1.

The results in Table B.1 should be compared with those in Table 4.2 for the MLP trained and tested with data that has been normalised using means and standard deviations calculated by the "means-filtering" process. From Table B.1, the average RMS error when simply using the global mean and variance for normalisation is 1.73mm. Meanwhile, the average RMS error using data normalised according to the method detailed in Section 3.4.1 is 1.64mm.

Overall, this is a relative improvement of over 5%. All articulatory channels show an improvement in RMS error expressed in millimetres, apart from for the tongue body y-coordinate, which becomes marginally worse by 0.01mm. The biggest improvement is observed for the velum coordinates; `v_y` RMS error improves by 55.3%. This normalisation method is therefore deemed to be useful. All experiments in this thesis make use of the training, validation and test data sets which have been normalised using the file-specific means derived by the means-filtering process.

Articulator	RMS error	RMS error(mm)	Correlation
upper lip x	0.178	1.02	0.566
upper lip y	0.147	1.24	0.694
lower lip x	0.158	1.24	0.604
lower lip y	0.138	2.77	0.745
lower incisor x	0.165	1.00	0.542
lower incisor y	0.120	1.22	0.788
tongue tip x	0.130	2.57	0.777
tongue tip y	0.117	2.65	0.831
tongue body x	0.124	2.26	0.802
tongue body y	0.111	2.18	0.832
tongue dorsum x	0.129	2.10	0.781
tongue dorsum y	0.139	2.39	0.693
velum x	0.153	0.64	0.654
velum y	0.162	0.94	0.570

Table B.1: Performance of the initial MLP architecture shown in Figure 4.2, containing 50 hidden units, when recovering articulation from acoustics for the unseen test set. The average of the RMS error values given is 1.73mm. In contrast with the MLP whose results are given in Table 4.2, this MLP has been trained and tested using data from the `fsew0` corpus which has been normalised using global means and variances.

Appendix C

MLP Output Lowpass Filtering

Lowpass filtering is one way of imposing an articulatory constraint on estimated articulatory trajectories. In so doing, we draw on our knowledge of human articulation; that the articulators move consistently and relatively slowly. But, what cutoff frequencies should be used when lowpass filtering the different channels of MLP output?

The graphs in Figures C.1 to C.7 present the full results of employing an empirical approach to answer this question. Each of the fourteen estimated articulatory trajectories for all the utterances in the validation set were lowpass filtered with cutoff frequencies ranging from 2-49Hz using a second order Butterworth filter. Since the frame shift of the processed EMA data and neural network output is 10ms, the 50Hz point is in fact the result obtained by not filtering the MLP output. The optimal lowpass filtering cutoff frequency for each channel is identified as the one which gives the best compromise between low RMS error and high correlation compared with the corresponding real target articulator trajectory. In each case, this is indicated with a vertical dotted line. For the reader's convenience, the optimal cutoff frequencies used for lowpass filtering the MLP output are summarised here:

3Hz `ul_x`, `ll_x` and `li_x`

5Hz `ul_y`, `v_x` and `v_y`

6Hz `tt_x`, `tb_x` and `td_y`

7Hz `li_y`, `tb_y` and `td_x`

8Hz `ll_y`

9Hz `tt_y`

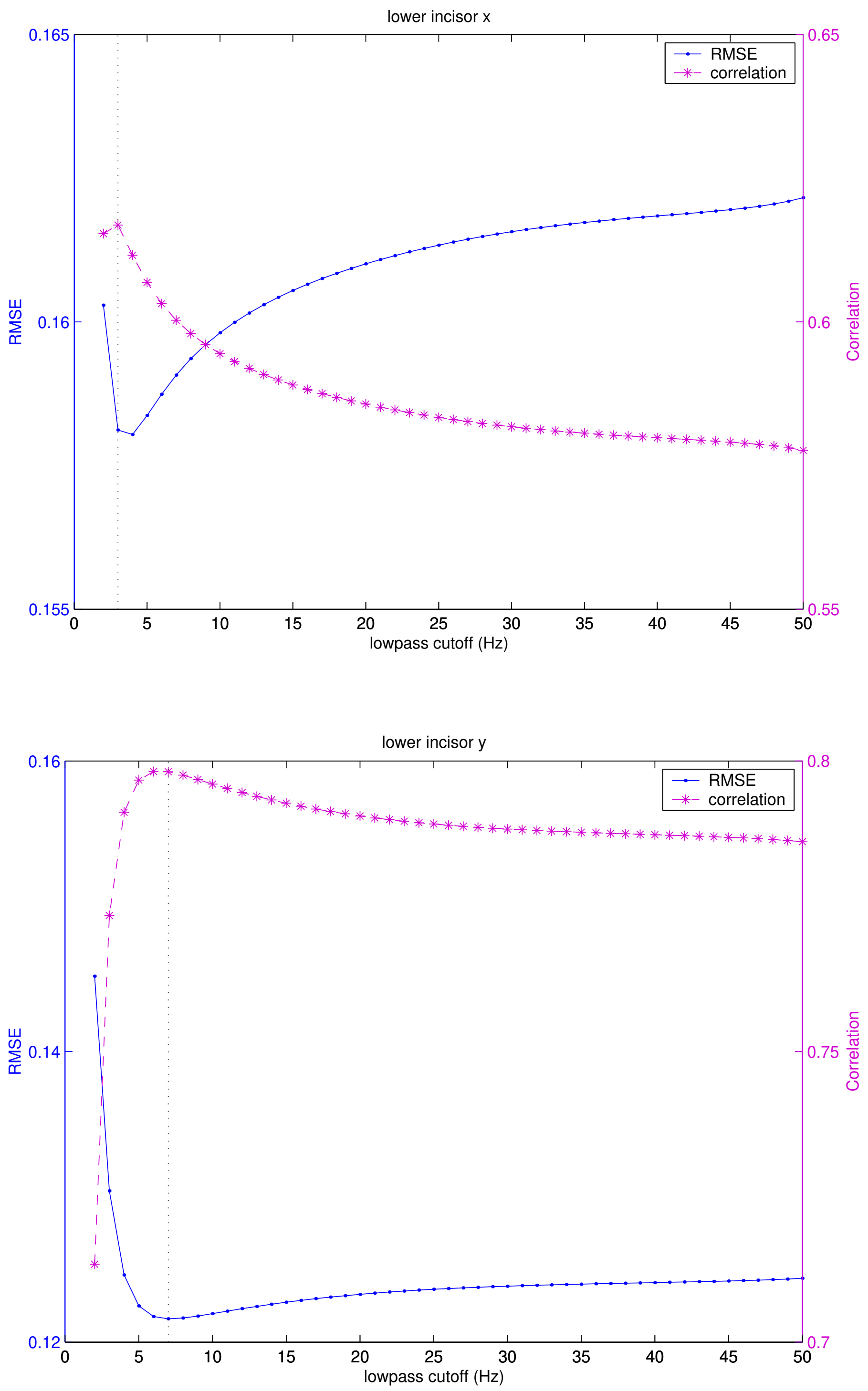
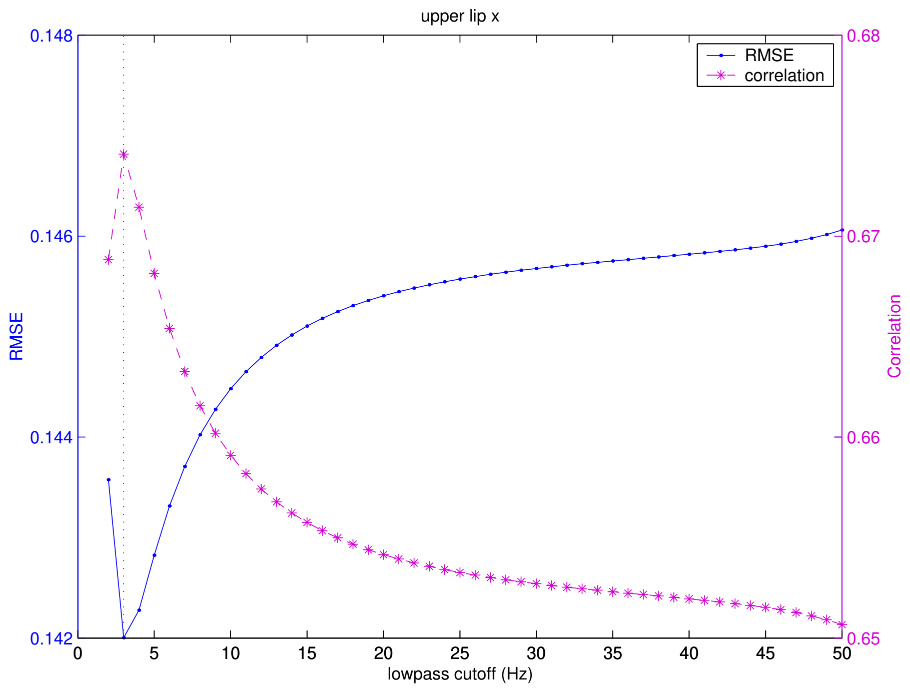


Figure C.1: RMS error and correlation as a function of cutoff frequency for lowpass filtering of the **lower incisor** channels of MLP estimated articulatory trajectories.



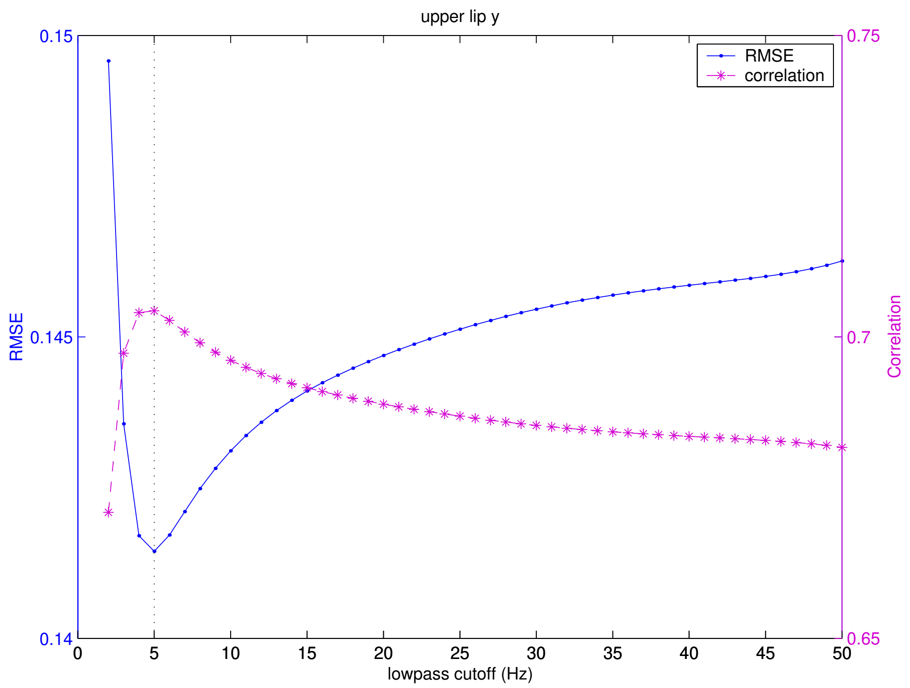
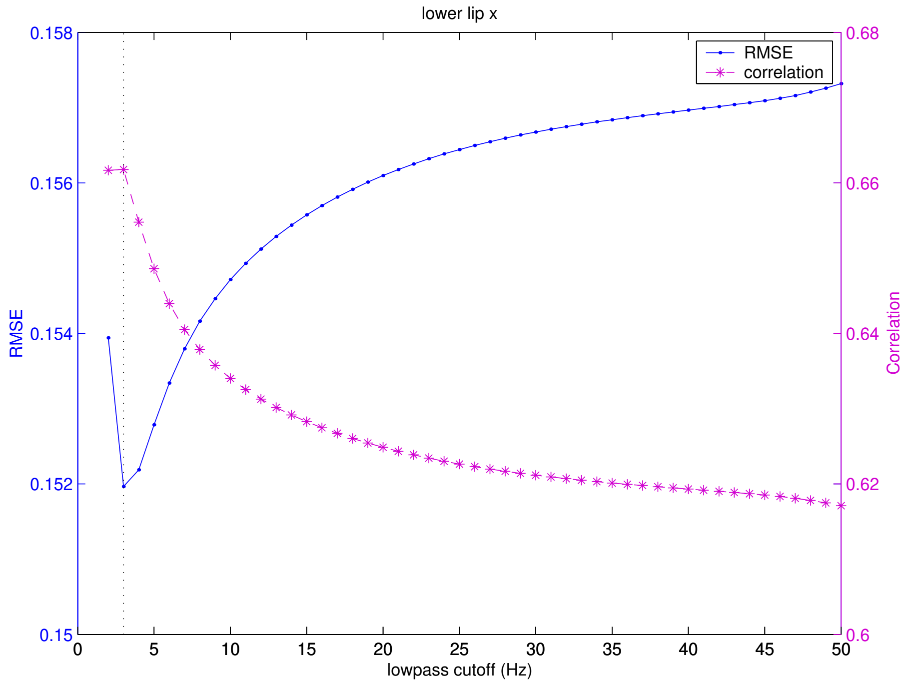


Figure C.2: RMS error and correlation as a function of cutoff frequency for lowpass filtering of the **upper lip** channels of MLP estimated articulatory trajectories.



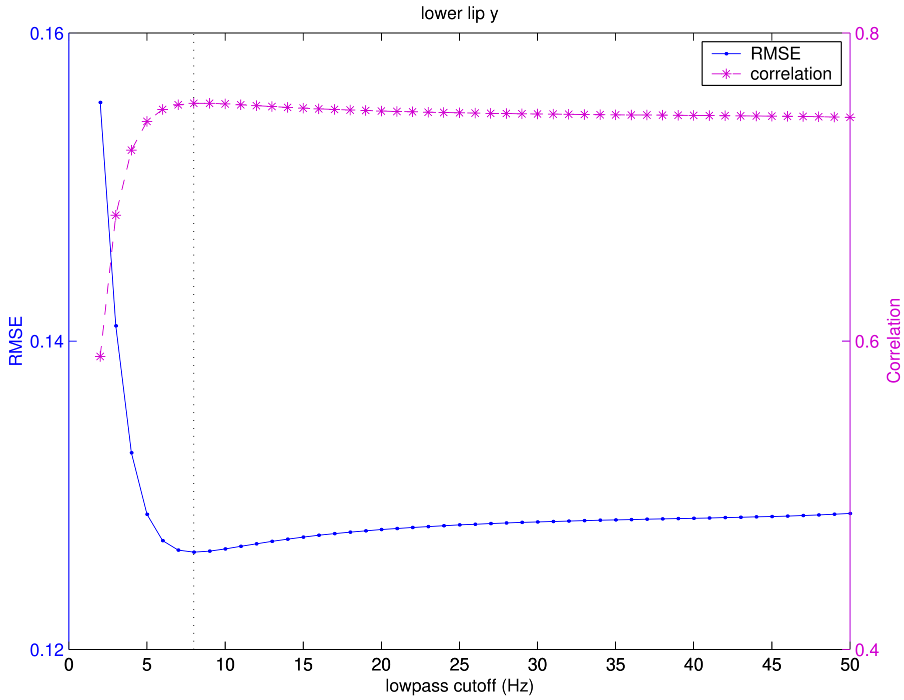


Figure C.3: RMS error and correlation as a function of cutoff frequency for lowpass filtering of the **lower lip** channels of MLP estimated articulatory trajectories.

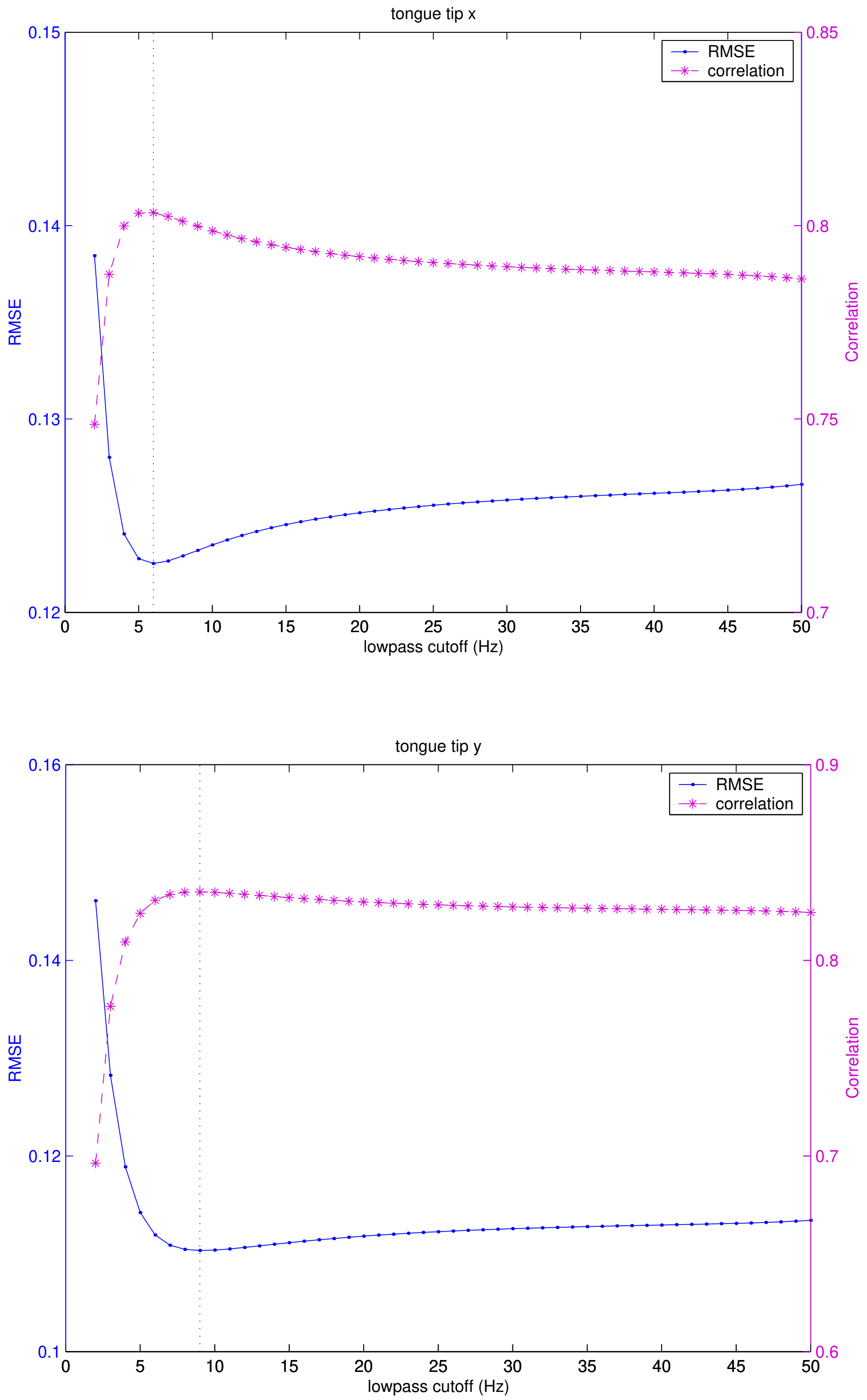
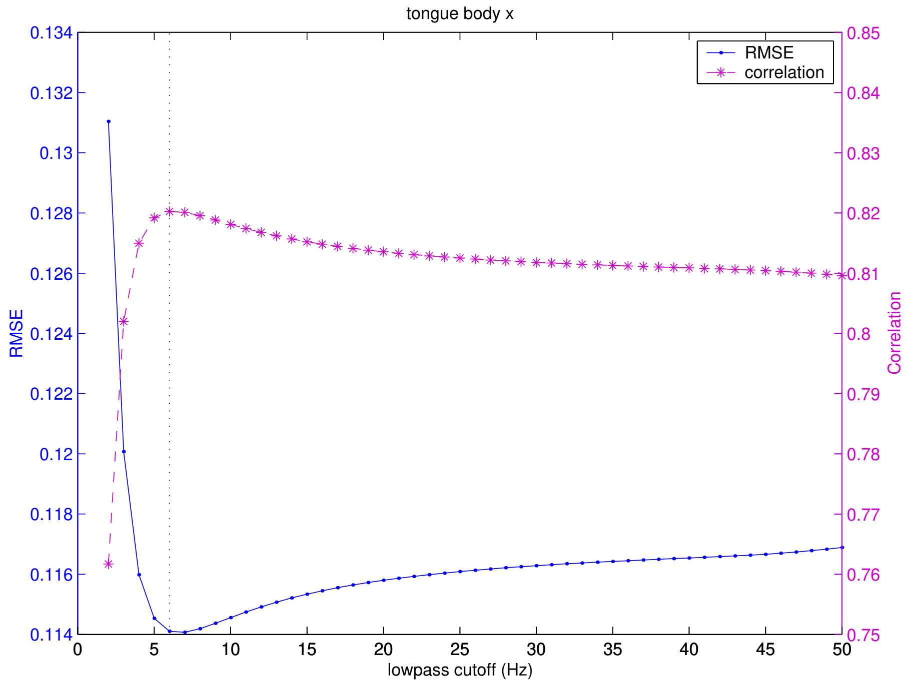


Figure C.4: RMS error and correlation as a function of cutoff frequency for lowpass filtering of the **tongue tip** channels of MLP estimated articulatory trajectories.



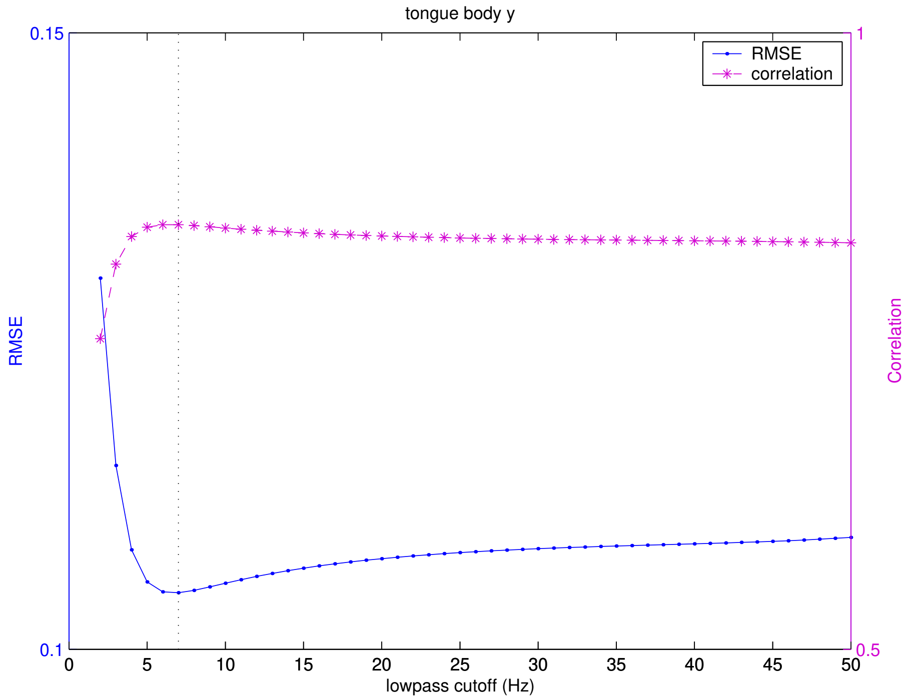
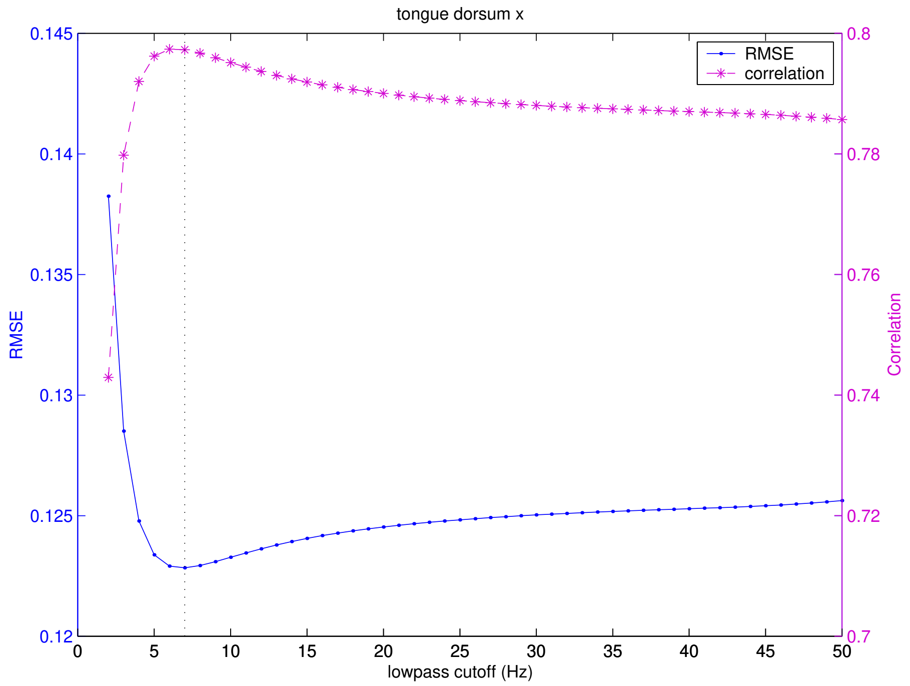


Figure C.5: RMS error and correlation as a function of cutoff frequency for lowpass filtering of the **tongue body** channels of MLP estimated articulatory trajectories.



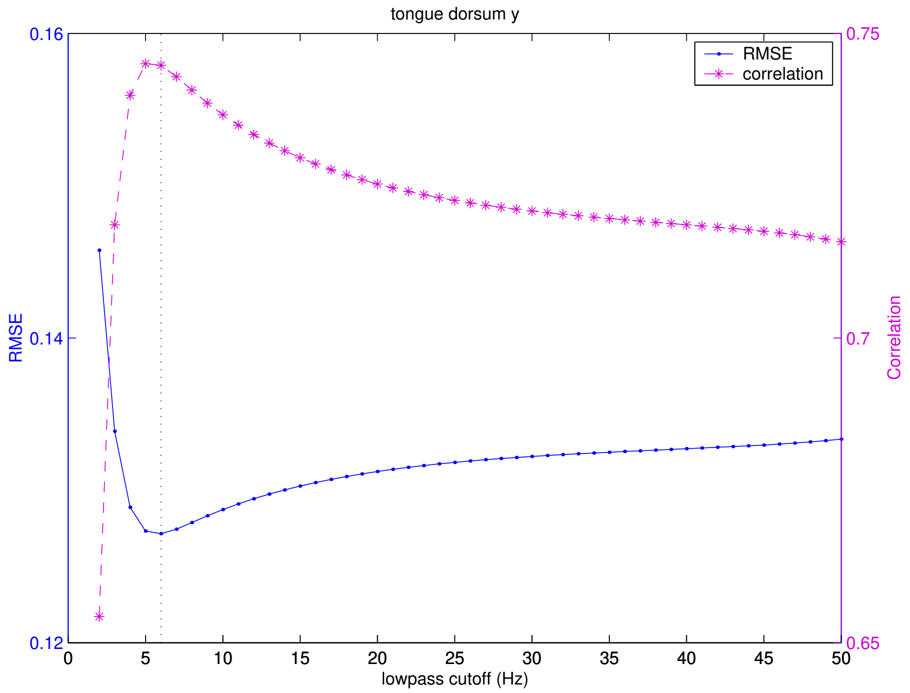
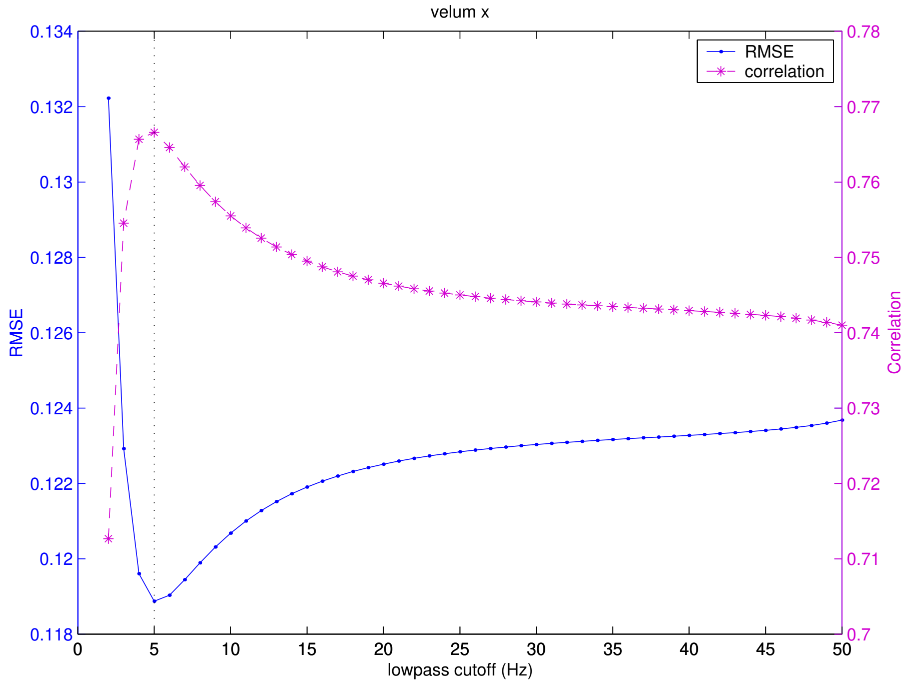


Figure C.6: RMS error and correlation as a function of cutoff frequency for lowpass filtering of the **tongue dorsum** channels of MLP estimated articulatory trajectories.



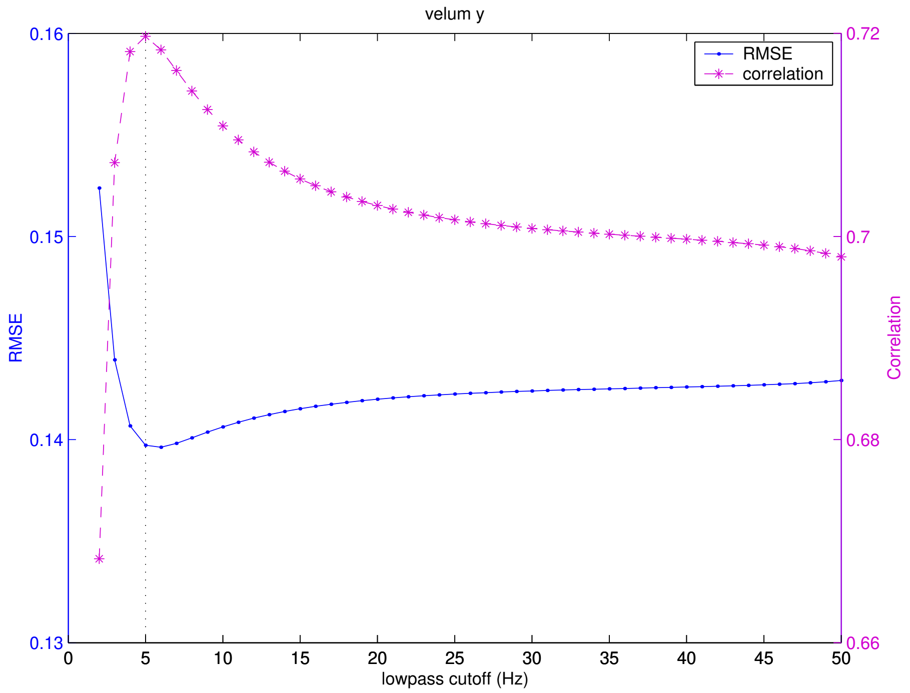


Figure C.7: RMS error and correlation as a function of cutoff frequency for lowpass filtering of the **velum** channels of MLP estimated articulatory trajectories.

Bibliography

Atal, B. S., Chang, J. J., Mathews, M. V. & Tukey, J. W. (1978), 'Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer sorting technique', *J. Acoust. Soc. Am.* **63**, 1535–1555.

Berry, M. J. A. & Linoff, G. (1997), *Data Mining Techniques: for Marketing, Sales and Customer Support*, John Wiley & Sons.

Bishop, C. (1995), *Neural Networks for Pattern Recognition*, Oxford University Press.

Bishop, C. M. (1994), Mixture density networks, Technical Report NCRG/4288, Neural Computing Research Group, Department of Computer Science, Aston University, Birmingham, B4 7ET, U.K.

Blackburn, C. S. (1996), Articulatory Methods for Speech Production and Recognition, PhD thesis, Trinity College Cambridge and Cambridge University Engineering Department.

Blackburn, C. S. & Young, S. J. (1995*a*), Learning new articulator trajectories for a speech production model using artificial neural networks, *in* 'IEEE International Conference on Neural Networks', Vol. 4, pp. 2046–2051.

Blackburn, C. S. & Young, S. J. (1995*b*), A novel self-organising speech production system using pseudo-articulators, *in* 'International Congress of Phonetic Sciences', Vol. 2, pp. 238–241.

Blackburn, C. S. & Young, S. J. (1995*c*), Towards improved speech recognition using a speech production model, *in* 'European Conference on Speech Communication and Technology', Vol. 3, pp. 1623–1626.

Blackburn, C. S. & Young, S. J. (1996), Pseudo-articulatory speech synthesis for recognition using automatic feature extraction from X-ray data, *in* 'Proc. ICSLP '96', Vol. 2, Philadelphia, PA, pp. 969–972.

Blum, A. (1992), *Neural Networks in C++*, Wiley.

Borowski, E. J. & Borwein, J. M. (1991), *Dictionary of Mathematics*, second edn, HarperCollins.

Bourlard, H., Hermansky, H. & Morgan, N. (1996), 'Towards increasing speech recognition error rates', *Speech Communication* **18**(3), 205–231.

Bridle, J. S. (1990), Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition, *in* 'Neurocomputing: Algorithms, Architectures and Applications', Springer-Verlag, pp. 227–236.

Browman, C. P. & Goldstein, L. (1992), 'Articulatory phonology: An overview', *Phonetica* **49**, 155–180.

Callan, D. E., Callan, A. M., Kroos, C. & Vatikiotis-Bateson, E. (2000), Neural processes underlying perception of audio-visual speech production, *in* 'Proc. 5th Seminar on Speech Production', Kloster Seeon, Bavaria, pp. 273–276.

Chen, M. (1995), 'Acoustic parameters of nasalised vowels in hearing-impaired and normal-hearing speakers', *J. Acoust. Soc. Am.* **98**(5), 2443–2453.

Chennoukh, S., Sinder, D., Richard, G. & Flanagan, J. (1997*a*), 'Articulatory based low bit-rate speech coding', *J. Acoust. Soc. Am.* **102**(5 (Pt.2)), 3163.

Chennoukh, S., Sinder, D., Richard, G. & Flanagan, J. (1997*b*), 'Methods for acoustic-to-articulatory mapping and voice mimic systems', *J. Acoust. Soc. Am.* **101**(5).

Chennoukh, S., Sinder, D., Richard, G. & Flanagan, J. (1997*c*), Voice mimic system using an articulatory codebook for estimation of vocal tract shape, *in* 'Proc. Eurospeech', Rhodes, Greece, pp. 429–432.

Dang, J. & Honda, K. (2000), Estimation of vocal tract shape from speech sounds via a physiological articulatory model, *in* 'Proc. 5th Seminar on Speech Production', Kloster Seeon, Bavaria, pp. 233–236.

Deng, L., Ramsay, G. & Sun, D. (1997), 'Production models as a structural basis for automatic speech recognition', *Speech Communication* **22**(2–3), 93–111.

Deng, L. & Sun, D. (1994*a*), Phonetic recognition using HMM representation of overlapping articulatory features for all classes of English sounds, *in* 'Proc. ICASSP '94', Adelaide, Australia, pp. I–45–I–48.

Deng, L. & Sun, D. X. (1994*b*), 'A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features', *J. Acoust. Soc. Am.* **95**(5), 2702–2719.

Dusan, S. (2000), Statistical Estimation of Articulatory Trajectories from the Speech Signal using Dynamical and Phonological Constraints, PhD thesis, Dept. of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada.

Dusan, S. & Deng, L. (1998*a*), Estimation of articulatory parameters from speech acoustics by Kalman filtering, *in* 'Proceedings of the Inaugural CITO Researcher Retreat', Hamilton, Canada.

Dusan, S. & Deng, L. (1998*b*), Recovering vocal tract shapes form MFCC parameters, *in* 'Proceedings of the International Conference on Spoken Language Processing', Vol. 7, Sydney, Australia, p. 3087.

Dusan, S. & Deng, L. (2000), Acoustic-to-articulatory inversion using dynamical and phonological constraints, *in* 'Proc. 5th Seminar on Speech Production', Kloster Seeon, Bavaria, pp. 237–240.

Dusan, S., Deng, L. & Wong, A. (1997), Classification of articulatory and acoustic patterns of speech in the task spaces, *in* 'Proceedings of the TRIO/ITRC Researcher Retreat', Kingston, Canada.

Erler, K. & Freeman, G. H. (1996), 'An HMM-based speech recognizer using overlapping articulatory features', *J. Acoust. Soc. Am.* **100**(4), 2500–2513.

Flanagan, J., Ishizaka, K. & Shipley, K. (1980), 'Signal models for low bit-rate coding of speech', *J. Acoust. Soc. Am.* **68**, 780–791.

Fowler, C. A. (1996), 'Listeners do hear sounds, not tongues', *J. Acoust. Soc. Am.* **99**(3), 1730–1741.

Frankel, J. & King, S. (2001*a*), ASR — articulatory speech recognition, *in* 'Proc. Eurospeech', Vol. 1, Aalborg, Denmark, pp. 599–602.

Frankel, J. & King, S. (2001*b*), Speech recognition in the articulatory domain: Investigating an alternative to acoustic HMMs, *in* 'Proc. Workshop on Innovations in Speech Processing', pp. 37–46.

Frankel, J., Richmond, K., King, S. & Taylor, P. (2000), An automatic speech recognition system using neural networks and linear dynamic models to recover and model articulatory traces, *in* 'Proc. ICSLP', Beijing, China.

Fugimura, O. (1982), 'Computer-controlled X-ray microbeam – method, history, and sample results', *J. Acoust. Soc. Am.* **71 (Pt. 2)**, S31.

Fujimura, O. (1967), 'Medical implications of on-line computer experiments in speech research', *Tokyo J. Med. Sci.* **75**, 235–239.

Fujimura, O., Kiritani, S. & Ishida, H. (1973), 'Computer-controlled radiography for observation of articulatory and other human organs', *Comp. Biol. Med.* **3**, 371–384.

Furui, S. & Sondhi, M. M., eds (1992), *Advances in Speech Signal Processing*, Marcel Dekker Inc, New York.

Hardcastle, W. J. & Marchal, A., eds (1990), *Speech Production and Speech Modelling*, Kluwer Academic Publishers, Netherlands.

Hassibi, B. & Stork, D. G. (1993), Second order derivatives for network pruning: Optimal brain surgeon, *in* S. J. Hanson, J. D. Cowan & C. L. Giles, eds, 'Advances in Neural Information Processing Systems', Vol. 5, Morgan Kaufmann, San Mateo, CA, pp. 164–171.

Hertz, J., Krogh, A. & Palmer, R. G. (1991), *An Introduction to the Theory of Neural Computation*, Vol. 1, Addison Wesley.

Hinton, G. E. (1986), Learning distributed representations of concepts, *in* 'Proceedings of the 8th Annual Conference of the Cognitive Science Society', Hillsdale: Erlbaum, pp. 1–12.

Hogden, J., Lofqvist, A., Gracco, V., Zlokarnik, I., Rubin, P. & Saltzman, E. (1996), 'Accurate recovery of articulator positions from acoustics: New conclusions based on human data', *J. Acoust. Soc. Am.* **100**(3), 1819–1834.

Hogden, J., Nix, D. & Valdez, P. (1998), An articulatorily constrained, maximum likelihood approach to speech recognition, Technical report, Los Alamos National Laboratory, Los Alamos, NM 87545.

Hogden, J. & Valdez, P. (2000), Bridging the gap between speech production and speech recognition, *in* 'Proc. 5th Seminar on Speech Production', Kloster Seeon, Bavaria, pp. 241–244.

Holzrichter, J. F., Burnett, G. C., Ng, L. C. & Lea, W. A. (1998), 'Speech articulator movements using low power EM-wave sensors', *J. Acoust. Soc. Am.* **103**(1), 622–625.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J. & Hinton, G. E. (1991), 'Adaptive mixtures of local experts', *Neural Computation* **3**(1), 79–87.

Johnson, K., Ladefoged, P. & Lindau, M. (1993), 'Individual differences in vowel production', *J. Acoust. Soc. Am.* **94**(2), 701–714.

Kalman, R. E. (1960), 'A new approach to linear filtering and prediction problems', *Trans. ASME, Journal of Basic Engineering* pp. 35–45.

King, S., Taylor, P., Frankel, J. & Richmond, K. (2000), Speech recognition via phonetically-featured syllables, *in* 'Phonus Nr 5: Workshop on Phonetics and Phonology in ASR', number 5, pp. 15–34.

Kobayashi, T., Yagyu, M. & Shirai, K. (1991), Application of neural networks to articulatory motion estimation, *in* 'Proc ICASSP', pp. 489–492.

Kuc, R., Tutuer, F. & Vaisnys, J. R. (1985), Determining vocal tract shape by applying dynamic constraints, *in* 'Proc. ICASSP', Tampa, Florida, pp. 1101–1104.

Kullback, S. & Leibler, R. A. (1951), 'On information and sufficiency', *Annals of Mathematical Statistics* **22**, 79–86.

Ladefoged, P., Harshman, R., Goldstein, L. & Rice, L. (1978), 'Generating vocal tract shapes from formant frequencies', *J. Acoust. Soc. Am.* **64**, 1027–1035.

Larar, J. N., Schroeter, J. & Sondhi, M. M. (1988), 'Vector quantization of the articulatory space', *IEEE Trans. Acoust. Speech and Sig. Process.* **36**(12), 1812–1818.

LeCun, Y., Denker, J., Solla, S., Howard, R. E. & Jackel, L. D. (1990), Optimal brain damage, *in* D. S. Touretzky, ed., 'Advances in Neural Information Processing Systems II', Morgan Kauffman, San Mateo, CA.

Liberman, A. & Mattingly, I. (1985), 'The motor theory revised', *Cognition* **21**, 1–36.

Lindblom, B. (1990), Explaining phonetic variation: A sketch of the H&H theory, *in* Hardcastle & Marchal (1990).

Lindblom, B. (1996), 'Role of articulation in speech perception: Clues from production', *J. Acoust. Soc. Am.* **99**(3), 1683–1692.

Lindblom, B., Lubker, J. & Gay, T. (1979), 'Formant frequencies of some fixed-mandible vowels and a model of speech motor programming by predictive simulation', *J. Phon.* **7**, 146–161.

Luettin, J. (1997), 'Artist: Articulatory representation to improve speech technologies', http://www.idiap.ch/vision/artist.html. (Juergen Luettin is PI).

Maeda, S. (1990), Compensatory articulation during speech: Evidence from the analysis and synthesis of vocal-tract shapes using an articulatory model, *in* Hardcastle & Marchal (1990).

McGowan, R. S. & Faber, A. (1996), 'Introduction to papers on speech recognition and perception from an articulatory point of view', *J. Acoust. Soc. Am.* **99**(3), 1680–1682.

Mermelstein, P. (1973), 'Articulatory model for the study of speech production', *J. Acoust. Soc. Am.* **53**(4), 1070–1082.

Møller, M. (1993), 'A scaled conjugate gradient algorithm for fast supervised learning', *Neural Networks* **6**(4), 525–533.

Moore, R. K. (1996), 'Critique: The potential role of speech production models in automatic speech recognition', *J. Acoust. Soc. Am.* **99**(3), 1710–1713.

Mozer, M. C. & Smolensky, P. (1989), Skeletonization: A technique for trimming the fat from a network via relevance assessment, *in* D. S. Touretzky, ed., 'Advances in Neural Information Processing Systems 1', Morgan Kaufmann, pp. 107–115.

Muller, E. & McLeod, G. (1982), 'Perioral biomechanics and its relation to labial motor control', *J. Acoust. Soc. Am. Suppl. 1* **78**(S38).

Nadler, R., Abbs, J. H. & Thompson, M. (1985), 'The new nationally shared X-ray microbeam facility: Status report', *J. Acoust. Soc. Am. Suppl. 1* **78**, S38.

Nearey, T. M. (1996), 'Critique: Phonological contrast and articulatory properties', *J. Acoust. Soc. Am.* **99**(3), 1714–1717.

Nelson, W. (1977), 'Articulatory feature analysis - i. initial processing considerations', Memorandum, Bell Laboratories.

Ohala, J. J. (1996), 'Speech perception is hearing sounds, not tongues', *J. Acoust. Soc. Am.* **99**(3), 1718–1725.

Okadome, T., Suzuki, S. & Honda, M. (2000), Recovery of articulatory movements from acoustics with phonemic information, *in* 'Proc. 5th Seminar on Speech Production', Kloster Seeon, Bavaria, pp. 229–232.

Orr, M. J. L. (1992), A mickey mouse guide to Kalman filtering, DAI working paper 232, Dept. Artificial Intelligence, Edinburgh Univ.

O'Shaughnessy, D. (1996), 'Critique: Speech perception: Acoustic or articulatory?', *J. Acoust. Soc. Am.* **99**(3), 1726–1729.

Ouni, S. & Laprie, Y. (1999), Design of hypercube codebooks for the acoustic-to-articulatory inversion respecting the non-linearities of the articulatory-to-acoustic mapping, *in* 'Proc. Eurospeech', Vol. 1, Budapest, Hungary, pp. 141–144.

Owens, F. J. (1993), *Signal Processing of Speech*, The Macmillan Press.

Papcun, G., Hochberg, J., Thomas, T. R., Laroche, F., Zachs, J. & Levy, S. (1992), 'Inferring articulation and recognising gestures from acoustics with a neural network trained on X-ray microbeam data', *J. Acoust. Soc. Am.* **92**(2), 688–700.

Perkell, J. S., Cohen, M. H., Svirsky, M. A., Matthies, M. L., Garabieta, I. & Jackson, M. T. T. (1992), 'Electromagnetic midsagittal articulometer systems for transducing speech articulatory movements', *J. Acoust. Soc. Am.* **92**(6), 3078–3096.

Plaut, D. C., Nowlan, S. J. & Hinton, G. E. (1986), Experiments on learning by back-propagation, Technical Report CMU-CS-86-126, Carnegie-Mellon University, Pittsburgh, PA.

Rahim, M. G., Kleijn, W. B., Schroeter, J. & Goodyear, C. C. (1991), Acoustic-to-articulatory parameter mapping using an assembly of neural networks, *in* 'Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing', pp. 485–488.

Rahim, M., Goodyear, C., Kleijn, W., Schroeter, J. & Sondhi, M. (1993), 'On the use of neural networks in articulatory speech synthesis', *J. Acoust. Soc. Am.* **93**(2), 1109–1121.

Reed, R. (1993), 'Pruning algorithms - a survey', *IEEE Trans. Neural Networks* **4**(5), 740–747.

Richard, G., Lin, Q., Zussa, F., Sinder, D., Che, C. & Flanagan, J. (1995), 'Vowel recognition using an articulatory representation', *J. Acoust. Soc. Am.* **98**(5 (Pt. 2)).

Richmond, K. (1999), Estimating velum height from acoustics during continuous speech, *in* 'Proc. Eurospeech', Vol. 1, Budapest, Hungary, pp. 149–152.

Richmond, K. (2001*a*), Mixture density networks, human articulatory data and acoustic-to-articulatory inversion of continuous speech, *in* 'Proc. Workshop on Innovation in Speech Processing', Institute of Acoustics, pp. 259–276.

Richmond, K. (2001*b*), 'Modelling the uncertainty of recovering articulation from acoustics', Submitted to Computer Speech and Language.

Robinson, T., Cook, G., Ellis, D., Fosler-Lussier, E., Renals, S. & Williams, G. (2000), 'Connectionist speech recognition of broadcast news'. Submitted to Speech Communication, 2000.

Rose, R. C., Schroeter, J. & Sondhi, M. M. (1994), An investigation of the potential role of speech production models in automatic speech recognition, *in* 'Proceedings of the International Conference on Speech and Language Processing', Vol. 2, pp. 575–578.

Rose, R. C., Schroeter, J. & Sondhi, M. M. (1996), 'The potential role of speech production models in automatic speech recognition', *J. Acoust. Soc. Am.* **99**(3), 1699–1709.

Roweis, S. (1997), Towards articulatory speech recognition: Learning smooth maps to recover articulator information, *in* 'Eurospeech', Vol. 3, pp. 1227–1230.

Roweis, S. (1999), Data Driven Production Models for Speech Processing, PhD thesis, California Institute of Technology, Pasadena, California.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986), Learning internal representations by error propagation, *in* D. E. Rumelhart & J. L. McClelland, eds, 'Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. I: Foundations', Bradford Books/MIT Press, Cambridge, MA., pp. 318–362.

Schmidbauer, O., Casacuberta, F., Castro, M. J., Hegerl, G., Höge, H., Sanchez, J. A. & Zlokarnik, I. (1993), 'Articulatory representations and speech technology', *Language and Speech* **36**(2,3), 331–351.

Schroeder, M. R. (1967), 'Determination of the geometry of the human vocal tract by acoustics measurements', *J. Acoust. Soc. Am.* **41**, 1002–1010.

Schroeter, J. & Sondhi, M. M. (1989), Dynamic programming search of articulatory codebooks, *in* 'Proc. ICASSP', Glasgow.

Schroeter, J. & Sondhi, M. M. (1992), *Speech Coding Based on Physiological Models of Speech Production*, in Furui & Sondhi (1992), chapter 8, pp. 231–268.

Schuster, M. (1999), On Supervised Learning from Sequential Data with Applications for Speech Recognition, PhD thesis, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology.

Shirai, K. & Kobayashi, T. (1986), 'Estimating articulatory motion from the speech wave', *Speech Communication* **5**, 159–170.

Sietsma, J. & Dow, R. J. (1991), 'Creating artificial neural networks that generalize', *Neural Networks* **4**, 67–79.

Swingler, K. (1996), *Applying Neural Networks: A Practical Guide*, Academic Press, London.

Wakita, H. (1973), 'Direct estimation of the vocal tract shape by inverse filtering of acoustic speech waveforms', *IEEE Trans. Audio Electroacoust.* **21**, 417–427.

Wakita, H. (1979), 'Estimation of vocal-tract shapes from acoustical analysis from the speech wave: The state of the art', *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-27**, 281–285.

Welch, G. & Bishop, G. (1995), An introduction to the Kalman filter, Technical Report TR 95-041, Department of Computer Science, University of North Carolina.

Werbos, P. J. (1988), Backpropagation: past and future, *in* 'Proceedings of the IEEE International Conference on Neural Networks', Vol. I, San Diego, pp. 343–353.

Wrench, A. A. & Hardcastle, W. J. (2000), A multichannel articulatory speech database and its application for automatic speech recognition, *in* 'Proc. 5th Seminar on Speech Production', Kloster Seeon, Bavaria, pp. 305–308.

Wrench, A. & Richmond, K. (2000), Continuous speech recognition using articulatory data, *in* 'Proc. ICSLP 2000', Beijing, China.

Zachs, J. & Thomas, T. R. (1994), 'A new neural network for articulatory speech recognition and its application to vowel identification', *Computer Speech and Language* **8**, 189–209.

Ziert, A., Hoole, P., Honda, M., Kaburagi, T. & Tillman, H. G. (2000), Extracting tongues from moving heads, *in* 'Proc. 5th Seminar on Speech Production', Kloster Seeon, Bavaria, pp. 313–316.

Zlokarnik, I. (1994), Untersuchungen zur Spracherkennung unter Verwendung von Messungen der elektromagnetischen Artikulographie, PhD thesis, Technische Universität München, FRG.

Zlokarnik, I. (1995*a*), 'Adding articulatory features to acoustic features for automatic speech recognition', *J. Acoust. Soc. Am.* **97**(5 (Pt.2)), 3246.

Zlokarnik, I. (1995*b*), A speech recognizer using electromagnetic articulography. (English short version of PhD thesis).