Available online at www.sciencedirect.com
ScienceDirect
Speech Communication 49 (2007) 361–383
SPEECH COMMUNICATION
www.elsevier.com/locate/specom

Inverting mappings from smooth paths through R^n to paths through R^m: A technique applied to recovering articulation from acoustics

John Hogden [a,*], Philip Rubin [b], Erik McDermott [c], Shigeru Katagiri [d], Louis Goldstein [e]

[a] M.S. B265, Los Alamos National Laboratory, Los Alamos, NM 87545, USA
[b] Haskins Laboratories, 300 George Street, Suite 900, New Haven, CT 06511, USA
[c] NTT Communications Science Laboratories, NTT Corporation, Kyoto, Japan
[d] Department of Information System Design, Faculty of Engineering, Doshisha University, 1-3 Tatara Miyakodani, Kyotanabe-shi, Kyoto 610-0394, Japan
[e] Haskins Laboratories, 300 George Street, Suite 900, New Haven, CT 06511, USA

Received 6 May 2006; received in revised form 7 February 2007; accepted 12 February 2007

Abstract

Motor theories, which postulate that speech perception is related to linguistically significant movements of the vocal tract, have guided speech perception research for nearly four decades but have had little impact on automatic speech recognition. In this paper, we describe a signal processing technique named MIMICRI that may help link motor theory with automatic speech recognition by providing a practical approach to recovering articulator positions from acoustics. MIMICRI's name reflects three important operations it can perform on time-series data: it can reduce the dimensionality of a data set (manifold inference); it can blindly invert nonlinear functions applied to the data (mapping inversion); and it can use temporal context to estimate intermediate data (contextual recovery of information). In order for MIMICRI to work, the signals to be analyzed must be functions of unobservable signals that lie on a linear subspace of the set of all unobservable signals. For example, MIMICRI will typically work if the unobservable signals are band-pass and we know the pass-band, as is the case for articulator motions. We discuss the abilities of MIMICRI as they relate to speech processing applications, particularly as they relate to inverting the mapping from speech articulator positions to acoustics. We then present a mathematical proof that explains why MIMICRI can invert nonlinear functions, which it can do even in some cases in which the mapping from the unobservable variables to the observable variables is many-to-one. Finally, we show that MIMICRI is able to infer accurately the positions of the speech articulators from speech acoustics for vowels. Five parameters estimated by MIMICRI were more linearly related to articulator positions than 128 spectral energies.

Keywords: Speech inverse problem; Dimensionality reduction; Channel normalization

1. Introduction

In their classic review of speech perception research, Liberman et al. (1967) conclude, "If the listener possesses some device for recovering the articulatory events from their encoded traces in the sound stream, then he should perceive the phonemes well and, indeed, evade several limitations that would otherwise apply in auditory perception". This conclusion is the source of the "motor theory" of speech perception, which has guided human speech perception research for many years.

While the motor theory has had little impact on automatic speech recognition systems, it seems likely that automatic systems that possess a device for recovering articulatory events would see benefits similar to what is expected in human speech perception. In fact, being able to invert the mapping from articulation to acoustics is a topic of interest for automatic speech processing for a

* Corresponding author. Tel.: +1 505 665 0134; fax: +1 505 665 5220.
E-mail addresses: hogden@lanl.gov (J. Hogden), rubin@haskins.yale.edu (P. Rubin), mcd@cslab.kecl.ntt.co.jp (E. McDermott), skatagir@mail.doshisha.ac.jp (S. Katagiri), goldstein@haskins.yale.edu (L. Goldstein).

0167-6393/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.specom.2007.02.008

variety of reasons (McGowan and Faber, 1996; Rose et al., 1996), not the least of which is that being able to recover articulator positions may improve the performance of speech recognition (Blackburn and Young, 2001; Deng, 1998; Deng et al., 1997; Frankel and King, 2001a,b; Hogden et al., 1998; Hogden and Valdez, 2000; Kirchhoff, 1998; McDermott and Nakamura, 2006; Richards and Bridle, 1999; Zlokarnik, 1995).

Of course, the statement that being able to recover articulation would aid in human speech perception or automatic recognition begs for an explanation of how the articulation can be recovered. In their revision of the motor theory, Liberman and Mattingly (1985) conclude that knowledge of the relationship between acoustics and articulation is innate in humans, in part because phenomena were discovered that "differed from anything that association learning could reasonably be expected to produce" – a claim that is echoed in the more recent work of Guenther et al. (1998). In associative learning, which is also known as *supervised* learning and is exemplified by multilayer perceptrons, the learning algorithm must be given paired input/output data in order to learn the mapping from the input to the output. But how can the mapping between acoustics and articulation be learned by people who do not produce speech, e.g., prelinguistic infants or people who are incapable – from birth – of controlling their articulators? While Guenther et al. agree that humans cannot be using associative learning to arrive at the perception-production link, they also argue that the perception-production link cannot be innately specified because the mapping differs between individuals and changes over time within individuals. Blesser's work (1972) showing that people learn to understand "spectrally inverted" speech (speech in which the energy at low frequencies is transformed into energy at high frequencies and vice versa) makes it even more difficult to understand how the perception-production link could be innate. The argument that people do not learn the perception-production link by association, combined with the argument that the link between acoustics and production cannot be innate, would seem to suggest that the perception-production link is not used in human perception.

There is, however, a flaw in the reasoning presented above: it is possible to use blind algorithms instead of association to learn the relationship between two signals, e.g., acoustics and articulation. The word "blind" is used in the same sense as used by Beneviste and Goursat (1984) and more recent publications, e.g., (Qiu and Hua, 1997) – given two sequences, $\boldsymbol{x}$ and $\boldsymbol{y}$, the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$ must be discerned from observations of $\boldsymbol{y}$ without the need to know $\boldsymbol{x}$ or time-align $\boldsymbol{x}$ and $\boldsymbol{y}$. Blind algorithms, sometimes called *unsupervised* algorithms, are not associative learning algorithms because they do not need to be given examples of $\boldsymbol{x}$ to learn the mapping between $\boldsymbol{x}$ and $\boldsymbol{y}$. If we assume that people use blind algorithms to learn the mapping between acoustics and articulation, then we can explain how people who never produce speech can learn the mapping between acoustics and articulation, and how people can learn to understand distorted speech for which an innate map seems unlikely.

In this paper, we apply insights gained from speech perception research to the problem of automatic speech recognition. The focus in this paper is to use one particular blind algorithm – MIMICRI (Hogden et al., 2003) – to help solve three important problems in signal analysis: blind function inversion, dimensionality reduction, and using temporal context – the preceding and subsequent values of the signal – to infer the current value of a signal. Each of the three problems is important on it own, but taken together they offer a potential solution to the long-standing problem of learning the relationship between acoustics and articulation. Although many proposals have been made for using articulatory-based representations to improve ASR (see McDermott and Nakamura, 2006 for a survey of such work), MIMICRI offers a much deeper theoretical insight into the relation between acoustics and articulators. Whereas most articulatory modeling for ASR has used articulatory features obtained from theoretical considerations, or trajectory modeling with minimal smoothing constraints, MIMICRI allows one to make very specific use of continuity constraints in relating acoustics to articulators. This relationship is intrinsic to the nature of speech, and not merely imposed by the engineer. We believe that those who come to understand the MIMICRI framework will be able to exploit this gain in understanding along lines similar to previous work in articulatory-based ASR, but in a much more principled manner.

In the remainder of this introduction, we give an outline of what can be found in the rest of the paper, highlighting some of the more important points. The outline includes a brief description of the three problems MIMICRI addresses and some mathematical background that is important for all of the subsequent sections (the detailed mathematics can be found in Section 2). We direct the reader to subsequent sections for more complete information on each of the topics.

Section 1.1 discusses the first problem MIMICRI addresses: blind function inversion, in which we can only observe a nonlinearly transformed version of a signal for which we would like to measure the amplitude over time. An example of this type of problem is trying to infer the input to a nonlinear microphone when we can only observe the output. Our goal will be to infer and invert the nonlinear function using known constraints on the unobservable data. Section 1.2 discusses the second problem: dimensionality reduction. Dimensionality reduction arises, for example, when we are given a series of short-time spectra of a signal and would like to characterize the spectra using as few parameters as possible. Section 1.3 discusses the use of temporal context. Each of these problems arises in many application areas, but many applications require the solution to only some of the problems. For example, we may want to invert the effect of a nonlinear microphone on a one-dimensional signal (such as a sound wave) but not be

concerned with reducing the dimensionality of the signal (since it has only one dimension). In contrast, the mapping between articulator positions and acoustics involves function inversion, dimensionality reduction, and, in some cases, use of context information to invert many-to-one mappings. Section 3 discusses our application of MIMICRI to inverting the mapping from acoustics to articulation.

As discussed more thoroughly below, MIMICRI approaches each of the problems as an inversion problem in which there is some memoryless function, $f: R^n \rightarrow R^m$, $f(x(t)) = y(t)$, that converts an unobservable signal, $\boldsymbol{x} = [x(1),x(2),\ldots x(T)]'$, to an observable signal, $\boldsymbol{y} = [y(1), y(2),\ldots y(T)]'$. For pedagogical reasons, assume that both n and m equal 1 unless otherwise stated, which allows us to treat signals as standard vectors in a Euclidean space. Cases in which $n < m$, i.e, dimensionality reduction, are discussed in Sections 1.2 and 2.6.

We will not discuss functions with memory, in which $y(t)$ may be a function of x at times other than t, i.e., $y(t) = f(x(t),x(t-1),x(t-2)\ldots)$, but a good discussion and bibliography can be found in (Tsimbinos, 1995). While the assumption that the function is memoryless is often not strictly true, it can be a useful approximation. For example, in modeling the mapping from articulation to acoustics, researchers typically assume that a sufficiently short acoustic window centered at t can be treated as being the result of the articulator configuration at t.

The proof presented in Section 2 shows that if $\boldsymbol{x} = \boldsymbol{C}\boldsymbol{v}$ where $\boldsymbol{C}$ is a matrix with more rows than columns (and other minor constraints discussed in Section 2.3), $\boldsymbol{v} \in R^d$ is a column vector, $d < T$, and if we can find a memoryless function, $g: R \rightarrow R$, which transforms every possible observable trajectory $\boldsymbol{y}$ to a new signal $\hat{\boldsymbol{x}} = [g(y(1)), g(y(2)),\ldots,g(y(T))]'$ such that $\hat{\boldsymbol{x}} = \boldsymbol{C}\hat{\boldsymbol{v}}$, then $g()$ inverts $f()$ to within an affine transformation. The precise meaning of this statement is given in Section 2, but for many applications we can think of $g()$ as the inverse of $f()$. We call any process that finds a $g()$ meeting this requirement, i.e., in which the constraints on $\hat{\boldsymbol{x}}$ mimic the constraints on $\boldsymbol{x}$, MIMICRI. Note that MIMICRI is blind in that it never requires $[x(t),y(t)]$ pairs to learn $g()$, but does require that we know the constraint matrix, $\boldsymbol{C}$. Recovering $\boldsymbol{x}$ is merely a side-effect of making $\hat{\boldsymbol{x}}$ mimic the constraints on $\boldsymbol{x}$.

The proof given in Section 2 shows us how to recognize when we have found a $g()$ that inverts $f()$ to within an affine transformation, but does not specify the algorithm needed to find $g()$. In practice, the implementation of MIMICRI will change depending on the parameterization of $g()$. An example in which $g()$ is chosen from a small class of functions is given at the beginning of Section 2.4. We use a more general parameterization of $g()$ in the computational formula we give in Section 2.7. The parameterization used in Section 2.7 only assumes that $g(y(t))$ can be approximated by $g(a_i)$, where a_i is a quantized version of $y(t)$. Increasing the number of quantization levels allows us to approximate increasingly general functions by making a_i a better approximation of $y(t)$, and $g(a_i)$ a better approximation of $g(y(t))$. With this parameterization, $g()$ does not need to be continuous but must have only a finite number of discontinuities (unless we have an infinite number of quantization levels). MIMICRI has also been implemented for this very general parameterization of $g()$ using an iterative approach (Hogden, 2000) similar to the well known EM algorithm (Dempster et al., 1977).

In Section 1.1, we discuss how $\boldsymbol{C}$ can be used to characterize fairly general constraints on signals. In particular, a set of bandlimited $\boldsymbol{x}$ can be characterized by correctly choosing $\boldsymbol{C}$. Given that $\boldsymbol{C}$ can be used to characterize bandlimited signals, it may be clearer, albeit not quite as accurate, to characterize MIMICRI by saying that if we know the pass-band of the signal we cannot observe, $\boldsymbol{x}$, then we can invert any memoryless, invertible nonlinearity applied to $\boldsymbol{x}$ by finding a transform of the signal we can observe, $\boldsymbol{y}$, so that our transformed $\boldsymbol{y}$ has the same pass-band as $\boldsymbol{x}$. The caveats are that we need to be able to see many examples of $\boldsymbol{y}$, and we can only invert to within an affine transformation.

Finding the mapping from $\boldsymbol{y}$ to $\boldsymbol{x}$ is not necessarily the same as inverting $f()$ because $\boldsymbol{y}$ can contain much more information than $y(t)$. For example, if $f()$ is many-to-one, we cannot recover $x(t)$ from $y(t)$ alone, but we may be able to exploit the sequence information in $\boldsymbol{y}$ to infer $x(t)$. A very large body of work exists on inferring $\boldsymbol{x}$ in the presence of many-to-one mappings (Badin et al., 1995; Dusan and Deng, 2000; Hiroya and Honda, 2004; Kuc et al., 1985; McGowan and Lee, 1996; Moody, 1999; Ouni and Laprie, 2005; Richards et al., 1997; Roweis and Alwan, 1997; Schroeter and Sondhi, 1994; Shirai and Kobayashi, 1986; Sorokin and Trushkin, 1996; Suzuki et al., 1998), and we give a more complete discussion of many-to-one mappings in Sections 1.3 and 2.5. MIMICRI has been shown to be able to invert complicated, discontinuous, and even many-to-one mappings (Hogden et al., 2003). However, unlike other approaches, which assumed that $f()$ was either known, approximated using simulations, or learned via a supervised approach, MIMICRI infers $f()$ blindly.

1.1. Blind function inversion

A common problem in signal processing is that unwanted nonlinearities can be introduced by signal transmission or transduction. For example, it has been shown that microphone nonlinearities can negatively affect the performance of speaker recognition systems (Reynolds, 1996), and that compensating for microphone nonlinearities can improve performance in speaker recognition (Quatieri et al., 2000).

Practical approaches to blind inversion use assumptions about $\boldsymbol{x}$ to allow the inversion. In an example of the kind of knowledge that can be used to invert nonlinearities, the work of Morris and Clements (2001) shows that a class of memoryless, nonlinear functions applied to speech

signals can be approximately inverted by assuming that all x belong to the class of autoregressive signals. Levin (2002) also describes a system that can be portrayed as blind nonlinearity inversion for memoryless functions, although Levin views it in a very different light. Continuing our notation, i.e., $f(x(t)) = y(t)$, Levin transforms y to a "canonical" signal, z, which is invariant to $f()$, by using a transform that is dependent on y. If we let $\boldsymbol{Z}$ be the set of all signals that can be output by Levin's technique, then it is easier to see the connection between Levin's technique and a blind function inverter. (See Levin's work for more information about $\boldsymbol{Z}$.) If we apply Levin's technique to any signal, x, drawn from $\boldsymbol{Z}$, then x will be unchanged. If we take x and transform it into y by any memoryless invertible nonlinearity and then apply Levin's technique to y, the output will be x, so the nonlinearity is inverted. Levin's technique, therefore, can be thought of as a blind inversion algorithm when the set of input signals is drawn from $\boldsymbol{Z}$. Since $\boldsymbol{Z}$ is not a particularly good characterization of speech signals, Levin's technique is not a blind function inverter for speech signals. Nonetheless, Levin's technique may be sufficient for finding canonical representations of speech signals, which is, in fact, all that Levin claims.

Bendat (1998) uses information about the probability density function (PDF) of x to invert memoryless functions. To do the inversion, Bendat transforms y to $\hat{x}$ in such a way that the PDF of elements in $\hat{x}$ matches the PDF of the elements in x. Bendat's approach was used to improve speaker recognition when the testing segments were subjected to nonlinear transformations (Balchandran and Mammone, 1998).

MIMICRI's basic assumption – in addition to $f()$ being memoryless – is that x lies in a linear subspace of the set of all signals. That is, $\boldsymbol{x} = \boldsymbol{C}\boldsymbol{v}$, where $\boldsymbol{v} \in R^d$ is a column vector, $d < T$, and $\boldsymbol{C}$ is a matrix that contains our knowledge about what signals can be produced. To better understand how $\boldsymbol{C}$ contains our knowledge about x, recall that x is the weighted sum of the columns of $\boldsymbol{C}$. One of the many possible examples of this constraint is shown pictorially in Fig. 1. In Fig. 1 we see that the columns of the $\boldsymbol{C}$ matrix are sine waves. The vector $\boldsymbol{v}$ comprises three values (the same as the number of columns in $\boldsymbol{C}$), and x is the sum of the sines weighted by the values of $\boldsymbol{v}$. Although a particular set of values is given for $\boldsymbol{v}$ in this example (1/6, 1/3, and 1/2), the set of all possible x values, given the constraints in this $\boldsymbol{C}$, is constructed by varying $\boldsymbol{v}$ over all possible real values. In this case, we have chosen $\boldsymbol{C}$ to fit the assumption that x can be any signal that is the weighted sum of the sine waves in $\boldsymbol{C}$. By correctly choosing $\boldsymbol{C}$, we can characterize many constraints, including the constraint that x trajectories have all their energy in a particular frequency band.

As discussed further in Section 2.3, MIMICRI does not work with any arbitrary $\boldsymbol{C}$, but the constraints on $\boldsymbol{C}$ are very weak.

In contrast to the signals discussed above, which are memoryless mappings from a scalar valued $x(t)$ to a scalar valued $y(t)$, the mapping from articulator positions to acoustics is typically thought of as a memoryless mapping from a vector of articulator positions to a vector of acoustic features (Coker, 1976; Fant, 1970; Flanagan, 1972; Maeda, 1979; Markel and Gray, 1976; McGowan, 1987; Mermelstein, 1973; Oppenheim, 1969; Rubin et al., 1981; Schroeder, 1967; Scully, 1979; Wakita, 1973). As discussed in Sections 1.2 and 2.6, MIMICRI's theory is also applicable to mappings from vector spaces to vector spaces (and finite sets of vectors to finite sets of vectors). Given that MIMICRI can invert mappings from vectors to vectors and that speech signals are produced by articulator motions that are bandlimited (Muller and McLeod, 1982; Nelson, 1977; Nix, 1998), it is reasonable to try to use MIMICRI to invert the mapping from articulation to acoustics. In fact, we show results of doing so in Section 3.

1.2. Function inversion as dimensionality reduction

As discussed in Section 2.6, the proof of MIMICRI can be extended to allow $x(t)$ to be a vector in an n-dimensional space, $x(t) \in R^n$, and $y(t)$ to be a vector in an m-dimensional space, $y(t) \in R^m$, where $m \geqslant n$. The ability to invert such a function makes MIMICRI a dimensionality-reduction technique as well as a function-inversion technique.

Dimensionality-reduction is a first stage in many speech-processing applications. For speech recognition, speaker recognition, and speech compression, speech has often been characterized by 20 or so acoustic parameters, e.g. Mel-cepstra, for each time-window of speech, with about 50 windows per second. These acoustic parameters are a reduced dimensionality representation of the windowed speech but are not known to be optimal for any application. We cannot rule out the possibility that some other

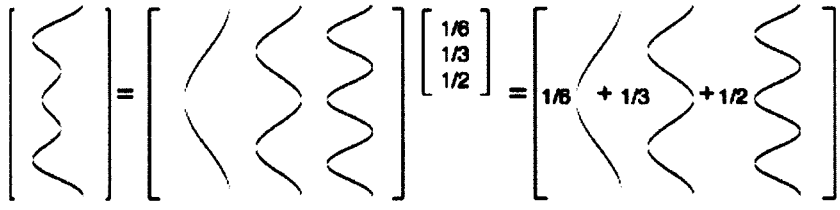

Fig. 1. One variant of the constraint equation, $\boldsymbol{x} = \boldsymbol{C}\boldsymbol{v}$, is shown pictorially here. The unobservable signal, x, shown on the left, is the sum of the columns of the $\boldsymbol{C}$ matrix (shown as sines) weighted by the values in the vector $\boldsymbol{v}$, i.e., 1/6, 1/3, and 1/2. By varying the elements of $\boldsymbol{v}$, we can get all the signals that are possible under this constraint.

acoustic parameters may capture more relevant information in fewer parameters. A review of previously cited papers about inverting the articulatory-to-acoustics mapping and papers on modeling speech production (Beautemps et al., 2001; Coker, 1976; Gupta and Schroeter, 1993; Hirayama et al., 1992; Kaburagi and Honda, 2001; Perrier et al., 1997; Rubin et al., 1981; Saltzman and Munhall, 1989) suggests that speech production can be sufficiently described using only 8–10 articulators. If this belief is correct, the phonetic information in speech can be captured using about 8–10 parameters, and the mapping from articulation to commonly used acoustic parameters is a mapping from, say, R^{10} to R^{20} – a large increase in dimensionality.

The consequences of increasing the dimensionality of a data set can be significant. Since high-dimensional mappings are difficult to visualize, suppose articulation could be represented by one parameter, say the height of the tongue, and acoustics by two parameters. The mapping from articulation to acoustic would then be R^1 to R^2. An example of such a mapping is shown in Fig. 2. In this example, articulator positions vary from 0 to 10 and are mapped into the parameters $[y_1(t), y_2(t)]$ using the equations $y_1(t) = \log(x(t) + 1)$ and $y_2(t) = x^2(t)$. Note that the mapping does not increase the intrinsic dimensionality of the articulation. Instead, the articulator positions are mapped to a one-dimensional curve embedded in the two-dimensional acoustic space. Conversely, our one-articulator vocal tract cannot make any sounds not on the curve. Similarly, if the mapping from articulation to acoustics is a mapping from R^{10} to R^{20}, the set of acoustic feature vectors produced by speaking can be represented by a surface "the constraint surface" with no more than 10 intrinsic dimensions embedded in the 20-dimensional acoustic feature space (Kimber, 1994). Any data point that lies off the surface must be the result of acoustic noise.

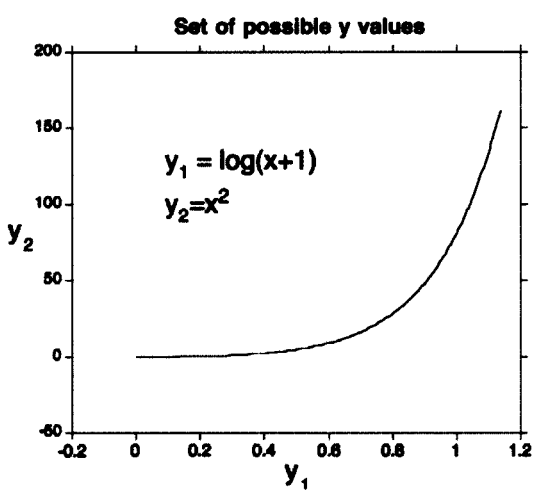

Fig. 2. This is a simple mapping from R^1 to R^2. The set of $[y_1, y_2]$ positions lies on a one-dimensional curve embedded in the two-dimensional space. This illustrates the general principle that mapping to a higher-dimensional region does not increase the intrinsic dimensionality.

Suppose we performed a common first step in speech recognition, i.e., estimating PDFs over acoustic space. In the R^1 to R^2 example, such an approach would mean trying to estimate the probability of points on a one-dimensional curve using two-dimensional PDFs. Not only would this approach mean that we would have to estimate more PDF parameters when working in the high-dimensional space, but virtually none of the points covered by the two-dimensional PDFs would be on the curve, and, therefore, they would have to be the result of acoustic noise. In high-dimensional acoustic spaces, the PDFs must be largely estimating the probability of the acoustic noise background. As the background noise changes, we should expect a change in the performance of the recognizer.

MIMICRI evolved from a dimensionality-reduction technique (Hogden, 1991, 1995; Hogden et al., 1992, 1996a, 1993). Like other nonlinear dimensionality reduction techniques (Carreira-Perpinan, 2001; Kambhatle and Leen, 1997; Roweis and Saul, 2000; Tenenbaum et al., 2000), MIMICRI is able to find manifolds in high-dimensional spaces. However, dimensionality-reduction algorithms do not typically do function inversion, so the solution obtained by dimensionality-reduction algorithms changes when different nonlinearities are applied to a signal. Since MIMICRI inverts nonlinear functions, MIMICRI should give the same solution (to within an affine transformation) regardless of nonlinear transformations, as long as the assumptions are met. Dimensionality reduction without the ability to invert nonlinear transformations has obvious disadvantages when we need to normalize for nonlinear microphones or for differences between speakers.

1.3. Function inversion as a means to recover contextual information

As mentioned in Section 1.1, a many-to-one $f()$ does not preclude being able to infer $\boldsymbol{x}$ from $\boldsymbol{y}$. A simple example makes this clear and introduces the relationship between a many-to-one mapping and the need to use context. Suppose that we have a many-to-one $f: R \rightarrow R$, such as $y(t) = f(x(t)) = x(t)^2$. Given the observable sequence $\boldsymbol{y} = [1,4,9]$, there are eight possible sequences of $x(t)$ values that could have produced the observable sequence. However, if we know that the domain is restricted such that $x(t) = x(t-1) + 1$, then we can infer that $\boldsymbol{x} = [1,2,3]$. In this case, our prior knowledge and context tell us that $f^{-1}(9) = 3$. Note that $f^{-1}(9)$ does not always equal 3 even given the constraint, because $\boldsymbol{y} = [9,4,1]$ must map to $\boldsymbol{x} = [-3,-2,-1]$. Therefore, while $f(x(t))$ is many-to-one, the mapping from $\boldsymbol{x}$ to $\boldsymbol{y}$ is invertible, given the restricted domain. In fact, the constraint is sufficient to infer the value of $x(3)$ that produced the signal $y(3)$ in $\boldsymbol{y} = [1,4,y(3)]$. Note that this case provides an example of using context to recover information – one of the tasks that MIMICRI can perform.

In order to use context, we need to know the relationships between data values over time. For example, suppose

that $y(t)$ is an n-dimensional vector that is a nonlinear function of the preceding d vectors. If we know the function, then we can use the preceding observations to predict the next, i.e., we can use context information. However, learning the function can be very difficult without constraints on the problem. If we have to infer the nonlinear function from training data, the difficulty of inferring the function increases rapidly with both d and n. Furthermore, unless we can restrict the class of functions (exponential? polynomial?) using some prior knowledge, the amount of data required to learn the function can be prohibitive. MIMICRI restricts the class of functions to memoryless functions that transform y values to x values where the x values have a known constraint. MIMICRI's constraints rule out many functions and will give bad results if the assumptions are incorrect, but the constraints make the problem of inferring the function manageable if the assumptions are correct.

Although we have only mathematically proven that MIMICRI can invert one-to-one mappings, simulations demonstrate that some many-to-one functions can be inverted with MIMICRI's assumption (Hogden et al., 2003). MIMICRI's ability to learn a mapping that allows us to use context makes MIMICRI especially applicable for speech processing. It is often strongly asserted, based on theoretical and empirical evidence, that the mapping from articulation to acoustics is many-to-one, implying that we need to be able to use context information. The issue of whether the mapping is many-to-one has been addressed in detail elsewhere (Atal et al., 1978; Boe et al., 1992; Fant, 1970; Flanagan, 1972; Hogden et al., 1996b; Schroeter and Sondhi, 1994; Sondhi and Resnick, 1983). Ultimately, conclusions from empirical tests of whether the mapping is many-to-one are contingent on how different two signals can be and still be considered the same. For example, should two acoustic segments be considered the same if their formant frequencies vary by less than 3 standard deviations (Lindblom et al., 1979); if they are perceived as the same vowel (Perkell et al., 1993); if humans cannot reliably distinguish them (Fowler and Turvey, 1980); if they are clustered into the same category by an acoustic clustering algorithm (Roweis, 1999); or if they are acoustically identical, in which case we never see the same signal twice? If we use a definition of "same" that includes more acoustic signals, then we can reasonably expect more articulator configurations to produce the same signal. Similar problems exist for determining the similarity of articulator positions, since a tiny change in the position of the tongue tip near the roof of the mouth is far more phonetically significant than the same magnitude of movement of the tongue tip well inside the mouth.

While empirical results regarding many-to-one mappings are difficult to interpret unambiguously, we find theoretical arguments for a many-to-one mapping for a few phoneme classes very compelling. For example, Moody (1999) suggests that recovery of articulator positions from acoustics can be achieved with good accuracy for vowels, but, since different stop consonants create segments of silence, the mapping from articulator positions to acoustics is many-to-one for stop-consonant closures. It has also been noted that unvoiced fricatives give little information about articulator positions behind the constriction (Richards et al., 1997), so the mapping from articulation to acoustics for unvoiced fricatives is also likely to be many-to-one or approximately many-to-one, at least for some articulators that are not critical (Papcun et al., 1992).

Lindblom (1996) argues that "parsing the signal into its articulatory components seems unnecessary because, if it contains enough distinctive (auditory) information to permit recovery of gestures, it also ought to contain enough (auditory) information to permit lexical access without going by way of articulation. Why make a detour unless articulatory information is richer and more distinctive than acoustic information?" However, Lindblom's question ignores the difficulty of learning the relationship between between data preceding t and the data at t. As discussed above, we need to learn relationships between observable data values in order to use the context information in the sequence, and prior knowledge can simplify the learning problem tremendously. If we had prior knowledge that told us what constraints are most appropriate for speech acoustic data, then it would make sense to apply the constraints for processing speech acoustics. However, while we know relatively little about acoustic constraints on speech, we know that the articulators are constrained to follow physical laws. The physical laws constrain the relationships between observable data values, thereby simplifying the problem of using context. The ability to use the constraints imposed by physical laws is one reason for detouring to use articulator information. Since MIMICRI allows us to use constraints approximating physical laws (such as the constraint that articulator motions are bandlimited) the use of MIMICRI is an attempt to use the physical laws to make the detour learnable.

2. Mathematical development

As stated above, MIMICRI assumes there is some memoryless function, $f: R^n \rightarrow R^m$, $f(x(t)) = y(t)$, that converts an unobservable signal, $\boldsymbol{x} = [x(1), x(2), \ldots, x(T)]'$, to an observable signal, $\boldsymbol{y} = [y(1), y(2), \ldots, y(T)]'$. MIMICRI's goal is to find a memoryless, possibly nonlinear function, $g: R^m \rightarrow R^n$, which transforms every observable trajectory $\boldsymbol{y}$ to

$$\hat{\boldsymbol{x}} = [g(y(1)), g(y(2)), \ldots, g(y(T))]' \tag{1}$$

and which approximates, in a reasonable sense, the inverse of $f()$. We prove that we can achieve this goal if we can find a $g()$ such that there exists, for every possible $\boldsymbol{y}$ (recall, from Section 1.1, that $\boldsymbol{x} = \boldsymbol{C}\boldsymbol{v}$, so not all $\boldsymbol{y}$ values are possible) a $\hat{\boldsymbol{v}}$ for which $\hat{\boldsymbol{x}} = \boldsymbol{C}\hat{\boldsymbol{v}}$.

If $g()$ is the true inverse of $f()$, then $h \equiv g \circ f$ is the identity mapping, $h(x) = x$. However, we will be satisfied with proving that $h(x)$ must be affine. That is,

$$h(x) = ax + b. \tag{2}$$

Since an affine function is a linear function with an additive constant, we prove that $h(x)$ is affine by proving that

$$\tilde{h}(x) \equiv h(x) - h(0) \tag{3}$$

is linear, which implies that $h(x)$ is affine in general and linear in those cases in which $h(0) = 0$.

In Section 2.1, we prove that $\tilde{h}$ is necessarily linear by showing that

$$\tilde{h}(z_1 + z_2) = \tilde{h}(z_1) + \tilde{h}(z_2) \tag{4}$$

for all real values of z_1 and z_2, i.e., when $\tilde{h} : R \rightarrow R$. Eq. (4) is known as Cauchy's functional equation and is discussed in a large body of literature. In particular, functions which satisfy (4) are linear under physically realistic conditions, e.g., if $\tilde{h}(x)$ is continuous at any point and satisfies (4), it is provably linear (Aczel and Dhombres, 1989). Furthermore, the proof that Cauchy's functional equation implies linearity does not hold just for cases in which the values of z_1 and z_2 are real, but extends to many classes of values. While we have not taken the step of clearly delineating all the cases in which our result holds, it is of particular interest that the results can be extended to cases in which the elements of $\boldsymbol{x}$ are themselves vectors, as in the case in which $x(t)$ comprises measurements of different signals, all of which are bandlimited although not necessarily with the same pass-band.

Our proof that $h()$ is necessarily affine does not take into account all of the equations implicit in our assumptions. Unfortunately, using the additional equations does not further constrain the class of $h()$. We show in Section 2.2 that any affine $h()$ will meet the assumptions, so we can only invert $f()$ to within an affine transformation.

Our proof does not hold for all $\boldsymbol{C}$, so, in Section 2.3, we discuss the constraints on $\boldsymbol{C}$. The constraints are weak and are unlikely to be of practical concern. While the proof places very minimal constraints on $h()$ – that $h()$ is continuous at some point – we can only prove that we have inverted $f()$ if we have an infinite amount of data – the $\boldsymbol{y}$ values for every $\boldsymbol{v}$. In any realistic setting, we can expect that $f()$ is at least piecewise continuous with a relatively small number of discontinuities and, therefore, can be inverted by a function with the same qualities. If we know more about $f()$, it is reasonable to expect that we will need less data to infer $g()$. Thus, in Section 2.4, we discuss how much data is likely to be required for realistic problems. Finding a $g()$ that meets the requirements of the proof can be done without understanding the proof. Those who wish to implement MIMICRI can find an algorithm in Section 2.7.

2.1. Proof that $h()$ is necessarily affine

Assume the columns of $\boldsymbol{C}$ are orthogonal so that:

$$\boldsymbol{C}'\boldsymbol{C} = \boldsymbol{I} \text{ and } \boldsymbol{I} \text{ is the identity matrix.} \tag{5}$$

If not, then replace $\boldsymbol{C}$ by a matrix whose columns form an orthogonal basis for the column space of $\boldsymbol{C}$. Replacing $\boldsymbol{C}$ by an orthogonal matrix with the same column space changes the values of $\boldsymbol{v}$ used to get $\boldsymbol{x}$ values, but can always be done and does not affect the generality of our argument. Call the number of columns in $\boldsymbol{C}$ (which is also the dimensionality of $\boldsymbol{v}$) d. By construction, the rank of $\boldsymbol{C}$ is equal to the number of columns, and the number of columns is less than the number of rows. Since the dimensionality of the row space is equal to the dimensionality of the column space (Strang, 1980), $\boldsymbol{C}$ must have d linearly independent rows and T–d rows that are linear combinations of the other rows. Put simply, $\boldsymbol{C}$ is a matrix with more rows than columns, and such matrices have some rows that are linear combinations of others.

Find a set of linearly independent rows of $\boldsymbol{C}$ that form a basis of the row space and call the set of indices for those rows I. (Note that $\boldsymbol{I}$, not I, is being used as the identity matrix) Let I^c be the set of row indices not in I. All rows with indices that are in I^c can be written as linear combinations of rows with indices in I, therefore, letting r be an index in I^c, we write the element in row r, column j of $\boldsymbol{C}$ as:

$$c_{rj} = \sum_{i \in I} \alpha_{ri} c_{ij}. \tag{6}$$

The dot product of row r of $\boldsymbol{C}$ with $\boldsymbol{v}$ gives $x(r)$, therefore:

$$x(r) = \sum_j c_{rj} v_j \tag{7}$$

where v_j is the jth element of $\boldsymbol{v}$. Substituting (6) into (7) and manipulating gives:

$$x(r) = \sum_j v_j \sum_{i \in I} \alpha_{ri} c_{ij} = \sum_{i \in I} \alpha_{ri} \sum_j v_j c_{ij} = \sum_{i \in I} \alpha_{ri} x(i). \tag{8}$$

Since $\hat{\boldsymbol{x}}$ is also the product of $\boldsymbol{C}$ and a vector, (8) must also hold for $\hat{\boldsymbol{x}}$. So we can conclude that

$$\hat{x}(r) = \sum_{i \in I} \alpha_{ri} \hat{x}(i). \tag{9}$$

Having already defined the function $h \equiv g \circ f$, we note that:

$$\hat{x}(i) = g(y(i)) = g(f(x(i))) = g \circ f(x(i)) = h(x(i)), \tag{10}$$

which can be substituted into the right-hand side of (9) to get:

$$\hat{x}(r) = \sum_{i \in I} \alpha_{ri} h(x(i)). \tag{11}$$

Substituting (10) into the left-hand side of (11) gives:

$$h(x(r)) = \sum_{i \in I} \alpha_{ri} h(x(i)). \tag{12}$$

Substituting (8) into the left side of (12) gives:

$$h\left(\sum_{i \in I} \alpha_{ri} x(i)\right) = \sum_{i \in I} \alpha_{ri} h(x(i)). \tag{13}$$

Eq. (13) is actually T–d equations – one equation for each $r \in I^c$. This fact is used in Section 2.4, but only one value of

r, i.e., one of the T–d equations, is needed to complete the proof. Furthermore, if we find a $g()$ such that $\hat{\boldsymbol{x}}$ is in the column space of $\boldsymbol{C}$ for every $\boldsymbol{x}$, then (13) is true for all real values of $x(i)$, $i \in I$.

Since (13) is true for all values of $x(i)$, it must be true for $x(i) = 0$ for all i, producing the equation:

$$h(0) = \sum_{i \in I} \alpha_{ri} h(0) = h(0) \sum_{i \in I} \alpha_{ri}. \tag{14}$$

From (14) we can conclude that if there exists one $r \in I^c$ for which $\sum_{i \in I} \alpha_{ri} \neq 1$, then $h(0)$ must be zero. Given this fact, it might seem unlikely that we would see a case in which $h(0) \neq 0$, but there is an important class of signals for which $h(0) \neq 0$. Suppose we want to define a class of signals with a component at 0 Hz. To create signals in this class, we could set every entry in one column of $\boldsymbol{C}$ to the same value, i.e., $c_{rj} = c_{ij} \neq 0$ for all r and k and for one j. In this case, (6) gives $c_{rj} = \sum_{i \in I} \alpha_{ri} c_{rj} = c_{rj} \sum_{i \in I} \alpha_{ri}$, which implies that $\sum_{i \in I} \alpha_{ri} = 1$ for all $r \in I^c$. So, if $\boldsymbol{x}$ can have a frequency component at 0 Hz, then $h(0) \neq 0$.

To prove (4), find an expression for $\tilde{h}(z_1)$ for all real z_1. Let p be a value for which $\alpha_{rp} \neq 0$ and let $I^{\bar{p}}$ be the set I with p removed. The definition in (3) allows us to write:

$$\tilde{h}\left(\sum_{i \in I} \alpha_{ri} x(i)\right) = h\left(\sum_{i \in I} \alpha_{ri} x(i)\right) - h(0). \tag{15}$$

Substituting the values $x(p) = \frac{z_1}{\alpha_{rp}}$ and $x(i) = 0 \ \forall i \in I^{\bar{p}}$ into (15) gives:

$$\tilde{h}(z_1) = h\left(\alpha_{rp} \frac{z_1}{\alpha_{rp}} + \sum_{i \in I^{\bar{p}}} \alpha_{ri} x(i)\right) - h(0). \tag{16}$$

Applying (13), which holds for all $x(i)$, to the right-hand side of (16) gives:

$$\begin{aligned} \tilde{h}(z_1) &= \alpha_{rp} h\left(\frac{z_1}{\alpha_{rp}}\right) + \sum_{i \in I^{\bar{p}}} \alpha_{ri} h(0) - h(0) \\ &= \alpha_{rp} h\left(\frac{z_1}{\alpha_{rp}}\right) + h(0)\left(\sum_{i \in I} \alpha_{ri} - \alpha_{rp} - 1\right). \end{aligned} \tag{17}$$

Similarly, let $q \neq p$ be a value for which $\alpha_{rq} \neq 0$. (We require that at least one $r \in I^c$ has two or more values of j for which $\alpha_{rj} \neq 0$; see Section 2.3) Let $I^{\bar{q}}$ be the set I with q removed. Use the particular choice $x(q) = \frac{z_2}{\alpha_{rq}}$ and $x(i) = 0 \ \forall i \in I^{\bar{q}}$ to derive:

$$\tilde{h}(z_2) = \alpha_{rq} h\left(\frac{z_2}{\alpha_{rq}}\right) + h(0)\left(\sum_{i \in I} \alpha_{ri} - \alpha_{rq} - 1\right). \tag{18}$$

To find an expression for $\tilde{h}(z_1 + z_2)$, let $I^{\bar{p}\bar{q}}$ be the set I without p and q and choose the values $x(p) = \frac{z_1}{\alpha_{rp}}$, $x(q) = \frac{z_2}{\alpha_{rq}}$, and $x(i) = 0 \ \forall i \in I^{\bar{p}\bar{q}}$. This approach produces the equation

$$\begin{aligned} \tilde{h}(z_1 + z_2) &= \alpha_{rp} h\left(\frac{z_1}{\alpha_{rp}}\right) + \alpha_{rq} h\left(\frac{z_2}{\alpha_{rq}}\right) \\ &\quad + h(0)\left(\sum_{i \in I} \alpha_{ri} - \alpha_{rp} - \alpha_{rq} - 1\right). \end{aligned} \tag{19}$$

If one recalls that (14) implies that $h(0) = 0$ if $\sum_{i \in I} \alpha_{ri} \neq 1$, it is a simple matter to show that adding (17) and (18) gives (19) regardless of the value of $\sum_{i \in I} \alpha_{ri}$. This step completes the proof that $\tilde{h}(z_1 + z_2) = \tilde{h}(z_1) + \tilde{h}(z_2)$, implying that $h(x)$ is affine in general and linear in the special case where $h(0) = 0$.

2.2. No additional information can be inferred about $h()$

Recall that (13) is actually several equations. If $f()$ is invertible, then there is at least one solution to all of the equations shown in (13) because setting $g = f^{-1}$ will make $h()$ be the identity transform, which is clearly a solution. We have proven that $h()$ must be affine by only considering one of these equations, but it might be hoped that requiring $g()$ to satisfy all of these equations might restrict $h()$ to a smaller class of functions. Unfortunately, even given an infinite amount of data, this is not the case. Let $h(x) = ax + b$. Substituting this general affine equation into (1) gives:

$$\hat{\boldsymbol{x}} = a \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_T \end{bmatrix} + b = \boldsymbol{C}\boldsymbol{a}\boldsymbol{v} + b. \tag{20}$$

If b is 0, then it is clear that $\hat{\boldsymbol{x}} = \boldsymbol{C}\hat{\boldsymbol{v}}$ with $\hat{\boldsymbol{v}} = \boldsymbol{a}\boldsymbol{v}$, which is the constraint we are trying to meet. If b is not 0, as in the case in which $\boldsymbol{C}$ has a column, j, for which $c_{rj} = c_{kj}$ for all r and k, then the b value added to each row of $\boldsymbol{C}\boldsymbol{a}\boldsymbol{v}$ to get $\hat{\boldsymbol{x}}$ can be constructed by multiplying $\boldsymbol{C}$ by a special vector with its jth element being b/c_{rj} and the other elements all being 0. In this case, $\hat{\boldsymbol{v}}$ would be $\boldsymbol{a}\boldsymbol{v}$ plus this special vector.

2.3. The proof does not hold for all C

The discussion following (17) indicates that there exist values of $\boldsymbol{C}$ for which we cannot guarantee that $h(x)$ is linear or affine. For example, if, for each $r \in I^c$ and for each j, $\alpha_{rj} = 0$, then we cannot guarantee linearity. However, these conditions are contrary to our assumption that $\boldsymbol{C}$ has fewer columns than rows, because such a $\boldsymbol{C}$ would have as many non-zero rows as it has non-zero columns. If $\alpha_{rj} = 0$ for all j and all $r \in I^c$ except for one $[r,j]$ pair, then $h(x)$ can be shown to be homogeneous but cannot, in general, be shown to be linear. With additional constraints, it may be possible to prove that a homogeneous equation is linear (Maxson and Meyer, 2001), but such work is not pursued here. Thus, our proof holds for cases in which, for at least one $r \in I^c$, there are two or more values of j for which $\alpha_{rj} \neq 0$. We have not ruled out the possibility that choosing

different rows of C to act as the basis for the rest of the rows, or using a different set of basis vectors for the column space of C will alleviate this problem. Perhaps future work will clarify this issue. Until then, we can only conclude that the proof holds under the conditions stated above. Nonetheless, the proof is very general and will hold for most values of $\boldsymbol{C}$ encountered in practical applications.

2.4. Solutions can be found even with small data sets

In general, we expect to know more about $f()$ than the fact that it is continuous at one point. In some cases, we can use that knowledge to invert $f()$ without requiring an infinite amount of data. We provide a concrete, albeit somewhat contrived, example to make the point that an infinite amount of data is not necessary for function inversion given sufficient prior information about $f()$. Let

$$\boldsymbol{C} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ 0 & 2 \end{bmatrix}. \tag{21}$$

In this case, $\alpha_{31} = 1$ and $\alpha_{32} = 1$, i.e., the third row of $\boldsymbol{C}$ is the sum of 1 times the first row and 1 times the second row. This fact implies that $x_3 = x_1 + x_2$ for all $\boldsymbol{v}$. Now suppose we know that $f(x) = x^b$ for a real but unknown b. We can try to find an inverse of the same form, e.g., $g(y) = y^c$. If we observe the data $\boldsymbol{y} = [1\ 8\ 27]$, then we know that we have found the $g()$ that inverts $f()$ when we find a real c for which $\hat{x}_3 = \hat{x}_1 + \hat{x}_2$, that is, $1^c + 8^c = 27^c$. There is only one solution to this equation, so we would conclude, from one $\boldsymbol{y}$, that b is 3, c is 1/3, and $\boldsymbol{x} = [1\ 2\ 3]$.

The T–d equations of (13) are helpful when the amount of data is finite. Assume, without loss of generality, that the d linearly independent rows of $\boldsymbol{C}$ are indexed by values from 1 to d, leaving the remaining T–d rows as linear combinations of the first rows. This approach allows us to rewrite (13) as:

$$\begin{bmatrix} g(y_{d+1}) \\ g(y_{d+2}) \\ \vdots \\ g(y_T) \end{bmatrix} = \begin{bmatrix} \alpha_{d+1,1} & \alpha_{d+1,2} & \cdots & \alpha_{d+1,d} \\ \alpha_{d+2,1} & \alpha_{d+2,2} & \cdots & \alpha_{d+2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{T,1} & \alpha_{T,2} & \cdots & \alpha_{T,d} \end{bmatrix} \begin{bmatrix} g(y_1) \\ g(y_2) \\ \vdots \\ g(y_d) \end{bmatrix} = \boldsymbol{A} \begin{bmatrix} g(y_1) \\ g(y_2) \\ \vdots \\ g(y_d) \end{bmatrix}. \tag{22}$$

Consideration of (22) reveals that we can sometimes find $g()$ for y values present in the data even when we cannot find $g()$ for every y. Another example clarifies this point. Let

$$\boldsymbol{C} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ 7 & -1 \\ 3 & 1 \\ -3 & 3 \end{bmatrix}, \quad \boldsymbol{v} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} 4 \\ 2 \\ 4 \\ 6 \\ 6 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} 1/4 \\ 1/2 \\ 1/4 \\ 1/6 \\ 1/6 \end{bmatrix}. \tag{23}$$

Typically, $\boldsymbol{x}$ and $\boldsymbol{v}$ need to be inferred. We would only expect to be given $\boldsymbol{C}$ and $\boldsymbol{y}$. We give $\boldsymbol{x}$ and $\boldsymbol{v}$ merely for comparison to the solution obtained using (22). Since $\boldsymbol{x}$ contains only values chosen from the set [2, 4, 6], we cannot expect to find $g()$ for every x value from such limited data.

The first two rows of $\boldsymbol{C}$ are linearly independent and, therefore, can be used to construct the third through fifth rows. As described above, $\boldsymbol{A}$ tells us how to construct the third through sixth rows of $\boldsymbol{C}$ from the first two rows of $\boldsymbol{C}$. We find the first row of $\boldsymbol{A}$ by noting that the third row of $\boldsymbol{C}$ is 3 times the first row of $\boldsymbol{C}$ minus 4 times the second row. Similarly, the second row of $\boldsymbol{A}$ tells us that the fourth row of $\boldsymbol{C}$ is 2 times the second row of $\boldsymbol{C}$ minus the first row of $\boldsymbol{C}$. Continuing this procedure we get:

$$\boldsymbol{A} = \begin{bmatrix} 3 & -4 \\ 2 & -1 \\ 0 & 3 \end{bmatrix}. \tag{24}$$

As described in the discussion of (14), we know that $h(0) = 0$ because the elements in the third row of $\boldsymbol{A}$ do not sum to one. Thus, we expect to be able to recover $\boldsymbol{x}$ to within a linear transformation. Writing our example in the form of (22), we get:

$$\begin{bmatrix} g(1/4) \\ g(1/6) \\ g(1/6) \end{bmatrix} = \begin{bmatrix} 3 & -4 \\ 2 & -1 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} g(1/4) \\ g(1/2) \end{bmatrix}. \tag{25}$$

The equation in the third row of (25) gives $g(1/6) = 3g(1/2)$, which can be plugged back in to (25) to get $g(1/4) = 2g(1/2)$. This gives

$$\begin{bmatrix} \hat{x}(1) \\ \hat{x}(2) \\ \hat{x}(3) \\ \hat{x}(4) \\ \hat{x}(5) \end{bmatrix} = \begin{bmatrix} g(1/4) \\ g(1/2) \\ g(1/4) \\ g(1/6) \\ g(1/6) \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 2 \\ 3 \\ 3 \end{bmatrix} g(1/2). \tag{26}$$

As expected, $\hat{\boldsymbol{x}}$ is a scalar multiple of $\boldsymbol{x}$. We could also solve the problem by setting the two equations for $g(1/6)$ equal to each other.

2.5. Some consequences of many-to-one mappings

The preceding example can be modified slightly to explore a many-to-one mapping. Let

$$\boldsymbol{C} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ 7 & -1 \\ 3 & 1 \\ 5 & 1 \end{bmatrix} \quad \boldsymbol{v} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \quad \boldsymbol{x} = \begin{bmatrix} 4 \\ 2 \\ 4 \\ 6 \\ 8 \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} 1/4 \\ 1/2 \\ 1/4 \\ 1/6 \\ 1/6 \end{bmatrix}. \tag{27}$$

The only differences between (23) and (27) are that the last row of $\boldsymbol{C}$ has been changed – and as a result, the last row of $\boldsymbol{x}$ changed. Note that the last row of $\boldsymbol{y}$ did not change. In

this example, both $x = 6$ and $x = 8$ map to $y = 1/6$, giving a many-to-one-mapping. The last row of $\boldsymbol{A}$ necessarily changes, so we now have:

$$\begin{bmatrix} g(1/4) \\ g(1/6) \\ g(1/6) \end{bmatrix} = \begin{bmatrix} 3 & -4 \\ 2 & -1 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} g(1/4) \\ g(1/2) \end{bmatrix}. \tag{28}$$

The discussion is simplified by changing (28) to:

$$\begin{bmatrix} \hat{x}(3) \\ \hat{x}(4) \\ \hat{x}(5) \end{bmatrix} = \begin{bmatrix} 3 & -4 \\ 2 & -1 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} \hat{x}(1) \\ \hat{x}(2) \end{bmatrix}. \tag{29}$$

If we set $\hat{x}(1) = \hat{x}(3)$, since they are both equal to $g(1/4)$, we conclude that $\hat{x}(1) = \hat{x}(3) = 2\hat{x}(2)$, that $\hat{x}(4) = 3\hat{x}(2)$, and that $\hat{x}(5) = 4\hat{x}(2)$. This solution correctly recovers $\boldsymbol{x}$ to within a linear transformation; however, ignoring the trivial solution $\hat{\boldsymbol{x}} = 0$, $\hat{x}(4) \neq \hat{x}(5)$ even though both are equal to $g(1/6)$. Alternately, set $\hat{x}(4) = \hat{x}(5)$, which leads to the conclusion that $\hat{x}(1) = \hat{x}(2) = \hat{x}(4) = \hat{x}(5) = -\hat{x}(3)$. In this case, $\hat{x}(1) \neq \hat{x}(3)$ despite the fact that both are equal to $g(1/4)$. Thus, if $g()$ is constrained to be one-to-one, (29) gives a set of equations whose only exact solution is $\hat{\boldsymbol{x}} = 0$.

Least squares and maximum likelihood techniques can be used to find a nontrivial (nonzero) solution. For example, we can find a nonzero least squares solution by minimizing the error function:

$$E = (\hat{x}(1) - \hat{x}(3))^2 + (\hat{x}(4) - \hat{x}(5))^2 \tag{30}$$

subject to the constraints of Eq. (29) and subject to the requirement that the solution be nonzero, e.g.,

$$\hat{x}(1)^2 + \hat{x}(2)^2 = 1. \tag{31}$$

The solution, in this case, is $\hat{x}(1) \cong 0.81$ and $\hat{x}(2) \cong 0.47$.

For real world data, many-to-one mappings are the rule, not the exception. Consider the mapping from articulation to acoustics. We cannot expect a speaker to use exactly the same articulator position twice. Furthermore, since there will always be some noise in the data, we cannot expect to see exactly the same acoustic signal twice. Without replicated acoustic values, we cannot solve for $\hat{\boldsymbol{x}}$. To get around this problem, we can quantize (or vector quantize) the acoustics to force the quantized acoustics to have repeated values, but in doing so, we create many-to-one mappings.

2.6. A generalization to multidimensional problems

The proof we have provided is very general, but stating it in its most general way is beyond the scope of this paper and risks obscuring the simplicity of the idea. Nonetheless, there is one particular generalization that is worth discussing here, since it is important for dimensionality reduction problems and problems involving recovering articulator positions from acoustics. Consider the case in which each element in $\boldsymbol{x}$ is a vector of real values instead of a single real value. A minimal departure from our previous problem is to let the constraint be:

$$\begin{bmatrix} x_1(1) & x_2(1) & \dots & x_n(1) \\ x_1(2) & x_2(2) & \dots & x_n(2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(T) & x_2(T) & \dots & x_n(T) \end{bmatrix} = \boldsymbol{C} \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{d1} & v_{d2} & \dots & v_{dn} \end{bmatrix} \tag{32}$$

where $x(t) = [x_1(t)\ x_2(t)\ \cdots\ x_n(t)]$ and

$$f(x(t)) : R^n \to R^m, \quad f(x(t)) = y(t) = [y_1(t)\ \ y_2(t)\ \ \cdots\ \ y_m(t)]. \tag{33}$$

For this statement to make sense, $g(y): R^m \to R^n$ and $h(x): R^n \to R^n$. As before, it can be proven that $h(x)$ is affine, which in this case means that

$$\hat{\boldsymbol{x}} = h(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{a} + \boldsymbol{b}. \tag{34}$$

The only difference between (34) and (2) is that $\boldsymbol{a}$, $\boldsymbol{x}$, and $\hat{\boldsymbol{x}}$ are now matrices, and $\boldsymbol{b}$ is a vector.

The proof for the multidimensional case is nearly the same as the proof given in Section 2.1. Since vector addition and multiplication follow nearly the same rules as scalars, no major changes to the proof are necessary.

2.7. An algorithm for finding the MIMICRI mapping

In this section we give a computational approach for finding g – which is the goal of MIMICRI. Many of the variables used in this section are defined in the comments immediately preceding Section 2.1. The solution we present assumes that the values of $y(t)$ are quantized and uses a least squares algorithm, as suggested in Section 2.5. In practical problems, we need to learn $g()$ from a finite number of observable sequences, and the sequences are not necessarily all the same length. To accommodate these considerations, we introduce some additional notation. Assume we have a collection of sequences of observable y values in which each sequence can have a different number of elements. The number of elements in sequence i is T_i. We write sequence i as:

$$\boldsymbol{y}_i = [y_i(1), y_i(2), \dots, y_i(T_i)]'. \tag{35}$$

Since $\boldsymbol{C}$ must have the same number of rows as $\boldsymbol{y}_i$, $\boldsymbol{C}$ will vary by the sequence length. We use $\boldsymbol{C}_i$ to denote the matrix corresponding to sequence i. We also generalize (5) to:

$$\boldsymbol{C}_i'\boldsymbol{C}_i = \boldsymbol{I}. \tag{36}$$

With the new notation, the goal of the algorithm is to find $g()$ such that:

$$\hat{\boldsymbol{x}}_i = \boldsymbol{C}_i\hat{\boldsymbol{v}}_i \quad \forall i \tag{37}$$

The elements of the y sequences are assumed to be quantized, and so they are restricted to a finite set of values, i.e.,

$$y_i(t) \in \{a_1 \quad a_2 \quad \cdots \quad a_Q\}. \tag{38}$$

Since the $y_k(t)$ values are selected from a finite set, the $\hat{x}_i(t)$ are also limited. Specifically:

$$\hat{x}_i(t) \in \{ g(a_1) \quad g(a_2) \quad \cdots \quad g(a_Q) \}. \tag{39}$$

We collect the possible values of $\hat{x}_i(t)$ in a vector:

$$\mu \equiv [g(a_1) \quad g(a_2) \quad \cdots \quad g(a_Q)]'. \tag{40}$$

This lets us write every possible $\hat{\boldsymbol{x}}_i$ as:

$$\hat{\boldsymbol{x}}_i = \boldsymbol{S}_i \mu \tag{41}$$

where the $\boldsymbol{S}_i$ selects the component of μ to insert into each position in $\hat{\boldsymbol{x}}_i$. That is, create a matrix,

$$\boldsymbol{S}_i = \begin{bmatrix} s_{i11} & s_{i12} & \cdots & s_{i1Q} \\ s_{i21} & s_{i22} & \cdots & s_{i2Q} \\ \vdots & \vdots & \ddots & \vdots \\ s_{iT1} & s_{iT2} & \cdots & s_{iTQ} \end{bmatrix}, \tag{42}$$

in which

$$s_{ijk} = \begin{cases} 1 & \text{if } y_i(j) = a_k \\ 0 & \text{otherwise} \end{cases}. \tag{43}$$

With the definitions given above, we can set about finding μ (which, for our purposes, gives us g) to meet our constraint as closely as possible. If the constraints shown in (36) and (37) were met exactly, then:

$$\boldsymbol{C}_i' \hat{\boldsymbol{x}}_i = \boldsymbol{C}_i' \boldsymbol{C}_i \boldsymbol{v}_i = \boldsymbol{v}_i, \tag{44}$$

$$\boldsymbol{C}_i \boldsymbol{C}_i' \hat{\boldsymbol{x}}_i = \boldsymbol{C}_i \boldsymbol{v}_i = \hat{\boldsymbol{x}}_i, \tag{45}$$

and

$$\boldsymbol{C}_i \boldsymbol{C}_i' \boldsymbol{S}_i \mu = \boldsymbol{S}_i \mu. \tag{46}$$

Eq. (46) suggests a least squares problem in which we minimize the squared length (indicated by the vertical bars) of the difference between the right and left side of (46), i.e.:

$$E = \sum_i | \boldsymbol{C}_i \boldsymbol{C}_i' \boldsymbol{S}_i \mu - \boldsymbol{S}_i \mu |^2. \tag{47}$$

For generality, we give the solution for the multidimensional problem, in which $\hat{x}_i(t)$ and the elements of μ are row vectors. To ensure that the dimensionality of the solution is that same as the number of columns in μ, we restrict the possible solutions to those in which the columns of μ are orthogonal and all have length one. In multiple dimensions, (47) can be written:

$$\begin{aligned} E &= \text{trace}\left[\sum (\boldsymbol{C}_i \boldsymbol{C}_i' \boldsymbol{S}_i \mu - \boldsymbol{S}_i \mu)' (\boldsymbol{C}_i \boldsymbol{C}_i' \boldsymbol{S}_i \mu - \boldsymbol{S}_i \mu) \right] \\ &= \text{trace}\left[\sum \mu' (\boldsymbol{S}_i' \boldsymbol{S}_i - \boldsymbol{S}_i' \boldsymbol{C}_i \boldsymbol{C}_i' \boldsymbol{S}_i) \mu \right]. \end{aligned} \tag{48}$$

It has been reported that the solution to quadratic problems of this form is well known (Edelman et al., 1998) and that the columns of μ are the eigenvectors of $\sum_i (\boldsymbol{S}_i' \boldsymbol{S}_i - \boldsymbol{S}_i' \boldsymbol{C}_i \boldsymbol{C}_i' \boldsymbol{S})$ associated with the smallest eigenvalues. We note that

$$E = \sum_j \mu_j' \left[\sum_i (\boldsymbol{S}_i' \boldsymbol{S}_i - \boldsymbol{S}_i' \boldsymbol{C}_i \boldsymbol{C}_i' \boldsymbol{S}) \right] \mu_j \tag{49}$$

where μ_j is a column of μ, and that, $\mu_j' \mu_j = 1$. This fact implies that each term in (49) is a Rayleigh quotient (Strang, 1980) that is minimized by setting each μ_j to an eigenvector of $\sum_i (\boldsymbol{S}_i' \boldsymbol{S}_i - \boldsymbol{S}_i' \boldsymbol{C}_i \boldsymbol{C}_i' \boldsymbol{S})$ with the smallest eigenvalue possible. The eigenvectors associated with the smallest eigenvalues meet our requirements of being mutually orthogonal, having length 1, and minimizing the error.

The matrices $\sum \boldsymbol{S}_i' \boldsymbol{S}_i$ and $\sum \boldsymbol{S}_i' \boldsymbol{C}_i$ can be found without explicit multiplication:

$$\sum \boldsymbol{S}_i' \boldsymbol{S}_i = \begin{bmatrix} n_1 & 0 & 0 \\ 0 & n_2 & 0 \\ 0 & 0 & \ddots \end{bmatrix} \tag{50}$$

where n_i is the number of occurrences of a_i.

To find $\boldsymbol{C}'\boldsymbol{S}$, create a matrix, $\boldsymbol{M}$, the same size as $\boldsymbol{C}'\boldsymbol{S}$, but set every element to 0. Starting with $i = 1$, let j index the column in row i of $\boldsymbol{S}$ that has a value of 1. (Recall that there is only one nonzero value in every row of $\boldsymbol{S}$.) Add column i of $\boldsymbol{C}'$ to column j of $\boldsymbol{M}$. After repeating this process for all i, $\boldsymbol{M} = \boldsymbol{C}'\boldsymbol{S}$.

When the column space of $\boldsymbol{C}$ contains the vector with all elements equal – such as when we allow the unobservable signals to have energy at 0 Hz – then there is a degenerate solution that will generally show up as the eigenvector associated with the smallest eigenvalue. The solution is to map every a_i to the same value. In this case, the error will be 0 for that solution (which is why it will generally be the eigenvector associated with the smallest eigenvalue). The solution is degenerate in that all it does is add an arbitrary 0 Hz component to the signal, which has no information content. In such cases, the smallest eigenvector needs to be discarded, and the eigenvectors associated with the next smallest eigenvalues should be used instead.

3. Learning the mapping from acoustics to articulator positions

The previous sections have argued that the ability to map blindly from speech acoustics to articulator positions may create a representation of speech that is more robust to microphone and speaker differences, requires fewer dimensions to capture the same amount of information, and aids in using context to infer missing data. In this section we bolster the idea that such a transformation can be found by using MIMICRI on a set of human data. This study builds on previous work showing that the mapping from acoustics to articulator positions can be learned (for some data sets) using a simple supervised learning algorithm (Hogden et al., 1996b). We extend the previous work by showing that MIMICRI, an unsupervised algorithm, can also learn the mapping, albeit only to within an affine transformation.

The final result of this work is a comparison of the accuracy of MIMICRI articulator position estimates to the accuracy of positions estimated using a supervised learning algorithm. To the extent that MIMICRI estimates articulator positions as accurately as the supervised approach, we conclude that MIMICRI is doing well. It is important, therefore, not to give MIMICRI any advantage that the supervised approach does not have. To ensure that our conclusions are conservative, we give a very slight advantage to the supervised algorithm by estimating some parameters to optimize the supervised algorithm, not MIMICRI. Even ignoring the possible advantage of the supervised approach, our conclusion is that MIMICRI performs nearly as well as supervised learning.

Recall that our claim is only that MIMICRI can recover articulator positions to within an affine transformation. For applications such as speaker-dependent speech recognition, it would not be necessary to transform the MIMICRI estimates to articulator positions, just as it is not necessary to transform Mel-cepstra to articulator positions before using them. Nonetheless, to test our claim, we find the affine transformation that makes the MIMICRI-derived features best match articulator positions, and then calculate correlation coefficients. These testing steps are accomplished using multiple regression, which gives us a way to estimate the error of the correlations. While multiple regression is a type of supervised learning, it is extremely uncommon for researchers to calculate error bars around multiple correlation coefficients by dividing the data into multiple training and testing sets. We follow the standard statistical approach of testing for a linear relationship instead of dividing the data into training and testing sets.

3.1. Data

The data consist of simultaneously collected acoustic and articulator measurements of utterances produced by a Swedish speaker. Each utterance contains a vowel-to-vowel transition spoken in a /g/ context, e.g., /guog/. The vowels in the utterances are {/i/, /e/, /æ/, /a/, /o/, /u/, /ɛ/} and the three Swedish, front-rounded vowels {/ʉ/, /y/, /ø/}. The speaker produced each transition from one vowel to a different vowel resulting in 90 utterances in a first set. The process was repeated to create three sets of 90 utterances each. The sound pressure versus time waveforms for the utterances were used to find the beginning and end-points of the utterances. Low-amplitude portions of the signals at the beginning and end of the utterances were included, but we avoided using acoustic samples during the nearly silent portion of the /g/ production in which the vocal tract is closed (also called the /g/ closure). So the utterances included /g/ releases, vowels, and transitions to /g/ closures, but very little information from /g/ closures.

Articulator positions were measured using the Haskins Laboratories EMMA system (Perkell et al., 1992; Whalen et al., 1990). The x and y coordinates of each coil were measured 625 times per second with 12 bit accuracy. One coil was placed on the upper incisors and one on the bridge of the nose to be used as a fixed reference frame. Four EMMA receiver coils were placed on the tongue. We refer to these receiver coils (from back of the tongue to front) as the tongue rear (TR), tongue body (TB), tongue dorsum (TD), and tongue tip (TT) coils. One receiver coil, placed on the lower incisors to capture jaw movement, is called the JA coil. Coils were also placed on the upper lip (UL) and lower lip (LL). The coil positions measured at each time were rotated and translated as needed to ensure that the reference coils were in the same positions in each frame and to force the x-axis to lie on the occlusal plane. Receiver coils placed on the lips and jaw have an estimated accuracy of ± 0.5 mm, and coils placed on the tongue have an estimated accuracy of ± 1.0 mm.

Fig. 3 shows the power (the variance) in the articulator trajectories after low-pass filtering using various cut-off frequencies. For example, the points on the curves at 5 Hz show the variance in the EMMA coil positions after removing frequency components above 5 Hz. The difference between the total power in the signal (i.e. the variance before filtering) and the power in the signal low-passed to 20 Hz would not be discernable on the plot. As can be seen, the articulator trajectories have little power above 5 Hz. We used an approach here that is unlike our previous work. In our previous work, the articulator positions were low-pass filtered to 15 Hz before further processing. In this

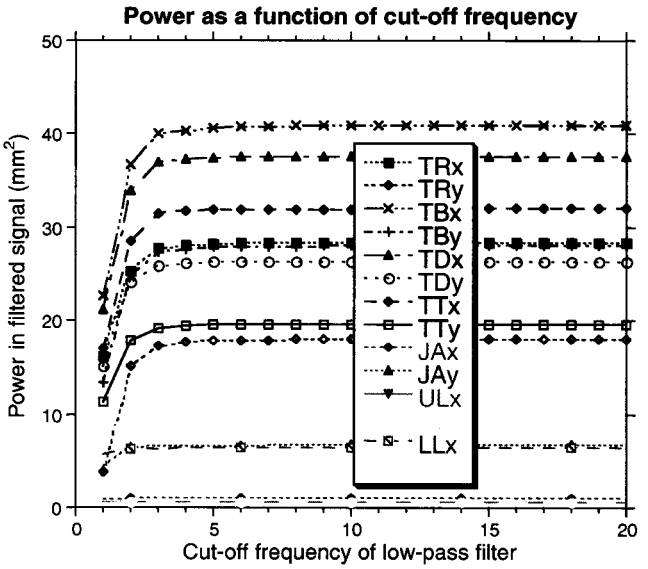

Fig. 3. Each curve shows the variance in the position of an articulator (the power of the articulator) after it has been low-pass filtered at different cut-off frequencies. When the signals are low-pass filtered at lower cut-off frequencies, they have less energy because the energy above the cut-off has been removed. The maximum amount of energy is at 312.5 Hz, but would not be noticeably higher than the amount of energy at 20 Hz in this plot. Very little extra energy is seen as we increase the cut-off frequency above 5 Hz, an indication that most of the energy in the articulator trajectories is below 5 Hz.

study, the articulator trajectories are not filtered before being compared to the estimated articulator positions.

Smooth spectra were calculated from the speech acoustics exactly as in Hogden et al. (1996b). We give a brief description of the signal processing used to calculate smoothed spectra here, and refer the reader to the Hogden et al. paper for more details. To calculate the smoothed spectra, the speech acoustics were filtered to remove frequencies above 10 kHz, pre-emphasized using a fixed filter, and then sampled at 20 kHz with 12 bits/sample precision. Hamming windows (25.6 ms) were positioned over the speech, centered at each time an articulator position sample was taken. The speech windows were used to calculate 32 cepstrum coefficients, which were then used to calculate a smoothed spectrum approximating the transfer function of the vocal tract. The transfer function was only estimated for frequencies below 5 kHz. The smoothed spectra were normalized by setting the total energy of each slice to 1.

Note that by using cepstrum analysis, we are attempting to remove the glottal source characteristics from the spectra. We do this because vocal-fold motions have high-frequency components, and so do not meet the smoothness constraint assumed by MIMICRI.

The resulting spectral slices were vector quantized so that each spectral slice was represented by a vector quantization (VQ) codeword. Results obtained using various numbers of VQ codewords are described below. A standard LBG algorithm was used in this work (Gray, 1984) – an approach unlike our earlier work, which used frequency-sensitive vector quantization (Ahalt et al., 1990).

3.2. Time alignment

Even though MIMICRI attempts to learn the mapping between acoustics and articulation using only acoustic data, we need to be able to compare the MIMICRI articulator estimates to the measured articulator positions in order to evaluate how well the mapping is learned. As noted in the previous work (Hogden et al., 1996b), the acoustics and articulator measurements may not be perfectly time aligned, or it may be that acoustics at time $t + d$ best predict articulator positions at time t. Thus we need to know which delay, d, gives the best time alignment. As discussed below, we find the time delay by using a supervised algorithm to learn the mapping between acoustics and articulation at varying time delays. We then choose the delay that gives reconstructed articulator trajectories that best match the measured trajectories.

The supervised learning algorithm simply creates a look-up table mapping each acoustic VQ codeword to an articulator configuration. As such, it is capable of learning extremely complex nonlinear mappings. Similar look-up tables have been used in the previously cited work on inverting the articulation-to-acoustics mapping. The process used to create the look-up table is the same as that in (Hogden et al., 1996b) – for each VQ codeword, find the average of the articulator configurations that occur at the same time (with the time delay) as the VQ codeword. Articulator trajectories are reconstructed from quantized acoustics by replacing each VQ codeword in a quantized acoustic signal with the mean articulator configuration given in the codebook and then low-pass filtering the results. We report results obtained using 1024 VQ codewords, since these results will be compared to MIMICRI in Section 3.4.

The results of the supervised learning are not intended to suggest how well the supervised learning algorithm would generalize to other data sets. For the supervised algorithm, using more VQ codewords will give a more accurate articulator trajectory estimation for data in the training set. In the extreme, if we use one VQ codeword for every acoustic signal, we will get nearly 100% correlation between estimated and actual articulator trajectories (the only differences being the result of filtering the estimated trajectories). We would not expect the nearly perfect results to generalize to a test set. We restrict our use of the supervised results to determining the relative time offset and to showing the best results we could expect from MIMICRI (as discussed in Section 3.4). Estimating the generalization performance of the supervised learning algorithm is not pursued here. However, since we are choosing the time-delay parameter to optimize results for the supervised learning, not MIMICRI, we may be overestimating the advantage that the supervised approach has over MIMICRI. Any advantage gained by the supervised approach will make MIMICRI appear to perform worse by comparison and will only give us a more conservative estimate of how well MIMICRI works.

Correlations between measured articulator trajectories and reconstructed trajectories for various delays and cut-off frequencies are shown in Fig. 4. We use delays that are multiples of the sampling period, 1.6 ms. The correlations are calculated over all articulator measurements. For example, given a sequence of measured tongue body x (TBx) positions, $x(t)$, a sequence of estimated TBx positions, $\hat{x}(t)$, a sequence of measured TBy positions of the tongue body, $y(t)$, and the estimated TBy positions, $\hat{y}(t)$, and the means of the sequences, μ_x, $\mu_{\hat{x}}$, μ_y, and $\mu_{\hat{y}}$, respectively, we calculate the overall correlation, r, using the equation:

$$r = \frac{\sum_t [(x(t)-\mu_x)(\hat{x}(t+d)-\mu_{\hat{x}}) + (y(t)-\mu_y)(\hat{y}(t+d)-\mu_{\hat{y}})]}{\sqrt{\sum_t [((x(t)-\mu_x)^2 + (y(t)-\mu_y)^2)((\hat{x}(t+d)-\mu_{\hat{x}})^2 + (\hat{y}(t+d)-\mu_{\hat{y}})^2)]}}. \quad (51)$$

Eq. (51) is only one of many possible ways to combine the correlations calculated over all articulators into one number. We do not know of any objective criterion for choosing one method over another. Nonetheless, (51) seems to be a reasonable equation for the limited uses of time-alignment and confirming previous expectations about the cut-off frequency.

From Fig. 4, we see that a delay of 15 sampling periods (24 ms, which is also approximately the duration of an

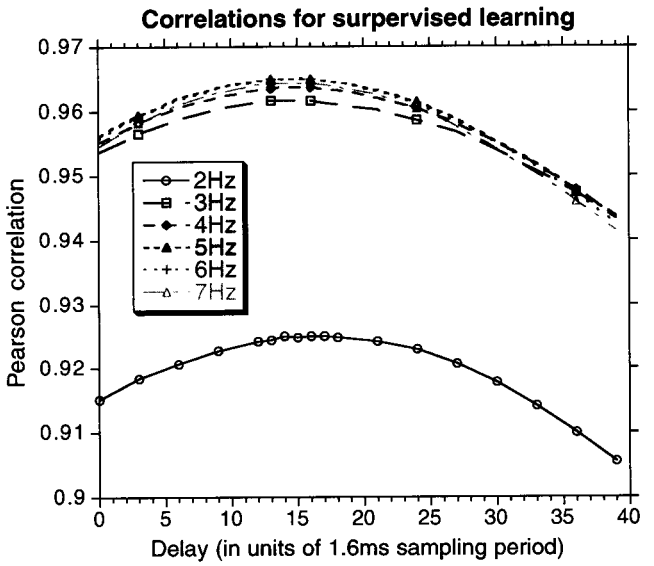

Fig. 4. This figure shows the overall correlation between the actual articulator positions and the articulator positions inferred from acoustics changes as the acoustics are shifted in time with respect to the articulator measurement. The peak at a time delay of 15 sample periods (24 ms) suggests that the optimal time-alignment was reached.

acoustic window) gives the best time-alignment. The results also confirm that a 5 Hz or 6 Hz cut-off frequency is optimal, a finding that agrees with the results shown in the Fig. 3. The remainder of the study will assume a 24 ms delay and a 5 Hz cut-off frequency.

3.3. Varying the number of VQ codewords and the dimensionality

Both from the perspective of signal quantization and MIMICRI mapping, using more VQ codewords will, typically, improve our ability to recover/reconstruct articulator trajectories. There are exceptions, however. Our vector quantization routine will not find the same VQ codebook when run with different initial settings. Some variability in articulator recovery can be expected to result from differences in the vector quantization, although those differences should decrease as the number of VQ codewords increases. Also, the accuracy of MIMICRI estimation will decrease beyond some number of VQ codewords, even on the training data. If we do not have enough data to provide many examples of each VQ codeword, we cannot expect to learn an inverse map accurately. Thus, determining the number of VQ codewords is largely a matter of using as many as we can without getting a decrease in articulator position estimation – as in simulation work reported previously.

A practical consideration when choosing the number of VQ codewords is the time it takes to solve a problem. Finding the MIMICRI solution involves finding eigenvectors of a square matrix with the number of rows equal to the number of VQ codewords. Since the computational difficulty of calculating eigenvectors, the time it takes to learn a VQ codebook, the time it takes to quantize an acoustic signal, and the amount of data required for training all increase with the number of VQ codewords, it is worthwhile to consider ways to decrease the number of VQ codewords with minimum impact to accuracy.

For a fixed signal reconstruction quality with uniform bit quantization, the number of quantization bits decreases with increases in sampling frequency. Quadrupling the sampling frequency is roughly equivalent to doubling the number of quantization levels (Lyons, 2004). VQ is not uniform, but it seems likely that oversampling will also improve the accuracy for VQ. Since the articulator trajectories have little energy above 5 Hz, a sampling frequency of 10–12 Hz should adequately sample them. We calculate 625 VQ codewords from acoustics every second (one VQ codeword for each articulator position sample). Since MIMICRI estimates articulator positions for each VQ codeword, this approach corresponds to nonuniformly sampling the articulation at 625 Hz – much higher than the required 10 Hz. Hogden et al. (1996b) showed that this sampling rate allows relatively few VQ codewords (about 256) to result in good estimates of articulator trajectories from acoustics for a supervised algorithm. More VQ codewords/second could easily be obtained from the speech acoustics if necessary, and further work exploring the relationship between the number of VQ codewords and the sampling rate would be worthwhile.

Using the same VQ codebooks found for the supervised learning, we quantized the acoustic data and then used MIMICRI to find mappings from acoustics to paths having all their energy below 5 Hz. Note that creating the VQ codebook, like MIMICRI processing, does not require articulator measurements. Given the theory, the smooth paths should be related to articulator trajectories by an affine transform. Fig. 5 shows the accuracy of articulator trajectories recovered using MIMICRI as we change the number of VQ codewords and the dimensionality of the solution. Since the articulator positions estimated by MIMICRI can only be expected to be within an affine transformation of the measured articulator trajectories, and since the MIMICRI solutions have fewer than 14 dimensions (14 is the number of articulator position measurements made at each time), we must find the affine transformation before making the comparison.

We use multiple regression (Dillon and Goldstein, 1984) to find the affine mapping that maximizes the correlation by treating the MIMICRI estimates as the independent variables and the measured positions as the dependent variables. The affine mapping was calculated over all trajectories in the data set and then applied to the estimated articulator positions. Eq. (51) was then used to calculate an overall correlation.

Note that adding a dimension to the MIMICRI solution is equivalent to adding an independent variable to a multiple regression problem. Since adding an independent variable never decreases the accuracy of multiple regression estimation and typically increases it, adding dimensions

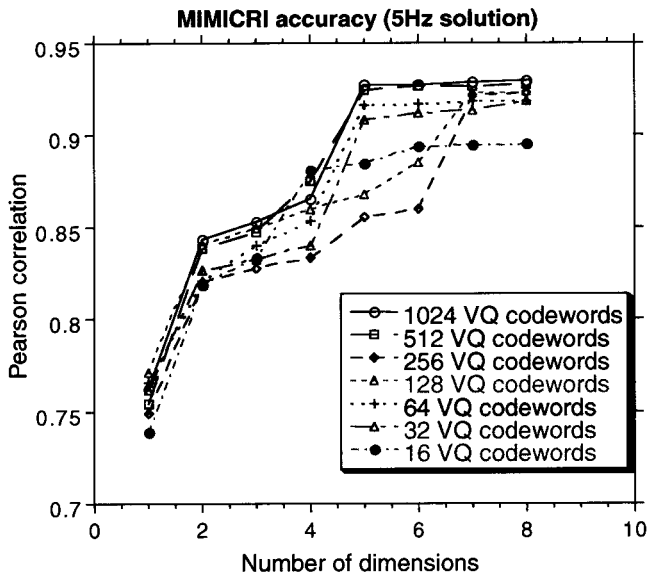

Fig. 5. Each curve shows how the overall correlation between measured and MIMICRI estimated articulator positions increases with dimensionality for a given number of VQ codewords. The estimation accuracy increases with the dimensionality of the solution but does not always increase as we increase the number of VQ codewords. The accuracy plateaus at the 5-dimensional, 1024 VQ codeword solution. This solution is used for further analysis.

to the MIMICRI solution will generally increase the accuracy, even when the dimension is random noise. The increase in correlation with increases in independent variables is typically factored out by multiplying correlations by a shrinking factor (Dillon and Goldstein, 1984). As the number of samples increases, the shrinking factor approaches 1. In our problem, the number of samples is large (sampling at 10 Hz would give approximately 2380 independent samples) so applying the shrinking factor has little effect on the plotted data. As a check, we also did pilot studies in which multiple regression was used to relate the articulator positions estimated by the supervised learning algorithm to the measured positions. Doing multiple regression on the supervised algorithm eliminates any advantage MIMICRI has over supervised learning. The changes in the correlations for the supervised algorithm were negligible. Thus, we report raw correlations.

Fig. 5 shows that the estimation accuracy does not improve monotonically as the number of VQ codewords increases. In fact, the results for 128 codewords and 256 codewords are worse for some dimensionalities than results using 16 codewords. All of the curves have leveled out by the time we get a 7-dimensional solution, perhaps suggesting that a 7-D solution is appropriate for this problem. For solutions with 5 dimensions or more, the 1024 codeword solution is the best and is only slightly better than the solution with 512 codewords. The 1024 codeword solutions show some – albeit very little – improvement as we go to more than 5 dimensions. As long as we use 5 or more dimensions, it appears that the 1024 codeword solution is near optimal.

3.3.1. Dimensionality considerations

It is not clear how many dimensions are needed to represent the articulator position data. Our measurements do not capture important aspects of articulation (e.g., pharyngeal lowering or nasalization), so one could imagine that more than 14 dimensions are needed to represent articulator positions accurately. However, Fig. 5 shows that the 5-dimensional solution estimates the measured articulator trajectories nearly as well as the 8-dimensional solution for 1024 VQ codewords, suggesting that the articulator positions we measured probably lie close to a 5-dimensional manifold.

Basing our selection of dimensionality on Fig. 5 alone may be a mistake. The correlation shown in Fig. 5 is taken over all the articulator trajectories and could hide improvements in some trajectories. Fig. 6 shows the estimation accuracy for each measured articulator as we increase the number of dimensions in the solution. Fig. 6 also shows that the estimation accuracy nearly stops increasing beyond 5 dimensions for each of the individual articulator measurements. The extent to which the curves level-off beyond 5 dimensions is fairly dramatic, and we do not have an adequate explanation of the phenomena, but it agrees with the results of the overall correlation in suggesting that the data lie on a 5-dimensional manifold. Since we do not know whether other, unmeasured articulators are being estimated more accurately beyond 5 dimensions, it is not possible to determine from this data alone how many dimensions may be required for applications such as speech recognition, but it does seem that the measured articulators can be described fairly well by the 5-D solution.

Fig. 6a–d suggests the possible meaning of the dimensions. For example, in Fig. 6c, we see that the estimation accuracies for the lower lip and jaw all increase dramatically in going from the 1-dimensional solution to the 2-dimensional solution, and that all of them are nearly constant after that. The similarity of the patterns suggests that these measurements are all related to one underlying factor – a factor that is estimated well for the 2-dimensional solution. Similarly, the x-positions of coils placed on the tongue show a similar pattern of improvement as the dimensionality increases, and the measurements of the upper lip seem to be related to a single factor. The y-positions of the coils placed on the tongue show similar patterns for the TB, TD, and TT coils, but the TR coil pattern is different, suggesting that the height of the rear of the tongue is controlled by a different underlying factor than the height of the front of the tongue. This interpretation of the meaning of the underlying factors is certainly debatable (e.g., ULy seems to get an larger boost in accuracy at 5 dimensions than ULx), but corresponds fairly well with our understanding of speech production.

Figs. 7 and 8 give scatterplots of the estimated versus actual positions of the articulators. To reduce the number of plotted points to a manageable size, the measured and estimated trajectories were subsampled to approximately 10.4 samples per second. The scatterplots show that the

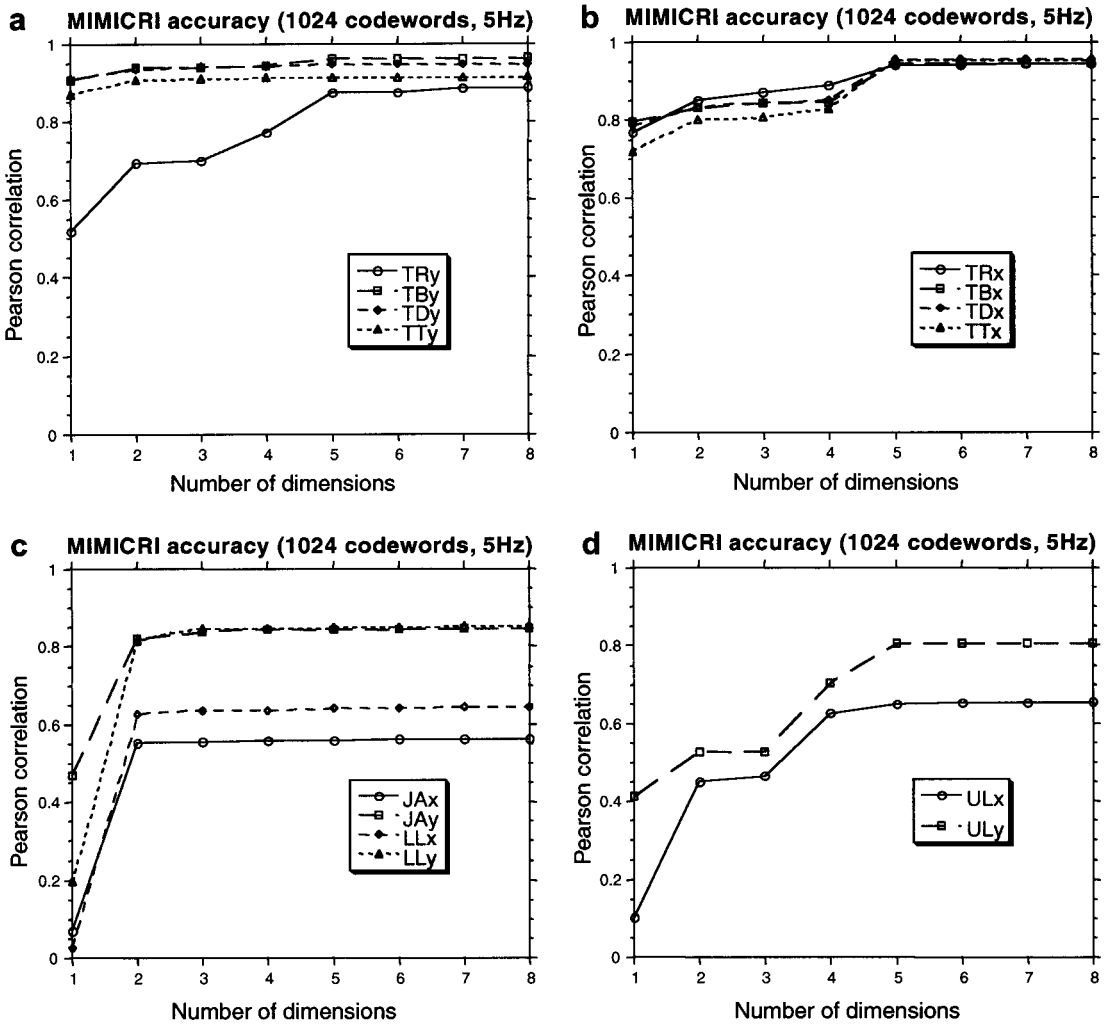

Fig. 6. Each curve shows the correlation between the estimated and actual positions for one articulator. Articulators with similar patterns of improvement are grouped together, except that the estimation accuracy of y-positions of EMMA receiver coils placed on the tongue seems to show different patterns of improvement between the tongue rear and the tongue tip. The fact that we observe roughly 5 patterns of improvement combined with fact that the accuracy plateaus at or before 5 dimensions for all the articulators further recommends the 5-D solution.

measured positions of coils on the tongue have a range of approximately 20–25 mm. The lower lip and y-positions of the jaw move over a range of about 15–20 mm. In contrast, the upper lip shows little motion (a range of approximately 5 mm). The x-position of coil on the lower lip also shows little variability (a range of approximately 7 mm). In agreement with the MIMICRI theory, the scatterplots show clear linear trends for tongue motions. However, the trends are not as clear for the lip motions and are worst for the x-position of the jaw.

3.4. Supervised learning vs. MIMICRI

Both MIMICRI and the supervised learning technique discussed in Section 3.2 estimate the mean articulator positions associated with each VQ codeword. They also both find the estimated articulator trajectories using the same technique – they replace the VQ codewords with the estimated articulator means and then smooth the trajectories with a low-pass filter. The only difference is how they estimate the means: the supervised algorithm uses the time-aligned acoustics and articulator trajectories to calculate the means, whereas MIMICRI estimates the means using only the VQ codeword sequences. Since the supervised algorithm has access to the articulator data, the means it calculates are the actual sample means for the data. MIMICRI only approximates the sample means.

Since the trajectory estimation accuracy will increase with increasing accuracy in the mean estimates, the supervised results give an appropriate upper performance bound on MIMICRI, making a comparison of the results interesting. Fig. 9 shows the comparison of the supervised and MIMICRI learning algorithms broken out by articulator for the 5-D, 1024 codeword solution with a 5 Hz cut-off frequency. For coils placed on the tongue, MIMICRI correlations are only a few percentage points lower than the upper

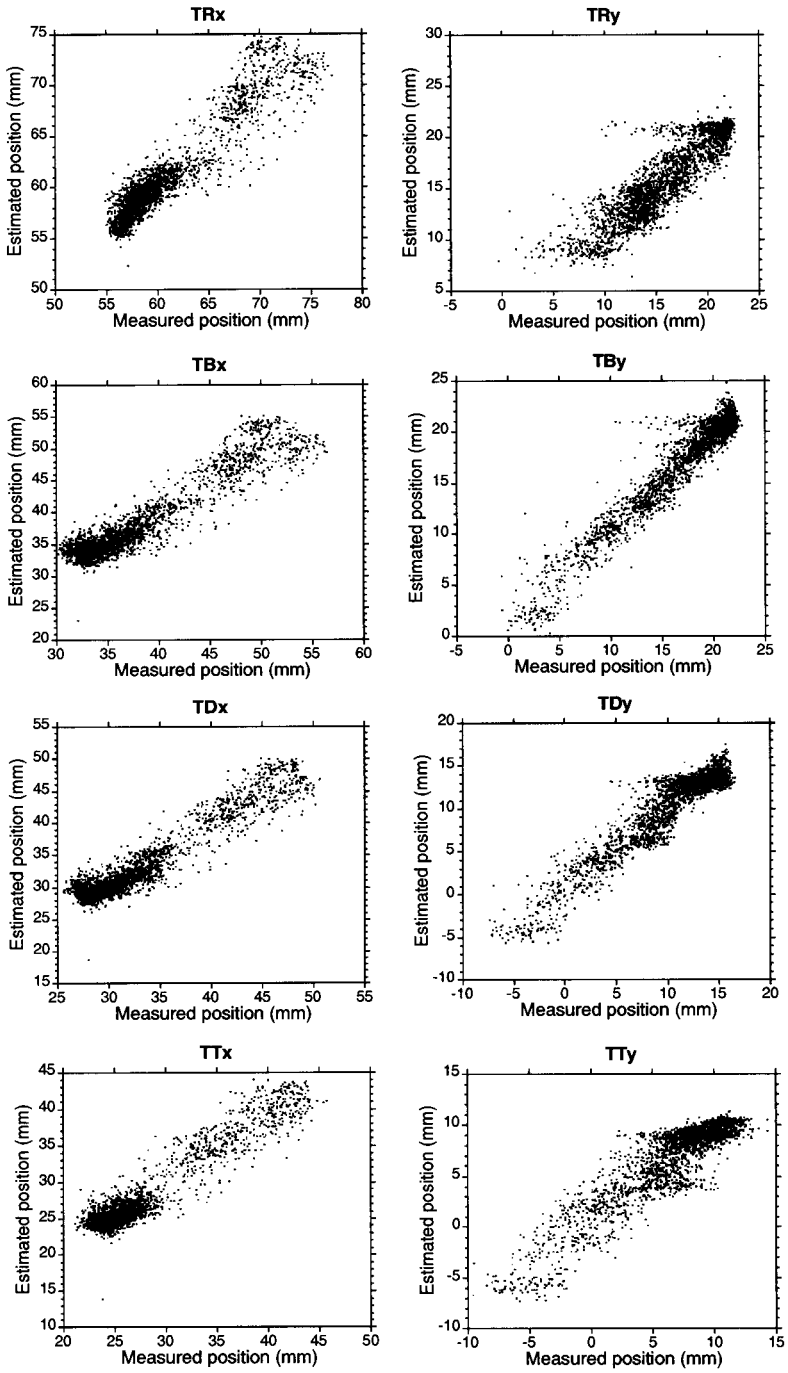

Fig. 7. These scatterplots show the relationship between estimated and actual positions of each EMMA receiver coil placed on the tongue. The linear trend predicted by the theory is obvious in these plots. EMMA receiver coils placed on the tongue are reported to be accurate to within about 1 mm.

limit given by the supervised learning algorithm. The exception is that the y-position of the tongue rear coil correlation is about 8 percentage points lower. The correlations of the y-positions of coils on the lips and the jaw are also within about 8 percentage points of the supervised learning, but the x-positions of the lips and jaw are not estimated as well. If the information about the x-positions was not in the acoustics, then the supervised algorithm could not have learned it, so it is clear that MIMICRI is missing information. However, it may also be the case that MIMICRI is capturing information about articulators that were not measured in this data. In any case, since it is difficult to collect articulator measurements, it may be that MIMICRI's ease of use more than compensates for the loss in accuracy that we experience because of using MIMICRI instead of a supervised learning algorithm.

MIMICRI only recovers articulator positions from acoustics to within an affine transform – an accomplishment

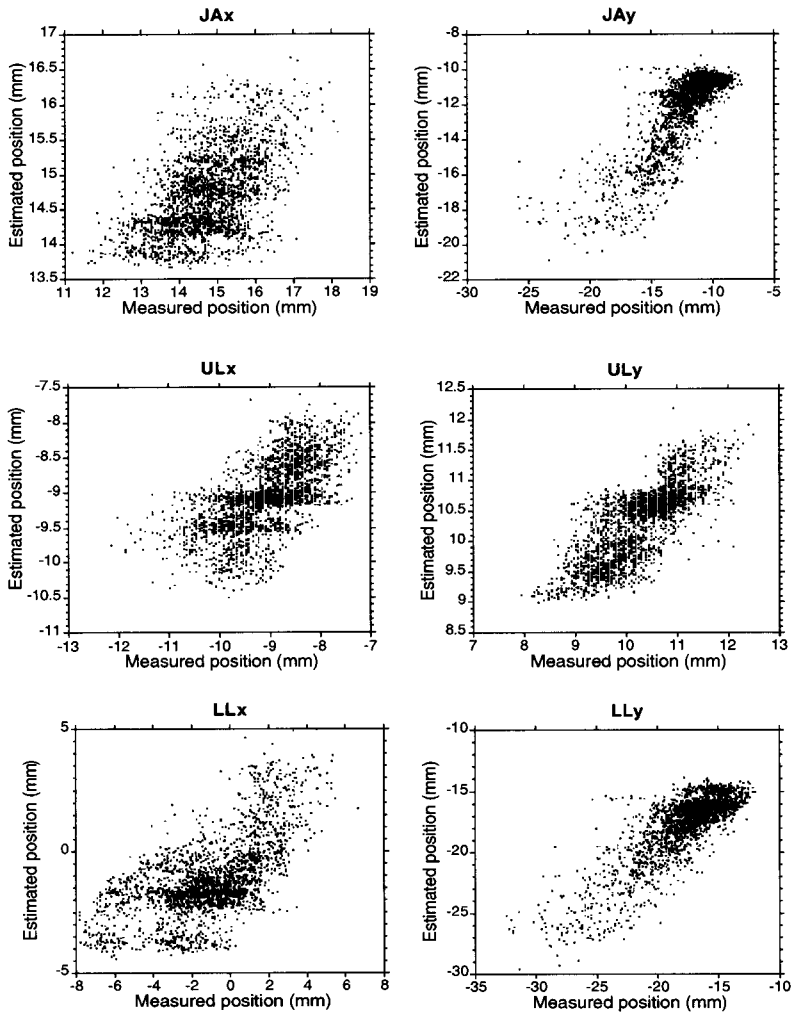

Fig. 8. These scatterplots show the relationship between estimated and actual positions of EMMA receiver coils on the jaw and lips. The articulator positions have more noise and a much smaller range of variation in the measured positions than the plots shown in Fig. 7. EMMA receiver coils placed on the jaw and lips are reported to be accurate to within about 0.5 mm.

that is less impressive if the acoustic data are themselves within an affine transform of the articulator data (Yehia et al., 1998). To evaluate the importance of nonlinearities in this task, we ran multiple regression analysis to find the best affine transformation from the smoothed spectra (128 energies at each time sample) to articulator positions. The results indicate that, for vowel data, the problem of recovering articulator positions from spectra is not as nonlinear as one might expect, but that MIMICRI does better at capturing articulator positions with 5 dimensions than the spectrum does with 128 dimensions.

The curve labeled "Spectrum" in Fig. 9 shows correlations between measured articulator positions and positions estimated using the best affine transform of the spectra. For x-positions of coils placed on the tongue, the difference between the MIMICIRI correlation and the spectrum correlation is roughly 5–7%. For the y-positions of points on the tongue the correlation difference varies more – MIMICRI is better by about 3% for the tongue rear but the difference gets smaller moving forward on the tongue until the spectrum and MIMICRI correlations are about the same. For the lip y-positions, MIMICRI outperforms the spectrum by about 5% or 6%, with the exception being the upper lip x, for which the spectrum is only about 3% (or less) worse. For the jaw y, MIMICRI is about 3% better. For the x-position of the jaw, the spectrum and MIMICRI perform about equally, but neither do well.

4. Summary and conclusions

This paper describes a signal processing technique called MIMICRI that may prove helpful in human and automatic speech recognition by providing psycholinguistic theoreticians with a practical approach to recovering articulator positions from acoustics. In describing this technique, we have presented a detailed mathematical proof that bandlimited signals (and other suitably constrained signals) have a remarkable property: If we know the bandlimit (or constraint), we can use a process called MIMICRI to recover the signal, to within an affine transformation, from

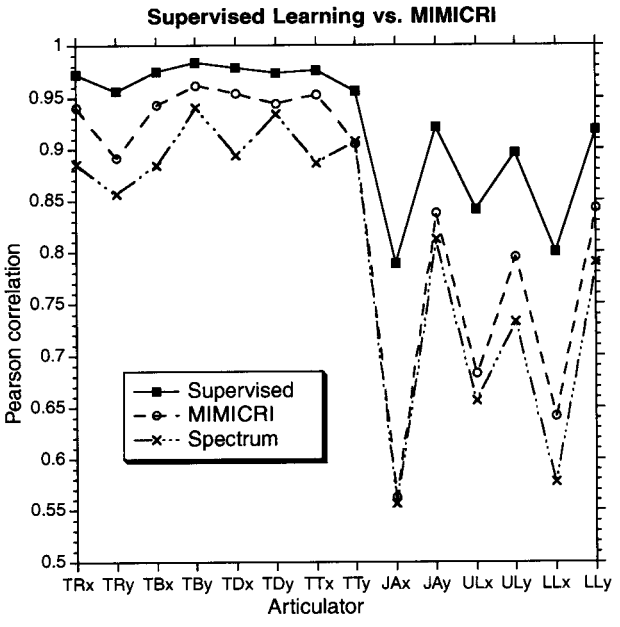

Fig. 9. The solid line gives estimation accuracy for a simple supervised learning algorithm. The dashed line gives the estimation accuracy for MIMICRI. Note that the value on the y-axis is 0.5, highlighting differences between the supervised algorithm and MIMICRI. For positions on the tongue, MIMICRI correlations are a few percentage points worse than the supervised algorithm, but MIMICRI performance is worse for movements along the x-axis for the jaw and lips.

a nonlinearly transformed version of the signal without knowing the nonlinear transformation and without having time-aligned training data for use in learning the transformation. Based on the proof, we have provided a computational solution to the problem of inverting unknown nonlinear transformations to within an affine transformation. The mathematical work greatly increases our understanding of MIMICRI. It has improved our understanding of how the proof applies to finite data sets and how many-to-one mappings figure into the framework.

We have also discussed why MIMICRI's ability to invert nonlinear mappings makes it useful in cases in which we need to reduce the dimensionality of a signal or use context to invert many-to-one mappings. We have argued that, because of these abilities, MIMICRI should be studied for its potential to solve the speech inversion problem of recovering articulator positions from acoustics. Articulator trajectories are approximately bandlimited and so should be recoverable from speech signals even if the mapping from articulator positions to acoustics is not known, and, possibly, even if the mapping is many-to-one. An exception is that vocal-fold vibrations have energy at frequencies higher than other articulator trajectories, and so don't fit the MIMICRI assumptions well. In our work, we sidestep this problem by using cepstrum analysis to reduce the glottal source contribution to acoustics. We speculate that vocal-fold vibrations may be incorporated in the future if the fundamental frequency of the vocal folds is itself a nonlinear function of some other variable (tension?) that varies smoothly.

To bolster the claim that articulator positions can be recovered from acoustics, we have applied our algorithm to acoustic recordings of vowel-to-vowel transitions produced by one speaker. For vowel production, the most critical aspects of articulation are the horizontal and vertical positions of the tongue and the lip opening. We have found that the resulting articulator position estimates are strongly correlated with measured articulator positions, demonstrating MIMICRI's ability to invert mappings. Although the estimated articulator positions are not recovered as accurately when we use MIMICRI as when we use a supervised algorithm, the correlation differences between MIMICRI and the supervised learning algorithm for the tongue x and y positions are typically 3–4% with larger error for y positions of the upper and lower lip, about 7–10%.

Since collecting articulator positions is always difficult, the increased estimation error may be acceptable as an alternative to collecting articulator measurements. In fact, there is a compelling reason to believe that articulator positions estimated by MIMICRI can give better performance results than those obtained by a supervised approach for some applications. With the current technology, we are never able to obtain measurements of all the articulators that may be involved in speech production. For example, the velum position, larynx height, and pharynx opening are difficult to measure. If articulator measurements are not made, then supervised learning algorithms will not learn to estimate them. Since MIMICRI does not need articulator measurements to learn the relationship between acoustics and articulation, MIMICRI might outperform supervised learning on tasks in which the unmeasured articulators provide critical information.

While the results show that MIMICRI was able to do the inversion to within an affine transformation, they also show that, for the vowel data we studied, the mapping from spectra to articulator positions has a large linear component. An affine transformation from spectra to articulator positions gives reasonably good correlations to measured articulator positions for this data. MIMICRI outperformed estimating articulator positions using an affine transformation of the spectra by about 5% for x-positions of the tongue, 3% for the y-position of the rear of the tongue, and 4–5% for y-positions of the upper and lower lips. MIMICRI is about as accurate as the spectrum-estimated positions for estimating the y-position for the tip of the tongue, which is not a critical articulator for the data we studied. We expect MIMICRI to outperform spectra to a greater degree on more complicated data sets, such as sets that include consonants, in which more nonlinearity can be expected.

The quality of MIMICRI shown most strongly by this data was the ability to reduce the dimensionality of a signal while retaining information about articulation. The acoustic speech signals processed by MIMICRI consisted of sequences of 128-dimensional acoustic feature vectors (smoothed spectra). Since each acoustic vector was nonlinearly generated from 32 cepstrum coefficients, it was clear

that the dimensionality could be reduced from 128 to no more than 32. Furthermore, since speech is produced by the motions of relatively few articulators, it seemed likely that the dimensionality could be reduced even further. MIMICRI reduced the 128-dimensional spectra down to 5-dimensional vectors that were more highly correlated with articulator positions than the spectra. Again, since we were not able to measure all of the potentially interesting articulators, it is possible that more dimensions would be required for applications such as speech recognition. Nonetheless, the dimensionality reduction that MIMICRI achieved was remarkable.

The particular speech inversion problem we studied did not provide a clear answer to whether MIMICRI can invert any many-to-one mappings that occur in speech production. It is possible that many-to-one mappings are not a significant problem for the data we examined (vowel-to-vowel transitions), but are a problem for other data sets. We hope to address this issue in future work.

It is likely that MIMICRI will be improved by the addition of further knowledge about the relationship between speech production and acoustics. One relatively simple improvement would be to add or infer information about articulator variability for given vector quantization (VQ) codewords. The algorithm studied here assumed that the articulator position variance stayed constant across the VQ codewords. However, in simulation work, it was important to allow the variance to change between VQ codewords in order to compensate for many-to-one mappings. In real articulator data, we are likely to see changes in the articulatory variability across VQ codewords, and we may see many-to-one mappings. Therefore, this approach seems like a good path to improving accuracy.

To the extent that MIMICRI can invert the mapping from articulation to acoustics, articulator configurations hold the promise of being a very robust characterization of speech. For example, a nonlinear microphone will change most characterizations of acoustics (spectra, cepstra, linear prediction coefficients, etc.), but should not affect the articulator positions inferred by MIMICRI, since the microphone nonlinearities are just another layer of nonlinearity to invert and are almost certainly easier to invert than the articulation-to-acoustics transform itself. See (Abuelma'atti, 1990; Quatieri et al., 2000) for examples of nonlinear microphone modeling. Furthermore, if the articulator positions of any particular individual can be treated as nonlinear transformations of the articulator positions of some hypothetical canonical speaker, then MIMICRI will be robust to speaker differences as well as microphone differences. At this point we can only conjecture that such robustness is possible, but it is certainly a worthwhile goal to pursue.

Of course, many of these same goals might be achievable using supervised learning algorithms to find the mapping from acoustics to articulation, but a different set of time-aligned articulator and acoustic measurements would be needed for each individual and each microphone. Gathering such data sets would be essentially impossible – a fact which emphasizes the need to learn the mapping using a blind algorithm such as MIMICRI.

What we have presented in this paper is not intended to be a specific algorithm to be incorporated in an ASR system, or a specific component of a psycholinguistic model of speech. Rather, as applied to speech, the work presented here can be construed as a *theory* about the nature of the information in the speech signal, a theory which may explain many experimental findings and which has implications for the development of specific models and experiments in both of those domains.

The theory is that the information in speech can be represented in a small number (5–10) of smoothly changing signals of very limited bandwidth. It is easiest to visualize these signals as articulatory movements, but what we have demonstrated is that there is nothing special about that interpretation of them. There is a family of such signals, related by affine transformations, that are (informationally) equivalent. The theory predicts that certain transformations of the speech signal that afford recovery in the way we have described here should be intelligible to listeners (they are information-preserving), while other transformations that do not allow recovery will be unintelligible. For example, speech remains intelligible despite the degradations introduced by sinewave speech (Remez et al., 1981), spectral inversion (Blesser, 1972), infinite peak clipping (Licklider and Pollack, 1948), and re-synthesis using only the phase spectrum (Paliwal and Alsteris, 2005). These are all non-linear transforms of the original signal, where the nature of the transform does not vary over time. If the smoothly changing signals are the carriers of speech information, the theory presented here suggests that these smooth signals can be recovered in all these cases, which would explain their intelligibility. Of course, whether recovery can actually succeed in all these cases needs to be tested in computational experiments.

Other kinds of manipulations have been shown to destroy intelligibility, and in these cases the theory predicts that recovery of the original smooth information functions will be impossible. For example, time-reversing a sentence will degrade its intelligibility almost completely. It is likely that smooth signals constructed after time reversal will not be the same as smooth signals constructed from the original signals (they should be time-reversed versions). However, when time-reversal is applied in short chunks of <50 ms, perception is only minimally impaired (Saberi and Perrott, 1999). Here we would predict that the rate of change of the information signals is likely slow enough that, on average, there is not much change in the 50 ms chunks, so recovery of the original underlying smooth signals is still possible. Relatedly, replacing even larger (55–125 ms) chunks of speech with silence can be tolerated, just in cases where the silences replace sections of the signal during which there is little change over time, for example the temporal center of vowels (Strange et al., 1983). Again these hypotheses need to be tested in computational experiments.

The theory also makes predictions about findings in multi-modal speech perception. For example, the equivalence and integrability of audio, visual, tactile, somatosensory information in speech (e.g., Rosenblum, 2005) could be explained, since the sensory information in each modality results from a nonlinear transform of the smooth articulator movements during speech. However, we know that the information in individual modalities may not be sufficient by itself to maintain intelligibility. For example, visual information from the face of a talker is not by itself sufficient to maintain intelligibility. The theory here predicts again that differences in intelligibility should be correlated with differences in function recoverability, and the relevant computational experiments can be performed to test this.

Psycholinguistic models need to provide an account of the fundamental informational structure of speech that underlies the sensory information that listeners encounter, and the account should provide insight as to which transformations of speech affect intelligibility. At present, most psycholinguistic models only account for particular transformations and experiments. While models have been proposed that are capable of accounting for the individual findings noted above, these models are not related to one another and do not make general predictions about what kinds of transforms will distort intelligibility and which will not. Such predictions are made by the theory presented here and they can be tested.

The issue of how this theory should be incorporated into psycholinguistic models or into ASR implementations is, of course, an entirely unanswered question. But we think that attempts to model human speech recognition need to incorporate a realistic model of the nature of the information in speech, and the theory we have presented here takes the first steps at formalizing that.

Acknowledgments

The authors are indebted to two anonymous reviewers whose comments significantly improved the readability and accuracy of the paper. The authors also gratefully acknowledge the technical editing help of Charmian Schaller, without whom the reader would have to wade through numerous punctuation errors and many other grammatical problems.

References

Abuelma'atti, M., 1990. Harmonic and intermodulation distortion in carbon microphones. Appl. Acoust. 31, 233–243.

Aczel, J., Dhombres, J., 1989. In: Rota, G.-C. (Ed.), Functional Equations in Several Variables with Applications to Mathematics, Information Theory, and the Natural and Social Sciences. In: Encyclopedia of Mathematics and its Applications Series, Vol. 31. Cambridge University Press, Cambridge.

Ahalt, S., Krishnamurthy, A., Chen, P., Melton, D., 1990. Competitive learning algorithms for vector quantization. Neural Networks 3, 277–290.

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. Amer. 63 (5), 1535–1555.

Badin, P., Beautemps, D., Laboissière, R., Schwartz, J.-L., 1995. Recovery of vocal tract geometry from formants for vowels and fricative consonants using a midsagittal-to-area function conversion method. J. Phonetics 23, 221–229.

Balchandran, R., Mammone, R., 1998. Non-parametric estimation and correction of non-linear distortion in speech systems. Proc. Internat. Conf. Acoust. Speech Signal Process. Vol. 2, 749–752.

Beautemps, D., Badin, P., Bailly, G., 2001. Linear degrees of freedom in speech production: analysis of cineradio- and labio-film data and articulatory-acoustic modeling. J. Acoust. Soc. Amer. 109 (5), 2165–2180.

Bendat, J., 1998. Nonlinear Systems Techniques and Applications. Wiley, New York.

Beneviste, A., Goursat, M., 1984. Blind Equalizers. IEEE Trans. Communications 32, 871–883.

Blackburn, C.S., Young, S., 2001. Enhanced speech recognition using an articulatory production model trained on X-ray data. Comput. Speech Language 15, 195–215.

Blesser, B., 1972. Speech perception under conditions of spectral transformation: I. Phonetic characteristics. J. Speech Hear. Res. 15, 5–41.

Boe, L.J., Perrier, P., Bailly, G., 1992. The geometric vocal tract variables controlled for vowel production: proposals for constraining acoustic-to-articulatory conversion. J. Phonetics 20, 27–38.

Carreira-Perpinan, M.A., 2001. Continuous latent variable models for dimensionality reduction and sequential data reconstruction. Unpublished Ph.D. Dissertation, University of Sheffield, Sheffield, UK.

Coker, C., 1976. A model of articulatory dynamics and control. Proc. IEEE 64 (4), 452–460.

Dempster, A., Laird, N., Rubin, D., 1977. Maximum likelihood from incomplete data via the EM algorithm (with discussion). J. Roy. Statist. Soc. Ser. B 39, 1–38.

Deng, L., 1998. A dynamic, feature-based approach to the interface between phonology and phonetics for speech modeling and recognition. Speech Comm. 24, 299–323.

Deng, L., Ramsay, G., Sun, D., 1997. Production models as a structural basis for automatic speech recognition. Speech Comm. 22, 93–111.

Dillon, W., Goldstein, M., 1984. Multivariate Analysis: Methods and Applications. Wiley, New York.

Dusan, S., Deng, L., 2000. Acoustic-to-articulator inversion using dynamical and phonological constraints. In: Proc. 5th Seminar on Speech Production, pp. 237–240.

Edelman, A., Arias, T., Smith, S., 1998. The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. 20 (2), 303–353.

Fant, G., 1970. Acoustic Theory of Speech Production, second ed. Mouton and Co, The Hague.

Flanagan, J., 1972. Speech Analysis, Synthesis, and Perception, second ed. Springer, New York.

Fowler, C., Turvey, M., 1980. Immediate compensation in bite-block speech. Phonetica 37, 306–326.

Frankel, J., King, S., 2001a. ASR – articulatory speech recognition. Proc. Eurospeech, 599–602.

Frankel, J., King, S., 2001b. Speech recognition in the articulatory domain: Investigating an alternative to acoustic HMMs. In: Proc. Workshop for Innovations in Speech Processing.

Gray, R., 1984. Vector quantization. IEEE Acoust. Speech Signal Process. Mag., 4–29.

Guenther, F., Hampson, M., Johnson, D., 1998. A theoretical investigation of reference frames for the planning of speech movements. Psychol. Rev. 105 (4), 611–633.

Gupta, S., Schroeter, J., 1993. Pitch-synchronous frame-by-frame and segment-based articulatory analysis by synthesis. J. Acoust. Soc. Amer. 94 (5), 2517–2530.

Hirayama, M., Vatikiotis-Bateson, E., Honda, K., Koike, Y., Kawato, M., 1992. Physiologically based speech synthesis. Poster presented at Neural Information Processing Systems (92).

Hiroya, S., Honda, M., 2004. Estimation of articulatory movements from speech acoustics using an HMM-based speech production model. IEEE Trans. Speech Audio Process. 12 (2), 175–185.

Hogden, J., 1991. Low-dimensional phoneme mapping using a continuity constraint. Unpublished Doctoral Dissertation, Stanford University, Stanford, CA.

Hogden, J., 1995. A maximum likelihood approach to estimating speech articulator positions from speech acoustics. In: Neural Information Processing Systems (95), Workshop on Neural Networks for Speech Processing, Vail, CO.

Hogden, J., 2000. USA Patent # 6,052,662.

Hogden, J., Valdez, P., 2000. Bridging the gap between speech production and speech recognition. In: Proc. 5th Seminar on Speech Production: Models and Data, Kloster Seeon, Bavaria, May 1st–4th.

Hogden, J., Rubin, P., Saltzman, E., 1992. An unsupervised method for learning to track tongue position from an acoustic signal. J. Acoust. Soc. Amer. 91 (4), 2443 (A).

Hogden, J., Saltzman, E., Rubin, P., 1993. Tracking moving objects with unsupervised neural networks. Paper presented at the World Conference on Neural Networks, Portland, OR.

Hogden, J., Rubin, P., Saltzman, E., 1996a. An unsupervised method for learning to track tongue position from an acoustic signal. Bull. Communication Parlee 3, 101–116.

Hogden, J., Zlokarnik, I., Lofqvist, A., Gracco, V., Rubin, P., Saltzman, E., 1996b. Accurate recovery of articulator positions from acoustics – new conclusions based on human data. J. Acoust. Soc. Amer. 100 (3).

Hogden, J., Nix, D., Valdez, P., 1998. Maximum likelihood continuity mapping: bridging the gap between production and recognition. Paper presented at the 9th Hub-5 Conversational Speech Recognition Workshop, Linthicum Heights, MD, USA, September 24th–25th.

Hogden, J., Valdez, P., Katagiri, S., McDermott, E., 2003. Blind inversion of multidimensional functions for speech enhancement. Proc. Eurospeech Conf., 1409–1412.

Kaburagi, T., Honda, M., 2001. Dynamic articulatory model based on multidimensional invariant-feature task representation. J. Acoust. Soc. Amer. 110 (1), 441–452.

Kambhatle, N., Leen, T., 1997. Dimension reduction by local PCA. Neural Computation 9, 1–18.

Kimber, D., 1994. Geometric methods for nonparametric modeling of dynamical systems. Unpublished Ph.D. Dissertation, Stanford University, Stanford, CA.

Kirchhoff, K., 1998. Robust Speech Recognition using Articulatory Information (No. TR-98-037). International Computer Science Institute, Berkeley, CA.

Kuc, R., Tutuer, F., Vaisnys, J.R., 1985. Determining vocal tract shape by applying dynamic constraints. In: Proc. Internat. Conf. on Acoustics Speech and Signal Processing, Tampa, FL.

Levin, D., 2002. Representations of sound that are insensitive to spectral filtering and parameterization procedures. J. Acoust. Soc. Amer. 111 (5), 2257–2271.

Liberman, A., Cooper, F., Shankweiler, D., Studdert-Kennedy, M., 1967. Perception of the speech code. Psychol. Rev. 74 (6), 431–461.

Liberman, A., Mattingly, I., 1985. The motor theory of speech perception revised. Cognition 21, 1–36.

Licklider, J., Pollack, I., 1948. Effects of differentiation, integration, and infinite peak clipping upon the intelligibility of speech. J. Acoust. Soc. Amer. 20, 42–51.

Lindblom, B., 1996. Role of articulation in speech perception: clues from production. J. Acoust. Soc. Amer. 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. Phonetics 7, 146–161.

Lyons, R., 2004. Understanding Digital Signal Processing, second ed. Prentice-Hall, Upper Saddle River, NJ.

Maeda, S., 1979. An articulatory model of the tongue based on a statistical analysis. J. Acoust. Soc. Amer. 65, S22.

Markel, J., Gray, A., 1976. Linear Prediction of Speech. Springer, New York.

Maxson, C., Meyer, J., 2001. How many subspaces force linearity? Amer. Math. Monthly 108 (6), 531–536.

McDermott, E., Nakamura, A., 2006. Production-oriented models for speech recognition. IEICE Trans. Special Issue on Statistical Modeling for Speech Processing E89-D (3), 1006–1014.

McGowan, R., 1987. Articulatory synthesis: numerical solution of a hyperbolic differential equation. Haskins Laboratories Status Report on Speech Research, SR-89/90.

McGowan, R., Faber, A., 1996. Introduction to papers on speech recognition and perception from an articulatory view. J. Acoust. Soc. Amer. 99 (3), 1680–1682.

McGowan, R., Lee, M., 1996. Task dynamic and articulatory recovery of lip and velar approximations under model mismatch conditions. J. Acoust. Soc. Amer. 99 (1), 595–608.

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

Moody, J., 1999. Visualizing speech with a recurrent neural network trained on human acoustic-articulator data. Unpublished Ph.D. Dissertation, University of California, San Diego, CA.

Morris, R., Clements, M., 2001. Maximum-likelihood compensation of zero-memory nonlinearities in speech signals. In: Proc. Internat. Conf. on Acoustics, Speech, and Signal Processing, pp. 289–292.

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

Nelson, W., 1977. Articulatory feature analysis – I. Initial processing considerations. Bell Laboratories, Memorandum.

Nix, D., 1998. Machine learning methods for inferring vocal-tract articulation from speech acoustics. Unpublished Ph.D. Thesis, University of Colorado, Boulder, CO.

Oppenheim, A., 1969. Speech analysis-synthesis system based on homomorphic filtering. J. Acoust. Soc. Amer. 45 (2), 458–465.

Ouni, S., Laprie, Y., 2005. Modeling the articulatory space using a hypercube codebook for acoustic-to-articulatory inversion. J. Acoust. Soc. Amer. 118 (1), 444–460.

Paliwal, K., Alsteris, L., 2005. On the usefulness of STFT phase spectrum in human listening tests. Speech Comm. 45, 153–170.

Papcun, G., Hotchberg, J., Thomas, T., Laroche, F., Zacks, J., Levy, S., 1992. Inferring articulation and recognizing gestures from acoustics with a neural network trained on X-ray microbeam data. J. Acoust. Soc. Amer. 92 (2), 688–700.

Perkell, J., Cohen, M., Svirsky, M., Matthies, M., Garabieta, I., Jackson, M., 1992. Electromagnetic midsagittal articulometer systems for transducing speech articulatory movements. J. Acoust. Soc. Amer. 92 (6), 3078–3096.

Perkell, J., Mathies, M., Svirsky, M., Jordan, M., 1993. Trading relations between tongue-body raising and lip rounding in production of the vowel /u/: A pilot 'motor equivalence' study. J. Acoust. Soc. Amer. 93 (5), 2948–2961.

Perrier, P., Labiossiere, R., Abry, C., Maeda, S., 1997. Speech production: models and data. Speech Comm. 22, 82–92.

Qiu, W., Hua, Y., 1997. A GCD method for blind channel identification. Digital Signal Processing 7, 199–205.

Quatieri, T.F., Reynolds, D.A., O'Leary, G.C., 2000. Estimation of handset nonlinearity with application to speaker recognition. IEEE Trans. Speech Audio Process. 8 (5), 567–584.

Remez, R.E., Rubin, P.E., Pisoni, D.B., Carrell, T.D., 1981. Speech perception without traditional speech cues. Science 212, 947–950.

Reynolds, D., 1996. The effects of handset variability on speaker recognition performance: experiments on the Switchboard corpus. In: Proc. IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing, pp. 113–116.

Richards, H., Bridle, J., 1999. The HDM: a segmental hidden dynamic model of coarticulation. Proc. IEEE Internat. Conf. Acoustics Speech Signal Process. 1, 357–360.

Richards, H., Bridle, J., Hunt, M., Mason, J., 1997. Dynamic constraint weighting in the context of articulatory parameter estimation. Proc. Eurospeech 97.

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

Rosenblum, L.D., 2005. The primacy of multimodal speech perception. In: Pisoni, D., Remez, R. (Eds.), Handbook of Speech Perception. Blackwell, Malden, MA, pp. 51–78.

Roweis, S., 1999. Data driven production models for speech processing. Unpublished Ph.D. Thesis, California Institute of Technology, Pasadena, CA.

Roweis, S., Alwan, A., 1997. Towards articulatory speech recognition: learning smooth maps to recover articulator information. Proc. Eurospeech 3, 1227–1230.

Roweis, S., Saul, L., 2000. Nonlinear dimensionality reduction by locally

Schroeder, M., 1967. Determination of the geometry of the human vocal tract by acoustic measurements. J. Acoust. Soc. Amer. 41 (4), 1002–1010.

Schroeter, J., Sondhi, M., 1994. Techniques for estimating vocal-tract shapes from the speech signal. IEEE Trans. Speech Audio Process. 2 (1), 133–150.

Scully, C., 1979. Model prediction and real speech: fricative dynamics. In: Lindblom, B., Ohman, S. (Eds.), Frontiers of Speech Comm. Research. Academic Press, New York, pp. 35–48.

Shirai, K., Kobayashi, T., 1986. Estimating articulatory motion from speech wave. Speech Comm. 5, 159–170.

Sondhi, M., Resnick, J.R., 1983. The inverse problem for the vocal tract: numerical methods, acoustical experiments, and speech synthesis. J. Acoust. Soc. Amer. 73 (3), 985–1002.

Sorokin, V., Trushkin, A.V., 1996. Articulatory-to-acoustic mapping for inverse problem. Speech Comm. 19, 105–118.

Strang, G., 1980. Linear Algebra and Its Applications, second ed. Academic Press, New York.

Strange, W., Jenkins, J., Johnston, T., 1983. Dynamic specification of coarticulated vowels. J. Acoust. Soc. Amer. 74, 695–705.

Suzuki, S., Okadome, T., Honda, M., 1998. Determination of articulatory positions from speech acoustics by applying dynamic articulatory constraints. In: Proc. Internat. Conf. on Spoken Language Perception, pp. 2251–2254.

filtering of acoustic speech waveforms. IEEE Trans. Audio Electroacoust. AU-21 (5), 417–427.

Whalen, D., Wiley, E., Rubin, P., Cooper, F., 1990. The Haskins Laboratories' pulse code modulation (PCM) system. Behavioral Res. Methods Instruments Comput. 22 (6), 550–559.

Yehia, H., Rubin, P., Vatikiotis-Bateson, E., 1998. Quantitative association of vocal-tract and facial behavior. Speech Comm. 22 (1–2), 23–43.

Zlokarnik, I., 1995. Adding articulatory features to acoustic features for automatic speech recognition. J. Acoust. Soc. Amer. 97 (5 pt. 2), 3246 (A).