LA-UR-02-5325

Approved for public release;
distribution is unlimited.

Title: THE MANIFOLD ADVANTAGES OF ARTICULATORY REPRESENTATIONS, INCLUDING MICROPHONE AND SPEAKER NORMALIZATION

Author(s): John E. Hogden Z# 116911, CCS-3
Patrick F. Valdez Z# 150153, CCS-3
Leonid Gurvits Z# 174471, CCS-3

Submitted to: Center for Language and Speech Processing Workshop 2002
Johns Hopkins University, Baltimore Maryland
June 1 - Aug 23, 2002

Los Alamos
NATIONAL LABORATORY

The Manifold Advantages of Articulatory Representations, Including Microphone and Speaker Normalization

John Hogden (hogden@lanl.gov),
Patrick Valdez, Leonid Gurvits
Los Alamos National Laboratory

I'm going to be presenting work done with the help of a Leonid Gurvits, whois a really good mathematician and Patrick Valdez, who actually did most of the work. I'm just along for the ride.

Outline

- Why recovering an articulatory representation should (not?) be a speech processing step.
- MALCOM: A stochastic model of speech based on articulation.
- MALCOM inverts functions.
- CO-MALCOM can be used for speech recognition

I'm going to be making two broad points during my talk. The first is that we should do a transformation from speech acoustics to articulator positions as part of our speech processing. The second point I will try to make is that we can do a transformation from speech sounds to articulator positions.

So, more specifically, I'll start off by talking about why we should not use an articulatory representation. Then I'll talk about a stochastic model of speech that, during training, finds the way to transforms acoustics to articulator positions. The interesting thing about this algorithm is that it finds the mapping from acoustics to articulation using only acoustic data, which is a bit like saying we can find a nonlinear regression between x and y using only x data.

To support the claim that MALCOM can invert the mapping from acoustics to articulation, I am going to discuss a mathematical proof, simulation results, and an experiment that all argue that MALCOM is a fairly general function inverter.

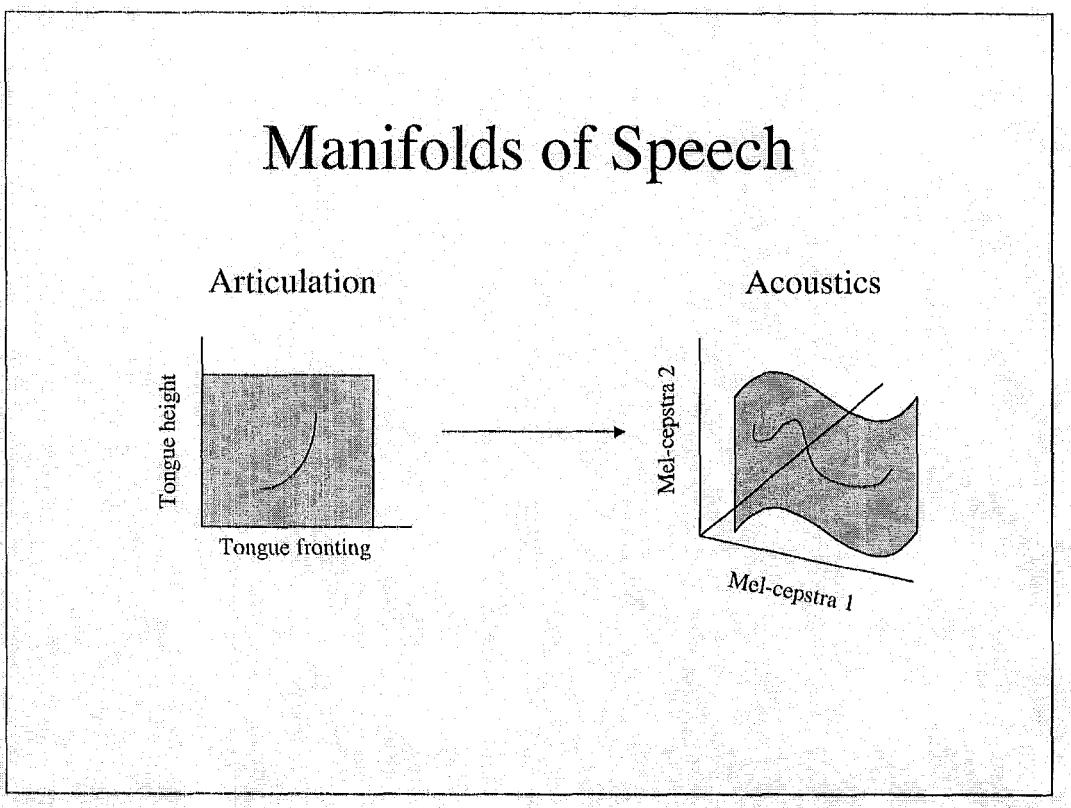
Finally, if I have time, I'll tell you about how I think MALCOM can be combined with current speech recognition algorithms. The interesting point of this last topic will be that we don't have to give up current trellis models to incorporate articulation into our recognition algorithms

Why not articulation?

- It's impossible to recover articulation from acoustics.
- Information will be lost in the processing.

Back when I was a graduate student studying ways to recover articulator positions from acoustics, there were good reasons to believe I was wasting my time. For example, everyone knew that it was impossible to recover articulator positions from acoustics because many different articulator positions could be used to produce that same acoustics. Everyone knew this on the basis of simulations of the vocal tract and various mathematical models of simplified tracts. However, there is now a lot of empirical evidence that the simulations wildly overestimated the extent of the problem. For example, the simulations would predict that tongue positions could change by two centimeters without changing the speech acoustics. However, numerous studies using measured acoustics and measured articulator positions have been able to recover articulator positions from acoustics an order of magnitude better than was predicted by the simulations. I have to be a bit careful here too, because the predictions of the simulations would vary by an order of magnitude depending on their assumptions, but people didn't seem to notice that at the time.

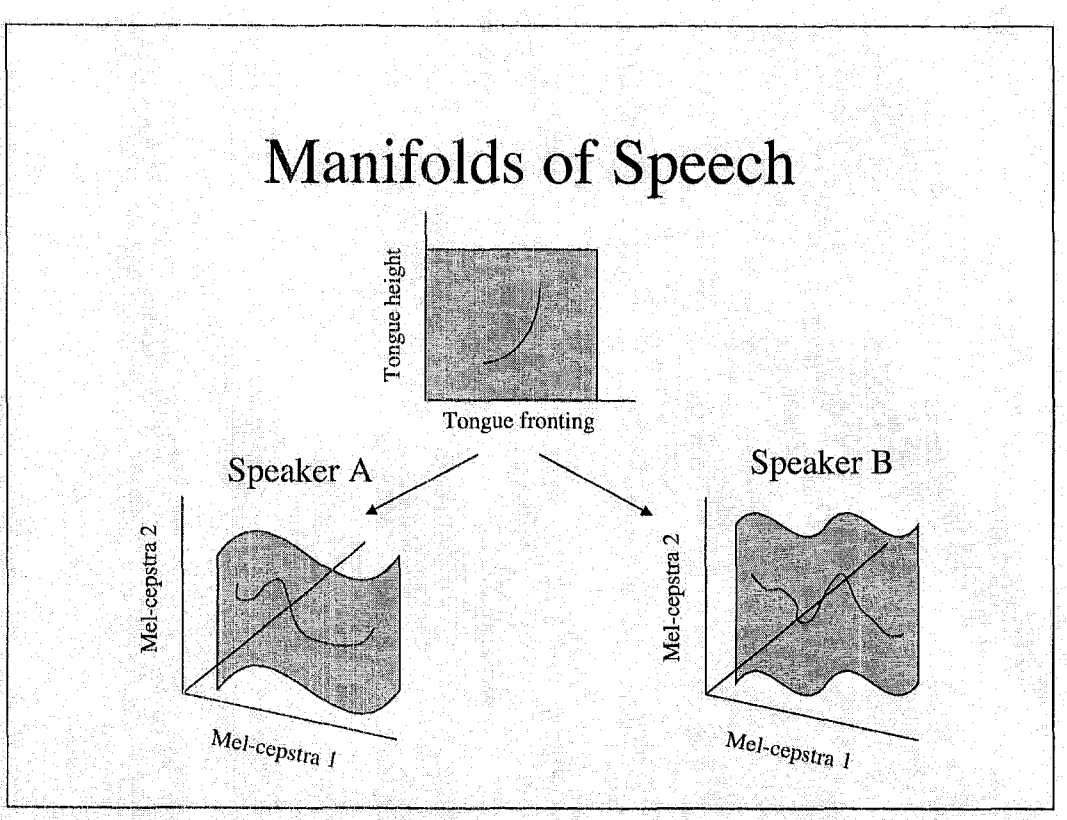
A second very good reason not to try to recover articulator positions was that it would require an extra step of processing which would inevitably throw out some information from the acoustics. Clearly, we would have more information working on the raw acoustics. This argument actually contradicts the first argument. If there is a many to one mapping from articulation to acoustics, then articulation contains more information than acoustics. So if we can invert the many-to-one mapping, which I believe is possible using dynamic information, then we might expect more information in recovered articulator positions than in acoustics.



So let's talk about why you might want to recover articulator positions. In this picture I show, in blue, the set of all articulator positions that can be achieved by a hypothetical vocal tract. In this case I've shown only two dimensions, but there is some such set no matter how many articulator dimensions there are. The red curve is intended to represent a particular trajectory.

This image shows the set of acoustics that can be produced by all the different articulator positions. The shape of the set is not correct or important. What you can be sure of, though, is that the set of acoustics is really just the articulator space contorted and embedded in a higher dimensional space. Similarly, if we have 8 articulator parameters that we transform into 30 -dimensional acoustic vectors, we can be sure that the set of acoustic values that can be produced is an 8 dimensional manifold in a 30 dimensional space. The exception to that is acoustic noise. Having an air conditioner in the background will lead to signal that lie off the manifold. When we use a speech recognition algorithm to estimate Gaussians over the acoustic space, we are really estimating the parameters of 30 dimensional Gaussians to capture the pdfs over an 8 dimensional space. Clearly, most of the area of any Gaussian isn't even on the articulatory manifold. An implication of this is that most of the parameters of the Gaussians are being used to estimate the probability of acoustic noise. When the noise environment changes, the parameters won't be accurate anymore, and we will see recognition performance decrease.

Furthermore, trajectories that are smooth in articulator space are going to be warped like crazy in the acoustic space. That means it will be harder to use temporal context to filter out noise. For example, articulator motions have very little energy above, say, 8-15 hz. If we have noise in our articulator positions estimates, we use a low pass-filter to get rid of most of the noise. We can't do that in acoustic space because then we really would be throwing away information. So we are stuck using the previous time step to predect the next and that means we use much less context to get rid of noise than when we work in an articulator space.



Furthermore, the manifold for a different speaker will have a different shape. The dimensionality won't changebut the shape will. If we change the microphone, the manifold will have yet another shape. I speculate that the articulator trqajectories contain most of the information about what is being said, and the mapping from articulation to acoustics has most of the information about who is saying it. To the extent that we can transform the acoustic space back to an articulatory space, we should be able to better separate information about who is talking from information about what is being said.

A final advantage of recovering an articulatory representation is that it helps circumvent the problems with the conditional independence assumption that are often cited for HMM-based recognition systems. After all, it is pretty clear that if you know the articulator position at time t, then knowing articulator positions or acoustics at other times gives very little additional information about the acoustics at t\, and that is what conditional independence means.

Why articulation?

- Better conditional independence assumption.
- Fewer parameters to be learned.
- Easier to use temporal context.
- Separates speaker/microphone info from content.
- Information may be gained in the processing.

So just to summarize, I think an articulatory representation would be better than an acoustic representation for several reasons.

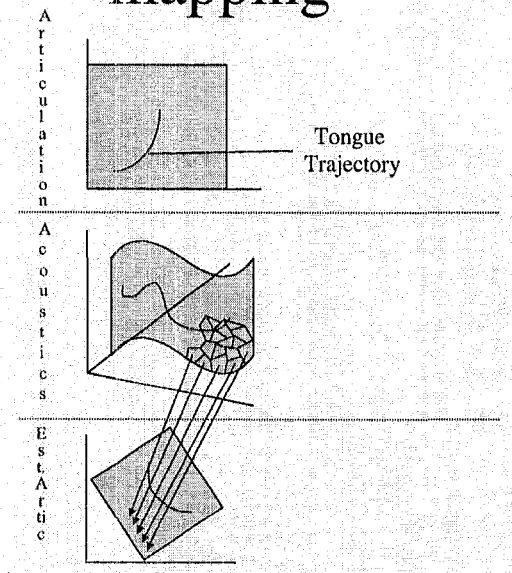
MALCOM

Maximum Likelihood
Continuity Mapping

I'm now going to describe an algorithm called maximum ..., or MALCOM for short, that can recover articulator positions without articulator measurements.

It is important that we can do the recovery without articulator measurements -- we don't want to have to measure articulator positions because the mapping between articulation and acoustics will differ between people and we don't want to have to measure articulator positions for everyone.

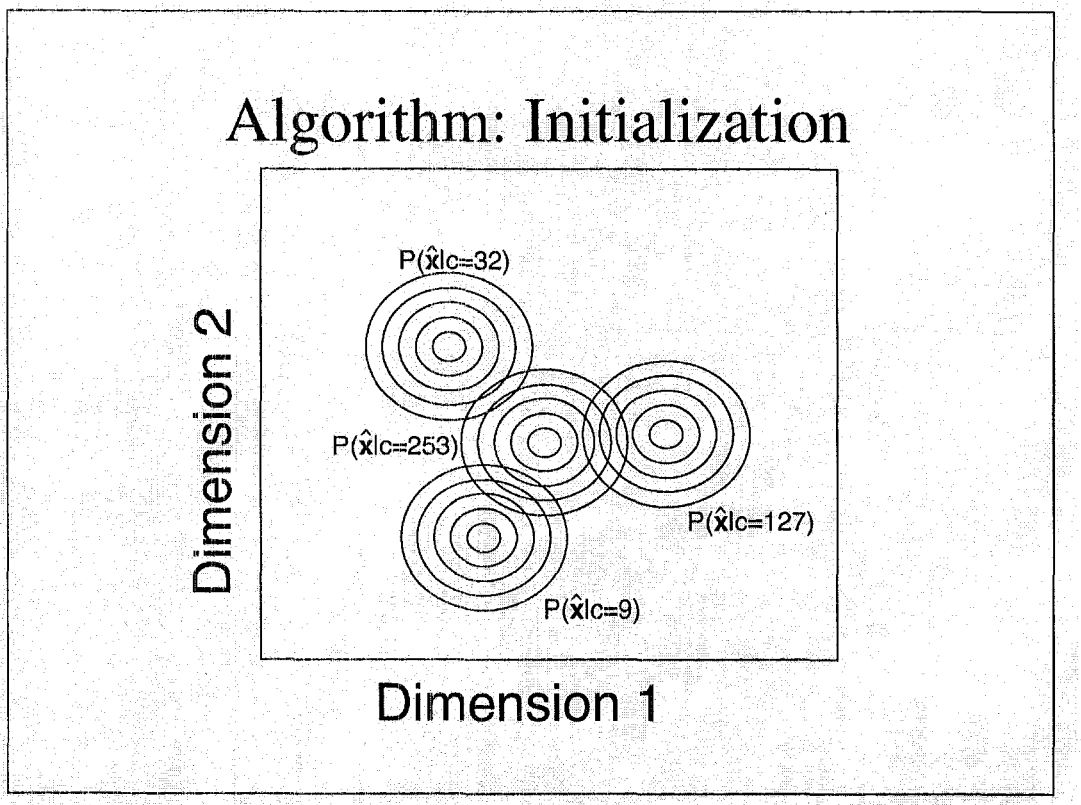
Algorithm: Non-parametric mapping



So here is the articulator space like in the last slides, and here is the articulator manifold embedded in acoustic space. To use MALCOM, we first run vector quantization in the acoustic space, which divides up the manifold like this. Then we feed sequences of VQ codes into MALCOM. From the sequences of VQ codes, MALCOM learns a mapping to a new space which we call a continuity map. So a smooth articulator trajectory maps to a an acoustic sequence, which maps to a sequence of VQ codes, which then maps to a sequence of positions in the continuity map.

Now we know that articulator trajectories have little energy above around 8 Hz. So MALCOM tries to make the mapping from VQ codes to CM positions to make trajectories through the CM have no energy above about 8Hz.

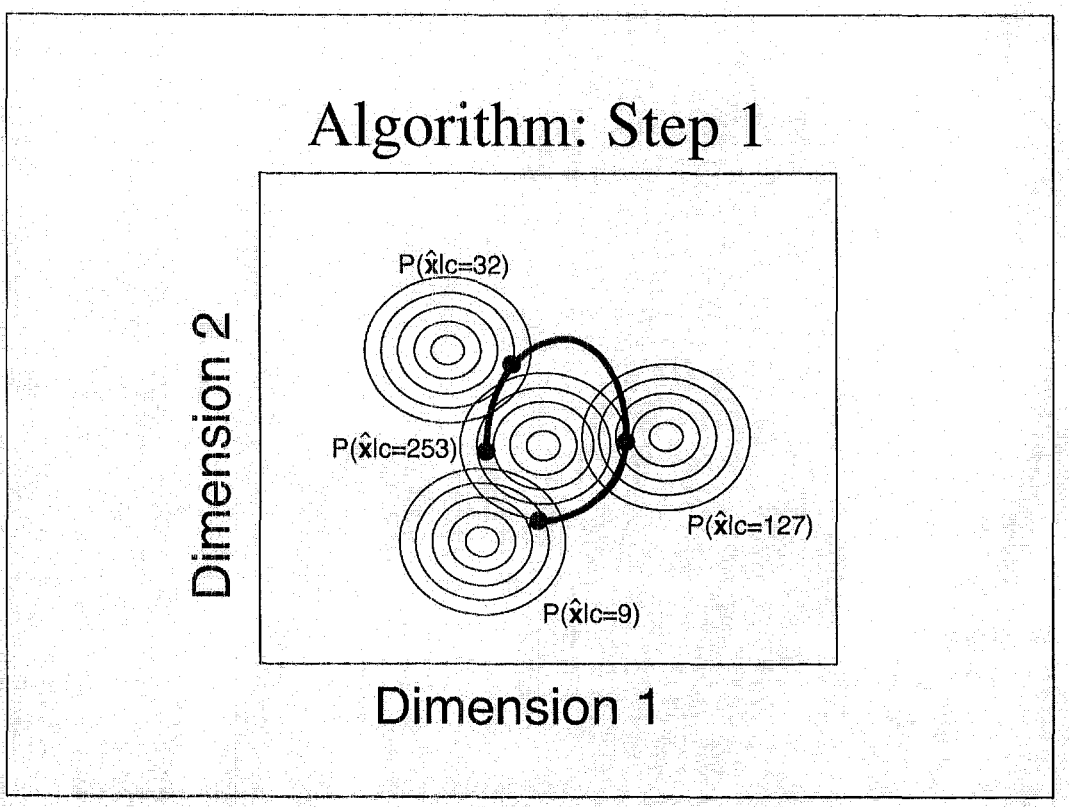
In a bit I will show why this recovers an affine transformation of the articulator positions. What that means is that we won't really get back the articulator positions, but that it will be possible to rotate, scale, translate or reflect the set of positions we get back so that they do match articulator positions.



I lied a bit on the last slide. We aren't really going to map the acoustic regions to points in a new space, instead we will map each region to a pdf over the new space. For example, this set of concentric elipses is supposed to represent the level curves of a Gaussian distribution diving the probability of a position in this space, x, given a region in acoustic space, where each region is refered to by a code, *c*.

Note that if the axes here were articulation, then this would be a kind of stochastic mapping between articulation and acoustics. Given an acoustic region, we could find the probability of each articulator position. Similarly, we could use Bayes' law to get the probability of an acoustic region given an articulator positions.

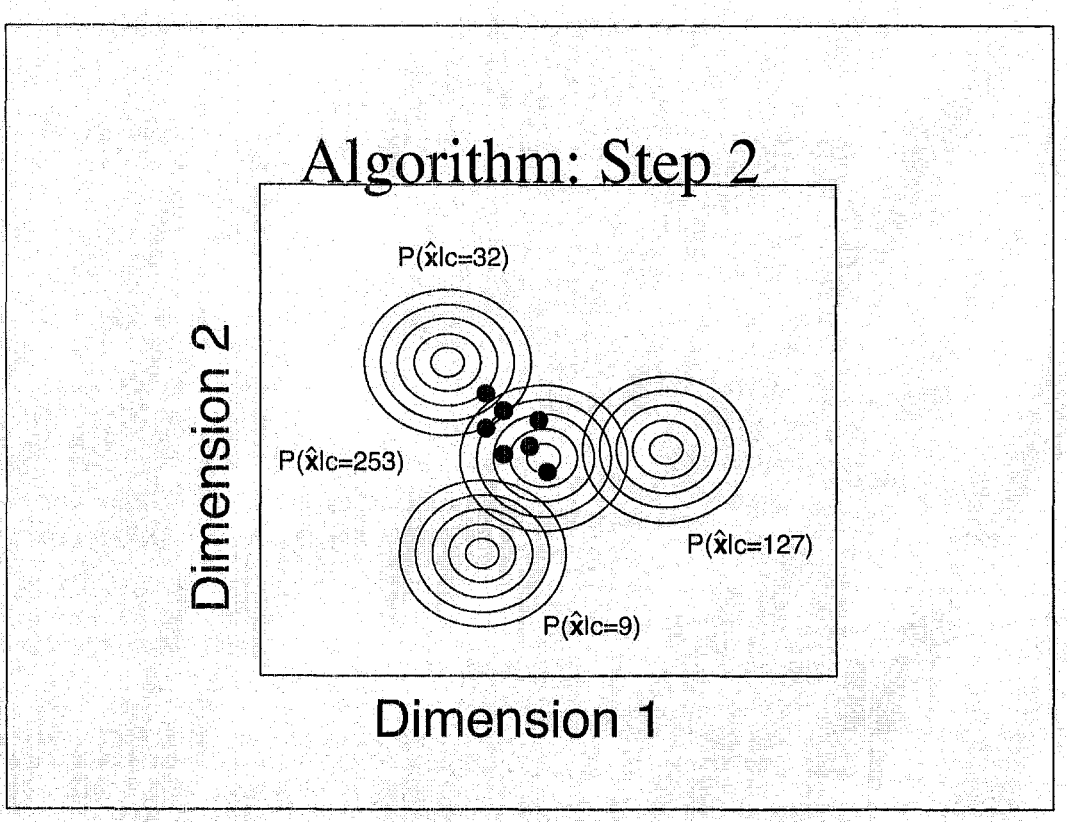
Obviously this random mapping has nothing to do with articulation, but we're going to modify the means and covariances of the gaussians using an EM like algorithm to get better estimates.



The first step of the algorithm is to pretend that the mapping is actually the mapping between acoustics and articulation. Then given a sequence of codes, we can find the most probable smooth path through the space.

By a smooth path, I mean that if we took a fourier transform of the path, we would find that it had no energy above some cut-off frequency, for example, 8 hz.

This is the estimated path position for the first sound type in the sequence, which in this case is sound type 253.



Given lots of sequences of acoustic regions, we could find lots of paths through this constructed space. Given lots of positions corresponding to region 253, for example, we can easily imagine adjusting the mean and covariance structure of this pdf to increase the probability of those points given the pdf.

Algorithm: Summary

- Letting i indicate the current iteration, repeat the following two steps:

$$\hat{\mathbf{X}}^i = \left[\hat{x}_1^i \quad \hat{x}_2^i \quad \cdots \quad \hat{x}_T^i\right] = \underset{\mathbf{X}}{\operatorname{argmax}} \prod_t P\left[x_t \middle| c_t, \hat{\varphi}^{i-1}\right]$$

$$\hat{\varphi}^i = \left[\mu_1 \quad \sigma_1 \quad \cdots \quad \mu_C \quad \sigma_C\right] = \underset{\varphi}{\operatorname{argmax}} P\left[\hat{x}_t^i \middle| c_t, \varphi\right]$$

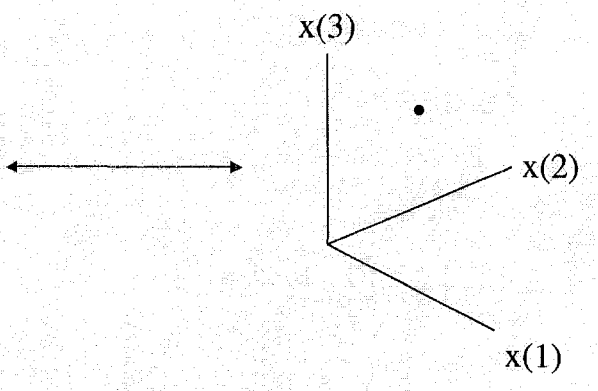
In fact, the whole MALCOM training process is just to repeat those two steps.

Step 1) is to find the most probable paths given the pdf parameters. And step 2 is to increase the probability of the paths by adjusting the pdf paramters.

Each step increases the probability of the paths, so we end up with a kind of maximum likelihood solution.

Bandpass signals lie on a linear subspace

$$\begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(T) \end{bmatrix}$$



So far I've given you no reason to believe that the paths recovered by MALCOM converge to the actual articulator paths. Now I'm going to start explaining why we should expect MALCOM to get back articulation.

In order to understand it, however, we first need to think about articulator trajectories, that is sequences of articulator positions, as points in a high dimensional space.

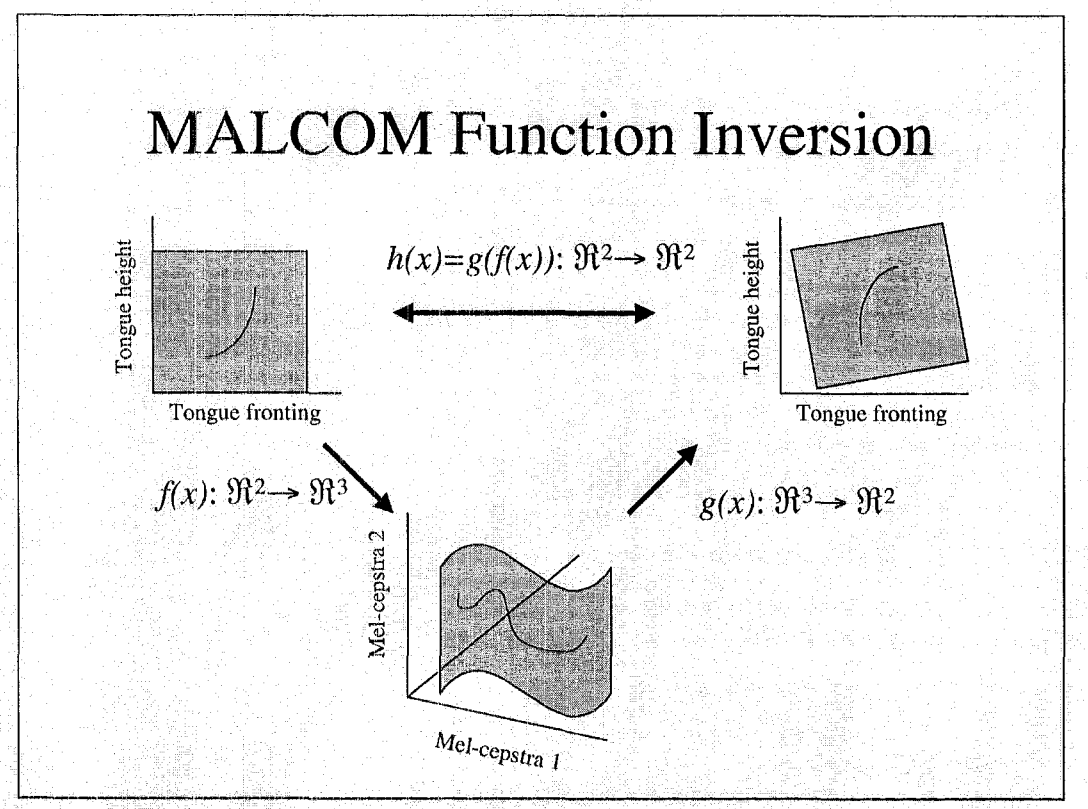
For example, the sequence x(1), x(2), ... can be thought of as a vector or a point in a high dimensional space. If we look at all the trajectories that have no energy above some cutoff frequency, we will find that they lie on a hyperplane cutting through the path space.

Bandpass signals lie on a linear subspace

$$\begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(T) \end{bmatrix} = \begin{bmatrix} \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ \cos_0 & \cos_1 & \sin_1 & \cdots & \cos_{f_c} & \sin_{f_c} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}$$

We know that because the set of all signals with no energy above some cutoff frequency can be represented as a matrix multiplication, where we multiply a matrix whose columns are low frequency cosine and sine waves by some arbitrary vector.

So smooth paths lie on a linear subspace of the path space.

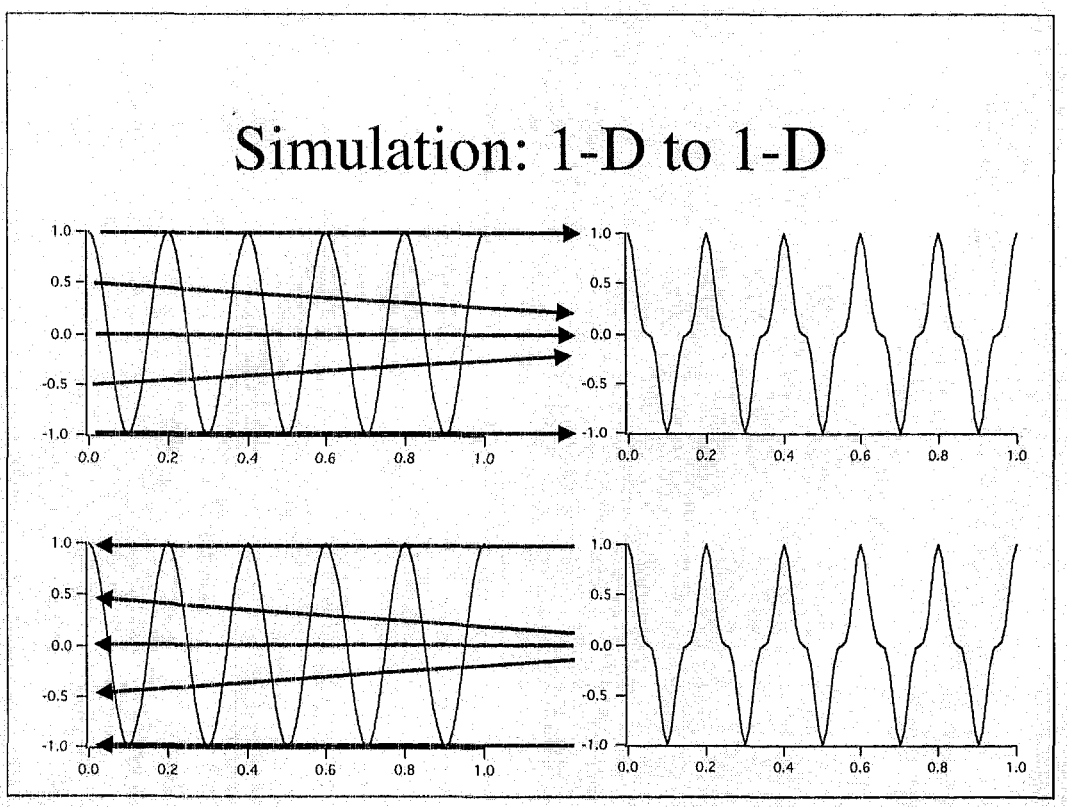


Let's look again at the mappings of speech production. Articulator trajectories make the speech, and articulator trajectories lie on a hyperplane. The vocal tract transforms the trajectories by the non-linear function f(x) into the acoustic space, where the trajectories do not lie on a hyperplane. Finally, MALCOM finds a mapping, g(), from acoustic space back to some space where the paths are again smooth, assuming that is possible.

We can also think about this composite mapping h(x), which is just g() of f(). But since h() maps from smooth paths to smooth paths, it is actually mapping from points lying on a hyperplane to points lying on a hyperplane.

Interestingly, we have a proof that, for hyperplanes of interest, a mapping from the hyperplane to itself must be affine. So the trajectories that MALCOM recovers must be only an affine transformation of the original articulator trajectories.

Typically, we don't care too much about affine differences. For example, the affine transformations of a 1D signal are simply changing the amplitude of the signal or adding a D.C. component. To the extent that these are irrelevant for the problem at hand, MALCOM essentiallly recovers the articulator positions, and inverts the nonlinear transformation, f().



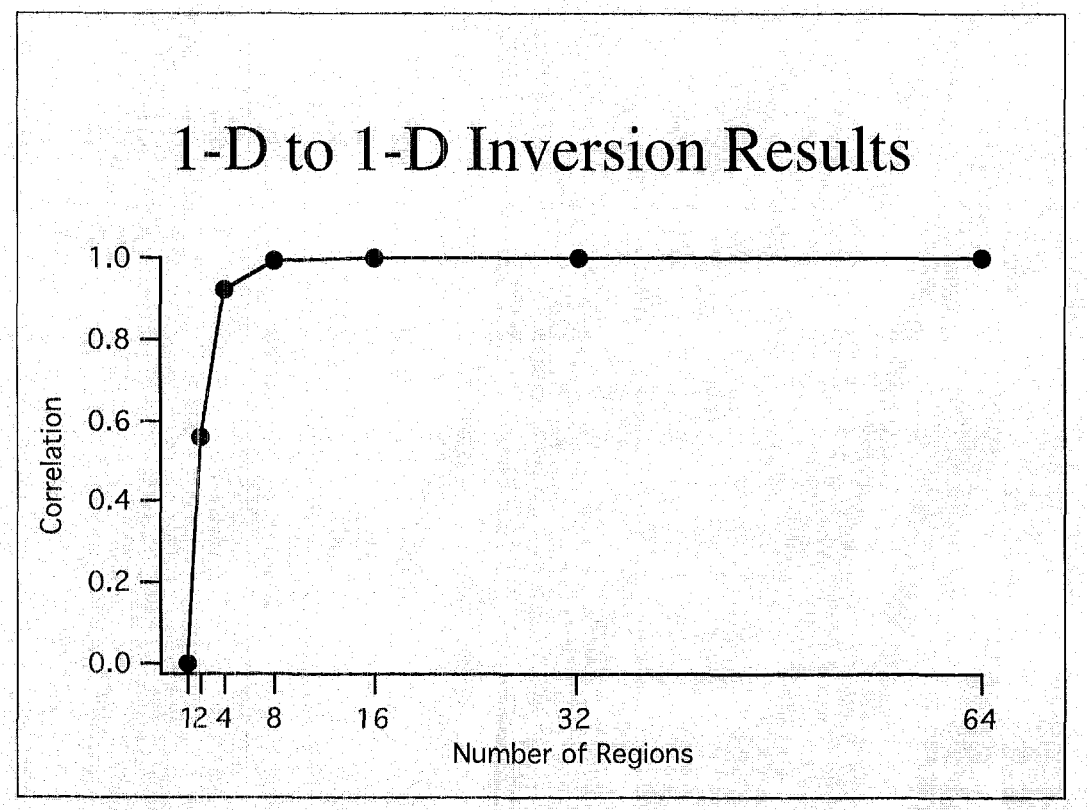
Our proof basically says that any function that maps all points on a linear subspace to the same linear subspace then the function is affine. However, we will never get to see all points on a linear subspace, so you might ask whether we can invert functions when we only get a sampling of points.

We did some simulations to address the issue. We started out looking at 1dimensional signals. We generated a bunch of random smooth paths, which are signals that have random amplitudes and phases of frequencies below some cutoff. Then we do a non-linear warping of the signal. Then we quantize the signal into some number of regions and run MALCOM to see if we can recover the original signal.

So if our random smooth path looked like this, which isn't very random because it has only one cosine wave but go with me on this, then it would get warped by our nonlinearity and MALCOM would try to find a mapping that inverts the warping.

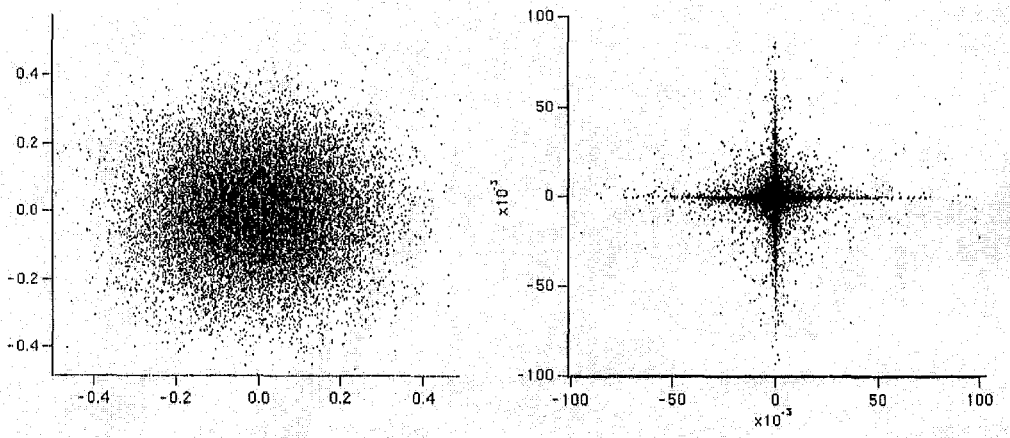
Simulation: 1D to 1D mapping

- Input
  - Cut-off frequency set to 5Hz
  - Sampled at 100 samples/sec
  - Each input is the sum of random amounts of frequencies below the cut-off
  - 100 input signals of 200 samples/signal
- Input/output function
  - cubic



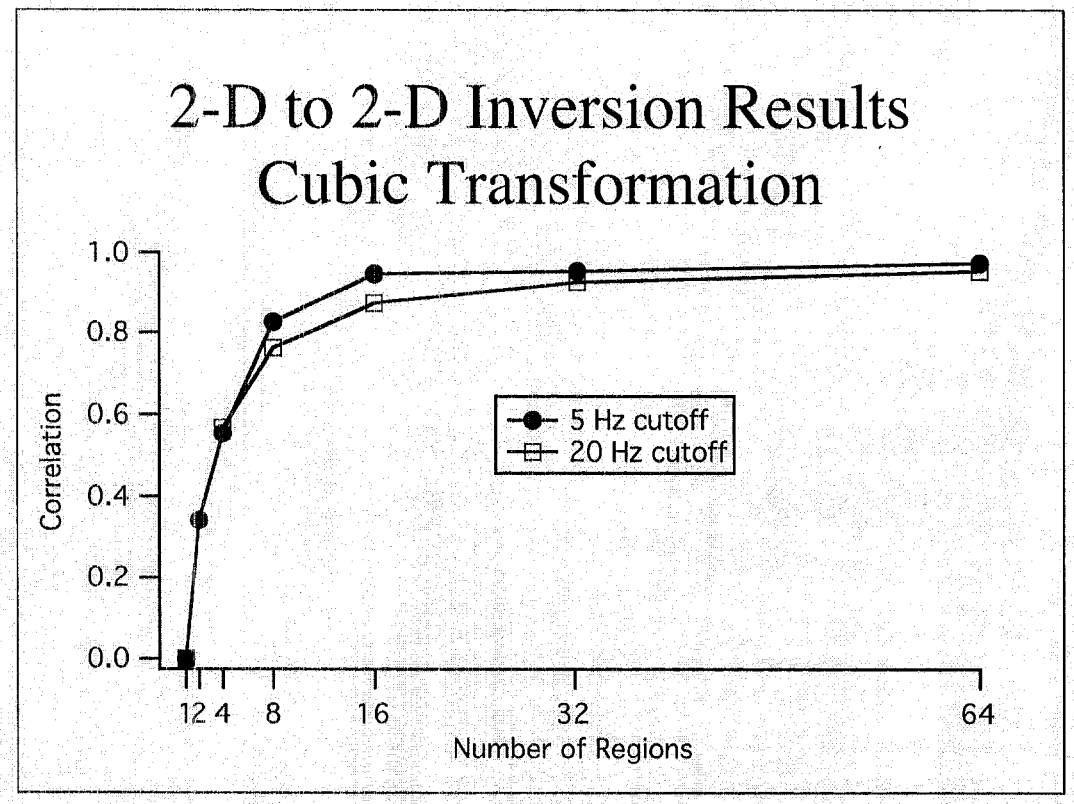
By the way, we chose a cubic warping because Reynolds and Quartieri used a cubic warping to model the effects of a carbon button microphone.

Simulation: 2-D to 2-D Cubic Transformation



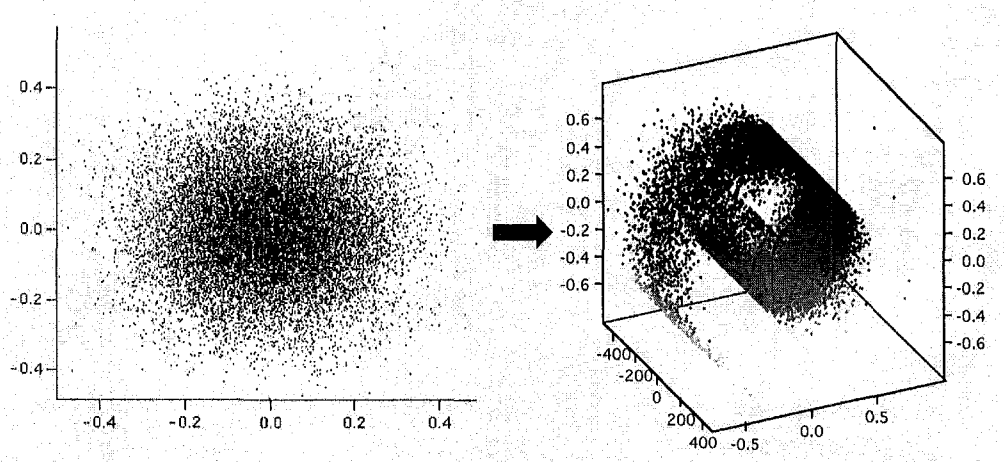
We tried out essentially the same simulation using 2-D paths. So when we generate a couple hundred smooth paths through a 2-D space and look at the points on the paths, we see a distribution of points that looks like this.

When we apply a cubic warping, our distribution is transformed to look like this, which is a pretty severe warping.

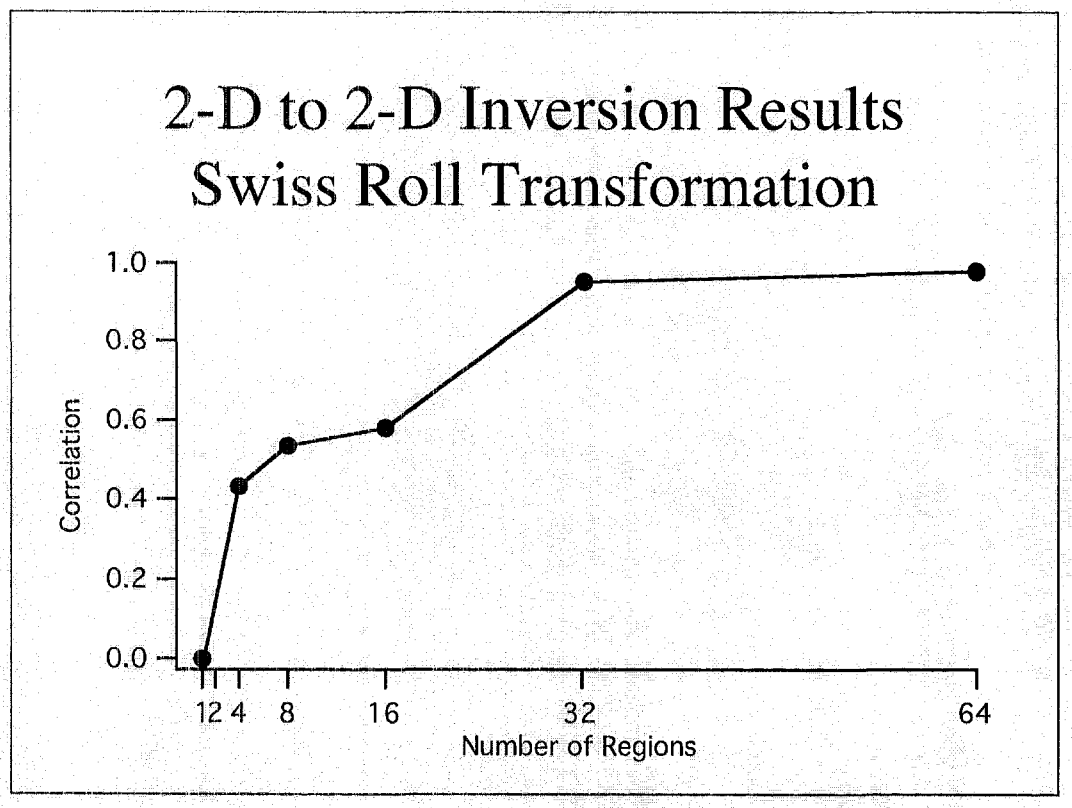


Again, as measured by the correlation between the MALCOM output and the unwarped paths, MALCOM was able to invert the nonlinearity.

Simulation: 2-D to 2-D Swiss Roll Transformation

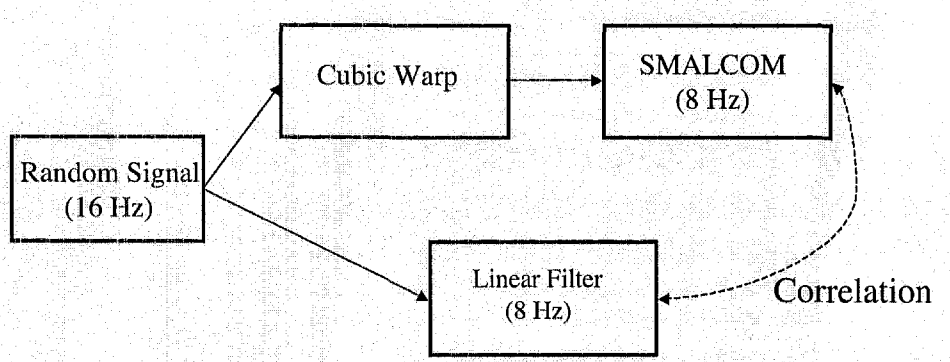


We also looked at a warping that embedded a 2D manifoold in a 3D space. In this case we used a "Swiss Roll" transformation that looks like this.



Again, once we used enough VQ codes, we were able to invert the nonlinear warping.

Frequency Mismatch

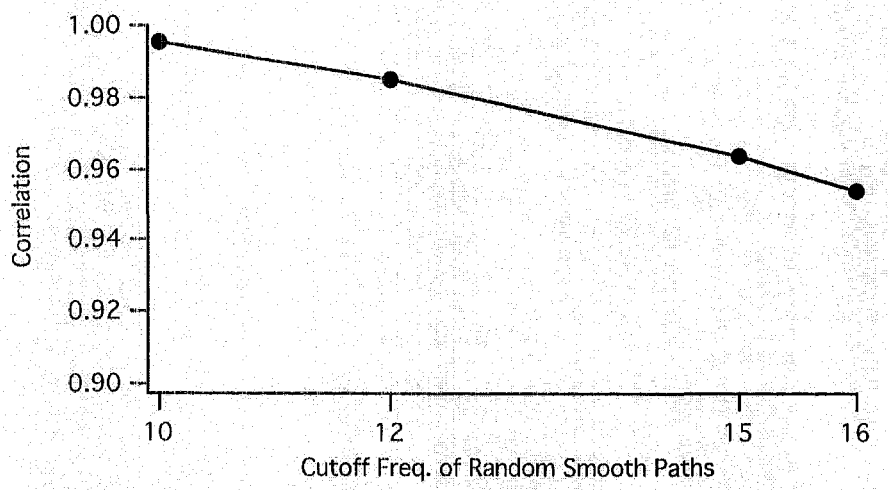


We also realize that we will never see real data that has a strict cutoff frequency. There will always be some energy above the cutoff, so the unwarped signals will not necessarily lie exactly on a hyperplane.

So we did some simulations to see whether MALCOM will completely fall apart when the assumptions are off a bit. In this case, we created randome smooth paths with energy up to 16 Hz, did a cubic transformation on the input signal, ran MALCOM, and then found the correlation between the MALCOM output and the filtered random signal.

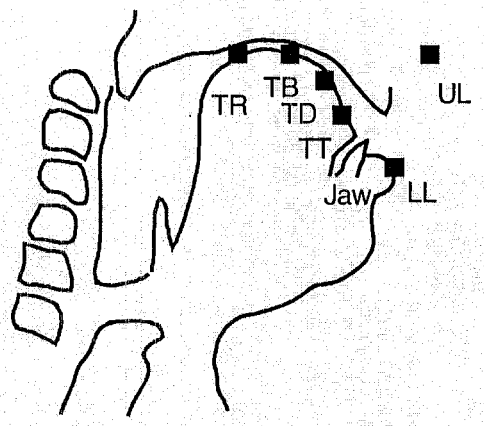
Actually, we tried several different cut-off freqeuncies for the random smooth paths, but always told MALCOM that the input signal only had energy up to 8 Hz.

1-D to 1-D Frequency Mismatch (SMALCOM told 8Hz.)

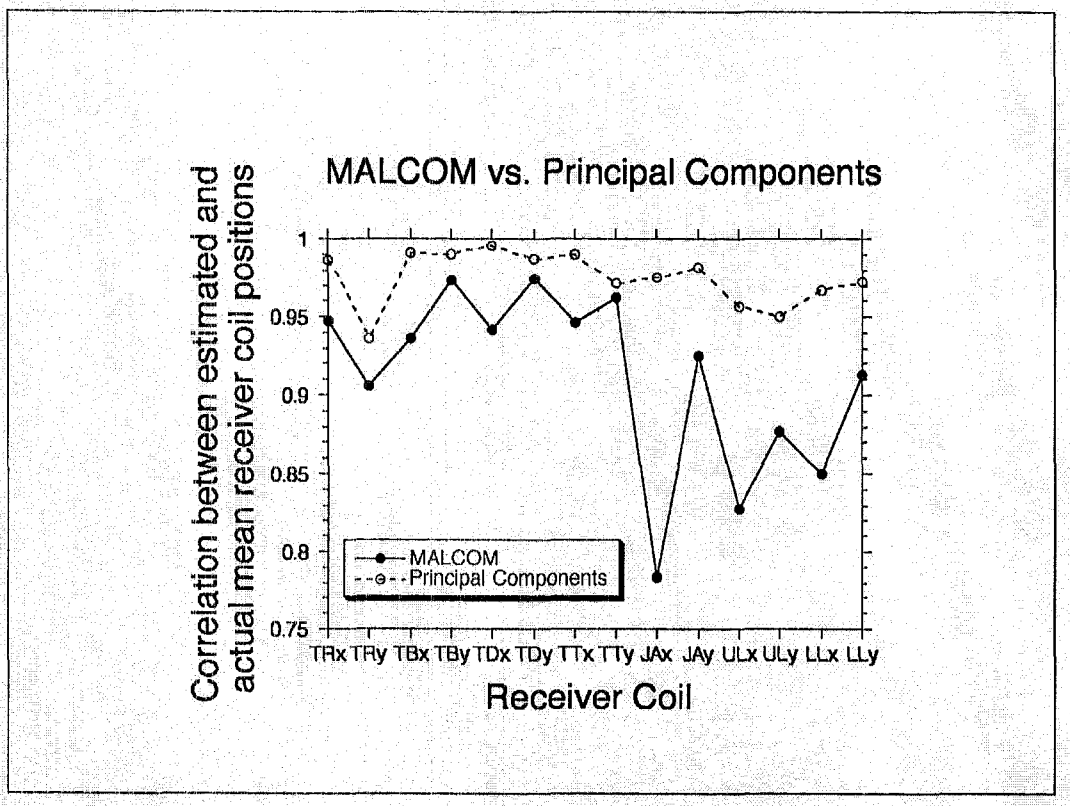


Again, the correlations were high. This plot shows how the asymptotic correlation changed with changing the cutoff frequency of the input signal. Noteice that the correlation axis starts at .9, so all the correlations are high, but they do drop off as the mismatch between reality and the MALCOM assumptions increases.

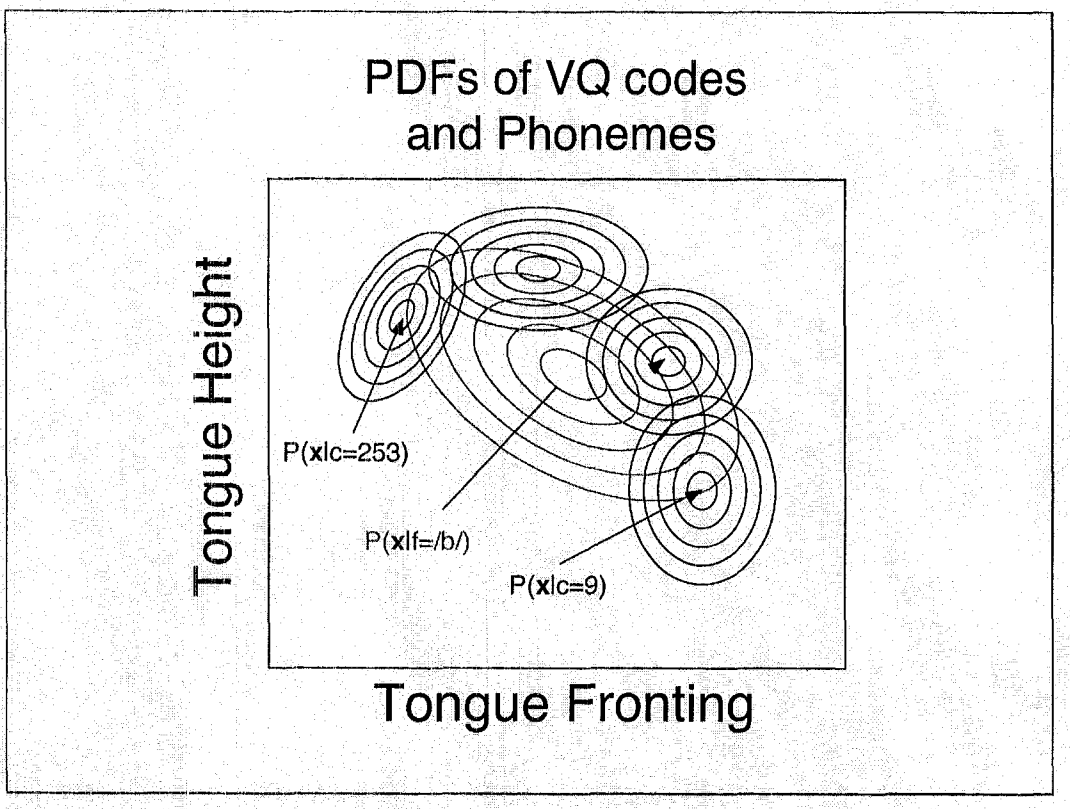
Articulator Data



We also did an experiment with measured articulator positions and recorded speech. In this case we had a speaker produce vowel to vowel transtions and we recorded the positions of pellets glued to his tongue, jaw, and lips. The pellets are labelled ...



We then ran MALCOM just on the recorded speech, but we found that the output of MALCOM was highly correlated with the articulator positions. This plot shows the results. Again note that the y axis starts at a correlation of 0.75.



So if we want to do speech recognition, we could start placing Gaussians over the MALCOM output and infer phonemes from that. For example, in addition to the gaussinas that give the probability of points in the recovered articulator space given VQ codes, we could have distributions that give the probability of points in the recovered articulator space given phonemes.

CO-MALCOM

	/p/	/b/	/m/	/i/	/a/	/u/
t=1	0.2	0.6	0.1	0.01	0.01	0.01
t=2	0.01	0.01	0.01	0.6	0.2	0.1
t=3	0.01	0.01	0.01	0.6	0.2	0.1
t=4	0.1	0.2	0.6	0.01	0.01	0.01

Then MALCOM could look at sequences of VQ codes, produce estimated articulator trajectories, and using those extra Gaussians, we could output something like this:

At each time we would have the probability of each phoneme. With this technique, the values in a row will always sum to 1, as a good probability should.

CO-MALCOM

	+voice	-voice	+nasal	-nasal
t=1	0.9	0.1	0.8	0.2
t=2	0.5	0.5	0.6	0.4
t=3	0.1	0.9	0.2	0.8

Alternately, we could try to recover articulator features, such as voicing or nasalization. This could give us a speed advantage.

CO-MALCOM

Conditional-Observable MALCOM

- Training: Maximize $P[\mathbf{f}|\mathbf{X}(\mathbf{c}, \varphi), \phi]$
- Recognition:
  - Find $\mathbf{X}(\mathbf{c}, \varphi)$
  - For each time, find $P[\mathbf{f}|\mathbf{X}(\mathbf{c}, \varphi), \phi]$

If we take this approach, however, we should try to optimize all the MALCOM parameters at the same time. We have a algorithm, called CO-MALCOM that let's us optimize both the paramters of the distribution that give the mapping between VQ codes and articulator positions and the paramters that give the relationship between phonemes and articulator position jointly, which should give better recognition.

HMM/MALCOM Hybrid

- Use currently available language models to get the probabilities of words given previous words.
- Use currently available word models to get the probabilities of phonemes given words.
- Use CO-MALCOM to get probabilities of phonemes given acoustics.
- Use slightly modified forward algorithm to combine the above probabilities for recognition.

We also have a way to use a slightly modified forward algorithm to combine CO-MALCOM with the trellis models that people use now

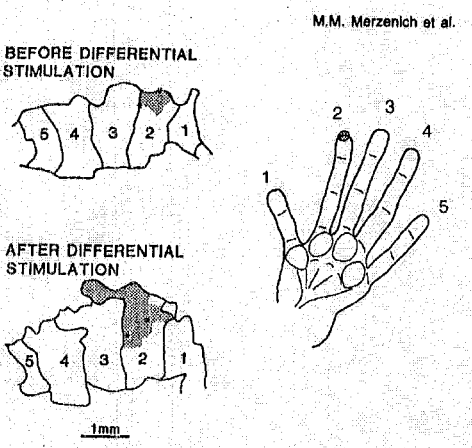
So doing MALCOM based recognition does not mean that we have to throw out the word and language models we use now. I often just think of MALCOM as an alternate acoustic model.

Conclusions

Articulator positions recovered from acoustics should:

- Be more robust to the noise environment
- Be easier to use with interpolation schemes
- Be invariant to microphone non-linearities
- Be more robust to speaker differences
- Allow better modeling of the conditional independence of speech signals
- Allow us to better build speech production knowledge into our recognition algorithms
- Not interfere with the usage of a standard trellis model

Somatosensory Homunculus



I'm going to take a couple minutes giving you a bit of background on the work I'll be presenting. When I was a graduate student I went to a talk about the work of Michael Merzenich. Merzenich was looking at the brain's somatosensory homunculus. It turns out that you can draw a picture of a monkeys hand on a monkeys brain so that when you poke the monkey's thumb, the thumb part of the brain will light up, and when you poke the pinky, the pinky part of the brain lights up.

So numbering the fingers 1-5, we find that they are represented by region 1-5 on the brain, where those regions are positioned like in this figure. Merzenich took a nerve from the monkey's pinky and put it in the monkeys thumb. Of course, at first when the nerve was touched, the pinky part of the brain lit up, but a month later the brain reorganized itself so when you touched the nerve, the thumb part of the brain lit up.

The brain was able to do that reorganization because we don't touch one nerve at a time, we tend to touch lots of adjacent nerves at the same time. All the brain needs to do is represent nerves by nearby locations locations if they tend to be active at the same time.

Similarly, since we know that sounds produced sufficiently close in time must have been produced by similar articulator configuratons, if we represent sounds by nearby location if they tend to occur close in time, we can get back information about articulator positions from sounds.