Underlying representations for speech modelling

Simon King
Centre for Speech Technology Research
University of Edinburgh

- acoustic signal is the observed output of some **underlying** process
- the true underlying process is articulation
- can we use this fact when modelling/synthesising/recognising speech?

CUED 25/02/2003

Some current CSTR projects

- speech recognition using phonological or pseudo-articulatory features
  with Mirjam Wester

- speech recognition using linear dynamical models
  with Joe Frankel and Fiona Kenney

- **join cost and smoothing for concatenative synthesis**
  with Jithendra Vepa

- articulatory-controllable speech modification
  with Yoshi Shiga

- acoustic-to-articulatory inversion
  Korin Richmond

Hidden vs. underlying articulation

- all speech production involves articulation(!)
  - sometimes it is known (it has been observed and measured)
  - usually it is not known
- can we use information gathered from observed articulation to
  - make better models / do better signal processing
  - even for speech with unobserved articulation

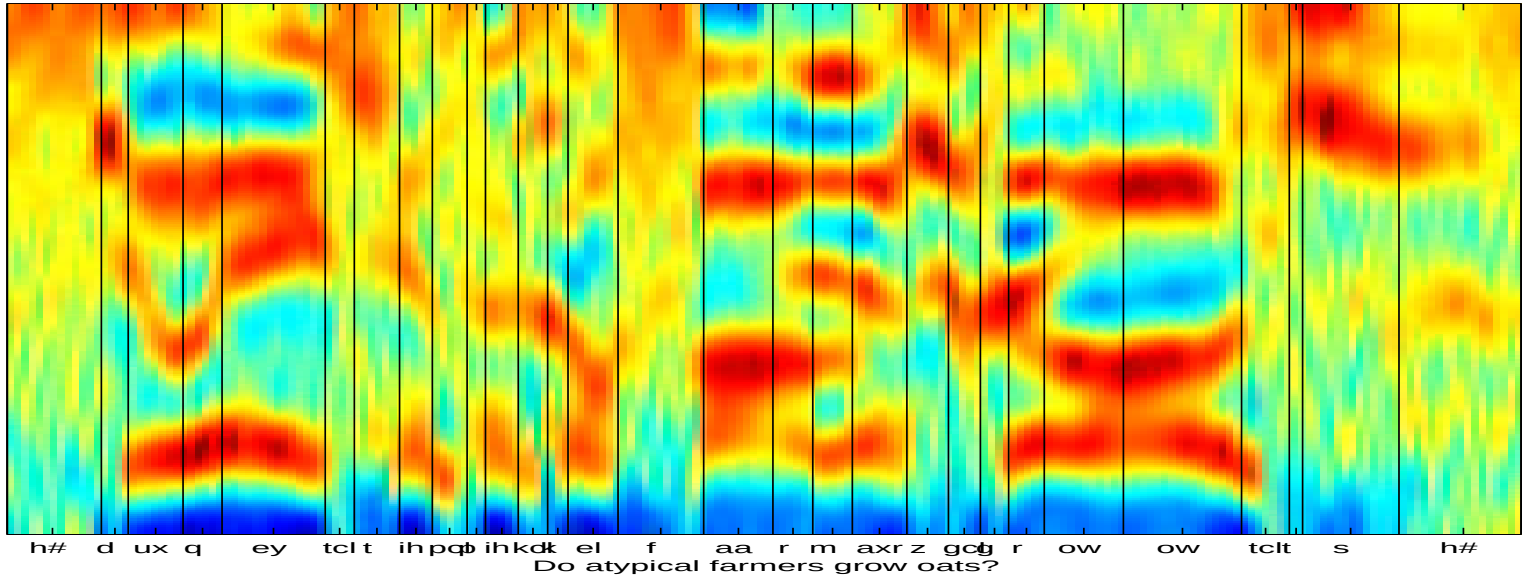
Speech production motivation

Articulator movements have particular characteristics

- continuous, smooth trajectories
- asynchronous (loosely coupled)
- limited velocities and accelerations
- almost constantly moving

Acoustics are manifestation of this system

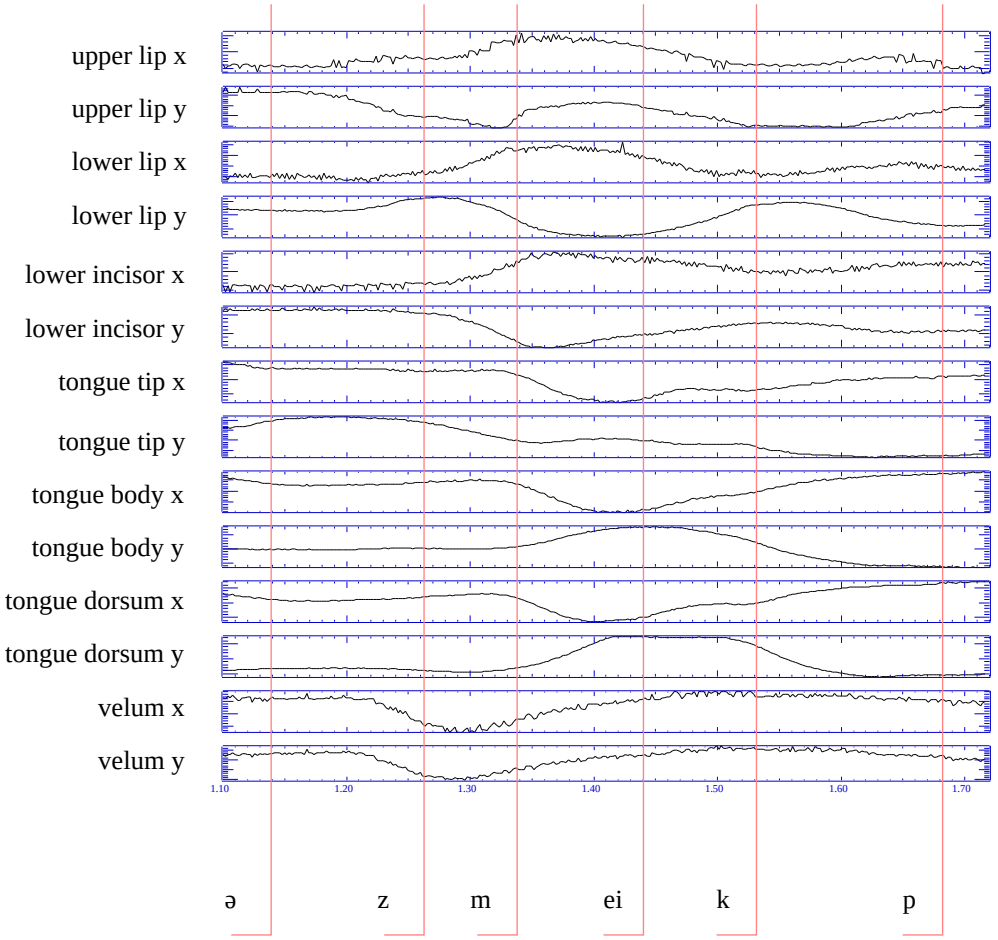
- piecewise continuous in spectral domain



Measuring articulation

- can measure movement of the articulators during speech production
- Electromagnetic articulography (EMA)
- QMUC are building a corpus of parallel speech signal and articulatory measurement data (MOCHA - uses TIMIT sentences)

The true picture of speech production is revealed by articulatory data.



A toy example

Small inventory:

- two tokens of “never”, one token of “down”

Limited domain:

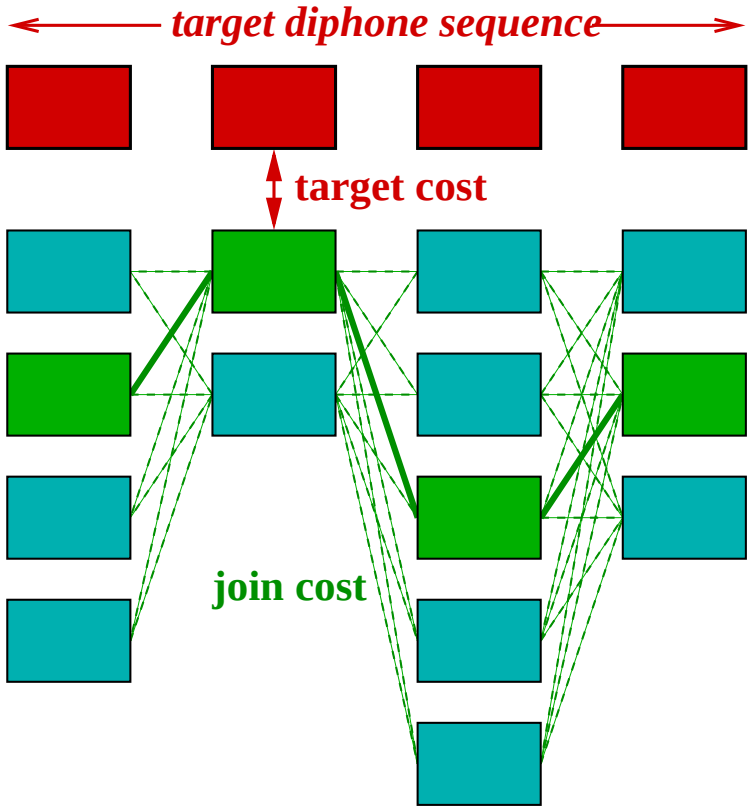
- we can only say one thing: “never down”

Unit selection: need to choose which of the “never” tokens to use

- using conventional acoustic measure to select units
- compare with an articulatory measure

Join cost

What we need is a good **join cost** for concatenative synthesis



Back to the basics: articulation

- two tokens of the word “never”
  1. from the context “... never compile ...”
  2. from the context “... never changes ...”

Articulator movement during each token of this word will be

- *roughly* the same, otherwise it wouldn’t sound like “never”

but will still vary, within certain limits

Similarities

We expect basic similarities in articulation

- tongue tip raised and velum lowered for [n]
- velum raised, tongue body and tip lowered for [ɛ]
- upper incisor meets lower lip for [v]
- constriction released for [ə]

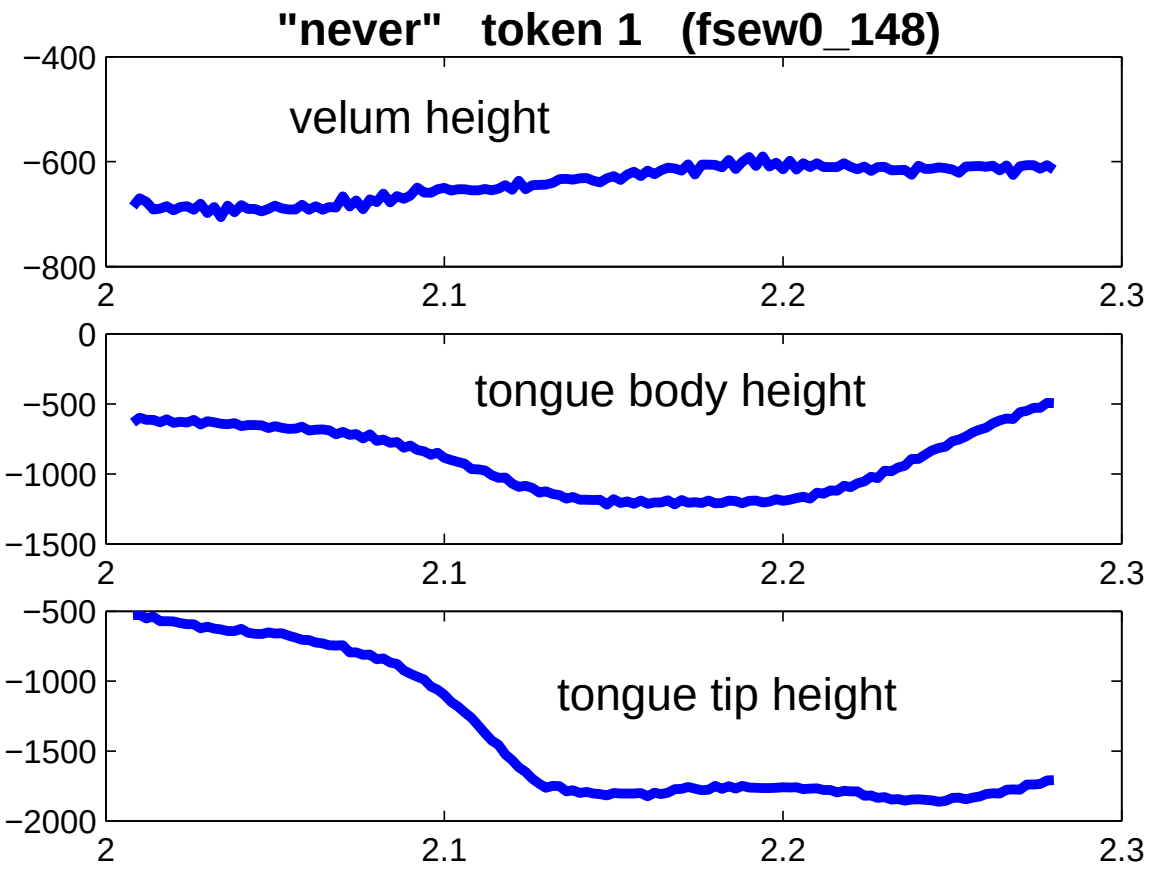
Differences

Differences largely due to context.

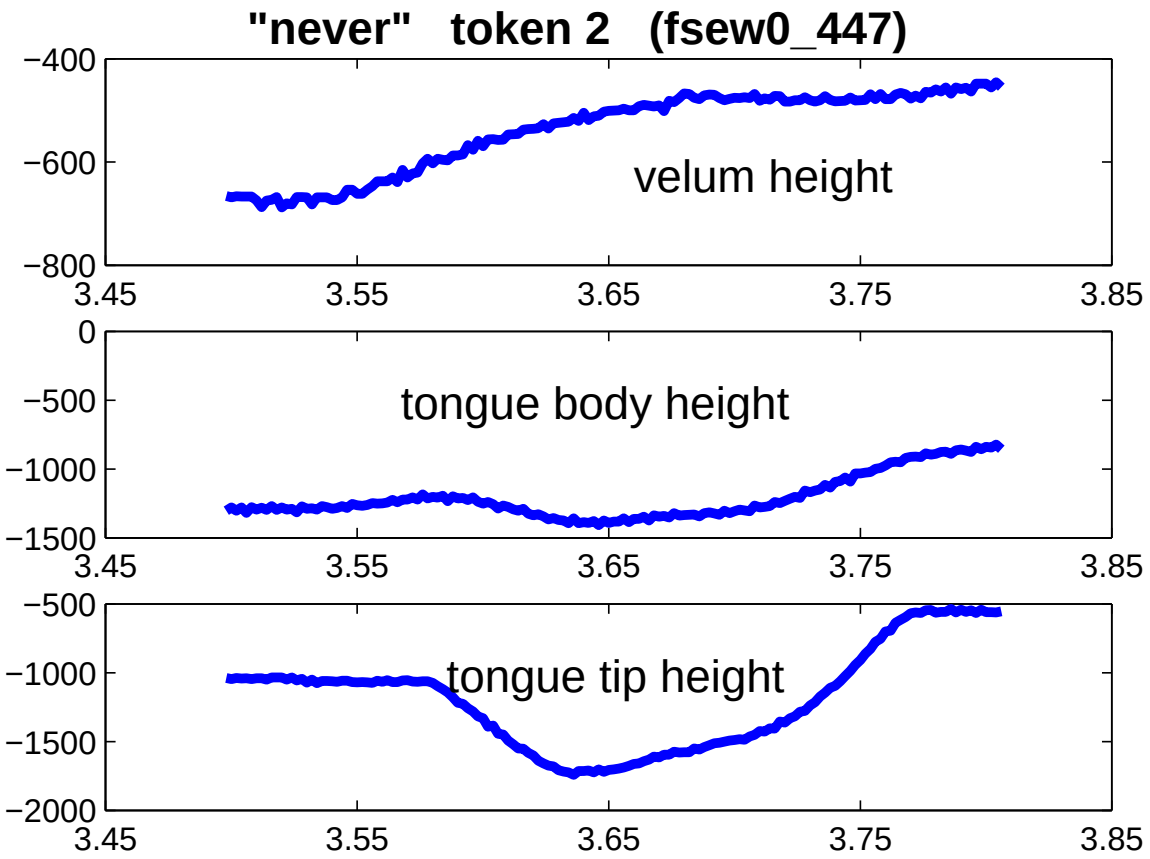
Consider only right-context

- Token 1 is from the context “... never compile ...”
  towards end of [nɛvə] we expect
  - tongue tip lowering for upcoming [k]
  - tongue body/dorsum raised & moved back to make closure for [k]
- Token 2 is from the context “... never changes ...”
  - tongue tip/body raised to make closure for [ʧ]

Here's some of the articulation for "never", token 1



And for “never”, token 2



Concatenation

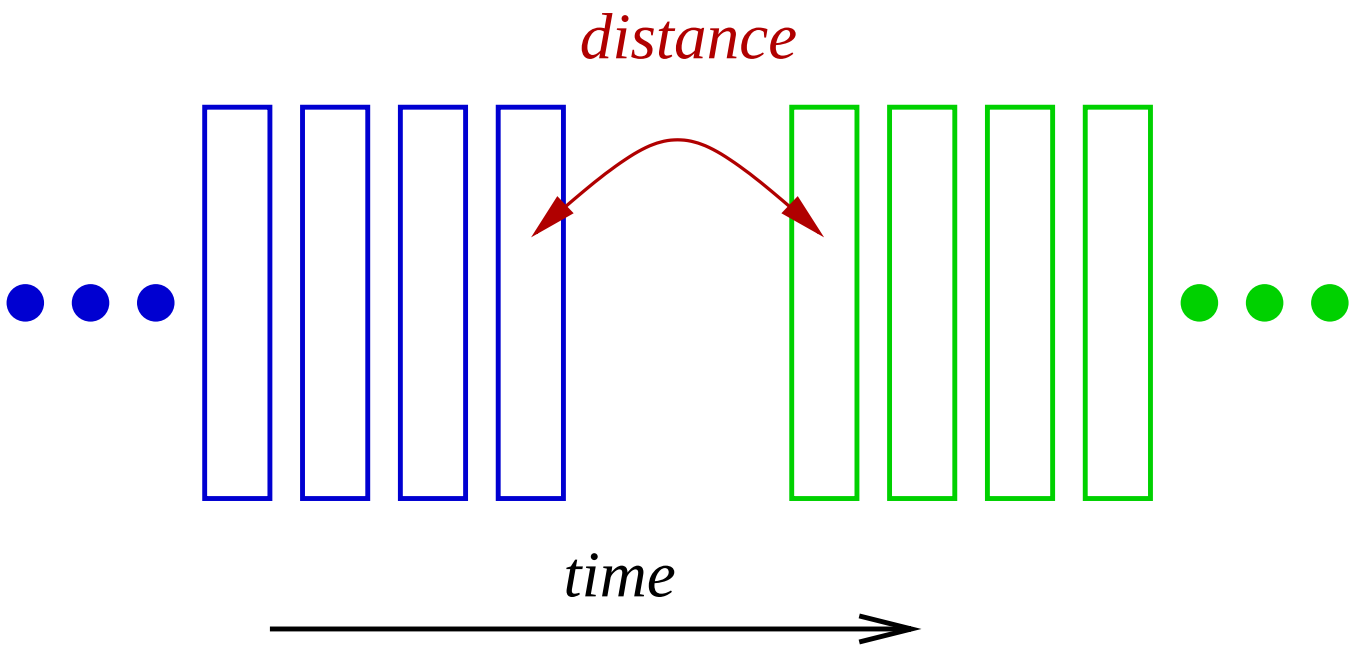
If we want to make “... never down ...”

which token of “never” should we choose?

- could use an acoustic measure
  - compare spectra at end of each token of “never” with start of “down”
  - choose most similar
- could use a context-based measure
  - compare contexts of each token:
    “... never **compile** ...” vs. “... never **changes** ...”
  - choose context most similar to “**down**”

Acoustic distance

e.g. Euclidean distance between Mel-scale cepstral coefficients



Results of acoustic join cost

(large cost = acoustically dissimilar, implies bad match)

left unit	right unit	acoustic distance
“never”, token 1	“down”	4.02
“never”, token 2		7.24

$\longrightarrow$ acoustic cost says to use token 1 of “never”

Context measure

Which of [k] or [ʧ] will have the most similar contextual effect on the preceding segment to [d] ?

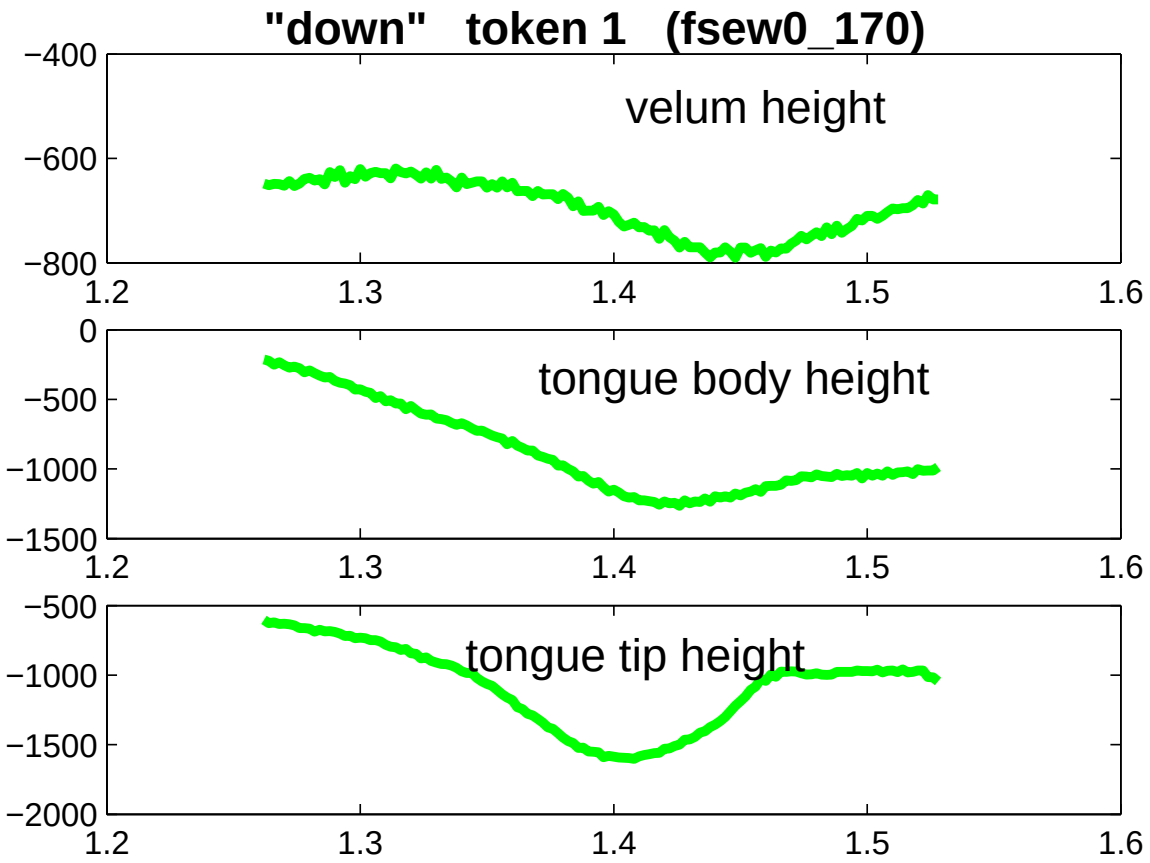
- place of articulation
  - [ʧ] most similar to [d]
- manner of articulation
  - [k] most similar to [d]

$\longrightarrow$ features of context are hard to formulate as a cost

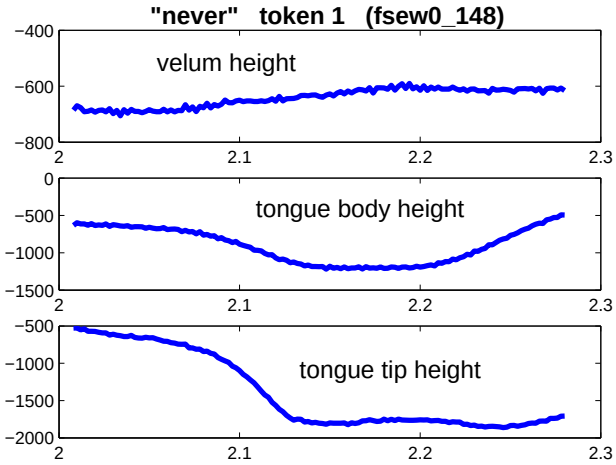
Articulatory measure

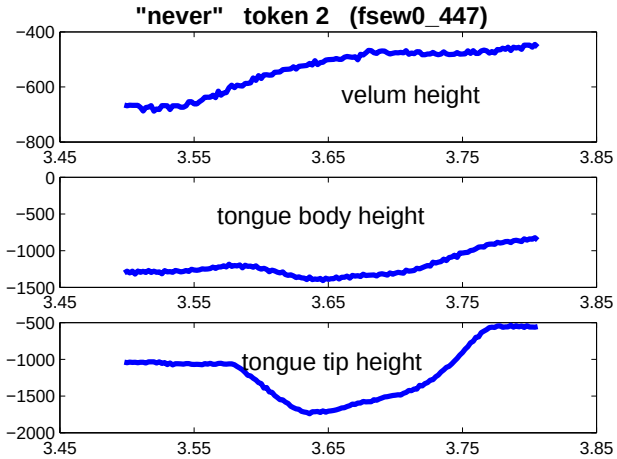
A more sophisticated “place of articulation” that allows easier formulation into a **join cost**

Our only token of “down”

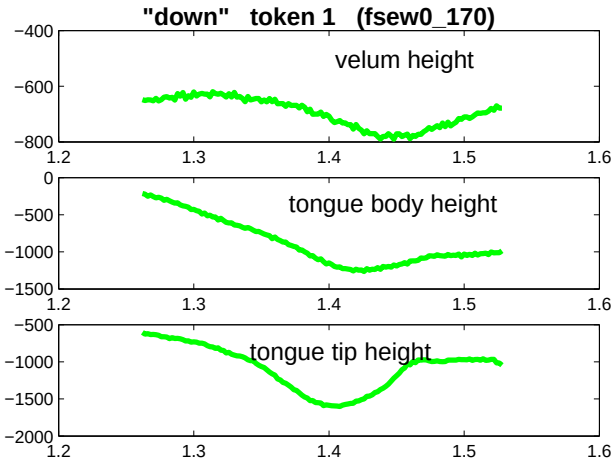


Concatenation

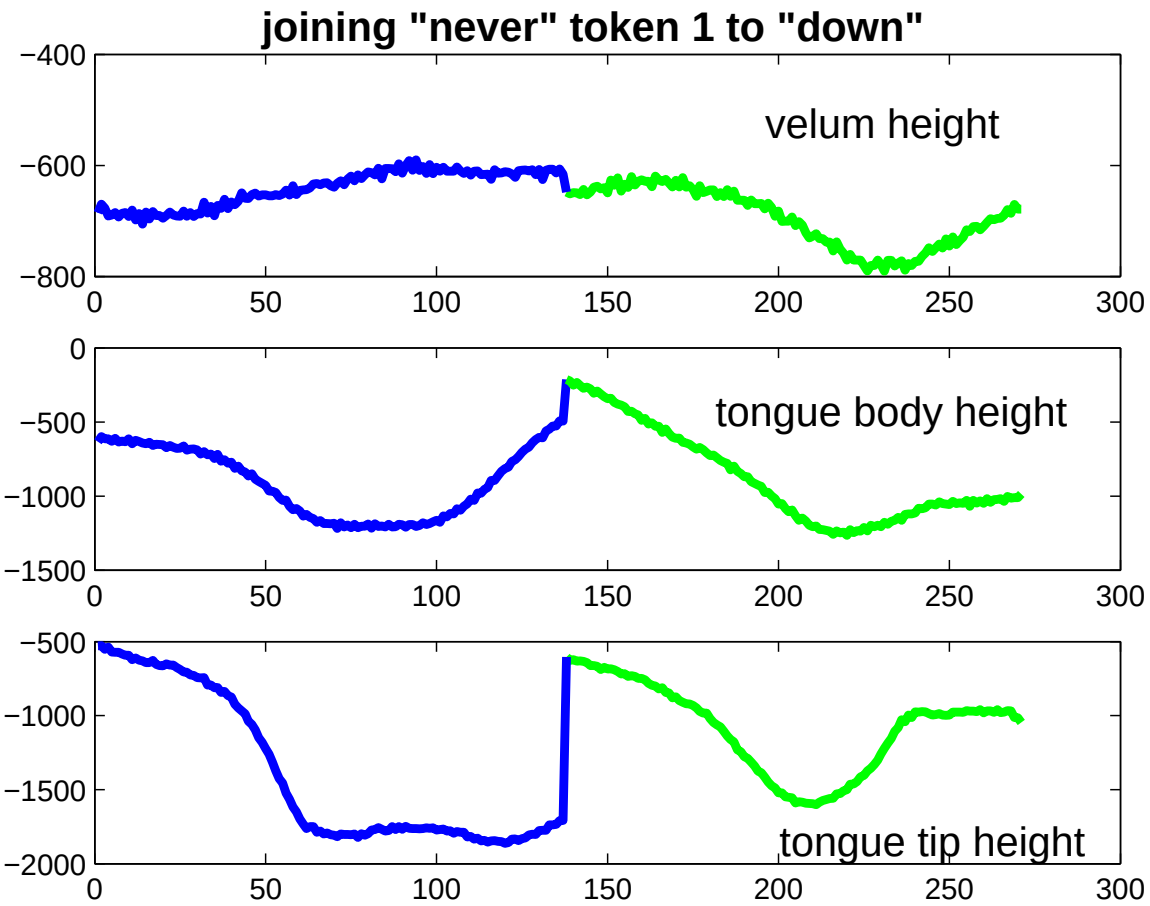




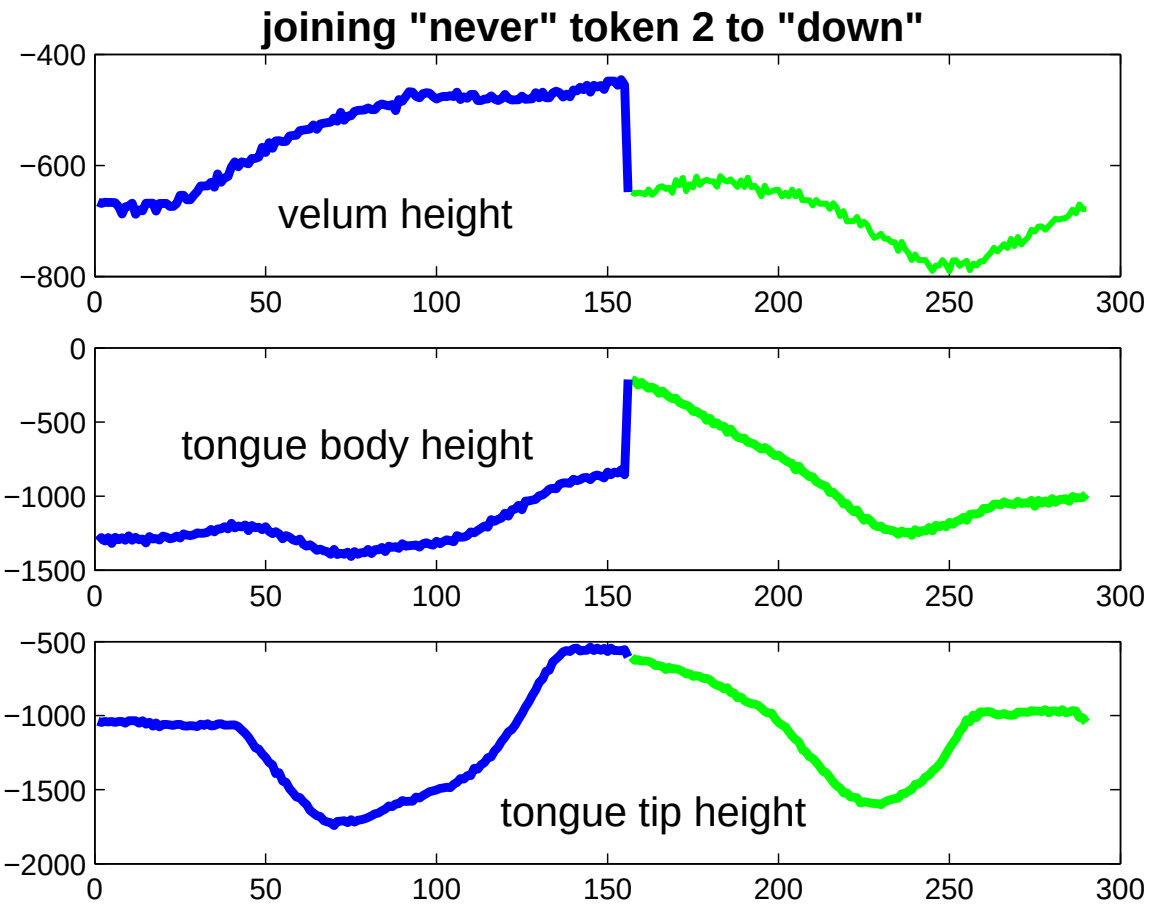
+



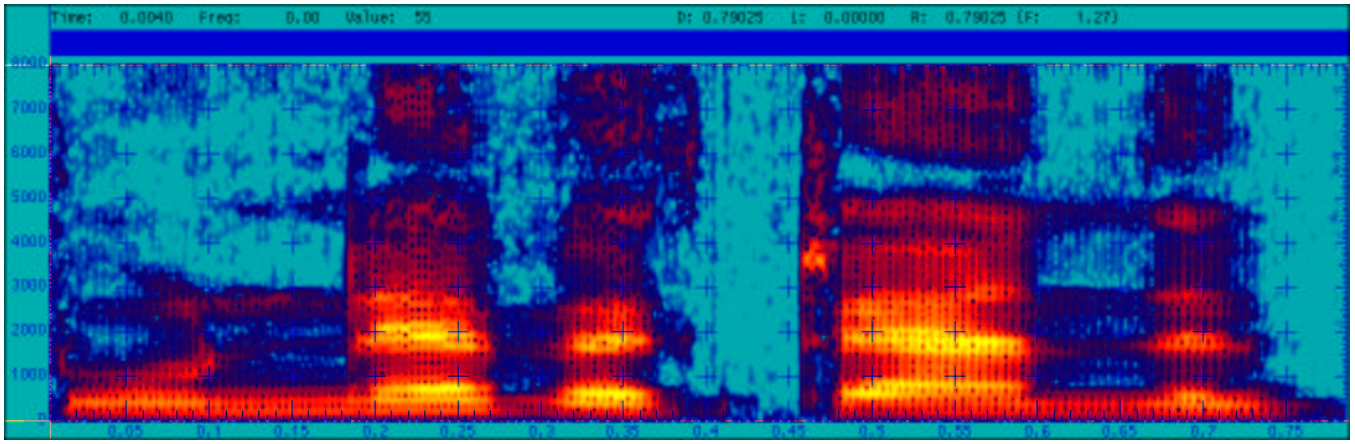
Joining token 1 of “never” with “down”



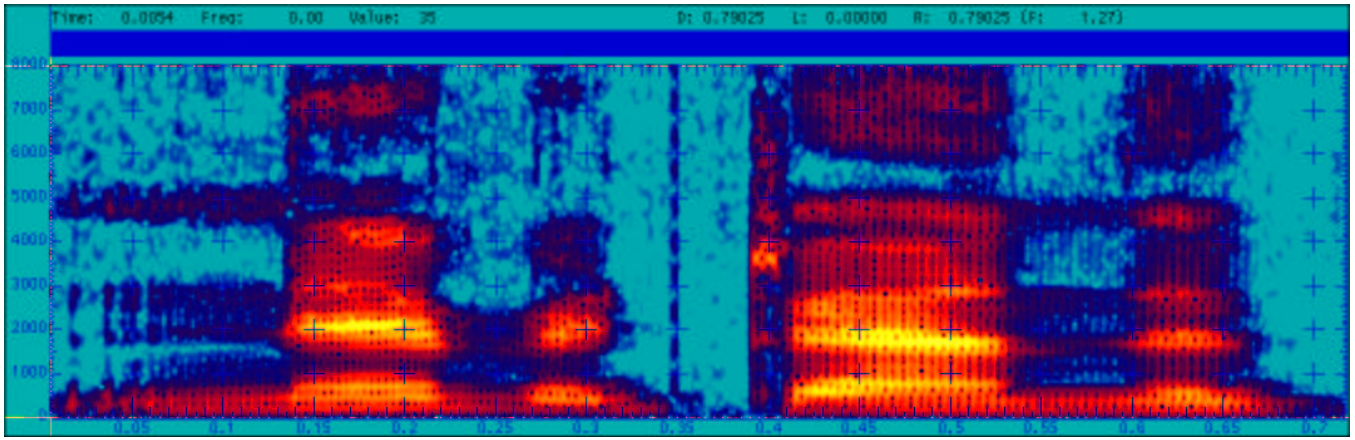
Joining token 2 of “never” with “down”



1.


Time: 0.0040 Freq: 0.00 Value: 55
D: 0.79025 L: 0.00000 R: 0.79025 (F: 1.27)

2.


Time: 0.0054 Freq: 0.00 Value: 35
D: 0.79025 L: 0.00000 R: 0.79025 (F: 1.27)

What if we don't know the real articulation?

- infer it from the acoustics (*Korin Richmond*)
  - almost certainly needs speaker-specific parallel acoustic+articulatory training data
- or infer something that has similar properties
  - what are the important properties?
  - what kind of model has those properties?

Properties of articulation

Those that seem important to consider when concatenating speech

- continuous
- smooth
  - a consequence of particular dynamic characteristics
  - e.g. limited range, velocity and acceleration

We need a model with those properties

Linear Dynamic Model (LDM)

$$\mathbf{y}_t = H\mathbf{x}_t + \varepsilon_t$$

$$\mathbf{x}_t = F\mathbf{x}_{t-1} + \eta_t$$

$$with\ \varepsilon_t \sim N(\mathbf{v},C)\ and\ \eta_t \sim N(\mathbf{w},D)$$

- Linear segment model with Markovian structure
- Continuous state which encapsulates dynamics

Well known in the field of control systems and optimal estimation as the **Kalman filter**

Why use a subspace?

- modelling directly in the observation space
  (e.g. a spectral parameterisation of the speech signal)
  - fixes the dimensionality
  - therefore fixes the complexity of the motion
- modelling in a subspace
  - allows any dimensionality
  - can choose complexity of the motion to suit the data

Properties of the state space

$$\mathbf{y}_t = H\mathbf{x}_t + \varepsilon_t$$

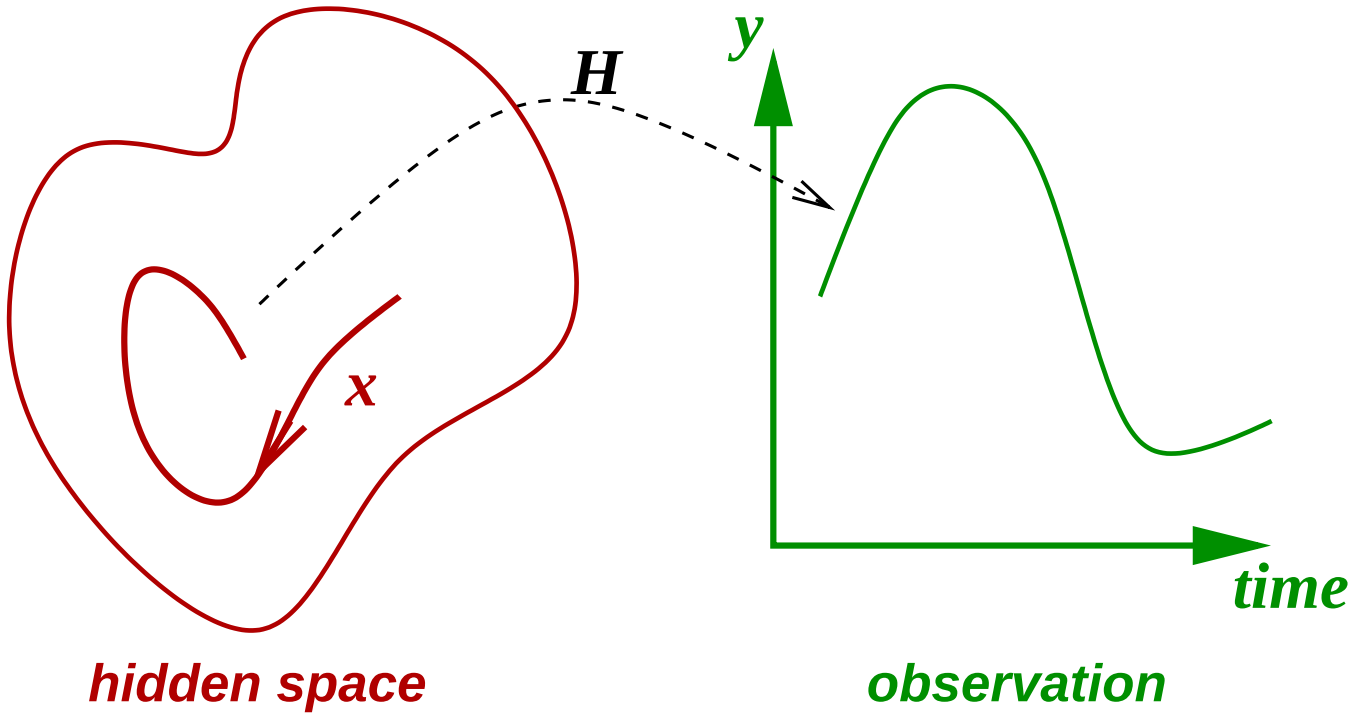
$$\mathbf{x}_t = F\mathbf{x}_{t-1} + \eta_t$$

$$with\ \varepsilon_t \sim N(\mathbf{v},C)\ and\ \eta_t \sim N(\mathbf{w},D)$$

- dynamics expressed in the state space
- state dimension dictates complexity of trajectories
- continuous within segments and can also be over boundaries

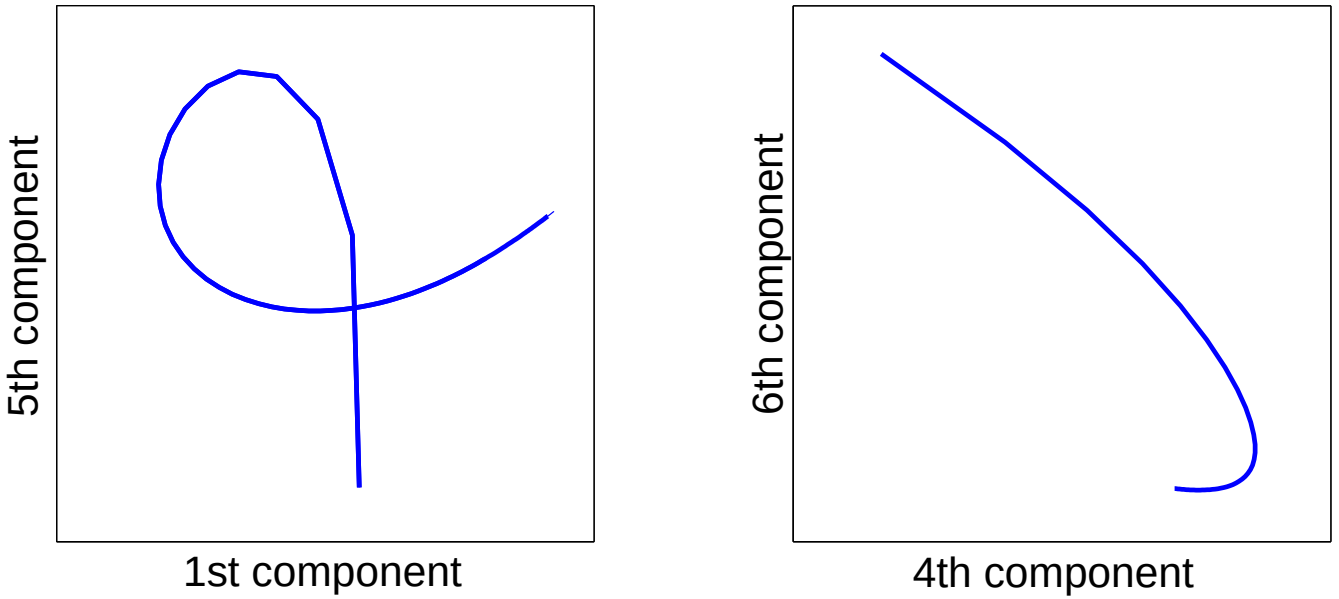
Subspace modelling

The model operates in an underlying **subspace** in which it makes smooth, continuous motion (a trajectory)



Actual state trajectory

2−d slices through an 8−d space showing mean ’ah’ state trajectory



Properties of the state-observation mapping

$$\mathbf{y}_t = H\mathbf{x}_t + \varepsilon_t$$

$$\mathbf{x}_t = F\mathbf{x}_{t-1} + \eta_t$$

$$with\ \varepsilon_t \sim N(\mathbf{v},C)\ and\ \eta_t \sim N(\mathbf{w},D)$$

- Observations are weighted sum of state values
- Noise accounts for confidence on each stream of data

Training

- Expectation-Maximisation algorithm

Current system is very simple

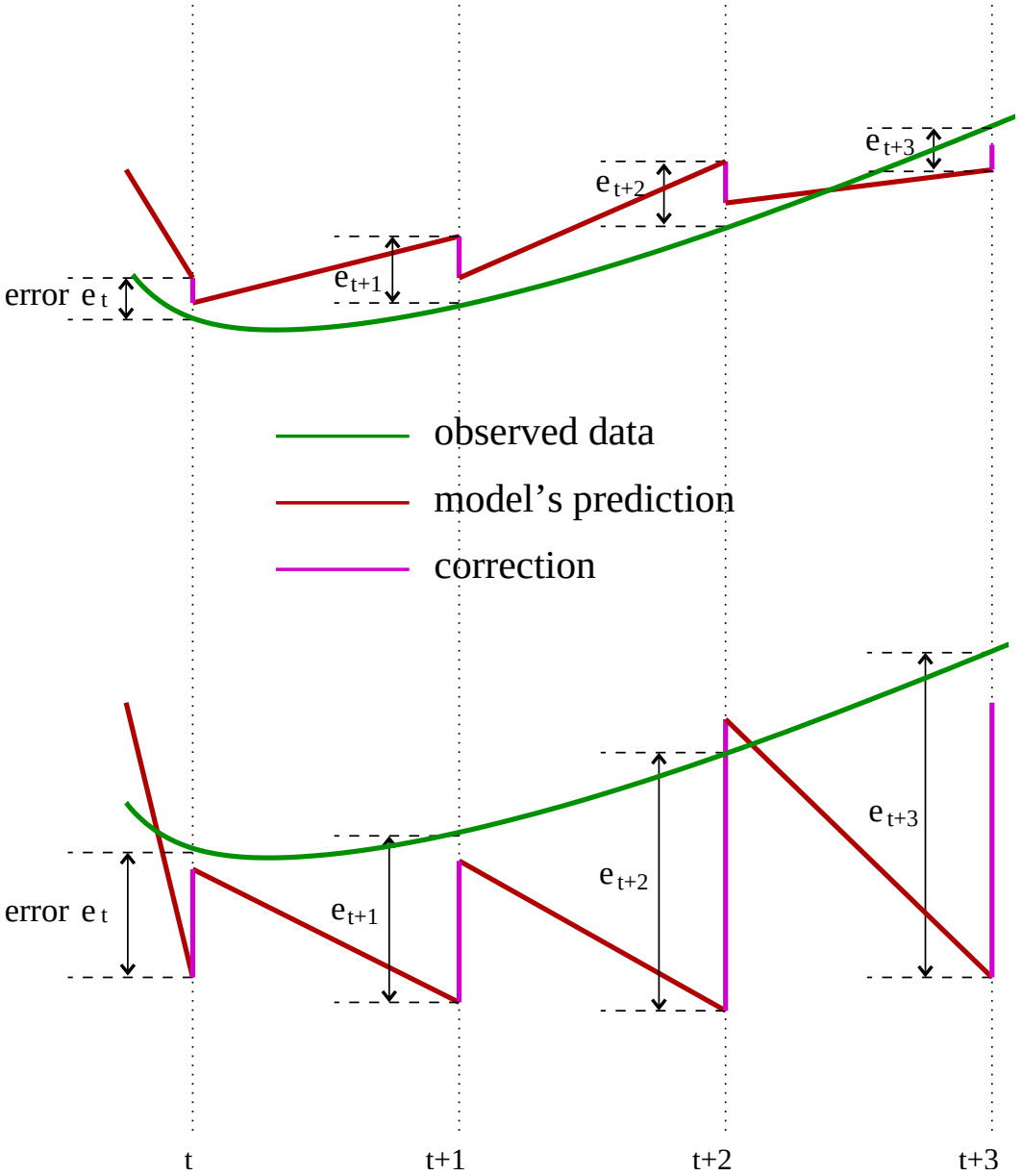
- context-independent phone models
- training data labelled at phonetic (i.e. model) level
- “embedded” training from sentence-level labelled data coming soon

Things we can do with LDMs

- speech recognition
- **join cost** calculation
- optimal parameter **smoothing** for disguising concatentation discontinuities

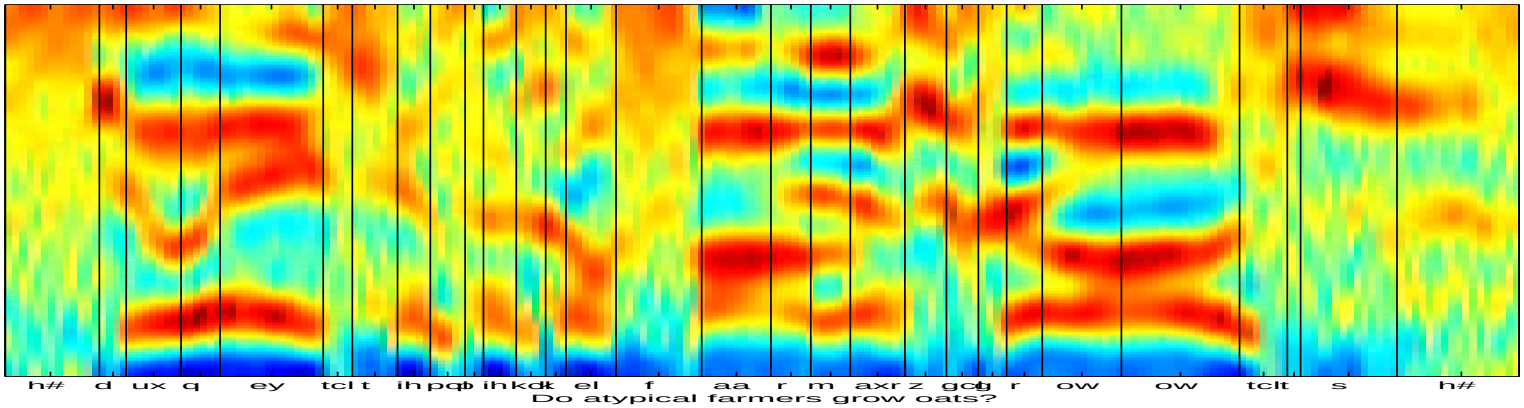
MAP evaluation

Sum errors $\mathbf{e}_t$ (normalised by covariance) to compute likelihood that model generated $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$.

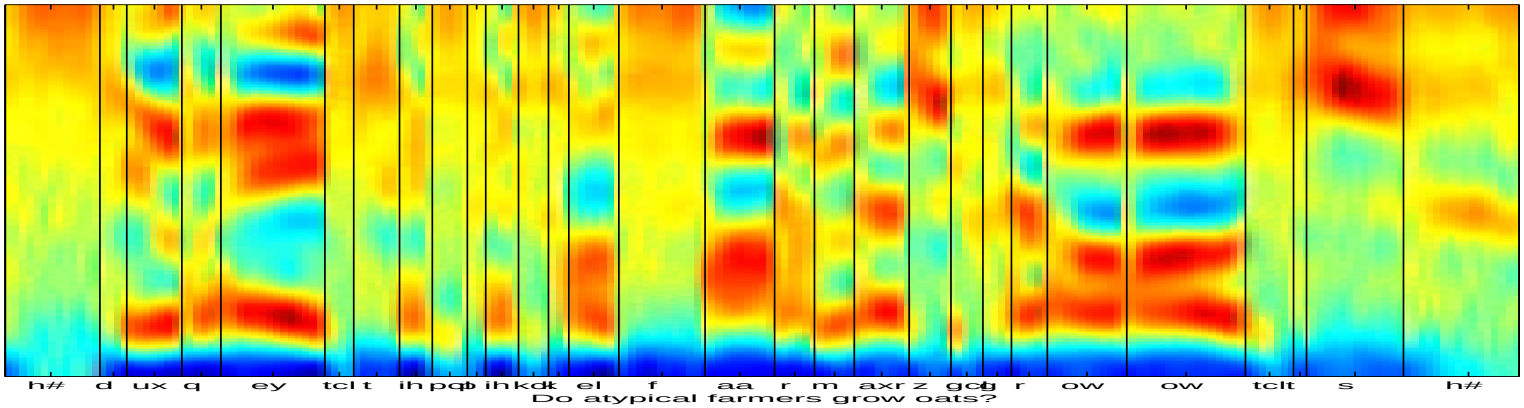

error e_t
e_{t+1}
e_{t+2}
e_{t+3}
observed data
model's prediction
correction
t
t+1
t+2
t+3

Filtering in action

Spectrogram of observations $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$



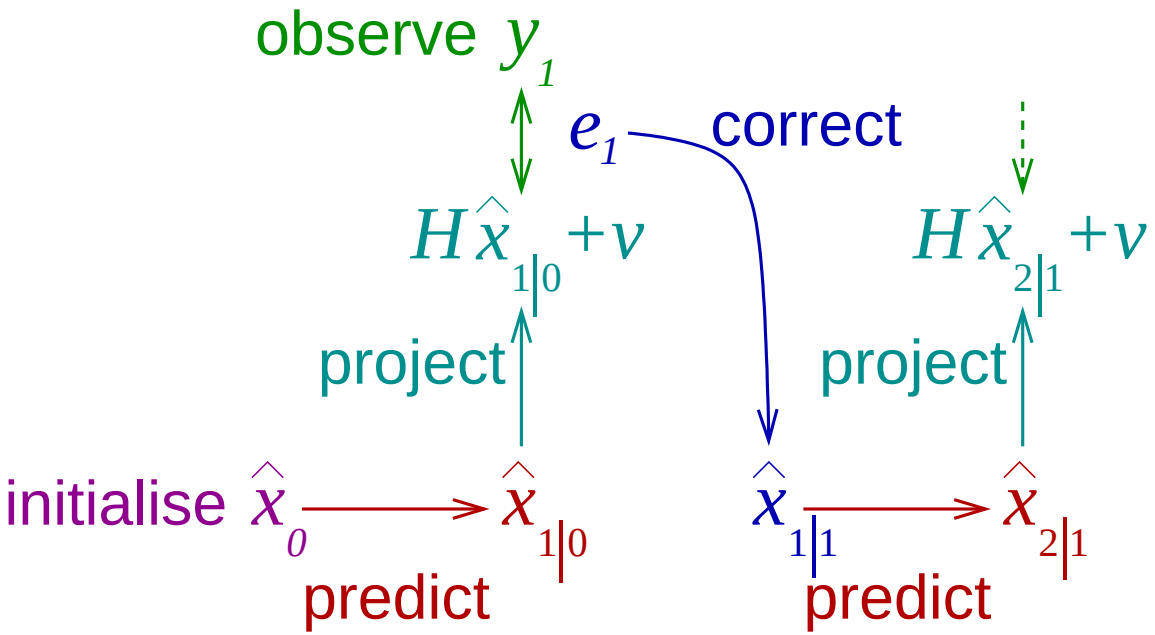
Spectrogram generated from model predictions



LDM-based join cost

- instead of measuring distance in acoustic domain, use a subspace model (the LDM)
- in fact, use the LDM as a probabilistic generative model
- use prediction error as a join cost

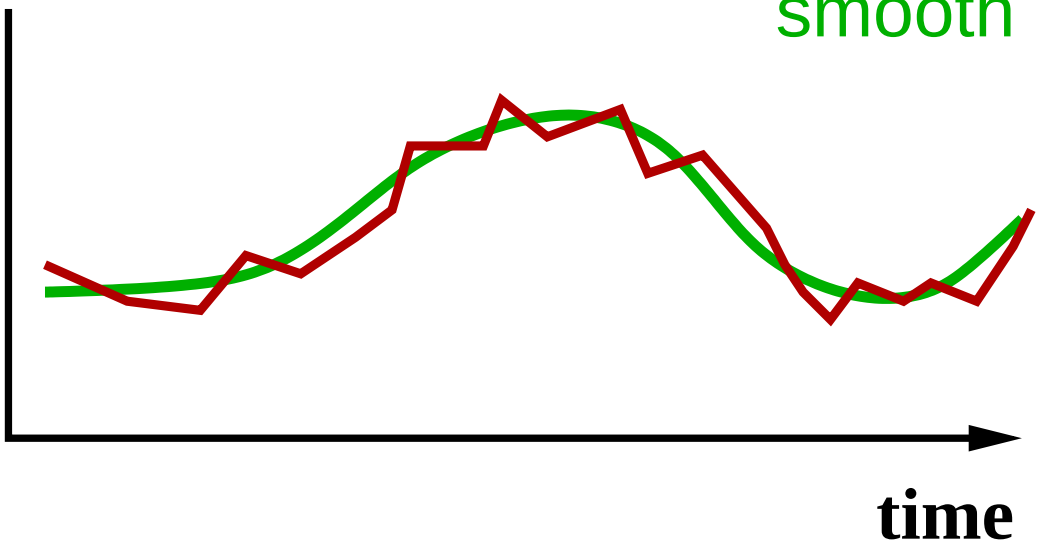
LDM as generative model



LDM as generative model - natural speech

noisy observation

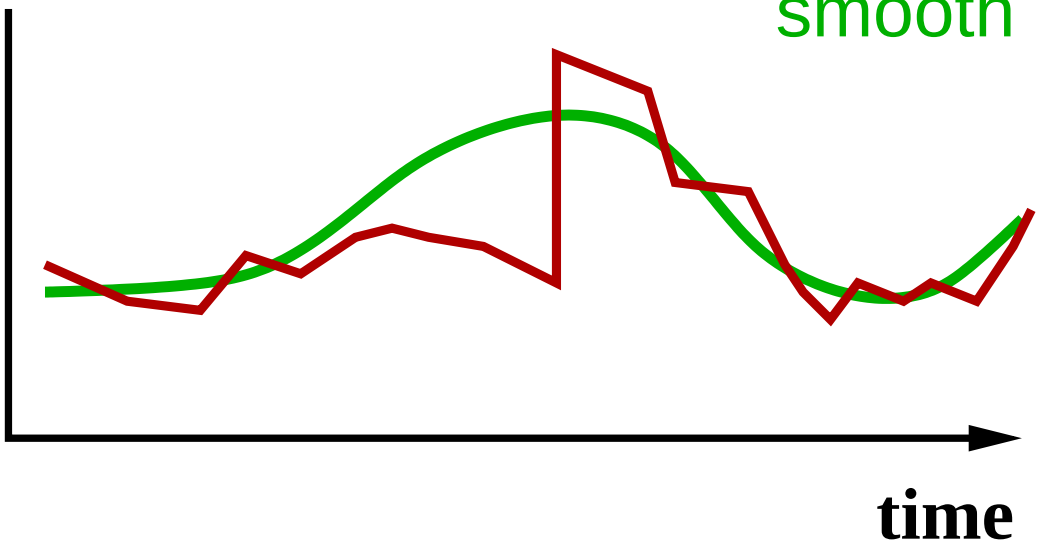
smooth "true" trajectory



LDM as generative model - joined speech

noisy observation

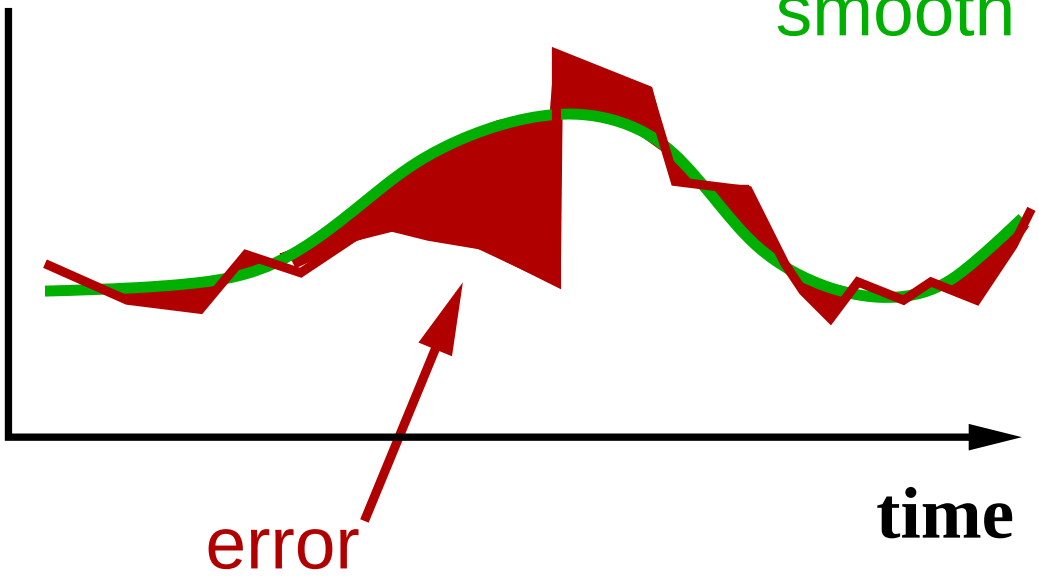
smooth "true" trajectory



LDM measuring error

noisy observation

smooth "true" trajectory



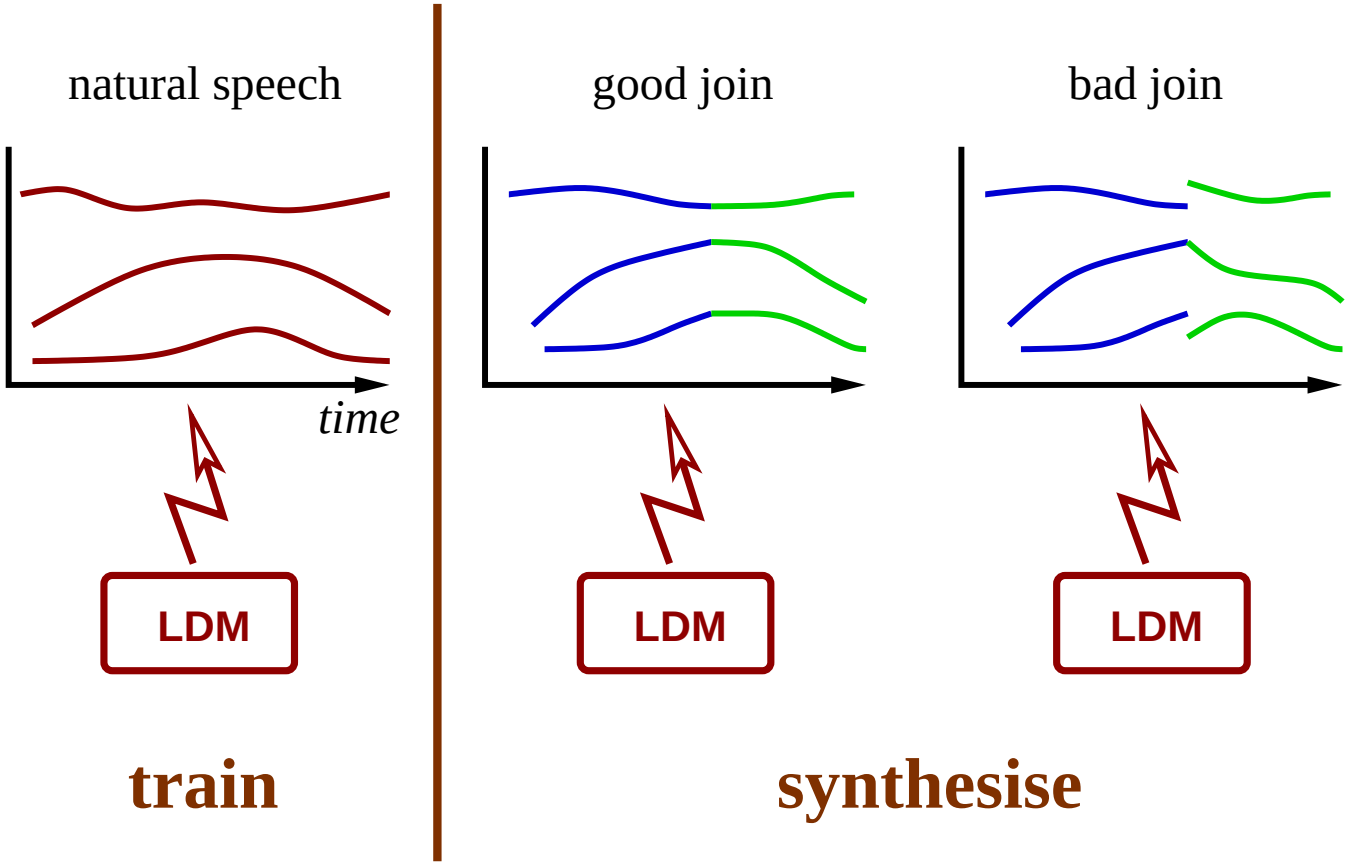
LDM computing join cost

Use the error between

- actual observations (e.g. LSFs)
- model's smooth predicted trajectory

Because model's predicted trajectory will hopefully be like that of natural speech

LDM join cost



Designing a join cost

How can we know whether our join cost function is any good?

A good join cost should

- correlate well with perceived join discontinuity

and to measure that, we need

- perceptual data

Perceptual data

Need to

- synthesise various stimuli containing both good and bad joins
- gather data on perceptual prominence of join

then

- examine correlations between objective join costs and perceptual ratings

Perceptual experiment

- five American English diphthongs, each in two carrier sentences
  - these are known to be hard segments to make joins in

diphthong	*sentences*
ey	More **plac**es are in the pipeline. The government sought autho**riz**ation of his citizenship.
ow	European shares resist **glob**al fallout. The speech sym**pos**ium might begin on Monday.
ay	This is **high**ly significant. Primitive **trib**es have an upbeat attitude.
aw	A large **hous**ehold needs lots of appliances. Every picture is worth a **thous**and words.
oy	The **boy** went to play tennis. Never exp**loit** the lives of the needy.

Perceptual experiment

- synthesise many versions of each (using *rVoice*)
- manually select a subset of about 30 per sentence with a range of join qualities
- present to subjects
  - who rate them on 1–5 scale

[see recent papers in on CSTR web site for details]

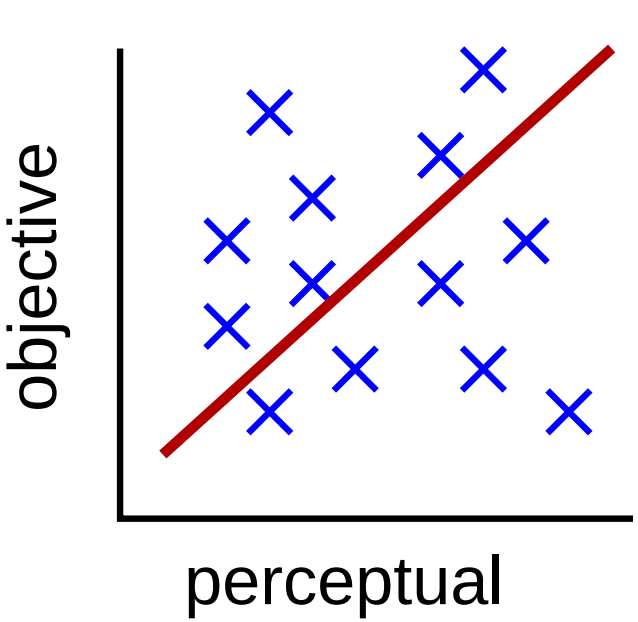
Data (i.e. results) from perceptual experiment

After discarding unreliable subjects

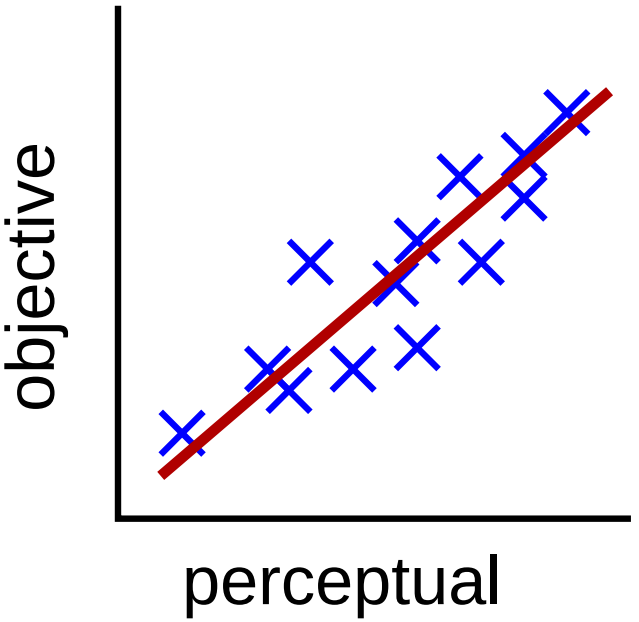
- for each of about 30 examples of each diphthong in each sentence
  - average perceptual rating for join cost, on 1–5 scale

We can now correlate these ratings with objective measures based on acoustic signal

Using this data to evaluate objective join costs



poor correlation



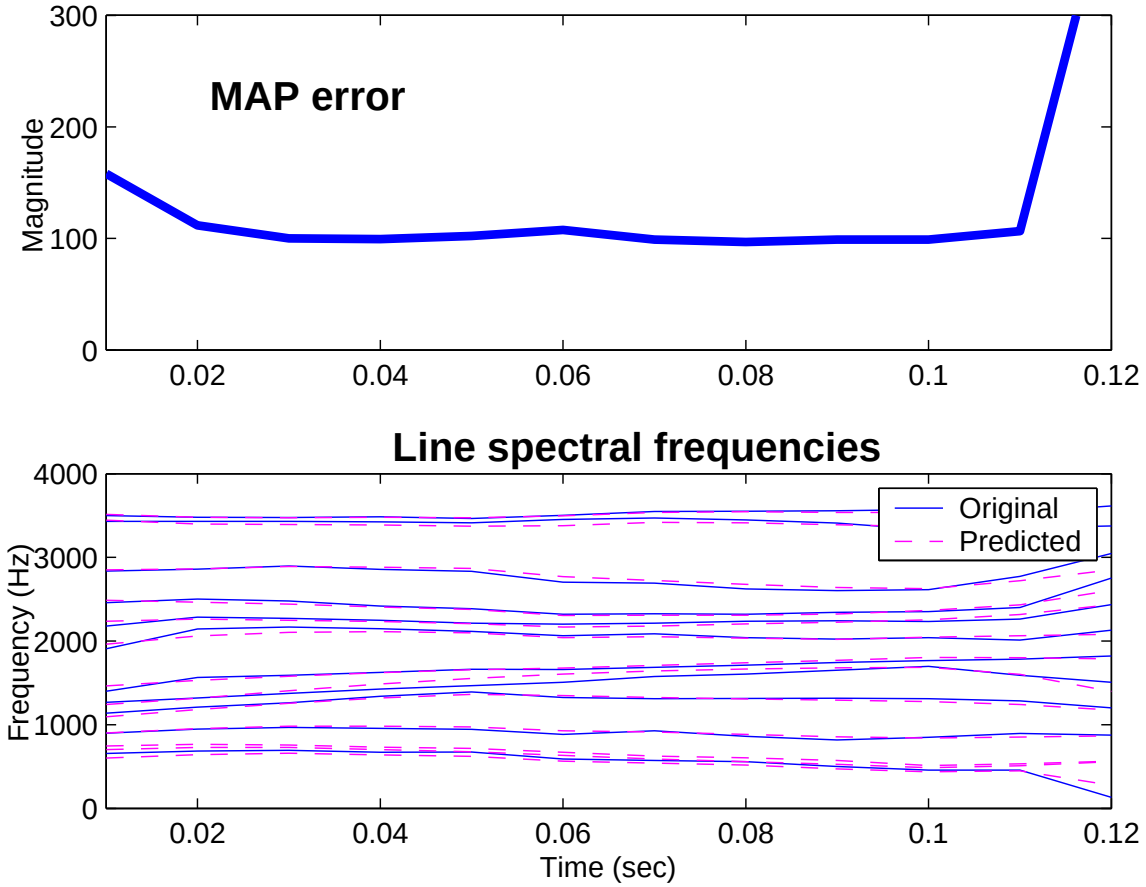
good correlation

Latest work

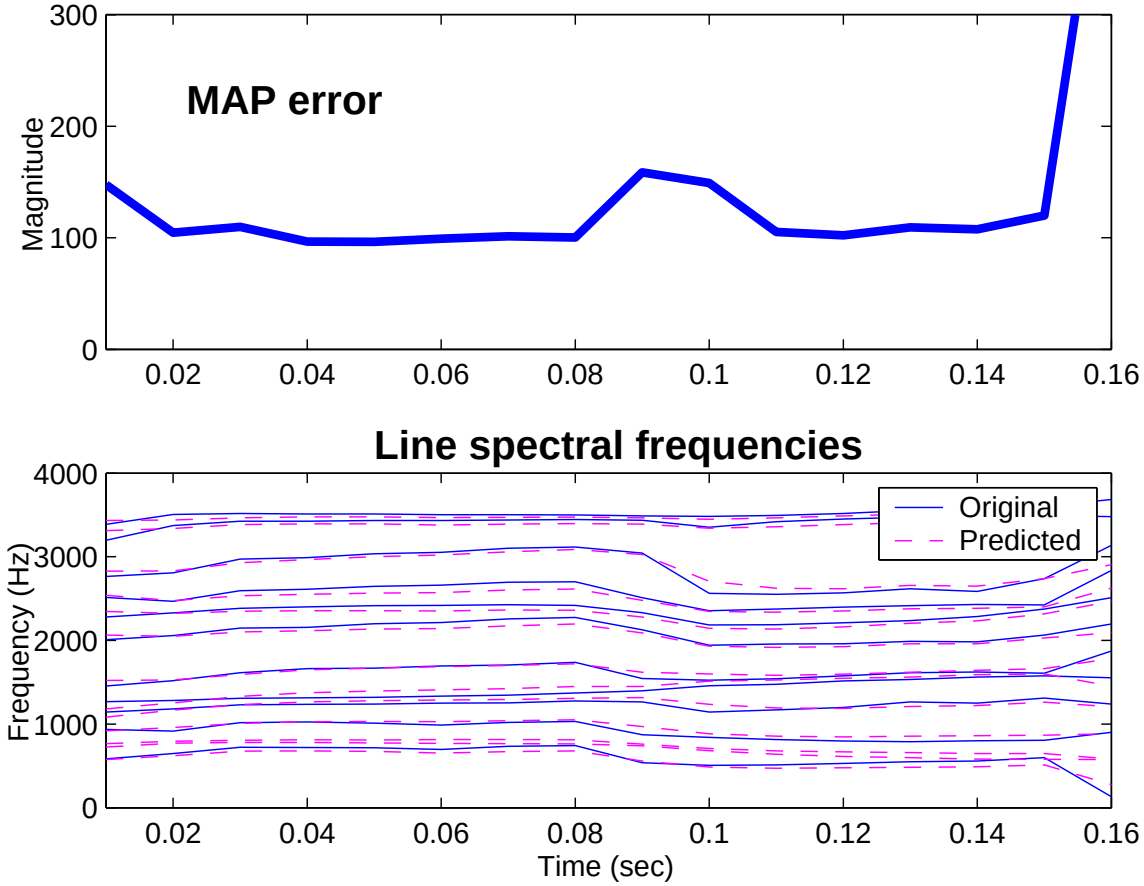
In *rVoice* joins are always made mid-phone (units are diphones)

- train one LDM per phone class
- for a candidate join between two units from the inventory
  - “run” the LDM along the phone being joined
  - use it to estimate the error of the observed data
  - if the joined phone is similar to a natural one, the error will be low

LDM join cost - good join



LDM join cost - bad join



Latest results

Correlations between LDM join cost and perceptual data

	MFCC	LSF	MCA	MCA wgts.	LDM MAP
ey	0.21	0.37	0.36	**0.44**	**0.58**
	0.66	**0.58**	**0.46**	**0.60**	0.17
ow	0.31	0.21	0.19	0.19	0.26
	0.56	0.40	**0.46**	**0.52**	0.34
ay	0.39	0.01	0.03	-0.02	**0.56**
	0.66	**0.61**	**0.45**	**0.49**	**0.59**
aw	0.34	**0.66**	0.35	**0.49**	-0.02
	0.77	**0.78**	**0.57**	**0.62**	**0.50**
oy	0.17	0.20	**0.53**	**0.55**	**0.45**
	-0.01	0.17	0.30	0.39	-0.14

bold = statistically significant ($p < 0.01$)

Join smoothing

However good the join cost is, some join smoothing is usually necessary

- typically smooth the line spectral frequencies (LSFs)
- they have better interpolation properties than LPCs
- can synthesise directly from LSFs

Problem with current methods

- LSFs come in pairs
  - typically need 6 pairs
  - each pair describes something *a bit like* a formant peak + bandwidth
  - so must modify them in pairs
- worse still, each LSF pair is not independent of the other pairs
  - so should not even modify individual pairs of LSFs at all

Join smoothing

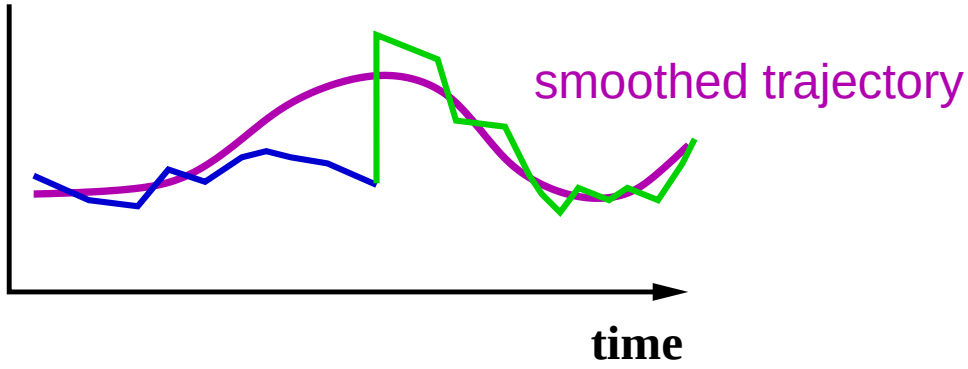
- interpolating individual LSFs can result in
  - sets of LSF values that would never occur in natural speech
- need a model which constrains the set of LSFs to be consistent
  - i.e. ones which **could occur in natural speech**

further

- want LSF **trajectories** to be
  - like those in natural speech

LDM join smoothing

- since the LDM is a generative model
  - can use it to generate smoothed LSF trajectories for joined phones



We'll be testing this method in a perceptual experiment

Conclusion

- if we pay attention to what the underlying articulation is, we can probably
  - devise a better join cost for concatenative synthesis
  - perform smoothing which results in more natural speech
- which hopefully leads on to being able to manipulate the articulation of recorded speech

links: www.cstr.ed.ac.uk www.rhetorical.com