Generating natural-sounding semisynthetic speech stimuli for sociophonetic experiments

Daniel Lawrence
School of Philosophy, Psychology & Language Sciences
The University of Edinburgh

dlawrenc@staffmail.ed.ac.uk
@danielplawrence

THE UNIVERSITY *of* EDINBURGH

E·S·R·C ECONOMIC & SOCIAL RESEARCH COUNCIL

Introduction

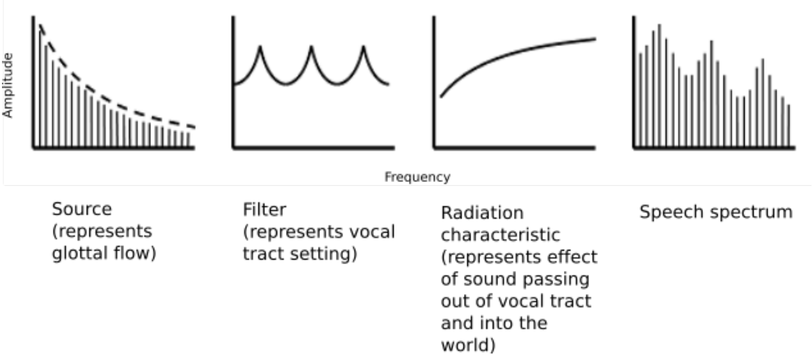
A typical aim of a sociophonetic perception study is to explore the impact of a single variable on a social judgment. Options:

- Use phonetically diverse natural stimuli (e.g. Clopper & Pisoni, 2004)
- Use stimuli performed by variable speakers (e.g. Evans & Iverson, 2004)
- Use stimuli performed by phoneticians (e.g. Kubisz, 2014)
- Use synthetic or semisynthetic stimuli (e.g. Kendall & Fridland, 2012; Hay, Warren & Drager, 2006)

Parametric synthesis

Basic source-filter theory (Fant, 1960):

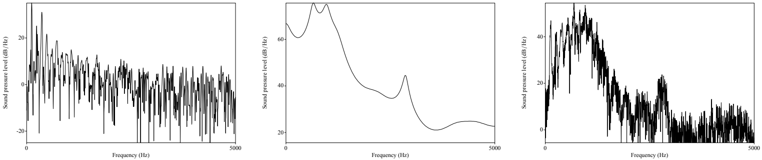
- Treat the speech signal as a combination of a sound source and vocal tract resonances, plus the effect of radiation from the lips:



Parametric synthesis

Basic source-filter theory (Fant, 1960):

- Treat the speech signal as a combination of a sound source and vocal tract resonances, plus the effect of radiation from the lips:

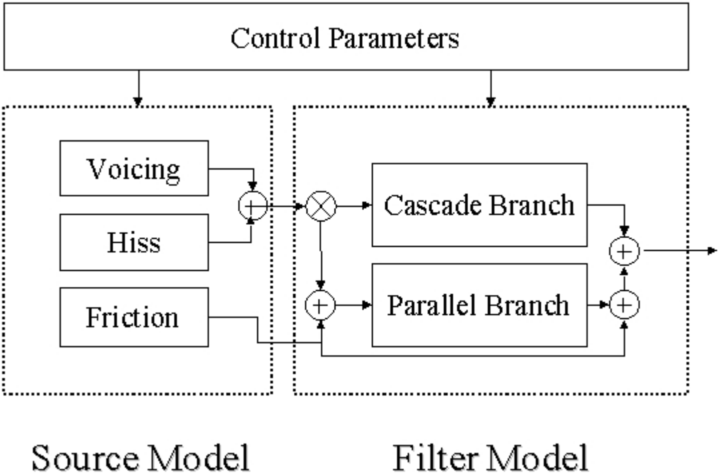


- To synthesize speech, we need to generate an excitation source and pass it through a set of digital filters

Parametric synthesis

Parametric synthesis:

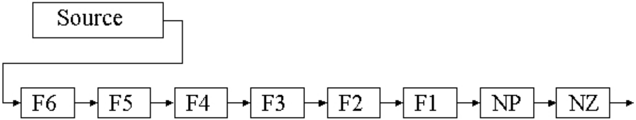
- Basic schematic of the Klatt (1980) synthesizer



Parametric synthesis

Parametric synthesis:

- Cascade branch of the Klatt (1980) synthesizer



- Each filter boosts the frequencies to match the resonances it represents.

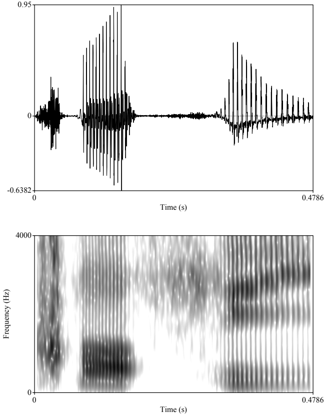
Parametric synthesis

In practice:

- Specify parameters for every time point.

N	V/C	Sym	Name	Min	Max	Typ
1	V	AV	Amplitude of voicing (dB)	0	80	0
2	V	AF	Amplitude of frication (dB)	0	80	0
3	V	AH	Amplitude of aspiration (dB)	0	80	0
4	V	AVS	Amplitude of sinusoidal voicing (dB)	0	80	0
5	V	F0	Fundamental freq. of voicing (Hz)	0	500	0
6	V	F1	First formant frequency (Hz)	150	900	450
7	V	F2	Second formant frequency (Hz)	500	2500	1450
8	V	F3	Third formant frequency (Hz)	1300	3500	2450
9	V	F4	Fourth formant frequency (Hz)	2500	4500	3300
10	V	FNZ	Nasal zero frequency (Hz)	200	700	250
11	C	AN	Nasal formant amplitude (dB)	0	80	0
12	C	A1	First formant amplitude (dB)	0	80	0
13	V	A2	Second formant amplitude (dB)	0	80	0
14	V	A3	Third formant amplitude (dB)	0	80	0
15	V	A4	Fourth formant amplitude (dB)	0	80	0
16	V	A5	Fifth formant amplitude (dB)	0	80	0
17	V	A6	Sixth formant amplitude (dB)	0	80	0
18	V	AB	Bypass path amplitude (dB)	0	80	0
19	V	B1	First formant bandwidth (Hz)	40	500	50
20	V	B2	Second formant bandwidth (Hz)	40	500	70
21	V	B3	Third formant bandwidth (Hz)	40	500	110
22	C	SW	Cascade/parallel switch	0(CASC)	1(PARA)	0
23	C	FGP	Glottal resonator 1 frequency (Hz)	0	600	0
24	C	BGP	Glottal resonator 1 bandwidth (Hz)	100	2000	100
25	C	FGZ	Glottal zero frequency (Hz)	0	5000	1500
26	C	BGZ	Glottal zero bandwidth (Hz)	100	9000	6000
27	C	B4	Fourth formant bandwidth (Hz)	100	500	250
28	V	F5	Fifth formant frequency (Hz)	3500	4900	3750
29	C	B5	Fifth formant bandwidth (Hz)	150	700	200
30	C	F6	Sixth formant frequency (Hz)	4000	4999	4900
31	C	B6	Sixth formant bandwidth (Hz)	200	2000	1000
32	C	FNP	Nasal pole frequency (Hz)	200	500	250
33	C	BNP	Nasal pole bandwidth (Hz)	50	500	100
34	C	BNZ	Nasal zero bandwidth (Hz)	50	500	100
35	C	BGS	Glottal resonator 2 bandwidth	100	1000	200
36	C	SR	Sampling rate	5000	20 000	10 000
37	C	NWS	Number of waveform samples per chunk	1	200	50
38	C	G0	Overall gain control (dB)	0	80	47
39	C	NFC	Number of cascaded formants	4	6	5

Parametric synthesis



Play

Parametric synthesis

Parametric synthesis in Praat:

- Praat implements the Klatt synthesizer through 'KlattGrid' objects
- You can imagine these as 2D grids with time on the X-axis and the Klatt settings on the Y-axis
- http://www.fon.hum.uva.nl/praat/manual/KlattGrid.html

```
#Create a KlattGrid
Create KlattGrid... aa 0 0.5 6 1 1 6 1 1 1
```

Parametric synthesis

Parametric synthesis in Praat:

```
#Add voicing amplitude, vowel formants, and pitch
    targets
Add voicing amplitude point... 0.0 0
Add voicing amplitude point... 0.04 90
Add voicing amplitude point... 0.25 90
Add voicing amplitude point... 0.5 90
Add pitch point... 0.0 150
Add pitch point... 0.5 150
```

Parametric synthesis

Parametric synthesis in Praat:

```
Add oral formant frequency point... 1 0.1 750
Add oral formant bandwidth point... 1 0.1 70
Add oral formant frequency point... 2 0.1 1250
Add oral formant bandwidth point... 2 0.1 120
Add oral formant frequency point... 3 0.1 2500
Add oral formant bandwidth point... 3 0.1 200
Add oral formant frequency point... 4 0.1 3900
Add oral formant bandwidth point... 4 0.1 300
#Synthesis
Play
To Sound
```

- Can you modify the code to generate a high front vowel?
- What about a diphthong?
- Try running ‘klatt_cardinal.psc’

Parametric synthesis

Pros of fully-parametric synthesis:

- Fine-grained control over parameters
- Given unlimited time and accurate measurements of the parameters of a source item, in principle possible to synthesize any speech sound
- Stimuli fully replicable as long as parameters are published

Parametric synthesis

Cons of fully-parametric synthesis:

- Properties of the glottal source particularly difficult to imitate.
- This means that tokens often have a 'robotic' quality – perhaps not appropriate for some sociophonetic applications.
- Parameter-setting can be very time consuming, particularly if we want to model dynamic properties of vowels.

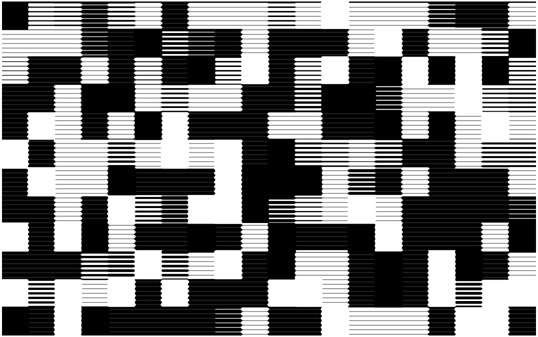
LPC inverse-filtering

An alternative: LPC inverse-filtering

- This technique has been implemented in a number of sociophonetic studies – as far back as Graff, Labov & Harris, 1984.
- Detailed technical outline in Alku et al. 1999
- This is what Bartek Plichta's *Akustyk* does...
- ...although I don't know the details of how BP has implemented it.

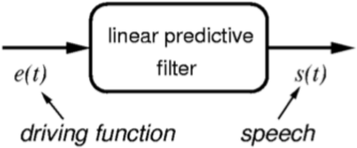
Linear Predictive Coding

- A technique for estimating the spectral envelope of a time-varying speech signal.
- Origins in early television encoding – black and white TV images are 3D grids (x,y,brightness) which vary over time.



- However, values do not vary randomly – the brightness of any point is likely to be related to the adjacent points

Linear Predictive Coding



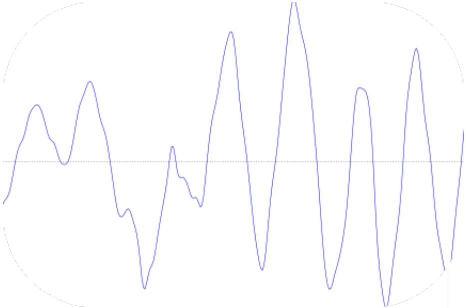
$$\hat{s}(t) = \sum_{j=1}^{p} a_j s(t-j)$$

$$s(t) = e(t) + \sum_{j=1}^{p} a_j s(t-j)$$

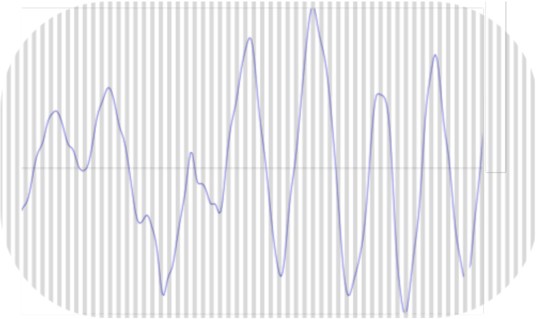
t = discrete time; p = filter order

Image courtesy of Simon King

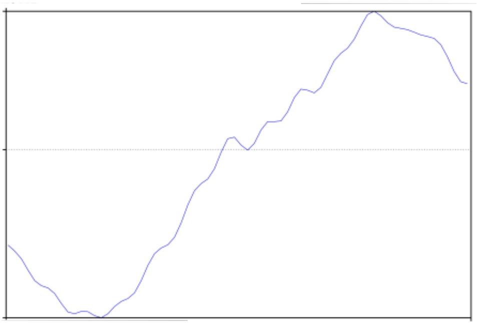
Linear Predictive Coding



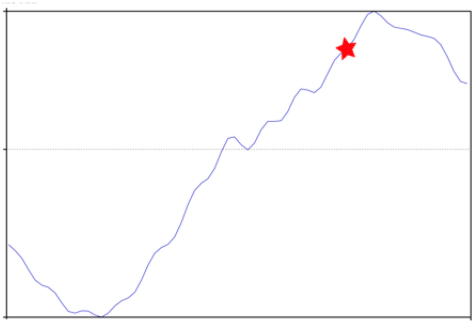
Linear Predictive Coding



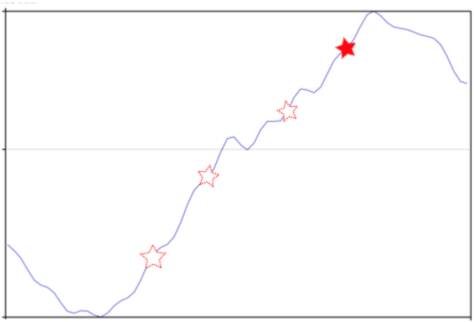
Linear Predictive Coding



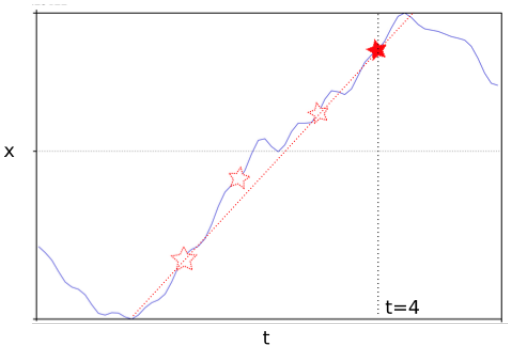
Linear Predictive Coding



Linear Predictive Coding

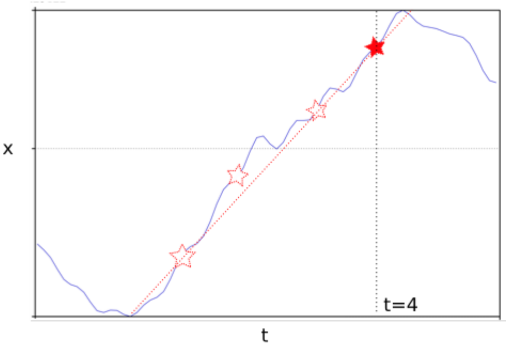


Linear Predictive Coding



$$x[t] = a_1 x[t-1] + a_2 x[t-2] + a_3 x[t-3]$$

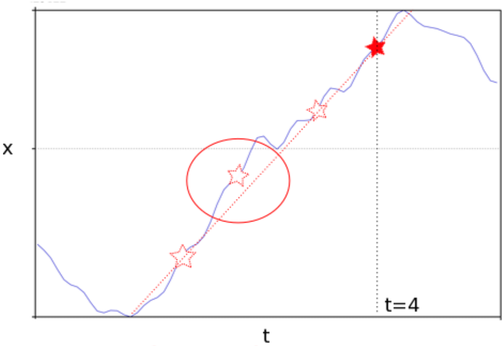
Linear Predictive Coding



$$x[t] = a_1 x[t-1] + a_2 x[t-2] + a_3 x[t-3]$$

$$[a_1 \quad a_2 \quad a_3]$$

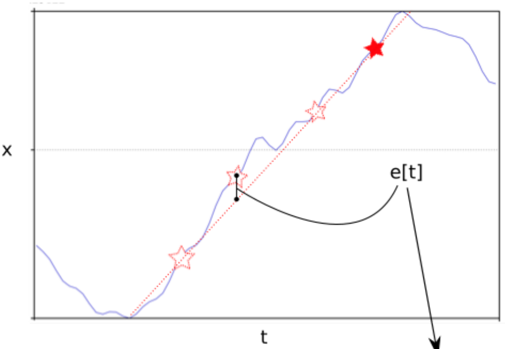
Linear Predictive Coding



$$x[t] = a_1 x[t-1] + a_2 x[t-2] + a_3 x[t-3]$$

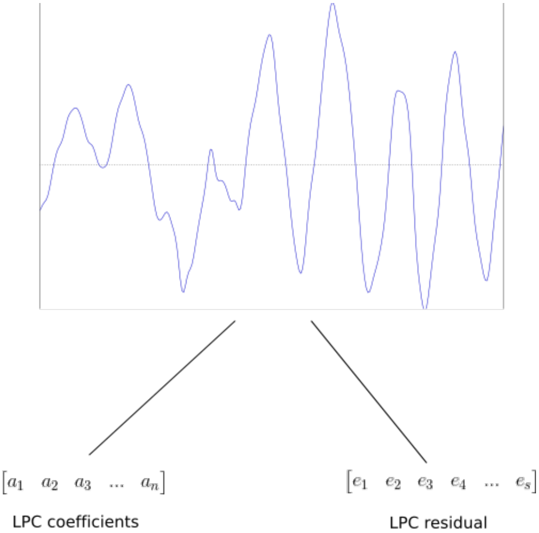
$$[a_1 \quad a_2 \quad a_3]$$

Linear Predictive Coding

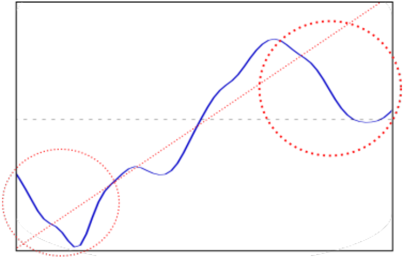


$$x[t] = a_1 x[t-1] + a_2 x[t-2] + a_3 x[t-3] + e[t]$$

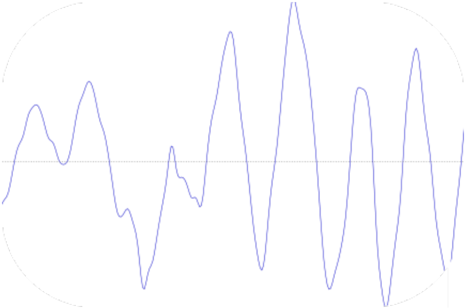
Linear Predictive Coding



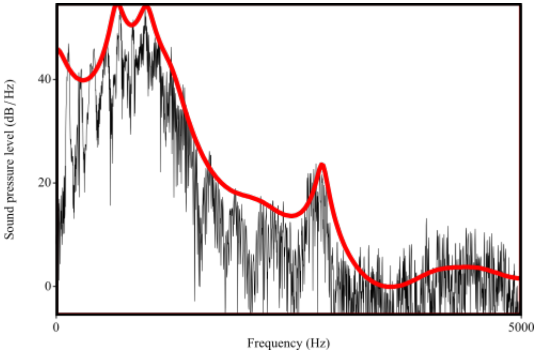
Linear Predictive Coding



Linear Predictive Coding



Linear Predictive Coding

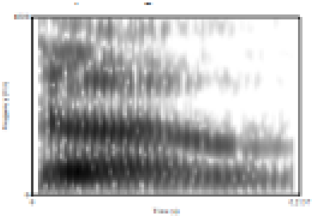


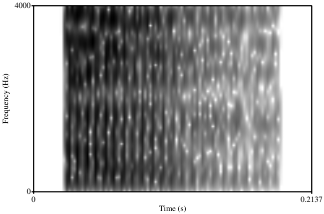
Linear Predictive Coding

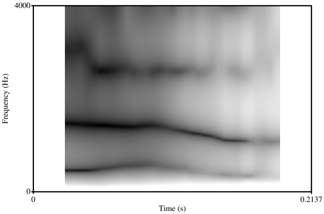
- Estimating the LPC filter is an optimization problem – we find the best set of *a* values for the given signal
- The difference between the LPC model and the actual signal is the *prediction residual* – together, the estimated LPC filter and residual encode the entire signal:

$$e(n) = x(n) - \widehat{x}(n)$$

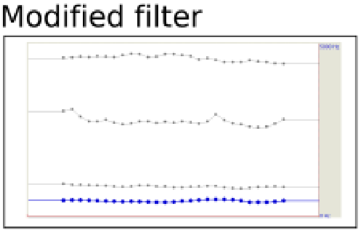
- Because the largest errors tend to be around the largest changes in the signal, the residual approximates the pitch periods, 'whitening' the signal by removing spectral characteristics.

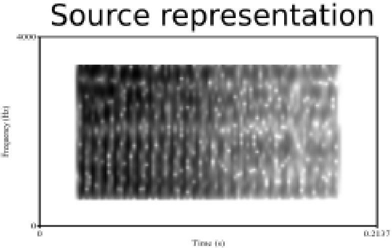



4000
0
Frequency (Hz)
0
0.2137
Time (s)

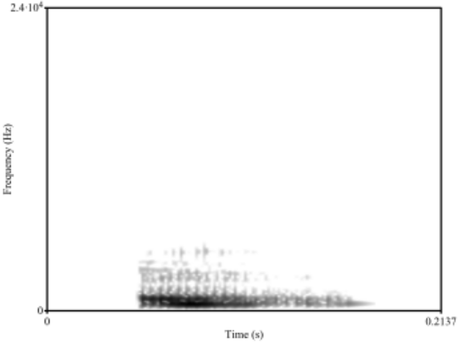

4000
0
Frequency (Hz)
0
0.2137
Time (s)

- Now we can excite a digital filter bank with our natural source representation



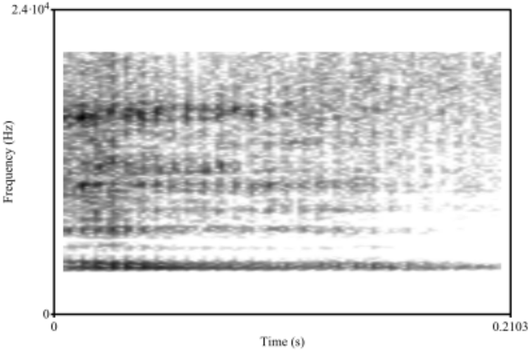


- Problem: LPC analysis results in the loss of the high-frequency component of the original sound



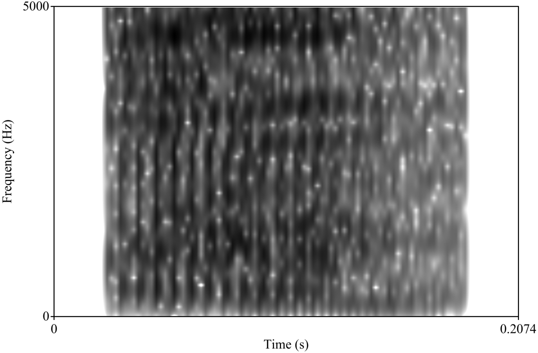
Play

- Solution: Restore the HF component of the original sound after synthesis

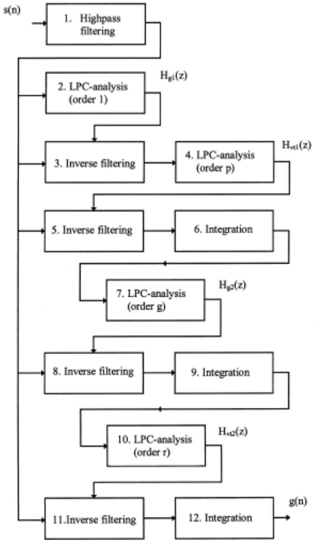


Play

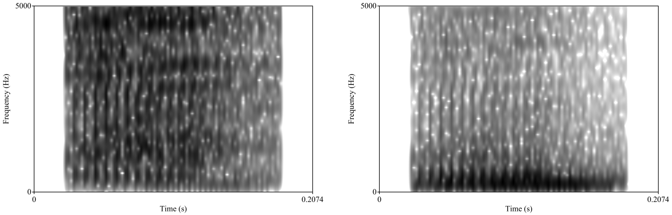
- Problem: How can we ensure that we have a good source model?



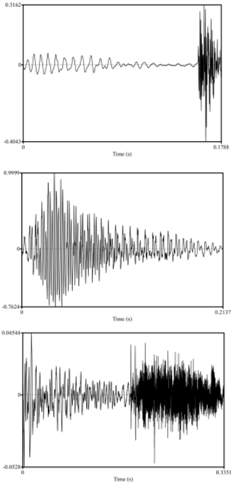
- Solution: Following Alku (1992), filter iteratively, estimating spectral characteristics of glottal flow and vocal tract with different LPC orders



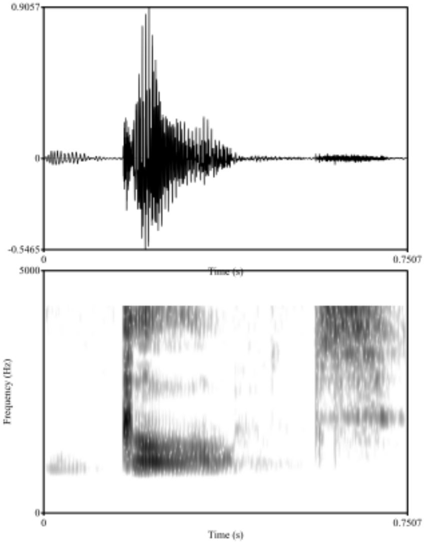
- Result: very high quality source-filter separation



- Finally, embed the vowel in a lexical item by splicing at zero-crossing points



- End result:



Play

Complete process

Resynthesis

LPC inverse-filtering in Praat:

```
#Estimate the LPC filter for a selected sound
#First we need to resample
Resample: 10000, 50
To LPC (burg): 8, 0.025, 0.005, 50
```

Resynthesis

```
#Take the inverse of this filter to get a
    representation of the source
selectObject: "Sound untitled_10000"
plusObject: "LPC untitled_10000"
Filter (inverse)
```

Resynthesis

```
#Generate a formant object and add 400 Hz to F2
selectObject: "LPC untitled_10000"
To Formant
selectObject: "Formant untitled_10000"
Formula (frequencies): "if row = 2 then self + 400
    else self fi"
```

Resynthesis

```
#Combine the source and filter representations to
    make a new vowel
selectObject: "LPC untitled_10000"
selectObject: "Sound untitled_10000"
plusObject: "Formant untitled_10000"
Filter
Play
```

- Try recording yourself producing a vowel (ctr+r). Select your vowel token and run 'LPC_cardinal_iaif.praat'

Summary

- A range of options available when preparing perception experiments
- Trade off between naturalness and control of phonetic detail
- In some cases, the face validity of the experiment may be more important than others
- In some cases, a lack of naturalness might even strengthen our arguments!
- Importance of explicitness about manipulation methods: no black boxes
- *Praat* is capable of very sophisticated analysis and manipulations, and is open source

References

- Alku, P., Tiitinen, H., & Naatanen, R. (1999). A method for generating natural-sounding speech stimuli for cognitive brain research. Clinical Neurophysiology, 110(8), 1329-1333.
- Clopper, C. G., & Pisoni, D. B. (2004). Some acoustic cues for the perceptual categorization of American English regional dialects. Journal of Phonetics, 32(1), 111-140.
- Evans, B. G., & Iverson, P. (2004). Vowel normalization for accent: An investigation of best exemplar locations in northern and southern British English sentences. The Journal of the Acoustical Society of America, 115(1), 352-361.
- Hay, J., Nolan, A., & Drager, K. (2006). From fush to feesh: Exemplar priming in speech perception. The linguistic review, 23(3), 351-379.
- Klatt, D. H. (1980). Software for a cascade/parallel formant synthesizer. the Journal of the Acoustical Society of America, 67(3), 971-995.
- Kubisz, A. (2014). The role of gendered sociolinguistic variants as perceptual cues. York Working Papers in Linguistics 1
- Kendall, T., & Fridland, V. (2012). Variation in perception and production of mid front vowels in the US Southern Vowel Shift. Journal of Phonetics, 40(2), 289-306.

Links

- Formant manipulation script on Github: https://github.com/danielplawrence/semisynthetic
- Will Styler's resynthesis scripts: https://github.com/stylerw/styler_praat_scripts/tree/master/source_filter_vowel_resynth
- Similar stuff from Sam Kirkham: http://samkirkham.com/scripts/index.html
- Instructions for source-filter synthesis in *Praat*: http://www.fon.hum.uva.nl/praat/manual/Source-filter_synthesis.html
- PraatR: http://www.aaronalbin.com/praatr/