Duration

□ Duration for each phones:
- fixed (100ms)
- average
- statistically modeled
- natural

□ Overall speaking rate
- global figure
- need duration contour

Festival approach

□ Collection of 153 features per segment
- phonetic feature plus context
- syllable type, position
- phrasal position
- *no* phone names

□ domain:
- absolute, log, or
- zscores ((X-mean)/stddev)

□ CART or Linear Regression similar results
- 26ms RMSE 0.78 correlation

Other duration approaches

- □ Syllable-based methods
  - – Predict syllable times, then segment durations
  - – But segment times don't correlate with syllable times
- □ Sums of Products model:
  - – Linear Regression is: $W_0.F_0 + W_1.F_1 + ... + W_nF_n$
  - – SoP model is $W_0.(F_0 * F_1 * ...) + W_i.(F_i * F_{i+1}...) + ...$
  - – finding the right mix is computationally expensive
  - – finding weights is easy
- □ Other learning techniques:
  - – neural nets ...
- □ None predict varying speaking rate

Building a duration model

□ Need data:
– suitable speech data

□ Need Labels:
– all the labels/structure necessary

□ Need feature extraction:
– Should be *same* format as in synthesis

□ Need training algorithm

□ Need testing criteria

KDT Database

□ KED Timit databases:
  – 452 phonetically balanced sentences
  – *"She had your dark suit in greasy wash water all year."*

□ Hand labeled phonetically

□ Recorded with EGG

□ Collated into festival utterance structures

Building a duration model

Need to predict a duration for every segment
What features help predict duration?

□ Phone:
- type: vowel, stop, frictative

□ Phone context:
- preceding/succeeding phones (types)

□ Syllable context:
- onset/coda, stressing
- word initial, middle final

□ Word/phrasal:
- content/function
- phrase position

□ Others?

Extracting training data

`dumpfeats`

□ -relation Segment

□ -feats durfeats.list

□ -output durfeats.train

□ utt0, utt1, utt2 ...

Festival Utterance feature names

- □ segment_duration
- □ name n.name p.name
- □ ph_*:
  - – ph_vc
  - – ph_vheight ph_vlng ph_vfront ph_vrnd
  - – ph_cplace ph_ctype ph_cvox
- □ pos_in_syl syl_initial syl_final
- □ Syllable context:
  - – R:SylStructure.parent.syl_break
  - – R:SylStructure.parent.R:Syllable.p.syl_break
  - – R:SylStructure.parent.stress

Full list is in Festival manual
Note features **and** pathnames

Train and test data

Guidelines

□ Approx 10% data for test

□ Could be partitioning or
– every nth utterance

□ For timit let's use:
– train: utts 001-339
– test: utts 400-452

```
dumpfeats -relation Segment -feats durfeats.list
     -output durfeats.train kdt_[0-3]*.utt

dumpfeats -relation Segment -feats durfeats.list
     -output durfeats.test kdt_4*.utt
```

```
0.399028 pau 0 sh 0 0 0 0 0 0 0 0 - f 0 0 0 0 p - 0 1 1 0 0 0
0.08243 sh pau iy - 0 0 0 0 0 0 - + 0 1 l 1 - 0 0 0 1 0 1 0 0
0.07458 iy sh hh - f 0 0 0 0 p - - f 0 0 0 0 g - 1 0 1 1 0 0
0.048084 hh iy ae + 0 1 l 1 - 0 0 + 0 3 s 1 - 0 0 0 1 0 1 1 1
0.062803 ae hh d - f 0 0 0 0 g - - s 0 0 0 0 a + 1 0 0 1 1 1
0.020608 d ae y + 0 3 s 1 - 0 0 - r 0 0 0 0 p + 2 0 1 1 1 1
0.082979 y d ax - s 0 0 0 0 a + + 0 2 a 2 - 0 0 0 1 0 1 1 1
0.08208 ax y r - r 0 0 0 0 p + - r 0 0 0 0 a + 1 0 0 1 1 1
0.036936 r ax d + 0 2 a 2 - 0 0 - s 0 0 0 0 a + 2 0 1 1 1 1
0.036935 d r aa - r 0 0 0 0 a + + 0 3 l 3 - 0 0 0 1 0 1 1 1
0.081057 aa d r - s 0 0 0 0 a + - r 0 0 0 0 a + 1 0 0 1 1 1
0.0707901 r aa k + 0 3 l 3 - 0 0 - s 0 0 0 0 v - 2 0 0 1 1 1
0.05233 k r s - r 0 0 0 0 a + - f 0 0 0 0 a - 3 0 1 1 1 1
0.14568 s k uw - s 0 0 0 0 v - + 0 1 l 3 + 0 0 0 1 0 1 1 1
0.14261 uw s t - f 0 0 0 0 a - - s 0 0 0 0 a - 1 0 0 1 1 1
0.0472 t uw ih + 0 1 l 3 + 0 0 + 0 1 s 1 - 0 0 2 0 1 1 1 1
0.04719 ih t n - s 0 0 0 0 a - - n 0 0 0 0 a + 0 1 0 1 1 0
0.0964501 n ih g + 0 1 s 1 - 0 0 - s 0 0 0 0 v + 1 0 1 1 1 0
0.0574499 g n r - n 0 0 0 0 a + - r 0 0 0 0 a + 0 1 0 0 1 1
0.0441101 r g iy - s 0 0 0 0 v + + 0 1 l 1 - 0 0 1 0 0 0 1 1
```

Build CART model

wagon needs

□ feature descriptions:
– names and types (class/float)
– make_wagon_desc durfeats.list durfeats.train durfeats.desc
– and edit output

□ tree build options:
– stop size (20?)
– held out data ?
– stepwise

□ Change domain:
– absolute, log, zscores
– ensure testing done in (absolute) domain

```
wagon -desc feats.desc -data feats.train -stop 20 -output dur.tree
Dataset of 12915 vectors of 26 parameters from: feats.base.train
RMSE 0.0278 Correlation is 0.9233 Mean (abs) Error 0.0171 (0.0219)

wagon_test -desc feats.desc -data feats.test -tree dur.tree
RMSE 0.0313 Correlation is 0.8942 Mean (abs) Error 0.0192 (0.0246)
```

Testing the model

□ Use wagon_test on test data:
– is this a good test set

□ On “real” data:
– Add new tree to synthesizer
– test it

□ Does it sound better:
– can you tell?

Other prosody

□ Power/energy variation:
- Build power contour for segments
- Need underlying power
- segments are naturally different power

□ Segmental/spectral variation:
- shouting isn't just volume
- can spectral qualities be varied

Using prosody

□ Predict default "neutral" prosody:
  – but that's boring
  – but it avoids making mistakes

□ What about emphasis, focus, contrast?

Emphasis

□ How *is* emphasis rendered

– raised pitch, different accent type

– phrasing, duration, power

– some combination

– not well understood

□ *Where* is emphasis required

– on the focus of the sentences

– (where/what is the “focus”)

Emphasis Synthesis

Record an emphasis database:

He _did then _know what _had occurred.
_Tarzan and _Jane raised _their heads.
...

Synthesize as:

This is a short example
_This is a short example
This _is a short example
This is _a short example
...

Semantic correlates of prosody

□ Same pitch contour may “mean” different things
– surprise/redundacy contour

□ “L*..” good at focus (sort of)

□ Find focus/contrast in text is AI hard
– but in concept to speech its given (maybe)

□ What is the relationship between concept and speech

Speech Styles

□ Multiple dimensions

□ Emotion:
– happy, sad, angry, neutral

□ Speech genre:
– news, sportscaster, helpful agent

□ Simpler notions:
– text reader vs conversation

□ Delivery style:
– polite, command
– speaking in noise

Voice characteristics

- □ How much is spectral and how much prosody:
  - – Elvis reading the news
  - – Bart Simpson delivering a sermon
  - – Teletubbies as Darth Vader

Prosodic style models

□ It costs time to get/label data:
– how do you prompt for intonational variation?

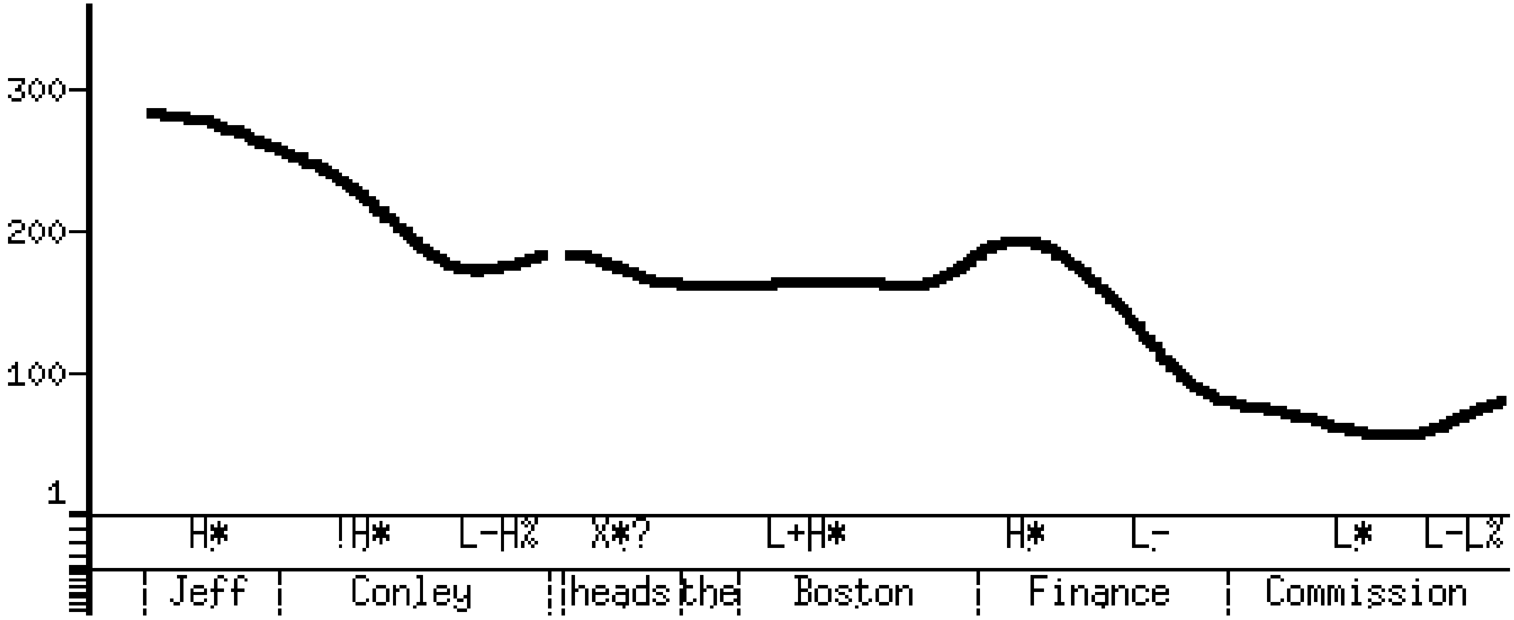
□ Build basic models from *lots* of data

□ Collect small amount data in style

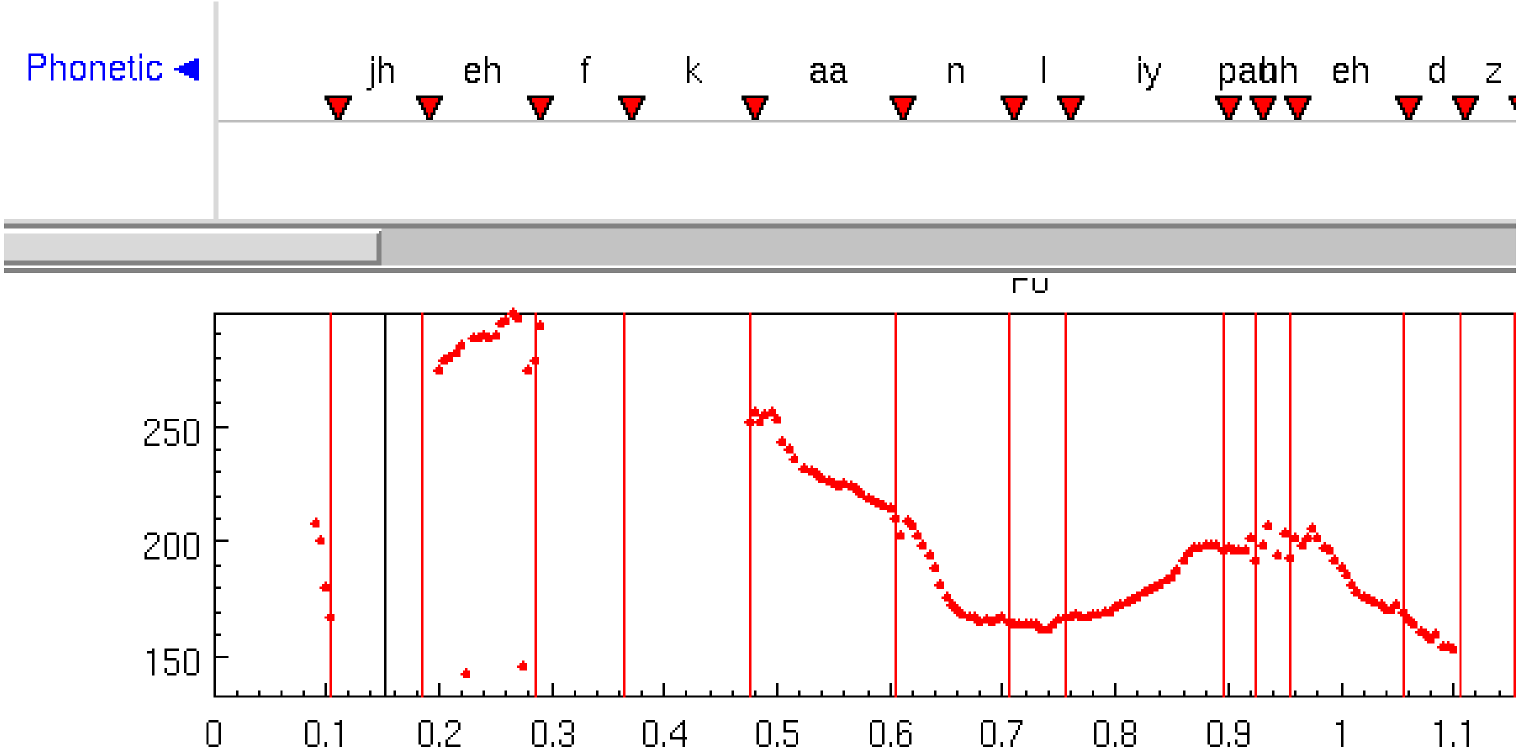
□ Interpolate the models:
– (easier said than done)

□ How can you tell if its right?

Finding the F0



Raw F0


Phonetic
jh
eh
f
k
aa
n
l
iy
pau
ah
eh
d
z
F0
250
200
150
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1

Extracting F0

□ Need to know pitch range

□ No pitch during unvoiced sections

□ Segmental perturbations (micro-prosody)

□ Pitch doubling and halving errors common

Finding the right answer

monitoring the signal more directly

□ Record electrical activity in larynx

□ Attach electrodes to throat and record with speech

□ Wave signal has implicit information but

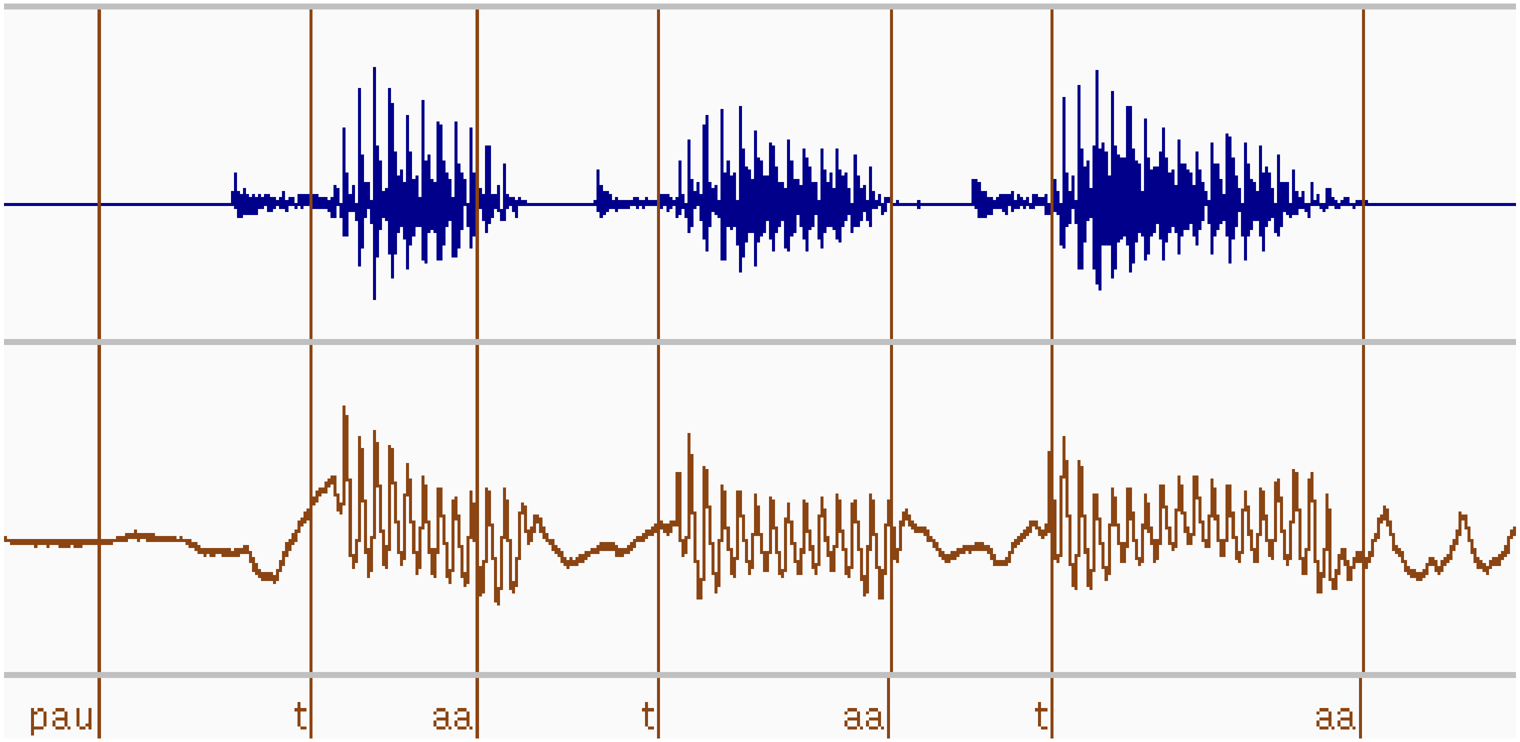
□ elctroglottograph (EGG) info is more direct

(sometimes called larynograph LAR)
But,

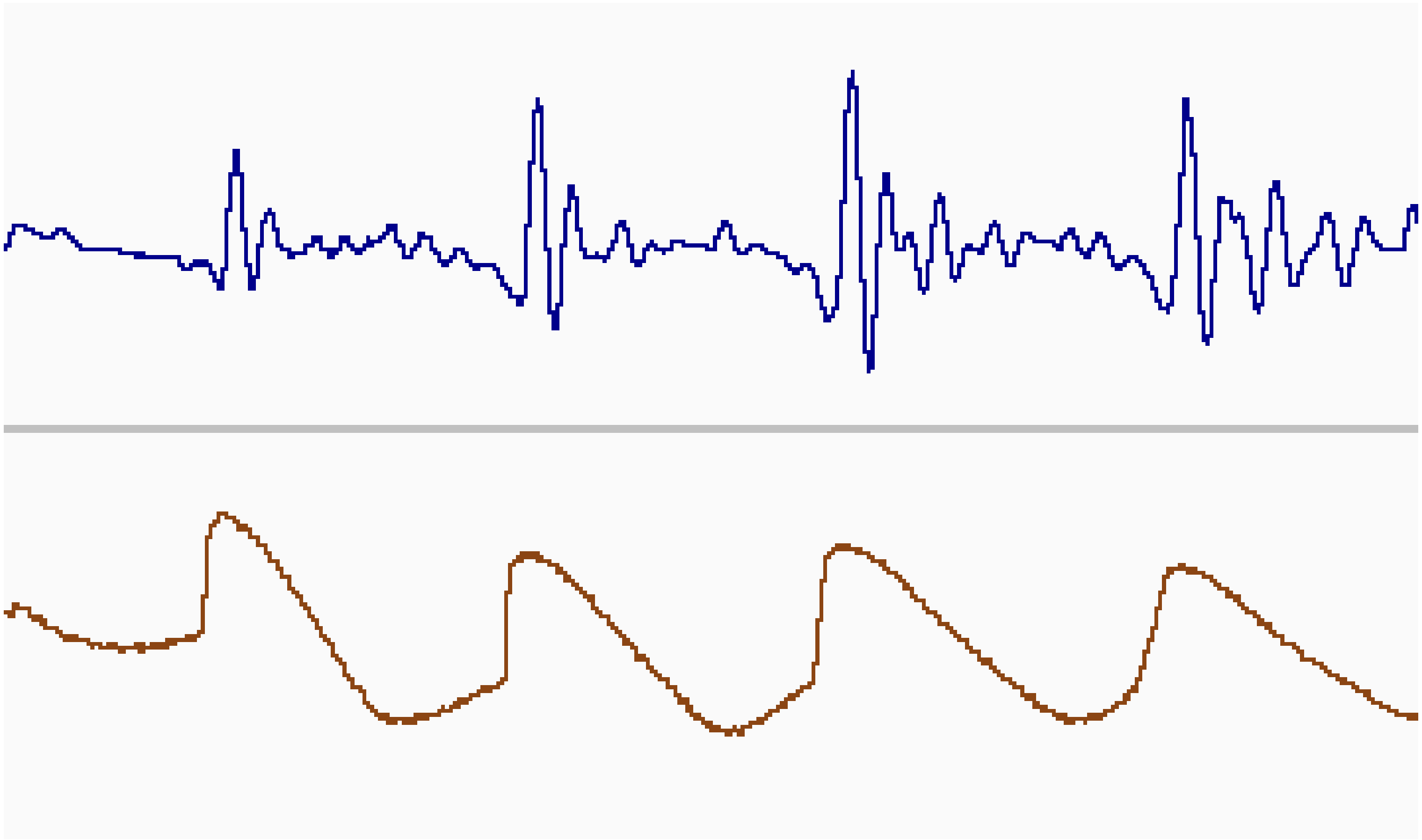
□ Specialized equipment

□ must be recorded as *same* time

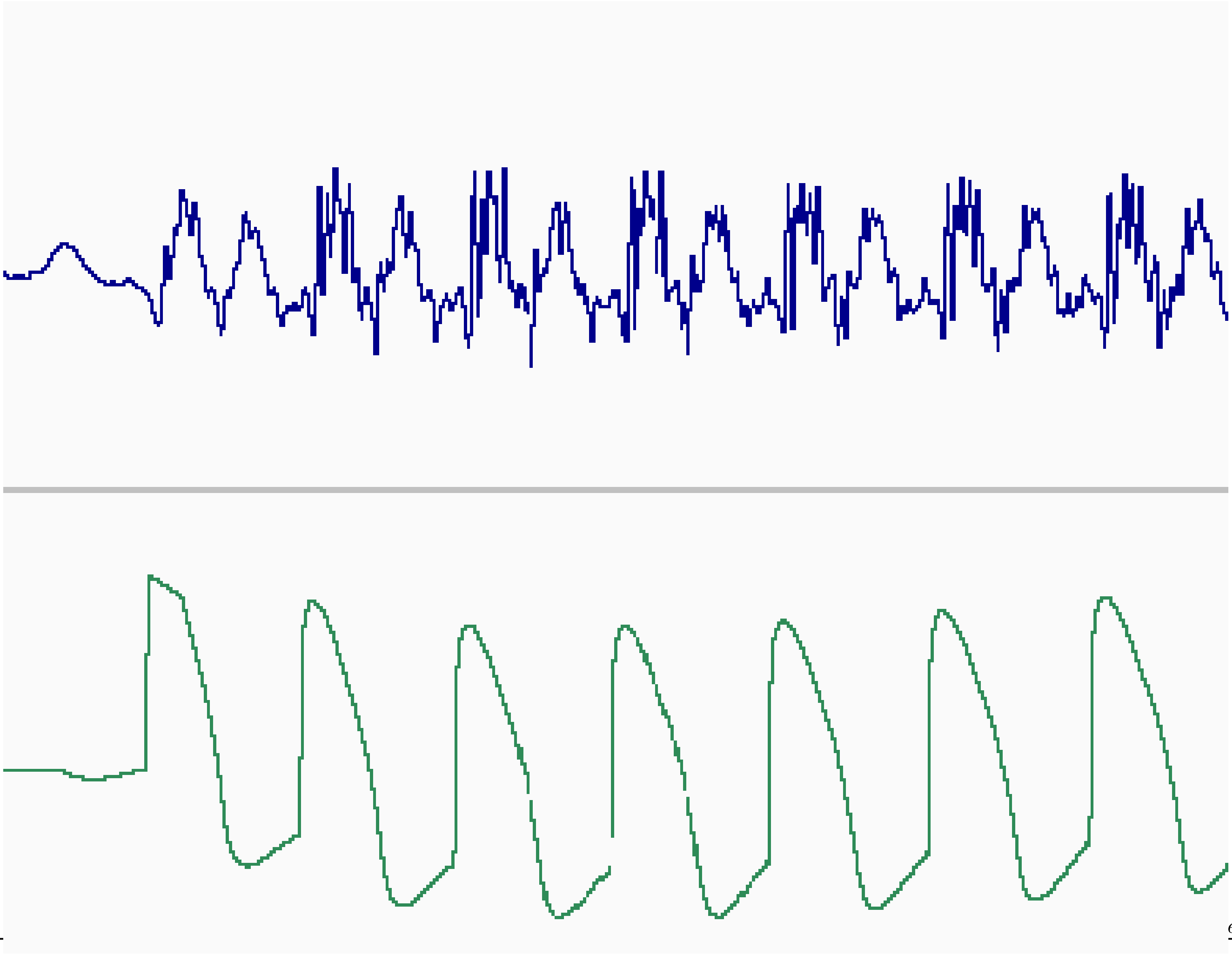
Wave plus EGG signal



Wave plus EGG signal



Wave plus EGG signal



Pitch Detection Algorithm

(many different)

□ Low pass filter

□ autocorelation

□ Linear interpretation through unvoiced regions

□ smoothing

Two uses of F0 extraction

□ F0 contour:
- pitch at 10ms intervals
- used for F0 modeling

□ Pitch periods:
- actual position of glottal pulse
- used in prosody modification

Linguistic/Prosody Summary

From words to pronunciations, durations and F0

□ Pronunciation:

– lexicons

– letter to sound rules

– post-lexical rules

□ Prosody:

– phrasing

– intonation: accents and F0 generation

– duration

– power

Testing prosodic models

Do measures correlate with human perception

Phenomena	Measureable	Measure	Alternative
Pitch	F0	Hz	Log/zscore/Bark scale
Timing	Duration	ms	Log/zscore
Energy	Power	log RMS	

Typically measure correlate but not linearly
What about tied models?

Exercises for April 9th

Do either 1 or 2

1. Build an intonation system that adds small hat accents to all stressed syllables, without declination, except on final syllables, which rise for a question and fall for all other sentence types.
2. Build a duration model using CART for one of the CMU Arctic databases. Use given example features and report the RMSE and Correlation. Modify the features/wagon parameters etc to show (at least) one model that is better (or two that are worse). Run these models and comment how they actually sound.

You need to use the General Intonation system to build this. You'll need an accent assignment decision tree and a Scheme function predict the targets for each syllable.
The accent assignment tree is as mentioned above

```
(set! int_accent_cart_tree
 '
  ((R:SylStructure.parent.gpos is content)
   ((stress is 1)
    ((Accented))
    ((position_type is single)
     ((Accented))
     ((NONE))))
   ((NONE))))
```

And you'll need to write a function that generates the F0 targets, something like

```
(define (targ_func1 utt syl)
  "(targ_func1 UTT ITEM)
Returns a list of targets for the given syllable."
  (let ((start (item.feat syl "syllable_start"))
        (end (item.feat syl "syllable_end")))
    (cond
     ((string-matches (item.feat syl "R:Intonation.daughter1.name") "A
       (list
        (list start 110)
        (list (/ (+ start end) 2.0) 140)
        (list end 100)))
      (
        ;; End of utterance as question
        ;; target for mid point and high end pont
      )
      ( ;; End of utterance but not question
        ;; target for mid point and low end pont
      ))))
```

The condition `(equal? nil (item.next syl))` will be true for the last syllable in the utterance and

```
(string-matches (item.feat syl
    "R:SylStructure.parent.R:Word.last.R:Token.parent.punc") "?")
```

will be true if there is a question mark at the end of the utterance.

You also need to set up these functions as the intonation method

```
(voice_ked_diphone)
(Parameter.set 'Int_Method 'General)

(set! int_general_params
      (list
       (list 'targ_func targ_func1)))
```

Basic data in http://festvox.org/dbs/. cstr_us_kdt, or any of the 7 arctic databases.
Utterances are in DB/festival/utts Some basic features in feats.basic
For more features (or add your own) see Festival manual appendix
http://festvox.org/docs/manual-1.4.2/

```
dumpfeats -relation Segment -feats .../feats.basic
     -output .../durfeats.train .../kdt_[0-3]*.utt
make_wagon_desc durfeats.list durfeats.train durfeats.desc
```

(edit durefests.desc)

```
wagon -desc feats.desc -data feats.train -stop 20 -output dur.tree
wagon_test -desc feats.desc -data feats.test -tree dur.tree
```

```
(define (Duration_Simple_Tree utt)
  "Duration_Simple_Tree utt)
predicts Segment durations with a simple CART tree (that returns
absolute times in seconds)."
  (let ((end 0))
    (mapcar
     (lambda (s)
       (let ((dur (wagon_predict s simple_duration_cart_tree)))
         (set! dur (* (Parameter.get 'Duration_Stretch) dur))
;        (format t "%s %f\n" (item.name s) dur)
         (set! end (+ dur end))
         (item.set_feat s "end" end)
         ))
     (utt.relation.items utt 'Segment))
    utt))

(voice_ked_diphone)
(Parameter.set 'Duration_Method Duration_Simple_Tree)
(set! simple_duration_cart_tree
      (car (load "/home/awb/data/kdt/dur.tree" t)))
```