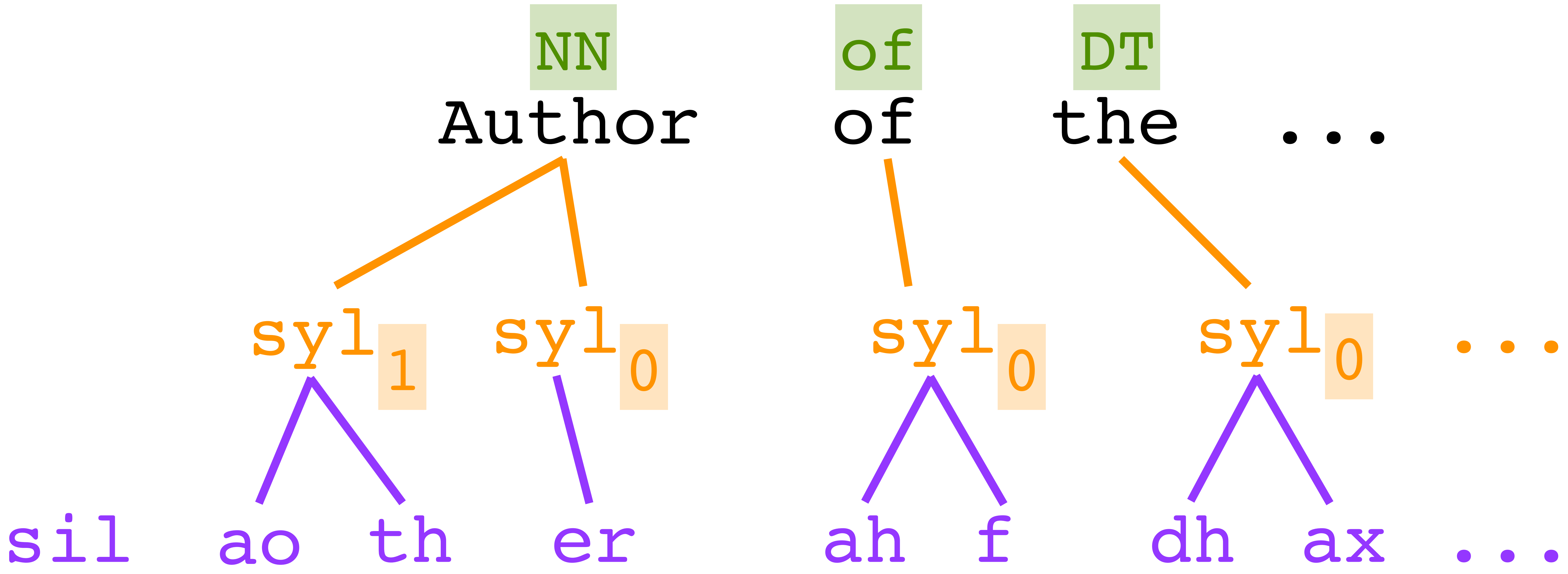
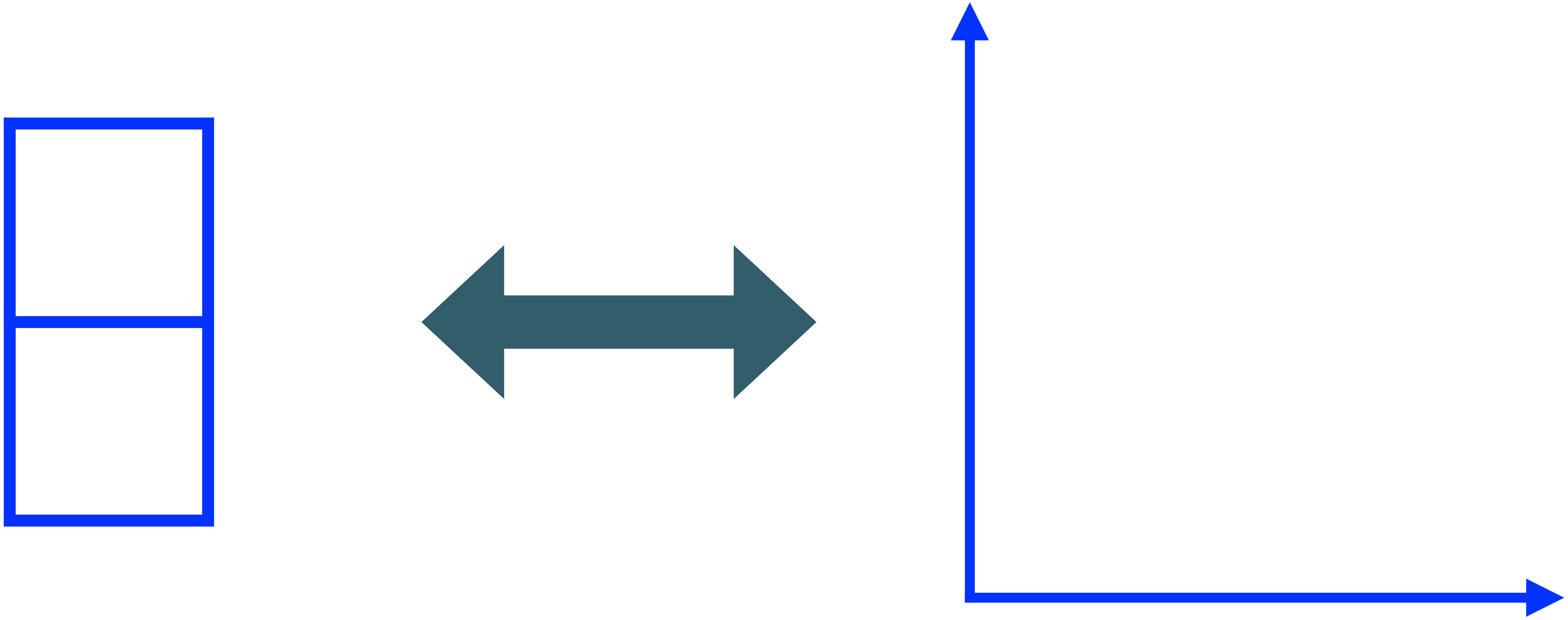
Orientation

- Modules 2 & 3
  - Unit selection speech synthesis
- Modules 4 & 5
  - The database
  - Evaluation
- Module 6

Anything can be represented as a sequence of vectors, even this:



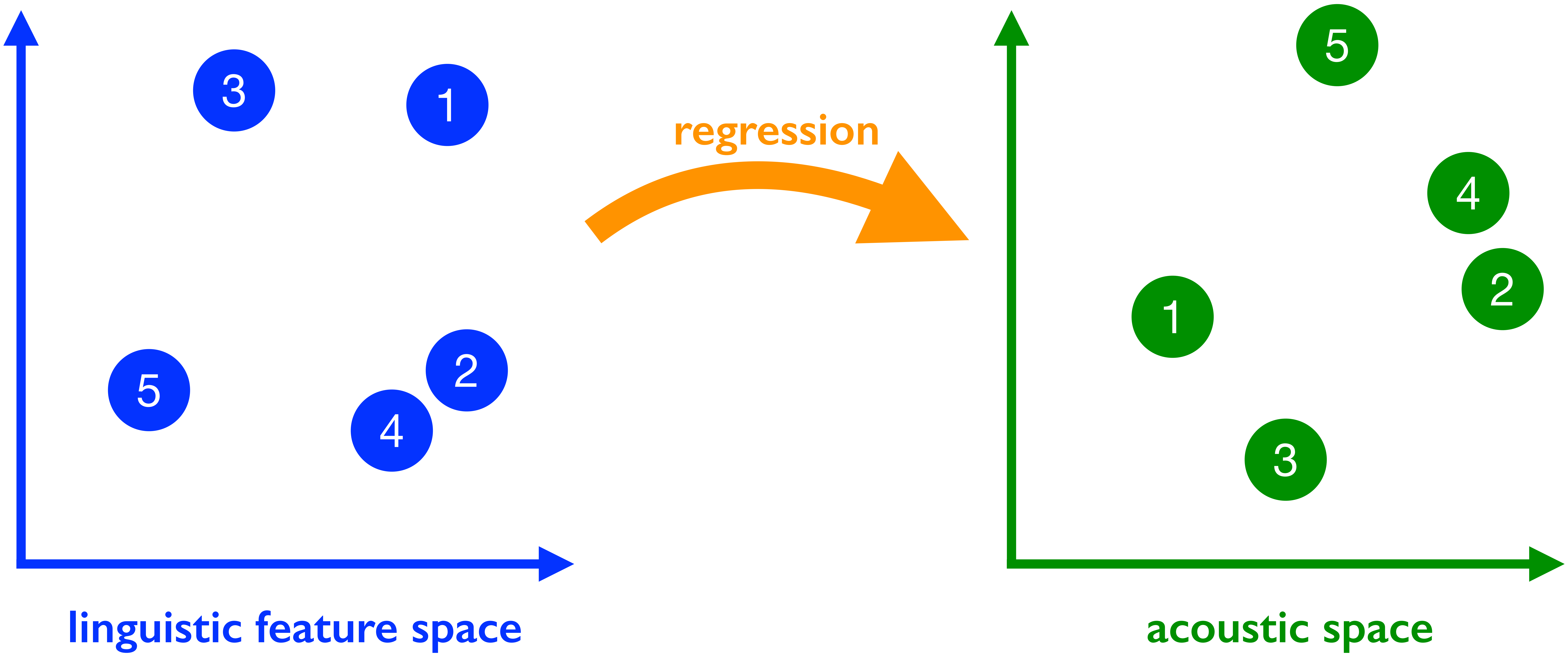
Feature vectors live in a "feature space"



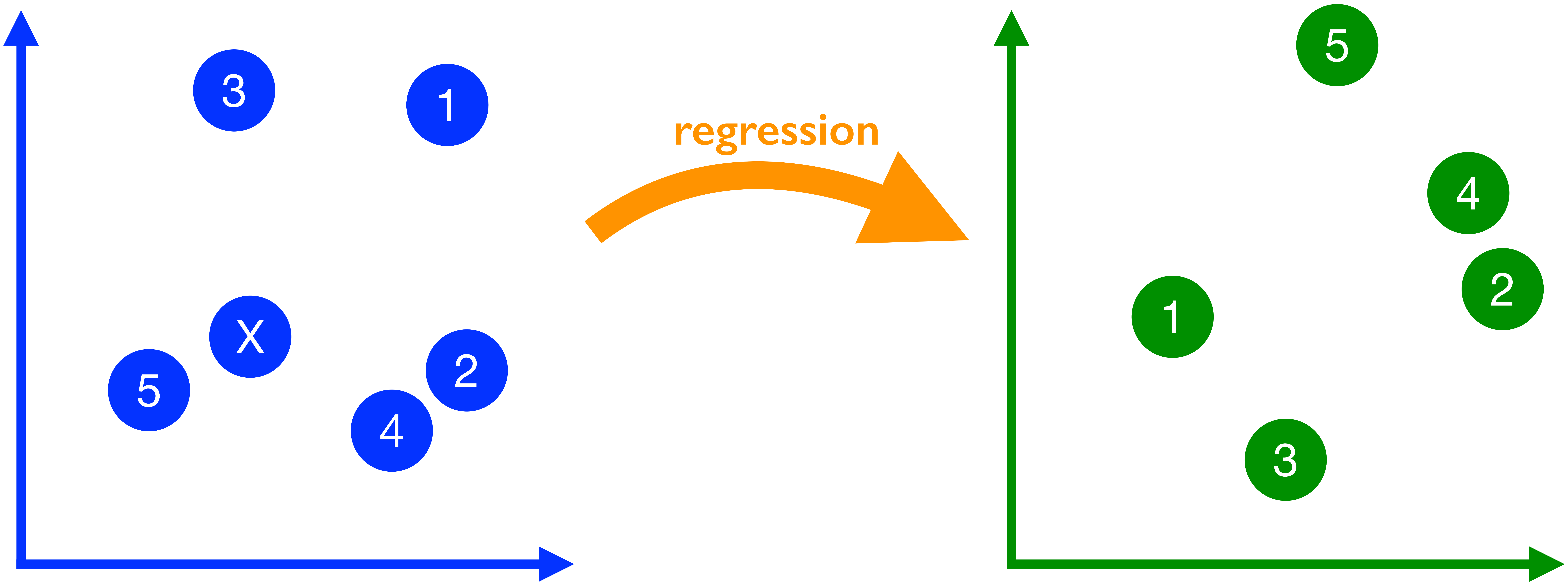
feature vector

feature space

Speech synthesis using an exemplar-based regression function (unit selection)



Speech synthesis using an exemplar-based regression function (unit selection)

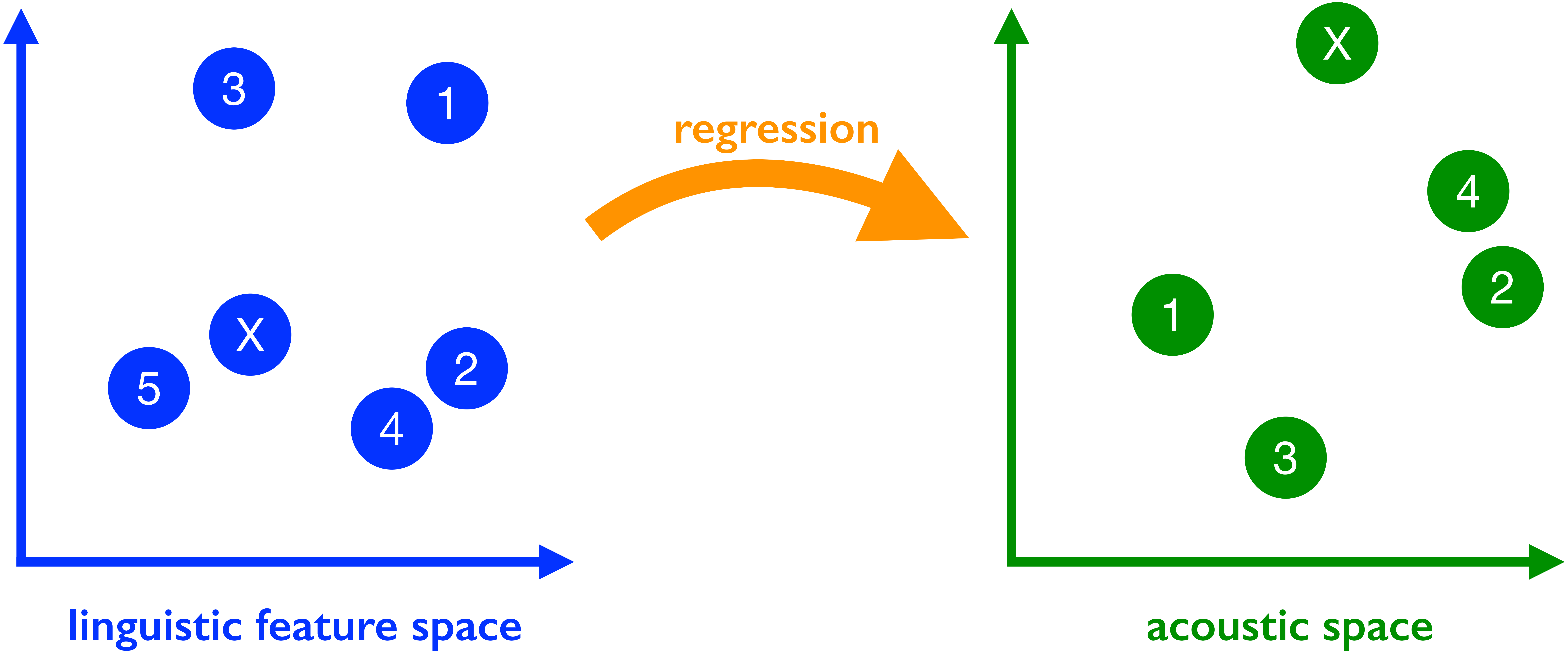


regression

linguistic feature space

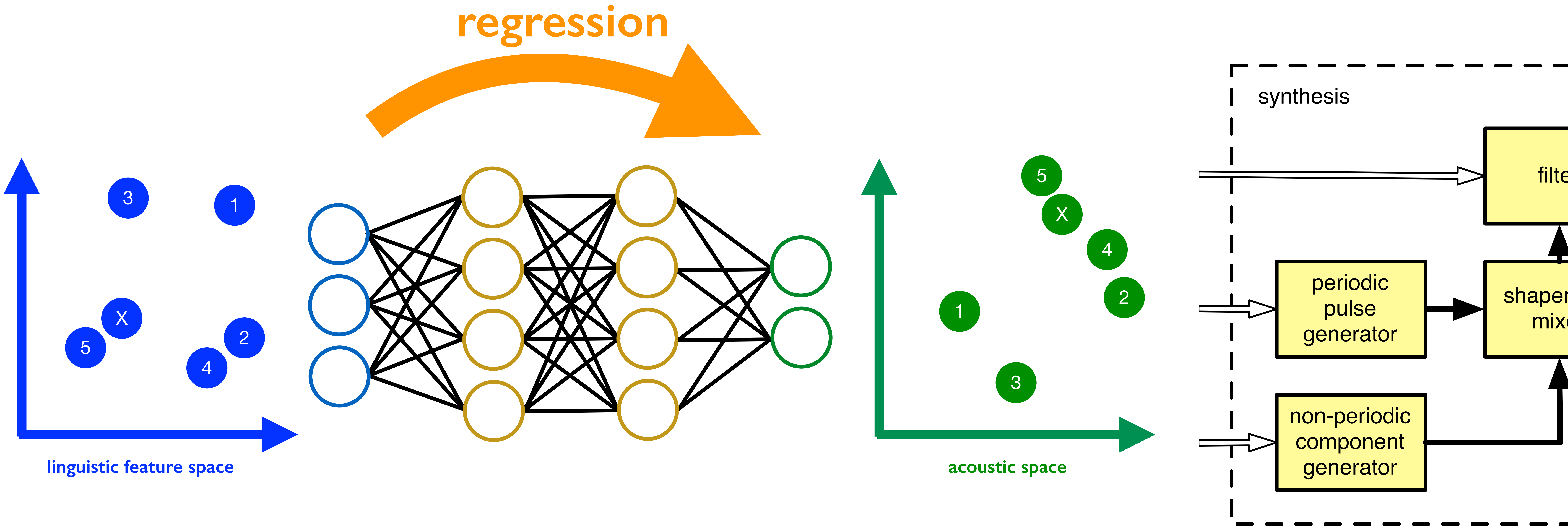
acoustic space

Speech synthesis using an exemplar-based regression function (unit selection)

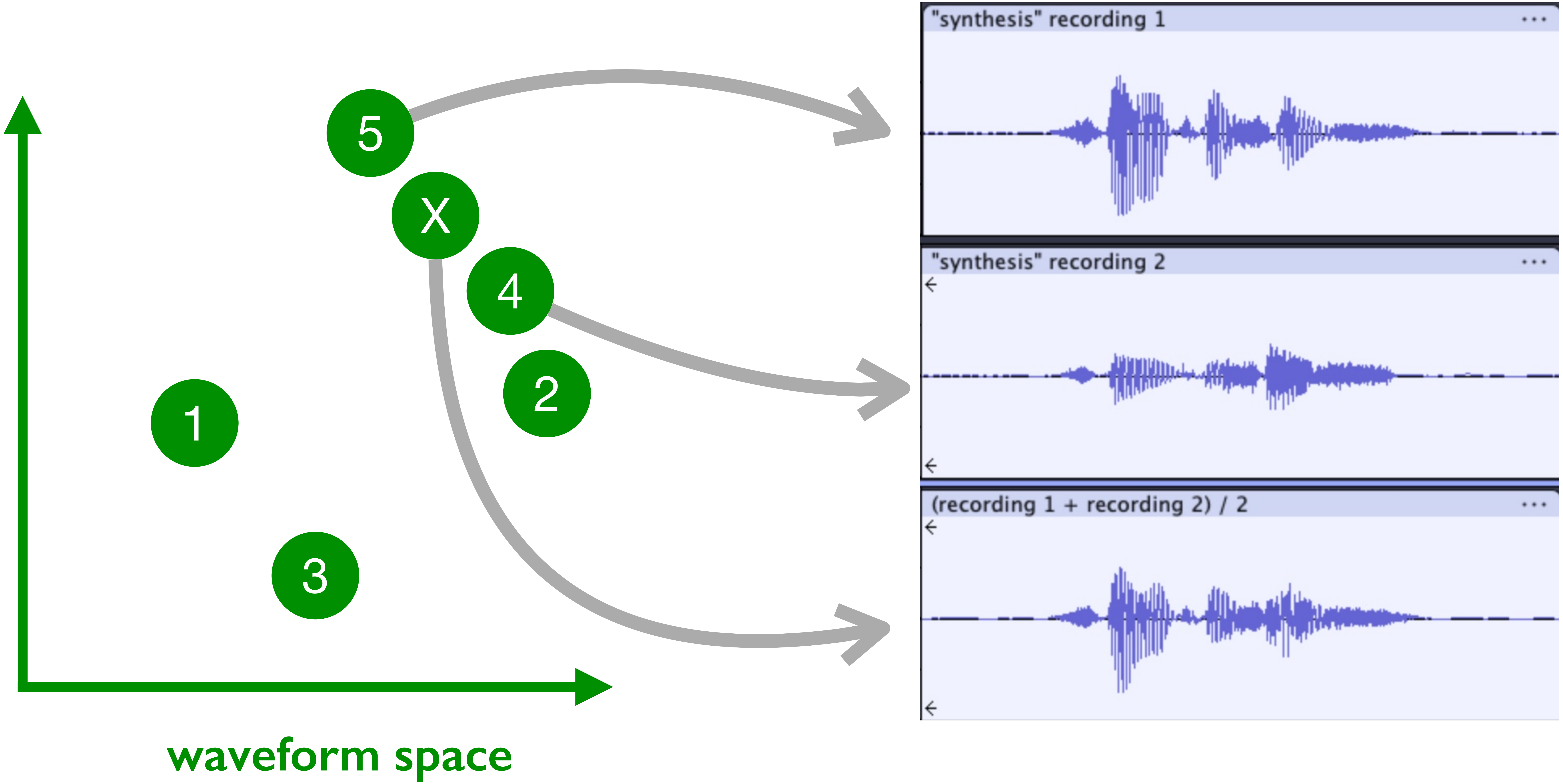


Speech synthesis using a "smooth" regression function (neural network)

```
[0 0 1 0 0 1 0 1 1 0 ... 0.2 0.0]
[0 0 1 0 0 1 0 1 1 0 ... 0.2 0.1]
...
[0 0 1 0 0 1 0 1 1 0 ... 0.2 1.0]
[0 0 1 0 0 1 0 1 1 0 ... 0.4 0.0]
[0 0 1 0 0 1 0 1 1 0 ... 0.4 0.5]
```



Why we need to choose a suitable representation



Orientation

- Modules 2 & 3 ✓
  - Unit selection speech synthesis
- Modules 4 & 5
  - The database
  - Evaluation
- Module 6

What we have learned about data, so far - the use case

- What will the TTS system be used for?
- What sort of things does it need to say?
- Who will listen to it?
- *plus various technical requirements (computation, memory, platform, latency,...)*

What we have learned about data, so far - a dataset creation pipeline

- Identify a source of data
- Obtain (a lot) more data than we require
- Curate the data
  - define "good data"
  - filter out "bad" data (simplest: discard it ; alternatively: fix/repair)
  - select a dataset of the desired size

- We only considered text data, but the recipe works for speech too (later in the course)

What we have learned about data, so far - requirements for neural approaches

- In simple terms
  - many neural models need a lot more speech data than we could record in the studio
  - *not true for all models (e.g., the one we are using in the assignment)*

- So, for data-hungry models we have no choice but to create a dataset automatically, by
  - Automatically curating (filtering, annotation, selection,...) a dataset from 'found' data

- **REMEMBER: do not do this for the assignment!**
  - You must *only* use purposely-recorded speech (your own + corpora approved by us)

What we have learned about data, so far - requirements for neural approaches

- Is data selection still relevant?
- Yes, in at least two ways:

1. Large-scale curation of ‘found’ speech data still involves selecting “good” data
   - may lack reliable transcription, other labels, and meta-data

2. Purposely-recorded speech is still useful (in fact, *essential* for most commercial products)
   - generally much higher quality
   - we can control everything: speaker identity, content, speaking style, ...
   - crucially, we have the (ethical *and* legal) right to use it

What we have learned about evaluation, so far

- Methods for synthesis have rapidly advanced
  - yet approaches to evaluation have **barely changed**

- Evaluation is even more relevant than before
  - so we'll need to revisit this topic later in the course (whilst reading recent papers)

What we have *not* yet covered properly: **objective evaluation**

- Approaches to subjective evaluation have **barely changed**

- But there *have* been advances in **objective (instrumental) evaluation**
  - the videos for Module 5 are outdated
  - we will need to come back to this topic later in the course

  - objective measures are based on properties of speech signals
    - *some* of which we might obtain through classical speech signal processing

Orientation

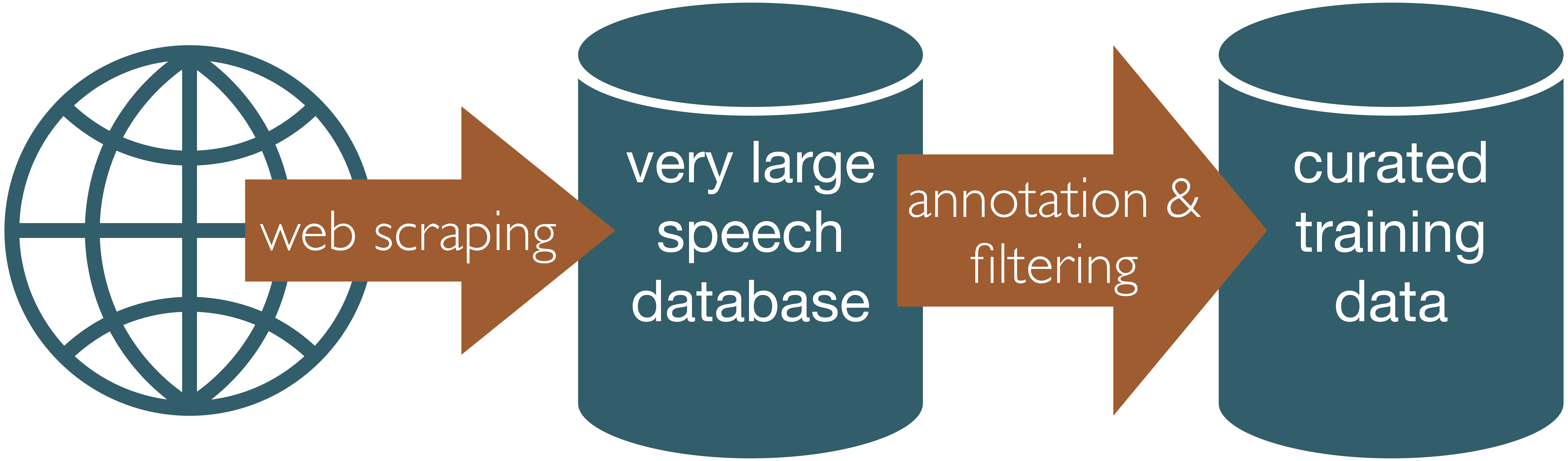
- Modules 2 & 3 ✓
  - Unit selection speech synthesis
- Modules 4 & 5 ✓
  - The database
  - Evaluation
- Module 6

Orientation

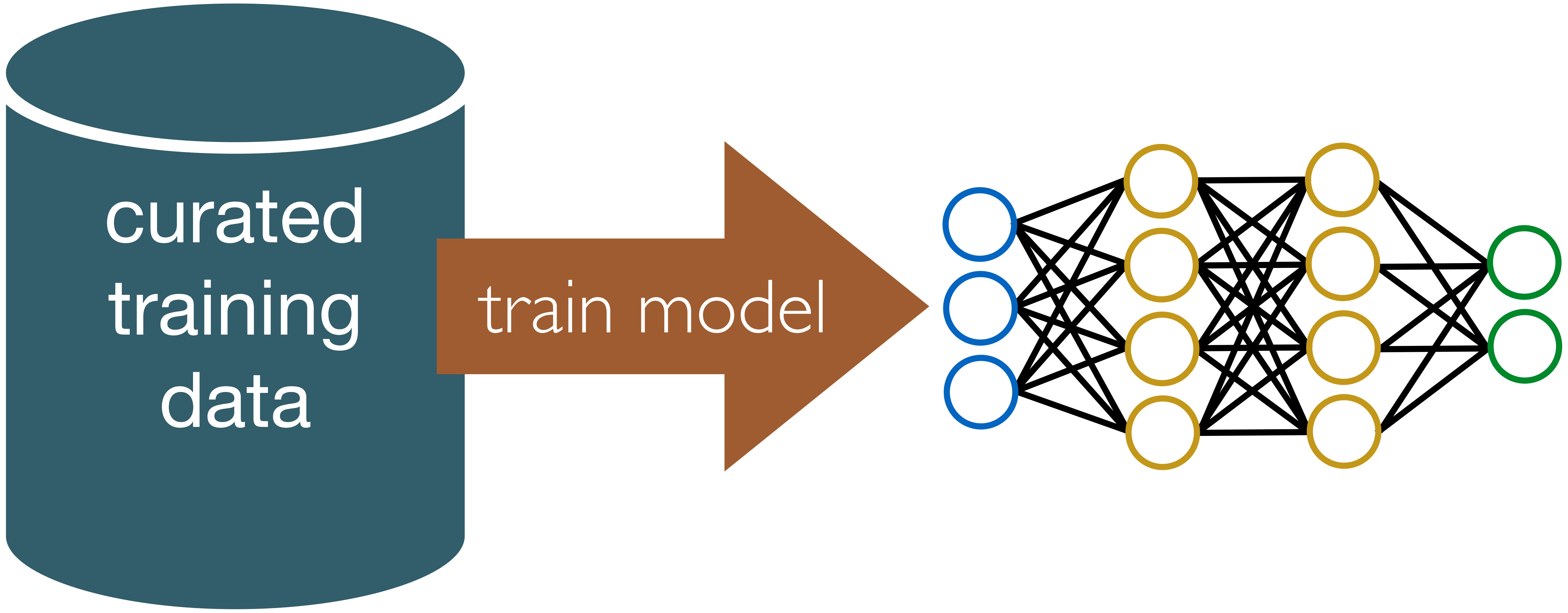
- Module 6 (today's class)
  - Parameterising speech
  - In other words, **representing** the speech signal

- A 'deep dive' into F0 estimation
  - F0 is a key feature we want to extract
  - RAPT is a classical example of a signal processing algorithm

Representations are employed when creating the training data



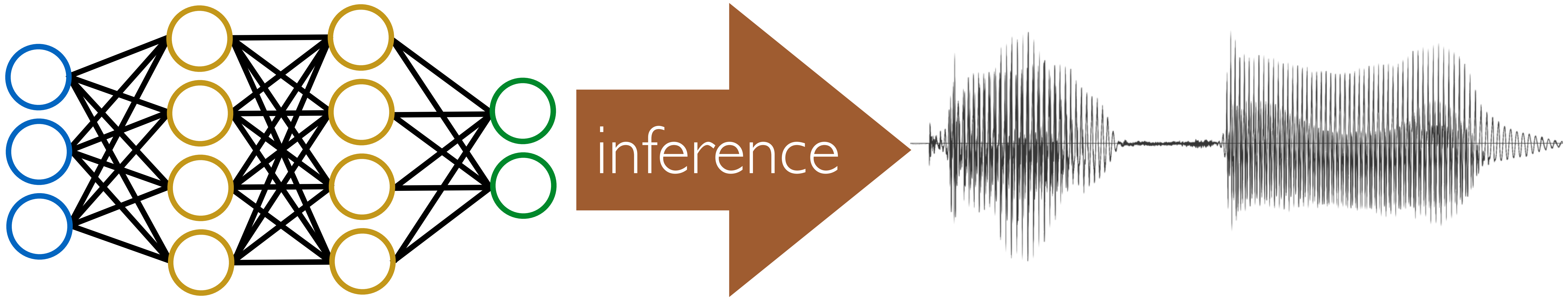
Representations are employed when training a model



curated training data

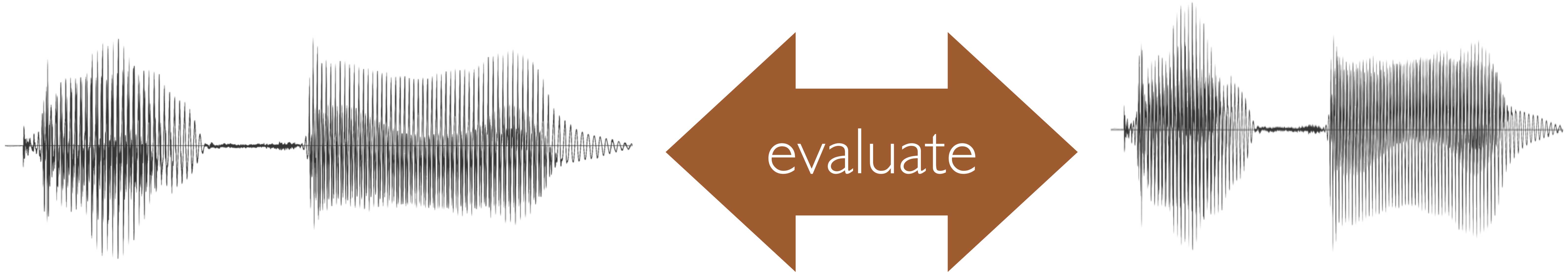
train model

Representations are employed when performing synthesis



inference

Representations are employed when objectively evaluating the output

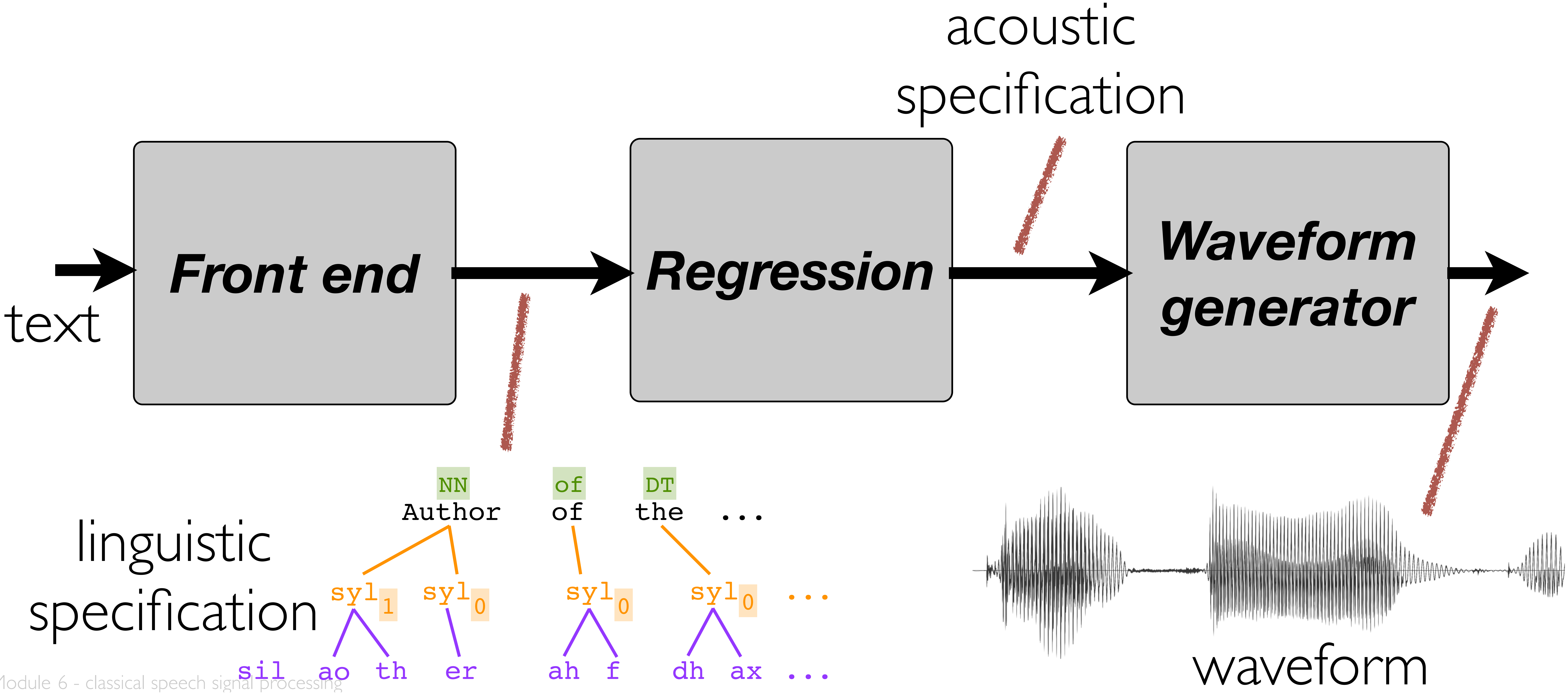


evaluate

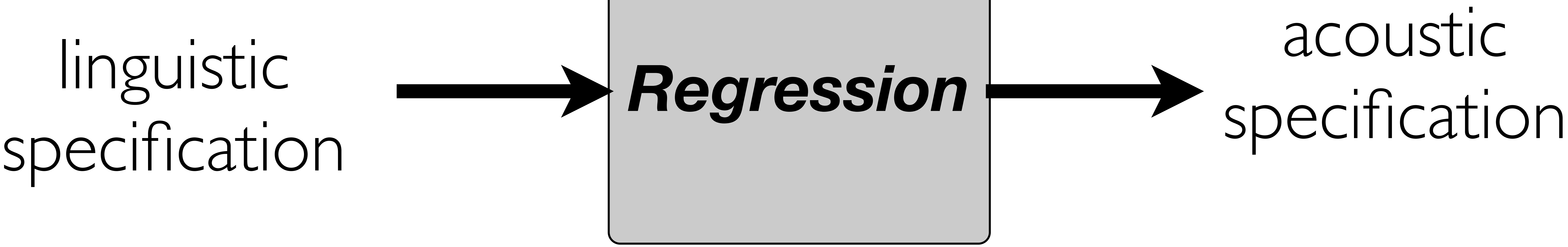
synthetic
speech

natural
speech

Representations are employed everywhere inside a Text-to-Speech **system**



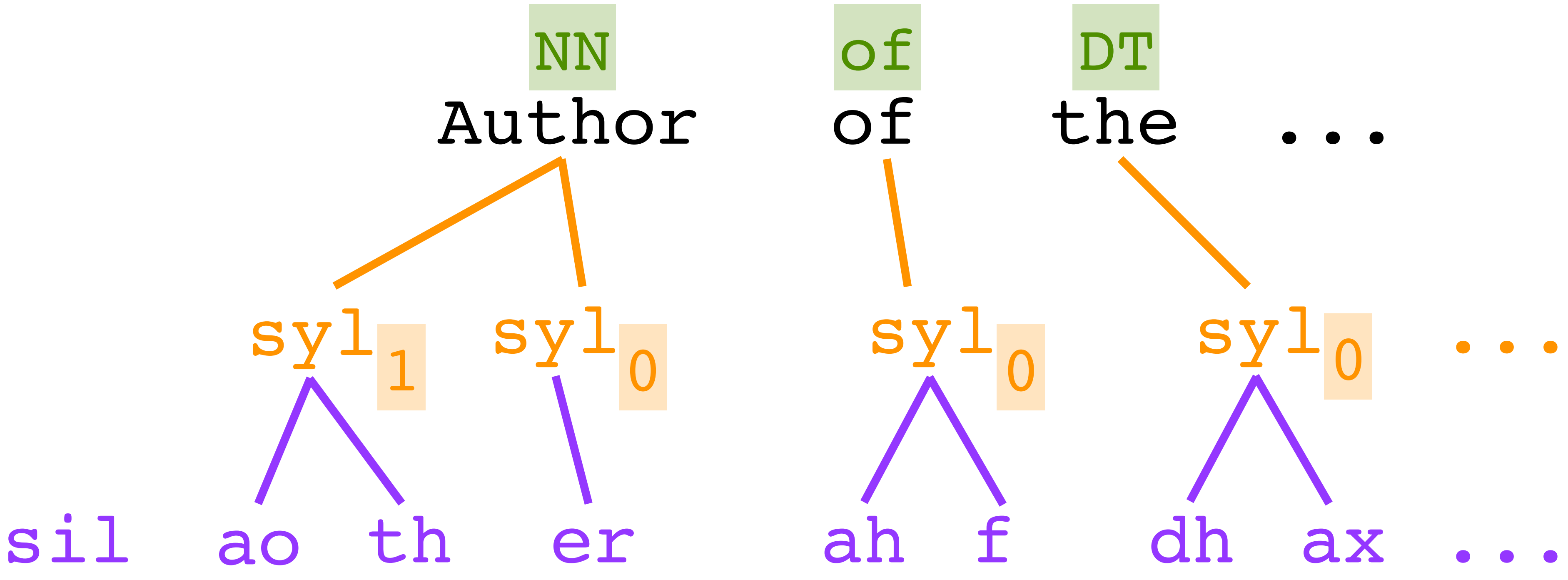
Representations are employed everywhere inside a Text-to-Speech **system**



linguistic specification

Regression

acoustic specification



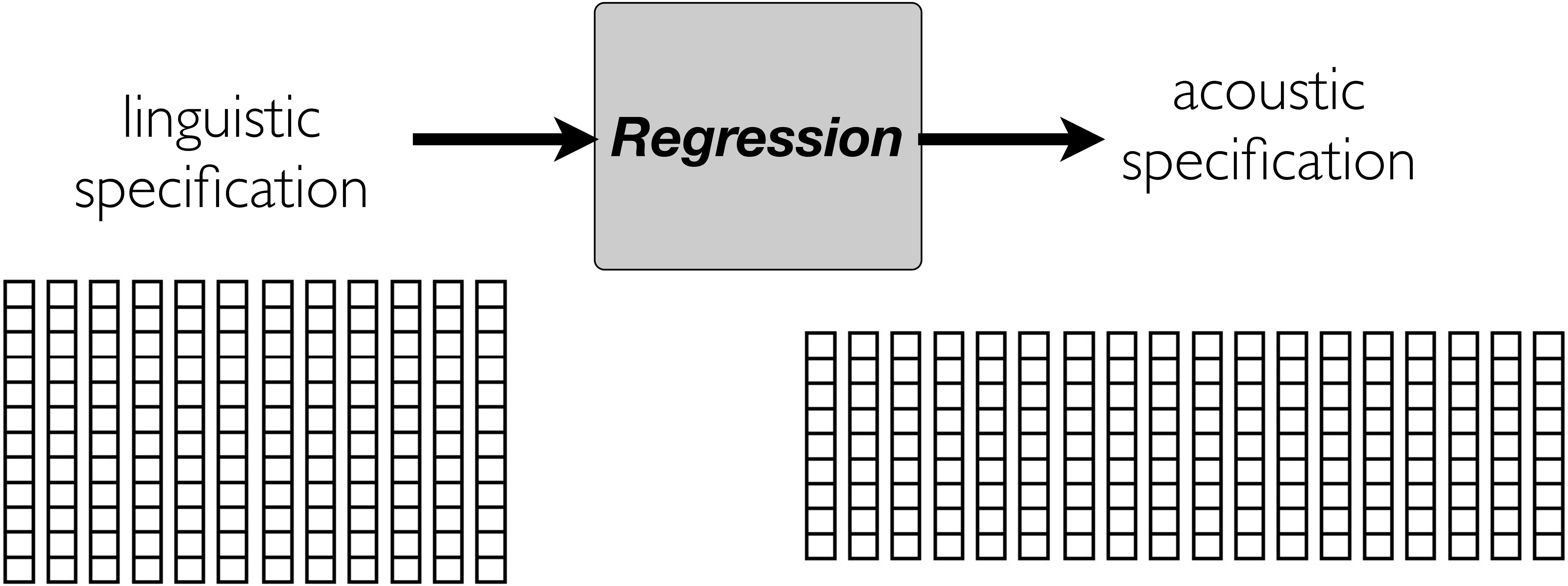
NN of DT

Author of the ...

syl$_1$ syl$_0$ syl$_0$ syl$_0$ ...

sil ao th er ah f dh ax ...

Representations are employed everywhere inside a Text-to-Speech **system**



linguistic specification

Regression

acoustic specification

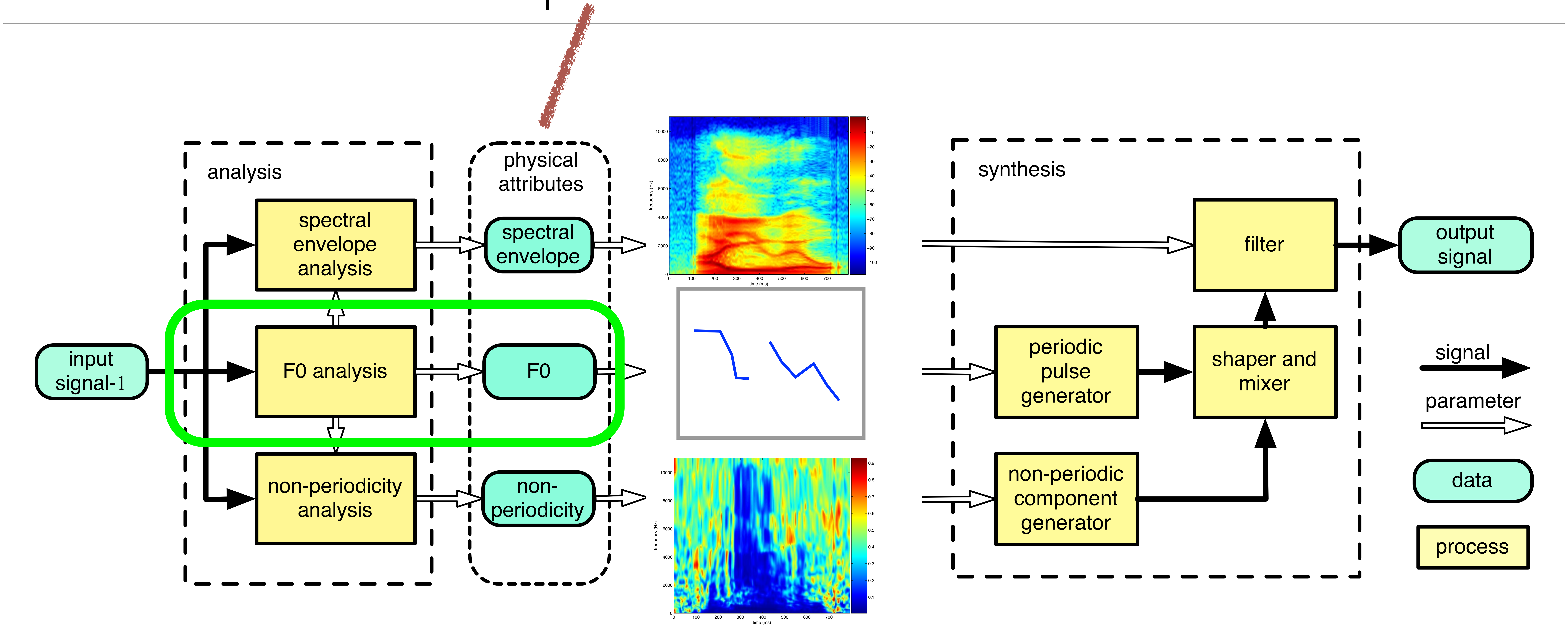
The acoustic specification (i.e., a **representation** of speech)

- Must be able to extract this from the waveform
  - today: using classical speech signal processing
    - e.g., estimating F0, spectral envelope, ...
  - *later in the course: using machine learning*
    - *i.e., learned representations such as audio tokens*

- Must be able to reconstruct the waveform from the acoustic specification
  - today: using classical speech signal processing - the source-filter model
  - *later in the course: using machine learning*
    - *neural vocoders, neural audio codecs*

A classical vocoder

acoustic specification

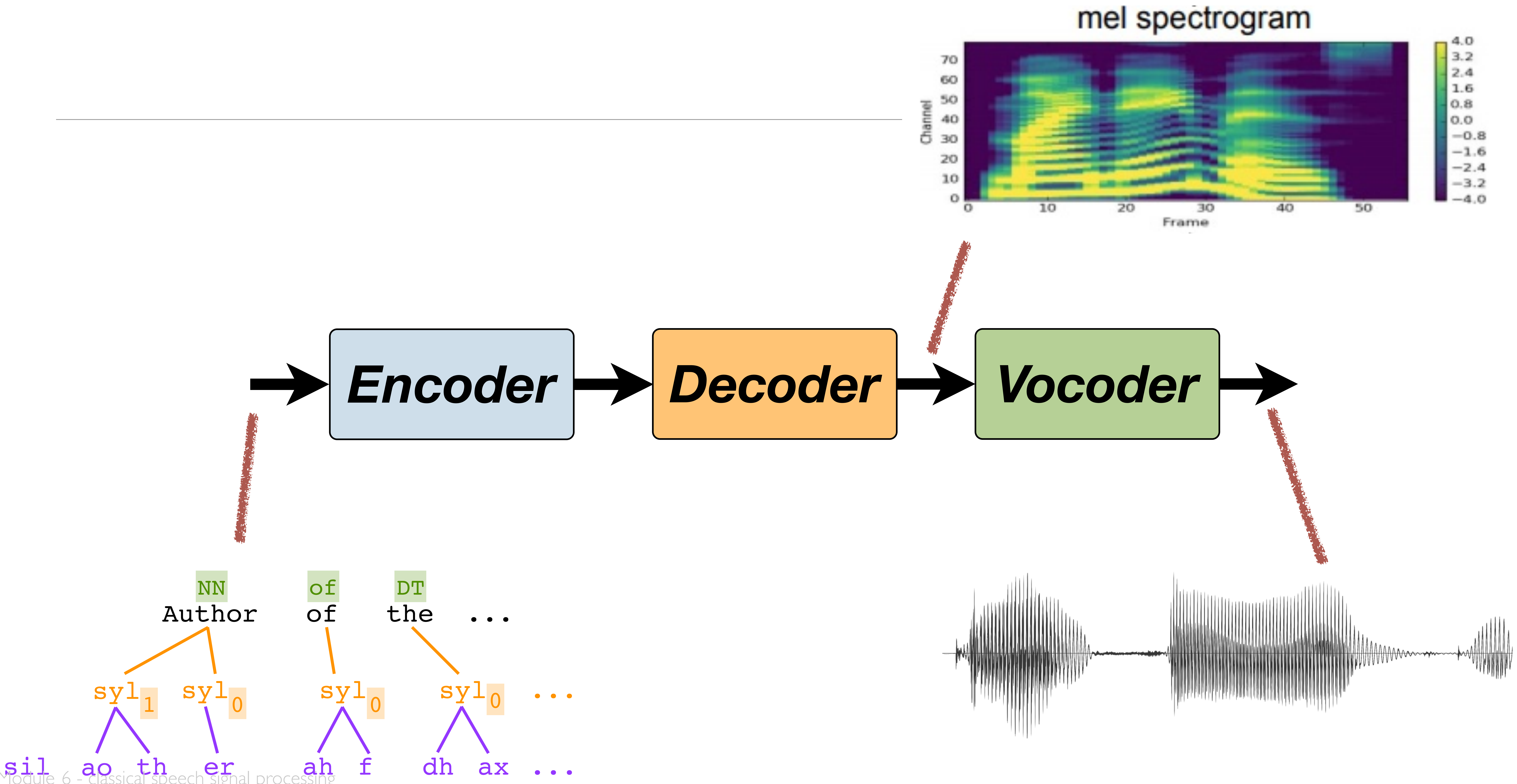


Terminology!

- Representation

- Acoustic specification

- Speech parameters



Encoder

Decoder

Vocoder

F0 estimation (‘pitch tracking’)

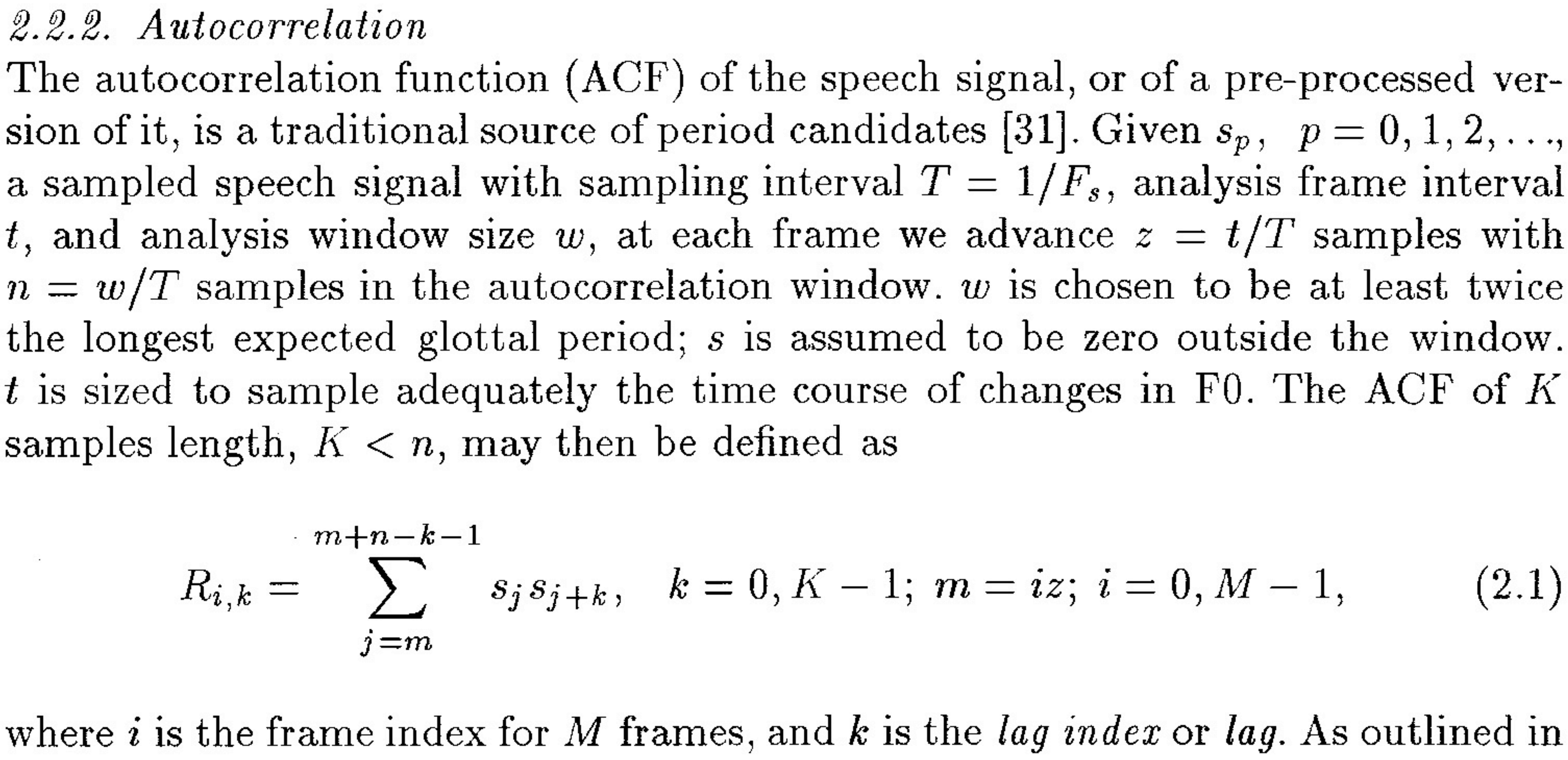
- Discussion points

David Talkin "A Robust Algorithm for Pitch Tracking (RAPT)"

Warm-up

- check your units !
  - time
  - frequency
  - sampling rate
  - sampling interval
  - samples
  - frame
- convert between time and samples
- describe a frame of samples from a longer waveform

What’s the relationship between samples and frames in Equation 2.1 ?



2.2.2. Autocorrelation

The autocorrelation function (ACF) of the speech signal, or of a pre-processed version of it, is a traditional source of period candidates [31]. Given $s_p, \ p = 0, 1, 2, \ldots$, a sampled speech signal with sampling interval $T = 1/F_s$, analysis frame interval t, and analysis window size w, at each frame we advance $z = t/T$ samples with $n = w/T$ samples in the autocorrelation window. w is chosen to be at least twice the longest expected glottal period; s is assumed to be zero outside the window. t is sized to sample adequately the time course of changes in F0. The ACF of K samples length, $K < n$, may then be defined as

$$R_{i,k} = \sum_{j=m}^{m+n-k-1} s_j s_{j+k}, \quad k = 0, K-1; \ m = iz; \ i = 0, M-1, \qquad (2.1)$$

where i is the frame index for M frames, and k is the *lag index* or *lag*. As outlined in

These equations are the *almost* same, except for notation

$$R_{i,k} = \sum_{j=m}^{m+n-k-1} s_j s_{j+k}, \quad k = 0, K-1;\; m = iz;\; i = 0, M-1, \tag{2.1}$$

$$r_t(\tau) = \sum_{j=t+1}^{t+W} x_j x_{j+\tau},$$

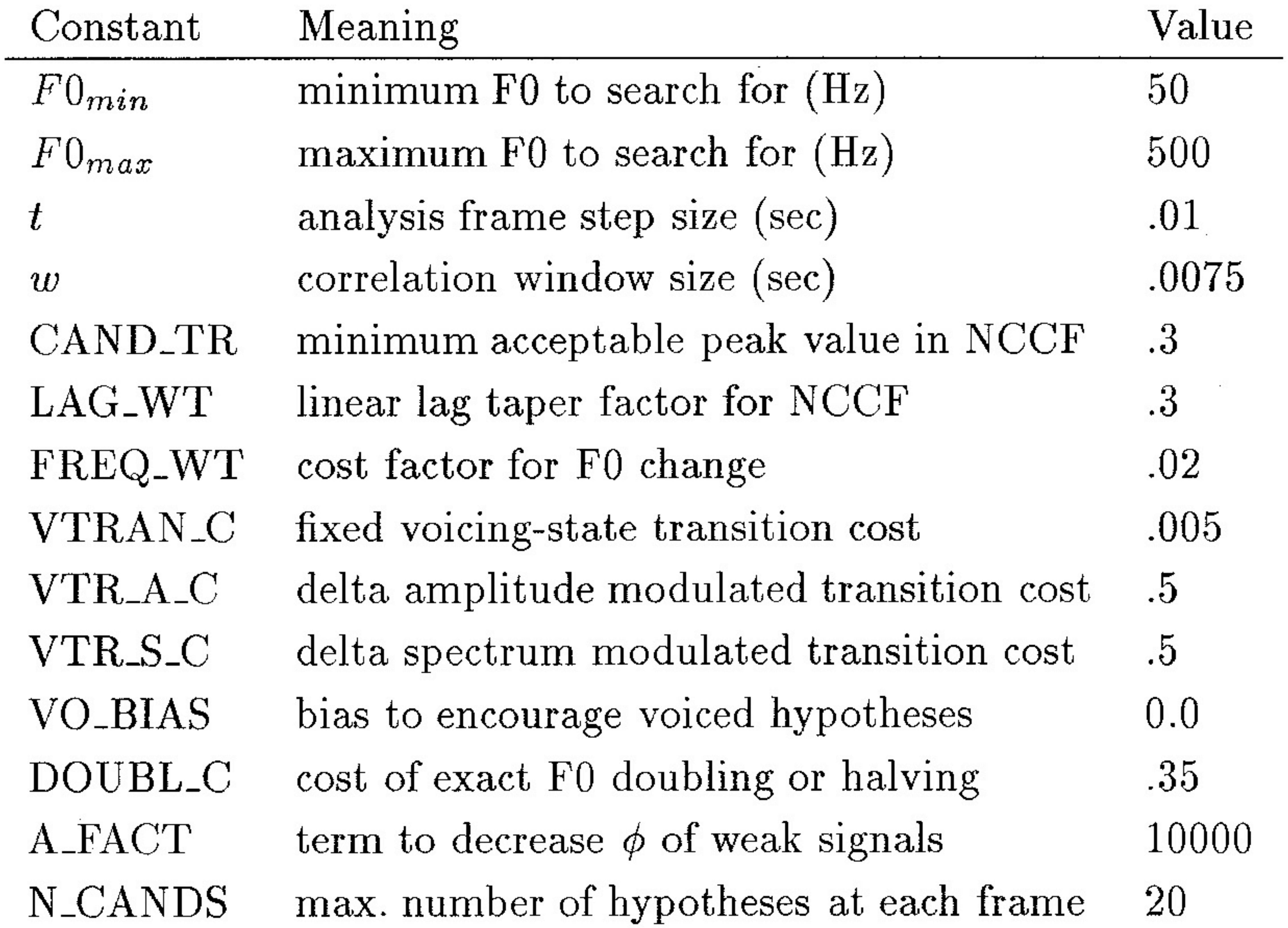
0
198.60
198.70
198.80
198.90
200
100
0
198.6
198.65
198.7
198.75
198.8
198.85
198.9
198.95

0
198.60
198.70
198.80
198.90
200
100
0
198.6
198.65
198.7
198.75
198.8
198.85
198.9
198.95

Draw a diagram that shows candidate generation

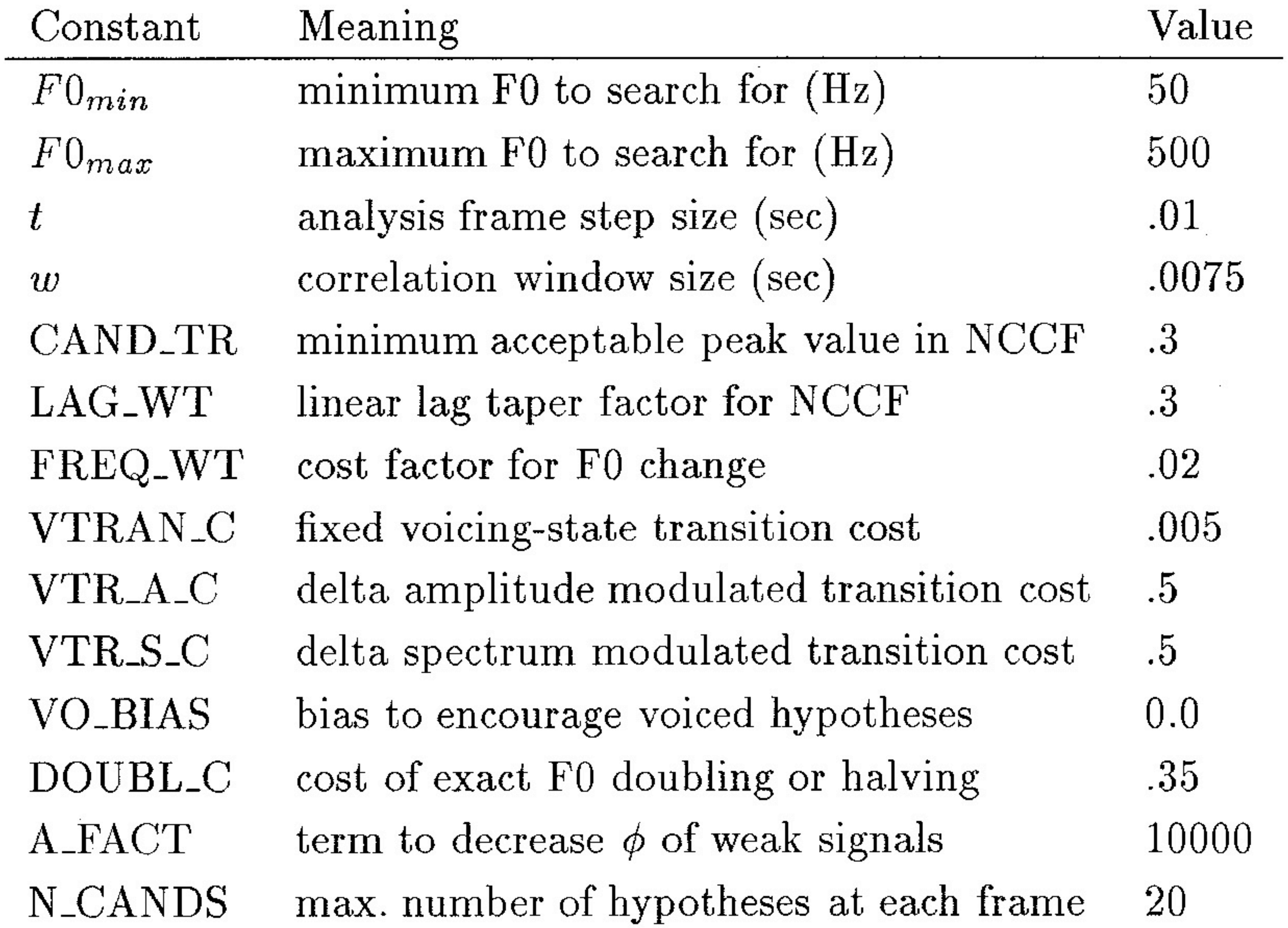
- Hint : start with Figure 2 (the correllogram)

Annotate **N_CANDS** on your diagram



Constant	Meaning	Value
$F0_{min}$	minimum F0 to search for (Hz)	50
$F0_{max}$	maximum F0 to search for (Hz)	500
t	analysis frame step size (sec)	.01
w	correlation window size (sec)	.0075
CAND_TR	minimum acceptable peak value in NCCF	.3
LAG_WT	linear lag taper factor for NCCF	.3
FREQ_WT	cost factor for F0 change	.02
VTRAN_C	fixed voicing-state transition cost	.005
VTR_A_C	delta amplitude modulated transition cost	.5
VTR_S_C	delta spectrum modulated transition cost	.5
VO_BIAS	bias to encourage voiced hypotheses	0.0
DOUBL_C	cost of exact F0 doubling or halving	.35
A_FACT	term to decrease ϕ of weak signals	10000
N_CANDS	max. number of hypotheses at each frame	20

Find a diagram in the slides on which you can annotate **CAND_TR**



Constant	Meaning	Value
$F0_{min}$	minimum F0 to search for (Hz)	50
$F0_{max}$	maximum F0 to search for (Hz)	500
t	analysis frame step size (sec)	.01
w	correlation window size (sec)	.0075
CAND_TR	minimum acceptable peak value in NCCF	.3
LAG_WT	linear lag taper factor for NCCF	.3
FREQ_WT	cost factor for F0 change	.02
VTRAN_C	fixed voicing-state transition cost	.005
VTR_A_C	delta amplitude modulated transition cost	.5
VTR_S_C	delta spectrum modulated transition cost	.5
VO_BIAS	bias to encourage voiced hypotheses	0.0
DOUBL_C	cost of exact F0 doubling or halving	.35
A_FACT	term to decrease ϕ of weak signals	10000
N_CANDS	max. number of hypotheses at each frame	20

Draw a diagram describing the dynamic programming

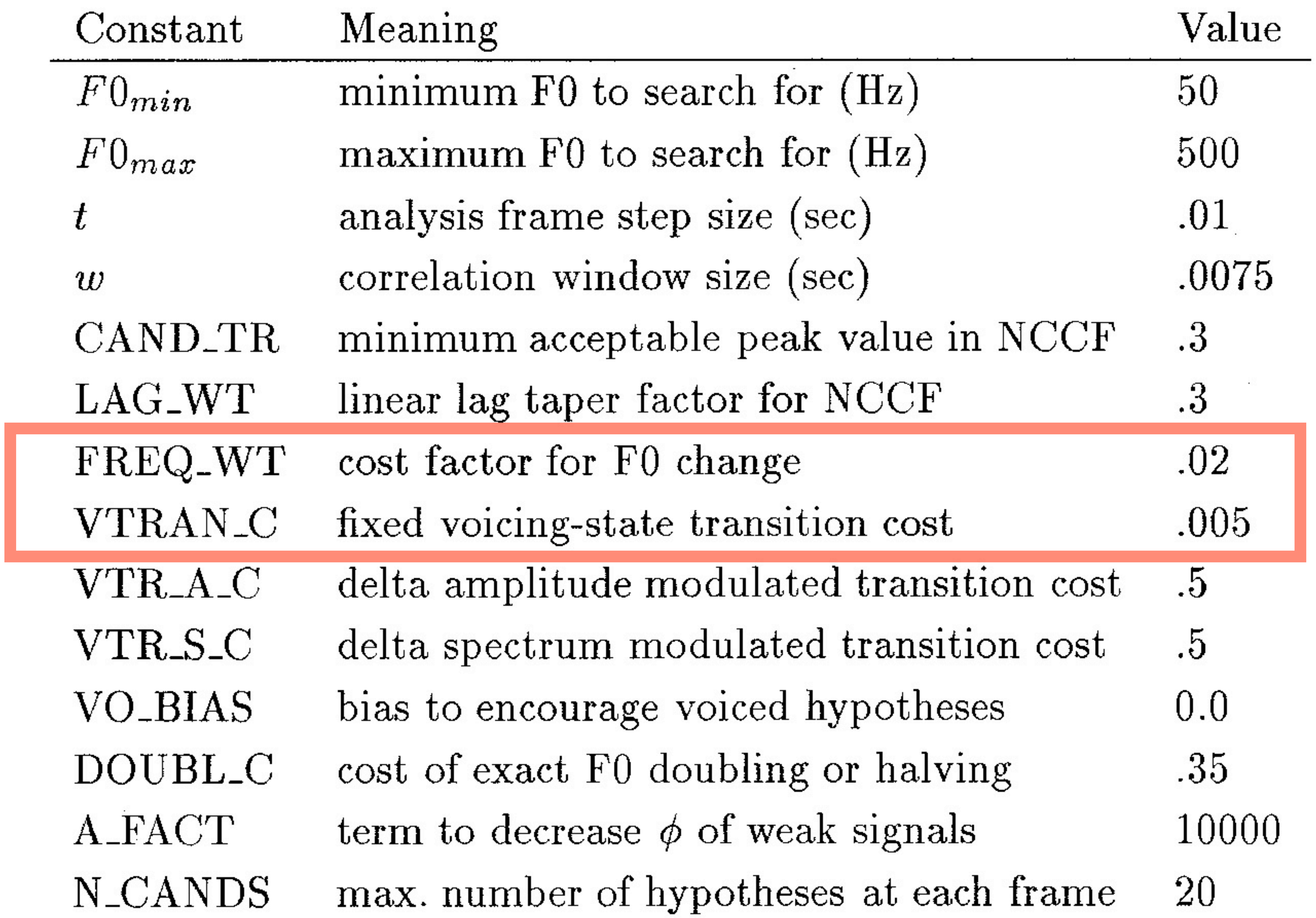
- What are the states?
  - and how many are there?

- What are the transitions?

- What is the local cost?
  - Hint: it’s different for voiced vs unvoiced candidates

- What is the transition cost?
  - Hint: it depends on voicing status

Annotate your diagram describing the dynamic programming with



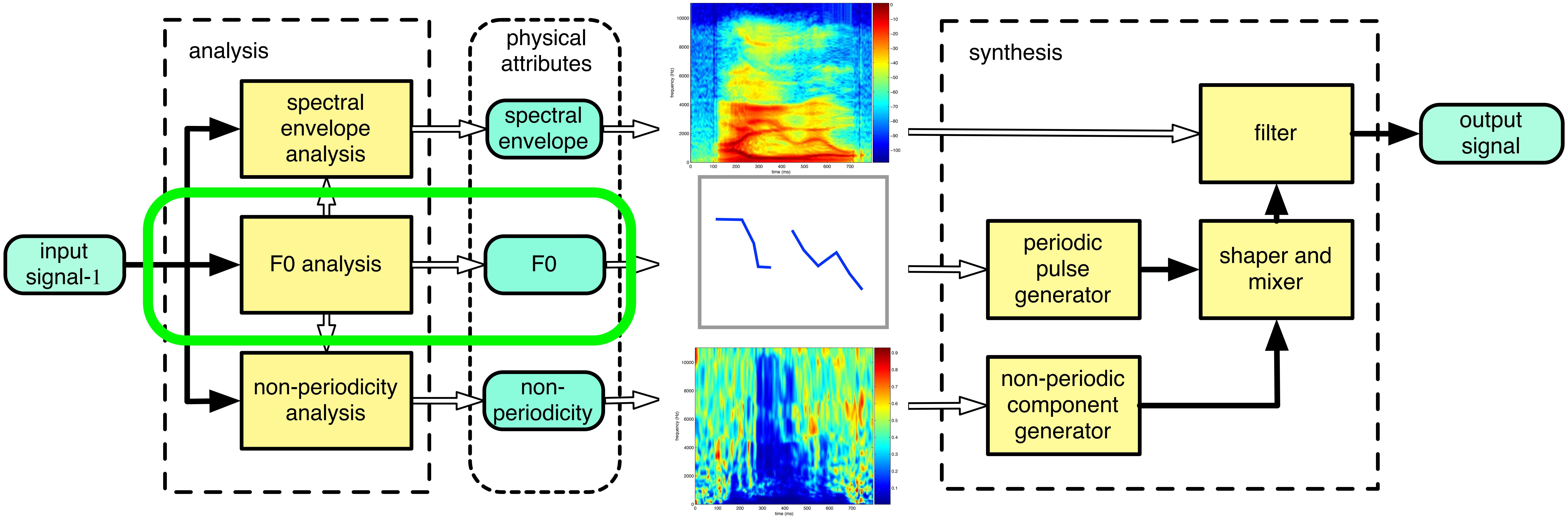
Constant	Meaning	Value
$F0_{min}$	minimum F0 to search for (Hz)	50
$F0_{max}$	maximum F0 to search for (Hz)	500
t	analysis frame step size (sec)	.01
w	correlation window size (sec)	.0075
CAND_TR	minimum acceptable peak value in NCCF	.3
LAG_WT	linear lag taper factor for NCCF	.3
FREQ_WT	cost factor for F0 change	.02
VTRAN_C	fixed voicing-state transition cost	.005
VTR_A_C	delta amplitude modulated transition cost	.5
VTR_S_C	delta spectrum modulated transition cost	.5
VO_BIAS	bias to encourage voiced hypotheses	0.0
DOUBL_C	cost of exact F0 doubling or halving	.35
A_FACT	term to decrease ϕ of weak signals	10000
N_CANDS	max. number of hypotheses at each frame	20

What next?

- We have decomposed speech into
  - F0, plus a V/UV decision
  - smooth spectral envelope, parameterised as the mel-cepstrum
  - band aperiodicity parameters
- We've seen how to reconstruct the waveform

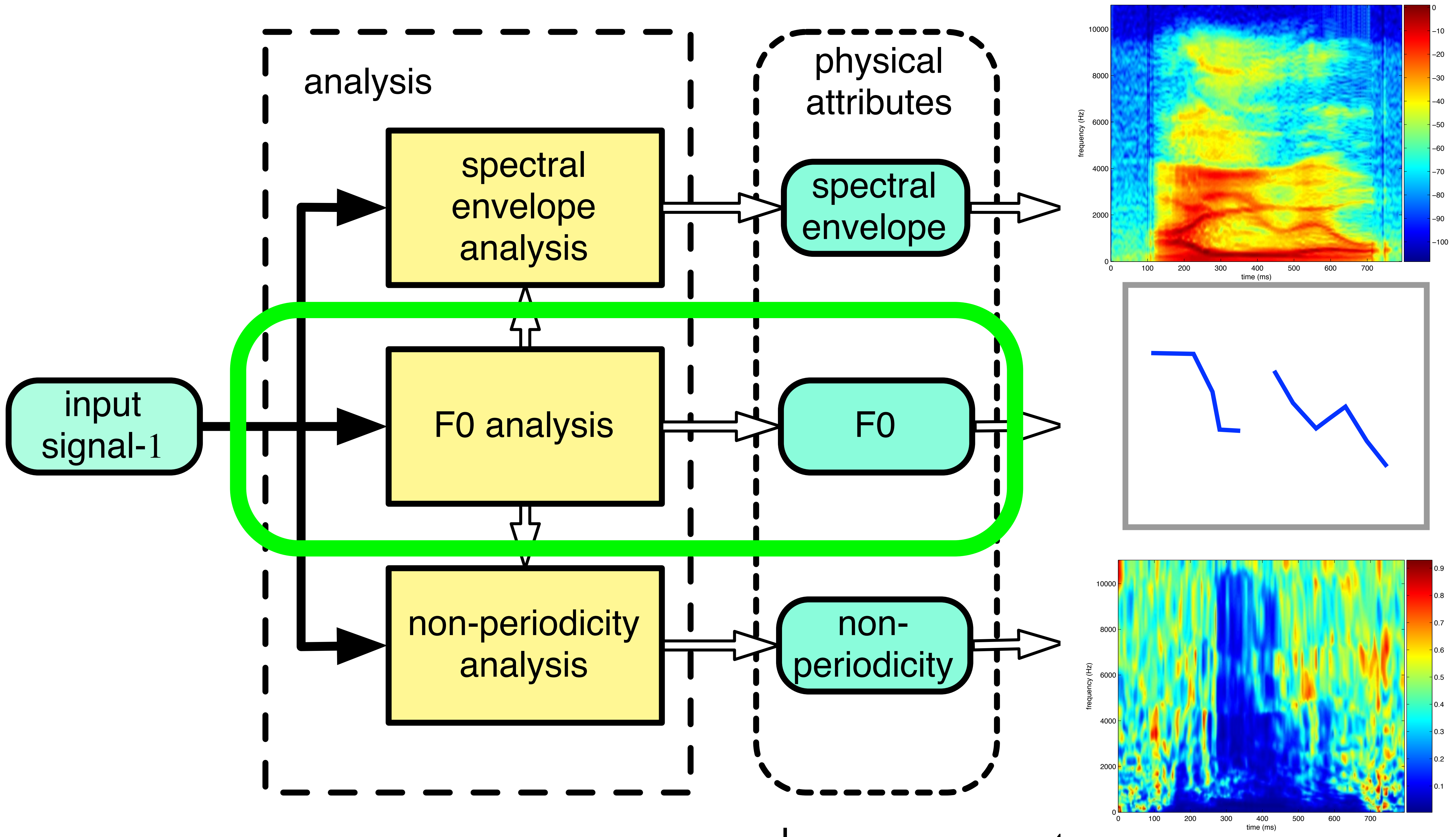
- Now we can train a **statistical model**, using the vocoder's *speech parameters* as the model's *acoustic specification*

What next?

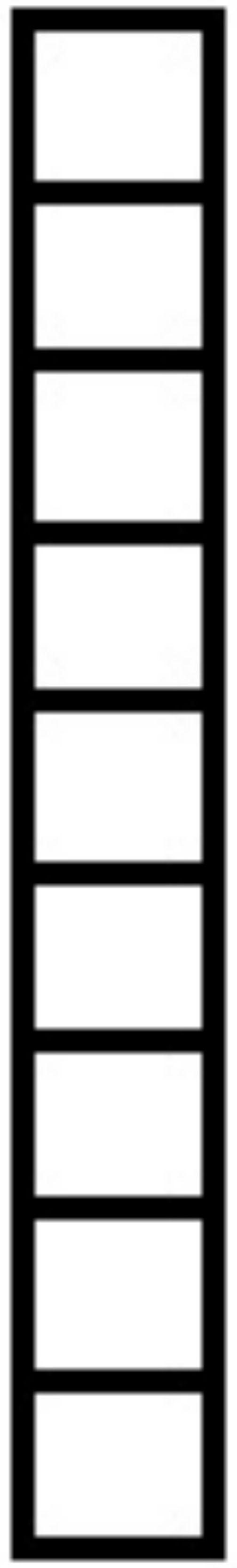


Figures: Hideki Kawahara

Speech parameters

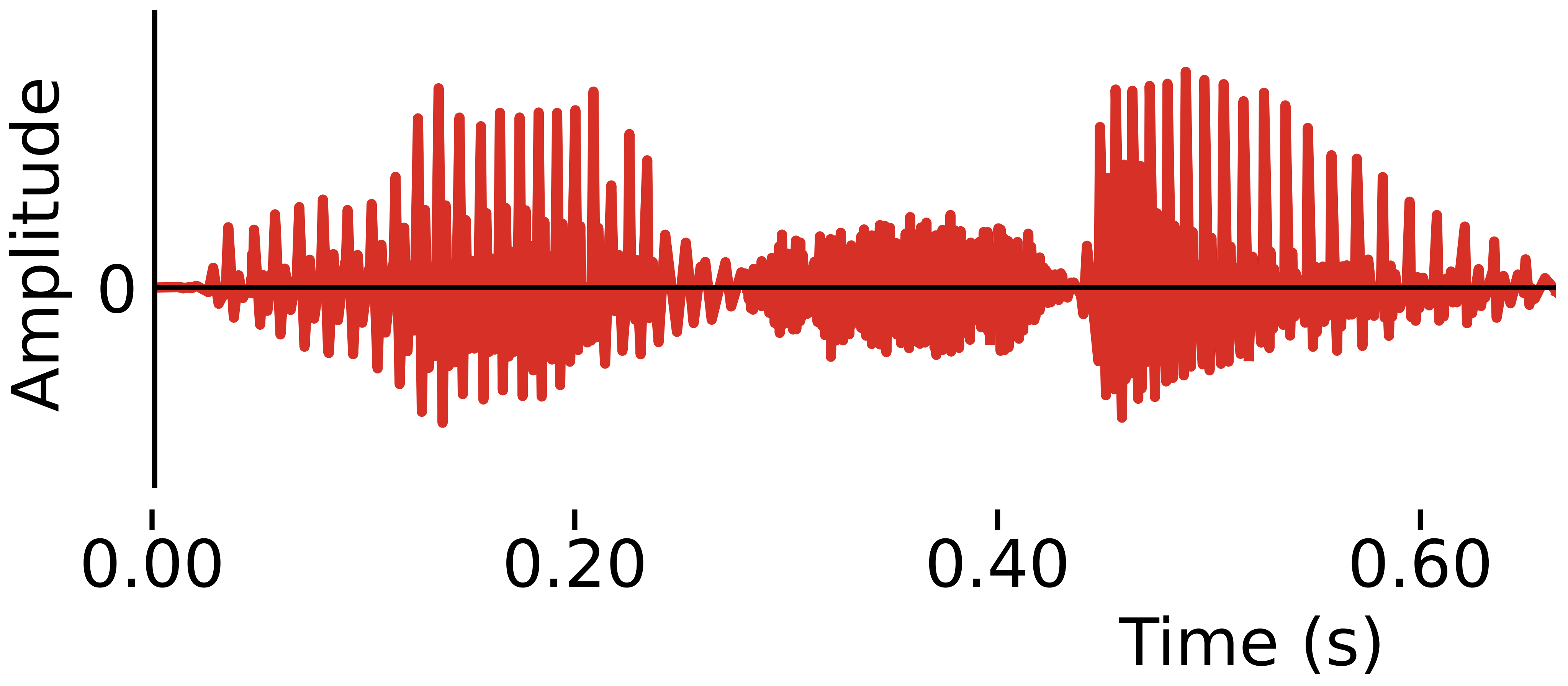


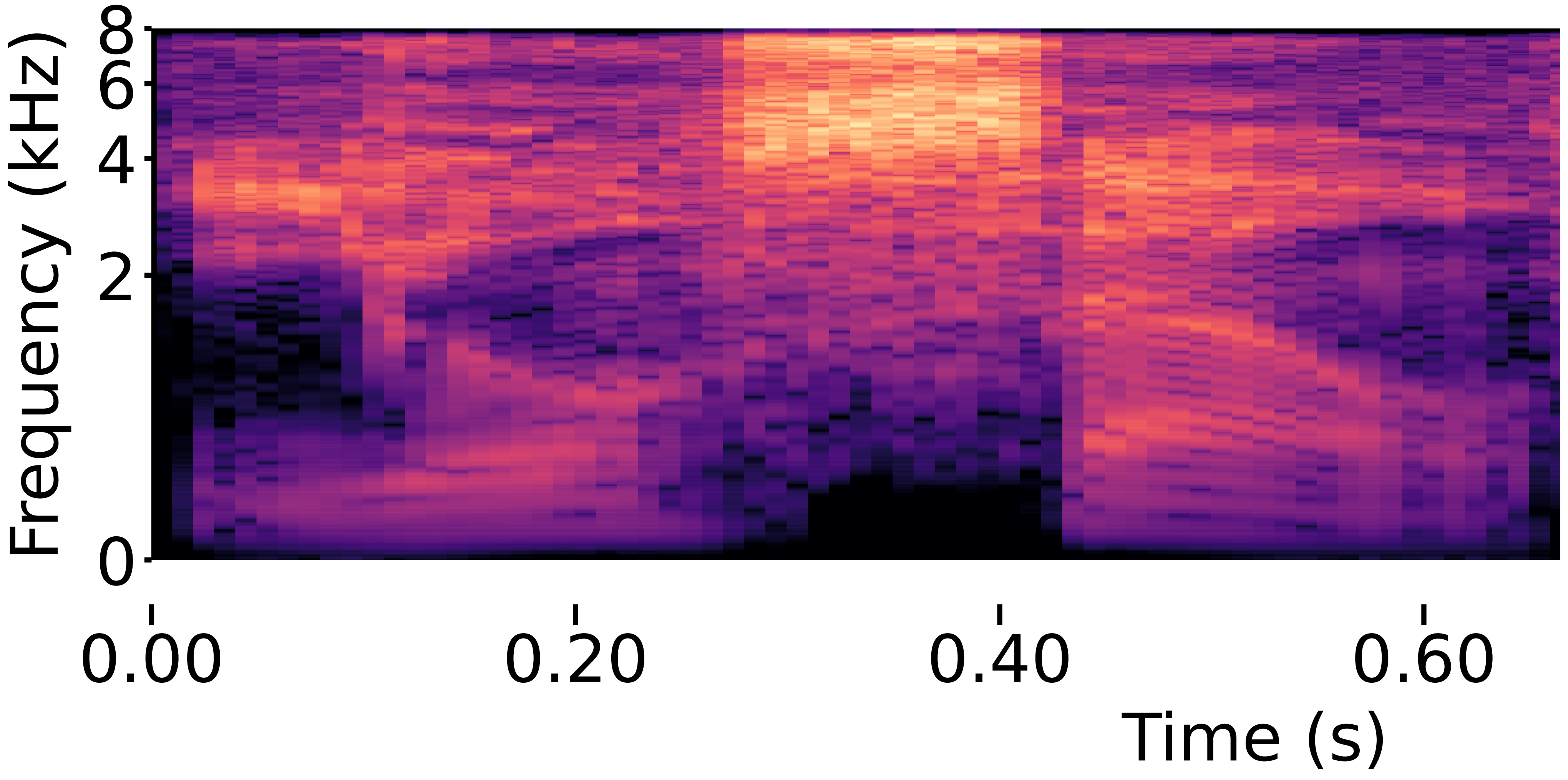
speech parameters



feature vector

Speech parameters in more recent approaches


Amplitude
Time (s)


Frequency (kHz)
Time (s)