Specialization Module

Speech Technology

Timo Baumann
baumann@informatik.uni-hamburg.de

Universität Hamburg, Department of Informatics
Natural Language Systems Group

Parametric Speech Synthesis:
Vocoding & HMM parameter estimation

Process diagram of Speech Synthesis

input

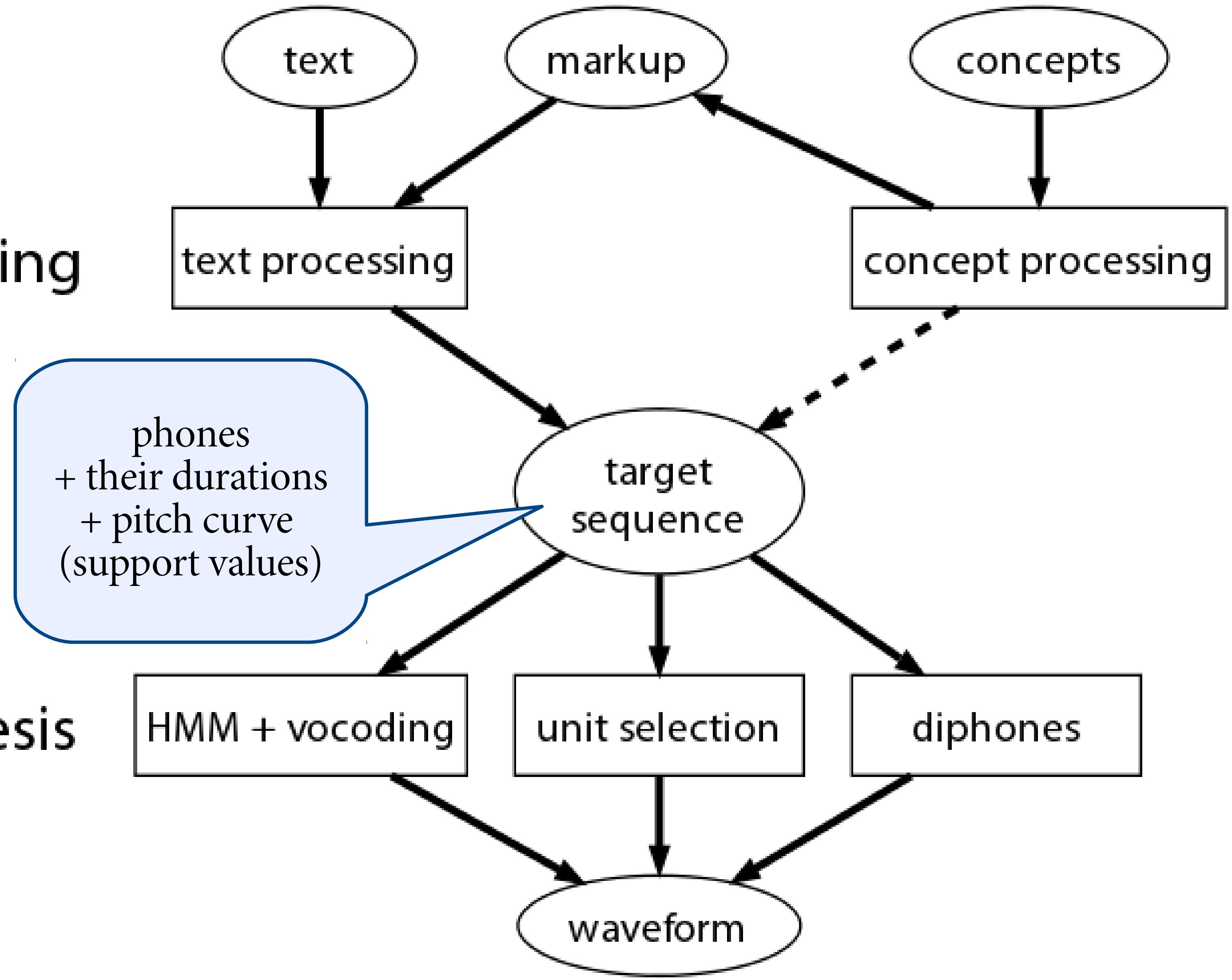
✓ symbolic processing

problem: text is missing detail

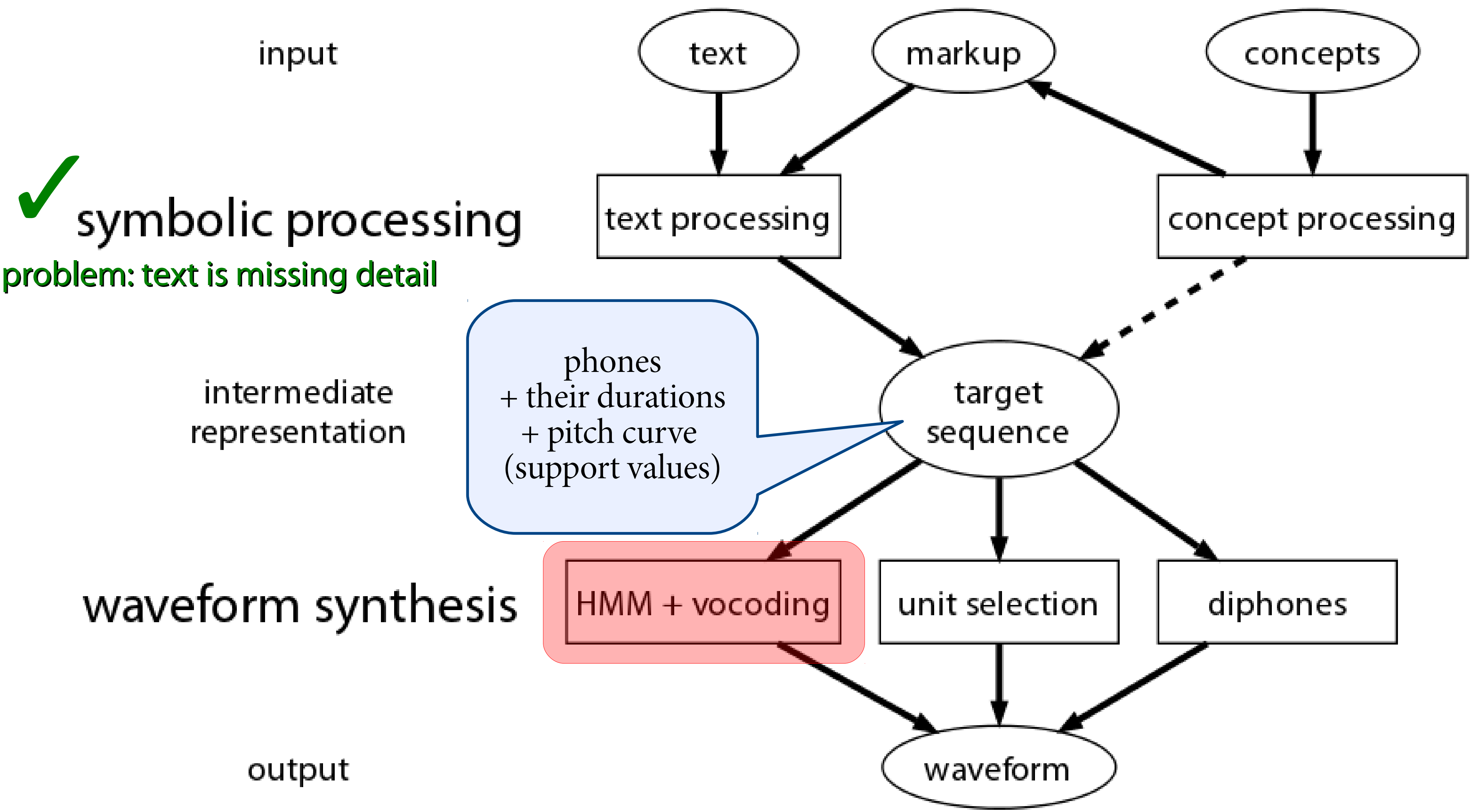
intermediate
representation

waveform synthesis

output

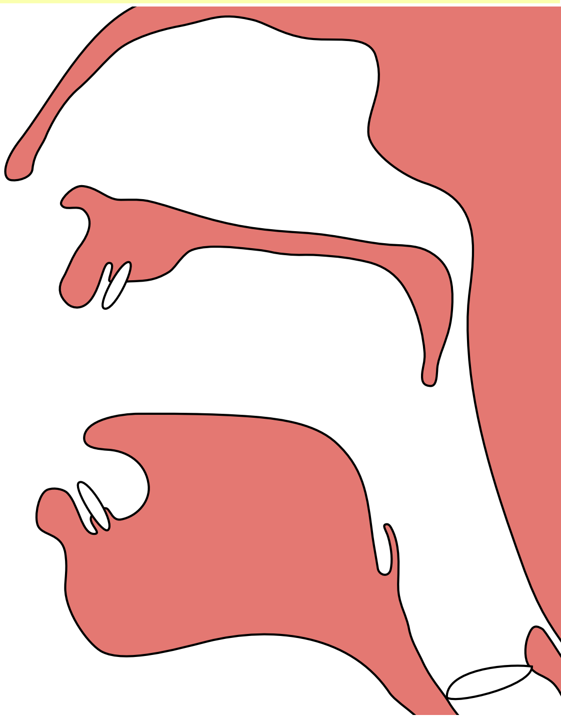


Process diagram of Speech Synthesis

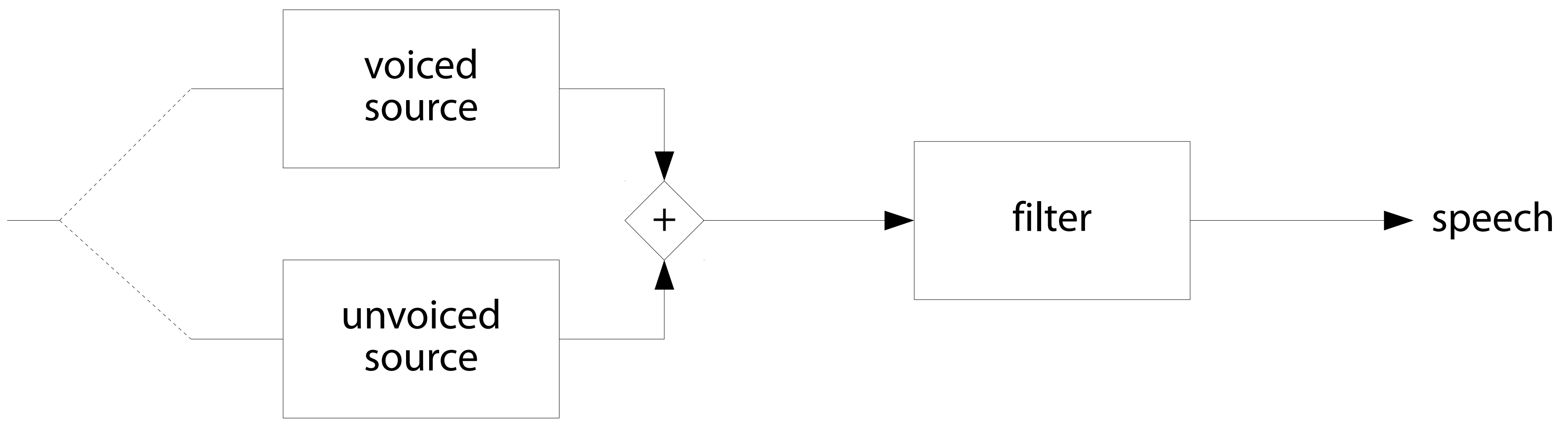


Idea: Filtering

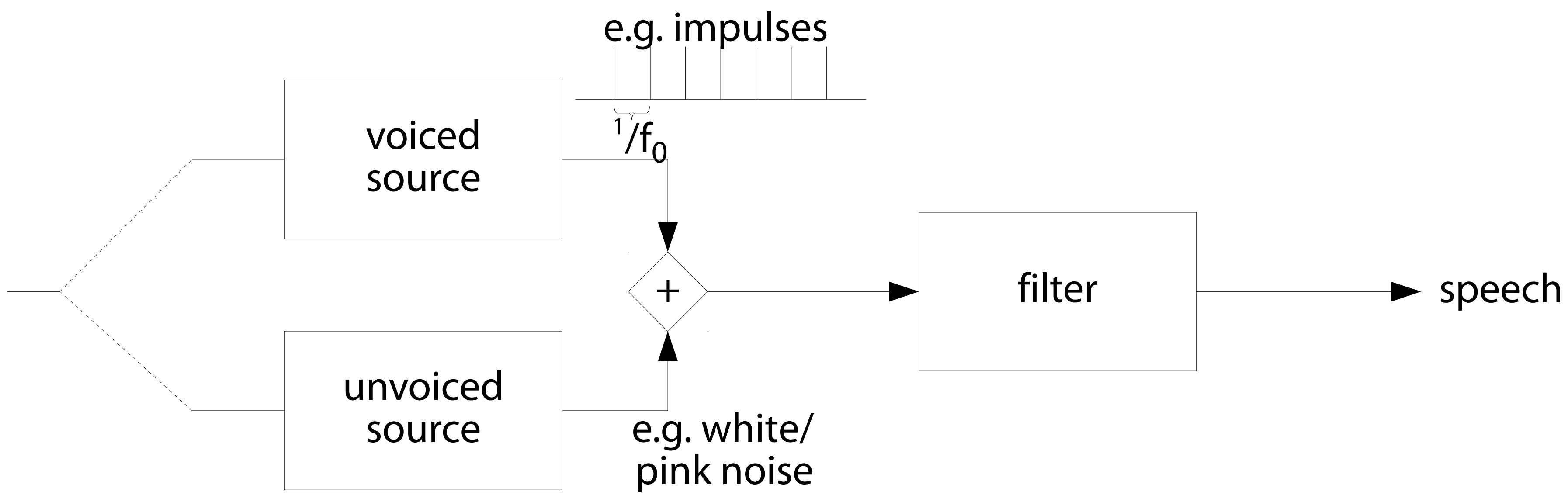
- the glottal folds produce a primary (saw-tooth-like) signal
  - rich in overtones/harmonics
- the vocal tract acts as a (frequency) filter
  - mostly attenuation
- if we know primary signal and filter parameters, we just need to combine the two



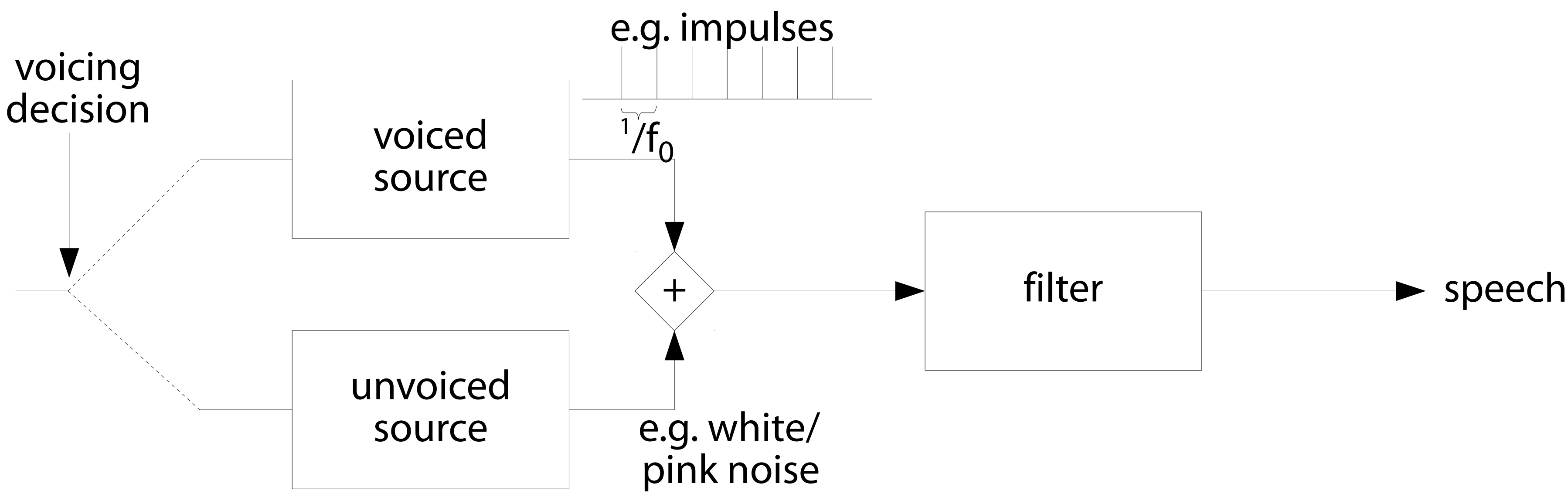
A Simple Vocoder Design



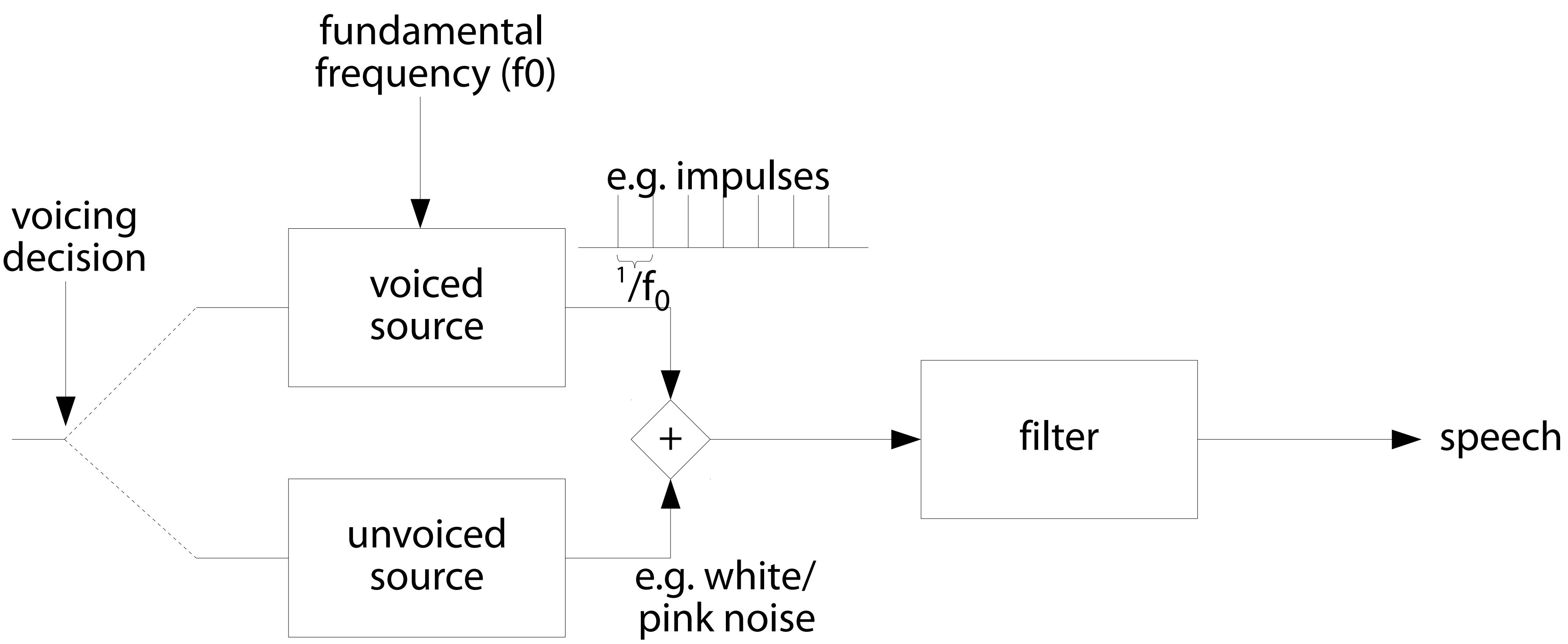
A Simple Vocoder Design



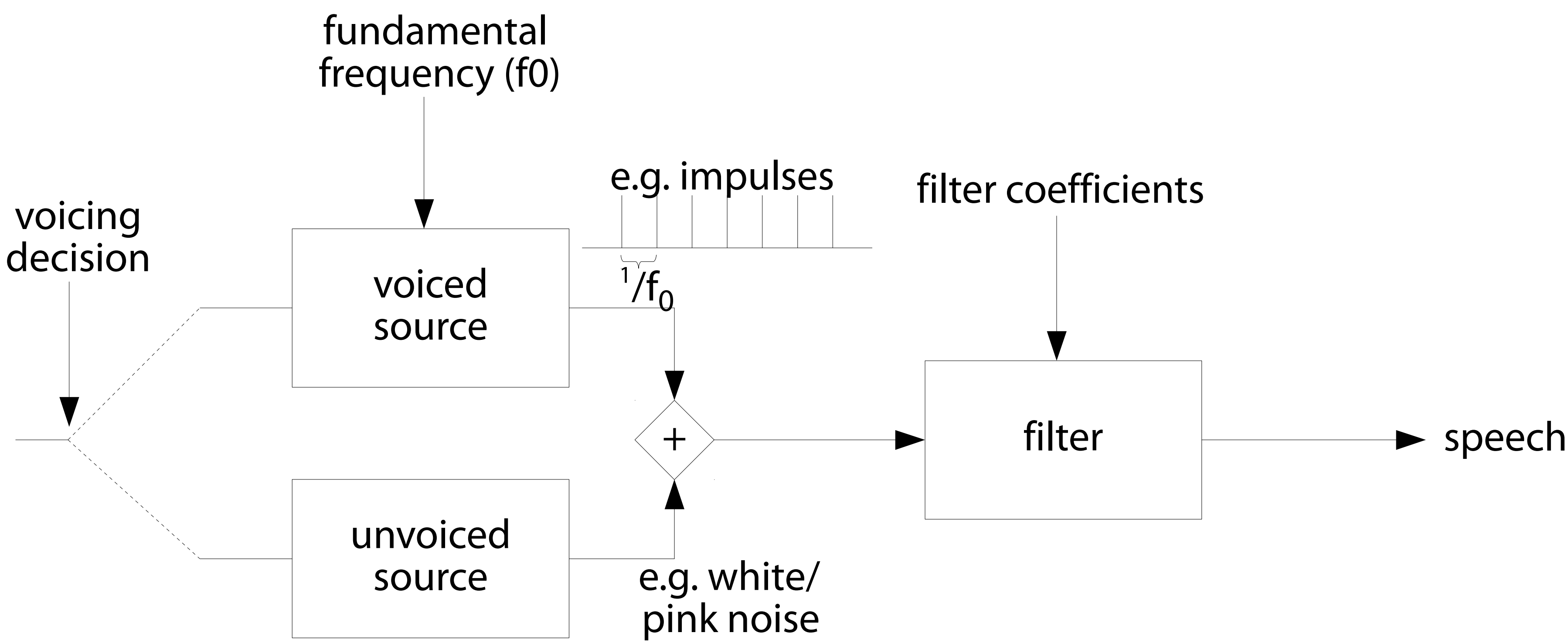
A Simple Vocoder Design



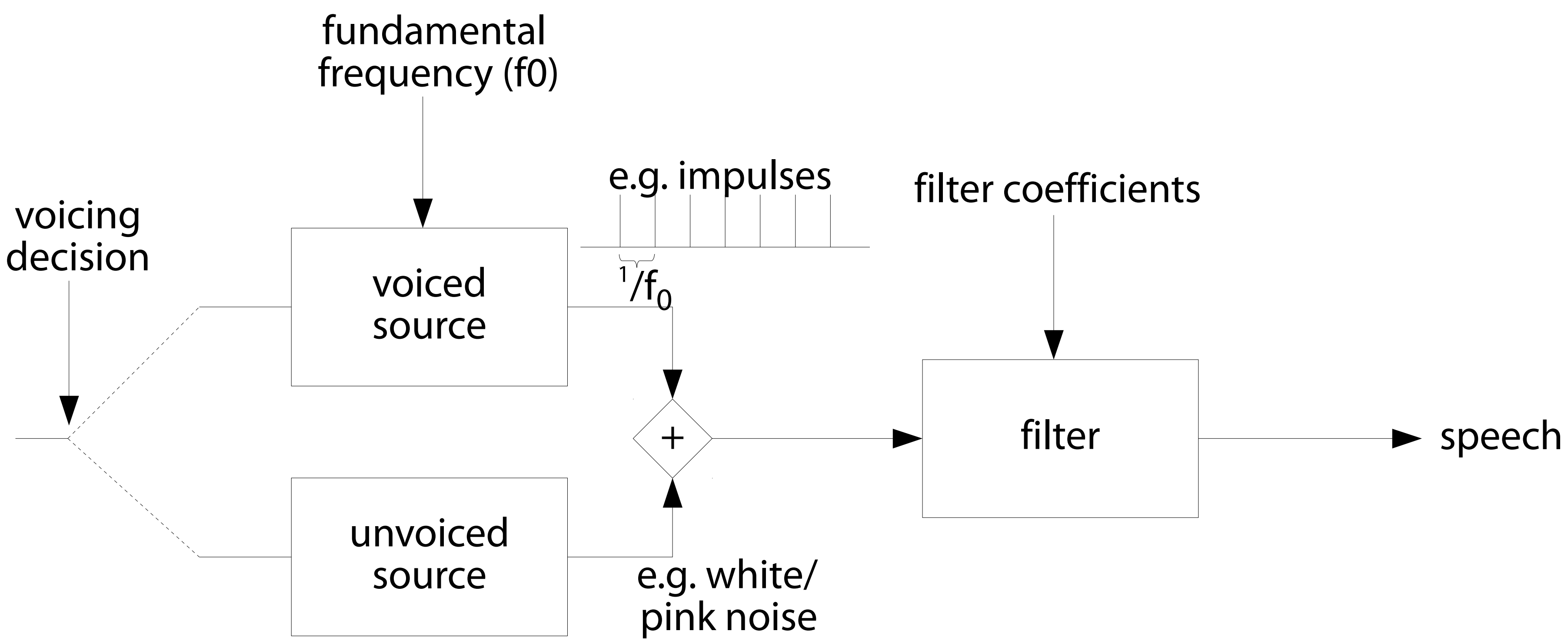
A Simple Vocoder Design



A Simple Vocoder Design

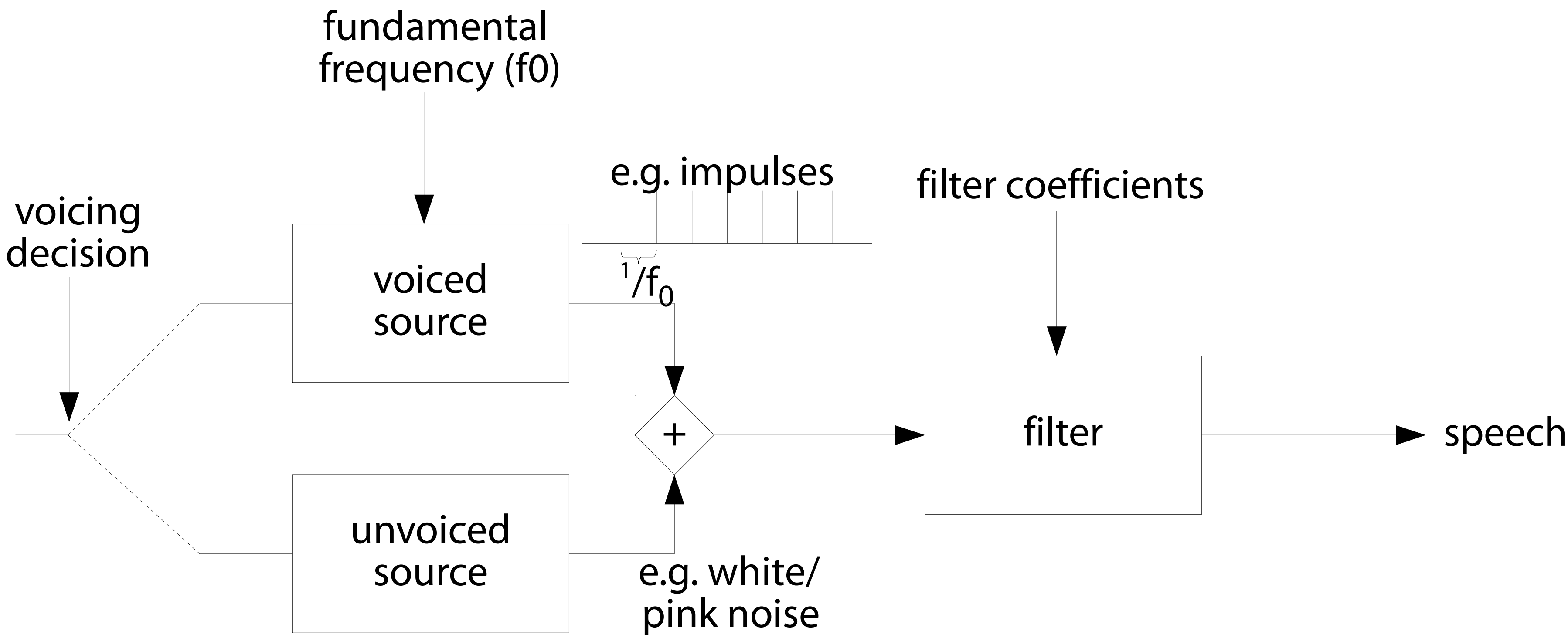


A Simple Vocoder Design



- few parameters in the standard model
  - still, good parameters are the bottleneck (remember eSpeak?)
- extensions: mixed voicing, model for primary signal, ...

A Simple Vocoder Design



- few parameters in the standard model
  - still, good parameters are the bottleneck (remember eSpeak?)
- extensions: mixed voicing, model for primary signal, ...

Parameters for Speech Synthesis

- previously for recognition:
  - reduce signal to a more compact representation
  - conventionally: „acoustic-phonetic“ parameters like MFCCs
  - rizing: parameters optimized with NNs
- for speech synthesis:
  - design a vocoder that allows for good re-synthesis performance from parameter streams
  - old-school: rule-based generation of parameters from target sequence
  - current: HMM-based generation of parameter streams
  - rizing: NN-based generation of parameter streams

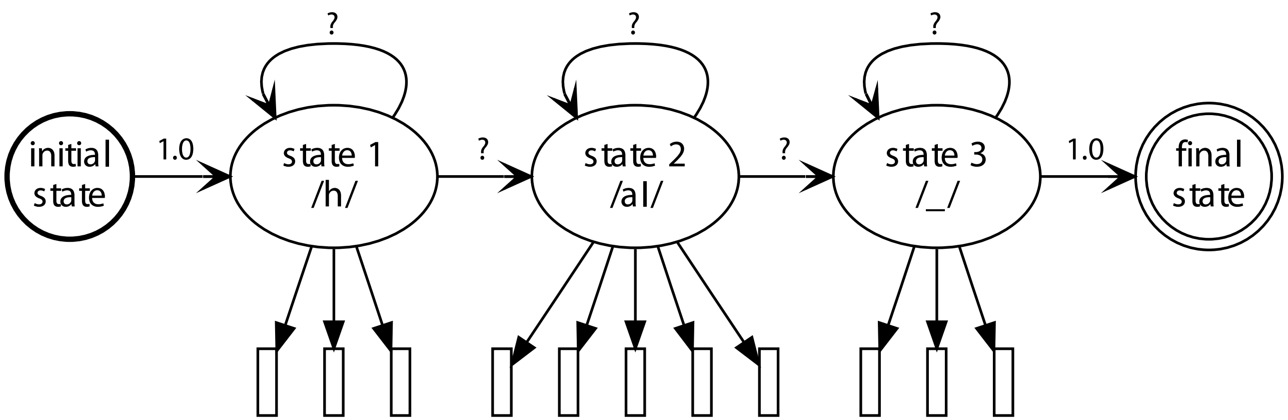
Main Difference Between Recognition vs. Synthesis

Main Difference Between Recognition vs. Synthesis

we know what to say but we don't know what to understand

- search is necessary for speech recognition
  - HMMs are excellent for search, RNNs are still comparatively harder to train
- no search is required for speech synthesis
  - we already know the state sequence (from target sequence)
  - all we want is to find a likely parameter emission sequence to feed to the synthesizer
  - optimal emissions given a state sequence can be found by solving a linear equation (details e.g. in Taylor, 2009)
    - much cheaper than search!!

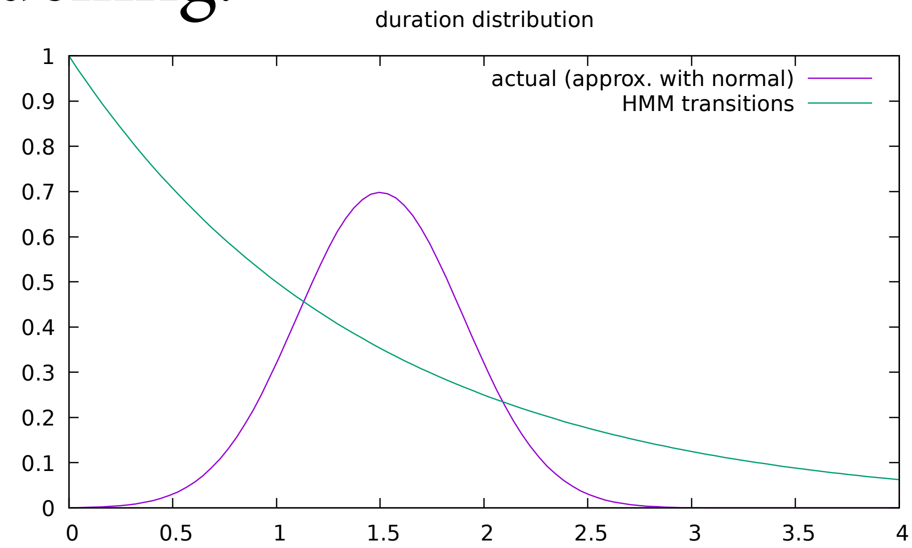
HMMs for Parameter Estimation



- challenges:
  - estimate emission parameters (already solved for recognition)
  - HMMs bad at duration modelling
    - good enough to accept speech timing, but too bad to generate
  - „most likely“ emission is always at μ – is that good?

State Duration Modelling

- HMMs are bad at duration modelling:

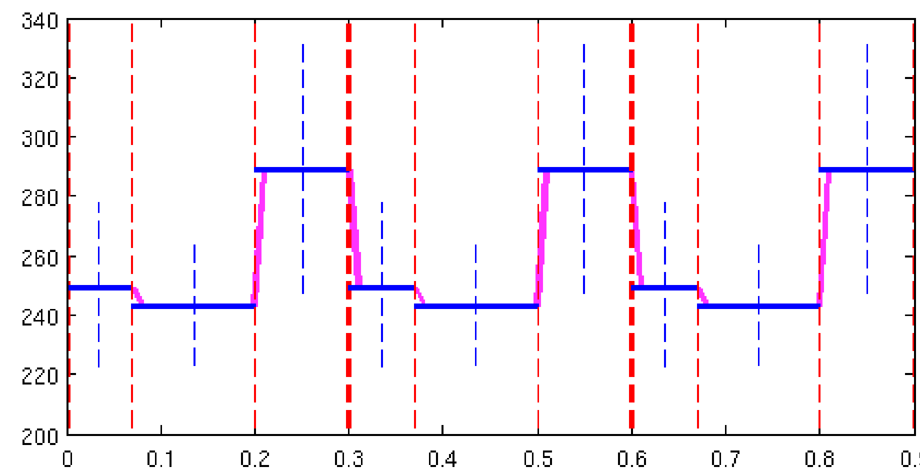


- finding state durations means that we do have to conduct a search (optimize how long to stay in a given state)

- much better: use external duration model (e.g. decision trees) that use target sequence, linguistic information, ...
  - better timings
  - avoids the need for a search

Dynamic Features

- Challenge: μ is always the most likely observation:
  - non-realistic contours
  - disregards continuous nature of speech
  - in recognition, we used Δ-features to capture continuous change

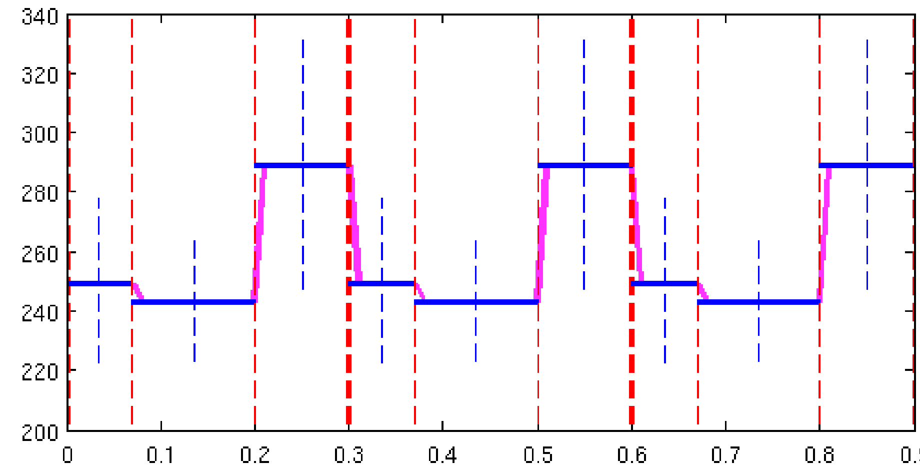


- Solution: introduce *dynamic features*
  - Δ-constraint can be added to the linear equation and little extra cost

images from Taylor (2009).

Dynamic Features

- Challenge: μ is always the most likely observation:
  - non-realistic contours
  - disregards continuous nature of speech



  - in recognition, we used Δ-features to capture continuous change

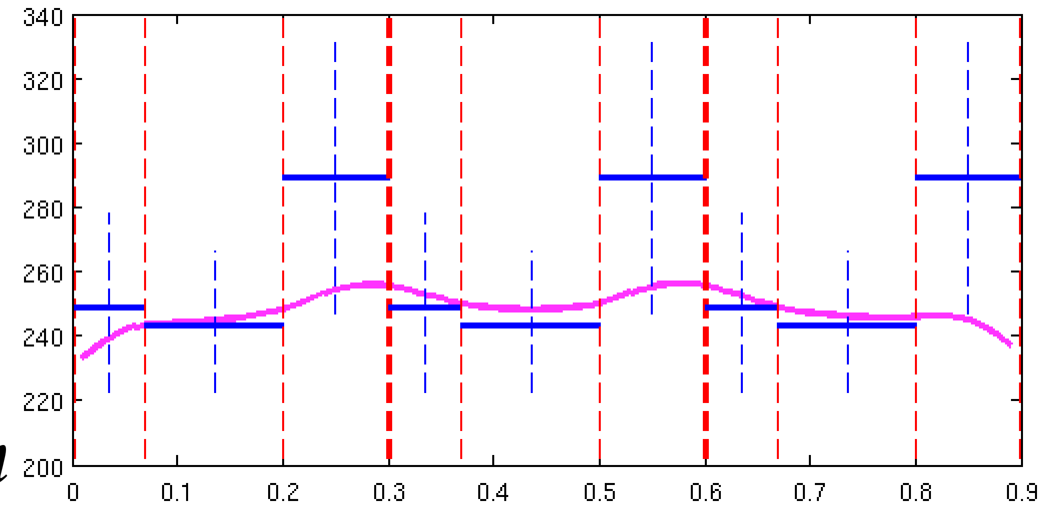
Δ-feature: $(\text{feature}_i - \text{feature}_{i-1})$

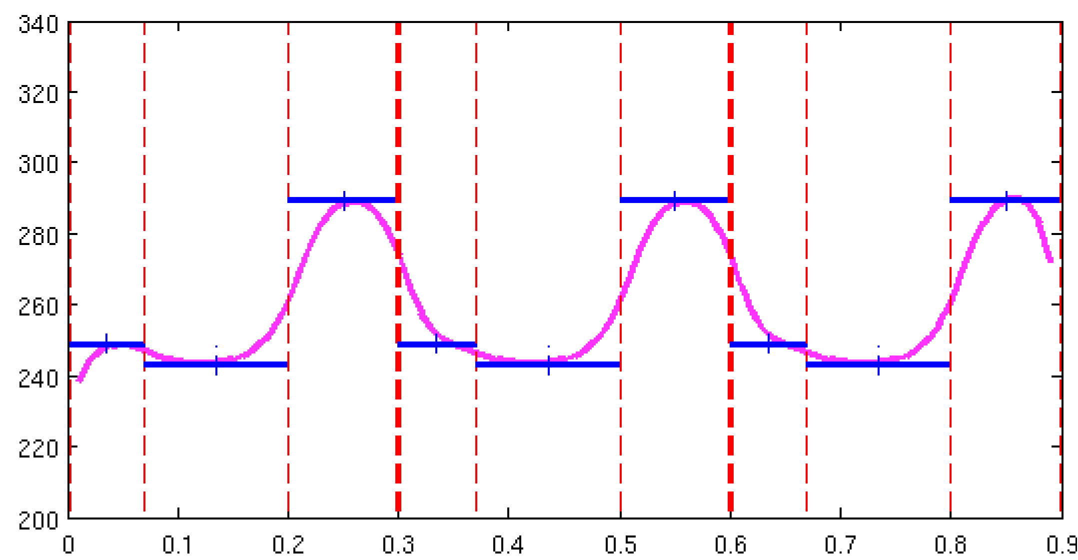
- Solution: introduce *dynamic features*
  - Δ-constraint can be added to the linear equation and little extra cost

images from Taylor (2009).

Dynamic Features II

- contours become continuous but blurred
- optimize to boost σ as well (not just μ)
- *Global Variance optimization*
  - unfortunately, this cannot be done as a simple constraint but requires a local search





Summary

- Speech synthesis does not need to search as it can be formulated as a (linear) optimization problem
-
- Vocoder is not trained but designed
  - interpretable input
- *optimality criterion* of the HMM approach is far from optimal
  - still, it's good enough, can be improved with NNs
  - change input to vocoder outside of the optimization (after the break)

Thank you.

baumann@informatik.uni-hamburg.de

https://nats-www.informatik.uni-hamburg.de/SLP16

Further Reading

- Speech Synthesis in General:
  - D. Jurafsky & J. Martin (2009): *Speech and Language Processing*. Pearson International. InfBib: A JUR 4204x
- Details of Speech Synthesis:
  - P. Taylor (2009): *Text-to-Speech Synthesis*. Cambridge University Press.
- Recent work on HMM-based and NN-based Parametric Synthesis by
  - Heiga Zen (e.g. Tutorial at the UK Speech Conference: http://research.google.com/pubs/pub42624.html)

Notizen

Desired Learning Outcomes

- know the vocoder and be able to relate it to the source-filter model
- understand the limitations of vocoding and parameter estimation, discuss their relative importance
- understand the optimization process in HMM-based speech synthesis
- be able to discuss the advantage of feature stream independence over unit-selection synthesis