Statistical Parametric Speech Synthesis

with focus on LDM based TTS

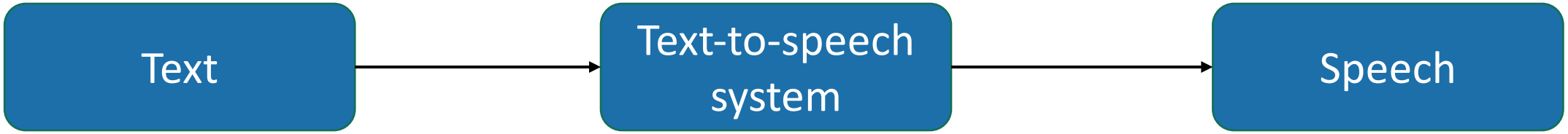
Gagandeep Singh

Outline

- Introduction
- Features used in TTS
- HMM based TTS
- Linear Dynamical Models
- LDM based TTS
- Evaluation
- Neural speech synthesis

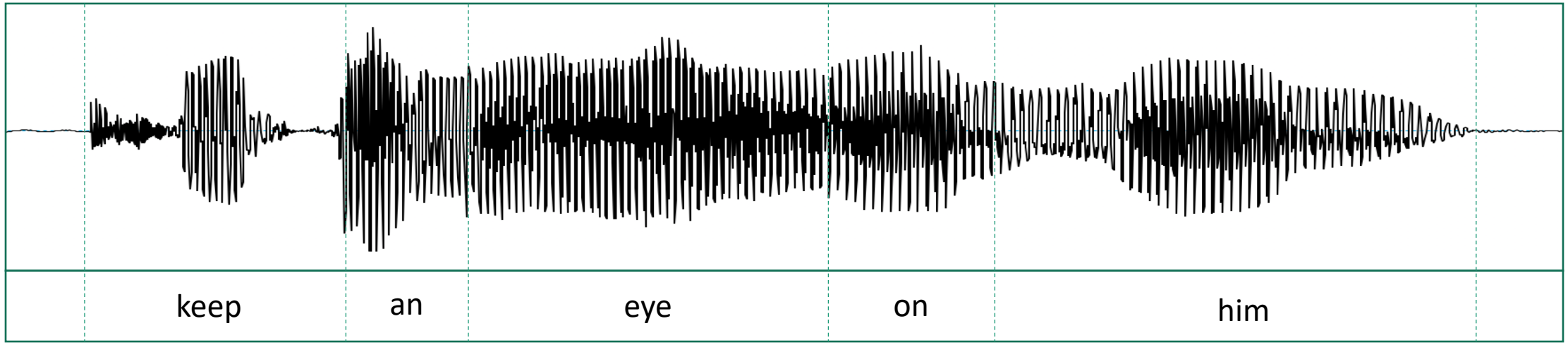
Some of the presentation ideas have been borrowed from:
Prof. Simon King's presentation on SPSS http://spcc.csd.uoc.gr/SPCC2017/SummerSchoolCrete_2017_KingI.pdf
Dr. Vassilis Tsiaras' presentation on LDMs http://spcc.csd.uoc.gr/SPCC2017/SummerSchoolCrete_2017_TsiarasI.pptx

TTS problem

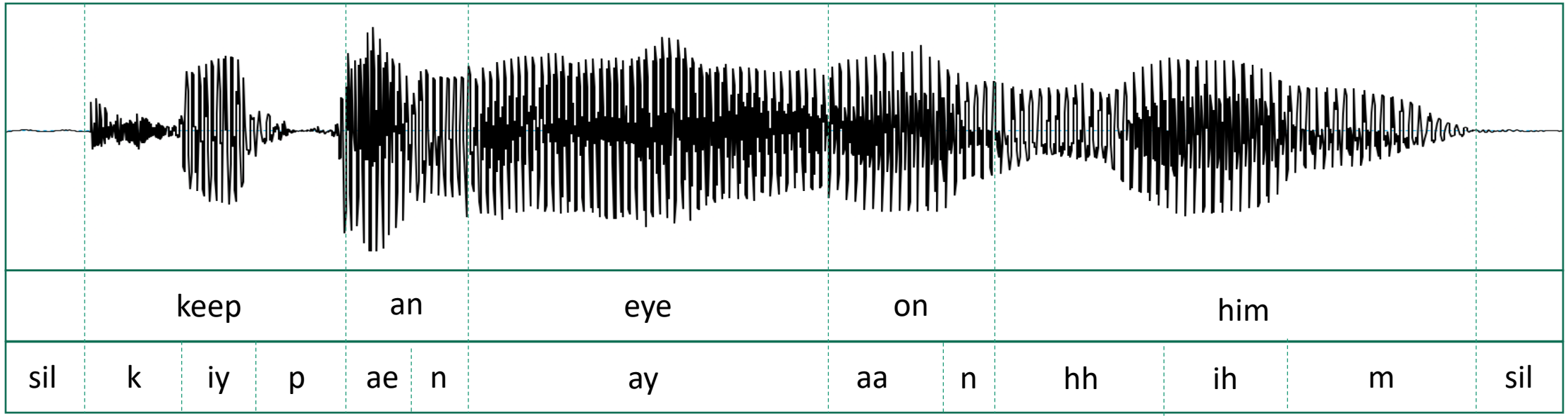


Keep an eye on him.

Speech as a linear sequence of units



Speech as a linear sequence of units

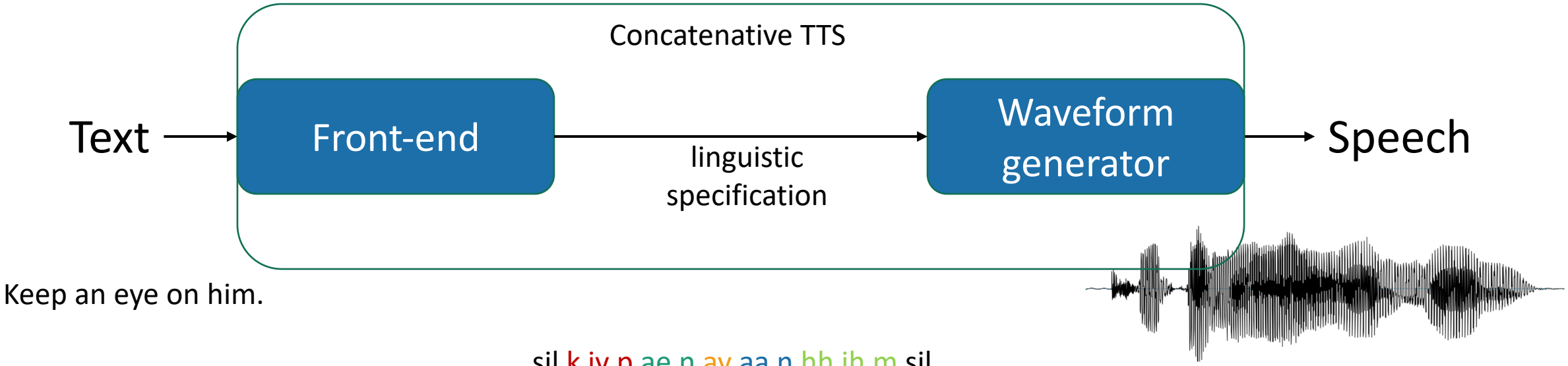


- There are no actual hard boundaries though
- In speech recognition, this allows us to join models of small units (e.g. phonemes) to make models of larger units (e.g. words)
- In speech synthesis, this enables a concatenative approach to synthesis.

Concatenative synthesis

- Based on concatenating together small units to produce an utterance
- Done using a time-domain joining algorithm
- Very natural sounding speech
- Requires a large database of speech
- Not really possible to do voice adaptation or conversion

Concatenative synthesis



Statistical parametric speech synthesis

- Speech is generated from statistical models rather than from stored exemplars
- Statistical models capture the probability distribution of the acoustic features, given the linguistic specification

Classic three stage pipeline

Keep an eye on him.

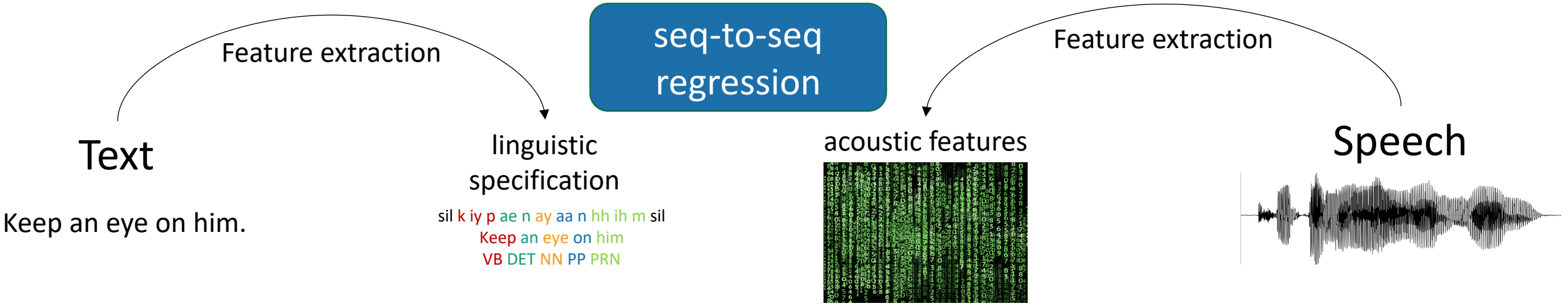
linguistic specification

sil k iy p ae n ay aa n hh ih m sil
Keep an eye on him
VB DET NN PP PRN

acoustic features

- Front-end could be same as in concatenative synthesis
- Linguistic specification is encoded as linguistic features
- Statistical model produces acoustic features given linguistic features
- Waveform generator is NOT same as in concatenative synthesis

Regression model



- The problem boils down to a sequence-to-sequence regression
- Length of linguistic feature sequence is much less than acoustic feature sequence length
- Speech waveform is difficult to model directly

Front-end text analysis

- part-of-speech (POS) tagging and syntactic analysis
- word segmentation
- text normalization
  - e.g. Room no. 4202 at 40 St. George st.
    Room number forty two o two at forty saint George street.
- prosody prediction
- finding word pronunciations
  - e.g. : The record shows that he did not read the conditions.
- generating linguistic features and feature vectors

Linguistic features

Keep an eye on him. : sil k iy p ae n ay aa n hh ih m sil

xx~xx-#+k=iy:1_0/A/0_0_0/B/0-0-0:1-0&1-1#1-1$1-1>0-0<0-0|0/C/0+0+0/D/0_0/E/0+0:1+0&1+0#0+0/F/0_0/G/0_0/H/0=0:1=1&0/I/0_0/J/5+5-1

xx~#-k+iy=p:1_3/A/0_0_0/B/1-1-3:1-1&1-5#1-5$1-3>0-1<0-2|iy/C/1+0+2/D/0_0/E/content+1:1+5&1+2#0+2/F/det_1/G/0_0/H/5=5:1=1&L-L%/I/0_0/J/5+5-1

#~k-iy+p=ae:2_2/A/0_0_0/B/1-1-3:1-1&1-5#1-5$1-3>0-1<0-2|iy/C/1+0+2/D/0_0/E/content+1:1+5&1+2#0+2/F/det_1/G/0_0/H/5=5:1=1&L-L%/I/0_0/J/5+5-1

k~iy-p+ae=n:3_1/A/0_0_0/B/1-1-3:1-1&1-5#1-5$1-3>0-1<0-2|iy/C/1+0+2/D/0_0/E/content+1:1+5&1+2#0+2/F/det_1/G/0_0/H/5=5:1=1&L-L%/I/0_0/J/5+5-1

iy~p-ae+n=ay:1_2/A/1_1_3/B/1-0-2:1-1&2-4#2-4$2-3>1-1<1-1|ae/C/1+1+1/D/content_1/E/det+1:2+4&2+2#1+1/F/content_1/G/0_0/H/5=5:1=1&L-L%/I/0_0/J/5+5-1

p~ae-n+ay=aa:2_1/A/1_1_3/B/1-0-2:1-1&2-4#2-4$2-3>1-1<1-1|ae/C/1+1+1/D/content_1/E/det+1:2+4&2+2#1+1/F/content_1/G/0_0/H/5=5:1=1&L-L%/I/0_0/J/5+5-1

ae~n-ay+aa=n:1_1/A/1_0_2/B/1-1-1:1-1&3-3#3-3$2-2>1-1<2-2|ay/C/1+0+2/D/det_1/E/content+1:3+3&2+1#2+2/F/in_1/G/0_0/H/5=5:1=1&L-L%/I/0_0/J/5+5-1

n~ay-aa+n=hh:1_2/A/1_1_1/B/1-0-2:1-1&4-2#4-2$3-2>1-1<1-1|aa/C/1+1+3/D/content_1/E/in+1:4+2&3+1#1+1/F/content_1/G/0_0/H/5=5:1=1&L-L%/I/0_0/J/5+5-1

ay~aa-n+hh=ih:2_1/A/1_1_1/B/1-0-2:1-1&4-2#4-2$3-2>1-1<1-1|aa/C/1+1+3/D/content_1/E/in+1:4+2&3+1#1+1/F/content_1/G/0_0/H/5=5:1=1&L-L%/I/0_0/J/5+5-1

aa~n-hh+ih=m:1_3/A/1_0_2/B/1-1-3:1-1&5-1#5-1$3-1>1-0<2-0|ih/C/0+0+0/D/in_1/E/content+1:5+1&3+0#2+0/F/0_0/G/0_0/H/5=5:1=1&L-L%/I/0_0/J/5+5-1

n~hh-ih+m=#:2_2/A/1_0_2/B/1-1-3:1-1&5-1#5-1$3-1>1-0<2-0|ih/C/0+0+0/D/in_1/E/content+1:5+1&3+0#2+0/F/0_0/G/0_0/H/5=5:1=1&L-L%/I/0_0/J/5+5-1

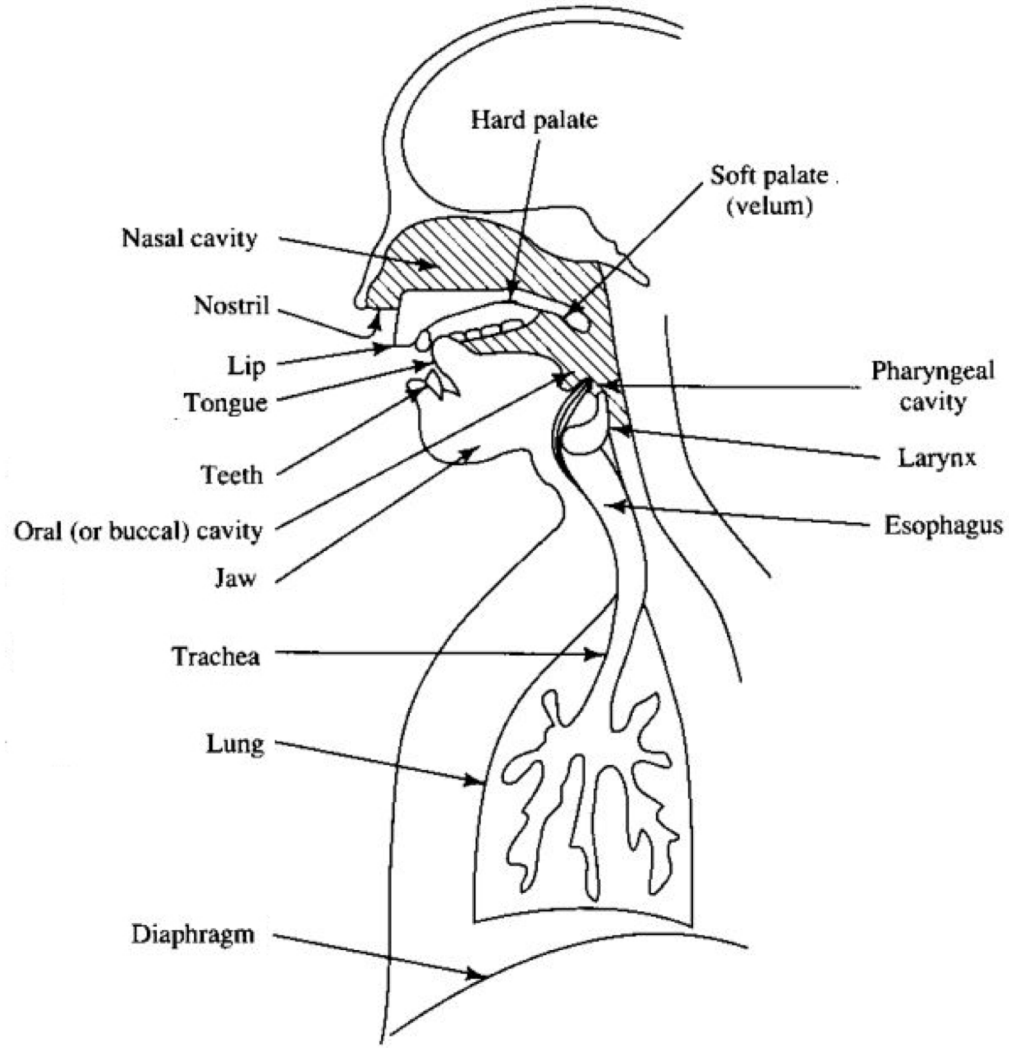
hh~ih-m+#=xx:3_1/A/1_0_2/B/1-1-3:1-1&5-1#5-1$3-1>1-0<2-0|ih/C/0+0+0/D/in_1/E/content+1:5+1&3+0#2+0/F/0_0/G/0_0/H/5=5:1=1&L-L%/I/0_0/J/5+5-1

ih~m-#+xx=xx:1_0/A/0_0_0/B/0-0-0:1-0&1-1#1-1$1-1>0-0<0-0|0/C/0+0+0/D/0_0/E/0+0:1+0&1+0#0+0/F/0_0/G/0_0/H/0=0:1=1&0/I/0_0/J/5+5-1

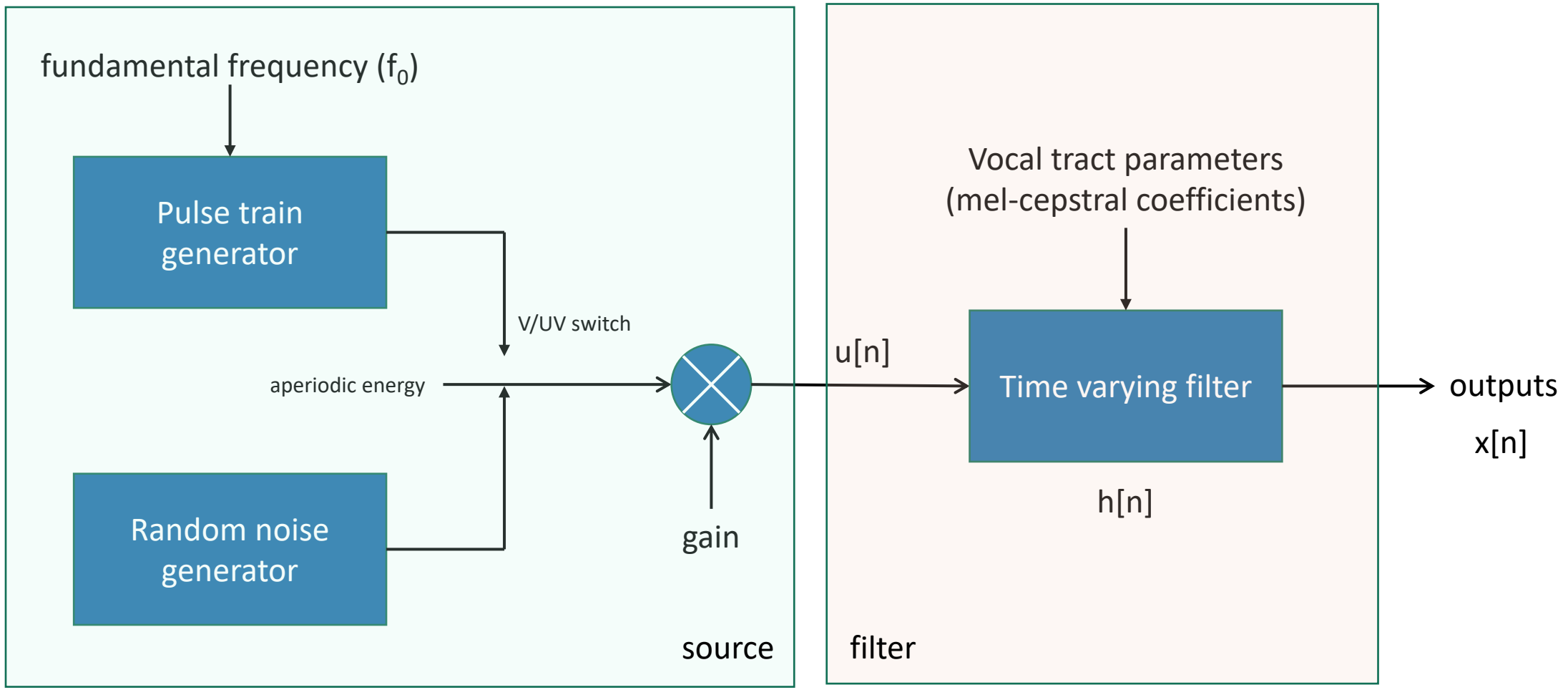
Also called context-dependent phone.
These are encoded to numerical feature vectors.

Speech production

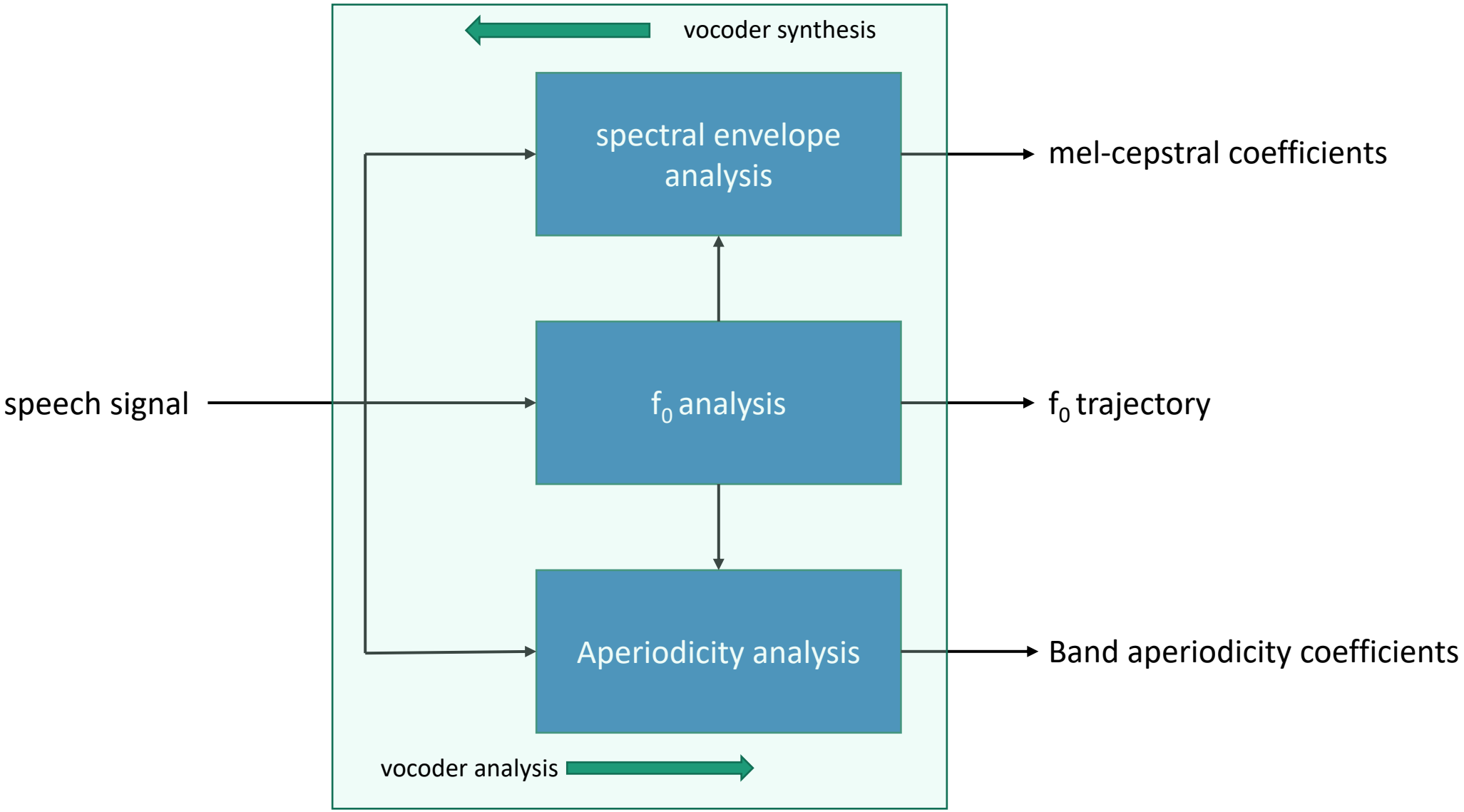
- voiced sounds
- unvoiced sounds



Source filter model



One way of decomposing speech signal



Mel-cepstral analysis

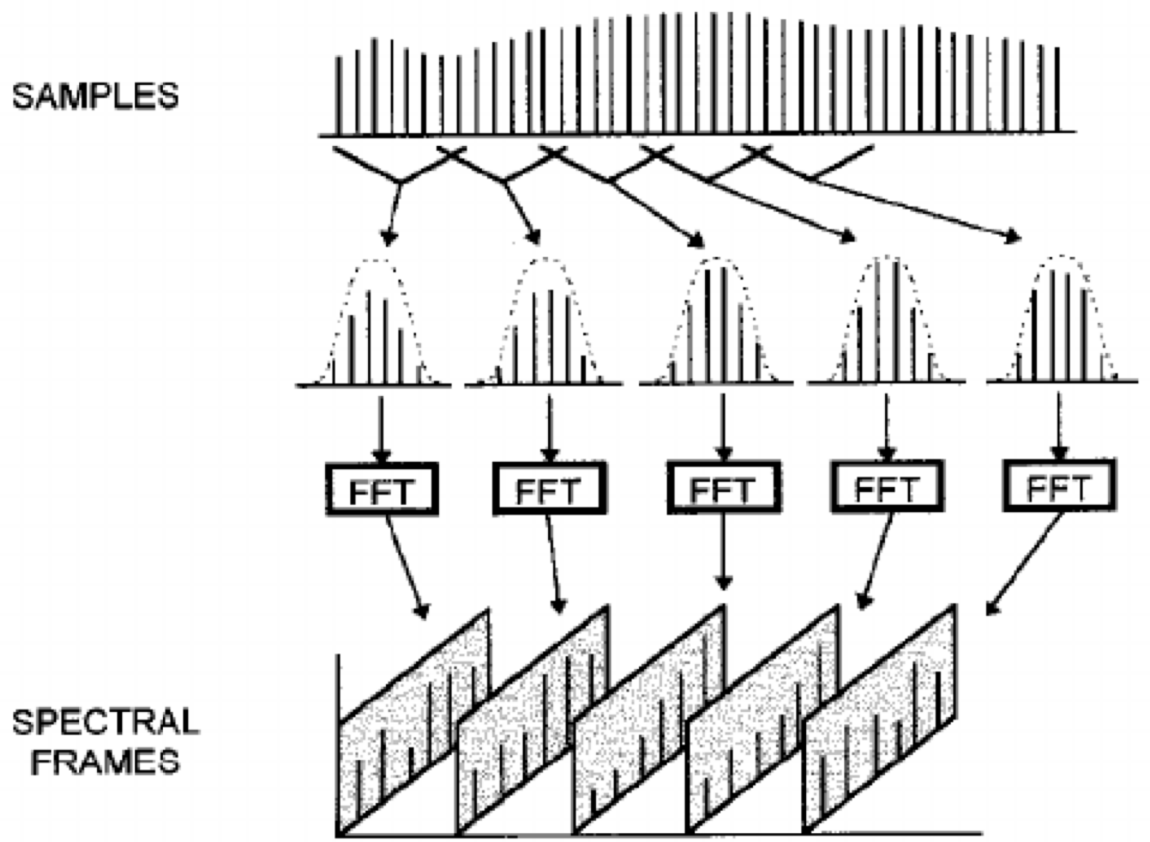
- Speech signal x[n] : discrete time signal, some sampling frequency e.g. 16 kHz
- $x[n] = u[n] * h[n]$
  - where u[n] is the source excitation
  - h[n] is the vocal tract impulse response
- Extract vocal tract characteristics from x[n]

speech signal

x[n]

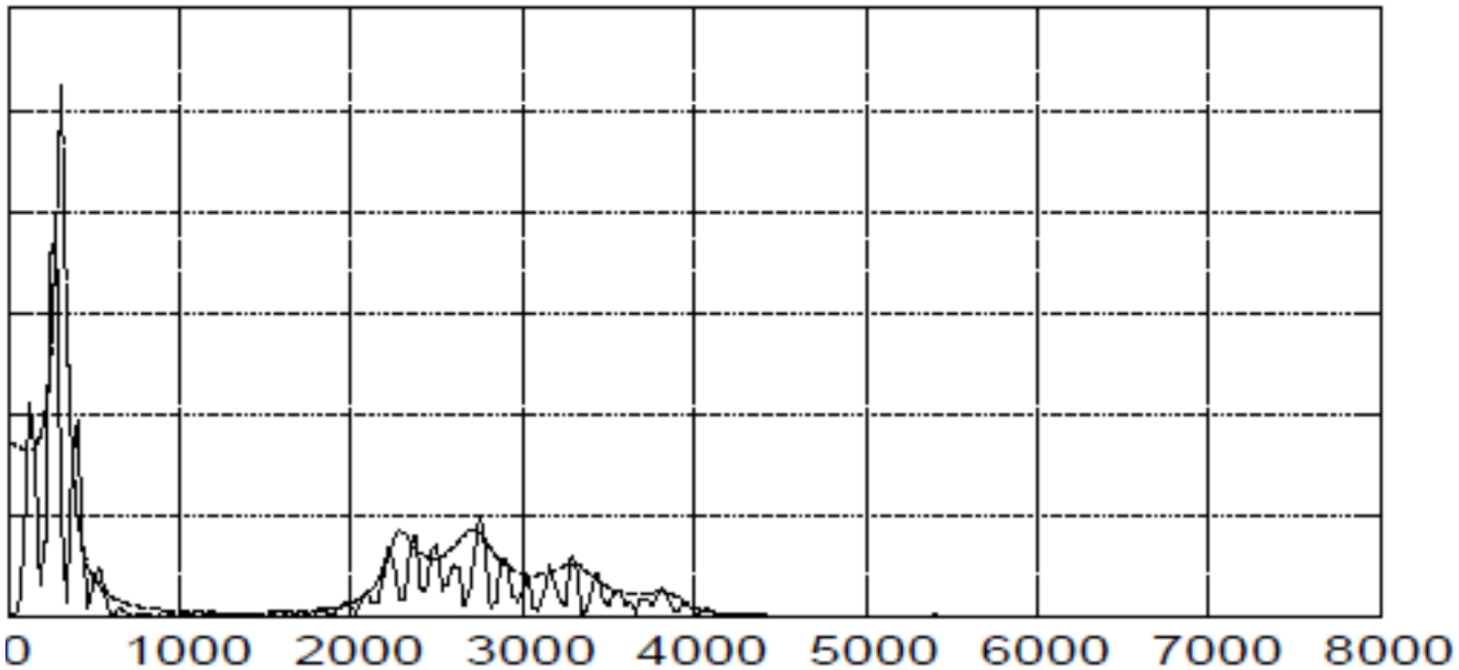
Short-term discrete Fourier transform (ST-DFT)

- window size e.g. 25 ms
- window shift e.g. 5 ms

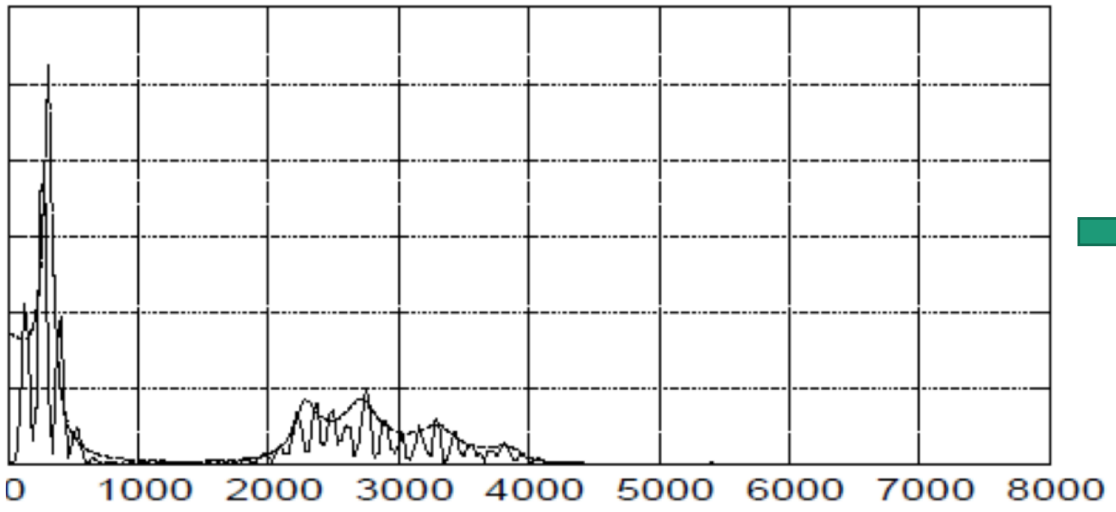


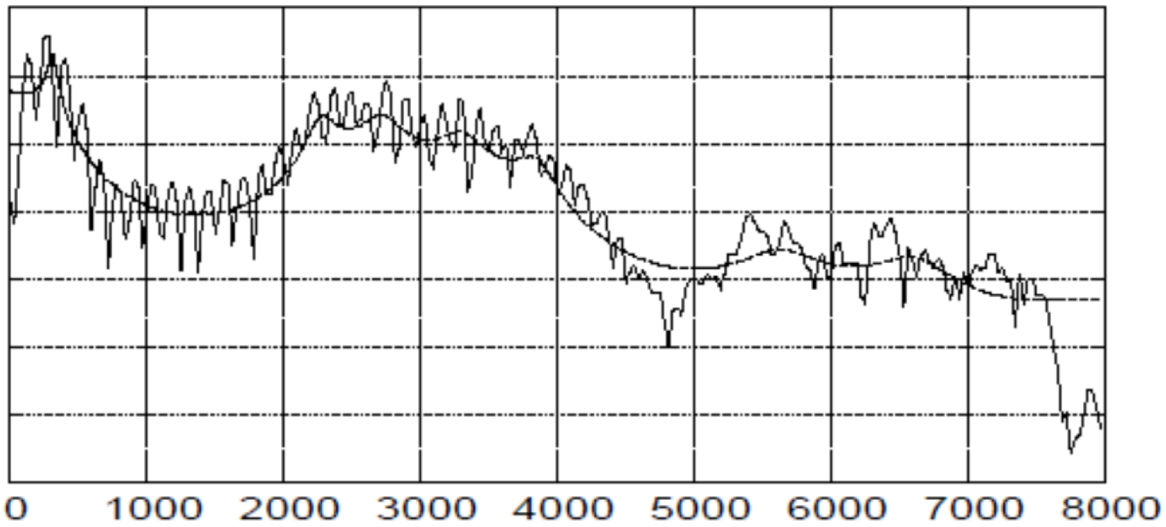
speech signal
x[n]

ST-DFT
$\mathcal{F}\{x[n]\}$



speech signal x[n] → ST-DFT $\mathcal{F}\{x[n]\}$ → logarithm of magnitude spectrum





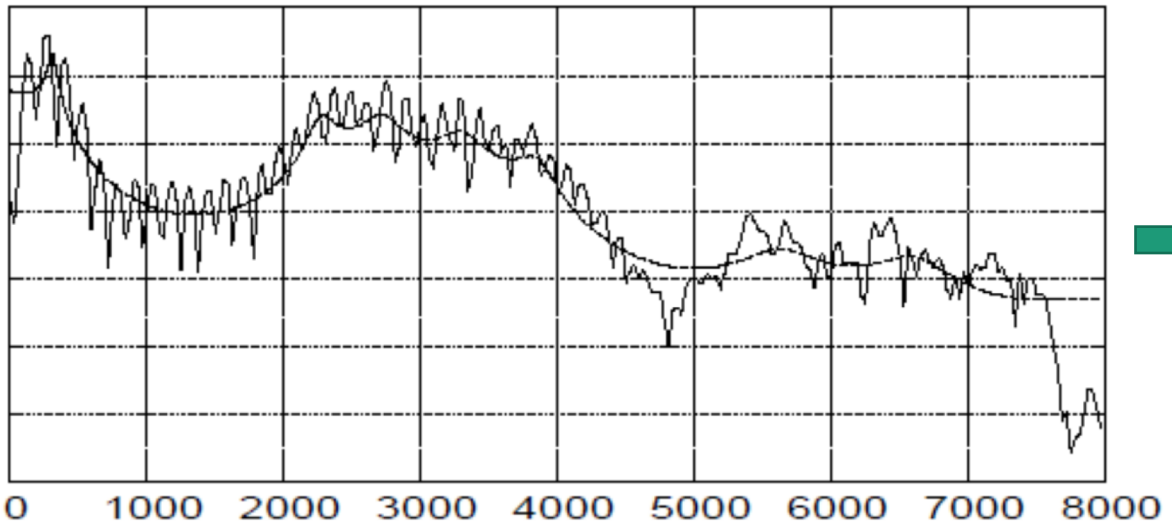
- $x[n] = u[n] * h[n] \Rightarrow \mathcal{F}\{x[n]\} = \mathcal{F}\{u[n]\} \times \mathcal{F}\{h[n]\}$
- $\log(|\mathcal{F}\{x[n]\}|) = \log(|\mathcal{F}\{u[n]\}|) + \log(|\mathcal{F}\{h[n]\}|)$

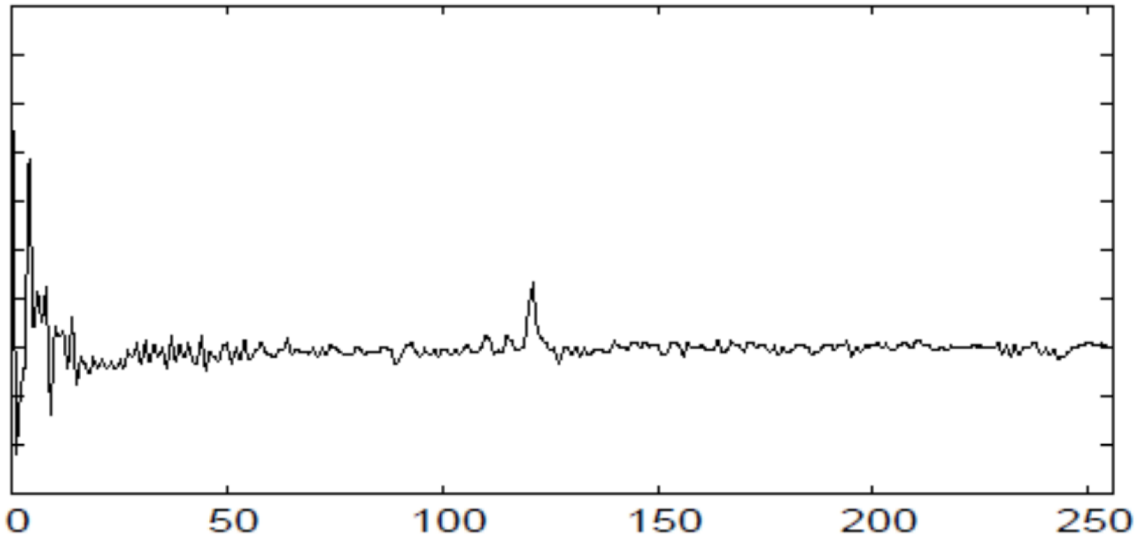
speech signal
x[n]

→ ST-DFT
$\mathcal{F}\{x[n]\}$

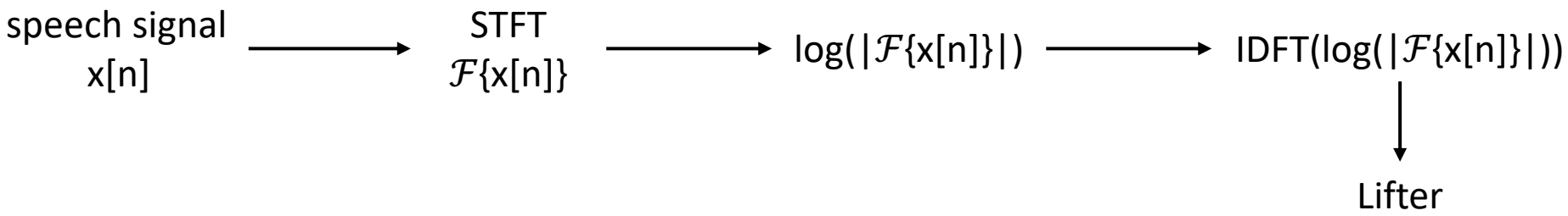
→ $\log(|\mathcal{F}\{x[n]\}|)$

↓ frequency warping
and Inverse DFT

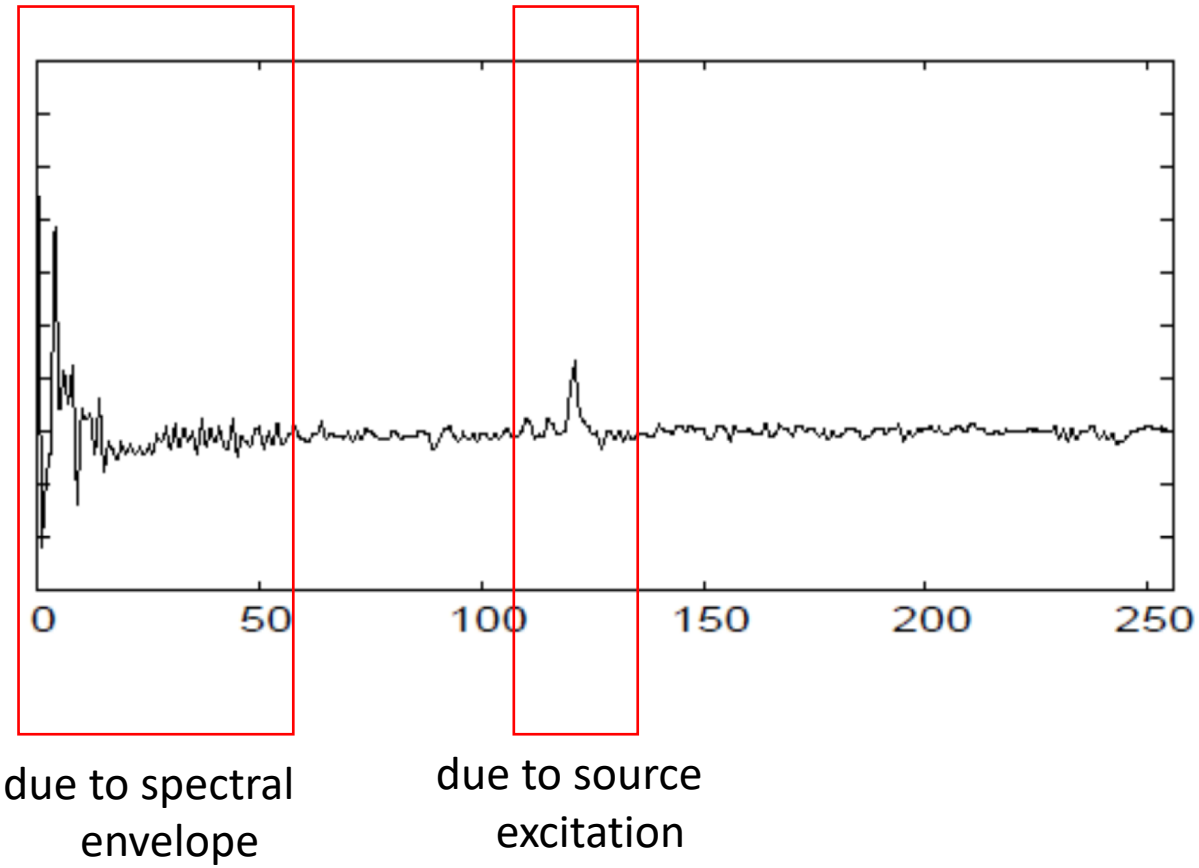




- This results in cepstrum (inverse of spectrum!)

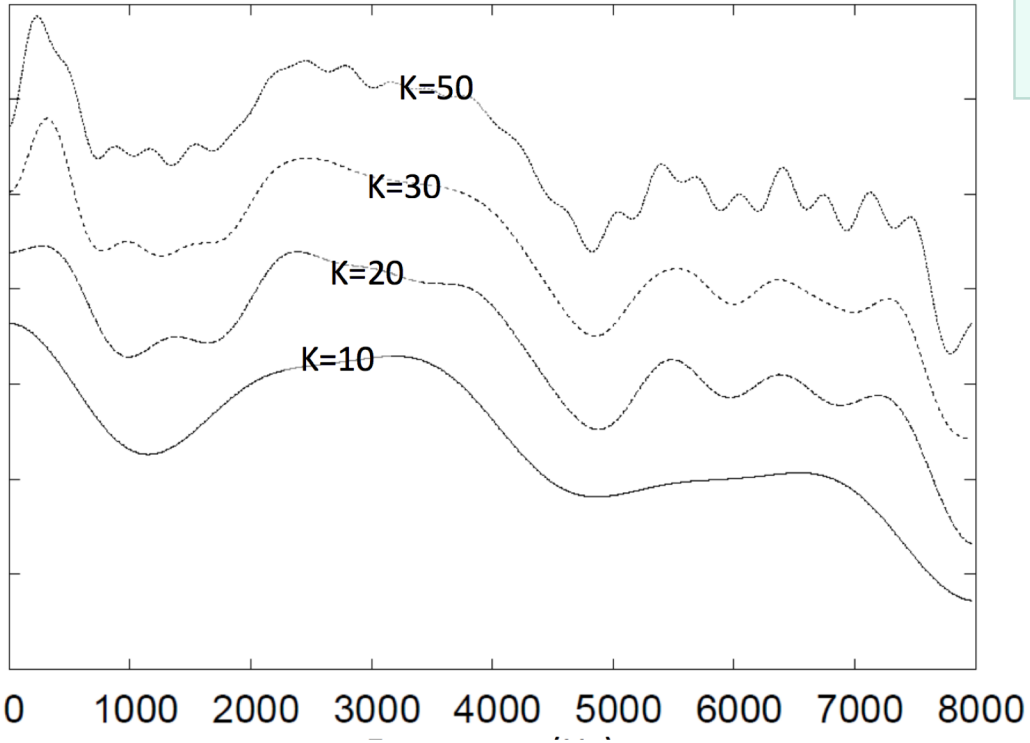


- Filter out the slow moving component
- Called liftering (how creative is that)
- Result is mel-cepstral coefficients
- Not same as MFCCs



speech signal x[n] → STFT $\mathcal{F}\{x[n]\}$ → $\log(|\mathcal{F}\{x[n]\}|)$ → IDFT($\log(|\mathcal{F}\{x[n]\}|)$) → lifter → Mel-cepstral coefficients → DFT → exponential → Magnitude spectral envelope

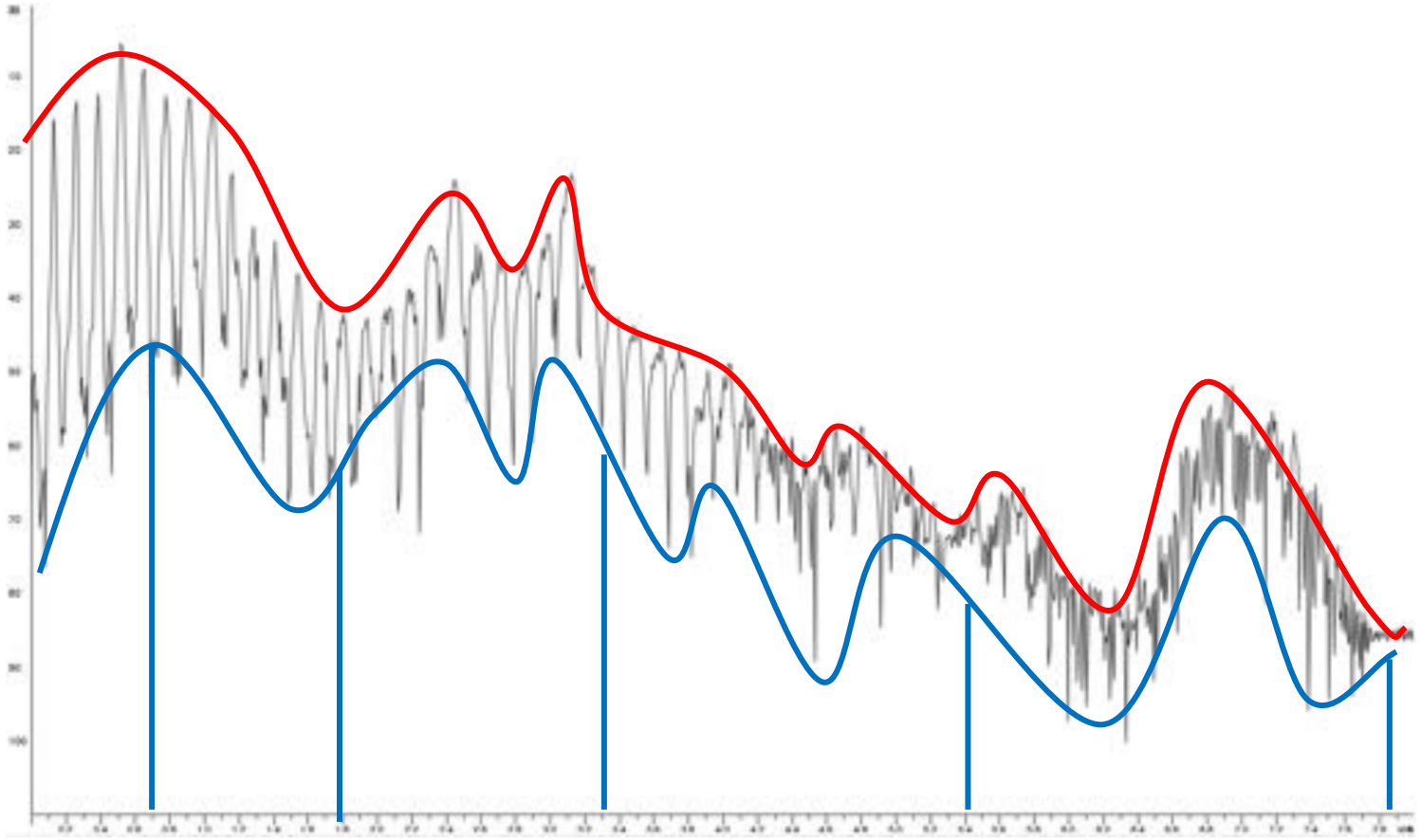
Reverse previous operations



What about phase information though?

- Minimum phase spectrum can easily be created from magnitude spectrum
- Use of algorithms e.g. Griffin-Lim

Band aperiodicity



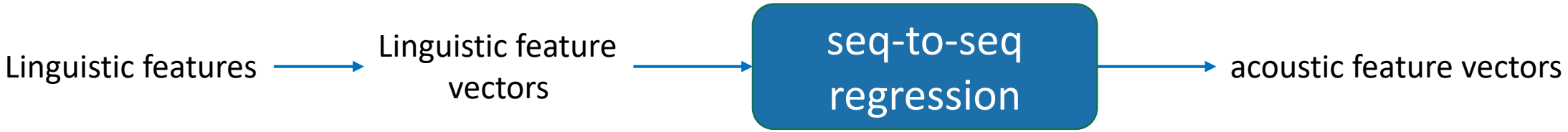
- Aperiodic energy
- Reduced resolution by averaging across broad frequency bands
- Typically around 3 to 5 bands (on a Mel scale)

STRAIGHT[1] acoustic features

- Per each frame at a frame rate of typically every 5ms
  - Mel-cepstral features (MCEPs or MGCs) : typically 40 or 60
  - Fundamental frequency (f_0) : one per frame
    - Voiced-unvoiced binary feature
  - Band-aperiodic features (BAPs) : typically 3-5 per frame
- These only correspond to vocoders like STRAIGHT[1] or WORLD[2]
- Other vocoders like Magphase[3], Vocaine[4] have completely different set of features

1 H. Kawahara, "STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds," Acoustical science and technology, vol. 27, no. 6, pp. 349– 353, 2006.
2 M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," IEICE Trans. on Information and Systems, vol. 99, no. 7, pp. 1877–1884, 2016.
3 F. Espic, C. Valentini-Botinhao, and S. King, "Direct Modelling of Magnitude and Phase Spectra for Statistical Parametric Speech Synthesis," in Proc. Interspeech, Stockholm, Sweden, August, 2017.
4 Y. Agiomyrgiannakis "Vocaine the vocoder and applications in speech synthesis." 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2015): 4230-4234.

Regression between input and output



Some options are:

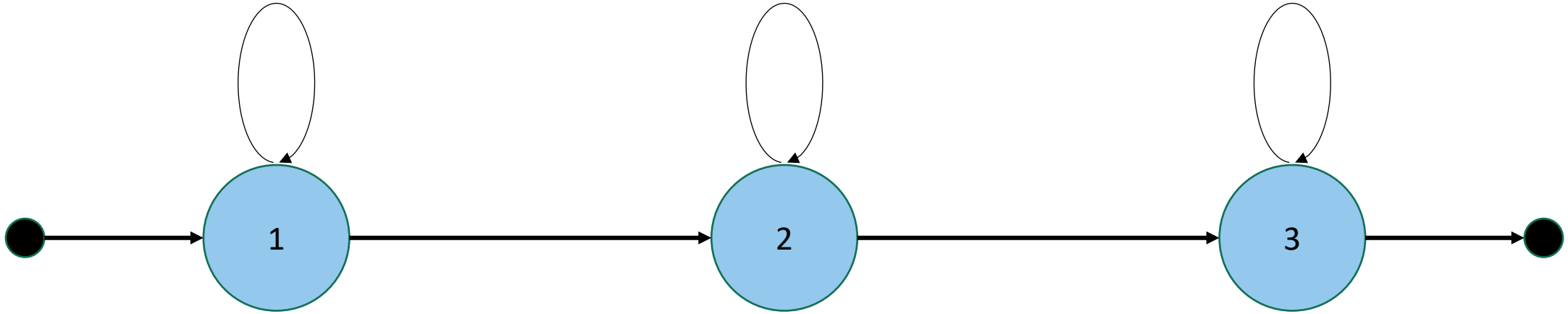
- Decision trees and HMMs
- Decision trees and LDMs
- Neural networks
- Neural seq-2-seq models e.g. with encoder and attention based decoder

Output generation overview

- For each “context-dependent” phone in the input, predict duration using a duration model
- Create a sequence of frames for that phone
- For example:

phone	sil	k	iy	p	ae	n	ay	aa	n	hh	ih	m	sil
duration	10	12	18	10	32	29	60	34	27	29	21	15	10

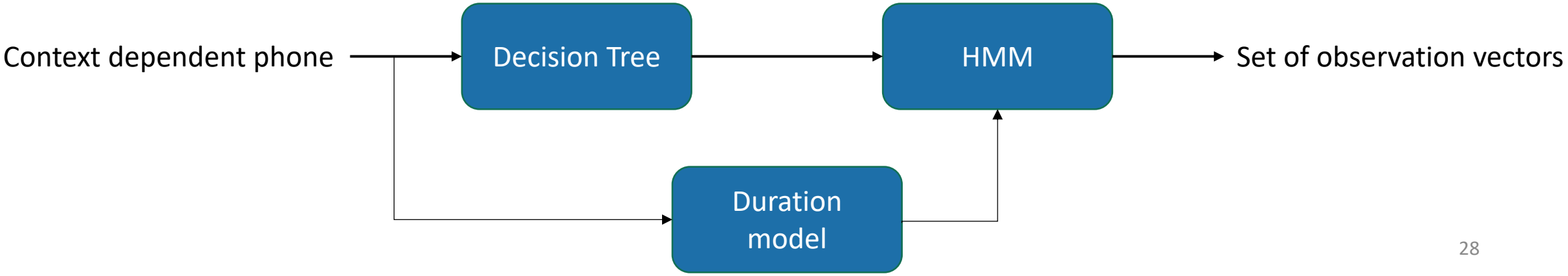
Context dependent phone model

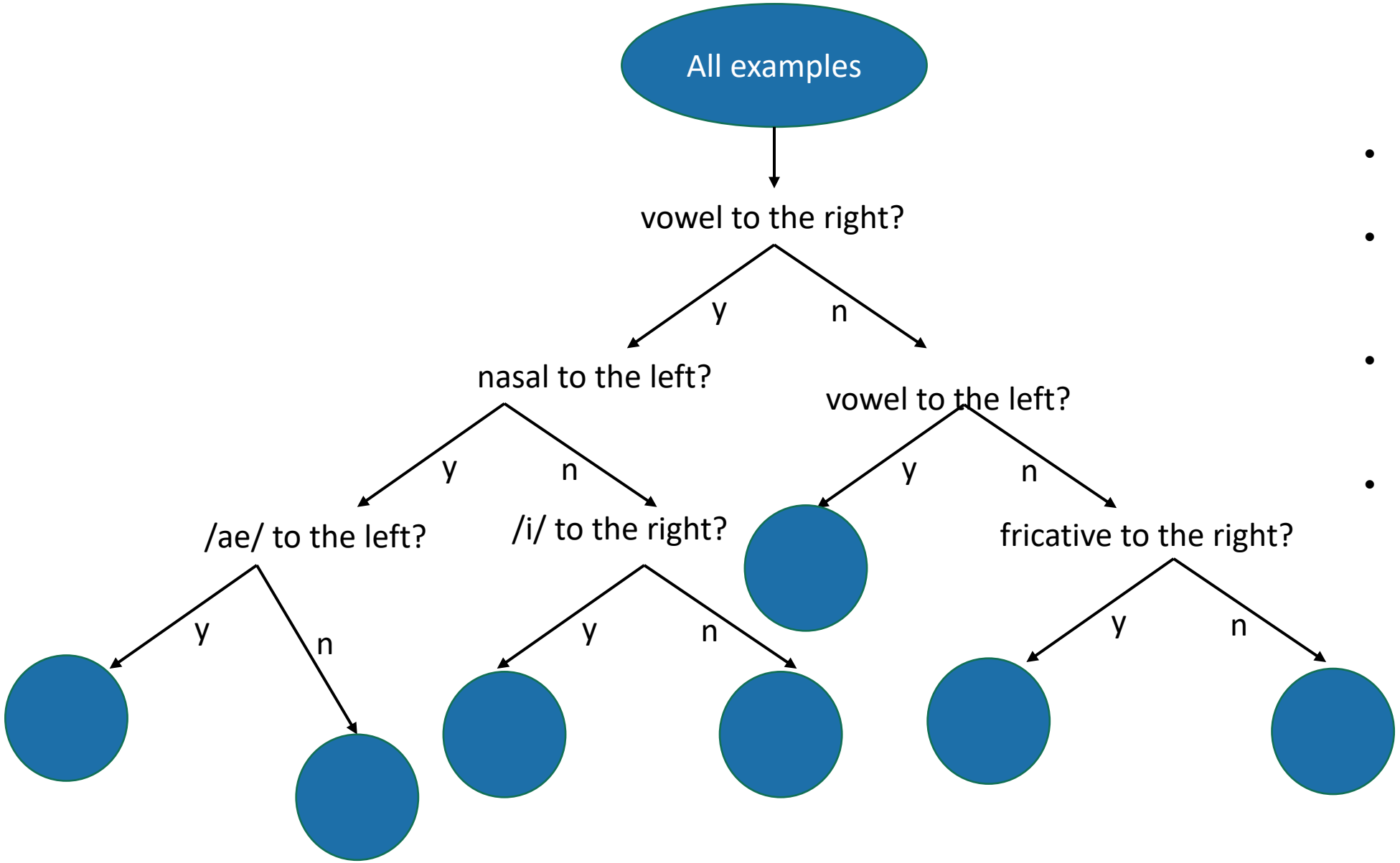


- Inspired from ASR
- 3 or 5 state HMM phone model
- One HMM for each context dependent phone?
- Number of context-dependent phones are extremely large
- Either a few examples per phone or no example at all
- Sharing parameters between various states, i.e. two context-dependent phones p_1 and p_2 can have same HMM

Clustering

- Decision tree based clustering solves two problems:
  - Sharing parameters between context-dependent phones
  - Mapping linguistic feature vector to an HMM
- Clustering done based on linguistic features





- Standard decision tree clustering algorithm
- We already have a question list and all the answers
- Fit an HMM state to each of the leaf node examples
- This even works for contexts with no examples at all

Training HMMs

- Initially training data is aligned to phone sequence using HMM forced alignment.
- All examples of a given phone are used to create a decision tree.
- Thus each given phone according to its context has a corresponding HMM.
- HMM outputs are usually Gaussian distributed or a mixture of Gaussians.
- Fitting an HMM involves estimating the distribution parameters e.g. mean and variance from the corresponding examples.
- HMMs are used in conjunction with decision trees

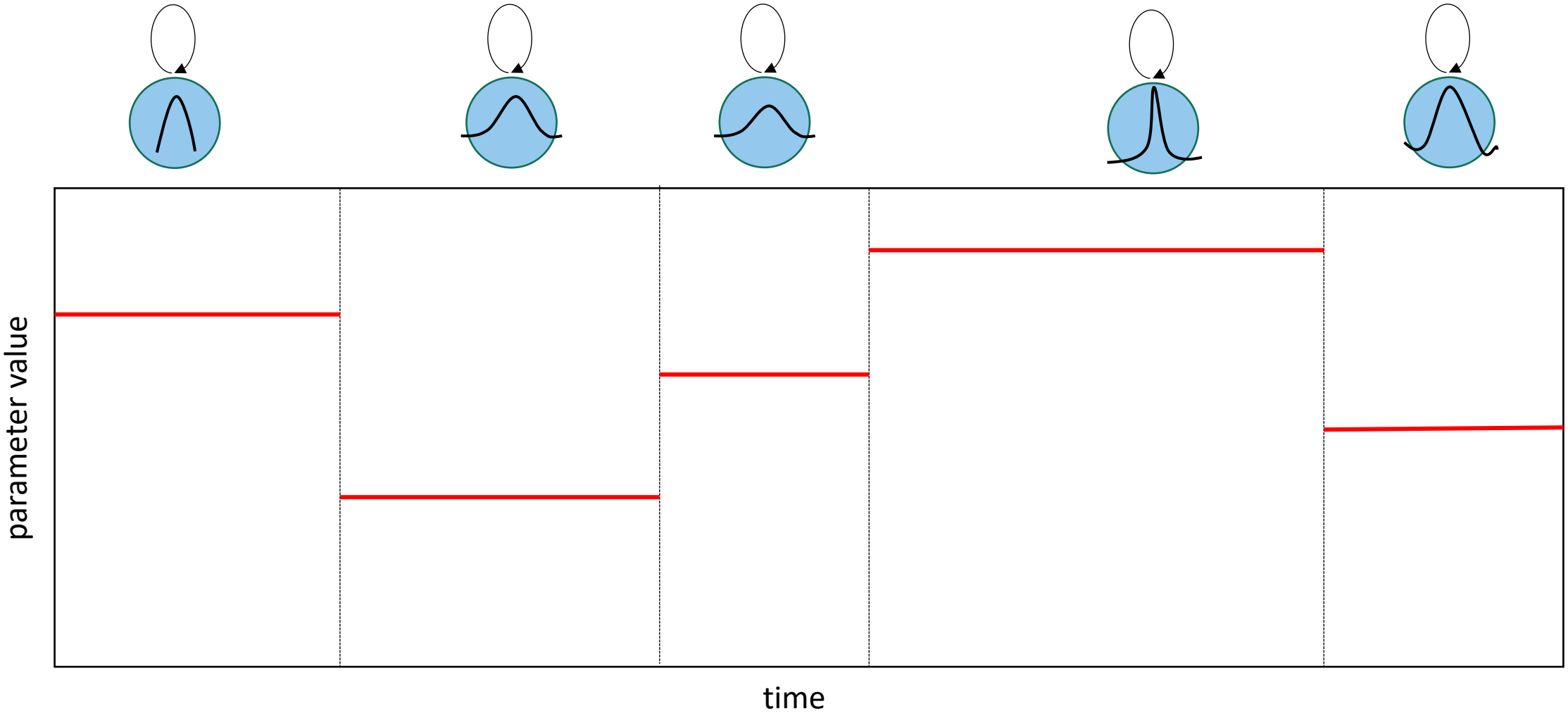
What about duration?

- Assume some duration distribution for each context-dependent phone e.g. Gaussian distribution
- Create another decision tree to map the context-dependent phone to duration distribution
- Use corresponding examples to find distribution parameters

Parameter generation

- Get the HMMs corresponding to each of the context-dependent phones
- Concatenate all the HMM states
- Find the number of frames to generate for each HMM state
- Emit the observation vectors according to maximum likelihood
- Observation vectors are modeled using Gaussian distribution

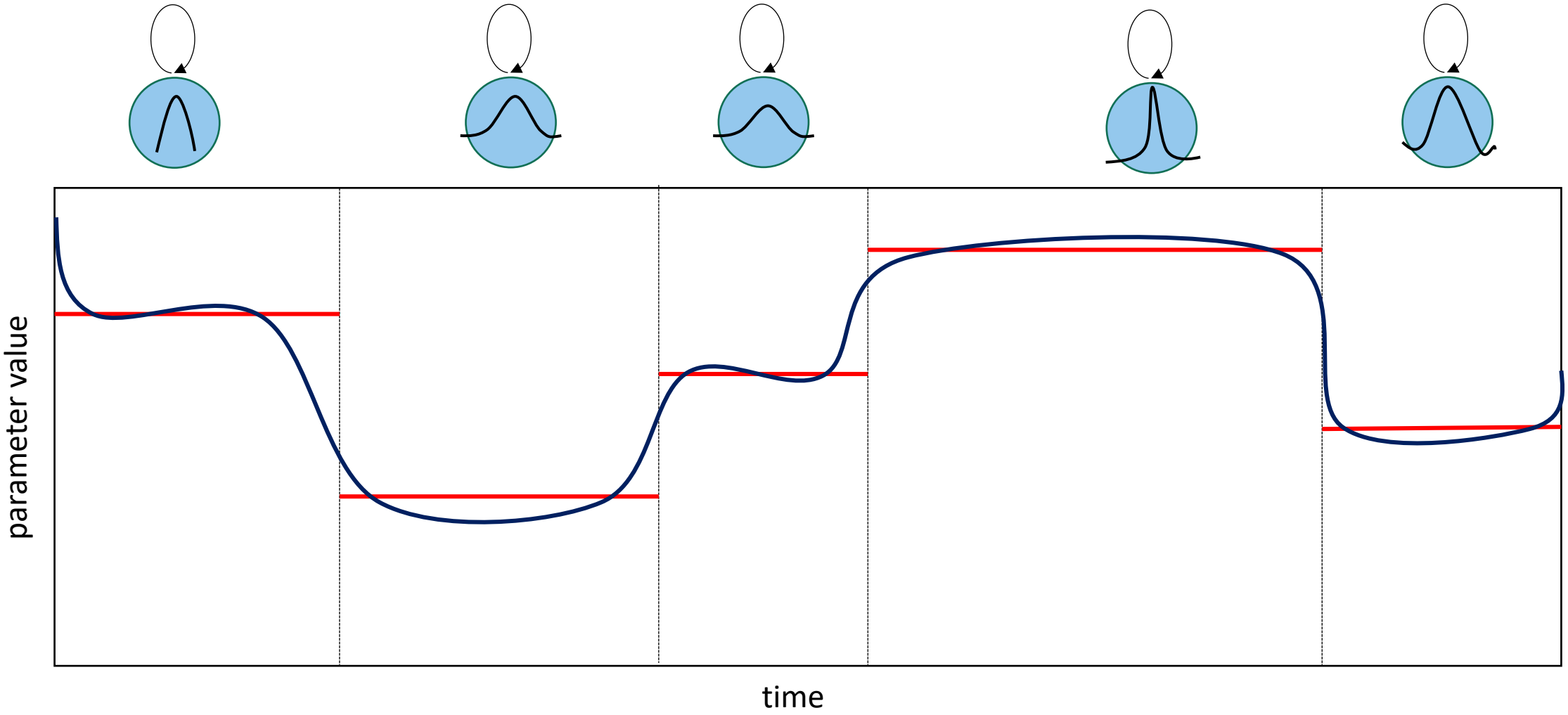
Trajectory generation



Smoothing the trajectory

- Observation vectors to include not just the parameter values but also velocity and acceleration
- Called deltas and double-deltas
- $\boldsymbol{\Delta}_{\mathrm{t}} = \frac{\boldsymbol{c}_{t+1} - \boldsymbol{c}_{t-1}}{2}$
- $\boldsymbol{\Delta\Delta}_{\mathrm{t}} = \frac{\boldsymbol{\Delta}_{t+1} - \boldsymbol{\Delta}_{t-1}}{2}$
- Take these into consideration during generation: maximum likelihood parameter generation (MLPG)

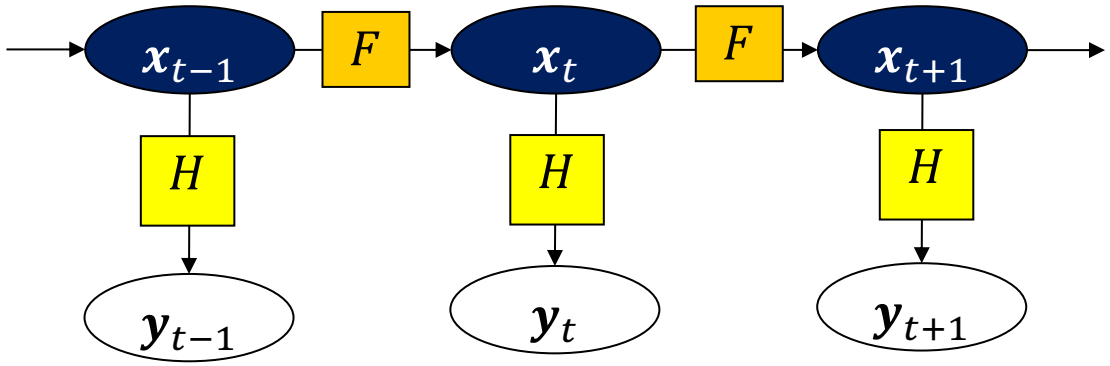
Parameter generation

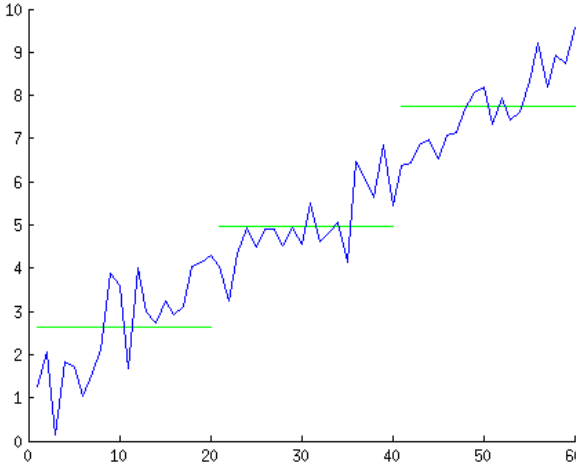


Linear Dynamical Models

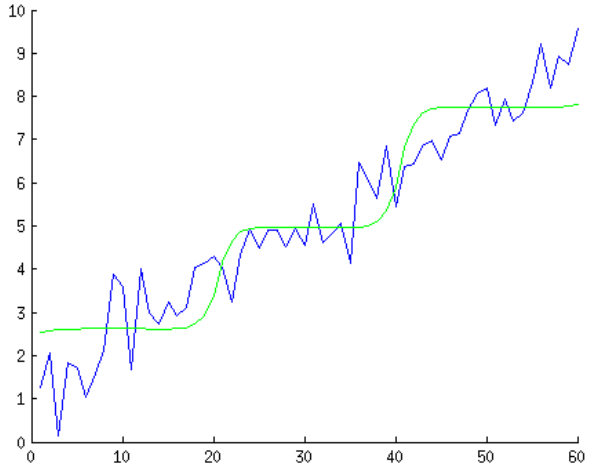
- Class of continuous state space models with a linear state evolution equation.
- Continuous vector hidden state $\boldsymbol{x}_t$
- State space is *n* dimensional while observation space is *m* dimensional
- For an LDM q
  - state evolution: $\boldsymbol{x}_t = F_q \boldsymbol{x}_{t-1} + \boldsymbol{w}_t, \quad \boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{g}_q, Q_q)$
  - Observation generation: $\boldsymbol{y}_t = H_q \boldsymbol{x}_t + \boldsymbol{v}_t, \quad \boldsymbol{v}_t \sim \mathcal{N}(\boldsymbol{\mu}_q, R_q)$
  - Initial state: $\boldsymbol{x}_1 \sim \mathcal{N}(\boldsymbol{g}_{1,q}, Q_{1,q})$

Linear Dynamical Models

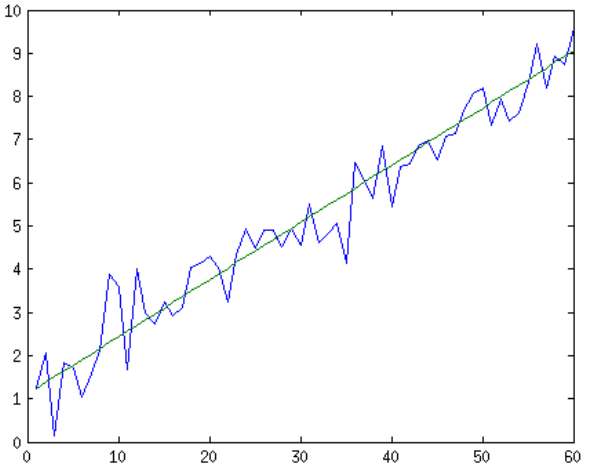




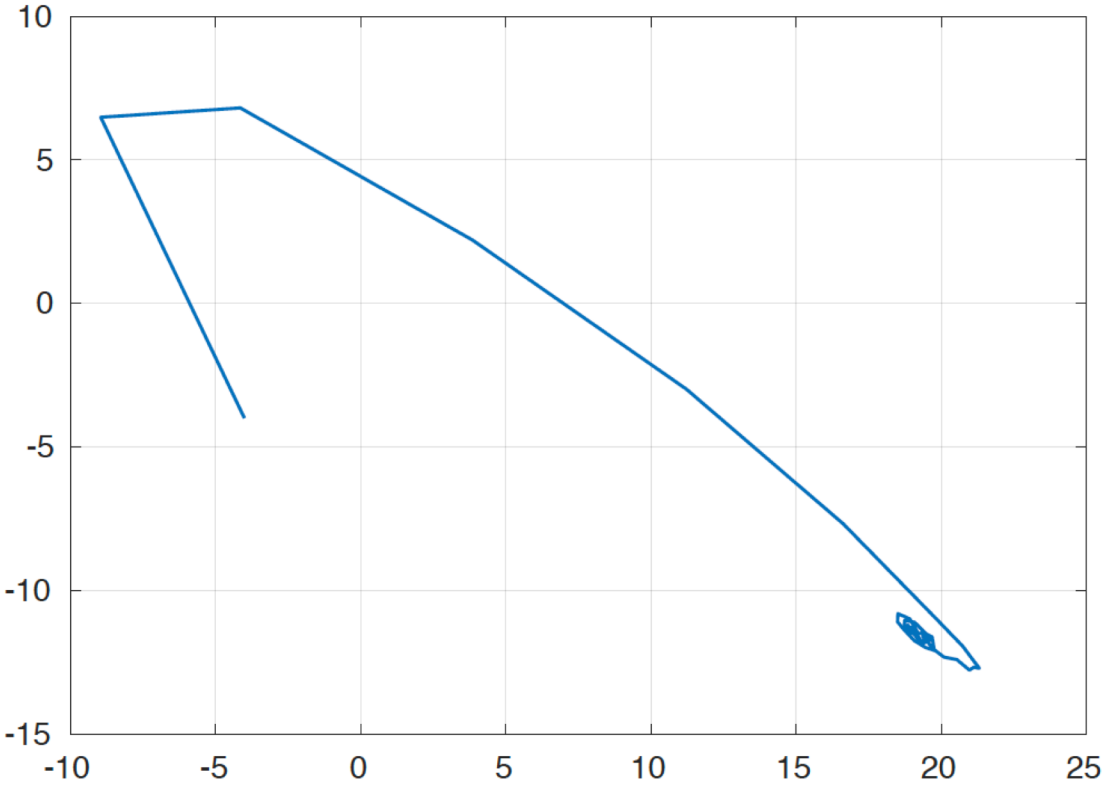
HMM modeling



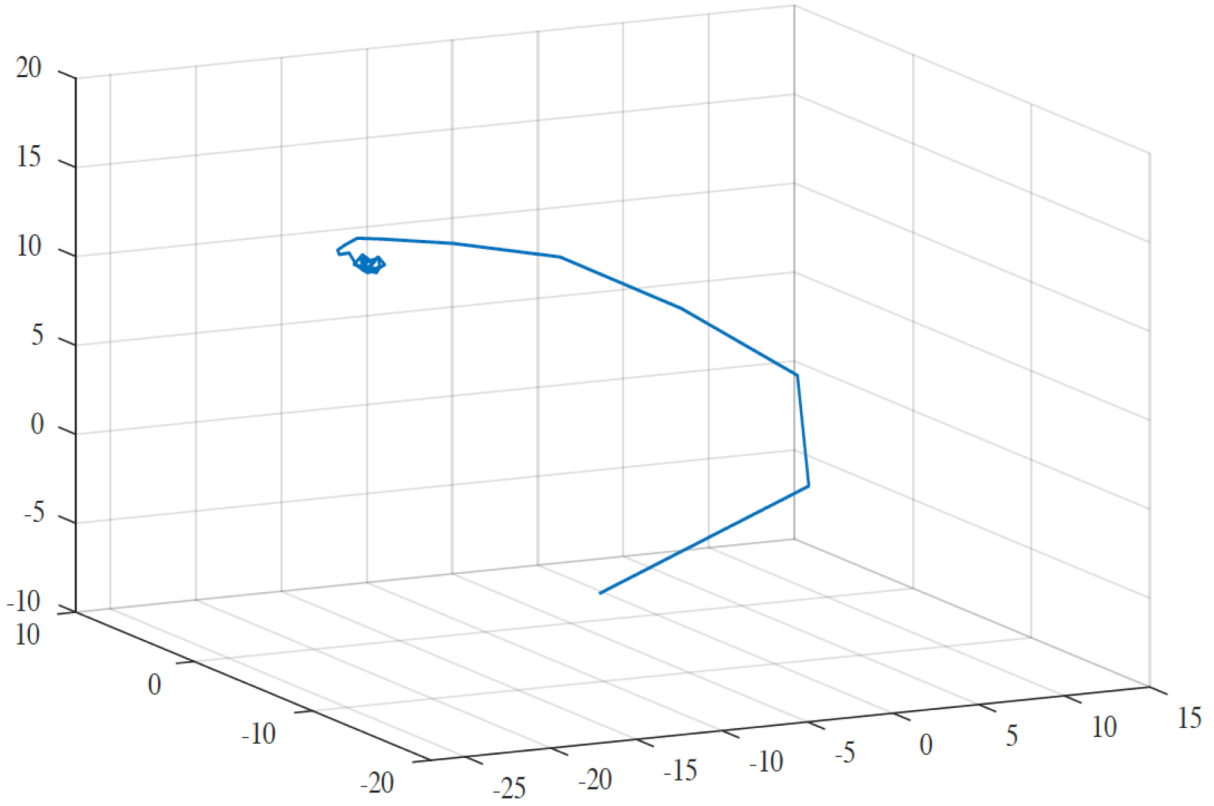
HMM with dynamic features modeling



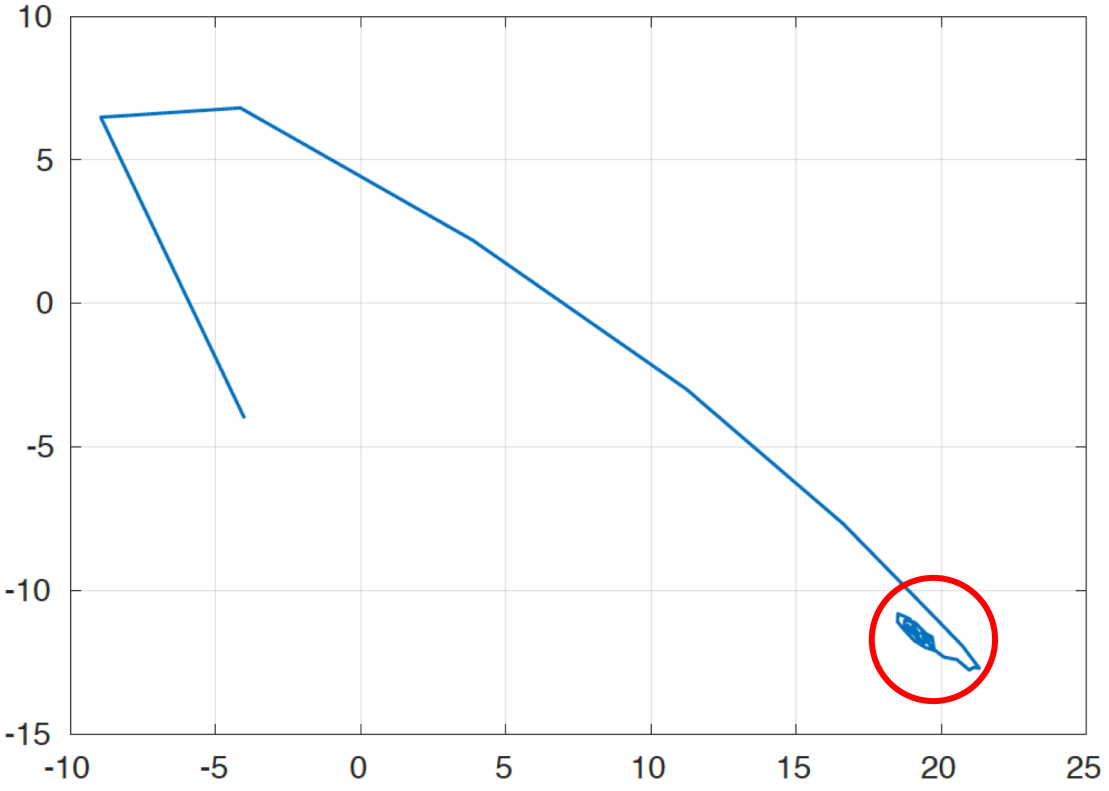
LDM modeling



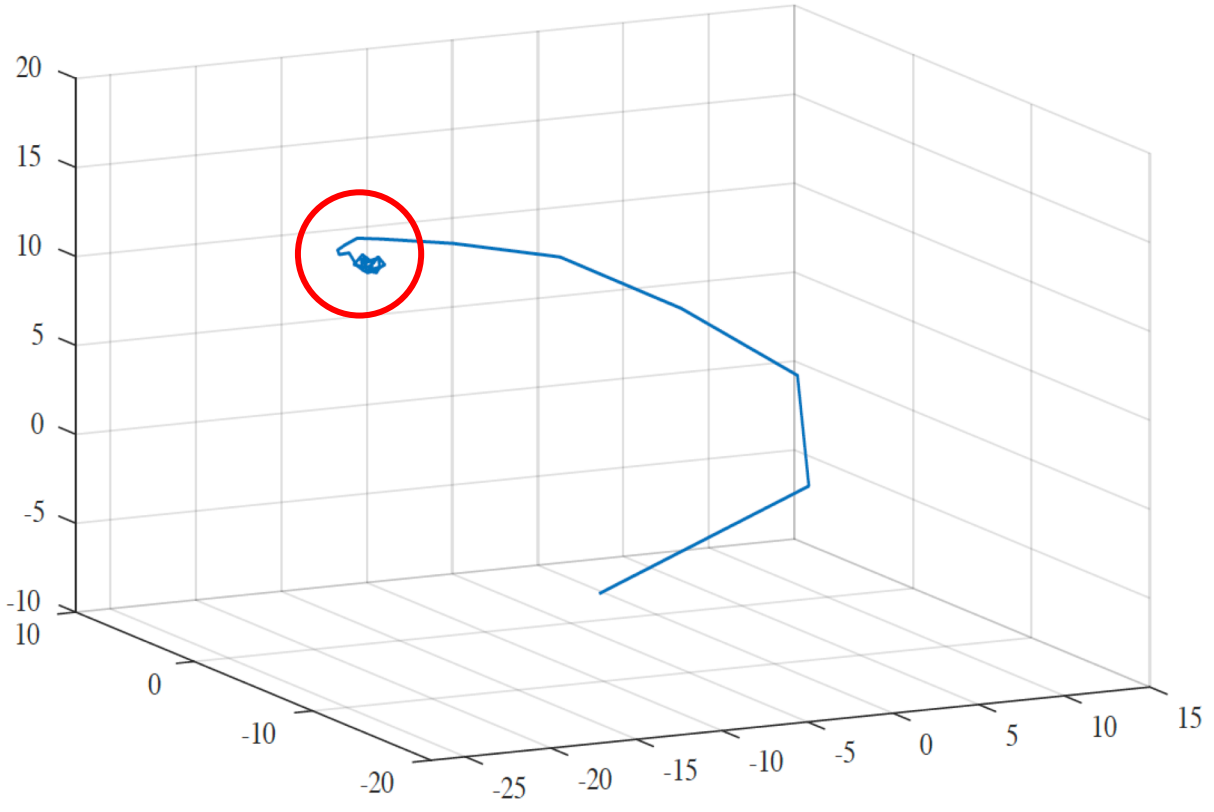
2 dimensional state space



3 dimensional observation space



2 dimensional state space

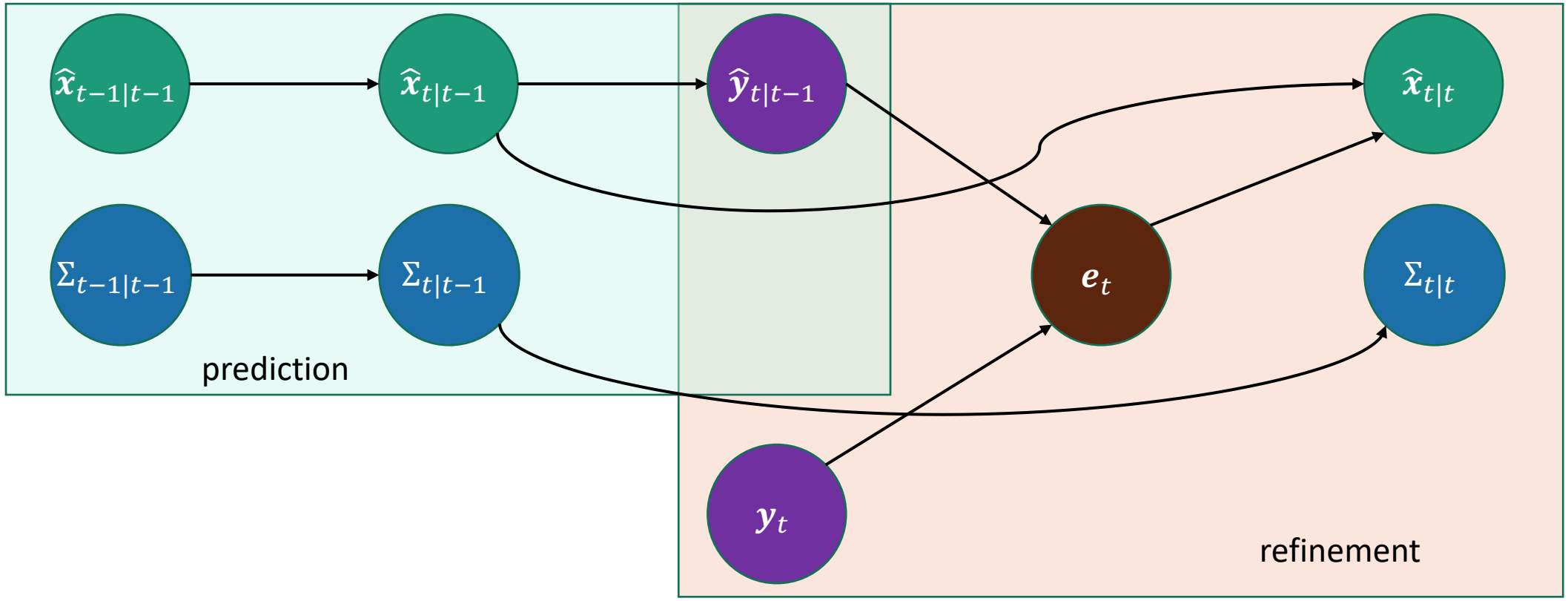


3 dimensional observation space

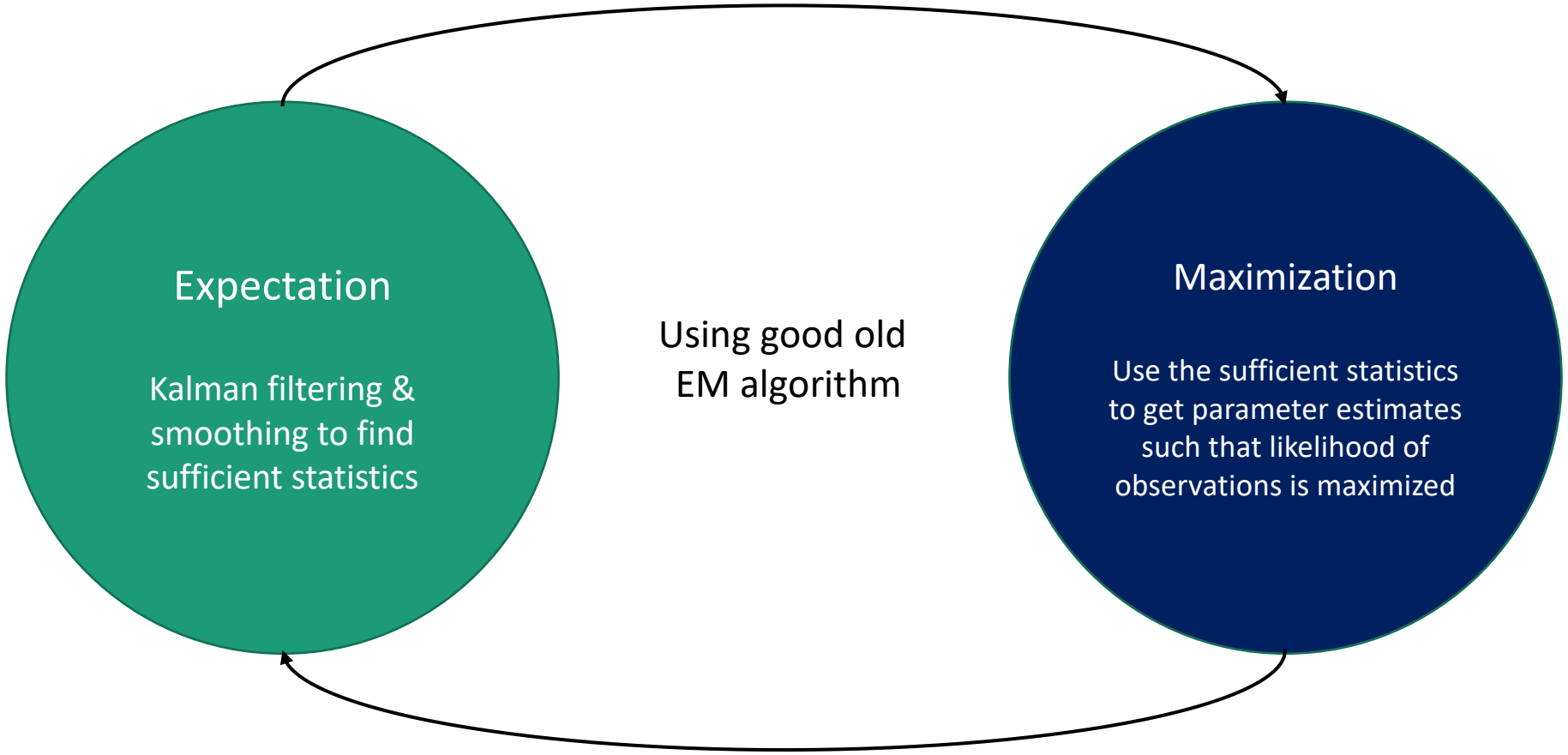
- State, and consequently observation, vector converges if state transition matrix is ‘well-behaved’

Kalman filtering

- Inferring state x_t given observations $\boldsymbol{y}_{1:t}$
- Similar is smoothing which infers $\boldsymbol{x}_t$ given all observations $\boldsymbol{y}_{1:T}$



Learning LDM parameters



Second order LDM

- Current state depends on two previous states
- For an second order LDM q

$$\boldsymbol{x}_t = F_q \boldsymbol{x}_{t-1} + G_q \boldsymbol{x}_{t-2} + \boldsymbol{w}_t \ , \qquad \boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{g}_q, Q_q)$$

$$\boldsymbol{y}_t = H_q \boldsymbol{x}_t + \boldsymbol{v}_t \ , \qquad \boldsymbol{v}_t \sim \mathcal{N}(\boldsymbol{\mu}_q, R_q)$$

$$\boldsymbol{x}_1 \sim \mathcal{N}(\boldsymbol{g}_{1,q}, Q_{1,q}) \ , \qquad \boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{g}_{0,q}, Q_{0,q})$$

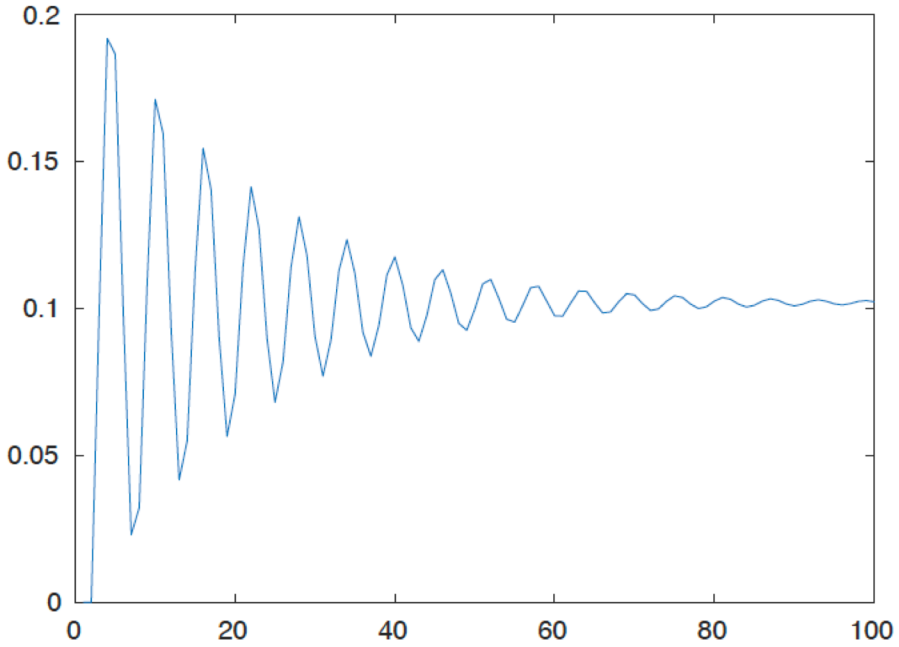
- Can be written as a first order LDM

$$\boldsymbol{x}'_t = F' \boldsymbol{x}'_{t-1} + \boldsymbol{w}_t \ , \qquad \boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{g}', Q')$$

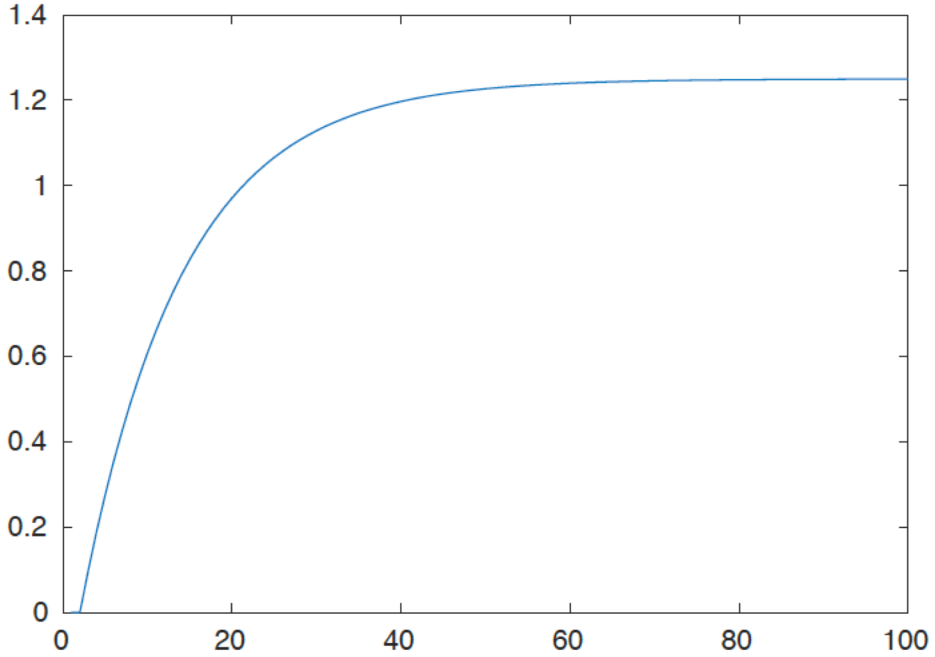
$$\text{where, } \boldsymbol{x}'_t = \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t-1} \end{bmatrix} , \qquad F' = \begin{bmatrix} F & G \\ \mathrm{I} & 0 \end{bmatrix} , \qquad \boldsymbol{g}' = \begin{bmatrix} \boldsymbol{g} \\ 0 \end{bmatrix} , \qquad Q' = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}$$

$$\boldsymbol{y}_t = H' \boldsymbol{x}'_t + \boldsymbol{v}_t$$

$$\text{where, } H' = \begin{bmatrix} H & 0 \end{bmatrix} , \quad \boldsymbol{v}_t \sim \mathcal{N}(\boldsymbol{\mu}, R)$$



Second order LDM trajectory (n = 1)



First order LDM trajectory (n = 1)

Why second order LDMS?

- Are more flexible
- Are more smooth
- Can reduce the number of parameters. It was found that SO-LDM with diagonal transition matrices F & G can provide results similar to full transition matrix of FO-LDM
- Thus reduction from n^2 to 2n per LDM

LDMs for TTS motivation

- The movement of the various articulators can be characterized by a critically damped spring-mass system.

$$\frac{\mathrm{d}^2 x(t)}{\mathrm{d}t^2} + 2\phi\frac{\mathrm{d}x(t)}{\mathrm{d}t} + \phi^2(x(t) - u) = w(t)$$

- This can be expressed as a discrete-time system

$$x_{t+1} = 2\phi x_t - \phi^2 x_{t-1} + (1-\phi)^2 u + w_t$$

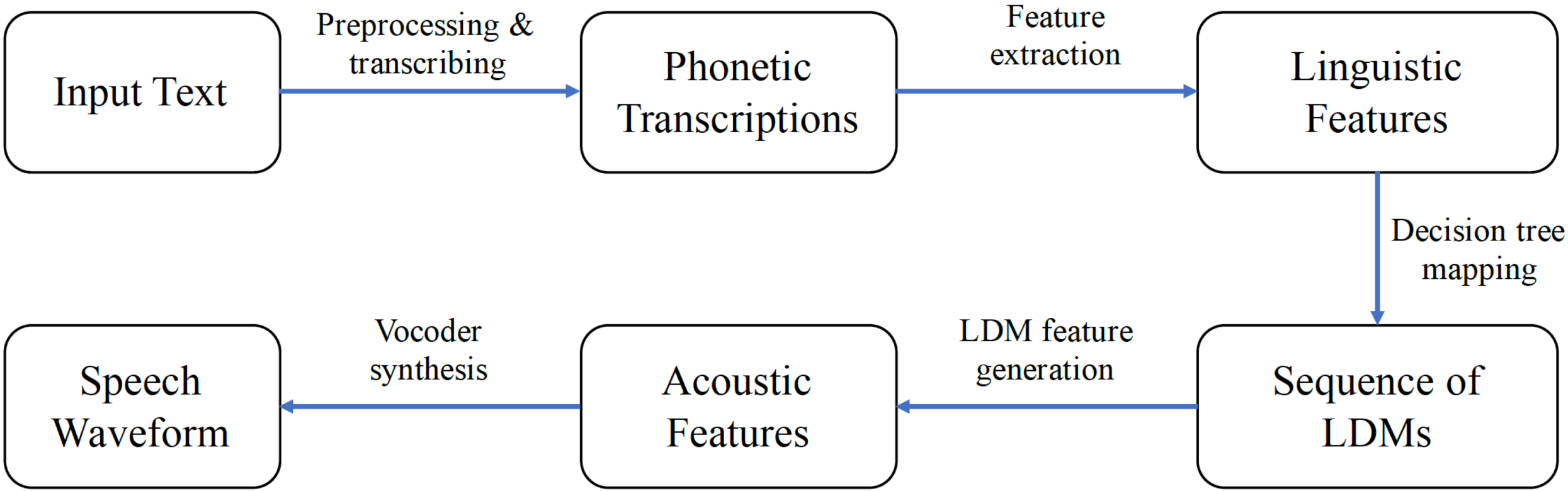
- If articulators are considered to operate in a lower dimensional subspace than acoustic features, we get:

$$\boldsymbol{x}'_t = F'\boldsymbol{x}'_{t-1} + \boldsymbol{w}_t \ , \qquad \boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{g}', Q')$$

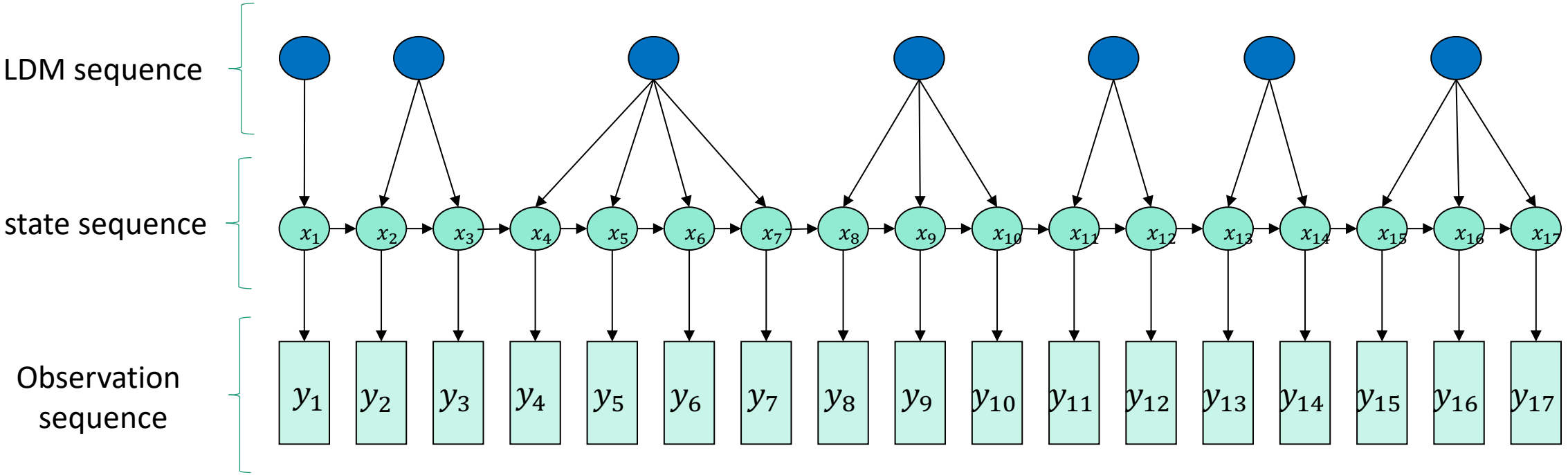
$$\text{where, } \boldsymbol{x}'_t = \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t-1} \end{bmatrix} , \qquad F' = \begin{bmatrix} 2\varphi & -\varphi^2 \\ \mathrm{I} & 0 \end{bmatrix} , \qquad \boldsymbol{g}' = \begin{bmatrix} (\mathrm{I}-\varphi)^2\boldsymbol{u} \\ 0 \end{bmatrix} ,$$

$$Q' = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} , \qquad \varphi = \mathrm{diag}(\phi_1, \phi_2, ..., \phi_n) \ , \qquad Q = \mathrm{diag}(q_1, q_2, ..., q_n)$$

LDMs for TTS


Input Text
Preprocessing & transcribing
Phonetic Transcriptions
Feature extraction
Linguistic Features
Decision tree mapping
Sequence of LDMs
LDM feature generation
Acoustic Features
Vocoder synthesis
Speech Waveform

Feature generation



An external duration model determines the number of frames for each LDM

Training LDMs for TTS

- Club together all the segments belonging to one LDM
- Train LDM on this set of segments
- HMMs are still convenient to force align and segment the training data
- Switching LDMs can be used for forced alignment and segmentation
- Decision tree clustering can be done using HMMs or LDMs (LDM clustering is very slow)

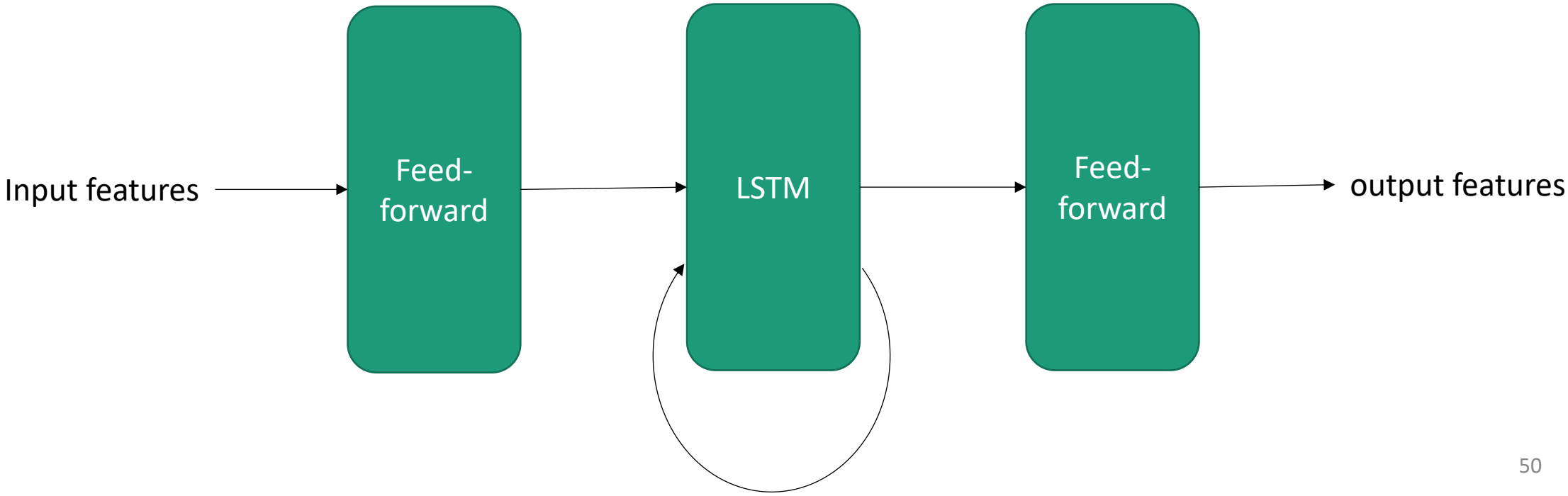
Neural speech synthesis

- For feed-forward models, replace the decision tree with neural network

- Linguistic feature sequence length is much shorter than acoustic feature sequence length
- Up-sample the input sequence to same time-scale as output sequence
- Get wider context by stacking together multiple inputs

Recurrent or hybrid models

- In order to get more context into account, use recurrent neural networks (LSTMs, GRUs, etc.) or a hybrid of the two



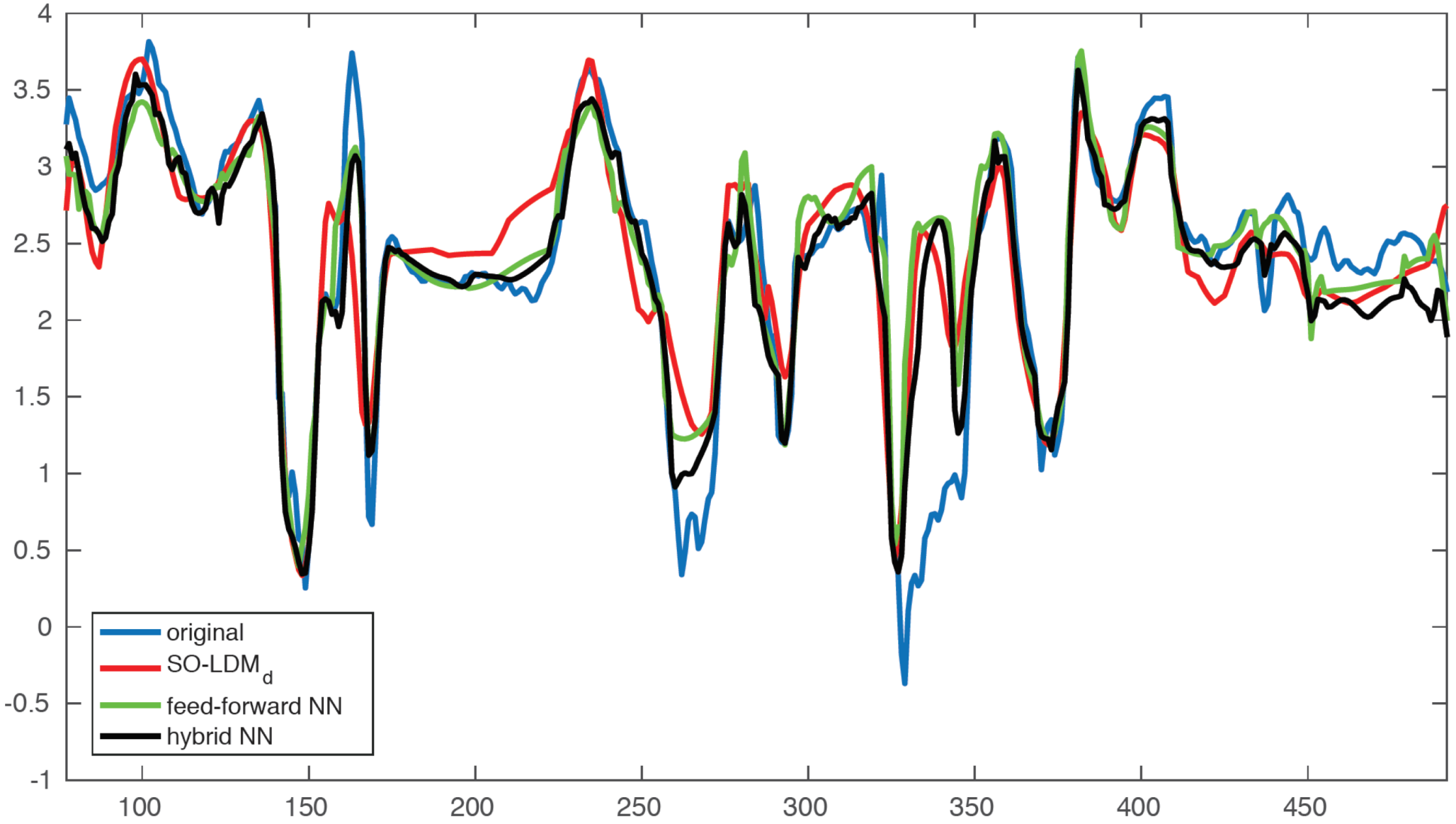
LDMs as simple RNNs

- LDMs have no non-linearity involved
- State progresses without any external input being fed into them at each time step
- LDMs are somewhat similar to a decoder in an encoder-decoder RNN architecture: x_1 can be considered encoded vector, but in LDMs, it is only fed in the beginning
- $\boldsymbol{x}_t = \sigma_x(F\boldsymbol{x}_{t-1} + \boldsymbol{g});\ \boldsymbol{y}_t = \sigma_y(H\boldsymbol{x}_t + \boldsymbol{\mu})$

Evaluation

Models were evaluated based on MCD and PESQ

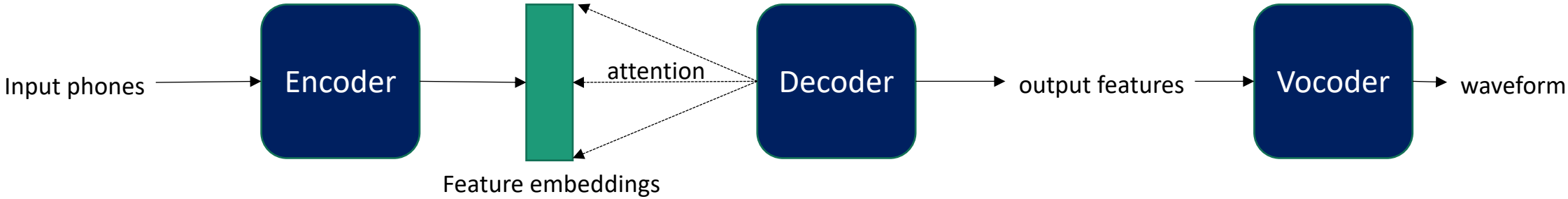
Model	Details	No. Param.	MCD	PESQ
2^{nd} order LDM	diagonal F, G	2,804,230	4.06	2.62
2^{nd} order LDM	full F, G	3,756,610	4.06	2.61
1^{st} order LDM	–	3,174,600	4.08	2.61
Autoregressive HMM	–	1,058,200	4.43	2.53
Feed-forward NN	4 layers $\times$ 1024 units	3,819,560	3.71	2.68
Feed-forward NN	6 layers $\times$ 512 units	1,648,680	3.71	2.69
Hybrid LSTM + FF	512 FF + 384 $\times$ 3 LSTM + 512 FF units	4,070,440	**3.62**	**2.73**



Example of parameter trajectory generated by various models

Some more neural speech synthesis models

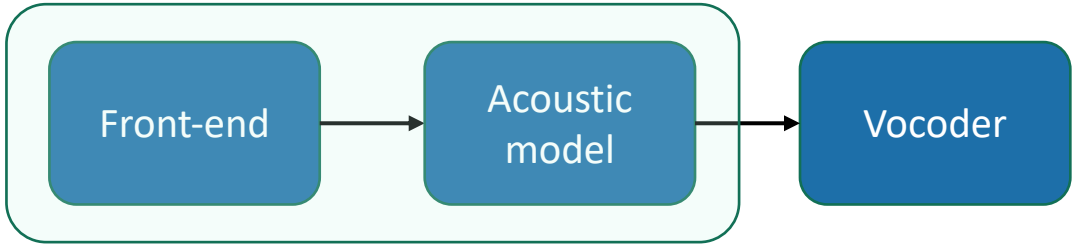
- Encoder decoder model with attention
- Encoder encodes the input phonemes or characters into some embeddings to remove the hand-engineered input features
- Attention corresponds to the alignment between input and output features (remember they operate on different time scales)
- Decoder serves as an acoustic model to generate acoustic features
- Advantages:
  - No need of input hand-engineered features
  - No explicit duration model
  - No need of external alignment
  - There could be dialect and speaker embeddings to be conditioned upon
- Examples: char2wav[1], voiceloop[2]

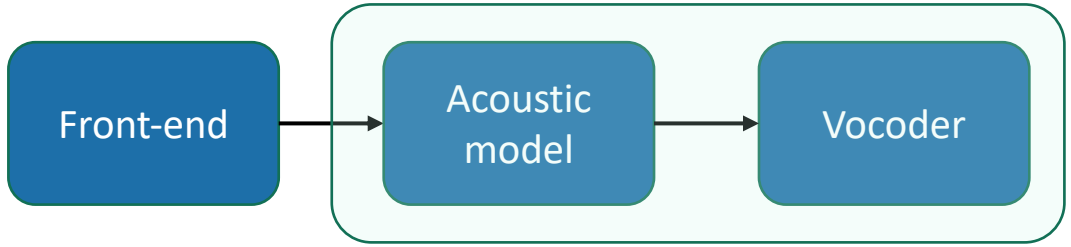


1 J. Sotelo, et al. Char2wav: End-to-end Speech Synthesis. (2017).
2 Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani. Voice synthesis for in-the-wild speakers via a phonological loop. CoRR, abs/1707.06588, 2017. URL http://arxiv.org/abs/1707.06588.

Combining various TTS components

- Traditional TTS system has three modules:
  - Front-end
  - Acoustic model
  - Vocoder
- Neural TTS models have tried to combine a few of them:
  - Tacotron[1] combines front end and acoustic model, so that it can take normalized text as input
  - Wavenet[2] combines acoustic model and vocoder, and produced raw waveform directly





1. Y. Wang, RJ. Skerry-Ryan, D. Stanton, Y. Wu, R.J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, Tacotron: Towards end-to-end speech synthesis. In Proceedings of Interspeech, August 2017
2. A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," CoRR, vol. abs/1609.03499, 2016.

Neural Vocoders

- Traditional vocoders can be replaced with neural vocoders.
- Input is some sort of acoustic features and output is a waveform
- Huge memory of neural vocoder prevents any artifacts which would normally occur in signal-processing based vocoders
- Most popular examples are Wavenet based vocoders. Wavenet uses dilated convolutions and produces samples autoregressively
- More recently there has been parallel Wavenet[1]
- Another example is SampleRNN[2]

1. A. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, H. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis. Parallel WaveNet: Fast high-fidelity speech synthesis. CoRR, abs/1711.10433, 2017.
2. S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville and Y. Bengio. Samplernn: An unconditional end-to-end neural audio generation model. arXiv preprint arXiv:1612.07837, 2016.

Conclusions

- Statistical parametric speech synthesis essentially maps input linguistic features to output acoustic features
- HMMs have been historically the most popular TTS models
- LDMs are more flexible can perform better than HMMs
- Neural networks synthesize speech of better quality, but LDMs require less working memory
- There has been a shift in paradigm from classical three stage pipeline to more ‘end-to-end’ systems

Thank You!