Linear Dynamical Models in Speech Synthesis

Vassilis Tsiaras
Technical University of Crete
July 2015

Contents

- Definition of LDMs
- Motivation of using LDMs
  - LDMs vs HMMs
  - Modelling capabilities
- LDMs
  - Properties
  - Inference
  - Learning
- Linguistic to phonetic mappings
  - Decision tree clustering
- Concatenation of speech parameter segments
- Global variance
- Implementation issues

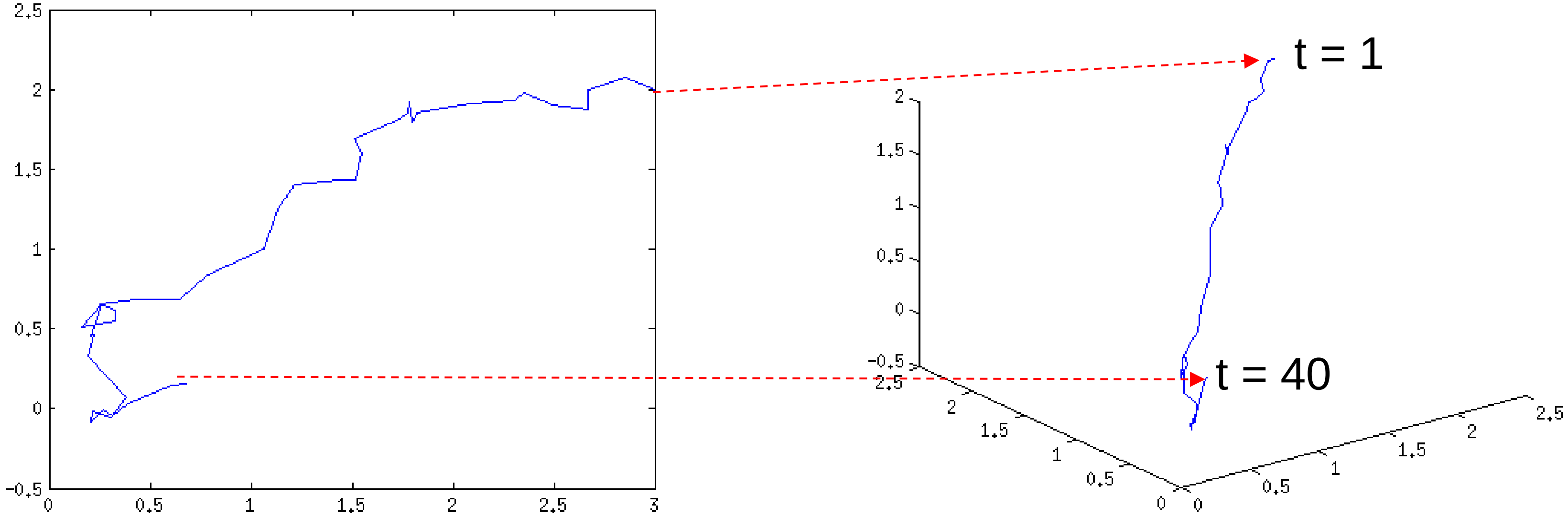
Linear Dynamical Models

- An LDM is a generative model with a time-varying multivariate unimodal Gaussian output distribution.
- An LDM is specified by the following pair of equations:

$$x_1 = N(g_1, Q_1)$$

$$x_t = Fx_{t-1} + g + w \qquad w \sim N(0,\ Q) \qquad x_t \in \mathbb{R}^n$$

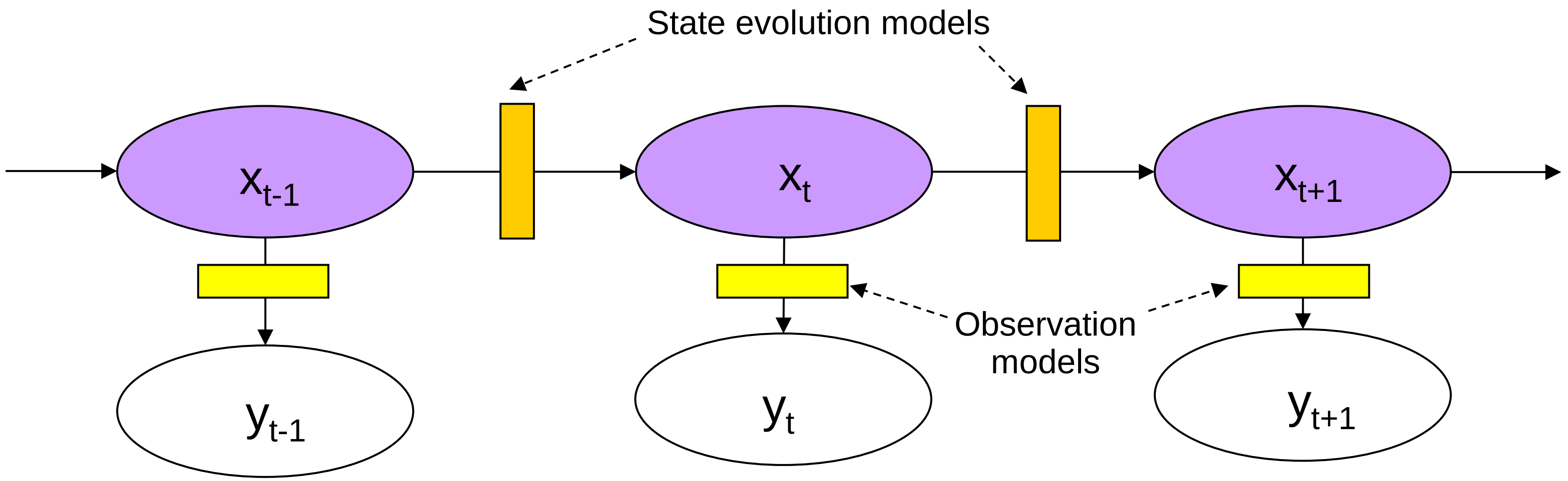
$$y_t = Hx_t + \mu + v \qquad v \sim N(0,\ R) \qquad y_t \in \mathbb{R}^m$$



Hidden state space (n=2) Observation space (m=3)

Linear Dynamical Models

- LDMs are described by a hidden Markov chain



- State evolution Model: $p(x_t/x_{t-1}) = \mathrm{N}(x_t; F \cdot x_{t-1} + g, Q)$
  - x_t: Abstract state, Articulators, Sinusoidals, e.t.c.

- Observation Model: $p(y_t/x_t) = \mathrm{N}(y_t; H \cdot x_t + \mu, R)$
  - y_t: (mceps, F0, bap, phi), Sinusoidal parameters, Raw speech, etc

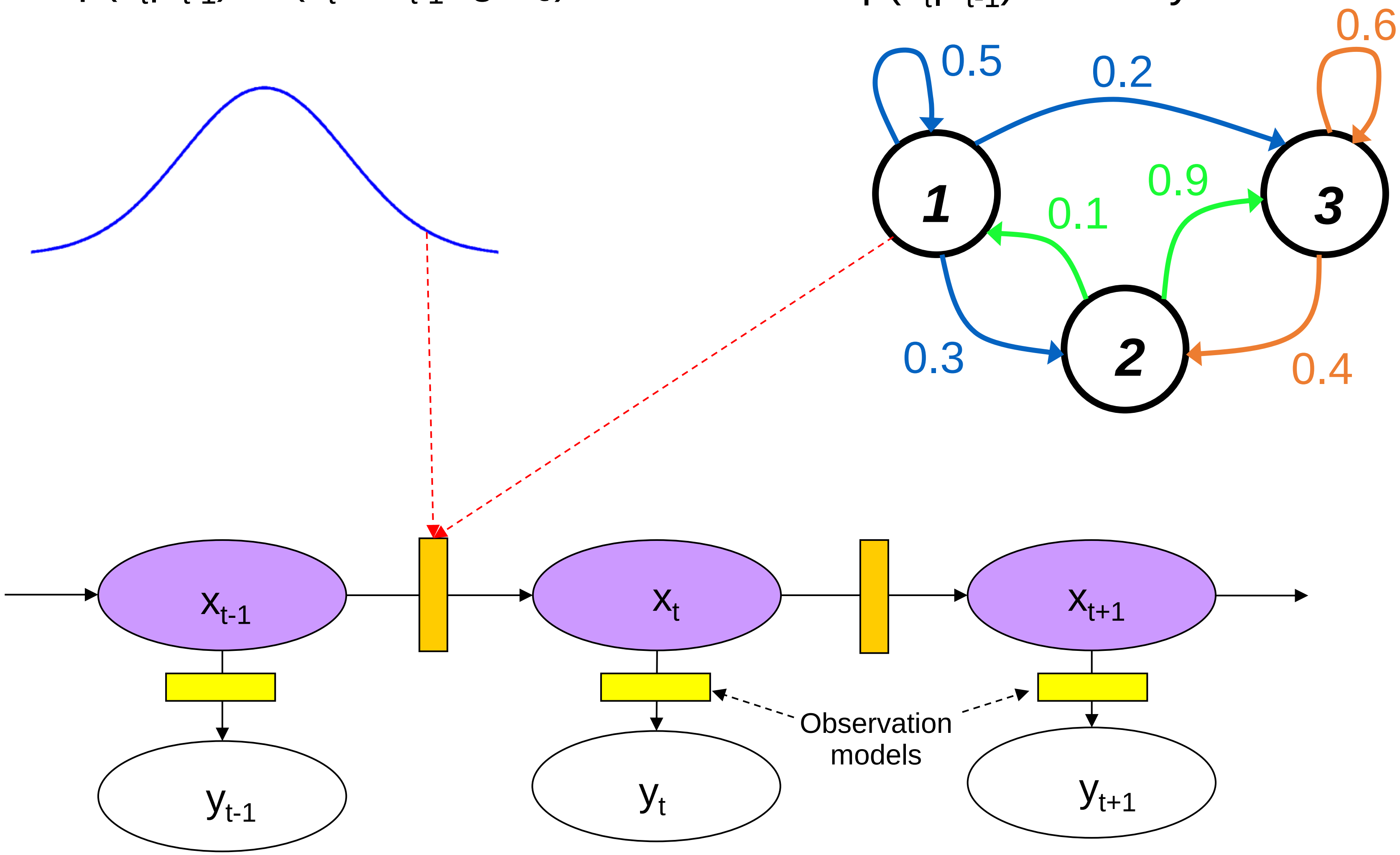
LDMs vs HMMs

LDM: x_t continuous vector

HMM: $x_t \in \{1, 2, \ldots, N\}$

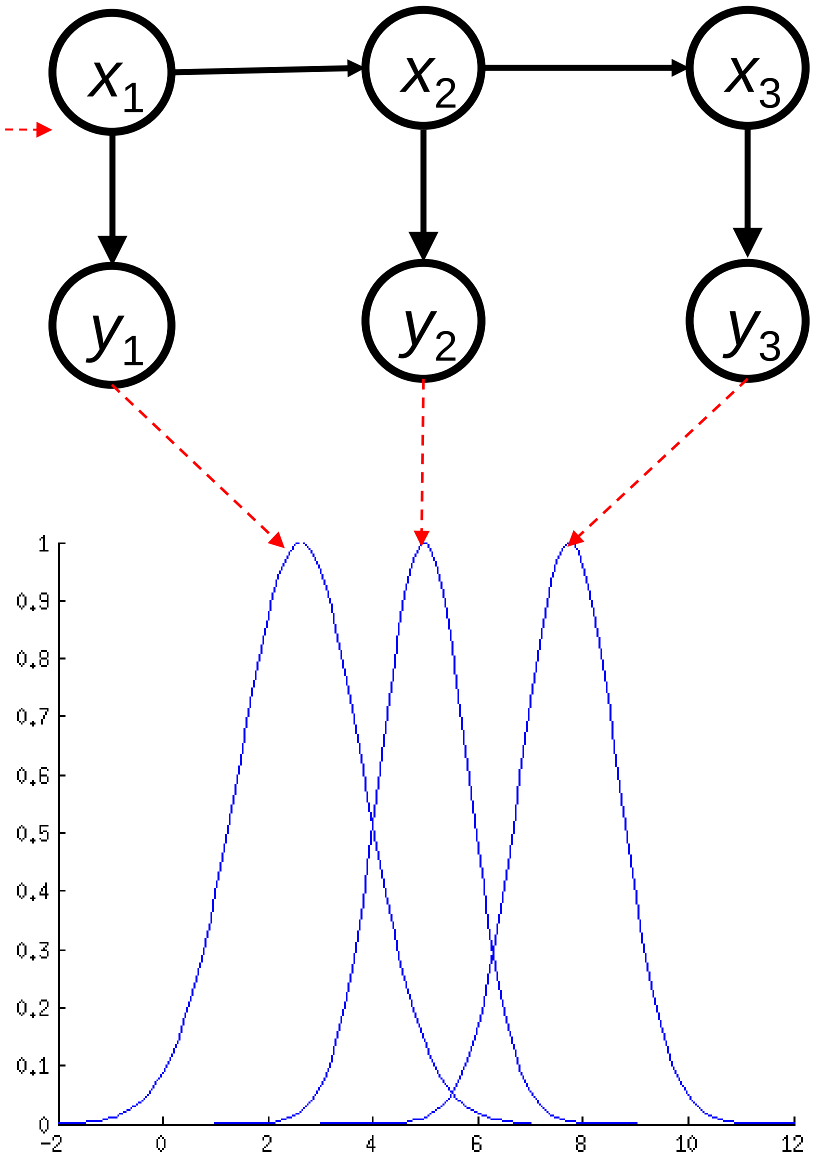
LDM: $p(x_t|x_{t-1})=N(x_t; Fx_{t-1}+g, Q)$

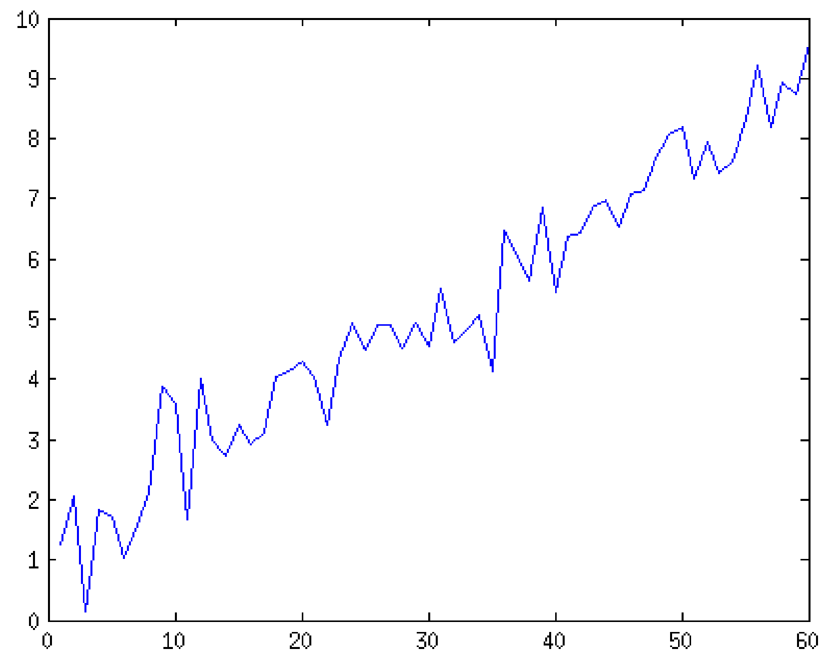
HMM: $p(x_t|x_{t-1})$ arbitrary



Modelling an artificial signal with HMMs and LDMs

- An artificial signal is modelled with
  - a 3-state HMM,
  - a 3-state trajectory HMM and
  - a single state LDM

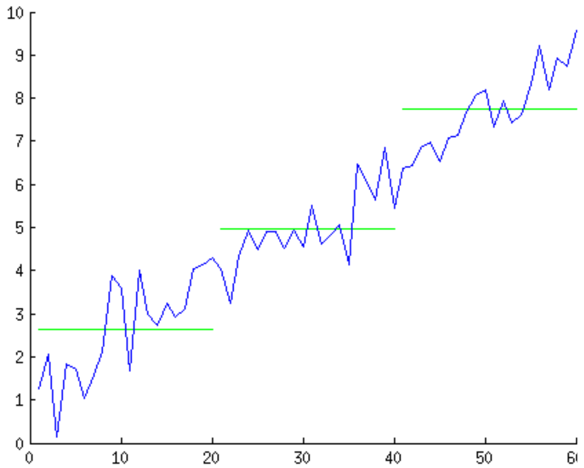




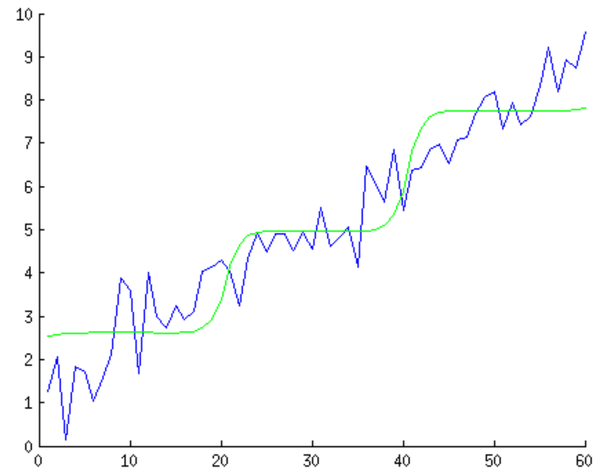
The signal:

x = linspace(1, 9, 60)+0.5*randn(1, 60)

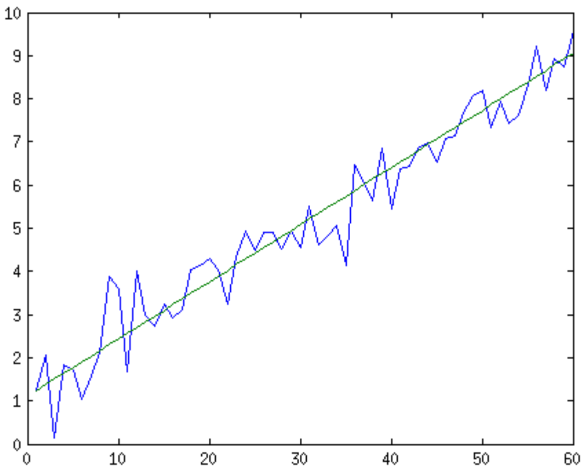
Synthesis with HMMs and LDMs



HMMs
Number of parameters
3 x (mean + std) +
transition matrix =
6 + 9 = 15



Trajectory HMMs
Number of parameters
3 x 3 x (mean + std) +
transition matrix =
18 + 9 = 27



LDM
Number of parameters
1 x (g1 + Q1 + F + g + Q
+ H + μ + R) =
8

In this example, an LDM generates a trajectory that is closer to the original using fewer parameters than 3-state HMMs or trajectory HMMs

LDMs and Autoregressive Models

- A p-order vector autoregressive (AR) model.

$$z_k = \sum_{i=1}^{p} A_i z_{k-i} + w$$

- The corresponding LDM

$$\begin{bmatrix} z_k \\ z_{k-1} \\ \vdots \\ z_{k-p+2} \\ z_{k-p+1} \end{bmatrix} = \begin{bmatrix} A_1 & A_2 & \cdots & A_{p-1} & A_p \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} z_{k-1} \\ z_{k-2} \\ \vdots \\ z_{k-p+1} \\ z_{k-p} \end{bmatrix} + \begin{bmatrix} w \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

$$x_{k+1} = F x_k + w'$$

$$y_k = x_k$$

LDMs and Sinusoidal Models

- Scalar noisy observations y_t of a periodic signal represented with a finite Fourier series plus a noise term

$$y_t = c_1 e^{j2\pi f_1 t} + c_2 e^{j2\pi f_2 t} + \cdots + c_k e^{j2\pi f_k t} + v$$

where the coefficients c_i are complex numbers

- By setting

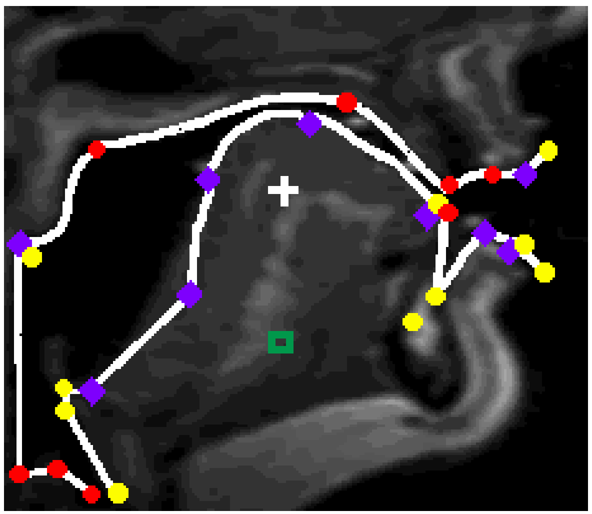
$$x_t = \begin{bmatrix} e^{j2\pi f_1 t} \\ \vdots \\ e^{j2\pi f_k t} \end{bmatrix}, \quad F = \begin{bmatrix} e^{j2\pi f_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{j2\pi f_k} \end{bmatrix}, \quad H = [c_1, c_2, \cdots, c_k]$$

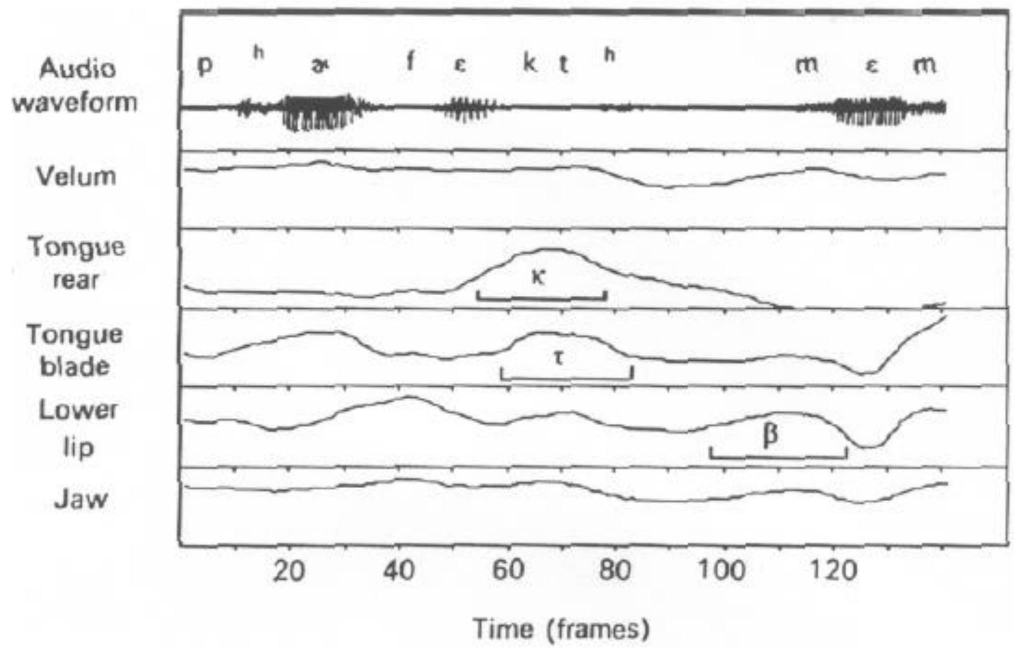
- The evolution of the periodic signal can be written as

$$x_t = F x_{t-1}$$
$$y_t = H x_t + v \quad v \sim N(0, R)$$

LDMs – State Evolution

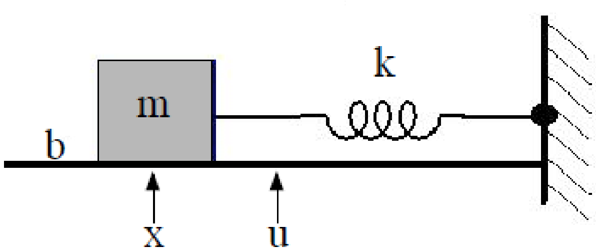
- Researchers at Haskins Laboratories developed differential equations that describe how the articulators move to produce a particular utterance.





- The motions of the articulators are simulated with critically-damped spring-mass models

$$\frac{d^2x(t)}{dt^2} + 2S\frac{dx(t)}{dt} + S^2(x(t) - u) = w$$



LDMs – State Evolution

- The differential equation

$$\frac{d^2 x(t)}{dt^2} + 2S\frac{dx(t)}{dt} + S^2(x(t) - u) = w$$

can be converted into a second order recurrence relation

- Therefore the motion of articulators can be connected with the dynamics of acoustic parameters with a State-Space Model

$$\begin{aligned} x_1 &= N(g_1, Q_1) & & & \\ x_t &= F x_{t-1} + g + w & w &\sim N(0, Q) & x_t &\in \mathbb{R}^n \\ y_t &= h(x_t) + v & v &\sim N(0, R) & y_t &\in \mathbb{R}^m \end{aligned}$$

- The hidden space variables, *x*, correspond to the states of articulators
- The observation space variables, *y*, correspond to speech parameters
- However the mapping between the two spaces may be non-linear

LDMs – Factor Analysis

- Factor analysis is a statistical method for modelling the covariance structure of high dimensional static data using a small number of latent (hidden) variables

$$x = w \qquad w \sim N(0, I) \qquad w \in \mathbb{R}^n$$

$$y = Hx + v \qquad v \sim N(\mu, R) \qquad y, v \in \mathbb{R}^m, \qquad R \text{ is diagonal}$$

- Number of parameters: $m + m \times n$, instead of $m \times m$ of a full *R*.
- Example n = 1, m = 2

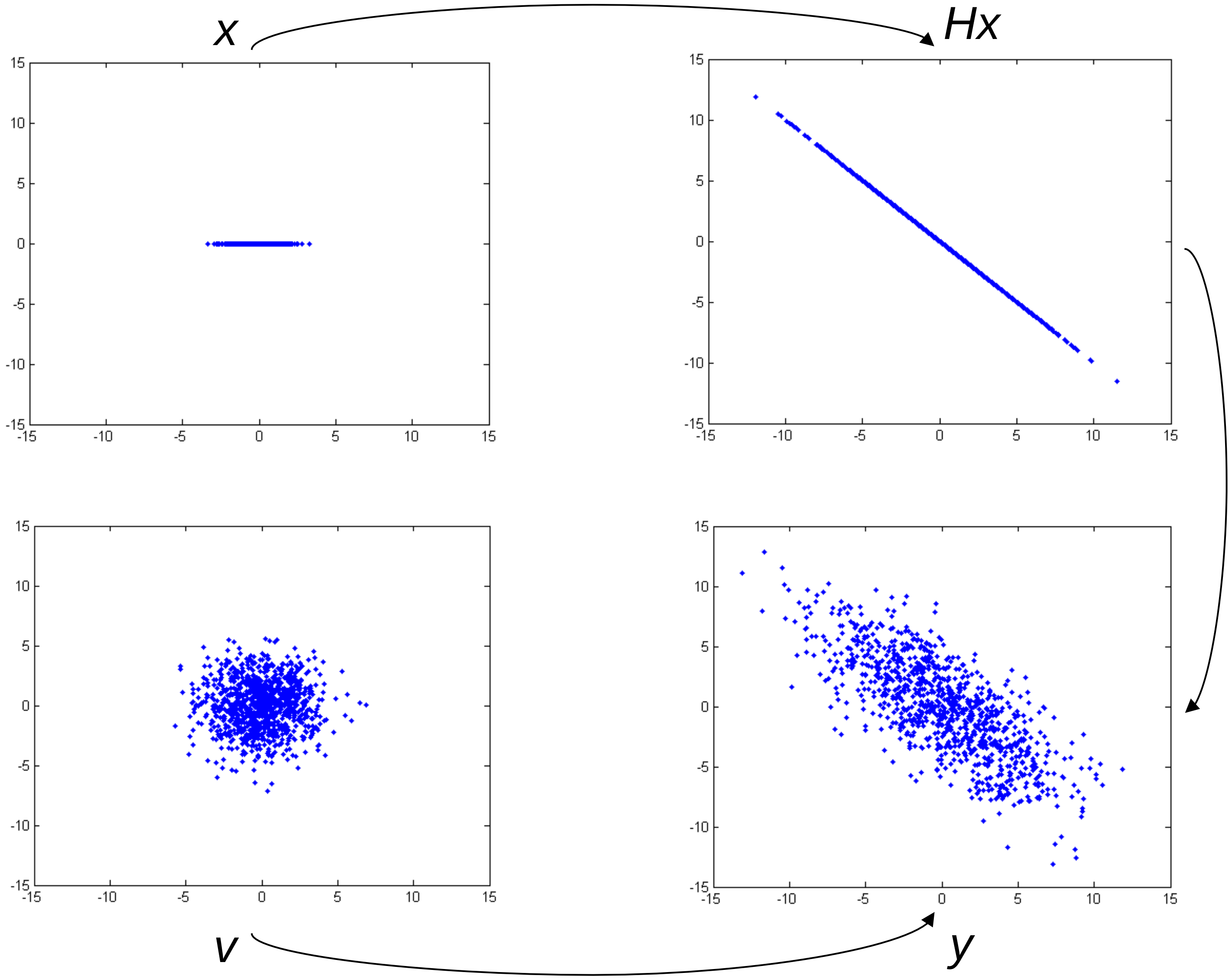
$$w \sim N(0,1)$$

$$v \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\right)$$

$$H = 5 * \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

H scales *x* by 5 and rotates it by 45° clockwise

LDMs – Factor Analysis



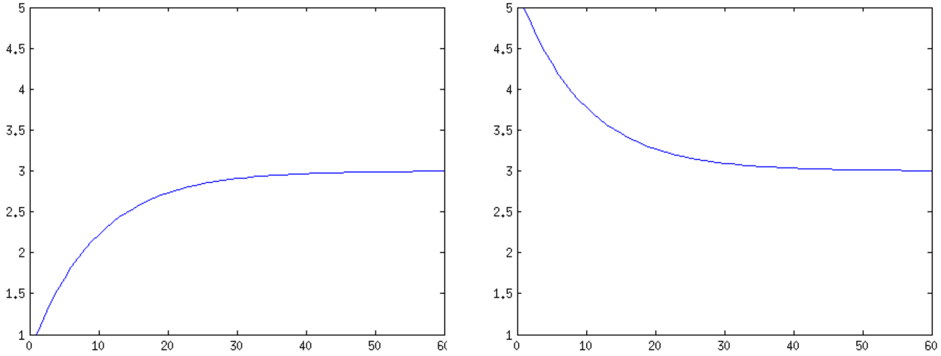
LDMs – Dynamics

- LDMs are stochastic models and can explain a huge number of time series using a small number of parameters
- The deterministic part of the dynamics of a first-order LDM is:

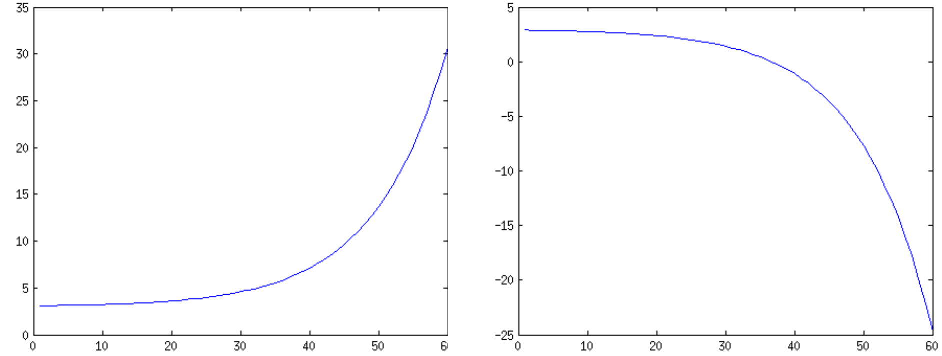
$$x_t = Fx_{t-1} + g$$

- In speech synthesis, stable models should be used.
  - The transition matrix F is constrained to have spectral radius less than one (All the eigenvalues of F have absolute values less than one).
- Target value of a stable LDM: $(I - F)^{-1}g$

Examples of trajectories of $x_t = Fx_{t-1} + g$



Stable first-order LDMs. Target value=3



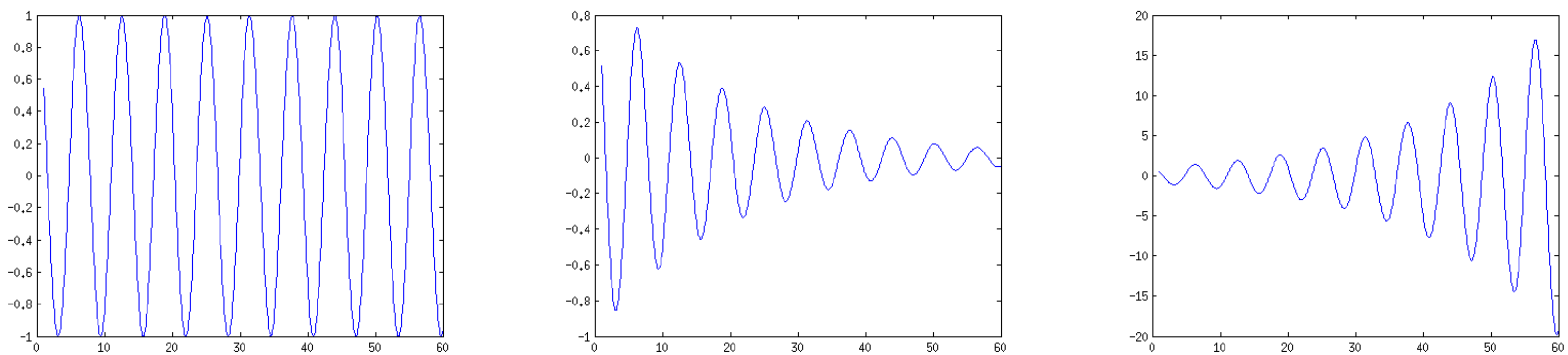
Unstable first-order LDMs.

LDMs – Dynamics

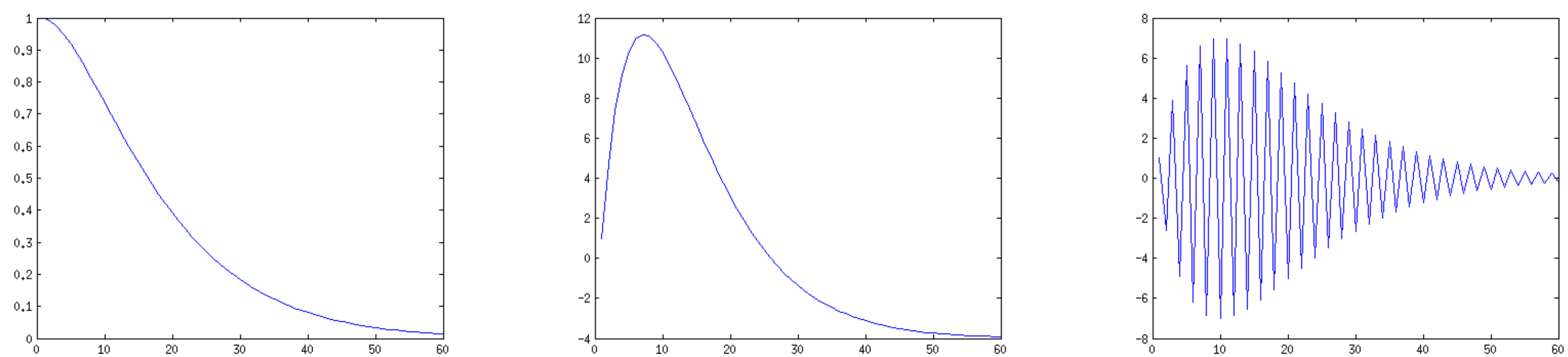
- The deterministic part of the dynamics of a second-order LDM is:

$$x_t = F_1 x_{t-1} + F_2 x_{t-2} + g$$

- The above recurrence relation can give oscillations



- Trajectories of second-order critically dumped linear dynamics



Second-order critically dumped LDMs.
They are stable and converge to a target value

The tree basic problems for HMMs and LDMs

- **Evaluation:** Given an LDM with parameters θ and an observation sequence $Y = [y_1, y_2, \cdots, y_T]$, calculate the probability that model θ has generated sequence Y.

- **Inference:** Given an LDM with parameters θ and an observation sequence $Y = [y_1, y_2, \cdots, y_T]$, calculate the probability of hidden states x_t that produced this observation sequence Y.

- **Learning:** Given a training observation sequence $Y = [y_1, y_2, \cdots, y_T]$, determine an LDM with parameters θ that best fit the training data.

LDMs - Inference

- We assume that the parameters θ of an LDM are known.
- There are two approaches, in order to infer the hidden state sequence $\mathrm{X} = [x_1, x_2, \cdots, x_T]$ statistics from an observation sequence $Y = [y_1, y_2, \cdots, y_T]$.

1. Solving a weighted least squares problem
   - Derivation of square root Kalman filter

2. Using the properties of Gaussian distributions and of Markov chain of probabilistic interactions
   - The equations and the algorithms are similar to HMM case
   - This method can be used to derive equations and recursive algorithms for any distribution of the exponential family (Gaussian, exponential, alpha-stable, …)

LDMs - Inference

- From the equations of LDM

$$x_1 = g_1 + w_1 \qquad w_1 \sim N(0,\ Q_1) \qquad x_1 \in \mathbb{R}^n$$

$$x_t = Fx_{t-1} + g + w \qquad w \sim N(0,\ Q) \qquad x_t \in \mathbb{R}^n$$

$$y_t = Hx_t + \mu + v \qquad v \sim N(0,\ R) \qquad y_t \in \mathbb{R}^m$$

it follows that

$$x_1 = g_1 + w_1$$
$$Hx_1 = y_1 - \mu - v$$
$$Fx_1 - x_2 = -g - w$$
$$Hx_2 = y_2 - \mu - v$$
$$\cdots$$
$$Fx_{T-1} - x_T = -g - w$$
$$Hx_T = y_T - \mu - v$$

$$\begin{bmatrix} I & 0 & 0 & \cdots & 0 & 0 \\ H & 0 & 0 & \cdots & 0 & 0 \\ F & -I & 0 & \cdots & 0 & 0 \\ 0 & H & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & F & -I \\ 0 & 0 & 0 & \cdots & 0 & H \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_T \end{bmatrix} = \begin{bmatrix} g_1 \\ y_1 - \mu \\ -g \\ y_2 - \mu \\ \vdots \\ -g \\ y_T - \mu \end{bmatrix} - \begin{bmatrix} w_1 \\ v \\ w \\ v \\ \vdots \\ w \\ v \end{bmatrix}$$

$$Ax = b - \varepsilon \Rightarrow \varepsilon = b - Ax, \qquad A \in \mathbb{R}^{(n+m)T \times nT}$$

$$\underset{x}{\text{minimize}}\ E[\varepsilon^T \varepsilon] \Rightarrow A^T \Sigma^{-1} A x = A^T \Sigma^{-1} b$$

LDMs - Inference

- **Weighted least squares problem**

$$Ax = b - \varepsilon \Rightarrow \varepsilon = b - Ax \qquad A \in \mathbb{R}^{2Tn \times Tn}$$

$$\underset{x}{\text{minimize}} \; \|\varepsilon\|^2 \Rightarrow A^T \Sigma^{-1} A x = A^T \Sigma^{-1} b$$

Normal equations

- Naïve solution $x = (A^T \Sigma^{-1} A)^{-1} A^T \Sigma^{-1} b$
- Matrix $A^T \Sigma^{-1} A$ is block tri-diagonal.
  - The structure of matrix $A^T \Sigma^{-1} A$ allows recursive solution.
  - Solving the system $A^T \Sigma^{-1} A x = A^T \Sigma^{-1} b$ using LU decomposition of $A^T \Sigma^{-1} A$ leads to Kalman filter
  - Solving the system $A^T \Sigma^{-1} A x = A^T \Sigma^{-1} b$ using orthogonalization of $A^T \Sigma^{-1} A$, e.g., QR decomposition, leads to square root Kalman filter

LDMs - Inference

- Derivation of Kalman filter based on the properties of Gaussian distribution and the properties of the probabilistic interactions.

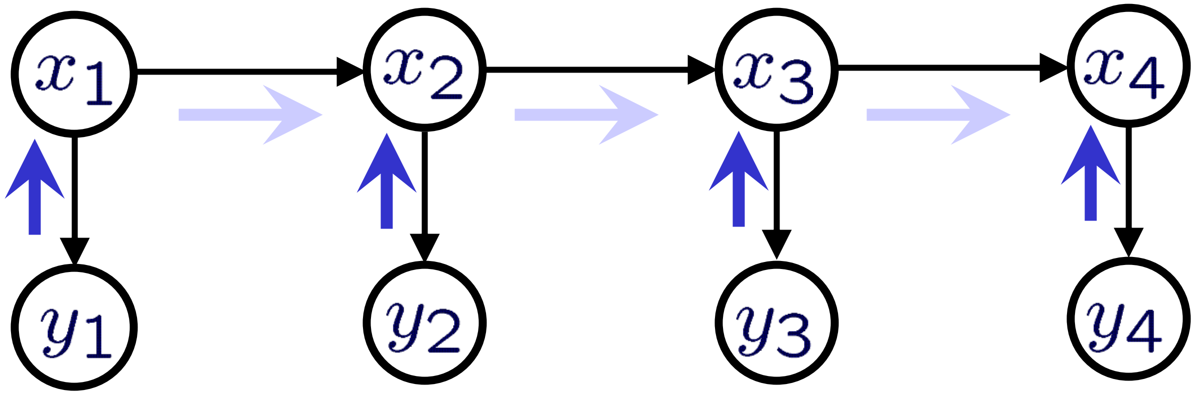
- Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ be an n-dimensional random vector with distribution $x \sim N(\mu, \Sigma)$, where x_1 and x_2 are two sub-vectors of respective dimensions p and q, with p+q = n, $\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$, $\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$

- **Theorem**
- The marginal distributions of x_1 and x_2 are also normal with mean vector μ_i and covariance matrix Σ_{ii} (i=1,2), respectively.
- The conditional distribution of x_i given x_j is also normal with mean vector $\mu_{i|j} = \mu_i + \Sigma_{ij}\Sigma_{jj}^{-1}(x_j - \mu_j)$

  and covariance matrix $\Sigma_{i|j} = \Sigma_{ii} - \Sigma_{ij}\Sigma_{jj}^{-1}\Sigma_{ij}^T$

LDMs - Inference

- Filtering



$$\alpha_t(x_t) \triangleq p(x_t \mid y_1, \ldots, y_t)$$

$$\hat{\alpha}_t(x_t) = \frac{1}{c_t} p(y_t | x_t) \int p(x_t | x_{t-1} = z)\, \hat{\alpha}_{t-1}(z) dz$$

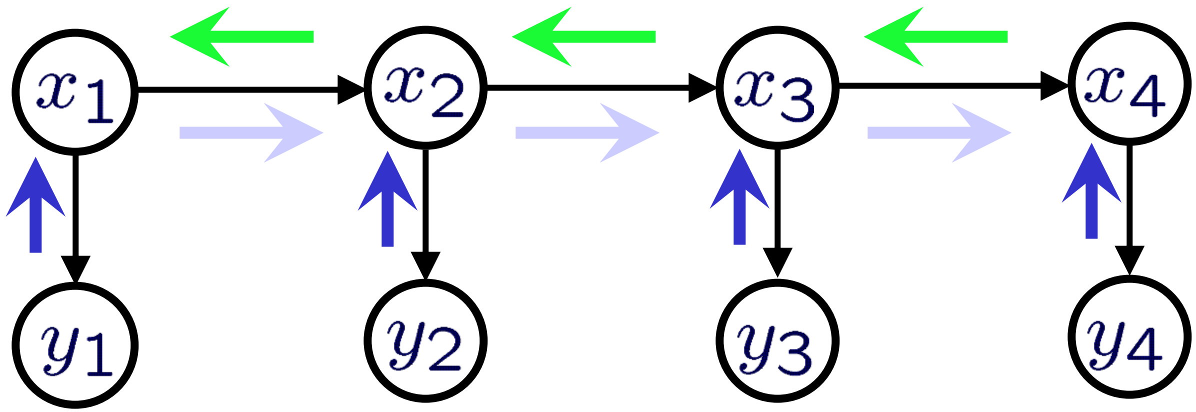
Normalization constant

Prediction: $p(x_t \mid y_1, \ldots, y_{t-1})$

Update: $p(x_t \mid y_1, \ldots, y_t)$

LDMs - Inference

- Smoothing



$$p(x_t \mid y) \propto \underbrace{p(x_t \mid y_1, \ldots, y_t)}_{\alpha_t(x_t)} \underbrace{p(y_{t+1}, \ldots, y_T \mid x_t)}_{\beta_t(x_t)}$$

- The *forward-backward* algorithm updates filtering via a *reverse-time* recursion:

$$\hat{\beta}_{t-1}(x_{t-1}) = \frac{1}{c_t} \int p(x_t = z | x_{t-1}) p(y_t | x_t = z) \hat{\beta}_t(z) dz$$

LDMs - Inference

- Smoothing
  - Backward recursion

$$\hat{\beta}_{t-1}(x_{t-1}) = \frac{1}{c_t}\int p(x_t = z|x_{t-1})p(y_t|x_t = z)\hat{\beta}_t(z)dz$$

  - Sequential recursion

$$\hat{\alpha}_{t-1}(x_{t-1})\hat{\beta}_{t-1}(x_{t-1}) = \int p(x_{t-1}|x_t = z, y_{1:t-1})\hat{\alpha}_t(z)\hat{\beta}_t(z)dz$$

- For the learning problem, the following marginal probabilities are inferred from the observation

$$p(x_t|Y) = \frac{1}{c_t}\hat{\alpha}_t(x_t)\hat{\beta}_t(x_t)$$

$$p(x_{t-1}, x_t|Y) = \frac{1}{c_t}\hat{\alpha}_{t-1}(x_{t-1})p(x_t|x_{t-1})p(y_t|x_t)\hat{\beta}_t(x_t)$$

The set of Kalman filtering equations

Prediction (Time Update)

(1) Project the state ahead

$$\hat{x}_{t|t-1} = F\hat{x}_{t-1|t-1} + g$$

(2) Project the error covariance ahead

$$\hat{\Sigma}_{t|t-1} = F\hat{\Sigma}_{t-1|t-1}F^T + Q$$

Correction (Measurement Update)

(1) Compute the Kalman Gain

$$K_t = \hat{\Sigma}_{t|t-1}H^T\left(H\hat{\Sigma}_{t|t-1}H^T + R\right)^{-1}$$

(2) Update estimate with measurement y_t

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t\left(y_t - H\hat{x}_{t|t-1} - \mu\right)$$

(3) Update Error Covariance

$$\hat{\Sigma}_{t|t} = \hat{\Sigma}_{t|t-1} - K_t H\hat{\Sigma}_{t|t-1}$$

Algorithm 5: Kalman Filter

Data: Observations, $y_{1:T}$, and model parameters: $F, g, Q, H, \mu, R, g_1, Q_1$

Result: logL $= \log(p(y_{1:T}))$ and statistics $\hat{x}_{t|t}$, $\hat{\Sigma}_{t|t}$, $t \in \{1, \ldots, T\}$,
$\hat{x}_{t|t-1}$, $\hat{\Sigma}_{t|t-1}$, $t \in \{2, \ldots, T\}$

/* Initialization */

$\hat{x}_{t|t-1} = g_1; \quad \hat{\Sigma}_{t|t-1} = Q_1; \quad \text{logL} = 0$

for t = *1:T* **do**

  /* Prediction */

  if $t > 1$ **then**

    $\hat{x}_{t|t-1} = F\hat{x}_{t-1|t-1} + g$

    $\hat{\Sigma}_{t|t-1} = F\hat{\Sigma}_{t-1|t-1}F^T + Q$

  /* Update */

  $e_t = y_t - (H\hat{x}_{t|t-1} + \mu)$

  $\hat{\Sigma}_{e_t} = H\hat{\Sigma}_{t|t-1}H^T + R$

  $K_t = \hat{\Sigma}_{t|t-1}H^T\hat{\Sigma}_{e_t}^{-1}$

  $\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t e_t$

  $\hat{\Sigma}_{t|t} = \hat{\Sigma}_{t|t-1} - K_t H \hat{\Sigma}_{t|t-1}$

  $\text{logL} = \text{logL} + \log(\mathcal{N}(e_t; 0, \hat{\Sigma}_{e_t}))$ /* $c_t = \mathcal{N}(e_t; 0, \hat{\Sigma}_{e_t})$ */

Algorithm 6: Kalman Smoother

Data: Statistics $\hat{x}_{t|t}$, $\hat{\Sigma}_{t|t}$, $\hat{x}_{t|t-1}$, $\hat{\Sigma}_{t|t-1}$ calculated from Kalman filter, and model parameter F

Result: Statistics $\hat{x}_{t|T}$, $\hat{R}_{t|T}$, $t \in \{1, \dots, T\}$ and $\hat{R}_{t,t-1|T}$, $t \in \{2, \dots, T\}$

$$\hat{R}_T = \hat{\Sigma}_{T|T} + \hat{x}_{T|T}\hat{x}_{T|T}^T$$

for t = T:2 **do**

$$J_t = \hat{\Sigma}_{t-1|t-1}F^T\hat{\Sigma}_{t|t-1}^{-1}$$
$$\hat{x}_{t-1|T} = \hat{x}_{t-1|t-1} + J_t(\hat{x}_{t|T} - \hat{x}_{t|t-1})$$
$$\hat{\Sigma}_{t-1|T} = \hat{\Sigma}_{t-1|t-1} + J_t(\hat{\Sigma}_{t|T} - \hat{\Sigma}_{t|t-1})J_t^T$$
$$\hat{\Sigma}_{t,t-1|T} = J_t\hat{\Sigma}_{t|T}$$

$$\hat{R}_{t-1|T} = \hat{\Sigma}_{t-1|T} + \hat{x}_{t-1|T}\hat{x}_{t-1|T}^T$$
$$\hat{R}_{t,t-1|T} = \hat{\Sigma}_{t,t-1|T} + \hat{x}_{t|T}\hat{x}_{t-1|T}^T$$

LDMs - Learning

- The parameters of an autoregressive (AR) model can be specified by solving closed form equations (e.g., the Yule-Walker equations).
- There is no closed form solution to parameter identification in LDMs.
- Parameters can be estimated by minimizing the log-likelihood

$$
\begin{aligned}
Q(\theta_i, \theta) = const - \frac{1}{2}\log|Q_1| - \frac{1}{2}E\left[(x_1 - g_1)^T Q_1^{-1}(x_1 - g_1)|Y, \theta_i\right] - \frac{T-1}{2}\log|Q| \\
- \frac{1}{2}\sum_{t=2}^{T} E\left[(x_t - Fx_{t-1} - g)^T Q^{-1}(x_t - Fx_{t-1} - g)|Y, \theta_i\right] \\
- \frac{T}{2}\log|R| - \frac{1}{2}\sum_{t=1}^{T} E\left[(y_t - Hx_t - \mu)^T R^{-1}(y_t - Hx_t - \mu)|Y, \theta_i\right] \quad (54)
\end{aligned}
$$

- Numerical optimization algorithms
  - Steepest ascent
  - Expectation maximization algorithm

LDMs - Learning

- **EM-algorithm**

  - Repeat until convergence
    - **E-step:** Given an estimate of the parameters of the model, compute the sufficient statistics,

      and the expected log-likelihood

    - **M-step:** Update the parameters of the model

LDMs - Learning

- **E-step:** Smoothed state estimates

$$E[x_t|y_{1:T}] = \hat{x}_{1|T}$$

$$E[x_t x_t^T|y_{1:T}] = \hat{\Sigma}_{t|T} + \hat{x}_{t|T}\hat{x}_{t|T}^T = \hat{R}_{t|T}$$

$$E[x_t x_{t-1}^T|y_{1:T}] = \hat{\Sigma}_{t,t-1|T} + \hat{x}_{t|T}\hat{x}_{t-1|T}^T = \hat{R}_{t,t-1|T}$$

- Sufficient statistics

$$\zeta_1 = \sum_{t=1}^{T-1} \hat{x}_{t|T}$$

$$\zeta_2 = \sum_{t=2}^{T} \hat{x}_{t|T}$$

$$\zeta_3 = \sum_{t=1}^{T} \hat{x}_{t|T}$$

$$\zeta_4 = \sum_{t=1}^{T} y_t$$

$$\Gamma_1 = \sum_{t=1}^{T-1} \hat{R}_{t|T}$$

$$\Gamma_2 = \sum_{t=2}^{T} \hat{R}_{t|T}$$

$$\Gamma_3 = \sum_{t=1}^{T} \hat{R}_{t|T}$$

$$\Gamma_4 = \sum_{t=2}^{T} \hat{R}_{t,t-1|T}$$

$$\Gamma_5 = \sum_{t=1}^{T} y_t \hat{x}_{t|T}^T$$

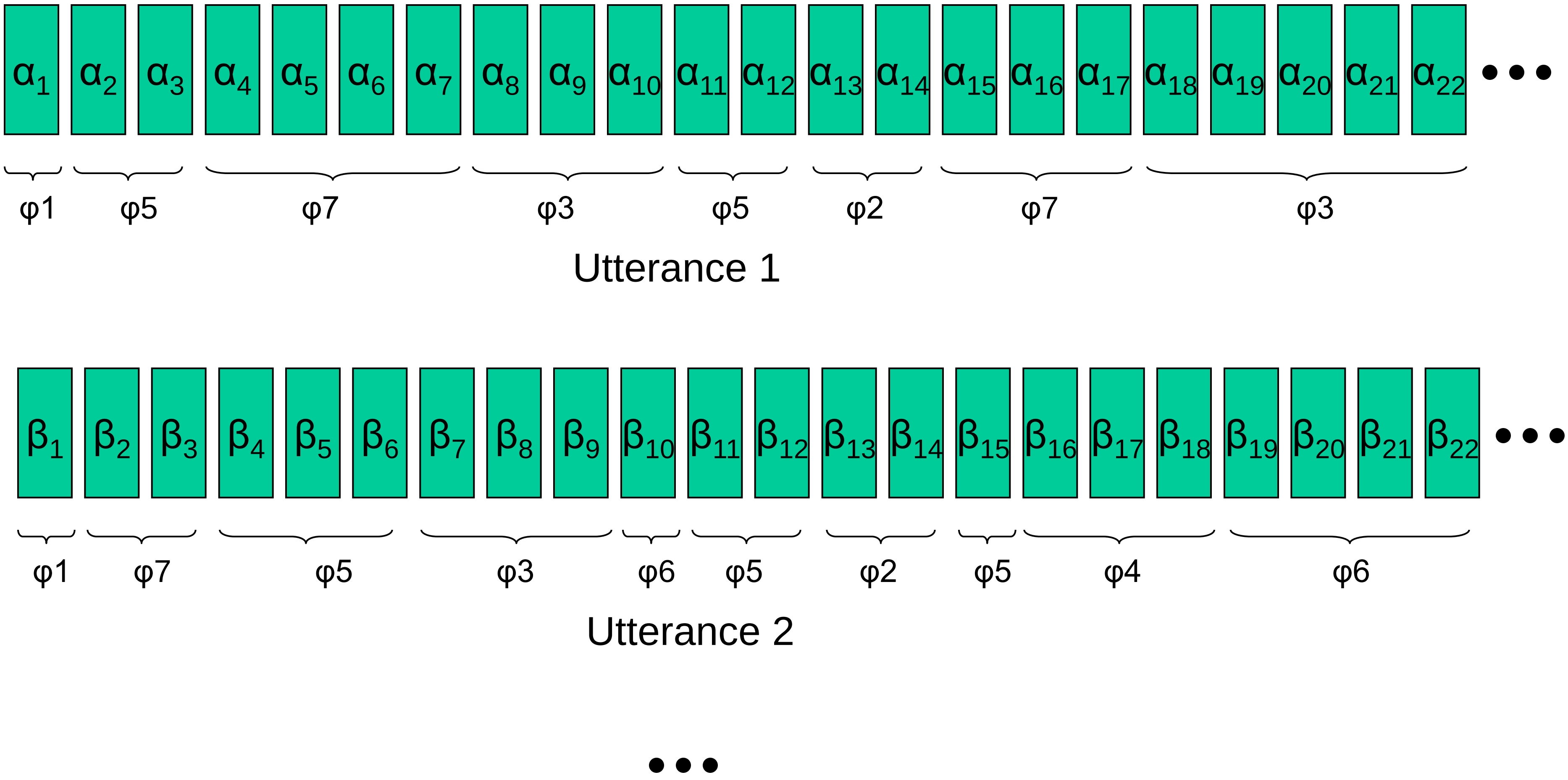
$$\Gamma_6 = \sum_{t=1}^{T} y_t y_t^T$$

LDMs - Learning

- **M-step:** Compute the parameters of the model

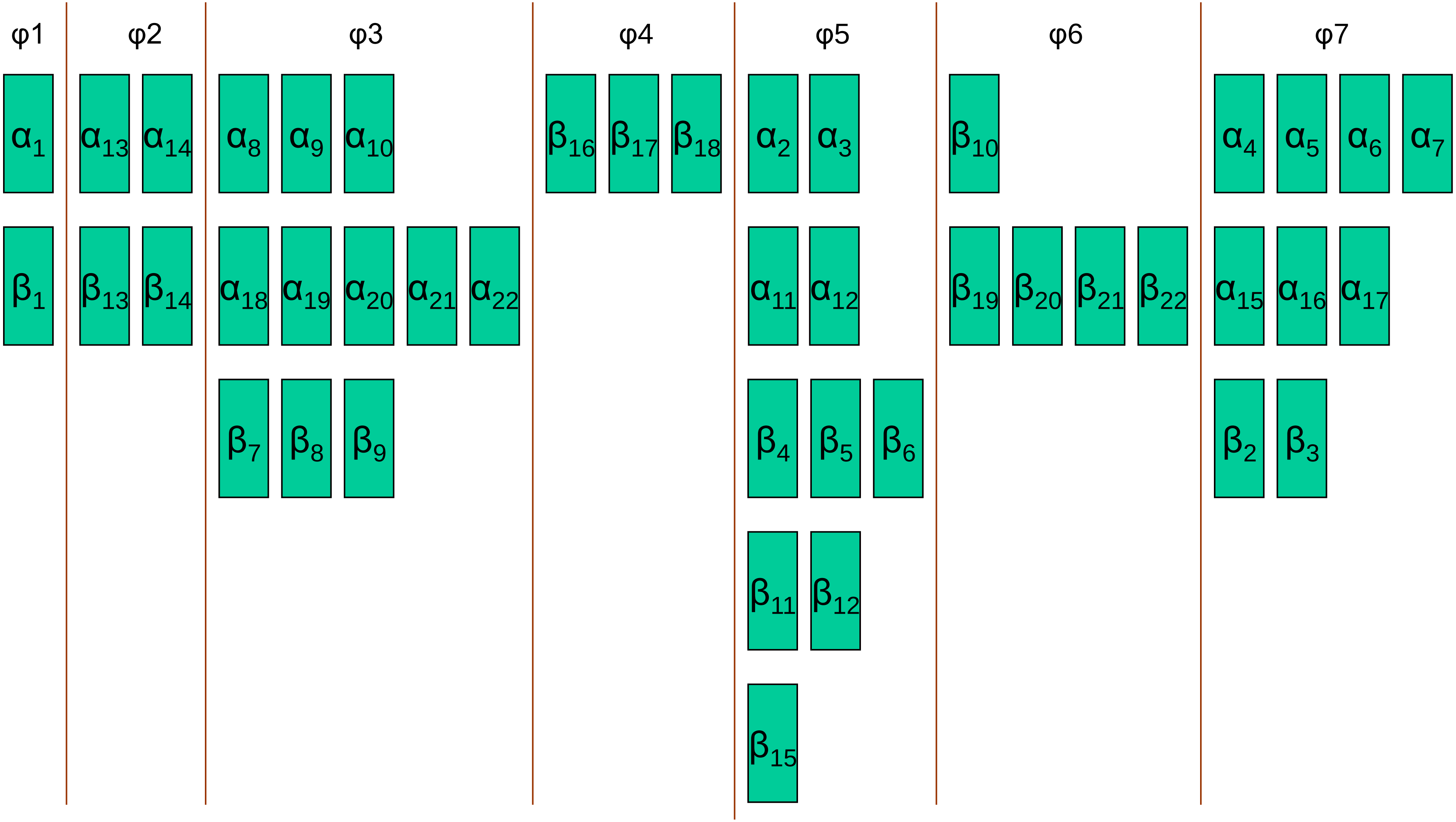
$g_1 = \hat{x}_{1\|T}$
$Q_1 = \hat{R}_{1\|T} - g_1 g_1^T$
$F = (\Gamma_4 - \frac{1}{T-1}\zeta_2\zeta_1^T)(\Gamma_1 - \frac{1}{T-1}\zeta_1\zeta_1^T)^{-1}$
$g = \frac{1}{T-1}(\zeta_2 - F\zeta_1)$
$Q = \frac{1}{T-1}\left(\Gamma_2 - F\Gamma_4^T - g\zeta_2^T\right)$
$H = \left(\Gamma_5 - \frac{1}{T}\zeta_4\zeta_3^T\right)\left(\Gamma_3 - \frac{1}{T}\zeta_3\zeta_3^T\right)^{-1}$
$\mu = \frac{1}{T}(\zeta_4 - H\zeta_3)$
$R = \frac{1}{T}\left(\Gamma_6 - H\Gamma_5^T - \mu\zeta_4^T\right)$

Training LDMs for speech synthesis



Each utterance consists of segments of phones or subphones.

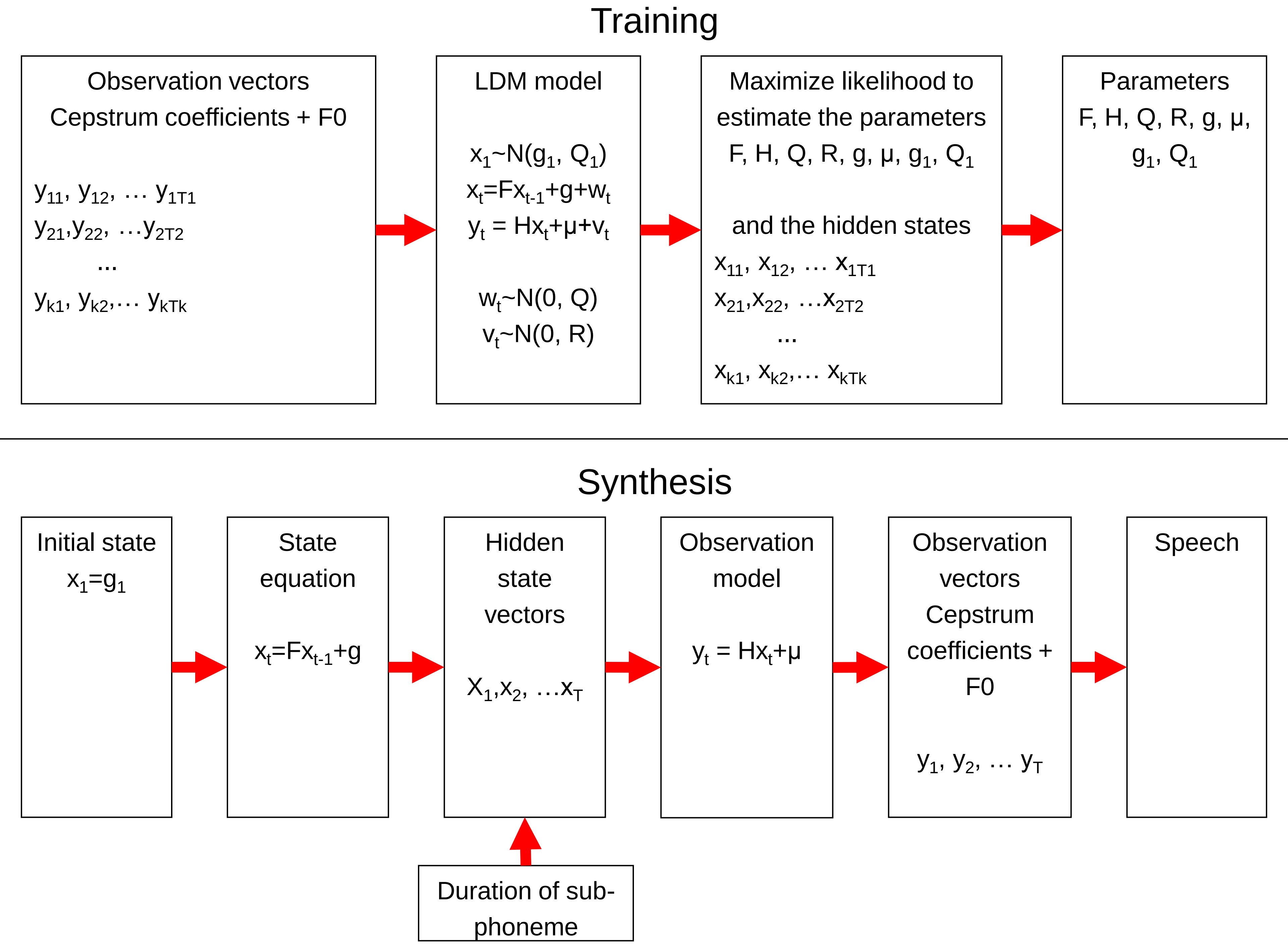
Training LDMs for speech synthesis



Train an LDM for each label φ1, φ2, φ3, φ4, φ5, φ6, φ7, ...

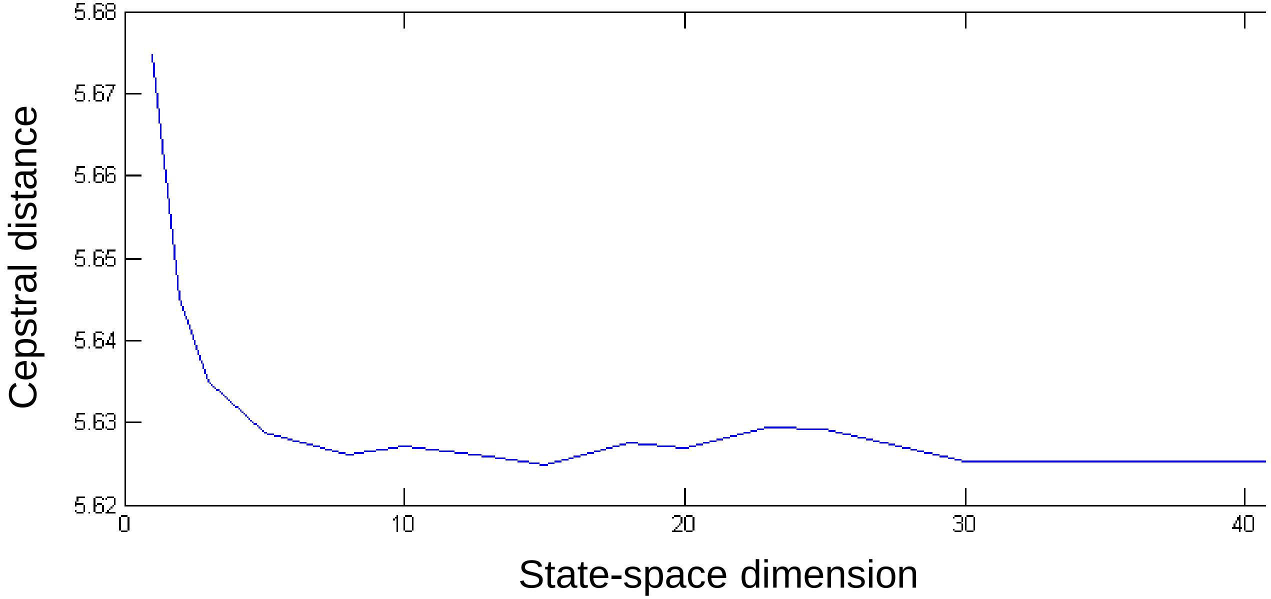
EM Algorithm

- Training an LDM for label φ_i
- Initial guesses of F, H, Q, R, g, μ, g_1, Q_1
- Kalman smoother (E-step):
  - Clear the sufficient statistics variables
  - For each example $y_{i1}, \dots y_{iT}$ in φ_i
    - Compute distributions of $X_1, \dots, X_T$ given data $y_{i1}, \dots y_{iT}$ and F, H, Q, R, g, μ, g_1, Q_1.
    - Accumulate the sufficient statistics into global variables
- Update parameters (M-step):
  - Update F, H, Q, R, g, μ, g_1, Q_1 based on sufficient statistics.
- Repeat until convergence (local optimum)


Training
Observation vectors
Cepstrum coefficients + F0
$y_{11}, y_{12}, \ldots y_{1T1}$
$y_{21}, y_{22}, \ldots y_{2T2}$
...
$y_{k1}, y_{k2}, \ldots y_{kTk}$
LDM model
$x_1 \sim N(g_1, Q_1)$
$x_t = Fx_{t-1} + g + w_t$
$y_t = Hx_t + \mu + v_t$
$w_t \sim N(0, Q)$
$v_t \sim N(0, R)$
Maximize likelihood to estimate the parameters
F, H, Q, R, g, μ, g_1, Q_1
and the hidden states
$x_{11}, x_{12}, \ldots x_{1T1}$
$x_{21}, x_{22}, \ldots x_{2T2}$
...
$x_{k1}, x_{k2}, \ldots x_{kTk}$
Parameters
F, H, Q, R, g, μ, g_1, Q_1
Synthesis
Initial state
$x_1 = g_1$
State equation
$x_t = Fx_{t-1} + g$
Hidden state vectors
$X_1, x_2, \ldots x_T$
Observation model
$y_t = Hx_t + \mu$
Observation vectors Cepstrum coefficients + F0
$y_1, y_2, \ldots y_T$
Speech
Duration of sub-phoneme

LDM configurations

- **Optimization of LDM training configurations:**
  - The ideal state-space dimension is between 6 and 9
    - Low dimensional dynamics produce high dimensional observations (e.g., 40 cepstral coefficients)



- Matrices Q and R should be diagonal
- The parameter μ is necessary
- Stability constraints should be enforced to LDMs
- All models can have the same matrix H

LDM - Maximum likelihood trajectory generation

- The likelihood of a given LDM and observation sequence Y is

$$P(Y|\theta) = \int_X P(X,Y|\theta)dX = \int_X P(Y|X,\theta)P(X|\theta)dX$$

- Sub-optimum state sequence $\hat{X}$ is determined, independently of Y

$$\hat{X} = \arg\max P(X|\theta)$$

- Since the maximum likelihood estimate of a Gaussian is its mean, the state sequence can be found by the following iteration:

$$\hat{x}_1 = g_1$$

$$\hat{x}_t = F\hat{x}_{t-1} + g, \quad t \in \{2, \cdots, T\}$$

- The maximum of

$$P\left(Y\middle|\hat{X},\theta\right) = \prod_{t=1}^{T} N(y_t; H\hat{x}_t + \mu, R)$$

is attained when:

$$y_t = H\hat{x}_t + \mu, \quad t \in \{1, 2, \cdots, T\}$$

LDM - Maximum likelihood trajectory generation

$$\hat{x}_1 = g_1$$

for t = 1:T

if (t > 1)

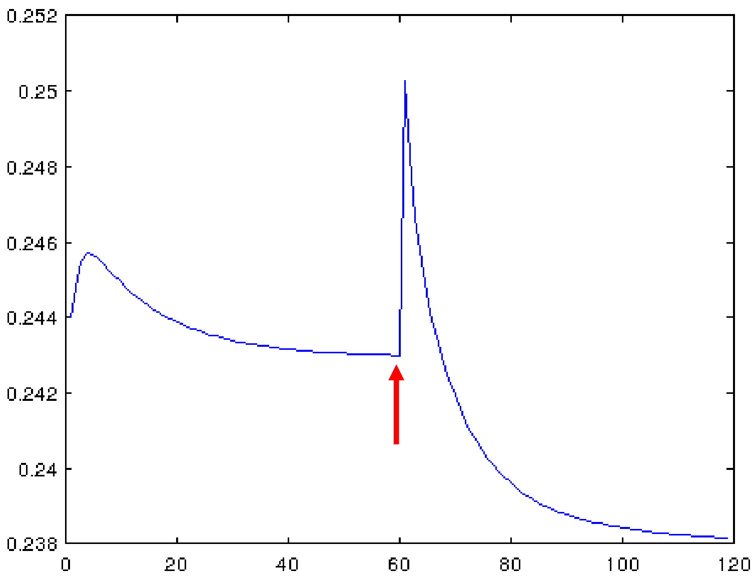
$$\hat{x}_t = F\hat{x}_{t-1} + g$$

$$y_t = H\hat{x}_t + \mu$$

- Very low computational requirements
- LDMs are suited for real time speech production

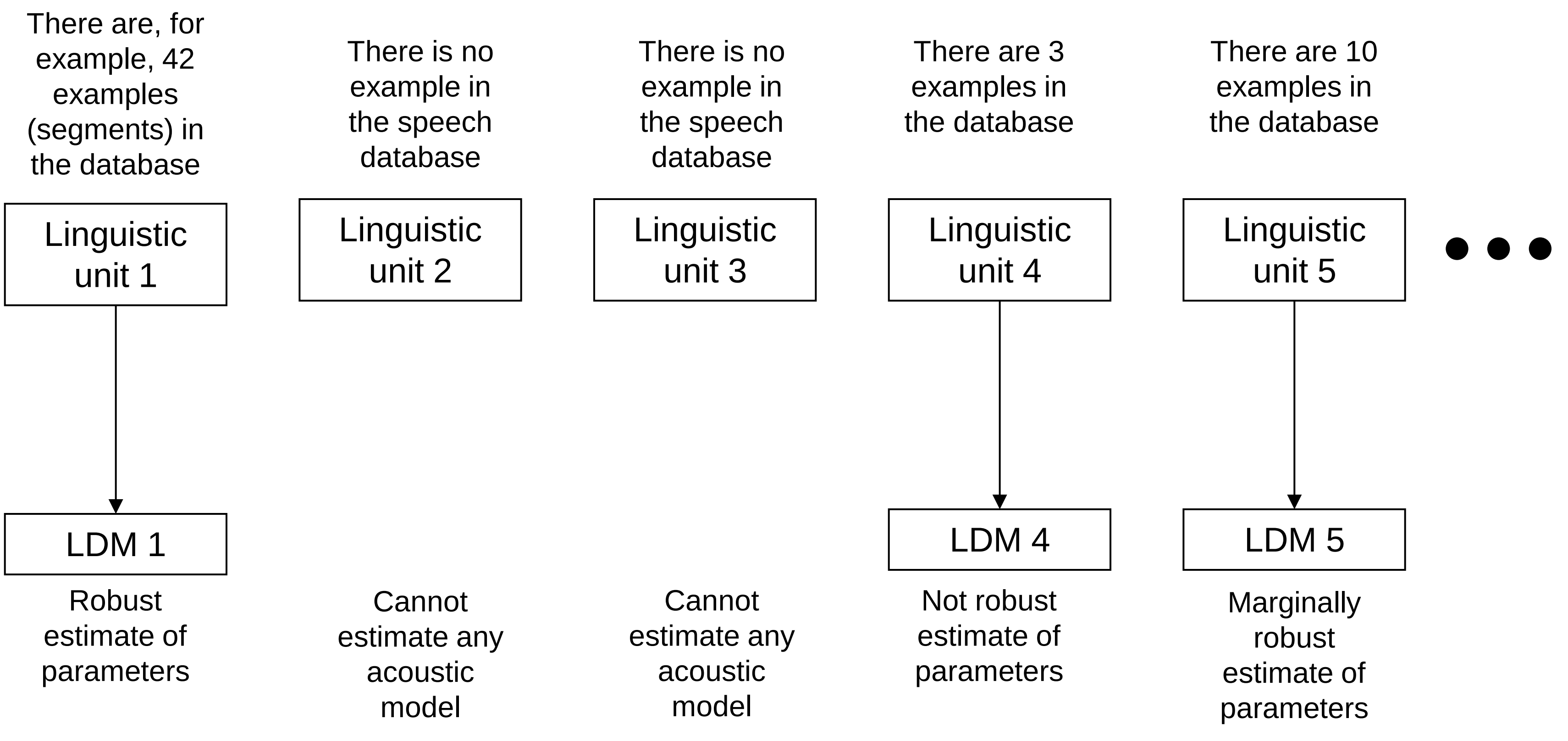
LDMs - Experiments

- Second-order LDMs fit better the ceptrum than first-order LDMs.
  - Mean cepstral distance
  - Informal listening tests
- First-order LDMs fit better the continuous F0 than second-order LDMs.
  - Informal listening tests
- Discontinuities between neighbouring segments in synthesized speech
  - A common parameter H alleviates the problem



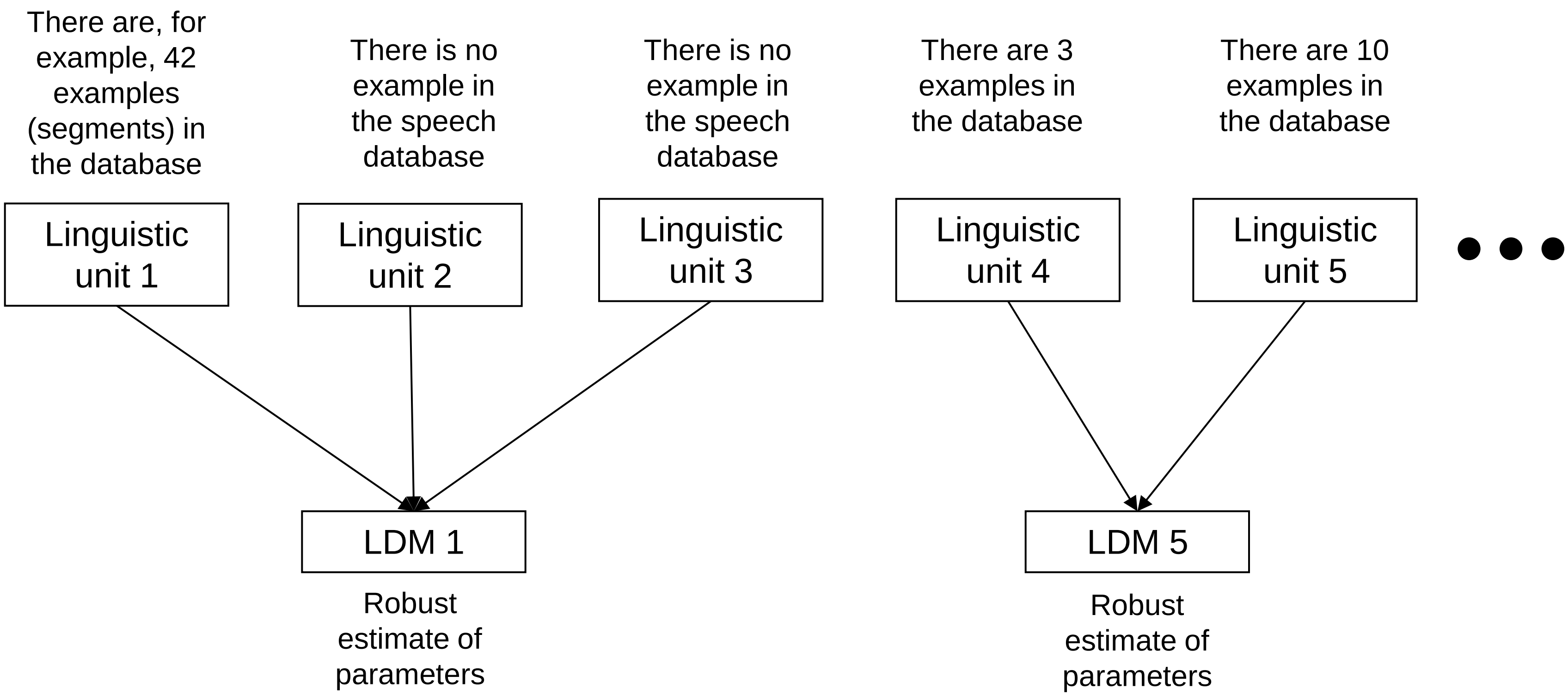
Linguistic-to-Acoustic Mappings

- The simplest map is for each linguistic (phonetic and prosodic contextual unit) unit to assign an acoustic model (an LDM).
- Not enough training samples to robustly train all models
- Example:



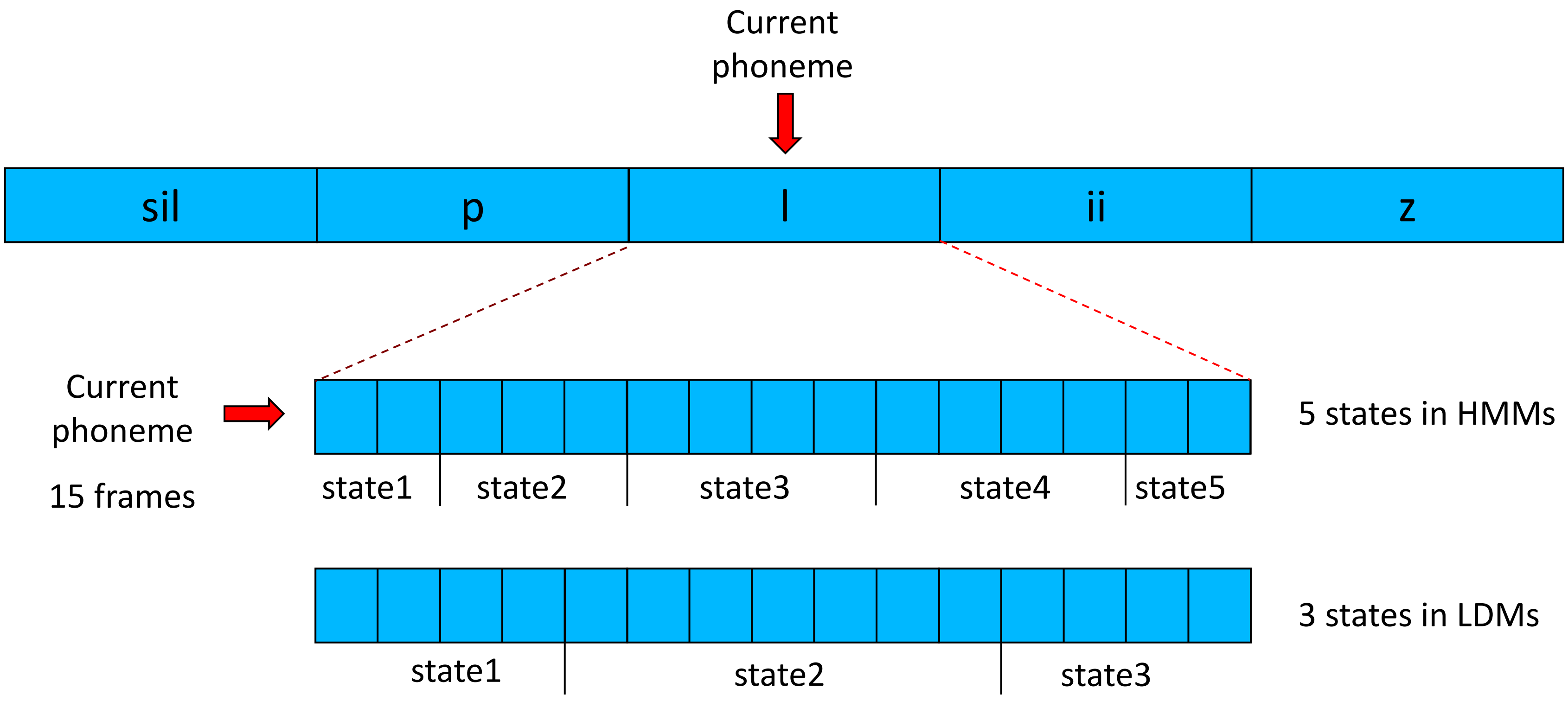
Linguistic-to-Acoustic Mappings

- A solution: Use the same LDM for more than one linguistic units.
  - Cluster linguistic units in an way that is close to optimal, using binary decision trees.



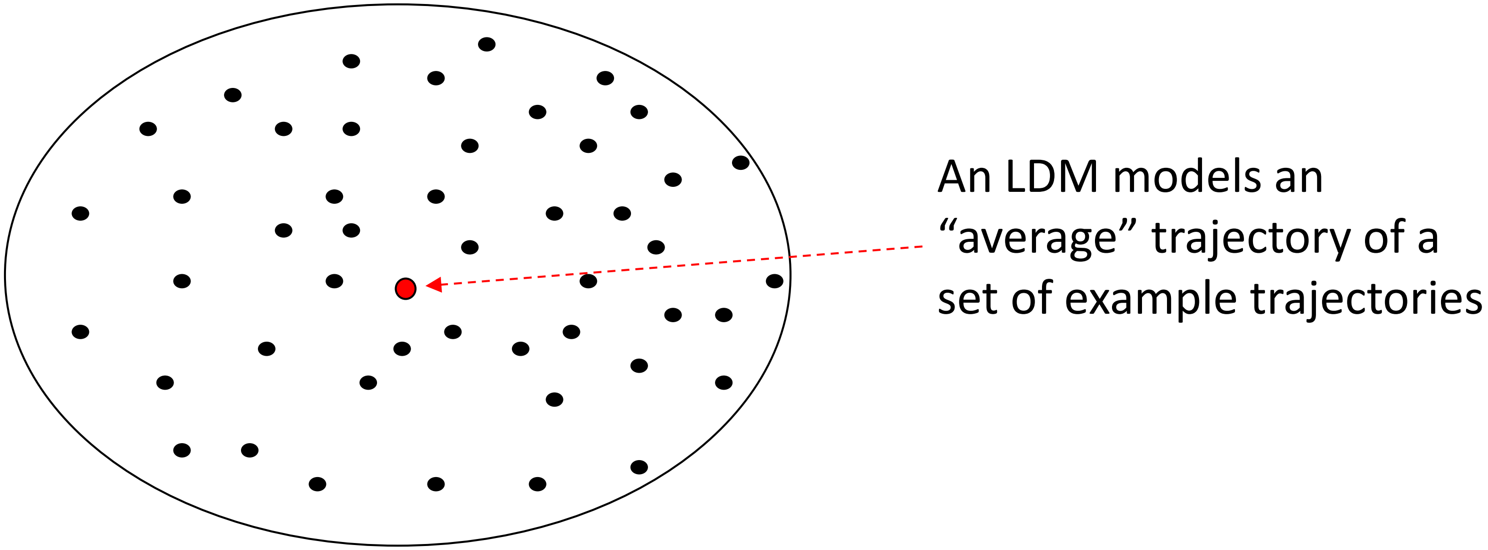
LDM: Decision Tree Clustering

- The LDM models are trained using full context labelling
- The context is independent of the number of states



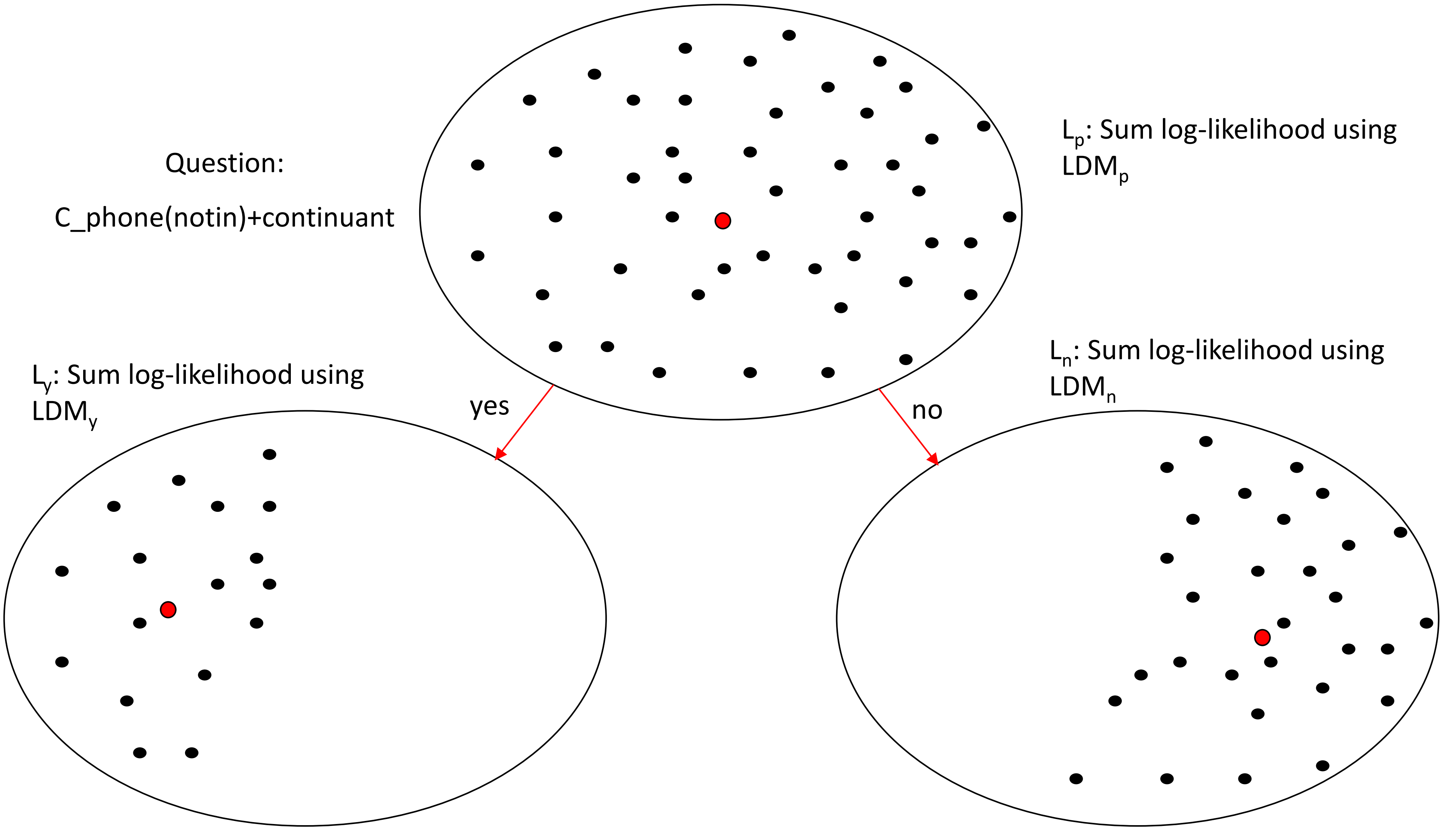
LDM: Decision Tree Clustering

- The LDM models are trained using full context labelling
  - The number of possible pentaphons far exceeds the number of training examples
  - Solution: One LDM models many pentaphons that have similar speech parameters
  - The training examples are clustered according to linguistic questions and how well they fit to LDM that models the examples of a cluster.
  - Initially, all training examples are modelled with one LDM.



LDM: Decision Tree Clustering

- Hierarchical top-down clustering. Split if $L_y + L_n > L_p$ + MDL_threshold

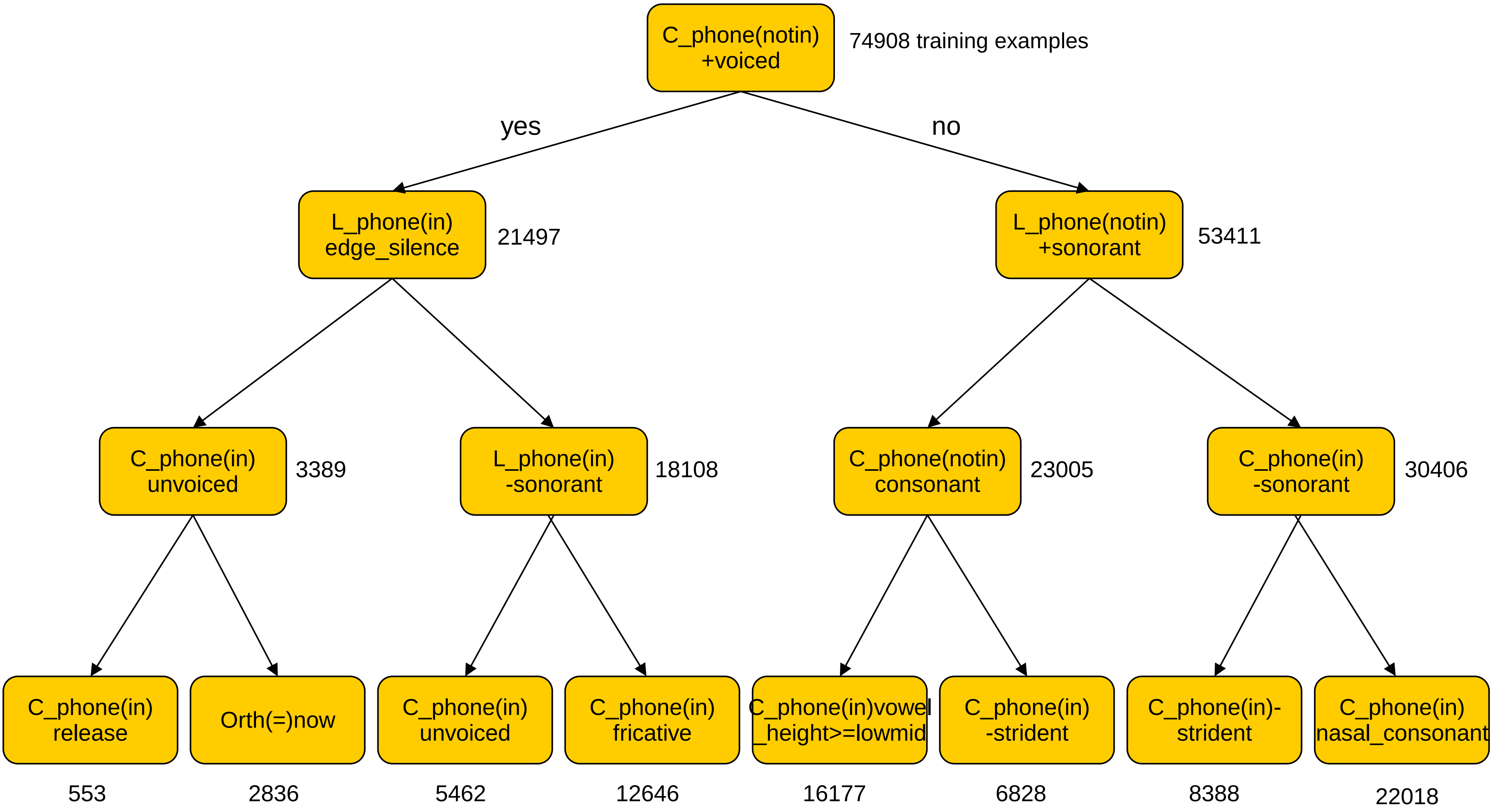


LDM: Decision Tree Clustering Algorithm

- Create the root node of the decision tree, which contains all examples
- queue.put(rootNode)
- While(is_not_empty(queue))
  - node = queue.pop()
  - Find the question that has the largest $L_y + L_n$
    - For each question *//Do this using Parallel Processing*
      - Split the examples associated with the current node
      - Fit an LDM to "yes" examples and calculate L_y
      - Fit an LDM to "no" examples and calculate L_n
      - Check if $L_y + L_n > L_p$ + MDL_threshold and store $L_y + L_n$
  - If a (best) question is found
    - Create tree node yesNode that contains the "yes" examples
    - Create tree node noNode that contains the "no" examples
    - queue.put(yesNode)
    - queue.put(noNode)

Application of LDMs to TTS – Clustering

- Part of the Decision Tree of mceps



Application of LDMs to TTS – Global Variance

- Global Variance (GV) is defined as an intra-utterance variance of a speech parameter trajectory and is modelled by a Gaussian distribution.
- The GV algorithm constrain the synthesized trajectories to have the same GV as the GV of the corresponding training samples.
- In speech parameter generation, the optimum parameter sequence is determined so as to maximize an objective function consisting of the LDM and GV log pdfs

$$L = \frac{1}{T} \log P(Y \mid \bar{X}, \theta_{LDM}) + \log P(v \mid \theta_{GV})$$

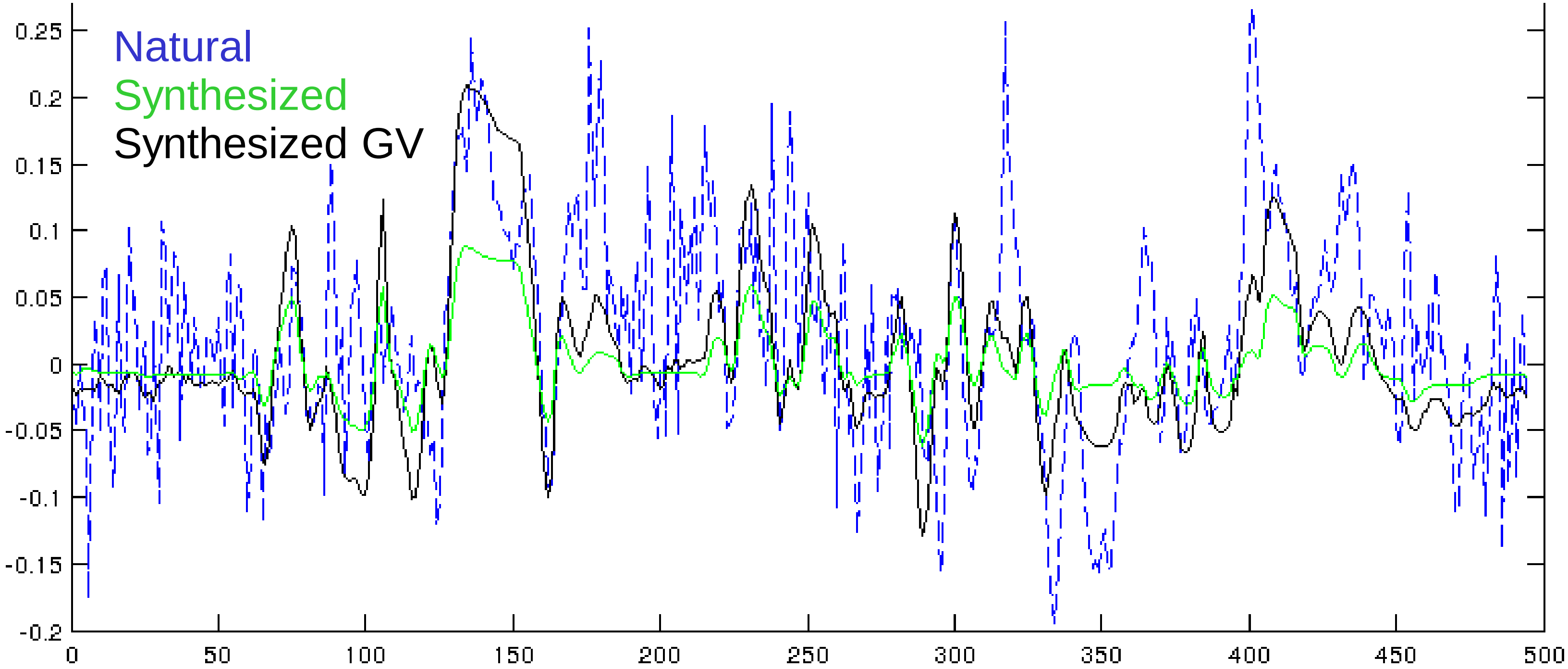
where θ_{LDM} and θ_{GV} are the parameters of the distributions of LDM and GV, Y are the trajectories of speech parameters (e.g., Cepstrum), vector v has the variances of Y trajectories, T is the duration of trajectories, and hidden state $\bar{X}$ is

$$\bar{X} = \arg\max P(X \mid \theta_{LDM})$$

- The objective function L is maximized by a steepest decent algorithm

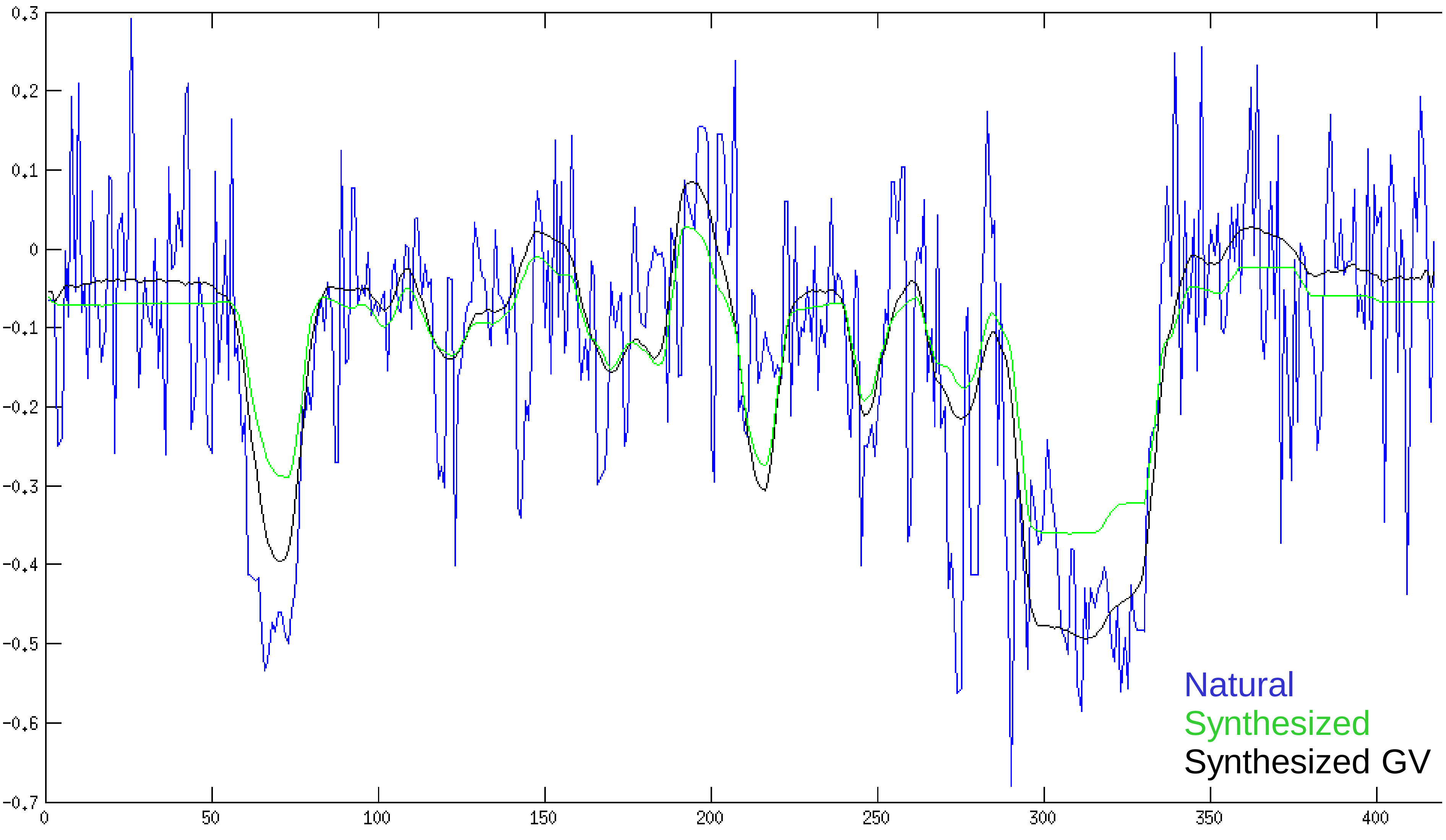
Application of LDMs to TTS – Global Variance

- GV has been applied both to traditional LDMs and to LDMs with critically dumped target-dynamics.
- In informal subjective listening tests the volunteers preferred the GV LDM synthesized speech from the LDM synthesized speech



Trajectories of c(32)

Application of LDMs to TTS – Global Variance



Trajectories of c(16)

LDMs – Footprint

- LDM footprint
- Matrices *H,* and *R* are globally tied
  - Their contribution to the total number of parameters is minimal
  - Matrix *Q* is constant ($Q = I$).
  - Matrix *F* and vector *g* are different for every model (leaf of the clustering tree)
  - n^2 parameters for *F* and *n* parameters for *q*, where $n < m$ (*m* is the number of static features).
  - Total number of parameters ≈ $(n^2 + 3n + m) \times$ number of leafs in clustering trees
- HSMM footprint
  - Total number of parameters ≈ $6m \times$ number of leafs in clustering trees
  - + elements of transition matrix x number of leafs in cluster trees
- If the number of clustering leafs are equal, then LDM uses 1/3 of the parameters of HSMM
- Alternatively LDM can use finer clustering, improving the quality of synthesized speech

LDMs – Implementation issues

- The software was implemented in Matlab.
  - It has been written from scratch and does not depend on HTS
- Those parts of the software that are computationally demanding have been implemented in C
  - The BLAS and LAPACK numerical libraries were used for the matrix operations
- The software uses the conventional Kalman filter, but there is the option to switch to the square root Kalman filter in ill conditioned models (relatively few samples).

Samples: March 2015

Samples from the training set

HSMM duration. Synthesized Cepstrum, Band aperiodicity and F0

LDM	LDM GV	HSMM GV
herald_264	herald_264	herald_264
herald_439	herald_439	herald_439

Samples from the test set

HSMM duration. Synthesized Cepstrum, Band aperiodicity and F0

LDM	LDM GV	HSMM GV
herald_413	herald_413	herald_413
herald_752	herald_752	herald_752
hvd_720	hvd_720	hvd_720
mrt_150	mrt_150	mrt_150

Samples: July 2015

Samples from the training set

Natural duration.

Second-order LDMs: Cepstrum, Band aperiodicity and phase

First-order LDMs: F0

LDM	LDM GV
herald_24	herald_24
herald_26	herald_26
herald_142	herald_142
herald_198	herald_198
herald_264	herald_264
herald_439	herald_439