Alex Sorin, IBM Watson Speech Technologies, Haifa, Israel.

Mixed
Unit Selection – HMM
Speech Synthesis

Speech Synthesis Summer School. Heraklion, Crete, Greece. July - Aug 2015

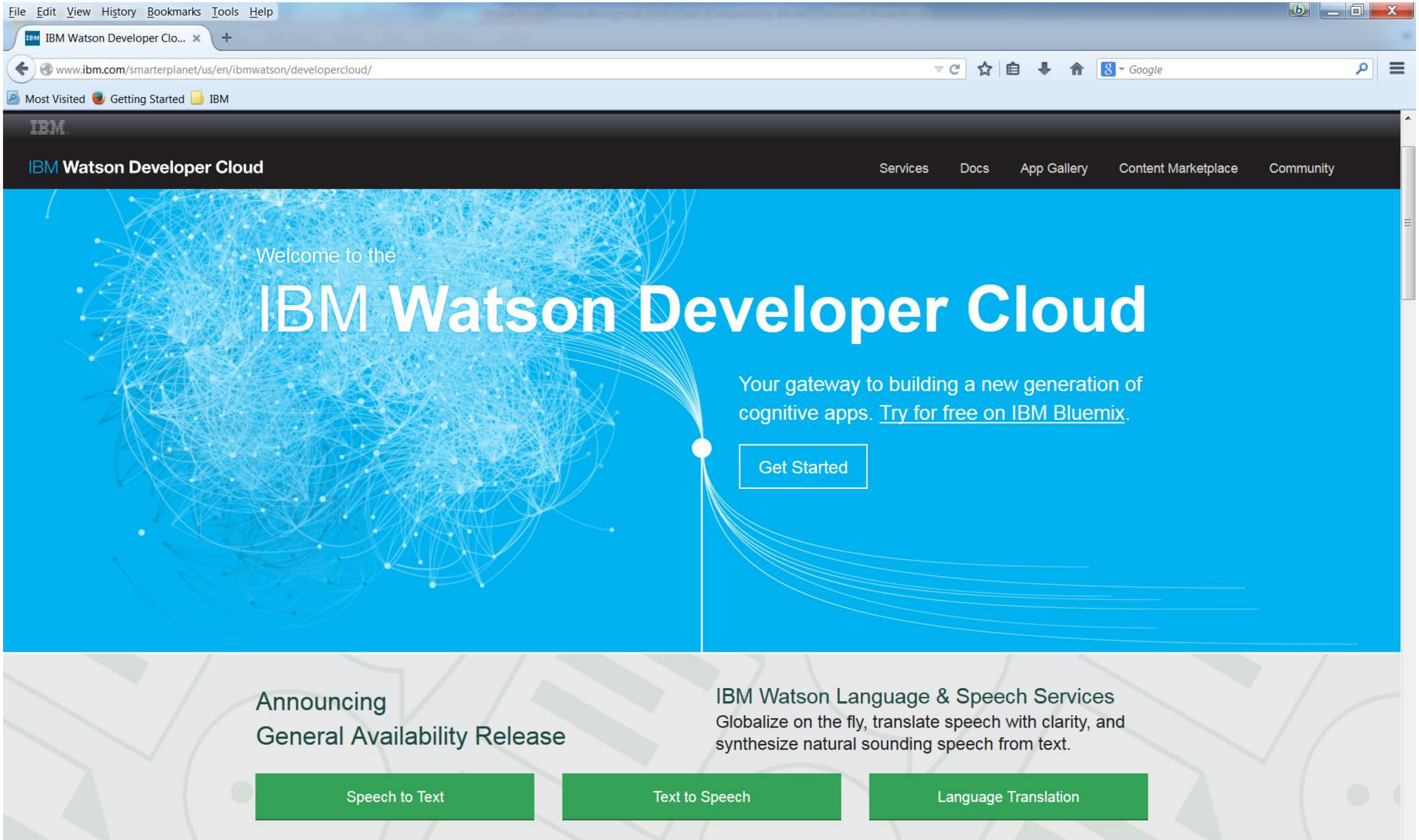
The outline

- Motivation, promises and challenges

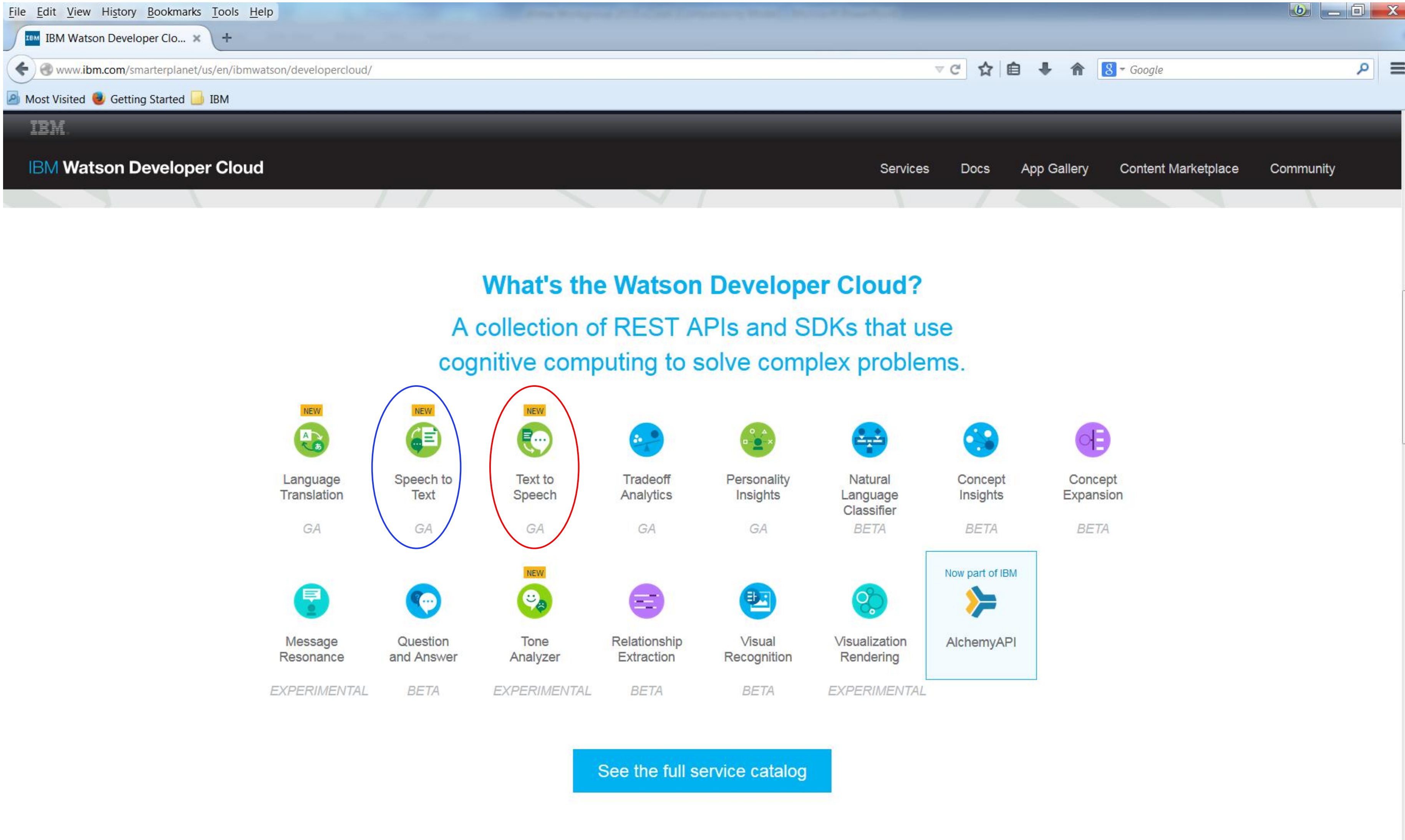
- Challenges one-by-one
  - Better understanding and formulation
  - Proposed solutions
  - Evaluations

TTS at IBM – the modern history

- Past products – Server based and Embedded *trainable* unit selection TTS systems
  - Robert Donovan, 1990s
  - Sub-phone level units
  - The embedded system: parameterized segments, 10 – 20 MB voices, deployed in Honda cars as a part of the embedded ViaVoice driver interface
- 2010 – 2014: Joint Development Agreement (JDA) with Nuance Communications
  - Tens of IBM researchers conducted exploratory work on ASR and TTS aiming at advancing Nuance products
- Since 2013 – IBM cognitive computing products
  - 2014 a new unit – IBM Watson Group to meet demand for cognitive innovations
  - Open cognitive platform – developer cloud including ASR/TTS as a service



http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/



http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/

- The vision and approaches presented here were developed in collaboration with Nuance under the Joint Development Agreement

- The evaluation results were obtained using the data and experimental TTS voices provided by Nuance

Concatenative TTS vs. HMM TTS

Concatenative TTS

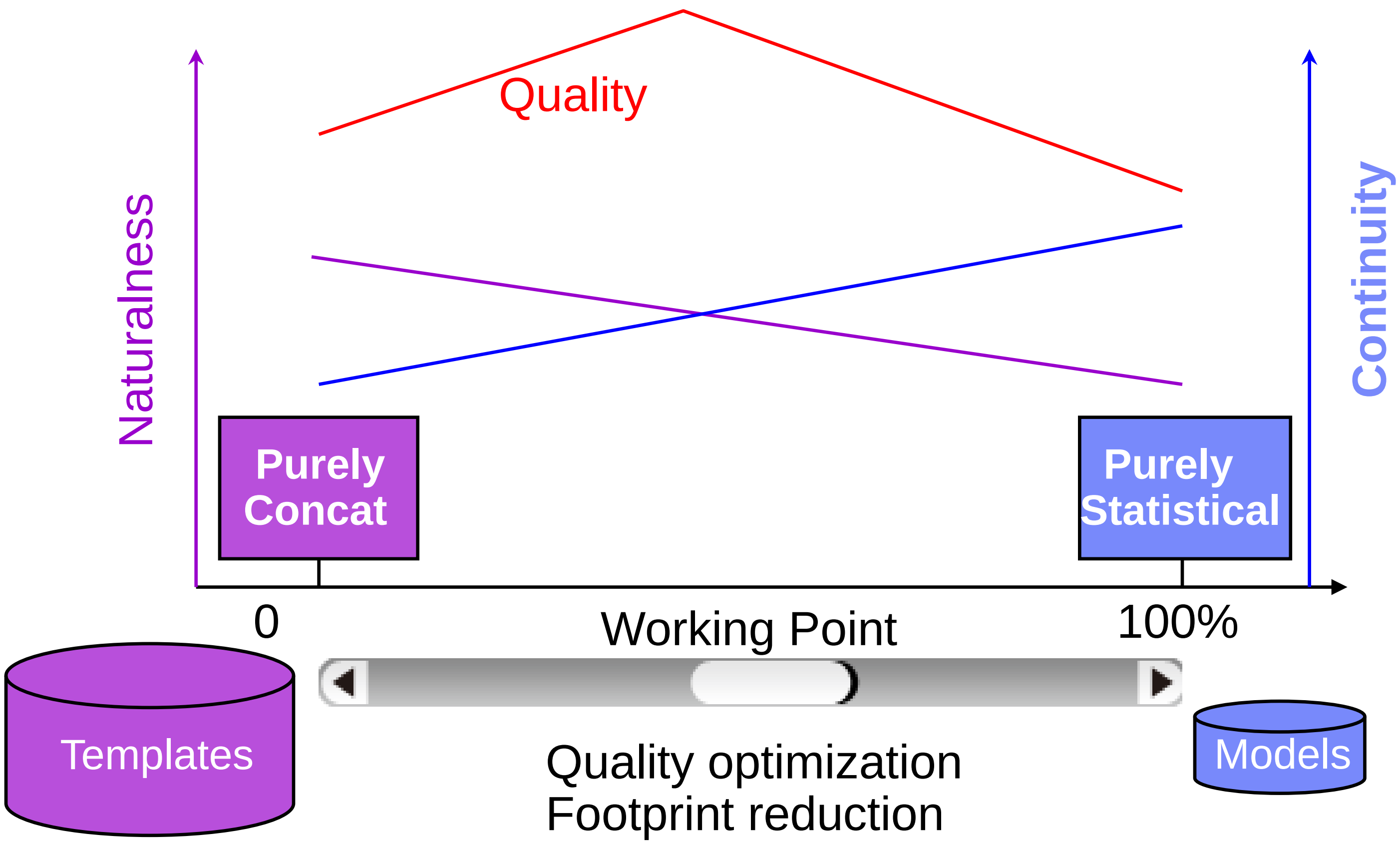
+ Crisp, natural sound
+ Natural prosody
- High sensitivity to the ‘training data’ - domain, sparsity, alignment accuracy
- Glitches – discontinuities at joints, bad occurrences
- Speech manipulation is limited
- Big footprint – memory & CPU

HMM TTS

- Muffled, artificial sound
- Averaged, flat prosody
+ Robustness - generalization capabilities, tolerance to the dataset size and alignment accuracy
+ Continuity, stable quality
+ Ease of speech manipulation
+ Small footprint

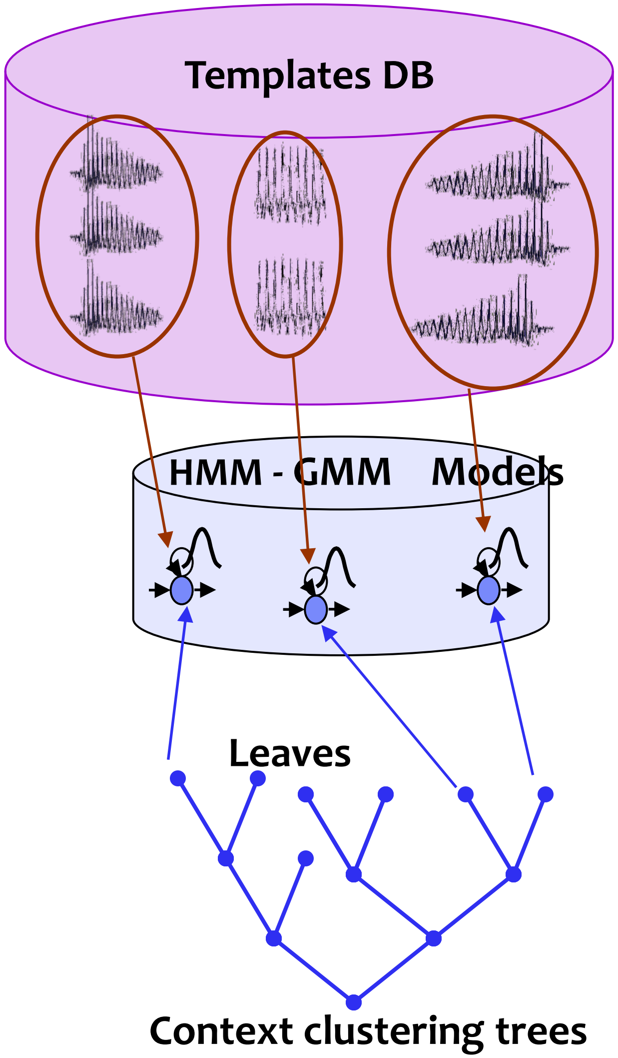
Why mixed speech synthesis

- To benefit from the respective advantages of the two paradigms
  - Natural and crisp sound of the unit selection TTS
  - Continuity, generalization, ease of manipulation, low footprint of the statistical parametric TTS

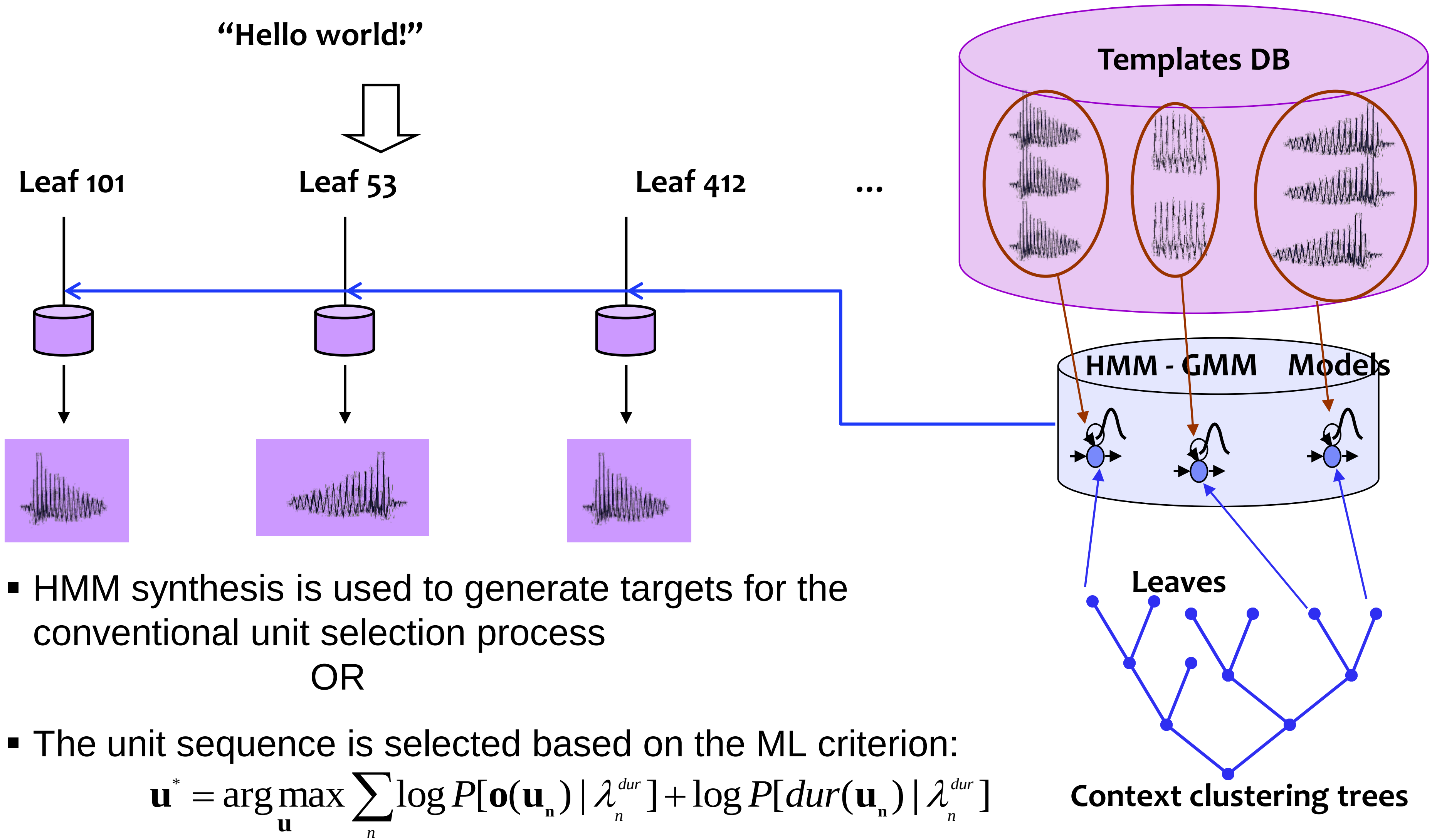


Hybrid TTS voice

- Take a data corpus that you would use for the Unit Selection voice building
- Train an HMM (or HsMM) voice on this corpus
  - E.g. 3 or 5 states per phone
- As a byproduct you get a mapping of the speech segments to leaf nodes (and their respective CD HMMs)
- Retain all the natural segments (*templates*) and their associated leaf labels
- Manual inspection/correction of pronunciation and phonetic alignment is important
- Dual nature
  - HMM system
  - Unit selection system
  - And each template is mapped to a leaf node (HMM model) - **important**



HMM based Unit Selection Synthesis



- HMM synthesis is used to generate targets for the conventional unit selection process

OR

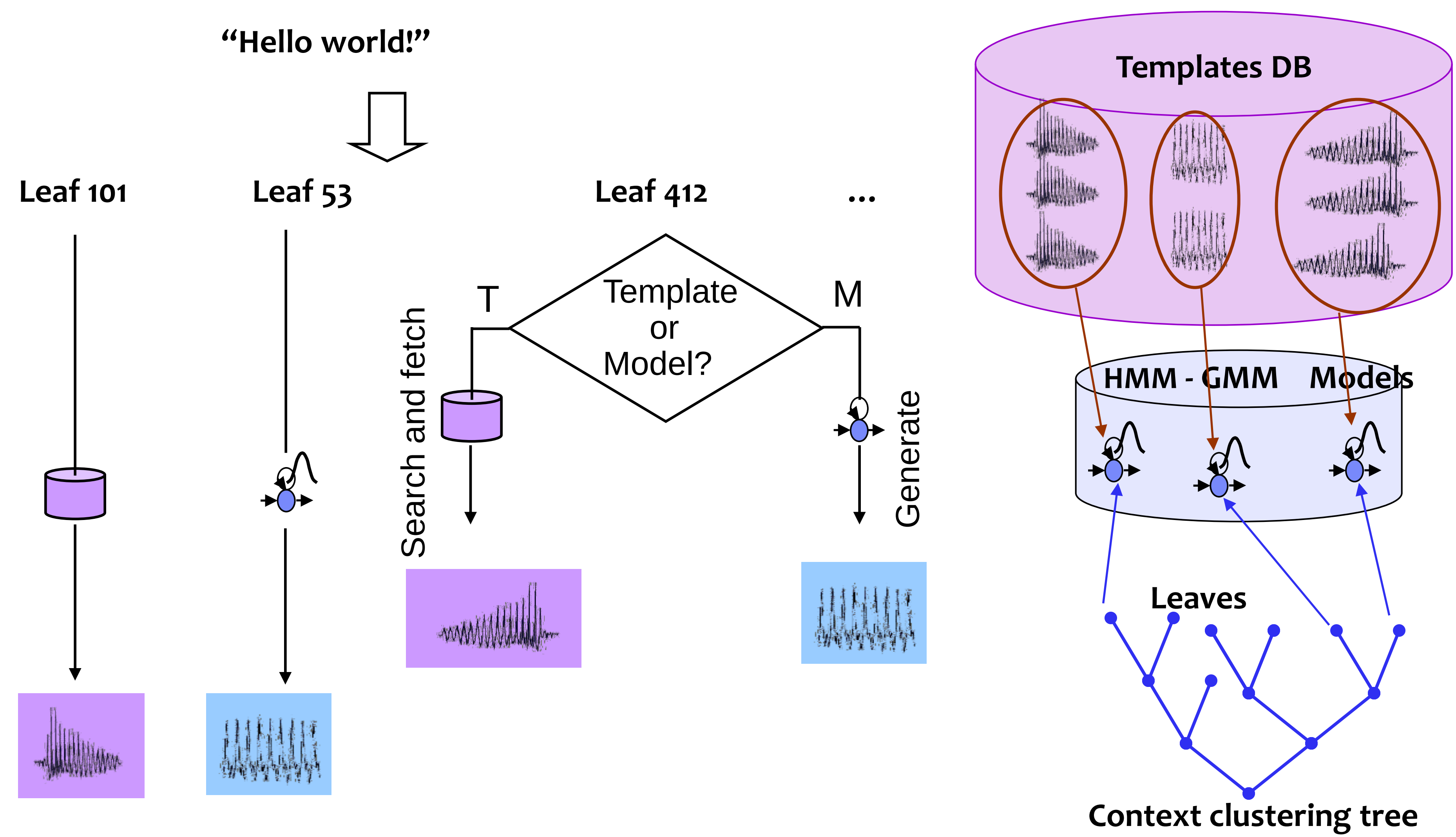
- The unit sequence is selected based on the ML criterion:

$$\mathbf{u}^* = \arg\max_{\mathbf{u}} \sum_n \log P[\mathbf{o}(\mathbf{u}_n) \mid \lambda_n^{dur}] + \log P[dur(\mathbf{u}_n) \mid \lambda_n^{dur}]$$

HMM based Unit Selection Synthesis

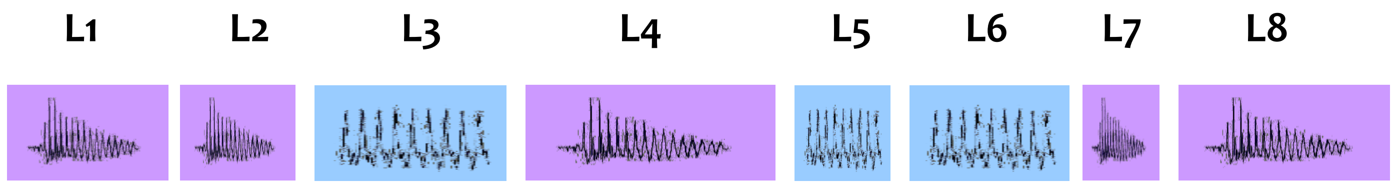
- A lot of publications since 2004
  - HMM-based target prediction: Kawai et al, 2004, ISCA SSW5
  - Ling and Wang, ICASSP 2007: ML based unit selection
- Recent publications
  - Yansuo Yu et al, SHRC Peking University – a winning submission to Blizzard 2013
- *This approach itself does not fully realizes the idea of a hybrid system*
  - *The output signal is a concatenation of natural segments. In particular, the sparsity issue remains unsolved.*

Mixing natural and generated segments – essentially hybrid synthesis



Mixed speech synthesis and related challenges

- Statistical parametric models based unit selection plus…
- *Splicing natural segments (templates) and model-based segments in the output speech signal*



CHALLENGES

1. Voice quality mismatch between the model and template segments – heterogeneous quality
2. When to use templates and when to use models?
   - How to define and control the working point at the HMM -- UnitSelection axis?
3. How to assure smoothness across model – template joints?
4. How to reduce discontinuities at template - template joints?

Mixed speech synthesis - publications

- Okubo et al, IEICE - Transactions on Information and Systems, 2006
  - The first proposed system with diphone level segments/models for voice mimicking app.
  - Ad hoc on-line template/model decision – in response to the local sparsity observed
- Aylett and Yamagishi, LangTech 2008
  - Diphone hybrid system for voice mimicking app
  - Ad hoc on-line template/model decision – in response to the local sparsity observed
- Pollet and Breen, Interspeech 2008
  - Subphone level segments/models. Statistical framework for template/model decision.
- Tiomkin et al, IEEE Transactions on Audio, Speech & Language Processing, 2011
  - Subphone level.
  - Ad hoc on-line template/model decision – in response to the local sparsity observed
- Sorin, Shechtman and Pollet. Interspeech 2011, 2012, 2014.
  - Subphone and frame level.
  - Offline template/model decision based on a statistical psychoacoustic measure.

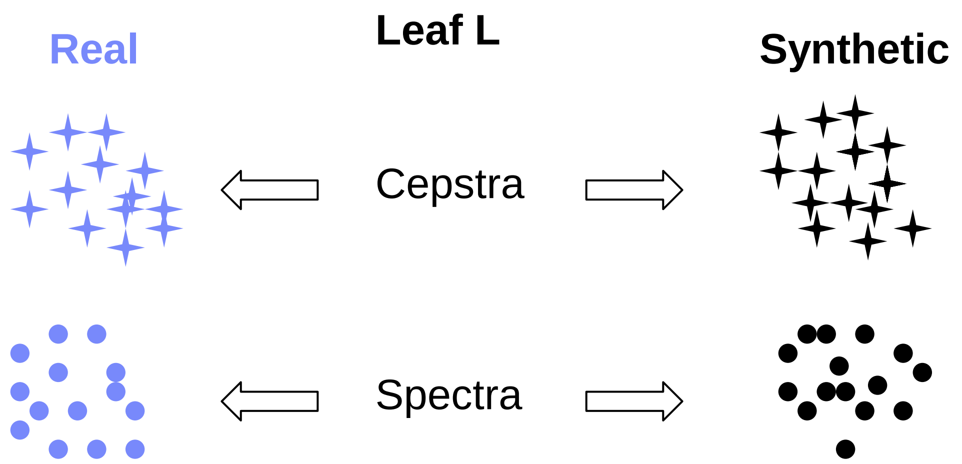
Challenge 1. Voice quality mismatch between natural and generated segments

- Switching between muffled generated segments and crisp natural segments would lead to patch-like heterogeneous speech quality
- Enhancement of statistically generated speech is a long-standing and still relevant issue tackled by numerous research works
  – Global Variance optimization (Toda and Tokuda, 2007) is the most popular approach
- This issue is especially relevant to the mixed synthesis

- The approach presented below yields tractable and simple method for effective enhancement of statistically generated speech
- Like in the GV approach we will observe differences between statistically generated and natural cepstral coefficients
- Unlike the GV approach we will
  – Observe the cepsrum vector componets structure rather than their dynamic range
  – Explain and parameterize the differences using cepstrum mathematical properties

Development setup

- Re-synthesize statistically all the sentences used for the voice training
- Collect all the *synthetic* cepstrum vectors associated with a selected leaf *L*
  - All these vectors were emitted from the leaf Gaussian $N(\mu_L, \Sigma_L)$
- Collect all the *real* cepstrum vectors associated with the leaf *L*
- Transform all the cepstrum vectors to respective spectrum envelops
- Thus for each leaf L we have two clusters – *real* and *synthetic*
  - For each cluster we have a collection of spectra and a collection of cepstrum vectors

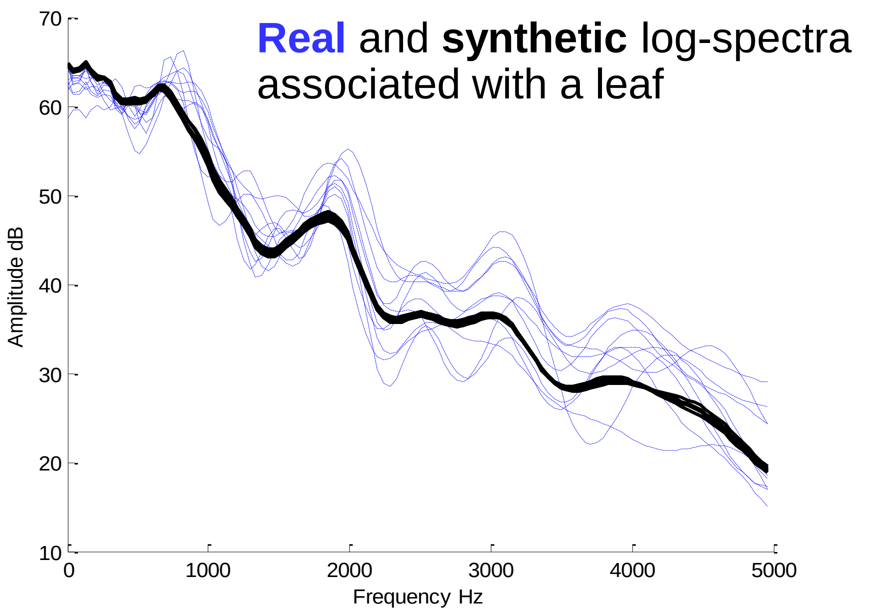


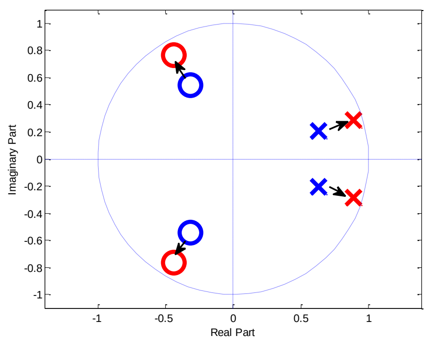
Spectrum over-smoothing effect and its interpretation

- Real spectra exhibit much higher peaks and deeper valleys than the synthetic spectra
- Averaging flattens the spectrum structure
  - Cepstra averaging is equivalent to log-spectra averaging
  - ML trajectory passes close to the Gaussian means
  - In some sense the average is not representative
- Zero-Pole representation is useful for analysis and parameterization of the spectrum flattening

$$S(z) = \prod_{m=1}^{M}(1 - z^{-1}z_m) \Big/ \prod_{k}^{K}(1 - z^{-1}p_k),\ |p_k| < 1, |z_m| < 1$$

- Flattening – moving poles and zeros away from the unit circle towards the origin of Z-plane





Spectrum flattening – cepstrum attenuation

- Let's express cepstral coefficients c_n in terms of poles and zeros

$$\log S(z) = \sum_{m=1}^{M} \log(1 - z^{-1} z_m) - \sum_{k=1}^{K} \log(1 - z^{-1} p_k)$$

$$\log S(z) = \sum_{n=1}^{\infty} c_n z^{-n}$$

$$\boxed{c_n = \sum_{m=1}^{M} \frac{z_m^n}{n} - \sum_{k=1}^{K} \frac{p_k^n}{n}}$$

- When $|z_m|$ and $|p_k|$ become smaller (moving away from the unit circle) the cepstral coefficients c_n decay faster with n
- It means that on the average the synthetic cepstra should exhibit faster attenuation than the real cepstra for the same leaf cluster
- Let's see if we observe this phenomenon

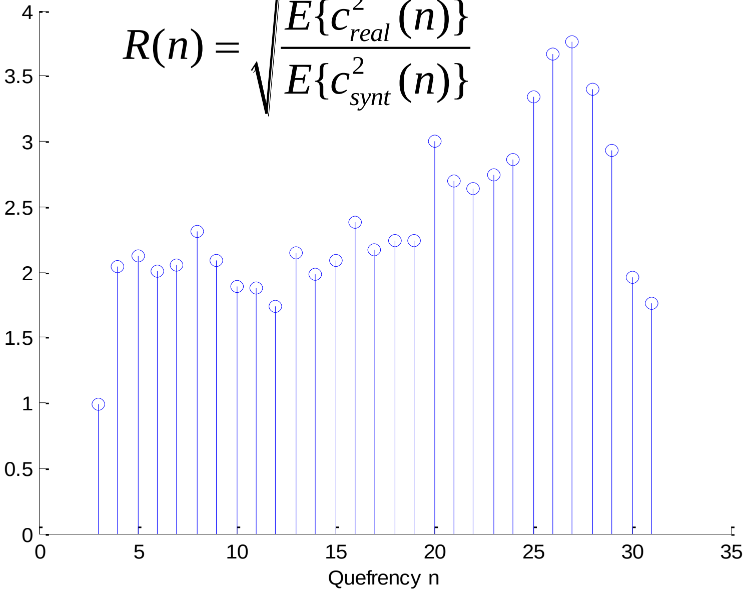
Observing and parameterization of the cepstrum attenuation

- We observe the extra-attenuation in the synthetic cepstra dividing the averaged squared real vectors by the averaged squared synthetic vectors
- To reduce the over-smoothing effect we would like to push the poles and zeros back towards the unit circle
- The simplest way is to push them uniformly and without changing their radial locations

$$\tilde{z}_m = \rho \cdot z_m \quad \tilde{p}_k = \rho \cdot p_k \quad 1 < |\rho| < \frac{1}{\max(|z_m|,|p_k|)}$$

2nd Moment Ratio vector

$$R(n) = \sqrt{\frac{E\{c_{real}^2(n)\}}{E\{c_{synt}^2(n)\}}}$$



- This lead to the exponential *liftering* of the synthetic cepstrum vectors

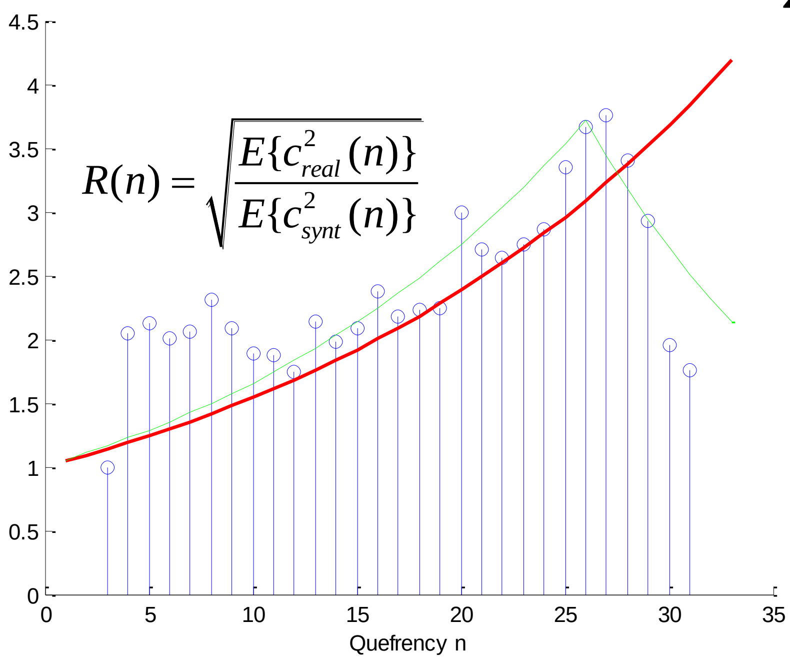
$$\tilde{c}_n = \rho^n \cdot c_n = \sum_{m=1}^{M} \frac{(\rho \cdot z_m)^n}{n} - \sum_{k=1}^{K} \frac{(\rho \cdot p_k)^n}{n}$$

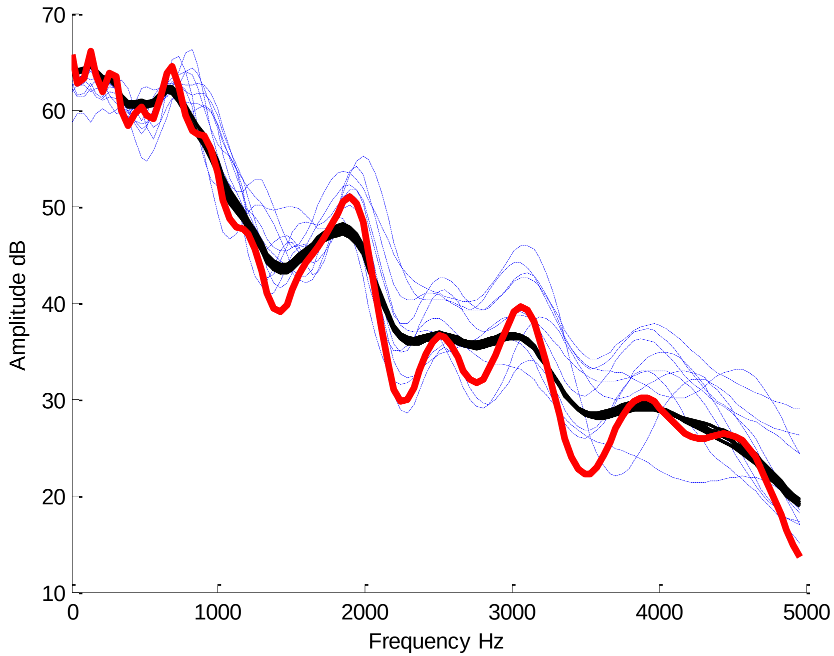
Enhancement parameter estimation

- Let's estimate the exponent base ρ using the LMS exponential approximation of the 2nd moment ratio vector R

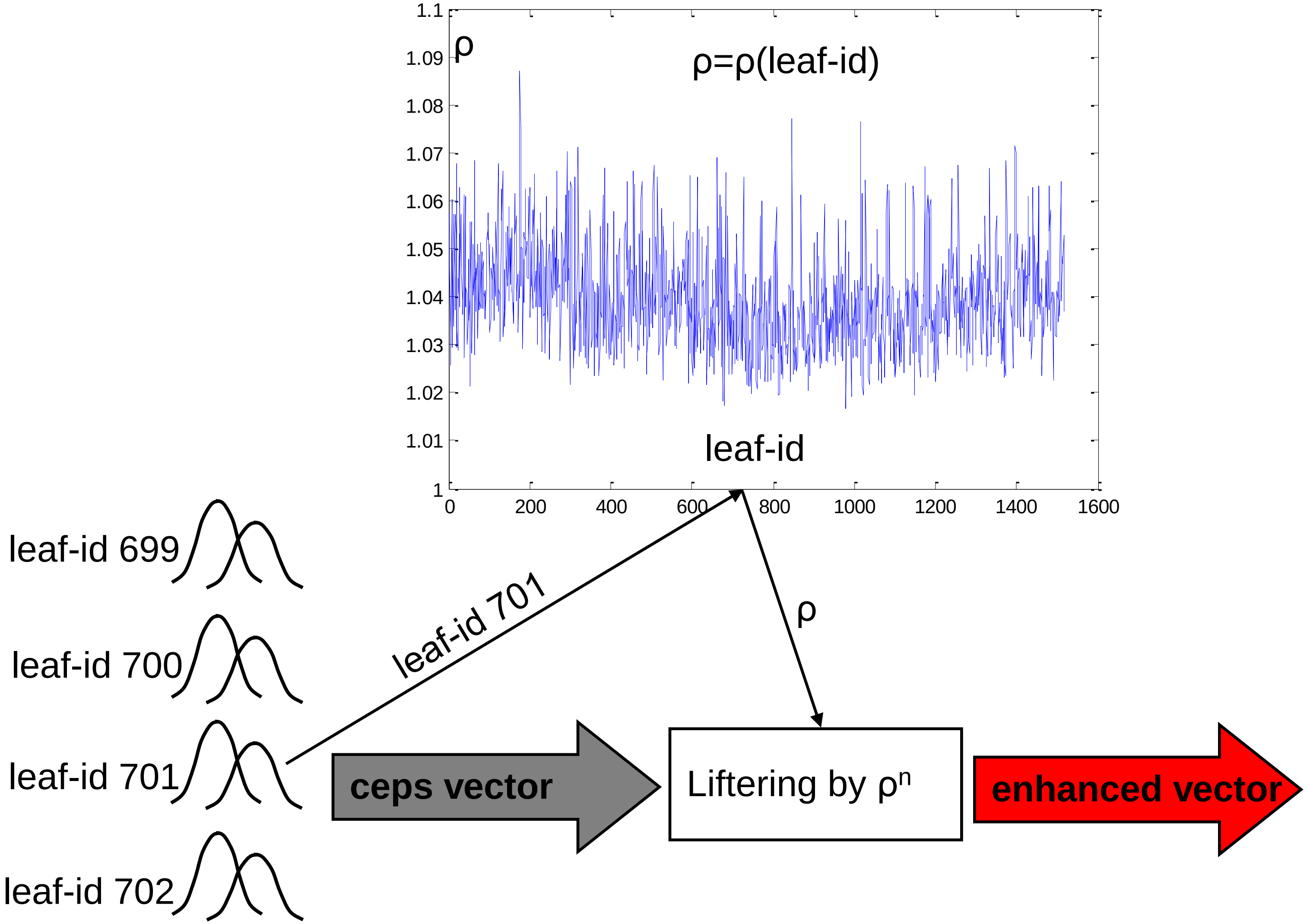
$$\log \rho = \frac{\sum_n n \cdot \log R(n)}{\sum_n n^2}$$

$$R(n) = \sqrt{\frac{E\{c_{real}^2(n)\}}{E\{c_{synt}^2(n)\}}}$$

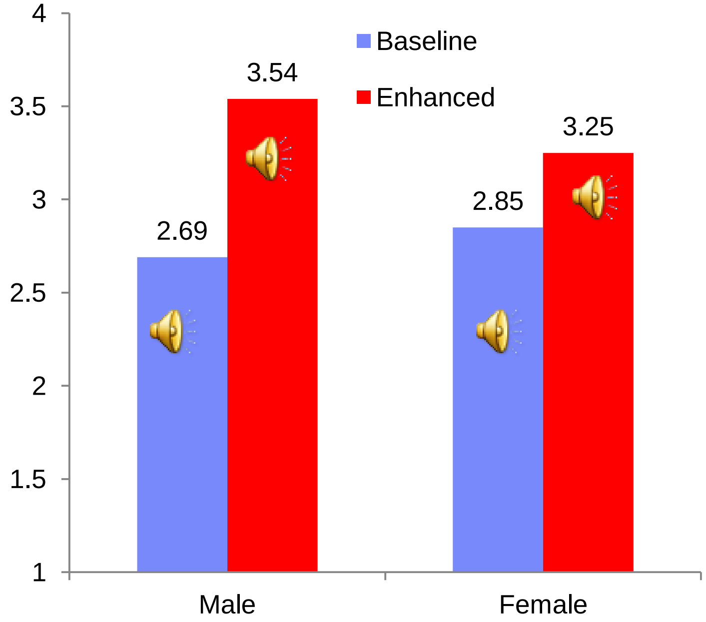




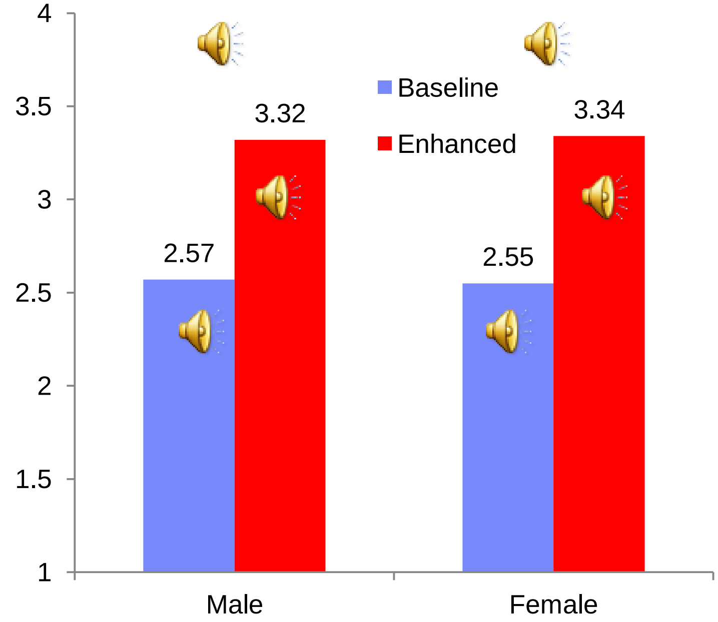
Adaptive statistical enhancement of model segments



Quality and naturalness



Model – template similarity



Challenge 2. Template vs. Model decision

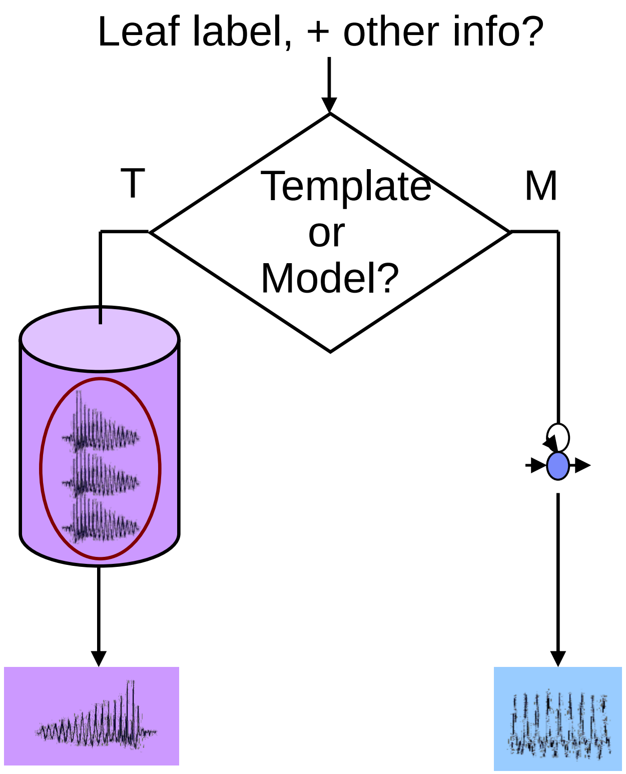
- When to use template and when to use model?

- The decision may be static or dynamic
  - Dynamic: made in in synthesis time depending on the input text
  - Static: a leaf is marked as “template” or as “model” offline prior to the synthesis

➢ In this chapter we consider an offline decision
  - Enables defining and controlling the working point at the HMM -- UnitSelection axis
  - Enables voice size reduction prior to deployment

- The offline decision may be based on *psychoacoustic* properties of speech segments containing in the leaf cluster and/or *phonological* information

➢ We focus on the psychoacoustic aspect
  - Automation, language independent

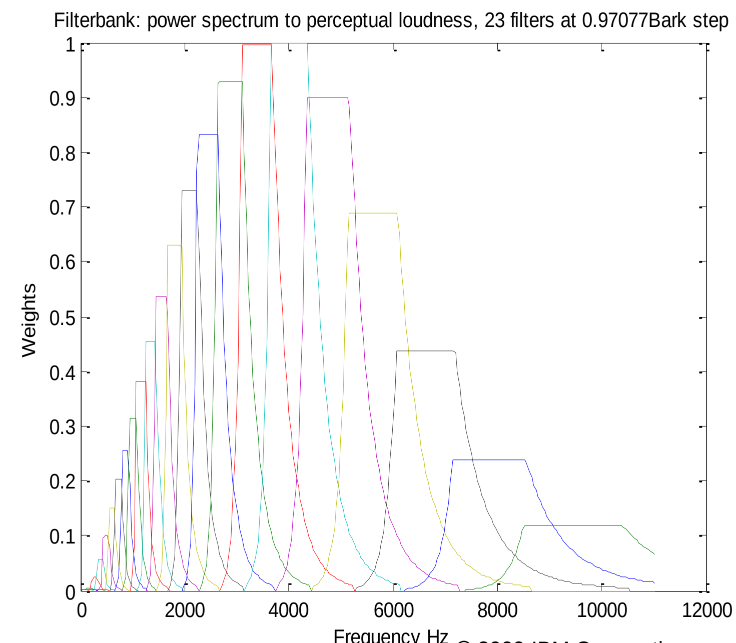


How to devise a Psychoacoustic Modelability scoring?

- *Modelability* score – a degree of perceptual transparency of replacing a natural speech segment by a segment generated from a statistical parametric model trained on similar natural segments
- Observation: segments generated from statistical models are highly stationary
  – HMM TTS emits slowly evolving spectral envelope and stationary excitation
  – HMM TTS does not reproduce transient sounds well enough
- Research hypothesis: Temporal stationarity is indicative of modelability
  – Highly stationary speech segments are transparently replaceable my models
  – Replacement of non-stationary speech segments is audible
- Approach:
  – Develop a segment temporal stationarity score
  – Develop a leaf-cluster modelability score based on the stationarity scores of the containing segments

Segmental perceptual stationarity score

- Divide a segment to T overlapping frames at a high frame update rate, e.g. 1kHz
  - – Use frame length slightly greater than the maximal pitch period
- For example, when 3 segments per phone are used the segment is typically longer than 25 ms and contains tens of frames
- Convert *t*-th frame (*t=1,…,T*) to a Perceptual Loudness Spectrum (*PLS*) adopting the transformation utilized in the Perceptual Linear Predictive ASR front-end
  - – STFT, power spectrum
  - – Filter bank defined on the Bark-scale
  - – Power of 0.33



- PLS vector: $\mathbf{V}(t) = [v_1(t), \ldots, v_N(t)]$
  N is the number of frequency bands (23)
- $v_k(t)$ is a perceptual loudness associated with *k*-th critical frequency band

Segmental perceptual stationarity score

- 1st and 2nd empirical moments of k-th component $v_k(t)$ of the PLS vector

$$M1_k = \frac{1}{T}\sum_{t=1}^{T} v_k(t) \quad M2_k = \frac{1}{T}\sum_{t=1}^{T} v_k^2(t)$$

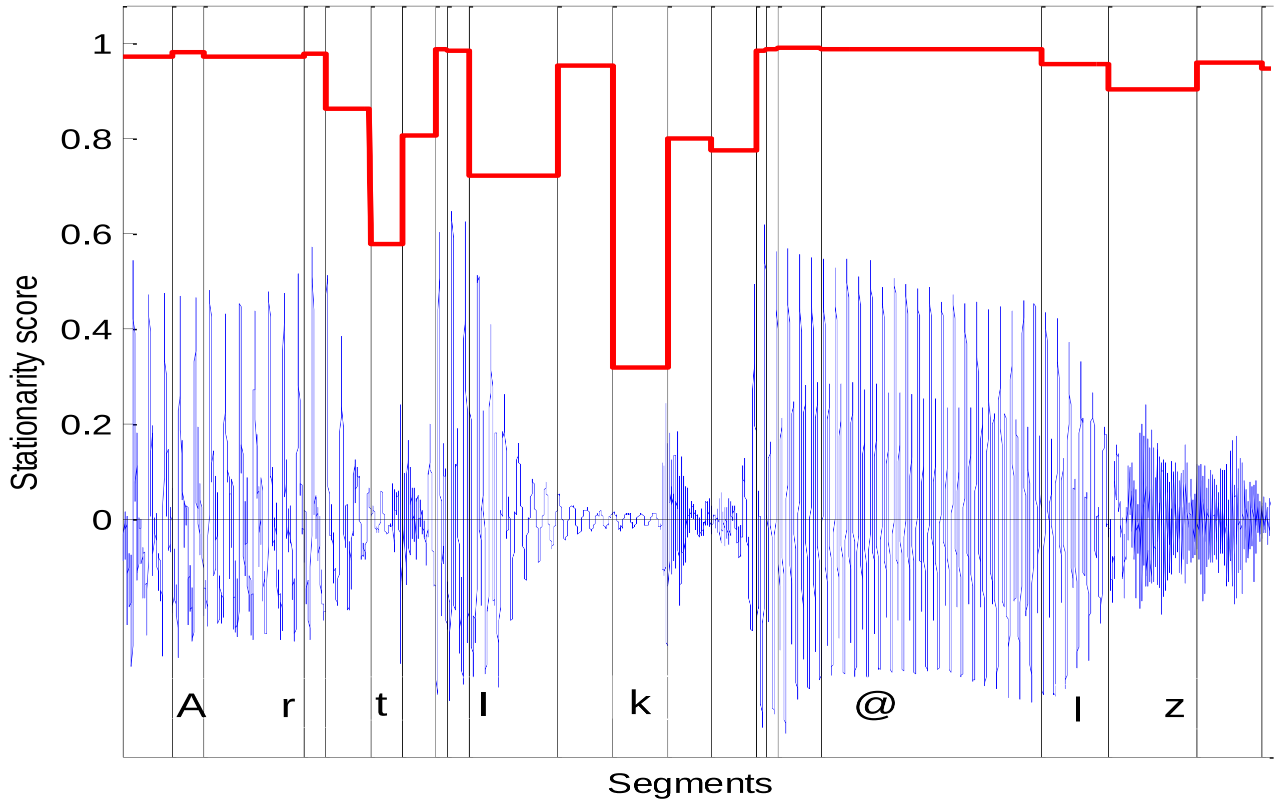
- *Non-stationarity* measure - aggregated relative variability of all the PLS components

$$R = \sum_{k=1}^{N}\left(M2_k - M1_k^2\right) \Big/ \sum_{k=1}^{N} M2_k = 1 - \sum_{k=1}^{N} M1_k^2 \Big/ \sum_{k=1}^{N} M2_k, \; 0 \le R \le 1 - 1/T$$

- A reasonable basis for the stationarity score is $(1-R) \in [1/T, 1]$
- Finally we define the stationarity score S
  - Defined on [0,1]
  - S = 1 for a perfectly stationary segment **V**(1)=**V**(2)=…=**V**(T)
  - S = 0 for singular δ-like segment V(t)=0, $t \neq t_0$

$$\boxed{S = \frac{1 - R - 1/T}{1 - 1/T} = \frac{\sum_{k=1}^{N} M1_k^2 \Big/ \sum_{k=1}^{N} M2_k - 1/T}{1 - 1/T}}$$

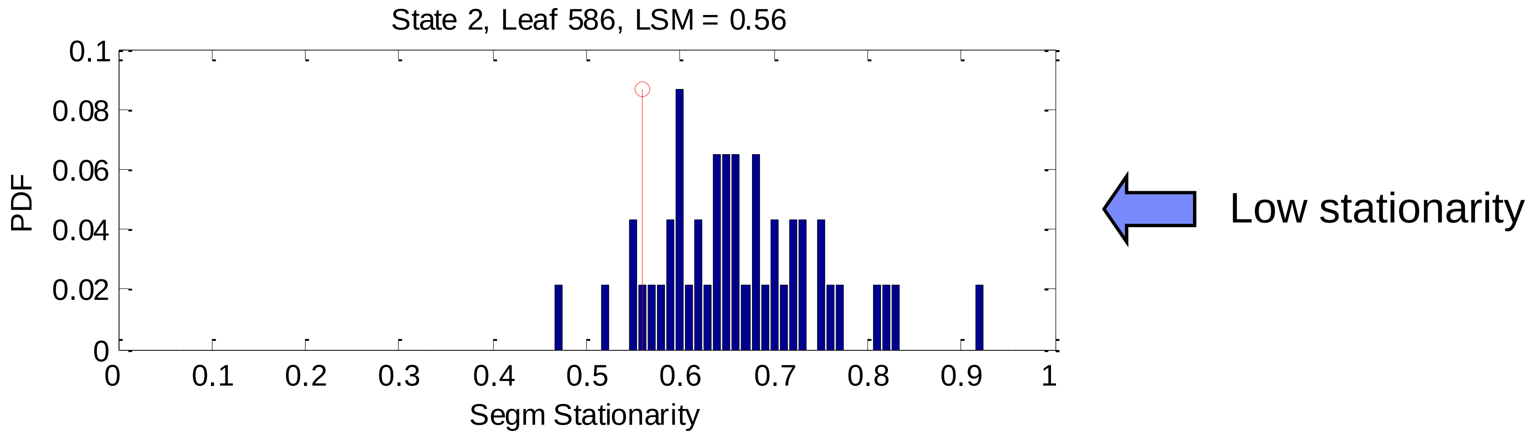
Segment-wise stationarity contour of a natural speech signal

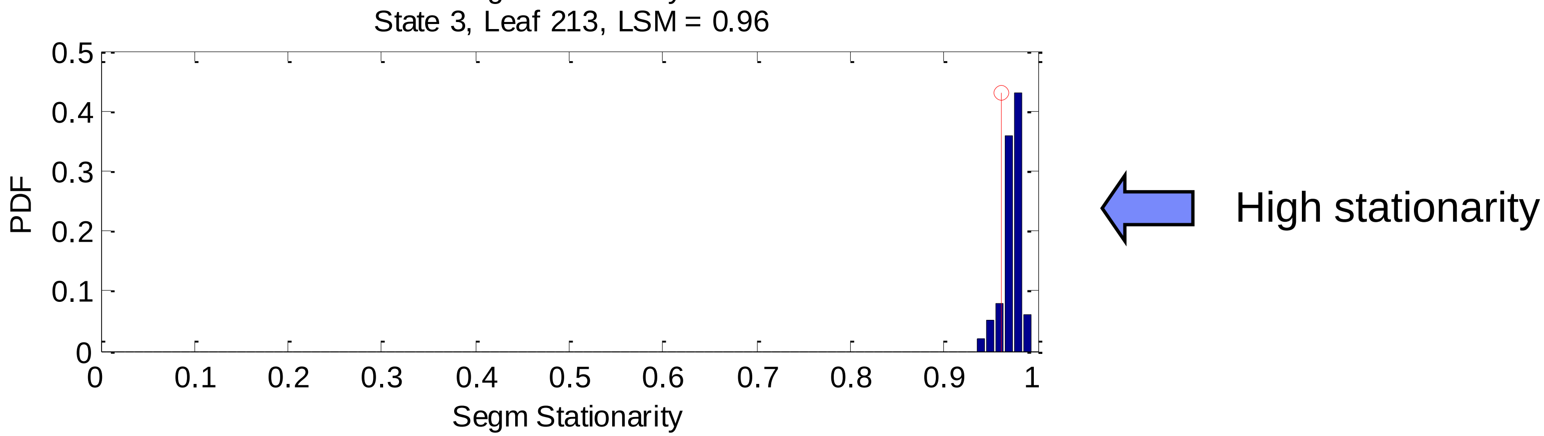


- Stationary segments – slowly evolving spectral envelope and periodic or gaussian excitation, e.g., sustain vowels, fricatives consonants
- Non-stationary segments – all the others, e.g. transients, plosives.

Stationarity Measure of a Leaf Cluster

- Let's define a Leaf Stationarity Measure (LSM) as a low percentile (e.g. 10%) of the segmental stationarity score distribution within the leaf cluster





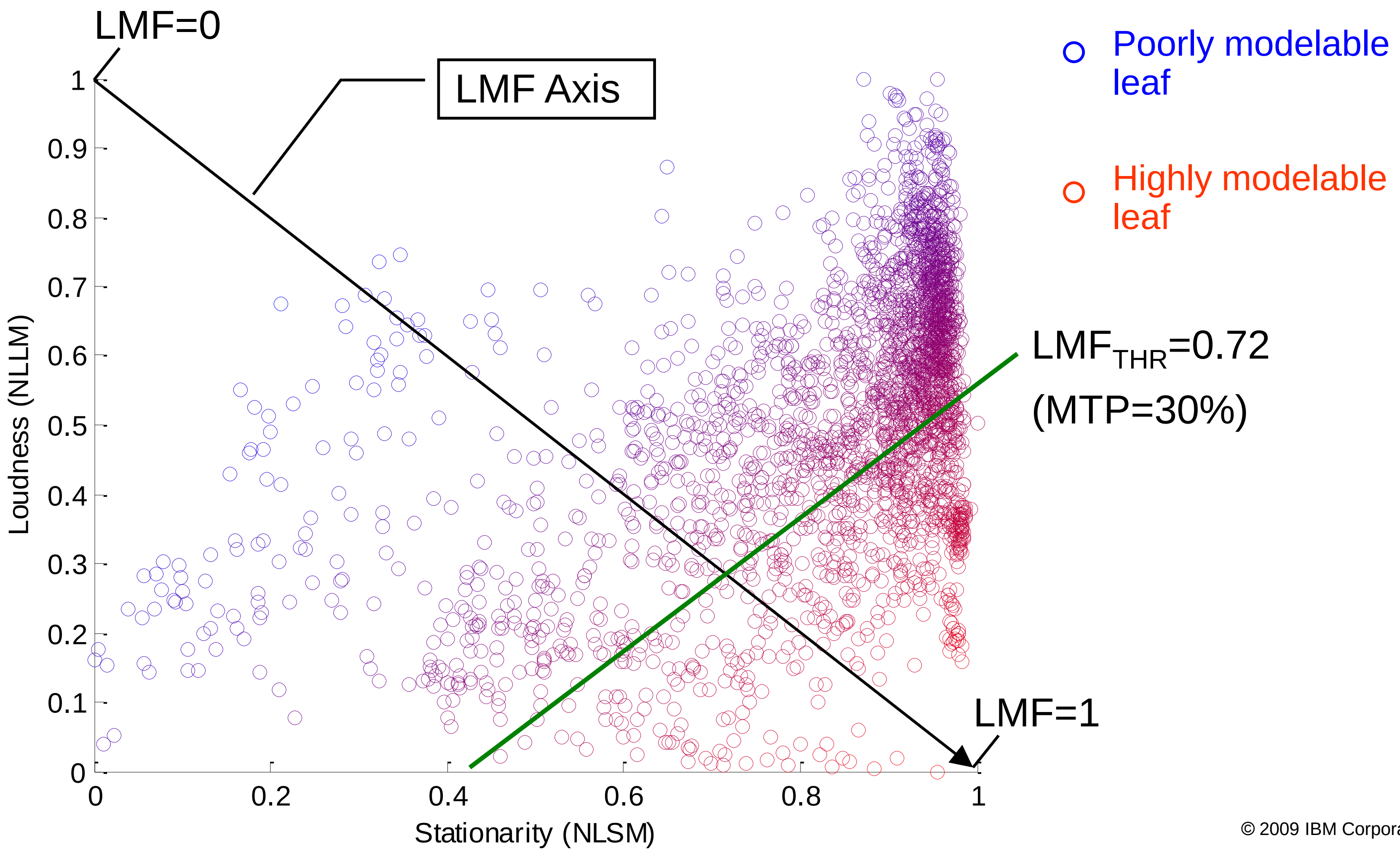
Taking loudness in consideration and voice-level normalization

- The most stationary leaves typically represent the loudest parts of vowels. Their model-based generation is highly audible – revealed by informal evaluation.
- Let's also measure the loudness to take it in consideration
- Perceptual loudness score L of a segment: $L = \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{N} v_k(t) = \sum_{k=1}^{N} M1_k$
- Let's define a Leaf Loudness Measure (LLM) as a high percentile (e.g. 90%) of the loudness score distribution within the leaf-cluster
- The absolute values of the LSM and LLM are irrelevant when we consider a fixed voice dataset. Let's normalize them at the voice level

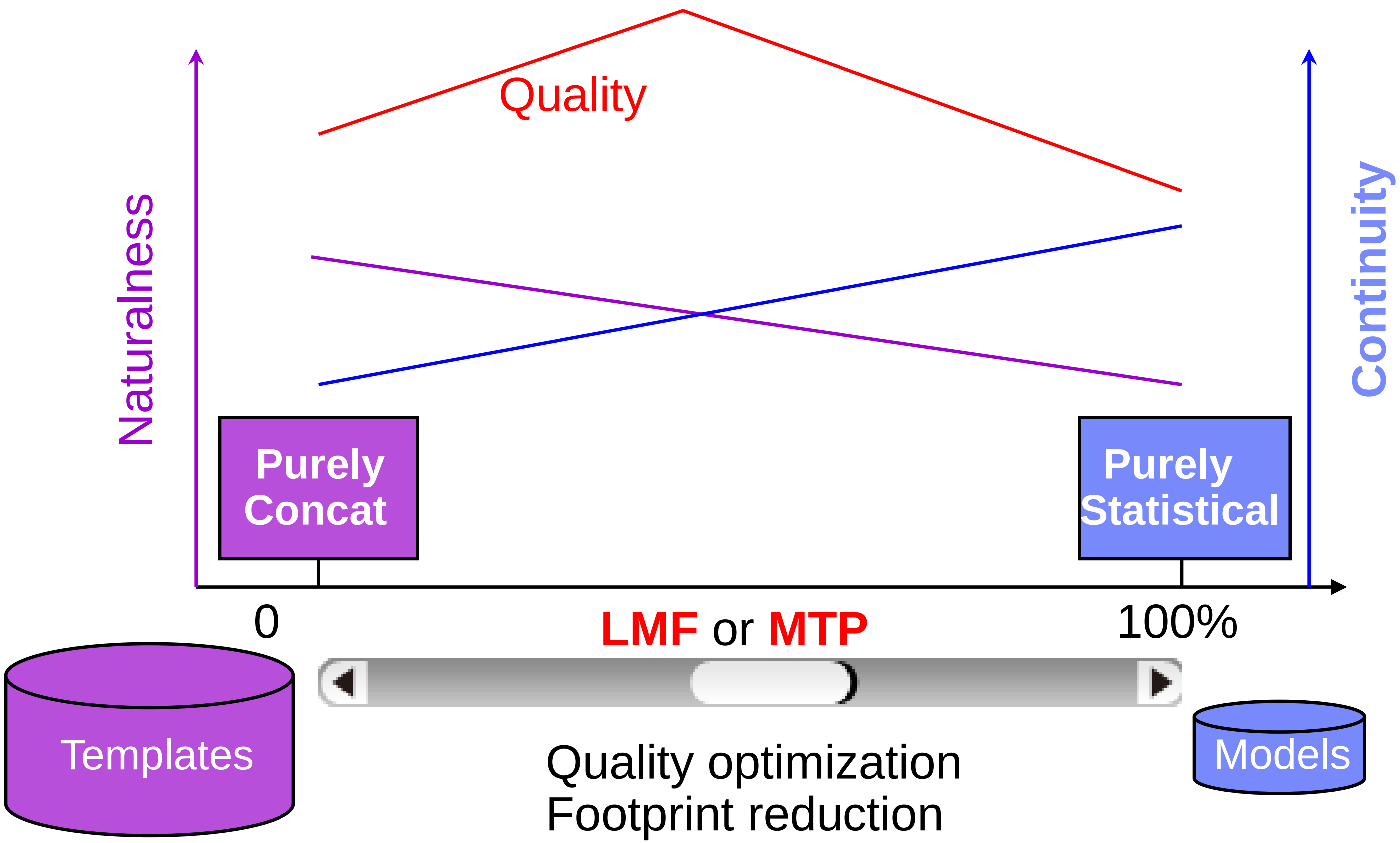
$$nLSM_l = \frac{LSM_l - \min_k LSM_k}{\max_k LSM_k - \min_k LSM_k} \qquad nLLM_l = \frac{LLM_l - \min_k LLM_k}{\max_k LLM_k - \min_k LLM_k}$$

Leaf Modelability Factor (LMF)

$$LMF_l = 0.5 \cdot \left[nLSM_l + \left(1 - nLLM_l\right) \right]$$

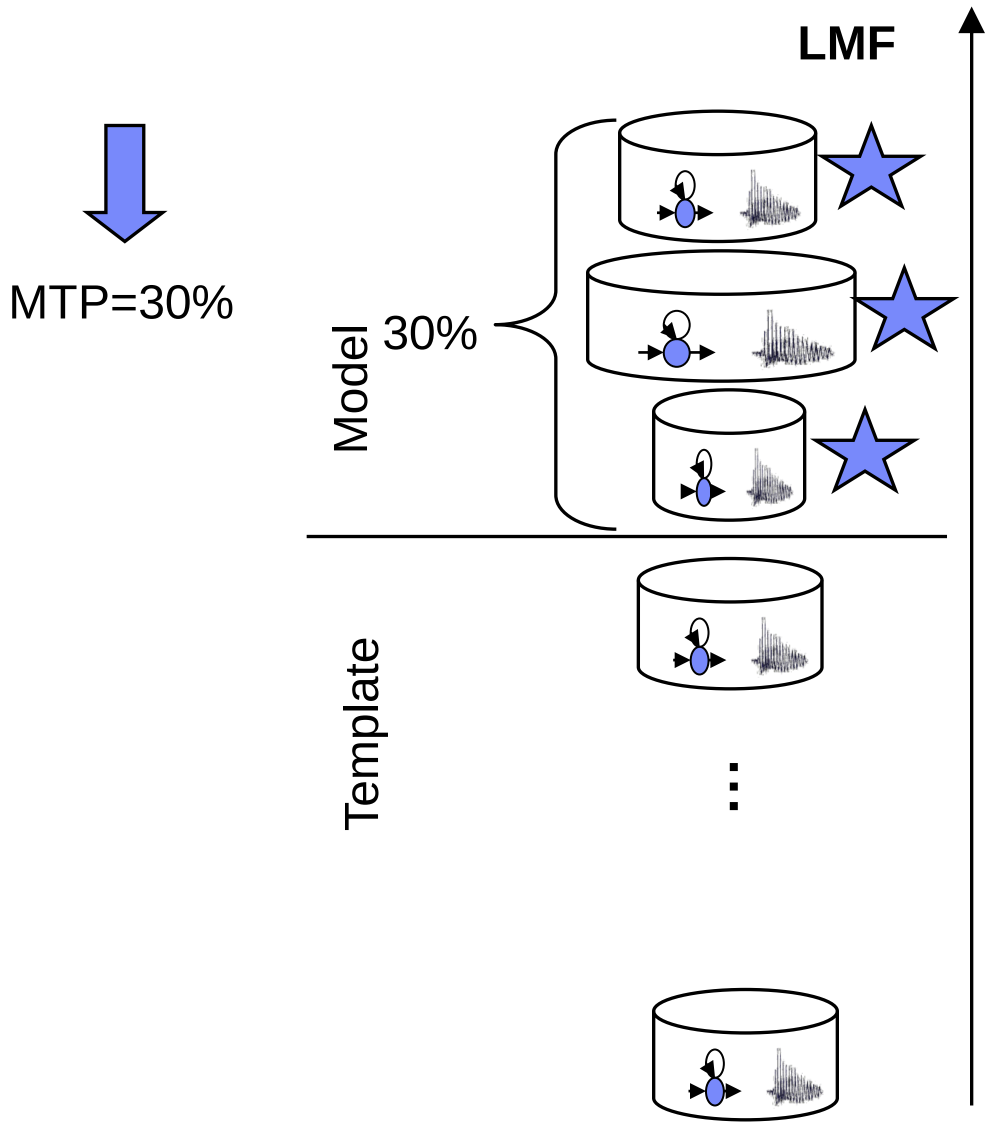


- LMF threshold or related to it Model/Template Proportion (MTP) defines a working point on the "Unit Selection HMM" axis
- MTP is the percentage of the voice dataset represented by models
  - Can be measured as % of segments or as % of the total speech duration



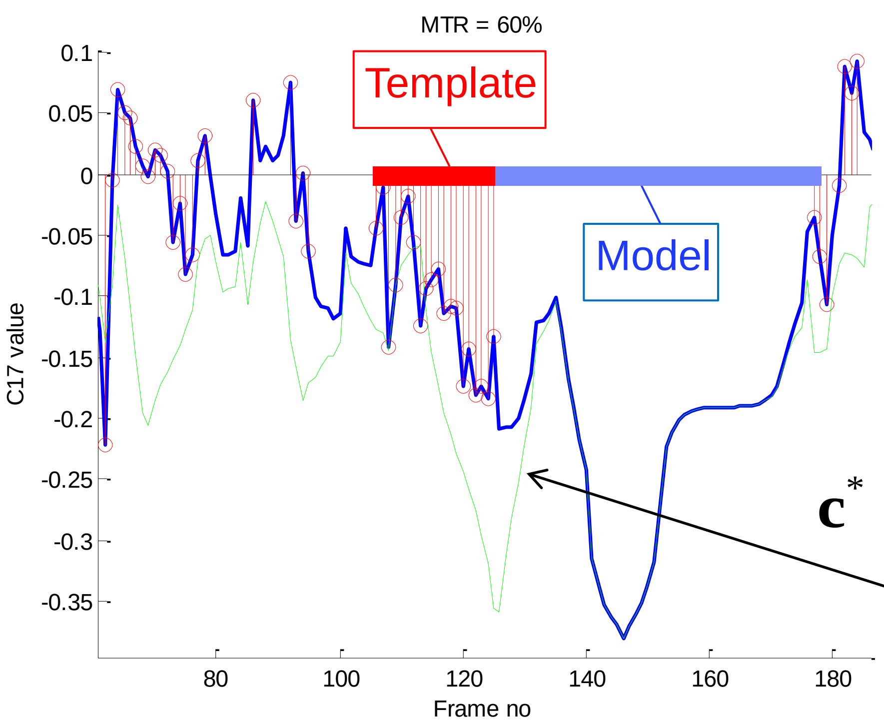
Configuring a mixed synthesis system for a given MTP level

- Sort the leaves by their LMF values
- Select the most modelable leaves containing together MTP% of the speech data
- Declare the selected leaves “model”.
  Declare the remaining leaves “template”
- Prune the voice dataset



Challenge 3. How to assure smoothness at model – template joints

- Cepstral coefficients and F0 within model segments are obtained by the classical Maximum Likelihood parameter generation algorithm which is not aware of the template segments



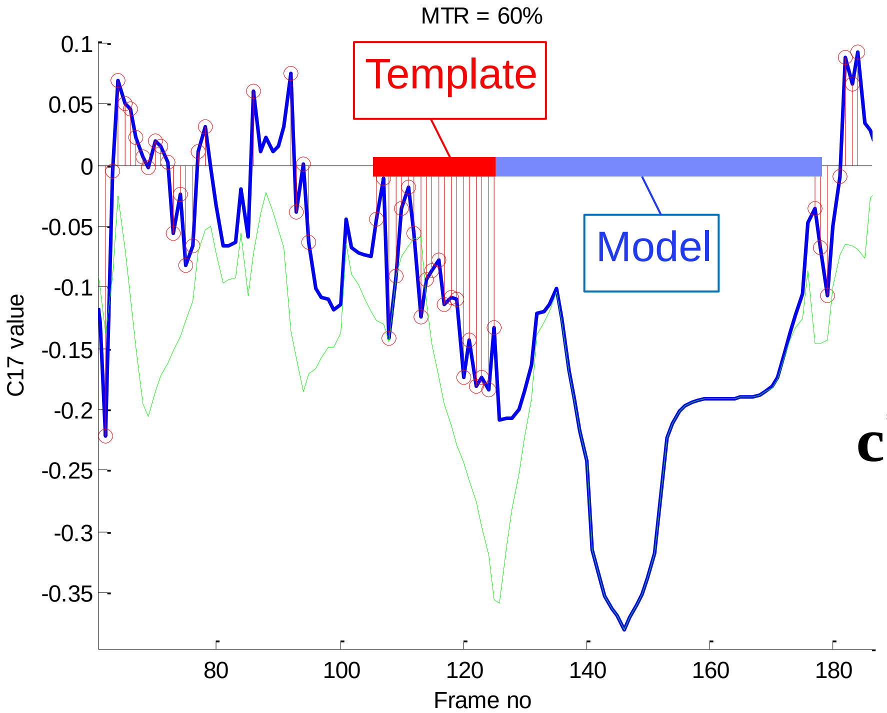
- Parameter values derived from template frames
- Maximum Likelihood trajectory (HMM synthesis)
- “Ideal” trajectory that we would like to see

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \left(\mathbf{c}^T \mathbf{W}^T \boldsymbol{\Sigma}^{-1} \mathbf{W} \mathbf{c} - 2\mathbf{c}^T \mathbf{W}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} \right)$$

$$\mathbf{W}^T \boldsymbol{\Sigma}^{-1} \mathbf{W} \cdot \mathbf{c}^* = \mathbf{W}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}$$

Constrained ML trajectory (not new, e.g. Tiomkin et al 2011)

- Find the ML trajectory passing through the points (n_k,t_k) given by the template frames
  - Constrained ML trajectory or ML interpolation in the acoustic parameter space



$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \left(\mathbf{c}^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \mathbf{W} \mathbf{c} - 2 \mathbf{c}^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} \right)$$

$$s.t.: \quad \mathbf{c}(n_k) = t_k$$

Hmm… optimization with equality-style constraints

$$\mathbf{c}^{*} = \arg\min_{\mathbf{c}} \left(\mathbf{c}^{\mathbf{T}} \mathbf{W}^{\mathbf{T}} \boldsymbol{\Sigma}^{\mathbf{-1}} \mathbf{W} \mathbf{c} - 2 \mathbf{c}^{\mathbf{T}} \mathbf{W}^{\mathbf{T}} \boldsymbol{\Sigma}^{\mathbf{-1}} \boldsymbol{\mu} \right)$$

$$s.t.: \quad c_{n_k} = t_k$$

- Do not use Lagrange multipliers which are useful for solving a general problem

max F(x), s.t.: G(x) = const

- It yields an overcomplicated solution

- There is a simple and efficient solution to our problem

An exercise

$$\min_{\mathbf{x}} \left\| \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ t \\ x_3 \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right\|^2$$

$$\min_{\mathbf{x}} \left\| \begin{bmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \\ a_{31} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} - t \cdot \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right\|^2$$

Now more formally and applying to our case

$$\arg\min_{\mathbf{c}}\left(\mathbf{c}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{W}\mathbf{c}-2\mathbf{c}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right)$$

$$\mathbf{c}=\mathbf{T}\mathbf{t}+\mathbf{M}\mathbf{m}$$

- Vector **t** is known – its components are the parameter values at **t**emplate frames
- Vector **m** is unknown – its components are the parameter values at **m**odel frames
- The role of the matrices **T** and **M** is to place the template and model components at their respective positions in the entire combined trajectory **c**

A toy example:

$$\underset{\mathbf{T}}{\begin{bmatrix}1&0&0\\0&1&0\\0&0&0\\0&0&0\\0&0&1\end{bmatrix}}\cdot\underset{\mathbf{t}}{\begin{bmatrix}t_1\\t_2\\t_3\end{bmatrix}}+\underset{\mathbf{M}}{\begin{bmatrix}0&0\\0&0\\1&0\\0&1\\0&0\end{bmatrix}}\cdot\underset{\mathbf{m}}{\begin{bmatrix}m_1\\m_2\end{bmatrix}}=\underset{\mathbf{c}}{\begin{bmatrix}t_1\\t_2\\m_1\\m_2\\t_3\end{bmatrix}}$$

- Substitution of $\mathbf{c} = \mathbf{T} \cdot \mathbf{t} + \mathbf{M} \cdot \mathbf{m}$

  in $\arg\min_{\mathbf{c}} \left(\mathbf{c}^{\mathbf{T}} \mathbf{W}^{\mathbf{T}} \boldsymbol{\Sigma}^{\mathbf{-1}} \mathbf{W} \mathbf{c} - 2\mathbf{c}^{\mathbf{T}} \mathbf{W}^{\mathbf{T}} \boldsymbol{\Sigma}^{\mathbf{-1}} \boldsymbol{\mu} \right)$

  yields

$$\mathbf{m}^{*} = \arg\min_{\mathbf{m}} [\mathbf{m}^{T} \mathbf{M}^{T} \mathbf{W}^{T} \boldsymbol{\Sigma}^{-1} \mathbf{W} \mathbf{M} \mathbf{m} - 2\mathbf{m}^{T} \mathbf{M}^{T} \mathbf{W}^{T} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} - \mathbf{W} \mathbf{T} \mathbf{t})]$$

- Finally the unknown points on the constrained trajectory are obtained by solving the linear equation:

$$\mathbf{M}^{T} \mathbf{W}^{T} \boldsymbol{\Sigma}^{-1} \mathbf{W} \mathbf{M} \cdot \mathbf{m}^{*} = \mathbf{M}^{T} \mathbf{W}^{T} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} - \mathbf{W} \mathbf{T} \mathbf{t})$$

Constrained ML trajectory – concluding notes

- The matrix of the equations inherits the sparse diagonal structure from the classical unconstrained solution
  - Only the delta relations tie frames to each other
  - For the usual delta calculation algorithm any single equation cannot tie more then 3 consecutive frames
- We throw out many equations present in the classical unconstrained system
- Hence the whole set of the equations can be split to independent separately solved subsets of a small size
  - Two consecutive template frames lead to a split

$$t \quad t \quad \underbrace{m \quad m \quad m} \quad t \quad t \quad t \quad \underbrace{m \quad m \quad m} \quad t \quad t \quad \underbrace{m} \quad t \quad t$$

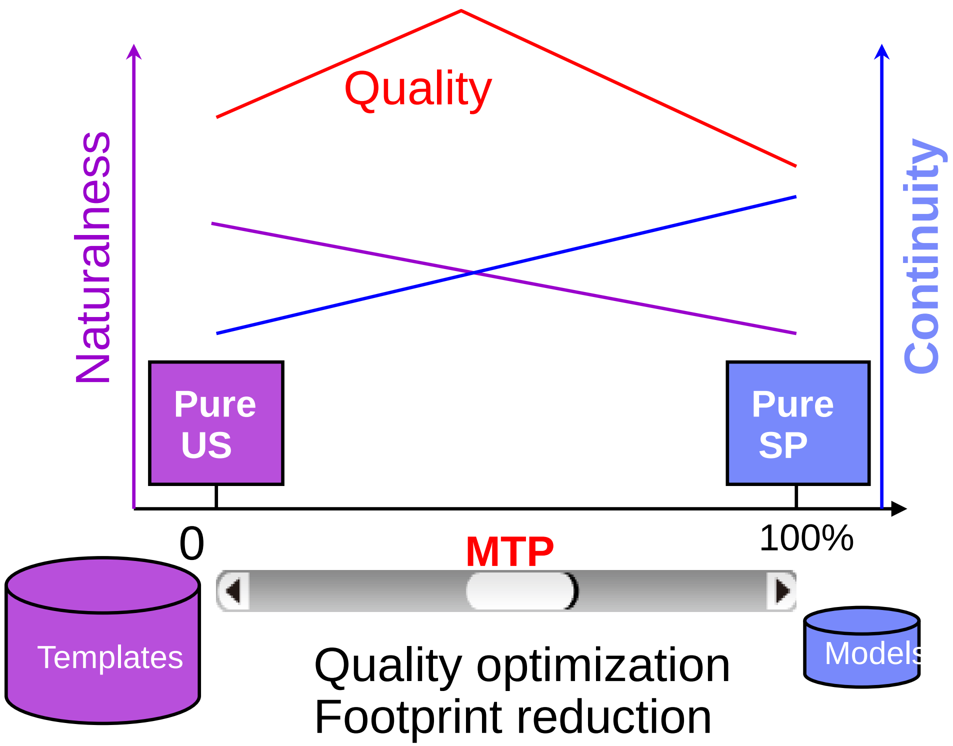
How to deal with the phases?

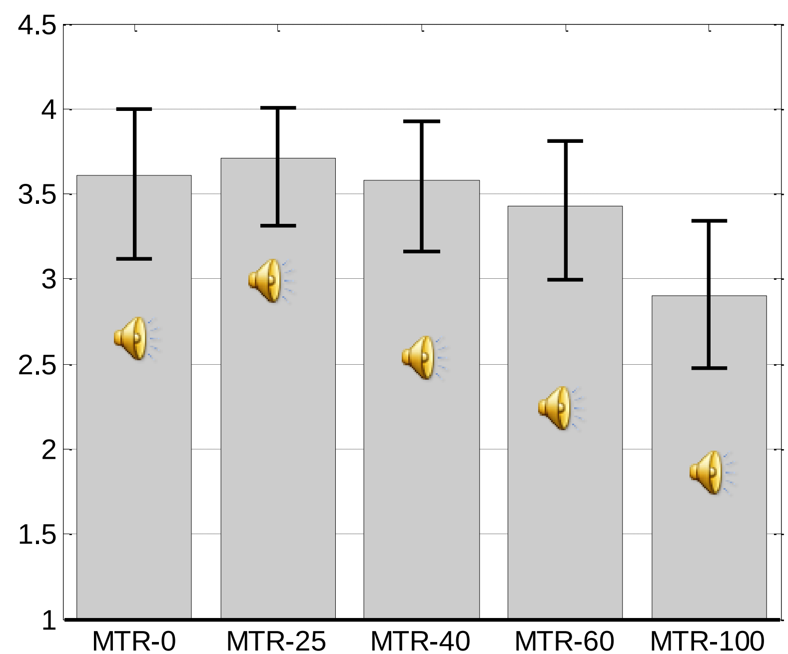
- Case 1. The template segments are represented by their waveforms.
  - Generate the model segments waveform
  - Find the best (e.g. max correlation) time offsets between the template and model waveform
  - Shift and overlap-add

- Case 2. The template segments are parameterized – we used a harmonic + noise representation
  - Convert the model segments to the same harmonic + noise structure
  - Interpolate/smooth respective harmonic phases over the template-model joints
  - Convert to the waveform
  - Overlap-add

Reality vs. the Idea – feasibility test

- The idea begin to look realistic

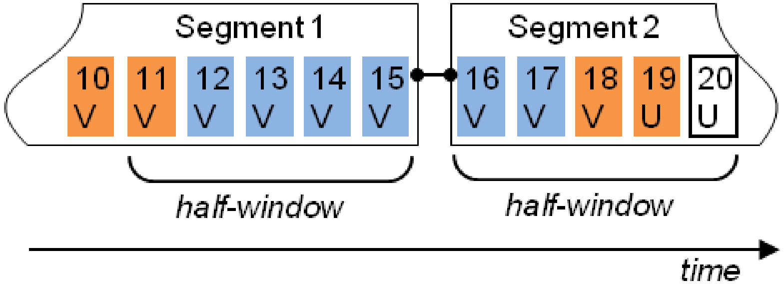


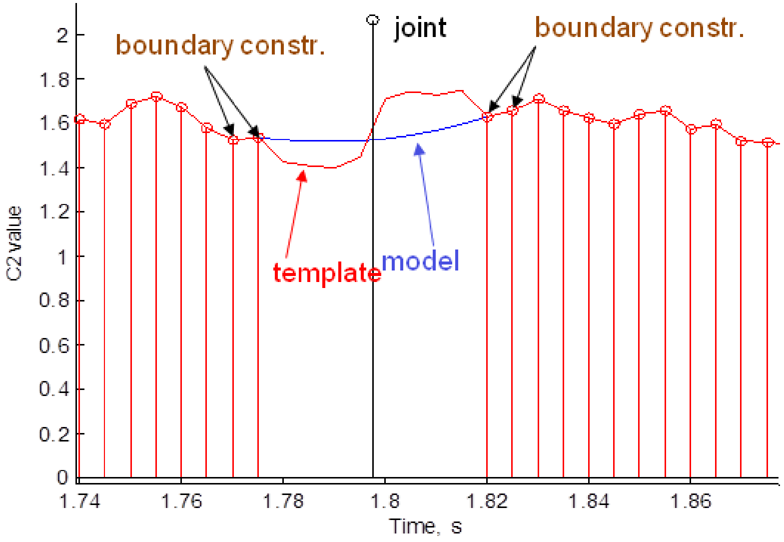


Challenge 4. How to reduce discontinuities at template-template joints

- Partially addressed by the works that used the mixed synthesis as a means to overcome the sparseness of the units inventory
  - When there is no *suitable* unit generate the segment from the model using the constrained ML parameter trajectories
  - The notion *of suitable* includes a low joint cost (Tiomkin et al, 2011)
- The drawback – insertion of a model segment ad hoc might be audible even if it smoothly joins the surrounding natural segments
- An alternative approach – generate from the model only a *small amount of frames* surrounding the joint
  - Virtually inaudible
  - <u>All</u> the voiced joints may be processed to guarantee smoothness

Frame level template by model substitution for joints smoothing

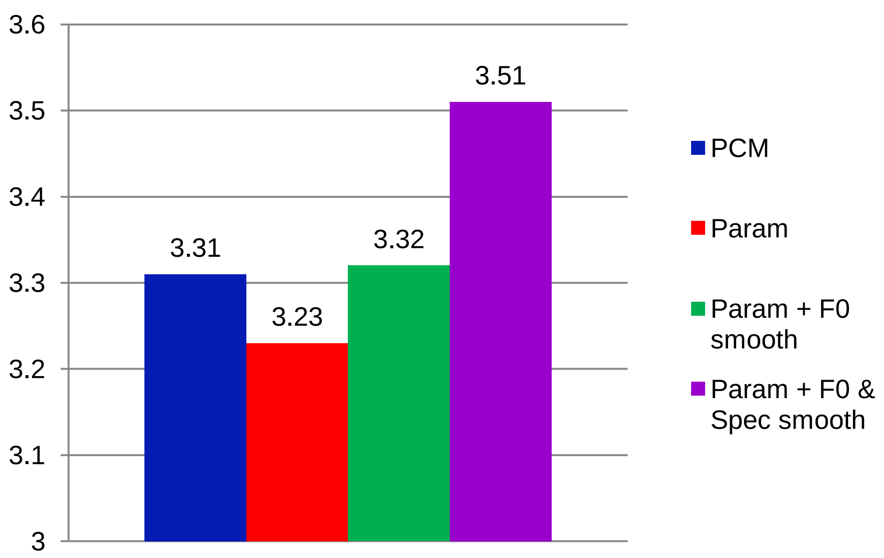




- The blue frames are replaced by the model
- The brown frames establish the boundary constraints for the ML trajectory generation

Comparative evaluation of the frame-level joints smoothing effect

- Phase smoothing is facilitated by the full parameterization
  - Not only the model segments but also the template segments are parameterized
- Full parameterization with the joints smoothing outperforms the PCM based segments without the joints smoothing



Acknowlegements

- Thanks to my colleague Slava Shechtman from IBM and our partner Vincent Pollet from Nuance who worked together with me on the matters presented and discussed in this lecture

Thanks for your attention!