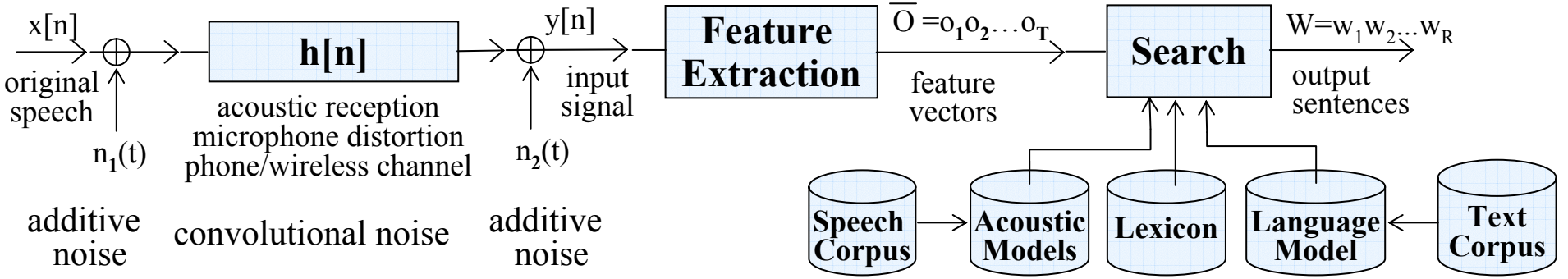
14.0 Robustness for Acoustic Environment

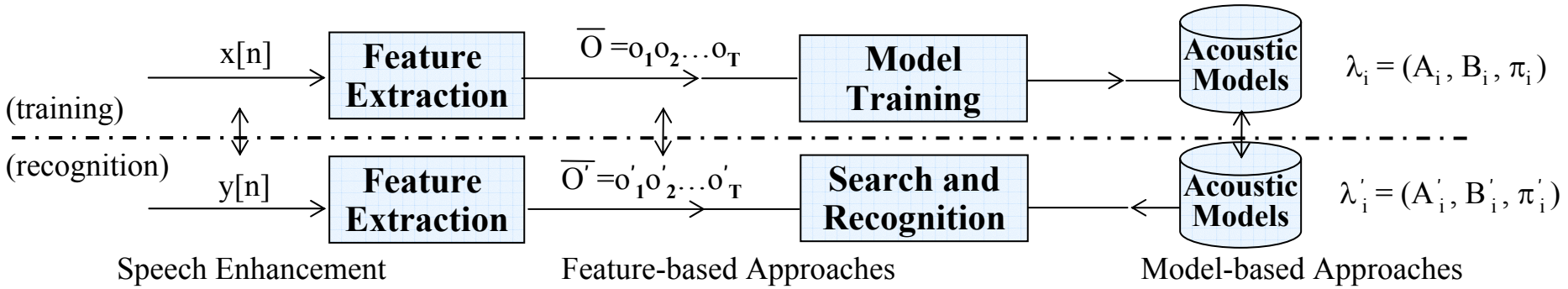
References: 1. 10.5, 10.6 of Huang

2. "Robust Speech Recognition in Additive and Convolutional Noise Using Parallel Model Combination" Computer Speech and Language, Vol. 9, 1995
3. "A Vector Taylor Series Approach for Environment Independent Speech Recognition", International Conference on Acoustics, Speech and Signal Processing, 1996
4. "Signal Bias Removal by Maximum Likelihood Estimation for Robust Telephone Speech Recognition", IEEE Trans. on Speech & Audio Processing, Jan 1996
5. "RASTA Processing of Speech", IEEE Trans. on Speech & Audio Processing, April 1994
6. 3.8 of Duda, Hart and Stork, "Pattern Classification", John Wiley and sons, 2001
7. "An Application of Discriminative Feature Extraction to Filter-bank-based Speech Recognition", IEEE Trans. on Speech & Audio Processing, Feb 2001
8. "Suppression of Acoustic Noise in Speech Using Spectral Subtraction" ,IEEE Trans. on Acoustics, Speech and Signal Processing, Apr 1979

Mismatch in Statistical Speech Recognition

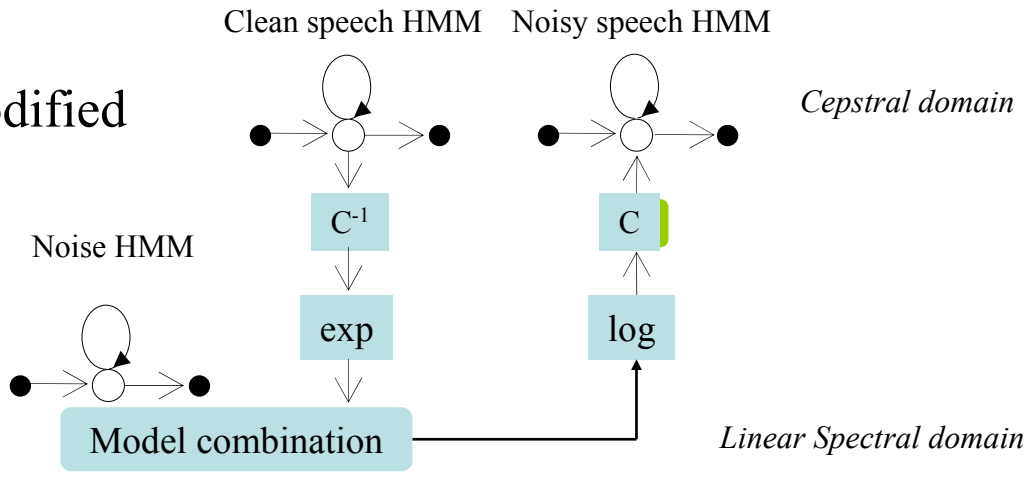


- **Mismatch between Training/Recognition Conditions**
  - Mismatch in Acoustic Environment — Environmental Robustness
    - additive/convolutional noise, etc.
  - Mismatch in Speaker Characteristics — Speaker Adaptation
  - Mismatch in Other Acoustic Conditions
    - speaking mode:read/prepared/conversational/spontaneous speech, etc.
    - speaking rate, dialects/accents, emotional effects, etc.
  - Mismatch in Lexicon — Lexicon Adaptation
    - out-of-vocabulary(OOV) words, pronunciation variation, etc.
  - Mismatch in Language Model — Language Model Adaptation
    - different task domains give different N-gram parameters, etc.
- **Possible Approaches for Acoustic Environment Mismatch**



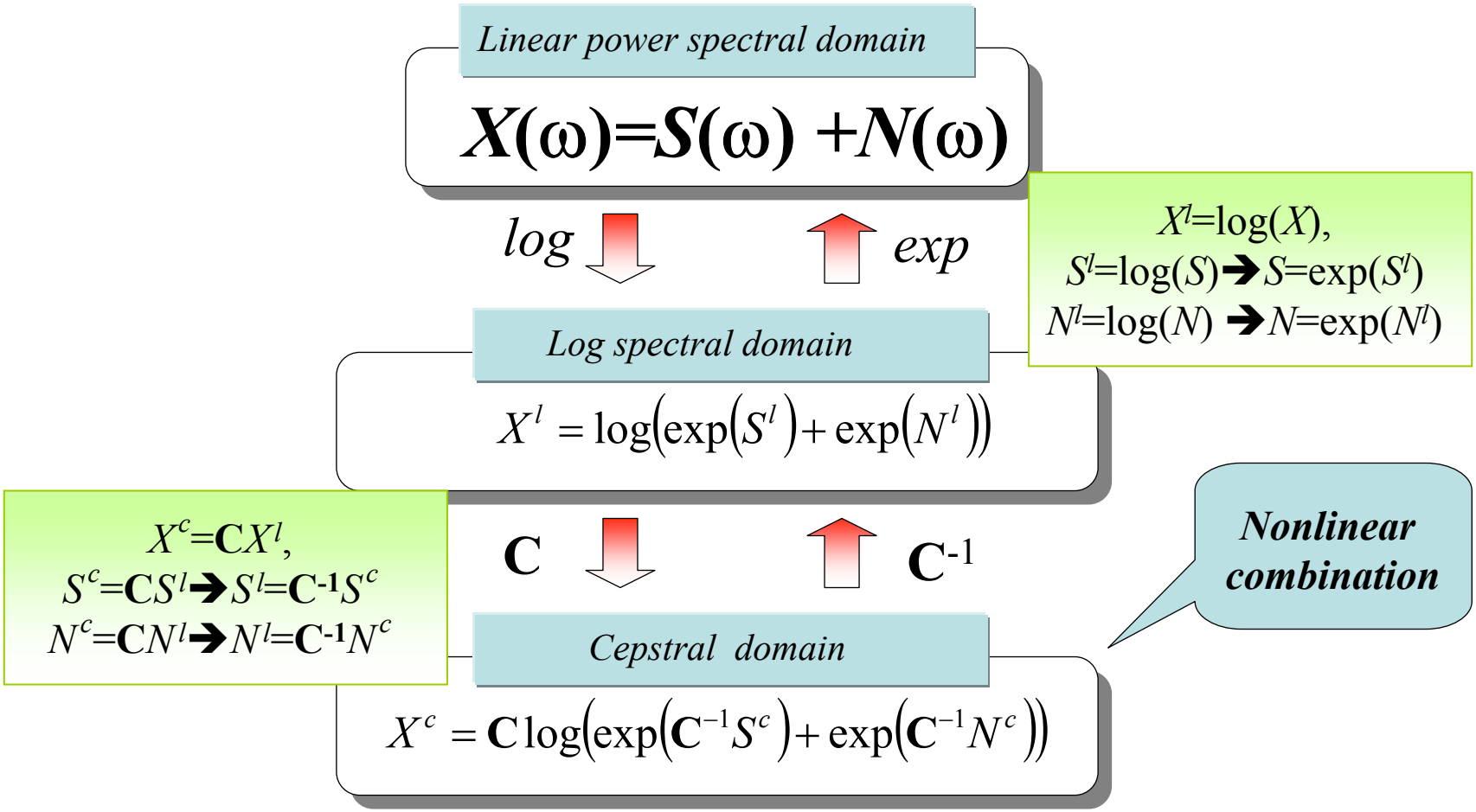
Model-based Approach Example 1— Parallel Model Combination (PMC)

- **Basic Idea**
  - primarily handling the additive noise
  - the best recognition accuracy can be achieved if the models are trained with matched noisy speech, which is impossible
  - a noise model is generated in real-time from the noise collected in the recognition environment during silence period
  - combining the noise model and the clean-speech models in real-time to generate the noisy-speech models
- **Basic Approaches**
  - performed on model parameters in cepstral domain
  - noise and signal are additive in linear spectral domain rather than the cepstral domain, so transforming the parameters back to linear spectral domain for combination
  - allowing both the means and variances of a model set to be modified
- **Parameters used :**
  - the clean speech models
  - a noise model



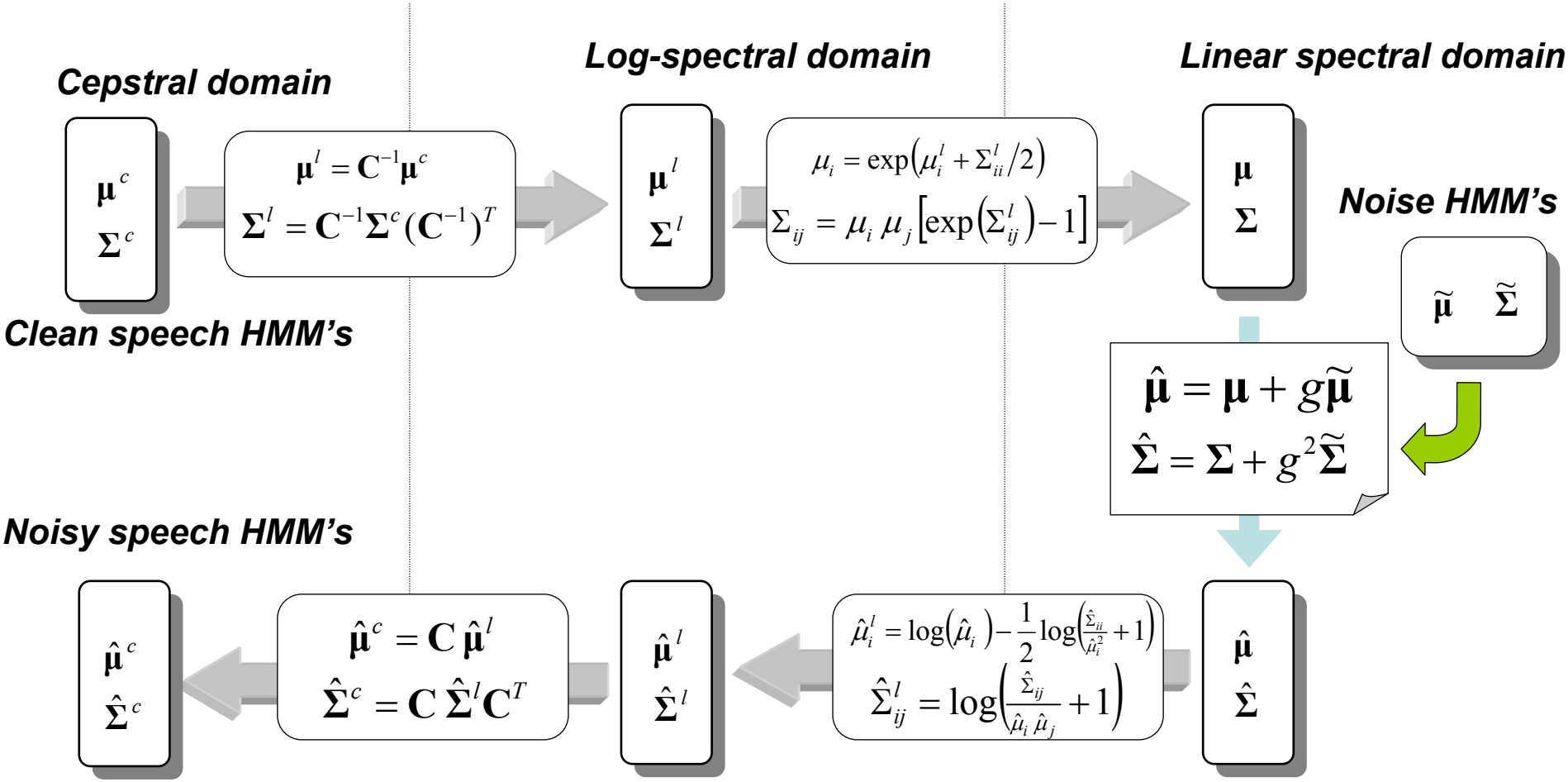
Model-based Approach Example 1 — Parallel Model Combination (PMC)

- **The Effect of Additive Noise in the Three Different Domains and the Relationships**



Model-based Approach Example 1 — Parallel Model Combination (PMC)

- **The Steps of Parallel Model Combination (Log-Normal Approximation) :**
  - based on various assumptions and approximations to simplify the mathematics and reduce the computation requirements



Model-based Approach Example 2— Vector Taylor's Series

- **Basic Approach**
  - Similar to PMC, the noisy-speech models are generated by combination of clean speech HMM's and the noise HMM
  - Unlike PMC, this approach combines the model parameters directly in the log-spectral domain using Taylor's Series approximation
  - Taylor's Series Expansion for l-dim functions:

$$f(x) = f(c) + \frac{df(c)}{dx}(x-c) + \frac{1}{2}\frac{d^2 f(c)}{dx^2}(x-c)^2 + \cdots \frac{1}{n!}\frac{d^n f(c)}{dx^n}(x-c)^n \cdots$$

- **Given a nonlinear function z=g(x, y)**
  - x, y, z are n-dim random vectors
  - assuming the mean of x, y, μ_x, μ_y and covariance of x, y, Σ_x, Σ_y are known
  - then the mean and covariance of z can be approximated by the Vector Taylor's Series

$$\mu_z{}^i = g(\mu_x{}^i, \mu_y{}^i) + \frac{1}{2}\left(\frac{\partial^2 g(\mu_x{}^i, \mu_y{}^i)}{\partial x^{i\,2}}\Sigma_x{}^{ii} + \frac{\partial^2 g(\mu_x{}^i, \mu_y{}^i)}{\partial y^{i\,2}}\Sigma_y{}^{ii}\right)$$

$$\Sigma_z{}^{ij} = \left(\frac{\partial g(\mu_x{}^i, \mu_y{}^i)}{\partial x_i}\frac{\partial g(\mu_x{}^j, \mu_y{}^j)}{\partial x_j}\right)\Sigma_x{}^{ij} + \left(\frac{\partial g(\mu_x{}^i, \mu_y{}^i)}{\partial y_i}\frac{\partial g(\mu_x{}^j, \mu_y{}^j)}{\partial y_j}\right)\Sigma_y{}^{ij}, \quad \text{i, j: dimension index}$$

- **Now Replacing z=g (x, y) by the Following Function**

$$\mathbf{X}^l = \log(\exp(\mathbf{S}^l) + \exp(\mathbf{N}^l))$$

  - the solution can be obtained

$$\mu_x{}^i = \log(e^{\mu_s{}^i} + e^{\mu_n{}^i}) + \frac{1}{2}\frac{e^{\mu_s{}^i + \mu_n{}^i}}{(e^{\mu_s{}^i} + e^{\mu_n{}^i})^2}(\Sigma_s{}^{ii} + \Sigma_n{}^{ii})$$

$$\Sigma_x{}^{ij} = \left(\frac{e^{\mu_s{}^i}}{e^{\mu_s{}^i} + e^{\mu_n{}^i}}\right)\left(\frac{e^{\mu_s{}^j}}{e^{\mu_s{}^j} + e^{\mu_n{}^j}}\right)\Sigma_s{}^{ij} + \left(\frac{e^{\mu_n{}^i}}{e^{\mu_s{}^i} + e^{\mu_n{}^i}}\right)\left(\frac{e^{\mu_n{}^j}}{e^{\mu_s{}^j} + e^{\mu_n{}^j}}\right)\Sigma_n{}^{ij}$$

Feature-based Approach Example 1— Cepstral Mean Subtraction(CMS) and Signal Bias Removal

- **Primarily for Covolutional Noise**
  - convolutional noise in time domain becomes additive in cepstral domain (MFCC)
    $y[n] = x[n]*h[n] \Rightarrow \bar{y} = \bar{x}+\bar{h}$, $\bar{x}, \bar{y}, \bar{h}$ in cepstral domain
  - most convolutional noise changes only very slightly for some reasonable time interval
    $\bar{x} = \bar{y}-\bar{h}$ if $\bar{h}$ can be estimated
- **Cepstral Mean Subtraction(CMS)**
  - assuming $E[\bar{x}] = 0$, then $E[\bar{y}] = \bar{h}$, averaged over an utterance or similar, or a longer time interval

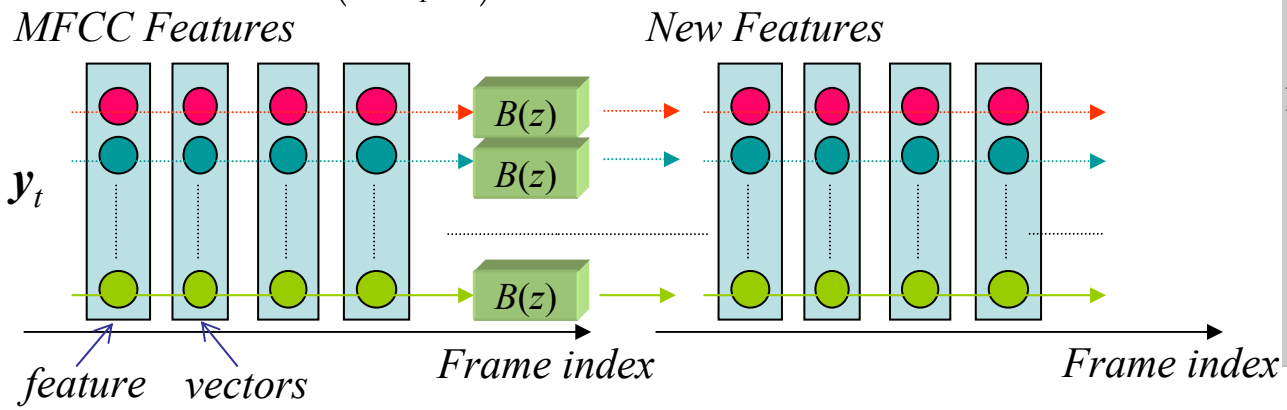
    $\bar{x}_{\mathbf{CMS}} = y-E[\bar{y}]$
  - CMS features are immune to convolutional noise
    x[n] convolved with any h[n] gives the same $\bar{x}_{\mathbf{CMS}}$
  - CMS doesn't change delta or delta-delta cepstral coefficients
- **Signal Bias Removal**
  - estimating $\bar{h}$ by the maximum likelihood criteria

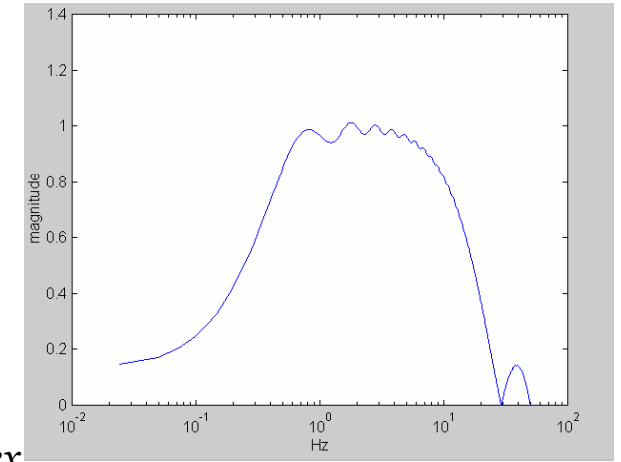
    $\bar{h}^* = \arg\max_{\bar{h}} \text{Prob}[\bar{Y} = (\bar{y}_1\bar{y}_2\ldots\bar{y}_T) \mid \lambda , \bar{h}]$, λ : HMM for the utterance $\bar{Y}$
  - iteratively obtained via EM algorithm

Feature-based Approach Example 2 — RASTA (Relative Spectral) Temporal Filtering

- **Temporal Filtering**
  - each component in the feature vector (MFCC coefficients) considered as a signal or "time trajectories" when the time index (frame number) progresses
  - the frequency domain of this signal is called the "modulation frequency"
  - performing filtering on these signals
- **RASTA Processing :**
  - assuming the rate of change of nonlinguistic components in speech (e.g. additive and convolutional noise) often lies outside the typical rate of the change of the vocal tract shape
  - designing filters to try to suppress the spectral components in these "time trajectories" that change more slowly or quickly than this typical rate of change of the vocal tract shape
  - a specially designed temporal filter for such "time trajectories"

$$B(z)=\frac{a_0+a_1z^{-1}+a_3z^{-3}+a_4z^{-4}}{\left(1-b_1z^{-1}\right)z^{-4}}$$

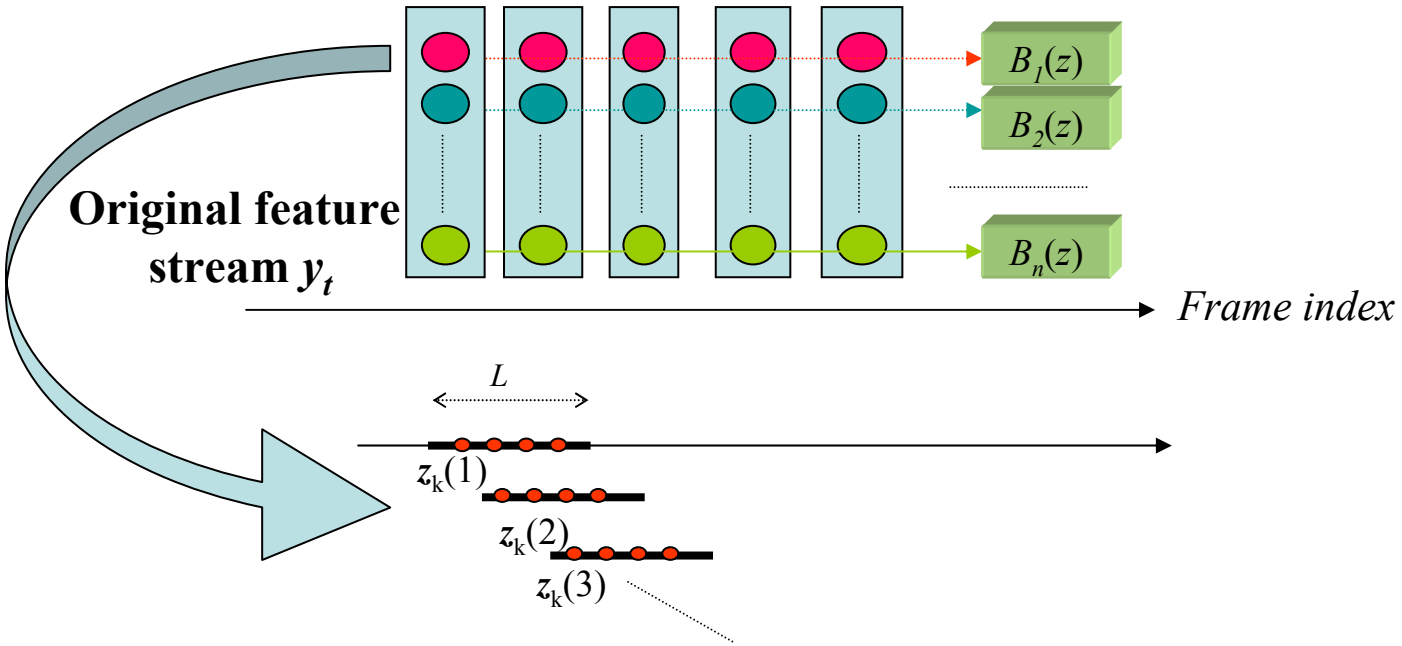




Modulation Frequency (H_z)

Features-based Approach Example 3 — Data-driven Temporal Filtering (1)

- **PCA-derived temporal filtering**
  - temporal filtering is equivalent to the weighted sum of a sequence of a specific MFCC coefficient with length L slided along the frame index
  - maximizing the variance of such a weighted sum is helpful in recognition
  - the impulse response of $B_k(z)$ can be the first eigenvector of the covariance matrix for z_k, for example
  - $B_k(z)$ is different for different k



Linear Discriminative Analysis (LDA)

- **Linear Discriminative Analysis (LDA)**
  - while PCA tries to find some "principal components" to maximize the variance of the data, the Linear Discriminative Analysis (LDA) tries to find the most "discriminative" dimensions of the data among classes
- **Problem Definition**
  - w_j, μ_j and $\mathbf{U}_j$ are the weight (or number of samples), mean and covariance for the random vectors of class j, $j=1 \ldots\ldots N$, μ is the total mean

    $\text{within - class scatter matrix}: \mathbf{S}_W = \Sigma_{j=1}^{N} w_j U_j$

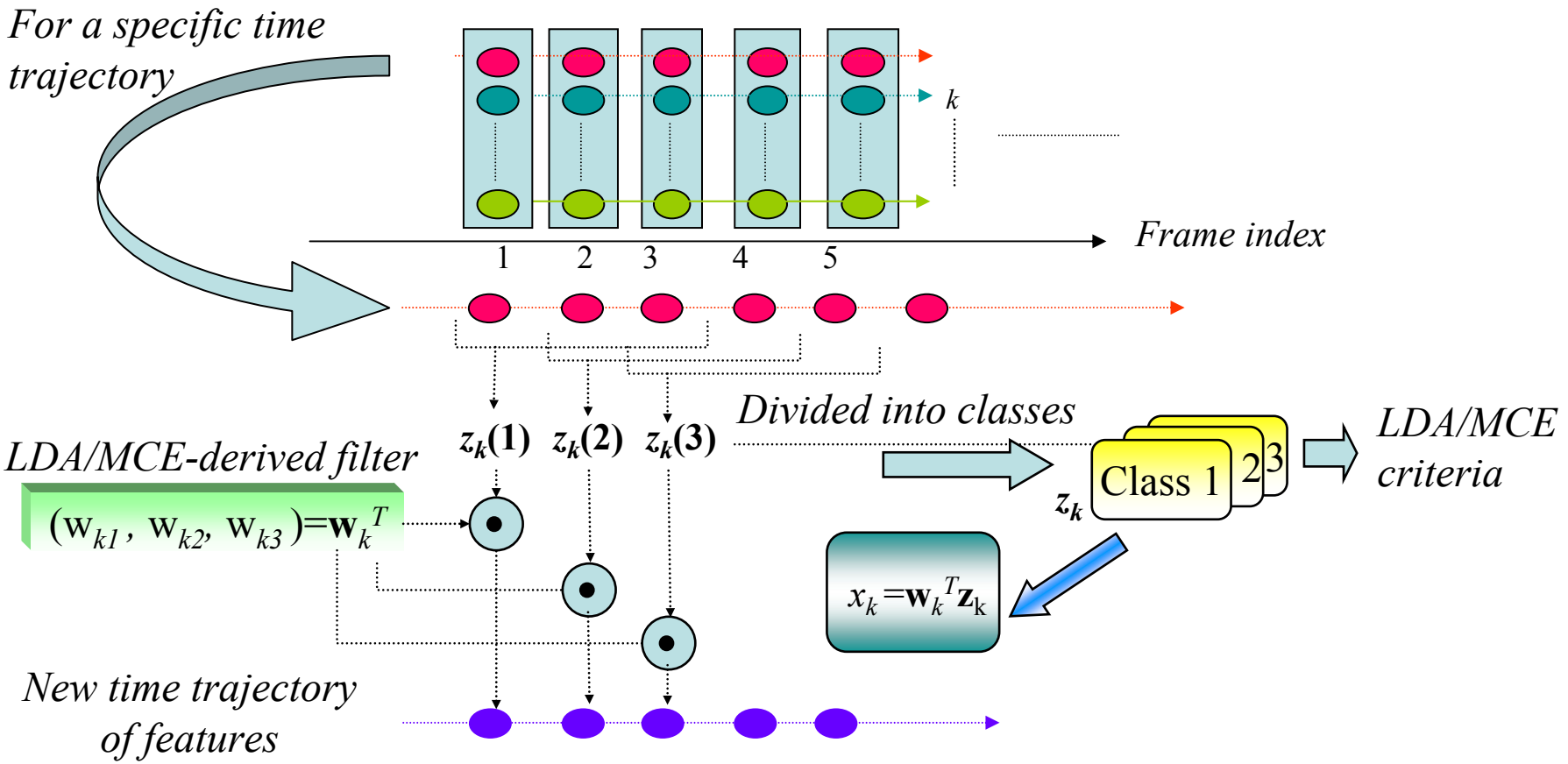
    $\text{between - class scatter matrix}: \mathbf{S}_B = \Sigma_{j=1}^{N} w_j (\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T$
  - find $\mathbf{W}=[\boldsymbol{w}_1 \boldsymbol{w}_2 \ldots\ldots \boldsymbol{w}_k]$, a set of orthonormal basis such that

    $$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \frac{tr\left(\mathbf{W}^T \mathbf{S}_B \mathbf{W}\right)}{tr\left(\mathbf{W}^T \mathbf{S}_w \mathbf{W}\right)}$$
  - tr(M): trace of a matrix M, the sum of eigenvalues, or the "total scattering"

    $\mathbf{W}^T\mathbf{S}_{B,W}\mathbf{W}$: the matrix $\mathbf{S}_{B,W}$ after projecting on the new dimensions
- **Solution**
  - the columns of $\mathbf{W}$ are the eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_B$ with the largest eigenvalues

Features-based Approach Example 3 — Data-driven Temporal Filtering (2)

- **LDA/MCE-derived Temporal Filtering**

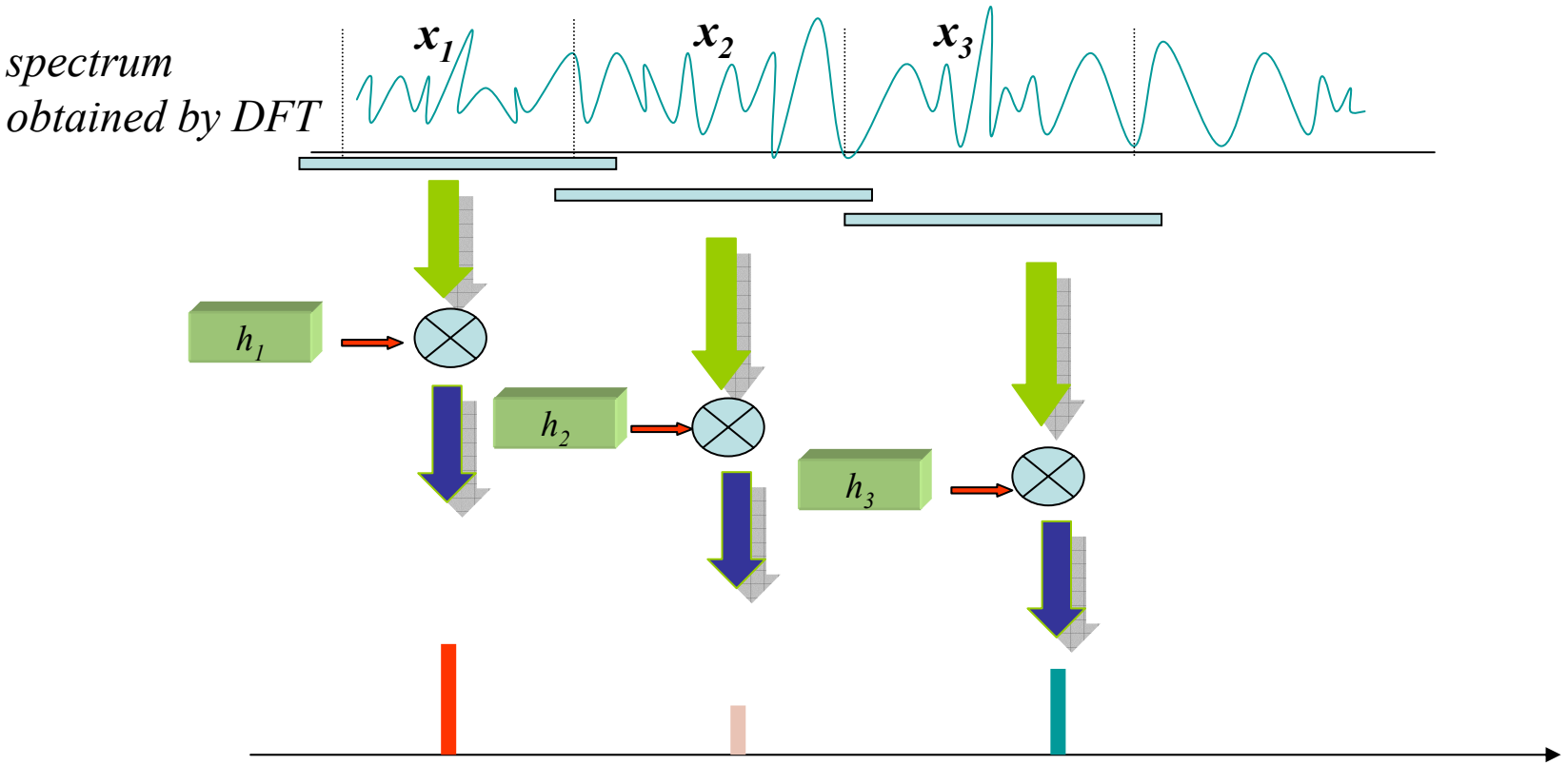


  - Filtered parameters are weighted sum of parameters along the time trajectory (or inner product)

Features-based Approach Example 4 — Data-driven Filter-banks

- **Data-driven Filter Bank**
  - the triangular shapes for the filler bank is not necessarily optimized
  - for PCA

    h_k can be the first eigenvector of the covariance matrix for x_k with the largest eigenvalue
  - MCE approach have been used



Speech Enhancement Example — Spectral Subtraction (SS)

- **Speech Enhancement**
  - producing a better signal by trying to remove the noise
  - for listening purposes or recognition purposes
- **Background**
  - assuming speech signal x[n] and noise n[n] are statistically independent

    $y[n] = x[n] + n[n]$
  - for power spectral densities in frequency domain

    $E(|Y(w)|^2) \approx E(|X(w)|^2)+E(|N(w)|^2)$, or $E(|X(w)|^2)=E(|Y(w)|^2)-E(|N(w)|^2)$
- **Spectrum Subtraction**
  - $|N(w)|$ estimated by averaging over M frames of locally detected silence parts, or up-dated by the latest detected silence frame

    $|N(w)|_i = \beta |N(w)|_{i-1}+(1- \beta) |N(w)|_{i,n}$

    $|N(w)|_i$: $|N(w)|$ used at frame i

    $|N(w)|_{i,n}$: latest detected at frame i
  - signal amplitude estimation

    $|\hat{X}(w)|_i = |Y(w)|_i - |N(w)|_i$, if $|Y(w)|_i - |N(w)|_i > \alpha |Y(w)|_i$

    $= \alpha |Y(w)|_i$ if $|Y(w)|_i - |N(w)|_i \leq \alpha |Y(w)|_i$

    transformed back to $\hat{x}[n]$ using the original phase

    performed frame by frame
  - useful for most cases, but may produce some “musical noise” as well
  - many different improved versions