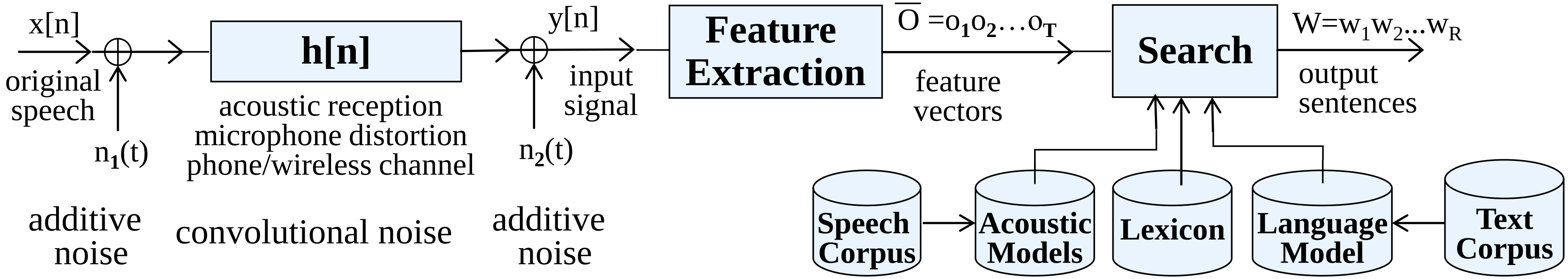
# 15.0 Robustness for Acoustic Environment

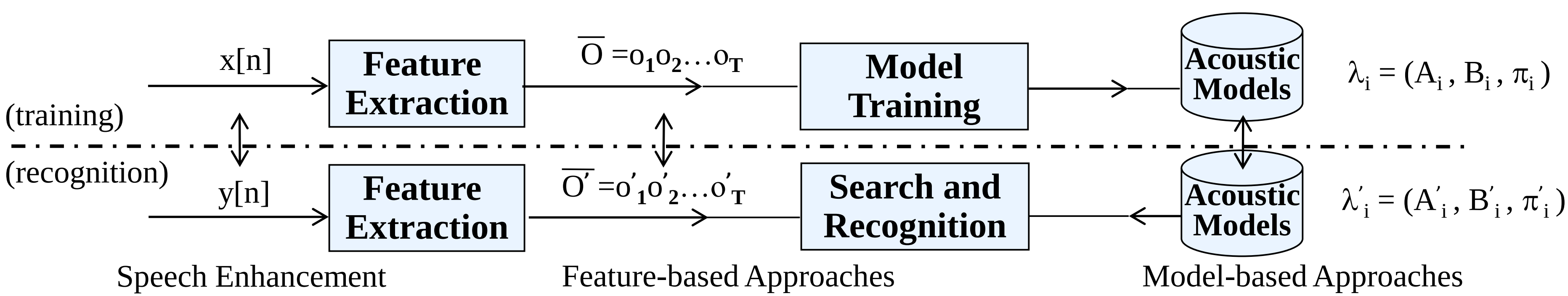
**References**:


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. "Cepstral Domain Segmental Feature Vector Normalization for Noise Robust Speech Recognition", Speech Communication, Vol. 25, pp. 133-147, August 1998
6. "Non-linear Transformation of the Feature Space for Robust Speech Recognition", in Proceedings of International Conference on Acoustics, Speech and Signal Processing, 2002, pp. 401-404
7. "RASTA Processing of Speech", IEEE Trans. on Speech & Audio Processing, April 1994
8. 3.8 of Duda, Hart and Stork, "Pattern Classification", John Wiley and sons, 2001
9. "Optimization of Temporal Filters for Constructing Robust Features in Speech Recognition", IEEE Trans. on Speech and Audio Processing, May 2006
10. "Suppression of Acoustic Noise in Speech Using Spectral Subtraction" ,IEEE Trans. on Acoustics, Speech and Signal Processing, Apr 1979
11. "A Perceptually Constrained GSVD-based Approach for Enhancing Speech Corrupted by Color Noise", IEEE Transactions on Audio, Speech and Language Processing, Jan 2007

# Mismatch in Statistical Speech Recognition

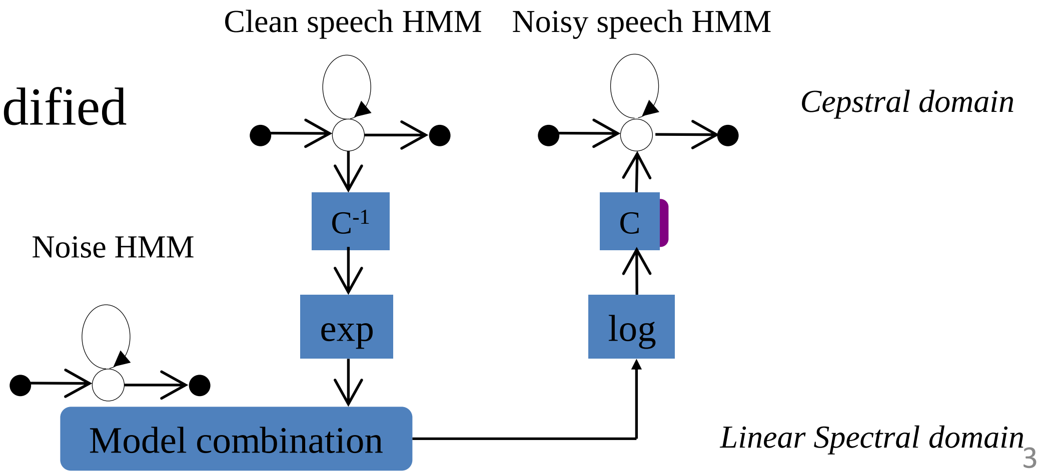



- **Mismatch between Training/Recognition Conditions**
  - Mismatch in Acoustic Environment — Environmental Robustness
    - additive/convolutional noise, etc.
  - Mismatch in Speaker Characteristics — Speaker Adaptation
  - Mismatch in Other Acoustic Conditions
    - speaking mode:read/prepared/conversational/spontaneous 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**

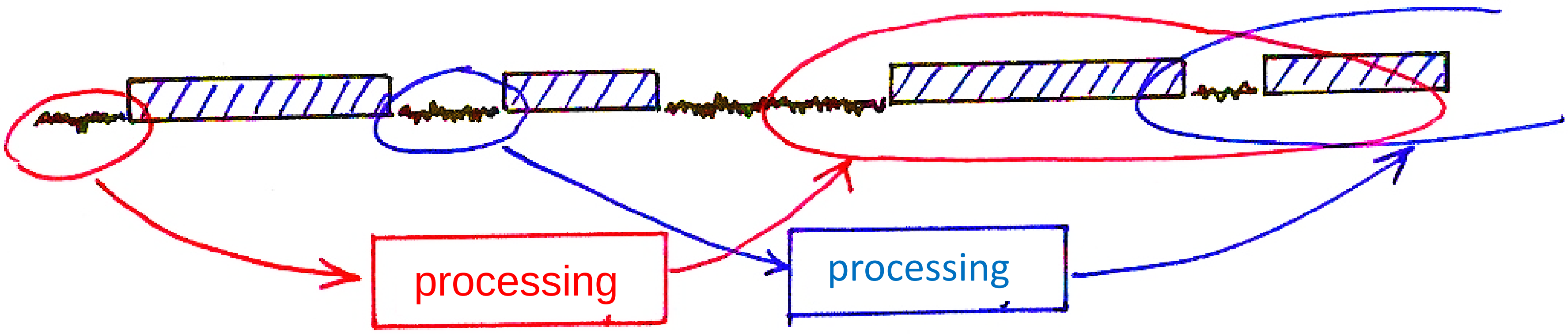


# Model-based Approach Example 1— Parallel Model Combination (PMC)

- **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



# Parallel Model Combination (PMC)



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

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

*Linear power spectral domain*

$$X(\omega)=S(\omega)+N(\omega)$$

*log* ↓ ↑ *exp*

$X^l=\log(X)$,
$S^l=\log(S) \rightarrow S=\exp(S^l)$
$N^l=\log(N) \rightarrow N=\exp(N^l)$

*Log spectral domain*

$$X^l=\log\left(\exp\left(S^l\right)+\exp\left(N^l\right)\right)$$

**C** ↓ ↑ $\mathbf{C}^{-1}$

$X^c=\mathbf{C}X^l$,
$S^c=\mathbf{C}S^l \rightarrow S^l=\mathbf{C}^{-1}S^c$
$N^c=\mathbf{C}N^l \rightarrow N^l=\mathbf{C}^{-1}N^c$

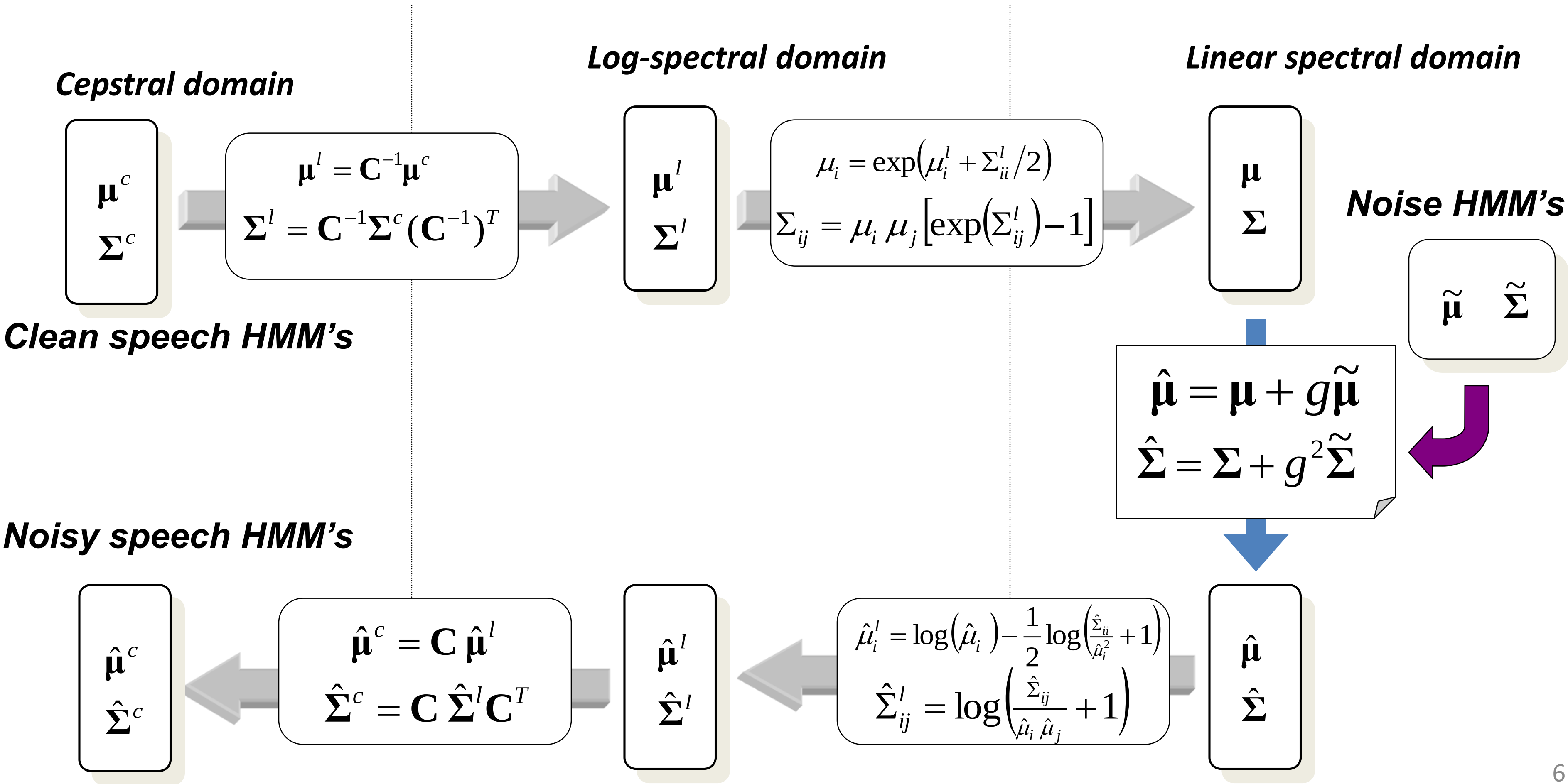
*Cepstral domain*

$$X^c=\mathbf{C}\log\left(\exp\left(\mathbf{C}^{-1}S^c\right)+\exp\left(\mathbf{C}^{-1}N^c\right)\right)$$

***Nonlinear combination***

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

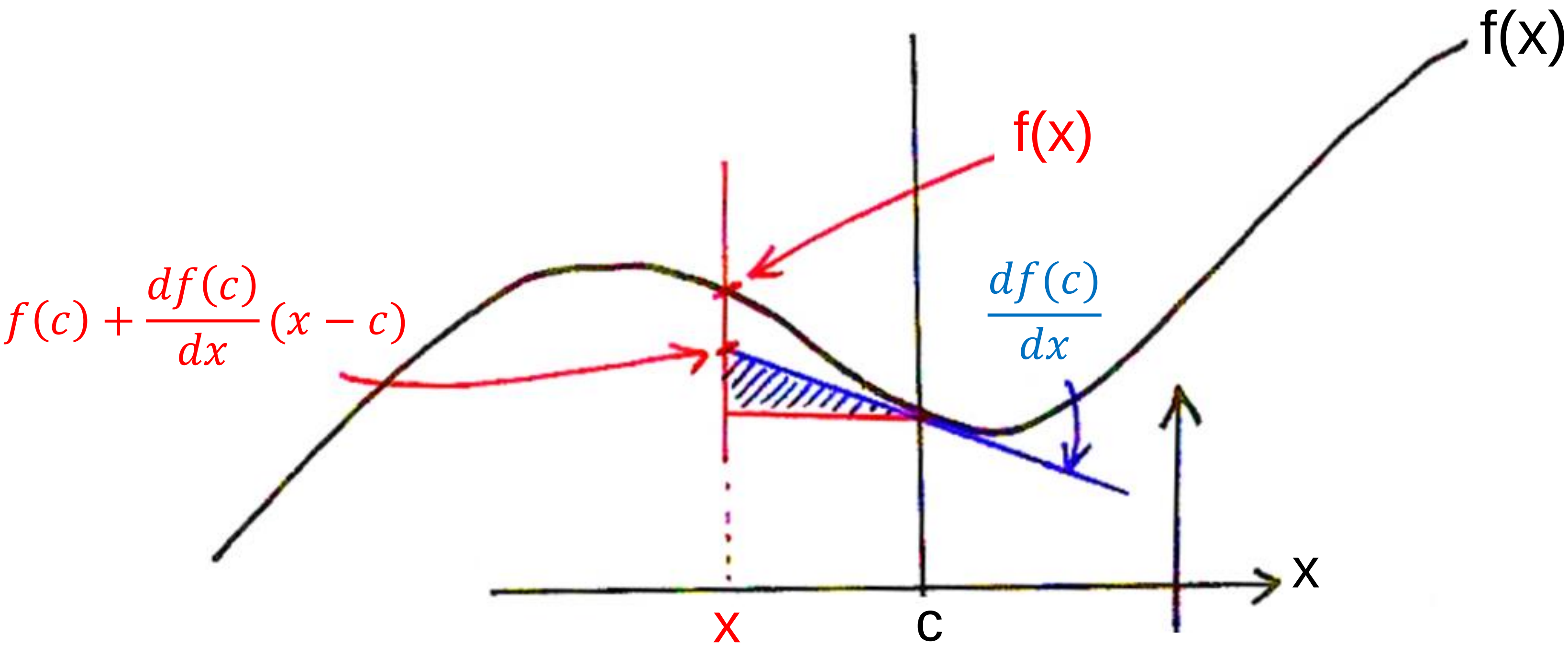
- **The Steps of Parallel Model Combination (Log-Normal Approximation) :**
  - based on various assumptions and approximations to simplify the mathematics and reduce the computation requirements



# Model-based Approach Example 2— Vector Taylor's Series (VTS)

- **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:

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



# Vector Taylor's Series (VTS)

- **Given a nonlinear function z=*g*(x, y)**
  - x, y, z are n-dim random vectors
  - assuming the mean of x, y, $\mu_x$, $\mu_y$ and covariance of x, y, $\Sigma_x$, $\Sigma_y$ are known
  - then the mean and covariance of z can be approximated by the Vector Taylor's Series

$$\mu_z^{\,i} = g(\mu_x^{\,i}, \mu_y^{\,i}) + \frac{1}{2}\left(\frac{\partial^2 g(\mu_x^{\,i}, \mu_y^{\,i})}{\partial x^{i2}}\Sigma_x^{\,ii} + \frac{\partial^2 g(\mu_x^{\,i}, \mu_y^{\,i})}{\partial y^{i2}}\Sigma_y^{\,ii}\right)$$

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

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

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

  - 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 Moment Normalization (CMS, CMVN) and Histogram Equalization (HEQ)

- **Cepstral Mean Subtraction(CMS) - Originally for Convolutional 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 a moving window, or a longer time interval

    $\bar{x}_{\mathbf{CMS}} = \bar{y} - E[\bar{y}]$
  - CMS features are immune to convolutional noise

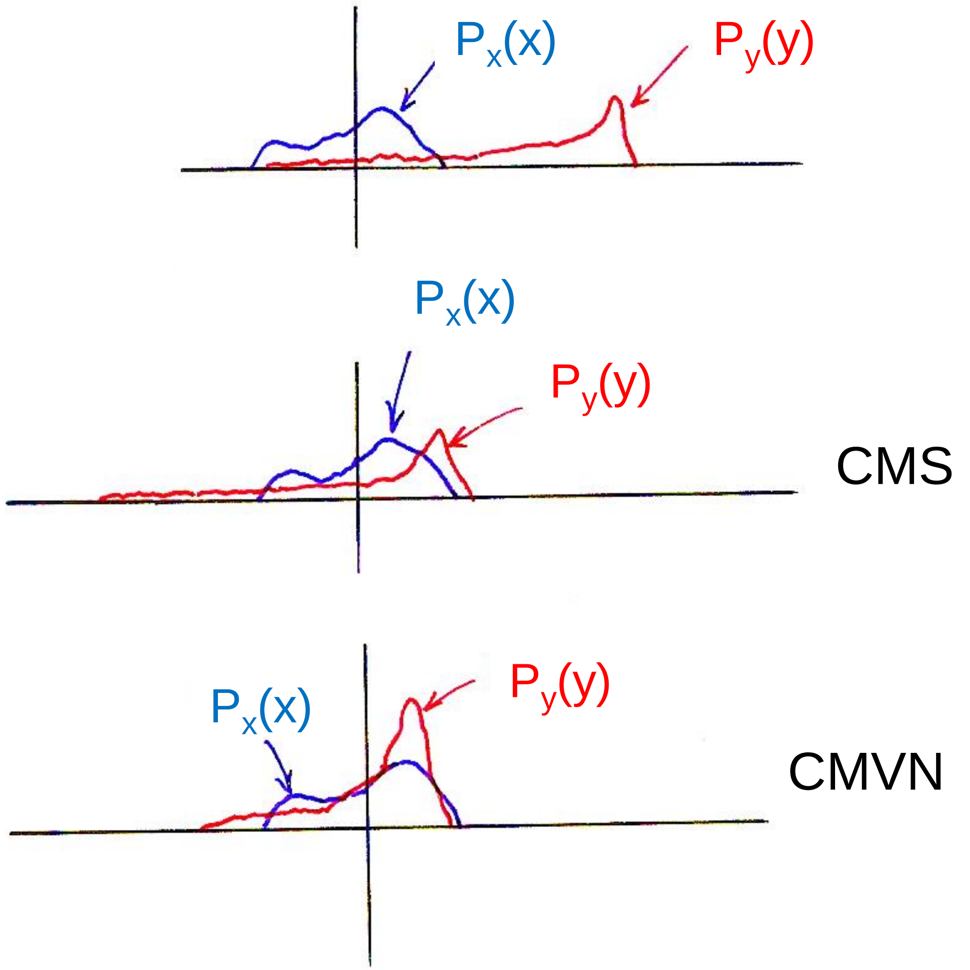
    x[n] convolved with any h[n] gives the same $\bar{x}_{\mathbf{CMS}}$
  - CMS doesn't change delta or delta-delta cepstral coefficients
- **Signal Bias Removal**
  - estimating h by the maximum likelihood criteria

    $\bar{h}^* = \underset{\bar{h}}{\arg\max}\ \mathrm{Prob}[\bar{Y} = (\bar{y}_1 \bar{y}_2 \ldots \bar{y}_T) \mid \lambda, \bar{h}]$ , $\lambda$ : HMM for the utterance $\bar{Y}$
  - iteratively obtained via EM algorithm
- **CMS, Cepstral Mean and Variance Normalization (CMVN) and Histogram Equalization (HEQ)**
  - CMS equally useful for additive noise
  - CMVN: variance normalized as well $x_{CMVN} = x_{CMS} / [\mathrm{Var}(x_{CMS})]^{1/2}$
  - HEQ: the whole distribution equalized $y = CDF_y^{-1}[CDF_x(x)]$
  - Successful and popularly used

# Cepstral Moment Normalization

- **CMVN: variance normalized as well**

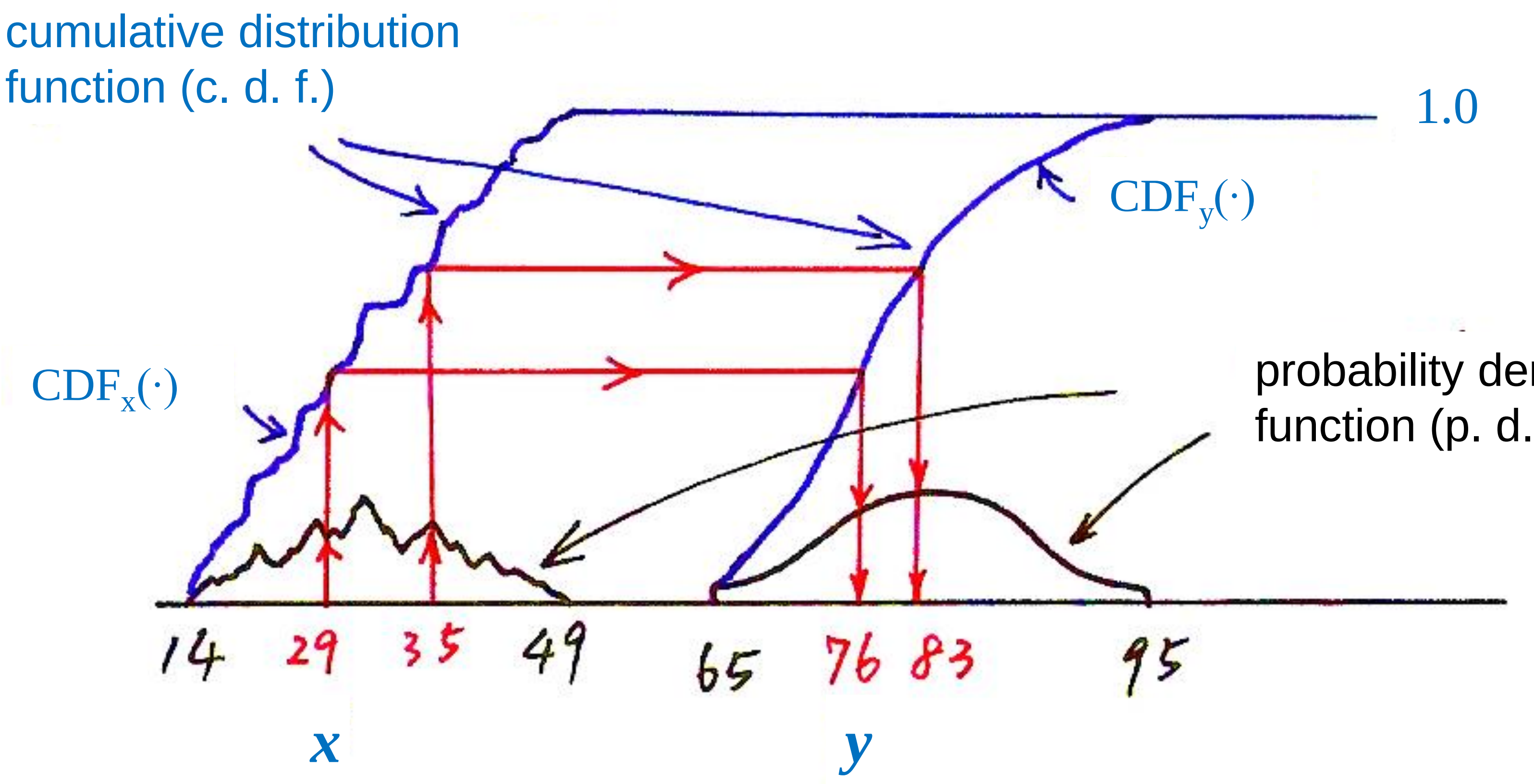
$$x_{CMVN} = x_{CMS}/[Var(x_{CMS})]^{1/2}$$



# Histogram Equalization

- **HEQ: the whole distribution equalized**

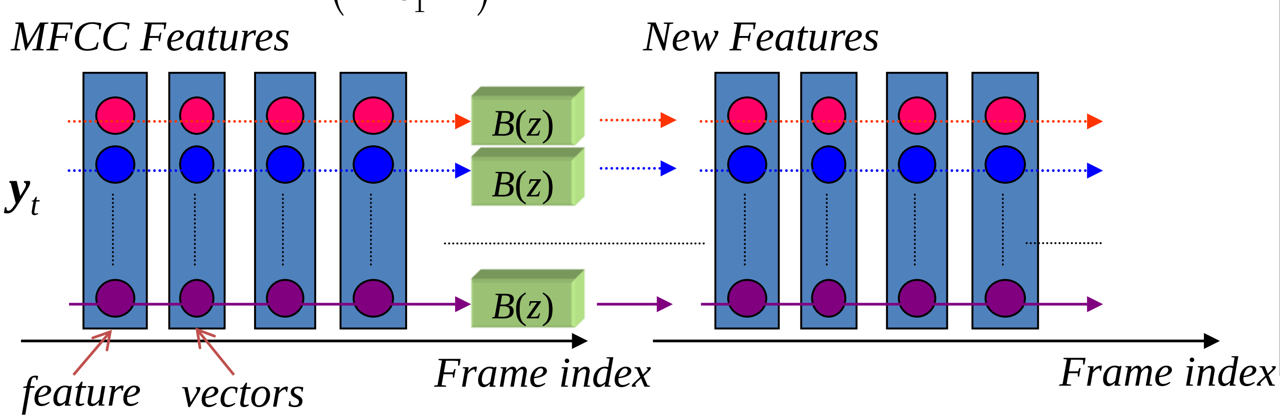
$$y = CDF_y^{-1}[CDF_x(x)]$$

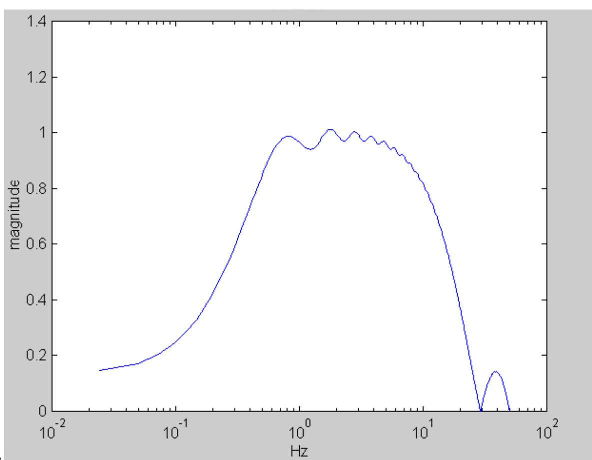


# Feature-based Approach Example 2 — RASTA (Relative Spectral) Temporal Filtering

- **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"

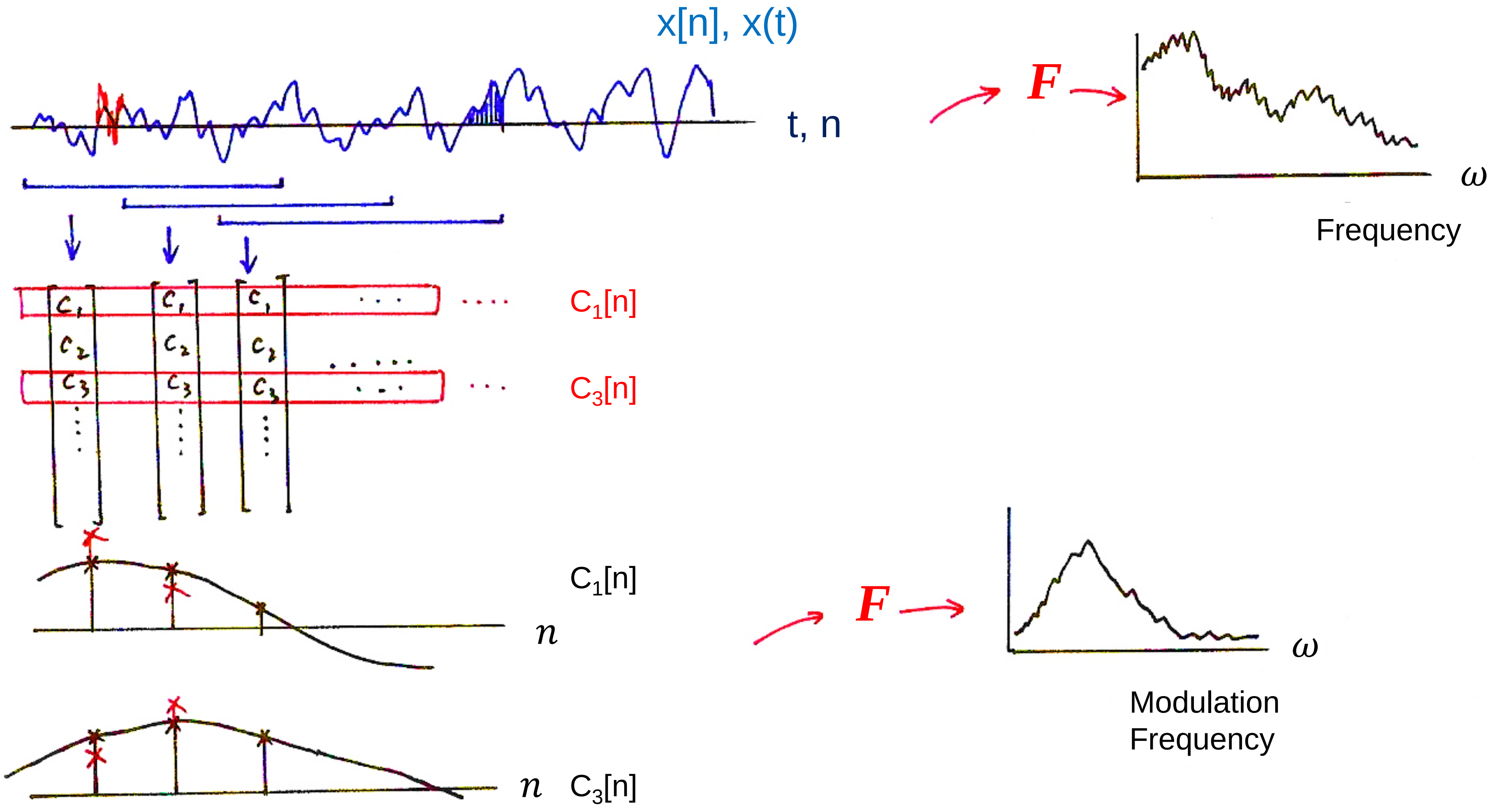
$$B(z) = \frac{a_0 + a_1 z^{-1} + a_3 z^{-3} + a_4 z^{-4}}{\left(1 - b_1 z^{-1}\right) z^{-4}}$$



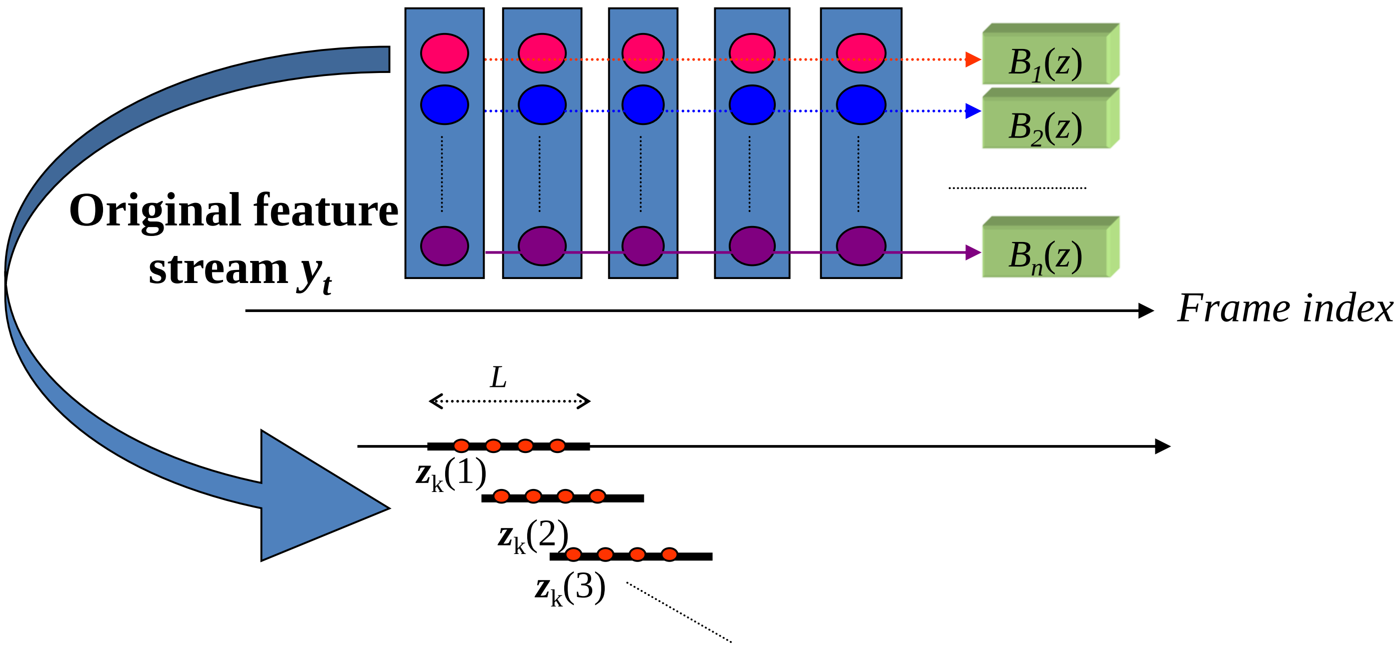

*Modulation Frequency (Hz)*

# Temporal Filtering

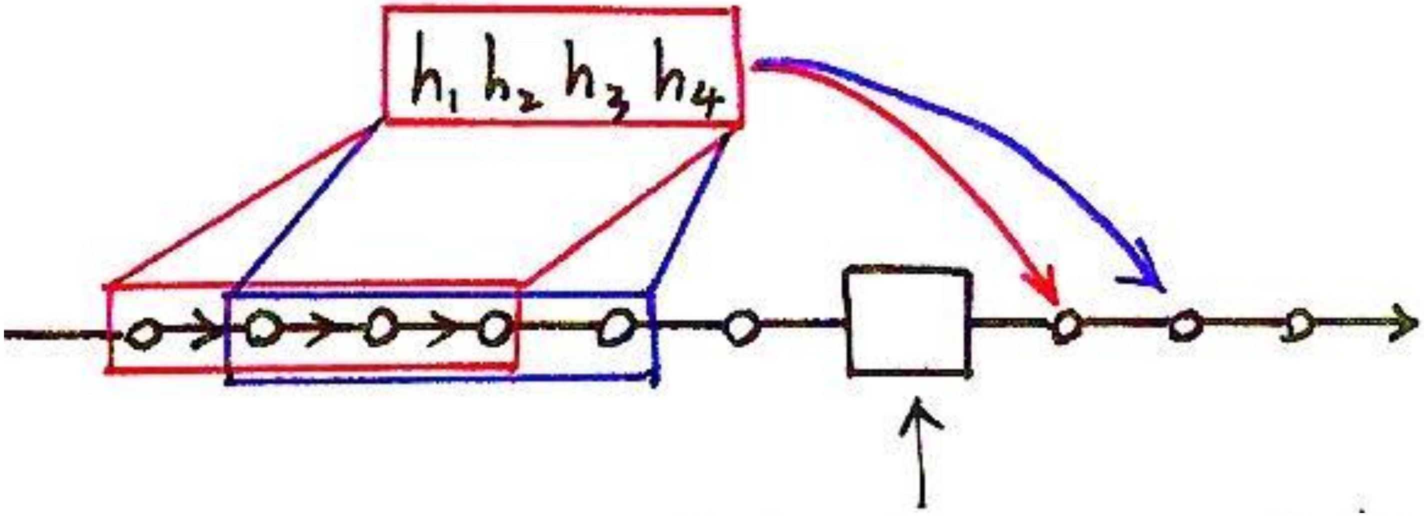


# Features-based Approach Example 3 — Data-driven Temporal Filtering (1)

- **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



# Filtering




filtering: convolution

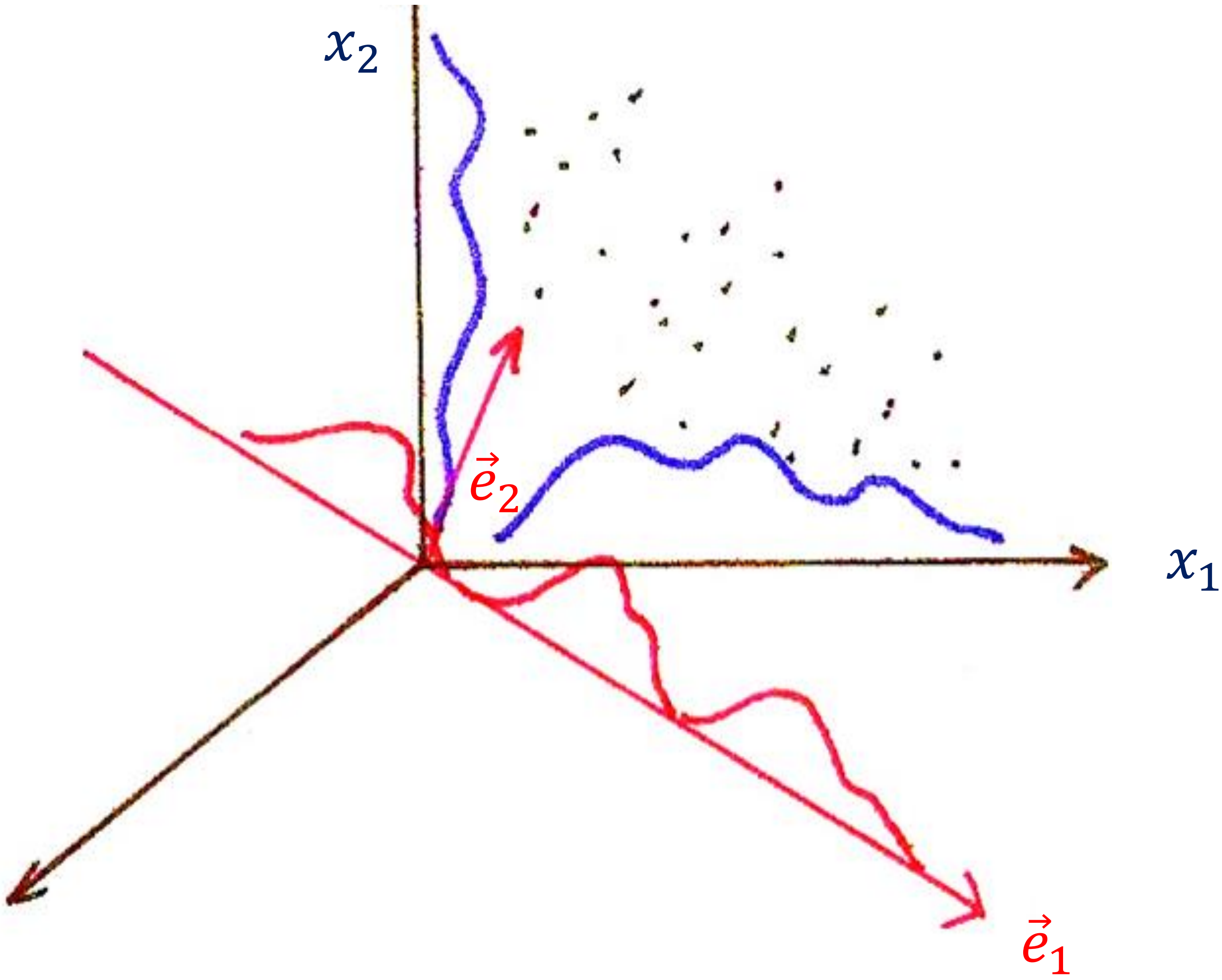
# PCA (P.13 of 13.0)

$$\begin{bmatrix} \cdots & e_{1k}^T & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ x_k \\ \vdots \end{bmatrix} = \vec{e}_1 \cdot \vec{x}$$

$$= \underbrace{|\vec{e}_1|}_{\substack{\parallel \\ 1}} |\vec{x}| \cos\theta$$

$$\vec{y} = A^T \vec{x} = \begin{bmatrix} \overrightarrow{e}_1^T \\ \overrightarrow{e}_2^T \\ \vdots \\ \overrightarrow{e}_k^T \end{bmatrix} \vec{x}$$

# PCA (P.12 of 13.0)



# Principal Component Analysis (PCA) (P.11 of 13.0)

- **Problem Definition:**
  - for a zero mean random vector $\boldsymbol{x}$ with dimensionality $\boldsymbol{N}$, $x \in R^N$, $E(x)=0$, iteratively find a set of $k$ ($k \le N$) orthonormal basis vectors $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_k\}$ so that
    (1) *var* $(\boldsymbol{e}_1^T \boldsymbol{x})$=*max* *(x has maximum variance when projected on* $e_1$ *)*
    (2) *var* $(\boldsymbol{e}_i^T \boldsymbol{x})$=*max*, subject to $\boldsymbol{e}_i \perp \boldsymbol{e}_{i-1} \perp \ldots \perp \boldsymbol{e}_1$ , $2 \le i \le k$
    *(x has next maximum variance when projected on* $e_2$ *, etc.)*
- **Solution: $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_k\}$ are the eigenvectors of the covariance matrix Σ for *x* corresponding to the largest *k* eigenvalues**
  - new random vector $\boldsymbol{y} \in \boldsymbol{R}^k$ : the projection of $x$ onto the subspace spanned by $\boldsymbol{A}=[\boldsymbol{e}_1\ \boldsymbol{e}_2\ \ldots\ \boldsymbol{e}_k]$, $\boldsymbol{y}=\boldsymbol{A}^T\boldsymbol{x}$
  - a subspace with dimensionality $k \le N$ such that when projected onto this subspace, y is "closest" to $x$ in terms of its "randomness" for a given k
  - var $(e_i^T x)$ is the eigenvalue associated with $e_i$
- **Proof**
  - var $(e_1^T x) = e_1^T E(x x^T) e_1 = e_1^T \Sigma e_1$ = max, subject to $|e_1|^2=1$
  - using Lagrange multiplier

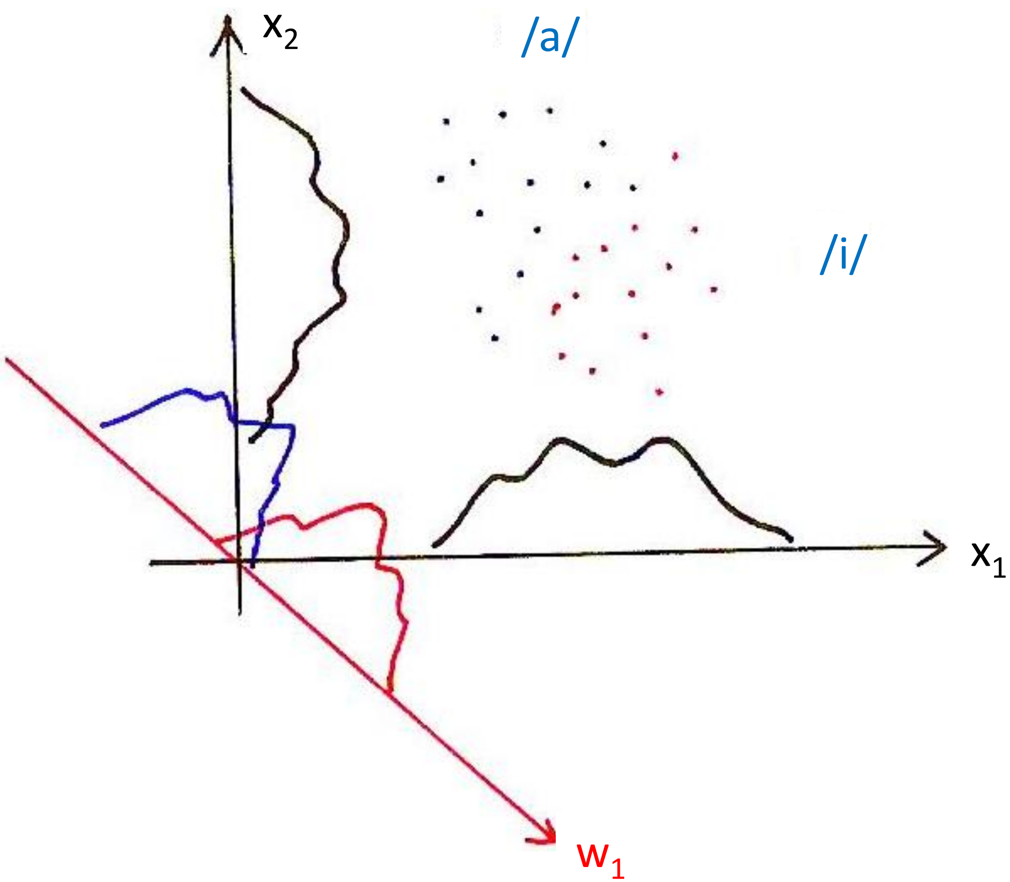
$$J(e_1)= e_1^T E(x x^T)e_1 - \lambda(|e_1|^2-1)\ , \quad \frac{\partial J(e_1)}{\partial e_1} = 0$$

$$\Rightarrow E(xx^T)\, e_1 = \lambda_1 e_1\ ,\ var(e_1^T x) = \lambda_1 = \max$$

  - similar for $e_2$ with an extra constraint $e_2^T e_1 = 0$, etc.

# Linear Discriminative Analysis (LDA)

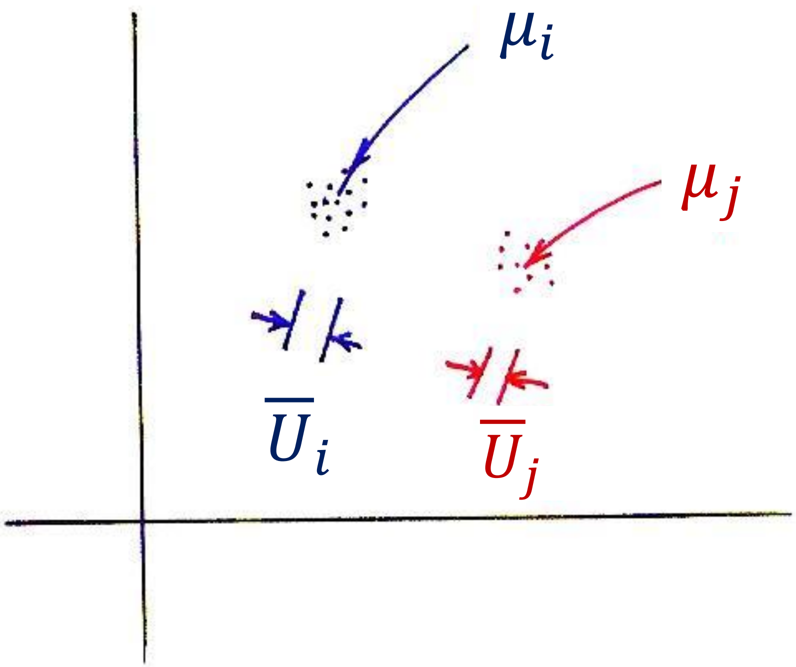
- **Linear Discriminative Analysis (LDA)**
  - while PCA tries to find some "principal components" to maximize the variance of the data, the Linear Discriminative Analysis (LDA) tries to find the most "discriminative" dimensions of the data among classes



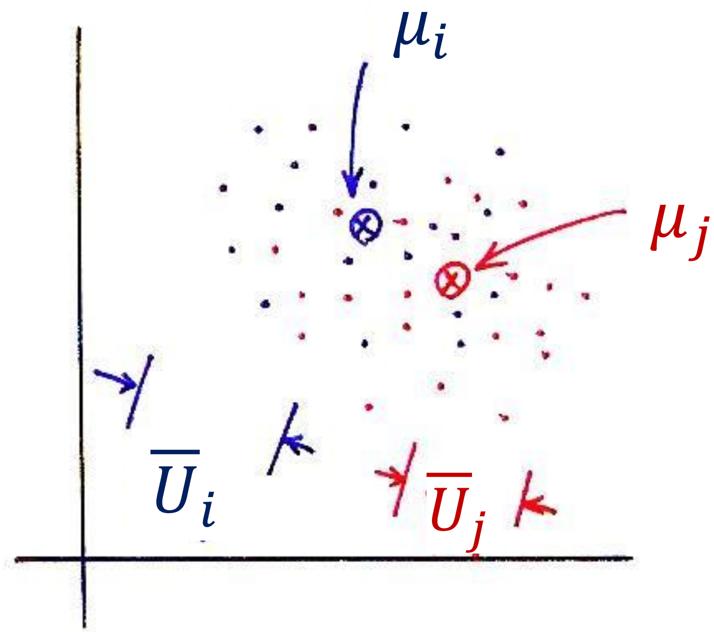
# Linear Discriminative Analysis (LDA)

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

desired

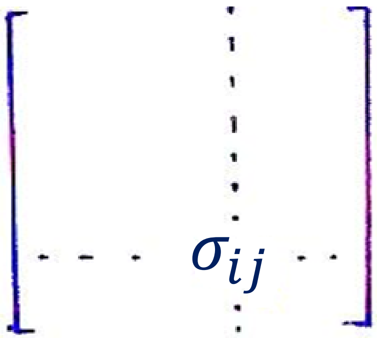



undesired



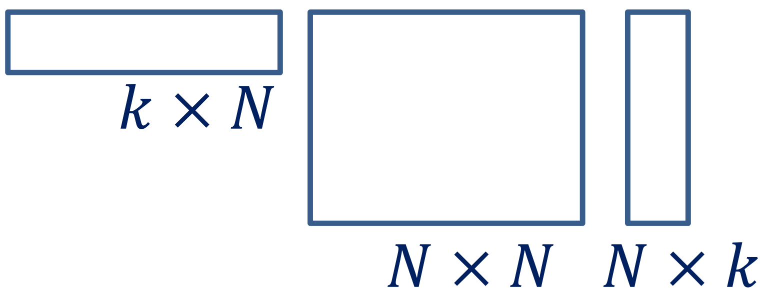
# Linear Discriminative Analysis (LDA)

$\overline{U} =$




$$\sigma_{ij} = E[(x_i - \overline{x}_i)(x_j - \overline{x}_j)]$$

$W^T S W =$



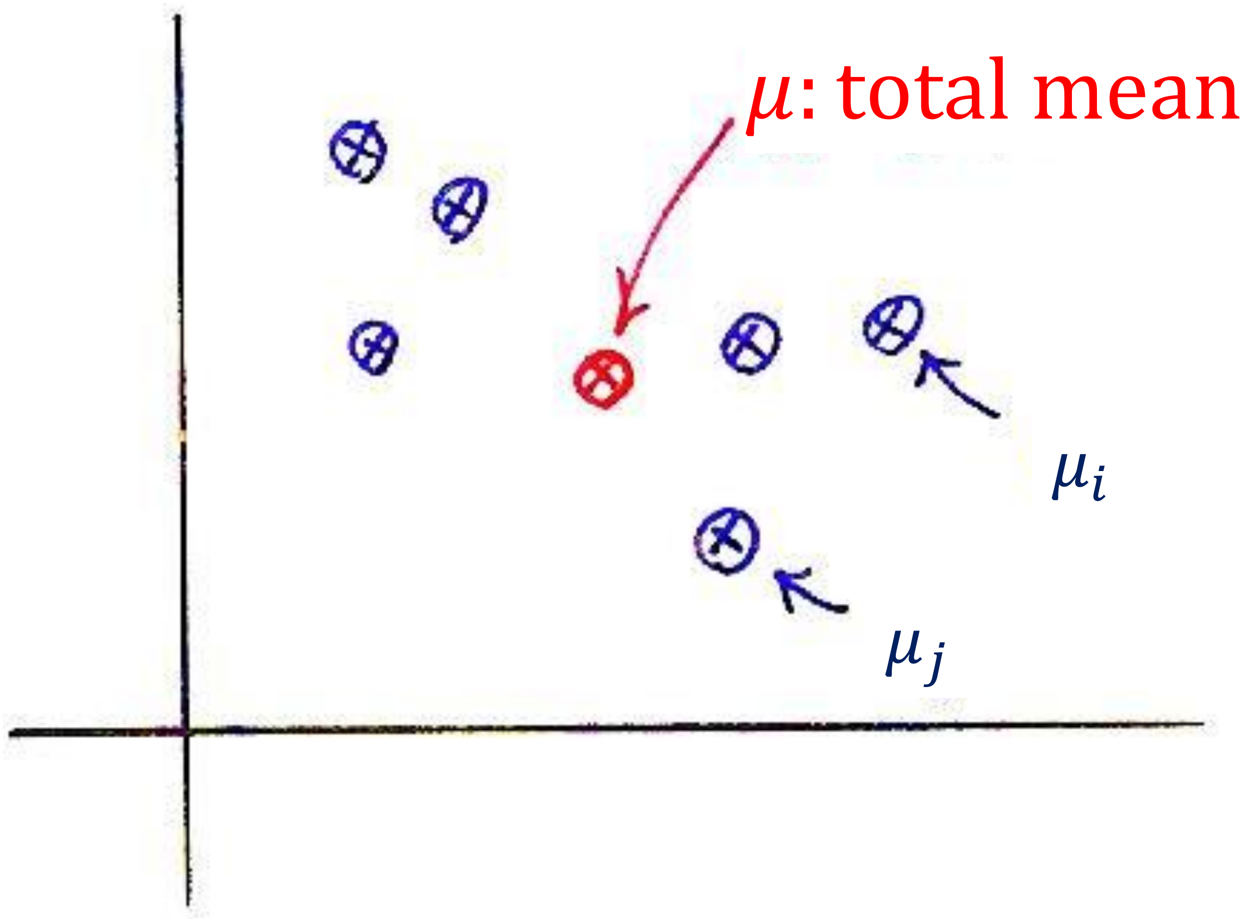

$$\frac{tr(\mathbf{S}_B)}{tr(\mathbf{S}_w)} = \max$$

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \frac{tr(\mathbf{W}^T \mathbf{S}_B \mathbf{W})}{tr(\mathbf{W}^T \mathbf{S}_w \mathbf{W})}$$

- **tr(M): trace of a matrix M, the sum of eigenvalues, or the “total scattering”**
- **$W^T S_{B,W} W$: the matrix $S_{B,W}$ after projecting on the new dimensions**

# Linear Discriminative Analysis (LDA)

- **Between-class scatter matrix:** $\mathbf{S}_B = \sum_{j=1}^{N} w_j(\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T$



# Linear Discriminative Analysis (LDA)

- **Problem Definition**
  - $\boldsymbol{w}_j$, $\boldsymbol{\mu}_j$ and $\mathbf{U}_j$ are the weight (or number of samples), mean and covariance for the random vectors of class *j*, *j=1……N,* μ is the total mean

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

    $$\text{between-class scatter matrix}: \mathbf{S}_B = \sum_{j=1}^{N} w_j (\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T$$

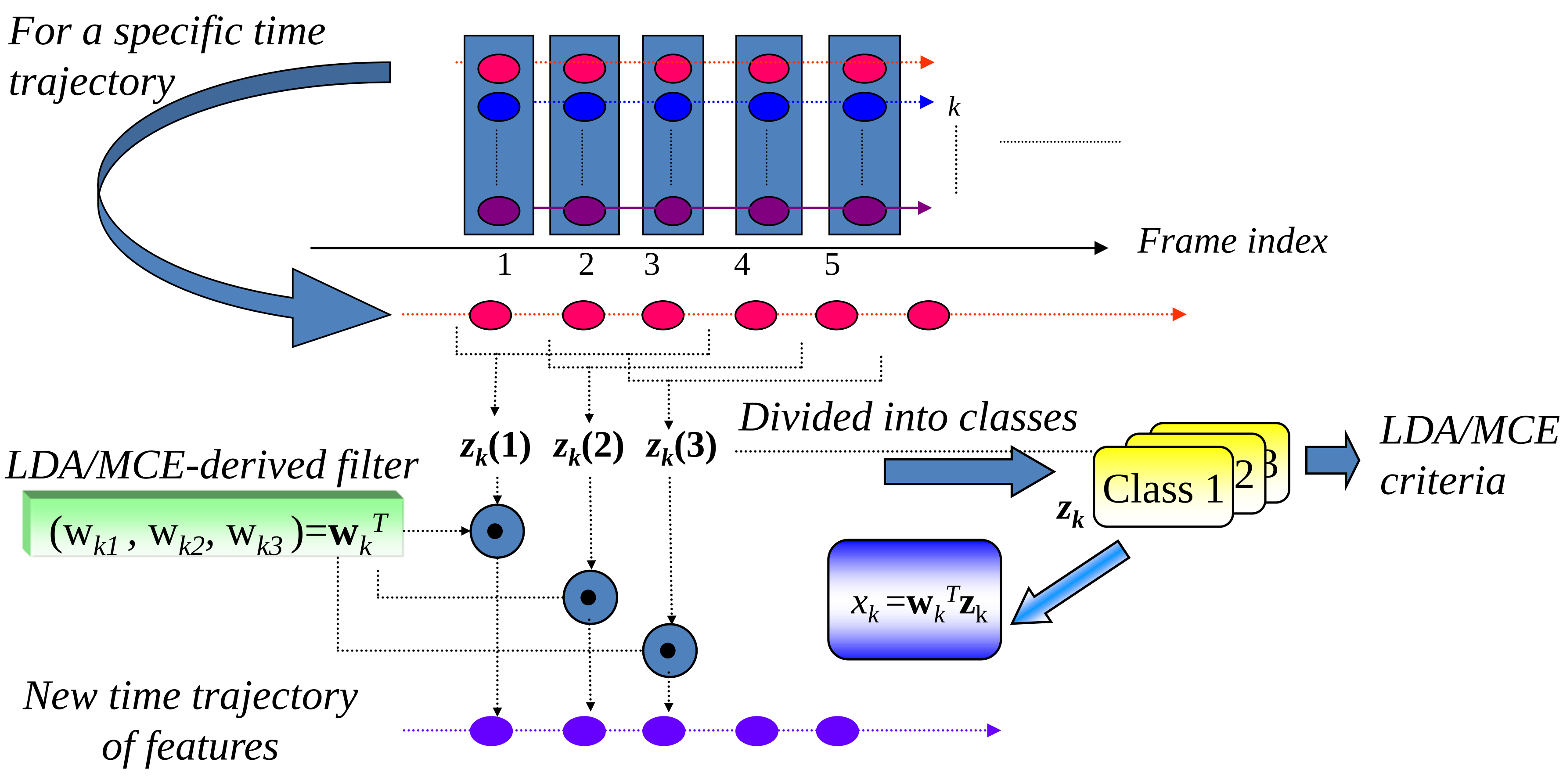
  - find $\mathbf{W}=[\boldsymbol{w}_1\ \boldsymbol{w}_2 \ldots\ldots \boldsymbol{w}_k]$, a set of orthonormal basis such that

    $$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \frac{tr(\mathbf{W}^T \mathbf{S}_B \mathbf{W})}{tr(\mathbf{W}^T \mathbf{S}_w \mathbf{W})}$$

  - tr(M): trace of a matrix M, the sum of eigenvalues, or the “total scattering” $W^T S_{B,W} W$: the matrix $S_{B,W}$ after projecting on the new dimensions
- **Solution**
  - the columns of **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)

# Speech Enhancement Example 1 — Spectral Subtraction (SS)

- **Speech Enhancement**
  - producing a better signal by trying to remove the noise
  - for listening purposes or recognition purposes
- **Background**
  - Noise n[n] changes fast and unpredictably in time domain, but relatively slowly in frequency domain, N(w)

    $y[n] = x[n] + n[n]$
- **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

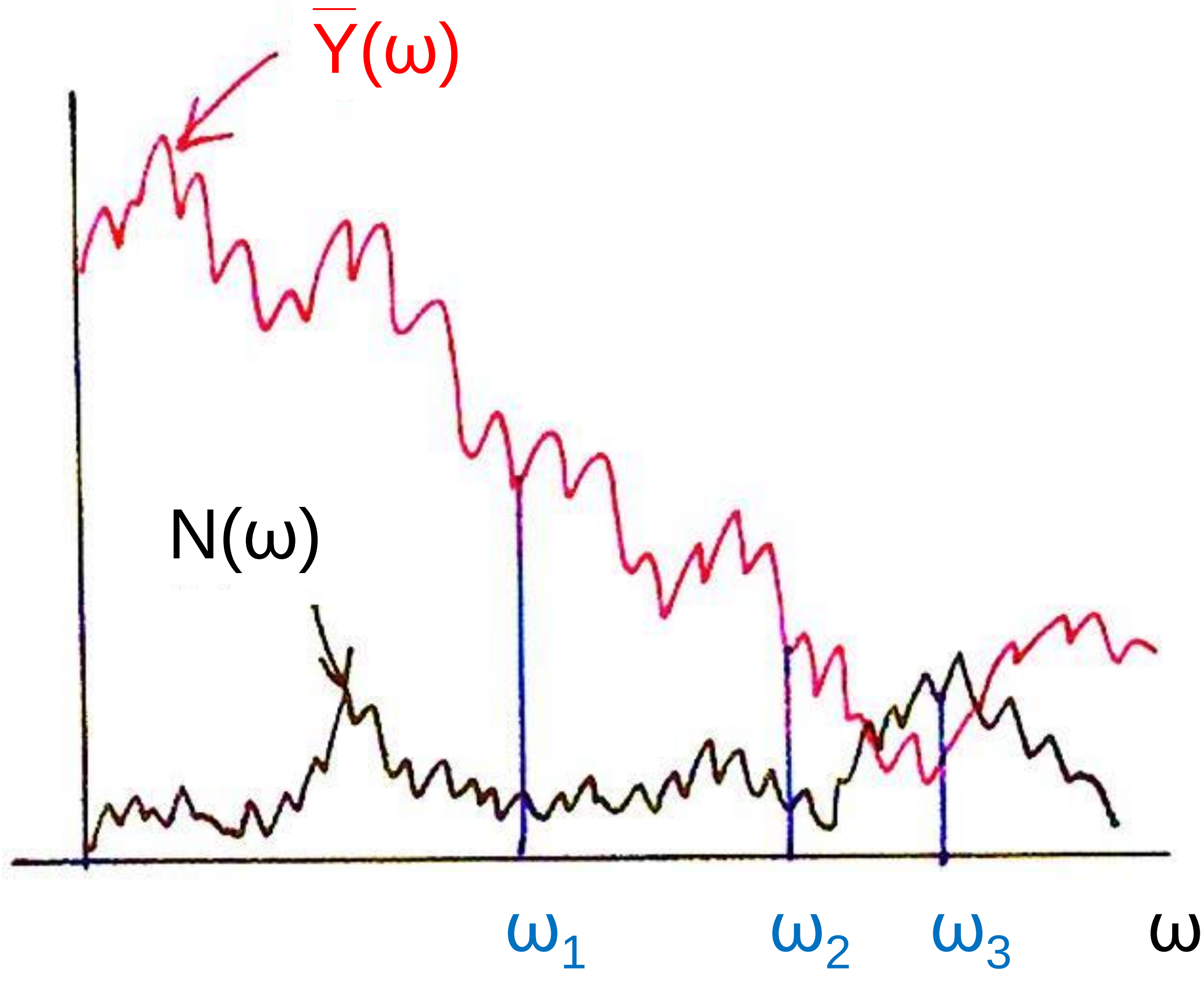
    $|X(w)|_i = |Y(w)|_i - |N(w)|_i$ , if $|Y(w)|_i - |N(w)|_i > \alpha\, |Y(w)|_i$

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

    transformed back to x[n] using the original phase

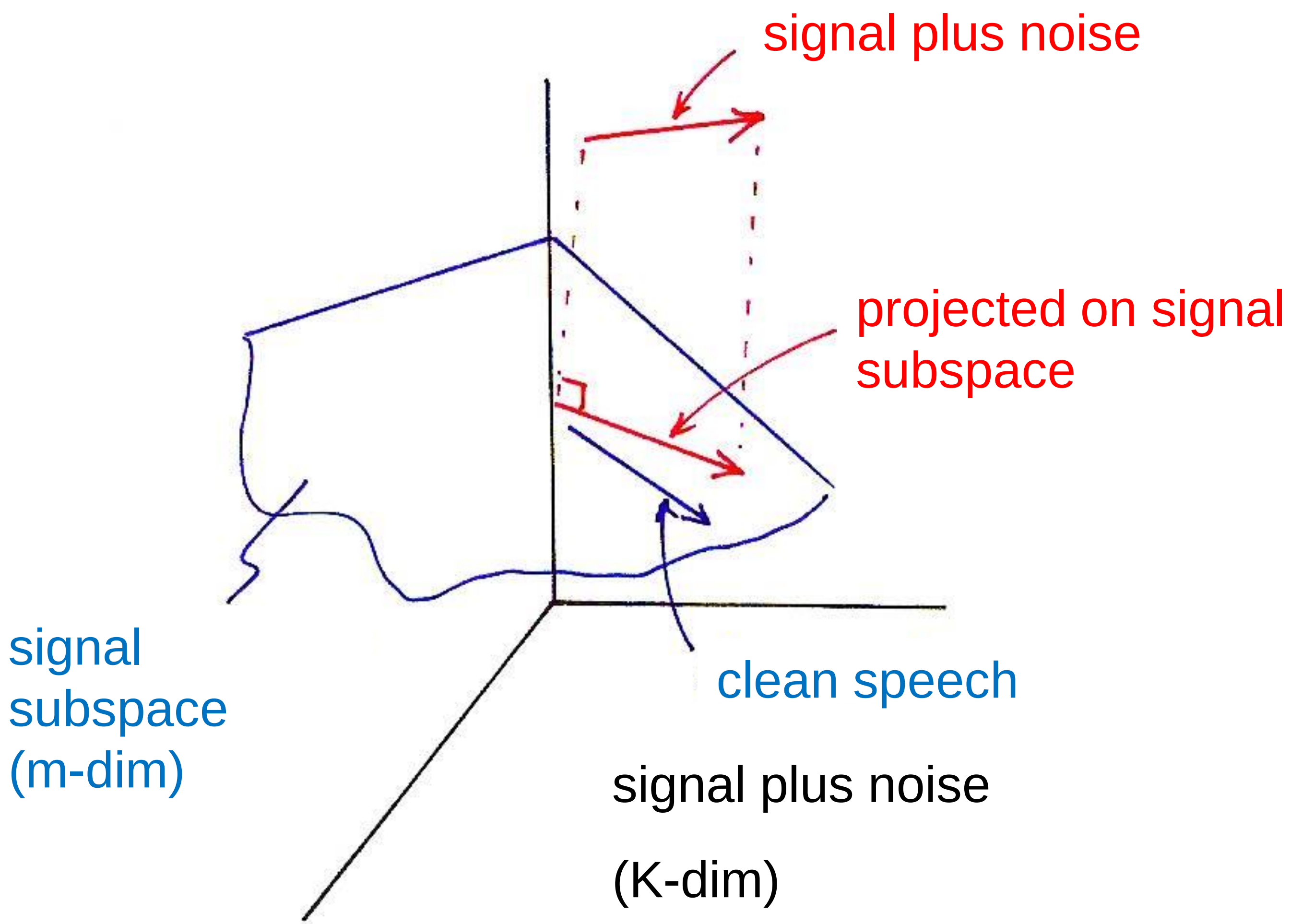
    performed frame by frame
  - useful for most cases,^but may produce some "musical noise" as well
  - many different improved versions

# Spectral Subtraction



# Speech Enhancement Example 2 — Signal Subspace Approach

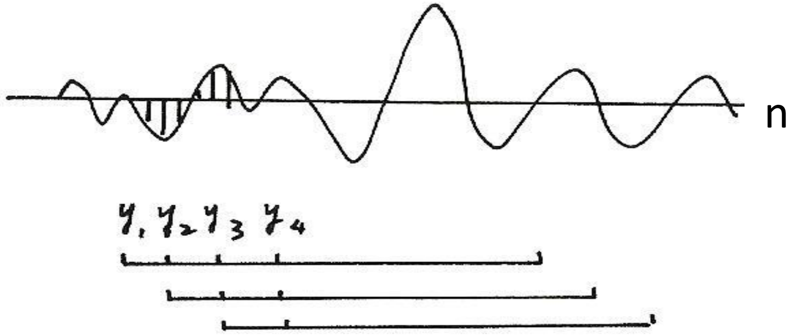
- **Signal Subspace Approach**
  - representing signal plus noise as a vector in a K-dimensional space
  - signals are primarily spanned in a m-dimensional signal subspace
  - the other K-m dimensions are primarily noise
  - projecting the received noisy signal onto the signal subspace



# Speech Enhancement Example 2 — Signal Subspace Approach

- **An Example**
  - Hankel-form matrix

    signal samples: $y_1 y_2 y_3 \ldots y_k \ldots y_L \ldots y_M$




$$H_y = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 \cdots & y_k \\ y_2 & y_3 & y_4 \cdots\cdots & & y_{k+1} \\ y_3 & y_4 \cdots\cdots\cdots & & & y_{k+2} \\ \vdots & \vdots & \vdots & & \vdots \\ y_L & y_{L+1} & & & y_M \end{bmatrix}$$

— $H_y$ for noisy speech

— $H_n$ for noise frames

  - Generalized Singular Value Decomposition (GSVD)

$U^T H_y X = C = \mathrm{diag}(c_1, c_2, \ldots c_k),\ c_1 \geq c_2 \geq \ldots \geq c_k$

$V^T H_n X = S = \mathrm{diag}(s_1, s_2, \ldots s_k),\ s_1 \leq s_2 \leq \ldots \leq s_k$

subject to

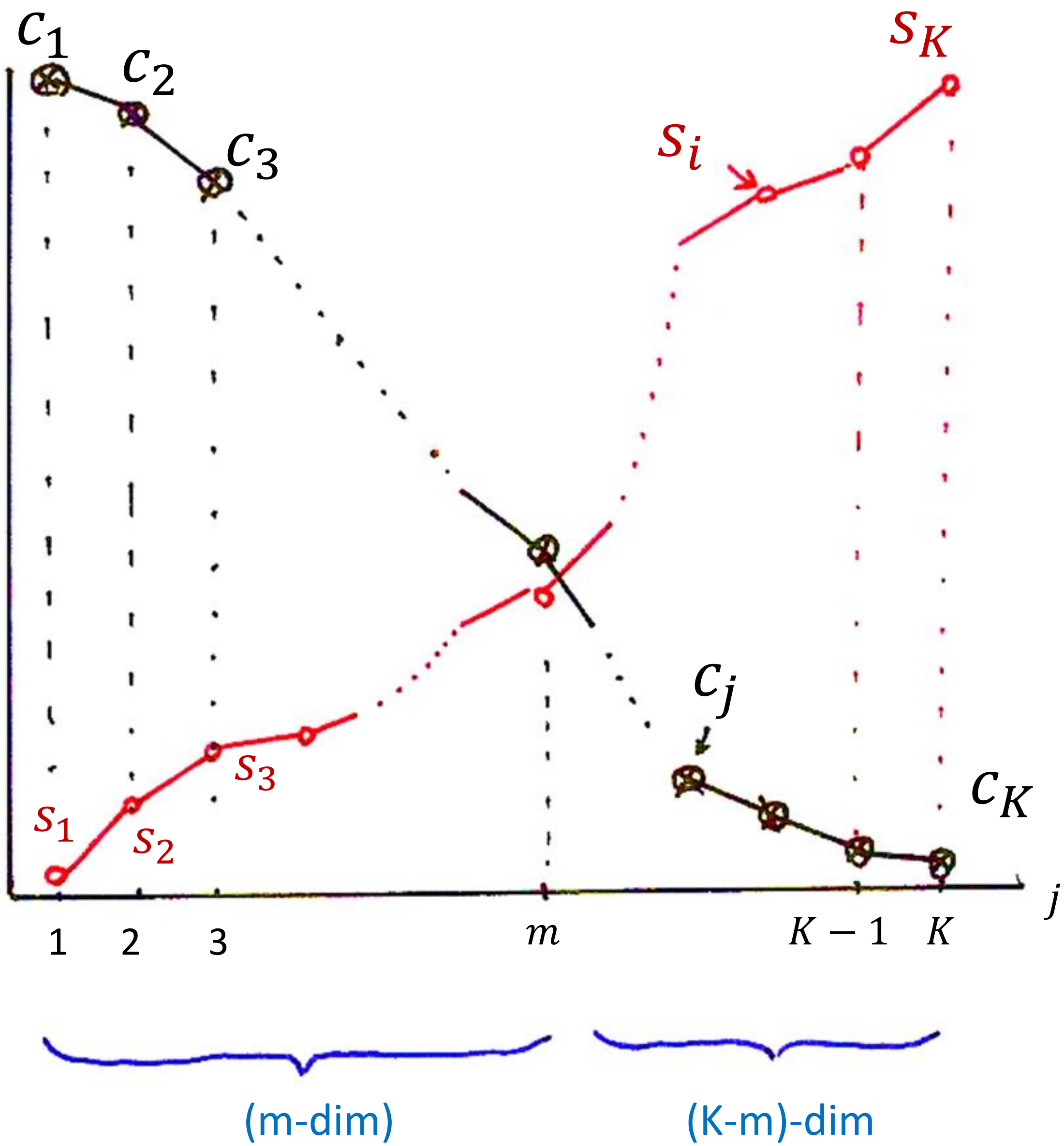
U, V, X : matrices composed by orthogonal vectors

Which gives $c_i > s_i$ for $1 \leq i \leq m$, signal subspace

$s_i > c_i$ for $m+1 \leq i \leq k$, noise subspace

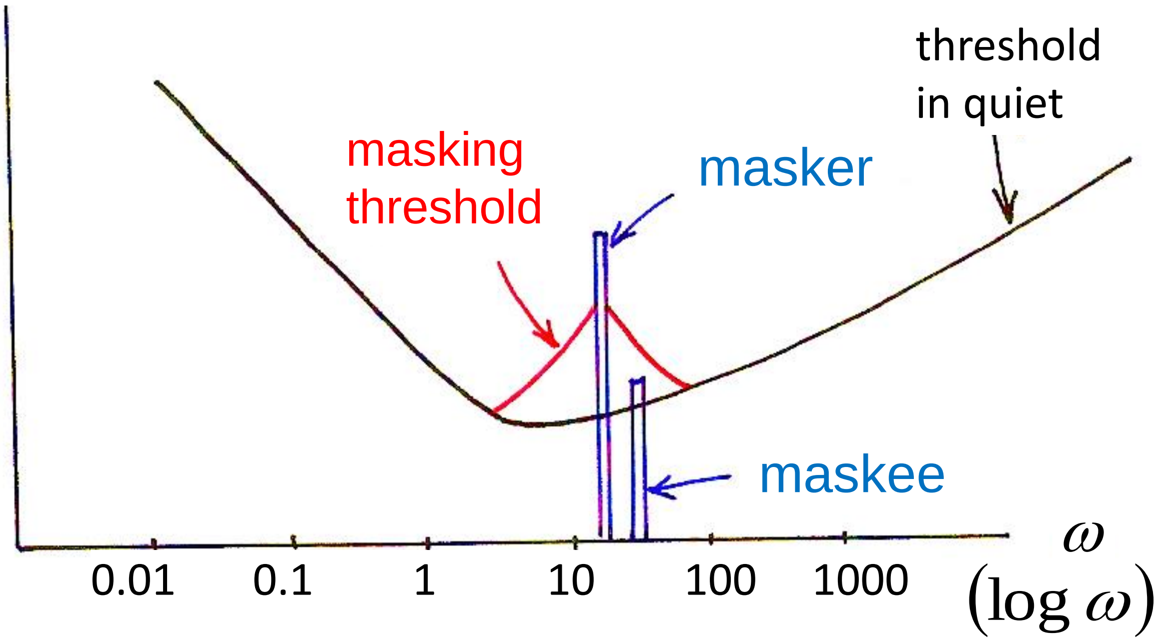
$c_i^2 + s_i^2 = 1,\ 1 \leq i \leq K$

# Signal Subspace



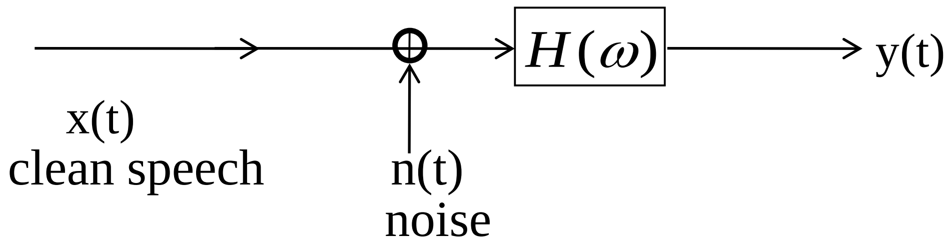
# Speech Enhancement Example 3 — Audio Masking Thresholds

- **Audio Masking Thresholds**
  - without a masker, signal inaudible if below a "threshold in quite"
  - low-level signal (maskee) can be made inaudible by a simultaneously occurring stronger signal (masker).
  - masking threshold can be evaluated
  - global masking thresholds obtainable from many maskers given a frame of speech signals
  - make noise components below the masking thresholds



# Speech Enhancement Example 4 — Wiener Filtering

- **Wiener Filtering**
  - estimating clean speech from noisy speech in the sense of minimum mean square error given statistical characteristics




$$E[(y(t)-x(t))^2]=\min$$

  - an example solution : assuming x(t), n(t) are independent

$$H(\omega)=\frac{S_x(\omega)}{S_x(\omega)+S_n(\omega)}=\frac{1}{1+S_n(\omega)/S_x(\omega)}$$

$$S_x(\omega)=E\{|F[x(t)]|^2\},\ S_n(\omega)=E\{|F[n(t)]|^2\}$$