CHAPTER 5

CHIRPLET DECOMPOSITION

5.1. INTRODUCTION

Of the many possible solutions that exist for analyzing signals in different domains by going beyond time and frequency, the promising ones are those which adapt the tilings in the time-frequency plane. However, much effort has been laid in constructing a proper representation of the signal in time and frequency, e.g., adaptive TFRs, optimized STFT, signal decomposition in terms of fixed set of time-frequency atoms, etc., because of their easy interpretation as well as computation. The successful application of time-frequency analysis methods has stimulated recent interest in considering representations that tile the time-frequency plane in a nonrectangular fashion in which case the existing methods are inappropriate. The shear in frequency and shift in time subspace considered in the earlier chapter is one such analysis methodology. As one should essentially operate on a non-finite vocabulary, estimating the parameters is a difficult task but it gives a better representation. The solution is to choose a dictionary of finite vocabulary and then project the signal onto the space spanned by the dictionary. However, unless we assume some apriori information about the characteristics of the signal under analysis, the residue will be of more energy. Atomic decomposition, chirp hunting are such approaches (Bultan, 1999 and O'Neil *et al*, 1998). It is the purpose of the present work to find a robust mechanism that adapts the tilings to suit the signal's characteristics and eliminate the

difficulties associated with finite size dictionaries. The outline of the algorithm is as follows:

We try to decompose/synthesize a signal as a linear combination of Gaussian chirplets. The computed / specified spectrogram of the signal to be analyzed/synthesized is modeled as a mixture of normal densities. This modeling of spectrogram as a mixture of normal densities is accomplished using the incremental variant of the Expectize-Maximize (EM) algorithm. K-means clustering algorithm is used to pass initial estimates to EM algorithm and the realizations corresponding to the normalized spectrogram are generated using band-rejection algorithm. After the t-f plane is well modeled, we derive a set of mapping rules that synthesize the signal corresponding to the component.

5.2. CHIRPLETS

It is known that the Fourier transform represents a signal as a linear combination of the weighed complex exponentials (waves). Similarly, wavelet transform expands a signal in terms of the scaled and shifted versions of the mother wavelet. In simple terms, a wavelet is an amplitude modulated version of the wave with a rapidly decaying envelope and when it satisfies certain criteria (like the admissibility condition, etc.) we call it a mother wavelet. The waves localize the events in frequency and wavelets offer varying time-frequency resolutions as discussed in Chapter 2. For signals of fractal in nature, these representations perform better than Fourier transform-based representations. However, the tilings are not adaptive and hence they also perform prominently only for certain classes of signals. In an attempt to motivate ourselves to tile the time-frequency plane in a nonrectangular fashion and generalize all existing classes, we have considered chirps

which act as shear operator in frequency (convolving with a chirp in time domain causes shearing in time direction) in the earlier chapter. As wavelets are to waves, chirplets are to chirps (shown in Fig. 5.1). Different tilings obtained by choosing different atoms are depicted in Fig. 5.2.

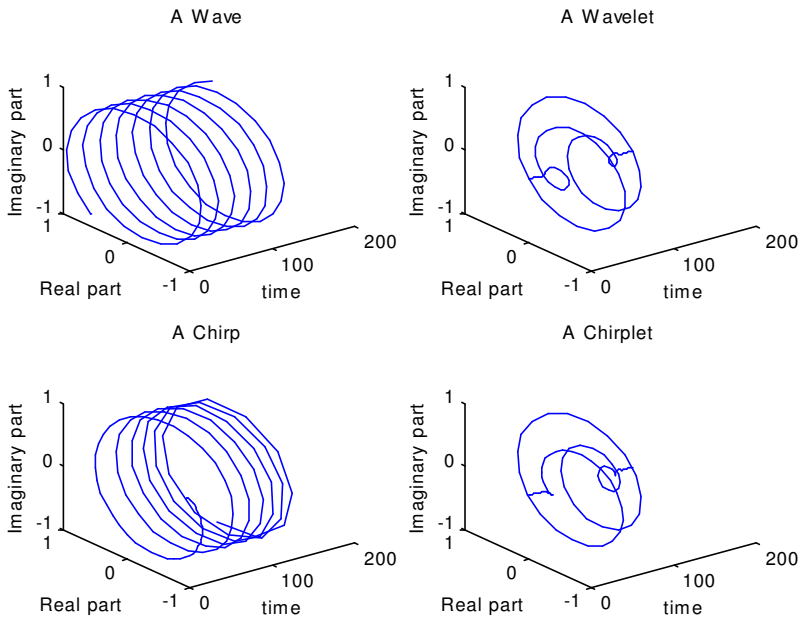
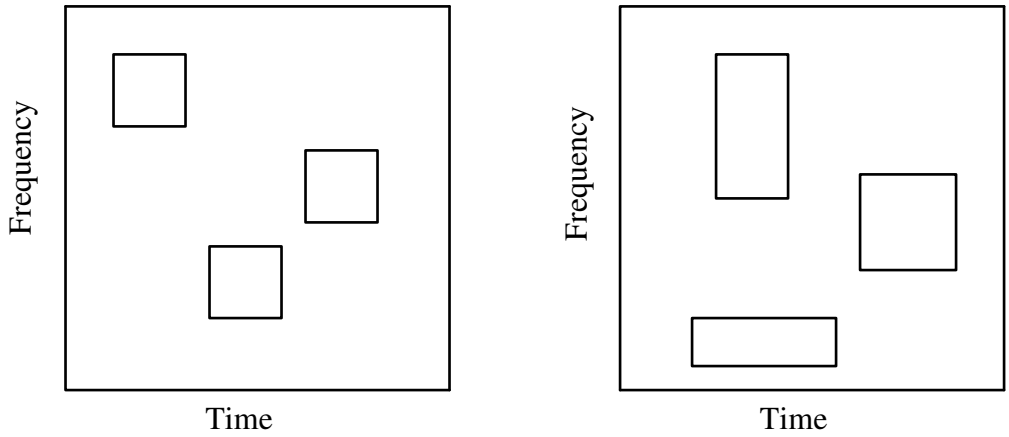
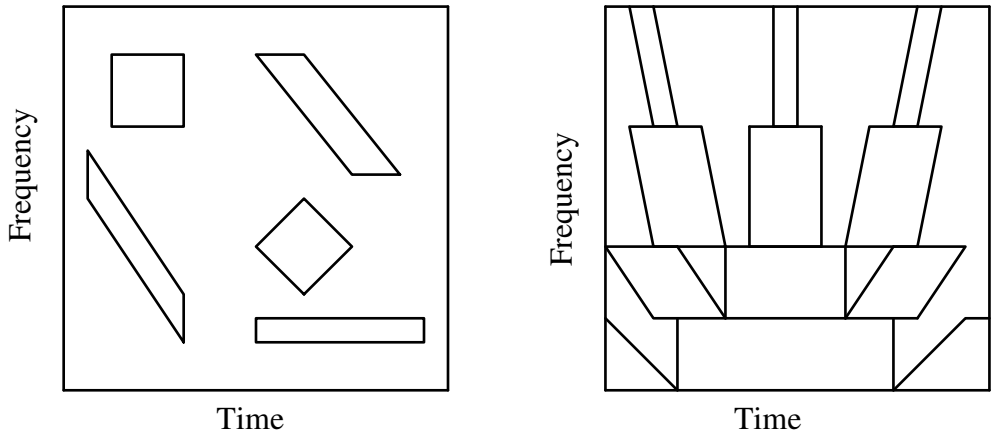


Fig. 5.1 Different atoms considered for signal analysis





(c) **(d)**

Fig. 5.2. Time-frequency tilings obtained by (a) STFT, (b) Wavelet analysis, (c) Chirptlet decomposition and (d) Shear-time representation

It can be observed from Fig. 5.2(a) that employing constant windows in the STFT analysis tile the time-frequency plane in a fixed manner, i.e., rectangular throughout. However, the wavelets being proportional bandwidth representations go on scaling the analysis grid and yet maintain the same area (any arbitrary tiling obtained as a result of modification in representing the scheme has to satisfy the uncertainty principle) shown in Fig. 5.2(b). Employing chirplets to decompose a signal unifies the above tilings as shown in Fig. 5.2(c), i.e., rotation, shearing, scaling and shifting. The tilings obtained by convolving with a chirp are exemplified in Fig. 5.2(d). Hence, we proceed with the following assumption that any signal can be represented as a weighed sum of Gaussian chirplets given by:

$$s(t) = \sum_{i} c_i \, g_i(t; t_c, f_c, \beta, \alpha) \quad , \tag{5.1}$$

where

$$g_{t_c,f_c,\beta,\alpha}(t) = \left(\frac{\alpha}{\pi}\right)^{\frac{1}{4}} e^{\frac{\alpha(t-t_c)^2}{2}} e^{j\beta(t-t_c)^2/2 + j\omega_c(t-t_c)}. \quad (5.2)$$

The parameters t_c, f_c, β and α represent, respectively, the location in time, the location in frequency, the chirp rate and the amplitude modulation parameter; and *g* is defined such that

$$\left\| g_{t_c,f_c,\beta,\alpha} \right\|^2 = \int \left| g(t; t_c, f_c, \beta, \alpha) \right|^2 dt = 1. \quad (5.3)$$

There exists different methodology to construct nice wavelets and so as with chirplets eventhough the latter was not extensively investigated. We have considered the Gaussains for simplicity sake and tractability in evaluation. Moreover, they are the ones which meet the equality sign in uncertainty principle which states that if two functions $f(t)$ and $F(\omega)$ form a Fourier integral pair, then they both cannot be short of duration, i.e., $Dd \geq \frac{1}{2}$ where

$$d^2 = \frac{1}{E}\int t^2 \left| f(t) \right|^2 dt,\ D^2 = \frac{1}{2\pi E}\int \omega^2 \left| F(\omega) \right|^2 d\omega \text{ and } E = \int \left| f(t) \right|^2 dt. \quad (5.4)$$

Now our objective is to find the parameters of these chirplets by modeling the t-f space as a mixture of normal pdfs. It requires interpretation of the spectrogram as an arbitrary bivariate pdf which would be modeled as a mixture of normal pdfs.

5.3. MIXTURE MODELING

Any random signal space can be modeled by a mixture of normal densities which is commonly known as mixture modeling. The arbitrary pdfs can be represented as a weighed combination of the normal pdfs as

$$f_X(x) = \sum_{i=1}^{N} \lambda_i \, f_i(x;\theta_i), \tag{5.5}$$

where x is the random vector with histogram specified by the spectrogram, N the number of normal distributions $f_i(x;\theta_i)$, where θ_i is the parameter vector of the i^{th} component consisting of the mean vector μ_i and covariance matrix $\sum_i$ and λ_i are the mixing weights, constrained to be positive and unity sum. In general, there are an infinite number of different *M*–mixture Gaussian densities that can be used to tile up the signal space. Hence, modeling the signal space can be regarded as a many-to-one, non-invertible mapping. The tractability in estimating the component parameters is due to the availability of the tools like the EM algorithm. Recently, t-f plane was modeled using the finite mixture modeling given in (Coates *et al*, 1998) to facilitate separation of multicomponent signals. To model the t-f space by means of mixture modeling, we first have to generate realizations that will drive the EM algorithm in such a way that their histogram resembles the spectrogram to be modeled. At first it may appear strange that we are tying to view the t-f space as a mixture of normal pdfs by applying a random process modeling technique to a deterministic problem. However, we are using it just as a means to accomplish the task of decomposing the signal into chirplets. We will consider the random vector (RV) generation in the following subsection.

5.3.1. Random Vector Generation

Generation of random vectors (RVs) for the specified bivariate pdf requires interpretation of the spectrogram as an arbitrary bivaraite pdf. Hence, we need to normalize the time dependent spectrum in such a way that it obeys the fundamental relation that a pdf should satisfy, i.e., integrating a pdf should give rise to one. The normalization can be done as follows:

Let $SP(t,\omega)$ be the spectrogram of the signal under analysis, $S_t(t)$ the time marginal and $S_\omega(\omega)$ the frequency marginal. Then, the normalization of the spectrogram should be done in such a way that:

$$\int S_t(t)\,dt = 1, \int S_\omega(\omega)\,d\omega = 1, \int SP(t,\omega)\,dt = S_\omega(\omega) \text{ and } \int SP(t,\omega)\,d\omega = S_t(t). \tag{5.6}$$

Then, the marginals of the scaled spectrogram are given by:

$$S_t^{modified}(t) = \frac{S_t(t)}{\iint SP(t,\omega)\,dt\,d\omega} \text{ and } S_\omega^{modified}(\omega) = \frac{S_t(t)S_\omega(\omega)}{\int S_\omega(\omega)\,d\omega} \;\forall\, S_t(t) \neq 0. \tag{5.7}$$

Now our task is to generate realizations corresponding to this scaled spectrogram. There exist different algorithms to generate RVs, namely, conditional pdf method, rejection algorithm, inverse transformation, etc (Devroye, 1986). The conditional pdf method is quite useful when we do not have closed form expressions for the conditional pdf and the disadvantage is that we cannot exploit any apriori information or speed up the procedure in generating the samples (or realizations). Whereas the band-rejection algorithm works on acceptance-rejection criteria that is one way sampling a target pdf to obtain the desired pdf, where target pdf is a pdf close to the desired pdf. The Band-rejection

algorithm is described in Table 5.1. It is essentially a sampling scheme that accepts the samples from the target pdf depending upon an acceptance criterion and associates them to the desired pdf. The rejection ratio defined in the algorithm determines the speed at which we can generate the desired realizations. A lesser rejection ratio means the target pdf is close to the pdf we wish and the ideal rejection ratio is one. In the present context of mixture modeling it serves two-fold purposes: Firstly, the lesser rejection ratio speeds up the process of RV generation and secondly it is a measure of our initial estimate in the modeling phase. Hence, the difficult task is in choosing a good target pdf that results in a lesser rejection ratio. Since we are dealing with mixture of bivarite normal densities, we should have an efficient way of generating samples for a normal bivarite pdf and then shuffle the samples of individual realizations according to the mixture probabilities. This algorithm can be found in generating realizations for a given Hidden Marokov Model (HMM), called Toy-Markov model generator (Rabiner *et al*, 1993). It is very logical to borrow the concept from Toy-Markov generator to compute realizations for the mixture density because we can consider the mixture density to be a one state HMM. Once the realizations for all the components in the mixture are generated, we can mix the realizations in a specified manner to obtain the realizations for the mixture density. The algorithm is described in Table 5.2. We review the EM algorithm in the following subsection.

5.3.2. EM Algorithm

It has long since been recognized that computing the maximum-likelihood (ML) parameter estimates can be a highly complicated task in many relevant estimation problems. The EM algorithm presented by Dempster et al (Dempster *et al*, 1977) is a

Table. 5.1: Band-rejection algorithm for generating random vectors of a specified pdf

Step-1. Initialization:

- Target pdf: $g(\mathbf{y})$
- Desired pdf (spectrogram to be modeled): $f(\mathbf{y})$
- A uniform distribution in the interval *[0,1] : U [0,1]*
- Rejection ratio: $R = max \frac{g(\mathbf{y})}{f(\mathbf{y})} \ \forall \ f(\mathbf{y}) \neq 0$
- Number of samples required: N
- set $i=0$ and $j=0$;

Step-2. Accept-Reject:

- Select a sample from uniform distribution : $r \sim U[0,1]$
- Select a sample from target pdf : $[\ \mathbf{y}_i\ ] \sim g(\mathbf{y})$
- *if* $r < R\frac{g(\mathbf{y}_i)}{f(\mathbf{y}_i)}$ *then* $j=j+1$;

  Associate the sample $[\ \mathbf{y}_i\ ]$ *with* $f(\mathbf{y})$,

  if $j < N$ go to step-3

  else stop.

else go to step-3;

Step-3. Loop:

- Repeat until N samples from $g(\mathbf{y})$ are associated to $f(\mathbf{y})$: $i=i+1;$ go to step-2.

Table. 5.2: Generation of RVs for a specified mixture of normal pdfs

Step-1. Initialization:

- For i=1:M, generate N_i realizations for the i^{th} component of the target pdf: where $N_i \geq \lambda_i N$.
- Partition the uniform random variable *U [0,1]* into *M*-segments according to the probability given by λ

Step-2. Mixing:

- Select a sample from uniform distribution : $r \sim U[0,1]$
- If the sample falls in the i^{th} segment, select a sample from i^{th} component of the mixture $g(\mathbf{y})$

Step-3. Loop:

- Repeat *step-2* until *N* samples for $g(\mathbf{y})$ are generated

general iterative method to compute ML estimates if the observed data can be regarded as "incomplete", like the formulation of $\mathbf{y} = h(\mathbf{x})$, where *h* is a noninvertible transformation and $\mathbf{x}$ is the complete data. As finding the ML estimates of θ by maximizing $ln\, f(\mathbf{y};\theta)$ is too difficult, we maximize $ln\, f(\mathbf{x};\theta)$. Since we do not have access to the complete data, at best we can maximize the conditional expectation of the complete data log-likelihood, given the incomplete data $\mathbf{y}$. This is given by

$$E\,[ln\, f_{X;\Theta}(\mathbf{x},\theta)/\,\mathbf{y}\,] = \int_X f_{X/Y;\Theta}(\mathbf{x}/\mathbf{y};\theta)\, ln\, f_{X;\Theta}(\mathbf{x};\theta)\, dx\,. \tag{5.8}$$

In the above equation, computation of the term $f_{X/Y;\Theta}(\boldsymbol{x}/\boldsymbol{y};\theta)$ requires an estimate of the unknown parameter vector θ. For this reason, the expectation of the likelihood function is maximized iteratively starting with an initial estimate of θ, and updating the estimate as described in Table 5.3. It has been demonstrated in (Vaseghi, 1996) that each iteration of the EM algorithm improves the convergence of the likelihood function and is monotonically non-decreasing. Before we apply EM algorithm to mixture modeling problem, we need to define the complete data and the incomplete data.

Table 5.3: EM Algorithm

Step-1. Initialization:

- Select an initial parameter estimate $\hat{\theta}_0$, and
- For i = 0, 1, … until convergence

Step-2. Expectation:

- Compute $U(\theta,\hat{\theta}_i) = E[\ln f_{X;\Theta}(\boldsymbol{x};\theta)/\boldsymbol{y};\hat{\theta}_i]$

Step-3.Maximization:

- Select $\hat{\theta}_{i+1} = \arg\max_{\theta} U(\theta,\hat{\theta}_i)$

Step-4.Convergence:

- If not converged then go to *step-2*.

As usual the observation vectors form the incomplete data and the complete data may be viewed as the observation vectors with a label attached to each vector $\boldsymbol{y}(k)$ to indicate the component of the mixture that generated it. They are defined as:

Incomplete data (observed data): $\mathbf{y}(n)$ $n=0,1,\ldots N-1$.

Complete data (unobserved data) : $[\mathbf{x}(n)=\mathbf{y}(n)_k]$ $n=0,1,\ldots N-1$, $k \in (1,\ldots,M)$.

Now the probability of the complete data, given the observed data, is the probability that the observation vector has the label k. Hence the E-step in Table 5.3 can be written as:

$$\begin{aligned} U(\theta,\hat{\theta}_i) &= E[\ln f_{X;\Theta}(\mathbf{x}(n);\theta)/\mathbf{y}(n);\hat{\theta}_i] \\ &= E[\ln f_{X;\Theta}(\mathbf{y}(n),k;\theta)/\mathbf{y}(n);\hat{\theta}_i] \\ &= \sum_{n=0}^{N-1}\sum_{k=1}^{M}\frac{f_{Y,K;\Theta}(\mathbf{y}(n),k/\hat{\Theta}_i)}{f_{Y/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i}\ln f_{Y,k;\Theta}(\mathbf{y}(n),k;\Theta), \end{aligned} \tag{5.9}$$

where $\Theta = \{\theta_k = [\lambda_k, \mu_k, \Sigma_k]\ k = 1,2,...,M\}$ are the parameters of the mixture density. The joint probability density function of $\mathbf{y}(n)$ and the k^{th} component of the mixture density is given by:

$$f_{Y,K;\Theta}(\mathbf{y}(n),k/\hat{\Theta}_i) = \lambda_{k_i}\ f_k(\mathbf{y}(n)/\hat{\theta}_{k_i}), \tag{5.10}$$

where $f_k(\mathbf{y}(n)/\theta_k)$ is a Gaussian density of mean vector μ_k and the covariance matrix Σ_k is given as:

$$f_k(\mathbf{y}(n)/\hat{\theta}_{k_i}) = \frac{1}{\sqrt{2\pi|\Sigma_k|}}\exp\left(-\frac{1}{2}(\mathbf{y}(n)-\mu_k)^T\Sigma_k^{-1}(\mathbf{y}(n)-\mu_k)\right). \tag{5.11}$$

The pdf of $\mathbf{y}(n)$ and the mixture of N Gaussian densities is given by:

$$f_{\mathbf{Y}/\Theta}(y(n)/\hat{\Theta}_i) = \sum_{k=1}^{M}\lambda_{k_i} f_k(y(n);\hat{\theta}_{k_i}). \tag{5.12}$$

Substitution of the Gaussian densities of Eqn. (5.10) and Eqn. (5.11) into Eqn. (5.9) yields

$$U(\theta,\hat{\theta}_i) = \sum_{n=0}^{N-1}\sum_{k=1}^{M}\frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)}\ln[\lambda_k f_k(\mathbf{y}(n);\theta_k)]$$

$$= \sum_{n=0}^{N-1}\sum_{k=1}^{M}\left(\frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)}\ln\lambda_k + \frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)}\ln[f_k(\mathbf{y}(n);\theta_k)]\right) \tag{5.13}$$

To maximize the above equation, we should set the derivative of the likelihood function with respect to λ_k, μ_k *and* Σ_k to zero. The weight of the k^{th} component is given by:

$$\hat{\lambda}_{k_{i+1}} = \arg\max_{\lambda_k} U[\theta,\hat{\theta}_{i+1}]. \tag{5.14}$$

However, we should set the derivative to zero subject to the constraint that $\sum_{m=1}^{M}\lambda_k = 1$. As our constraint is an equality condition, we use the Lagrangean method of optimization (Rau, 1970) which can be formulated as described below:

Let the objective function be $f(x_1, x_2 ..., x_M)$ and constraint function be $g(x_1, x_2 ..., x_M)$. Define $L = f(x) - \zeta g(x)$, where ζ is the Lagrangean multiplier, which need be determined. To optimize *L,* set its partial derivative to zero, i.e., $\frac{\partial}{\partial x_i}(L) = 0$. Solving the expressions obtained by setting the partial derivatives of *L* to zero gives the optimal solution. This algorithm can be extended to cases involving multiple constraints. In the present context, our objective function is $U(\theta,\hat{\theta}_i;\lambda_k)$ and the constraint function is $\sum_{k=1}^{M}\lambda_k - 1 = 0$. Hence,

$$L = \sum_{n=0}^{N-1}\sum_{k=1}^{M}\left(\frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)}\ln\lambda_k + \frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)}\ln[f_k(\mathbf{y}(n);\theta_k)]\right) - \zeta\left(\sum_{k=1}^{M}\lambda_k - 1\right). \tag{5.15}$$

Setting the partial derivative with respect to λ_k to zero, we get

$$\frac{\partial L}{\partial \lambda_k} = \sum_{n=0}^{N-1} \left(\frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} \frac{1}{\lambda_k}) \right) - \zeta = 0. \tag{5.16}$$

The solution of the above equation is given by

$$\zeta = \sum_{n=0}^{N-1} \left(\frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} \frac{1}{\lambda_k}) \right) \quad for\ k = 1,2,....,M. \tag{5.17}$$

By equating the above expression for $k = a$ and $k = b$, we can represent λ_a in terms of λ_b as:

$$\lambda_b = \lambda_a \frac{\sum_{n=0}^{N-1} \left(\frac{\hat{\lambda}_{b_i} f_b(\mathbf{y}(n)/\theta_{b_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} \right)}{\sum_{n=0}^{N-1} \left(\frac{\hat{\lambda}_{a_i} f_a(\mathbf{y}(n)/\theta_{a_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} \right)} . \tag{5.18}$$

By using the above result, we can rewrite the constraint function as:

$$\lambda_a + \sum_{k=1\ k\neq a}^{M} \lambda_k \frac{\sum_{n=0}^{N-1} \left(\frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{k/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} \right)}{\sum_{n=0}^{N-1} \left(\frac{\hat{\lambda}_{a_i} f_a(\mathbf{y}(n)/\theta_{a_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} \right)} = 1 . \tag{5.19}$$

Simplifying the above equation by noting the fact that $\sum_{k=1}^{M} \sum_{n=0}^{N-1} \left(\frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{k/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} \right) = N$,

the parameter λ_k is given by

$$\hat{\lambda}_{k_{i+1}} = \frac{1}{N} \sum_{n=0}^{N-1} \frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} . \tag{5.20}$$

To obtain the remaining parameters, we may require the following matrix identities (Mix, 1995):

$$\frac{d}{dX}(X^T AX) = X(A + A^T),\ \frac{d}{dX}|X| = |X| X^{-1} \text{ and } \frac{d}{dX}(A^T X^{-1} A) = -X^{-1} AA^T X^{-1}.$$

(5.21)

Similarly, the mean vector μ_k can be obtained by maximizing Eqn. (5.15) with respect to μ_k and we get :

$$\frac{\partial}{\partial \mu_k} U(\theta, \hat{\theta}_i) = \sum_{n=0}^{N-1} \frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{Y/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} \left(0 + \frac{\partial}{\partial \mu_k} \ln[f_k(\mathbf{y}(n); \theta_k)]\right) = 0. \quad (5.22)$$

Using the product rule of the logarithm and Eqns. (5.10) and (5.11), we can express Eqn. (5.22) as:

$$\sum_{n=0}^{N-1} \frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} \frac{\partial}{\partial \mu_k} \left(ln\left(\frac{1}{\sqrt{2\pi|\Sigma_k|}}\right) + ln\left[exp\left[-\frac{1}{2}(\mathbf{y}(n) - \mu_k)^T \Sigma_k^{-1} (\mathbf{y}(n) - \mu_k)\right] \right) = 0. \quad (5.23)$$

Using the matrix identies given in Eqn. (5.21), we get

$$\sum_{n=0}^{N-1} \frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} \left(-\frac{1}{2}(\mathbf{y}(n) - \mu_k) 2(\Sigma_k^{-1} + \Sigma_k^{-1^T}) \right) = 0, \quad (5.24)$$

which gives

$$\hat{\mu}_{k_{i+1}} = \frac{\sum_{n=0}^{N-1} \frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{Y/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} \mathbf{y}(n)}{\sum_{n=0}^{N-1} \frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{Y/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)}}. \quad (5.25)$$

Similarly $\hat{\Sigma}_{k_{i+1}}$ can be obtained by maximizing $U[\theta, \hat{\theta}_{i+1}]$ with respect to Σ_k as:

$$\frac{\partial}{\partial \Sigma_k} U(\theta, \hat{\theta}_i) = \sum_{n=0}^{N-1} \frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{Y/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} \left(0 + \frac{\partial}{\partial \Sigma_k} \ln[f_k(\mathbf{y}(n); \theta_k)]\right) = 0.$$

(5.26)

Expanding Eqn. (5.26) using the relation in Eqn. (5.12), we obtain

$$\sum_{n=0}^{N-1} \frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} \frac{\partial}{\partial \Sigma_k} \left(ln\left(\frac{1}{\sqrt{2\pi|\Sigma_k|}} \right) + ln\left[exp\left[-\frac{1}{2}(\mathbf{y}(n)-\mu_k)^T \Sigma_k^{-1}(\mathbf{y}(n)-\mu_k) \right] \right] \right) = 0. \tag{5.27}$$

Using the matrix identities given in Eqn. (5.21), the partial derivative of the term inside the bracket of the left hand side of the above equation gives us:

$$\sum_{n=0}^{N-1} \frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} \left(-\frac{1}{2} \frac{|\Sigma_k| \, \Sigma_k^{-1}}{|\Sigma_k|} + \frac{1}{2}(\Sigma_k^{-1}\mathbf{y}(n)-\mu_k)(\mathbf{y}(n)-\mu_k)^T \Sigma_k^{-1}) \right) = 0. \tag{5.28}$$

Then, it is straightforward to get Σ_k as:

$$\hat{\Sigma}_{k_{i+1}} = \frac{\sum_{n=0}^{N-1} \frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)} (\mathbf{y}(n)-\hat{\mu}_{k_i})(\mathbf{y}(n)-\hat{\mu}_{k_i})^T}{\sum_{n=0}^{N-1} \frac{\hat{\lambda}_{k_i} f_k(\mathbf{y}(n)/\theta_{k_i})}{f_{\mathbf{Y}/\Theta}(\mathbf{y}(n)/\hat{\Theta}_i)}}, \tag{5.29}$$

which completes the parameter estimation process. Eventhough EM algorithm results in an accurate model depending on the initial estimate, it is computationally intensive. Recently, variants of EM algorithm which improve the convergence rate significantly have been proposed (Fessler *et al*, 1994 and Neal *et al*, 1993). We try to re derive Eqn. (5.9) using the incremental view point of (Neal *et al*, 1993) in the following section.

5.3.3. Incremental-Based EM Algorithm

There exist many variants of the EM algorithm, some of them differ from the standard algorithm in the way the E-step and M-steps are implemented. The M-step of the

algorithm may be partially implemented, with the new estimate for the parameters improving the likelihood given the distribution found in E-step, but necessarily maximizing it. Such a partial M-step also results in the true likelihood improvement, referred to as *Generalized EM* (GEM) algorithms (Neal *et al*, 1993). The basic idea behind the *Incremental-based EM algorithm* is the partial implementation of the E-step. In many cases, partial implementation of the E-step is also natural. As the unobserved variables are commonly independent, they influence the likelihood of the parameters only through simple sufficient statistics. If the statistics for the E-step are incrementally collected and the parameters are frequently estimated, it should speed up the convergence, since the information from the new data contributes to the parameter estimation more quickly than the standard algorithm. The incremental EM algorithm is described in Table 5.4. This view point can be applied when the observed data $\boldsymbol{y} = (\boldsymbol{y}^1, \boldsymbol{y}^2, ..., \boldsymbol{y}^L)$ are L independent sets and the complete data can be decomposed as $\boldsymbol{x} = (\boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^L)$. The general approach is to collect subset of the data that essentially meets the above requirement and then update the parameters based on the subset until the convergence of the parameters. In the context of mixture modeling, the spectrogram of the signal under analysis/synthesis, we can generate the L independent sets of observed data by performing the RV generation algorithm that many times. Then, the joint probability for Y and X can be factored as:

$$f_{X,Y/\Theta}(\boldsymbol{x}, \boldsymbol{y} / \theta) = \prod_{l=1}^{L} f_{X^l,Y^l/\Theta^l}(\boldsymbol{x}^l, \boldsymbol{y}^l / \theta^l), \tag{5.30}$$

where θ^l represents the parameter estimates of the l^{th} segment.

Table 5.4: Incremental-based EM algorithm

Step-1.Initialization:

- Select an initial parameter estimate for each of the L segments
- $\hat{\theta}_0^l \quad l = 1, 2, ..., L$, and for the whole data $\hat{\theta}_0^{tot}$
- For i = 0, 1, … until convergence

Step-2: Expectation:

- Choose the data segment, l, to be updated
- Set $U(\theta^m, \hat{\theta}_i^m) = U(\theta^m, \hat{\theta}_{i-1}^m)$ for $m \neq l$
- Compute $U(\theta^l, \hat{\theta}_i^l) = E[\ln f_{X^m;\Theta^m}(\boldsymbol{x}^m; \theta^m) / \boldsymbol{y}; \hat{\theta}_{i-1}^m]$

Step-3: Maximization:

- Select

$$\hat{\theta}_i^{tot} = \arg\max_{\theta} [\, U(\theta, \hat{\theta}_{i-1}^{tot}) - U(\theta, \hat{\theta}_{i-1}^l) + U(\theta, \hat{\theta}_i^l) \,]$$

Step-4: Convergence:

- If not converged then go to step-2.

Now, we can expand the E-step in Table 5.4 as:

$$\begin{aligned} U(\theta, \hat{\theta}_i^{tot}) &= E[\ln f_{X;\Theta}(\mathbf{x}(n); \theta) / \mathbf{y}(n); \hat{\theta}_i] \\ &= E[\ln f_{X;\Theta}(\mathbf{y}(n), k; \theta) / \mathbf{y}(n); \hat{\theta}_i]. \end{aligned} \tag{5.31}$$

By imposing the constraint on the data segments that they are independent, Eqn. (5.31) can be rewritten as:

$$U(\theta,\widehat{\theta}_{\mathrm{i}}^{\mathrm{tot}}) = E[ln\prod_{l=1}^{L}\left(f_{X^l;\Theta^l}(\mathbf{y}^l(n),k;\theta^l)/\mathbf{y}^l(n);\widehat{\theta}_i^l\right)]. \tag{5.32}$$

Taking the logarithmic operation of the product of the conditional pdfs, we get

$$U(\theta,\widehat{\theta}_{\mathrm{i}}^{\mathrm{tot}}) = E[\sum_{l=1}^{L} ln[\, f_{X^l;\Theta^l}(\mathbf{y}^l(n),k;\theta^l)/\mathbf{y}^l(n);\widehat{\theta}_i^l\,]. \tag{5.33}$$

As expectation operator is linear, we can rewrite the above equation as:

$$U(\theta,\widehat{\theta}_{\mathrm{i}}^{\mathrm{tot}}) = \sum_{l=1}^{L} E[\, ln\left(f_{X^l;\Theta^l}(\mathbf{y}^l(n),k;\theta^l)/\mathbf{y}^l(n);\widehat{\theta}_i^l\right)]. \tag{5.34}$$

Now, we implement the partial E-step on the data segment *m* and then Eqn. (5.34) can be split as

$$\begin{aligned} U(\theta,\widehat{\theta}_i^{tot}) &= E[ln\left(f_{X^m;\Theta^m}(\mathbf{y}^m(n),k;\theta^m)/\mathbf{y}^m(n);\widehat{\theta}_i^m\right)] \\ &\quad + \sum_{l=1\ l\neq m}^{L} E\,[\ ln\left(f_{X^l;\Theta^l}(\mathbf{y}^l(n),k;\theta^l)/\mathbf{y}^l(n);\widehat{\theta}_i^l\right)]. \end{aligned} \tag{5.35}$$

Since we are updating the parameters for the m^{th} segment, the expected log-likelihood of the other data segments remain unchanged. Hence,

$$U(\theta,\widehat{\theta}_{\mathrm{i}}^{\mathrm{tot}}) = E[ln\left(f_{X^m;\Theta^m}(\mathbf{y}^m(n),k;\theta^m)/\mathbf{y}^m(n);\widehat{\theta}_{i-1}^m\right)] + \sum_{l=1\ l\neq m}^{L} U(\theta,\widehat{\theta}_{i-1}^l). \tag{5.36}$$

Adding and subtracting the term $U(\theta,\widehat{\theta}_{i\text{-}1}^m)$ to the right hand side of the above equation gives us:

$$\begin{aligned} U(\theta,\widehat{\theta}_i^{tot}) &= E[ln\left(f_{X^m;\Theta^m}(\mathbf{y}^m(n),k;\theta^m)/\mathbf{y}^m(n);\widehat{\theta}_{i-1}^m\right)] + \sum_{l=1\ l\neq m}^{L} U(\theta,\widehat{\theta}_{i-1}^l) \\ &\quad + U(\theta,\widehat{\theta}_{i\text{-}1}^m) - U(\theta,\widehat{\theta}_{i\text{-}1}^m) \\ &= E[ln\left(f_{X^m;\Theta^m}(\mathbf{y}^m(n),k;\theta^m)/\mathbf{y}^m(n);\widehat{\theta}_{i-1}^m\right)] + \sum_{l=1}^{L} U(\theta,\widehat{\theta}_{i-1}^l) - U(\theta,\widehat{\theta}_{i\text{-}1}^m). \end{aligned} \tag{5.37}$$

By denoting $E[ln\left(f_{X^m;\Theta^m}(\boldsymbol{y}^m(n),k;\theta^m)/\boldsymbol{y}^m(n);\hat{\theta}_{i-1}^m\right)]$ as $U(\theta,\hat{\theta}_i^m)$, we obtain

$$U(\theta,\hat{\theta}_i^{tot}) = U(\theta,\hat{\theta}_{i-1}^{tot}) + U(\theta,\hat{\theta}_i^m) - U(\theta,\hat{\theta}_{i-1}^m). \tag{5.38}$$

It appears that eventhough we perform partial E-step on a selected segment, we have to maximize the likelihood function on the whole data. However, usage of sufficient statistics leads to an efficient way of implementing the M-Step (Neal *et al*, 1993). Sufficient statistics for a Gaussian process are the mean and the covariance matrix (Kay, 1993). For a mixture of densities, the parameters that characterize the whole process are the weight vector, the mean vector and the covariance matrix. As we do not have the idea of sufficient statistics for the mixture density, we resort to the conventional method of maximizing Eqn. (5.38) to obtain the parameters. The expected log-likelihood has to be maximized with respect to λ_k, μ_k *and* Σ_k to obtain the parameters $\hat{\lambda}_{k_{i+1}}, \hat{\mu}_{k_{i+1}}$ *and* $\hat{\Sigma}_{k_{i+1}}$ as:

$$\begin{aligned}
\lambda_{k_i} &= \lambda_{k_{i-1}} + \frac{1}{N_m}\left(\lambda_{k_i}^m - \lambda_{k_{i-1}}^m\right), \\
\mu_{k_i} &= \frac{N\lambda_{k_{i-1}}\mu_{k_{i-1}} + N_m\lambda_{k_i}^m\mu_{k_i}^m - N_m\lambda_{k_{i-1}}^m\mu_{k_{i-1}}^m}{N\lambda_{k_i}} \quad \textit{and} \\
\Sigma_{k_i} &= \frac{N\lambda_{k_{i-1}}\Sigma_{k_{i-1}} + N_m\lambda_{k_i}^m\Sigma_{k_i}^m - N_m\lambda_{k_{i-1}}^m\Sigma_{k_{i-1}}^m}{N\lambda_{k_i}}.
\end{aligned} \tag{5.39}$$

where N is the total number of realizations of $\boldsymbol{y}$, N_m number of realizations of the segment $\boldsymbol{y}^m$; and $\lambda_{k_i}^m, \mu_{k_i}^m$ and $\Sigma_{k_i}^m$ are the parameters of the m^{th} segment. Using Eqns. (5.20), (5.25) and (5.29) to compute the parameters $\lambda_{k_i}^m, \mu_{k_i}^m$ and $\Sigma_{k_i}^m$, respectively, on the subset of the data labeled $\boldsymbol{y}_m$ reduces the complexity to a direct maximization of the likelihood function, since we are computing only $\lambda_{k_i}^m, \mu_{k_i}^m$ and $\Sigma_{k_i}^m$ at each iteration and are using the earlier estimates adequately to obtain λ_k, μ_k *and* Σ_k. The proof of Eqn.

(5.39) can be given on similar lines as was done for the EM algorithm. A comparison of the Incremental EM and the standard EM algorithm is shown in Fig. 5.3. for the parameters given in Table 5.5.

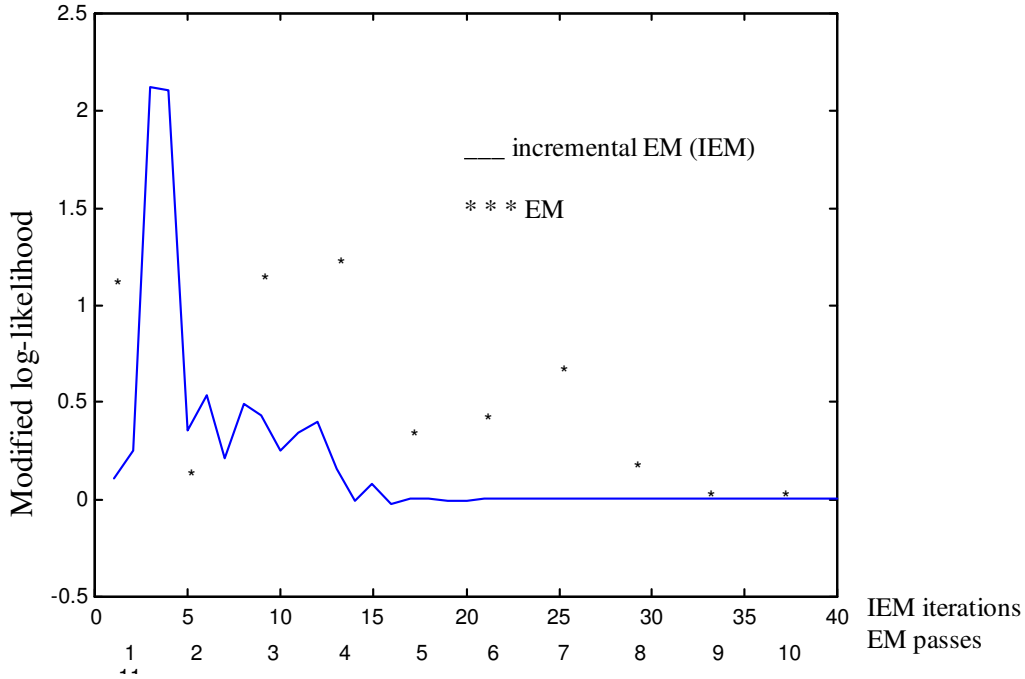


Fig. 5.3. Comparison of incremental-based and standard EM algorithms

Table 5.5: A comparison of the estimated components from the EM and the Incremental EM algorithms

		Mixture model parameters	Initial estimates	Estimated parameters after 10 passes of EM algorithm	Estimated parameters after 5 passes of incremental EM algorithm
Component 1	μ	$\begin{bmatrix}40\\64\end{bmatrix}$	$\begin{bmatrix}20\\40\end{bmatrix}$	$\begin{bmatrix}40.892\\63.923\end{bmatrix}$	$\begin{bmatrix}40.086\\64.135\end{bmatrix}$
	Σ	$\begin{bmatrix}30 & 0\\0 & 20\end{bmatrix}$	$\begin{bmatrix}80 & 20\\20 & 40\end{bmatrix}$	$\begin{bmatrix}22.82 & 1.287\\1.287 & 13.120\end{bmatrix}$	$\begin{bmatrix}22.82 & 1.244\\1.244 & 13.161\end{bmatrix}$
	λ	0.3	0.5	0.345	0.342
Component 2	μ	$\begin{bmatrix}70\\64\end{bmatrix}$	$\begin{bmatrix}20\\40\end{bmatrix}$	$\begin{bmatrix}69.601\\64.932\end{bmatrix}$	$\begin{bmatrix}69.619\\64.135\end{bmatrix}$
	Σ	$\begin{bmatrix}30 & 0\\0 & 20\end{bmatrix}$	$\begin{bmatrix}40 & -8\\-8 & 30\end{bmatrix}$	$\begin{bmatrix}29.861 & 0.756\\0.756 & 21.030\end{bmatrix}$	$\begin{bmatrix}29.852 & 0.742\\0.741 & 20.891\end{bmatrix}$
	λ	0.7	0.5	0.654	0.657

Total number of samples *(N)* = 128
Number of segments *(I)* = 4
Number of samples in each segment (N_l) = 32
Number of components = 2

It can be observed from the figure that the IEM converges within the 6th pass of the EM algorithm, i.e., at 20th iteration of the IEM algorithm (one pass of the EM algorithm is

equivalent to L iterations of the IEM, where L is the number of segments of the data), whereas EM algorithm converges only after the 10^{th} pass. On the y-axis is shown the modified likelihood defined as:

$$U_{ml} = U(\theta,\hat{\theta}_i) - U(\theta,\hat{\theta}_{i-1}) \quad . \tag{5.40}$$

Since each estimate of the parameter should result in an improved likelihood, expressed in terms of the log-likelihood as:

$$U(\theta,\hat{\theta}_i) \geq U(\theta,\hat{\theta}_{i-1}) \,, \tag{5.41}$$

the modified likelihood function, U_{ml}, is non-negative and becomes zero when the parameters converge. The complexity of computing the parameters can be reduced by noting the fact that convergence rate depends on the initial estimates passed in the *Initialization* step of Table 5.4. A good initial estimate always results in quick convergence. K-means clustering can be used to classify the segments into the desired number of components and estimate the mean vectors (McLachlan, 1987). The obvious question that arises is: K-means clustering and mixture modeling themselves are two random signal space modeling techniques. Then, why cannot we use K-means instead of Mixture modeling? The reason is that K-means clustering cannot associate a classified sample to another, i.e., the clusters are disjoint, whereas in mixture modeling, a sample can have joint observation in adjacent clusters since Guassian densities can be overlapping, with the result that in an area of overlap a data point can be associated with various probabilities to different components of the Gaussian mixture (Vaseghi, 1996). However, this procedure helps us in automating the selection of number of components in the mixture after the realizations are generated using band-rejection algorithm. We will now review K-means clustering algorithm.