if the right-hand side of (2.6) is a good approximation of $x(t)$. Observe that the subspace spanned by the sinc functions in (2.6) has dimension

$$N := N_2 - N_1 + 1 \approx N_2 - N_1 = 2WT.$$

For this reason, the set of waveforms bandlimited to W and of approximate duration T is said to have $2WT$ **degrees of freedom** or **approximate dimension** $2WT$.

2.3. Plotting Linear Combinations of Waveforms

Suppose we want to plot

$$y(t) = \sum_{j=1}^{n} c_j x_j(t).$$

In many cases, x_j is a scaled shift of some fixed waveform x, say

$$x_j(t) = x\big(s_j(t - m_j)\big).$$

For example, in Problem 2–7, we would take $x(t) = e^t$ with $s_j = \beta_j$ and $m_j = 0$. In (2.6) we would take $x(t) = \mathrm{sinc}(t)$ with $s_j = 2W$ and $m_j = j\tau$. When the x_j have the form of scaled shifts, we can plot a linear combination at points t_i very easily in MATLAB. Observe that

$$y(t_i) = \sum_{j=1}^{n} c_j x\big(s_j(t_i - m_j)\big), \quad i = 1, \ldots, M,$$

says that the row vector $[y(t_1), \ldots, y(t_M)]$ is the product of the row vector $c = [c_1, \ldots, c_n]$ and the $M \times n$ matrix with ij entry $x\big(s_j(t_i - m_j)\big)$. This suggests the MATLAB function `lincmb` below to compute the vector $[y(t_1), \ldots, y(t_M)]$ if the row vector of coefficients c, the row vector of shifts m, and the row vector of scale factors s are given. Before listing the function `lincmb`, we give several examples of how to use it.

Example. To plot $e^t + 2e^{-t} + 4e^{-t/3}$ on $[0, 1]$, use the following commands

```
t=linspace(0,1,200);
c = [ 1 2 4 ];
s = [ 1 -1 -1/3 ];
m = zeros(size(s));
y = lincmb(t,c,'exp',m,s);
plot(t,y)
```

to get the graph shown at the top in Figure 2.1. To plot the functions e^t, e^{-t}, and $e^{-t/3}$ themselves, replace the last two command lines with

```
[y,xmat] = lincmb(t,c,'exp',m,s);
plot(t,xmat)
```

The result is shown at the bottom in Figure 2.1.

Example 2.11. We use `lincmb` to plot the linear combination

$$y(t) = \sum_{k=3}^{9} e^{-(t-k)^2/2}/\sqrt{2\pi}$$

for $t \in [0, 12]$. In addition, we plot the functions $e^{-(t-k)^2/2}/\sqrt{2\pi}$ on the same graph. The first step is to create an M-file with the desired function:

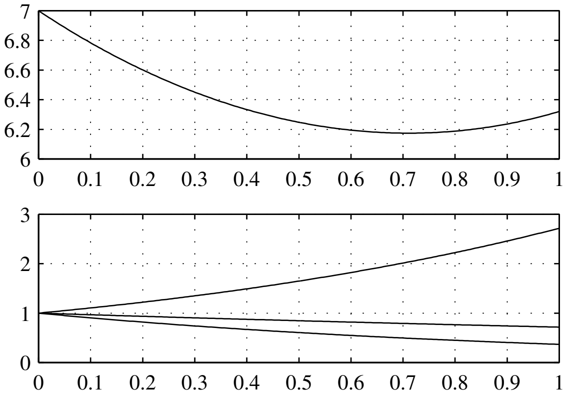

Figure 2.1. The linear combination $e^t + 2e^{-t} + 4e^{-t/3}$ (top) and the functions e^t, e^{-t}, and $e^{-t/3}$ (bottom).

```
function x = gaussdens(t)
x = exp(-t.^2/2)/sqrt(2*pi);
```

We then use the following commands:

```
t=linspace(0,12,200);
m = [3:9];
c = ones(size(m));
[y,xmat] = lincmb(t,c,'gaussdens',m);
plot(t,y,'--',t,xmat)
```

See Figure 2.2. Notice that since the scale factors are all one, the fifth argument of `lincmb` can be omitted.

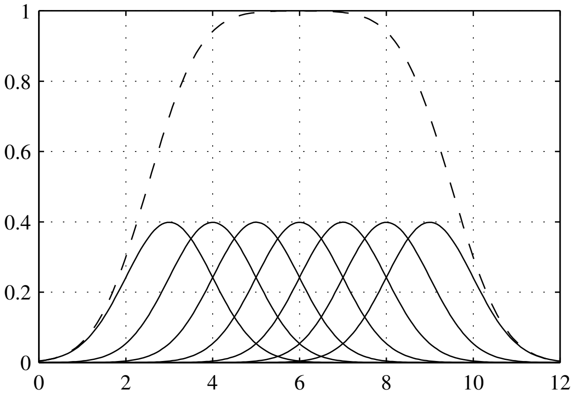

Figure 2.2. Linear combinations of Gaussians (dashed line) and individual Gaussians (solid lines).

Example 2.12 (Sampling Theorem Interpolation). Suppose that `x` is a row vector of samples of a bandlimited function x that is bandlimited to W. Let `tau` denote the corresponding row vector of sampling times. If `t` is a row vector of reconstruction times at which we want to compute (2.6), then the following MATLAB function will produce the required computations.

```
function y = smpthm(t,x,tau,W)
%
% Use the sampling theorem to reconstruct
% a bandlimited waveform.
%
% t = ARRAY of reconstruction times
% x = ROW vector of waveform samples
% tau = ROW vector of sampling times
% W = bandwidth of waveform

twoW = 2*W;
sp = tau(2)-tau(1);    % sampling period
y = twoW*sp*lincmb(t,x,’sinc’,tau,twoW);
```

To illustrate its use, we consider the standard Gaussian density, which is approximately bandlimited to $W = 1$. The following commands produced Figure 2.3, in which the samples are indicated by the circles, and the approximate reconstruction is given by the solid line. The reader can experiment with the parameter `Ts`. As `Ts` increases, the reconstruction breaks down. The reader should also experiment with plotting the reconstruction outside the interval where the waveform samples are taken. In this case, the samples and the reconstruction should be plotted on different graphs.

```
W = 1;           % BW of waveform
fsMin = 2*W;     % Nyquist rate
TsMax = 1/fsMin; % Nyquist period
Ts = .5;         % Actual sampling period
Tasamp = -5;     % Interval to sample
Tbsamp = 5;
n = ceil((Tbsamp-Tasamp)/Ts); % # samples
tsamp = linspace(Tasamp,Tbsamp,n);
Taplot = -3;     % Interval to plot results
Tbplot = 3;
x = gaussdens(tsamp); % samples
t = linspace(Taplot,Tbplot,200);
y = smpthm(t,x,tsamp,W);      % sampling thm
plot(tsamp,x,’o’,t,y)
```

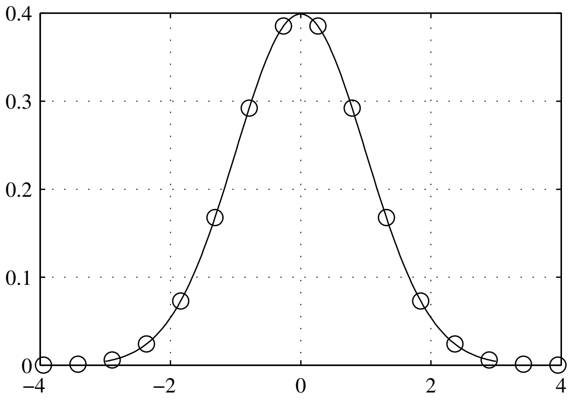

Figure 2.3. Reconstruction of a Gaussian density using the finite-sum approximation (2.6) of the sampling-theorem formula.

Here is the function `lincmb`:

```
function [y,xmat] = lincmb(t,c,xfun,m,varargin)
%
% Usage:  lincmb(t,c,xfun,m)
%     or  lincmb(t,c,xfun,m,s)
% where if the optional argument s is
% omitted, it is taken to be 1.
%
% Compute
%   y(t_i) = \sum_j c_j xfun_j(t_i)
% and
%   xmat = matrix with ij element xfun_j(t_i)
% where xfun_j(t_i) = xfun( s_j(t_i-m_j) )
%
% t = ARRAY of times at which the
%     linear combination will be evaluated.
% c = ROW vector of coefficients of the
%     linear combination.
% xfun = STRING containing the name of the
%        underlying function to be used.
% m = ROW vector of shifts.
% s = OPTIONAL ROW vector of scale factors.
%     WARNING: If length(s)>1, then s and
%     m are assumed to have the same length;
%     otherwise an ERROR may result.
% y = ARRAY output (same dimension as t)
%
sizt = size(t); % Make t a column vector
lt = prod(sizt);
tt = reshape(t,lt,1);
lm = length(m);
if nargin==4
  tmat = repmat(tt,1,lm);
  mmat = repmat(m,lt,1);
else
  s = varargin{1};
  ls = length(s);
  if ls==1
    tmat = repmat(s(1)*tt,1,lm);
  else
    tmat = tt*s;
  end
  mmat = repmat(s.*m,lt,1);
end
xmat = feval(xfun,tmat-mmat);
yloc = xmat*(c.’);
y = reshape(yloc,sizt);
```

Remark. The inner product space V in this section is totally arbitrary. However, if $V = \mathbb{C}^d$ is equipped with the standard Euclidean inner product $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{v}^H \mathbf{u}$, where the superscript ^{H} denotes the complex conjugate transpose, then it is not hard to see that (6.11) can be expressed as

$$\mathscr{W}^H \mathscr{W} c = \mathscr{W}^H \mathbf{v}, \tag{6.16}$$

where $\mathscr{W}$ denotes the $d \times n$ matrix whose columns are $\mathbf{w}_1, \ldots, \mathbf{w}_n$,

$$\mathscr{W} := \big[\, \mathbf{w}_1 \,|\cdots|\, \mathbf{w}_n \,\big].$$

It is easy to see that (6.10) becomes

$$\widehat{\mathbf{v}} \;=\; \mathscr{W} c \;=\; \mathscr{W}(\mathscr{W}^H \mathscr{W})^{-1} \mathscr{W}^H \mathbf{v}. \tag{6.17}$$

6.1.1. Approximation of Functions

Let $v(t)$ be a given function to be approximated using well-known functions $w_1(t), \ldots, w_n(t)$. One approach is to seek coefficients $c_1, \ldots, c_n$ that minimize

$$\bigg\| v - \sum_{j=1}^n c_j w_j \bigg\|^2 \;=\; \int \bigg| v(t) - \sum_{j=1}^n c_j w_j(t) \bigg|^2 dt.$$

Let $W := \operatorname{span}\{w_1, \ldots, w_n\}$. Then the optimal coefficients are given by (6.11), where

$$G_{ij} \;=\; \int w_j(t) w_i(t)\, dt \quad \text{and} \quad b_i \;=\; \int v(t) \overline{w_i(t)}\, dt. \tag{6.18}$$

Example 6.12 (Least Squares Polynomial Approximation). In Example 5.10 we considered the uniform approximation of continuous functions on a finite interval $[a, b]$ by polynomials. In that example, the distance between two continuous functions, say v and u, was measured by the uniform norm,

$$\|v - u\|_\infty \;:=\; \max_{a \le t \le b} |v(t) - u(t)|.$$

In the next two problems we measure distance in the least squares sense,

$$\|v - u\|_2 \;:=\; \bigg(\int_a^b |v(t) - u(t)|^2\, dt \bigg)^{1/2}.$$

Now observe that

$$\|v - u\|_2^2 \;\le\; \int_a^b \|v - u\|_\infty^2\, dt \;=\; \|v - u\|_\infty^2 (b - a).$$

Let $\widehat{v}_2$ denote the best approximation of v in the least squares sense, and let $\widehat{v}_\infty$ denote the best approximation of v in the uniform norm. Then we have the following relationships:

$$\begin{aligned} \|v - \widehat{v}_2\|_2 \;&\le\; \|v - \widehat{v}_\infty\|_2 \\ &\le\; \|v - \widehat{v}_\infty\|_\infty \sqrt{b-a} \\ &\le\; \|v - \widehat{v}_2\|_\infty \sqrt{b-a}. \end{aligned}$$

Problem 6–17. You wish to approximate the function $v(t) = t^3$ for $0 \le t \le 1$ using a polynomial of degree 1; i.e., $\widehat{v}(t) = c_1 + c_2 t$. Find numerical values of c_1 and c_2 that minimize the mean squared error,

$$\int_0^1 \big| v(t) - \widehat{v}(t) \big|^2\, dt.$$

Problem 6–18. Find the best approximation (in the sense of mean squared error) of $v(t) = t^2, 0 \le t \le 1$, by a polynomial $\widehat{v}(t)$ of degree at most one that also satisfies $\int_0^1 \widehat{v}(t)\, dt = 0$.

If $v(t)$ is not given theoretically, or if the integrals for b_i above cannot be computed numerically, then the foregoing approach is not possible. However, suppose samples $v(t_1), \ldots, v(t_m)$ are available. Put

$$\mathbf{v} := [v(t_1), \ldots, v(t_m)]' \quad \text{and} \quad \mathbf{w}_j := [w_j(t_1), \ldots, w_j(t_m)]'.$$

Using the standard inner product on $\mathbb{R}^n$ or $\mathbb{C}^n$ and the corresponding Euclidean distance, we have

$$\bigg\| \mathbf{v} - \sum_{j=1}^n c_j \mathbf{w}_j \bigg\|^2 \;=\; \sum_{i=1}^m \bigg| v(t_i) - \sum_{j=1}^n c_j w_j(t_i) \bigg|^2. \tag{6.19}$$

If we put $\mathscr{W} := [\mathbf{w}_1, \ldots, \mathbf{w}_n]$, then in $Gc = b$, $G = \mathscr{W}^H \mathscr{W}$, and $b = \mathscr{W}^H \mathbf{v}$. Note that once we have found the coefficients $c_1, \ldots, c_n$, the approximation

$$\widehat{v}(t) \;=\; \sum_{j=1}^n c_j w_j(t) \tag{6.20}$$

can be evaluated for all t for which the $w_j(t)$ are defined, not just the sample times $t_1, \ldots, t_m$. Hence, for $t_i < t < t_{i+1}$, $\widehat{v}(t)$ can serve as an approximation of $v(t)$.

Remark. Whether we solve $Gc = b$ when G and b are given by (6.18) or by (6.19), to plot the projection, we must plot the waveform given by (6.20). If the w_j are scaled and shifted versions of common waveform, say $w_j(t) = w(s_j(t - m_j))$, then we can use the MATLAB function `lincmb` discussed in Section 2.3. In fact, in the case of (6.19), the matrix $\mathscr{W}$ is provided by `lincmb`.

Remark. Suppose that the $\mathbf{w}_j$ are linearly independent and that $m = n$. Then the $\mathbf{w}_j$ are a basis for n-dimensional Euclidean space, and the minimum value of (6.19) is zero when the c_j are the unique coefficients of the representation of $\mathbf{v}$ in this basis. In other words, the approximation $\widehat{v}(t)$ satisfies $\widehat{v}(t_i) = v(t_i)$ for all i; i.e., the approximation **interpolates** the data.

Application (Polynomial Approximation). We can compute polynomial approximations in the sense of (6.19) very easily using MATLAB. For example, suppose we want to approximate $v(t) = \sin(2\pi t)$ for $t \in [0, 1]$ using a polynomial of degree 4. This corresponds to projecting v onto the 5-dimensional space spanned by $1, t, t^2, t^3, t^4$. Use the following commands.

```
t = linspace(0,1,30);
v = sin(2*pi*t);
```