Sampling Theory m-Files

Introduction

To better understand the frequency domain of sampling theory a collection of MATLAB m-files was created for plotting the frequency spectrum following ideal sampling. The tools described in this note are useful in analyzing the spectra following ideal sampling of real lowpass and bandpass signals, and a positive spectrum complex signal. In all cases the function provide a means to visualize the spectra of a particular bandlimited analog signal spectrum following uniform sampling.

Real Lowpass Signal

The function `lp_samp()` plots the sampled spectrum of a bandlimited real lowpass signal having the classical triangular shaped spectrum. The input parameters are defined as follows:

```matlab
function lp_samp(fb,fs,fmax,N)
%function lp_samp(fb,fs,fmax,N)
%******************************************************************************
%* Lowpass Sampling Theorem Plotting Function *
%******************************************************************************

% fb = spectrum lowpass bandwidth in Hz
% fs = sampling frequency in Hz
% fmax = plot over [-fmax,fmax]
% N = number of translates, N positive and N negative
%
% This function automatically creates a MATLAB plot using a predefined
% spectral shape taken from the function lp_tri( ).
% %
% Mark Wickert Fall 2000
```

- Choose the number of translates $N$ so that a nice symmetrical spectrum is obtained, but the graphics draw time is minimal.

As a specific example consider the following MATLAB command sequence and the corresponding output plot.

```matlab
lp_samp(10,25,100,10)
Current plot held
```
Real Bandpass Signal

The function `bp_samp()` plots the sampled spectrum of a bandlimited real bandpass signal having a right-triangular shaped spectrum. The input parameters are defined as follows:

```matlab
function bp_samp(fl,fh,fs,fmax,N,shape)
%function bp_samp(fl,fh,fs,fmax,N,shape)
%***********************************************
%* Bandpass Sampling Theorem Plotting Function *
%***********************************************
% fl = spectrum lower frequency in Hz
% fh = spectrum upper frequency in Hz
% fs = sampling frequency in Hz
% fmax = plot over [-fmax,fmax]
% N = number of translates, N positive and N negative
% shape = basic spectral shape: 0-default upslope tri, 1-downslope tri
%
% This function automatically creates a MATLAB plot using a predefined
% spectral shape taken from the function bp_tri( ).
%
% Mark Wickert Fall 2000
```
As a specific example consider the following MATLAB command sequence and the corresponding output plot.

```matlab
» bp_samp(10,20,25,80,10,0)
Current plot held
Current plot released
» print -tiff -deps bp_samp.eps
```

Positive Spectrum Bandpass Signal

The function `bp_sampc()` plots the sampled spectrum of a bandlimited complex bandpass signal having a right-triangular shaped spectrum. The input parameters are defined as follows:

```matlab
%*******************************************************
%* Complex Bandpass Sampling Theorem Plotting Function *
%*******************************************************
%
% fl = spectrum lower frequency in Hz
```
% fh = spectrum upper frequency in Hz
% fs = sampling frequency in Hz
% fmax = plot over [-fmax,fmax]
% N = number of translates, N positive and N negative
% shape = basic spectral shape: 0-default upslope tri, 1-downslope tri
%
% This function automatically creates a MATLAB plot using a predefined
% spectral shape taken from the function bp_tri_positive( ).
%
% Mark Wickert Fall 2000

As a specific example consider the following MATLAB command sequence and the corresponding output plot.

```matlab
» bp_sampc(10,20,30,100,10,0)
Current plot held
Current plot released
» print -tiff -deps bp_sampc.eps
```

A Complex Signal
since only a single-sided spectrum is present.

![Diagram of bandpass sampling theorem for a complex signal](image)

Bandpass Sampling Theorem for a Complex Signal: Blk = orig, dotted = translates

\[
X(f) = \begin{cases} 
1 & \text{shape} = 0 \\
\text{shape} = 1 
\end{cases}
\]

Sampling Rate

f Hz

0

\(-f_{max}\)

\(f_{max}\)

\(f_l\)

\(f_h\)

\(f_s\)

Spectrum Magnitude

0

0.2

0.4

0.6

0.8

1

-100

-80

-60

-40

-20

0

20

40

60

80

100

Frequency in Hz
function lp_samp(fb,fs,fmax,N)
%function lp_samp(fb,fs,fmax,N)
%******************************************************************************
%* Lowpass Sampling Theorem Plotting Function *
%******************************************************************************
%
% fb = spectrum lowpass bandwidth in Hz
% fs = sampling frequency in Hz
% fmax = plot over [-fmax,fmax]
% N = number of translates, N positive and N negative
%
% This function automatically creates a MATLAB plot using a predefined
% spectral shape taken from the function lp_tri( ).
%
% Mark Wickert Fall 2000

% define the plot interval
f = -fmax:fmax/200:fmax;

% plot the lowpass spectrum in black
plot(f,lp_tri(f,fb),'k');

% overlay positive and negative frequency translates
hold
for n = 1:N
    plot(f,lp_tri(f-n*fs,fb),':r');
    plot(f,lp_tri(f+n*fs,fb),':g');
end
hold

title('Lowpass Sampling Theorem for a Real Signal: Blk = orig, dotted = translates')
ylabel('Spectrum Magnitude')
xlabel('Frequency in Hz')

function x = lp_tri(f, fb)
% x = lp_tri(f, fb):

x = zeros(size(f));

for k=1:length(f)
    if abs(f(k)) <= fb
        x(k) = 1 - abs(f(k))/fb;
    end;
end
function bp_samp(fl,fh,fs,fmax,N,shape)
%function bp_samp(fl,fh,fs,fmax,N,shape)
%*****************************************************************************
%* Bandpass Sampling Theorem Plotting Function *
%*****************************************************************************
% fl = spectrum lower frequency in Hz
% fh = spectrum upper frequency in Hz
% fs = sampling frequency in Hz
% fmax = plot over [-fmax,fmax]
% N = number of translates, N positive and N negative
% shape = basic spectral shape: 0-default upslope tri, 1-downslope tri
%
% This function automatically creates a MATLAB plot using a predefined
% spectral shape taken from the function bp_tri( ).
%
% Mark Wickert Fall 2000

% use default shape for only 5 input arguments
if nargin == 5
    shape = 0;
end

% define the plot interval
f = -fmax:fmax/200:fmax;

% plot the bandpass spectrum
plot(f,bp_tri(f,fl,fh,shape),'k');

% overlay positive and negative frequency translates
hold
for n = 1:N
    plot(f,bp_tri(f-n*fs,fl,fh,shape),':r');
    plot(f,bp_tri(f+n*fs,fl,fh,shape),':g');
end
hold

title('Bandpass Sampling Theorem for a Real Signal: Blk = orig, dotted = translates')
ylabel('Spectrum Magnitude')
xlabel('Frequency in Hz')

function x = bp_tri(f, fl, fh, shape)
% x = bp_tri(f, fl, fh, shape):

x = zeros(size(f));
if shape == 1
    for k=1:length(f)
        if abs(f(k)) <= fh
            if abs(f(k)) >= fl
                x(k) = (fh - abs(f(k)))/(fh-fl);
            end;
        end;
    end
else
    for k=1:length(f)
        if abs(f(k)) <= fh
            if abs(f(k)) >= fl
                x(k) = (abs(f(k))-fl)/(fh-fl);
            end;
        end;
    end
end

function bp_sampc(fl, fh, fs, fmax, N, shape)
%function bp_sampc(fl, fh, fs, fmax, N, shape)
%*****************************************************************************
%* Complex Bandpass Sampling Theorem Plotting Function *
%*****************************************************************************
%
% fl = spectrum lower frequency in Hz
% fh = spectrum upper frequency in Hz
% fs = sampling frequency in Hz
% fmax = plot over [-fmax,fmax]
% N = number of translates, N positive and N negative
% shape = basic spectral shape: 0-default upslope tri, 1-downslope tri
% This function automatically creates a MATLAB plot using a predefined
% spectral shape taken from the function bp_tri_positive( ).
% Mark Wickert Fall 2000

% use default shape for only 5 input arguments
if nargin == 5
    shape = 0;
end

% define the plot interval
f = -fmax:fmax/200:fmax;

% plot the bandpass spectrum
plot(f, bp_tri_positive(f, fl, fh, shape), 'k');
% overlay positive and negative frequency translates

hold
for n = 1:N
    plot(f,bp_tri_positive(f-n*fs,fl,fh,shape),'r');
    plot(f,bp_tri_positive(f+n*fs,fl,fh,shape),'g');
end

hold

title('Bandpass Sampling Theorem for a Complex Signal: Blk = orig, dotted = translates')
ylabel('Spectrum Magnitude')
xlabel('Frequency in Hz')

function x = bp_tri_positive(f, fl, fh, shape)
% x = bp_tri(f, fl, fh, shape):

x = zeros(size(f));

if shape == 1
    for k=1:length(f)
        if f(k) <= fh
            if f(k) >= fl
                x(k) = (fh-abs(f(k)))/(fh-fl);
            end;
        end;
    end;
else
    for k=1:length(f)
        if f(k) <= fh
            if f(k) >= fl
                x(k) = (abs(f(k))-fl)/(fh-fl);
            end;
        end;
    end;
end