Subject: Using IDL to make a signal filter
Posted by ICBM0926 on Mon, 14 Jul 2008 11:08:15 GMT
View Forum Message <> Reply to Message

I have an 1D vector data in IDL from an analytical laser formula which
contains 2 laser frequencies. I wrote a program trying to filter my 1D
vector and get the waveform of one of the frequencies. I used 1D FFT
and a mask
function(step function). I applied the mask function to the frequency
domain data. I've covered both positive and negative frequencies. I
did inverse 1d FFT to retrieve the signal. I found that
the amplitude of the signal is only half as it should be. Could
anybody tell me what went wrong?

Subject: Re: Using IDL to make a signal filter
Posted by Kenneth P. Bowman on Mon, 14 Jul 2008 15:06:45 GMT
View Forum Message <> Reply to Message

In article
<6961630e-ccbc-4e50-82c4-9124498ec7a0@s50g2000hsb.googlegroups.com>,
 ICBM0926 <ICBM0926@gmail.com> wrote:

> I have an 1D vector data in IDL from an analytical laser formula which
> contains 2 laser frequencies. I wrote a program trying to filter my 1D
> vector and get the waveform of one of the frequencies. I used 1D FFT
> and a mask
> function(step function). I applied the mask function to the frequency
> domain data. I've covered both positive and negative frequencies. I
> did inverse 1d FFT to retrieve the signal. I found that
> the amplitude of the signal is only half as it should be. Could
> anybody tell me what went wrong?

It sounds like you are doing things right, but it is easy to
make mistakes using FFTs.

You might want to look at the chapter on FFTs in my book
(http://tinyurl.com/zavp9), or create an artificial input
data set where you know exactly what the answer is and
use that to test your software.

Ken Bowman

Subject: Re: Using IDL to make a signal filter
Posted by Wox on Mon, 14 Jul 2008 17:33:06 GMT
View Forum Message <> Reply to Message

On Mon, 14 Jul 2008 04:08:15 -0700 (PDT), ICBM0926
<ICBM0926@gmail.com> wrote:

> I have an 1D vector data in IDL from an analytical laser formula which
> contains 2 laser frequencies. I wrote a program trying to filter my 1D
> vector and get the waveform of one of the frequencies. I used 1D FFT
> and a mask
> function(step function). I applied the mask function to the frequency
> domain data. I've covered both positive and negative frequencies. I
> did inverse 1d FFT to retrieve the signal. I found that
> the amplitude of the signal is only half as it should be. Could
> anybody tell me what went wrong?

In the example below, you see first a sum of three sinus waves, all
with amplitude 1. After filtering two of them out, 1 sinus is
remaining with amplitude 1, not 0.5 as in your case. Does this help?

```
pro test

; Time domain
freq1=2.
freq2=3.
freq3=4.
dtime=0.05
ntime=200

time=dtime*findgen(ntime)
 signal=sin(2*!pi*freq1*time)+sin(2*!pi*freq2*time)+sin(2*!pi *freq3*time)

; Frequency domain
nfreq=ntime/2+1
freq=findgen(nfreq)/(dtime*ntime)
freq=[freq,reverse(-freq[1:nfreq-1-(~(ntime mod 2))])]
fsignal=fft(signal,-1)

; Frequency domain filter
f_low = 0
f_high = 2.5
steep=20.
freqfilter= 1./(1.+(freq/f_high)^steep)
fsignalfilt=fsignal*freqfilter

; Back to time domain
signalfilt=fft(fsignalfilt,1)
```

```
; Plot
window,0
!P.MULTI=[0,2,2]
 plot,freq[0:nfreq-1],abs(fsignal[0:nfreq-1])^2,xtitle='frequ ency',ytitle='spectrum',title='Original
Spectrum'
 plot,freq[0:nfreq-1],freqfilter[0:nfreq-1],xtitle='frequency ',ytitle='filter',title='Filter'
 plot,freq[0:nfreq-1],abs(fsignalfilt[0:nfreq-1])^2,xtitle='f requency',ytitle='filtered
spectrum',title='Filtered spectrum'

window,1
!P.MULTI=[0,1,2]
plot,time,signal,xtitle='time',ytitle='signal'
plot,time,signalfilt,xtitle='time',ytitle='filtered signal'
end
```

Subject: Re: Using IDL to make a signal filter
Posted by Wox on Tue, 15 Jul 2008 09:32:48 GMT

View Forum Message <> Reply to Message

On Mon, 14 Jul 2008 23:21:33 -0700 (PDT), ICBM0926
<ICBM0926@gmail.com> wrote:

<snip>
```
> kxcen=7.757e+6
> kxband=7.757e+6
> NyKx=2.d*3.1415926/2.d*dx
> mkx=dindgen(2048)*dkx
> mask=dindgen(2048)
>  mask=1.d*(exp(-((mkx-kxcen)/(kxband/2.d))^8.d)+exp(-((mkx-(2 .d*NyKx-
> kxcen))/(kxband/2.d))^8.d))
```
<snip>

The filter above removes all negative frequencies. This is why you end
up with half the amplitude. Check fft documentation: negative
frequencies in the second half, so the second half of the filter (in
1/m space) should be a (reversed) copy of the first half. See below
how it's done. I got confused by your notation, so I transformed
things a bit.

```
pro onedanalytic
; constants
lambda=810.0e-9
c=299792458.d
n=2049
dx=800.0e-9/32.d
E0=1.d
```

```
; (m)-domain
k=2*!dpi/lambda
x=dx*(dindgen(n)-n/2)
E1=E0*exp(-(x/(3.822e-14*c))^2)*cos(k*x)
E2=E0*exp(-(x/(3.822e-14*c))^2)*cos(2.d*k*x)
E=E1+E2

; (1/m)-domain
Efft=fft(E,/double)
nkx=n/2+1
kx=dindgen(nkx)/(n*dx)
kx=[kx,reverse(-kx[1:nkx-1-(~(n mod 2))])]

; filter
 mask=exp(-(2*(kx*lambda-1))^8.d)+exp(-(2*((kx-dx)*lambda+1)) ^8.d)
Efft*=mask
ReE=fft(Efft,/inverse)

; plot
window,0
!P.Multi=[0,2,2]
plot,x,E1,title='E1'
plot,kx,mask,title='Filter: remove E2'
plot,x,ReE,title='Filtered E1+E2 = E1'
end;pro onedanalytic
```