Subject: Re: How to perform the 1-D signal filter?
Posted by duxiyu@gmail.com on Sat, 02 Feb 2008 15:14:05 GMT
View Forum Message <> Reply to Message

On Feb 2, 1:16 am, Wox <nom...@hotmail.com> wrote:
> On Fri, 1 Feb 2008 07:16:26 -0700, David Fanning <n...@dfanning.com>
> wrote:
>
>
>
>> Wox writes:
>
>>> Example below filters in time or frequency domain:
>
>>> ; Time domain
>>> freq1=2.
>>> freq2=3.
>>> freq3=4.
>>> dtime=0.1
>>> ntime=1000
>
>>> time=dtime*findgen(ntime)
>>> signal=sin(2*!pi*freq1*time)+sin(2*!pi*freq2*time)+sin(2*!pi *freq3*time)
>
>>> ; Time domain Filter
>>> f_low = 0
>>> f_high = 2.5
>>> timefilter = DIGITAL_FILTER(f_low*2*dtime, f_high*2*dtime, 50.,40)
>>> signal=convol(signal,timefilter)
>
>>> ; Frequency domain
>>> nfreq=ntime/2+1
>
>>> freq=findgen(nfreq)/(dtime*ntime)
>>> fsignal=fft(signal)
>
>>> ; Frequency domain filter (instead of time domain filter)
>>> if n_elements(timefilter) eq 0 then begin
>>>   steep=20.
>>>   freqfilter= 1./(1.+(freq/f_high)^steep)
>>>   fsignal*=freqfilter
>>> endif
>
>>> plot,freq,abs(fsignal[0:nfreq-1])^2,xtitle='frequency',ytitl e='spectrum'
>
>> Wonderful example, but I'm trying to understand this whole
>> subject. Do you think you could flush this out with a little
>> explanation of what you are doing and why you choose the terms

>> you use, etc.? What kind of frequency filter are you constructing
>> here? I don't necessarily see it doing any filtering of the signal,
>> at least if I pass it the original signal, rather than the signal
>> that had already been filtered in the time domain, as written
>> in your example.
>
>> Cheers,
>
>> Confused
>
> Ok, sorry for the confusion, but I was just illustrating that you can
> do the same filtering in the frequency domain as in the time domain.
> You do one or the other, not both at the same time. Btw, convolution
> in one domain becomes multiplication in the other:
>
> filtered = signal "convol" filter
> fft(filtered) = fft(signal) x fft(filter)
>
> But I guess you already knew all this.
>
> The filter used is the Kaiser-Bessel filter. At least I think
> digital_filter is using this. For the filter I constructed in the
> fourier domain, I'm not quit sure whether it is really identical to
> the KB filter, but if you plot it, it looks like a nice lowpass filter
> to me :-).
>
> I'm just typing this in a hurry... Did I answer your questions?

Thank you again for your detailed explaination.
Now I am clear about it.

Best regards,
Du