
Subject: Re: How to perform the 1-D signal filter?
Posted by duxiyu@gmail.com on Fri, 01 Feb 2008 14:54:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thank you very much.
But I still have some questions about the parameters in the procedure
Digital_Filter.

Result = DIGITAL_FILTER(Flow, Fhigh, A, Nterms [, /DOUBLE])

In the example, you set A=50 and Nterms=40 and I do not know where
their values come from and how I get them.

Seondly, I want to know the relation between the filter in time and
frequency domain.

if the following command is excuted, I think newfsignal is exactly
equal to fft(newsignal), isn't it?

```
"newsignal=convol(signal,timefilter)
fsignal=fft(signal)
newfsignal=fsignal*freqfilter"
```

And I also do not konw why you set steep=20.

Du

On Feb 1, 9:27 pm, Wox <nom...@hotmail.com> wrote:

> On Fri, 1 Feb 2008 02:20:52 -0800 (PST), "dux...@gmail.com"

>

>

>

> <dux...@gmail.com> wrote:

>> Dear all,

>

>> Here I give a signal example and hope somone can show me how to
>> perform the frequency filter on it.

>

>> ;creat a signal data with two peaks in frequency domain at 2 and 3 Hz.

>> t=findgen(1000)/10.

>> data=sin(2*!pi*2*t)+sin(2*!pi*3*t)

>

>> freq=findgen(501)/100.

>> v=fft(data)

>> plot,freq,abs(v[0:500])^2,xtitle='frequency',ytitle='spectru m'

>

>> I want to filter the signal with the frequency higher than 2.5 Hz. How
>> do I do this?

```

>
>> I have read the help files about Digital_Filter and Convolve, but I do
>> not know how to select the parameters for Signal_Filter.
>
>> Du
>
> 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=convolve(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',ytitle='spectrum'

```
