Hilbert IIR filter implementation - praveen - 07:12 29-07-03
Hello,

I wanted to know how to implement Hilbert transform using IIR filter. Any reference or article or suggestion will be great. I wanted to implement it on a DSP processor. Hardware structure, filter coefficient ?

waiting for reply
With regards
praveen

Re: Hilbert IIR filter implementation - Vladimir Vassilevsky - 09:13 29-07-03
Hello Praveen,

The hardware structure for IIR Hilbert transformer is standard cascaded allpass biquad sections. The number of sections depends on what kind of frequency range, phase, and gain accuracy do you need. The coefficients are usually optimized by Chebyshev approximation. For example, for voice band IIR Hilbert with 1% accuracy you need 6-th order filter at least.

Vladimir Vassilevsky,
Ph.D. DSP and Mixed Signal Design Consultant
http://www.abvolt.com

Re: Hilbert IIR filter implementation - Jerry Avins - 10:09 29-07-03
Vladimir,

For analog work, it is customary to use a pair of third-order allpasses whose outputs have relative quadrature but are not specified relative to the input phase. How is the IIR case handled, and how is identical delay for the two channels achieved?

Jerry

Re: Hilbert IIR filter implementation - Peter Brackett K1PO - 10:59 29-07-03
Praveen:

Often one does not really need a Hilbert Transformer, what one requires is a method of producing two sequences which are Hilbert Transforms of each other from a single input sequence. i.e. a pair of networks having approximately a 90 degree phase shift between there outputs with no amplitude difference.

If you don't care about phase linearity it's easy to generate a complex IIR filter whose two output sequences are approximate Hilbert Transforms of each other to whatever degree of precision you require in your approximation. And the approximation can be couched in terms of dB [or Np] of suppression of undesired responses. Neat, eh?

If you do require phase linearity as well as the requisite 90 degree phase difference between the outputs then you must also design an all-pass phase equalizer to linearize the phases, but often that is not required. If it is required then it is still quite possible if you have the capability to desing group delay phase equalizers. This latter all-pass group delay equalizer design is not a widespread method in common dsp toolkits.

To begin the design of a pair of networks which have a ninety degree phase shift between their outputs, one simply designs an appropriate complex IIR filter [i.e. a filter without even symmetry about zero frequency in its' attenuation characteristic] having the desired degree of approximation to the Hilbert Transform magnitude.

One method, perhaps the simplest of several methods, to accomplish this is to first design an appropriate "prototype" "real" low pass IIR filter to pass the desired bandwidth over which one desires the ninety degree phase [Hilbert Transform] relationship using say the bilinear transformation technique.

The "prototype" low-pass filter is a "real" filter which has real coefficients, and except for any single real pole or zero in the case of odd order, all other poles and zeros will occur in complex conjugate pairs, and it will have real sequences as inputs and outputs.

Then, having the z-plane pole and zero factors for this real low pass prototype filter in hand, simply rotate the pole zero constellation about zero in [say] the positive direction about z = zero in the z-plane to place the lower transition band in the appropriate location around zero frequency. You will then have a pole zero constellation where the poles and zeros do not occur in complex conjugate pairs. i.e. this rotated low pass will now be a complex filter with complex coefficients and complex sequence outputs, i.e. two sequences as outputs, a real sequence and an imaginary sequence. The real and imaginary output sequences of this complex rotated low pass filter will then be approximate Hilbert Transforms of each other to whatever degree of approximation was set up in

the low pass prototype. Such IIR "Hilbert Transformers" are very efficient [economical in multiply-adds and memory requirements] when compared to the FIR Hilbert Transformer techniques.
Peter
Consultant
Indialantic By-the-Sea, FL.

Re: Hilbert IIR filter implementation - Olli Niemitalo - 04:58 30-07-03
Here is described a very effective structure for an IIR allpass pair:
(old link: http://www.biochem.oulu.fi/~oniemita/dsp/hilbert/ , replaced by:)
http://yehar.com/ViewHome.pl?page=dsp/hilbert/
Juicy specs of the example filter:
* Takes 8 multiplications per input sample
* Phase difference 90 +/- 0.7 degrees over a band of width 0.998*Nyquist
Unfortunately, I haven't written a software to calculate the coefficients
to a specification. Someone could do it, for sure!
-olli

Re: Hilbert IIR filter implementation - Jerry Avins - 08:28 30-07-03
Thanks. That's very nice work and presentation. It is a digital
equivalent of the two-channel, third-order analog quadrature circuit I'm
familiar with. When Vladimir wrote "... for voice band IIR Hilbert with
1% accuracy you need 6-th order filter at least" I assumed he meant a
single sixth-order filter. I was probably wrong about that.
Jerry

Re: Hilbert IIR filter implementation - Jerry Avins - 09:35 31-07-03
Peter,
That's neat and almost intuitive. Please write it up with an example for
http://www.dspguru.com I think it belongs there.
Jerry

Re: Hilbert IIR filter implementation - Al Clark - 10:08 31-07-03
Peter,
I think I'm still missing something. Doesn't rotating the low pass
prototype just create a bandpass filter? How do you structure the filter
to get the hilbert pair? Maybe I need pictures or coffee. Can you create
a simple example?
Al Clark Danville Signal Processing, Inc. Purveyors of Fine DSP Hardware and other Cool Stuff
Available at **http://www.danvillesignal.com**

Re: Hilbert IIR filter implementation - Matt Boytim - 10:48 31-07-03
This may not be exactly what you're looking for, but is is certainly
related and interesting none the less.
"A Simple Method for Sampling In-Phase and Quadrature Components",
Charles M. Rader, Lincoln Labratory, IEEE Transactions on something
(sorry, that's all I can read on my copy), 1984, pages 821-824
Matt