Fixed point Toolbox for scilab users guide

S. Ygorra

stephane.ygorra@u-bordeaux1.fr

presentation

This toolbox is a integer number *C, and VHDL* source code generator for *iir* filters.
It aims at simplifying the conversion task from an ideal transfer function, to a working C source code, with integer numbers, and proper scaling for overflow avoidance, and noise minimization.

Moreover the author propose a modified operator for the implementation, in place of the classical delay operator, wich greatly improve performances (noise level, and frequential response sensitivity to coefficients) in the case of low frequency dynamics of the filter...

Contents

The toolbox contains a set of functions wich allow one to handle *iir* transfer functions as a cascade or a sum of second order (or less) filters, called cels.

It also contains functions for norms computation on such cels, useful for scaling and noise analysis.

The scaling and analysis can be performed for the classical following forms: ***df1***,***df2***,***df1t***,***df2t***, optimal noise ***state-space.***

For each form one can use a particular operator, different of the classical delay, which dramatically enhance performances (frequency response sensitivity and quantization noise level) in the case where cut off frequencies are litlle in regard of the sample frequencies.

the toolbox also contains functions wich allow one to define a filter from a given *template (low pass,bandpass,high pass, band stop*, eventually as 2 paralell allpass cels .

limitation

for instance **the c and vhdl source code generation only works for *df2* or *state-space* forms**...

USAGE

The easier way to use this toolbox consists of reading and further adapting the working examples , in the corresponding (ve carefull, because of very low transient, the low_pass example has a long computational time)

- ***band_pass_iir_example.sce, for a band_pass iir filter***
- ***band_stop_iir_example.sce, for a band_stop iir filter***
- ***high_pass_iir_example.sce, for a high_pass iir filter***
- ***high_pass_iir_example.sce, for a high_pass iir filter***
- ***low_pass_iir_example.sce, for a low_pass iir filter***

- arbitrary_transfer_function_example.sce , for an arbitrary transfer function

Each example generate the .h, *.c* and .vhd code source files of the corresponding *iir* filter in the same directory (???.c for c source files, ???.vhd for vhdl filter source file).

User can edit, and adapt this file to his own problem.

typically parameters have reasonable default values. The user may however modify the used operator, or implementation structure, and retain the best (keep the parameters wich generate minimum noise level)

Be careful to the declaration of integers in the generated c program: it is correct only for 16/32 bits arithmetics.
For example : if one use 24 bits arithmetic,

the types ***int_24*** and ***int_48*** will not correspond to the good sizes in the C program, and the user has to modify them manually.

the subdirectory generated_examples contains a complete application, corresponding to files generated by the examples...