The Delta-Sigma Toolbox

- Matlab functions for the design, simulation and realization of delta-sigma modulators.
- Developed at OSU by Dr. Schreier
- Download sites:

  `ftp://ftp.mathworks.com/pub/contrib/v5/control/delsig.zip`

  `http://www.mathworks.com/matlabcentral/fileexchange/Files.jsp?fileId=19`

- Documentation is in file `DelSig.pdf`

1

Installation

- Unzip file to `~/delsig`
- In non-existent, create directory `~/matlab`
- Copy file `mexopts.sh` into `~/matlab`
- Edit file `~/matlab/startup.m` to include:

```
path(path,'<full-path-to-home>/delsig');
```

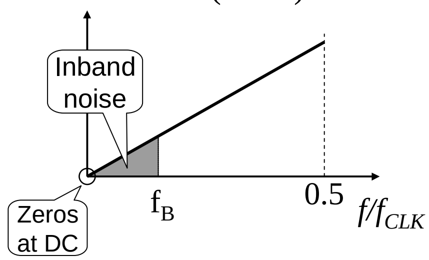
- Compile C functions using `mex file.c`
- Start matlab.

2

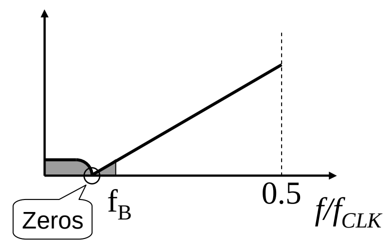
Noise Transfer Function

Simple NTF

NTF = $(1 - z^{-1})^L$

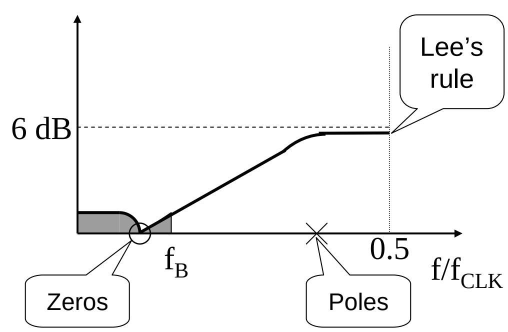


NTF with optimized zeros

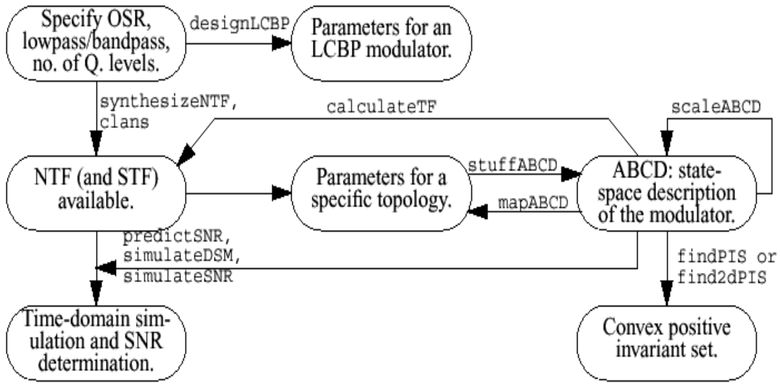


Even better

- Less in-band noise (optimized zeros)
- Meets stability requirements (Lee’s rule)



Design Flow



5

Key Functions (1)

- Modulator synthesis and simulation

`synthesizeNTF`	Noise transfer function (NTF) synthesis.
`clans`	Closed-loop analysis of noise shapers (NTF synthesis for multi-bit modulators).
`simulateDSM`	Simulate a delta-sigma modulator using either its NTF or its state-space description.
`simulateSNR`	Use simulateDSM to simulate a DSM with sine wave inputs of varying amplitudes and then determine the SNR for each.
`predictSNR`	SNR prediction for binary modulators (Ardalan & Paulos method)

6

Key Functions (2)

- Modulator Realization:

`realizeNTF`	Compute coefficients for a particular modulator topology.
`stuffABCD`	Create state-space description of a modulator given the coefficients for a particular topology.
`mapABCD`	Convert state-space description back to coefficients.
`scaleABCD`	Perform dynamic-range scaling.

7

Key Functions (3)

- Demonstrations and Examples:

`dsdemo1`	Synthesize 5th-order lowpass and 8th-order bandpass NTF.
`dsdemo2`	Time-domain simulation and SNR calculation.
`dsdemo3`	Modulator realization and dynamic range scaling.
`dsdemo4`	Continuous-time bandpass modulator design using LC tanks.
`dsdemo5`	Find positively-invariant sets for second and third-order modulators.
`dsdemo6`	Simulate element selection logic of mismatch-shaping DAC.
`dsdemo7`	Design hardware-efficient halfband filter.
`dsexample1`	Discrete-time lowpass modulator.
`dsexample2`	Discrete-time bandpass modulator.

8

Design Example

- Delta-sigma ADC for digital audio applications:
  - – SNR > 98 dB (16-bit resolution)
  - – Output data rate: 44.1 KS/s
  - – Use 1-bit quantizer
  - – Second-order noise transfer function (NTF)
- Quick lookup shows that oversampling ratio needs to be > 128.

  => Select OSR = 256 , and check later.
- Start code with:

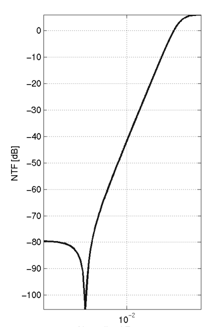
```
order = 2; OSR = 256; nlev = 2;
```

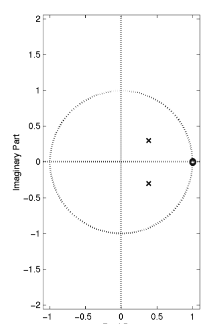
9

Synthesize Noise Transfer Function

```
>> ntf = synthesizeNTF(order,OSR,opt)

Zero/pole/gain:
     (z^2 - 2z + 1)
------------------------
(z^2 - 0.7639z + 0.2361)

>> [NUM,DEN] = tfdata(ntf,'v')

NUM =
    1.0000   -1.9999    1.0000
DEN =
    1.0000   -0.7639    0.2361
```

opt=1
Optimize zeros

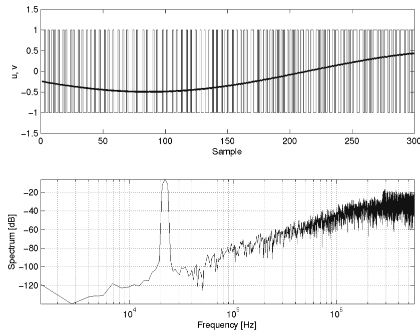


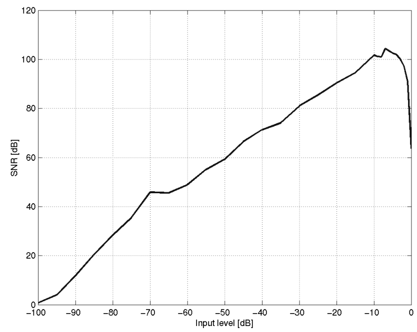


10

Modulator Simulation

```
N=8192; f=N/(2*OSR); u=0.5*sin(2*pi*f*[0:N-1]/N);
[v,xn,xmax,y]=simulateDSM(u,ntf,nlev);
fB = 1/(8*OSR); Au = [-100:5:-10 -9:1:0];
[snr,Au] = simulateSNR(ntf,OSR,Au,0,nlev,fB,14);
```





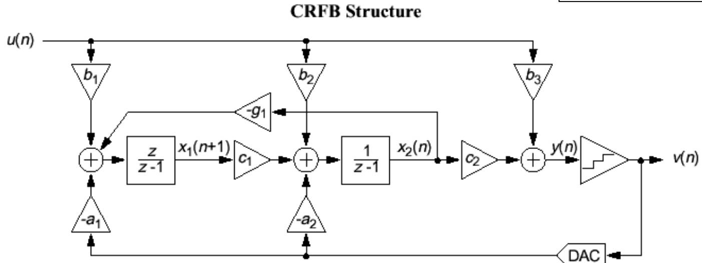
11

Calculate Coefficients

```
>> form = 'CRFB'; [a,g,b,c] = realizeNTF(ntf,form,1)
a = 0.4721    0.7639
g = 5.0199e-05
b = 0.4721    0.7639    1.0000
c = 1     1
```

Other forms:
- CRFF
- CIFB
- CIFF, etc

CRFB Structure



Even Order

12

Scaling

- Scale coefficients for optimum dynamic range:

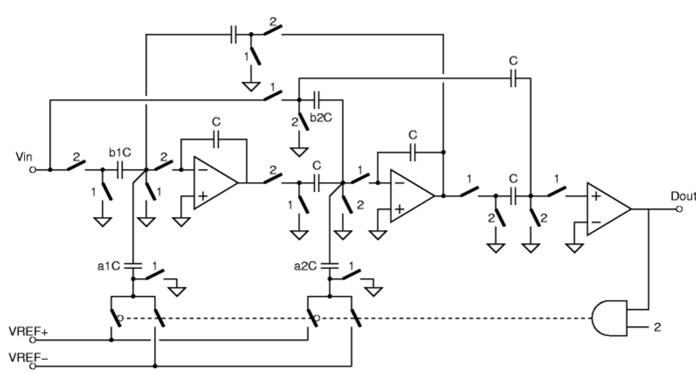
```
ABCD = stuffABCD(a,g,b,c,form)
[ABCDs,umax]=scaleABCD(ABCD,nlev,0,1,7)
umax = 0.9000
ABCDs = 1.0000  -0.0003   0.7286   -0.7286
        0.1530   0.9999   0.2919   -0.2919
             0   4.2349   1.0000         0

[a,g,b,c] = mapABCD(ABCDs,form)
a = 0.7286    0.1804
g = 3.2808e-04
b = 0.7286    0.1804    1.0000
c = 0.1530    4.2349
```

threshold for judging stability

signal limit

13

Circuit Level Implementation



$$f_{CLK} = 2 \times f_B \times OSR = 11.29 \text{ MHz}$$

14