5. Scaling, Product Round-Off Noise, Pole-Zero Ordering and Limit Cycles

5.1 Scaling of a Cascade Structure using a Structure Model

Verify the scaling of the 3rd-order lowpass filter in cascade form considered in the lecture slides (see fig. 5.1).

To this end, write a function ***cascadescale*** and a script to scale the cascade structure using the function ***cascadescale*** step-by-step, i.e. node by node.

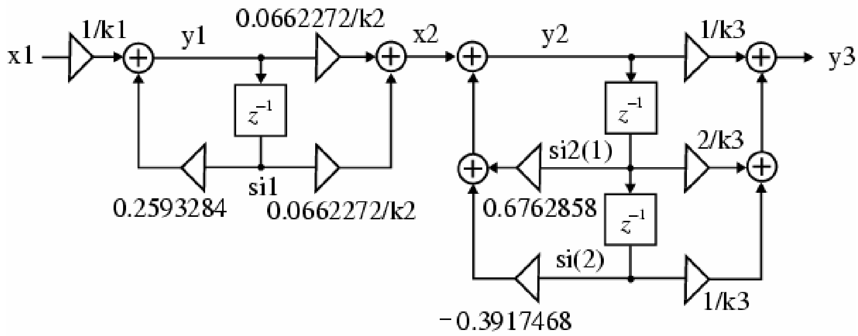

Fig. 5.1: Filter structure in cascade form with 3 scaling constants

5.2 Scaling of a Cascade Structure using the scale Method

Scale the second-order sections of the above cascade structure using the method `scale` available for filter objects.

Compare your results with that obtained in 5.1.

Hint: `realizemdl` can be used to generate a SIMULINK model of the scaled filter.

5.3 Product Round-off Noise of a Cascade Structure

Verify the product round-off noise power of the 3rd-order unscaled filter cascade structure using two different orderings considered in the lecture slides.

Use the function `noisepsd` to determine the power spectral density of the noise power, `avgpower` and `pow2db` to calculate the average product round-off noise power in dB.

5.4 Optimum Pole-Zero Pairing and Ordering of a Cascade Filter Structure

Find the optimum pole-zero pairing and ordering of a fixed-point IIR filter in cascade structure to minimize the product round-off noise power.

The filter specifications are as follows:

- $R_p = 1$ dB
- $R_s = 60$ dB
- $\omega_p = 0.5\,\pi$
- $\omega_s = 0.6\,\pi$

Use the commands

```
d = fdesign.lowpass(0.5,0.6,1,60);
hd = design(d,'ellip');
```

to design the filter.

Change the option `'Arithmetic'` to `'fixed'`.

Scale the second-order sections using the method `scale` and find the optimal pole-zero pairing and ordering using the method `reorder` for L_2- and L_∞-Scaling.

Hint: The product round-off noise power spectrum can be visualized e.g. by using the command `fvtool(hd,'analysis','noisepower');`

5.5 Granular Limit Cycles

Verify the granular limit cycles of the first-order IIR filter considered in the lecture slides.

To this end, write a MATLAB script using the function ***twosquant*** (see lab 4.1).

Hint: You can use the debugger to track the quantization cycles.

5.6 Overflow Limit Cycles

Verify the overflow limit cycles of the second-order IIR digital filter considered in the lecture slides. To this end, write a MATLAB script using the function ***twosquant***.

Hint: Note the overflow warning messages in the command window.

Make use of the MATLAB debugger to track the overflows and the overflow handling.

Verify that the direct-form structure has no overflow limit cycles if the filter coefficients are related by $|\alpha_1| + |\alpha_2| < 1$.