# Effects of finite word length representation of the filter coefficients

**Lab 10, SDP**

## Table of contents



## 1 Objective

Students should observe the effects of having fixed point coefficients in a digital filter, and be able to mitigate the effects.

## 2 Theoretical notions

### 2.1 Binary representation of fractionary numbers

TBD

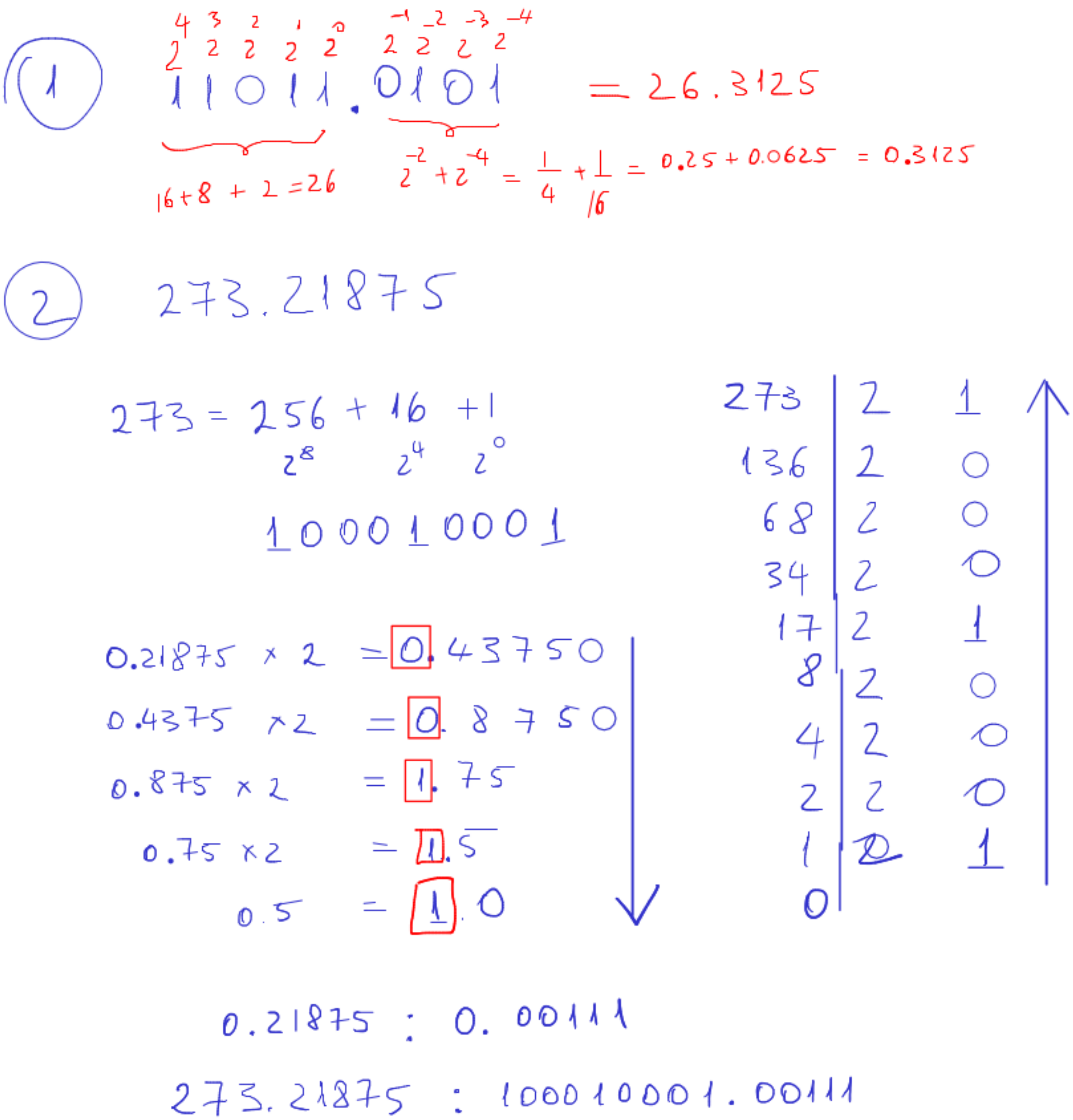


Figure 1: Binary representation of fractionary numbers

### 2.2 Filter implementation with second order sections (SOS)

Implementing filters with second order sections means a **series implementation**, as a sequence of sub-filters of order 2.

$$H(z) = H_1(z) \cdot ... \cdot H_n(z) \cdot Gain$$

where each $H_i(z)$ has order 2:

$$H_i(z) = \frac{b_0^{(i)} + b_1^{(i)} z^{-1} + b_2^{(i)} z^{-2}}{1 + a_1^{(i)} z^{-1} + a_2^{(i)} z^{-2}}$$

Example:

https://www.ni.com/docs/en-US/bundle/labview-digital-filter-design-toolkit-api-ref/page/lvdfdtconcepts/iir_sos_specs.html

## 3 Theoretical exercises

1. Convert the following binary number to the decimal value:

   11011.0101

2. Convert in binary fixed point format (signed, 6 integer bits, 6 fractionary bits - 1S6I6F the following numbers:

   273.21875

3. Convert in binary fixed point format (signed, 6 integer bits, 6 fractionary bits - 1S6I6F the following negative numbers. Negative numbers shall be represented in sign-value, 1's complement (C1) and 2's complement (C2) formats.

   a. -22
   b. -22.21875

4. Quantize the samples $x_1 = 0.42625$ and $x_2 = -0.4333$ the fixed point format 1S0I4F via:

   a. Truncation
   b. Rounding
   c. Truncation in absolute value

   The negative values shall be represented in C2 format.

## 4 Practical exercises

1. Use Matlab's `fdatool` to design a low-pass IIR filter, Butterworth type, order 4, with cutoff frequency of 4kHz for a sampling frequency of 44.1kHz. Export the coefficients of the direct form II implementation to the Matlab Workspace as `b` and `a`.

2. In Matlab's `fdatool`, set the filter arithmetic to "fixed-point arithmetic" and modify the following:

   a. Set the format to fixed point 1S2I7F. How does the filter's transfer function change?
   b. Increase the number of bits in the fractionary part. How does the filter's transfer function change? For what number of bits do you consider the errors to be negligible?
   c. Export the coefficients of the direct form II implementation to Matlab's Workspace as `b1` and `a1`.

3. Repeat the preceding exercise with the filter implemented in series form ("Second-Order-Sections"). Which implementation has smallest errors? Export the coefficients to Matlab's Workspace as `b2` and `a2`.

4. Load the `mtlb` audio signal from Matlab (`load mtlb;`). Use `filter()` to filter the signal with the original filter (`b` and `a`) and with the fixed point coefficients (`b1` and `a1`).

   a. Plot the difference between the two filtered signals.
   b. Plot the histogram of the difference signal. What is it's shape? What is the average value of the errors?

## 5 Final questions

1. TBD