Signal Processing Systems (521279S), Fall 2025
Part 2 : Implementing fixed-point filtering
Design tasks, deadline for return Thu 13.11.2025 23:59

Task 1. [Matlab's Fixed Point Toolbox (see Sec. 1 in **intro2.pdf**), 1p] Pick the parameters from Table below and answer to the following question using the toolbox. Return also the Matlab script that you used in Moodle.

Question: What bit string corresponds to the value of the expression $c \times d + c - d$ in fixed-point format sp_o.n_o if c is approximated by the format sp_c.n_c and d by the format sp_d.n_d first? In the computations, use the output format sp_o.n_o also for intermediate results, that is, for $c \times d$ and $c \times d + c$. In approximation of the inputs c and d, use rounding towards the nearest. In computations, use rounding towards zero.

Group	c	d	p_c	n_c	p_d	n_d	p_o	n_o
1	-2.177	2.314	8	3	6	2	11	3
2	-2.851	3.402	7	2	8	3	12	3
3	-6.296	2.156	8	3	7	3	12	4
4	-13.305	3.896	9	3	7	3	13	4
5	3.346	-12.564	7	2	8	2	12	2
6	-6.174	3.776	9	3	7	3	13	4
7	5.852	-7.306	8	2	7	2	12	2
8	3.553	-1.290	8	3	7	2	12	3
9	-14.353	1.654	8	2	7	3	12	3
10	-1.799	1.523	7	3	8	3	12	4
11	3.849	-1.298	9	2	6	2	12	2
12	1.359	-1.652	7	2	6	3	10	3
13	-2.842	14.317	7	2	8	2	12	2
14	-24.598	13.306	9	2	8	2	14	2
15	-6.925	1.648	8	3	6	3	11	4
16	28.346	-3.820	9	2	7	2	13	2
17	-6.353	1.784	9	2	7	3	13	3
18	4.596	-3.311	7	2	8	2	12	2
19	9.594	-2.524	9	2	8	3	14	3
20	-15.853	2.810	8	2	6	2	11	2
21	-10.299	1.395	9	3	7	3	13	4
22	2.302	-1.779	9	3	7	3	13	4
23	5.553	-7.553	8	3	7	2	12	3
24	5.927	-6.541	9	3	7	2	13	3
25	-3.924	1.776	8	3	6	3	11	4
26	6.343	-7.771	7	2	8	3	12	3
27	-12.551	7.157	9	3	8	3	14	4
28	12.850	-2.805	8	2	7	2	12	2
29	-18.599	3.657	9	2	8	3	14	3
30	13.604	-1.780	8	2	6	3	11	3

Task 2. [Filter design (Sec. 3), 1.5p] A list of filter specifications is given on the last page of this handout. Take the specification assigned to your group.

(a) Design a fixed-point FIR filter that satisfies the specification requirements. As explained in Chapter 3 of the background material, you can do it in the following steps:

1. Design a floating-point FIR filter using `filterDesigner` or `fdatool`.
2. Quantize the coefficients provided by the tool.
3. Check out the response of the quantized filter using `fvtool`.

Steps 2-3 are repeated as many times as needed to find a sufficient quantization. What is the number of coefficients (L) in that filter and what is the $\mathtt{s}p_c.n_c$ format of the quantized coefficients?

(b) Chapter 3.3. in the material explains how the filter specification can be tightened to find a proper fixed-point filter meeting the original specification. Try out this approach:

1. Using the values in your specification, decrease the passband ripple by 10-20% and increase the stopband attenuation by 5-10 dB. Design a floating-point FIR filter with these specifications.
2. Quantize the coefficients provided by the tool.
3. Check out the response of the quantized filter using `fvtool`.

Repeat steps 2-3 until the filter performance is close to the original specification. What is the number of coefficients (L) in the filter? What is the $\mathtt{s}p_c.n_c$ format of suitable quantized coefficients?

(c) Compare the filters you found in subtasks (a) and (b). Compute for both the number of bits required for storing the coefficients, N_{tot}:

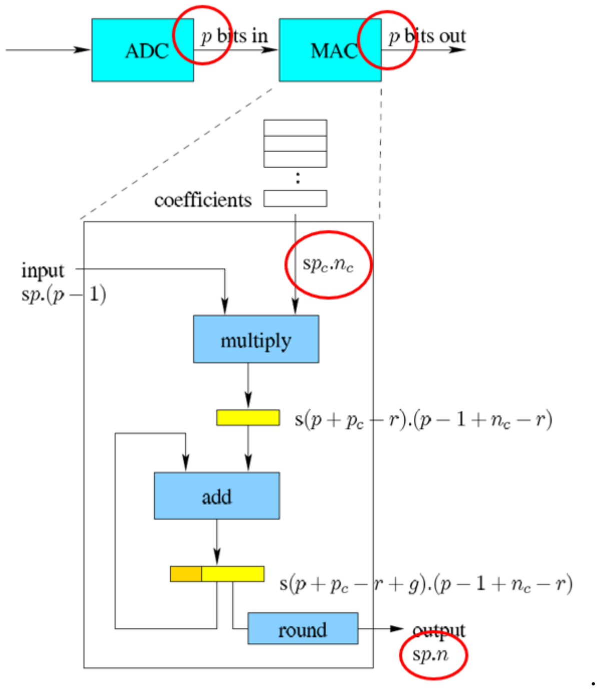
$$N_{\text{tot}} = Lp_c,$$

where L is the filter length and p_c is the coefficient word length. Select the filter for which N_{tot} is lower and use it in Task 3.

(d) For the filter you selected in (c), include in the report `fvtool` images to show that it works in passband/stopband.

Task 3. [Roundoff errors (Sec. 4), 1.5p] The filter you designed in Task 2 is implemented on a MAC unit. The Matlab function **FIR_Experiment.m** (in **dtask2.zip**) is used to simulate filtering. Its operation is described in Chapter 4 of the background material. Note that you do not have to make any changes to this function.

(a) Experiments are done using `FIR_Experiment`, which takes as input argument a bit specification `bitspec=[` p, p_c, n_c, r, g, n `]`, whose values control the data path as shown below.



The word length for signal input and output, p, is provided in the last column of the parameter table; take the value from there. p_c and n_c are the coefficient word and fraction lengths you determined in Task 2. r is used to reduce the fraction length of the multiplier output and will be varied in subtask (d).

In this subtask, consider the choice of g and n, which control the number of guard bits in the accumulator and the fraction length of the MAC output. The signal from ADC is considered to be in the fractional fixed-point format s$p.(p-1)$. Then, to ensure that there is no overflow in the accumulator, one must be able to represent the numbers in the range $[-A, +A]$, where $A = \sum_i |h_i|$ (h_i are the filter coefficients). A is the maximum possible value in the accumulator and therefore also the maximum possible value in the MAC output.

Based on A, we can select the values of g and n. For example, if $A < 1$ we can use $g = 0$ and $n = p - 1$. Explain why. Analyze your filter and select appropriate g and n.

(b) Next, perform the following experiment using the values found in (a) for `bitspec` and set $r = 0$.

Firstly, use `'sine'` as the first argument in the function call. As a result, the test signal is a sum of three sine waves (3.5 kHz, 7.5 kHz, 8.5 kHz; sampling frequency is 30 kHz). Using `freqz` function, plot the **spectrum** of the error signal and include the figure in your report. Note that the error signal is the third output of **FIR_Experiment()**.

Secondly, repeat the simulation using a random signal as input. To do that, use `'rand'` as the first argument. In this case also include a figure of the error spectrum in the report. How the error spectra are different? Why?

(c) Change the fraction length of the MAC unit output by increasing/decreasing the 6th parameter of `bitspec`. Examine the values $\{n-4,\ n-3, ..., n+4\}$ where n is the value determined in subtask (a). Simulate each case for random input signal and compute the power of the error signal in decibels. Tabulate or plot the result in the report (i.e. error power as a function of the fraction length). Explain the dependence observed between the fraction length and the error power.

(d) The length of the pipeline register used for the multiplication output can be reduced by increasing the value of r, the 6th parameter of `bitspec`. Set other parameters as in subtask (b), simulate, and compute the power of the error signal for $r = 0, 1, 2, ...$ How does quantization of the multiplication results change the power of the error signal? What length of the pipeline register seems to be sufficient?

Filter requirements

All specifications are for the sampling frequency $F_s = 30\,\text{kHz}$. Use it in `filterDesigner` (or `fdatool`).

Group number	Filter type	Cutoff frequencies Fpass, Fstop [kHz]	Passband ripple [dB]	Stopband attenuation [dB]	I/O signal word length
1	high-pass	6.8, 5.4	0.25	60	11
2	low-pass	4.6, 5.9	0.25	50	10
3	high-pass	5.9, 4.4	0.25	70	12
4	high-pass	6.4, 5.3	0.50	60	11
5	low-pass	5.1, 6.3	0.25	60	11
6	low-pass	5.0, 6.7	0.25	70	12
7	high-pass	6.1, 4.7	0.50	70	12
8	low-pass	4.9, 6.3	0.50	70	12
9	low-pass	4.9, 6.3	0.50	60	11
10	low-pass	5.3, 6.4	0.50	60	11
11	low-pass	4.7, 5.7	0.75	60	11
12	high-pass	6.4, 4.9	0.75	50	10
13	low-pass	4.6, 6.3	0.25	60	11
14	high-pass	5.5, 4.3	0.75	70	12
15	high-pass	6.0, 4.9	0.25	60	11
16	high-pass	6.4, 5.3	0.50	50	10
17	low-pass	5.1, 6.5	0.25	70	12
18	low-pass	4.6, 6.2	0.50	60	11
19	low-pass	4.8, 6.3	0.50	50	10
20	low-pass	4.4, 6.0	0.75	50	10
21	high-pass	6.2, 5.0	0.50	60	11
22	high-pass	5.6, 4.5	0.50	60	11
23	low-pass	5.1, 6.7	0.25	60	11
24	low-pass	4.4, 5.5	0.75	70	12
25	low-pass	4.8, 6.1	0.50	60	11
26	high-pass	6.0, 4.7	0.75	60	11
27	high-pass	5.9, 4.6	0.50	50	10
28	low-pass	5.2, 6.2	0.75	50	10
29	high-pass	6.5, 5.3	0.50	70	12
30	low-pass	5.5, 6.7	0.75	60	11