P2F18 ECE 4624 11 October 2018

1. Implement a sinusoidal signal generator in the form of a DDS (direct digital synthesizer), based on reading out an array of samples consecutively and repeatedly (think circular buffer). For an arbitrary fractional frequency, determine the closest number of periods, to fall between 10 and 50 periods, such that the repeated read-out of consecutive samples generates a unit amplitude sinusoid while maximizing the SFDR (spurious free dynamic range). Show the achieved, measured, SFDR for a fractional frequency of 0.33. Describe your process, motivations, and measurement procedure(s). Execute the measurement over 500 periods.
2. Another indicator of performance is TD (total distortion). Describe, and execute (for fractional frequency 0.33), your process for measuring TD.
3. Now run your DDS for 100 randomly picked fractional frequencies in the interval (0,0.5] and plot the SFDR and TD results versus fractional frequency. Are SFDR and TD related?
4. How do SFDR and TD change if the look-up table (LUT) constraint is that at most 1024 memory locations are available to you (you do not have to use all 1024)?
5. We're interested in using fast FFT-based circular convolution approaches to achieve the same output result as by direct linear convolution. Assume that 2 blocks of (real) input data are being handled simultaneously (as real and imaginary part of the complex input). For N-point FFTs and IFFTs with $N=2^M$, M=11 through 16, analyze the expected "per output sample" speed-up – over direct convolution – of the overlap-save and overlap-add methods operating in steady state, i.e. ignoring transients at power-up and power-down. Use that a complex N-FFT or N-IFFT requires $2N[\log_2(4N)-1]$ real MACs, and that isolated real adds take as much time as a real MAC. What is the optimal length of the UPR?
6. Write your own Matlab script to implement the overlap-save method for two blocks of input simultaneously. Show the difference of the direct linear convolution and overlap-save results, as verification that your overlap-save implementation works. Use both approaches to filter the data in **xn** (in P2F18data.mat) with the UPR **hn**, and show efficacy of your algorithms. For the FFT lengths indicated in part 5, evaluate the actual speed-up realized by the overlap-save approach and present your results graphically. You may use the Matlab commands tic, toc, etime, cputime, etc. Compare actual and expected results and behaviors.
7. In the vicinity of sample indices for which the filtered output takes on large positive or large negative values, what does the input signal look like? What does this particular filter, with UPR **hn**, detect? How does this make sense?
8. For the P2F18 SFG below, define the signal at the green branch node as state variable 1, i.e. $s_{1,n}$, and the signal at the red branch node as state variable 2, i.e. $s_{2,n}$. Write the state space representation, and use it to show that the signal flow graph (SFG) represents a second order section (SOS) discrete-time filter. Indicate the correspondence between the variables in the SFG (i.e. k_1, k_2, c_1, c_2, and c_3) and the standard Direct-Form coefficients in

$$H(z) = \frac{\sum_{m=0}^{2} b_m z^{-m}}{\sum_{k=0}^{2} a_k z^{-k}}; \; a_0 \equiv 1 \qquad (1)$$

9. Show also how a set of standard Direct-Form coefficients can be converted to an equivalent set of P2F18 SFG coefficients (equivalent means that the SFG can implement an SOS of the form given in (1)). Are there any restrictions?

10. What does the BIBO region look like in terms of the P2F18 coefficients?
11. Write a Matlab module implementing one clock-cycle of the P2F18 SFG, so that internally all outputs from the summing nodes are explicitly available. The inputs to this module are the SFG variable vectors, $\mathbf{c} = [c_0 \quad c_1 \quad c_2]$ and $\mathbf{k} = [k_1 \quad k_2]$, the present state, and the present value of the input. The outputs of the module are the output of the SFG and the next state.
12. Design and execute input-output experiments that show that your module operates as required, going from direct form coefficients to SFG coefficients and vice versa. Provide arguments and results.
13. Using $\mathbf{k} = [-0.53 \quad 0.91]$, measure and compare the frequency response of the filter that uses $\mathbf{c} = [-0.11 \quad 0 \quad 0]$ with the filter that uses $\tilde{\mathbf{c}} = [0 \quad 0 \quad -0.11]$. How do these responses compare to what is expected in theory? What are your observations?
14. Use DFT-based spectral analysis, including windows, to arrive at an alternative frequency response measurement process in which either the sin or the cos experiment part of the cos/sin frequency response measurement process is used. How does the use of a window help when the states are quantized to 12 bits?

PDF report to beex@vt.edu, due 1 November 2018, 11:59 PM

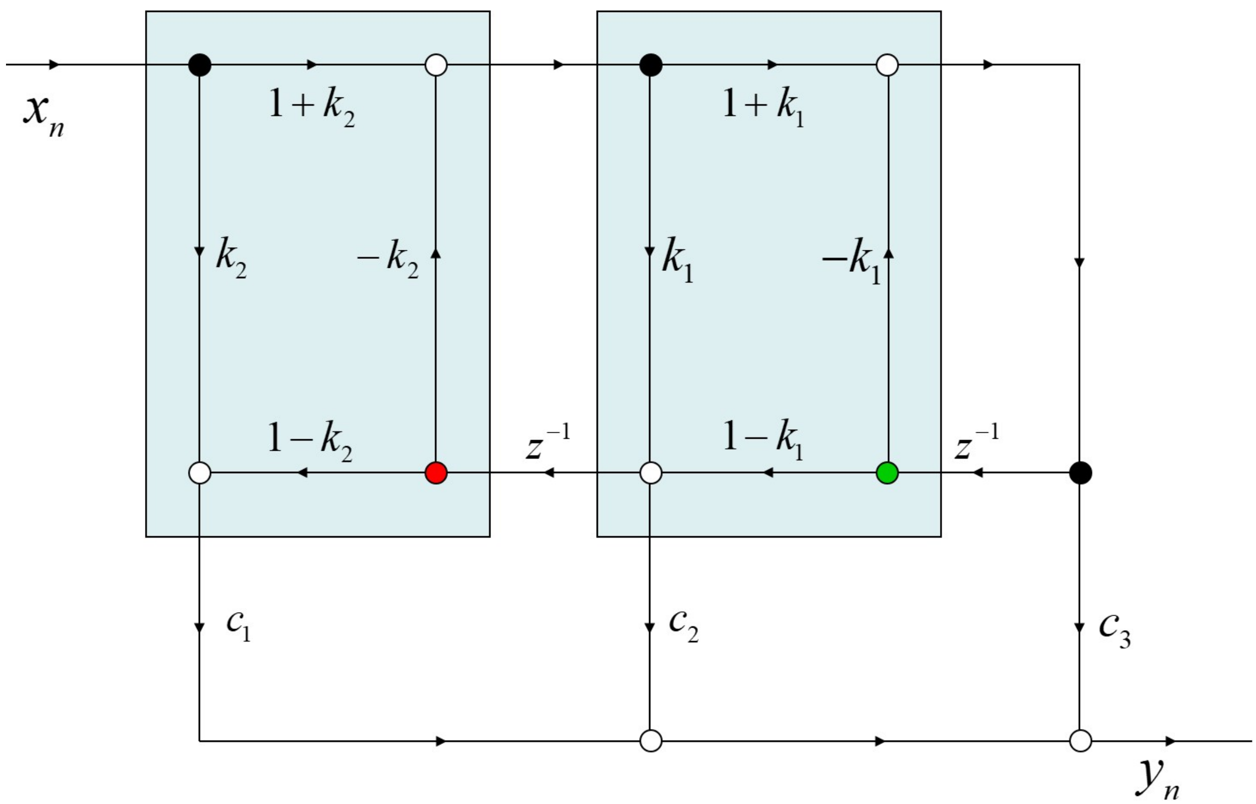

Fig. 1 P2F18SFG