Set #7

Due Tuesday December 9, 2025

Problems (4 required, 5 is quantization simulation in Jupyter):

1. Text problem 6.5. System function from SFG. Show your work.
2. Text problem 6.8. System function from SFG. Show your work.
3. Text problem 6.27 (a) **only**.
4. **10 Points Bonus**: Text problem 6.56a & c, **just** for the SFG of Figure P6.56c.
5. **Fixed-point filter modeling**: Design an equal-ripple FIR lowpass filter using

   ```
   b = FIR_d.fir_remez_lpf(1/8,1/6,1,80,n_bump=4)
   ```

   You will use `ssd.simple_quant()` to quantize the coefficients and compare filter characteristics before and after quantization.

   (a) Take the `b` coefficient array and quantize the coefficients to 16-bit and 14-bit rounding as follows

   ```
   bq16 = simple_quant(b,16,1.0,'sat') # 16-bit coefficient quantization
   bq14 = simple_quant(b,14,1.0,'sat') # 14-bit coefficient quantization
   ```

   (b) Was it is reasonable to quantize the coefficients with $X_{\max} = 1.0$? Is there any unnecessary loss of precision?

   (c) Compare the three filters in terms of magnitude response in dB, group delay, and pole-zero plots. Overlay plots are the best way to see the impact of quantization, except for the pole-zero plots, which cannot be overlayed at present. The function, `freqz_resp_list` works well here, e.g.,

   ```
   FIR_d.freqz_resp_list([b,bq16,bq14],[[1],[1],[1]],'dB')
   ```

   makes this comparison very easy, as you can load multiple filters in the function call. Note overlaying pole-zero plots is not easy, so you will have to make three plots.

   (d) Is linear phase maintained in spite the coefficient quantization? Why or why not? Can you predict the phase/group delay results from the quantized coefficients? From the pole-zero plots? What is most impacted in the overall frequency response.