

Assignment: Homework 2

What to Hand It In

1. Your code with appropriate comments and a **readme file**. You are responsible for making us understand and know how to run your code.
2. A report with answers, figures, and explanations.
3. Any sound files asked for in the problems.

How to Hand It In

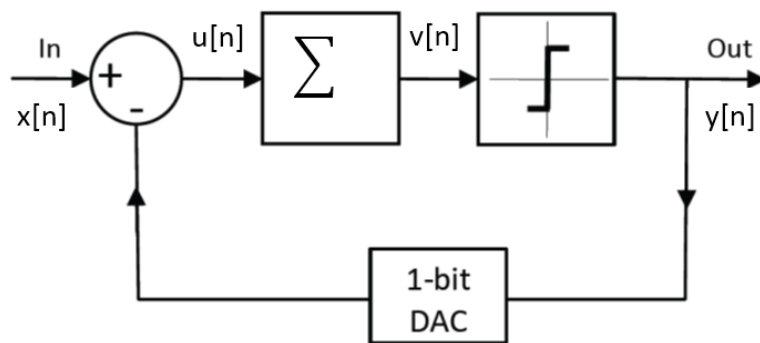
1. Put all your files in one folder. Compress this folder and name it `<firstname>_<lastname>_HW2.zip`. For example, "Zhiyao_Duan_HW2.zip".
2. Submit via blackboard to the appropriate entry.

When to Hand It In

It is due immediately at the beginning of the day (i.e., 0:00AM) specified on the course calendar. **Late assignments will receive a grade deduction of 20% of the full grade each day.**

Problems (10 points in total)

1. (3 points) Implement a function in Matlab to simulate a 1st order Sigma-Delta modulator. It takes a discrete-time analog signal $x[n]$ with dynamic range $[-1, 1]$ as input, and outputs a modulated discrete-time digital signal $y[n]$ that takes binary values -1 and 1 . As a reminder, the diagram of this modulator is shown in the following figure:



Syntax of this function is as follows:

```
function y = SigDelMod_1order_1bit(x, v_init)
% 1st order Sigma-Delta Modulator with 1-bit quantizer
%
% Input
% - x : input oversampled signal, a vector representing x[n]=x(nT)
% - v_init: initial value of the 1-order integrator, a scalar
% Output
% - y : output of the modulator, a vector representing y[n]=y(nT)
```

- a. (1 point) The 1st-order integrator in the diagram has a transfer function of $\frac{z^{-1}}{1-z^{-1}}$, i.e., the difference function is $v[n] = v[n-1] + u[n-1]$, where $u[n]$ is the input, and $v[n]$ is the output.
 - b. (1 point) The 1-bit quantizer will be a midriser quantizer with 2 levels. Different from the normal midriser formula, we will let the levels be at -1 and 1, instead of -0.5 and 0.5, i.e. $y[n] = 1$ if $v[n] \geq 0$, and $y[n] = -1$ if $v[n] < 0$.
 - c. (1 point) The 1-bit DAC is just an identity transformation, i.e., $y_a[n] = y[n]$.
2. (1.5 points) Use your Sigma-Delta modulator to process a direct current (DC) signal.
 - a. (0.5 points) Generate a DC signal with 100 samples. Try several different values of the signal within the range of $[-1, 1]$. Plot the modulator output. What are the differences?
 - b. (0.5 points) Take one DC signal in the previous question, run the modulator with several different initial values of the integrator within $[-1, 1]$. Plot the modulator output. What are the differences?
 - c. (0.5 points) If the value of the input DC signal is at some "round" numbers, e.g., 0.1, -0.3, 0.6, etc., what does the modulator output look like? If the value of the input DC signal is taking other numbers, e.g., 0.1001, -0.3173, 0.6358, etc. what does the modulator output look like? Can you explain the reasons of the differences if any?
 3. (2 points) Implement a digital decimator in Matlab. A digital decimator contains a digital low-pass filter and a decimator. Run the following experiment with different oversampling factor M (e.g., 2, 4, 8, 16, 32, 64, 128, 256).
 - a. (1 point) Use "fir1" function to implement an FIR low-pass filter. Make the order of the filter equal to the oversampling factor M . The cutoff frequency should be set to $1/M * Mfs/2$. Here Mfs is the oversampling rate of the signal that goes through the filter. Use "freqz" function to verify the frequency response of your filter.
 - b. (1 point) Generate a 1-second long DC signal with value of 0.3713, and oversampling rate of $M*fs$, where $fs = 8000\text{Hz}$. Feed this signal to the Sigma-Delta modulator. Then use your low-pass filter to filter the modulator output. You may find function "filtfilt" useful. Finally, decimate the output signal for a factor of M , i.e., keep only 1 sample for every M samples. The final output signal should be a vector with 8000 samples. The value of the output vector should be close to 0.3713. For each M , try 100 different initial values of the integrator, and calculate the mean and variance of one sample (e.g., the 4000-th) of the output signal. Plot the mean vs. M and variance vs. M , and explain the result.
 4. (2.5 points) Run the modulator and decimator on a sine wave.
 - a. (0.5 points) Generate a 2-second-long sine wave with frequency of 447Hz, peak amplitude of 0.9, and oversampling rate of $M*fs$, where $M=128$ and $fs = 8000\text{Hz}$. Feed this signal to the Sigma-Delta modulator and then filter and decimate the output. The final output signal should be vector with 16000 samples. Plot five cycles of the output signal. Does it look good? Save the signal as "447_8k.wav". Does it sound good?

- b. (1 point) Plot the power spectrum of the output signal using the “fft” function. Do you see a strong peak at the frequency 447Hz? Do you also see strong peaks at several other frequencies, which are artifacts introduced by the quantizer? Explain the reason of the artifacts. In order to attenuate the artifacts, you can add dither to the signal $v[n]$ before quantizing. Plot the power spectrum of the final output signal. Do you see less artifact?
 - c. (1 point) Change the oversampling rate M (e.g., 2, 4, 8, 16, 32, 64, 128, 256) of the sine wave and the system. Plot the power spectrum of the final output. Do you see SNR increase when M increases?
5. (1 point) Implement a Sigma-Delta modulator taking the order L as input. Name the function as `SigDelMod_1bit(x, L)`. Run your modulator on a DC signal or a sine wave with different L . Compare output resulted from the modulators with different orders.

```
function y = SigDelMod_1bit(x, L)  
% Lth-order Sigma-Delta Modulator with 1-bit quantizer  
%  
% Input  
% - x : input oversampled signal  
% - L : order of the integrator  
% Output  
% - y : output of the modulator
```