

ECE 535 — Digital Signal Processing
Matlab Project 1: Bandlimited Interpolation
George Mason University
Fall 2020

Due: Sunday, October 25 by 11:59 pm

Submit via Blackboard.

Reminder: No late projects will be accepted!

-
- Matlab project submissions must be your own individual work. This includes code, plots, answers to questions, and all other report components. You may discuss the project with others at a high level. However, the coding and the write-up (answering questions, explanations, etc.) are to be completed individually.
 - Matlab projects must be submitted a single pdf file.
 - Matlab project submissions must be typed or written neatly. All parts must appear in the order they are assigned.
 - All plots must be neatly labeled with x-axis and y-axis labels. Any plot that is not fully labeled will not be graded.
 - All plots must be fully referenced. I will not try to figure out what plots correspond to what parts of the project when grading. You can either assign figure numbers to the plots (i.e. The resulting plot is shown in Figure 1) or cite the location in which the plot appears (i.e. The resulting plot appears at the top of page 5).
 - Project submissions should include all analytical work (paper and pencil calculations), all relevant explanations and answers to questions (in complete sentences), and all Matlab code. Matlab code must be well-documented. It may be included in an appendix at the end of the report and appropriately referenced as needed. Alternatively, the relevant code may be included in each section of the report.
 - Remember that you can use the help command in Matlab to find out more about any command.

Note: This project is adapted from Computer-Based Exercises for Signal Processing Using Matlab by Burrus et al.

Overview

The process of interpolation essentially corresponds to estimating or reconstructing the values of a signal at locations (times) between the sample values. Figure 4.23 in O&S depicts a system for interpolating a signal by a factor of L . The signal $x[n]$ is first upsampled by inserting $L - 1$ between samples of $x[n]$. Then, the lowpass filter interpolates between nonzero values of $x_e[n]$ to generate the interpolated signal $x_i[n]$. When the lowpass filter is ideal, the interpolation is referred to as bandlimited interpolation.

Accurate bandlimited interpolation requires a carefully designed high-order lowpass filter. Two simple and very approximate procedures which are often used instead are zero-order hold and linear interpolation. For zero-order hold interpolation, each value of $x[n]$ is simply repeated L times:

$$x_i[n] = \begin{cases} x_e[0], & \text{for } 0 \leq n \leq L - 1 \\ x_e[L], & \text{for } L \leq n \leq 2L - 1 \\ x_e[2L], & \text{for } 2L \leq n \leq 3L - 1 \\ \vdots & \vdots \end{cases}$$

This can be accomplished by convolving $x_e[n]$ with the impulse response

$$h_{\text{zoh}}[n] = \sum_{k=0}^{L-1} \delta[n - k]$$

Zero-order hold interpolation is often used in digital-to-analog converters, resulting in analog “stairstep” waveforms.

Linear interpolation can be accomplished using a system with impulse response

$$h_{\text{lin}}[n] = \begin{cases} 1 - |n|/L, & \text{for } 0 \leq |n| \leq L - 1 \\ 0, & \text{otherwise} \end{cases}$$

Unlike the zero-order hold interpolator, the linear interpolator is noncausal and has zero group delay.

Note that the ideal bandlimited interpolator (a sinc function) also has a noncausal impulse response.

1. Given $x_e[n]$ as defined in Figure 4.23 of O&S write the difference equations that correspond to the zero-order hold (ZOH) interpolation filter

and the linear interpolation filter for the impulse responses $h_{\text{zoh}}[n]$ and $h_{\text{lin}}[n]$.

2. Enter the impulse responses for the ZOH and linear interpolators into Matlab using the value $L = 5$. Note that the linear interpolator is a noncausal filter. Since Matlab doesn't recognize negative indices, you must enter a causal version, and then "time advance" the result by relabeling the time axis appropriately. Assign the impulse response sequences to the Matlab variables `hzoh` and `hlin`, respectively. Use the Matlab function `stem` to plot the two impulse responses as a function of sample index n . Make sure the sample index values n are correct!
3. Use the Matlab command `fft` to compute the frequency responses of the two filters. (Use a value for the FFT length N that is sufficiently large to give you a smooth approximation to the DTFT.) Plot the magnitude of the frequency responses of the ZOH and linear interpolation filters on the same graph. Use a linear (not logarithmic) scale. Label the horizontal axis in terms of frequency, ω . Draw (by hand or via Matlab) the magnitude of the ideal interpolation filter on your plot, as well. Which filter is a better approximation to the ideal bandlimited interpolator?
4. Using the Matlab functions `upsample` and `filter`, implement the interpolation system with $L = 5$. (Note that, when using `filter` with an FIR filter, you'll want to set `b` equal to the filter impulse response and `a` equal to 1.) Perform the interpolation using (i) the ZOH and (ii) the linear interpolator. For input to the system, use the sequence `data1` that has been provided in `mp1DataFile.mat`. Assign the interpolated sequences to the Matlab variables `xzoh` and `xlin`.
5. Using `stem`, plot the sequences `xzoh` and `xlin` on a sample axis from $n = -10$ to $n = 20$. In establishing the sample (time) axis for the plot, be sure to take into account that $h_{\text{lin}}[n]$ is noncausal. You can check whether or not you have the time axis appropriately lined up for `xzoh1` and `xlin1` by noting that for both the ZOH and the linear interpolator, all the values of the original data are preserved exactly on the expanded sample (time) axis.
6. The Matlab variable `hsharp` contains the impulse response of an interpolation filter that more closely approximates the ideal lowpass filter

than either the linear interpolator or the ZOH interpolator. This filter (hsharp) is noncausal, and its impulse response has even symmetry about $n = 0$. Use stem to plot this impulse response as a function of n . Again, make sure the sample index values n are correct!

7. Plot (on a linear scale) the magnitude of the frequency response of the filter hsharp. How does it compare to the ZOH and linear interpolators? How does it compare to the ideal interpolation filter?
8. Interpolate the sequence data1 using hsharp. Assign the resulting sequence to the Matlab variable xsharp. Using stem, plot xsharp on a sample axis from $n = -10$ to $n = 20$.
9. In the items above, we have implemented the interpolation system with $L = 5$ for three different filters. In the final item, we want to compare the results with essentially ideal bandlimited interpolation. To do this, we need to be careful about comparing appropriate segments of $x_i[n]$.
 - (a) For each of the three filters, with the appropriate time advance incorporated, specify the range of n over which $x_i[n]$ is available. (Pay attention to the output of the filter command. How does its output differ from the output of conv, the convolution function?)
 - (b) From your answer to part (a), what is the largest interval common to all three filtered outputs?
 - (c) Another issue is that the Matlab function filter assumes that data1 is zero for $n < 0$, and consequently, there is a startup transient in filtering until the filter impulse response totally engages the data. The duration of this transient depends on the length of the FIR filter. The three filtered outputs should only be compared after the transient. Taking this into account together with your answer to part (b), xzoh, xlin, and xsharp should be compared only over a common time interval $n_1 \leq n \leq n_2$. Determine n_1 and n_2 .
10. As a final task, we evaluate the performance of the various interpolation filters compared with what would be obtained by essentially bandlimited interpolation. The sequence corresponding to perfect bandlimited interpolation is contained in the Matlab variable ideal1. A measure of

the average interpolation error is

$$e_x = \frac{1}{n_2 - n_1 + 1} \sum_{n=n_1}^{n=n_2} |x_i[n] - x_{\text{ideal}}[n]|^2.$$

Use Matlab to compute the average interpolation error for the three filters. Which interpolation filter has the best performance? Is this what you expected?