3DY4 Lab 2 Report

Jan 2024

Kyler Witvoet, 400393313

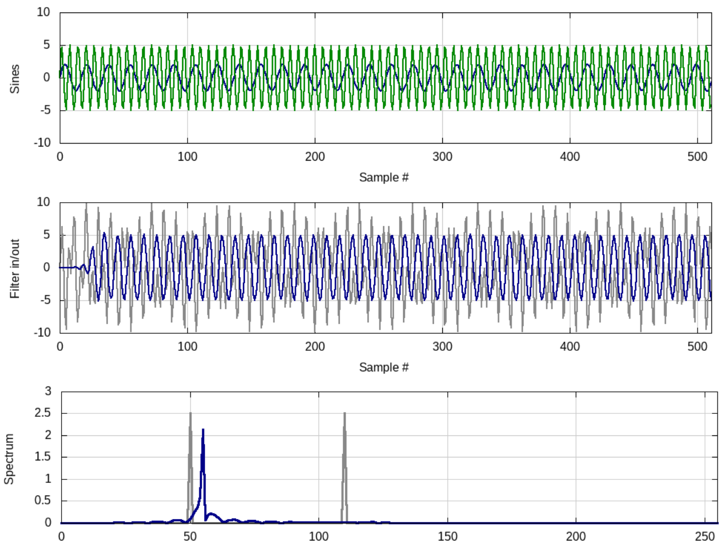
Josh Umansky, 400234265

TH1:

Sample Length	DFT Runtime (s)	IDFT Runtime (s)	DFT Runtime (-o1 Complier)	IDFT Runtime (-o1 Complier)
2^10	.127	.118	.108	.138
2^11	.504	.520	.509	.483
2^12	1.891	2.183	1.755	1.972
2^13	6.931	7.385	6.817	7.348
2^14	26.525	29.593	26.489	29.478

When analyzing the functions for DFT and IDFT, both have a runtime complexity of O(n^2), which do correlate to the experimental runtime analysis shown. When compiling using the –O1 modifier, a small runtime decrease is observed, which correlates to the basic code optimizations that are performed.

TH2: In the first plot of the resulting figure, we can see a graph of the two input sin waves with frequencies of 30Hz and 80Hz and amplitudes of 2 and 5 respectively. On the second plot, the pre and post filter data is graphed in the time domain, the grey representing the multiplication of the two input sin functions, and the purple function representing output response of the filter, you can see that it appears as a single sine wave. In the third plot, the frequencies of the pre and post filter signals are displayed. The grey shows the frequencies of the multiplied input sin wave which has frequencies of the positive addition of the original frequencies (80 + 30 = 110Hz) and the negative addition of the original frequencies (80 – 30 = 50 Hz). The purple shows the frequency of the filter response signal which appears to have a frequency around 60 Hz.



TH3:

To accomplish the convolution using block processing, we approached the problem like lab 1, since the method of overlapping adjacent blocks to preserve the convolution of the beginning elements was retained. The difference in implementation in C++ was that the definition of convolution as defined by the summation equation was used, instead of the IDFT of the multiplication of the DFT of the signal and filter in the frequency domain. After a block is processed, the end of the block (Size equal to the size of the filter) is concatenated to the next block, and processed, until every block has been processed.