Appendix 2. MATLAB Code

```
1 % SIO 207A Final Project
 2 % Ruipu Ji
 4 % Initialization and default plot settings.
 5 clear; clc; close all;
 7 set(0, 'DefaultAxesFontSize', 15);
 8 set(0, 'DefaultTextFontSize', 15);
10 set(0, 'DefaultTextInterpreter', 'latex');
11 set(0, 'DefaultLegendInterpreter', 'latex');
12 set(0, 'DefaultAxesTickLabelInterpreter', 'latex');
13
14 %% Data Set.
15 % Generate a 4096-point discrete-time signal x(n). ------
16 A = [100 10 1]; % Magnitude.
17 f = [160 237 240]; % Analog frequency [Hz].
18 Phi = [0 \ 0 \ 0]; % Phase angle [rad].
20 fs = 1000; % Sampling frequency [Hz].
21 n = 4096; % Number of data points.
22 t = (0:1:n-1)'*1/fs; % Time vector.
23
24 x = zeros(n,1); % Initialize the signal sequence as an empty array.
26 % Generate the discrete-time signal sequence x(n).
27 for idx = 1:size(A, 2)
      x = x + A(idx) *cos(2*pi*f(idx)*t+Phi(idx));
29 end
30
31 % Plot the first 256 data points of the signal x(n). -----
32 NFFT = 256;
33
34 figure('Position', [0, 0, 1800, 600]);
35
36 subplot(1,2,1);
37 hold on;
38 plot(0:1:NFFT-1, x(1:NFFT), 'b', 'LineWidth', 2);
39 grid on;
40 box on;
41 xlim([0 300]);
42 xticks(0:50:300);
43 ylim([-150 150]);
44 yticks(-150:50:150);
45 xlabel('Data Point Index $n$');
46 ylabel('$x(n)$');
47 title('The First 256 Data Points of Signal $x(n)$');
49 % 256-point FFT of the signal using the Kaiser-Bessel window. -----
50 % Kaiser-Bessel window (alpha = 2.5 or beta = alpha*pi = 7.85).
51 KaiserBesselWindow = kaiser(NFFT, 7.85);
53 % Window the first 256 data points of the signal x(n).
54 x 256 KB = x(1:NFFT) .* KaiserBesselWindow;
56 % Perform 256-point FFT on the windowed signal and calculate the logarithmic \checkmark
```

```
magnitude.
 57 X 256 KB magnitude = 20*log10(abs(fftshift(fft(x 256 KB, NFFT))));
 58 X 256 KB magnitude = X 256 KB magnitude - max(X 256 KB magnitude); % Normalize the ✓
results.
 59
 60 % Calculate bin width of the FFT.
 61 BinWidth = fs/NFFT;
 63 % Calculate the frequency bin index for analog frequencies.
 64 f idx = zeros(1, size(f,2));
 66 for idx = 1:size(f idx, 2)
             f idx(1,idx) = round(f(idx)/BinWidth);
 69
 70 % Plot the result.
 71 f fft = (-fs/2:BinWidth:fs/2-BinWidth)';
 72
 73 subplot(1,2,2);
 74 hold on;
 75 plot(f fft, X 256 KB magnitude, 'b', 'LineWidth', 2);
 76 xline(f(1), 'r--', 'LineWidth', 2);
 77 xline(f(2), 'g--', 'LineWidth', 2);
 78 xline(f(3), 'k--', 'LineWidth', 2);
 79 xline(-f(1), 'r--', 'LineWidth', 2);
 80 xline(-f(2), 'g--', 'LineWidth', 2);
 81 xline(-f(3), 'k--', 'LineWidth', 2);
 82 grid on;
 83 box on;
 84 xlim([-500 500]);
 85 xticks(-500:100:500);
 86 ylim([-120 0]);
 87 yticks(-120:20:0);
 88 xlabel('Analog Frequency $f$ [Hz]');
 89 ylabel('$|X(f)|$ [dB]');
 90 legend('|X(f)|', '|X(f)|', '|X(f)
 91 title('Logrithmic Magnitude of $X(f)$');
 92
 93 exportgraphics(gcf, 'Figure1-PartI.png', 'ContentType', 'image');
 95 %% Decimation Filter Design.
 96 N coefficients = 64; % Number of coefficients.
 97 fc passband = 40; % Passband cutoff frequency [Hz].
 98 fc stopband = 85; % Stopband cutoff frequency [Hz].
 99 Weight = [50 1]; % Define weight ratio vector [passband stopband].
100
101 % Filter design using an equiripple FIR filter design algorithm.
102 f parameter = [0 \text{ fc passband/(fs/2) fc stopband/(fs/2) 1}];
103 a parameter = [1 1 0 0];
104 h = firpm(N coefficients-1, f parameter, a parameter, Weight)'; % Filter design.
105
106\ \% Calculate the frequency response of the filter.
107 NFFT filter = 1024; % NFFT for the filter calculation.
108 f filter fft = (-0.5:1/NFFT filter:0.5-1/NFFT_filter)' * fs; % Frequency vector for <math>\checkmark
the plot.
109 RectangularWindow = rectwin(NFFT filter); % Rectangular window.
```

```
110 h padded = padarray(h, [NFFT filter-size(h,1) 0], 'post'); % Pad the filter to the
same length as NFFT.
111
112 % Perform 256-point FFT on the windowed signal and calculate the logarithmic \checkmark
magnitude.
113 H magnitude = 20*log10(abs(fftshift(fft(h padded.*RectangularWindow, NFFT filter))));
114 H magnitude = H magnitude - max(H magnitude); % Normalize the result.
116 % Plot the result.
117 figure('Position', [0, 0, 2500, 500]);
118
119 subplot(1,3,1);
120 hold on;
121 plot(0:1:N coefficients-1, h, 'b', 'LineWidth', 2);
122 grid on;
123 box on;
124 xlim([0 70]);
125 xticks(0:10:70);
126 ylim([-0.04 0.16]);
127 yticks (-0.04:0.04:0.16);
128 xlabel('$n$');
129 ylabel('$h(n)$');
130 title('Impulse Response $h(n)$');
131
132 subplot (1, 3, 2);
133 hold on;
134 plot(f filter fft, H magnitude, 'b', 'LineWidth', 2);
135 xline(fc passband, 'r--', 'LineWidth', 2);
136 xline(fc stopband, 'k--', 'LineWidth', 2);
137 grid on;
138 box on;
139 xlim([-500 500]);
140 xticks(-500:100:500);
141 ylim([-100 0]);
142 yticks(-100:20:0);
143 xlabel('Analog Frequency $f$ [Hz]');
144 ylabel('$|H(f)|$ [dB]');
145 legend('$|H(f)|$', 'Passband', 'Stopband', 'Location', 'northeast');
146 title('Logrithmic Magnitude of $H(f)$');
147
148 subplot(1,3,3);
149 hold on;
150 plot(f filter fft, H magnitude, 'b', 'LineWidth', 2);
151 grid on;
152 box on;
153 xlim([0 40]);
154 xticks(0:10:40);
155 ylim([-0.01 0.002]);
156 yticks(-0.01:0.002:0.002);
157 xlabel('Analog Frequency $f$ [Hz]');
158 ylabel('$|H(f)|$ [dB]');
159 title('Passband Ripples of $H(f)$');
161 exportgraphics(gcf, 'Figure2-PartII.png', 'ContentType', 'image');
163 %% Complex Basebanding and Desampling.
```

```
164 % Complex multiplication on the discrete-time signal x(n). ------
165 f0 = 250; % Center frequency [Hz].
166 w0 = 2*pi*f0/fs; % Center frequency in rad.
167 x complex = zeros(size(x,1), 1); % Initialization for the complex sequence.
168
169 for idx = 1:size(x,1)
170 x \text{ complex(idx)} = \exp(-1i*w0*idx)*x(idx);
171 end
172
173 % 256-point FFT of the complex sequence. -----
174 % Window the first 256 data points of the complex sequence.
175 x complex 256 KB = x complex(1:NFFT) .* KaiserBesselWindow;
176
177 % Perform 256-point FFT on the windowed signal and calculate the logarithmic \checkmark
magnitude.
178 X complex 256 KB magnitude = 20*log10(abs(fftshift(fft(x complex 256 KB, NFFT))));
179 X complex 256 KB magnitude = X complex 256 KB magnitude - max ✔
(X complex 256 KB magnitude); % Normalize the results.
181 % Plot the results.
182 figure('Position', [0, 0, 1800, 600]);
183
184 subplot(1,2,1);
185 hold on;
186 plot(f_fft, X_complex_256_KB_magnitude, 'b', 'LineWidth', 2);
187 x line(f(1)-f0, 'r--', 'LineWidth', 2);
188 xline(f(2)-f0, 'g--', 'LineWidth', 2);
189 x = (f(3)-f0, 'k--', 'LineWidth', 2);
190 xline(-f(1)-f0, 'r--', 'LineWidth', 2);
191 xline(-f(2)-f0, 'g--', 'LineWidth', 2);
192 xline(-f(3)-f0, 'k--', 'LineWidth', 2);
193 grid on;
194 box on;
195 xlim([-500 500]);
196 xticks(-500:100:500);
197 ylim([-120 0]);
198 yticks(-120:20:0);
199 xlabel('Analog Frequency $f$ [Hz]');
200 ylabel('$|X(f)|$[dB]');
201 legend('|X(f)|', '|x|pm f 1|x|', '|x|pm f 2|x|', '|x|pm f 3|x|', 'Location', 'northeast');
202 title('Logrithmic Magnitude of $X(f)$ after Complex Basebanding');
204 % Low pass filter the complex sequence in the time domain. ------
205 y complex = filter(h, 1, x complex);
207 % Window the data points (n = 256-511) of the signal y(n).
208 y complex 256 KB = y complex(NFFT+1:2*NFFT) .* KaiserBesselWindow;
209
210 % Perform 256-point FFT on the windowed signal and calculate the logarithmic \checkmark
magnitude.
211 Y complex 256 KB magnitude = 20*log10(abs(fftshift(fft(y complex 256 KB, NFFT))));
212 Y_complex_256_KB_magnitude = Y_complex_256_KB_magnitude - max 🗸
(Y complex 256 KB magnitude); % Normalize the results.
213
214 % Plot the results.
215 subplot(1,2,2);
```

```
216 hold on;
217 plot(f fft, Y complex 256 KB magnitude, 'b', 'LineWidth', 2);
218 xline(f(1)-f0, 'r--', 'LineWidth', 2);
219 xline(f(2)-f0, 'g--', 'LineWidth', 2);
220 xline(f(3)-f0, 'k--', 'LineWidth', 2);
221 xline(-f(1)-f0, 'r--', 'LineWidth', 2);
222 xline(-f(2)-f0, 'g--', 'LineWidth', 2);
223 xline(-f(3)-f0, 'k--', 'LineWidth', 2);
224 grid on;
225 box on;
226 xlim([-500 500]);
227 xticks(-500:100:500);
228 ylim([-120 0]);
229 yticks(-120:20:0);
230 xlabel('Analog Frequency $f$ [Hz]');
231 ylabel('$|Y(f)|$ [dB]');
232 legend(\$|Y(f)|\$", \$\pm f 1\$", \$\pm f 2\$", \$\pm f 3\$", 'Location', 'northeast');
233 title('Logrithmic Magnitude of $Y(f)$ of the Filtered Signal');
235 exportgraphics(gcf, 'Figure3-PartIII.png', 'ContentType', 'image');
236
237 % Desample the complex filtered sequence y(n) by a factor of 8. -----
238 factor downsample = 8;
239 x prime = y complex(1:factor downsample:end);
240
241 %% High Resolution Spectral Analysis.
242 % Window the data points (n = 256-511) of the signal x'(n).
243 x prime 256 KB = x prime(NFFT+1:2*NFFT) .* KaiserBesselWindow;
244
245 % Perform 256-point FFT on the windowed signal and calculate the logarithmic \checkmark
magnitude.
246 X prime 256 KB magnitude = 20*log10(abs(fftshift(fft(x prime 256 KB, NFFT))));
247 X prime 256 KB magnitude = X prime 256 KB magnitude - max(X prime 256 KB magnitude); 🗸
% Normalize the results.
249 % Calculate the upadated bin width of the FFT.
250 fs prime = fs/factor downsample;
251 BinWidth prime = fs prime/NFFT;
252 f prime fft = (-fs prime/2:BinWidth prime:fs prime/2-BinWidth prime)';
253
254 % Plot the results.
255 figure('Position', [0, 0, 800, 600]);
256 hold on;
257 plot(f prime fft, X prime 256 KB magnitude, 'b', 'LineWidth', 2);
258 xline(f(1)-f0+fs prime, 'r--', 'LineWidth', 2);
259 xline(f(2)-f0, 'g--', 'LineWidth', 2);
260 xline(f(3)-f0, 'k--', 'LineWidth', 2);
261 xline(-f(1)-f0+3*fs prime, 'r--', 'LineWidth', 2);
262 xline(-f(2)-f0+4*fs prime, 'g--', 'LineWidth', 2);
263 xline(-f(3)-f0+4*fs prime, 'k--', 'LineWidth', 2);
264 grid on;
265 box on;
266 xlim([-70 70]);
267 xticks(-70:10:70);
268 ylim([-120 0]);
269 yticks(-120:20:0);
```

```
270 xlabel('Analog Frequency $f$ [Hz]');
271 ylabel('$|X^\prime(f)|$ [dB]');
272 legend('\$|X^prime(f)|\$', '\$pm f 1\$', '\$pm f 2\$', '\$pm f 3\$', 'Location', \checkmark
'northeast');
273 title('Logrithmic Magnitude of $X^\prime(f)$');
274
275 exportgraphics(gcf, 'Figure4-PartIV.png', 'ContentType', 'image');
276
277 %% Second Iteration on LPF Design.
278 % Trial #1: Use passband/stopband weight ratio = 10. -----------
279 Weight2 = [10 1]; % Define weight ratio vector [passband stopband].
280 h2 = firpm(N coefficients-1, f parameter, a parameter, Weight2)'; % Filter design.
281
282 % Calculate the frequency response of the filter.
283 h2 padded = padarray(h2, [NFFT filter-size(h2,1) 0], 'post'); % Pad the filter to the \checkmark
same length as NFFT.
284
285 % Perform 256-point FFT on the windowed signal and calculate the logarithmic \checkmark
magnitude.
286 H2 magnitude = 20*log10(abs(fftshift(fft(h2 padded.*RectangularWindow, 🗸
NFFT filter))));
287 H2_magnitude = H2_magnitude - max(H2 magnitude); % Normalize the result.
288
289 % Plot the result.
290 figure('Position', [0, 0, 1800, 1000]);
291
292 subplot (2, 2, 1);
293 hold on;
294 plot(0:1:N coefficients-1, h2, 'b', 'LineWidth', 2);
295 grid on;
296 box on;
297 xlim([0 70]);
298 xticks(0:10:70);
299 ylim([-0.04 0.16]);
300 yticks(-0.04:0.04:0.16);
301 xlabel('$n$');
302 ylabel('$h(n)$');
303 title('Impulse Response $h(n)$');
304
305 subplot (2, 2, 2);
306 hold on;
307 plot(f filter fft, H2 magnitude, 'b', 'LineWidth', 2);
308 xline(fc passband, 'r--', 'LineWidth', 2);
309 xline(fc stopband, 'k--', 'LineWidth', 2);
310 grid on;
311 box on;
312 xlim([-500 500]);
313 xticks(-500:100:500);
314 ylim([-100 0]);
315 yticks(-100:20:0);
316 xlabel('Analog Frequency $f$ [Hz]');
317 ylabel('$|H(f)|$ [dB]');
318 legend('$|H(f)|$', 'Passband', 'Stopband', 'Location', 'northeast');
319 title('Logrithmic Magnitude of $H(f)$');
320
321 subplot(2,2,3);
```

```
322 hold on;
323 plot(f filter fft, H2 magnitude, 'b', 'LineWidth', 2);
324 grid on;
325 box on;
326 xlim([0 40]);
327 xticks(0:10:40);
328 ylim([-0.01 0.002]);
329 yticks(-0.01:0.002:0.002);
330 xlabel('Analog Frequency $f$ [Hz]');
331 ylabel('$|H(f)|$ [dB]');
332 title('Passband Ripples of $H(f)$');
334 % Low pass filter the complex sequence in the time domain.
335 y2 complex = filter(h2, 1, x complex);
336
337 % Desample the complex filtered sequence y(n) by a factor of 8.
338 x2 prime = y2 complex(1:factor downsample:end);
339
340 % Window the data points (n = 256-511) of the signal x'(n).
341 x2 prime 256 KB = x2 prime(NFFT+1:2*NFFT) .* KaiserBesselWindow;
342
343 % Perform 256-point FFT on the windowed signal and calculate the logarithmic \checkmark
magnitude.
344 X2 prime 256 KB magnitude = 20*log10(abs(fftshift(fft(x2 prime 256 KB, NFFT))));
345 X2 prime 256 KB magnitude = X2 prime 256 KB magnitude - max ✔
(X2 prime 256 KB magnitude); % Normalize the results.
346
347 % Plot the results.
348 subplot (2, 2, 4);
349 hold on;
350 plot(f prime fft, X2 prime 256 KB magnitude, 'b', 'LineWidth', 2);
351 x \text{line}(f(1) - f0 + fs \text{ prime}, 'r--', 'LineWidth', 2);
352 xline(f(2)-f0, 'g--', 'LineWidth', 2);
353 x \text{line}(f(3)-f0, 'k--', 'LineWidth', 2);
354 xline(-f(1)-f0+3*fs prime, 'r--', 'LineWidth', 2);
355 xline(-f(2)-f0+4*fs prime, 'g--', 'LineWidth', 2);
356 xline(-f(3)-f0+4*fs prime, 'k--', 'LineWidth', 2);
357 grid on;
358 box on;
359 \times \lim([-70 \ 70]);
360 xticks(-70:10:70);
361 ylim([-120 0]);
362 yticks(-120:20:0);
363 xlabel('Analog Frequency $f$ [Hz]');
364 ylabel('$|X^\prime(f)|$ [dB]');
365 legend('$|X^\prime(f)|$', '$\pm f 1$', '$\pm f 2$', '$\pm f 3$', 'Location', ✓
'northeast');
366 title('Logrithmic Magnitude of $X^\prime(f)$');
367
368 exportgraphics(gcf, 'Figure5-PartIV-Trial#1.png', 'ContentType', 'image');
369
370 % Trial #2: Use passband/stopband weight ratio = 100. -----
371 Weight3 = [100 1]; % Define weight ratio vector [passband stopband].
372 h3 = firpm(N coefficients-1, f parameter, a parameter, Weight3)'; % Filter design.
374 % Calculate the frequency response of the filter.
```

```
375 h3 padded = padarray(h3, [NFFT filter-size(h3,1) 0], 'post'); % Pad the filter to the \checkmark
same length as NFFT.
376
377 % Perform 256-point FFT on the windowed signal and calculate the logarithmic \checkmark
magnitude.
378 H3 magnitude = 20*log10(abs(fftshift(fft(h3 padded.*RectangularWindow, ✓
NFFT filter))));
379 H3 magnitude = H3 magnitude - max(H3 magnitude); % Normalize the result.
381 % Plot the result.
382 figure('Position', [0, 0, 1800, 1000]);
384 subplot (2, 2, 1);
385 hold on;
386 plot(0:1:N coefficients-1, h3, 'b', 'LineWidth', 2);
387 grid on;
388 box on;
389 \times lim([0 70]);
390 xticks(0:10:70);
391 ylim([-0.04 0.16]);
392 yticks(-0.04:0.04:0.16);
393 xlabel('$n$');
394 ylabel('$h(n)$');
395 title('Impulse Response $h(n)$');
397 subplot (2, 2, 2);
398 hold on;
399 plot(f_filter_fft, H3 magnitude, 'b', 'LineWidth', 2);
400 xline(fc passband, 'r--', 'LineWidth', 2);
401 xline(fc_stopband, 'k--', 'LineWidth', 2);
402 grid on;
403 box on;
404 xlim([-500 500]);
405 xticks(-500:100:500);
406 ylim([-100 0]);
407 yticks(-100:20:0);
408 xlabel('Analog Frequency $f$ [Hz]');
409 ylabel('$|H(f)|$ [dB]');
410 legend('$|H(f)|$', 'Passband', 'Stopband', 'Location', 'northeast');
411 title('Logrithmic Magnitude of $H(f)$');
412
413 subplot (2,2,3);
414 hold on;
415 plot(f filter fft, H3 magnitude, 'b', 'LineWidth', 2);
416 grid on;
417 box on;
418 xlim([0 40]);
419 xticks(0:10:40);
420 ylim([-0.01 0.002]);
421 yticks (-0.01:0.002:0.002);
422 xlabel('Analog Frequency $f$ [Hz]');
423 ylabel('$|H(f)|$ [dB]');
424 title('Passband Ripples of $H(f)$');
425
426 % Low pass filter the complex sequence in the time domain.
427 y3 complex = filter(h3, 1, x complex);
```

```
428
429 % Desample the complex filtered sequence y(n) by a factor of 8.
430 x3 prime = y3 complex(1:factor downsample:end);
432 % Window the data points (n = 256-511) of the signal x'(n).
433 x3 prime 256 KB = x3 prime(NFFT+1:2*NFFT) .* KaiserBesselWindow;
435 % Perform 256-point FFT on the windowed signal and calculate the logarithmic \checkmark
436 X3 prime 256 KB magnitude = 20*log10(abs(fftshift(fft(x3 prime 256 KB, NFFT))));
437 X3 prime 256 KB magnitude = X3 prime 256 KB magnitude - max ✔
(X3 prime 256 KB magnitude); % Normalize the results.
438
439 % Plot the results.
440 subplot(2,2,4);
441 hold on;
442 plot(f prime fft, X3 prime 256 KB magnitude, 'b', 'LineWidth', 2);
443 xline(f(1)-f0+fs prime, 'r--', 'LineWidth', 2);
444 xline(f(2)-f0, 'g--', 'LineWidth', 2);
445 xline(f(3)-f0, 'k--', 'LineWidth', 2);
446 xline(-f(1)-f0+3*fs prime, 'r--', 'LineWidth', 2);
447 xline(-f(2)-f0+4*fs prime, 'q--', 'LineWidth', 2);
448 xline(-f(3)-f0+4*fs prime, 'k--', 'LineWidth', 2);
449 grid on;
450 box on;
451 \times lim([-70 70]);
452 xticks(-70:10:70);
453 ylim([-120 0]);
454 yticks (-120:20:0);
455 xlabel('Analog Frequency $f$ [Hz]');
456 ylabel('$|X^\prime(f)|$ [dB]');
457 legend('\$|X^{\text{prime}}(f)|\$', '\$|pm f 1\$', '\$|pm f 2\$', '\$|pm f 3\$', 'Location', \checkmark
'northeast');
458 title('Logrithmic Magnitude of $X^\prime(f)$');
460 exportgraphics(gcf, 'Figure6-PartIV-Trial#2.png', 'ContentType', 'image');
461
```