Digital Signal Processing – exercises

Markus Kuhn

Michaelmas 2016 – Part II

Some of the exercises involve writing very short programs (< 20 lines). Preferably use MATLAB, Octave, or a similar technical-computing language. A MATLAB campus licence is available at

`https://www.cl.cam.ac.uk/teaching/matlab/`

Getting started with MATLAB:

```
https://www.cl.cam.ac.uk/teaching/1617/TeX+MATLAB/matlab-slides.pdf
https://uk.mathworks.com/help/matlab/getting-started-with-matlab.html
```

Please hand in both your source code and its output. Some tips to help this go smoothly:

- Start every program with a comment (`%`) that states your name, the exercise number, and the approximate time in minutes it took you to write it.
- Provide figures generated by your program in PDF. In MATLAB this can be done with `saveas(gcf, 'mgk25-ex10.pdf')`. You can also save both source code and its output into one PDF with a command like `publish('mgk25-ex10.m', 'pdf')`.
- Test your MATLAB programs after a `clear` instruction, to avoid it depending on other variables left in your workspace.

1 Sequences and systems

Exercise 1: What type of discrete system (linear/non-linear, time-invariant/ non-time-invariant, causal/non-causal, memory-less) is:

(*a*) $y_n = |x_n|$

(*b*) $y_n = -x_{n-1} + 2x_n - x_{n+1}$

(*c*) $y_n = \prod_{i=0}^{8} x_{n-i}$

(*d*) $y_n = \frac{1}{2}(x_{2n} + x_{2n+1})$

(*e*) $y_n = \dfrac{3x_{n-1} + x_{n-2}}{x_{n-3}}$

(*f*) $y_n = x_n \cdot \mathrm{e}^{n/14}$

(*g*) $y_n = x_n \cdot u_n$

(*h*) $y_n = \sum_{i=-\infty}^{\infty} x_i \cdot \delta_{i-n+2}$

2 Convolution

Exercise 2:

Prove that convolution is

(a) commutative

(b) associative

Exercise 3: MATLAB/Octave commands (similar to)

```
x = [ 0  0  0 -4  0 0  0  0  0 0 2  2 2 2 ...
      2  0 -3 -3 -3 0  0  0  0 0 1 -4 0 4 ...
      3 -1  2 -3 -1 0  2 -4 -2 1 0  0 0 3 ...
     -3  3 -3  3 -3 3 -3  3 -3 0 0  0 0 0 0 ];
n = 0:length(x)-1;
y = filter([1 1 1 1]/4, [1], x);
plot(n, x, 'bx-', n, y, 'ro-');
```

produced the plot on slide 17 to illustrate the 4-point moving average system. The standard library function `filter(b, a, x)` applies to the finite sequence x the discrete system defined by the constant-coefficient difference equation with coefficient vectors b and a (see slide 23 and "`help filter`").

Change this program to generate the corresponding plot for the

(a) exponential averaging system (slide 18)

(b) accumulator system (slide 19)

(c) backward difference system (slide 20)

Exercise 4: A *finite-length sequence* is non-zero only at a finite number of positions. If m and n are the first and last non-zero positions, respectively, then we call $n - m + 1$ the *length* of that sequence. What maximum length can the result of convolving two sequences of length k and l have?

Exercise 5: The length-3 sequence $a_0 = -3$, $a_1 = 2$, $a_2 = 1$ is convolved with a second sequence $\{b_n\}$ of length 5.

(a) Write down this linear operation as a matrix multiplication involving a matrix A, a vector $\vec{b} \in \mathbb{R}^5$, and a result vector $\vec{c}$.

(b) Use MATLAB/Octave to multiply your matrix by the vector $\vec{b} = (1, 0, 0, 2, 2)$ and compare the result with that of using the `conv` function.
Hint: MATLAB's `toeplitz` function may save you some typing effort here.

(c) Use the MATLAB/Octave facilities for solving systems of linear equations to undo the above convolution step.

Exercise 6:

(a) Find a pair of sequences $\{a_n\}$ and $\{b_n\}$, where each one contains at least three different values and where the convolution $\{a_n\} * \{b_n\}$ results in an all-zero sequence.

(b) Does every LTI system T have an inverse LTI system T^{-1} such that $\{x_n\} = T^{-1}T\{x_n\}$ for all sequences $\{x_n\}$? Why?

3 Fourier transform

4 Sampling

Exercise 7: Digital-to-analog converters cannot output Dirac pulses. Instead, for each sample, they hold the output voltage (approximately) constant, until the next sample arrives. How can this behaviour be modeled mathematically as a linear time-invariant system, and how does it affect the spectrum of the output signal?

Exercise 8: Many DSP systems use "oversampling" to lessen the requirements on the design of an analog reconstruction filter. They use (a finite approximation of) the sinc-interpolation formula to multiply the sampling frequency f_s of the initial sampled signal by a factor N before passing it to the digital-to-analog converter. While this requires more CPU operations and a faster D/A converter, the requirements on the subsequently applied analog reconstruction filter are much less stringent. Explain why, and draw schematic representations of the signal spectrum before and after all the relevant signal-processing steps.

Exercise 9: Similarly, explain how oversampling can be applied to lessen the requirements on the design of an analog anti-aliasing filter.

4.1 Band-pass sampling

Exercise 10:

(*a*) Simulate the reconstruction a sampled base-band signal in MATLAB/Octave, following these steps:

- Generate a one second long Gaussian noise sequence $\{r_n\}$ (using MATLAB function `randn`) with a sampling rate of 300 Hz.
- Use the `fir1(50, 45/150)` function to design a finite impulse response low-pass filter with a cut-off frequency of 45 Hz. Use the `filtfilt` function in order to apply that filter to the generated noise signal, resulting in the filtered noise signal $\{x_n\}$.
- Then sample $\{x_n\}$ at 100 Hz by setting all but every third sample value to zero, resulting in sequence $\{y_n\}$.
- Generate another low-pass filter with a cut-off frequency of 50 Hz and apply it to $\{y_n\}$, in order to interpolate the reconstructed filtered noise signal $\{z_n\}$. Multiply the result by three, to compensate the energy lost during sampling.
- Plot $\{x_n\}$, $\{y_n\}$, and $\{z_n\}$, all on top of each other in one figure, and compare $\{x_n\}$ with $\{z_n\}$.

(*b*) Why should the first filter have a lower cut-off frequency than the second?

<u>Exercise 11:</u>

(*a*) Simulate the reconstruction of a sampled band-pass signal in MATLAB/Octave, following these steps:

- Generate a 1 s noise sequence $\{r_n\}$, as in exercise 10, but this time use a sampling frequency of 3 kHz.
- Apply to that a band-pass filter that attenuates frequencies outside the interval 31–44 Hz, which the MATLAB Signal Processing Toolbox function `cheby2(3, 30, [31 44]/1500)` will design for you.
- Then sample the resulting signal at 30 Hz by setting all but every 100-th sample value to zero.
- Generate with `cheby2(3, 20, [30 45]/1500)` another band-pass filter for the interval 30–45 Hz and apply it to the above 30-Hz-sampled signal, to reconstruct the original signal. (You'll have to multiply it by 100, to compensate the energy lost during sampling.)
- Plot all the produced sequences and compare the original band-pass signal and that reconstructed after being sampled at 30 Hz.

(*b*) Why does the reconstructed waveform differ much more from the original if you reduce the cut-off frequencies of both band-pass filters by 5 Hz?

4.2 IQ sampling

<u>Exercise 12:</u> FM demodulation of a single radio station from IQ data:

- The file `iq-fm-96M-240k.dat` (on the course web page) contains 20 seconds of a BBC Radio Cambridgeshire FM broadcast, IQ sampled at the transmitter's centre frequency of 96.0 MHz, at a sample rate of 240 kHz, after having been filtered to 192 kHz bandwidth.
- Load the IQ samples into MATLAB using

```
f = fopen('iq-fm-96M-240k.dat', 'r', 'ieee-le');
c = fread(f, [2,inf], '*float32');
fclose(f);
z = c(1,:) + j*c(2,:);
```

- FM demodulate the complex baseband radio signal `z` (using `angle`)
- apply a 16 kHz low-pass filter (using `butter`, `filter`)
- reduce the sample rate from 240 kHz down to 48 kHz (keep only every 5th sample using the : operator)
- normalize amplitude ($-1 \ldots +1$), output as WAV (`wavwrite`), listen

Exercise 13: FM demodulation of multiple radio stations from IQ data:

- The file `iq-fm-97M-3.6M.dat` contains 4 seconds of Cambridgeshire radio spectrum, IQ sampled at a centre frequency of 97.0 MHz, with 2.88 MHz bandwidth and a sample rate of 3.6 MHz. Load this file into MATLAB (as in exercise 12).
- Shift the frequency spectrum of this IQ signal up by 1.0 MHz, such that the 96.0 MHz carrier of BBC Radio Cambridge ends up at 0 Hz.
- Apply a 200 kHz low-pass filter (`butter`).
- Display the spectrogram of the signal after each of the preceding three steps (using `spectrogram`). How does the displayed frequency relate to the original radio frequency?
- FM demodulate, low-pass filter, and subsample the signal to 48 kHz, and output it as a 16-bit WAV file, as in exercise 12.
- Estimate the centre frequencies of two other FM radio stations within the recorded band (using `spectrogram`), then demodulate these too.

5 Discrete Fourier transform

Exercise 14: Explain the difference between the DFT, FFT, and FFTW.

6 Deconvolution

Exercise 15: Use MATLAB to deconvolve the blurred stars from slide 29.

The files `stars-blurred.png` with the blurred-stars image and `stars-psf.png` with the impulse response (point-spread function) are available on the course-material web page. You may find the MATLAB functions `imread`, `double`, `imagesc`, `circshift`, `fft2`, `ifft2` of use.

Try different ways to control the noise (slide 87) and distortions near the margins (windowing). [The MATLAB image processing toolbox provides ready-made "professional" functions `deconvwnr`, `deconvreg`, `deconvlucy`, `edgetaper`, for such tasks. Do not use these, except perhaps to compare their outputs with the results of your own attempts.]

7 Spectral estimation

Exercise 16: Analog touch tone push-button telephones use a system called dual-tone multi-frequency signaling (DTMF) to communicate to the telephone switch which button is being pressed. Each button produces a combination of two sine tones:

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

(*a*) You receive a digital telephone signal with a sampling frequency of 8 kHz. You cut a 256-sample window out of this sequence, multiply it with a windowing function and apply a 256-point DFT. What are the indices where the resulting vector $(X_0, X_1, \ldots, X_{255})$ will show the highest amplitude if button $\boxed{9}$ was pushed at the time of the recording?

(*b*) The audio file `touchtone.wav` on the course-materials web page contains the recording of a DTMF-encoded sequence of buttons pressed on a telephone. Use the short-term Fourier transform, as implemented by MATLAB's `spectrogram` function, to produce a spectrogram image. Chose a window size that provides a good visual tradeoff for the resulting time and frequency resolution. Then read off which touch-tone button sequence was typed.

8 Digital filters

8.1 FIR filters

Exercise 17: Returning to the problem of decoding the DTMF touch-tone sequence in the `touchtone.wav` file from in exercise 16:

(*a*) Construct seven band-pass FIR filters, with centre frequencies chosen according to the table given in exercise 16 (skipping 1633 Hz, which is not used here). Choose the bandwidth of the filters such that the passband covers up to $\pm 1.8\%$ deviation from the correct frequency. Then pass the same `touchtone.wav` signal through each of these seven filters. Plot the result for each of the seven filters such that you can again read off the sequence of digits pressed.

(*b*) As a third approach to solve the same problem, implement seven AM demodulators, using the same centre frequencies as in the previous part. For each of these channels, down-convert the signal to a complex IQ signal by multiplying it with a phasor, such that the desired centre frequency ends up at 0 Hz. Then low-pass filter (with a cut-off frequency of 1.8% of the centre frequency), take the absolute value, and reduce the sampling rate of the result to 100 Hz, before plotting the output of all eight channels. Again, read off the sequence of digits pressed.

9 IIR filters

Exercise 18: Draw the direct form II block diagrams of the causal infinite-impulse response filters described by the following z-transforms and write down a formula describing their time-domain impulse responses h_i:

(*a*) $H(z) = \dfrac{1}{1 - \frac{1}{2}z^{-1}}$

(*b*) $H'(z) = \dfrac{1 - \frac{1}{4^4}z^{-4}}{1 - \frac{1}{4}z^{-1}}$

(*c*) $H''(z) = \dfrac{1}{2} + \dfrac{1}{4}z^{-1} + \dfrac{1}{2}z^{-2}$

Exercise 19:

(*a*) Perform the polynomial division of the rational function given in exercise 18 (a) until you have found the coefficient of z^{-5} in the result.

(*b*) Perform the polynomial division of the rational function given in exercise 18 (b) until you have found the coefficient of z^{-10} in the result.

(*c*) Use its z-transform to show that the filter in exercise 18 (b) has actually a finite impulse response and draw the corresponding block diagram.

Exercise 20: Consider the system $h : \{x_n\} \to \{y_n\}$ with $y_n + y_{n-1} = x_n - x_{n-4}$.

(*a*) Draw the direct form I block diagram of a digital filter that realises h.

(*b*) What is the impulse response of h?

(*c*) What is the step response of h (i.e., $h * u$)?

(*d*) Apply the z-transform to (the impulse response of) h to express it as a rational function $H(z)$.

(*e*) Can you eliminate a common factor from numerator and denominator? What does this mean?

(*f*) For what values $z \in \mathbb{C}$ is $H(z) = 0$?

(*g*) How many poles does H have in the complex plane?

(*h*) Write H as a fraction using the position of its poles and zeros and draw their location in relation to the complex unit circle.

(*i*) If h is applied to a sound file with a sampling frequency of 8000 Hz, sine waves of what frequency will be eliminated and sine waves of what frequency will be quadrupled in their amplitude?

10 Random sequences and noise

11 Audiovisual data compression