Applied Signal Processing Laboratory

Assignment 1 - Fundamentals of spectral analysis

Daniel Gaetano Riviello

A.A. 2024-2025

19/03/2025

Exercise 1 - Convolution custom function and effect of zero padding on DTFT

1. Create a function called **`my_conv`** that computes the linear or circular convolution of two sequences according to the user's choice.
2. Input arguments of the function are the two signal vectors and, if specified, a char for the choice, `'l'` for linear convolution, `'c'` for circular convolution. If no choice is specified, the function must compute the linear convolution.
3. The function definition starts with `function z = my_conv(x,y, varargin)` and it has to be saved with filename `my_conv.m`. The parameter `varargin` is an input variable that enables the function to accept any number of input arguments.
4. Use the variable `nargin` to check the total number of given input arguments.
5. The variable `varargin` is a $1 \times N$ cell array where N is the number of inputs that the function receives after the explicitly declared inputs.

Exercise 1

6. Neither `conv` nor `cconv` Matlab built-in functions can be used. Only one `for` loop can be used for each convolution.
7. Given the sequences `x` of length N and `y` of length M, the output `z` must have a length of $N + M - 1$ in case of linear convolution and $\max(N, M)$ in case of circular convolution.
8. Create an empty script and test the function with two random sequences of different length as input signal vectors. Plot the linear and circular convolution in separate figures and compare them with the output of the `conv` and `cconv` commands (use `hold on` for comparison).
9. Now create a triangular sequence $x(n)$ of $N = 45$ samples. Generate it as the linear convolution of 2 rectangular sequences of length $M = \frac{N+1}{2}$ and amplitude $1/\sqrt{M}$. Use **`my_conv`** to generate the triangular sequence. Plot it with `stem`.

Exercise 1

10 Compute the DFT $X(k)$ of $x(n)$ with the `fftshift(fft(x))` command. Since N is odd, the vector of frequencies is bounded between $-\frac{1}{2}+\frac{1}{2N}$ and $\frac{1}{2}-\frac{1}{2N}$ with step $\frac{1}{N}$. Plot $|X(k)|$.

11 Add zeros to the triangular sequence in order to reach $N_1 = 64$, $N_2 = 128$ and $N_3 = 256$ total samples. Recompute the new vector of frequencies with new bounds (N is now even) from $-\frac{1}{2}$ and $\frac{1}{2}-\frac{1}{N}$ with step $\frac{1}{N}$ and the DFT of the three new sequences. Plot them against the DTFT $X(e^{j2\pi f})$ of $x(n)$, defined as:

$$X(e^{j2\pi f}) = \frac{1}{M}\mathrm{DTFT}\left\{\mathrm{rect}_M(n) * \mathrm{rect}_M(n)\right\} = \frac{\sin^2(\pi f M)}{M\sin^2(\pi f)}.$$

Use `stem` for $|X(k)|$ and `plot` for the DTFT with at least 4000 points for the vector of frequencies f of the DTFT.

Exercise 2 - Analog time- and band-unlimited signal

1. Consider the following analog signal:
$$x(t) = 32\,t\,e^{-8t}H(t)$$
Define it in MATLAB as a symbolic expression, use `syms` for the variable t. Assume t real.
2. Compute its energy and check the result in MATLAB by means of symbolic integration (use `int`).
3. Find the CTFT $X(f)$ of $x(t)$ with the help of the table of Fourier transform pairs and properties. Check if the transform is correct by using `fourier` (symbolic Fourier transform). First, set the default Fourier parameters as `sympref('FourierParameters', [1 -2*pi])` and use `f` as transformation variable (create a new symbolic variable for `f`).

Exercise 2

4. Estimate the signal bandwidth which contains 99.9% of the energy:

$$\int_{-B_x}^{B_x} |X(f)|^2 \, \mathrm{d}f = 0.999 \int_0^{\infty} |x(t)|^2 \, \mathrm{d}t$$

   Define a new symbolic variable for the bandwidth B_x and assume it real and positive. Use first `int` to find the symbolic expression of the leftside definite integral.

5. Finally solve the above equation with `vpasolve` to find a numerical solution for B_x, use `eval` to convert it to a double variable.
6. Use $f_c = 6\, B_x$.
7. Find T_0 by considering close to zero the condition $x(t) < 10^{-6}$. Use `solve` and then `eval` with the positive solution.

Exercise 2

8. Compute N and round it up to the nearest next power of 2 (use `nextpow2`).
9. Update T_0 accordingly.
10. Use `matlabFunction` to convert the symbolic expressions $x(t)$ and $X(f)$ into MATLAB function handles.
11. Create the discrete-time signal $x(n)$ of $x(t)$ with N samples and $t_c = 1/f_c$, plot it with `stem` and compare it against $x(t)$, use `plot` with $M = 100N$ samples for the latter.
12. Compute the DFT $X(k)$ of $x(n)$, plot the magnitude with `stem` and compare it against $|X(f)|$ (use `plot` with M points from $-f_c/2$ to $f_c/2 - f_c/M$).
13. Compute Δf.
14. Look at the last comparison. Do $|X(k)|$ and $|X(f)|$ perfectly match for all frequencies? If not, explain why.

Exercise 3 - Band-limited signals

1. Generate the following bilateral sequence (with even symmetry):

$$x(n) = 2B_1\text{Sinc}\left(2B_1 t\right) + \left[4B_2\text{Sinc}(2B_2 t) - 3B_2\text{Sinc}^2\left(B_2 t\right)\right]\cos\left(2\pi f_0 t\right)$$

   with $t = nt_c$, $t_c = 1/f_c$, $n = -\frac{N}{2}, \ldots, \frac{N}{2} - 1$.

2. Use the following initial parameters:

B_1	B_2	f_0	f_c	N
16 Hz	6 Hz	10 Hz	64 Hz	128 samples

3. Find the values of T_0 and Δf.
4. Consider the following function in the frequency domain:

$$Y(f) = \text{rect}_{2B_1}(f) + \text{rect}_{2B_2}(f - f_0) + \text{rect}_{2B_2}(f + f_0) - \frac{3}{2}\left[\text{tri}\left(\frac{f - f_0}{B_2}\right) + \text{tri}\left(\frac{f + f_0}{B_2}\right)\right]$$

   where $\text{tri}\left(\frac{f}{B}\right)$ indicates a triangular function with base $2B$.

Exercise 3

5. Compute the DFT $X(k)$ of $x(n)$, plot with `stem` $|X(k)|$ and compare it against $Y(f)$ (use `plot` for the latter with $M = 100N$ points from $-f_c/2$ to $f_c/2 - f_c/M$ and step f_c/M). Use `triangularPulse` and `rectangularPulse` to generate the triangular and rectangular functions.
6. Now generate the sequences $x_a(n)$ and $x_b(n)$ by zero-padding $x(n)$ with 128 and 384 more samples respectively and repeat the plots (keep M as in the previous point). What is the new Δf?
7. Do $|X(k)|$ (or $|X_a(k)|$ or $|X_b(k)|$) and $Y(f)$ exactly match? Recall that the equivalent continuous-time signal of $x(n)$ is:

$$x(t) = x(n)\,\mathrm{rect}_{T_0}(t)$$

Exercise 3

8. Implement the analog convolution in the frequency domain to compute $Z(f) = Y(f) * W(f)$, where $W(f)$ is the Fourier transform of $\text{rect}_{T_0}(t)$ (use the tables).

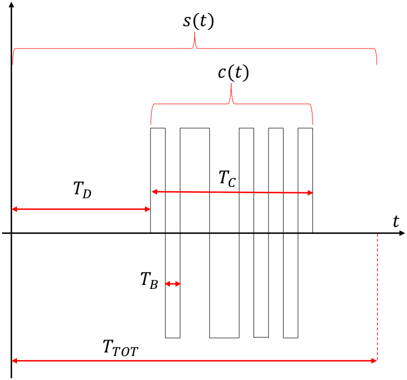
9. Use `find` to get initial and final indexes of $Y(f)$ different from zero, use those indexes with the frequency axis to find the total passband bandwidth B_w of $Y(f)$. Generate $\tilde{Y}(f)$ as the signal portion of $Y(f)$ different from zero. Represent $W(f)$ with a total bandwidth $f_c + B_w$, i.e. compute the length of the $W(f)$ vector as $Q = M\left(1 + \frac{B_w}{f_c}\right)$ with frequency axis ranging from $-\frac{f_c+B_w}{2}$ to $\frac{f_c+B_w}{2} - \frac{f_c+B_w}{Q}$. Use a `for` loop and compute the convolution integral with `trapz`. The function to be integrated is the product between $\tilde{Y}(f)$ and a slided portion of $W(f)$ at each iteration (use the sliding window technique on $W(f)$).

10. Finally plot the magnitude of the result of the convolution $|Z(f)|$ and check if it is more accurate w.r.t. $|X(k)|$.

Exercise 4 - Delay and frequency estimation through correlation

1. Consider the following signal:

$$x(t) = s(t) + w(t)$$



Where:

- $c(t)$ is a signal for remote sensing systems (i.e., radar or CDMA codes) and it is constructed from a pseudorandom binary sequence.
- $s(t) = c(t - T_D)$.
- T_D is the parameter (delay) to be estimated.
- $w(t)$ is a zero-mean Gaussian process with $\sigma_n^2 = 50$.
- $T_{\text{TOT}} = 5$ s.
- $T_C = 2$ s.
- $T_B = 1$ ms.

Exercise 4

2. Set the sampling frequency $f_c = 4$ kHz and the sampling interval $t_c = 1/f_c$.
3. Generate the sequence $b[k],\ k = 1, 2, \ldots, B$, as a Bernoulli process $\{0, 1\}$ of $B = \left\lfloor \frac{T_C}{T_B} \right\rfloor$ random variables with `binornd` and map the values $\{0, 1\}$ to $\{-1, 1\}$.
4. Generate $c(t)$ from $b[k]$ by considering that each kth sample of $b[k]$ has a duration equal to T_B in $c(t)$ (use `repmat` and vectorize the resulting matrix). Compute the number of samples M of $c(t)$.
5. Generate the delay T_D as a uniform random variable (`help rand`) between 0 and $[T_{\text{TOT}} - T_C]$. Round it to the third decimal place (`help round`).
6. Generate $s(t)$ by zero-padding $c(t)$ on the left (samples corresponding to T_D) and on the right (samples corresponding to $[T_{\text{TOT}} - T_C - T_D]$).
7. Generate $w(t)$ with `randn` and finally generate $x(t)$.

Exercise 4

7. Plot in one figure $s(t)$ and $x(t)$ (use a 2×1 `subplot`).
8. If d represents the number of samples corresponding to T_D, the delay in samples $\hat{d}$ can be estimated as:

$$\hat{d} = \arg\max_{m} \sum_{n=0}^{N-1} x\left[nt_c\right] \, c\left[(n-m)t_c\right]$$

   where N is length in samples of $s(t)$.
9. Write a custom function `my_xcorr` to compute the cross-correlation expressed above between 2 sequences. The function has two vectors as input arguments and returns the cross-correlation between them. Notice that if $a[n]$ has length A and $b[n]$ has length B, the cross-correlation will have length $(A + B - 1)$.
   - It is allowed to use at most one for loop.

Exercise 4

10 Plot the result of the cross-correlation. Set the x-axis (lags) in seconds instead of samples (in samples it would go from $-M+1$ to $N-1$).

11 Find the delay in samples $\hat{d}$ with `max`.

12 Finally find the delay in seconds $\hat{T}_D$ and check if it matches the previously generated delay T_D.

13 Re-run the script with $\sigma_n^2 = 100$ and re-plot the cross-correlation.

14 Is your algorithm still able to correctly estimate the delay?

Exercise 4

15 Consider the signal and the parameters described in slide 11 with the following differences:

- $c(t)$ is now replaced by $p(t) = c(t)\cos(2\pi f_0 t)$.
- $s(t) = p(t - T_D)$, where T_D is the same delay generated in bullet 5.
- f_0 is the frequency parameter to be estimated.
- $\sigma_n^2 = 50$ and $y(t) = s(t) + w(t)$.

16 Generate f_0 with `randi` by assuming it can take only one of the following values

$$f_0 = p \cdot 10\,\mathrm{Hz} \qquad p = 1, 2, \ldots, P \qquad P = 20$$

17 Jointly estimate the delay in samples and the frequency f_0 with the following expression:

$$\left[\hat{d}, \hat{f}_0\right] = \arg\max_{m, f_0} \sum_{n=0}^{N-1} y[nt_c]\, p[n-m] =$$

$$= \arg\max_{m, f_0} \sum_{n=0}^{N-1} y[nt_c]\ \left(c\left[(n-m)t_c\right] \cos\left[2\pi f_0 (n-m) t_c\right]\right)$$

Exercise 4

18 You need to perform P cross-correlations. Use an outer `for` loop to cycle the possible frequencies and call `my_xcorr` at each iteration. Save the result of all cross-correlations in a $P \times (N + M - 1)$ matrix $\mathbf{A}$.

19 Plot the 3-D surface of all cross-correlations in $\mathbf{A}$ with `mesh` or `surf`. Set the x-axis (lags) in seconds and the y-axis in Hz.

20 Use `max` twice (type `help max`) to find the cross-correlation peak in $\mathbf{A}$. In the first `max` call, save the indexes of the peaks along one dimension to finally retrieve $\hat{d}$ in samples and $\hat{f}_0$ with the second `max` call.

21 Compute the delay in seconds $\hat{T}_D$ from $\hat{d}$, check if $\hat{T}_D$ and $\hat{f}_0$ match the previously generated delay T_D and frequency f_0.

22 Re-run the script with $\sigma_n^2 = 100$ and re-plot the surface.

Exercise 5 - Spectral estimation

1. Consider the following signal:

   $$X(t) = \frac{20}{\sqrt{2\pi}\eta} \exp\left(-\frac{t^2}{2\eta^2}\right) \cos(2\pi f_1 t) + \cos(2\pi f_2 t) + \cos(2\pi f_3 t) + W(t)$$

   with $\eta = 0.01$, $f_1 = 100$ Hz, $f_2 = 500$ Hz, $f_3 = 510$ Hz, sampling frequency $f_c = 2$ kHz.
2. $W(t)$ is a zero-man white Gaussian noise with variance $\sigma_n^2 = 25$.
3. Generate 20 seconds of the signal. Use `randn` for $W(t)$.
4. Estimate the power spectral density $S_x(f)$ with the Welch periodogram with the following command:

   ```
   [Sx, f] = pwelch(x, window, n_overlap, NFFT, fs, 'centered');
   ```

   where initial parameters are:
   - `NFFT=128;`
   - `window=hamming(NFFT);`
   - `n_overlap=0;`
5. Plot the power spectral density (PSD) `Sx` in logarithmic scale, i.e., `f`, the vector of frequencies versus $10\log_{10}|S_x(f)|$.

Exercise 5

6. Why can't we see the two separate harmonic contributions?
7. Adjust the parameters of `pwelch` in order to decrease the estimation variance and to maintain a good spectral resolution. Take into account that:
   - `NFFT` is the number of samples for each segment.
   - `n_overlap` is the number of overlapping samples and has to be smaller than `NFFT`.
   - `window` can be `hamming(NFFT)`, `hann(NFFT)` or `rectwin(NFFT)`.

   Report the chosen parameters and plot your estimation.
8. Set the parameters in order to obtain:
   - a simple periodogram;
   - a Bartlett periodogram with $M = 25$ segments;
   - a Welch periodogram with Hamming window, $D = N/M$ samples and 50% overlap.

   Plot the 3 periodograms in the same figure with `hold on`. Discuss the variance and spectral resolution of each method.

Exercise 5

9. Estimate the autocorrelation function of $X(t)$ by using the two estimators $\hat{R}_N[l]$ and $\hat{R}'_N[l]$ for the autocorrelation presented in Lecture 5. You can use the `xcorr` command with the option `SCALEOPT` set to `'unbiased'` or `'biased'` to normalize the autocorrelation according to the first or second method respectively.
10. Plot both estimations with a 2×1 `subplot` in a single figure. Discuss the differences.
11. Filter the signal $X(t)$ with a low-pass filter by using the following code:

```
N = 165; M = floor(N/2); n = (-M:M)';
h = 0.3*sinc(0.3*n).*hamming(N);
y = filter(h, 1, x);
```

Exercise 5

12 Compute now for $Y(t)$, the output of the filtering operation on $X(t)$, the Welch periodogram with the 3 windowing functions: rectangular, hamming and hann with 50% overlap and $D = 1600$. Use `rectwin`, `hamming` and `hann` MATLAB functions. Plot the 3 periodograms in the same figure (use hold on). Which one has the lowest out-of-band attenuation?

13 Create a function called `my_PSD` that computes the simple or Bartlett periodogram of a signal vector `x` of length N.

14 Input compulsory parameters of the function are the signal vector and the length N. Optional parameter is the number of segments M (use `varargin`). If no optional parameter is given, the function outputs the simple periodogram, otherwise the Bartlett one.

15 For the simple periodogram:
- compute the squared magnitude of the spectrum of `x` and normalize it by N.
- Return the periodogram vector and a vector of normalized frequencies from $-\frac{1}{2}$ to $\frac{1}{2} - \frac{1}{N}$ with step $\frac{1}{N}$.

Exercise 5

16 For the Bartlett periodogram:

- check first if N is a multiple of M, if that is not the case, zero-pad the signal vector with L samples so that $\frac{N+L}{M}$ is an integer, compute D as the segment size in samples;
- use a for loop to create a sliding window to process each segment and compute the average periodogram.
- Within the for loop, compute the squared magnitude of the spectrum of each segment and normalize it by D.
- Return the periodogram vector and a vector of normalized frequencies from $-\frac{1}{2}$ to $\frac{1}{2} - \frac{1}{D}$ with step $\frac{1}{D}$.

17 To compute the spectrum, create a custom function called `my_DFT` with input parameters the input vector and the number of discrete frequencies for the DFT. The function outputs the centered spectrum of the input vector (use `circshift`).

- You cannot use any `for` loops or the command `dftmtx`.

Exercise 5

18 Test the function for both periodograms with the filtered sequence $Y(t)$, compare it with the `pwelch` function with the proper parameters for simple and Bartlett periodograms (use sampling frequency $f_c = 1$).

19 Plot in one figure the 2 simple periodograms and in another figure the 2 Bartlett periodograms in logarithmic scale (use `hold on` in each figure).

20 Both functions are local, i.e., they are contained in the same main script.

Report and Matlab scripts

- For each exercise include all requested plots and answers.
- Plots must contain labels for all axes, a title and a legend in case of multiple plots in one figure.
- The scripts must run correctly. If the script of an exercise doesn't work, the exercise will be considered failed.
- Deliver a separate Matlab file for each exercise (not a single Matlab file for the entire assignment). For exercise 1 and 4, the main script plus the function (2 separate `.m` files).
- Both the report and the Matlab files must be uploaded on the portal in a zip file.
- Naming rule: `Assignment1_lastname1_lastname2.zip`.
- Send an email to daniel.riviello@polito.it when you upload it.

Report delivery and deadlines

- Deliver a single pdf report for Assignment 1 “Fundamentals of spectral analysis”.
- The report must include all requested plots, comments and answers for all exercises (1 to 5).

Deadlines to get extra points for Assignment 1

- **Tue. 01/04/2025** at 23:59 for 1 point.
- **Tue. 08/04/2025** at 23:59 for 0.5 point.