Lab 1: Autocorrelation and Pitch Tracking

Due Friday, February 1st

Overview:

- This lab should be completed with your assigned lab partner(s).
- Each group must turn in a report composed using a word processor (e.g., Word, Pages, LaTex, etc...). The report should include cover page with both full names. The remaining pages should contain (in order) the answers and MATLAB scripts for the exercises. MATLAB figures can be pasted into the document or saved as PDF files. When working on the project, please follow the instructions and respond to each item listed. Your project grade is based on: (1) your MATLAB scripts, (2) your report (plots, explanations, etc. as required), and (3) your final results. For all labs, you must clearly write the problem number next to your solution and label the axes on all plots to get full credit. Submission can be done electronically in PDF format or on paper.
- Plagiarism is a very serious offense in Academia. Any figures in the paper not generated by you should labeled "Reproduced from [...]". Any portions of any simulation code (e.g., MATLAB, C, etc...) not written by you be clearly marked in your source files. The original source of any mathematical derivation or proof should be explicitly cited.

1 What is Correlation?

For DT energy-type signals, the *cross correlation* between $x[n]$ and $y[n]$ with lag ℓ is defined to be

$$r_{xy}[\ell] \triangleq \sum_{n=-\infty}^{\infty} x[n]y^*[n-\ell] = \sum_{n=-\infty}^{\infty} x[n+\ell]y^*[n].$$

This quantity measures how closely the two signals match each other when they are shifted and scaled. In particular, the squared Euclidean distance satisfies

$$\begin{aligned}\sum_{n=-\infty}^{\infty} |x[n]-Ay[n-\ell]|^2 &= \sum_{n=-\infty}^{\infty} |x[n]|^2 + |A|^2 \sum_{n=-\infty}^{\infty} |y[n-\ell]|^2 \\ &\quad - \sum_{n=-\infty}^{\infty} \left(A^* x[n]y^*[n-\ell] + Ax^*[n]y[n-\ell]\right) \\ &= E_x + |A|^2 E_y - 2Re\left\{A r_{xy}^*[\ell]\right\},\end{aligned}$$

where E_x is the energy in $x[n]$ and E_y is the energy in $y[n]$. Moreover, the minimum over A equals $E_x - |r_{xy}[\ell]|^2/E_y$ and is achieved by $A = r_{xy}[\ell]/E_y$.

This operation is very useful if one is trying to find a scaled and shifted copy of one signal as a component of another. For example, consider the case where $y[n] = Bx[n-n_0]$ for a known $x[n]$ and unknown parameters B and n_0. The autocorrelation can be used to compute the parameters.

The *autocorrelation* $r_{xx}[\ell]$ is defined to be the cross-correlation between $x[n]$ and itself (i.e., $r_{xy}[\ell]$ when $y[n] = x[n]$). The maximum autocorrelation is always $r_{xx}[0] = E_x$ and, hence, the *normalized autocorrelation* is defined to be $r_{xx}[\ell]/r_{xx}[0]$. Also, autocorrelation of a periodic signal with period N will take its maximum value of E_x when ℓ is an integer multiple of N. Thus, the autocorrelation is very useful for detecting periodicity in a signal. Also, if $y[n] = x[n] + Bx[n-n_0]$ has an echo with time delay n_0, then autocorrelation can be used to estimate B and n_0.

For some of the estimation problems mentioned above, it is important that the signal $x[n]$ have an autocorrelation function that is sharply peaked around lag 0. To understand the autocorrelation of various signals, use Matlab to compute and plot the autocorrelation of the following signals:

(a) $x[n] = \delta[n]$. The autocorrelation is sharply peaked but we will see why this waveform is not used in practice.

(b) $x[n] = \cos(\frac{2\pi n}{20})$ for $-30 \le n \le 30$. What do you observe about the autocorrelation in this case?

(c) $x[n] = \frac{2}{\pi n}\sin(\frac{\pi n}{2})$ for $-N \le n \le N$ with $N = 50$ and $N = 250$. How does $r_{xx}[\ell]$ compare with $x[\ell]$? Can you explain this?

2 Ultrasonic Measuring Tape

As an engineer at Ultrasonics Unlimited, your goal is to design a system that transmits some ultrasonic waveform, listens for the echo from nearby reflecting objects, and determines the range to the nearest reflecting object. Let us assume that the system operates in discrete-time with a sampling frequency of 100 KHz or $T = 10^{-5}$s. Let $x[n]$ be the transmitted waveform and

$$y[n] = \sum_{i=1}^{P} a_i x[n - k_i].$$

be the received waveform consisting of P reflections each with delay k_i and coefficient a_i. This model is based on the idea that one starts transmitting $x[n]$ and receiving $y[n]$ at the same time (say $n = 0$) using a receiver completely isolated from the transmitter. For example, if the channel is defined by $P = 1$, a_1 =1, and $k_1 = 20$, then $y[n] = x[n - 20]$ is simply a delayed version of $x[n]$.

To find the delays and coefficients, one can cross-correlate $y[n]$ with $x[n]$ because

$$\begin{aligned} r_{yx}[\ell] &= \sum_{n=-\infty}^{\infty} \sum_{i=1}^{P} a_i x[n - k_i] x[n - \ell] \\ &= \sum_{i=1}^{P} a_i r_{xx}[k_i - \ell]. \end{aligned}$$

Since $r_{xx}[0] = E_x \geq r_{xx}[\ell]$ for $\ell \neq 0$, it follows that $r_{yx}[\ell]$ will be large for $\ell = k_i$ for some $i \in \{1, \ldots, P\}$. But, there may other large values that make it difficult to find the true delays. If the transmitted waveform is chosen so that $r_{xx}[0] \gg r_{xx}[n]$ for $n \neq 0$, then one can find the delays k_i by finding the peaks in $r_{yx}[n]$.

(a) One engineer suggests that you choose

$$x[n] = \begin{cases} \cos(2\pi f n) & \text{if } n = 0, \ldots, 99 \\ 0 & \text{otherwise} \end{cases}$$

with $f = 0.1$. Plot this signal and the normalized autocorrelation $r_{xx}[n]/r_{xx}[0]$ (try `help xcorr`) using `stem`. Try using `[ac,lags]=xcorr(x)` to automatically get the correct time indices your $r_{xx}[n]$ plot.

(b) You remember that radars use "chirp" signals estimate distance. A linear chirp is a signal whose frequency increases linearly with time. So you suggest the signal

$$x[n] = \begin{cases} \cos\left(2\pi\left(\frac{n^2}{2\times 800}\right)\right) & \text{if } n = 0, \ldots, 99 \\ 0 & \text{otherwise.} \end{cases}$$

The instantaneous angular frequency of a signal equals the time-derivative of its phase. This implies that the instantaneous frequency of $x[n]$ is $\frac{n}{800}$ cycles per sample and its frequency increases linearly from 0 to $\frac{1}{8}$ cycles per sample. Plot this signal and the normalized autocorrelation $r_{xx}[n]/r_{xx}[0]$ using `stem`. Make sure you choose the time indices correctly for the $r_{xx}(n)$ plot.

(c) Based on your plots, which signal is more suited to this application and why?

(d) For each waveform, simulate the performance of your ultrasonic measuring tape. To be fair, one should normalize both waveforms to unit energy with $\tilde{x}[n] = x[n]/\sqrt{E_x}$. Then, generate a length-10000 vector defined by

$$y[n] = 5\tilde{x}[n - 2000] + z[n],$$

where $z[n]$ is standard Gaussian noise (see `help randn`). Use the largest value of $r_{yx}[n]$ to estimate the delay (see `help max`). Try 100 runs with independent noise samples and compute the fraction of times that each algorithm estimates the delay exactly right. Which waveform is better by this measure?

(e) On the class website, the signal "ultra_test.txt" contains a test vector based on the the Chirp signal. Use this signal as $y[n]$ (try `help load`) to cross-correlate, and estimate the distance to the nearest reflecting object assuming the speed of sound is 1000 feet per second.

3 Autocorrelation of Periodic Signals

For discrete-time power-type signals, recall that the time-average autocorrelation function is defined by

$$\overline{r}_{xx}[\ell] \triangleq \lim_{M\to\infty} \frac{1}{2M+1} \sum_{n=-M}^{M} x[n]x^*[n-\ell] = \lim_{M\to\infty} \frac{1}{2M+1} \sum_{n=-M}^{M} x[n+\ell]x^*[n].$$

Also, the autocorrelation at lag 0 equals the average power

$$\overline{r}_{xx}[0] = \lim_{M\to\infty} \frac{1}{2M+1} \sum_{n=-M}^{M} x[n]x^*[n] = \lim_{M\to\infty} \frac{1}{2M+1} \sum_{n=-M}^{M} |x[n]|^2 = P_x.$$

Now, consider a discrete-time periodic signal $x[n]$ with fundamental period N. Recall that this implies N is the smallest positive integer such that $x[n+N] = x[n]$ for all n. For this signal, it is easy to verify that

$$\begin{aligned}\overline{r}_{xx}[\ell+N] &= \lim_{M\to\infty} \frac{1}{2M+1} \sum_{n=-M}^{M} x[n+\ell+N]x^*[n] \\ &= \lim_{M\to\infty} \frac{1}{2M+1} \sum_{n=-M}^{M} x[n+\ell]x^*[n] \\ &= \overline{r}_{xx}[\ell].\end{aligned}$$

Thus, the autocorrelation function of a signal with period N is periodic with period N. From this, we see that $\overline{r}_{xx}[N] = \overline{r}_{xx}[0]$ and it follows that $\overline{r}_{xx}[\ell] \leq \overline{r}_{xx}[0] = \overline{r}_{xx}[N]$ for all ℓ. Thus, a signal with period N has an autocorrelation peak at lag N and no values larger than $\overline{r}_{xx}[N]$.

4 Pitch Tracking

To estimate the period of a periodic signal in noise, one assumes a fixed range of possibilities $\mathcal{H} = \{N_{min}, N_{min}+1, \ldots, N_{max}\}$ for the period and then computes a score function

$$A_j = \frac{\overline{r}_{xx}[j]}{\overline{r}_{xx}[0]}$$

for each lag $j \in \mathcal{H}$. Conceptually, the value of A_j indicates what fraction of the power can be explained by a signal with period j. A good estimate of the signal period is given by maximizing the score over the range of possibilities

$$\hat{N} = \arg\max_{j\in\mathcal{H}} A_j.$$

A common error for this method is frequency halving, where the score associated with the fundamental period is tied with or slightly smaller than the score associated with twice the fundamental period. To avoid this type of error, one can check if the score for $\lfloor \hat{N}/2 \rfloor$ is not too much lower than the largest score. For example, if $A_{\lfloor \hat{N}/2 \rfloor} \geq 0.8 A_{\hat{N}}$, then the algorithm should return $\hat{N}/2$ instead. Also, the value of $A_{\hat{N}}$ needs to be relatively large (e.g., greater than 0.5) for the signal to be strongly periodic. If the value of the maximum is too small, then the signal is probably not periodic. Naturally, the frequency estimate associated with the period estimate is given by $\hat{F} = F_s/\hat{N}$, where F_s is the sampling rate.

As described, this method has a few problems. First, it is generally infeasible to compute the time-average autocorrelation function and, if we could, the method would only work for perfectly periodic signals. Second, let $x(t)$ be the continuous band-limited interpolation of $x[n]$. If $x(t)$ is periodic with a non-integer period T (e.g., 11.5), then this method may incorrectly estimate the period to be an integer multiple of T (e.g., 23) instead of choosing 11 or 12.

In practice, this approach must also be adjusted to handle finite-time signals that are not periodic but have sections that are almost periodic.

4.1 Practical Method 1: Windowing

The first practical method does not use the time-average autocorrelation. Instead, one delays the signal by n_0, computes $y[n] = w[n]x[n+n_0]$ for some window function $w[n]$, and then uses the autocorrelation $r_{yy}[\ell]$ for energy-type signals. For a length-L block, a simple choice of $w[n]$ is the Hann window

$$w[n] = \begin{cases} \sin^2\left(\frac{\pi n}{L-1}\right) & \text{if } n = 0, \ldots, L-1 \\ 0 & \text{otherwise.} \end{cases}$$

As before, a good estimate of the signal period is given by maximizing the score over the range of possibilities

$$\hat{N} = \arg\max_{j \in \mathcal{H}} \frac{r_{yy}[j]}{r_{yy}[0]}.$$

It is very important to choose the window function to be wide enough that it covers multiple periods. In general, it is recommended that the window length be 3-4 times the longest period N_{max}. Similar to the above discussion, one can avoid frequency halving errors by returning $\hat{N}/2$ if $A_{\lfloor \hat{N}/2 \rfloor} \geq 0.8 A_{\hat{N}}$.

4.2 Practical Method 2: Lag Search

Another approach is to use fixed-length correlation to search over the set of possible lags. First, you fix a correlation length L (e.g., to roughly twice the largest period N_{max}). To estimate the pitch of signal $x[n]$ at time n_0, one can compute

$$A_j = \sum_{n=0}^{L-1} x[n+n_0] x^*[n+n_0-j]$$

for $j \in \mathcal{H} = \{N_{min}, N_{min}+1, \ldots, N_{max}\}$. Once again, one estimates the signal period by maximizing this over the range of possibilities

$$\hat{N} = \arg\max_{j \in \mathcal{H}} A_j.$$

Similar to the above discussion, one can avoid frequency halving errors by returning $\hat{N}/2$ if $A_{\lfloor \hat{N}/2 \rfloor} \geq 0.8 A_{\hat{N}}$.

4.3 Lab Exercises

(a) Write a Matlab function `ac_pitch(x,L,M,Fs,Fmin,Fmax)` that estimates the pitch of $x[n]$ at positions $n_0 = M(k-1)+1$ for $k = 1, 2, 3, \ldots$ using a window of size L. The parameters F_s, F_{min}, and F_{max} denote the sampling rate, the minimum search frequency, and the maximum search frequency. When truncating $x[n]$ at a particular n_0, it is better to truncate to a length $2L$ so that while computing the autocorrelation, a sub-window of size L can be moved within this $x[n]$, corresponding to each lag, for computing the dot product with `x(1:L)`. So, the last value of k above should be chosen so that just about $2L$ samples are left in $x[n]$ starting from the corresponding time index n_0. The function should return a vector of length `ceil((length(x)-2*L)/M)+1` that contains the frequency estimate (in Hz) for each value of n_0.

(b) To test your program, start by choosing $F_s = 22050$, $F_{min} = 100$, and $F_{max} = 200$. Then, let $L = 2\lceil F_s/F_{min} \rceil$ and $M = L/2$. Next, try it out with the constant frequency signal $x[n] = \cos\left(2\pi n(F_{min}+F_{max})/(2F_s)\right)^3$. Finally, try it with a chirp-like signal whose instantaneous frequency changes linearly from F_{min} to F_{max} but changes only by 1 Hz over the window size L. Thus, the length of the signal will be $Q = (F_{max} - F_{min})L$ and the signal will be given by

$$x[n] = \begin{cases} \cos\left(\frac{2\pi F_{min}}{F_s} n + \frac{2\pi(F_{max}-F_{min})}{QF_s} \frac{n^2}{2}\right)^3 & \text{if } n = 1, \ldots, Q \\ 0 & \text{otherwise.} \end{cases}$$

This function is a chirp that is raised to the 3rd power to add a few higher harmonics. Plot the true instantaneous frequency versus the output of your program.

(c) Next, record yourself singing (or playing on some instrument a few different notes). Then, try your program on this file. Make sure that F_{min} and F_{max} are set correctly to capture the true values. Plot the program output versus the known true values (e.g., you should know what notes you played). For instruments, a sampling rate of $F_s = 44100$ Hz is recommended.

(d) (Optional) Compare the results of your program with the publicly available program "Yet Another Algorithm for Pitch Tracking" (YAAPT) from:

`http://www.ws.binghamton.edu/zahorian/yaapt.htm`

5 Additional Information

To achieve a larger frequency resolution (and avoid the integer period problem discussed above), the easiest method is to first oversample the signal. For example, using band-limited interpolation by a factor of 2 to 4 may allow one to resolve the frequency more accurately. If high accuracy is required, it is also important to choose longer window lengths (e.g., 5-10 times the longest period N_{max}).

There are many ways to improve this basic pitch-tracking implementation. First, one can also estimate whether or not there is a periodic component and suppress the estimate if there it not. Also, the pitch of signal generally changes slowly and an outer tracking loop can be used to reduce the error rate due to artifacts. For example, one might try using a median filter to smooth the estimates or dynamic programming to discourage large jumps in the pitch.