Fourth set of exercises

1. DSP processors are missing op-codes for various standard operations, such as division, square-root, and trigonometric functions. In this exercise we will see how DSP programmers overcome this.

   (a) To perform division it is enough to calculate the inverse $y = 1/x$. We do this by starting with a guess y_0 and iterating $y_{n+1} = y_n(2 - xy_n)$. With a reasonable initial guess this only takes a few iterations to converge. (Try it!) Why does this work? **Note that when $\mathbf{xy_n = 1}$ then $\mathbf{y_{n+1} = y_n}$ so the correct answer is a *fixed point*. Since the function is quadratic in $\mathbf{y_n}$ there can only be 2 fixed points, one of which is $\mathbf{y_{n+1} = y_n = 0}$ and the other is the correct answer. As long as $\mathbf{x > 1}$ the sign of the correction is opposite the sign of the error (check by substituting $\mathbf{y_n = 1/x + \epsilon}$, so the correct answer is an *attractor*, i.e., if we start close enough we get even closer with each iteration.** Is this a *filter*? **No, it is AR but not linear.** How many clocks are required to calculate it using a DSP? **One MAC cycle for the $(\mathbf{2 - xy_n})$ and one more for the multiplication. Altogether 2 clocks per iteration.**

   (b) We often require the inverse square-root $y = 1/\sqrt{x}$. We could first find the square-root using the Newton-Raphson method $y_{n+1} = (y_n + x/y_n)/2$ and then invert. Why is this problematic? **It requires a division!** A better method uses $y_{n+1} = y_n(3 - xy_n^2)/2$. (Try it!) Why does this work? **Note that when $\mathbf{xy_n^2 = 1}$ then $\mathbf{y_{n+1} = y_n}$ so the correct answer is a *fixed point*. Similarly to what we did in the previous section, we can show that it is also an *attractor*.** Is this a *filter*? **No, it is AR but not linear.** How many clocks are required to calculate it using a DSP? **One multiplication for $\mathbf{y_n^2}$, another for the multiplication by x and one more for the subtraction, and one for the multiplication. We'll count one cycle for the shift, although some DSPs may be able to do it without an additional cycle. Altogether 5 clocks per iteration.**

   (c) We often require $e^{\mathrm{i}\omega n}$ which requires both $\sin(\omega n)$ and $\cos(\omega n)$ for consecutive values of n. If we start with the known values for $n = 0$ and precalculate $e^{\mathrm{i}\omega} = \cos\omega + \mathrm{i}\sin\omega$, we can find the new values by iterating $e^{\mathrm{i}\omega(n+1)} = e^{\mathrm{i}\omega}e^{\mathrm{i}\omega n}$. Write explicitly the update formulae in matrix form.

$$\begin{pmatrix} \mathbf{\cos(\omega)} & \mathbf{-\sin(\omega)} \\ \mathbf{\sin(\omega)} & \mathbf{\cos(\omega)} \end{pmatrix} \begin{pmatrix} \mathbf{\cos(\omega n)} \\ \mathbf{\sin(\omega n)} \end{pmatrix}$$

   What kind of matrix is this? **A rotation matrix.** Why is that obvious? **Each increment of n raises the base $e^{\mathbf{i}\omega}$ to the next power, which is a rotation.** How many real multiplications and additions are required for each update? **Four multiplies and two adds.** What is the problem with this method? (Hint: we encountered this problem when discussing ADPCM!) **Error accumulation.**

2. LPC analysis of speech involves several steps on a frame of digitized speech. Explain how each of these can be carried out.

   (a) Deciding whether the signal is voiced or unvoiced. **By observing whether the spectrum is continuous or lines.**

   (b) Finding the gain parameter (or the signal's energy). **From the definition of energy – sum the squared signal values!**

   (c) Finding the pitch. **from autocorrelation**

   (d) Finding the AR filter coefficients. **This is hard system identification problem for an AR system – solve Yule Walker equations!**

3. The *cepstrum* of a signal is defined to be the inverse Fourier transform of the logarithm of the signal's spectrum (i.e., of the Fourier transform of the signal).

   (a) *Cepstrum* is a purposeful distortion of the word *spectrum*. Where do you think *quefrency, liftering, maplitude,* and *saphe* come from? **Frequency, filtering, amplitude, and phase, respectively.**

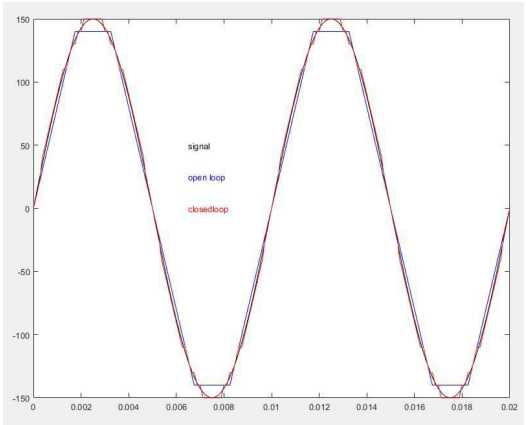
   (b) The *power cepstrum* is defined as the squared magnitude of the inverse Fourier transform of the logarithm of the squared magnitude of the signal's power spectrum (i.e., the Fourier transform of the signal). Why is using the power cepstrum easier than using the full cepstrum? (Hint: think about the logarithm.) **Taking the logarithnm of a non-negative real number avoids the multi-branched complex logarithm.**

(c) Explain why cepstral analysis is useful for voiced speech signals (based on the LPC model and the fundamental property of filters in the frequency domain). How can liftering be used to isolate the pitch? **Speech is generated by AR filtering the vocal-chord excitation signal, which in the frequency domain means multiplying the excitation signal by the frequency response of the vocal tract. After the logarithm this multiplication becomes an addition,and since the iFFT is linear, the cepstrum is the sum of two components - one due to the vocal chords (the pitch component), and the other to the vocal tract (the formant component). By liftering we may be able to separate these components, and read off the pitch and the formants.**

(d) Cepstrum was originally invented to handle seismic echoes. What can be said about the spectrum of a signal polluted by a strong echo, such as $y_n = x_n + x_{n-d}$? (Hint: Will there be frequencies that are attenuated or amplified?) **If after d times the sinusoid is 180 degrees out of phase, then it cancels out. If it is back in phase then there is constructive interference.** What is the frequency response of this echo system? **Write** $\mathbf{ye^{i\omega n} = e^{i\omega n} + e^{i\omega(n-d)}}$ **and find** $H(\omega) = \cos(\omega d/2)$**.** How can cepstral analysis help find the distance to a seismic event when the input spectrum is complicated? **The frequency response itself is sinusoidal so we need to Fourier analyze it.**

4. In the lecture we saw that *ADPCM* eliminates *error accumulation* by including the decoder inside the encoder (often called *closed loop* encoding). In this exercise we will see how this works. Assume that quantization restricts all encoded values to be multiples of $q = 10$. What happens if we perform (nonadaptive) DPCM on the signal $x_n = 7n$? With *open loop* encoding the encoder calculates the first finite difference signal $d = \hat{\Delta}x$, quantizes it to create the encoded signal $c = q\text{round}(d/q)$ and sends c to the decoder. The decoder constructs its estimate of the signal $y^{[1]}$ by accumulating the received encoded signal $y^{[1]} = \hat{\Upsilon}c$. With *open loop* encoding the encoder calculates, at each time n, the difference between x_n and the previously *reconstructed* signal value, quantizes, and sends the encoded signal c' to the decoder. The decoder constructs its estimate of the signal $y^{[2]}$ by accumulating the received encoded signal $y^{[2]} = \hat{\Upsilon}c'$. Compare the first few values of $y^{[1]}$ and $y^{[2]}$. Plot the results. What happens to the error over time?

The true signal is $\mathbf{x_n = 0, 7, 14, 21, 28, \ldots}$**, so the first finite difference is** $\mathbf{d = \hat{\Delta}x = 7, 7, 7, 7, 7, \ldots}$**. The quantized difference is** $\mathbf{c = 10, 10, 10, 10, 10, \ldots}$**, so the open loop reconstruction is** $\mathbf{y^{[1]} = \hat{\Upsilon}c = 0, 10, 20, 30, 40, \ldots = 10n}$**. The error increases linearly with n, since all the errors equal. In more general cases the error will sometimes be positive and sometimes negative (a *random walk*), so the accumulated error will increase more faster than the square root of n. For the closed loop case the decoder will also reconstructed the first value as** $\mathbf{y_1^{[2]} = 10}$**, but then the second value** $\mathbf{x_2 = 14}$ **will be encoded with respect to 10, i.e., the difference will be 4 which will be quantized to 0 and the reconstructed value will be** $\mathbf{y_2^{[2]} = 10}$**. The next value** $\mathbf{x_3 = 21}$ **will be compared to 10 and the quantized difference will be 10 so that** $\mathbf{y_3^{[2]} = 10 + 10 = 20}$**. So, the reconstruction will be** $\mathbf{y^{[2]} = 0, 10, 10, 20, 30, \ldots}$**. The reconstruction error never exceeds 5.** Repeat this for a sinusoidal signal of sampled at the rate a 100 Hz. pitch frequency would be sampled in a digital telephone channel, for example, take $x_n = 150\sin(2\pi(100\text{Hz}/8000\text{Hz})n)$. Plot the original signal along with the open and closed loop reconstructions.



5. In the lecture we mentioned that the inefficiency of the speech system is partially due to the transmitter and receiver being unmatched, i.e., not being inverse systems. What is the main signal processing function in the transmitter? What would be its inverse system? **The main function is AR filtering. The inverse of an AR filter is an MA filter. The required information extraction function is finding the peaks of the AR frequency response.** What is the main signal processing function in the receiver? What would be its inverse system? **The main function is the Fourier transform or a *filter bank*. The inverse would be the iFT. The required information insertion function would put basic information into frequency bins (in the simplest case, either place a tone or not.** Another inefficiency problem is the inability of the human speech production organs to react quickly enough. Is the problem in the vocal tract or in the brain? **It is mainly of the vocal tract, since the its muscles don't react as quickly as nerve pulses. Note that we perceive echos as close as 20 milliseconds, and that the visual system's flicker perception threshold is about 20 pictures per second.** What about the speech hearing system – does it have this problem? **We can perceive sounds of very short duration and phase reversals (remember the phase jump in the fax negotiation?). So, our hearing system detects very short signals.** Give arguments for and against communicating using the visual system instead of the auditory system. **Although we usually use *body language* to supplement speech communications (i.e., to carry low information rate signals), and lip motion to help against interference and noise, it would be possible to use it much more effectively. Most people can read much faster than they can listen to a person speaking (average 250 word per minute reading is twice the normal speech rate). Compressed video requires over a million bits per second!**