Signals and systems class, HSE, Spring 2015. A. Ossadtchi, Ph.D.

Lecture 1

Definition of signal

Signal is usually a measurable quantity reflecting the state of a system or a medium and tracking this state over some extended range of the carrier variable (time, space, etc....) . Signals usually carry information about a system or a medium. The information can be naturally coded (sound pressure variation) or artificially coded (Morse code) into a signal. Certain manipulations and transformations are performed on signals in order to extract this (naturally or artificially) encoded information from the signal.

Digital versus analog signals

Signals are defined on its carrier. The notion of carrier is similar to the notion of *support variable* for functions. Say, *s(t)* is a function *s* of *t* defined as dependence of *s* on variable *t*, called support (or independent) variable Usually, we will consider the carrier to be time and the signals then reflect the evolution of some physical process over time. However, the role of carrier variable can be performed by coordinates in space (x,y,z) or by angular coordinate and etc....

In most natural life cases the carrier is thought of as a continuous quantity. If t is a continuous quantity then for any two arbitrarily close values of t, say t_1 and t_3, we will be able to find another value t_2 that is situated between those t_1 and t_3 values. The same can be said about the amplitude of such signals. Often, such signals having continuously varied amplitude and defined over the continuous carrier are called *analog signals* and the corresponding processing is referred to as *analog processing*. Strictly speaking, analog signals should take on continuous (not discrete) values. Therefore, a signal taking values only from the set {0,1} can not be called analog signal even if it defined on a continuous carrier. However, in the modern world most of the signal processing is done using the digital computers. To be able to process the analog signals they need to be input into the computer. This is done via analog to digital conversion process by devices called Analog-to-Digital Converters (ADCs). This operation performs discretization of signal amplitude and the carrier. Discretization of the carrier is called sampling and corresponds to reading off the signals values at the discrete moments of time (collecting samples).Usually, these time moments are regularly spaced so that $t_{i+1}-t_i=\frac{1}{F_s}$ where F_s is called sampling frequency or sampling rate. If the samples are taken every second then the sampling rate is 1 Hz, if the samples are taken every millisecond then $F_s=1000$ Hz . Discretization of signals amplitude is called quantization that allows the signals to be represented in the computers using binary code. Modern ADC converters use 24 bits to represent the amplitude of the input signal. This means that signal peak-to-peak amplitude is said to belong to one of 2^{24} levels, see the figure below. For most signals such quantization yields sufficiently fine grain and therefore the effects of signal amplitude quantization are significantly less of a concern than quantization of a carrier variable (sampling). In what follows we will ignore signal amplitude quantization effects, although the concerns regarding the quantization arise when dealing with high performance systems and result into the problems of stability of such systems.

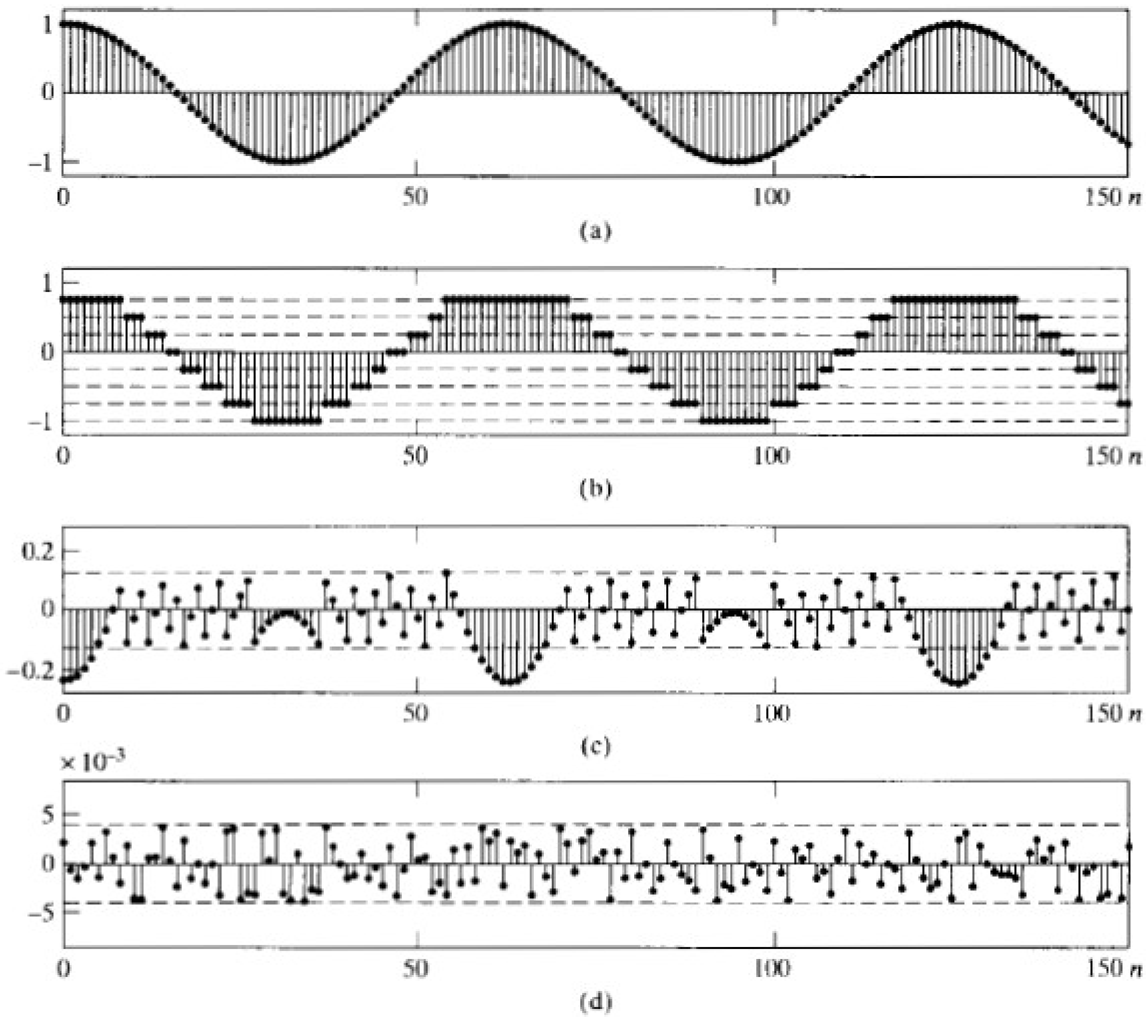

Figure 4.51 Example of quantization noise. (a) Unquantized samples of the signal $x[n] = 0.99\cos(n/10)$. (b) Quantized samples of the cosine waveform in part (a) with a 3-bit quantizer. (c) Quantization error sequence for 3-bit quantization of the signal in (a). (d) Quantization error sequence for 8-bit quantization of the signal in (a).

<u>Basic discrete signals (sequences)</u>

Discrete signals are often called sequences. The carrier for the discrete signal is a set $\mathbb{Z}$ of all integer numbers {...-100,....-1, 0, 1, …., 10, 100,....}. In order to develop our intuition let's think of a carrier for discrete signals as of discrete moments of time when the samples of such a signal have been taken. We will denote the carrier variable as *n* and to illustrate that we are dealing with discretized signals (or simply, sequences) we will use *s*[*n*] notation for the discrete signal or sequence.

So, the simplest non-trivial (not containing all zeros) sequence is so called pulse sequence defined to have zero values for all *n* but for $n = 0$ and $\delta[0]=1$. Find this pulse signal in the figure below

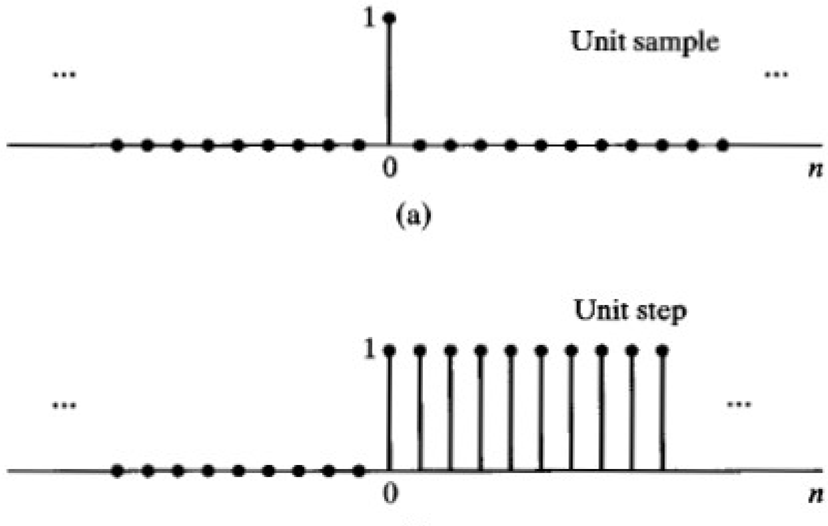

Figure 1: Impulse sequence (a) and step sequence (b). Copied from Oppenheim and Schafer's Discrete-time signal processing text, 1998

The next simplest sequence is so called unit step sequence efined to be $u[n] = 0$ for all $n<0$ and to be $u[n] = 1$ for all $n \geq 0$.

Exercise 1.1
Draw the following sequences $\delta[n-5]$, $\delta[n+5]$, $u[n-1]$, $u[n]-u[n-1]$, $\sum_{k=-\infty}^{n} \delta[k]$ for $n \in [-3,9]$

Any sequence can be written as a scaled summation of time-shifted impulse sequences. To see this, accomplish the following Exercise 1.2.

Exercise 1.2
Draw the following sequence $-1\,\delta[n+1]+2\,\delta[n]+3\,\delta[n-1]-3\delta[n-3]$ for $n \in [-3,9]$. Write a general expression for a sequence whose values are zero for all n<0 and for $n \geq 0$ are given by coefficients a_n . You can combine u[*n*] and $\delta[n]$ sequences to accomplish this.

The impulse sequence $\delta[n]$ will play the pivotal role in the discrete linear systems theory that we are going to cover later in this class, so get used to it and make it your good simple friend! How? Play with it! Also, don't forget about the unit step sequence.

Exercise 1.3
Draw the following sequence $-u[n+1]+2\mathrm{n}\,u[n]+3\mathrm{u}[n-1]-3\mathrm{u}[n-3]$ for $n \in [-3,9]$

Another important sequence is the real exponential sequence defined as $x[n]=A\alpha^n$ with base α being a real number, i.e. $\alpha \in \mathbb{R}$. Example of a real exponential sequence is shown in figure below. Is the base for this sequence greater or less than unity?

In case, the base α is a complex number, i.e. $\alpha=|\alpha|e^{j\omega_0}$ and $A=|A|e^{j\phi_0}$ then by simple

substitution and using Euler's formula we get
$x[n]=A\alpha^n=|A||\alpha|^n\cos(\omega_0 n+\phi_0)+j|A||\alpha|^n\sin(\omega_0 n+\phi_0)$

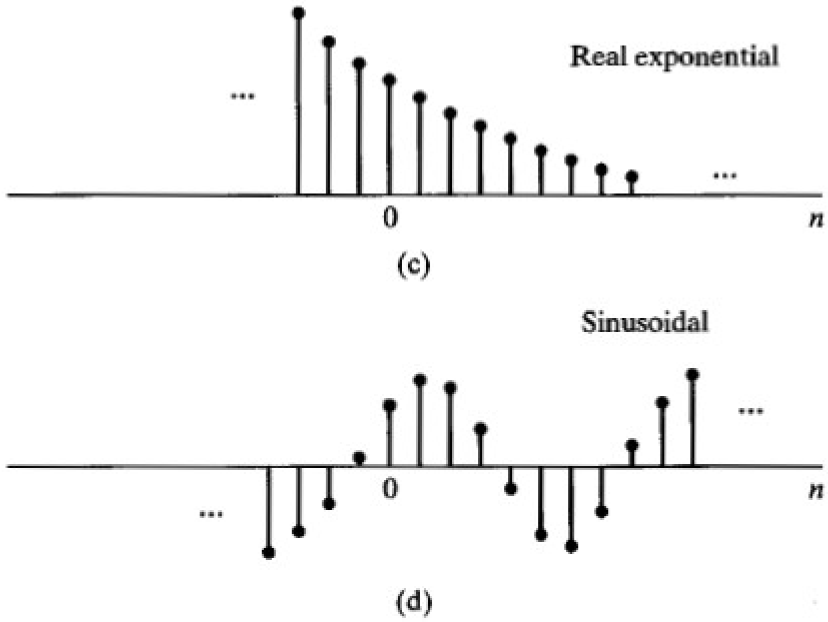

Figure 2: Copied from Oppenheim and Schafer's Discrete-time signal processing text, 1998

Exercise 1.4

Consider sequence $x[n]=|A||\alpha|^n\cos(\omega_0 n+\phi_0)+j|A||\alpha|^n\sin(\omega_0 n+\phi_0)$ for A = 1.0 and $\alpha=0.9e^{j0.2\pi}$ .What is ϕ_0 ? What is ω_0 ? Draw the real part of x[n] for *n* = 0,...10. Do the same for $\alpha=2e^{j0.2\pi}$. What is the role of $|\alpha|$?

For $|\alpha|=1.0$, we get $x[n]=|A|e^{j(\omega_0 n+\phi_0)}=|A|\cos(\omega_0 n+\phi_0)+j|A|\sin(\omega_0 n+\phi_0)$ and this sequence is called complex exponential sequence. The real part of the complex exponential sequence reduces to the cosinusoidal sequence
$x[n]=\Re\{|A|\cos(\omega_0 n+\phi_0)+j|A|\sin(\omega_0 n+\phi_0)\}=|A|\cos(\omega_0 n+\phi_0)$ and the imaginary part is simply a sinusoidal sequence.

We will make an extensive use of this complex exponential sequence when we study linear systems and the fundamentals of transform theory and the discrete time Fourier transform in particular. The use of complex counterparts of the real-life real signals allows to represent the operation of shift in time (or any other carrier variable) as a simple multiplication by a complex number. This allows to efficiently solve the problems of finding the shifts that, say, maximize the power of the sum of the shifted signals. This problem often arises in a wide range of signal processing applications. So, again, make the complex exponential sequence your friend! How? Treat it with a simple script in Matlab that generates the values of such a sequence for given A , ω_0 and ϕ_0 and plots its real and imaginary parts. Use *stem* command of Matlab to plot and define *j = sqrt(-1). Y*ou can either use the explicit formula via sine and cosine or use the *exp(j*w.*n)* expression to get the samples of this sequence.

Interesting properties of the complex exponential sequence

Observe that $x[n]=|A|e^{j(\omega_0 n+\phi_0)}=|A|e^{j((\omega_0+2\pi k)n+\phi_0)}=|A|e^{j(\omega_0 n+\phi_0)}e^{2\pi k}=|A|e^{j(\omega_0 n+\phi_0)}$. This means that the complex exponential sequence is indistinguishable for the frequency values whose difference is a multiple of 2π . So, when studying complex exponential sequence we will only need to consider the frequencies in the range of 2π , i.e. $-\pi\le\omega_0\le\pi$ or $0\le\omega_0\le 2\pi$.

What is the period of a complex exponential sequence $x[n]=|A|e^{j(\omega_0 n+\phi_0)}$? For the periodicity to happen we should have $x[n]=x[n+N]$ with *N* be strictly integer. Let us substitute this pair of arguments for the expression of the complex exponential sequence $x[n]=|A|e^{j(\omega_0 n+\phi_0)}=|A|e^{j(\omega_0 n+\omega_0 N+\phi_0)}$. Based on the observation stated in the previous paragraph we conclude that for the last equation to be true we should have $\omega_0 N=2\pi k$. So, we should find such smallest value of *k* that $N=\frac{2\pi k}{\omega_0}$ **<u>is an integer</u>** and only then call it period of *x*[*n*].

In some cases, however, for the discrete cosine and sine sequences we will not be able to find the period at all. Consider $x[n]=\cos(n)$ with $\omega_0=1$. Then, for the periodicity we should have $N=2\pi k$ and *N* should be an integer. Clearly, since π is not a rational number (can not be represented as the ratio of natural numbers) it is impossible to find such a *k* that would provide shift *N* to be an integer. Therefore, the sequence $x[n]=\cos(n)$ is not periodic!

As we showed above the cosine sequences with frequencies ω_0 and $\omega_0+2\pi r$ for integer *r* are indistinguishable and sine sequences are in antiphase. Consider a set of *N* sequences with frequencies $\omega_k=\frac{2\pi}{N}k$. Clearly, the sequence with frequency $\omega_0=\frac{2\pi}{N}0=0$ and sequence with $\omega_N=\frac{2\pi}{N}N=2\pi$ are identical as both cosine and sine values are the same for each *n*-th sample. $x_0[n]=\cos(0\mathrm{n})=1,\ x_N[n]=\cos(2\pi n)=1$ and $y_0[n]=\sin(0\mathrm{n})=0,\ y_N[n]=\sin(2\pi n)=0$. This sequence is a constant sequence and does not oscillate at all. Let's continue going down the spectrum of frequency values The next frequency is $\omega_1=\frac{2\pi}{N}1$ and fantastically, the cosine and sine sequences with this frequency oscillate as fast as the sequences with the corresponding frequency but on the side of the spectrum, i.e. the sequence with frequency value $\omega_{N-1}=\frac{2\pi}{N}(N-1)=2\pi-\frac{2\pi}{N}$, as

$x_{N-1}[n]=\cos(2\pi n-\frac{2\pi}{N}n)=\cos(\frac{2\pi}{N}n)$ (because *cos(x)* is a 2π periodic even function) and

$y_{N-1}[n]=\sin(2\pi n-\frac{2\pi}{N}n)=-\sin(\frac{2\pi}{N}n)$ (because *sin(x)* is a 2π periodic odd function) .

Therefore, the cosine sequences coincide exactly while the sine sequences are in antiphase. This property is used in designing efficient algorithms for Fourier analysis that we will address later in the course. Also, use this property in one of your homework problems.

Based on the above we should be careful when assigning physical meaning to high and low frequencies of the discrete time sequences. The oscillation frequency of discrete harmonic sequence with $\omega_k=\frac{2\pi}{N}k$ grows up to $\omega_{\lfloor N/2\rfloor}$ and then starts to slow down to become a flat sequence again for ω_N as it is shown in the figure below.

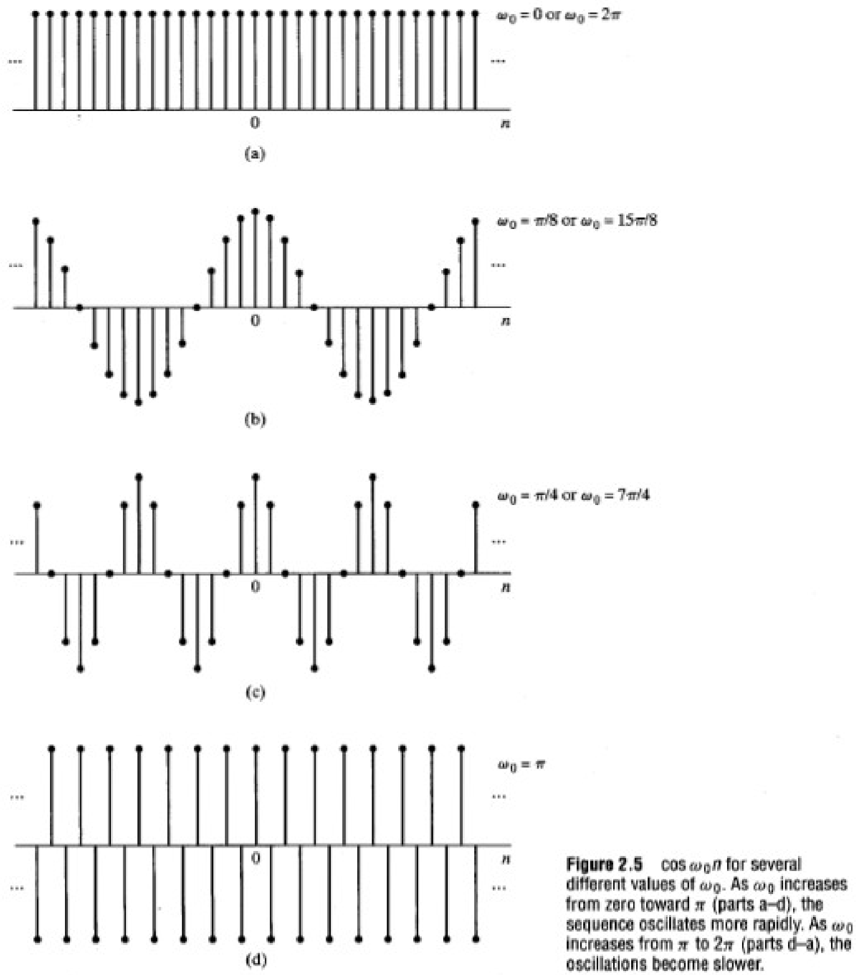

Figure 3: Copied from Oppenheim and Schafer's Discrete-time signal processing text, 1998

<u>Sequences as points in *N*-dimensional space</u>

A sequence can be thought of as a vector of dimension *N*, with *N* being the length of the sequence. Since a sequence is a vector it corresponds to a point in the *N*-dimensional space. Thinking of a sequence as of a vector allows to introduce the notions of collinear sequences, orthogonal sequences, projections of one sequence onto another. We can also think of the angle between the pair of sequences (or cosine of it) as the measure of similarity of the sequence shapes. This can be easily computed using the scalar product of normalized sequence. Indeed, if $\boldsymbol{x}^T \boldsymbol{y} = |\boldsymbol{x}||\boldsymbol{y}|\cos(\sphericalangle[x, y])$ then $\cos(\sphericalangle[x, y]) = \frac{\boldsymbol{x}^T \boldsymbol{y}}{|\boldsymbol{x}||\boldsymbol{y}|} = \tilde{\boldsymbol{x}}^T \tilde{\boldsymbol{y}}$, where the tilde denotes corresponding unit-norm vectors. The norm of the sequence to be used in this expression can be computed as $|\boldsymbol{x}| = \sqrt{\boldsymbol{x}^T \boldsymbol{x}}$ and corresponds simply to the length of the vector representing this sequence in the *N*-dimensional space.

The distance metric in this space can be purely Euclidean, or may depend on some properties of the sequences and scale the space accordingly. We will return to the question of representing sequences as vectors in the high-dimensional hyperspace after a brief introduction to random processes.

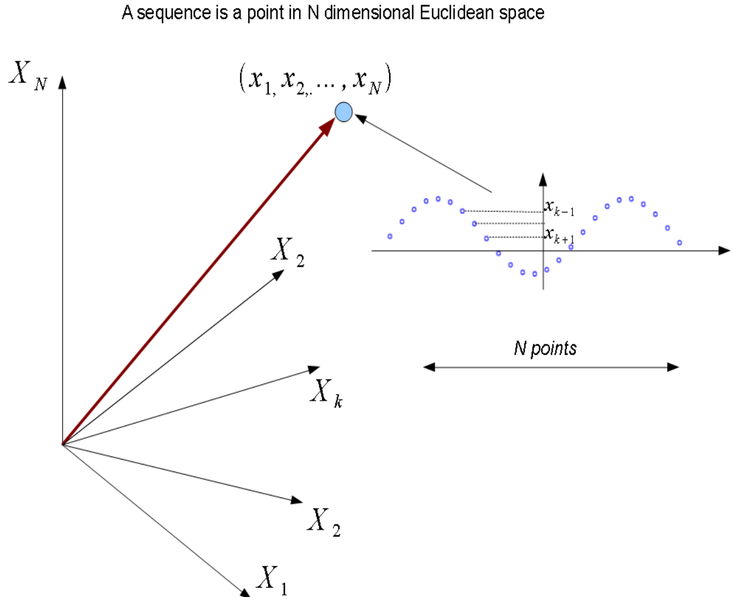

Figure 4: Any sequence x with N samples can be thought of as an N-dimensional vector. Then, a sequence corresponds to a single point in the N-dimensional space.

Exercise 1.5

Put together a simple Matlab program that generates a set of sequences with $x_k[n] = \cos(\frac{2\pi k}{N} n)$ for $k = 0{:}N$, and for n = 0:N-1 with N = 128. Store sequence values in the rows of matrix **X**. Then do this *figure, subplot(1,2,1), imagesc(X), subplot(1,2,2), imagesc(X*X').* The first figure shows colorcoded values of the rows of **X**. You can see how the oscillation frequency grows and the slows down. Now, think of rows of **X** as of vectors, explain what you see on the second figure. Below is the picture you should get and explain.

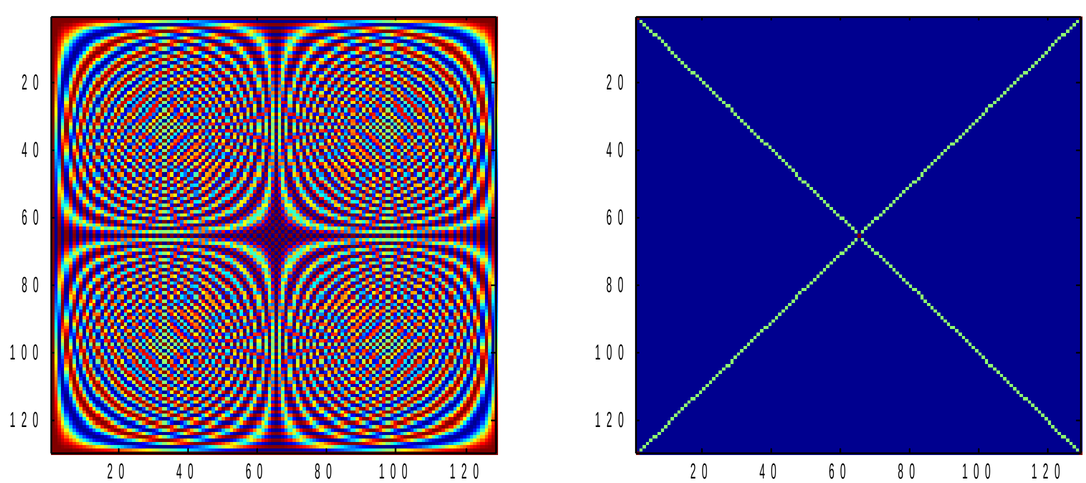

Figure 5: The result you should get if you properly accomplish exercise 1.5. Explain the second figure that plots color-coded matrix XX^T