0.1 CDMA

This course explains basic principles of CDMA (Code Division Multiple Access) - a method of sending digital signals of several users over a single digital channel.

Disclaimers: The course is *under construction* and still unfinished. Unlike the other courses in Algovision, this one covers the topic only partially. It describes just several aspects of the synchronous CDMA and doesn't deal with other methods that belong to the family of CDMA protocols, e.g., W-CDMA or DS-CDMA.

The main goal of the course is to develop geometric intuition that is behind CDMA. Since our space, in which our visualization is implemented, has just 3 dimensions, I have to restrict the geometric scenes to the cases of 2 and 3 users, but I am sure that it would be easy for a student to generalize the geometric view of CDMA to any higher number of users and dimensions.

Scene: *Signatures*

The first two scenes admit generally N users communicating simultaneously over a single digital communication channel. The number N can be changed using the field [**# of users**] and it must remain in the range $2 \leq N \leq 8$ to avoid plots that are too narrow. The original value of N is 3.

Each of our N users has a *signature*, which is a sequence of L real numbers $s_0, \ldots, s_{L-1}$. A signature represents a periodic signal $\sigma = \sigma(t)$: the periods of the signature signal are drawn in repeating white and pink intervals, which are divided into L equal subintervals, and the function in the i-th subinterval ($i = 0, \ldots, L-1$) is constant and has the value s_i. For reasons that will be clear later, L should be greater or equal to N.

Signatures can be changed by dragging any constant segment of the function up or down. Since a signature is a periodic function, changing the signature in one period interval induces changes in other periods of the same signature as well. There are two modes of drawing signatures - use the checkbox [**Signature outline**]. You will see later how signatures are used.

Due to constraints imposed by the display graphics we require that $L \leq 16$ and that values $s_0, \ldots, s_{L-1}$ remain in the light domain where a signature of the corresponding user is drawn (i.e., there is an upper bound to the size of s_i's). The number L can be changed using the field [**SGN width**].

The number P of periods of signatures that are shown in the display can be changed using the field [**Frequency**]. Algovision prevents setting frequency that is too high ($LP > 256$) or smaller than 1.

Scene: *Signals*

This scene shows not only signatures of all users, but also digital signals they transmit. Any user sends one bit of information during each period of the signature. Instead of using values 0 and 1, values of the user's signal are -1 for *false* and 1 for true. This means that during each period of the signature function the user's signal is constant and equal to either -1 or 1 and the values in different period intervals are not correlated in any way.

The signal that is transmitted through the channel and combines digital signals of all users is obtained by summing up for all users the product of the signature and the signal of the corresponding user. The combined signal is shown in the bottom of the screen under the title "Combined signal". The following scenes will explain how to decode the combined signal both assuming a perfect communication channel (which is easy) and a noisy channel (which is much more challenging).

Again, the signatures can be changes by the mouse, and users' signals can be changed in a similar way. In fact, the scene is more general than it was explained above; a signal of a user can have any value between -1 and 1 (including the extreme values). This is convenient in different situations that occur in practice, e.g., each user signal can get stronger and weaker.

The combined signal is determined by the users' signatures and signals and hence it cannot be changed directly.

Note that CDMA generalizes time multiplexing. If the signature of the i-th user is such that $s_i = 1$ and $s_0 = \cdots = s_{i-1} = s_{i+1} = \cdots = s_{L-1} = 0$, then the values of signals of the users repeat in a round-robin way, during each period each user sends exactly one bit. Arrange and observe this situation.

Let me repeat that the present course explains a *synchronous* CDMA, and hence all signatures and user signals are perfectly synchronized. An asynchronous version is beyond the scope of this course and will be described in a course that is now under construction.

Scene: *Signatures in 2D*

Now we arrive to geometric view of CDMA, which is very convenient in many situations. This scene is restricted to the case of two users and signatures of the width 2 items. The restriction allows us to show signatures as vectors in the plane, which admits very easy visualization.

A signature is mathematically an ordered pair of real numbers s_0 and s_1, i.e. a two dimensional vector. A vector can be displayed in a standard way in the plane using an arrow that starts in the origin of the coordinate system.

The signature of the first user is shown as a red vector, the signature of the second user as a green vector (the colors correspond to signature colors in column representation ([**Signature outline**] checked off).

Change the signatures by the mouse as explained in the previous scenes, and observe how the corresponding vectors change. The angle between the red and green vectors is very important for decoding, as we will see later. Note that originally the vectors were orthogonal, and this is the most convenient angle.

Change the first signature arbitrarily and then change the second signature in such a way that the vectors are orthogonal.

Scene: *Signals in 2D*

The scene shows again one period of 2 synchronous signatures of two users. Signatures of 2 users are shown in the same way as in the previous scene (both as a signal and as a vector). Moreover, signals of the users are shown and the combined signal is presented, again as a plot and a vector. We use the black color to draw the vector corresponding the combined signal.

Scene: *Samples in 2D*

The scene shows a geometric visualization of the result of sending signals of two users (1000 bits each), combined using a synchronous CDMA, through a communication channel with Gaussian noise.

At the beginning (Noise = 0), each point represents a combined signal corresponding to a pair of bits sent by the users. Since we have just four possible pairs of users' bits ([1,1], [1,-1], [-1,1], and [-1,-1]), just four points are visible (each corresponding to about 250 signature period intervals).

If the checkbox [**User 0**] is checked, black (red, resp.) points correspond to the bit value 1 (-1, resp.) of the user 0, similarly if the checkbox [**User 1**] is checked, the point color reflects the value of the user 1.

Click a black or red point in the geometric window using the right mouse button. The corresponding values of signals of users and the combined signal appear in the left part of the screen. If the checkbox [**Show vectors**] is checked, the geometric window shows also vectors corresponding to the signatures (red and green) and, if a black or red point is selected, the vector corresponding to the combined signal is shown as well.

Change the signatures and observe how the signals change. Given the same signature vector lengths, the smallest distance between points corresponding to different user signals is maximized when the signature vectors are orthogonal. Note that the smallest distance implies how easy or difficult is to distinguish user bits, when the combined signal is given.

Now, increase the noise either by writing a small integer into the field [**Noise**], or by clicking the button [**Low**]. The combined signal is superposed with Gaussian noise of low amplitude (i.e., values of two independent real-valued random variables are added to the two components of the combined signal; the random variables have Gaussian distribution with the mean 0 and a relatively low variance). Now, the points that correspond to the same pair of users' bits do not map any more to the same point of the plane, but form a relatively compact cluster. Use checkboxes [**User 0**] and [**User 1**] to see the original values of the users' bits. You can see that clusters are well separated and decoding of the combined signal is still quite easy.

Finally, click the button [**High**]. The combined signal is superposed with Gaussian noise of high amplitude. You will see that the clusters overlap and it becomes difficult or impossible to distinguish between red and black points just using information about their position, which prevents an errorless decoding of the signal received from the noisy channel.

Observe the situation for different values of the noise amplitude.

Scene: *Decoding in 2D*

The scene is the same as the preceding one, but there is a blue line passing through the origin that divides the plane into two halfplanes. The line can be rotated around the origin using the mouse with the left button pressed.

The goal is, for a given selection of a user (checkboxes [**User 0**] and [**User 1**]), positioning the blue line in such a way that the red points (corresponding to one value of bits sent by the selected user) appear in one halfplane, and the black points into the second halfplane.

The goal can be achieved easily for low amplitudes of the noise (provided the signature vectors are in an appropriate relationship, e.g. orthogonal). If the noise amplitude is high, certain number of errors is inevitable. Try to find the best position of the blue line (smallest number of errors).

Now, check [**Show normal**]. A yellow arrow appears that starts in the origin and is perpendicular to the blue separation line. Click an arbitrary sample by the right button of the mouse. The membership of the sample into one of the halfplanes determined by the blue separation line can be determined by observing whether the angle between the yellow arrow and the black arrow pointing to the sample is greater or smaller than 90 degrees. This in turn can be determined by the sign of the scalar product of the yellow vector and the black vector.

Note that it the yellow vector is collinear with the signature vector of certain user, the separation line gives usually very good separation of samples that correspond to bits 1 or -1 of the user. However, you can see that, especially under noise, the optimal yellow decoding vector that yields the smallest number of errors is slightly different from the signature vector of the user. It is beyond the scope of this course to analyze how the optimal decoding vector could be determined. [One method to minimize errors uses the smallest squares; we plan to add optional calculator displaying the sum of squares as a guide for a learner.]

Thus, decoding the signal of one user from the combined signal of CDMA received in a receiver can be done easily by the sign of the scalar product of the signal received during one signature period (and considered as a vector) with certain fixed vector, which is equal or close to the signature vector of the user. Since computing a scalar product is a simple and fast task, decoding CDMA signal doesn't represent a serious problem, once the decoding vector is known.

Scene: *Signatures in 3D*

Now, we repeat the previous 4 scenes for 3 users, and the geometrical visualization uses three dimensions. Three-dimensional vectors representing signatures of users are shown as red, green, and blue vectors in 3D. Use the mouse (left button) to rotate the 3D drawing.

Even though the vectors seem to be in a general position, you would find that they are mutually orthogonal when the scene appears. Change signatures in a way described above and observe how the corresponding vectors change.

Scene: *Signals in 3D*

Not only signatures of 3 users are displayed, but also users' signals (one bit for each user). The combined signal is both displayed in the left lower corner and represented as a 3D vector in the right geometric window. Change signatures and signals and observe how the corresponding vectors change.

Scene: *Samples in 3D*

The scene shows 1000 samples of the combined signal for different values of users' signals selected from $\{-1, 1\}$. When the scene appears, no noise is added, but you can involve noise in the same way as in the 2D scenes. One can again select one of the users and get samples colored as red and black according to the sign of the signal of the selected user, display values corresponding to a sample selected by a right mouse button click, display signature vectors and/or sample vector, rotate the image in the geometric window, and change the level of noise.

Observe that samples form 8 well-localized clusters that can easily be separated, provided the noise level is low. However, for high noise level the clusters merge together and it would be impossible to separate them by a plane.

If four- or higher dimensional space were available, we could have represented the situation with 4 or more users, and obtain 2^N clusters, where N is the number of users.

Scene: *Decoding in 3D*

This scene has all functionality of the previous one and asks you to separate red and black samples (corresponding to a particular choice of a user) by a blue plane that passes through the origin. While rotating of the whole image is by the mouse with the left button pressed, if the right button is pressed, only the plane rotates. [At the moment, the way of moving the plane is not well designed, and it will be amended soon].

The blue plane separates the space into two halfspaces. The membership to one of the halfspaces can again be determined by the sign of the scalar product of the vector pointing the selected sample and the normal vector of the plane. Check [**Show normal**] to show the normal to the plane.

To make the image clearer, the two sides of the plane have slightly different colors, and two normals (again slightly different yellow colors) are displayed on either sides of the plane.