EE521 Analog and Digital Communications

Questions

Problem 1: SystemView

Part A (25%)

Name the three fundamental types of SystemView data formats.

Response

- Voltage, real number, floating point
- Integer, signed integer, unsigned integer, token
- Digital, binary

Part B (25%)

Describe each of the three fundamental SystemView data formats, giving

- Numerical range
- Accuracy

If more than one numerical range is possible for a data format, give all of them.

Response

Internally, all SystemView data formats are carried as floating point. This simplifies interface between tokens and allows simplifications in models, such as we use when using a binary signal to model the output of a matched filter for one bit in a data stream.

Voltage

This is a floating point format.

Integer

The base format is signed 32-bit integer. Symbols must be unsigned integers.

Digital

There are two digital formats. For digital systems, values of 0 and 1 are used. For modeling digital outputs of analog systems with antipodal waveforms, and for input to modulators, -1 and +1 are sometimes used. The threshold for input of digital systems is 0.5 when 0 and 1 are expected, and is zero when -1 and +1 are expected.

Part C (25%)

Describe the importance of the SystemView Time Specification (sample rate, simulation time, total number of samples) to these considerations:

- The highest data rate in your simulation.
- Analog signals such as sine waves that have not been sampled.

Response

The SystemView Time Specification allows the user to set the SystemView sample rate and either of two interacting specifications, the number of samples or the run time. The number of samples can be set to a power of two with a spin button, which will double or half the number of samples in a run or loop. Other inputs are the number of loops or iterations, and whether or not the model is cleared between loops.

Part D (25%)

Which of these tokens have a different data rate on their output than their input? What is the output data rate as a function of the input data rate?

- The quantizer token.
- The symbol-to-bits converter token (for 8 bits per symbol)
- A forward error correction (FEC) encoder with a code rate of r
- The bits-to-symbol converter token (for 8 bits per symbol)
- A FEC decoder with a code rate of r
- The sampling token.

Response

There were two reasonable interpretations of "data rate" and both were accepted for full credit.

For data rate interpreted as sample rate:

- The quantizer doesn't change the sample rate
- The sample rate out of the symbol-to-bits converter is 8 times the input sample rate (for 8 bits per symbol).
- The sample rate out of a FEC is $1/r$ times the input, and increases.
- The bits-to-symbol converter has 1/8 the sample rate on its output as it does on its input (for 8 bits per symbol).
- The output bit rate of FEC decoders is r times the input bit rate.
- The sampling token changes the sample rate from the SystemView sample rate, which is used to model time-continuous analog signals, to a sample rate specified as a setup parameter. The sample rate must be less than or equal to the SystemView sample rate for the token to function, and the model will not execute unless this condition is met.

For data rate interpreted as effective signal information rate in bits per second:

- The quantizer token reduces the number of bits per sample to a value in the token's setup.

- The symbol-to-bits converter token (for 8 bits per symbol) increases the sample rate but does not change the data rate.
- A forward error correction (FEC) encoder with a code rate of r increases the data rate by a factor if $1/r$. Since the input and output are both bit streams, the sample rate is increased by the same factor. The data rate in bits per second changes by the same factor. The extra information is used in error detection and correction.
- The bits-to-symbol converter token (for 8 bits per symbol) inverts the actions of the symbol-to-bits converter. The information rate does not change.
- A FEC decoder with a code rate of r inverts the actions of the FEC encoder. The information rate is reduced by a factor of r.
- The sampling token does not change the information rate if the signal is sampled at or above the Nyquist limit.

Problem 2

Part A (20%)

For a linear block code with the generator matrix *G*:

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

The codewords are

$$UM = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (2.2)$$

- What is the codeword length *n*? The data word length *k*? What is the code rate?
- Is this a systematic code? Why?
- Verify the codeword corresponding to message (1,1,1). Show your work.

Response

This is a (10,3) linear block code. The code rate is *k*/*n* or 3/10.

This is a systematic code. The right three columns form a 3 by 3 identity matrix. This is the format used in the text; see Sklar's equation (6.27) page 333.

The codeword for the message (1,1,1) can be found by adding the columns of the generator matrix using binary arithmetic. This can be done by inspection from (2.1): (0,0,0,1,0,0,1,1,1,1). Note that this is the last row in the *UM* matrix, and could have been taken from that matrix.

Part B (50%)

- What is $d_{\min}$ for this code?
- What is the maximum number of unambiguously correctable bits *t*?

Response

The $d_{\min}$ for this code is found from the smallest Hamming weight of the codewords, exclusive of the zero codeword, as 5. The maximum number of unambiguously correctable bits *t* is [(5-1)/2] or 2. Note that there is a truncation to the next lowest integer in the computation of *t*; note Sklar's equation (6.44) page 345 and the line of text immediately following the equation.

Part C (30%)

For the received codeword

$$cr = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.3}$$

- Write the syndrome.
- What is the next step in automated decoding? Do we need the full standard array? Why? Pick out the correct codeword using the minimum Hamming distance between the received codeword and each member of the valid codewords.

Response

The syndrome is found by the product

$$s = cr \cdot H^T = cr \cdot \begin{bmatrix} I_7 \\ P \end{bmatrix} \tag{2.4}$$

Since the last three bits of the received codeword are zero, the syndrome is simply the first seven bits of *cr*.

Decoding is the operation of finding the valid codeword with the smallest Hamming distance to the received codeword. Decoding can proceed by using the syndrome to find error pattern for the zero codeword and adding this error pattern to the received codeword to form the corrected codeword, as described in Sklar section 6.4.8 pages 336-340. Or, for small code like the example used for this problem, the Hamming distance from the received codeword to each of the valid codewords can be examined. We see from the table of valid codewords given in (2.2) that we have a Hamming distance of 2 from the codeword for the message (0,1,1). Thus that is the message, extracted as the message corresponding to the correct codeword, or the last three bits of the corrected codeword.

Problem 3

Part A (25%)

For the cyclic code defined by the denominator polynomial

$$D(X) = 1 + X^4 \tag{3.1}$$

and the code vector

$$U^{(0)}(X) = X^2 + X^3 \tag{3.2}$$

find the other code vectors $U^{(i)}$.

Part B (25%)

Show that

$$g(X) = 1 + X \tag{3.3}$$

is a generator for the cyclic code defined by (3.1), by showing that the remainder *r(X)* is zero for the quotient

$$\frac{D(X)}{g(X)} = q(X) + \frac{r(X)}{g(X)} \tag{3.4}$$

where *r(X)* is degree zero – i.e. either 0 or 1.

Part C (25%)

Find *n*, *k* and *p* for the cyclic code defined by (3.1) and (3.3).

Part D (25%)

Find the codeword for the message vector

$$m(X) = 1 + X^2 \tag{3.5}$$

Response

This is an ordinary binary cyclic code as described in Sklar's sections 6.7.1 and 6.7.2 pages 357-359. This is simpler than the systematic cyclic codes. Some assumed that it was a systematic code because we spent much more time on that, but I meant to ask for responses for the much more simple ordinary cyclic codes (I gave full credit for systematic code results for part D). The codewords are simply left-shifted, end-around, from any valid codeword, plus the zero codeword and the all-ones codeword. From (3.1) we see that n=4 for this code, and from (3.2) we are presented (0,0,1,1) as a codeword, implying that k=3 and p=1. Thus the other codewords are (1,0,0,1), (1,1,0,0), (0,1,1,0), and the zero and all-ones codewords.

We are given the generator by (3.3), which also gives us p=1, verifying the values of k and p we have inferred. We can use the division theorem to find the remainder by subtracting powers of *X* times *g(X)* from *D(X):*

$$\begin{aligned} D_1(X) &= D(X) - X^3 \cdot g(X) \\ &= 1 + X^4 - \left(X^3 - X^4\right) \\ &= 1 + X^3 \end{aligned} \tag{3.6}$$

in binary arithmetic. Repeating this process twice more gives us

$$\begin{aligned} D_2(X) &= 1 + X^2 \\ D_3(X) &= 1 + X \\ D_4(X) &= 0 \end{aligned} \tag{3.7}$$

so that *g(X)* divided *D(X).*

We have already found *n, k,* and *p* as 4, 3 and 1 respectively. The codeword for the message given is

$$\begin{aligned} U(x) &= m(X) \cdot g(X) \\ &= \left(1 + X^2\right) \cdot \left(1 + X\right) \\ &= 1 + X + X^2 + X^3 \end{aligned} \tag{3.8}$$

For the systematic format, we have

$$U_S = R(X) + X^{n-k} \cdot m(X) \tag{3.9}$$

where *R(X)* is the remainder

$$\frac{X^{n-k} \cdot m(X)}{g(X)} = q(X) + \frac{R(X)}{g(X)} \tag{3.10}$$

or, we add a polynomial of order n-k-1 to $X^{n-i} \cdot m(X)$ to give us a valid codeword. We can find *R(X)* by adding powers of *X* times *g(X)* to $X^{n-i} \cdot m(X)$ until we have reduced the result to order *p-1*, which, for *p*=1, simply a zero or one. Or, we can look at $X^{n-i} \cdot m(X)$ and the existing codewords and see that the remainder is zero, and that $X \cdot m(X)$ is the codeword.

Problem 4

Part A (75%)

Draw ***one, not all, of the following*** for the K=3, rate 1/3 convolutional code generated by

$$\begin{aligned} g_1(X) &= X + X^2 \\ g_2(X) &= 1 + X \\ g_3(X) &= 1 + X + X^2 \end{aligned} \tag{4.1}$$

Denote the states according to Table 1 below.

Table 1 Convolutional Code State Letter Designations

State of Convolution Filter [last bit, previous bit]	State Designation
[0,0]	a
[1,0]	b
[0,1]	c
[1,1]	d

Show all work so that your logic is clear.

Option 1: The State Transition Table

This is the table presented in class and used in the demonstrations. You may use the format given in Table 2 below

Response

Table 1 is filled out.

Table 2 State Transition Table

Input Bit	State [lst bit,prv bit]	Outputs (u_1,u_2,u_3)	Next State
0	a [0,0]	(0,0,0)	a [0,0]
0	c [0,1]	(1,0,1)	a [0,0]
0	b [1,0]	(1,1,1)	c [0,1]
0	d [1,1]	(0,1,0)	c [0,1]
1	a [0,0]	(0,1,1)	b [1,0]
1	c [0,1]	(1,1,0)	b [1,0]
1	b [1,0]	(1,0,0)	d [1,1]
1	d [1,1]	(0,0,1)	d [1,1]

Option 2: State Diagram

Draw a state diagram in the format of Sklar's Figure 7.5 page 390. Name the states with letters according to Table 1 above, and use solid arrows to designate state transitions when the input bit is a 0 and dotted arrows to designate state transitions when the input bit is a 1.

Response

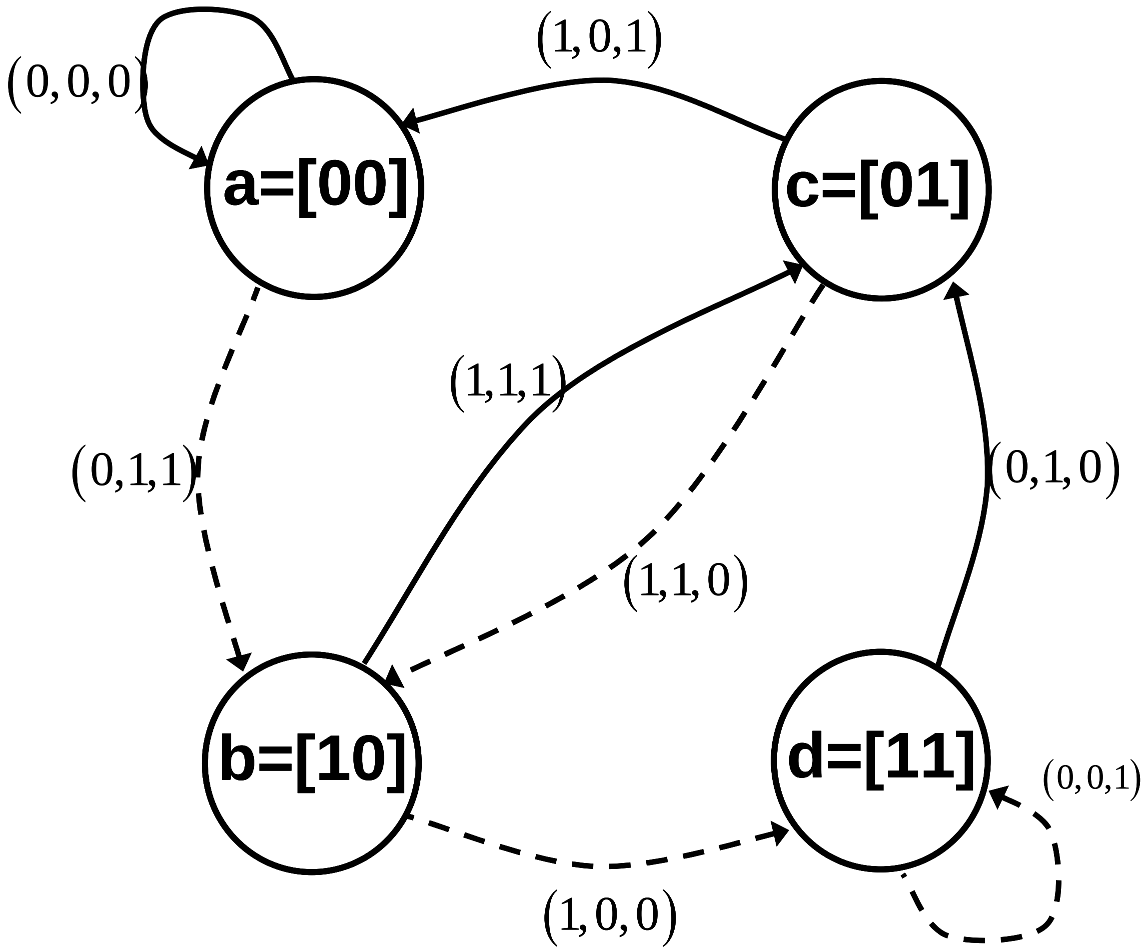
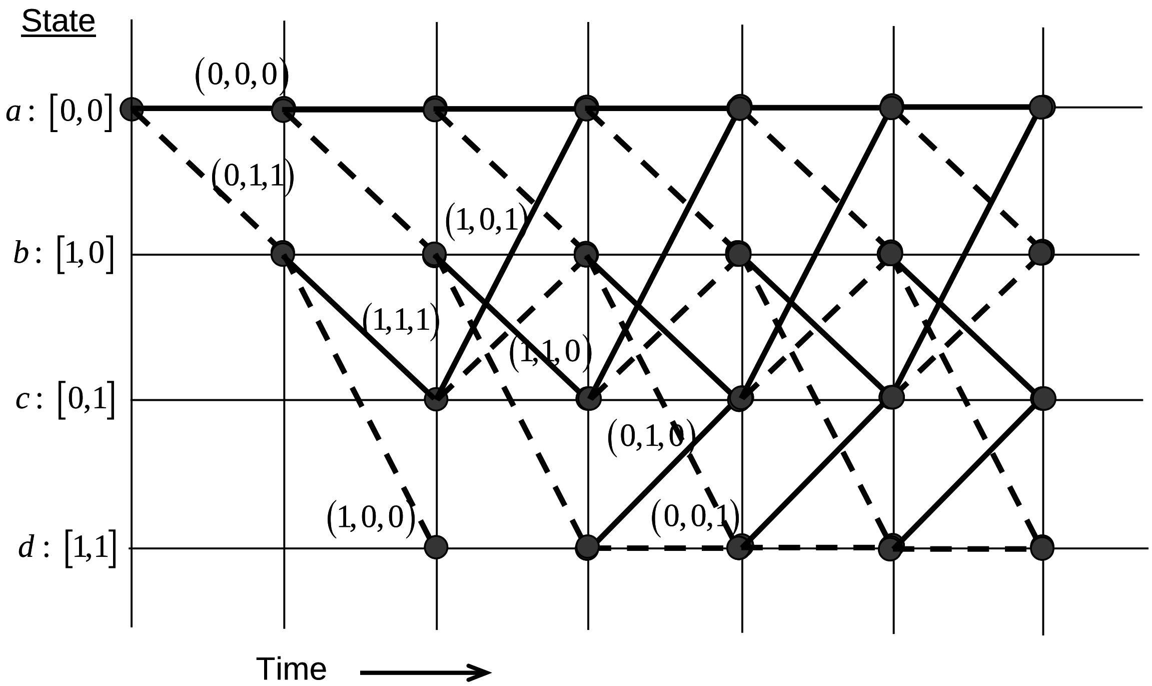


Figure 1 State Diagram

Option 3: Trellis Diagram

Draw a trellis diagram in the format of Sklar's Figure 7.7 page 394. Draw all paths, beginning with state a [0,0] for six input bits.

Response



Part B (25%)

Given an input,

$$m(X) = [1 \quad 0 \quad 1] \tag{4.2}$$

use any of results of Part A to provide the first five codeword triplets (u_1,u_2,u_3). Assume that the signal is preceded and followed by zeros.

Response

Using Table 2 and the message as given we construct the table, including output triplets, as Table 3 below.

Table 3 Convolutional Encoding

Input Bit No.	Bit	State	Triplet	Next State
1	1	a	(0,1,1)	b
2	0	b	(1,1,1)	c
3	1	c	(1,1,0)	b
4	0	b	(1,1,1)	c
5	0	c	(1,0,1)	a