Tutorial 3

12.04.2007

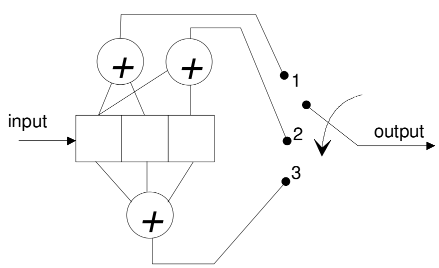
Exercise 1

Generator polynomials of a convolutional code, given in binary form, are $\mathbf{g}_1 = [1 \quad 1 \quad 0]$, $\mathbf{g}_2 = [1 \quad 0 \quad 1]$ ja $\mathbf{g}_3 = [1 \quad 1 \quad 1]$.

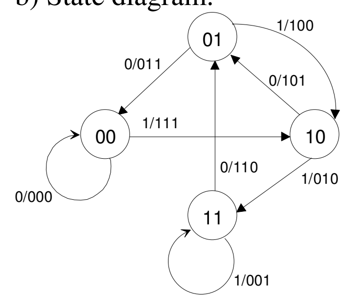
a) Sketch the encoding circuit.
b) Sketch the state diagram.
c) Find the transfer function $T(D)$.
d) What is the minimum free distance of the code?

Solution 1

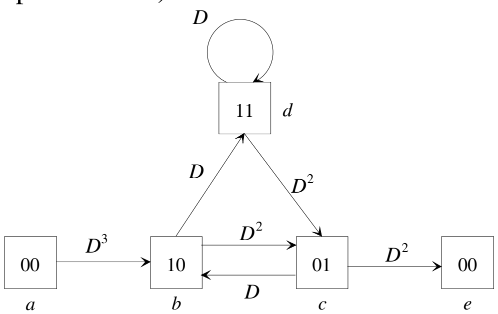
a) Encoding circuit:



b) State diagram:



c) Modified state diagram indicating the weights of transitions (node 00 split into two):



d) State equations:

$$X_b = D^3 X_a + DX_c$$
$$X_d = DX_b + DX_d$$
$$X_c = D^2 X_b + D^2 X_d$$
$$X_e = D^2 X_c .$$

From the second equation we get

$$X_d = \frac{D}{1-D} X_b ,$$

which can be inserted into the third equation:

$$
\begin{aligned}
X_c &= D^2 X_b + \frac{D^3}{1-D} X_b \\
&= D^2 \left[D^3 X_a + DX_c \right] + \frac{D^3}{1-D} \left[D^3 X_a + DX_c \right] \\
&= \frac{\left[D^2 - D^3 \right]\left[D^3 X_a + DX_c \right] + D^3 \left[D^3 X_a + DX_c \right]}{1-D} \\
&= \frac{D^5 X_a + D^3 X_c - D^6 X_a - D^4 X_c + D^6 X_a + D^4 X_c}{1-D} \\
&= \frac{D^5 X_a + D^3 X_c}{1-D} .
\end{aligned}
$$

Solving for X_c and inserting into the fourth equation results in

$$
\begin{aligned}
T(D) = \frac{X_e}{X_a} &= \frac{D^7}{1-D-D^3} \\
&= D^7 + D^7 \left(D + D^3 \right) + D^7 \left(D + D^3 \right)\left(D + D^3 \right) + ... \\
&= D^7 + D^8 + D^9 + ...
\end{aligned}
$$

From $T(D)$ we see that $d_{free} = 7$. Minimum free distance can also be seen directly from the modified state diagram of part c. The minimum-weight route $a \to e$ is $a \to b \to c \to e$. The weight of this route is $d_{free} = 7$.

Exercise 2

Calculate union bound for error probability for a Convolutional Code generated by the polynomials $\mathbf{g}_1 = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$ ja $\mathbf{g}_2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$.

Solution 2

Union bound for the error probability is evaluated as

$$p(e) \le \sum_{j=1}^{k} c_{j,d} P\left(c_i \to c_j\right) + C$$

where

C is the constant that comes from truncation of the sum (we do not sum to infinity but to some finite k and the sum from k+1 till infinity is bounded by the constant C).

$c_{j,d}$ is the amount of bit errors generated in the error path j

$P\left(c_i \to c_j\right) = Q\left(\sqrt{\frac{4Ed_{ij}^2}{2N_0}}\right)$ is the error probability of error on the error path with distance $d_{i,j}$

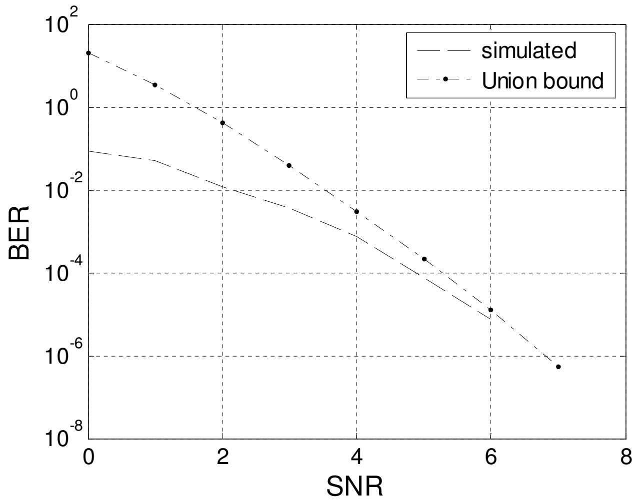
From the lecture material the code spectrum for this code is

d	5	6	7	8	9	10	11	12	13	14
A_d	1	4	12	32	80	192	448	1024	2304	5120

For this code the amount of bit error is d – 4.
We evaluate the union bound as

$$p(e) \le \sum_{j=1}^{k} c_{j,d} P\left(c_i \to c_j\right) + C$$

$$= \sum_{d=5}^{14} (d-4) \cdot A_d \cdot Q\left(\sqrt{\frac{4Ed_{ij}^2}{2N_0}}\right)$$



Exercise 3

Calculate the union bound for a Viterbi equalizer in a two tap channel.

H=[1 0.3]

Use only few first error paths.

Solution 3

Probability of error on a particular error path in Viterbi equalizer is

$$P\left\{c_i \to c_j\right\} \le 1 \cdot \left(\prod_{m=k}^{k+l-L-1} \frac{M-|\varepsilon_m|}{M} \right) \cdot Q\left(\frac{0.5\|\Delta Q\|}{\sigma} \right)$$

$$= \left(\prod_{m=k}^{k+l-L-1} \frac{M-|\varepsilon_m|}{M} \right) \cdot Q\left(\frac{0.5 d_{ij}}{\sigma} \right)$$

Where

$\frac{M-|\varepsilon_m|}{M}$ describes the probability of selecting certain path from the state in the trellis and making $|\varepsilon_m|$ symbol errors because of selecting this path. The amount of possible symbols is M. In Viterbi equalizer from each state there are M possible outgoing branches. By selecting one branch we can do $\frac{M-|\varepsilon_m|}{M}$ errors.

If the path is *l* trellis sections long we could do errors in each section. Total probability of error is multiplication of probabilities for each transition along the path. $\left(\prod_{m=k}^{k+l-L-1} \frac{M-|\varepsilon_m|}{M} \right)$. (We multiply only *l-L* transitions since last *L* transitions corresponds to correct symbols – otherwise the erroneous path would not merge with correct path).

For our BPSK transmission M=2 (two possible symbols) error $|\varepsilon_m|$ is 1.

In one errorneous path we do in averge $c_{j,d}$ information bit errors. As for the case of Convolutional Code for calculating the average bit error we have to average over bit errors over all the possible error paths.

$$p(e) \leq \sum_{j=1}^{k} c_{j,d} P\left(c_i \to c_j\right) + C$$

For simplifying our calculations we select *k* to be 3.

1. For given channel there is no error path with length 1.
2. There is one path with length 1 and with distance

$$\left\|\Delta d_{0,2}\right\|^2 = \frac{E_{rx}}{T}\left[\left(\mu_{00}-\mu_{01}\right)^2+\left(\mu_{00}-\mu_{10}\right)^2\right]$$
$$= \frac{E_{rx}}{T}\frac{1}{1.09}\left[\left(-1.3-(-0.7)\right)^2+\left(-1.3-0.7\right)^2\right]$$
$$= 4\frac{E_{rx}}{T}$$

In this path we do one error $c_{0,2} = 1/2$

Where, the term 1/1.09 normalizes the total channel power to 1.

3. There is one path with length 2 and with distance

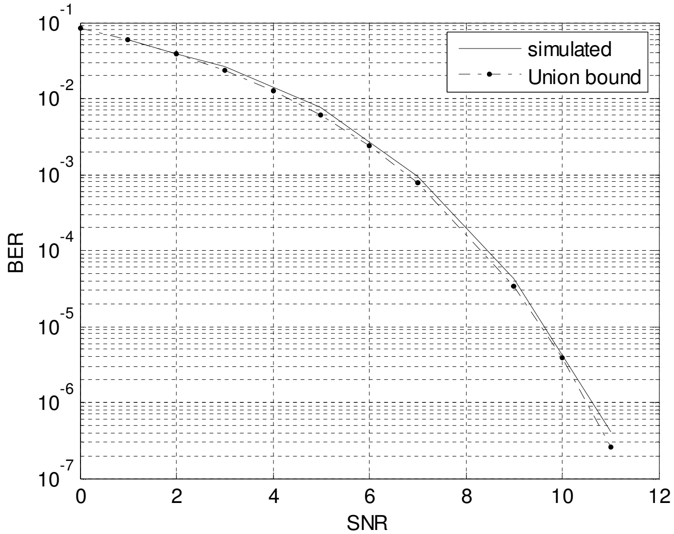
$$\left\|\Delta d_{0,3}\right\|^2 = \frac{E_{rx}}{T}\left[\left(\mu_{00}-\mu_{01}\right)^2+\left(\mu_{00}-\mu_{11}\right)^2+\left(\mu_{00}-\mu_{10}\right)^2\right]$$
$$= \frac{E_{rx}}{T}\frac{1}{1.09}\left[\left(-1.3-(-0.7)\right)^2+\left(-1.3-1.3\right)^2+\left(-1.3-0.7\right)^2\right]$$
$$= 10.20\frac{E_{rx}}{T}$$

in this path we do two errors $c_{0,3} = 2/3$

Total error probability is bounded as

$$p(e) \leq \sum_{j=1}^{k} c_{j,d} P\left(c_i \to c_j\right) + C$$
$$= h_d c_{0,2}\left(\frac{2-1}{2}\right)^2 Q\left(\sqrt{\frac{2E_b}{N_0}4}\right) + \sum_{j=3}^{\infty} h_d c_{j,d} P\left(c_i \to c_j\right)$$
$$\leq 2\cdot 1\left(\frac{1}{2}\right)^{2-1} Q\left(\sqrt{\frac{2E_b}{N_0}4}\right) + \frac{2^{N-1}}{2^N}3\cdot 2\left(\frac{2-1}{2}\right)^{3-1} Q\left(\sqrt{\frac{2E_b}{N_0}10.2}\right)$$
$$\leq 2\cdot 1\left(\frac{1}{2}\right)^{2-1} Q\left(\sqrt{\frac{2E_b}{N_0}4}\right) + \frac{1}{2}3\cdot 2\left(\frac{1}{2}\right)^{3-1} Q\left(\sqrt{\frac{2E_b}{N_0}10.2}\right)$$

Where h_d describes the amount of symbol errors for error event with length d



Exercise 4

The rate $1/2$ convolutional code has generators

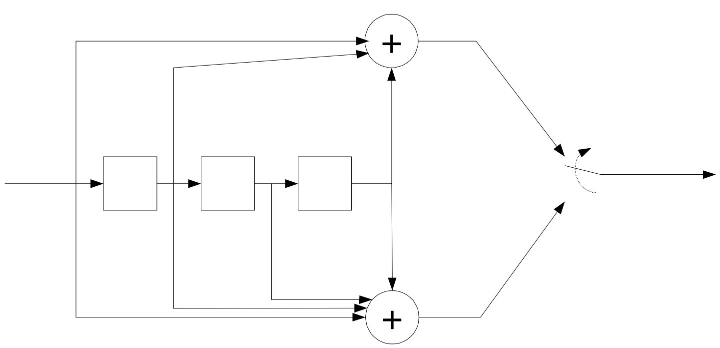
$$g^{(1)} = (1\,1\,0\,1)$$
$$g^{(2)} = (1\,1\,1\,1).$$

Represent this code as:

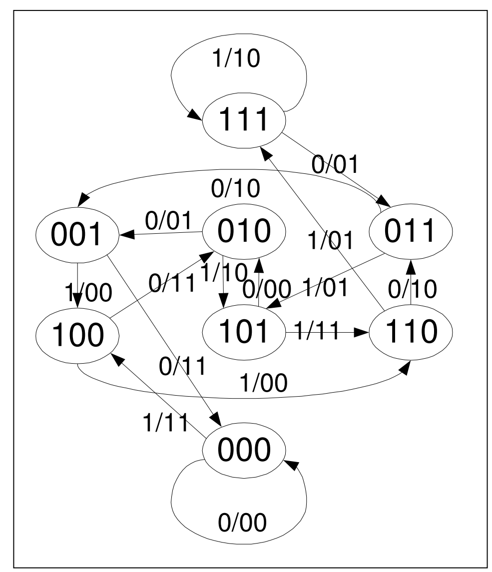
a) Shift register.
b) State diagram.
c) Tree.
d) Trellis.
e) Matrix.

Solution 4

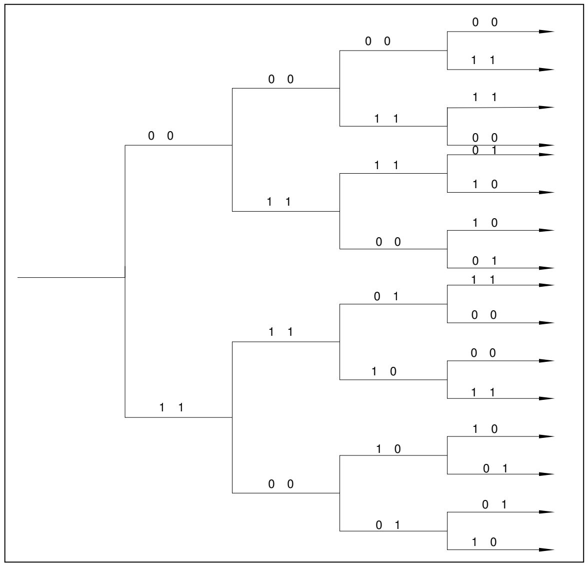
a)



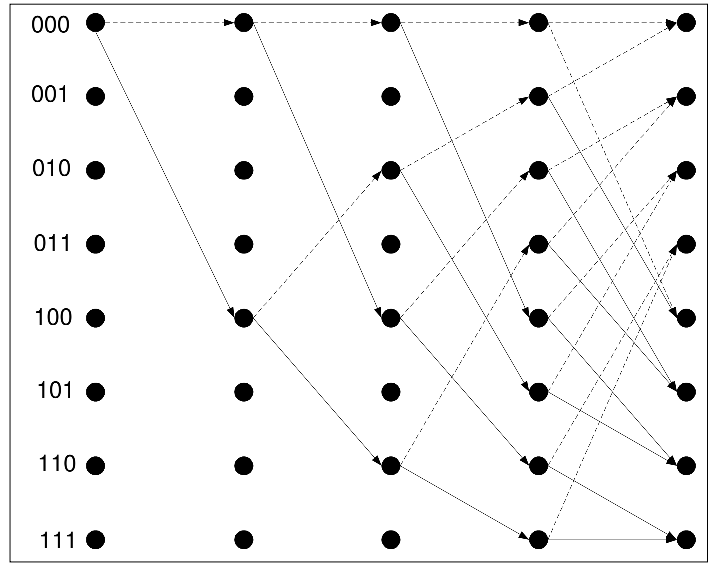
b)



c)



d)



e)

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix}$$

Exercise 5

Represent the code in the previous exercise as recursive systematic convolutional code.
Calculate the output (coded) sequence for both: systematic and corresponding nonsystematic encoder. The information sequence at the input is $u = [1\,0\,1\,0\,0]$.

Solution 5

a) recursive systematic convolutional code

$$u(z) = u_0 + u_1 z^1 + u_2 z^2 + \ldots$$

$$g^{(1)}(z) = 1 + z^2 + z^3$$

$$g^{(2)}(z) = 1 + z + z^2 + z^3$$

$$s(z) = \left[u_0 + u_1 z + u_2 z^2 + u_3 z^3 + u_4 z^4\right] \begin{bmatrix} 1 + z^2 + z^3 \\ 1 + z + z^2 + z^3 \end{bmatrix}$$

$$s(z) = \left[1 + z^2\right] \begin{bmatrix} 1 + z^2 + z^3 \\ 1 + z + z^2 + z^3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 + z^2 + z^3 + z^2 + z^4 + z^5 \\ 1 + z + z^2 + z^3 + z^2 + z^3 + z^4 + z^5 \end{bmatrix} = \begin{bmatrix} 1 + z^3 + z^4 + z^5 \\ 1 + z + z^4 + z^5 \end{bmatrix}$$

$$\left.\begin{matrix} s^{(1)} = [100111] \\ s^{(2)} = [110011] \end{matrix}\right\} s = [110100101111]$$

b) Systematic convolutional code

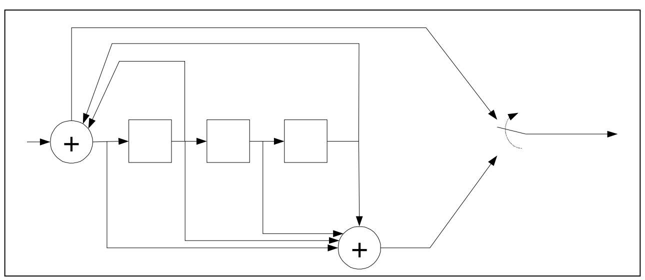
$$s^{(1)} = u(z) \cdot g^{(1)}(z)$$

$$s^{(2)} = u(z) \cdot g^{(2)}(z)$$

Divide both sequences with $g^{(1)}(z)$ and you get new sequence where one contains the systematic bit and the other is generated by a recursive digital filter.

$$s^{(1)} = u(z)$$

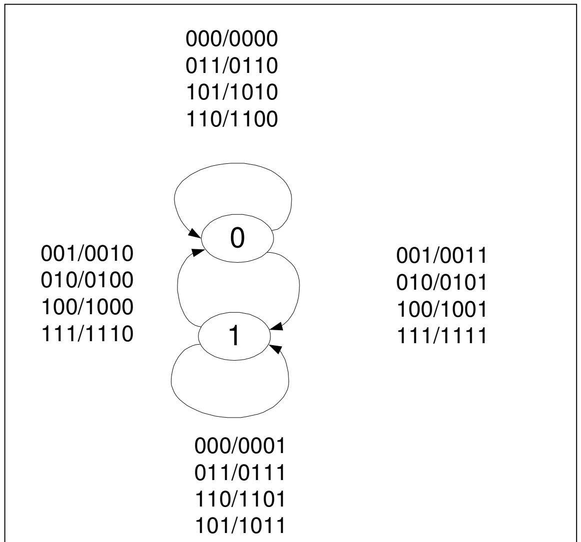
$$s^{(2)} = u(z) \cdot \frac{g^{(2)}(z)}{g^{(1)}(z)}$$



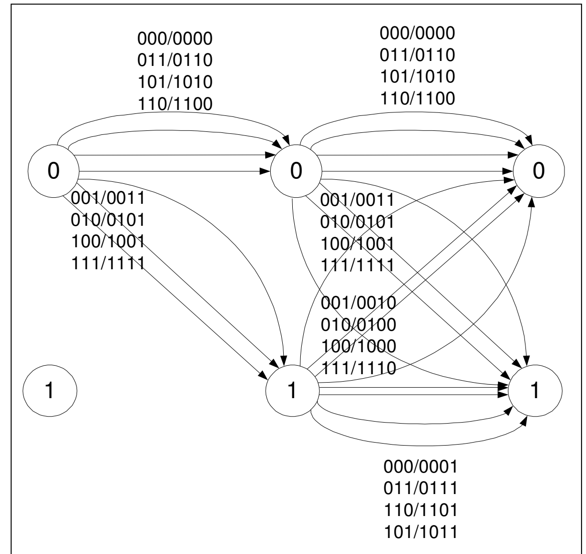
Exercise 6

The encoder starts from zero state, collects three bits and adds to them a parity bit calculated based on the previous state and the bit values. The information bits and parity bits are transmitted. In next interval the process is repeated but the initial state is equal to parity bit value in the next interval. Draw the trellis for given code.
(This type of code is called zigzag code).

Solution 7



The state diagram

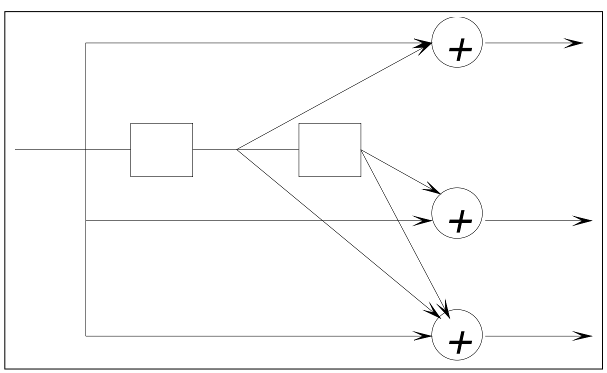


Trellis of the code

Exercise 8

The encoder block diagram is given in the figure below.

We find for it the impulse response, polynomial description, state space description, trellis description, and how Viterbi algorithm decodes



- An impulse response $g_i^{(j)}$ is obtained for the i-th output of an encoder by applying a single 1 at the j-th input followed by a string of zeros. In our example there is only one input stream $i = 1$ and three output steams $j = 3$. The impulse responses for the decoder in figure are

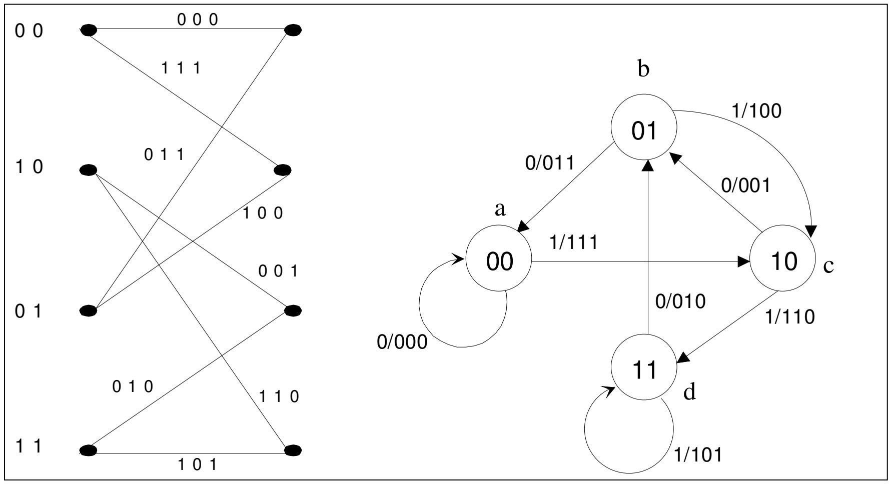
$$\overline{g}^{(0)} = (110)\overline{g}^{(1)} = (101)\overline{g}^{(0)} = (111).$$

- The transfer function matrix is found by applying the delay transform to the impulse responses. The indeterminate D indicates a delay and its exponent denotes the number of time units the coefficient is delayed.
- The D transform of the impulse response:

$$\mathrm{G}(D) = \left[1 + D \quad 1 + D^2 \quad 1 + D + D^2\right]$$

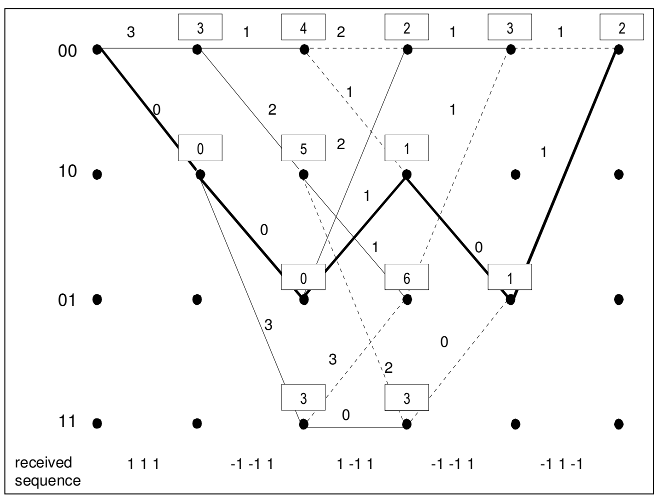
Solution 8

One step of the trellis is shown in the figure below. Compare to the state diagram shown on the right.



Decoding example: Decode the message using hard decision Viterbi algorithm.

- The transmitted bits were 111 001 011 000 000. This corresponds to message bits 10000.
- An encoded message at the output of the hard decision detector is 101 001 011 110 111
- The shift register values of the encoder are initially 00, and the message is terminated with a stream of zero bits.
- At the end encoder is filled with a stream of zeros which drives the decoder to known end state.
- Hard Decision Viterbi decoder: The transmitted sequence is compared to the candidate sequences and only hard decision of the decision are provided (1 0)

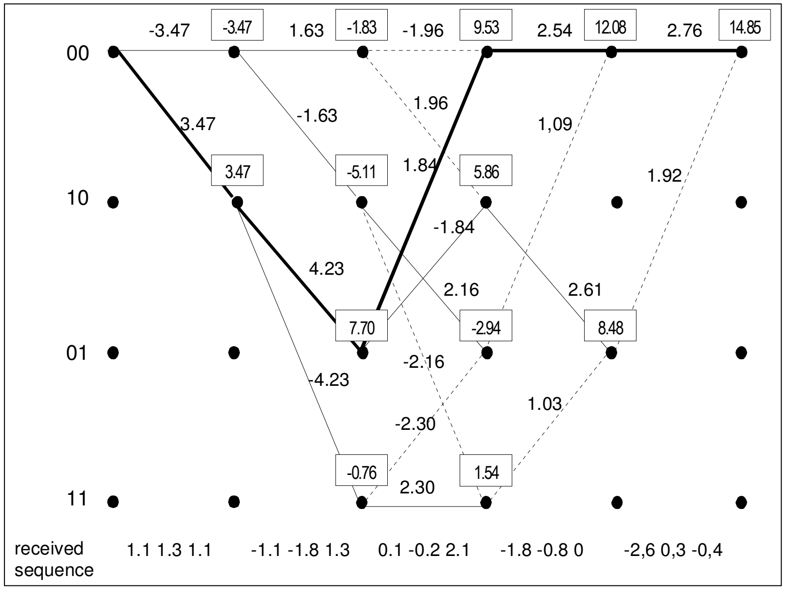


- Solid lines depict branch survivors, broken lines terminating branches. In case the metrics are equal we choose the lower path. The final survivor path shown in bold.

Soft Decision Viterbi decoder

- The received message at the output of a soft decision detector

y = [1.1139 1.2993 1.0568 -1.0944 -1.8377 1.2943 0.0662 -0.1643 2.0667 -1.8236 -0.7580 0.0334 -2.5940 0.2510 -0.4236]



Exercise 9

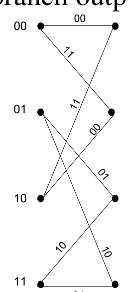
(Wicker, problem 12.9) Assume that the encoder from Problem 12.1 (see the book) is used over a symmetric memoryless channel. The bit metrics are as follows.

y/r	$\underline{0}$	0	1	$\underline{1}$
$y=0$	0	1	3	6
$y=1$	6	3	1	0

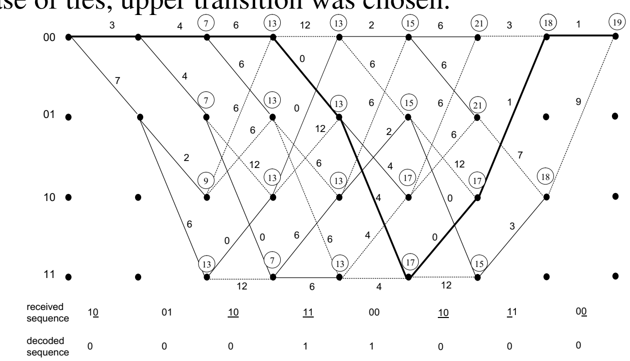
Find the maximum-likelihood code word corresponding to the following received sequence.

$$\mathbf{r} = \left(1\underline{0}, 01, \underline{1}\,\underline{0}, \underline{1}\,\underline{1}, 00, \underline{1}\,\underline{0}, \underline{1}\,1, 0\underline{0}\right).$$

One stage of the trellis and the branch outputs are shown below.



The decoding procedure is illustrated in the following figure (check it!). Dashed lines and solid lines indicate terminating branches and surviving branches, respectively. The survivor path is indicated with bold solid line. In case of ties, upper transition was chosen.

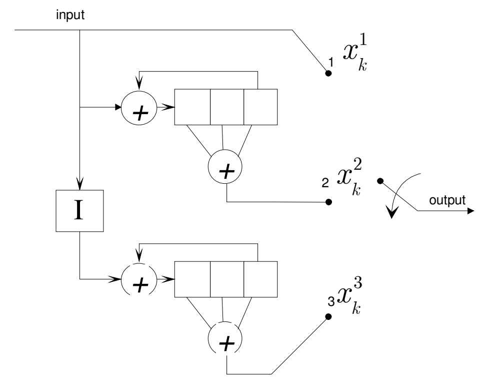


Exercise 10

Consider the encoder on Figure below.
Describe the decoders by the state diagram.
Assume that the I is a 3×3 block interleaver. Derive the output sequence c_k of the encoder if the input data sequence is

$$d_k = [1\,0\,1\,0\,1\,0\,0\,0\,1].$$

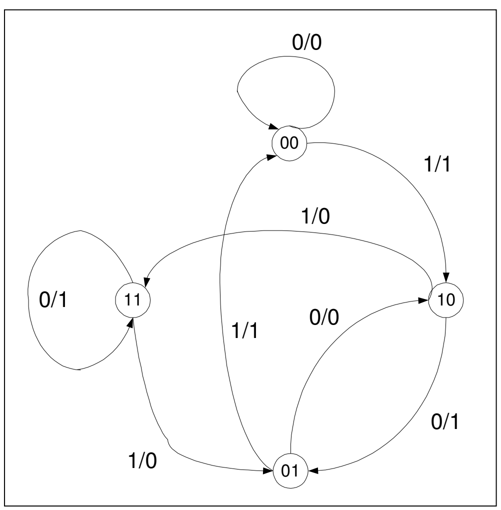
Give also the input sequence to the second encoder. (The data sequence after interleaver.)
Describe what kind of impact the two last bits of the input sequence have on the termination of the trellis of each encoder.



The bit sequence at the interleaver output is

$$d_k = [1\,0\,1\,0\,1\,0\,0\,0\,1] \Rightarrow d_{i,k} = [1\,0\,0\,0\,1\,0\,1\,0\,1]$$

The encoder state diagram is



Solution 10

The encoder outputs are

$$x_k^1 = [1\,0\,1\,0\,1\,0\,0\,0\,1]$$

$$x_k^2 = [1\,1\,1\,0\,1\,1\,0\,1\,1]$$

$$x_k^3 = [1\,1\,0\,1\,1\,0\,1\,1\,1]$$

Two last bits set the encoder to the 00 state. For this example this state is reached by both of the constituent encoders.