Problem Set 6 Solutions

Problem 6.1 (rational realizations).

(a) Generalize Figure 2 of Chapter 9 to realize any causal rational impulse response $g(D) = n(D)/d(D)$ with $\nu = \max\{\deg n(D), \deg d(D)\}$ memory elements, where $n(D)$ and $d(D)$ are both polynomials in $\mathbb{F}_2[D]$.

Since $g(D)$ is causal, we may assume that D does not divide $d(D)$. By multiplying the numerator and the denominator by the same scalar, we may further assume that $d_0 = 1$.

The desired realization is then as follows. Let $v(D)$ denote a sequence that enters a shift register of length $\nu = \max\{\deg n(D), \deg d(D)\}$. From this shift register we can obtain the sequences $Dv(D), D^2v(D), \ldots, D^\nu v(D)$. By calculating an appropriate linear combination of these sequences, we can obtain the sequence $f(D) = (d(D) - 1)v(D)$, which is fed back to the shift register input as shown in Figure 1 below. The shift register input is then

$$v(D) = u(D) - (d(D) - 1)v(D).$$

Solving this equation, we obtain $u(D) = d(D)v(D)$, or

$$v(D) = \frac{u(D)}{d(D)}.$$

Now by calculating an appropriate linear combination of the shift-register contents $Dv(D), D^2v(D), \ldots, D^\nu v(D)$, we can obtain the output sequence

$$y(D) = n(D)v(D) = \frac{n(D)}{d(D)}u(D),$$

which is the desired input-output map.

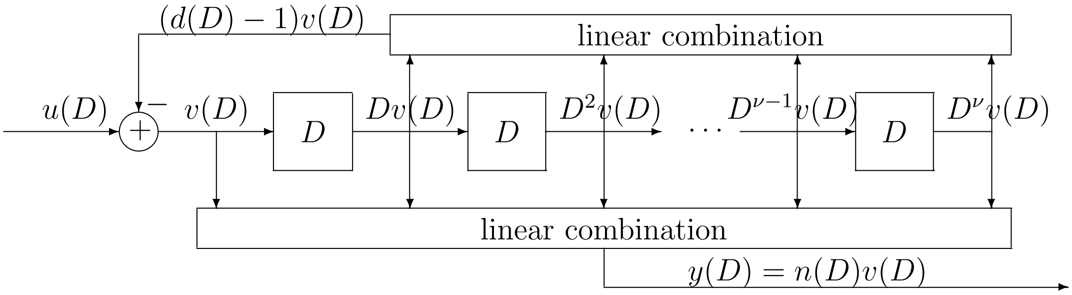


Figure 1. Realization of a linear system with impulse response $g(D) = \frac{n(D)}{d(D)}$ $(d_0 = 1)$.

(b) By a further generalization, show how to realize a rate-$1/n$ convolutional encoder with causal rational transfer function $\mathbf{g}(D) = \mathbf{n}(D)/d(D)$ with $\nu = \max\{\deg \mathbf{n}(D), \deg d(D)\}$ memory elements, where $\mathbf{n}(D)$ and $d(D)$ are polynomial.

By calculating n appropriate linear combinations of the shift-register contents $Dv(D), D^2v(D), \ldots, D^\nu v(D)$, we can obtain each of the n output sequences

$$y_j(D) = n_j(D)v(D) = \frac{n_j(D)}{d(D)}u(D), \quad 1 \le j \le n,$$

which give the n desired input-output maps. This realization is illustrated in Figure 2.

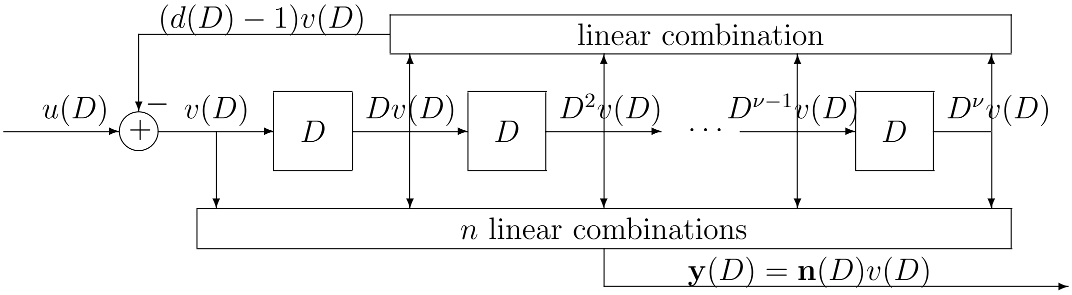


Figure 2. Realization of a rate-$1/n$ convolutional encoder with causal rational transfer function $\mathbf{g}(D) = \frac{\mathbf{n}(D)}{d(D)}$ ($d_0 = 1$).

Problem 6.2 (rational = eventually periodic).

Show that a Laurent D-transform $f(D) \in \mathbb{F}_2((D))$ is rational if and only if the corresponding Laurent sequence $\mathbf{f}$ is finite or eventually becomes periodic.

[Hints: (a) show that if a sequence $\mathbf{f}$ is eventually periodic with period P, then its D-transform $f(D)$ can be written as $f(D) = g(D)/(1 - D^P)$, where $g(D)$ is finite; (b) using the results of Problem 6.1(a), show that any causal rational Laurent D-transform $f(D) = n(D)/d(D)$ is the impulse response of a finite-state linear time-invariant system over $\mathbb{F}_2$, and therefore must be finite or eventually periodic.]

We first show that if $f(D)$ is finite, then $f(D)$ is rational. If $f(D)$ is finite with delay $\operatorname{del} f(D) = \delta \ge 0$, then $f(D)$ is rational because it is is polynomial. If $f(D)$ is finite with delay $\operatorname{del} f(D) = \delta < 0$, then $f(D)$ is rational because it can be written as $f(D) = (D^{-\delta} f(D))/(D^{-\delta})$, where both the numerator and denominator are polynomial.

Next, following hint (a), we show that if $f(D)$ is eventually periodic, then $f(D)$ is rational. If $f(D)$ is infinite and eventually periodic with period P starting at time γ, then $f(D)$ can be written as

$$f(D) = f_0(D) + D^\gamma \left(p(D) + D^P p(D) + D^{2P} p(D) + \cdots\right),$$

where $f_0(D)$ is finite with degree $\deg f_0(D) < \gamma$ and $p(D) \neq 0$ is polynomial with degree

$\deg p(D) < P$. Since $1 + D^P + D^{2P} + \cdots = 1/(1 - D^P)$, we can then write

$$f(D) = \frac{(1 - D^P)f_0(D) + p(D)}{1 - D^P}.$$

Since this is a ratio of finite sequences, $f(D)$ is rational.

Conversely, suppose that $f(D)$ is rational; *i.e.,* $f(D) = n(D)/d(D)$ for some polynomial $n(D)$ and $d(D) \neq 0$. Then we can prove that $f(D)$ is finite or eventually periodic by using hint (b). Using the result of Problem 6.1(a), we can realize a system with a causal rational impulse response $f(D) = n(D)/d(D)$ with $\nu = \max\{\deg n(D), \deg d(D)\}$ memory elements. (If $f(D)$ is not causal, consider instead the causal rational sequence $f'(D) = D^{-\mathrm{del}\, f(D)} f(D)$.) Since a realization with a finite number of memory elements has only a finite number of states, its impulse response must be finite or eventually periodic, because after the initial impulse, the system is autonomous (*i.e.,* there is no input), and an autonomous finite-state system must eventually cycle through a periodic sequence of states.

Problem 6.3 (input/output properties)

(a) If $\mathbf{y}(D) = u(D)\mathbf{g}(D)$ where $u(D)$ is Laurent and $\mathbf{g}(D) = \{n_j(D)/d_j(D)\}$ is causal and rational, show that $\mathbf{y}(D)$ is an n-tuple of formal Laurent series, $\mathbf{y}(D) \in (\mathbb{F}_2((D)))^n$.

Each rational function $g_j(D)$ may be identified with a formal Laurent series. Each $y_j(D)$ is then a convolution of two formal Laurent series, which is a well-defined formal Laurent series.

(b) Show that $\mathbf{y}(D)$ is rational if and only if $u(D)$ is rational; i.e., the rational subcode of $\mathcal{C} = \{\mathbf{y}(D) = u(D)\mathbf{g}(D) \mid u(D) \in \mathbb{F}_2((D))\}$ is

$$\mathcal{C}_r = \{\mathbf{y}(D) = u(D)\mathbf{g}(D), u(D) \in \mathbb{F}_2(D)\}.$$

If $u(D)$ is rational, then $y_j(D) = u(D)g_j(D)$ is the product of two rational functions and is thus rational. Conversely, if $y_j(D) = u(D)g_j(D)$ is rational, then $u(D) = y_j(D)/g_j(D)$ is the product of two rational functions and is thus rational. (We assume that at least one $g_j(D)$ is nonzero.)

(c) Show that $\mathbf{y}(D)$ is finite if and only if $u(D) = a(D)\mathrm{lcm}\{d_j(D)\}/\gcd\{n_j(D)\}$, where $a(D)$ is finite, $\mathrm{lcm}\{d_j(D)\}$ is the least common multiple of the denominators $d_j(D)$ of the $g_j(D)$, and $\gcd\{n_j(D)\}$ is the greatest common divisor of their numerators.

Since a finite sequence is rational, by part (b) we need consider only rational $u(D)$.

A rational function is finite if and only if when reduced to lowest terms its denominator polynomial is D^k for some k.

Again, the generator n-tuple $\mathbf{g}(D)$ has rational elements $g_j(D) = n_j(D)/d_j(D)$, which we may assume to have been reduced to lowest terms. To cancel all the denominator terms, $u(D)$ must be a multiple of all denominators, which means it must be a multiple $m(D)d(D)$ of their least common multiple $d(D) = \mathrm{lcm}\{d_j(D)\}$ for some finite $m(D)$.

If $u(D)$ has a nontrivial denominator term $b(D)$ other than D^k, then $y_j(D)$ can be finite only if $b(D)$ divides $n_j(D)$. Thus $\mathbf{y}(D)$ is finite only if $b(D)$ divides the greatest common divisor $n(D) = \gcd\{n_j(D)\}$; *i.e.,* $b(D) = n(D)/c(D)$ for some finite $c(D)$.

Thus we conclude that $\mathbf{y}(D)$ is finite if and only if

$$u(D) = \frac{m(D)d(D)}{n(D)/c(D)} = m(D)c(D)\frac{d(D)}{n(D)}$$

for some finite $m(D), c(D)$, which proves the proposition.

Problem 6.4 (SPC codes have a 2-state trellis diagram.)

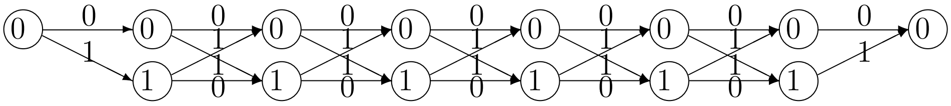
Show that if the (catastrophic) rate-1/1 binary linear convolutional code generated by $g(D) = 1 + D$ is terminated with deg $u(D) < \mu$, then the resulting code is a $(\mu + 1, \mu, 2)$ SPC code. Conclude that any binary linear SPC code may be represented by a 2-state trellis diagram.

In this case the terminated code is $\mathcal{C}_\mu = \{u(D)g(D) \mid \deg u(D) < \mu\}$, namely the set of all polynomials $y(D) = u(D)(1 + D)$ where $\deg u(D) < \mu$. Thus the total number of possibly nonzero input bits u_j is $k = \mu$, and the total number of possibly nonzero output bits y_j is $n = \mu + 1$, since

$$\deg y(D) = \deg u(D) + 1 < \mu + 1.$$

Finally, it is easy to see that a binary polynomial $y(D)$ has even Hamming weight if and only if $y(1) = 0$; *i.e.,* if and only if $y(D)$ is divisible by $1 + D$. Therefore $\mathcal{C}_\mu$ is the $(\mu + 1, \mu, 2)$ even-weight code; *i.e.,* the single-parity-check (SPC) code of length $\mu + 1$.

The rate-1/1 convolutional encoder with generator $g(D) = 1 + D$ may be realized by a binary shift register of length $\nu = 1$, which has 2 states. The trellis diagram of $\mathcal{C}_\mu$ is therefore a terminated 2-state trellis like this:



Two-state trellis for a binary $(\mu + 1, \mu, 2)$ single-parity-check code ($\mu = 6$).

Note that if this trellis were not terminated, then it would include an all-zero path in addition to the one associated with the all-zero state sequence, namely the path associated with the all-one state sequence. This proves that as a rate-1/1 convolutional encoder, the generator $g(D) = 1 + D$ is catastrophic. Indeed, the finite output sequence $y(D) = 1$ is generated by the infinite input sequence $u(D) = 1/(1 + D) = 1 + D + D^2 + D^3 + \cdots$.

Problem 6.5 (The $(7,4,3)$ Hamming code has an 8-state trellis diagram.)

Show that if the (catastrophic) rate-1/1 binary linear convolutional code generated by $g(D) = 1 + D + D^3$ *is terminated with* $\mu = 4$*, then the resulting code is a* $(7,4,3)$ *Hamming code.*

In this case the terminated code is $\mathcal{C}_4 = \{u(D)(1 + D + D^3) \mid \deg u(D) < 4\}$. Thus the total number of possibly nonzero input bits u_j is $k = 4$, and the total number of possibly nonzero output bits y_j is $n = 7$, since $\deg y(D) = \deg u(D) + 3 < 7$. Thus $\mathcal{C}_4$ is a $(7,4)$ binary linear block code with 16 codewords, namely the set of all polynomials of the form

$$u_0(1 + D + D^3) + u_1(D + D^2 + D^4) + u_2(D^2 + D^3 + D^5) + u_3(D^3 + D^4 + D^6).$$

By simply writing down the 16 codewords, we can establish that the minimum nonzero weight is $d = 3$, and in fact that $N_3 = 7, N_4 = 7$ and $N_7 = 1$. Thus $\mathcal{C}_4$ is a $(7,4,3)$ binary linear block code. Any code with these parameters is called a Hamming code.

Problem 6.6 (Viterbi algorithm decoding of SPC codes)

As shown in Problem 6.4, any $(\mu{+}1, \mu, 2)$ *binary linear SPC block code may be represented by a two-state trellis diagram. Let* $\mu = 7$*, and let the received sequence from a discrete-time AWGN channel be given by* $\mathbf{r} = (0.1, -1.0, -0.7, 0.8, 1.1, 0.3, -0.9, 0.5)$*. Perform Viterbi algorithm decoding of this sequence, using the two-state trellis diagram of the* $(8,7,2)$ *SPC code.*

We may first assign metrics to trellis branches so as to perform maximum-inner-product (MIP) detection: *i.e.,* maximize $\sum_j r_j s(y_j)$. (Alternatively, we could minimize the squared distance $||\mathbf{r} - s(\mathbf{y})||^2 = \sum_j (r_j - s(y_j))^2$.) In other words, we assign the metric r_j to branches corresponding to $y_j = 0$, and $-r_j$ to branches corresponding to $y_j = 1$.

At time 1 the two survivors to state 0 and state 1 therefore have metrics $+0.1$ and -0.1.

At time 2 we compare the two paths 00 and 11 to state 0, which have accumulated metrics -0.9 and $+0.9$, and choose the latter. Similarly we choose the path 01 to state 1, which has accumulated metric 1.1.

At time 3 we choose the path 011 to state 0, which has metric 1.8, and the path 111 to state 1, which has metric 1.6.

Time 4: path 0110 to state 0, metric 2.6; path 1110 to state 1, metric 2.4.

Time 5: path 01100 to state 0, metric 3.7; path 11100 to state 1, metric 3.5.

Time 6: path 011000 to state 0, metric 4.0; path 111000 to state 1, metric 3.8.

Time 7: path 1110001 to state 0, metric 4.7; path 0110001 to state 1, metric 4.9.

Time 8: path 11100010 to end state 0, metric 5.2.

Compare and contrast the performance and complexity of VA decoding to that of "Wagner decoding" (Problem 4.6) for this example.

For Wagner decoding, we first make hard decisions on every bit, yielding the word 01100010. Since this word has odd weight, it is not a codeword. We then find the least reliable hard decision, *i.e.,* the r_j with least magnitude; this occurs in the first position. We flip this bit to obtain the even-weight codeword 11100010, which must be the ML = MD = MIP codeword, as shown in Problem 4.6. Indeed, this is the same codeword as is decoded by the VA in this case.

Note that the two VA surviving paths at any time differ in precisely one position, the position of the least reliable hard decision up to that time. It can be seen that this will be true in general; this gives another proof of the optimality of Wagner decoding.

Wagner decoding clearly requires fewer arithmetic operations than VA decoding, although its logical structure is somewhat less regular.