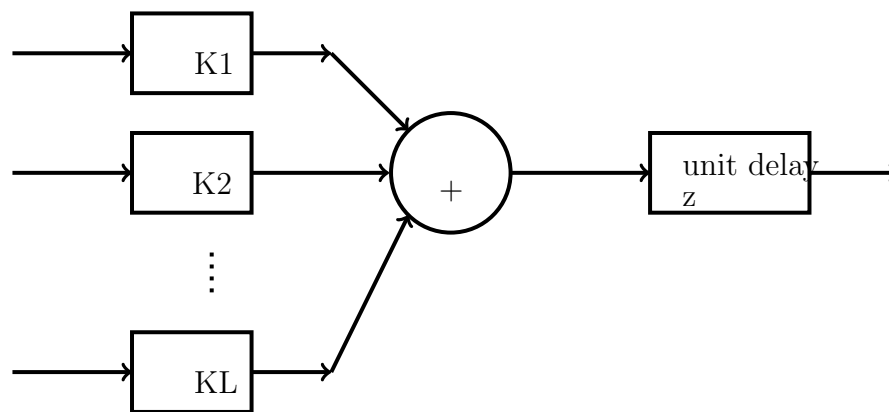


6.1 Cyclic Networks with LTI codes

In Figure 7.1, we see a model of a node in a graph. We want to look at each outgoing edge. Each edge is assumed to have unit delay. Notice that unit delays will be denoted with z , which is slightly unorthodox notation relative to the z^{-1} common in DSP, but is kept here for consistency with Raymond Yeung's book.



LTI systems

Figure 6.1. An LTI view of a node.

6.1.1 Example

In Figure 7.2, we see a cyclic graph. This graph actually has a routing solution, but for our purposes we will see what happens when the coding matrices are as shown in the figure (each node adds all its inputs). Table 7.1.1 shows what is going on at each edge over time.

Clearly, the resulting matrices will be invertible and decoding is possible. However we want to find a time-invariant perspective on this solution. Since everything is LTI, the idea

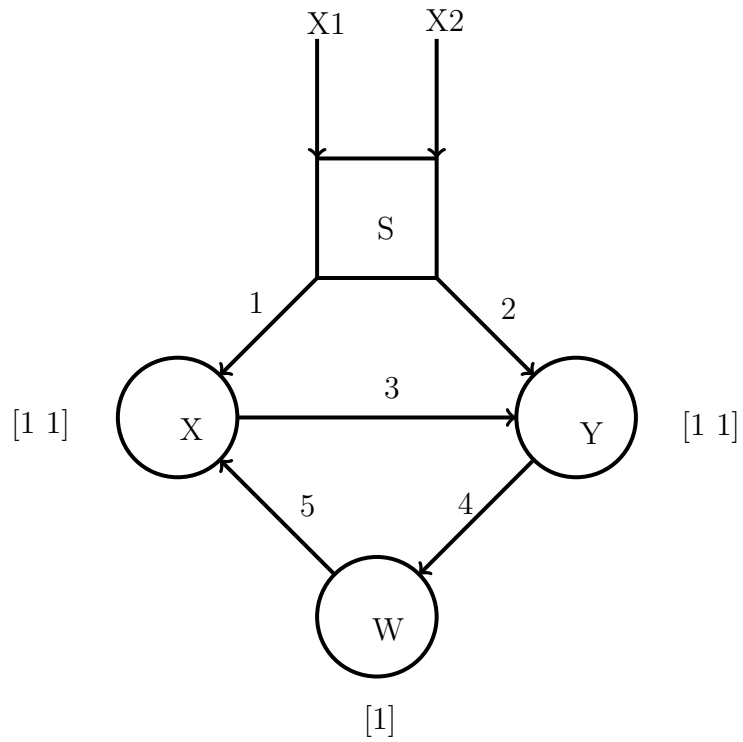


Figure 6.2. Example cyclic graph.

time	1	2	3	4	5
1	$x_1(1)$	$x_2(1)$	0	0	0
2	$x_1(2)$	$x_2(2)$	$x_1(1)$	$x_2(1)$	0
3	$x_1(3)$	$x_2(3)$	$x_1(2)$	$x_2(2) + x_1(1)$	$x_2(1)$
4	$x_1(4)$	$x_2(4)$	$x_1(3) + x_2(1)$	$x_2(3) + x_1(2)$	$x_2(2) + x_1(1)$
5	\vdots	\vdots	\vdots	\vdots	\vdots

Table 6.1. What each edge is sending over time.

is to rewrite the problem in z-transform domain:

$$B_1(z) = zX_1(z) \quad (6.1)$$

$$B_2(z) = zX_2(z) \quad (6.2)$$

$$B_3(z) = zB_1(z) + zB_5(z) \quad (6.3)$$

$$B_4(z) = zB_2(z) + zB_3(z) \quad (6.4)$$

$$B_5(z) = zB_4(z) \quad (6.5)$$

$$(6.6)$$

We can solve these to get

$$B_3(z) = \frac{z^2}{1-z^3}X_1(z) + \frac{z^4}{1-z^3}X_2(z) \quad (6.7)$$

$$B_4(z) = \frac{z^3}{1-z^3}X_1(z) + \frac{z^4}{1-z^3}X_2(z) \quad (6.8)$$

$$B_5(z) = \frac{z^4}{1-z^3}X_1(z) + \frac{z^3}{1-z^3}X_2(z) \quad (6.9)$$

$$(6.10)$$

In vector notation

$$B(z) = \begin{bmatrix} B_1(z) \\ B_2(z) \\ B_3(z) \\ B_4(z) \\ B_5(z) \end{bmatrix}, \quad X(z) = \begin{bmatrix} X_1(z) \\ X_2(z) \end{bmatrix}$$

$$B(z) = z \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_{K(z)}, + z \underbrace{\begin{bmatrix} I^{2 \times 2} \\ \mathbf{0} \end{bmatrix}}_{H_s(z)} X(z)$$

Solve for $B(z)$

$$B(z) = \underbrace{(I - zK(z))^{-1} zH_s(z)}_{G(z)} X(z)$$

Thus, we need to invert

$$(I - zK(z))^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -z & 0 & 1 & 0 & -z \\ 0 & -z & -z & 1 & 0 \\ 0 & 0 & 0 & -z & 1 \end{bmatrix}^{-1} = \frac{1}{\det(\cdot)} \begin{bmatrix} \text{compute det} \\ \text{of submatrices} \\ (5-1) \times (5-1) \\ \text{polynomials of degree} \\ \text{at most 4} \end{bmatrix}$$

We can always do this inversion, since the $\det(\cdot)$ is a function of z and therefore not identically zero. Consider $G(z)$ restricted to node X.

$$G(z) = \begin{bmatrix} z & 0 \\ \frac{z^4}{1-z^3} & \frac{z^3}{1-z^3} \end{bmatrix}, \quad G^{-1}(z) = \frac{1-z^3}{z^4} \begin{bmatrix} \frac{z^3}{1-z^3} & 0 \\ -\frac{z^4}{1-z^3} & z \end{bmatrix} = \begin{bmatrix} z^{-1} & 0 \\ -1 & \frac{1-z^3}{z^3} \end{bmatrix}$$

We observe that the decoder at node X is non-causal. (remember that z corresponds to unit delay in our notation, not z^{-1}) This is clearly a problem since it is aphysical. However, we can get around this by introducing delay as follows:

$$G^{-1}(z) = \underbrace{z^{-3}}_{\text{finite pure anticipation}} \underbrace{\begin{bmatrix} z^2 & 0 \\ -z^3 & 1-z^3 \end{bmatrix}}_{\text{causal}}$$

Since G^{-1} is in this form, we can introduce some delay and the decoding will be causal. The important point to note is that this delay will always be finite, because the $\det(G(z))$ involves at most degree $5-1$ polynomials in z . Also note that this delay is expected, since even the routing solution has a delay of 3.

In general the argument in this example works when the matrices are of size $|E| \times |E|$, where $|E|$ is the number of edges, and the input vector is of size R , the rate.

6.1.2 Choosing the field size

Notice that the $|E| \times R$ global coding matrix $G(z)$, has entries from rational transfer functions in z . This is the field we will be working with. Call it $F \langle z \rangle$. Thus, the α 's, which are elements of the coding matrices at each edge, also come from $F \langle z \rangle$. We want to make an existence argument, similar to the DAG case, to show that we can always achieve invertibility. Remind yourself that the proof in the DAG case relied on the fact that the final decoding matrices were polynomial functions of the α 's. However in this case we have,

$$G(z) = \frac{1}{\det(I - zK(z))} \underbrace{\text{Adj}(I - zK(z)) zH(z)}_{G'(z)}$$

which is not a polynomial in α , due to the dividing determinant. However, this is not a problem. Let $G_t(z)$ be defined as the restriction of $G(z)$ to destination t . (Assume this is square.) We already argued that $\det(I - zK(z))$ cannot be identically zero. This means

$$\det(G_t(z)) = 0 \iff \det(G'_t(z)) = 0$$

Therefore, we simply work with G'_t , which is an $R \times R$ matrix filled with polynomial functions of α 's, of degree at most $|E| - 1$. Therefore $\det(G'_t(z))$ is a polynomial of degree at most $R(|E| - 1)$. Call this polynomial $g_t(\alpha)$. If we assume $R = \min \text{ min cut}$, then at least the routing solution exists, and $g_t(\alpha)$ cannot be zero for all α . Now consider the determinant of the global matrix,

$$g(\alpha) = \prod_t g_t(\alpha)$$

this has degree at most $|T||R|(|E| - 1)$. Thus, by choosing a field size large enough, we can ensure a non-zero solution. The only problem remains that we need a non-zero solution α^* such that α^* are causal. The following theorem guarantees that we can do this:

Theorem 6.1. *If $g(\alpha_1, \alpha_2, \dots, \alpha_k)$ is a polynomial in α 's with coefficients not all zero from a field F , then if we restrict α 's to come from a subset $|E'|$ of F s.t.*

$$|E'| > \max \text{ degree of } \alpha \text{'s in } g$$

there exists α^ such that $g(\alpha^*)$ is not the zero polynomial.*

Proof: See Exercise 1. This follows from the exact same argument that we did in the DAG case except you just have to notice that we never used the structure of E' , just its size. \square

6.2 Exercises

1. Prove theorem 7.1.
2. Show that if the underlying field is of size L , then randomly choosing memoryless network coding matrices will succeed with $P = (\text{fill this in})$.
3. Show that $P \rightarrow 1$ as $L \rightarrow \infty$.
4. Comment on the difference between DAG and cyclic cases.
- 4a. Would you ever want to use a network code with memory?