# TSKS04 Digital Communication
## *Continuation Course*

Lecture 10

Convolutional codes: Encoding and decoding

Emil Björnson

Department of Electrical Engineering (ISY)
Division of Communication Systems

# Binary Sequences

Consider: $a = a_0, a_1, a_2, \ldots$ for $a_i \in \mathbb{F}_2$

- $D$-transform: Write sequence using delay operator $D$
- $A(D) = a_0 + a_1 D + a_2 D^2 + \cdots$

Convolution of binary sequences:

- $P(D) = p_0 + p_1 D + p_2 D^2 + \cdots + p_m D^m$
- Consider: $X(D) = A(D)P(D)$
- Time representation $x = a * p$:

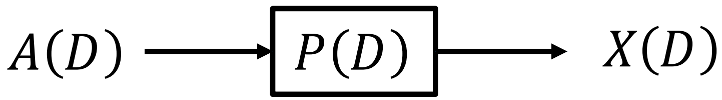
$$x_m = \sum_{k=0}^{m} a_k p_{m-k} = \sum_{k=0}^{m} a_{m-k} p_k$$

# Filtering of Sequences

Input sequence: $A(D) = a_0 + a_1 D + a_2 D^2 + \cdots$

Filter $p$:

- Impulse response: $p = p_0, \ldots p_m$
- Transfer function: $P(D) = p_0 + p_1 D + p_2 D^2 + \cdots + p_m D^m$



# Multiple Inputs and Multiple Outputs (1/2)

(semi-)infinite $k$-dimensional binary sequence

Input sequences

$$\bar{a} = \left[a^{(1)}, a^{(2)}, \dots, a^{(k)}\right] \quad A(D) = \left[A^{(1)}(D), A^{(2)}(D), \dots, a^{(k)}(D)\right]$$

$k \times n$ matrix impulse response: $G = \left\{g^{(i,j)}\right\} \quad G(D) = \left\{G^{(i,j)}(D)\right\}$

$$g^{(i,j)} = g_0^{(i,j)}, g_1^{(i,j)}, g_2^{(i,j)}, \dots$$

$k \leq n$

Each element is a (semi-)infinite binary sequence

Matrix convolution:

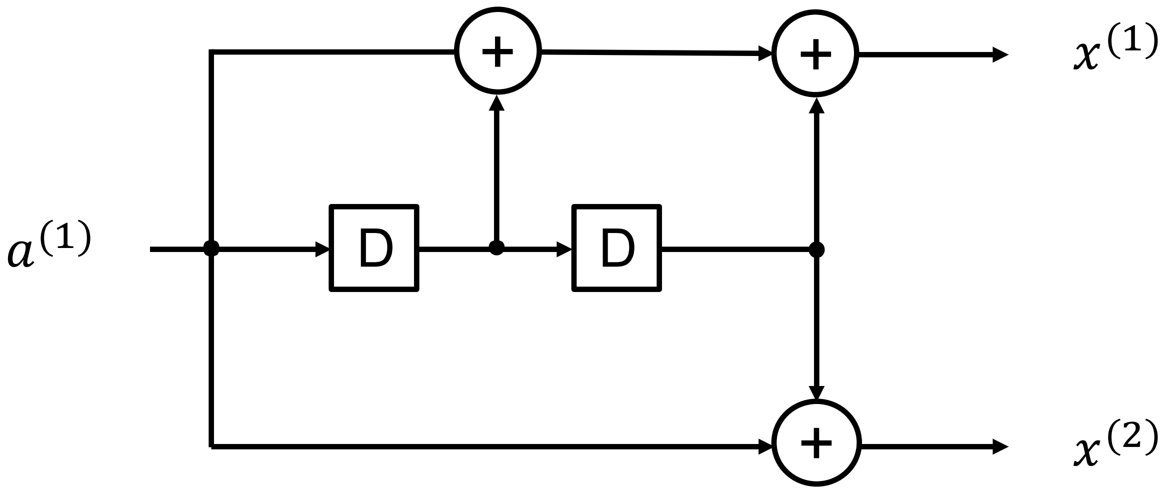
$$\bar{x} = \bar{a} * G = \left[x^{(1)}, x^{(2)}, \dots, x^{(n)}\right]$$

$$x_m^{(j)} = \sum_{i=1}^{k} \sum_{t=0}^{m} a_m^{(i)} g_{m-t}^{(i,j)} = \sum_{i=1}^{k} a^{(i)} * g^{(i,j)}$$

# Multiple Inputs and Multiple Outputs (2/2)

Example: $k = 1, n = 2$

$$G(D) = (1 + D + D^2, 1 + D^2)$$



# Convolutional Code

Source: Binary information sequence $\boldsymbol{u} = \{u[k]\}$

- $k$ inputs to encoder
- $n$ outputs from encoder

Coding rate: $R = \frac{k}{n}$

- Generator matrix: $G(D) = \begin{pmatrix} G^{(1,1)}(D) & \cdots & G^{(1,n)}(D) \\ \vdots & \ddots & \vdots \\ G^{(k,1)}(D) & \cdots & G^{(k,n)}(D) \end{pmatrix}$

Generator polynomials: $G^{(1,1)}(D), \ldots, G^{(k,n)}(D)$ ← Impulse responses

- Naming: Read coefficients from binary
- Example: $G(D) = (1 + D + D^2, 1 + D^2)$ is called [7,5]

# Basic Properties (1/2)

Nonrecursive (or polynomial)

- Codewords produced by passing information sequence through a *finite impulse response feedforward* filter
- All generator polynomials are polynomials (FIR filters)
- Memory $m$: Number of previous inputs used at current output

Recursive

- Codeword generation involves feedback (use of previous outputs)

# Basic Properties (2/2)

Systematic

- One of the output sequences is the information sequence
- One generator polynomial is $g(D) = 1$

Non-systematic

- None of the output sequences is the information sequence

# Catastrophic Encoder

If $G(D)$ is a generator matrix, then

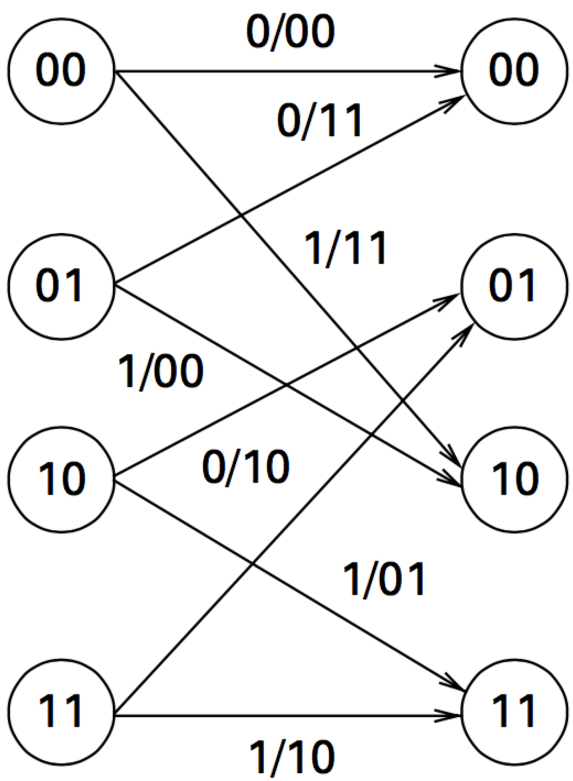
$$G'(D) = \underbrace{(1 + p_1 D + \cdots p_m D^m)}_{P(D)} G(D)$$

is it as well. They are equivalent in the sense that $P^{-1}(D)$ exists.

**What is the best encoder?**

**Definition:** A catastrophic encoder can give $X(D) = A(D)G(D)$ with finite Hamming weight even if $A(D)$ has infinite weight.

# Trellis representation (1/2)




State at time $n$

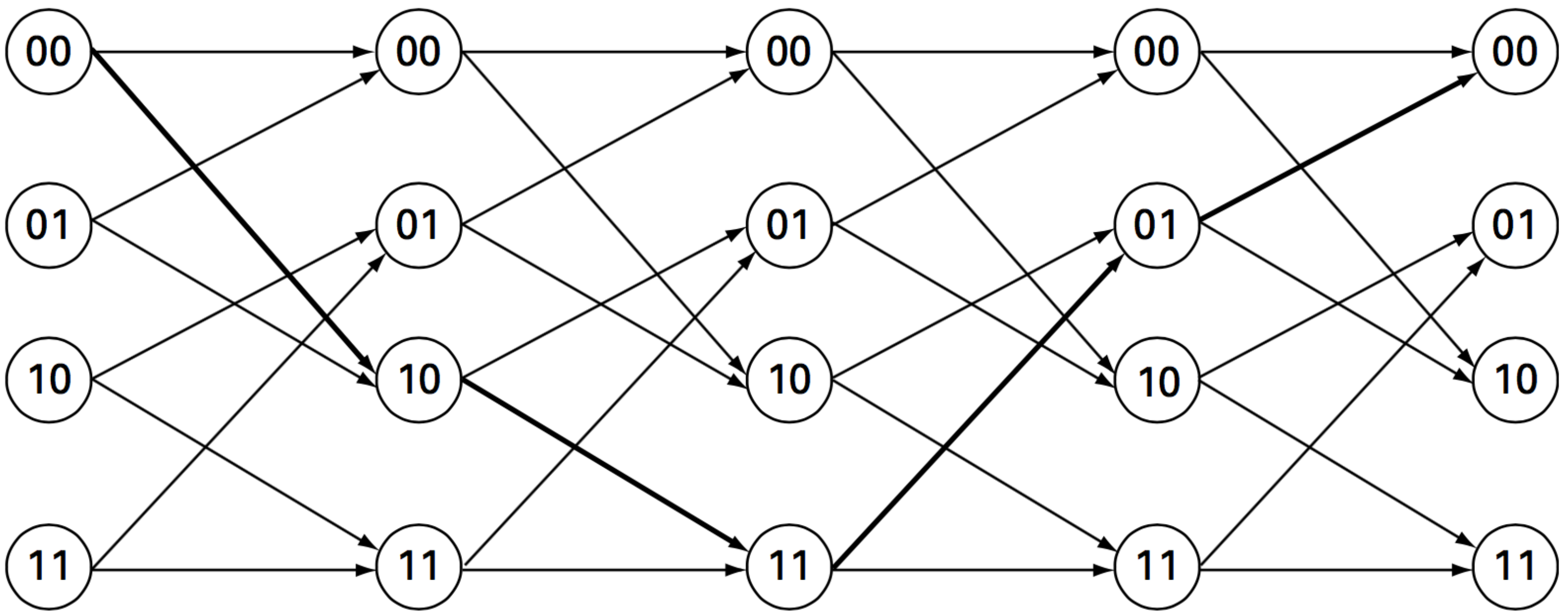
State at time $n+1$

Branch label:
$a_n / y_n^{(1)} y_n^{(2)}$

Number of states:
$2^m$ (here: 2)

Number of paths into/reaching a state:
$2^k$

# Trellis representation (2/2)




**One way from left to right represents**:

One data sequence

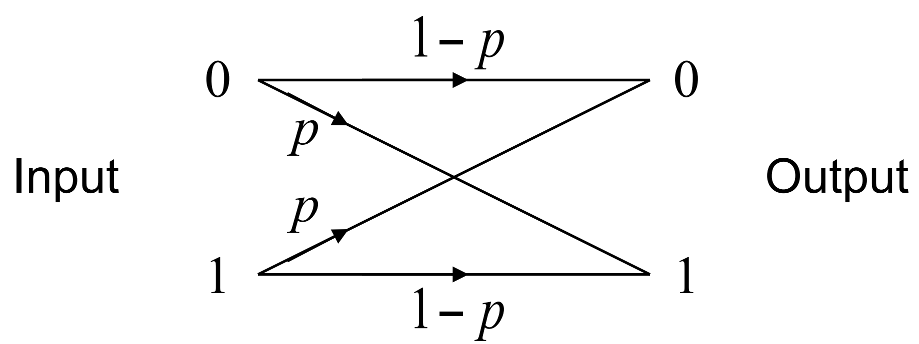
Produces one code sequence

# Decoding of Convolutional Code

**Decoding**: Obtain $\hat{a}$ from received bit sequence that contains errors

**Approach:**

- Use trellis representation and apply Viterbi algorithm
- Begin in state 00 and end in state 00
- If binary symmetric channel with $p \leq 0.5$:
  Choose path with fewest bit errors




$p$ = Bit error probability

LINKÖPING UNIVERSITY

www.liu.se