

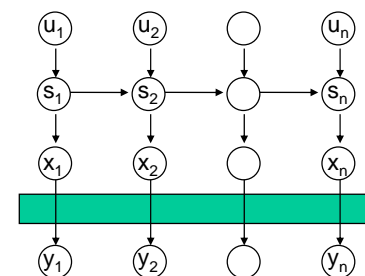
Számítógépes Hálózatok és Internet Eszközök

2007

7. Adatkapcsolati réteg -- konvolúciós kódok, Viterbi dekódolás, Turbo kódok

Konvolúciós kódok

- Konvolúciós kódok (Convolutional Codes)
 - Adatok és a redundancia a hibajavításhoz nincsenek szétválasztva.
 - k bitet n bitre képezünk le
 - A kimenet az utolsó k bittől és a belső állapottól függ.



u: eredeti üzenet
s: encoder (állapotai)
x: kódolt üzenet
zajos csatorna
y: fogadott üzenet

Konvolúciós kódok dekódolása: Viterbi Algoritmus

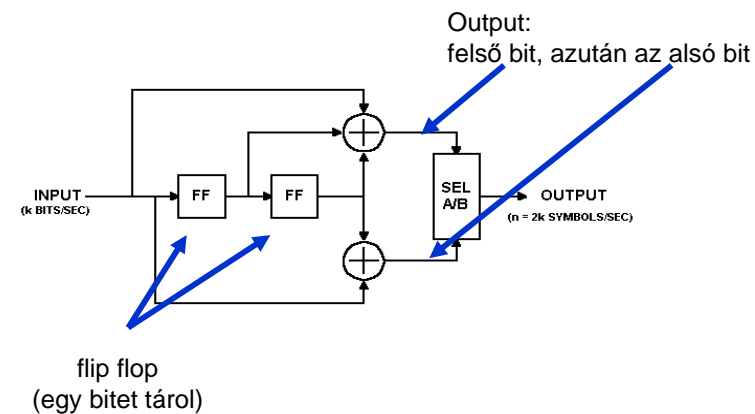
Viterbi algoritmusához a mai előadás alapja :

- Chip Fleming: **A Tutorial on Convolutional Coding with Viterbi Decoding**, <http://home.netcom.com/~chip.f/viterbi/tutorial.html>

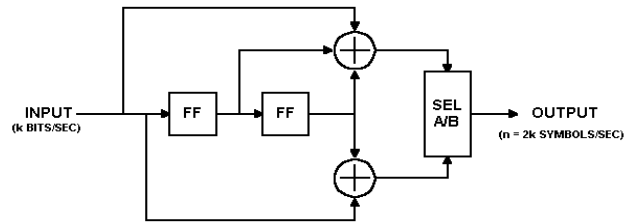
A Viterbi dekódolás eredete:

- Andrew J. Viterbi: **Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm**, *IEEE Transactions on Information Theory*, Volume IT-13, pp. 260-269, April 1967.
- Viterbi a Qualcomm egyik alapítója.

Konvolúciós Enkóder



Kódolás példa



Kezdetben mindkét flip flop tartalma 0

Input: 0 1 0 1 1 1 0 0 1 0 1 0 0 0 1

Output: 00 11 10 00 01 10 01 11 11 10 00 10 11 00 11

Befejezéskor „kiürítjük” az enkódert (flush): termináljuk az inputot két 0-val

Az állapot átmenet- és az output tábla

Aktuális állapot	Következő állapot	
	Input = 0 esetén	Input = 1 esetén
00	00	10
01	00	10
10	01	11
11	01	11

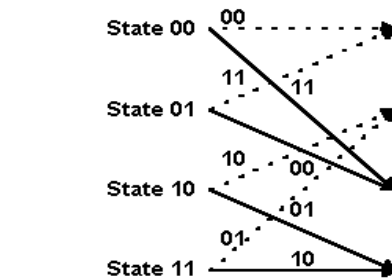
Állapot átmenet tábla

2^{k-1} sor, 2 oszlop

Aktuális állapot	Output	
	Input = 0:	Input = 1:
00	00	11
01	11	00
10	10	01
11	01	10

Output tábla

Állapot átmenetek

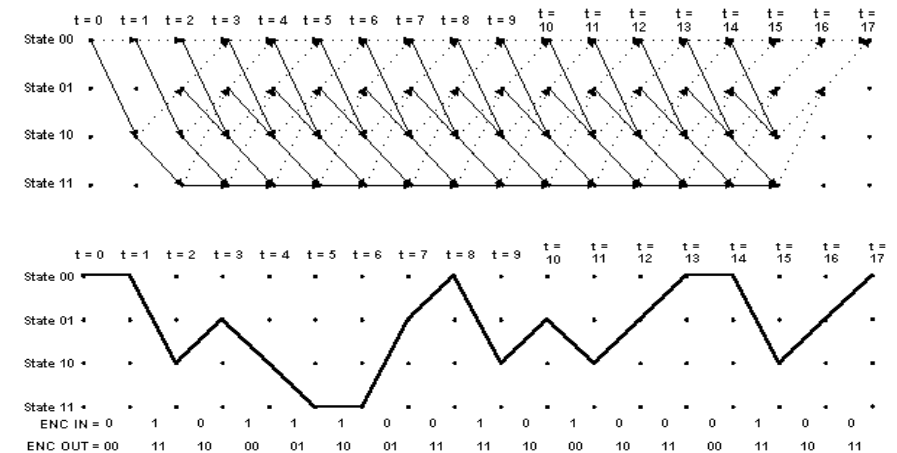


—————> input bit 1

.....> input bit 0

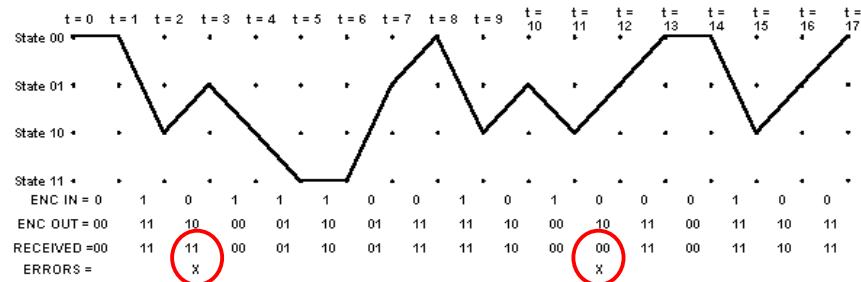
Az éleket megcímkézzük az output bitekkel (output kóddal)

Trellis-diagramm



Minden helyesen fogadott bitsorozatnak egyértelműen megfelel egy útvonal

Hiba

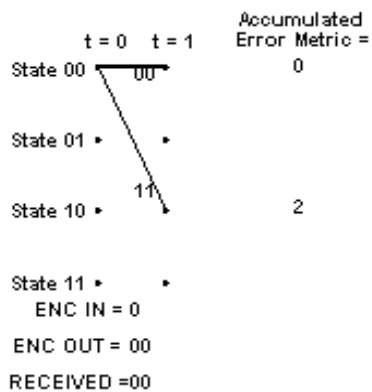


A korábban fogadott bitekből minden e élhez meghatározható, hogy hány bithibának kellett legalább történni akkor, ha az enkóder egy olyan utat járt be a trellis diagrammban, amely e-t tartalmazza.

Konvolúciós kódok dekódolása: Viterbi Algoritmus

- Dinamikus programozás a legvalószínűbb útvonal meghatározásához
- A dekódoláshoz szükséges
 - a lehetséges állapotátmenetek (a trellis) ismerete
 - a fogadott bitek megfigyelt sorozata (lehetséges hibával)
- Viterbi algoritmus meghatározza az állapotok legvalószínűbb sorozatát, amely a fogadott biteket magyarázza
 - Hardware implementáció lehetséges

Akkumulált hiba metrika

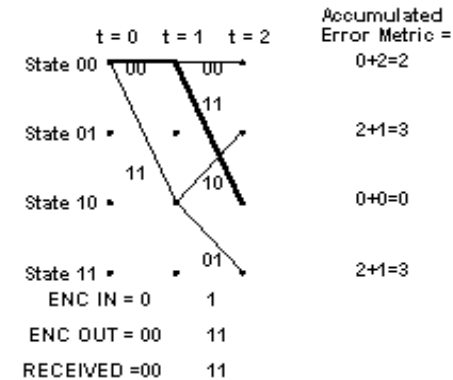


- Minden $t=1,2,\dots$ időpontban tekintjük a fogadott bitek távolságát azoktól a lehetséges bitektől, amiket t pillanatban fogadhattunk volna: a fogadott bitek távolságát a trellis diagramm éleinek címkeihez
- Használjuk pl. a Hamming távolságot a fogadott bitek x és az e él címkeje között. Jelölje $d(x,e)$ ezt a távolságot
- s állapothoz t időpontban az **akkumulált hiba** $err(s,t)$:

$$err(s,t) = \begin{cases} 0, & \text{ha } s = 00, \\ \infty, & \text{ha } s \neq 00. \end{cases}$$

$$err(s,t) = \min_{e=(s',s)} \{err(s',t-1) + d(x,e)\}$$

Akkumulált hiba



———— legvalószínűbb útvonal

Az akkumulált hiba az időben

t =	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
State 00_2		0	2	3	3	3	3	4	1	3	4	3	3	2	2	4	5	2
State 01_2			3	1	2	2	3	1	4	4	1	4	2	3	4	4	2	
State 10_2		2	0	2	1	3	3	4	3	1	4	1	4	3	3	2		
State 11_2			3	1	2	1	1	3	4	4	3	4	2	3	4	4		

A túlélő megelőző állapotok

t =	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
State 00_2	0	0	0	1	0	1	1	0	1	0	0	1	0	1	0	0	0	1
State 01_2	0	0	2	2	3	3	2	3	3	2	2	3	2	3	2	2	2	0
State 10_2	0	0	0	0	1	1	1	0	1	0	0	1	1	0	1	0	0	0
State 11_2	0	0	2	2	3	2	3	2	3	2	2	3	2	3	2	2	0	0

A túlélő megelőző állapotok

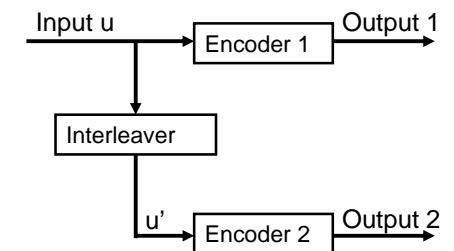
t =	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
State 00_2	0	0	0	1	0	1	1	0	1	0	0	1	0	1	0	0	0	1
State 01_2	0	0	2	2	3	3	2	3	3	2	2	3	2	3	2	2	2	0
State 10_2	0	0	0	0	1	1	1	0	1	0	0	1	1	0	1	0	0	0
State 11_2	0	0	2	2	3	2	3	2	3	2	2	3	2	3	2	2	0	0

Az állapotok visszakövetve

t =	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
	0	0	2	1	2	3	3	1	0	2	1	2	1	0	0	2	1	0

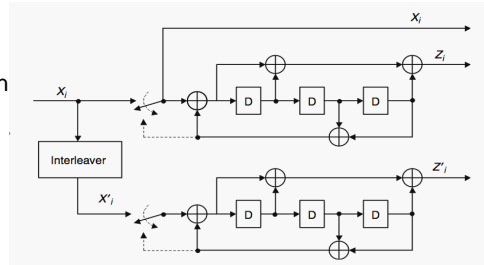
Turbo kódok

- Turbo-kódok lényegesen hatékonyabbak, mint a konvolúciós kódok
 - két konvolúciós kód kombinációja
 - két konvolúciós enkódert tartalmaz, melyek párhuzamosan dolgoznak a bemeneten
 - A bemenetet egy u.n. „interleaver” más sorrendbe rendezi (egy permutációt állít elő) a második konvolúciós kódhoz



Turbo kódok

- Példa: UMTS Turbo kód
 - 2 rekurzív konvolúciós kód
 - Párhuzamosan dolgoznak az inputon
 - Output: $X_1Z_1Z'_1X_2Z_2Z'_2X_3Z_3Z'_3 \dots$
 - Az „interleaver” az inputnak egy permutációját állítja elő



• pl. 1 0 0 1 0 0 0 1 1 1 0
 0 0 1 0 1 1 0 0 1 0 1

- Turbó kódok dekódolása hatékonyabban lehetséges, mint a konvolúciós kódé:

- sokkal robusztusabbak löketszerűen (burst) érkező hibákkal szemben

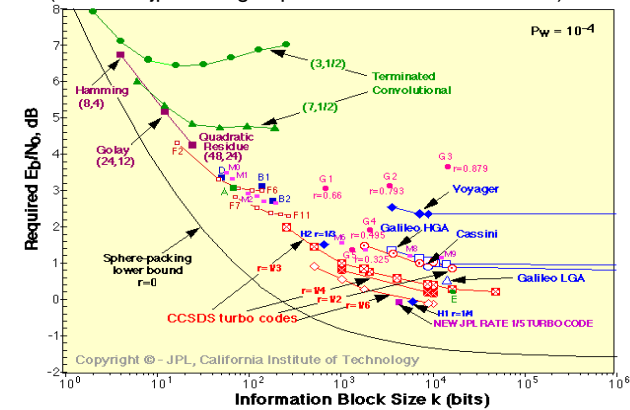
pl. 1 0 0 1 0 0 0 x x x x
 0 0 x 0 1 0 x 0 x 0 x

konvolúciós kódok számára ilyen hibák végzetesek, a permutáció „elosztja” a hibákat a sorozatban

Kódok összehasonlítása

- Kód ráta versus „signal-to-noise ratio”

- 1998: (www331.jpl.nasa.gov/public/AllCodesVsSize.GIF)



Irodalom

- Chip Fleming: A Tutorial on Convolutional Coding with Viterbi Decoding
<http://home.netcom.com/~chip.f/viterbi/tutorial.html>
- Berrou, C., Glavieux, A. and Thitimajshima, P., "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," ICC, pp 1064-1070, 1993.
- **Néhány könyv előre hibajavításról**
- S. Lin and D. J. Costello, *Error Control Coding*. Englewood Cliffs, NJ: Prentice Hall, 1982.
- A. M. Michelson and A. H. Levesque, *Error Control Techniques for Digital Communication*. New York: John Wiley & Sons, 1985.
- W. W. Peterson and E. J. Weldon, Jr., *Error Correcting Codes*, 2nd ed. Cambridge, MA: The MIT Press, 1972.
- V. Pless, *Introduction to the Theory of Error-Correcting Codes*, 3rd ed. New York: John Wiley & Sons, 1998.
- C. Schlegel and L. Perez, *Trellis Coding*. Piscataway, NJ: IEEE Press, 1997
- S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Englewood Cliffs, NJ: Prentice Hall, 1995.

További irodalom (Fleming)

- **Néhány cikk konvolúciós kódokról Viterbi dekódolással**
- For those interested in VLSI implementations of the Viterbi algorithm, I recommend the following paper and the papers to which it refers (and so on):
- Lin, Ming-Bo, "New Path History Management Circuits for Viterbi Decoders," *IEEE Transactions on Communications*, vol. 48, October, 2000, pp. 1605-1608.
- Other papers are:
- G. D. Forney, Jr., "Convolutional Codes II: Maximum-Likelihood Decoding," *Information Control*, vol. 25, June, 1974, pp. 222-226.
- K. S. Gilhousen et. al., "Coding Systems Study for High Data Rate Telemetry Links," Final Contract Report, N71-27786, Contract No. NAS2-6024, Linkabit Corporation, La Jolla, CA, 1971.
- J. A. Heller and I. M. Jacobs, "Viterbi Decoding for Satellite and Space Communications," *IEEE Transactions on Communication Technology*, vol. COM-19, October, 1971, pp. 835-848.
- K. J. Larsen, "Short Convolutional Codes with Maximal Free Distance for Rates 1/2, 1/3, and 1/4," *IEEE Transactions on Information Theory*, vol. IT-19, May, 1973, pp. 371-372.
- J. P. Odenwalder, "Optimum Decoding of Convolutional Codes," Ph. D. Dissertation, Department of Systems Sciences, School of Engineering and Applied Sciences, University of California at Los Angeles, 1970.
- A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Transactions on Information Theory*, vol. IT-13, April, 1967, pp. 260-269.