Translačné Kódy: *ich úlohou je prispôsobiť prenos obmedzeniam kanála*

Charakteristický polynóm - $p(\lambda)=det(\lambda I-B)$

Kapacita kanála - $C'=\log_2(max(root(det(\lambda I-B))))$

Prenosová a modulačná rýchlosť - $R_p=R_m*C'[\frac{bit}{symbol}]$

Rýchlosť kódu $R_k=\frac{k}{n}$ Nutná podmienka existencie **TK** $R_k\leq C'$!!!!

Postačujúca podmienka $\sum_j b_{ij}^n\geq 2^k$; $\forall i$ kde b_{ij}^n je príslušný prvok B^n

Lineárne Blokové Kódy: *všetky kódové slová majú rovnakú vzdialenosť*,

počet kódových slov = počtu informačných slov,

LBK k rozmerný podpriestor n rozmerného vektorového priestoru

Princíp linearity: Pre všetky kódové slová musí platiť, že lineárna kombinácia ľubovoľných kódových slov je tiež kódové slovo.

Hammingova váha kódového slova: w – počet nenulových prvkov kódového slova

Hammingova vzdialenosť: d – počet miest v ktorých sa 2 kódové slová líšia

platí $d(u,v)=w(u+v)$ - vzdialenosť dvoch vektorov u,v sa = váhe súčtu týchto vektorov

LBK (n,k,d_{min}) je definovaný generujúcou maticou $G_{k\,x\,n}$. Riadky matice G sú bázovými vektormi kódu.

Kódovanie: $c=i*G$ pre riadkové vektory i a c

$c=i*G^T$ pre stĺpcové vektory i a c

Ortogonalita - $G*H^T=0$; 0 reprezentuje nulovú maticu ==> pre každé slovo $c*H^T=\bar{0}$

Systematický tvar **LBK** – Ak $G=I_k:P$ ==> $H=P^T\,|\,I_{n-k}$

Detekčné a opravné schopnosti kódu: $t_{kor}=|\frac{(d_{min}-1)}{2}|$ $t_{det}=d_{min}-1$

tabuľka syndrómov

e (chybový vektor, všetky možné kombinácie)	$s=e*H^T$
e_1 - 1000.....0	
e_n - 0000.....1	

Dekódovanie: $s=v*H^T$, podľa výsledného s, odčítame z tabuľky odhad chyby e

$\hat{c}=\hat{e}+v$, ak je kód systematický, odhad informačného slova i je priamo subvektorom c

počet riadkov syndrómovej tabuľky $r=\sum_{\tau=1}^{t_{opr}}(\frac{n}{\tau})=\sum_{\tau=1}^{t_{opr}}(\frac{n!}{(n-\tau)!\,\tau!})$

Cyklické Kódy: $CK \subset LBK$, ich základom je generujúci polynóm g(x)

cykličnosť kódu – máme kódové slovo c, ak toto kódové slovo patrí do kódu c, potom aj všetky jeho cyklické posuny patria do kódu.

Kódovanie:

v nesystematickom tvare: $c(x)=i(x)*g(x)$

v systematickom tvare: 1. $i(x)*x^{(\deg g(x))}=i(x)*x^{(n-k)}$

2. $Q(x)=i(x)*x^{(\deg g(x))} mod\ g(x)$

3. $c(x)=i(x)*x^{(\deg g(x))}+Q(x)$

Dekódovanie:

v systematickom tvare: 1. $s(x)=v(x) mog\, g(x)$

2. e(x) dostaneme zo syndrómovej tabuľky

3. $c(x)=v(x)+e(x)$

4. i(x) = prvých k členov c(x)

v nesystematickom: 1. ; 2. ; 3. krok rovnako ako v systematickom

4. $i(x)=c(x):g(x)$

HW realizácia CK a konečné polia:

GF(p) ==> $x+y=(x+y) mod\,(p)$ $x\circ y=(x\circ y) mod\,(p)$

Vektorové polia:

$\alpha^i*\alpha^j=\alpha^{((i+j) mod (q-1))}$

2 primitívne polynómy 3.stupňa $p(x)=\begin{matrix}x^3+x+1\\x^3+x^2+1\end{matrix}$

Najvyššia $\alpha=\alpha^{(q-2)}$

Reed Solomon: $LBK \subset (CK \subset RS)$

RS kódy GF(2^r) v základnom tvare sú viazané na pole, v ktorom sú zostrojené.

Postup pred kódovaním: 1. vypočítam n,k,d_{min} $n = q - 1$ $d_{min} = n - k + 1$ $t = |\frac{n-k}{2}| = |\frac{d_{min}-1}{2}|$

$GF(2^r) = GF(q) \rightarrow r = deg\ p(x)$

2. z p(x) určíme prvky poľa (α)

3. g(x) má 2t za sebou idúcich koreňov

Kódovanie: najprv g(x) v konjunktívnom tvare $g(x) = (x - \alpha^j) * (x - \alpha^{j+1}) ... (x - \alpha^{j+(2t-1)})$

potom g(x) v disjunktívnom tvare – roznásobime koreňové činitele

i(x) rozdelíme si to na r-tice (deg p(x))

napíšeme si i(x) pomocou prvkov poľa (α)

v nesystematickom tvare: $c(x) = i(x) * g(x)$

v systematickom tvare: $i(x) * x^{n-k} + [(i(x) * x^{n-k}) mod\ g(x)]$

c(x) môže byť zapísané vo vektorovom tvare, binárnom tvare aj v tvare binárnej matice

Dekódovanie:

1.) Vypočítame si syndrómy, počet syndrómov = počtu koreňov g(x) = 2t

$\bar{s_k} = v(\alpha^k)$ k je 0 po 2t-1

2.) Určíme polynóm lokátorov a vyriešime ho

$\sigma(x) = x^t + \sigma_1 * x^{t-1} + ... + \sigma_{t-1} * x^1 + \sigma_t$ pre t=2 $s_2 + \sigma_1 * s_1 + \sigma_2 * s_0 = 0$

$s3 + \sigma_1 * s_2 + \sigma_2 * s_1 = 0$

3.) Určíme lokátorov chýb

pre t=2 polynóm lokátorov $x^2 + \sigma_1 * x^1 + \sigma_2 * x^2 = \sigma_{(x)}$

za σ(x) si dosadíme všetky prvky poľa ($\alpha^0 .... \alpha^{q-2}$) prvky dosádzame vzostupne!!!!

ak $\sigma(\alpha^k) = 0$ ==> chyba nastala pri x^k

4.) Výpočet hodnoty chyby

$\bar{S_k} = \sum_{(i=1)}^{(t)} Y_i * X_i^k$ k je od 0 po 2t-1, X_i je umiestnenie i-tej chyby z predchádzajúceho kroku

pre t=2

$\bar{s_0} = Y_1 * X_1^0 + Y_2 * X_2^0$ Vyriešime, Y_i je hodnota i-tej chyby

$\bar{s_1} = Y_1 * X_1^1 + Y_2 * X_2^1$

$e(x) = Y_1 * X_1 + Y_2 * X_2$

5.) $\hat{c}(x) = v(x) + \hat{e}(x)$

i (x) = prvých k členov c(x), znížime stupeň o n-k (2t)

Charakteristika a Kódovanie Zdroja:

Množstvo informácie, ktoré správy obsahujú $I_{(mi)} = -\log_2(pi)[bit]$

Entropia (stredná hodnota množstva, informácie)

$$H(M) = E[I(M)] = \sum_i pi * I(mi) = -\sum pi * \log_2(pi) \qquad [\frac{bit}{symbol}]$$

Kódovanie zdroja – úlohou je kódovať čo najefektívnejšie – prefixné kódy

- Shanon Fanov kód
- Hoffman

Shanon fanov kód:

algoritmus: 1.) správy sa zoradia podľa pravdepodobnosti zostupne

2.) správy rozdelíme do 2 skupín tak, aby súčet pravdepodobností v týchto skupinách bol čo najnižší (rozdiel súčtu početnosti).

2a.) dodatočné kritéria – v hornej polovici je väčší súčet

3.) pre 1. skupinu priradíme 0, a pre 2. skupinu priradíme 1

4.) rekurzívne opakujeme, až pokiaľ nedostaneme skupinu o 1 správe

Hoffmanov kód:

algoritmus: 1.) správy sa zoradia podľa pravdepodobnosti vzostupne a tvoria listové uzly stromu

2.) postupne spájame 2 uzly s najmenšou pravdepodobnosťou do jedného uzla:

a.) jeho pravdepodobnosť bude = súčtu pravdepodobnosti zlúčených

b.) ak mám viac možností, postupujem podľa dodatočných kritérií

c.) minimálna variácia (rozptyl) dĺžky kód slova

3.) opakujem, kým neostane jediný uzol ==> koreň stromu

Strom má n listov – počet prvkov abecedy

n-1 - počet medziľahlých listov

zložitosť závisí od implementácie $O(n * \log n)$ *alebo* $O(n)$

Stredná dĺžka kódového slova: $\bar{d} = E(d) = \sum_i^n p_i * d_i$ ⟸ dĺžka kódového slova pre správu i

$$\sum_i p_i = 1$$

dolné ohraničenie dĺžky kód. slova $d_{min} \geq \frac{H(M)}{\log_2(q)}$ q-počet prvkov abecedy, ktorou kódujeme

Univerzálne kódovanie – Ziv Lempelove kódy: sú to kompakčné kódy

LZ77 – sliding window

LZ78 – dictionary

LZ77 – výstup (pointer, dĺžka zhody, inovačný symbol – prvý v ktorom sa nezhoduje)

kódovanie: 1.) inicializuj buffer (naplň) ľubovoľným znakom, ktorý sa vyskytuje v postupnosti

2.) načítaj do look-ahead bufferu nezakódovaný reťazec, ktorý je maximálne možný

3.) hľadaj najdlhšiu zhodu načítaného reťazca v look-ahead zásobníku

4.) napíš adresu zač. Reťazca v zásobníku, dĺžku ktorá sa zhoduje, pripíš inovačný symbol, ak sa reťazec v slovníku nenachádza, napíš 0-vý vektor.

5.) posuň kópiu zakódovaného reťazca z look-ahead bufferu do zásobníka

6.) načítaj nezakódovaný reťazec rovnakej dĺžky do look-ahead buffera

7.) opakuj od kroku 3

dekódovanie: 0.) zakódovanú postupnosť rozdelím na n-tice

1.) inicializuj zásobník (nulami)

2.) prečítaj zakódovanú adresu, dĺžka a inovačný symbol – skopíruj reťazec danej dĺžky od udanej adresy a pripíš inovačný symbol

3.) posuň kópiu výstupu do zásobníka

4.) opakuj od kroku 2

LZ78 – slovníková metóda (index v slovníku, inovačný symbol ktorý je zhodný)

kódovanie: 1.) Inicializuj slovník (0=prázdny riadok, slovník začína indexom 1)

2.) nájdi najdlhší reťazec W v slovníku zhodný s aktuálnym vstupným reťazcom

3.) na výstup daj index slovníka reťazca W a odstráň W zo vstupu. Ak sa W nenachádza v slovníku, index = 0

4.) pridaj W spolu s nasledujúcim inovačným symbolom do slovníka

5.) opakuj od kroku 2