

Algebraické dekódování Reed Solomonových kódů

Algebraic decoding of Reed Solomon codes

Jiří Šicner

xsicne02@feec.vutbr.cz

Fakulta elektrotechniky a komunikačních technologií VUT v Brně.

Abstrakt: Tento dokument popisuje jednotlivé metody algebraického dekódování Reed Solomonových kódů. Jsou zde znázorněny postupy pro získání lokalizačního mnohočlenu pomocí vybraných metod dekódování, každá metoda je doplněna stručným příkladem pro lepší vysvětlení.

Abstract: This article describes the various methods of algebraic decoding of Reed Solomon codes. There are shown the procedures for obtaining localization of selected methods using polynomial decoding, each method is accompanied by a brief example for a better explanation.

Algebraické dekódování Reed Solomonových kódů

Jiří Šicner

Fakulta elektrotechniky a komunikačních technologií VUT v Brně
Email: xsicne02@feec.vutbr.cz

Abstrakt – Tento dokument popisuje jednotlivé metody algebraického dekódování Reed Solomonových kódů. Jsou zde znázorněny postupy pro získání lokalizačního mnohočlenu pomocí vybraných metod dekódování, každá metoda je doplněna stručným příkladem pro lepší vysvětlení.

1 Úvod

Při přenášení dat v komunikačních systémech díky potřebě neustále zvyšovat přenosové rychlosti se stále častěji setkáváme se vznikem chyb, které mají tendenci se shlukovat. Jedním ze způsobů ochrany jsou Reed Solomonovy kódy (dále jen RS kódy), ty jsou schopny přenos dostatečně zabezpečit. Byly publikovány v roce 1960 svými vynálezci Irvingem Reedem a Gusem Solomonem v tehdejší časopisu o vědě Journal of the Society for Industrial and Applied Mathematics, od svých vynálezců také přebrali svůj název.

2 Základní popis RS kódů a jejich kódování

RS kódy patří do kategorie BCH (Bose-Ray-Chaudhuri) kódů a jedná se tedy o korekční, cyklické, lineární kódy se symboly, které jsou tvořeny m -bitovými posloupnostmi, kde m je libovolné celé kladné číslo s hodnotou větší než 2. Kód bývá běžně zadán parametry (n, k) . Parametr k určuje, kolik symbolů vstupuje do kodéru a parametr n udává počet symbolů vystupujících z kodéru. Pro n a k pak podle [2] platí:

$$0 < k < n < 2^m + 2 \quad (1)$$

Pro nejběžnější RS(n, k) kód pak platí [2]:

$$(n, k) = (2^m - 1, 2^m - 1 - 2t), \quad (2)$$

kde t je označení pro korekční schopnost kódu a $n - k = 2t$ je počet symbolů pro zabezpečení.

Pro nebinární kódy je vzdálenost mezi dvěma kódovými slovy definována (analogově s Hammingovou vzdáleností) jako počet symbolů, ve kterých se posloupnost liší. Pro RS kódy je minimální kódová vzdálenost dána vztahem podle [2]:

$$d_{min} = n - k + 1 \quad (3)$$

Kód je schopen opravit jakoukoliv kombinaci t nebo méně chyb, kde t lze vyjádřit jako [2]:

$$t = \frac{d_{min} - 1}{2} = \frac{n - k}{2} \quad (4)$$

Než přejdeme k samotnému procesu kódování, je potřeba ještě definovat vytvářecí polynom. Ten je podle [2] definován jako:

$$g(X) = (X - \alpha)(X - \alpha^2) \dots (X - \alpha^{2t}) = \prod_{j=1}^{2t} (X - \alpha^j) = \sum_{j=0}^{2t} g_j X^j, \quad (5)$$

kde $\alpha^1, \alpha^2, \dots, \alpha^{2t}$ jsou prvky Galoisova tělesa.

Protože RS kódy mohou být systematické i nesystematické, používají se pro kódování dva přístupy. Nesystematické cyklické kodéry generují kódové slovo vynásobením vstupního polynomu $i(X)$ vytvářecím polynomem $g(X)$. Výstupní kódové slovo je tedy dáno [2]:

$$c(X) = i(X) \cdot g(X) \quad (6)$$

Kde koeficienty polynomů jsou prvky Galoisova pole $GF(2^m)$.

Pro systematické RS kódy pak platí vztah [2]:

$$c(X) = X^{n-k} \cdot i(X) + p(X) \quad (7)$$

podle výše uvedené definice máme:

$$\text{Rem} \left\{ \frac{c(X)}{g(X)} \right\} = 0 \quad (8)$$

$$\text{Rem} \left\{ \frac{X^{(n-k)} \cdot i(X) + p(X)}{g(X)} \right\} = 0 \quad (9)$$

$$\text{Rem} \left\{ \frac{X^{(n-k)} \cdot i(X)}{g(X)} \right\} + \text{Rem} \left\{ \frac{p(X)}{g(X)} \right\} = 0 \quad (10)$$

Jestliže posloupnost paritního polynomu $p(X)$ je menší než $n - k$ a posloupnost $g(X)$ je $n - k$, pak můžeme podle [2] psát:

$$\text{Rem} \left\{ \frac{p(X)}{g(X)} \right\} = p(X) \quad (11)$$

a dosazením této rovnice do rovnice 10 a upravením dostaneme:

$$-\text{Rem} \left\{ \frac{X^{(n-k)} \cdot i(X)}{g(X)} \right\} = p(X) \quad (12)$$

Dosažením rovnice 12 do 7 pak dostaneme konečný vztah pro systematické kódování:

$$c(X) = X^{n-k} \cdot i(X) + \text{Rem} \left\{ \frac{X^{(n-k)} \cdot i(X)}{g(X)} \right\}. \quad (13)$$

3 Obecný postup algebraického dekódování RS kódů

Pro dekódování RS kódů bylo vyvinuto mnoho algoritmů. V této kapitole se je přibližuje obecný postup algebraického dekódování RS kódů. Konkrétní metody, které budou popsány níže se pak liší víceméně pouze ve stanovení lokalizačního mnohočlenu.

Obecně lze podle [3] algebraické dekódování RS kódů rozdělit do následujících kroků:

1. Výpočet syndromů (určení zda nastala či nenastala chyba).
2. Stanovení chybového polynomu (lokátoru chyb), jehož kořeny nám určují, kde se nachází chyby v přenášeném kódovém slovu. Tento krok lze řešit více způsoby, existuje několik odlišných algoritmů pro nalezení chybového polynomu, např. Peterson-Gorenstein-Zierleův algoritmus, Berlekamp-Masseyův algoritmus, Euklidův algoritmus atd.
3. Nalezení kořenů chybového polynomu. To se obvykle provádí pomocí Chienova vyhledávání.
4. Stanovení hodnoty chyby. To je obvykle dosaženo použitím Forneyova algoritmu.
5. Oprava chyby.

3.1 Výpočet syndromů

Prvním krokem dekódování je výpočet syndromových rovnic. Ty nám indikují zda nastala, či nenastala při přenosu chyba. V případě, že všechny syndromové rovnice vyjdou nulové, při přenosu nenastala žádná chyba a další kroky dekódování vůbec neproběhnou. Pro výpočet jednotlivých syndromových rovnic se používá vztah:

$$s_j = r(\alpha^j) = \sum_{k=0}^{n-1} e_k \alpha^{jk}, \quad j = 1, 2, \dots, 2t. \quad (14)$$

3.2 Lokalizační mnohočlen

V tomto kroku se liší jednotlivé přístupy k dekódování RS kódů. V dalším textu jsou popsány jednotlivé metody a s nimi i jednotlivé algoritmy pro nalezení lokalizačního mnohočlenu.

Obecně je pak lokalizační mnohočlen podle [3] definován jako:

$$L(X) = \prod_{l=1}^v (1 - X_l x) = L_v X^v + L_{v-1} X^{v-1} + \dots + L_1 X + L_0, \quad (15)$$

kde $L_0 = 1$. Ostatní koeficienty L_v je nutné dopočítat některým z níže uvedených algoritmů.

3.3 Nalezení pozic chyb - Chienovo vyhledávání

Předpokládejme, že v tuto chvíli máme stanovený lokalizační mnohočlen. Dalším krokem je nalezení kořenů tohoto mnohočlenu. To spočívá v tom, že se postupně prověřují všechny prvky z Galoisova tělesa, jestli nejsou kořenem mnohočlenu. Existují i odlišné způsoby, ale pro tělesa používané pro korekční kódy a pro počet hledaných kořenů se jeví Chienovo vyhledávání jako nejvíce účinné [3].

Vezměme příklad, kdy máme $v = 3$ a lokalizační mnohočlen je tedy podle vztahu 15 ve tvaru:

$$L(X) = L_0 + L_1 X + L_2 X^2 + L_3 X^3 = 1 + L_1 X + L_2 X^2 + L_3 X^3$$

Vypočítáme $L(X)$ pro každý nenulový prvek Galoisova tělesa: $X = 1, X = \alpha, X = \alpha^2, \dots, X = \alpha^{q^m-2}$. To nám dává následující:

$$\begin{aligned} L(1) &= 1 + L_1(1) + L_2(1)^2 + L_3(1)^3 \\ L(\alpha) &= 1 + L_1(\alpha) + L_2(\alpha)^2 + L_3(\alpha)^3 \\ L(\alpha^2) &= 1 + L_1(\alpha^2) + L_2(\alpha^2)^2 + L_3(\alpha^2)^3 \\ &\vdots \\ L(\alpha^{q^m-2}) &= 1 + L_1(\alpha^{q^m-2}) + L_2(\alpha^{q^m-2})^2 + L_3(\alpha^{q^m-2})^3 \end{aligned}$$

V případě, že $L(X)$ vyjde nulové pro daný prvek GF, je indikována pozice chyby. Jestliže nejsou nalezeny žádné kořeny, znamená to neodstranitelné chyby.

3.4 Výpočet hodnoty chyb - Forneyův algoritmus

Po získání lokátoru chyb a získání jeho kořenů nám zbývá vyčíst hodnotu chyb. Pro toto byl vyvinut Forneyův algoritmus, pro jeho odvození se musíme vrátit k výpočtu syndromů.

V případě, že máme určené syndromy S_1, S_2, \dots, S_{2t} můžeme určit syndromový polynom $S(X)$ následovně:

$$S(X) = S_1 + S_2(X) + \dots + S_{2t} X^{2t-1} = \sum_{j=0}^{2t-1} S_{j+1} X^j. \quad (16)$$

Následně můžeme podle [3] definovat polynom pro určení hodnoty chyb $E(X)$:

$$E(X) = S(X) \cdot L(X) \pmod{X^{2t}} \quad (17)$$

Tato rovnice je nazývána klíčová rovnice (ang. key equation). Operace $\text{mod}X^{2t}$ zajišťuje zrušení všech členů polynomu, které jsou stupně $2t$ nebo vyšší. Protože již máme určený polynom pro určení chyb, stačí vypočítat derivaci $L(X)$ a můžeme podle [2] a [3] psát vztah pro hodnotu chyby M_l , $l \in 1, \dots, v$ která je obecně ve tvaru:

$$M_l = -\frac{E(P_l^{-1})}{L'(P_l^{-1})} \quad (18)$$

kde $L'(X)$ je derivace lokalizačního mnohočlenu.

3.5 Oprava chyby

V poslední fázi jsou již známy pozice chyb a jejich hodnoty v přenášené zprávě. Na základě těchto hodnot je sestaven chybový polynom $e(X)$. Tento polynom je přičítán k polynomu, který byl přijatý v dekodéru $r(X)$, jejich sečtením pak dostáváme opravenou vstupní zprávu $i(X)$. Toto vyjadřuje následující rovnice:

$$i(X) = r(X) + e(X). \quad (19)$$

4 Algoritmy pro stanovení lokalizačního mnohočlenu

4.1 Berlekamp Masseyův algoritmus

V této kapitole bude popsán Berlekamp - Masseyův algoritmus (dále jen BM algoritmus) pro stanovení chybového polynomu. Tento algoritmus je založen na postupné iteraci.

BM algoritmus můžeme formálně shrnout Blahutovou interpretací[2]. Jednotlivé fáze algoritmu jsou znázorněny v diagramu a tyto kroky jsou popsány níže:

1. Nastavení počátečních podmínek:

index iterace: $i = 0$

délka LSFR: $l^{(1)} = 0$

chybový mnohočlen: $L^{(1)}(X) = 0$

pomocný chybový mnohočlen: $A^{(1)}(X) = 0$

2. Kontrola, zda jsme dosáhli konce iterace, to je jestliže $i = 2t$.
3. Odhad chyby, která náleží k dalšímu vygenerovanému syndromu:

$$d^i = \sum_{n=0}^{l^i} L_n^{(i)} \cdot s_{i+1-n} \quad (20)$$

4. Kontrola, zda je odchylka $d^i = 0$ a v případě, že ano, pak aktuální LSFR o délce l^i a chybový mnohočlen $L^i(X)$ produkuje následující syndrom s_{i+1} . Proto přejděte ke kroku 5, v opačném případě přejděte ke kroku 6.

5. Jednoduše posuň pomocný LSFR o jednu pozici pomocí následujícího vztahu:

$$A^{(i)}(X) := X.A^{(i)}(X), \quad (21)$$

dále pokračuj krokem 12.

6. Jestliže $d^{(i)} \neq 0$, opravíme dočasný chybový mnohočlen $T^{(i)}(X)$. Toto vyjadřuje následující vztah:

$$T^{(i)}(X) = L^{(i)}(X) - d^{(i)} - d^{(i)} \cdot X.A^{(i)}(X). \quad (22)$$

7. Ověření, jestli je LSFR třeba zvětšit, nebo ne. V případě že ne, pokračujeme krokem 8, jinak krokem 9.
8. Obnovení chybového mnohočlenu pomocí následujícího vztahu:

$$L^{(i)}(X) := T^{(i)}(X) \quad (23)$$

vracíme se do kroku 5.

9. Normalizujeme poslední chybový mnohočlen $L(X)$ dělením $d^{(i)} \neq 0$ a výsledek se uloží do pomocného LSFR $A^{(i)}(X)$:

$$A^{(i)}(X) := \frac{L^{(i)}(X)}{d^{(i)}} \quad (24)$$

pokračujeme krokem 10.

10. Nyní můžeme přepsat $L^{(i)}(X)$, protože byl v předchozím kroku normalizován a uložen do $A^{(i)}(X)$. $L^{(i)}(X)$ je aktualizován následovně:

$$L^{(i)}(X) := T^{(i)}(X) \quad (25)$$

pokračujeme krokem 11.

11. Podle kroku 7 musí být prodloužen LSFR. To provedeme pomocí následujícího vztahu:

$$l^{i+1} = i + 1 - l^i. \quad (26)$$

pokračujeme krokem 12.

12. Inkrementujeme index iterace i a vrátíme se do kroku 2.

Pro podrobnější studium tohoto algoritmu bych doporučil literaturu [2] a [3].

4.2 Euklidův algoritmus

V této kapitole si ukážeme použití Euklidova algoritmu pro konstrukci chybového polynomu. Tento přístup k dekódování se často nazývá Sugiyama algoritmus.

Vyjdeme ze základní rovnice podle [3]:

$$E(X) = S(X)L(X) \pmod{X^{2t}} \quad (27)$$

Známe pouze $S(X)$ a t a chceme určit chybový polynom $L(X)$ a polynom pro určení velikosti chyb $E(X)$. Tento problém se jeví jako neřešitelný. Nicméně rovnici 27 můžeme podle [3] přepsat na:

$$\Theta(X)(X^{2t}) + L(X)S(X) = E(X) \quad (28)$$

pro libovolný polynom $\Theta(X)$. Uvedme, že podle [3] rozšířený Euklidův algoritmus vrací pro dvojici prvků (a, b) , dvojici prvků (s, t) takových, že:

$$as + bt = c \quad (29)$$

kde c je největší společný dělitel z a a b . Z této úvahy vychází Euklidova metoda a celý proces lze pak podle [1] shrnout do následujících kroků:

1. Spočítání syndromů a syndromového polynomu $S(X) = s_1 + s_2X + \dots + s_{2t}X^{2t-1}$.
2. Jestliže $S(X) = 0$, pak je odpovídající přijatý vektor považován za kódový vektor.
3. Jestliže $S(X) \neq 0$, pak je algoritmus inicializován jako:

$$\begin{aligned} r_{-1}(X) &= X^{2t} \\ r_0 &= S(X) \\ t_{-1}(X) &= 0 \\ t_0(X) &= 1 \\ i &= -1 \end{aligned} \quad (30)$$

4. Rekurzivní parametry jsou stanoveny následovně:

$$r_i(X) = r_{i-2}(X) - q(X)r_{i-1}(X) \quad (31)$$

$$t_i(X) = t_{i-2}(X) - q(X)t_{i-1}(X) \quad (32)$$

tato rekurze probíhá tak dlouho dokud je $(r_i(X)) \geq t_i$.

5. Když $(r_i(X)) < t_i$, rekurze skončí a můžeme určit:

$$E(X) = r_i(X) \quad (33)$$

$$L(X) = t_i(X) \quad (34)$$

Pro podrobnější studium Euklidova algoritmu bych doporučil literaturu [3] a [1].

4.3 Peterson-Gorenstein-Zierleův algoritmus

Peterson-Gorenstein-Zierleova (dále jen PGZ) metoda dekódování obsahuje inverzi dvou matic. Jednu pro výpočet chybového polynomu a druhou pro určení hodnot chyb.

Následující výpočty vycházejí ze vztahu podle [2]:

$$\mathbf{L} = \mathbf{S}^{-1} \cdot \mathbf{S}. \quad (35)$$

Řešení je založeno na následující teorému. Vandermondova matice \mathbf{S} tvořená syndromy není singulární a může být invertována, pokud je její dimenze $v \times v$, pokud je singulární, tak nemůže být invertována protože dimenze je větší než v , kde v je aktuální počet chyb, který nastal.

Abychom mohli vyřešit rovnici 35, musíme určit v pro schopnost invertace \mathbf{S} . Zpočátku nastavíme $v = t$, protože t je maximální možný počet chyb, a počítáme determinant \mathbf{S} . Pokud je $\det(\mathbf{S}) = 0$, při $v = t$ pak musíme dekrementovat v o jedna, postupně až na nulu, dokud nenajdeme v , pro které bude $\det(\mathbf{S}) \neq 0$. Jakmile je nalezeno správné v , spočítáme \mathbf{S}^{-1} a určíme $L = \mathbf{S}^{-1}\mathbf{S}$.

Následuje využití Chienova vyhledávání. Jsou nalezeny pozice chyb a zbývá pouze vyčíslit hodnotu chyby. To lze provést výše zmíněným Forneynovým algoritmem, nicméně v době vyvinutí PGZ metody tento algoritmus nebyl znám, a tak se tento problém řešil opět pomocí invertování matice.

Je to poměrně jednoduché, vyjdeme ze vztahu $\mathbf{M} = \mathbf{P}^{-1}\mathbf{S}$ podle [2]. Z Chienova vyhledávání již známou matici pozic chyb \mathbf{P} invertujeme pro spočítání vektoru velikosti chyb \mathbf{M} :

$$\mathbf{M} = \mathbf{P}^{-1}\mathbf{S}. \quad (36)$$

5 Příklad na dekódování

Pro upřesnění výkladu jednotlivých algoritmů uvedeme příklad. Máme přijatou zprávu $r = (000\ 111\ 000\ 101\ 000\ 000\ 000)$. Posloupnost byla zakódována kódem RS (7, 3), který je definován polem $\text{GF}(2^3)$ nad základním polynomem $p_i(X) = 1 + X^2 + X^3$.

Nejprve je potřeba přepsat přijatou zprávu do polynomiálního tvaru, který je nutný pro výpočet syndromů:

$$r(X) = \alpha^4 X + \alpha^3 X^3$$

Následuje samotný výpočet syndromů:

$$s_1(\alpha) = r(\alpha) = \alpha^5 + \alpha^6 = \alpha^3$$

$$s_2(\alpha) = r(\alpha^2) = \alpha^6 + \alpha^2 = \alpha$$

$$s_3(\alpha) = r(\alpha^3) = 1 + \alpha^5 = \alpha$$

$$s_4(\alpha) = r(\alpha^4) = \alpha + \alpha = 0$$

Nyní potřebujeme stanovit lokalizační mnohočlen, to zde pro názornost provedeme všemi výše uvedenými algoritmy.

i	$r_i = r_{i-2} - q_i r_{i-1}$	q_i	$t_i = t_{i-2} - q_i t_{i-1}$
-1	$X^{n-k} = X^4$		0
0	$S(X) = \alpha^3 + \alpha X + \alpha X^2$		1
1	$X + \alpha^5$	$\alpha^2 + \alpha^6 X + \alpha^6 X^2$	$\alpha^2 + \alpha^6 X + \alpha^6 X^2$

Tabulka 1: Znázornění jednotlivých iterací Euklidova algoritmu.

5.1 Stanovení lokalizačního algoritmu pomocí BM algoritmu

i	$L(X)$	$A(X)$	l
0	1	1	0
1	$1 + \alpha^3 X$	α^4	1
2	$1 + \alpha^5 X$	$\alpha^4 X$	1
3	$1 + \alpha^5 X + \alpha^6 X^2$	$\alpha^5 + \alpha^3 X$	2
4	$1 + \alpha^4 X + \alpha^4 X^2$		

Tabulka 2: Znázornění jednotlivých iterací BM algoritmu.

Iterace končí v našem případě když $i = 2t = 4$ a získáváme lokalizační mnohočlen $L(X) = 1 + \alpha^4 X + \alpha^4 X^2$

5.2 Stanovení lokalizačního algoritmu pomocí Euklidova algoritmu

Jednotlivé iterace algoritmu jsou znázorněny v tabulce 1. Jestliže je stupeň polynomu ve sloupci r_i menší než stupeň polynomu ve sloupci t_i , algoritmus se ukončí. Potom můžeme psát:

$$L_1(X) = \alpha^6 X^2 + \alpha^6 X + \alpha^2$$

Polynom $L_1(X)$ takto získaný, je násobený prvkem Galoisova pole $\lambda \in GF(2^3)$ za cílem převést jej do normovaného polynomu. Tato hodnota λ je v našem příkladě $\lambda = \alpha$. Potom:

$$L(X) = X^2 + X + \alpha^3$$

Vidíme, že jsme získali lokalizační polynom stejný jako BM algoritmem.

5.3 Stanovení lokalizačního algoritmu pomocí PGZ algoritmu

Vzhledem k tomu, že nemáme žádné informace o skutečném počtu chyb v přijímači, musíme nejprve očekávaný počet chyb ν určit. Zpočátku nastavíme $\nu = t = 2$. Otestujeme, zda je determinant matice $\nu \times \nu$ různý od nuly:

$$\det(\mathbf{S}) = \det \begin{vmatrix} s_1 & s_2 \\ s_2 & s_3 \end{vmatrix} = (s_1 s_3 - s_2^2)$$

Dosažením za jednotlivé syndromy dostaneme:

$$\det(\mathbf{S}) = (\alpha^3 \alpha - \alpha^2) = \alpha^4 - \alpha^2 = \alpha^5 \neq 0$$

Protože je $\det(\mathbf{S}) \neq 0$ můžeme vypočítat \mathbf{S}^{-1} :

$$\mathbf{S}^{-1} = \begin{bmatrix} \alpha^3 & \alpha^3 \\ \alpha^3 & \alpha^5 \end{bmatrix}$$

Nyní už můžeme vypočítat hledaný lokalizační mnohočlen:

$$\begin{bmatrix} L_1 \\ L_2 \end{bmatrix} = \begin{bmatrix} \alpha^3 & \alpha^3 \\ \alpha^3 & \alpha^5 \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha^4 \\ \alpha^4 \end{bmatrix}$$

Lokalizační mnohočlen je tedy $L(X) = \alpha^4 X^2 + \alpha^4 X + 1$, převedením do normovaného polynomu opět dostaneme stejný výsledek jako u BM a Euklidova algoritmu.

Po stanovení lokalizačního mnohočlenu je na řadě Chienovo vyhledávání pro nalezení pozic chyb:

$$\begin{aligned} L(\alpha^0) &= \alpha^0 + \alpha^0 + \alpha^3 = \alpha^3 \\ L(\alpha) &= \alpha^2 + \alpha + \alpha^3 = \alpha^5 \\ L(\alpha^2) &= \alpha^4 + \alpha^2 + \alpha^3 = \alpha^6 \\ L(\alpha^3) &= \alpha^6 + \alpha^3 + \alpha^3 = \alpha^6 \\ L(\alpha^4) &= \alpha + \alpha^4 + \alpha^3 = 0 \leftarrow \\ L(\alpha^5) &= \alpha^3 + \alpha^5 + \alpha^3 = \alpha^5 \\ L(\alpha^6) &= \alpha^5 + \alpha^6 + \alpha^3 = 0 \leftarrow \end{aligned}$$

Nalezené pozice je třeba invertovat:

$$\begin{aligned} (\alpha^4)^{-1} &= \frac{\alpha^0}{\alpha^4} = \alpha^3 = P_1 \\ (\alpha^6)^{-1} &= \frac{\alpha^0}{\alpha^6} = \alpha^1 = P_2 \end{aligned}$$

Nyní již můžeme vypočítat hodnotu hledaných chyb. Pro to potřebujeme derivaci lokalizačního mnohočlenu, což je $L'(X) = 1$ a polynom pro odhad chyby $E(X)$.

$$\begin{aligned} E(X) &= L(X) \cdot S(X) \pmod{X^4} \\ E(X) &= (1 + \alpha^4 X + \alpha^4 X^2) \cdot (\alpha^3 X + \alpha X^2 + \alpha X^3) \pmod{X^4} \\ E(X) &= \alpha^5 X^2 + \alpha^3 X \end{aligned}$$

$$\begin{aligned} M_1 &= \frac{E(P_1^{-1})}{L'(P_1^{-1})} = \frac{E(\alpha^4)}{L'(\alpha^4)} = \frac{\alpha^3}{1} = \alpha^3 \\ M_2 &= \frac{E(P_2^{-1})}{L'(P_2^{-1})} = \frac{E(\alpha^6)}{L'(\alpha^6)} = \frac{\alpha^4}{1} = \alpha^4 \end{aligned}$$

Tím dostáváme potřebné hodnoty pro sestavení chybového polynomu:

$$\begin{aligned}(P_1, M_1) &= (\alpha^3, \alpha^3) \\ (P_2, M_2) &= (\alpha^1, \alpha^4)\end{aligned}$$

Chybový polynom má pak tvar:

$$e(X) = \alpha^4 X + \alpha^3 X^3$$

Přičtením chybovému vektoru $e(X)$ k přijatému polynomu $r(X)$ získáme nulový vektor, což je hledaná vstupní zpráva kodéru.

6 Závěr

V tomto dokumentu bylo popsáno algebraické dekodování Reed Solomonových kódů, byly zde představeny jednotlivé přístupy ke stanovení lokalizačního mnohočlenu. Pro podrobnější studium těchto metod bych doporučil především literaturu [1] a [3]. Díky dobrým vlastnostem RS kódů jsou v dnešní době hojně používány v mnoha digitálních zařízeních, v poslední době je například zabezpečení těmito kódy použito v zabezpečení FEC u standardů pro pasivní optické sítě (EPON, GPON, 10GEPON, XG-PON), kde se používá pro dekodování částečně modifikované Euklidovy metody.

Poděkování

Článek vznikl za podpory grantového projektu GAČR č. 102/09/1846 "Vícetónová modulace realizovaná překryvnou bankou filtrů" a výzkumného záměru MSM002163513 "Elektronické komunikační systémy a technologie nových generací".

Literatura

- [1] FARELL, P.G., MOREIRA, J.C. Essentials of Error-Control Coding. John Wiley, 2006, ISBN-13 978-0-470-02920-6.
- [2] HANZO, L., LIEW, T.H., YEAP, B.L. Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels. JohnWiley, 2002, ISBN: 0470847263.
- [3] MOON, T.K. Error Correction Coding: Mathematical Methods and Algorithms. Wiley-Interscience, 2005, ISBN-13: 978-0070010697.
- [4] MORELOS, R.H. The Art of Error Correcting Coding Wiley-Interscience, 2002, ISBN: 0471 49581 6.
- [5] NĚMEC, K. Datová komunikace. Skripta. VUT FEKT, Brno 2007.

- [6] PURSER, M. Introduction to Error-correcting codes. Boston, London: Artec House, 1995. ISBN 0-89006-784-8.
- [7] SKALAR, B. Digital Communications, Fundamentals and applications Prentice-Hall, 2003, ISBN 0-13-084788-7.
- [8] ŠICNER, Jiří Srovnání algoritmů dekodování Reed-Solomonova kódu: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2011.