

METODE UNTUK MENGKOREKSI KESALAHAN (ERROR) DENGAN MENGGUNAKAN MATRIKS JARANG

Daud Andreaw / 0222198

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Kristen Maranatha

Jln. Prof. Drg. Suria Sumantri 65, Bandung 40164, Indonesia

Email : gallagher_fetch22@yahoo.co.id

ABSTRAK

Terkadang, pada saat pengirim menyampaikan informasi (data) ke penerima pada proses komunikasi terdapat beberapa kesalahan-kesalahan yang terjadi. Kesalahan (*error*) tersebut dapat terjadi karena *noise*, *interferensi*, *jamming*, atau adanya gangguan sinyal lain. Ada dua strategi dasar untuk mengetahui terjadinya *error*, yaitu dengan cara mendeteksi *error* dan mengkoreksi *error*.

Metode yang digunakan untuk merancang suatu pengkoreksi *error* yang baik yaitu menggunakan algoritma *Bit Flip* (BF) dengan matriks jarang (*sparse matrix*) pada kode *Gallager*. Kode *Gallager* atau disebut juga *Low-Density Parity-Check Codes* (LDPC codes) dengan beberapa *rate* dan panjang blok dapat membentuk spesifikasi matriks *parity-check* jarang yang acak. Dan karena kebenaran suatu *codeword* adalah benar jika memiliki *parity-checks*.

Kode LDPC merupakan kode yang baik dari rentetan kode yang ada ketika proses dekoding mencapai optimal sehingga mampu mendekati batas kapasitas *Shannon* daripada menggunakan kode *Turbo*, serta performansi kode ini tidak hanya menampilkan untuk kanal biner simetris saja, tetapi juga mampu untuk beberapa model kanal yang lain.

Kata kunci : *Bit Flip (BF)*, *Matriks jarang*, *Kode Gallager (Kode LDPC)*

AN ERROR-CORRECTION METHOD BY USING A SPARSE MATRIX

Daud Andreaw / 0222198

Departement of Electrical Engineering, Faculty of Techniques

Maranatha Christian University

Jln. Prof. Drg. Suria Sumantri 65, Bandung 40164, Indonesia

Email : gallagher_fetch22@yahoo.co.id

ABSTRACT

Sometimes, there are some errors occurs when the transmitter sends information (data) to the receiver at the process of communication. Those errors are due to noise, interference, jamming, or the other signal disturbances. There are two ways to find out when errors occurs, using an error-detection and an error-correction.

A method for designing a good error-correcting codes using a Bit Flip (BF) algorithm with a sparse matrix of Gallager codes. The Gallager codes also called as a Low-Density Parity-Check Codes (LDPC codes) with any rate and blocklength can be created simply by specifying the shape of a sparse random parity-check matrix. And, because the validity of a codeword is validated if its parity-checks.

LDPC codes are “very good,” in that sequences of codes exist when optimally decoded, achieve information rates up to the Shannon limit as that of Turbo codes, and these codes performs not only for the binary-symmetric channel performance but also be able for any other channel models.

Keywords : Bit Flip (BF), Sparse Matrix, Gallager Codes (LDPC Codes)

DAFTAR ISI

	Halaman
ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
 BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	1
1.3 Tujuan.....	2
1.4 Pembatasan Masalah.....	2
1.5 Sistematika Pembahasan.....	2
 BAB II LANDASAN TEORI	
2.1 <i>Low-Density Parity-Checks codes</i> (LDPC codes).....	3
2.2 Matriks <i>generator G</i> dan matriks <i>A parity-check</i>	4
2.3 Kode LDPC reguler.....	5
2.4 Konstruksi matriks <i>parity-check</i>	6
2.5 Teorema matriks jarang (<i>sparse matrix</i>).....	8
2.6 .Grafik <i>Tanner (Tanner Graphs)</i>	8
2.7 Proses enkoding pada kode LDPC.....	9
2.8 Transmisi yang melewati kanal <i>Gaussian (Gaussian channel)</i>	12
2.9 Teknik proses dekoding untuk kode LDPC.....	15
2.10 Kanal biner dengan <i>symmetric stationary ergodic noise</i>	18
2.11 Kanal biner simetris (<i>Binary Symmetric Channel/BSC</i>).....	18
2.12 Kode <i>Linear</i>	19

2.13 Kode <i>Turbo</i>	19
2.14 Kode <i>Mackay-Neal</i>	20
2.15 Pencapaian untuk dekoding yang optimal.....	21

BAB III PROSES DAN CARA KERJA

3.1 Algoritma pengulangan pada proses dekoding LDPC.....	22
3.1.1 Langkah horisontal.....	23
3.1.2 Langkah vertikal.....	24
3.1.3 Inisialisasi dan akhir dari algoritma proses dekoding.....	28
3.2.1 Formula rasio dekoder <i>log likelihood</i>	31

BAB IV ANALISA HASIL SIMULASI

4.1 Perangkat Lunak untuk kode LDPC.....	37
4.2 Data Pengamatan dan Analisa.....	37

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan.....	43
5.2 Saran.....	43

DAFTAR PUSTAKA	44
-----------------------------	----

LAMPIRAN A : Fungsi M-File Matlab.....	A
---	---

LAMPIRAN B : Percobaan dengan algoritma <i>Bit Flip</i> (BF) untuk proses dekoding.....	B
---	---

LAMPIRAN C : Percobaan pada data pengamatan dan analisa untuk kode LDPC.....	C
--	---

DAFTAR TABEL

Tabel 4.1 Pengamatan Proses Dekoding dengan Ga,Hs,10,5,.5	38
Tabel 4.2 Bit error BF pada Percobaan 1	38
Tabel 4.3 Pengamatan Proses Dekoding dengan Ga,Hs,20,10,1	39
Tabel 4.4 Bit error BF pada Percobaan 2	39
Tabel 4.5 Pengamatan Proses Dekoding dengan Ga,Hs,50,15,1	40
Tabel 4.6 Bit error BF pada percobaan 3	41

DAFTAR GAMBAR

Gambar 2.1 Proses pembuatan kode (enkoding)	4
Gambar 2.2 Proses pengkodean (dekoding)	4
Gambar 2.3 Matriks A <i>parity-check</i>	5
Gambar 2.4 Matriks A <i>parity-check</i> untuk kode reguler.	7
Gambar 2.5 Antar dua grafik (<i>bipartite graph</i>) yang diasosiasikan dengan matriks A <i>parity-check</i>	9
Gambar 2.6 Hasil permutasi dari baris dan kolom	10
Gambar 2.7 Representasi dari AWGN	14
Gambar 2.8 Kepadatan spektral daya <i>noise white gaussian</i> (WGN)	15
Gambar 2.9 Fungsi Autokorelasi WGN	15
Gambar 2.10 Sebuah pohon <i>parity-check</i> yang diasosiasikan dengan grafik <i>Tanner</i>	17
Gambar 3.1a Sebuah subset dari grafik <i>Tanner</i> yang menunjukkan sebuah pohon untuk bit c_n sebagai akar	25
Gambar 3.1b Sebuah bagian dari grafik <i>Tanner</i> yang sebenarnya dengan node untuk c_1 sebagai akar yang menunjukkan sebuah <i>cycle</i>	26
Gambar 3.2 Pohon tier ke-2	28
Gambar 3.3 Diagram alir subroutine algoritma dekoding dengan BF	31
Gambar 3.4 Proses informasi melewati grafik yang ditentukan oleh A	32
Gambar 3.5 Kondisi independen diantara kumpulan bit	34
Gambar 3.6 Diagram alir algoritma <i>log likelihood</i> untuk kode LDPC	36
Gambar 3.7 Diagram alir algoritma kode <i>Gallager</i> (LDPC)	38
Gambar 4.1 Grafik data pengamatan untuk Kode LDPC	41