プログラミング課題における計算機シミュレーションによる簡易ディジタル系モデルの誤り率特性

栗原　義武 *　村上　竜都 *　白井みゆき *　占部　弘治 *
小野　智裕 *　小西　　耀 *　近藤　大智 *　加藤　幹大 *

Bit Error Rate of Digital System by Computer Simulation for Programming Course

Yoshitake KURIHARA*　Ryuto MURAKAMI*　Miyuki SHIRAI*　Koji URABE*
Tomohiro ONO*　Yo KONISHI*　Daichi KONDO*　Mikihiro KATO*

Bit error rate (BER) performance has been obtained to compare with the performance of signal processing systems in digital systems.

This paper shows that we find BER performance of simple digital systems based on both central limit theory and Box-Muller method through computer simulation. We assume that the noise of our model is Additive White Gaussian Noise (AWGN) with threshold detection of 0.5. We also compare BER performences under gcc in Linux system with that under Visual C in Windows OS.

1. 緒言

本校電子制御工学科においては、平成 28 年度から第 5 学年の学生実験の改善に取り組んでいる [1],[2]。その中の課題の一つとして、ディジタル系簡易モデルについて誤り率特性を求めるシミュレーション課題について述べた [2]。

電子制御実験 2 の実施において、TA (Teaching Assistant) 役の学生は、研究室において Linux 系の gcc を使って誤り率特性のシミュレーションを求める準備を行う。その他の学生は、教育用電子計算機システムまたは学生用実験室において MS-Windows 系 OS の Visual C を使って誤り率特性を求めるプログラミング課題を実施する。

本稿では、白色ガウス雑音を仮定した簡易ディジタル系モデル (図 1) において、C 言語プログラミングによって計算機シミュレーションにより誤り率特性を求め、比較および考察を行う。

2. 誤り率特性の理論計算

簡易ディジタル系モデル [2] における誤り率特性を求める試みにより、誤り率 (BER) の理論値の計算式を導出できるようになったものの、ごく一部の学生には、

- 誤り率の理論値を計算する式の導出をしてもグラフが描けない。
- 誤り率の理論値のグラフを曲線で描くにはデータ不足。

などの問題点が見られる場合がある。

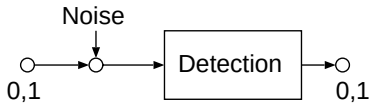

図 1　簡易ディジタル系モデル

しきい値 0.5 で AWGN を仮定した図 1 の BER の理論値は、

$$\mathrm{BER} = \frac{1}{2}\mathrm{erfc}\left(\frac{0.5}{\sqrt{2}\sigma}\right) \tag{1}$$

となるが、この式を導出できても誤り率特性のグラフを描けない学生が意外に多く存在する。ここで σ は雑音の実効値で、正規乱数と仮定すると標準偏差となる。誤り率特性のグラフは、横軸に SNR(Signal to Noise Ratio; 信号対雑音比; SN 比) の値 X [dB] を

$$X = 20\log_{10}\frac{1}{\sigma} \tag{2}$$

で求め、縦軸に BER の値 Y を式 (1) で求めることで描けるようになるが、求めた座標データが不足する場合もある。

令和 4 年 9 月 30 日受付 (Received Sept. 30, 2022)

* 新居浜工業高等専門学校電子制御工学科（Department of Electronics and Control Engineering, National Institute of Technology (KOSEN), Niihama College, Niihama, 792-8580, Japan）

σ の値をいくつか選び、式 (2) と式 (1) を使って、グラフを描けるようになってくる。この段階では、データ数が不足する傾向が見られる。

誤り率の理論値を曲線で描くために、式 (2) より、

$$\sigma = 10^{-\frac{X}{20}} \tag{3}$$

のように、X と σ の関係式を導出する課題に取組む。この課題で式 (3) を導き出せる学生は、更に、

$$Y = \frac{1}{2}\mathrm{erfc}\left(\frac{10^{\frac{X}{20}}}{2\sqrt{2}}\right) \tag{4}$$

という計算式を導き出すことによって、X と Y の連続関数として曲線を描くことができる。

3. C 言語によるシミュレーション

誤り率特性の計算機シミュレーションは、C 言語または C++ 言語を利用したプログラミング課題としている [2]。

雑音としては加法性白色ガウス雑音 (AWGN) を仮定しているため、正規乱数が必要となる。C 言語では、標準的な機能としては `rand( )` を疑似乱数として利用できるが、この値は、Linux 系の gcc では 31 bit の非負の整数値であり、また、MS-Windows 系の Visual C では 15 bit の非負の整数値であるため、一様乱数や正規乱数を得るにはプログラムを作成することが、電子制御実験 2 の課題となっている [2]–[7]。

3 – 1　先の検討 (中心極限法)

誤り率を求める先の試みでは、中心極限法によるシミュレーションを行った [2]。ディジタルデータ数が 100 000 個で SNR が $X = 20$ dB の場合、$\sigma = 0.1$ での実行結果は誤り個数が 0 個となり、BER は対数軸で表示できない。誤り個数が得られたシミュレーション結果と誤り率の理論値は、良く一致した [2]。

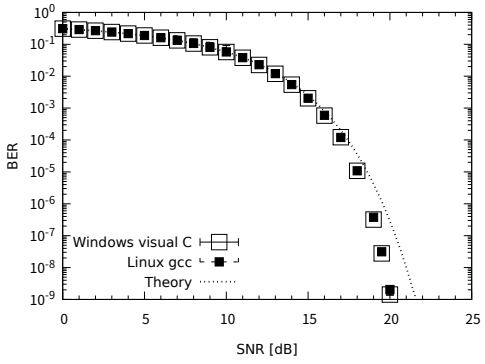

図 2　中心極限法 (先の検討) による誤り率特性

図 2 に示す誤り率特性のグラフは、先の試みと同じ中心極限法によるプログラムで求めた結果であるが、少なくとも 2 桁の誤り個数が得られるまでディジタル入力データの個数を十分に増加させた点に違いがある。図 2 より、SNR=16 dB 程度までは、誤り率のシミュレーション結果と理論値が良く一致しているが、SNR が 17 dB よりも大きくなるにつれて、シミュレーション値は理論値から外れる傾向にある [3],[5]。また、MS-Windows 系の Visual C と Linux 系の gcc では、計算機環境が違ってもシミュレーション結果に大きな違いは見られなかった。

3 – 2　ボックスミュラー法

正規乱数を生成する方法としてボックスミュラー法がある。ボックスミュラー法を適用して正規乱数を生成するには、2 個の一様乱数が必要となる。本稿では、中心極限定法と同様に `rand( )` の値を `RAND_MAX` の値で割ることで一様乱数を求めた [2],[7]。ここに、`RAND_MAX` は `rand( )` の最大値で、表 1 に示すように、計算機環境によって値が異なる。

表 1　計算機環境による `rand( )` と `RAND_MAX` の値

計算機環境	MS-Windows 系 Visual C	Linux 系 gcc
`rand( )`	15 bit	31 bit
`RAND_MAX`	0x7fff	0x7fff ffff
(10 進数)	32 767	2 147 483 647

本稿のシミュレーションでは、ノイズをボックスミュラー法で作成した。ボックスミュラー法は、計算機シュミレーションにおいて、一様分布の乱数から標準正規分布に従う乱数を発生させる手法として知られている。またその式は、U_1,U_2 が独立に [0,1] 上の一様乱数であるとき、

$$N_1 = \sqrt{-2\log_e U_1}\cos(2\pi U_2) \tag{5}$$

$$N_2 = \sqrt{-2\log_e U_1}\sin(2\pi U_2) \tag{6}$$

となり、この N_1, N_2 は標準正規分布に従う。このように作成した N_1, N_2 に係数 σ を乗じたものをノイズ成分とした。また、一様乱数の発生において、本稿では通常考えられる `rand( )` を最大値で除する手法でプログラムを作成した。プログラム上で、U_1, U_2 は [0,1] 上の実数値を取る一様乱数であるので、C 言語では `rand( )/RAND_MAX` により求めることになる。ただし、`rand( )` の値も、`RAND_MAX` の値も整数値のため、そのまま計算すると整数の除算により、商が全て 0 となってしまう場合がある。そこで、(double) または (float) による変数の型変換 (キャスト) を行う必要がある。

4. シミュレーション結果による特性比較

C 言語プログラミングにより図 1 のディジタル系簡易モデルを検討する手順として、まず、ディジタル入力は標準乱数 `rand( )` の整数値を 2 で割った剰余により、0 または 1 の 2 値を得る。これがディジタル信号の入力データとなる。図 1 におけるノイズ (Noise) は AWGN を仮定する。AWGN として、中心極限法とボックスミュラー法の両方の場合とも計算機シミュレーションを検討する。ディジタル信号成分とノイズ成分を足し合わせたものを、0.5 のしきい値で 0 か 1 に振り分け出力データとし、もとの入力データと比較して誤りを数えた。この手順を、所望の SNR に応じて式 (2) または式 (3) により求めたノイズの係数 σ を変えながら、両手法で求めた標準正規乱数に掛けることを繰り返すことで誤り率特性を導出した。

誤り率特性のグラフにおいて、シミュレーション結果の場合、縦軸 BER は、誤り個数を全データ個数で除して求める。横軸 SNR [dB] は、AWGN の標準偏差 σ により式 (2) によって定義される X の値である。

4－1　Linux 系 gcc による誤り率特性

図 3 に、Linux 環境の gcc でコンパイルして求めた計算機シミュレーションによる誤り率特性のグラフを示す。

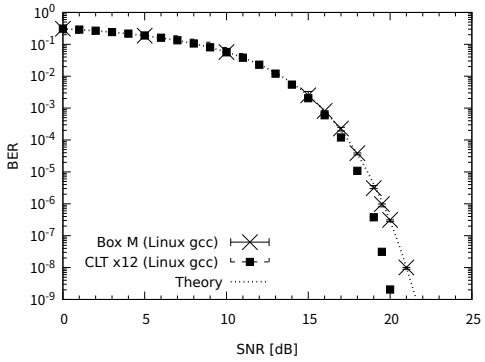

図 3　Linx 系 gcc による誤り率特性

図 3 より、中心極限法を用いた計算機シミュレーション結果が、SNR=16 dB を超えると理論値から外れてしまうのに対して、ボックスミュラー法では、誤り率の理論値と良く一致していることが分かる。

4－2　MS-Windows 系 Visual C による誤り率特性

図 4 に、MS-Windows 環境の Visual C でコンパイルして求めた計算機シミュレーションによる誤り率特性のグラフを示す。Box M はボックスミュラー法を用いたシミュレーション結果で、CLT ×12 は 12 個の一様乱数を足して 6 引いて正規乱数とみなす中心極限法を用いたシミュレーション結果である。いずれも、図 3 と同一のプログラムをコンパイルした結果で、計算機環境の違いだけである。

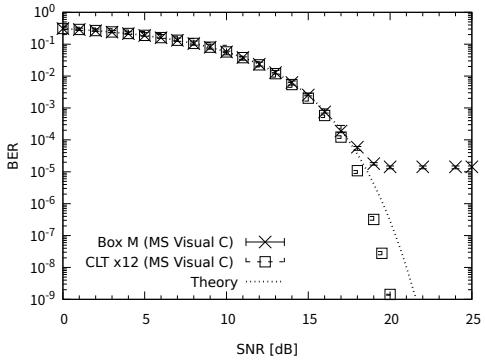

図 4　MS-Windows 系 Visual C による誤り率特性

4－3　ボックスミュラー法を用いた特性比較

AWGN にボックスミュラー法を用い C 言語による計算機シミュレーションにより誤り率特性を導出し、比較したグラフを図 5 に示す。MS-Windows 環境の Visual C および Linux 系の gcc でコンパイルして求めた誤り率のシミュレーション結果を、計算機環境の違いによって比較できる。

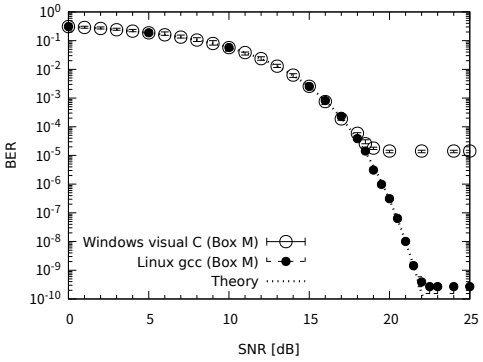

図 5　ボックスミュラー法を用いた誤り率特性

図においてボックスミュラー法のプログラムは同一ソースであるが計算機環境の違いにより、実行結果に違いがみられた。UNIX 系の計算環境の gcc でコンパイルした実行結果 (Linux gcc) は SNR が 20 dB 程度まで理論値と良く一致している。しかし、MS-Windows 系 Visual C 環境 (Windows visual C) では SNR が 17 dB を超えると理論値から大きく外れ、BER は 10^{-5} 程度にとどまる。更に SNR が 22 dB を超えると、Linux 系 gcc であっても理論値から大きく外れ、BER が 10^{-10} 程度において、MS-Windows 系 Visual C と同様の現象がみられる。

4－4　プログラム実行結果の考察

計算機シミュレーションにおいて、C 言語に限らずプログラミングの際に、真の乱数を扱うことは不可能であるため、疑似乱数となる。

MS-Windows 系 Visual C 環境において、疑似乱数 `rand( )` は 15 bit の非負の整数で表現される。また、 Linux 系 gcc 環境において、同一プログラムであっても `rand( )` は 31 bit の非負の整数を取る。よって、プログラム実行すると、 MS-Windows 系 Visual C では、2^{15} 個のうち 1 個は、`rand( )` の値が 0(ゼロ) となる。同様に、Linux 系 gcc では、2^{31} 個のうち 1 個は、`rand( )` の値が 0 となる。

本稿における計算機シミュレーションにおいては、[0, 1] の一様乱数を得るため、多くの文献に見られるよう、`rand( )` をその最大値 `RAND_MAX` で割って求めた [2]。この一様乱数は、ボックスミュラー法のシミュレーションの場合には、U_1 および U_2 として、式 (5) および式 (6) を用いて疑似正規乱数の計算を行う。$U_1 = 0$ のとき、N_1 と N_2 の絶対値は、無限大となる。

MS-Windows 系 Visual C の場合、$1/\mathrm{RAND\_MAX} = 1/2^{15}$ の確率で $U_1 = 0$ となる。このとき、$N_1 = \infty$ または $N_1 = -\infty$ となる。もし、元のディジタル入力データが 0 であれば、雑音が $N_1 = \infty$ の場合誤りとなるが、$N_1 = -\infty$ の場合しきい値 0.5 より小となるので誤りは生じない。元のディジタル入力データが 1 の場合には逆になる。

MS-Windows 系 Visual C による計算機シミュレーションに

よってボックスミュラー法による AWGN を仮定した誤り率を求める場合、SNR をどんなに大きくしても、誤り率の実行結果は、理論的に $1/2^{16} = 1.5258789 \times 10^{-5}$ 程度になると考えられる。また、Linux 系 gcc による計算機シミュレーションによって同様に誤り率を求める場合、SNR をどんなに大きくしても、誤り率の実行結果は、理論的に $1/2^{32} = 2.3283 \times 10^{-10}$ 程度になると考えられる。

中心極限法のシミュレーションの場合、本稿では、疑似一様乱数を 12 個足して 6 引いたものを標準正規乱数の疑似乱数とみなしている。このように求めた疑似乱数の取りうる値は、[−6, 6] となる。疑似乱数のヒストグラムを求めると、正規分布と良く一致しているように見えるが [2]、真の正規乱数は [−∞, ∞] であるので、これが誤り率の理論値とシミュレーション値の差の原因となる。

例えば、SNR が 20 dB のとき、$\sigma = 0.1$ となる。本稿で検討した中心極限法を用いた場合の疑似正規乱数の取りうる範囲は、[−0.6, 0.6] となる。計算機シミュレーションにおけるしきい値検出 (Detection) において、しきい値は 0.5 であるので、入力データ 0 の場合、疑似乱数の [0.5, 0.6] の値となるものが誤り個数としてカウントされる。真の正規乱数を仮定した場合には、標準偏差が $\sigma = 0.1$ の正規分布の確率密度関数を [0.5, ∞] の範囲で積分して求めたものが理論値となる。

5. 結言

ディジタル系簡易モデルにおいて、C 言語プログラミングの標準乱数を利用した計算機シミュレーションにより誤り率特性を求めた。

中心極限法によるシミュレーション結果において、BER が 10^{-3} 程度までは理論値と良く一致しているが、BER が 10^{-3} よりも大きくなるにつれて理論値から外れる傾向にあることが明らかとなった。また、MS-Windows 環境の Visual C と Linux 環境の gcc に違いは見られなかったことから、計算機環境が違ってもシミュレーション結果はほぼ変わらないことが明らかになった。

正規乱数の発生手法として良く知られているボックスミュラー法によるシミュレーション結果においては、教育用の 15bit 整数値の標準乱数である計算機環境 (MS-Windows 系 Visual C) においては、誤り率が 10^{-5} 以上の範囲においては誤り率の理論値とシミュレーション値は良く一致するが、それ以上 SN 比を大きくしても誤り率が小さくならないことが明らかとなった。研究室レベルの 31bit 整数値の標準乱数である計算機環境 (Linux 系 gcc) であれば、誤り率が 10^{-10} 程度まで、誤り率の理論値とシミュレーション値が良く一致することが明らかとなった。

Linux 環境の gcc を用いたシミュレーション結果において、中心極限法は BER が 10^{-3} 程度までは理論値と良く一致しているが、BER が 10^{-3} よりも大きくなるにつれて理論値から外れる傾向にあることに対して、ボックスミュラー法は、少なくとも BER が 10^{-9} までだと理論値と良く一致することが明らかとなった。

MS-Windows 環境の Visual C を用いたシミュレーション結果において、中心極限法は BER が 10^{-3} 程度までは理論値と良く一致しているが、BER が 10^{-3} よりも大きくなるにつれて理論値から外れる傾向にあり、更に SNR が 22 dB を超えると、より大きく外れることに対して、ボックスミュラー法は、10^{-3} 程度までは理論値と良く一致しているが、SNR が 17 dB を超えると理論値から大きく外れ、BER は 10^{-5} 程度にとどまることが明らかとなった。

計算機環境の違いによって、疑似乱数のビット数と誤り率のシミュレーション結果を考察すると、真の正規乱数を仮定した数式を数学的 (解析的) に積分して求めた誤り率の理論値とは大きく外れるものの、疑似乱数のビット数に基づいて計算したボックスミュラー法を用いた場合の誤り率の理論値とは良く一致することが明らかとなった。

本稿では、C 言語による計算機シミュレーションで、一様乱数を計算する際、非負の整数値を取る乱数の基本機能を利用して、その最大値で除した値を一様乱数としたが、一様乱数の発生手法を検討 [5] することによって、ボックスミュラー法を用いた誤り率のシミュレーション結果が異なることも考えられる。また、本稿で検討した中心極限法では、一様乱数を 12 個足して、疑似正規乱数を求めたが、この個数を増やすことによって、誤り率の理論値と良く一致する場合も考えられる。これらについては、今後の検討課題としたい。

本稿では、簡易ディジタル系モデルについての検討であったが、今後、現実の通信系または記録系のディジタル系システムについて検討する場合には、図 1 において、符号化や復号化、波形等化についても考慮する必要がある。

本稿で AWGN としてボックスミュラー法を用いて疑似正規乱数による計算機シミュレーションで誤り率特性を求めたとき、どんなに SNR を高くしても BER が残留する現象が見られたが、これは、エラーフロア現象とも類似している [8],[9]。

光通信では、誤り訂正後のビット誤り率を 10^{-15} 以下という極めて低い値にすることが要求される場合がある [9]。本稿で検討したボックスミュラー法のシミュレーションでは、Linux 系 gcc の結果でも BER=10^{-10} 程度で残留するため、本稿の内容を光通信へ発展させるには、AWGN の発生手法について、更に検討を要する。

一方、光衛星間通信実験衛星においては、上りリンクの所要性能が BER=10^{-3} との報告もある [8]。このような光無線通信に、本稿の内容を適用するには、Linux 系 gcc でも、MS-Windows 系 Visual C でも、問題なく実施可能である。

ディジタル磁気記録系においても、誤り訂正を行う前の生の誤り率においては、従来、10^{-3} 程度の誤り率による性能で十分であるとされ、多くの研究報告においては、10^{-4} の誤り率を達成する SNR の評価が多い [5], [10]–[16]。この場合でも、いずれの計算機環境によるシミュレーション結果にも問題ないといえる。

電子制御実験 2 では、本稿で述べた簡易ディジタル系モデルの誤り率のシミュレーションのほか、放射線の測定と測定結果が正規分布状とみなす統計処理などもあり [2]、本稿と大いに関連する内容である。

また、本稿の実験テーマを実施する際に、電子制御工学科

における情報処理教育による C 言語または C++ 言語教育も必要になる。本稿で検討した、計算機環境の違いによる実行結果については、乱数を取り扱うプログラミング教育にも関連する内容であり、今後の教育改善が期待される。

なお、研究室においては UNIX 系 OS において 48 ビット整数演算を利用した疑似乱数発生関数が備わっている場合があるが、教育用環境で標準的な利用ができないため、本稿では割愛した。

謝辞

電子制御工学科の電子制御実験 2 担当と関係のみなさまに感謝いたします。

参考文献

[1] 松木剛志, 田中大介, 出口幹雄, 占部弘治, 白井みゆき, 栗原義武, “卒業研究の基礎分野を導入した実験手法の試み”, 工学教育, Vol.**66**, No.4, pp.38–44, (2018).

[2] 栗原義武, 松木剛志, 白井みゆき, 佐々木直人, 村上倫平, 辰本一夢, 合田拓海, “卒業研究に関連した学生実験におけるディジタル系プログラミング教育”, 新居浜高専紀要, Vol.**54**, pp.1–5, (2017).

[3] 村上竜都, 栗原義武, 近藤大智, 白井みゆき, 占部弘治, 小野智裕, “誤り率シミュレーションのための C/C++ 言語における機種依存に関する一検討”, 2022 年映像情報メディア学会年次大会, 13C-4, (2022).

[4] 栗原義武, 白井みゆき, 占部弘治, “正規乱数を利用したディジタル系の誤り率シミュレーションに関する一検討”, 2021 映像情報メディア学会冬季大会, 21D-2, (2021).

[5] 森實響, 山口翔大, 栗原義武, “正規乱数を仮定した場合の理論とシミュレーションに関する一検討”, 2017 映像情報メディア学会冬季大会, 24C-1, (2017).

[6] 仲村拓也, 三輪大貴, 栗原義武, 松岡優太, “白色ガウス雑音を仮定した疑似乱数シミュレーションの一検討”, 令和元年度電気関係学会四国支部連合大会論文集, 18-23, p.229, (2019).

[7] 佐々木直人, 合田拓海, 辰本一夢, 村上倫平, 栗原義武, “学生実験のためのディジタル技術に基づく誤り率特性”, 平 28 年度電気関係学会四国支部連合大会講演論文集, 18-18, p.276, (2016).

[8] 岡本英二, 荘司洋三, 豊嶋守生, 高山佳久, “4-4 リピータモードによる誤り訂正符号の効果の実証実験”, 情報通信研究機構季報, Vol.**58**, Nos.1/2, pp.73–82, (2012).

[9] 杉原堅也, 松本渉, 宮田好邦, 杉原隆嗣, 久保和夫, “100Gbps 超級光コア・メトロネットワーク向け誤り訂正技術”, 三菱電機技報, Vol.**59**, No.6, pp.367–370, (2016).

[10] 栗原義武, 小泉祐貴, M.Z Ahmed, 高石雅章, 安藤毅, 大沢寿, 岡本好弘, “垂直磁気記録への CITI 符号の適用”, 新居浜高専紀要, Vol.**41**, pp.51–58, (2005).

[11] Y. Kurihara, Y. Takeda, Y. Takaishi, Y. Koizumi, H. Osawa, M.Z. Ahmed, and Y. Okamoto, “Constructive ITI Coded PRML System Based on Two Track Model for Perpendicular Magnetic Recording,” Journal of Magnetism and Magnetic Materials, Vol.**320**, No.22, pp.3140–3143, (2008).

[12] Y. Kurihara, H. Osawa, Y. Okamoto, P.J. Davey, D.J. Mapps, M.Z. Ahmed, and T. Donnelly, “RLL coded PR (1,2,3,4,3,2,1)ML system for double layer perpendicular magnetic recording,” The 8th Joint MMM-Intermag Conference, Marriott Rivercenter, San Antonio, Texas, USA, CA-08, pp.133–134, Jan. 7–11 (2001).

[13] Y. Kurihara, H. Osawa, N. Fujimoto, H. Saito, Y. Okamoto, H. Muraoka, and Y. Nakamura, “Simplification of PRML System for Perpendicular Magnetic Recording,” 7th European Magnetic Materials and Applications Conference (EMMA ’98), Hotel Boston, Zaragoza, Spain, Fr-P40, p.260, Sep. 9–12 (1998).

[14] H. Osawa, Y. Kurihara, Y. Okamoto, H. Saito, H. Muraoka, and Y. Nakamura, “PRML Systems for Perpendicular Magnetic Recording,” Journal of Magnetic Society of Japan, Vol.**21**, No.S2, pp.399–405, (1997).

[15] Y. Kurihara, H. Osawa, Y. Okamoto, H. Muraoka, and Y. Nakamura, “Performance of PRML Systems in Perpendicular Magnetic Recording by a Bi-layered Main-Pole Head,” 6th European Magnetic Materials and Applications Conference (EMMA ’95), Technishe Universitat Wien, Wien (Vienna), Austria, We-S05, p.168, Sep. 4–8 (1995).

[16] 大沢寿, 栗原義武, 岡本好弘, 西田靖孝, 村岡裕明, 中村慶久, “垂直記録におけるパーシャルレスポンス方式の誤り率特性”, 電子情報通信学会和文論文誌 C-II, Vol.**J77-C-II**, No.4, pp.190–196, (1994).