電気学会論文誌 D（産業応用部門誌）
IEEJ Transactions on Industry Applications
Vol.145 No.4 pp.200–207　DOI: 10.1541/ieejias.145.200

論　文

LFSR を用いた URNG における風向データの削減

非会員　別役　拓哉*　上級会員　市川　周一*a)

Reducing Wind Direction Data for the Unpredictable Random Number Generator based on Linear Feedback Shift Register

Takuya Betchaku*, Non-member, Shuichi Ichikawa*a), Senior Member

（2024 年 3 月 25 日受付，2024 年 8 月 16 日再受付）

Chiba and Ichikawa (2023) proposed an unpredictable random number generator (URNG) that samples a linear feedback shift register (LFSR) at varying intervals fluctuated by weather data. Although their URNG meets the criteria of the Diehard and NIST tests, it requires a substantial amount of weather data. This study examines various design options to reduce wind direction data. Our results show that the resulting random sequence passes the Diehard and NIST tests under the following conditions: (1) use of a 64-bit or longer LFSR, (2) incorporation of a robust hash function (e.g., MD5) for data selection, and (3) adoption of 64 or greater number of observations with sufficient entropy.

キーワード：乱数，URNG, TRNG, LFSR
Keywords: random number, URNG, TRNG, LFSR

1. はじめに

近年，多くの分野で，乱数生成は必須の技術となっている。乱数には，物理現象から生成される真性乱数（TRN；True Random Number）と，確定的アルゴリズムにより数値的に生成される疑似乱数（PRN；Pseudo Random Number）がある。TRN の生成にはハードウェア（TRNG；True Random Number Generator）が必須であるが，PRN はソフトウェアでも生成することができる。さらに，TRN と PRN の中間的な乱数として URN（Unpredictable Random Number）が提案されている[1]。

正岡ら[2] は，ソフトプロセッサに 128 ビットの LFSR（Linear Feedback Shift Register）を追加し，適切な条件下で LFSR のサンプリングを行うことにより URN が生成できることを示した。この URNG（Unpredictable Random Number Generator）のエントロピー源は，LFSR を読み出す間隔の揺らぎである。さらに鴨狩と市川[3] は，正岡らの URNG について LFSR の仕様とサンプリング間隔を変化させて乱数品質を調査し，適切な設計要件・使用条件を検討した。

正岡らの URNG には LFSR（ハードウェア）が必須であるが，LFSR 自体はソフトウェアでエミュレートすることが可能である。適切なエントロピー源で LFSR の読み出し間隔に揺らぎを与えれば，ソフトウェアで URNG を実現することができる。

千葉と市川[4] は，LFSR を用いた URNG のエントロピー源として，気象データを使用することを提案した。気象データは自然由来のエントロピー源であるため，TRNG のように予測不能な乱数を生成できる。一方，気象データは公共機関により一般公開されており，インターネットから無料で入手可能である[5]。そのため通信機能が使用できれば，ソフトウェアだけで実質的に予測不能な乱数（URN）を生成できる。千葉と市川[4] は，中部地方 27 か所の風向データ（12 年分）をエントロピー源として使用し，適切な乱数生成方法を検討するとともに，生成された乱数列が Diehard テスト[6] および NIST テスト[7] に合格することを示した。

千葉と市川[4] は，気象データをエントロピー源とする URNG の実現性を検討するため，大量の風向データ（2.5×10^6 個）を利用して乱数を生成した。手法の妥当性を検討するには，データ（エントロピー源）の不足を避ける必要があったためである。しかしデータ量が大きいと，実装時にメモリを多く使用し，ダウンロード時の通信量も

a) Correspondence to: Shuichi Ichikawa. E-mail: ichikawa@tut.jp
* 豊橋技術科学大学大学院電気・電子情報工学専攻
〒441-8580　愛知県豊橋市天伯町雲雀ヶ丘 1-1
Department Electrical and Electronic Information Engineering, Graduate School of Engineering, Toyohashi University of Technology
1-1, Hibarigaoka, Tempaku-cho, Toyohashi, Aichi 441-8580, Japan

大きくなる。実用性の観点から，データ量と乱数品質の関係を再検討し，データ量を削減することが望まれる。

また千葉と市川[4] は，気象データの選択にハッシュ関数を使用することを提案し，ハッシュ関数の選択が乱数品質に影響することを示した。特定地点の気象データには固有の偏りや時系列の依存性があるため，複数地点・複数時刻のデータをハッシュ関数で選択することにより，乱数性の低下を防ぐことができる。千葉と市川[4] は，単純なハッシュ関数でも乱数テストに合格することを示したが，より洗練されたハッシュ関数を評価していない。

本研究では，千葉と市川[4] の URNG について，その設計原則は維持したままデータ量を削減する方法について検討する。データ量を削減するにはハッシュ関数の選択も重要になると予想されるので，新たに MD5[8] と SHA256[9] について検討を行う。観測地点数や観測期間の削減と乱数品質の関係についても評価する。

なお本稿は，著者らによる研究会発表[10] に大幅な加筆を行ったものである。

2. 気象データと LFSR を用いた URNG

まず本章では，千葉と市川[4] が提案した URNG の設計と，その評価方法について概観する。

〈2·1〉 URNG の設計 LFSR を用いた URNG は，先行研究[2][3] によって提案された。8 ビット LFSR の例を Fig. 1 に示す。LFSR は帰還多項式に対応するタップシーケンスで表現され，Fig. 1 のタップシーケンスは [8, 6, 5, 4] である。本研究では，特に明記がない限り，千葉と市川[4] が採用した 32 ビット LFSR（タップシーケンス [32, 30, 17, 12, 3, 1]）[11] を実験に用いる。

LFSR の値を読む際に，n 回目のサンプリング間隔 $S(n)$ を以下の式で決める。サンプリング間隔とは，$(n-1)$ 回目のサンプルのあと帰還多項式を $S(n)$ 回適用して，n 回目のサンプルを行うという意味である。

$$S(n) = \alpha(n) + \beta \cdots\cdots (1)$$

ここで定数 β は基本となるサンプリング間隔（基本間隔），$\alpha(n)$ は風向データから生成する揺らぎで $0 \le \alpha(n) \le 15$ である。風向データ（16 方位）を 4 bit に符号化する方法は，Fig. 2 に示すとおりである（グレイコード）。

揺らぎ $\alpha(n)$ の生成方法を比較検討するため，千葉と市川[4] は以下の 4 つの方法を用いて URN を生成した。風向データは，時系列順に m 個の要素を持つ配列 $wdata$ に格納する。各要素のデータは 16 方位のグレイコード（0〜15）である。風向は短時間で変わりにくいので，時系列順に風向データを用いる方法（Method 1）では，乱数性が低下する可能性がある。そこで Method 2〜4 では，配列 $wdata$ へのアクセス順序に簡単なハッシュ関数を適用した。

Method 1. 時系列順に風向データを使用するため，揺らぎ $\alpha(n) = wdata[(n-1) \mod m]$ とする。

Method 2. 除算法のハッシュ関数で風向データの利用順序を変更する。$\alpha(n) = wdata[8(n-1) \mod m]$，ここで m は係数 8 と互いに素で，なるべく大きい値を使用する。

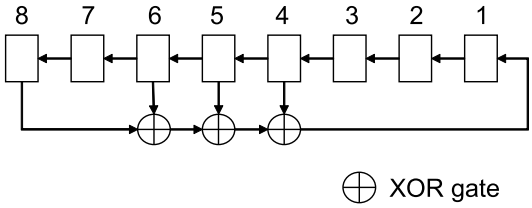


Fig. 1. An example of 8-bit LFSR [8, 6, 5, 4][3].

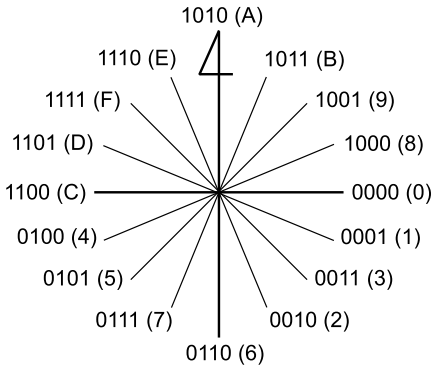


Fig. 2. Enconding of wind direction[3].

Method 3. 除算法のハッシュ関数で風向データの利用順序を変更する。$\alpha(n) = wdata[x(n-1) \mod m]$，ここで係数 $x = (\sqrt{5}-1) \times 2^{m+1}$ としている。この方法は乗算法で用いる係数 $(\sqrt{5}-1)/2$ を除算法に適用し，固定小数点演算で実現したものともいえる。

Method 4. 乗算法のハッシュ関数[12] で風向データの利用順序を変更する。$\alpha(n) = wdata[\lfloor m(A(n-1) \mod 1) \rfloor]$，ここで $A = (\sqrt{5}-1)/2$。

さらに本研究では，乗算法や除算法より複雑なハッシュ関数についても新たに検討した。

MD5. MD5[8] は 128 ビットを出力するハッシュ関数である。揺らぎ $\alpha(n) = wdata[\mathrm{MD5}(n)$ の上位 32 ビット $\mod m]$ とする。

SHA256. SHA256[9] は，256 ビットを出力するセキュアハッシュ関数である。揺らぎ $\alpha(n) = wdata[\mathrm{SHA256}(n)$ の上位 32 ビット $\mod m]$ とする。

〈2·2〉 評価方法

〈2·2·1〉 Diehard テスト 本研究では，先行研究[4] と同じく Diehard テスト[6] を利用して乱数品質の評価を行う。Diehard テストは NIST テスト[7] ほど厳格ではないが，必要なデータ量が少ないため試行錯誤の多い研究段階に適している。

Diehard テストは全 18 種のテストからなり，各テストで 1〜100 個（合計 313 個）の p 値を出力する。合格判定の基準は定められておらず，利用者の判断に委ねられているので，本研究でも先行研究の基準にならって，以下のように乱数品質を評価する。

Table 1. Diehard evaluation criteria[2].

Decision	Condition
PASS	$0.005 \le p < 0.995$
WEAK	$0.000001 \le p < 0.005$, or $0.995 \le p < 0.999999$
FAIL	$p < 0.000001$, or $0.999999 \le p$

入力が理想的乱数であれば p 値は区間 [0,1) で均等に分布することが期待されるので，Table 1 に示した基準で各 p 値の成功（PASS）／弱成功（WEAK）／失敗（FAIL）を判定する。FAIL の発生確率（期待値）は 2×10^{-6} なので，入力が乱数であれば（ほぼ）発生しない。WEAK の発生確率（期待値）は 1×10^{-2} なので，WEAK が 1% 程度発生することは正常である。

単純に 313 個の p 値について PASS/WEAK/FAIL の個数を示すと，多くの p 値を出力するテストのウエイトが大きく見えてしまう。そこで以下の方法により，各テストの結果を判定する。各テストで出力される p 値の個数が 3 個以上であれば，得られた p 値の分布が一様であるかどうかの判定を Kolmogorov-Smirnov 検定により行い，得られた p 値を Table 1 に示した基準で判定する。テストの出力する p 値が 1 個か 2 個であれば，以下に述べる方法で結果を判定する。各テストで出力される p 値に，ひとつでも FAIL が含まれれば，そのテストは FAIL。出力される p 値に FAIL がなく，WEAK だけであれば，そのテストは WEAK。出力される p 値が FAIL がなく，PASS を含んでいれば，そのテストは PASS とする。

こうして計算した全 18 テストの結果（PASS/WEAK/FAIL の内訳）により，乱数列の品質を評価する。全ての実験について PASS/WEAK/FAIL の内訳を示すと冗長になるため，以下，本稿では FAIL の個数に着目して，乱数品質を評価することとする。

〈2·2·2〉 NIST テスト NIST SP 800-22[7] は，NIST（National Institute of Standards and Technology）により規定された乱数および疑似乱数の統計検定スイートであり，広く使用されている。NIST テストは 15 種のテストからなり，結果の解釈方法についても明確に定義されている。

NIST テスト[7] では 1 回のテストで約 10^6 ビットを使用し，そのテストを 1000 回以上行うことが推奨されている。Diehard テスト[6] では約 10^8 ビットのデータ量が必要であるが，NIST テストには合計 10^9 ビット程度が必要になる。従って本研究では，試行錯誤を行う際に Diehard テストを用い，最終的な乱数品質の検定に NIST テストを用いることにする。

3. データ量の削減

本章では，先行研究[4] の追実験と，風向データ削減方法の検討を行う。

〈3·1〉 期間の削減 千葉と市川[4] は，27 か所の 12 年分の風向データを用いて URN を生成し，その URN を Diehard テストと NIST テストで検証した。Diehard テストは 4 種のハッシュ関数（Method 1〜4）全てで合格したが，必要なデータ量の最低限については検討されていなかった。

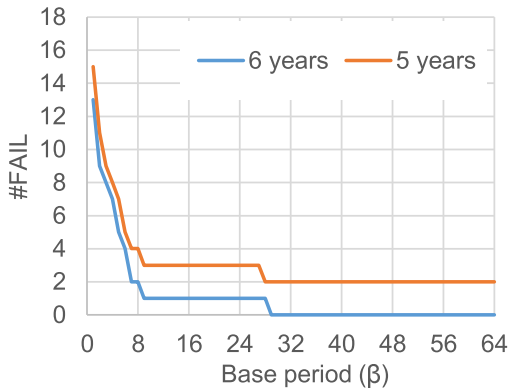


Fig. 3. Diehard results of Method 1 (5 years, 6 years).

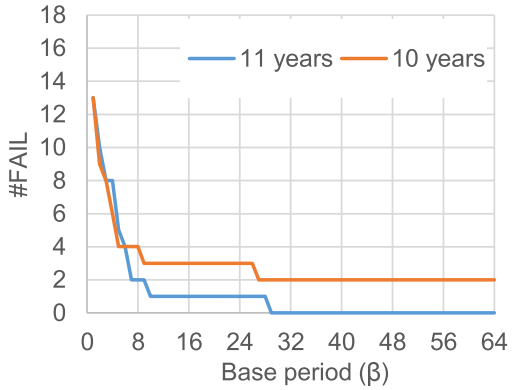


Fig. 4. Diehard results of Method 2 (10 years, 11 years).

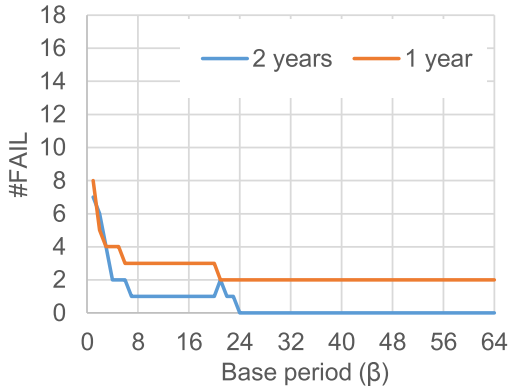


Fig. 5. Diehard results of Method 3 (1 year, 2 years).

そこで本節では先行研究[4] の追試を行い，データ量を 12 年分から 1 年単位で減らしながら Diehard テストで乱数品質を検証した。Fig. 3〜Fig. 7 は，その結果を図示したものである。

それぞれの図は，〈2·1〉節で説明した方法で生成した URN を Diehard テストで評価した結果を示したものである。横軸は基本サンプリング間隔であり，(1) 式の β に相当する。縦軸は FAIL したテストの数で，0 になれば Diehard テストに合格したと判定する。いずれも基本間隔 β が増

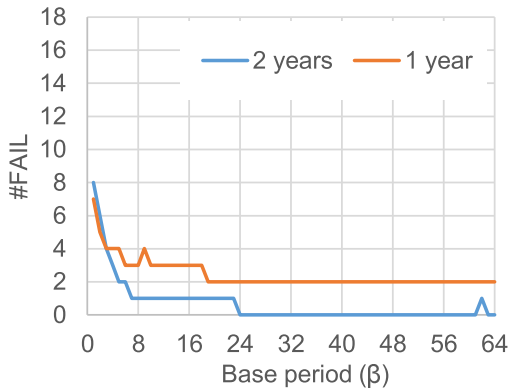


Fig. 6. Diehard results of Method 4 (1 year, 2 years).

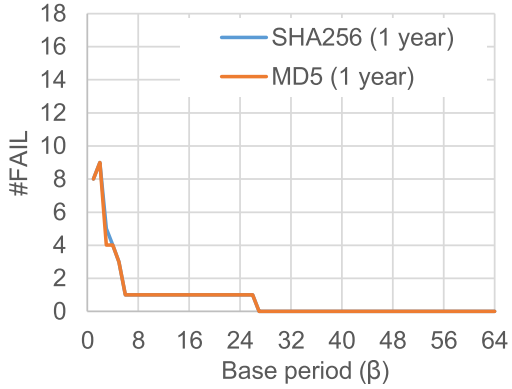


Fig. 7. Diehard results of MD5 and SHA256 (1 year).

加すると乱数品質は向上し，データ量が十分であれば概ね $\beta \geq 32$ で FAIL しなくなる。

Method 1 は，データ 5 年分のとき $\beta \geq 32$ でも Diehard テストに合格できないが，データが 6 年分あると合格する（Fig. 3）。データ 5 年分で $\beta \geq 32$ のとき FAIL したのは Binary Rank テスト（31×31，32×32）であり，これはエントロピー不足で数列の中に線形従属性が現れたことを示す。同様に Method 2 は 11 年分で合格し（Fig. 4），Method 3 と 4 では 2 年分あれば合格する（Fig. 5，Fig. 6）。

MD5 と SHA256 は，風向データ 1 年分でも $\beta \geq 27$ で合格する（Fig. 7）。これに対して Method 1〜4 はハッシュ関数としての機能が低いため，揺らぎ $\alpha(n)$ に現れるエントロピーが不十分で，必要な風向データ量が大きくなる。データ量を削減するためには，良質なハッシュ関数の利用が必須であることが分かった。

27 か所 1 年分の風向データ量は 115 KiB 程度であるが，このサイズでも無視できるほど小さいとは言えない。そこで以下の章ではハッシュ関数として MD5 を採用し，風向データの使用量を更に削減することを試みる。

〈3･2〉 地点数の削減 風向データの量は，観測地点の数に概ね比例している[†]。本節では観測地点の数を減らして，データ量を削減することを試みる。

[†] 各地点の観測回数が同じであれば，地点毎のデータ量は同じになる。しかし欠測状況や観測品質が地点毎に異なるため，データ量にも差が生じる[(4)]。

Table 2. Entropy of wind direction for each observation point.

Observation Point	Entropy	Relative Entropy
Shizuoka	3.877239	0.969310
Kofu	3.877195	0.969299
Niigata	3.875439	0.968860
Iida	3.857038	0.964260
Kanazawa	3.855254	0.963813
Takayama	3.842137	0.960534
Mishima	3.787677	0.946919
Nagano	3.781706	0.945427
Matsumoto	3.777814	0.944454
Gifu	3.775223	0.943806
LakeKawaguchiko	3.774503	0.943626
Takada	3.760104	0.940026
Aikawa	3.734653	0.933663
Hamamatsu	3.725353	0.931338
Irago	3.668932	0.917233
Karuizawa	3.667610	0.916903
Suwa	3.656269	0.914067
Fushiki	3.645322	0.911331
Toyama	3.580482	0.895120
Nagoya	3.563186	0.890797
Ajiro	3.548454	0.887114
Fukui	3.546188	0.886547
Omaezaki	3.522868	0.880717
Tsuruga	3.332330	0.833083
Wajima	3.294481	0.823620
Irozaki	3.092012	0.773003

観測地点により，風向データの含むエントロピーは異なっている。例えば地形等の影響で風向に偏りがあれば，風向データのエントロピーは減少し，生成される乱数列の品質も低下することが予想される。

Table 2 に，各観測地点の風向データのエントロピーをまとめた。先行研究[(4)] では 27 か所のデータを使用していたが，富士山は実質的に観測データがないため[††]，表から除外している。風向データは Fig. 2 に示す 4 ビット（16 方位）であるため，エントロピーは最大で 4 ビットとなる。参考のため，相対エントロピーも表示した。

26 地点のうち，エントロピー最大の地点は静岡，最小の地点は石廊崎となる。そこで，静岡（1 地点），石廊崎（1 地点），静岡と石廊崎（2 地点）のデータ（各 12 年分）を用いて乱数列を生成し，その品質を Diehard テストで評価した。結果を Fig. 8 にまとめる。いずれの場合も，基本間隔 $\beta \geq 35$ で Diehard テストに合格している。

Fig. 7（27 地点 1 年分）と比べると，FAIL が無くなる β の値が大きくなっているが，これはデータ量が小さいためと考えられる。27 地点 1 年分の風向データは約 115 KiB，1 地点 12 年分の風向データは約 51 KiB である。実際，FAIL が無くなる条件は，静岡のみでは $\beta \geq 31$ であるが，静岡と石廊崎の 2 地点では $\beta \geq 27$ となり，小さい間隔で合格するようになる。

地点毎の差については，$28 \leq \beta \leq 30$ でエントロピーの小さい石廊崎が合格し，エントロピーの大きい静岡が FAIL

[††] 品質 0 の項目が並び，実質的な観測結果がない。

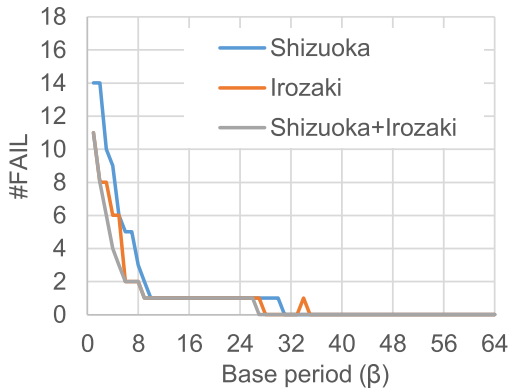


Fig. 8. Diehard results of Shizuoka and Irozaki (12 years).

している。一方 $\beta = 34$ では石廊崎だけが FAIL しており，静岡は合格している。このように，静岡と石廊崎の差は明確であるとはいえない。

以上の結果によれば，今回扱った 26 地点の範囲内では，風向データのエントロピーは乱数品質に大きな影響を与えていない。観測地点の選択より，データ量の影響が大きいと考えられる。

〈3·3〉 地点数と期間の削減 〈3·2〉節の結果を踏まえて，1 地点（石廊崎）のデータだけを使用し，データ量を削減して乱数品質を評価した。2014 年 1 月 1 日 1 時からの石廊崎の風向データを時系列順に使用し，それぞれ 32 回，64 回，128 回の観測値をエントロピー源として乱数生成を行った。評価に使用する 6 つのデータセットの関係を Fig. 9 に図示する。データセット 32(a) はデータセット 64(a) の前半，またデータセット 32(b) はデータセット 64(a) の後半部分と同じデータである。風向データは 1 時間に 1 回の計測であるため，32 回は約 1.3 日，64 回は約 2.7 日，128 回は約 5.3 日分のデータに相当する。

URNG のエントロピー源として使用するデータセットが固定（不変）であれば，それは URNG ではなく PRNG である。従って，URNG で継続的に乱数を生成する際には，エントロピー源を随時更新することが前提となる[4]。気象観測では最新のデータが逐次追加されてゆくため，新しいデータが利用可能になったらデータセットを更新することにより，生成される乱数列が実質的に予測できなくなる。本節では，データセット更新による乱数品質の変動を観察するため，同一サイズについて複数のデータセット（例：64(a) と (b)）で乱数品質を検証する。Fig. 9 において，データセット 64(a) は更新前，64(b) は更新後という関係になる。

Fig. 10 は，6 つのデータセットに対する Diehard テストの結果をまとめたものである。データセット 32(a) では，β を大きくしても FAIL 数が 2 以下にならない。データセット 32(a) の観測期間内では，風向が常に一定（西）であり，データセットに含まれるエントロピーがゼロになっている。従って揺らぎ $\alpha(n)$ は定数となり，サンプリング間隔が常に一定になる。このとき，32 ビット LFSR では周期

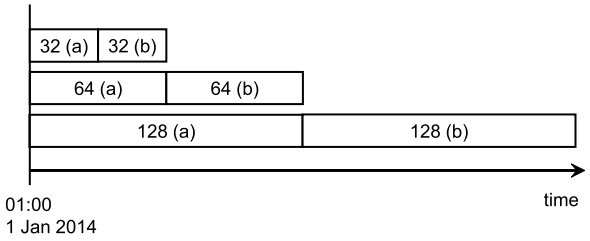


Fig. 9. Evaluation data set.

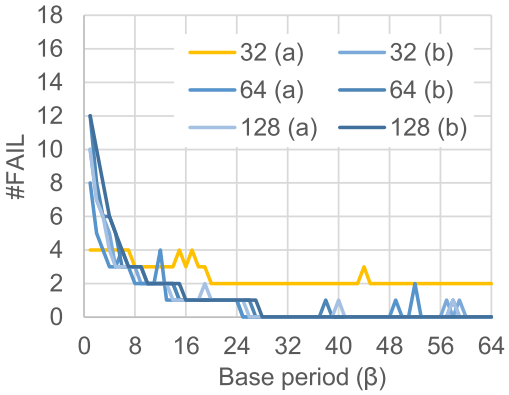


Fig. 10. Diehard results of Irozaki (32–256 data).

Table 3. Diehard and NIST test results of Irozaki (32–128 data, 32-bit LFSR, $\beta = 32$).

Dataset	Size (B)	Diehard (FAIL)	NIST Result
32 (a)	16	2	fail (Rank, RandomExcursions)
32 (b)	16	0	pass
64 (a)	32	0	pass
64 (b)	32	0	fail (NonOverlappingTemplate)
128 (a)	64	0	fail (LongestRun)
128 (b)	64	0	pass

不足のため，Binary Rank（31×31）と（32×32）に合格することができない。32(a) 以外のデータセットにはエントロピー（風向の変化）が含まれるため，$\beta \geq 28$ の多くの β で FAIL が 0 となる。

観測回数が 32 回でもデータセット 32(b) では FAIL を 0 にできることから，観測回数不足が 32(a) の不合格の直接的原因であるとは言えない。気象観測では同じ風向が継続することは珍しくないので，観測回数の少ないデータセットではエントロピーが枯渇する可能性が高くなる，ということは言える。

次に，Diehard テストより厳格な NIST テストにより，データセット毎の乱数品質を評価する。Fig. 10 より，データセット 32(a) 以外では $\beta = 32$ で Diehard テストに合格するので，$\beta = 32$ で乱数列を生成して NIST テストにより評価する。評価結果を Table 3 にまとめる。Table 3 において，Size はエントロピー源のサイズ（バイト単位）を表している。風向は 16 方位なので，1 回の観測が 4 ビット=0.5 バイトになる。Diehard の FAIL 数は，Fig. 10 の $\beta = 32$ から抜き出したものである。NIST テストに fail した場合は，fail したテスト名をカッコ内に示した。

Table 3 に示した通り，データセット 32(a), 64(b), 128(a) は NIST テストに不合格となった。32(a) は Diehard テストにも合格しないので，NIST テストで不合格となるのは自然である。64(b) と 128(a) は，Diehard テストには合格するが，NIST テストには失敗した。32(a) と (b)，64(a) と (b)，128(a) と (b) がそれぞれ時系列的に連続していることから，データサイズが 32，64，128 のいずれの場合も，採用する観測期間によって乱数品質が変動することが分かった。

そこで，安定して乱数検定に合格する条件を検討した。市川[13] は，LFSR を用いた PRNG について評価結果を報告している。その中の Leap-ahead LFSR に関する評価は，本研究において $\alpha(n) = 0$ とした場合に相当する。市川によれば，LFSR 長が概ね 40 ビット未満である場合は，基本サンプリング間隔 β を大きくしても Diehard テストに合格することはできない。LFSR 長が 44 ビット以上で，$\beta = 64$ のときは，NIST テストにも合格したことを報告している。

市川の結果[13] によれば，Table 3 の評価条件では LFSR 長と β が小さすぎて，気象データのエントロピーが枯渇したとき乱数品質が低下する（乱数テストに合格できない）ことが予想される。そこで LFSR 長を 64 ビットとし，基本間隔 β を 64 まで大きくして，生成された乱数列を Diehard テストおよび NIST テストで評価した。64 ビット LFSR のタップシーケンスは，$[64, 61, 56, 31, 28, 23]$ とした[11]。

64 ビット LFSR による評価結果を Table 4 に示す。Diehard テストは，64(b) を除く全てのデータセットで合格した。データセット 64(b) では OPERM5 に FAIL するが，Diehard テストの OPERM5 にはバグがあることが知られている[14]。過去の研究でも OPERM5 で不可解な FAIL が多く観察されていることから，Brown[14] のアドバイスに従い，64(b) における OPERM5 の FAIL は誤検出であると判断する。NIST テストにおいては，データセット 32(a) で fail したが，他の 5 つのデータセットは合格となった。データ数が 32 でも Diehard テストには合格できるが，より厳格な NIST テストではエントロピーが不足した可能性がある。

これらの結果から，LFSR 長を 64 ビットとし，データセットに含まれる観測回数を 64 以上にすれば，Diehard および NIST テストに合格すると予想される。64 回（32 バイト）程度のデータであれば通信コストも微小で，攻撃者の予測を避けるために定期的にデータセットを更新してもコスト面・実用面の問題は発生しないと考えられる。

Table 4. Diehard and NIST test results of Irozaki (32–128 data, 64-bit LFSR, $\beta = 64$).

Dataset	Size (B)	Diehard (FAIL)	NIST Result
32 (a)	16	0	fail (NonOverlappingTemplate)
32 (b)	16	0	pass
64 (a)	32	0	pass
64 (b)	32	1 (OPERM5)	pass
128 (a)	64	0	pass
128 (b)	64	0	pass

4. 考　察

〈4·1〉 風向データのサイズ　3 章までの結果から，適切な LFSR，適切な β，質の良いハッシュ関数を用いることにより，比較的エントロピーの低い観測地点の風向データであっても，64 回程度（32 バイト相当）のデータで揺らぎを与えることにより，乱数検定に合格する乱数列が得られることが分かった。千葉と市川[4] の URNG では 27 か所 12 年分（1335 KiB）の風向データを使用していたが，本研究では最小で 32 バイト程度まで削減できることを示した。

ただし，著者らは「最少の風向データを用いた URNG」の使用を推奨しているわけではない。本研究は「乱数テストに合格する URNG」を設計するための必要条件を示したものである。URNG の本質は「実質的に予測不能」であることであり，乱数テストに合格することは，その要件の一部に過ぎない。実際，疑似乱数生成器（PRNG）の出力は予測可能であるが，乱数テストに合格する PRNG を設計することは難しくない†。

URNG の出力を予測困難にするのは外部から与えるエントロピーであり，本研究においては風向データである。その意味で，使用するデータ量を大きくするほど，出力の予測は困難になる。

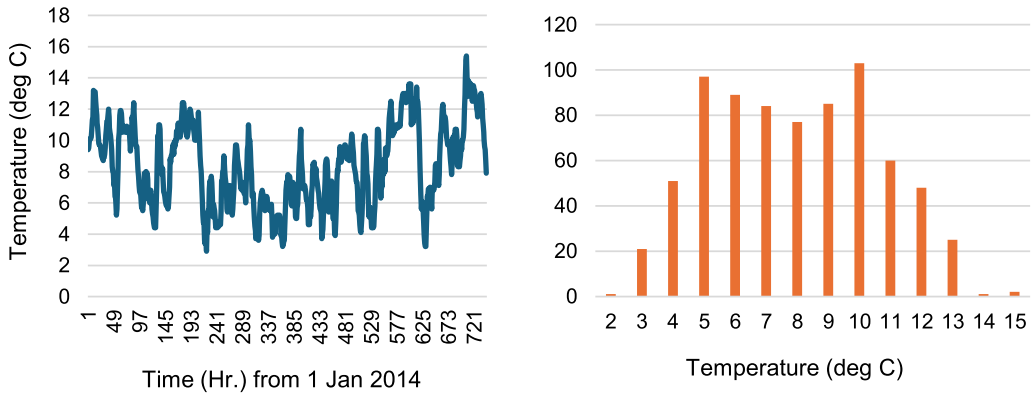
また〈3·3〉節で示した通り，1 か所の観測データだけを用いると，風向が変わらない場合に即座にエントロピーが枯渇してしまう。地理的に離れていて風向データの相関が少ない複数地点からのデータをエントロピー源にすることにより，エントロピー枯渇の危険性を減らすことができる。同様に，観測地点が故障で欠測になった場合もエントロピー枯渇に直結するが，観測地点を複数使うことにより冗長性を増すことができる。攻撃者による観測データへの干渉についても，観測地点が少ないと脆弱になるので，複数地点を採用することが推奨される。

このように，エントロピー源として使用する風向データの構成（地点，期間，等）は，多面的な視点で検討し決定すべきである。

〈4·2〉 エントロピー源の選択　〈4·1〉節では，風向データ量を減らす場合の問題点と，URNG 設計において配慮すべき点について考察した。

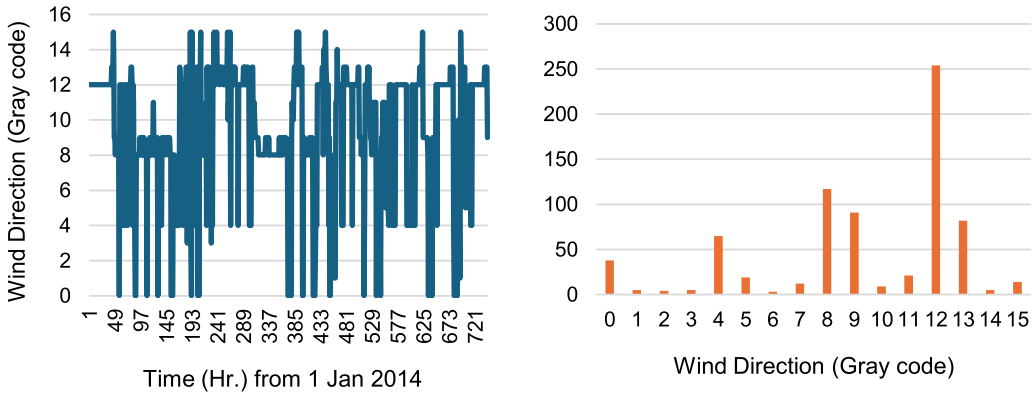
これに対し，風向というエントロピー源に問題があるなら，風向以外の気象データを利用するというアプローチも考えられる。本研究の目的は，1 章でも述べた通り，千葉と市川[4] の URNG で用いる風向データの量を削減することであるから，風向以外のデータの利用については基本的に本論文の範囲外である。しかしながら，今後の展望を含めて本節で若干の考察を加える。

† LFSR を用いた PRNG については，市川[13] を参照して頂きたい。



(a) Time series (b) Histogram

Fig. 11. Temperature of Irozaki in Jan. 2014.



(a) Time series (b) Histogram

Fig. 12. Wind direction of Irozaki in Jan. 2014.

Fig. 11 は，石廊崎の 2014 年 1 月における気温を示している。Fig. 11(a) は 1 カ月の気温変化を時系列で示したものであり，Fig. 11(b) は気温の整数部分に着目して出現度数を数えたヒストグラムである。同様に，Fig. 12 は石廊崎の 2014 年 1 月における風向を示している。Fig. 12(a) 左端で同じ風向が続く部分が，〈3·3〉節のデータセット 32(a) に相当する。ヒストグラムを見る限り，風向は大きく変わっているが，出現頻度に大きな偏りがある。気温は変化が大きくない（連続的である）が，出現頻度の偏りは風向より小さい。

風向データは元々 16 方位に離散化されているため，URNG で使用する際に前処理が不要であった。それに対し，気温データは連続的なデータであるため，何らかの前処理で離散化する必要がある。前処理の方法によって URNG の乱数品質が変わることが予想されるので，データの性質に合わせた前処理・離散化手法を比較検討する必要がある。また，風向は短時間で変動するが，気温の変動は 1 日〜1 年単位で周期が長い。従って，データセットのサイズが小さい場合には，気温は変動が小さくエントロピーが不足する可能性がある。このように，データの性質に合わせたエントロピー抽出手法を検討する必要がある。

Table 5 は，AMEDAS の 6 つの観測項目について，石廊崎の 2014 年 1 月のデータからエントロピーを求めたものである。気温については，Fig. 11(b) のヒストグラムからエントロピーを計算している。同様に風向は Fig. 12(b) から計算した。雨量は 0.5 mm 単位，風速はデータの整数部分，日照は 0〜1 を 0.1 刻み，湿度は 0〜100% を 10% 刻みで区分して，各区分の出現頻度からエントロピーを計算している。この区分（前処理）は全く素朴かつ直感的なもので，処理方法を変えればエントロピーも変わる。本節における素朴な考察のための試験的データと御理解いただきたい。

Table 5. Entropy of weather data in Jan. 2014, Irozaki.

Item	Entropy (bit)
Temperature (deg C)	3.368
Rain (mm)	0.300
Wind Speed (m/s)	3.539
Wind Direction	2.976
Sunshine (%)	1.767
Humidity (%)	2.240

Table 5 で見る限り，風速や気温もエントロピー抽出に利用できる可能性がある。また，湿度や日照も，補助的に使用できる可能性がある。しかしこれらのデータを使用するには多くの検討項目と作業項目が残されており，それら全ては今後の研究に委ねることとする。

5. おわりに

本研究では，千葉と市川[4] の提案した URNG について，その設計を再検討し，使用する風向データの量を削減することを試みた。32 ビット LFSR を使用した場合，地点数と観測期間の両方を削減することが可能で，風向データを 64 回分まで削減しても，Diehard テストに合格できることを示した。NIST テストは Diehard テストより厳格であるが，64 ビット LFSR を使用し，64 回分の風向データを使用することで，NIST テストにも合格することを確認した。

本研究では URNG の設計指針（の一部）について定量的に検討した。乱数生成器は単体で使われることは少なく，より大きなシステムの構成要素として用いられることが多い。システムの目的や要件に合わせて乱数生成器を選択・設計すべきであり，本研究の成果は，選択・設計のための基礎的データを提供したことである。利用者は，システム全体の設計要件を配慮しつつ，本研究で示した必要条件に対して十分なマージンを持った設計をすることが望まれる。

本研究では千葉と市川[4] の研究に従い，風向データを URNG のエントロピー源として使用した。その他の気象データ（気温など）あるいは複数データ源を組み合わせてエントロピー源とする URNG も考えられるが，それについては今後の課題とする。

謝　辞

本研究の一部は JSPS 科研費 20K11733 および 24K14878 の支援による。

文　献

（1） A. Suciu, S. Banescu, and K. Marton: “Unpredictable random number generator based on hardware performance counters”, Digital Information Processing and Communications (ICDIPC 2011), pp.123–137, Springer-Verlag (2011)

（2） H. Masaoka, S. Ichikawa, and N. Fujieda: “Random Number Generation from Internal LFSR and Fluctuation of Sampling Interval”, *IEEJ Transactions on Industry Applications*, Vol.141, No.2, pp.86–92 (2021)
正岡秀崇・市川周一・藤枝直輝：「内蔵 LFSR とサンプリング間隔の揺らぎを利用した乱数生成手法」, 電学論 D, Vol.141, No.2, pp.86–92 (2021)

（3） H. Kamogari and S. Ichikawa: “Evaluation of a random number generator based on an internal linear feedback shift register”, *IEEJ Transactions on Industry Applications*, Vol.143, No.2, pp.87–93 (2023)
鴨狩滉斗・市川周一：「内蔵 LFSR を用いた乱数生成方法の評価」, 電学論 D, Vol.143, No.2, pp.87–93 (2023)

（4） A. Chiba and S. Ichikawa: “Evaluation of Random Number Generator Utilizing Weather Data and LFSR”, *IEEJ Transactions on Industry Applications*, Vol.143, No.2, pp.80–86 (2023)
千葉歩武・市川周一：「気象データと LFSR による乱数生成手法の評価」, 電学論 D, Vol.143, No.2, pp. 80–86 (2023)

（5） 気象庁：「過去の気象データ・ダウンロード」, https://www.data.jma.go.jp/risk/obsdl/index.php

（6） G. Marsaglia: “Diehard battery of tests of randomness (Archived)”, https://web.archive.org/web/20160125103112/http://stat.fsu.edu/pub/diehard/

（7） A. Rukhin, et al.: “A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications”, NIST SP 800-22 (Rev. la) (2010)

（8） IETF: “The MD5 Message-Digest Algorithm”, RFC 1321 (1992)

（9） NIST: “Secure Hash Standard (SHS)”, FIPS 180-4 (2015)

（10） T. Betchaku and S. Ichikawa: “Improvement of the URNG that utilizes weather data and LFSR”, Papers of Technical Meeting, IEE Japan, IIS-23-014 (2023)
別役拓哉・市川周一：「気象データと LFSR を用いた URNG の改良」, 電学次世代産業システム研, IIS-23-014 (2023)

（11） M. Živković: “A table of primitive binary polynomials”, Mathematics of Computation, Vol.62, No.205, pp.385–386 (1994)

（12） D.E. Knuth: “The Art of Computer Programming”, Vol.3, Addison-Wesley (1973)

（13） S. Ichikawa: “Pseudo-Random Number Generation by Staggered Sampling of LFSR”, Proc. Eleventh International Symposium on Computing and Networking (CANDAR 2023), pp.134–140 (2023)

（14） R.G. Brown, D. Eddelbuettel, and D. Bauer: “Dieharder: a random number test suite version 3.31.1”, https://webhome.phy.duke.edu/˜rgb/General/dieharder.php

別役拓哉（非会員） 2023 年豊橋技術科学大学電気・電子情報工学課程卒業。同年，同大学大学院電気・電子情報工学専攻博士前期課程入学。

市川周一（上級会員） 1985 年東京大学理学部卒業。1987 年東京大学大学院理学系研究科修士課程修了。1987 年新技術事業団，1991 年三菱電機（株），1994 年名古屋大学工学部助手。1997 年豊橋技術科学大学工学部講師。同助教授，准教授を経て，2011 年沼津工業高等専門学校制御情報工学科教授。2012 年より豊橋技術科学大学大学院工学研究科教授。現在に至る。理学博士。IEEE (senior member)，電子情報通信学会（シニア会員），ACM，情報処理学会，各会員。