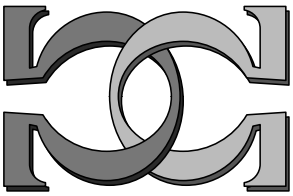

CDMTCS Research Report Series

Non-uniformity in the Quantis Random Number Generator

A. A. Abbott[1], L. Bienvenu[2], G. Senno[3]

[1]ENS, Paris, France & University of Auckland, NZ
[2]Université Paris 7, France,
[3]Universidad de Buenos Aires, Argentina

CDMTCS-472
November 2014

Centre for Discrete Mathematics and Theoretical Computer Science

Non-uniformity in the Quantis Random Number Generator

Alastair A. Abbott*, Laurent Bienvenu† and Gabriel Senno‡

November 20, 2014

Abstract

In this short report we present a detailed analysis of long sequences of bits produced by a Quantis quantum random number generator. We find that the output is slightly biased, and nearby bits are partially correlated with a correlation length of 2 bits. We briefly discuss possible physical origins for this, as well as possible normalisation techniques that can help correct for this 'on the fly', as opposed to post-processing produced bits in bulk. The non-uniformity found is small, but it is nonetheless important to understand and characterise this given the recent growth in importance of quantum random number generators.

1 Background

In recent years, there have been many proposals for quantum random number generators (QRNGs) [13, 14, 10, 4, 15], as well as several commercial devices produced [5, 9, 11]. These devices attempt to make use of quantum indeterminism to produce better quality random numbers than possible with classical, generally pseudorandom, devices. Beyond the level of unpredictability provided by indeterminism, it is important that QRNGs are also able to produce data with the correct statistical properties. In particular the output bits should be stable, independent and uniformly distributed. While pseudorandom number generators are entirely deterministic, they are carefully designed to give the correct statistics. QRNGs, on the other hand, are by their very nature prone to experimental imperfections which can lead to non-uniformly distributed bits. For this reason, normalisation and randomness extraction techniques are important in QRN generation, and their use needs to be weighed against the need to high bit-rate streams of random bits [1].

Previous studies have shown that statistical tests can be used to distinguish between classical and quantum random number generators [3], but did not look in detail at the non-uniformity of the quantum bit-sequences generated.

*Centre Cavaillès, République des Savoirs, USR 3608 CNRS, Collège de France & Ecole Normale Supérieure, Paris, France; Department of Computer Science, University of Auckland, New Zealand

†LIAFA, CNRS & Université Paris 7, France

‡Departamento de Computación, FCEN, Universidad de Buenos Aires, Argentina

In this short note we provide such an analysis, and present the results performed on bits generated from a Quantis QRNG [5] which the data "fails"—i.e. appears non-uniformly distributed. Specifically, we identify bias and non-independence between bits that should not be present in data sampled from a uniform distribution. While any physical device is bound to have imperfections at some level, we believe a more effective normalisation technique could significantly improve the quality of the output bits without the need for complicated post-processing randomness extraction.

2 Technical details

We used a sample of 10×2^{32} ($\approx 40 \times 10^9$) bits generated with a Quantis-PCI-1 device (purchased in 2004, bits generated in 2009).[1] We verified that the device was not behaving erroneously by running the NIST battery of statistical tests [12] on 1000 sequences of 1×10^6 bits and the DIEHARD tests [7] on a collection of 1×10^9 bits as described in the Randomness Test Report available online;[2] in both cases all tests were passed.

3 Analysis

Rather than the batteries of statistical tests used by the NIST and DIEHARD suites on shorter sequences, we ran some simple tests on the longer sequence of bits we obtained to see if we could detect any deviation from the expected uniform distribution.

3.1 Frequency count

Initially we ran a simple frequency test on the collected data (with $n = 10 \times 2^{32}$). Here we count the number of σs that the observed frequency count is from the $n/2$ expected for a uniform distribution ($\sigma = \sqrt{n}/2$), i.e.

$$\frac{f(0) - n/2}{\sigma},$$

where $f(0)$ is the number of 0s in the sequence.

Table 1: Frequency counts in un-normalised sequence under assumption of uniform distribution.

Bit	Count	Num. σs from expected mean	p-value
0	21473947676	-8.577407939	$\approx 9.7 \times 10^{-18}$
1	21475725284		

Performing a χ^2 test, we see that the probability of such a deviation from uniformity is $p \approx 9.7 \times 10^{-18}$. It is clear from this that the device is slightly biased towards outputting 1. This was confirmed by running some of the simple NIST tests on the full set of collected bits: when blocks larger than 400 million bits were used the bits began to fail the tests.

[1] This is the same data used in [3].

[2] `http://www.idquantique.com/images/stories/PDF/quantis-random-generator/quantis-test.pdf`

3.2 Non-independence

If this non-uniformity were simply due to bias, applying von Neumann normalisation (replacing 01 and 10 with 0 and 1, respectively, and deleting blocks of 00 and 11; [17]) should unbias the output. To test this we applied von Neumann normalisation and found, in the normalised output sequence, that while the frequency of 0s and 1s were consistent with a uniform distribution, the frequencies of 2-bit blocks (00, 01, 10, 11) were not—the probability that such a deviation from the expected counts would be obtained from a uniform distribution is $p \approx 0.00025$. This suggests the output is not independently distributed, although this could also be the result of sampling from a non-identically distributed source.

Table 2: Frequency counts in normalised sequence under assumption of uniform distribution.

Bit	Count	p-value
0	5369535132	0.1847
1	5369397686	
00	1342477466	0.00025
01	1342294455	
10	1342285745	
11	1342408742	

Looking once more at the un-normalised output, we examined the conditional distributions in order to confirm non-independence. These results are presented in Table 3, where we assume the bits are produced by a Bernoulli distribution (i.e. independent and identically distributed) with the bias calculated empirically from the frequency counts in Table 1. The statistic calculated (number of σs from the expected counts) is for the distribution of the *last* bit in the block conditioned on the *first* bit. Thus, we consider four conditional distributions: $p(x_i|x_{i-1} = 0)$, $p(x_i|x_{i-1} = 1)$, $p(x_i|x_{i-2} = 0)$, and $p(x_i|x_{i-2} = 1)$.

From these data we can examine directly the 1- and 2-bit condition distributions, i.e. $p(a|0)$ and $p(a|1)$ for $a \in \{0, 1, 00, 01, 10, 11\}$. One can see that after a 0, the device becomes much more biased towards producing a 1, whereas after a 1 the bias shifts towards producing a 0. In other words, there is a preference to alternate outcomes on top of the underlying bias. Further, from the 3-bit block data it appears that this non-independence extends for two-bits: the counts are much closer to those expected, but the the probability of such a deviation is still only $p \approx 0.013$.

To verify this hypothesis further, and to confirm the extent of the non-independence, we shuffled the data as follows: the original sequence $x_0x_1 \ldots x_{n-1}$ (we assume n is a multiple of 3; if not, the final 1 or 2 bits are ignored to make this the case) was transformed to the sequence $x_0x_3x_6 \ldots x_{n-3}x_1x_4 \ldots x_{n-2}x_2x_5 \ldots x_{n-1}$. If the non-independence is limited to 2 adjacent bits, then adjacent bits in the shuffled sequence should be independent and applying von Neumann normalisation to this sequence should give a sequence that appears normally distributed. Indeed, looking again at the condition distributions it seems that this normalisation removes the bias and non-independence:

A χ^2 test showed the probability of a deviation at least as large as this is $p = 0.34$; for k-bit blocks with $k \leq 5$ the shuffled sequence thus appeared uniform under these tests.

Table 3: Statistics for 2- and 3-bit blocks in un-normalised data under assumption of biased Bernoulli distribution.

Distribution	Bits	Count	Num. σs from expected mean
$p(x_i\|x_{i-1}=0)$	00	5367507429	-15.28007247
	01	5369535132	
$p(x_i\|x_{i-1}=1)$	10	5369397686	13.95313768
	11	5368396233	
$p(x_i\|x_{i-2}=0)$	000	1788926275	2.47870286
	010	1789995059	
	001	1789450366	
	011	1789557512	
$p(x_i\|x_{i-2}=1)$	100	1789504445	-1.965655696
	110	1789578477	
	101	1790133116	
	111	1789412400	

Table 4: Statistics for 2-bit blocks in shuffled, normalised data under assumption of uniform distribution.

Distribution	Bits	Count	Num. σs from expected mean
$p(x_i\|x_{i-1}=0)$	00	1342241891	1.756225097
	01	1342150899	
$p(x_i\|x_{i-1}=1)$	10	1342188938	0.304376517
	11	1342173168	

Performing the same analysis on the sequence $x_0x_2\ldots x_{n-2}x_1x_3\ldots x_{n-1}$ gave strong evidence the dependency extends further than to adjacent bits: while there wasn't evidence that the normalised sequence was not uniformly distributed, a χ^2 test under the assumption of a Bernoulli source indicates adjacent bits in the un-normalised sequence were not independent with a p value of $p \approx 0.00156544$. Thus, it appears the probability distribution for each bit depends on the previous two bits in the sequence.

The reason that the normalised sequence appeared uniform in this case is likely a combination of two-factors: firstly, the normalisation sufficiently decreased the length of the sequence by discarding bits that the non-uniformity could no longer be detected; secondly, the non-independence appears to affect the blocks 01 and 10 by roughly the same amount, so non-uniformity shows only in the correlations in the normalised sequence, and more bits are needed in order to detect this. However, with this non-independence present, normalisation necessarily would not remain as effective for longer sequences.

4 Interpretation and proposed normalisation

The output of the Quantis device has already been passed through an unbiasing procedure. The technique apparently used [6] consists simply of switching the labelling of the detectors as 0 or 1 after each detection. This is equivalent to xor-ing the raw output of the device $x_1x_2x_3\ldots$ with the sequence $010101\ldots$

This technique does not increase the entropy nor the algorithmic complexity of bit-sequences generated with the device. Rather, it simply changes bias for correlation, which does not amount to an effective normalisation or randomness extraction method. While this process should remove any global bias at the 1-bit level as long the probability of detection at one of the two detectors is independent of its labelling as 0 or 1 (which seems very reasonable), the bias should remain if one looked only at bits in odd (or even) positions in the generated sequence.

It is curious to note that we did not find this to be the case at all. There is no difference in distributions at odd and even positions in the generated data, and while this could be due to the output not directly reflecting the raw output (e.g. occasional bits being missed), the clear bias we found is more curious. One possibility is that the earlier model of the Quantis device we used actually made use of an alternative normalisation procedure, such as von Neumann's technique or an iterated version thereof [8], instead.

Alongside the bias we observed, we also observed clear evidence of non-independence. The most obvious possible cause of this is detector dead-time [13], resulting from a detector becoming inactive for a short time period after detection and leading to correlations between nearby bits. We observed such a correlation of length 2 bits, which would correspond to at least 8 bits of correlation in the raw data if von Neumann normalisation was used (in which case bias itself would indicate correlation in the un-normalised data).

However, this description of dead-time should lead to a slight tendency to observe 'change' in the raw output (a bias towards 01 and 10, i.e. changing detectors), which would in turn result in a tendency to 'stay the same' (a bias towards 00 and 11) in the processed output, whether it be via Quantis' procedure of changing detector labelings, or von Neumann normalisation. We observed exactly the opposite: a bias towards change in the processed output of the device. Unfortunately we do not have a sufficiently detailed understanding of the operation of the device to further speculate on the cause of this dependence, but it would be interesting to understand this properly. There are several other possibilities beyond dead-time that could affect the quality of the output of the Quantis device [1].

While the output can be unbiased by post-processing—either by shuffling and normalising as discussed in the previous section, or by using seed-based randomness extractors as allowed for in the most recent version of the Quantis software [16]—the anomalies described above should also be able to be removed at the time of generation by using a better unbiasing technique. Of course, one cannot hope to extract more uniform bits than the entropy of output sequence, and this is what such post-processing approaches aim to do. However, in many applications there can be benefit to unbiasing the data on the fly, improving the quality of the bit-stream without post-processing.

With the length of the dependence known, one can use normalisation techniques which are as efficient as von Neumann normalisation, but which will work even in the presence of dependence. While this can be done with Markov-chain models [2], the 'shuffling' technique described in the previous section is simpler yet and could easily be implemented on the fly by

buffering $2k$ bits, where k is the length of the dependence, shuffling and then applying von Neumann normalisation. Furthermore, by increasing the buffer length and using an iterated version of the procedure, the entropy bound can be approached, improving the efficiency. This would greatly increase the quality of the output with minimal effort and no significant loss in efficiency, while lessening the need for more costly post-processing based randomness-extraction.

5 Conclusion

We analysed in detail long sequence of bits generated by a Quantis quantum random number generator. We found that the sequence showed clear signs of bias and systematic correlation between nearby bits, despite of normalisation applied by the QRNG. While this non-uniformity was not large, it is important to understand well the statistical quality of bits produced by such devices. Understanding this can help both to design of more robust QRNGs and better normalisation techniques that can provide 'on the fly' normalisation to correct for systematic non-uniformity of a physical origin. We briefly mentioned some such possibilities for improving normalisation.

Acknowledgment

We acknowledge C. S. Calude and M. J. Dinneen for providing access to the Quantis bits from [3].

References

[1] Alastair A. Abbott, Cristian S. Calude, and Karl Svozil. A quantum random number generator certified by value indefiniteness. *Mathematical Structures in Computer Science*, 24(3):e240303, 2014.

[2] M. Blum. Independent unbiased coin flips from a correlated biased source—a finite state Markov chain. *Combinatorica*, 6(2):97–108, 1986.

[3] C. S. Calude, M. J. Dinneen, M. Dumitrescu, and K. Svozil. Experimental evidence of quantum randomness incomputability. *Physical Review A*, 82(022102), 2010.

[4] Martin Fürst, Henning Weier, Sebastian Nauerth, Davide G. Marangon, Christian Kurtsiefer, and Harald Weinfurter. High speed optical quantum random number generation. *Optics Express*, 18(12):13029–13037, 2010.

[5] ID Quantique. *QUANTIS. Quantum number generator.* `http://www.idquantique.com/random-number-generators/products/products-overview.html`.

[6] ID Quantique. Private communication., 18/11/2014.

[7] G. Marsaglia and A. Zaman. Towards a universal random number generator. *Statistics & Probability Letters*, 9(1):35–39, 1990. `http://www.stat.fsu.edu/pub/diehard/`.

[8] Y. Peres. Iterating von Neumann's procedure for extracting random bits. *The Annals of Statistics*, 20(1):590–597, 1992.

[9] PicoQuant. *Quantum Random Number Generator 'PQRNG 150'*. `http://www.picoquant.com/products/pqrng150/pqrng150.htm`.

[10] S. Pironio, A. Acín, S. Massar, A. Boyer de la Giroday, D. N. Matsukevich, P. Maunz, S. Olmchenk, D. Hayes, L. Luo, T. A. Manning, and C. Monroe. Random numbers certified by Bell's Theorem. *Nature*, 464(09008), 2010.

[11] qutools. *quRNG – Quantum Random Number Generator*. `http://www.qutools.com/products/quRNG/index.php`.

[12] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Special Publication 800-22, NIST, 2010.

[13] A. Stefanov, N. Gisin, O. Guinnard, L. Guinnard, and H. Zbinden. Optical quantum random number generator. *Journal of Modern Optics*, 47(4):595–598, 2000.

[14] M. Stipčević and B. M. Rogina. Quantum random number generator based on photonic emission in semiconductors. *Review of Scientific Instruments*, 78(4):045104, 2007.

[15] T. Symul, S. M. Assad, and P. K. Lam. Real time demonstration of high bitrate quantum random number generation with coherent laser light. *Applied Physics Letters*, 98(231103), 2011.

[16] M. Troyer and R. Renner. A randomness extractor for the Quantis device. Technical paper on randomness extractor, ID Quantique, September 2012. `http://www.idquantique.com/images/stories/PDF/quantis-random-generator/quantis-rndextract-techpaper.pdf`.

[17] J. von Neumann. Various techniques used in connection with random digits. *National Bureau of Standards Applied Math Series*, **12** (1951), 36–38. In A. H. Traub, editor, *John von Neumann, Collected Works*, pages 768–770. MacMillan, New York, 1963.