# True random number generators with flicker noise: stochastic model, min-entropy calculation and online test

**Rok Zitko**
rok.zitko@ijs.si

Jožef Stefan Institute

**Research Article**

**Keywords:** tochastic model, flicker noise, Gaussian processes, autocorrelations, min-entropy, online health tests

**Posted Date:** February 23rd, 2024

**DOI:** https://doi.org/10.21203/rs.3.rs-3970536/v1

**Additional Declarations:** No competing interests reported.

# True random number generators with flicker noise: stochastic model, min-entropy calculation and online test

Rok Žitko[1,2*]

[1*]Department of theoretical physics, Jožef Stefan Institute, Jamova 39, Ljubljana, SI-1000, Slovenia.

[2]Faculty of mathematics and physics, University of Ljubljana, Jadranska 19, Ljubljana, SI-1000, Slovenia.

Corresponding author(s). E-mail(s): rok.zitko@ijs.si;

**Abstract**

Flicker (pink, $1/f$) noise is ubiquitous in all electronic devices, including in oscillator circuits used in true random number generators (TRNGs) based on jitter. Flicker noise produces strong serial correlations with very slow decay in time. We present a stochastic model for counter-mode TRNGs that takes the effects of such time correlations into account. Key parameters in the model are the spectral strengths of the pink and white noise components, as well as the low-frequency cutoff for the flicker noise spectrum. The random bits are defined as the least significant bit of consecutive integer-valued count numbers. We present the dependence of autocorrelations and min-entropy of generated random bits on model parameters. The autocorrelation between the bits is suppressed by increasing the strength of either pink or white noise, but it remains long-ranged (power-law decay). The power-law exponent depends linearly on the strength of pink noise, while the prefactor depends exponentially on both strengths. We determine the min-entropy per bit from a careful analysis of long-time sequences. It approaches the value of 1 approximately as a stretched exponential function of the flicker noise strength: highly entropic random bit generators can thus be designed even in the presence of strong flicker noise. We also propose an effective and efficient online health test for generators of this type.

**Keywords:** stochastic model, flicker noise, Gaussian processes, autocorrelations, min-entropy, online health tests

## 1 Introduction

True random number generators (TRNGs) are devices that provide random bits with guaranteed entropy properties based on a well-understood physical process that is known to be random, such as the measurement process in a quantum system or the sampling of a thermally fluctuating system [1–4]. The modern approach to designing a TRNG is based on establishing a stochastic model, i.e., a mathematical description of the noise source with random variables, which allows determining the lower bound of the min-entropy of generated numbers [5, 6]. This is the required [7, 8] or recommended [9] practice in standards for random number generation for cryptographic applications. Setting up a stochastic model requires good understanding of the working principle of the noise source. Mathematically, a stochastic model takes the form of a random process or a distribution function depending on a set of parameters, such that the set of distributions encompasses the actual distribution in the real system, including the cases when the quality of random numbers is degraded [5]. The parameters of the stochastic model for an actual TRNG device are measured experimentally. Stochastic models are also the basis for designing effective and efficient online health tests.

In counting-mode oscillator TRNGs, one oscillator is used to generate the slow clock for a circuit that counts the periods of a second oscillator, using the least-significant bit (LSB) of the total count number in each period of the first oscillator as the random bit [10–12]. The simplest oscillators that can be implemented in CMOS technology or using programmable logic (FPGA) are elementary ring oscillators (EROs), chains of an odd number of NOT gates [13, 14]. The randomness in such rings is due to jitter, i.e., the phase noise of the oscillation [15]. One often distinguishes deterministic and random jitter, as well as global and local jitter. The deterministic jitter is due to predictable changes in the circuit (e.g. due to oscillations in other parts of the integrated circuit, oscillating voltage in the power lines, RF irradiation of the chip, etc.), while the random jitter is unpredictable and ultimately due to thermal or quantum fluctuations (e.g. Johnson-Nyquist noise of agitating charge carriers in electric conductors, electron quantum tunneling events, etc.). Global jitter affects a large area of the chip, while the local jitter is due to local processes within each oscillator. Ideally, only local random jitter would be used for random number generation. To filter out the global and deterministic noises, one typically uses differential configurations, where two nominally identical EROs are located in immediate vicinity on the chip [12], with the slow clock generated by frequency division of one of the two periodic signals. It is often assumed that the dominant contribution to jitter is the Johnson-Nyquist noise produced solely within the oscillators and that this noise is white (i.e., it has a spectrum that does not depend on the frequency). In such case the random processes are not correlated in time, so that a stochastic model in terms of independent and identically distributed Gaussian variables can be easily constructed. The correlation length between the count numbers is finite: only pairs of consecutive numbers are (anti)correlated because of the events close to the end time of each counting period, which can fall on either side of the time boundary. The min-entropy can thus be easily computed from the conditional probabilities on blocks of two random bits.

In reality, no oscillator has purely white noise spectrum down to zero frequency and eventually the flicker (pink) noise with the $1/f$ spectrum takes over [16, 17]. The $1/f$ noise is ubiquitous in all metals and semiconductors [18, 19] and by now relatively well understood, with the exception of the actual microscopic processes causing it in different materials and devices: there seems to be no universality as concerns the physical mechanisms [19]. Given that the $1/f$ noise is unavoidable, this begs the question about its effects on the security of TRNG designs, especially given that flicker noise is expected to become increasingly relevant in future due to shrinking transistor sizes [17]. The $1/f$ noise is namely strongly correlated: the autocorrelation function is logarithmic and practically never decays to zero on the time-scale of random number generator operation [20]. We show an example of value distribution, autocorrelations and a periodogram from an actual FPGA implementation in Fig. 1, evidencing strong flicker noise effects. Furthermore, strictly speaking flicker noise is not even stationary, but depends on the age of the device [20].

One could hence be led to believe that these issues represent an insurmountable obstacle for building high entropy TRNGs. In the following, we present a stochastic model for counter-mode TRNGs [12, 17] where the spectrum of count numbers has a $1/f$ (pink) behaviour at low frequencies and becomes constant (white) at high frequencies. The parameters of the model are the strengths of the pink and white noise components, as well as the low-frequency cutoff of the $1/f$ noise spectrum. The frequency cutoff is determined by some characteristic time in the generation process (e.g. related to the count accumulation time set by the period of the slow oscillator), but it is essentially an empirical parameter in the model to be established by experimental measurements. (In the following we also find that its value is actually not very important as regards the min-entropy of random numbers and the reliability of the online health test.)

We first introduce the stochastic model in full detail. We discuss the dependence of the random bit autocorrelation on model parameters,

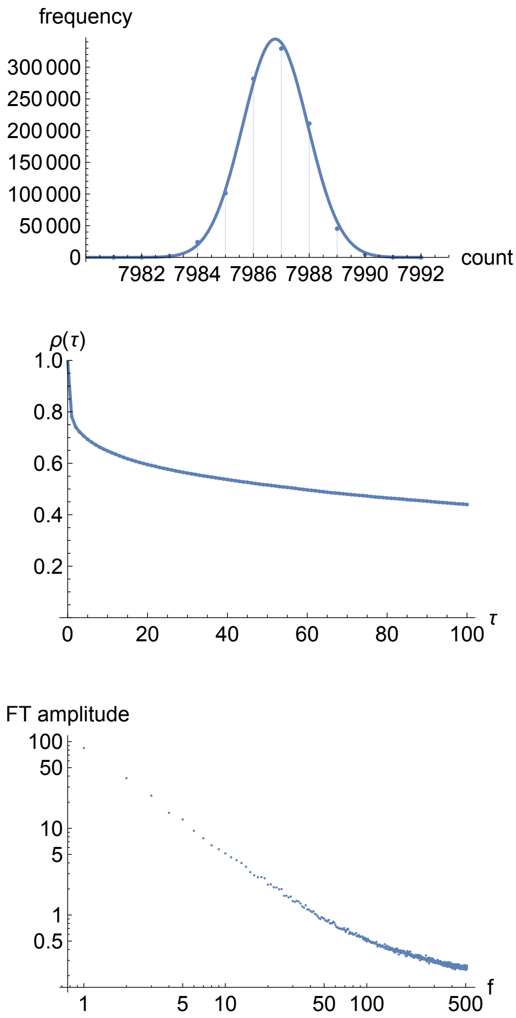

**Fig. 1** Properties of a sequence of $10^6$ count values in a true random number generator based on two length-19 elementary ring oscillators implemented in a Cyclone V SE 5CSEBA6U23I7 FPGA. We show the distribution of count values, the autocorrelation function, and a periodogram (average of size 1024 blocks). They are characteristic of a noise spectrum dominated by the flicker noise.

then proceed with the analysis of entropy production, taking special care to correctly estimate the min-entropy in the presence of long-time (power-law) correlations. We show how the autocorrelation between the bits are strongly suppressed by increasing the noise amplitudes. Importantly, even for pure flicker noise spectrum (no white noise component) one finds that increasing the amplitude is sufficient to produce highly entropic random numbers. Finally, we also present an effective and efficient on-line health test for generators of this type.

## 2 Stochastic model

The device that we are modelling consists of two nominally identical oscillators [12]. The frequency of the first is divided by $M$ to generate the slow clock. During each period of the slow clock, we count the number of zero to one (raising signal) transitions of the second clock; the number of these fast clock ticks is an integer-valued random variable $X_i$. The device is sensitive to fluctuations of the relative frequency of the oscillators. For simplicity, we ascribe these fluctuations (jitter) to the second clock and assume the slow clock to be absolutely stable; this leads to no loss of generality. We measure the time in units of period of the slow clock, $\tau_s = 1$, so that the corresponding sampling frequency is $f_s = 1$, and hence the Nyquist frequency is $f_N = 1/2$. The random bits are generated by taking the LSB of $X_i$; the corresponding random variable is $R_i$.

We will not concern ourselves with the statistical properties of each individual tick of the jittered fast clock. This is difficult to measure reliably even using fast oscilloscopes or spectrum analysers with differential probes, typically leading to overestimated variance (on-chip time-to-digital converters are a possible solution for embedded in-situ characterization) [12, 17, 21]. Instead, the count numbers of a realization of the random process, $x_i$, are easy to reliably characterize, including their variance, autocorrelation, and periodogram, simply by capturing the actual values. This "event count" perspective is also in line with the standard interpretation of the $1/f$ noise as arising from counting the number of relaxation events from a large set of fluctuators with an approximately flat distribution of energies $E$, so that the relaxation times are $\tau = \tau_0 \exp(E/k_B T)$, where $1/\tau_0$ is the attempt frequency, $k_B$ the Boltzmann constant, and $T$ the temperature [19]; the key quantity in the theory is the number of events per time interval itself, not so much the times between the consecutive events.

With the aim of building a stochastic model suitable for simulations, we introduce an auxiliary real-valued random variable $Y_i$, such that its spectrum is [16, 17]

$$S_Y(f) = \frac{a_p}{f} + a_w, \tag{1}$$

where $a_p$ and $a_w$ characterize the amplitude of the pink noise and white noise, respectively. The variance $\sigma_Y^2$ is given as

$$\sigma_Y^2 = 2\int_{f_{\min}}^{f_{\max}} S_Y(f)\mathrm{d}f = A_p(f_{\min})a_p + A_w(f_{\min})a_w, \tag{2}$$

where we have introduced the low-frequency cutoff $f_{\min}$ as an important additional parameter of the model, while $f_{\max} = f_N = 1/2$ is simply the Nyquist frequency. The prefactors are $A_w(f_{\min}) = 1 - 2f_{\min} \approx 1$ and

$$A_p(f_{\min}) = 2(\ln f_{\max} - \ln f_{\min}). \tag{3}$$

Hence

$$\sigma_Y^2 \approx 2\ln\frac{f_{\max}}{f_{\min}} a_p + a_w. \tag{4}$$

It is important to notice that the variance depends not only on the amplitude of the pink noise, but also on the low-frequency cutoff $f_{\min}$. The upper corner frequency for pink noise is $a_p/a_w$; above this frequency the noise becomes white. If $a_p/a_w \gg f_N = 0.5$, the noise spectrum can be considered as purely pink.

To generate a realization of the time series $x_i$ (and bit sequence $r_i$) from $y_i$ in numerical stochastic simulations, we simply take the integer part (floor) of $y_i$:

$$x_i = \lfloor y_i \rfloor. \tag{5}$$

In stochastic simulations, the variates $y_i$ are generated using the Timmer-König (TK) algorithm for the given power spectrum $S_Y(f)$ [22]. This method simulates a Gaussian process. The correctness of Gausianness assumption is corroborated by our test devices: configurations with two EROs in differential setup on an FPGA exhibited very well-defined Gaussian distributions with hardly any outliers. (If the setup was not differential, we observed non-Gaussian distributions, sometimes with multiple peaks, and sometimes with numerous outliers.)

The distribution generated by the TK algorithm of $y_i$ is a Gaussian centered at some mean value $\mu$ with variance $\sigma_Y^2$. The value of mean $\mu$ is typically beyond control in the TRNG and possibly slowly changes with time. We will assume the worst-case scenario of $\mu$ being a half-integer: in this case the variance of $X_i$ is the lowest for a given variance of $Y_i$. (In fact, the value of $\mu$ is actually found to be unimportant in the presence of a flicker component with a sizable amplitude.) In the TK algorithm, the integration over the frequency $f$ is effectively replaced by a sum over frequencies from $f_{\min}$ to $f_{\max}$ in steps of $f_{\min}$. A better approximation for $A_p(f_{\min})$ is then

$$A_p(f_{\min}) = 2(\gamma - \ln f_{\min}). \tag{6}$$

with $\gamma \approx 0.577216$ being the Euler constant. After generating $y_i$, we then take the floor to obtain $x_i$, and take the LSB to finally obtain $r_i$.

For the purposes of this work, we have implemented a computer code for the TK algorithm to perform large-scale stochastic simulations. It is available on a public repository [23]. Since inverse fast Fourier transform (FFT) is used in the TK algorithm, we will mostly use $f_{\min} = 1/2^m$ with integer $m$, so that the window size of $2^m$ permits efficient FFT evaluation.

Empirically, the parameters $a_p$, $a_w$ and $f_{\min}$ are most easily extracted from the periodograms for count values $x_i$ by fitting the estimated power spectrum to function $S_Y(f)$ or, for more accuracy, to $S_X(f)$. This can be performed by capturing the count number data [17].

## 3 Autocorrelations

We first consider the autocorrelation function of the random bit variable $R_i$ as a function of $a_p$, $a_w$ and $f_{\min}$. Pure white noise is uncorrelated, while pink noise has a slow logarithmic decay [20]. This might immediately lead to concerns, given that the $1/f$ noise is ubiquitous and unavoidable, thus in any realistic system the autocorrelation function of $Y_i$ (and $X_i$) is expected to never truly drop to zero. In practice, however, this is not necessarily a problem. Since the random bits are taken as the LSB of variates $x_i$, with increasing variance $\sigma_Y^2$ we generally expect that the autocorrelations of $R_i$ will decrease because of stronger randomization. In fact, we find that the autocorrelation of the random bits has a power-law behavior, not logarithmic.

To explore this in more detail, we first consider the case of pure pink noise (setting $a_w \equiv 0$) at fixed lower cutoff. For $a_p = 0.05$, $f_{\min} = 2^{-15}$, so that $\sigma_Y^2 \approx 1.028$, we find that the normalized autocorrelation function decays as

$$\rho(\tau) = a\tau^{-\alpha} \tag{7}$$

with $a = 0.148$ and $\alpha = 0.987$, see Fig. 2. We collected 1000 sequences of $2^{27}$ bits each; the curve fitting was performed on the average of autocorrelation functions and we have verified that the fit quality is within the error bars of the estimated $\rho(\tau)$. The fitting is performed in the time-delay range from $\tau = 2$ to $\tau = 64$. We observed small exponential contributions that are present in $\rho(\tau)$ at short time delays; in order to improve the asymptotic fit and the extracted value of the power-law exponent we therefore included two exponential terms in the Ansatz:

$$\rho(\tau) = a\tau^{-\alpha} + \epsilon_1 \exp(-\tau/\tau_1) + \epsilon_2 \exp(-\tau/\tau_2). \tag{8}$$

Further examples for a range of $a_p$, $a_w$ and $f_{\min}$ are tabulated in Tab. 1. We find that the autocorrelation does not depend on $f_{\min}$ for $f_{\min}$ low enough (we observed that the cases of $f_{\min} = 10^{-10}$ slightly deviate from the clear power-law behavior, while for $f_{\min} \leq 10^{-15}$ deviations were imperceptible). This is an interesting observation in light of the result for the variance, Eq. (3), which depends on $f_{\min}$ logarithmically. The exponent $\alpha$ depends on $a_p$ as a linear function:

$$\alpha \approx 19.8 a_p. \tag{9}$$

At the same time, the prefactor $a$ decays exponentially:

$$a \approx 0.835 \exp(-34.7 a_p). \tag{10}$$

Thus even though the autocorrelations in principle persist indefinitely, with increasing $a_p$ they decay more quickly (with a higher power-law exponent) and furthermore their amplitude is exponentially damped. In the presence of additional white noise, we find that the decay remains of power-law type with the same value of the exponent $\alpha$, only the prefactor decreases with an additional exponential prefactor

$$a \propto \exp(-a_w/c) \tag{11}$$

with $c \approx 0.1$.

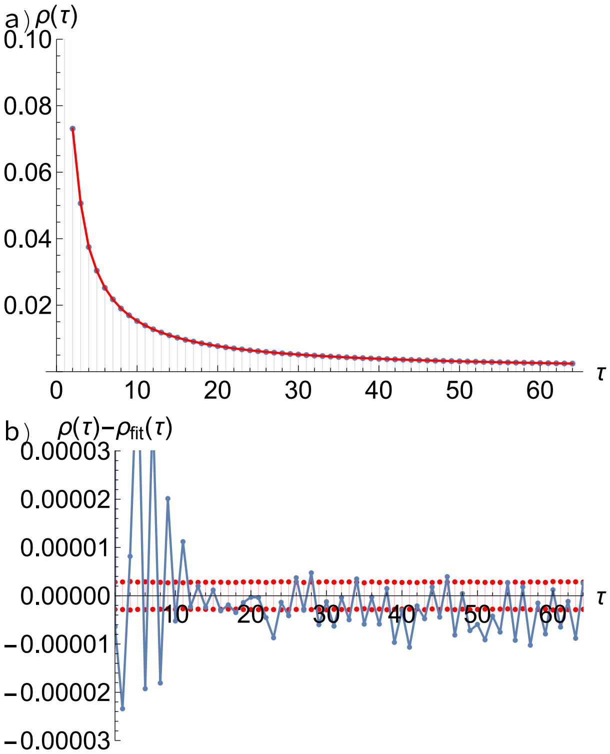

**Fig. 2** Normalized autocorrelation function $\rho(\tau)$ of the random-bit variable $R_i$. The parameters are $a_p = 0.05$, $a_w = 0$, $f_{\min} = 2^{-15}$. In the asymptotic tail for large $\tau$, the power-law fit leads to residual errors of comparable magnitude to the statistical noise.

## 4 Min-entropy calculation

Since random bits have long correlations with power-law decay, the min-entropy $H_{\min}$ should be computed using conditional probability with the condition that extends to a large number of past bit values. One should then take the probability of the most likely event, $p_{\max}$, and finally $H_{\min} = -\log_2 p_{\max}$. A simple consideration shows that for a system with a strictly positive autocorrelation function the most likely events at any finite sequence length consist of sequences of all zeros or all ones. It is thus sufficient to accumulate the statistics of such sequences for different sequence lengths, $l$. We denote by $p_0(l)$ and $p_1(l)$ the probabilities of runs of $l$ zeros and $l$ ones, respectively. We define $H_{\min}(l) = -\log_2[\max\{p_0(l), p_1(l)\}]$. If $H_{\min}(l)/l$ converges to some finite value with increasing $l$, we take that value as the estimate of

**Table 1** Autocorrelation properties

| $a_p$ | $a_w$ | $f_{\min}$ | $\sigma_Y^2$ | $a$ | $\alpha$ |
|---|---|---|---|---|---|
| 0.05 | 0 | $2^{-10}$ | 0.681 | 0.157 | 1.092 |
| 0.05 | 0 | $2^{-15}$ | 1.028 | 0.146 | 0.982 |
| 0.05 | 0 | $2^{-20}$ | 1.375 | 0.145 | 0.979 |
| 0.01 | 0 | $2^{-15}$ | 0.206 | 0.533 | 0.370 |
| 0.02 | 0 | $2^{-15}$ | 0.411 | 0.416 | 0.473 |
| 0.03 | 0 | $2^{-15}$ | 0.617 | 0.295 | 0.622 |
| 0.04 | 0 | $2^{-15}$ | 0.822 | 0.207 | 0.797 |
| 0.05 | 0 | $2^{-15}$ | 1.028 | 0.146 | 0.982 |
| 0.06 | 0 | $2^{-15}$ | 1.234 | 0.103 | 1.172 |
| 0.07 | 0 | $2^{-15}$ | 1.440 | 0.0723 | 1.360 |
| 0.1 | 0 | $2^{-15}$ | 2.06 | 0.025 | 1.913 |
| 0.04 | 0 | $2^{-20}$ | 1.100 | 0.206 | 0.786 |
| 0.05 | 0 | $2^{-20}$ | 1.375 | 0.145 | 0.979 |
| 0.05 | 0 | $2^{-15}$ | 1.028 | 0.146 | 0.982 |
| 0.05 | 0.025 | $2^{-15}$ | 1.053 | 0.114 | 0.982 |
| 0.05 | 0.05 | $2^{-15}$ | 1.078 | 0.0890 | 0.982 |
| 0.05 | 0.075 | $2^{-15}$ | 1.103 | 0.0696 | 0.982 |
| 0.05 | 0.1 | $2^{-15}$ | 1.128 | 0.0543 | 0.982 |
| 0.05 | 0.15 | $2^{-15}$ | 1.178 | 0.0337 | 0.987 |

min-entropy per bit of the random process, $H^*$:

$$H^* = \lim_{l\to\infty} \frac{H_{\min}(l)}{l}. \tag{12}$$

It is thus important to study the asymptotic properties of $H_{\min}(l)/l$.

Using simulations, we have established that it is possible to fit $H_{\min}(l)/l$ of a process with pure pink noise with a power-law function:

$$\frac{H_{\min}(l)}{l} = h\left(1+\frac{b}{l^\beta}\right). \tag{13}$$

We used $10^{11}$ samples in each run, and averaged over 100 such runs in order to obtain results with very low statistical noise (on the order of $10^{-4}$ for $l = 35$). We find that $b$ and $\beta$ depend slightly on the interval on which the fit is performed, while $h$ is more robust. For example, comparing $l \in [5 : 35]$ and $l \in [15 : 35]$ for one particular case, we found $h$ that differ by 3 per-mil.

For $a_p = 0.05$, $a_w = 0$ and $f_{\min} = 10^{-15}$, the case considered already in the previous section, we find $h = 0.633$, $b = 0.685$, $\beta = 0.683$, see Fig. 3. From this we conclude that the min-entropy per bit of this sequence is 0.633. Note that at this value of the exponent $\beta$ it takes $l \approx 570$ to converge within one percent of the asymptotic value, and $l \approx 17600$ to get within one per-mil. This demonstrates the importance of extrapolating to large values of $l$ in order to reliably assess the min-entropy, especially since $b$ is positive, hence the asymptotic value is approached from above (i.e., with a tendency for overestimation). For this value of $a_p$, the min-entropy $h$ depends little on $f_{\min}$. For $f_{\min} = 2^{-20}$, $f_{\min} = 2^{-15}$ and $f_{\min} = 2^{-10}$ we find the same result for $h$ within one per-mil.

With increasing noise amplitude, the autocorrelations decay faster, and $H_{\min}(l)/l$ values tend towards 1. In these cases, the extrapolation is no longer necessary (and, in fact, becomes mathematically ill-posed). Instead, we can reliably extract the asymptotic value $H^*$ from shorter sequences (e.g. up to 8) by directly accumulating the statistics for all bit sequences. This is a reliable procedure for $H^* \gtrsim 0.99$. We find that it is possible to extract min-entropy within one percent absolute error and one percent relative error.

In the presence of white noise, we find that $h$ increases, but the functional form of $H_{\min}(l)/l$ is no longer well approximated by a pure power-law function. In fact, when both components are present and sizable, we were unable to reliably extract the asymptotic value of $h$. The reason is the limited range of $l$ values that are accessible in stochastic simulations; in the absence of a known analytical expression for this general case reliable fits do not appear possible. Solving this issue remains an open challenge for future work. The problem is, however, rather academic and concerns only cases of relatively low min-entropy. For large enough noise amplitudes, i.e., for $H_{\min}$ above 0.98 per bit, it is sufficient for all practical purposes to extract $H_{\min}$ by direct computation on short sequences (order 8).

At fixed $a_p$, decreasing $f_{\min}$ leads to slightly higher min-entropy. This may at first seem surprising, since decreasing $f_{\min}$ suggests enhanced long-time correlations, but the effect is actually quite expected, because we are enabling additional fluctuating modes, thus overall randomness can only be increased. The effect is, however, rather weak.

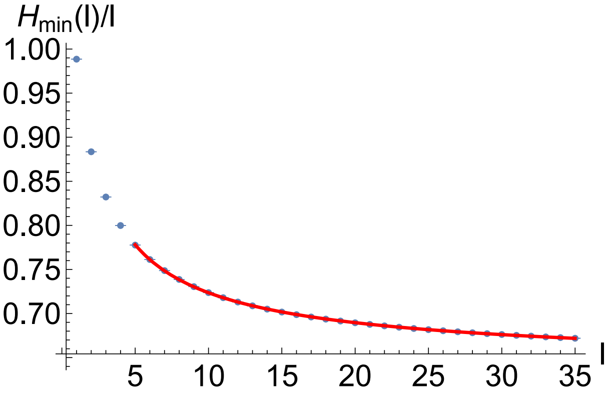

**Fig. 3** Min-entropy per bit estimates $H_{\min}(l)/l$ (blue symbols). The parameters for the stochastic simulation are $a_p = 0.05$, $a_w = 0$, $f_{\min} = 10^{-15}$. The red curve is a $h\left(1 + b/l^{\beta}\right)$ fit, which here gives $h = 0.633$, $b = 0.684$, $\beta = 0.683$. The fit is performed in the interval $l \in [5 : 35]$.

## 5 Min-entropy parameter dependence

Having established the general methodology, we now systematically study the parameter dependence of $H^* = \lim_{l\to\infty} H_{\min}(l)/l$ as a function of $a_p$, in the absence of any white noise by setting $a_w = 0$. We fix $f_{\min} = 10^{-15}$ in this section, since taking a lower value modifies the results less then by one percent which is our target precision. The results are shown in Fig. 4, where we show both a finite-length result for $l = 8$ as well as the asymptotic value. We note that taking the $l = 8$ estimated leads to a severe overestimation of the true min-entropy for low $a_p$; the ratio of the two values is found to be diverging as $a_p \to 0$.

The attempts to fit $H^*$ vs. $a_p$ curves with various well-known functional shapes were not fully satisfactory; the best result for $a_p \in [0.01 : 0.16]$ (corresponding to $H^*$ approximately between 0.1 and 0.99) were obtained for a stretched exponential function:

$$H^*(a_p) \approx 1 - \exp\left[-(a_p/\lambda)^k\right] \tag{14}$$

with $\lambda = 0.050$ and $k = 1.35$.

## 6 Statistical testing

It is of some practical interest to investigate the discerning power of commonly used standard batteries of statistical tests for random number

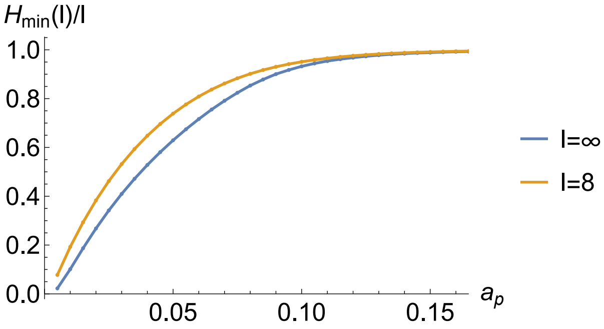

**Fig. 4** Min-entropy per bit for pure flicker noise model as a function of $a_p$. We show the min-entropy estimated from length $l = 8$ sequences and the min-entropy determined from extrapolating to infinitely long sequences.

generators to detect the correlations present in the case of pure pink noise jitter. We find that all common tests batteries indeed pass for sufficiently large values of $a_p$, but some are more sensitive to the particular type of autocorrelations present in our bit streams. For practical purposes of testing real-life devices it is also useful to identify the specific tests that detect the actual deficiency. In all stochastic simulations performed for purposes of these tests we have used simulation parameters $f_{\min} = 10^{-15}$ and $\mu = 0.5$.

For `dieharder`, we find that $a_p$ needs to be larger than 0.3 to pass all tests. The first tests to fail for too low $a_p$ are `sts_runs`, `sts_serial`, `dab_bytedistrib` and `dab_filltree2`. (We have used `dieharder` version 3.31.1, with command line switches `-a -g 200 -Y 1 -k 2`. )

`TestU01` test battery `alphabit` passes for $a_p \gtrsim 0.21$ for 10 MB sizes (the first test to fail at lower $a_p$ is `MultinomialBitsOver`) and for $a_p \gtrsim 0.28$ for 1 GB sizes (same test). The test battery `rabbit` test passes for $a_p \gtrsim 0.21$ for 10 MB sizes (first tests to fail is `AutoCor`), and for $a_p \gtrsim 0.28$ for 1 GB sizes (first test to fail are `AutoCor` and `RunOfBits`). The `Crush` test passes for $a_p \gtrsim 0.36$ (typical failing tests at lower values are `BirthdaySpacings, t = 4`, `CollisionOver, t = 20`, `RandomWalk1 H (L = 10000)` and `AutoCor, d = 1`). It is interesting to observe that the batteries designed for testing hardware generators, such as `alphabit` and `rabbit` start failing at much lower values of $a_p$ than the batteries typically used to test pseudo-random generators, such as `crush`. This points in the direction that some tests are actually sensitive to slight deficiencies of the simulation procedure. (We have used `TestU01` version 1.2.3 with the default settings. For `alphabit` we used $r = 0$, $s = 32$.)

`PractRand` tests are performed for sequences of increasing length (factors of 2). We find that the length where the first failure is reported depends on $a_p$. At $a_p = 0.3$ and $a_p = 0.4$ a failure is usually detected at size 64 GB (failing test is `TMFn(2+0):wl`), for $a_p = 0.5$ at 256 GB (same test failing), at $a_p = 0.6$ at 512 GB (same test, as well as `FPF-14+6/16:all`, at $a_p = 0.7$ at 1 TB (`TMFn(2+0):wl`, `FPF-14+6/16:all` and `FPF-14+6/16:(0,14-0)` failing). Again, we speculate that these failures at large values of $a_p$ are actually due to `PractRand` detecting nonidealities due to the generation process itself: we generate random numbers in non-overlapping blocks determined by the block size in Timmer-Könnig algorithm (which is itself fixed by $f_{\min}$). If the block size is commensurate with the sampling size in some test, this could be detected as a non-random pattern. To test this hypothesis, we increased the block size from $2^{15}$ to $2^{20}$. This indeed pushed the size where first failures were reported to higher values (by a factor or 2 or 4). Similar effect is found by using a block size which is a product of powers of different primes, e.g., $2^7 \cdot 3^5 = 31104$, which is close in value to $2^{15}$ but nevertheless is effective in pushing up the data size before first failure is detected. (We have used `PractRand` version 0.95 with the default parameters, i.e., standard battery of tests and the default threshold setting `-e 0.1`.)

Finally, we compare NIST entropy assessment [24] against our min-entropy calculations, see Fig. 5. The results show the well-known tendency towards underestimation of min-entropy by the NIST tests at large values of min-entropy. More peculiar are the results for low $a_p$, where the NIST test slightly overestimates the min-entropy. We speculate that this is a finite-size effect (data sets were 10 MB in size).

## 7 Online test

The main failure mechanism in oscillator-based TRNGs is an insufficient amount of jitter in clock frequency or a total failure (no oscillation). A concern is also oscillator frequency locking, where multiple oscillators synchronize in phase, although this is mainly an issue in designs with a large number of oscillator rings on the same chip. Online tests must be able to detect non-tolerable entropy deficiencies in as little time as possible (so that no compromised random bits are ever presented to the user) and should be tailored to the stochastic model. For entropy sources based on oscillator circuits in counter mode it is easy to implement embedded online tests. In particular, to target the main failure mechanism (lack of jitter) one can implement variance measurement of the consecutive integer count values $x_i$ [17, 25]. We thus consider the quantity [12]

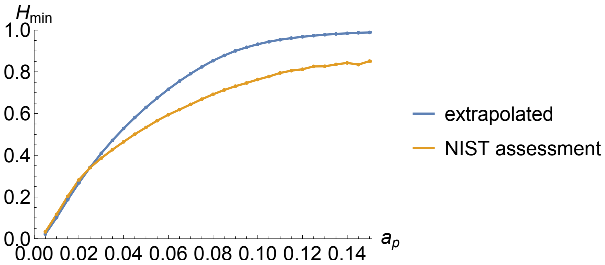

**Fig. 5** Min-entropy per bit for pure flicker noise model as a function of $a_p$: out asymptotic estimate vs. NIST entropy assessment result.

$$\sigma_R^2 = \frac{1}{2(N-1)} \sum_{i=2}^{N} \frac{1}{2} \left(x_i - x_{i-1}\right)^2 , \qquad (15)$$

which is essentially an estimator for the 2-sample Allan variance with equal time between measurements and measurement time, $T = \tau$. It is easy to compute in real-time in digital circuits [12]. This quantity can be used to directly quantify the randomness properties of the oscillator. It is less sensitive to long-time correlations compared to the regular variance, because only neighboring value pairs enter the expression. Importantly, it is not affected by the value of $f_{\min}$, unlike the standard variance [see Eqs. (2) and (3)]. We have established that the min entropy is not affected by $f_{\min}$, provided that $f_{\min}$ is small enough, thus the online test must be based on a quantity which also has this property. Using the variance defined as $\sigma_R^2$ is thus crucial. In Fig. 6 we plot the min-entropy as a function of $\sigma_R^2$ for the case of pure pink noise.

Let us now set the goal of achieving $H_{\min} \geq 0.98$. We find that this requires $\sigma_R^2 > 0.52$ (and corresponds to $a_p = 0.132$). This defines the

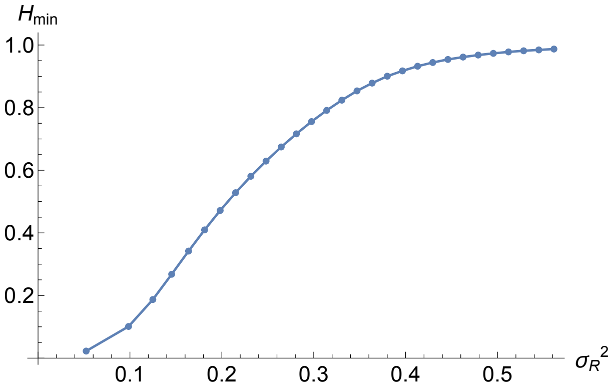

**Fig. 6** Min-entropy per bit as a function of $\sigma_R^2$ for pure pink noise.

region $A_{\text{good}}$ in the parameter space, while the complementary set is denoted $A_{\text{bad}}$. The region $A_{\text{real}} \subset A_{\text{good}}$, that corresponds to $H_{\min} \geq 0.995$, is defined by $\sigma_R^2 > 0.639$ (and corresponds to $a_p = 0.169$). The alarm is raised for $\sigma_R^2 < 0.580$. We performed stochastic simulation to determine the probability for false alarms (false positives, i.e., RNG in $A_{\text{real}}$ triggering alarm) and silent failures (false negatives, i.e., RNG in $A_{\text{bad}}$ not triggering alarm) for different test sample sizes $N$, see Table 2. These two probabilities should be low in order to guarantee both availability and security. We see that the probabilities decrease by an order of magnitude for each increase of size of 1000. For $N = 10000$ we thus expect false positives/negatives at a rate lower than $10^{-10}$, which for RNG rates on the scale of Mbit/s corresponds to a few events per day. Raising the value to $N = 20000$ would make such events practically non-existent in the typical life-time of the device, baring an actual degradation of the noise properties.

**Table 2** Probabilities for silent failures and false alarms

| Test sample size $N$ | probability of silent failure (false negative) | probability of false alarm (false positive) |
|---|---|---|
| 1000 | $9.5\ 10^{-3}$ | $2.3\ 10^{-2}$ |
| 2000 | $4.6\ 10^{-3}$ | $2.3\ 10^{-3}$ |
| 3000 | $2.6\ 10^{-5}$ | $2.6\ 10^{-3}$ |
| 4000 | $1.4\ 10^{-6}$ | $3.1\ 10^{-5}$ |
| 5000 | $9.5\ 10^{-8}$ | $4.2\ 10^{-6}$ |
| 6000 | $< 10^{-8}$ | $4.3\ 10^{-7}$ |
| 7000 | $< 10^{-8}$ | $5.6\ 10^{-8}$ |

## 8 Conclusion

The main result of this work was to show that the presence of a large component of non-white noise, such as $1/f$ noise with long-ranged autocorrelations, does not preclude building a highly entropic source for TRNGs. While such noise leads to some inconveniences, it can be analysed by determining how the min-entropy estimate varies with the block size and extrapolating the result to infinite length. Based on such results it is possible to devise appropriate online tests that guarantee a suitable amount of randomness. We speculate that many of the existing oscillator-based TRNGs that were designed on the assumption of pure white noise actually have some component of pink noise. Our work provides a method to appropriately calibrate the online health test to avoid any potential security weakness.

**Data availability statement.** Datasets that have been used in the preparation of this manuscript have been posted on a public repository [26]. The software required to reproduce these datasets is available from GitHub [23].

**Acknowledgments.** This work was supported by the Government Office for the Protection of Classified Information (Research Project No. V1-2119), the Slovenian Research and Innovation Agency (Research Project No. V1-2119 and Research Core Funding No. P1-0416).

## References

[1] Johnston, D.: Random Number Generators - Principles and Practices. DeG PRESS, Berlin/Boston (2018)

[2] Stipčević, M., Koç, Ç.K.: True random number generators. In: Open Problems in Mathematics and Computational Science, pp. 275–315. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10683-0_12 . https://doi.org/10.1007/978-3-319-10683-0_12

[3] Herrero-Collantes, M., Garcia-Escartin, J.C.: Quantum random number generators. Reviews of Modern Physics **89**(1), 015004 (2017) https://doi.org/10.1103/revmodphys.89.015004

[4] Kollmitzer, C., Petscharnig, S., Suda, M., Mehic, M.: Quantum random number generation. In: Quantum Science and Technology, pp. 11–34. Springer, Cham (2020). https://doi.org/10.1007/978-3-319-72596-3_2 . https://doi.org/10.1007/978-3-319-72596-3_2

[5] Killmann, W., Schindler, W.: A proposal for: Functionality classes for random number generators. AIS 20 / AIS 31, Version 2.0, English Translation, BSI (2011). https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_Functionality_classes_for_random_number_generators_e.html

[6] Peter, M., Schindler, W.: A proposal for functionality classes for random number generators, Version 2.35 - DRAFT, September 2, 2022 (2022)

[7] BSI: Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren. Version 3.0 (15.05.2013) (2013). https://www.bsi.bund.de/dok/6618284

[8] BSI: Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren. Version 3 (15.05.2020) (2020). https://www.bsi.bund.de/dok/6618252

[9] Turan, M.S., Barker, E., Kelsey, J., McKay, K.A., Baish, M.L., Boyle, M.: Recommendation for the entropy sources used for random bit generation. NIST Special Publication SP 800-90B (2018). https://doi.org/10.6028/NIST.SP.800-90B

[10] Valtchanov, B., Aubert, A., Bernard, F., Fischer, V.: Modeling and observing the jitter in ring oscillators implemented in FPGAs. In: 2008 11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems. IEEE, Bratislava, Slovakia (2008). https://doi.org/10.1109/ddecs.2008.4538777 . https://doi.org/10.1109/ddecs.2008.4538777

[11] Lubicz, D., Bochard, N.: Towards an oscillator based TRNG with a certified entropy rate. IEEE Transactions on Computers **64**(4), 1191–1200 (2015) https://doi.org/10.1109/tc.2014.2308423

[12] Allini, E.N., Skórski, M., Petura, O., Bernard, F., Laban, M., Fischer, V.: Evaluation and monitoring of free running oscillators serving as source of randomness. IACR Transactions on Cryptographic Hardware and Embedded Systems (2018)

[13] Sunar, B., Martin, W., Stinson, D.: A provably secure true random number generator with built-in tolerance to active attacks. IEEE Transactions on Computers **56**(1), 109–119 (2007) https://doi.org/10.1109/tc.2007.250627

[14] Baudet, M., Lubicz, D., Micolod, J., Tassiaux, A.: On the security of oscillator-based random number generators. Journal of Cryptology **24**(2), 398–425 (2010) https://doi.org/10.1007/s00145-010-9089-3

[15] Sullivan, D.B., Allan, D.W., Howe, D.A., Walls, F.L.: Characterization of clocks and oscillators. NIST technical note 1337 (1990)

[16] Hajimiri, A., Limotyrakis, S., Lee, T.H.: Jitter and phase noise in ring oscillators. IEEE JOURNAL OF SOLID-STATE CIRCUITS **34**, 790 (1999)

[17] Haddad, P., Berdnard, F., Fischer, V., Teglia, Y.: On the Assumption of Mutual Independence of Jitter Realizations in P-TRNG Stochastic Models. Design, Automation and Test in Europe (DATE 2014), Dresden, Germany. IEEE (2014)

[18] Paladino, E., Galperin, Y.M., Falci, G., Altshuler, B.L.: 1/f noise: Implications for solid-state quantum information. Reviews of Modern Physics **86**(2), 361–418 (2014) https://doi.org/10.1103/revmodphys.86.361

[19] Kogan, S.: Electronic Noise and Fluctuations in Solids. Cambridge University Press, Cambridge (1996)

[20] Keshner, M.S.: 1/f noise. Proceedings of the IEEE **70**, 212 (1982)

[21] Fischer, V., Lubicz, D.: Embedded Evaluation of Randomness in Oscillator Based Elementary TRNG. IACR 2013 (2014)

[22] Timmer, J., Koenig, M.: On generating power law noise. Astronomy and Astrophysics **300**, 707 (1995)

[23] Žitko, R.: sumi - Gaussian noise stochastic simulation tool. GitHub repository (2023). https://github.com/rokzitko/sumi

[24] NIST: EntropyAssessment. GitHub repository (2023). https://github.com/usnistgov/SP800-90B_EntropyAssessment

[25] Petura, O., Laban, M., Allini, E.N., Fischer, V.: Two Methods of the Clock Jitter Measurement Aimed at Embedded TRNG Testing. Trustworthy Manufacturing and Utilization of Secure Devices (TRUDEVICE) (2018)

[26] Zitko, R.: Datasets for reproducing the results in "True random number generators with flicker noise: stochastic model, min-entropy calculation and online test". Zenodo (2024). https://doi.org/10.5281/zenodo.10680355 . https://doi.org/10.5281/zenodo.10680355