CHAOS
Theory and Applications
in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.4 / No.2 / 2022 / pp.64-70
https://doi.org/10.51537/chaos.1112243

Experimental Validation of a Chaotic Jerk Circuit Based True Random Number Generator

R. Chase Harrison *,[1], **Benjamin K. Rhea** *,[2], **Ariel R. Oldag** *,[3], **Robert N. Dean** *,[4] and **Edmon Perkins** †,[5]

*Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849, United States, †LAB2701, Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, NC 27695, United States.

ABSTRACT A method for true random number generation by directly sampling a high frequency chaotic jerk circuit is explored. A method for determination of the maximum Lyapunov exponent, and thus the maximum bit rate for true random number generation, of the jerk system of interest is shown. The system is tested over a wide range of sampling parameters in order to simulate possible hardware configurations. The system is then implemented in high speed electronics on a small printed circuit board to verify its performance over the chosen parameters. The resulting circuit is well suited for random number generation due to its high dynamic complexity, long term aperiodicity, and extreme sensitivity to initial conditions. This system passes the Dieharder RNG test suite at 3.125 Mbps.

KEYWORDS
Random number generation
Dieharder
Chaotic circuits
Chaos theory
Nonlinear dynamics
Jerk oscillator

INTRODUCTION

Chaos and randomness have gone hand-in-hand since the inception of the idea that both natural and man-made systems could produce wildly different behaviors given seemingly identical conditions. There has always been a struggle to determine the outcome of future processes given only present information, but the feasibility of these endeavors has only recently been quantified in terms of entropy and randomness. Efforts to achieve this understanding have shed light on the inherent nonlinear dynamics. As such, these properties can be exploited to achieve a randomness that can be understood and measured, yet retain the useful unpredictable nature of these dynamics.

Security and communications systems today, including financial security, RFID, and cryptography, rely on this idea of randomness Sundaresan *et al.* (2015); Volos (2013). Specifically, these technologies depend heavily on random bits being readily available to process into various encryption schemes. It is important that the random numbers in these systems exhibit various statistical properties that are indicative of a theoretically perfect random sequence. These are qualitatively grouped into terms such as "strong" or "weak" random numbers. If the random numbers are less than ideal or lack statistical randomness, the biases and dependencies in the bit stream can potentially compromise encrypted systems. Thus, it is imperative that the random numbers can be trusted to be theoretically random. In order to evaluate the statistical properties of random numbers, the bit sequences are submitted to various RNG tests, many of which are bundled into test suites such as NIST's Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications and Duke University's Dieharder Random Number Test Suite Bassham III *et al.* (2010); Brown *et al.* (2013). Most systems today use a pseudo-random number generator (PRNG), since they are easily integrated into electronic systems; however, true random number generators (TRNGs) provide fundamental advantages since their numbers are truly "random" in addition to statistical randomness.

PRNGs are exceptional choices for producing statistically random numbers quickly, even though they lack true randomness. These pseudorandom bit sequences are produced using various algorithms, which range in both computational requirements and complexity Akhshani *et al.* (2014); Han and Kim (2017); Li *et al.* (2010). Because there are no physical processes limiting these algorithms, the implementation of the algorithm can be made as fast as possible, and pseudorandom numbers can be made when needed without regard to lack of supply. However, the fact that these RNG schemes are entirely software based presents inherent weaknesses to the strength of these schemes. First, an exact replica of the scheme can be copied across many systems. For this reason, if a portion of the sequence is known, the rest of the sequence

Manuscript received: 11 May 2022,
Revised: 4 June 2022,
Accepted: 5 June 2022.

[1] rch0012@auburn.edu
[2] bkr0001@auburn.edu
[3] anr0025@auburn.edu
[4] deanron@auburn.edu
[5] edmonperkins@ncsu.edu (**Corresponding Author**)

can be extrapolated. Second, the initial state of the system (e.g., the "seed") exactly determines all future outputs of the system. Hence, the true randomness of PRNGs is extremely low despite their adoption throughout electronic, security, and communication systems.

Rather than producing statistically random numbers that can be replicated at will, true random number generators instead are based on a physical process (here, a chaotic circuit) that generates entropy. After the chaotic circuit generates entropy, a method is used to extract a bit of information from the dynamic response. Together, the chaotic system and the method of bit extraction may be considered a TRNG. One method for quantifying the entropy in a system of interest is in terms of a maximum Lyapunov exponent (MLE). This is the rate of divergence of two trajectories with almost identical initial conditions that are allowed to propagate in time. The maximum Lyapunov exponent also determines the maximum rate at which truly random bits can be extracted from the system Wolf *et al.* (1985). In essence, new information is available from the system at a specified rate, and once that information is extracted, there is a period of time before new information is available to be extracted again. Thus, a system with a large MLE gets this new information, and thus random bits, faster than a system with a small MLE.

Characteristics such as sensitivity to initial conditions, aperiodicity, and spread spectrum power density make chaotic systems ideal candidates for true random number generation Guinee and Blaszczyk (2009); Ergun and Ozoguz (2007); Pareschi *et al.* (2009); Blaszczyk and Guinee (2008). Chaotic systems that can be represented by sets of differential equations have the ability to be quantified as potential RNGs both from the ideal equations and from the implementation in hardware Valtierra *et al.* (2017); Sprott (2000); Tavas *et al.* (2010); Saito and Fujita (1981). Unfortunately, most chaotic systems that can be easily described with differential equations often do not lend themselves to simple electronic circuit implementation, while processes that are seemingly chaotic can be difficult to quantify accurately without prior knowledge of the underlying mechanics of the system.

Since the final goal of a TRNG is to get bits from a physical hardware process into a digital system, a scheme for forming these random bits from a process must be chosen. Many methods to achieve this are available, including sampling the process with an analog-to-digital converter, observing resulting clock jitter, and multiple oscillator sampling Cicek *et al.* (2014). Although the calculation of a maximum Lyapunov exponent in a system sets the maximum rate at which bits can be extracted from the system, there is no information determined about which bit sampling method to use. Often, the bit sampling method will necessitate using various post processing techniques to correct for the biases inherent in most sampling techniques of these physical systems. This is done to ensure the statistical randomness needed in order to pass the stringent testing that is required for random number generators Pareschi *et al.* (2010).

THE IDEAL JERK CHAOTIC SYSTEM

Many third order differential equations that exhibit chaotic behavior have previously been explored by Sprott Sprott (2010). These systems are known as "jerk" systems, because of their dependence on the third derivative with respect to time. Jerk systems have been implemented as a Josephson junction circuit Yalçin (2007), a diode-based circuit design Njitacke *et al.* (2017), and a smoothly-adjustable nonlinearity circuit Kengne *et al.* (2019).

This work focuses on a jerk oscillator that has a nonlinear term that is easily implemented in electronics: a signum function. Specifically, each of the integration stages can be implemented with operational amplifiers, and the signum function can be implemented as a high speed comparator Harrison *et al.* (2016). The jerk system of interest is given below as a third order differential equation in (1) and (2).

$$\dddot{x} = -0.5\ddot{x} - \dot{x} - x + sgn(x) \tag{1}$$

where

$$sgn(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases} \tag{2}$$

The nonlinearity in this system is caused by the signum function. The nonlinear dynamics for this type of system was well-defined by Sprott Sprott (2010). A modern implementation of this circuit using a comparator instead of a saturated operational amplifier is used in the current paper. A phase space plot of this system is shown in Fig. 1.

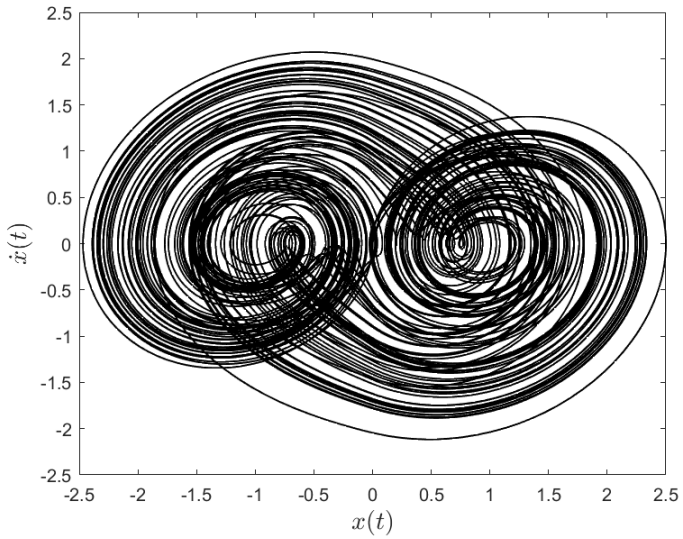

Figure 1 The ideal jerk equation's simulated phase space.

Next, the maximum Lyapunov exponent of the system needs to be estimated so that the maximum bit rate for random number generation can be found. The method chosen to accomplish this is a direct measurement of the divergence rate for many pairs of simulated trajectories that have almost identical initial conditions. The sensitivity of these chaotic systems to initial conditions causes trajectories that are different only by an amount well below measurement thresholds of real systems to quickly diverge. The MLE is then calculated by comparing the inital offset between the two systems with the time it takes for the difference in trajectories to reach a chosen threshold. This calculation is given with the following equation:

$$MLE = \frac{ln(\frac{threshold}{offset})}{time} \tag{3}$$

where *threshold* is the chosen divergence limit, *offset* is the initial difference in states of the two trajectories, and *time* is the final time taken to reach the threshold. Then, this translates into a theoretical maximum bit rate as follows:

$$bitrate = \frac{MLE}{ln\,2} \approx 1.443 * MLE \tag{4}$$

The bit rate given in eq. (4) has units of s^{-1}.

For the jerk system, 1000 time domain simulations are performed in order to determine the MLE by using a MATLAB script to implement the differential equations with a fixed time step. Initial conditions for both systems (the x, $\dot{x}$, and $\ddot{x}$ states) are randomized such that they were between -1 and $+1$, so that the trajectories remain in the chaotic region of the attractor rather than becoming globally unstable. Then, a very small offset (between 10^{-12} and 10^{-8}) is applied to the second system's x state. The threshold limit is chosen to be between 10^{-4} and 10^{-1}. The systems are then simulated forward in time until they reach the specified threshold. After this point, the systems diverge quickly.

The resulting MLE from many simulations with various initial conditions, thresholds, and offsets are between 0.152 and 0.153 for the jerk system. The MLE calculated in Sprott (2000) for the same system when using another method with some approximations is similar to this value. This then gives a bit rate of 0.218 and 0.221 bits per second. Examination of the system reveals that the system has a natural "pseudo frequency" of oscillation (when considering the time to complete one orbit around half of the attractor) of approximately 0.2 Hz. Thus, the bit rate of the system itself is approximately 1 bit per cycle. By framing the bit rate this way, the system can be scaled to any frequency, and the bit rate will remain constant with respect to the system. Specifically, when the system is implemented at high frequency in an electronic circuit, the circuit will be able to generate random bits at this higher natural frequency, as long as the system is represented accurately.

RANDOM NUMBER GENERATION

In order to obtain random bits from the system, the system is sampled at a fixed rate using an analog-to-digital converter (ADC). There are a number of different parameters for an ADC that can be chosen in regards to sampling, including voltage range, sampling frequency, and bits of resolution. In order for the jerk system to be a true random number generator, the sampling parameters of the ADC, which samples the system to produce bits that are statistically random, must be determined. These parameters are then replicated in hardware, and the physical electronic circuit is sampled in order to get truly random bits.

Simulation of an ADC sampling the system is achieved in MATLAB by implementing the jerk system equation. The 32-bit floating point value for x (since the x variable is sampled in hardware) is then converted to an n-bit sample value based on the chosen resolution of n bits and the ADC's maximum voltage. Successive samples are taken at approximately the natural pseudo frequency of the jerk system (i.e., at 0.2 Hz). For each sample, every bit of the sample except the lowest bit is discarded, and the lowest remaining bits are concatenated to form 8-bit random bytes.

The Dieharder test suite is used to evaluate the bit sequences generated from this simulation. There are 114 tests of randomness in this suite, but some tests are simply variations of other tests. However, in evaluating the sequences for randomness, the 114 tests are viewed as independent. Each test returns a P-value between 0 and 1, which is interpreted as follows: a P-value that is between 0.005 and 0.995 is considered to have passed that test, and a P-value of exactly 0 or 1 is considered to have failed. P-values that are under 0.005 and above 0.995 are "weak" and can be further resolved to either pass or fail through more testing. Due to the amount of data that Dieharder requires, some sequences that are actually random will produce weak P-values in approximately 1% of tests. An example output of Dieharder is shown in Fig. 2.

```
#=============================================================================
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown
#=============================================================================
   rng_name    |           filename             |rands/second|
 file_input_raw|           hard_3p125MHz_B16.bin|  1.92e+07  |
#=============================================================================
        test_name   |ntup| tsamples |psamples|  p-value |Assessment
#=============================================================================
   diehard_birthdays|   0|       100|     100|0.97980759|  PASSED
      diehard_operm5|   0|   1000000|     100|0.65329807|  PASSED
  diehard_rank_32x32|   0|     40000|     100|0.84539595|  PASSED
    diehard_rank_6x8|   0|    100000|     100|0.16086332|  PASSED
   diehard_bitstream|   0|   2097152|     100|0.52860159|  PASSED
        diehard_opso|   0|   2097152|     100|0.53328270|  PASSED
```

Figure 2 Partial output of Dieharder testing. More tests and results are given than are shown here.

The P-values returned by the Dieharder tests are such that if the tested bit sequence is statistically random, the P-values are uniformly distributed from 0 to 1. This allows for both individual test results and the results from the Dieharder suite as a whole to be analyzed quickly. This can be visually represented by plotting the test results versus a uniform distribution, in order to see the agreement. An example of this process is shown in Fig. 3 for simulation and hardware results. The P-values are sorted in ascending order (the type of test for each P-value is not taken into account) when plotted as a cumulative frequency against a straight line (the uniform distribution). Although this is only simulated data, it provides a baseline from which to build a hardware system that closely matches the parameters from the simulation equation and the analog-to-digital converter. From this testing, it is discovered that bit sequences gathered from ADC resolutions below 10-bits do not pass the Dieharder suite, indicating that these sequences are not statistically random. Above the 10-bit resolution mark, the P-values from the generated bit sequences are close to the desired uniform distributions.

These simulation test results are performed by sampling at 1 bit per cycle, and almost ideal results are obtained at the 12-bit resolution level. When the sampling frequency is increased to 2.5 bits and 5 bits per cycle at 12-bits of resolution, the bit sequences still pass most or all of the Dieharder test suite. These plots indicate that statistically random bits can be obtained from a system that is sampled faster than the maximum Lyapunov exponent dictates for true random number generation. Thus, it is imperative that the implementation of the random number generation take into account the theoretical limitation for true randomness. Beyond this limit, the system as a whole cannot be truly random, even though statistical randomness might be achieved at a higher bit rate.

HARDWARE CIRCUIT DESIGN AND TESTING

The circuit presented in this paper, which is shown in Fig. 4, includes additional components to make testing easier. Preliminary simulations of this circuit were presented in Harrison *et al.* (2019), and a preliminary design of this circuit was described in Harrison *et al.* (2016) and Harrison *et al.* (2017). Specifically, the current circuit has pin headers used to power the board that are exchanged with a micro USB connector to enable power to come directly from a number of readily available power sources, including a host computer with an open USB port or a mobile battery bank for testing inside of an enclosed space. A PMOS transistor is placed after this connector in order to apply power to the rest of the circuit in a more controlled fashion (via unshorting a jumper on the board

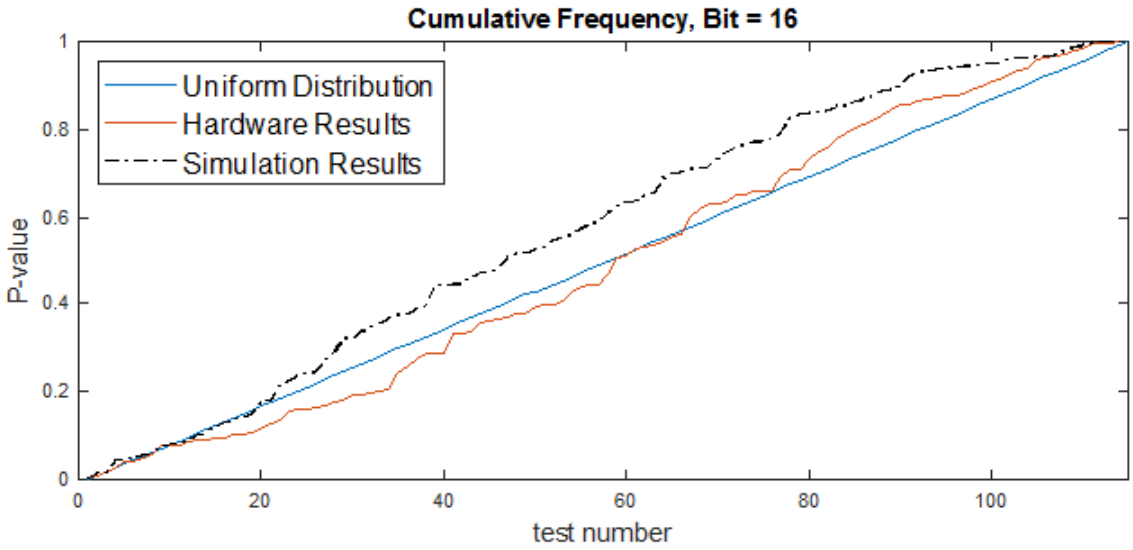

Figure 3 Cumulative frequency plot of Dieharder test results from simulating the 16^{th} bit of an ADC in software, and hardware implementation of ADC sampling of the jerk circuit at 3.125 MSPS.

from ground) than just plugging in the connector. Also, in testing previous versions of the board, it was discovered that the circuit could enter a periodic railing state upon being powered, but it could then be forced into the normal chaotic state by temporarily shorting the x node of the op amp integrators to ground.

A small momentary switch is added to this node to facilitate this correction. Finally, the signum output is taken from the $-Q$ pin on the comparator so as to not interfere with the Q signal in the feedback path. The Q pin is the signum output of the comparator, which is directly connected to the rest of the feedback summing circuitry. In earlier versions of the board, probing this pin directly caused some undesired changes to the chaotic attractor, since the oscilloscope loaded the pin. Essentially, if probed, the signum output would no longer be accurate. To remedy this, the chip has a $-Q$ output that can be probed instead, which can be inverted if needed. By probing the $-Q$ pin, this leaves the Q output undisturbed. As another solution, the Q output could instead have been buffered with another chip (e.g., an operational amplifier), but the current board was designed with a minimal part count in mind. The pseudo-fundamental frequency of the board was maintained at 4 MHz. A picture of the front and back of the updated board is shown in Fig. 5.

It should be noted that the 15kΩ resistors are a deviation from the ideal operational amplifier integration circuit, and, in fact, these convert the ideal integrators into active amplifier circuits with a low pass filter. Since the gain of these integrators was high, the operational amplifier integrators could easily saturate on startup, which would prevent oscillations from occurring. The 15kΩ resistors prevent saturation of the feedback capacitors. The 15kΩ value for these resistors was found from trial and error: a resistance that is too high causes the capacitor to saturate, while a resistance that is too low causes the oscillations to be considerably damped (e.g., the chaotic signal would stay on each side of the attractor for too long before switching, which reduces the Lyapunov exponent of the implemented circuit). This value was chosen for good oscillation characteristics compared with the ideal circuit.

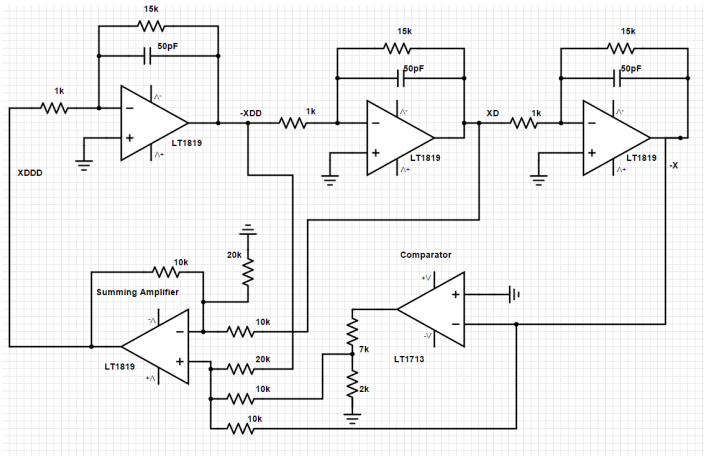

Figure 4 A schematic of the electronic implementation of the jerk oscillator.

For testing of this circuit as a true random number generator, ADC sampling is achieved using a Handyscope HS6 USB Oscilloscope from TiePie Engineering connected to a host computer running the provided MultiChannel software. The circuit is powered from a USB port on the same computer. The HS6 allows for up to 16 bit streaming ADC sampling, but at that resolution the sampling speed is limited to 3.125 MS/s, slightly less than the desired 4 MS/s to achieve a 1 bit per cycle random output. The full scale voltage of the ADC is chosen to be ±2 V since the circuit has peaks of approximately ±1 V when AC coupled to the oscilloscope and powered using the single 5V supply from the USB port. This results in a loss of approximately 1 bit of resolution when compared to a full scale voltage that is smaller, but the next lowest supported by the MultiChannel software is 800 mV. A screenshot of the circuit being sampled in this software is shown in Fig. 6.

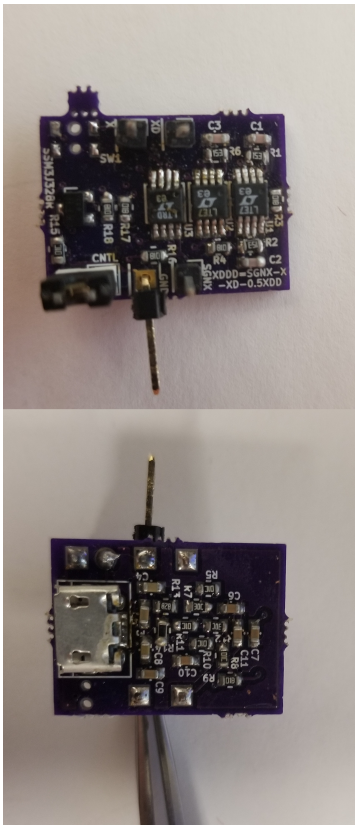

Figure 5 The front and back of the populated circuit board that implements the jerk equation at 4 MHz.

Figure 6 Data readout from the Handyscope HS6 from the hardware circuit.

32 GB of data is collected from the x node of the circuit at 3.125 MS/s and then split using a MATLAB script into sixteen 2 GB files, one with each separate bit of the 16-bit samples concatenated together. These files are then subjected to Dieharder testing in the same manner as the jerk equation simulation data. These results are shown in Fig. 3.

For the hardware circuit, the bit sequences pass most of the Dieharder tests starting at the 13^{th} bit and do not fail any tests at the 16^{th} bit of the ADC sample. At the 12^{th} bit and above, the sequences systematically fail certain sets of tests, most notably the sts_serial and rgb_lagged_sum series of tests. These tests involve skipping many bits in a row and thus the input file is rewound multiple times for each of these tests. This can potentially make the bit sequence seem like it is repeating itself, but this is unavoidable with this setup of Dieharder without a much larger input file size.

A concern for this method of sampling (i.e., taking one bit per sample of a high resolution sample) is that the lowest bits are masked by noise in the system, and thus the randomness ultimately achieved may be due to noise and not to the chaotic dynamics in the system. Although the noise floor is a useful metric for linear systems, noise and nonlinearity can produce unintuitive dynamics. For instance, noise can cause a system undergoing chaos to become regular Lepik and Hein (2005), but it can also drive a system undergoing regular motion to become chaotic Perkins and Balachandran (2012). Further, noise can cause stochastic resonance Perkins and Balachandran (2015), modify the hysteresis curve of nonlinear oscillators Perkins (2017); Perkins and Fitzgerald (2018), affect the dynamics of intrinsic localized modes in coupled oscillator arrays Perkins *et al.* (2016); Balachandran *et al.* (2015), and cause learning to be degraded in an adaptive oscillator circuit Li *et al.* (2021).

In order to investigate whether the noise had an effect on the randomness of the jerk oscillator, two additional tests are performed using Dieharder. In the first test, the +5V power rail on the board is used to generate random bits using the same sampling parameters at the 16^{th} bit of resolution. In the second test, a separate chaotic circuit with higher fundamental frequency is likewise sampled. Both of these bit sequences fail the majority of the Dieharder suite with 2 GB input files. If the sampling or thermal noise in the jerk chaos board was providing the randomness in order to pass Dieharder, then the sequences generated from the noise should pass the test suite. These results show that the statistical randomness achieved using this particular nonlinear circuit and bit extraction technique is likely coming from the nonlinear dynamics, instead of from electronic noise.

Although there is no real way to prove randomness, the results from the Dieharder RNG test suite indicate that the jerk circuit is able to provide statistically random bits from a hardware source under various circumstances. This test suite is widely used to stringently test pseudorandom number generators with much larger bit sequences available on demand. Since the chaotic jerk circuit is implemented in a small form factor with commercial off the shelf components, it can easily be integrated into other systems requiring truly random bits at high speeds. The test results from both the simulated and hardware TRNG are in good agreement with each other. Overall, the jerk oscillator circuit is an ideal candidate for random number generation.

CONCLUSION

A scheme for extracting true random numbers directly from a chaotic jerk system is shown. This system is implemented in high speed electronics on a small printed circuit board and sampled in accordance with the necessary parameters found from the simulation results. The bit sequences generated from the physical system pass the Dieharder Random Number Generator test suite, which enables this system to function as a fast random bit generator for many different applications. Overall, this system shows high dynamic complexity in a compact form, which is desirable for a true random number generator.

Acknowledgements

Support for this project under grant number **G00009556/7** from the U.S. Army (research sponsor) and Torch Technologies, Inc. (prime contractor) is greatly appreciated.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Availability of data and material

Not applicable.

LITERATURE CITED

Akhshani, A., A. Akhavan, A. Mobaraki, S.-C. Lim, and Z. Hassan, 2014 Pseudo random number generator based on quantum chaotic map. Communications in Nonlinear Science and Numerical Simulation **19**: 101–111.

Balachandran, B., E. Perkins, and T. Fitzgerald, 2015 Response localization in micro-scale oscillator arrays: influence of cubic coupling nonlinearities. International Journal of Dynamics and Control **3**: 183–188.

Bassham III, L. E., A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, *et al.*, 2010 *Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications*. National Institute of Standards & Technology.

Blaszczyk, M. and R. A. Guinee, 2008 A true random binary sequence generator based on chaotic circuit. In *IET Irish Signals and Systems Conference (ISSC 2008)*, pp. 294–299.

Brown, R. G., D. Eddelbuettel, and D. Bauer, 2013 Dieharder: A random number test suite. Open Source software library, under development, URL http://www. phy. duke. edu/˜ rgb/General/dieharder. php .

Cicek, I., A. E. Pusane, and G. Dundar, 2014 A novel design method for discrete time chaos based true random number generators. INTEGRATION, the VLSI journal **47**: 38–47.

Ergun, S. and S. Ozoguz, 2007 A chaos-modulated dual oscillator-based truly random number generator. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pp. 2482–2485, IEEE.

Guinee, R. A. and M. Blaszczyk, 2009 A novel true random binary sequence generator based on a chaotic double scroll oscillator combination with a pseudo random generator for cryptographic applications. In *Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for*, pp. 1–6, IEEE.

Han, M. and Y. Kim, 2017 Unpredictable 16 bits LFSR-based true random number generator. In *SoC Design Conference (ISOCC), 2017 International*, pp. 284–285, IEEE.

Harrison, R. C., B. K. Rhea, A. N. Ramsey, R. N. Dean, and J. E. Perkins, 2019 A true random number generator based on a chaotic jerk system. In *2019 SoutheastCon*, pp. 1–5, IEEE.

Harrison, R. C., B. K. Rhea, F. T. Werner, and R. Dean, 2017 A compact and low power realization of a high frequency chaotic oscillator. Additional Conferences (Device Packaging, HiTEC, HiTEN, & CICMT) **2017**: 4.

Harrison, R. C., B. K. Rhea, F. T. Werner, and R. N. Dean, 2016 A 4 MHz chaotic oscillator based on a jerk system. In *International Conference on Applications in Nonlinear Dynamics*, pp. 41–51, Springer.

Kengne, J., R. L. T. Mogue, T. F. Fozin, and A. N. K. Telem, 2019 Effects of symmetric and asymmetric nonlinearity on the dynamics of a novel chaotic jerk circuit: Coexisting multiple attractors, period doubling reversals, crisis, and offset boosting. Chaos, Solitons & Fractals **121**: 63–84.

Lepik, Ü. and H. Hein, 2005 On response of nonlinear oscillators with random frequency of excitation. Journal of sound and vibration **288**: 275–292.

Li, C.-Y., T.-Y. Chang, and C.-C. Huang, 2010 A nonlinear PRNG using digitized logistic map with self-reseeding method. In *VLSI Design Automation and Test (VLSI-DAT), 2010 International Symposium on*, pp. 108–111, IEEE.

Li, X., M. R. E. U. Shougat, T. Mollik, A. N. Beal, R. N. Dean, *et al.*, 2021 Stochastic effects on a hopf adaptive frequency oscillator. Journal of Applied Physics **129**: 224901.

Njitacke, Z., L. K. Kengne, *et al.*, 2017 Antimonotonicity, chaos and multiple coexisting attractors in a simple hybrid diode-based jerk circuit. Chaos, Solitons & Fractals **105**: 77–91.

Pareschi, F., G. Scotti, L. Giancane, R. Rovatti, G. Setti, *et al.*, 2009 Power analysis of a chaos-based random number generator for cryptographic security. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pp. 2858–2861, IEEE.

Pareschi, F., G. Setti, and R. Rovatti, 2010 Statistical testing of a chaos based CMOS true-random number generator. Journal of Circuits, Systems, and Computers **19**: 897–910.

Perkins, E., 2017 Effects of noise on the frequency response of the monostable duffing oscillator. Physics Letters A **381**: 1009–1013.

Perkins, E. and B. Balachandran, 2012 Noise-enhanced response of nonlinear oscillators. Procedia Iutam **5**: 59–68.

Perkins, E. and B. Balachandran, 2015 Effects of phase lag on the information rate of a bistable duffing oscillator. Physics Letters A **379**: 308–313.

Perkins, E. and T. Fitzgerald, 2018 Continuation method on cumulant neglect equations. Journal of Computational and Nonlinear Dynamics **13**.

Perkins, E., M. Kimura, T. Hikihara, and B. Balachandran, 2016 Effects of noise on symmetric intrinsic localized modes. Nonlinear Dynamics **85**: 333–341.

Saito, T. and H. Fujita, 1981 Chaos in a manifold piecewise linear system. Electronics and Communications in Japan (Part I: Communications) **64**: 9–17.

Sprott, J. C., 2000 Simple chaotic systems and circuits. American Journal of Physics **68**: 758–763.

Sprott, J. C., 2010 *Elegant chaos: algebraically simple chaotic flows*. World Scientific.

Sundaresan, S., R. Doss, S. Piramuthu, and W. Zhou, 2015 Secure tag search in RFID systems using mobile readers. IEEE Transactions on Dependable and Secure Computing **12**: 230–242.

Tavas, V., A. S. Demirkol, S. Ozoguz, A. Zeki, and A. Toker, 2010 An ADC based random bit generator based on a double scroll chaotic circuit. Journal of Circuits, Systems, and Computers **19**: 1621–1639.

Valtierra, J. L., E. Tlelo-Cuautle, and Á. Rodríguez-Vázquez, 2017 A switched-capacitor skew-tent map implementation for random

number generation. International Journal of Circuit Theory and Applications **45**: 305–315.
Volos, C. K., 2013 Image encryption scheme based on coupled chaotic systems. Journal of Applied Mathematics and Bioinformatics **3**: 123.
Wolf, A., J. B. Swift, H. L. Swinney, and J. A. Vastano, 1985 Determining lyapunov exponents from a time series. Physica D: Nonlinear Phenomena **16**: 285–317.
Yalçin, M. E., 2007 Multi-scroll and hypercube attractors from a general jerk circuit using josephson junctions. Chaos, Solitons & Fractals **34**: 1659–1666.

How to cite this article: Harrison, R. C., Rhea, B. K., Oldag, A. R., Dean, R. N. and Perkins, E. Experimental Validation of a Chaotic Jerk Circuit Based True Random Number Generator. Chaos Theory and Applications, 4(2), 64-70, 2022.