Chaos and Randomness in an Electric Circuit

Juan Marcos Avina
DEPARTMENT OF PHYSICS AND ASTRONOMY
UNIVERSITY OF NORTHERN COLORADO, Greeley, CO, 80639
FACULTY ADVISOR: DR. MATTHEW SEMAK

ABSTRACT

To pursue my interests in electrical engineering and nonlinear dynamics, I began researching chaotic circuits. This research focuses on the behavior of Chua's circuit and its application to the generation of random numbers. Chua's circuit, a simple electronic design, is well-known for its ability to generate a chaotic signal. This circuit consists of one inductor, one resistor, two capacitors, and Chua's diode. Chua's diode is made of two negative impedance converters wired in parallel, and functions as the source of nonlinearity in the circuit. Chaotic behavior can only occur when a nonlinear element is present. The chaotic signal from this circuit is used to cause electronic jitter necessary for producing a true random number generator (RNG). The output of the RNG is converted to a readable file so that a statistical analysis of the randomness can be done. These tests show that the output of the RNG more closely mimics the behavior of a series of unbiased coin flips than does the typical RNG found in most computers. Once the behavior of the RNG is verified to be sufficiently random, I hope to investigate its use in applications such as data encryption.

Background Information

The distinction between random and chaotic is an important one to make. Chaotic events are deterministic and sensitive to initial conditions. Random events have no memory of previous data points. This means that if one is to generate a random number using a RNG, the resulting numbers will have zero correlation with each other.

As for a circuit, there are three criteria for generating a chaos [1]:

1. One or more nonlinear elements
2. One or more locally active resistors
3. Three or more energy storage elements

For an ideal RNG, it is convenient to obtain a bitstring where each bit has the same probability of being a one or a zero (much like flipping a series unbiased coins). This makes testing the quality of the RNG a much more streamlined process [2].

Chua Circuit

Figure 1 shows the schematic for Chua's circuit modeled in LTspice. Chua's diode is the main source of nonlinearity as well as locally negative resistance in the circuit. Figure 2 shows the I-V curve of the diode, demonstrating its nonlinearity and negative resistance.

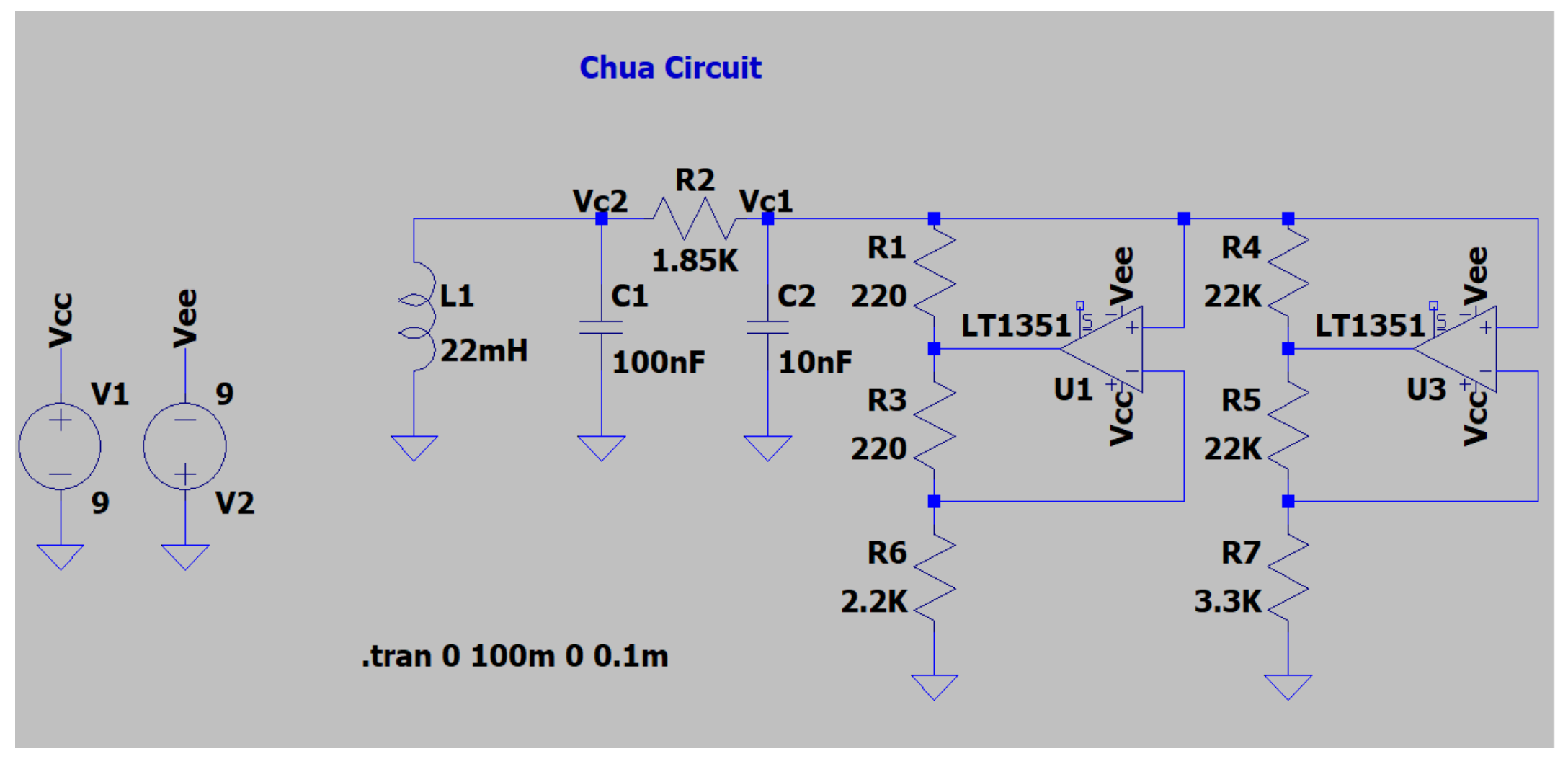

Figure 1.
Schematic of Chua's Circuit made in LTspice. The voltage drop over C1 and C2 (capacitor one and two, respectively) is measured to observe the chaotic output. Alternatively, the current through L1 (the inductor) can also be measured for this same purpose.

A printed circuit board was designed for ease of transportation and construction, with the hope that future students may do research with chaos generators more easily. Figure 3 shows the final result for the printed Chua Circuit

Chua's Circuit was modeled in the LTspice. Figure 4 shows the phase space diagram plotting the voltage drop over the first capacitor versus the voltage drop over the second capacitor.

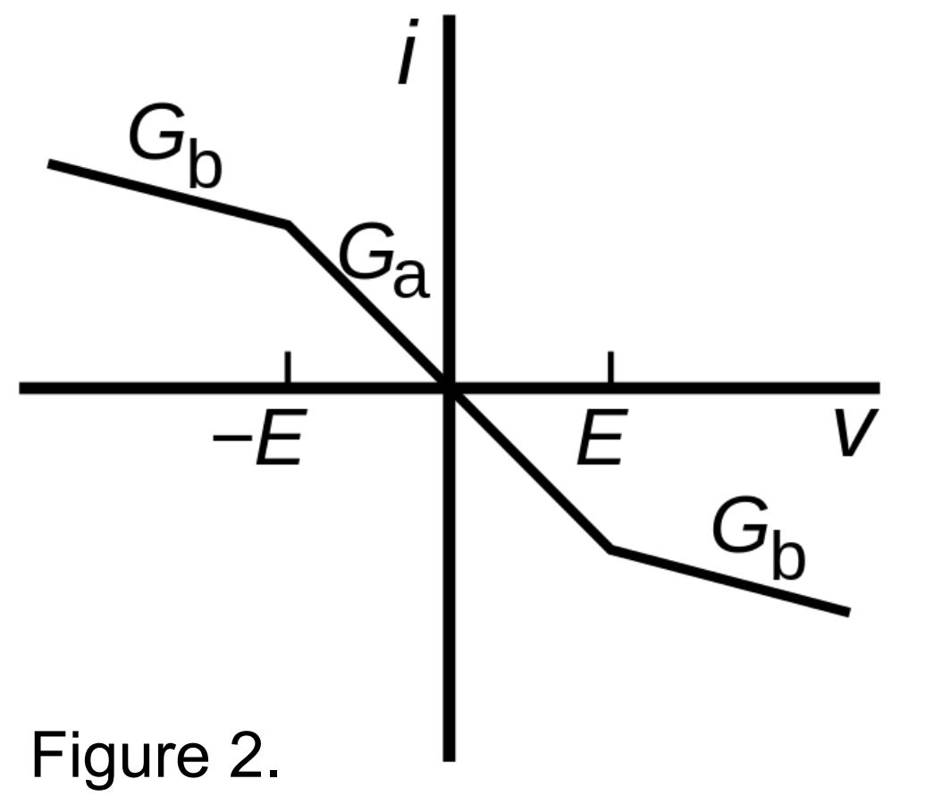

Figure 2.
I-V curve of Chua's diode. The diode here can be seen to have a locally active nonlinear-negative resistance.

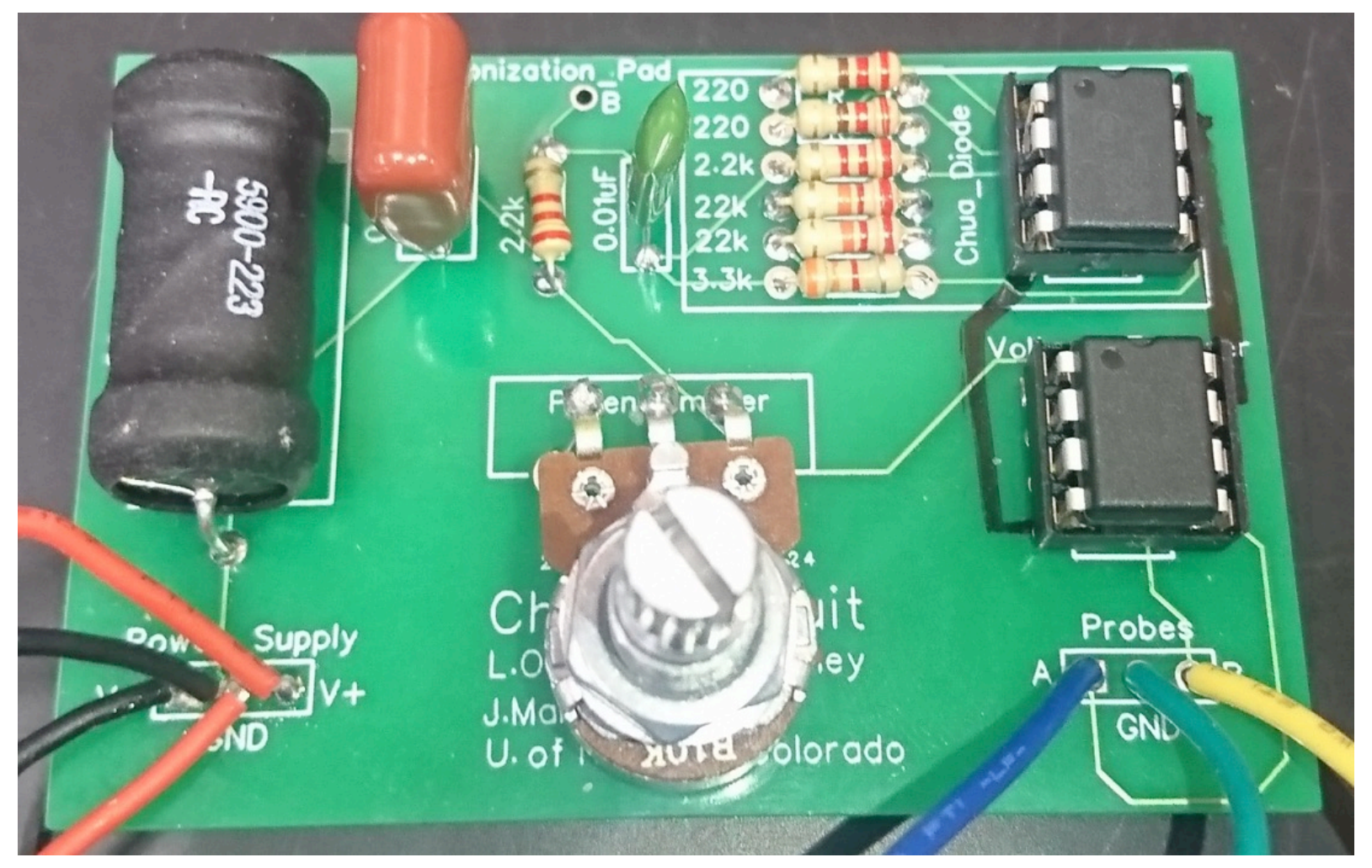

Figure 3.
Fully assembled Chua's Circuit

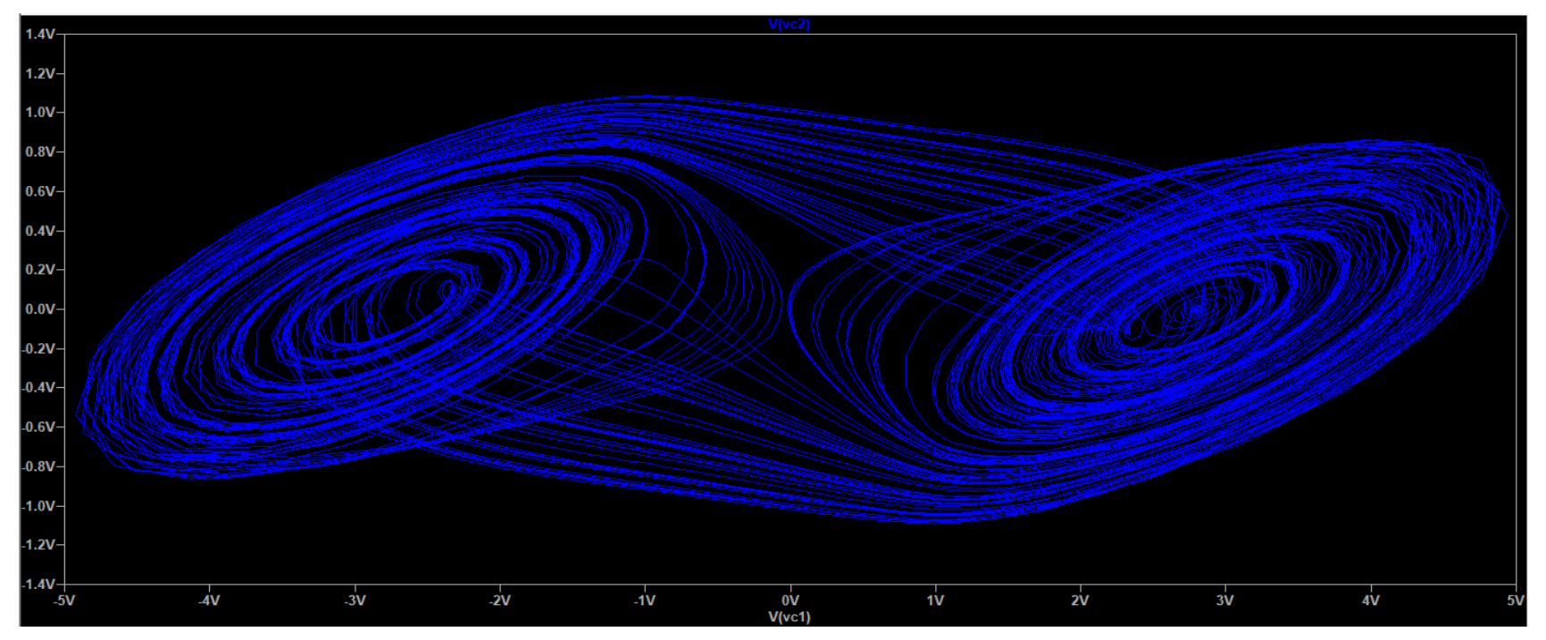

Figure 4.
Phase Space diagram for Chua's Circuit.

Random Number Generator

The RNG was based on a design where a slow jittered oscillator was fed into the clock input of a D-Type flip flop, while a fast oscillator was fed into the data input of the flip flop. This can be referred to as an oscillator sampling RNG [3],[4].

In the RNG, the slow jittered oscillator was created by adding a triangle waveform to the chaotic waveform from Chua's Circuit. Figure 5 shows a diagram of how the RNG was wired up.

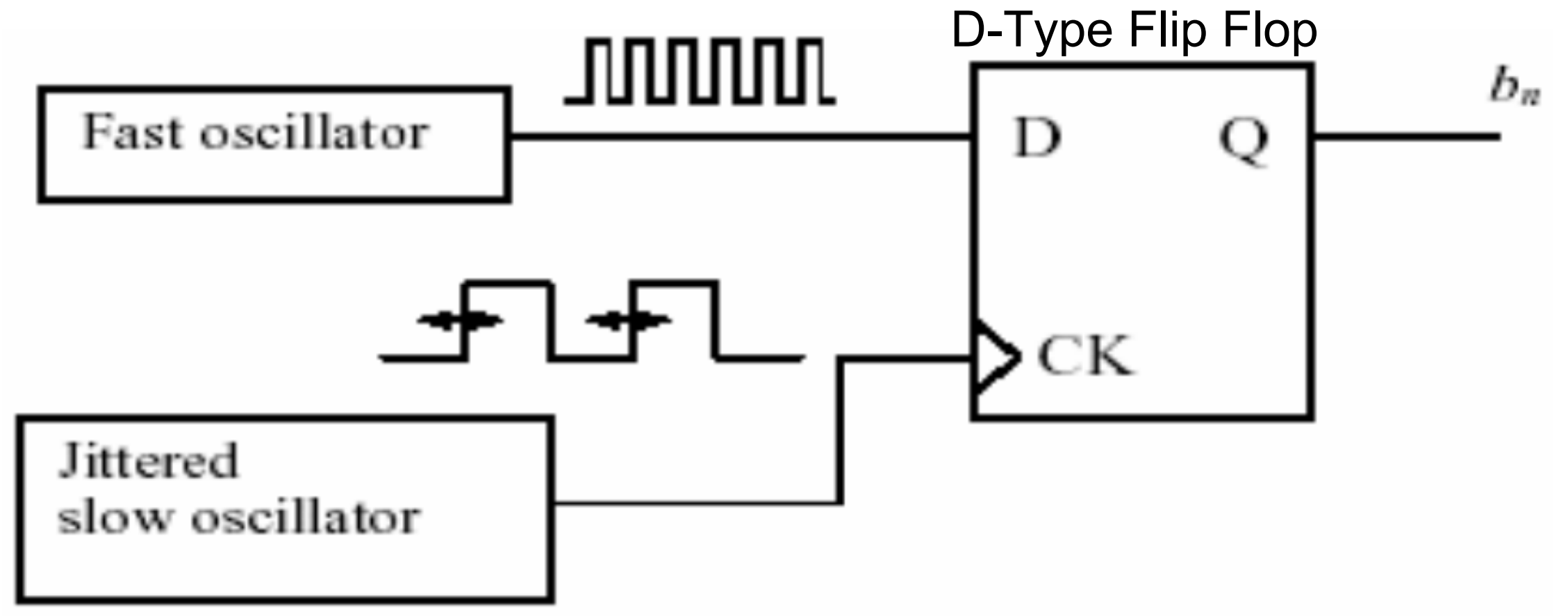

Figure 5.
Diagram of how RNG circuit is constructed. The D-type flip flop will store whatever logic level is applied to its data (D) input during the rising edge of the clock input. The rising edge of the slow jittered oscillator is ambiguous, and produces a random bitstring on the output (Q) of the flip flop.

References

[1] M. P. Kennedy, "Three steps to chaos. I. Evolution," in *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, no. 10, pp. 640-656, Oct. 1993.
[2] SP 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications
[3] C.S. Petrie, J.A. Connelly, "A noise-based IC random number generator for application in cryptography", *IEEE Transaction on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 5, pp. 615-621, May 2000
[4] S. Kilinc, S. Ozoguz, K. Ozdemir, "True Random Number Generation Based on Double-Scroll Chaotic System" *Electroscope: Applied Electronics,* Vol. 2008 no. 3cvcv
[5] "The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness" , Florida State University, 1995.

NIST and DIEHARD Test Suites

NIST test suite: This is a set of tests developed by the National Institute of Standards and Technology for testing for randomness [2].

DIEHARD test suite: A test suite developed by George Marsaglia, is a set of tests for measuring the quality of randomness in a random number generator [5].

These are a set of hypothesis tests where the null hypothesis is that the data being tested display a behavior that is consistent with randomly distributed variables. R Studio software was used to perform these statistical tests on the bitstream data. The DIEHARDER test suite was performed due to its ease of access in R Studio as a quick and easy to use package.

Results

Table 1.
Results of the tests in the Diehard suite. Tests with a P-Value lower than 0.05 suggest we should reject the null hypothesis (that the data is random). The only test with a low enough P-Value to reject the null is the Oqso test. The rest of the tests suggest that the bitstream data is random.

Statistical Test	P-Value	Result
Birthdays	1.99E-01	FAIL TO REJECT
Operm5	5.63E-01	FAIL TO REJECT
Rank 32x32	2.76E-01	FAIL TO REJECT
Rank 6x8	4.23E-01	FAIL TO REJECT
Bitstream	6.78E-01	FAIL TO REJECT
Opso	6.80E-02	FAIL TO REJECT
Oqso	1.39E-87	REJECT
DNA	8.31E-01	FAIL TO REJECT
Count 1s Stream	4.41E-01	FAIL TO REJECT
Count 1s byte	6.16E-01	FAIL TO REJECT
Parking Lot	9.61E-01	FAIL TO REJECT
2dSphere	9.99E-01	FAIL TO REJECT
3dSphere	9.72E-01	FAIL TO REJECT
Squeeze	5.26E-01	FAIL TO REJECT
Sums	3.41E-01	FAIL TO REJECT
Runs	3.42E-01	FAIL TO REJECT
Craps	2.14E-01	FAIL TO REJECT

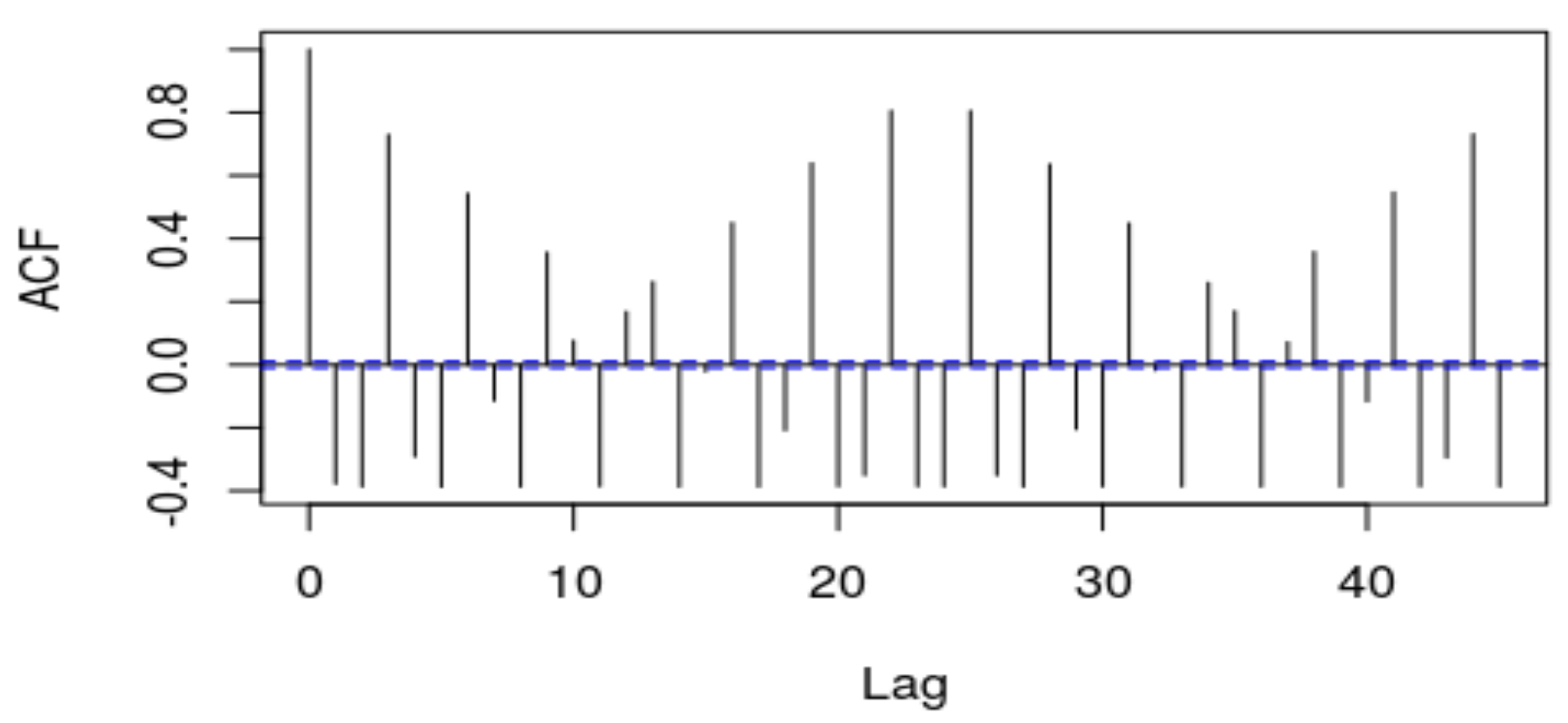

Figure 6.
Autocorrelation function of the output of the RNG.

Table 1 shows that most of the tests in the Diehard suite result in successes. One should keep in mind that a failure to reject the null hypothesis does not mean that the data is, in fact, random, but there is significant evidence to suggest that it is. The p-value from each test gives us a measure of how probable the null hypothesis is. In this case, the null hypothesis is that the data is random. A significant number of the tests ended within positive results.

As seen in Figure 6, the autocorrelation function for the RNG shows that there is a significant correlation amongst data points. In a sample from random numbers, the autocorrelation function should tend towards zero. Our autocorrelation function shows that the data is correlated to itself. This suggests that the data from the RNG is not random.

The discrepancy between the autocorrelation function and the Diehard tests remains to be investigated. The NIST tests are currently being conducted, and the results should be available in the near future.

Acknowledgements

I'd like to thank Dr. Galovich of UNC Physics and Astronomy. I'd also like to acknowledge Chris Koehler and Bernadette Garcia of the Colorado Space Grant Consortium.