Simulating Brownian Motion Based on Unpredictability in the Quantum Realm

PH102

Vraj Patel Kishen N Gowda N V Karthikeya Saumitra Sharma Rushil Shah Nidhin Harilal Vedanta Krishna Bhutani

Objective of the Project

- Generating true random numbers using quantum phenomenon i.e. observing quantity of electrons tunnelling in emitter reverse biased transistor.
- Simulating and studying Brownian Motion using the random numbers generated.

Theory/Principle

True Random Number Generator devices generate random numbers by using physical processes. Deterministic machines like computers are incapable of generating truly random numbers. Our device is based on amplifying random noise. This random noise involves the quantum effect of tunneling. We can safely assume Tunneling to be a random phenomenon. This assumption of tunneling being random is also supported by a paper published in nature (Ref. 1). We utilised the avalanche breakdown phenomenon of a common npn transistor.
According to the datasheet of the transistor (2N2222) used, Avalanche breakdown is observed to be an insulator for a reverse bias voltage of 6V around base-emitter junction.We then use again a common npn junction transistor (this time as amplifier across base-collector region) to amplify the current obtained. After rectification, we require to convert this random current values to give us a perfect digital output of 5V or (about) 0V. For this we use a Schmitt trigger inverter.
Brownian Motion : Brownian motion (or pedesis), also known as Wiener Process is the random motion of particles suspended in a fluid (a liquid or a gas) resulting from their collision with the fast-moving molecules in the fluid. In Brownian motion, the particle moves as though under the influence of random forces of varying direction and magnitude. A standard Wiener process on the interval [0,T] is a random variable W(t) that depends continuously on t in [0,T] and satisfies the following: W(0) = 0; For $0 <= t_1 < t_2 <= T$, W(t) - W(s) ~ $(t2 - t1)^{1/2}$ N(0,1), where N(0,1) is a normal distribution with zero mean and unit variance. For use on a computer, we discretize the Wiener process with a timestep dt as, dW ~ $(dt)^{1/2}$ N(0,1).

Diffusion Coefficient :
The diffusion coefficient is most simply understood as the magnitude of the molar flux through a surface per unit concentration gradient out-of-plane. It is analogous to the property of thermal diffusivity in heat transfer. In an aqueous (water) solution, typical diffusion coefficients are in the range of 10^{-10} to 10^{-9} m^2/s.
The theoretical value of the diffusion coefficient, D, is given by,

$$D = \frac{k_B T}{3\pi\eta T}$$

where T = temperature (Kelvin), k_B = Boltzmann's constant, η = viscosity, and d = particle diameter.

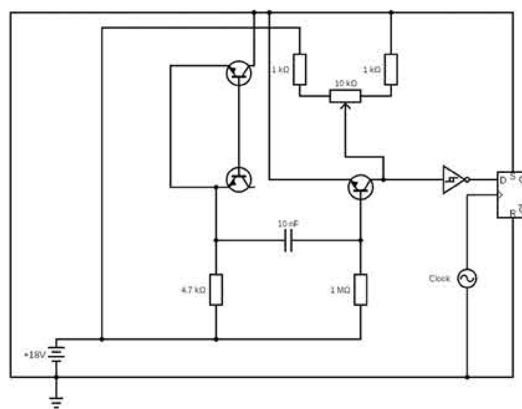
Coefficient of Correlation: The correlation coefficient is a statistical measure that calculates the strength of the relationship between the relative movements of the two variables. The range of values for the correlation coefficient is bounded by 1.0 and -1.0. As Brownian Motion are stochastic processes, the correlation coefficient of any two brownian motion must be close to zero.

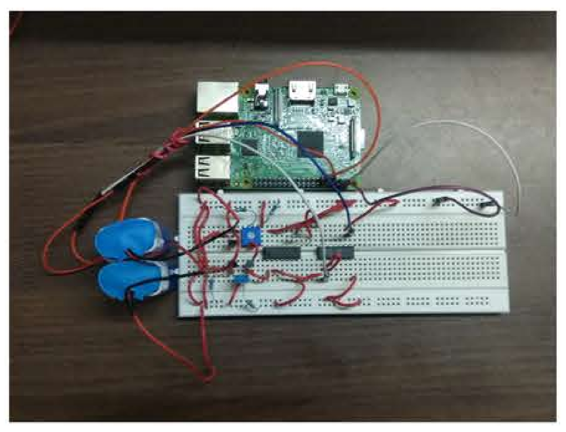
$$r = \frac{Cov(X,Y)}{\sigma_X \sigma_Y (N-1)}$$

Experimental Setup and Procedure

PART 1

1. Make the circuit as given in the above circuit diagram. Since we are making a high-gain amplifier in the circuit, make all the connections as short as possible to avoid picking up stray electromagnetic fields from the surrounding.
2. Try to place all transistors and capacitors as close as possible.
3. Cut the unused collector lead of the noise-generating transistor (A).
4. Give a 90% duty cycle clock of 1 Hz to the flip-flop and the 5V power supply to ICs through Raspberry Pi using the code given.
5. Rotate the potentiometer until you get a good amount of blinking in the LED.
6. Now remove the LED. Now take the input into the Raspberry Pi GPIO.
7. Use the code given below, making appropriate changes based on the pin numbers used.
8. Make appropriate changes for the number of bits required and you get a file made up of the random bits. Repeat the runs 3 times generating 4,00,000 bits on each run and get a set of 3 files.
9. Further analysis is carried out on these files to simulate Brownian Motion.





PART 2

1. Take 8 bits at a time and treat it as a single integer.
2. We get a uniform distribution of numbers from 0 to 255.
3. Convert these numbers to a normal distribution using the Box-Muller method.
4. Now we have a list of random numbers following a normal distribution.
5. Calculate the diffusion coefficient you desire from the simulation using the formula.
6. Now simulate a Wiener process which is a one-dimensional Brownian Motion but scale the displacements based on the diffusion coefficient calculated.
7. Now we take one Wiener process as the X-axis displacement and other as Y-axis displacement.
8. Now, we plot this and get the Brownian Motion Path.
9. Also, we plot the net displacement squared v/s time graph.
10. Also, we plot the net displacement squared v/s time graph.

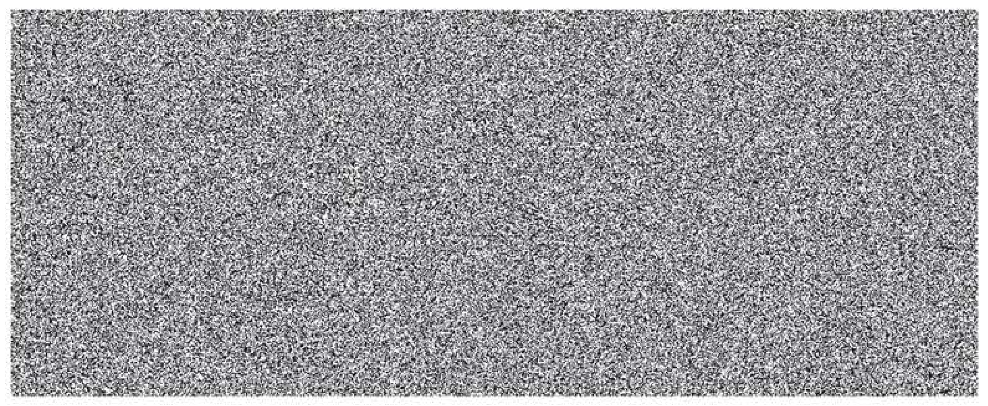
Observations, Discussions and Concluding Remarks

PART 1

Run Number	Arithmetic Mean Bytewise	Monte Carlo Pi Value	% Error in Pi	Chi-Squared Test %	Number of 0s	Number of 1s
1	127.2050	3.163326533	0.69	25.84	199803	200197
2	127.8013	3.147965919	0.20	25.36	199315	200685
3	127.6148	3.155166207	0.43	30.93	199336	200664

- Analysis:
  For an ideal random sequence of large length expectation of the various quantities is as follows:
  Arithmetic Mean Bytewise: 127.5
  Monte Carlo Pi: as close to real Pi as possible
  Chi-Squared Test %: should lie between 10% and 90%

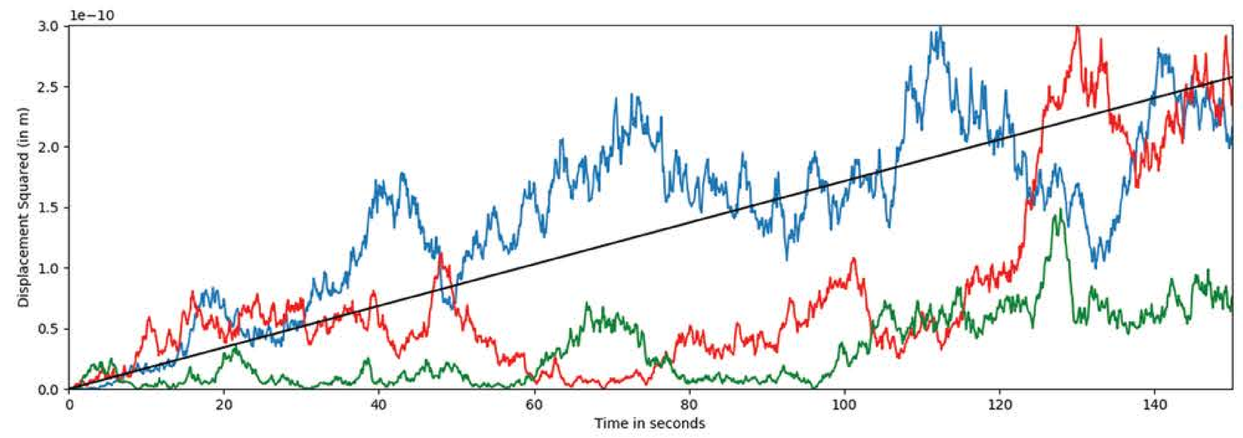
Number of zeroes should be close to the number of ones
We observe that the data obtained after running various tests for randomness says that the sequence that we obtained is indeed random.



Representation of Random Noise Generator (Black pixel = 1 and White pixel = 0)

- Errors
  1. The source is assumed to maintain a uniform probability of giving ones and zeroes over time. Therefore, while de-skewing, we will not get perfect uniform distribution.
  2. Variation in the surroundings can disrupt uniform distribution.
  3. Stray electromagnetic fields might be caught and current values amplified by the amplifier.

PART 2

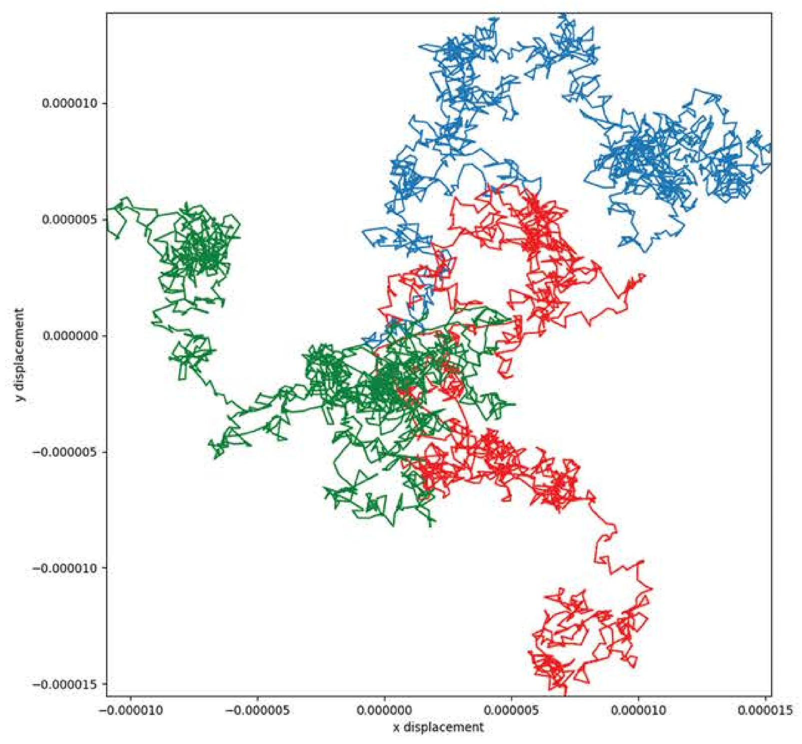


Displacement² vs Time

Estimated Diffusivity from simulated data:
From theory,

$$simulatedD = \frac{mean(dSquaredDisplacement)}{(2 * dimensions * \tau)}$$

From simulated data,
D1 =4.197943082912188 x 10^{-13}
D2 =4.0920671801477695 x 10^{-13}
D3 =4.4991802113610167 x 10^{-13}
simulatedD = (D1+D2+D3)/3 = 4.263063491473658 x 10^{-13}



Brownian Motion

Error Analysis:

$$standardError = \frac{\sigma(dSquaredDisplacement)}{(2 * dimensions * \tau * \sqrt{N})}$$

actualError = D - simulatedD
From Simulation Data:
Err1 = 1.0669802737044507 x 10^{-14}, Err2 = 1.0429709260666482 x 10^{-14}, Err3 = 1.1470266441194321 x 10^{-14}
standardError = (err1 + err2 + err3)/3 = 1.0856592812968437 x 10^{-14}
actualError = 2.71171545 x 10^{-15}

Coefficient of Correlation:

From simulation data,
$r_1 = 4.699927344005296 X 10^{-12}$ (Between W_1 and W_2)
$r_2 = 5.759526357948099 X 10^{-12}$ (Between W_1 and W_3)
$r_3 = 5.728296852854371 X 10^{-12}$ (Between W_2 and W_3)

As we can notice, the correlation factor is nearly zero in all the three cases, which is expected for any two Weiner processes

References

- Ramón Bernardo-Gavito, Nature, Dec. 2017, Extracting random numbers from quantum tunnelling through a single diode.
- https://lben.epfl.ch/files/content/sites/lben/files/users/179705/Simulating%20Brownian%20Motion.pdf
- https://www.comsol.com/multiphysics/diffusion-coefficient
- https://www.investopedia.com/terms/c/correlationcoefficient.asp