Analyzing Chaos and Symmetry in a random Linear Josephson Junction Array

Cliff Sun

April 7, 2026

Introduction

Josephson Junctions are a powerful building block for constructing Superconducting Quantum Architecture. Comprised of a superconducting sandwich (Often referred to as the S-I-S or Superconductor-Insulator-Superconductor System [Fig 1]), Josephson Junctions operate off the Josephson Effect. That is, Cooper Pairs have the ability to Quantum Tunnel across the Insulator, thus producing a supercurrent across the junction without an applied Voltage.

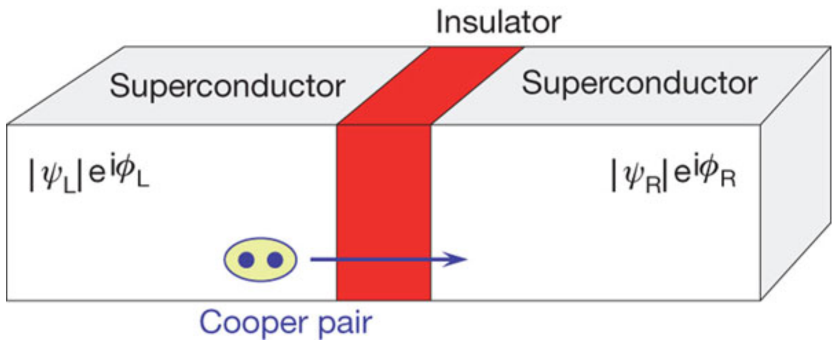

Figure 1: Josephson Junction S-I-S system

Described by the following equations:

$$\phi_2 - \phi_1 = c \tag{1}$$

$$j_s = j_c \sin(\phi_2 - \phi_1) \tag{2}$$

The Josephson Effect is a direct consequence of the Bardeen-Cooper-Schrieffer Theory of Superconductivity. That is, at suitably low temperatures, the electrons that generally make up the current suddenly "pair" up into pairs of electrons called a "Cooper Pair". As a result, all the Cooper Pairs can be described as a single macroscopic wave function $\sim \sqrt{n_s} e^{i\phi(r)}$ where r is the generalized coordinates that describe the evolution of the phase. This strange phenomena is the basis for superconductivity and is the reason why we see quantum phenomena on a macroscopic scale.

Typical applications of these junctions are for constructing Josephson Junction Arrays, which is the cornerstone of this paper. Generally, one can utilizing these arrays to study the behaviors of unconventional superconductors, or even used to perform Magnetic Resonance Imaging (MRI). Nonetheless, as we will see, Josephson Junctions Arrays are a powerful tool for detecting even the smallest of fluxes, and investigating into their behavior will allow a deeper understanding of how Josephson Junctions behave.

What is a Josephson Junction Random Junction Array?

A Josephson Junction Random Junction Array is composed of a n number of Josephson Junctions that span a rectangular border, connecting two superconducting electrodes together. Each junction has its own characteristic Josephson Junction Current-Phase Relationship (CPR) according to its dimensions and spatial location on the electrode. This CPR is nothing but Eq (2). A simple case is a 2-junction Superconducting Quantum Interference Device, or a SQUID. (Fig 2) This device has incredible flux-sensing abilities, generates an appreciable critical current graph when plotted against the applied magnetic flux close to flux quantums.

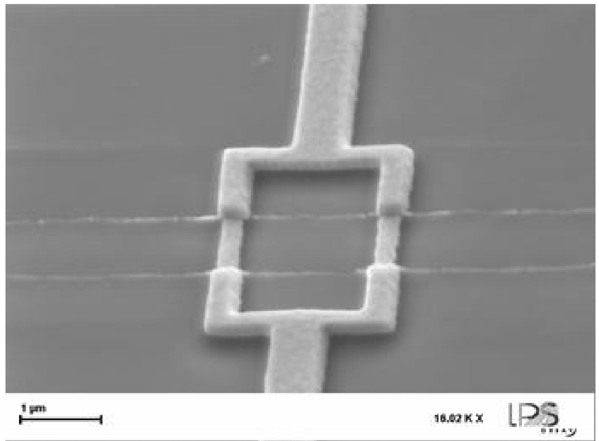

Figure 2: Superconducting Quantum Interference Device (or SQUID)

However, this paper is concerned with a general Josephson Junction Array (which is different than a SQUID!). With a SQUID, you'd be able to detect extremely small magnetic fluxes (like one flux quanta), and the resulting critical current generated by the SQUID can tell you which type of flux the device is reading (half-integer/integer multiple of flux quantum).

However, with a general JJ-Array, the critical current graphs produced are more chaotic and have experimentally demonstrated strange phenomena that were previously unresolved. This will be the focus of this paper. Specifically, we will be analyzing various symmetries and chaotic behaviors in the critical current graphs produced by the Josephson Junction Array when increasing the junctions' widths' standard deviation.

But before we begin our technical analysis of the general Josephson Junction Array, let's first begin by deriving the Josephson Effect.

Derivation

To begin this derivation, we first see that the cooper pairs within a superconductor can be described as a singular macroscopic wave function due to the consequences of the Bardeen–Cooper–Schrieffer (BCS) Theory. That is because the cooper pairs behave like Bosons and will condense to a lower ground state due to it being energetically favorable. Thus, we see that the wave function can be shown as:

$$\psi = \sqrt{n_s} e^{i\phi(r)} \tag{3}$$

With n_s being the particle density. We see that squaring the wave function yields the particle density. Then, let's assume a two body superconducting system with 2 superconductors placed roughly 1 nanometer apart from each other. Then the wave functions of each superconductor can be described as the following:

$$i\hbar\psi_1 = H_1\psi_1 \tag{4}$$

$$i\hbar\psi_2 = H_2\psi_2 \tag{5}$$

Next, let's couple the superconductors, utilizing perturbation theory, we see that the wave functions can be modified to satisfy the following:

$$i\hbar\psi_1 = E_1\psi_1 + \kappa\psi_2 \tag{6}$$

$$i\hbar\psi_2 = E_2\psi_2 + \kappa\psi_1 \tag{7}$$

Where κ is the coupling constant. The Hamiltonian has been replaced by Eigenvalues E_1 and E_2 since the Hamiltonian can no longer describe the total energy of the wave function. Now, measuring the voltage drop across the 2 superconductors is simply just:

$$E_2 - E_1 = -2eU \tag{8}$$

Where U is the Voltage across this insulator. If we plug in the macroscopic wave equation of the Cooper pairs into (2) and (3), we arrive at 2 separate equations from the real and imaginary parts of the equation:

$$\hbar\dot{\psi_1}\sqrt{n_{s,1}}\sin\phi_1 - \hbar\frac{\dot{n_{s,1}}}{2\sqrt{n_{s,1}}}\cos\phi_1 = \sqrt{n_{s,1}}E_1\sin\phi_1 + \sqrt{n_{s,2}}\kappa\sin\phi_2 \tag{9}$$

$$\hbar\dot{\psi_1}\sqrt{n_{s,1}}\cos\phi_1 - \hbar\frac{\dot{n_{s,1}}}{2\sqrt{n_{s,1}}}\sin\phi_1 = \sqrt{n_{s,1}}E_1\cos\phi_1 + \sqrt{n_{s,2}}\kappa\cos\phi_2 \tag{10}$$

In this expansion, we've assumed that $n_{s,1}$ and ϕ are time dependent. Thus, performing some equation manipulations, we arrive at the following equalities:

$$\dot{\phi_1} = \frac{E_1}{\hbar} + \frac{\kappa}{\hbar}\sqrt{\frac{n_{s,2}}{n_{s,1}}}\cos(\phi_2 - \phi_1) \tag{11}$$

$$\dot{n_{s,1}} = \frac{2\kappa}{\hbar}\sqrt{n_{s,1}n_{s,2}}\sin(\phi_2 - \phi_1) \tag{12}$$

Similarly, for ϕ_2 and $n_{s,2}$ we arrive at the following equalities:

$$\dot{\phi_2} = \frac{E_2}{\hbar} + \frac{\kappa}{\hbar}\sqrt{\frac{n_{s,2}}{n_{s,1}}}\cos(\phi_2 - \phi_1) \tag{13}$$

$$\dot{n_{s,2}} = -\frac{2\kappa}{\hbar}\sqrt{n_{s,1}n_{s,2}}\sin(\phi_2 - \phi_1) \tag{14}$$

Finding the difference between ϕ_1 and ϕ_2 gives us the following equality:

$$\hbar(\dot{\phi_1} - \dot{\phi_2}) = E_1 - E_2 = -2eU \tag{15}$$

Thus, we arrive at our first Josephson Equation. Allowing there to be no applied voltage due to the nature of a Superconductor, we see that:

$$\dot{\phi_1} - \dot{\phi_2} = 0 \implies \phi_1 - \phi_2 = c \tag{16}$$

For c $\in \mathbb{R}$. This implies that the following equality in equations (10) and (12):

$$\dot{n_{s,1}} = -\dot{n_{s,2}} \tag{17}$$

Assuming $\dot{n_{s,1}}$ and $\dot{n_{s,2}}$ are constant, or else the superconductors would receive a voltage, we arrive at the following equality:

$$\dot{n_{s,1}} = -\dot{n_{s,2}} = k$$
$$\delta n_{s,1} = \delta t \frac{2\kappa}{\hbar}\sqrt{n_{s,1}n_{s,2}}\sin(\phi_2 - \phi_1)$$
$$n_{s,1} = \frac{2\kappa}{\hbar}\sqrt{n_{s,1}n_{s,2}}\sin(\phi_2 - \phi_1)$$
$$\text{Let } j_c = A \times \frac{2\kappa}{\hbar}\sqrt{n_{s,1}n_{s,2}} \text{ and } j_s = A \times n_{s,1}$$
$$j_s = j_c \sin(\phi_2 - \phi_1)$$

Thus we have arrived at the 2nd Josephson Equation:

$$j_s = j_c \sin(\phi_2 - \phi_1) \tag{18}$$

This concludes the derivation. ■

Symmetry Analysis

Now that we've covered what a Josephson Junction Array is and walked through the derivation, let's start unraveling my algorithm for calculating the critical current. [GitHub Repository].

More specifically, the main focus of this section will be to make sense of the "current" function within the model:

```
26 def current(B, arrJ, arrC, y, numOfSegments):
27 # y is initial phase difference of the whole circuit, B is the magnetic field
28 #arrJ is the location of junctions, arrC is critical current associated with each junction
29
30     curr = 0 # summation of all currents in the entire junction
31
32     limit = int(len(arrJ) / 2) # number of junctions in the SQUID
33
34     for n in range(limit):
35         sizeOfSegment = float((arrJ[2 * n + 1] - arrJ[2 * n]) / numOfSegments)
36         for i in range(numOfSegments):
37             curr += arrC[n] * np.sin(y + (2 * np.pi * B) * (arrJ[2 * n] + i * sizeOfSegment)) * (1/numOfSegments)
38
39     # phase difference evolves according to 2 * pi * B
40
41     # curr += (critical current element in array)(sin(y + (2 * pi * B) * length)
42
43     return curr
```

Figure 3: Josephson Junction Array Current Function

Algorithm

To understand this function, we introduce a new equation describing the Meissner Phase Evolution across a Superconducting Electrode.

$$\Delta\phi = \phi_2 - \phi_1 = \Delta\phi_0 + 2\pi b x \tag{19}$$

That is, the phase difference $\phi_2 - \phi_1$ evolves spatially across the Superconducting Electrode with factor of $2\pi b$. In this case, b is a normalized flux, defined as $\frac{\Phi}{\Phi_0}$ where Φ_0 is the flux quanta and Φ is the applied magnetic flux. x is the normalized distance of the Array ranging from 0 to 1, where the length is thought to be 1 normalized unit length. This allows for the critical current graphs generated to be easily generalized to other systems. Clearly, (19) is displayed in line 37, but takes on a different form.

In order find the total sum across each junction, I utilized a numeric integration across each junction that has a super-current, and sum the mini $\sin\Delta\phi$ dx's across the junction according to (19). A numeric integration, apposed to a direct integration, produces more accurate graphs since a direct integration could cause floating point errors due to precision with the $\frac{1}{2\pi B}$ term. The rest of the program is focused around (19), and critical currents are found by varying the $\Delta\phi_0$ term and finding the corresponding maximum current associated with that specific Magnetic Field. Then, the program plots it using the Matplotlib Python Package.

In addition, the critical current densities signified by the 'arrC[n]' are made to be directly proportional to the junctions' widths. This allows us to predict phenomenon such as the "node-lifting" effect, representing the "shifting" up of the entire critical current graph, eliminating all the "zeros" of the graph.

Results

For a simple SQUID described by the array [0, 0.001, 0.999, 1], where the junctions would span 0 - 0.001, & 0.999 - 1. with a mean junction width of 0.001, a width $\sigma = 0$, the critical current graph is the following:

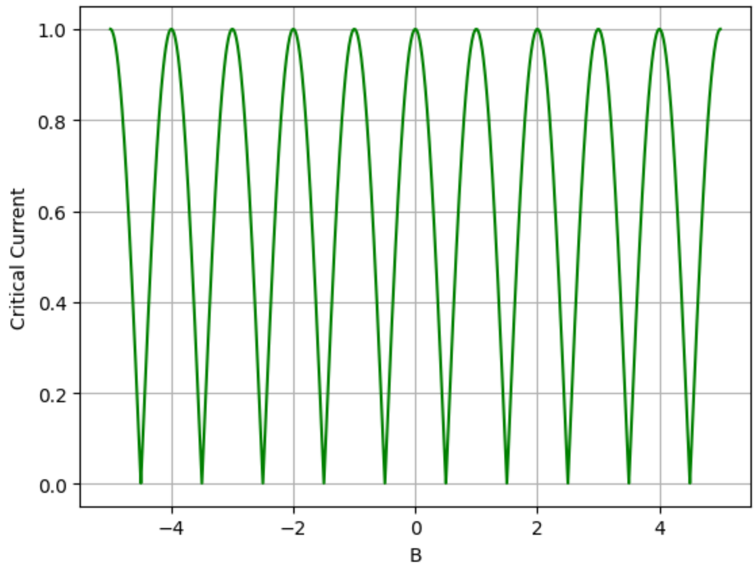

Figure 4: 2-Junction SQUID Critical Current graph

A more complex Josephson Junction Array with 5 symmetric, identically small-width junctions described by the array [0, 0.001, 0.25, 0.251, 0.5, 0.501, 0.75, 0.751, 0.999, 1] with a mean junction width of 0.001 and a width $\sigma = 0$, the critical current graph is the following:

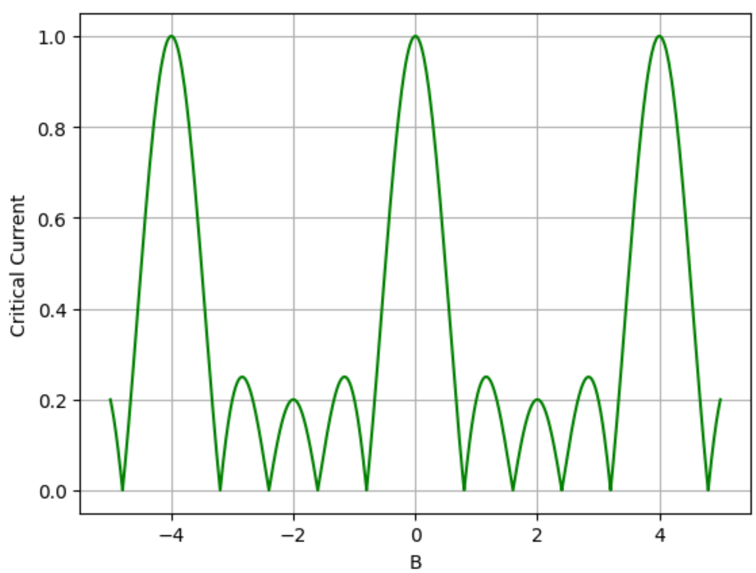

Figure 5: 5-Junction Array Critical Current graph

For a 2 junction Array described by the array of [0, 0.02, 0.99, 1] with a mean junction width of 0.015 and a width $\sigma = 0.005$, the critical current graph is the following:

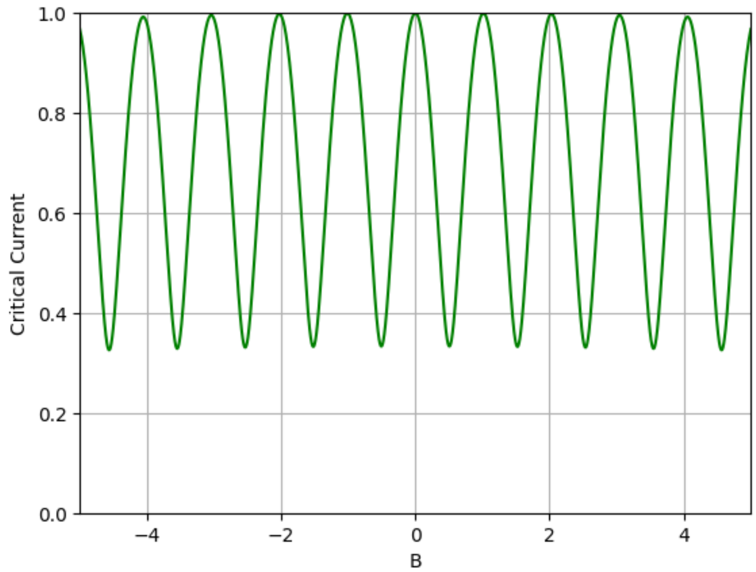

Figure 6: 2 Asymmetric Junction Array Critical Current graph

The massive node-lifting effect aligns with the theoretical predicted critical current graphs derived in Michael Tinkham's *Introduction to Superconductivity* Book.

For a chaotic 3 junction array described by the array [0, 0.2, 0.4, 0.6, 0.9, 1], with a mean junction width of 0.167 and a width $\sigma = 0.047$ the critical current graph is the following:

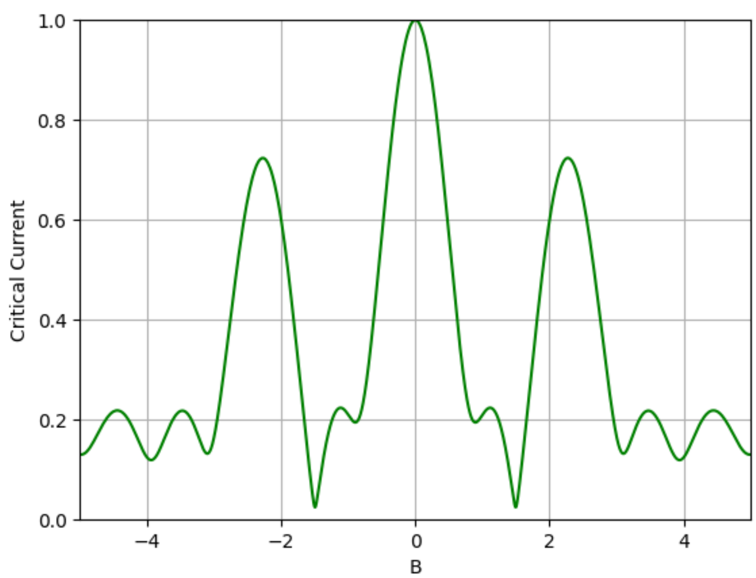

Figure 7: Asymmetric 3 Junction Array Critical Current graph

Symmetries

The Josephson Junction Array exhibits exceptional symmetry. For a symmetric, identically small-width JJ-Array, the patterns produced are periodic and are extremely symmetric. However, even with a random Array with a width standard deviation $\neq 0$, there will always be a symmetry across the y axis. That implies that the JJ-Array doesn't "see" the minus sign which is certainly an interesting phenomenon. A simple informal proof can be made to explain this phenomenon:

Proof. Suppose that we have an n-junction JJ-Array with a CPR described as the following:

$$I_s = \sum_{i=1}^{n} J_{c,i} \sin(\Delta\phi_i) \tag{20}$$

Where I_c is the observed super-current, $J_{c,i}$ is the super-current density at that specific junction, and $\Delta\phi_i$ is the phase difference across that junction. Then suppose that we have some critical current at the phase differences:

$$I_c = \sum_{i=1}^{n} J_{c,i} \sin(\Delta\phi_i) \tag{21}$$

This can be expanded out to the following:

$$I_c = \sum_{i=1}^{n} J_{c,i} \sin(\Delta\phi_0 + 2\pi b x_i) \tag{22}$$

Next, suppose we apply a negative flux of the same magnitude, but of different sign. We expand out the term to be the following:

$$I_c = \sum_{i=1}^{n} J_{c,i} \sin(\Delta\phi_0 - 2\pi b x_i) \tag{23}$$

But $\Delta\phi_0$ is a free-variable, so we can choose $\Delta\phi_0$ to be $\Delta\phi_0 + 4\pi b x_i$, which yields our y-axis reflected critical current value. Thus, this concludes the proof. □

Clearly, this isn't a formal proof, but it gives some insight into why this symmetry always exists across the y axis. Moreover, we can also plot the differences $I(B) - I(-B)$ to further convince you of this symmetry. For a 5 junction array, this graph is the following:

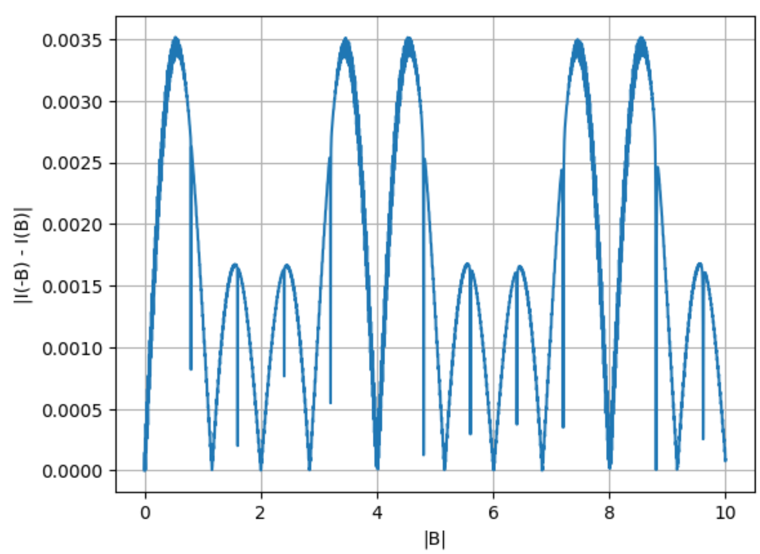

Figure 8: Symmetry across the y-axis for Fig 5

For the random 3 junction array, this plot looks like the following:

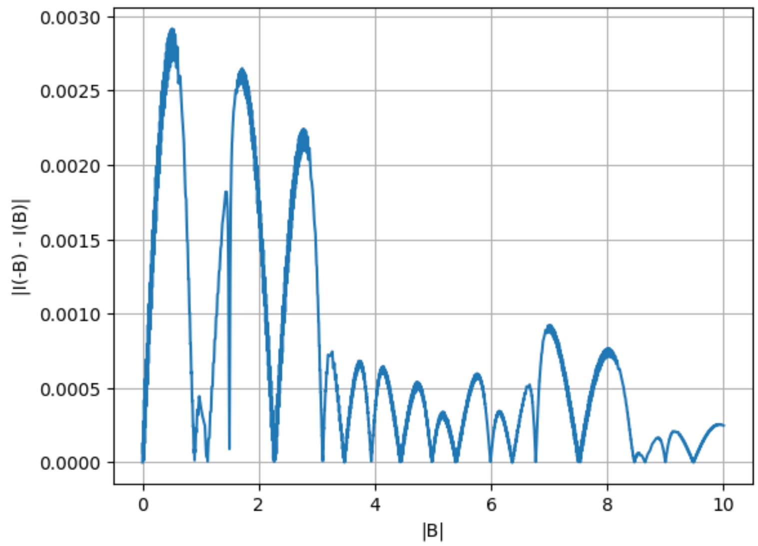

Figure 9: Symmetry across the y-axis for Fig 7

With the y axis representing the differences at the points of $-B$ and B, we notice that very insignificant values are plotted against $|B|$. This can be explained with the nature of the performed numeric integration, where small differences arise due to the small error produced when numerically integrating across the junction. This explains why this difference is non-zero. However, since this value is so small, we can conclude that a guaranteed symmetry exists across the y axis regardless of the configuration of the Josephson Junction Array.

Remarks

If you are familiar with the Light Diffraction Experiment and the graphs it generates, you'll notice that the pattern of the Critical Current graphs generated by symmetric, identically small-width JJ-Arrays are *very* similar to the light diffraction graphs. In fact, the critical current graphs for these specific Array configurations are *completely analogous* to the Light Diffraction Experiment in Quantum Mechanics.

This interesting phenomenon that arises from a seemingly completely unconnected Superconducting Array gives insight into the behavior of these Josephson Junctions, and how they tend to react to the externally applied flux.

Note: For those interested in playing around with this program, click the link in the beginning of this section to access the GitHub Repository. The title of the program is "JJ Array Model"

Chaos Analysis

From the previous couple of plots, you may or may not have noticed that in general, Josephson Junction Arrays with a width $\sigma \neq 0$ manifest a "node-lifting" effect in its critical current graphs, representing the "zeros" of the critical current function lifting up. This strange result is called the *Node-lifting phenomena* and has been observed experimentally. However, the cause of this effect has puzzled Condensed Matter Experimentalists for some time now. However, in this project, it was hypothesized that increasing the junctions' widths standard deviation (or randomness) would lead to this node-lifting effect. But now it comes down to the main question of:

How do we measure this phenomena?

Obviously, there are many, many ways of measuring this phenomena, like just manually making the junctions' widths more chaotic and looking at the graphs. However, I wanted to create a method that was a little more quantitative and not just "plug and chug" values in hopes to increase the junction widths' standard deviation. As an idea, I decided to try to attempt to formulate this algorithm that increases the junctions' widths' standard deviation a specified amount by the user. Then the program would plot the corresponding critical current graphs and noticing the critical current deviations from the unperturbed JJ-Array. This is quite an ambitious task, but as you'll see later, is actually possible.

Then, with this powerful program, we can generalize this "node-lifting" phenomenon analysis to a Chaos analysis, and seeing how the JJ-Array's generated critical current graphs vary when you increase its junctions' widths' standard deviation. But before I show the results of this model, I'll first derive the exact formula I used to obtain an array of junction widths given an inputted standard deviation:

Derivation

We begin this derivation with the definition of standard deviation:

$$\sigma = \sqrt{\frac{\Sigma(x-\mu)^2}{n}} \tag{24}$$

Where μ is the mean of the this population and n is the number of objects in this population. Performing some calculations, we see that:

$$\begin{gathered}\sigma = \sqrt{\frac{\Sigma(x-\mu)^2}{n}} \\ \implies n\sigma^2 = \Sigma(x-\mu)^2 \\ \implies n\sigma^2 = (x_1-\mu)^2 + (x_2-\mu)^2 + \dots\end{gathered}$$

Assuming a linear evolution within the $(x_n-\mu)^2$ terms, we have that:

$$\begin{gathered}n\sigma^2 = (x_1-\mu)^2 + (x_2-\mu)^2 + \dots \\ \implies n\sigma^2 = \delta_1 + \delta_2 + \delta_3 + \cdots + \delta_n\end{gathered}$$

Assuming a perfect Gaussian distribution allows us to assume axial symmetry across y = 0. Thus we see that each δ term can be multiplied by two, this will simplify our calculations. Performing some calculations, we see that:

$$\begin{gathered}n\sigma^2 = \delta_1 + \delta_2 + \delta_3 + \cdots + \delta_n \\ \implies \frac{n\sigma^2}{2} = \delta_1 + \delta_2 + \cdots + \delta_{\frac{n}{2}}\end{gathered}$$

However, with our assumption of a perfect Gaussian distribution coupled with our assumption of a perfect axial symmetry across y = 0; we can now deduce that the set $\Delta = \{\delta_1, \delta_2, ..., \delta_{\frac{n}{2}}\}$ consists of strictly increasing terms. Thus, I propose that fitting any ordinary evolution function $f(x)$ such that $f(x)$ is strictly increasing and assigning values to each δ_k will satisfy this formula as long as it is written in terms of δ_1. Therefore, I propose and prove a theorem of my statistical fitting law:

Theorem 1. *In the environment of an axial symmetric Gaussian distribution, given the formula*

$$n\sigma^2 = \delta_1 + \delta_2 + \delta_3 + \ \dots \ + \delta_n \tag{25}$$

We can reduce the term to be

$$\frac{n\sigma^2}{2} = \delta_1 + \delta_2 + \delta_3 + \ \dots \ + \delta_{\frac{n}{2}} \tag{26}$$

We notice that the set $\Delta = \{\delta_1, \delta_2, ..., \delta_{\frac{n}{2}}\}$ *is strictly increasing. Thus, fitting any arbitrary strictly increasing function* $f(x)$ *to the set* Δ *will produce the same corresponding* σ *as long as you write the* δ_2, δ_3, ..., $\delta_{\frac{n}{2}}$ *in terms of* δ_1 *such that*

$$\frac{n\sigma^2}{2} = \delta_1 + c_1\delta_1 + c_2\delta_1 + \ \dots \ + c_{\frac{n}{2}-1}\delta_1 \tag{27}$$

Where $\delta_2 = c_1\delta_1$, $\delta_3 = c_2\delta_1$, ..., *&* $\delta_{\frac{n}{2}} = c_{\frac{n}{2}-1}\delta_1$

Proof. Suppose that

$$\sigma = \sqrt{\frac{\Sigma(x_i - \mu)^2}{n}} \tag{28}$$

Assuming an axial symmetric Gaussian distribution across $y = 0$, we can compress this equation to be the following:

$$\sigma = \sqrt{\frac{2 \cdot \Sigma(x_i - \mu)^2}{n}} \tag{29}$$

Where $(x_i - \mu)^2$ is strictly increasing. Then choose some δ such that

$$\delta = \frac{n\sigma^2}{2(1 + c_1 + c_2 + \ \dots \ + c_{\frac{n}{2}-1})} \tag{30}$$

Where $c_1 < c_2 < c_3 < \ \dots \ < c_{\frac{n}{2}-1}$ without loss of generality. Then we have that $\delta = (x_1 - \mu)^2$, $c_1\delta = (x_2 - \mu)^2$, etc. Plugging that into equation (24) yields

$$\sigma = \sqrt{\frac{2\delta + 2c_1\delta + 2c_2\delta + \ \dots \ + 2c_{\frac{n}{2}-1}\delta}{n}} \tag{31}$$

But it follows that because of equation (25), we have the following equality:

$$\sigma = \sqrt{\frac{2\delta(1 + c_1 + c_2 + \ \dots \ + c_{\frac{n}{2}-1})}{n}} \tag{32}$$

Plugging in equation (26), we see that all of the values of the square root cancel out, and yields us

$$\sigma = \sqrt{\sigma^2} \iff \sigma = \sigma \tag{33}$$

And this cancellation of the square roots is valid since the standard deviation of a distribution will always be positive. This concludes the proof. □

Allowing $f(x) = x$, and choosing $\delta_1 = 1$ at x = 1, and allowing δ_k be equal to $f(k) = k\delta$ for $k \in \mathbb{N}$, we see that $\delta_2 = 2\delta_1$, $\delta_3 = 3\delta_1$, ..., $\delta_{\frac{n}{2}} = \frac{n}{2}\delta_1$. Thus we see that our formula can be rewritten as the following:

$$\tfrac{n\sigma^2}{2} = \delta + 2\delta + 3\delta + ... + \tfrac{n}{2}\delta$$

Where δ_1 has been simplified to δ. Applying the Gaussian summation formula yields us:

$$\tfrac{n\sigma^2}{2} = \delta\left(\tfrac{\frac{n}{2}(\frac{n}{2}+1)}{2}\right)$$
$$\implies \tfrac{n\sigma^2}{2} = \delta\left(\tfrac{n(n+2)}{8}\right)$$
$$\implies \delta = \sigma^2\left(\tfrac{4}{n+2}\right)$$

Thus we arrive at our final equation, assuming that $n \neq 0$, describing δ:

$$\delta = \sigma^2\left(\frac{4}{n+2}\right) \tag{34}$$

Using this equation, we can derive the various x_i terms using this delta such that:

$$x_i = \mu \pm \sigma\sqrt{i\frac{4}{(n+2)}} \tag{35}$$

Similarly, for odd junctions (with the assumption of a zero middle and that $n \neq 0$), we have that:

$$\delta = \sigma^2\frac{4n}{(n+1)(n-1)} \tag{36}$$

and solving for the various x_i terms yields:

$$x_i = \mu \pm \sigma\sqrt{i\frac{4n}{(n+1)(n-1)}} \tag{37}$$

And if you utilize this formula, and find the various x_n terms given some mean, then reflecting the values over the beginning value and multiplying by -1. You'll see that the corresponding σ and μ will match the inputted σ and inputted μ. Thus, this concludes this derivation. ■

Mini Problem + Easy Solution

One issue that I ran into when creating this algorithm was that half of the width array was negative, and junctions can't have a negative width! So as an easy fix, I shifted all the values up by a factor of the minimum value + some user inputted value, such that the rest of the widths are positive. Then we're able to apply this width array into the array to be perturbed.

Algorithm

The algorithm for this program is relatively straight forward, or as straight forward as it can be. However, we must address two separate cases. The first case is that when the user inputs a set number of junctions and a mean width, and another where the user inputs a custom array describing a more chaotic Josephson Junction Array. Let's first begin by addressing the easier case: the former.

Case 1: User-specified # of junctions & mean junction width

We begin by generating an array of widths that correspond to the user-inputted standard deviation. We then shift the values up such that the minimum value is now the user-inputted mean width. Then we go about the array and randomly apply these new widths from the original junction center. Think about expanding from the original junction center and ending up with a symmetric junction about the center, but with a different width. This new width would be one of the generated widths that correspond with the inputted standard deviation.

Case 2: User-specified custom JJ-Array

This case is a bit harder, but still nothing too challenging. We begin by finding the original standard deviation of the user-inputted array. Then we add the user-inputted standard deviation to this value, then generate and shift the corresponding junction widths. We then apply the same algorithm as stated above, randomly applying these new widths all throughout the array by expanding from the calculated junction center. In this case, the calculated junction center would be the junction centers of the user-inputted array.

Results

In the legend on the top right of the graphs, the standard deviation (listed as "Sigma") is recorded with the percentage of space the areas without a super-current occupies.

Applying this program to the SQUID in Fig 4, I set the standard deviations to be $\sigma = +0, +0.01,$ and $+0.02$

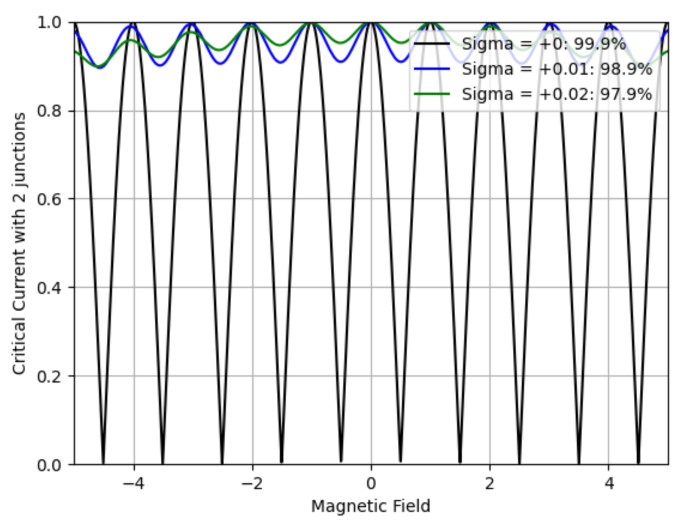

Figure 10: SQUID [Fig 4]: σ's = +0, +0.01, & +0.02

Take note of the massive node-lifting occurring.

Applying this program for the 5 Junction Array in Fig 5 with the same σ's, the graph comes out to be:

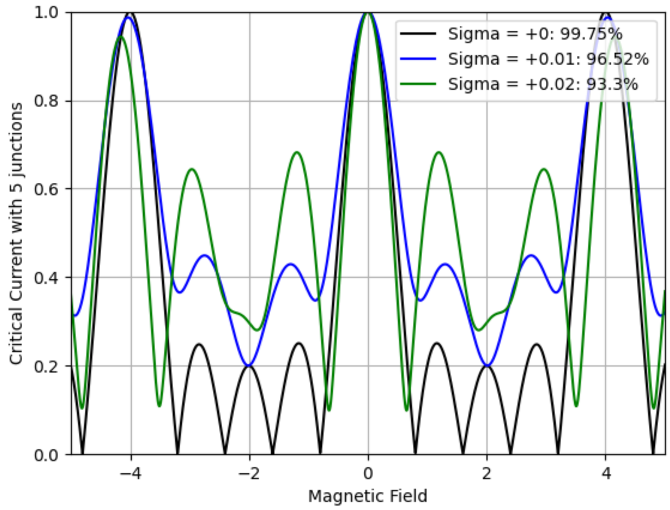

Figure 11: 5-Junction Array [Fig 5]: σ's = +0, +0.01, & +0.02

Notice the node-lifting occurring. Also note that it's not nearly as severe as the SQUID.

Applying this program to the Chaotic 3-Junction Array in Fig 7 with the same σ's, the code produces this graph:

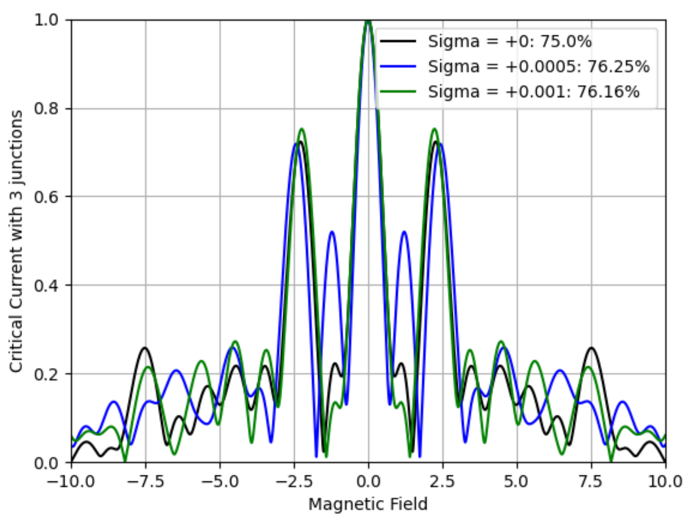

Figure 12: Chaotic 3-Junction Array [Fig 7] σ's = +0, +0.01, & +0.02

Chaos

We note that in general, increasing the junctions' widths' standard deviation manifests a "Node-lifting" Effect, representing the "zeros" of the critical current graphs lifting up. Additionally, we notice that the SQUID appears to be more sensitive to increases in the standard deviation. This result can be generalized such that symmetric, identically small-width JJ-arrays with a smaller amount of junctions are generally more sensitive to increases in the standard deviation. This result can be explained by thinking about the impact that the largest junction has in the context of its home JJ-Array. In this argument, we are considering a symmetric, small-width JJ-Array, and perturbing by the same standard deviation.

For a smaller number of Josephson Junctions, the largest Josephson Junction has a larger impact on the critical current graph since it would make up a larger fraction with respect to the total space occupied by all junctions. However, with a larger number of Josephson Junctions, the largest junction can't really affect the critical current too appreciably since its fraction with respect to the total space occupied by all the junctions. Thus, this argument does explain the reasoning behind why a smaller number of junctions in a symmetric, identically small-width junctions appears to be more sensitive to perturbations in the standard deviation.

As well, we notice that the custom-inputted JJ-Array doesn't really deviate too much from the case of $\sigma = 0$. This makes sense, however, when you consider the fact that the mean width of the junctions is relatively large (~ 0.15), so changes in the standard deviation by 0.01 and 0.02 wouldn't necessarily impact the critical current too much.

So we conclude that in general, a Node-lifting effect occurs when the junctions' widths' standard deviation in a symmetric, identically small-width Josephson Junction Array increases. As well, the "amount" of node-lifting generally decreases when the number of junctions increases, the reasoning is postulated to be related to the fraction of space occupied by the largest junction with respect to the total space occupied by all the junctions. This conclusion can be generalized to any JJ-Array by stating that if the mean width and the number of junctions are sufficiently large, then the array will be resistant to perturbations in the junctions' widths standard deviation. In other words, these types of arrays exhibit non-sensitive behavior.

In terms of chaos, the JJ-Arrays exhibit non-chaotic behaviors when increasing the widths' standard deviation since the shape and the amount of "valleys" in between the mountains are still preserved. However, critical current sensitivity in the JJ-Arrays appears to be an interesting topic to investigate.

Concluding Thoughts

The Josephson Junction Random Junction Array exhibits exceptional symmetry. For a symmetric, identically small-width JJ-Array, the patterns produced are periodic and are extremely symmetric. Additionally, there will always be a symmetry across the y axis regardless of the Array configuration.

In terms of chaos, the JJ-Arrays exhibit non-chaotic behaviors when increasing the widths' standard deviation. However, critical current sensitivity in the JJ-Arrays appears to be an interesting topic to investigate.