# SINGLE ELECTRON TUNNELING JUNCTIONS

PAUL RIECHERS

*Physics Department*
*University of California at Davis,*
*Davis, CA 95616*
*pmriechers@ucdavis.edu*

This paper and its associated Python programs explore the dynamics of single electron tunneling junction (SETJ) circuits. SETJs can potentially be used as nonlinear circuit elements to implement new paradigms for information processing. I show that an isolated SETJ circuit with a constant bias voltage and sinusoidal pump offset can produce a variety of interesting nonlinear behavior. By adjusting certain input parameters, the frequency and phase of tunneling events can become entrained to any super-harmonic of the pump, although the robustness of the basins of attraction typically diminishes as we increase the tunneling period. Capacitively coupled SETJs can interact to display new types of behavior, as the voltage across one tunneling junction will influence the voltage of its neighbors. After looking at the behavior exhibited by two-coupled SETJs, I examine the mutual interaction and tunneling outputs of arbitrarily large MxN SETJ arrays.

*Introduction*

Single electron tunneling junctions (SETJs) offer an opportunity to not only downsize electronic components, but also to enhance the functionality of large arrays of components by exploiting their nonlinear dynamics to access an emergent level of information processing. However, nanoparticle arrays of capacitively coupled SETJs have many interacting variables and associated parameters that gives rise to a complexity that is useful only to the extent that we understand these dynamics. Therefore, to better understand the range of behavior allowed by such tunneling junctions, I have studied the dynamics of isolated SETJ circuits, two-coupled SETJ circuits, and finally one-dimensional and two-dimensional systems of arbitrary size. I show how one can alter the behavior of an isolated SETJ, and explore how parameter values can change the behavior of large SETJ networks.

*Background*

Sandwiching a molecule with metal contacts forms a single capacitive junction. In the simplest isolated SETJ circuit, this capacitive junction is in series with a resistive element. Applying a voltage across these two elements would seem to yield a simple RC circuit, except that a single electron will tunnel through the tiny junction once a critical voltage (~ e / 2C$_{\text{junction}}$) has built up across the junction. Therefore, the isolated SETJ circuit behaves as a relaxation oscillator, discharging a single electron every time a threshold voltage has been obtained across the metal-molecule-metal junction. Figure 1, borrowed from Yang et al., shows a schematic circuit diagram of the isolated SETJ circuit [1]. The system becomes interesting because SETJs can be capacitively coupled to form a dense array of interacting tunneling junctions. The dynamics are further enriched since each SETJ can be driven by an applied voltage at one of its terminals.

1092 *T. Yang & L. O. Chua*

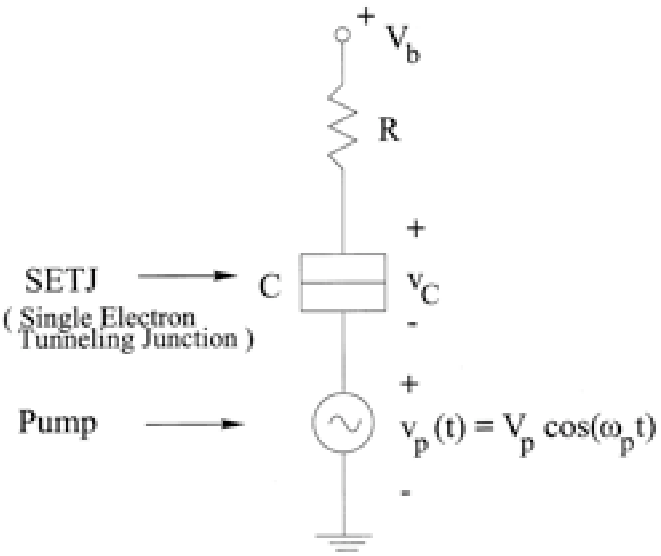

+ $V_b$
R
+
C
$v_C$
-
SETJ
( Single Electron Tunneling Junction )
Pump
+
$v_p(t) = V_p \cos(\omega_p t)$
-

Fig. 1. An isolated driven single-electron tunneling junction circuit.

*Dynamical System*

SETJ arrays are nonlinear and time-dependent dynamical systems. A tunneling event in one SETJ will discontinuously change the tunneling junction voltage of other coupled SETJs, implying that both discrete and continuous time phenomena are of interest, and a combination of differential equations and discrete difference equations must be used to accurately model the system. The primary variables of interest are the voltage across the tunneling junctions and the corresponding phase of a tunneling event.

The phase of a tunneling event can be described in various ways, but can be given as either a binary or continuous variable that takes on a new value for each tunneling event. The tunneling phase depends on when the last tunneling event took place in relation to the phase of the pump voltage. I chose to represent the grey-scale tunneling-phase output as:

$$f_{grey-scale}(\tau_{ij}^{last},\varphi(t)) = \frac{\tau_{ij}^{last} \bmod 2\pi}{2\pi}$$

and I represented the binary tunneling-phase output as:

$$f_{binary}(\tau_{ij}^{last},\varphi(t)) = \quad 1 \text{ if } \tau_{ij}^{last} \bmod 2\pi > \pi, \text{ or } \quad 0 \text{ if } \tau_{ij}^{last} \bmod 2\pi < \pi$$

where $\tau_{ij}^{last}$ is the time of the last tunneling event in the SETJ cell indexed as *ij*. Therefore, this definition applies to any SETJ cell, whether it is in an isolated circuit, one-dimensional lattice, or two-dimensional array. The following discussion of the dynamics describes the system in terms of the physically natural variables, such as voltages at nodes, or across the junctions and coupling capacitors (except that these variables are algebraically simplified to reveal the essential dynamical characteristics). However, given the physical evolution of the system, the aforementioned output variables, $f_{grey-scale}$ and $f_{binary}$, can easily be found.

An isolated SETJ circuit can be modeled as shown in Figure 1. For this case of a constant bias voltage, $V_b$, applied at one end of the cell, and a sinusoidal driving voltage applied at the other, the nonautonomous impulsive differential equation for the voltage across the junction, $v_C$, is:

$$\begin{aligned}\frac{dv_C(t)}{dt} &= \frac{1}{RC}(V_b - v_C(t) \\ &\quad - V_p \cos(\omega_p t)), \quad \text{if } |v_C(t)| < V_T, \\ \Delta v_C(t_i) &\triangleq v_c(t_i^+) - v_c(t_i) \\ &= \begin{cases} -2V_T, & v_c(t_i) \geq V_T, \\ 2V_T, & v_c(t_i) \leq -V_T, \end{cases}\end{aligned}$$

[1].

Notice that these equations consist of a continuous differential equation (the differential equation for *t* is implicitly assumed to be unity) and a simple discrete difference equation. The differential equation suffices to model the voltage across the junction when its magnitude is less than the threshold tunneling voltage, $V_T$. However, when the voltage across the junction exceeds $V_T$, it is energetically favorable for an electron to tunnel to the other side of the metal-molecule-metal junction. This tunneling behavior is modeled by the discrete difference equation, and is assumed to happen immediately when $v_C > V_T$. This is clearly an approximation, since each tunneling event is stochastic in nature (the events will be distributed near $V_T = e/2C$) [2], but for large resistances, R, this turns out to be a good approximation [3].

Although the previous interdependent set of equations most directly represents the isolated SETJ circuit with its natural variables, it is convenient to use algebraically simplified equations to isolate the dynamical activity and adjustable variables. Again borrowing notation and equations from Yang et al., the previous equations can be rewritten as:

$$\begin{aligned} \frac{d\theta}{dt} &= -\frac{\gamma}{\pi}\theta + \gamma(a - b\cos t)\,, \quad t \neq \tau_i\,, \\ \Delta\theta(\tau_i) &= -2\pi\,, \quad t = \tau_i\,, \\ |\theta(0)| &< \pi\,, \end{aligned} \quad \text{where} \quad \begin{aligned} a = \frac{V_b}{V_T}\,, \quad b = \frac{V_p}{V_T}\,, \quad \theta = \frac{2\pi C v_C}{e}\,, \\ \tau = \omega_p t\,, \quad \gamma = \frac{\pi}{RC\omega_p}\,. \end{aligned} \qquad [1]$$

This is similar in nomenclature to the dimensionless forms used for higher dimensional arrays (remember the isolated SETJ circuit is still zero-dimensional, as far as arrays are concerned), however each neighbor that an SETJ cell is coupled to brings along a new coupling capacitor with an accompanying voltage across it that will change the dynamics of the original system. For simplicity, a nearest neighbor approximation has been made, so that only the nearest SETJ neighbors, and the coupling capacitors between them, instantaneously influence a given SETJ cell. This is reminiscent of a similar approximation used in solid-state physics for oscillators in a lattice coupled to nearest neighbors with springs. Because cells at the edge of the lattice have a different number of nearest neighbors, these cells will have different dynamical equations describing their behavior. Again returning to the solid-state analogy, these edge cells can be thought of as lattice defects, and introduce reflection and other 'non-ideal' behavior into the system of coupled oscillators. In short, circular boundary conditions are not used, but rather the boundary conditions are stated and implemented explicitly.

For the one-dimensional lattice, I have used a set of equations similar to Yang et al.'s:

$$
\begin{aligned}
\frac{d\theta_1}{ds} &= \gamma\left(a_1 - b\cos s - \frac{(k+1)\theta_1}{k\pi} + \frac{\theta_2}{k\pi} + \frac{\vartheta_1}{k\pi}\right), \quad \text{if } \theta_1 < \pi\,, \\
\frac{d\theta_2}{ds} &= \gamma\left(a_2 - b\cos s - \frac{(k+2)\theta_2}{k\pi} + \frac{\theta_1}{k\pi} + \frac{\theta_3}{k\pi} - \frac{\vartheta_1}{k\pi} + \frac{\vartheta_2}{k\pi}\right), \quad \text{if } \theta_2 < \pi\,, \\
\vdots \quad & \vdots \\
\frac{d\theta_i}{ds} &= \gamma\left(a_i - b\cos s - \frac{(k+2)\theta_i}{k\pi} + \frac{\theta_{i-1}}{k\pi} + \frac{\theta_{i+1}}{k\pi} - \frac{\vartheta_{i-1}}{k\pi} + \frac{\vartheta_i}{k\pi}\right), \quad \text{if } \theta_i < \pi\,, \\
\vdots \quad & \vdots \\
\frac{d\theta_n}{ds} &= \gamma\left(a_n - b\cos s - \frac{(k+1)\theta_n}{k\pi} + \frac{\theta_{n-1}}{k\pi} - \frac{\vartheta_{n-1}}{k\pi}\right), \quad \text{if } \theta_n < \pi\,, \\
\frac{d\vartheta_1}{ds} &= \frac{\gamma}{k\varepsilon\pi}\left(\theta_1 - \theta_2 - \vartheta_1\right), \\
\vdots \quad & \vdots \\
\frac{d\vartheta_i}{ds} &= \frac{\gamma}{k\varepsilon\pi}\left(\theta_i - \theta_{i+1} - \vartheta_i\right), \\
& \vdots \\
\frac{d\vartheta_{n-1}}{ds} &= \frac{\gamma}{k\varepsilon\pi}\left(\theta_{n-1} - \theta_n - \vartheta_{n-1}\right), \\
& \Delta\theta_i = -2\pi, \quad \text{if } \theta_i \geq \pi, \quad i = 1, 2, \ldots, n-1, n\,, \\
& \theta_i(0) < \pi\,, \quad i = 1, 2, \ldots, n-1, n\,.
\end{aligned}
\qquad [1],
$$

where theta again represents the tunneling voltage across the tunneling junctions in the lattice. Some new variables introduced are *s*, which is a normalized time, k, which represents the ratio of the coupling resistance and load resistance (i.e. $k = \frac{R_{coupling}}{R}$), $\varepsilon$, which represents the ratio of the coupling capacitance to the tunneling junction capacitance (i.e. $\varepsilon = \frac{C_{junction}}{C_{coupling}}$), and $\upsilon$, which represent the voltages across the N - 1 capacitors in a 1xN lattice.

For a two-dimensional lattice with four nearest neighbors, two more capacitors and two more SETJ cells with associated voltages influence each inner cell of the array. The dynamical equations are thus more complicated, and the boundary conditions are more numerous. The impulsive differential equations describing the behavior of a four-coupled inner cell of a two-dimensional MxN SETJ array, located at *ij* are:

$$
\begin{aligned}
\frac{dx_{ij}}{dt} &= \gamma \left( u_{ij} - b\cos t - \frac{(k+4)x_{ij}}{k\pi} + \frac{x_{i,j-1}}{k\pi} \right. \\
&\quad + \frac{x_{i,j+1}}{k\pi} + \frac{x_{i-1,j}}{k\pi} + \frac{x_{i+1,j}}{k\pi} - \frac{q_{ij}}{k\pi} \\
&\quad \left. + \frac{q_{i,j+1}}{k\pi} - \frac{p_{ij}}{k\pi} + \frac{p_{i+1,j}}{k\pi} \right), \quad \text{if } x_{ij} < \pi\,, \\
\frac{dp_{ij}}{dt} &= \frac{\gamma}{k\varepsilon\pi}\left(x_{i-1,j} - x_{ij} - p_{ij}\right), \\
\frac{dq_{ij}}{dt} &= \frac{\gamma}{k\varepsilon\pi}\left(x_{i,j-1} - x_{ij} - q_{ij}\right), \\
&\Delta x_{ij} = -2\pi\,, \quad \text{if } x_{ij} \ge \pi\,, \\
&x_{ij}(0) < \pi\,, \quad 1 < i < M\,,\ 1 < j < N\,, \qquad [1],
\end{aligned}
$$

where $x_{ij}$ now takes the place of theta as the voltage across the junction. $p_{ij}$ represents the voltage across the coupling capacitor between $x_{i-1,j}$ and $x_{i,j}$. $q_{i,j}$ represents the voltage across the coupling capacitor between $x_{i,j-1}$ and $x_{i,j}$. This is somewhat of a strange notation, however, since there are no $p_{1,j}$ or $q_{i,1}$ capacitor voltages in the array (although, for programming purposes and consistency, they can be considered present but always zero). Notice also, that the applied bias voltage term $a = \frac{V_b}{V_T}$, has been replaced with the new notation, $u_{ij}$, to emphasize that this voltage term can have a different value at each cell. $u_{ij}$ is potentially an important term for inputting information to be processed into the system. The boundary conditions for the two-dimensional array are too numerous to list here, so I refer the interested reader to a theoretical overview by Yang et al. [1], or an example of implementation in my *TPL_CNN.py* code, which includes many subclasses of SETJ types with appropriate boundary conditions.

The dynamical system above is a rather general description of a capacitively coupled MxN SETJ array. By varying parameters and array types, systems can be designed to exhibit a wide range of behavior. As one example of the generality of this setup, consider that the simpler non-driven SETJ array described by Kikombo et al. [3] can be realized by setting b = 0 (i.e., turning the pump off creates an array of coupled relaxation oscillators).

*Methods*

Initial efforts to simulate an SETJ circuit led me to develop *Isolated_TimeSeries.py*, which ended up becoming my tool for plotting time series, phase portraits, and limit cycles. For bifurcation diagrams and Poincare maps, I created a *OneDimMap* class with appropriate subclasses and modules for specific applications. These programs use an early set of Runge-Kutta (RK4) integrators that I developed that can store trajectory information and tunneling events, or can store minimal information to run through transient time steps. Since time is an explicit variable in the SETJ differential equations, time must be updated explicitly in the intermediate steps of the RK4 integrator (but, although explicit time evolution in RK4 is theoretically appropriate, an accidental experiment with the time evolution in RK4 revealed that its omission typically led to negligible discrepancies in trajectories since the time dependence varies so slowly for typical parameter settings). An n-dimensional RK4 integrator eventually supplanted the one-dimensional RK4 for the first several programs developed for this project.

*TPL_CNN.py* and *CNN_lattice_simulator.py* were developed as a universal tool for SETJ array design and manipulation. While good headway has been made towards this goal, there are many ways that I hope to enhance these programs now that the infrastructure is in place. *TPL_CNN.py* allows for the creation and manipulation of any MxN SETJ array. One of the main features of this program is the structure and hierarchy of SETJ objects. When an SETJ array is created, each object in the array inherits the characteristics of a generic SETJ class. However, each object is further specialized depending on its connectivity in the array. There are currently sixteen subclasses of the SETJ class, to account for all of the possible configurations and boundary conditions of an MxN SETJ array. The program accommodates zero-dimensional arrays (isolated SETJ circuits), one-dimensional arrays, and two-dimensional arrays. An easy way to create these arrays is to make an instance of the *SETJarray* class. As shown below, MyArray = SETJarray(M,N,parameters) automatically creates an MxN array with the correct boundary conditions that can then be displayed with the names of each SETJ type in the array.

```
In [16]: MyArray.array
Out[16]:
array([[SETJ top left corner cell, SETJ top row cell, SETJ top right corner cell],
    [SETJ leftmost column cell, SETJ inner cell, SETJ rightmost column cell],
    [SETJ leftmost column cell, SETJ inner cell, SETJ rightmost column cell],
    [SETJ bottom left corner cell, SETJ bottom row cell,
     SETJ bottom right corner cell]], dtype=object)
```

Additionally, MyArray has many useful attributes and methods for SETJ array manipulation and analysis.

```
In [17]: MyArray.
MyArray.__class__     MyArray.gamma       MyArray.k3p        MyArray.q_current
MyArray.__doc__       MyArray.k           MyArray.k3q        MyArray.randomIC
MyArray.__init__      MyArray.k1p         MyArray.k3u        MyArray.shape
MyArray.__module__    MyArray.k1q         MyArray.k3x        MyArray.t
MyArray.array         MyArray.k1u         MyArray.k4p        MyArray.tau_i
MyArray.b                     MyArray.k1x         MyArray.k4q        MyArray.u_current
MyArray.binary        MyArray.k2p         MyArray.k4u        MyArray.x_current
MyArray.dt            MyArray.k2q         MyArray.k4x        MyArray.y_current
MyArray.epsilon       MyArray.k2u         MyArray.new_param
MyArray.evolvearray   MyArray.k2x         MyArray.p_current
```

In fact, most of the SETJ operations have been transferred to *SETJarray*. However, if one chooses to play with SETJ cells without the *SETJarray* class, there are still built in methods to do so.

```
n [18]: Alfred = MyArray.array[2][1]

In [19]: Alfred
Out[19]: SETJ inner cell

In [20]: Alfred*(2,3)
Out[20]:
array([[SETJ inner cell, SETJ inner cell, SETJ inner cell],
    [SETJ inner cell, SETJ inner cell, SETJ inner cell]], dtype=object)
```

*CNN_lattice_simulater.py* works *with TPL_CNN.py* to create, manipulate and analyze SETJ arrays. This interactive program runs the user through options for creating their array and setting parameters, and eventually displays the evolution of the dynamical system using the output variable of the user's choice. The *SETJarray.evolvearray* RK4 method in *TPL_CNN.py* performs the integration of the impulsive differential equations over the spatially extended array.

*Results*

### 1. An Isolated SETJ

Before attacking the complexity of large SETJ arrays, I began my study of SETJ dynamics by analyzing the isolated SETJ circuit. However, even a driven isolated SETJ circuit can display interesting dynamical behavior, as it is characterized by a second order nonlinear differential equation with impulsive behavior. Remember that the equations describing the isolated SETJ are:

$$\frac{d\theta}{dt} = -\frac{\gamma}{\pi}\theta + \gamma(a - b\cos t)\,, \quad t \neq \tau_i\,,$$
$$\Delta\theta(\tau_i) = -2\pi\,, \quad t = \tau_i\,,$$
$$|\theta(0)| < \pi\,,$$

where

$$a = \frac{V_b}{V_T}\,, \quad b = \frac{V_p}{V_T}\,, \quad \theta = \frac{2\pi C v_C}{e}\,,$$
$$\tau = \omega_p t\,, \quad \gamma = \frac{\pi}{RC\omega_p}\,.$$

[1].

First, let us consider the behavior of the system while $-\pi < \theta < \pi$, where the voltage across the junction is less than the tunneling voltage, and no tunneling events occur. In this range of theta, the equations for the system can be rewritten as:

$$\dot{t} = 1$$

$$\dot{\theta} = -\frac{\gamma}{\pi}\theta + \gamma(a - b\cos t)$$

Obviously there are no fixed points for time, but we can find any fixed points for the junction voltage by setting $\overset{«}{\theta} = -\frac{\gamma}{\pi}\theta^* + \gamma(a - b\cos t) = 0$, from which we find $\theta^* = \pi(a - b\cos t)$ for $| a + b |< 1$. Therefore, if $| a + b |< 1$, then the system will reach a steady-state where it oscillates about $\pi a$. This should come as no surprise: it implies that if the applied bias and pump voltage amplitude do not combine to exceed the voltage required for tunneling, then the voltage across the capacitive junction will build up to the bias voltage, and oscillate about that point in unison with the pump. We can also find that in this limiting case, the eigenvalues, $\lambda = 0, -\frac{\gamma}{\pi}$ correspond to the non-isolated temporal solutions of the steady state and the expected decay rate for an RC circuit. Figure 2 shows the time series and phase evolution of such a system with $a = 0.6$, $b = 0.2$, and $\gamma = 1/3$.

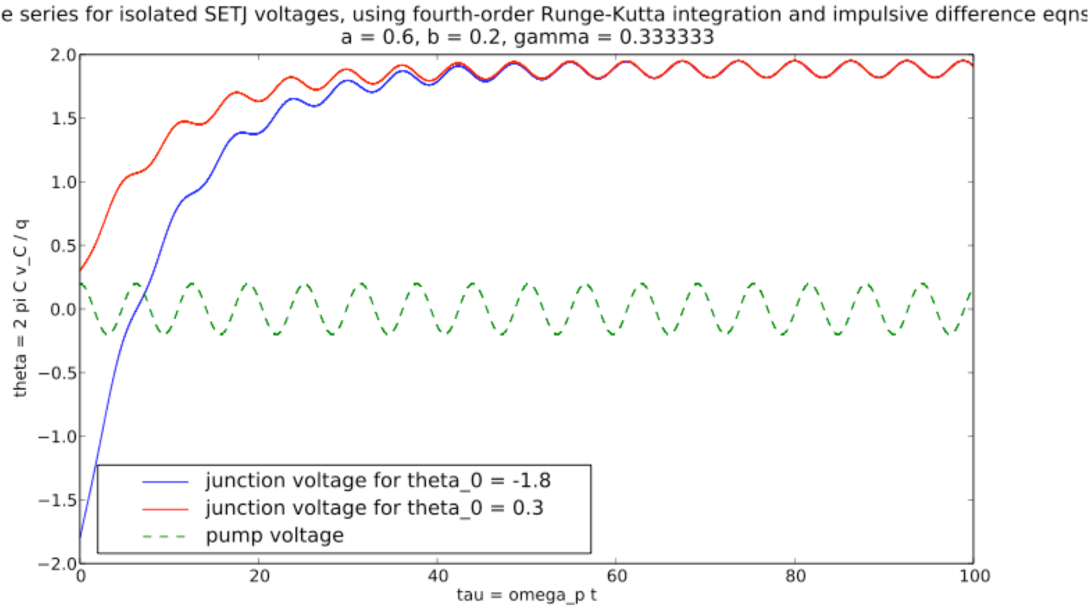


**Figure 2(a)**

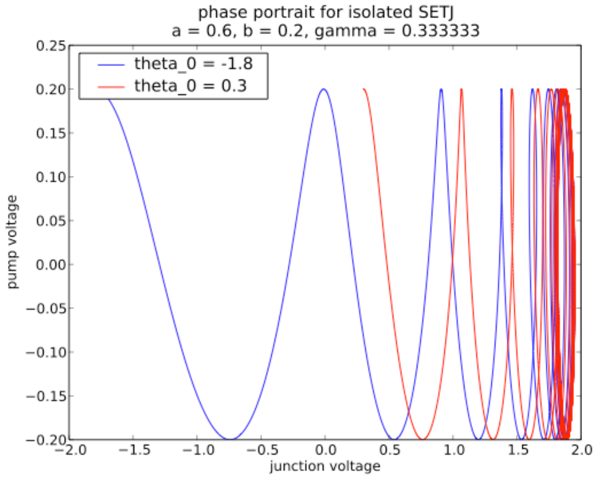


**Figure 2(b)**

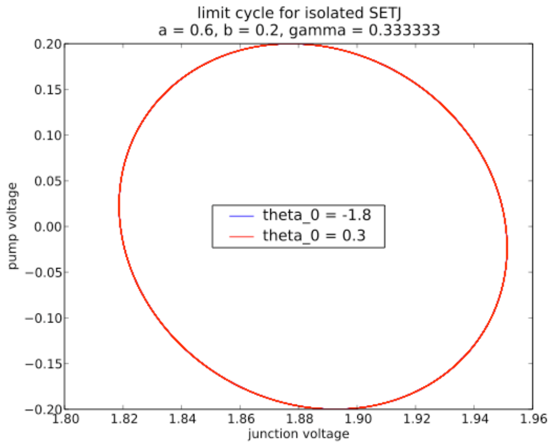


**Figure 2(c)**

Consider next, the behavior of this circuit when $a > 1$, but the pump voltage is set to zero so that b = 0. As shown in Figure 3, the system then behaves as a relaxation oscillator. When comparing the two panels in Figure 3, which are plots of the time series for two different initial voltages across the tunneling junction, it is evident that both trajectories have the same waveform. The waveforms are merely shifted along the time axis according to their initial condition, which is seen to be the y-intercept for each plot. We therefore deduce that there is no transient behavior for the isolated tunneling junction when b = 0.

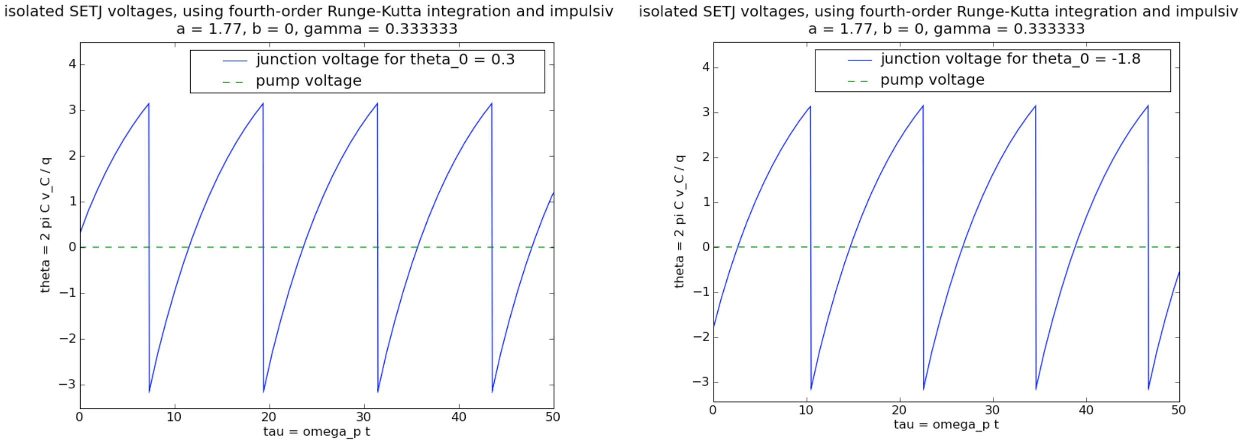


**Figure 3**

When the pump is turned on such that $a = 1.77$, $b = 2.0$, and $\gamma = 1/3$, as shown in Figure 4(a), any initial condition seems to snap to one of two possible waveforms after a short transient period. This entrainment to a super-harmonic of the pump is even more evident in panel (b) of Figure 4, where the steady-state trajectories for $\theta_0$ = 0.3 and $\theta_0$ = 0.8 are shown. The corresponding phase portrait shown in panel (c) of Figure 4 shows the two cycles converging to a single attractor. Once we get rid of transient behavior, we can see in panel (d) of Figure 4 that the two waveforms do indeed converge to the same limit cycle. However, what is not apparent from the figure is that these two waveforms are actually $2\pi$ out of phase with each other (using the pump phase as a common reference) so that the two trajectories are actually chasing each other around the limit cycle throughout their $4\pi$ period.

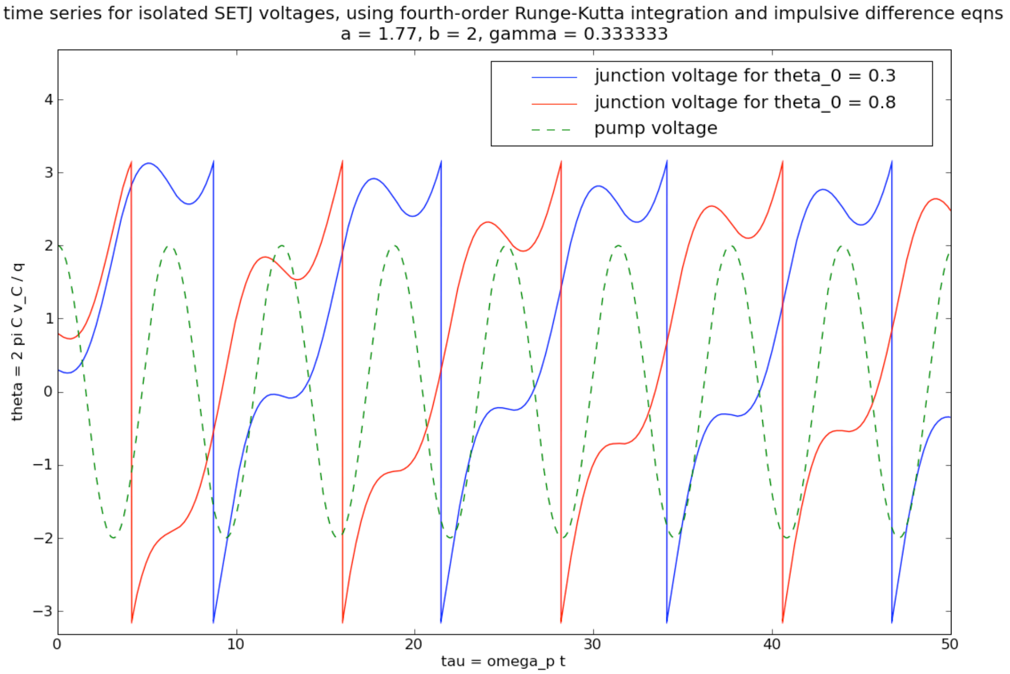


**Figure 4(a)**

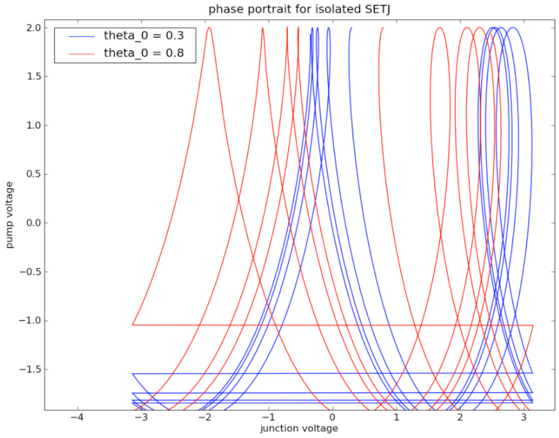


**Figure 4(b)**

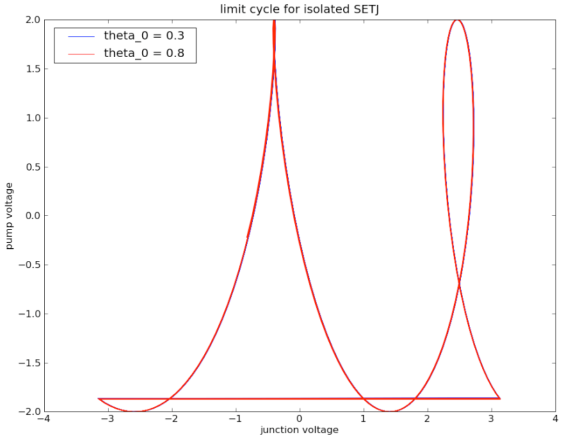


**Figure 4(c)**

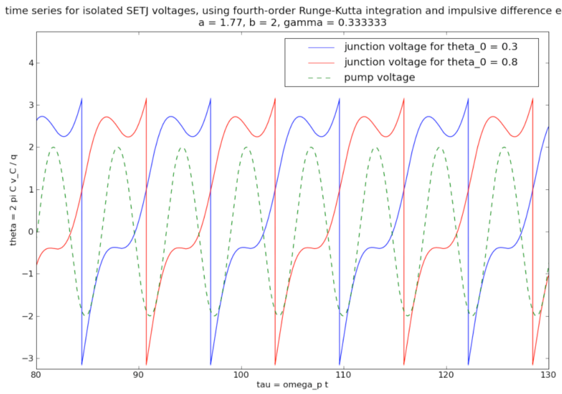
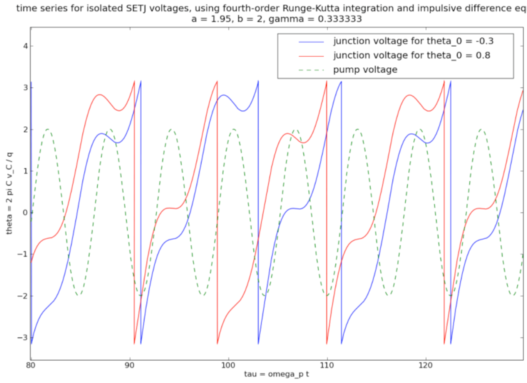


**Figure 4(d)**

Before we get too complacent about the idea of an isolated SETJ circuit producing bistable orbits, we should notice that this behavior is unique to the parameter space near $a = 1.77$. While keeping all other parameters the same, it seems that we can tune $a$ to produce a window of five stable states (at $a = 1.95$, as shown in Figure 5), or even orbits with arbitrarily long periods (see Figure 6 for an example of a parameter setting, $a = 1.49$, that has fourteen possible steady-state orbits, each with a period of $28\pi$).



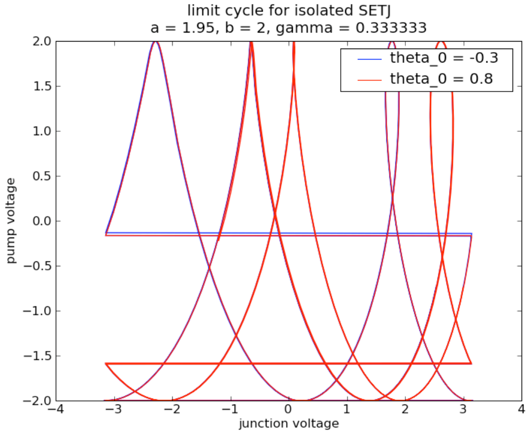


**Figure 5(a)**

**Figure 5(b)**

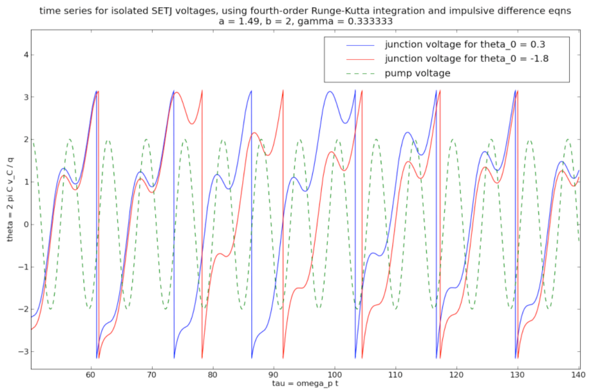


Figure 6(a)

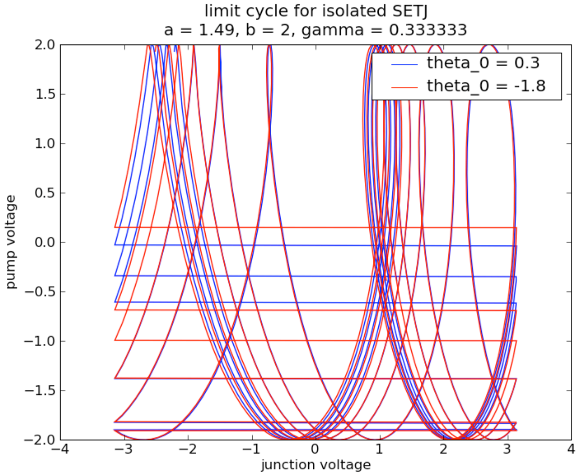


Figure 6(b)

To observe the underlying pattern of orbit periodicity, I have plotted the bifurcation diagram in Figure 7 as a function of the control parameter, $a$. For each value of $a$ from $-0.5 < a < 2.5$, a random initial junction voltage has been chosen, and a steady-state snapshot of $\theta$ has been taken thousands of times at a single phase of the pump cycle. Therefore, this bifurcation diagram suggests that a given value of $a$, which intercepts $N$ diagonal striae, corresponds to a waveform of period $\propto 2N\pi$ (the period should in fact be $2N\pi$ although the diagram is not quite sufficient evidence to prove this). Since each tightly packed $a$ value was plotted with a random $\theta_0$, we can also assume that the diagram has uncovered all of the attractors for this $a$, $b$, and $\gamma$.

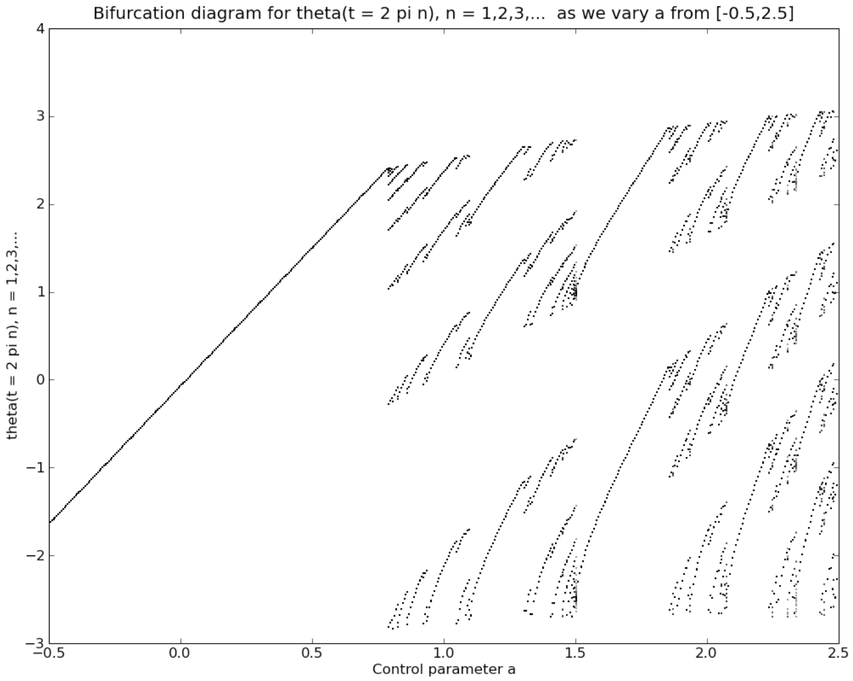


Figure 7

For any $a$, we can alternatively verify the number and stability of orbits by overlaying the Poincare return maps for periods of all integer multiples of the pump period, and observing its intersections with the diagonal. These maps are shown in Figure 9 for the previous $a$-values examined. Although the Poincare maps display local instabilities where the slope of $P(\theta)$ diverges, the steady-state orbits tend to be stable since the points of intersection usually all have local slope, or equivalently Floquet multipliers, less than one. The Poincare maps can all be seen to flatten out as periods progress, indicating that most $\theta_0$ lead to the same waveform. However, the plots of pump vs tunneling voltage (which would more naturally be mapped to a toroidal surface than the paper here) are perhaps the clearest way to see that all orbits converge to the same limit cycle with only an offset in phase by multiples of $2\pi$.

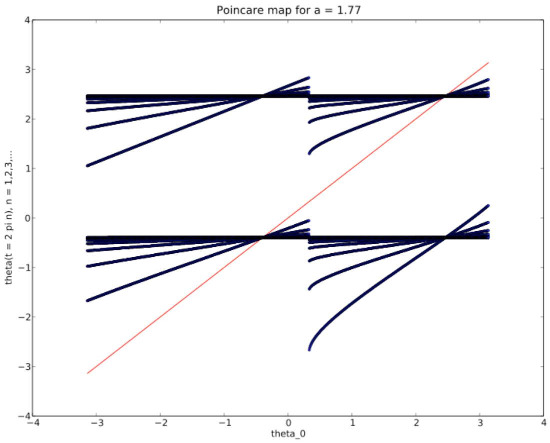


Figure 9(a)

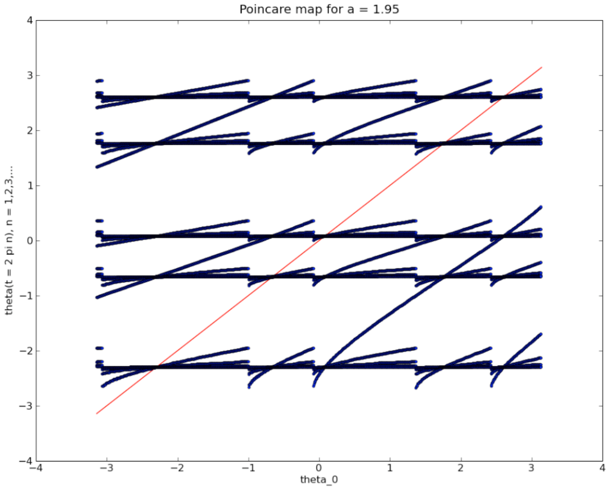


Figure 9(b)

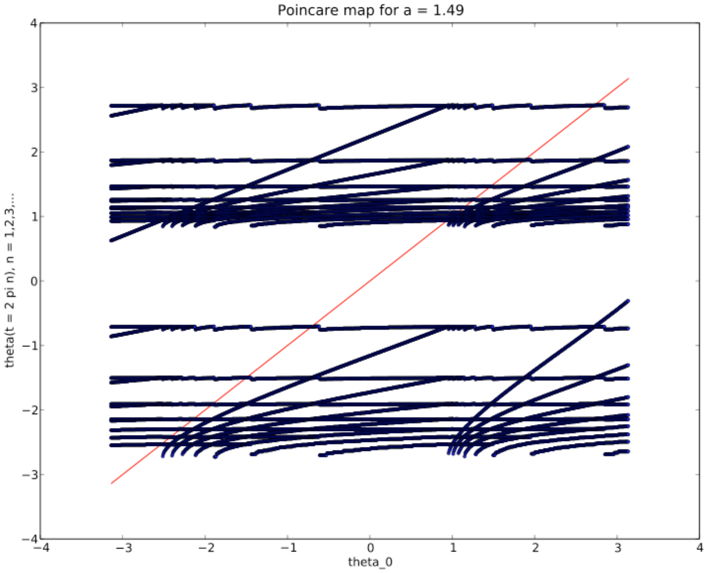


Figure 9(c)

## 2. Two-Coupled SETJs

Now that we know what to expect from the dynamics of an isolated SETJ, let us consider how the dynamics are enriched when an SETJ is coupled to another through a capacitor and a small resistance. Throughout my simulations, the most important requirement for strong interaction seemed to be a low $k = \frac{R_{coupling}}{R}$, as one might expect from the governing equations. As shown in Figure 10, a tunneling event causes the most drastic interaction between SETJs since a small discontinuity is induced in the non-tunneling junction. However, mutual interaction continually influences each SETJs behavior. Playing with a three-dimensional plot, which relates voltages in the pump and the first and second SETJ, helps one to make sense of the tangle of voltages. Panels (d) and (e) show how this single plot contains the information from both (b) and (c).

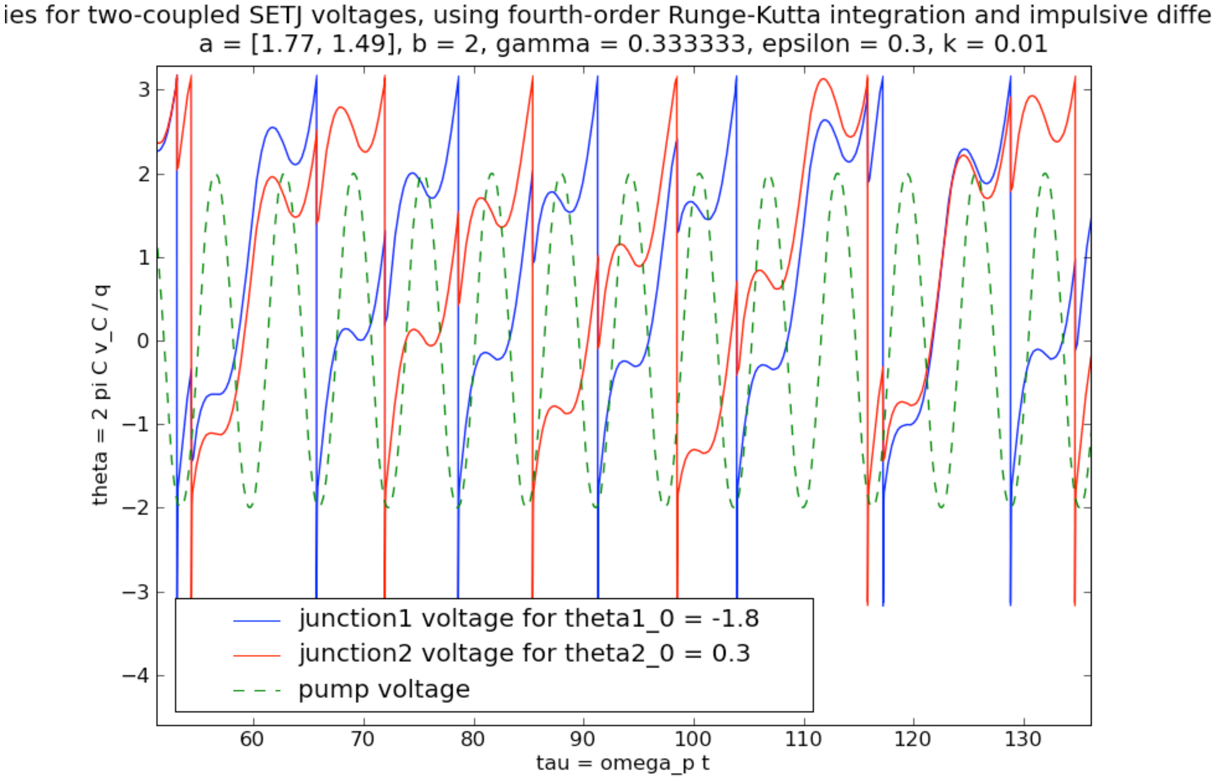


**Figure 10(a)**

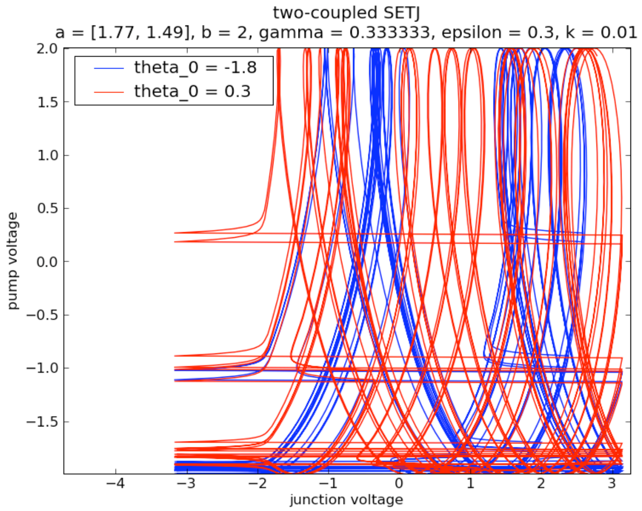


Figure 10(b)

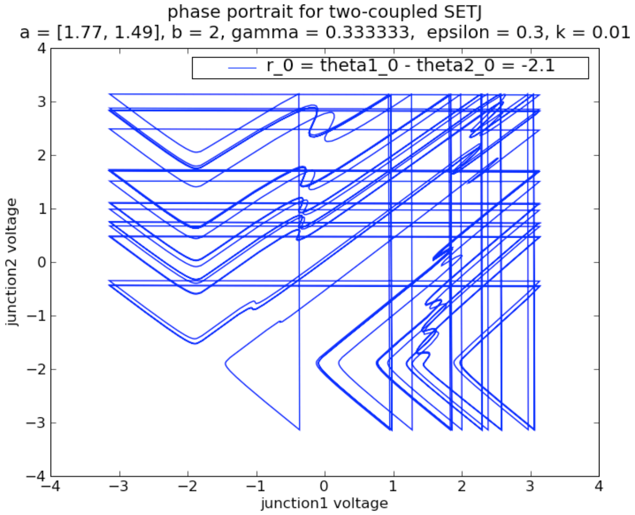


Figure 10(c)

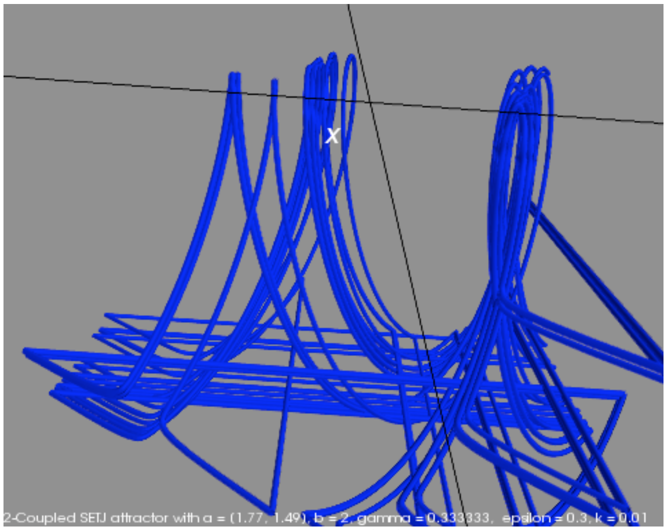

X
2-Coupled SETJ attractor with a = (1.77, 1.49), b = 2, gamma = 0.333333, epsilon = 0.3, k = 0.01

Figure 10(d)

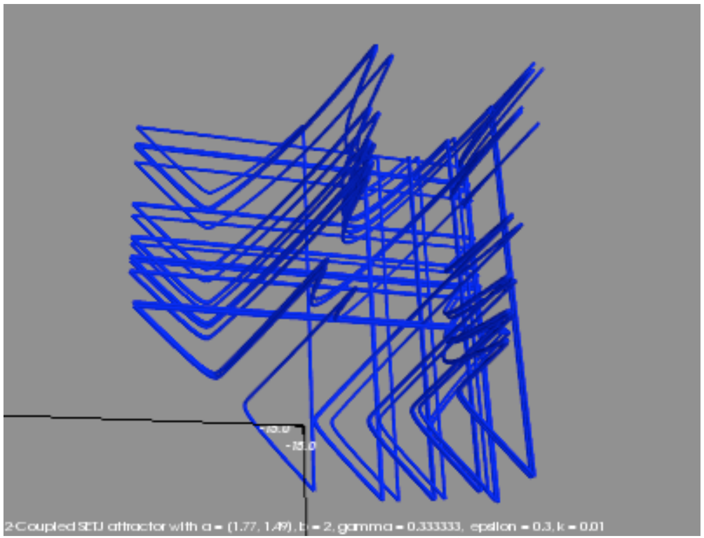

-15.0
-15.0
2-Coupled SETJ attractor with a = (1.77, 1.49), b = 2, gamma = 0.333333, epsilon = 0.3, k = 0.01

Figure 10(e)

As shown in Figures 11 and 12, two SETJs biased at different voltages can also find a compromise frequency to produce periodic behavior. This entrainment can be achieved by giving the SETJs bias voltages that are opposite in sign but equal in magnitude as shown in Figure 11 (a)-(c). However, unified behavior can also be achieved when bias voltages for each SETJ are similar, as shown in Figure 12 (a)-(c), where the bias voltages are $a_1 = 1.77$ and $a_2 = 1.6$.

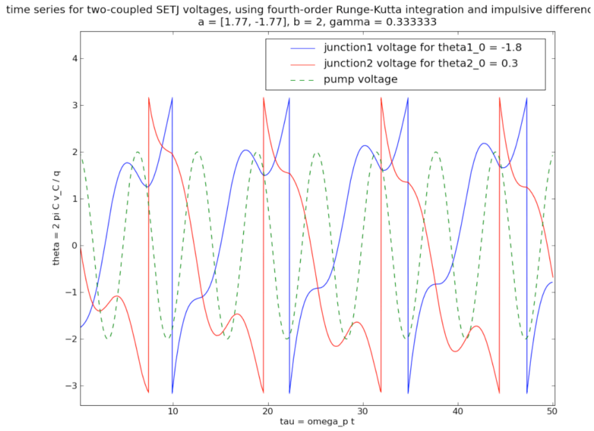


Figure 11(a)

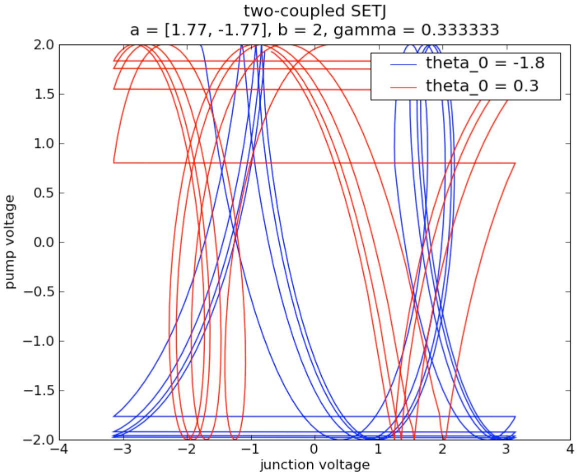


Figure 11(b)

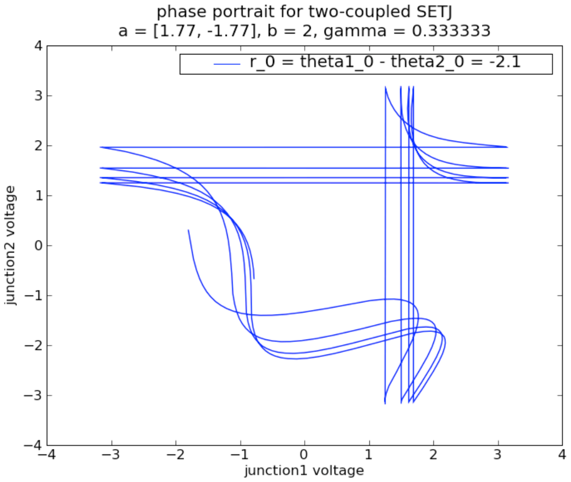


Figure 11(c)

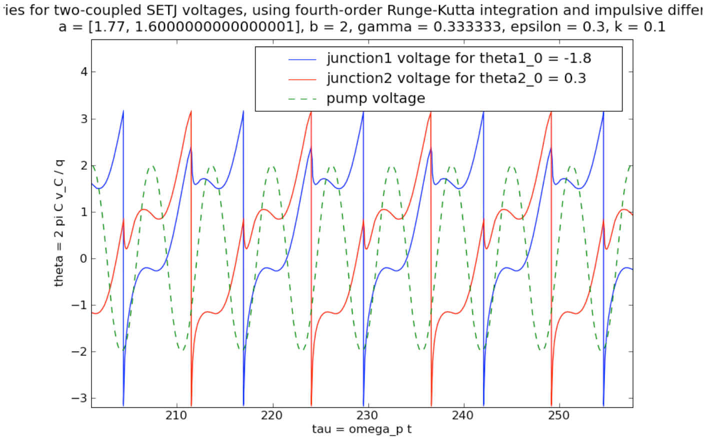


Figure 12(a)

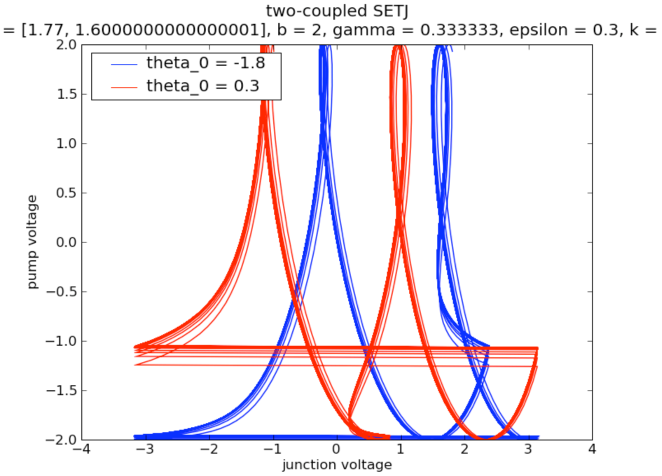


Figure 12(b)

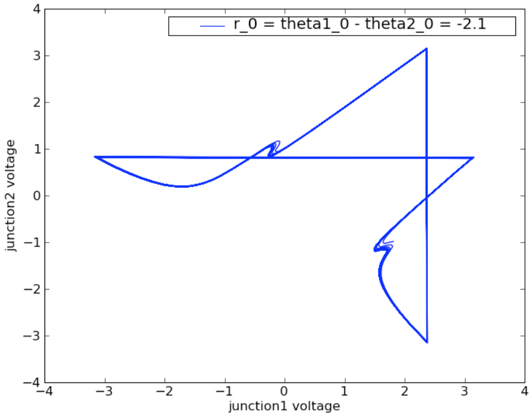


Figure 12(c)

## 3. SETJ Arrays

The behavior of large arrays of SETJs can be appreciated qualitatively through the colorful output simulations generated by *CNN_lattice_simulator.py*. This program can directly output voltages across tunneling junctions, or it can output tunneling phase logic states as described in [1]. Tunneling phase logic states indicate the pump phase at which the last tunneling event occurred, and can be a continuous or binary output function. The first two outputs are displayed as colorized grey-scale while the third necessarily only utilizes two colors.

The sequence of events in Figure 13 shows that the checkerboard bias voltage setting can be used to see how tunneling events can cascade through the array and even be reflected by the boundary cells. The output setting here is the voltage across the tunneling junction.

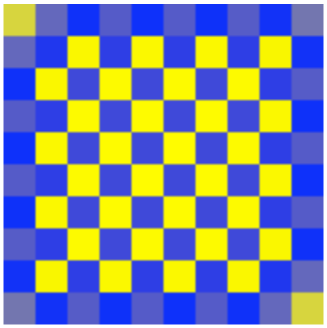


Figure 13(a)

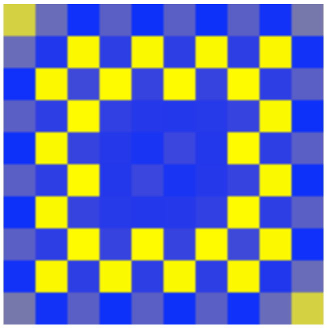


Figure 13(b)

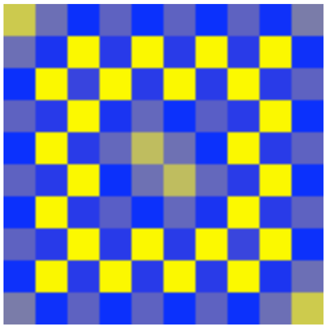


Figure 13(c)

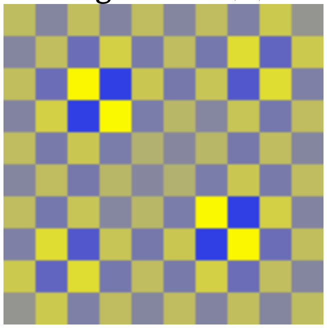


Figure 13(d)

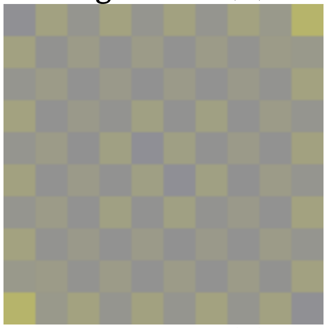


Figure 13(e)

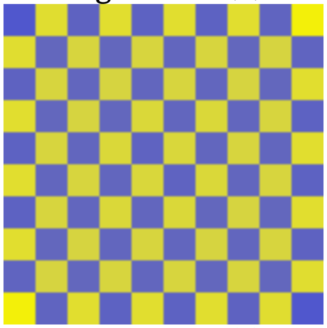


Figure 13(f)

Other settings can be used to show how SETJs might process information given as bias voltage inputs. As shown in Figure 14, my simulations confirm that SETJ cellular nonlinear networks (CNNs) could be used for edge detection, as first shown by Yang et al. [1].

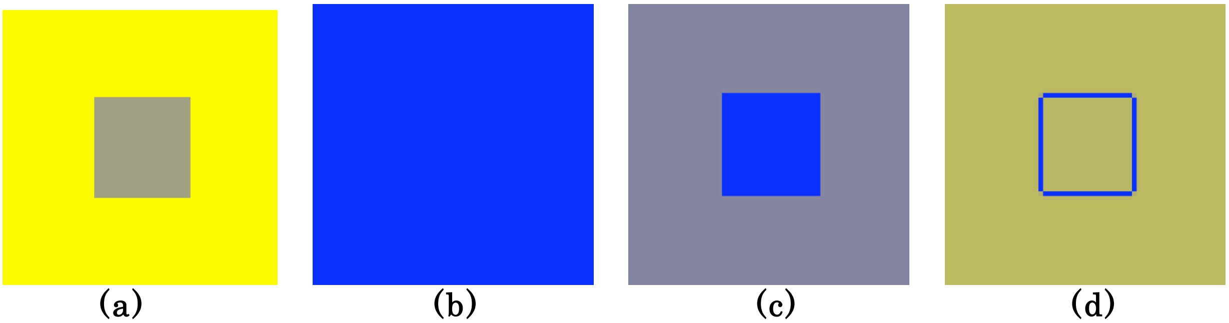


**Figure 14**

Panel (a) of Figure 14 shows the input voltage for the array. Panels (b),(c), and (d) show the continuous tunneling phase logic (TPL) output for the array at the normalized times, $\tau$ = 10, 25, and 100 respectively.

I have also created an option to encode images as grey-scale bias voltage distributions in SETJ arrays. I have found that one function of SETJ arrays is to reduce an image to two colors with an adjustable threshold grey-scale value. To obtain a polarized image, I have found that high k-values are desirable. Figure 15 shows that when the Poincare image is processed with low k values, the increased interaction between cells leads to more active or indecisive domain regions than if the image had been processed with a higher k-value. High b-values also seem to be desirable, as they tend to wash-away some of the straggling pixels in foreign domains. To evaluate the array's success at this task, these figures are compared with the same image, processed by *Gimp* using its *Threshold* tool.

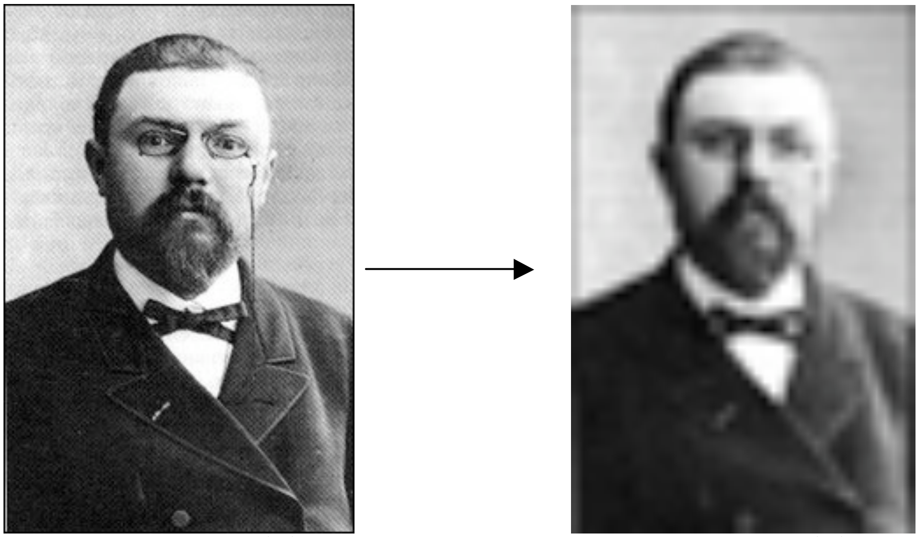


Figure 15(a)

Figure 15(b)

**Figure 15 (a), (b)**

Panel (a) shows the original image of Poincare. Panel (b) shows the lower resolution image used as bias voltage input for the simulations.

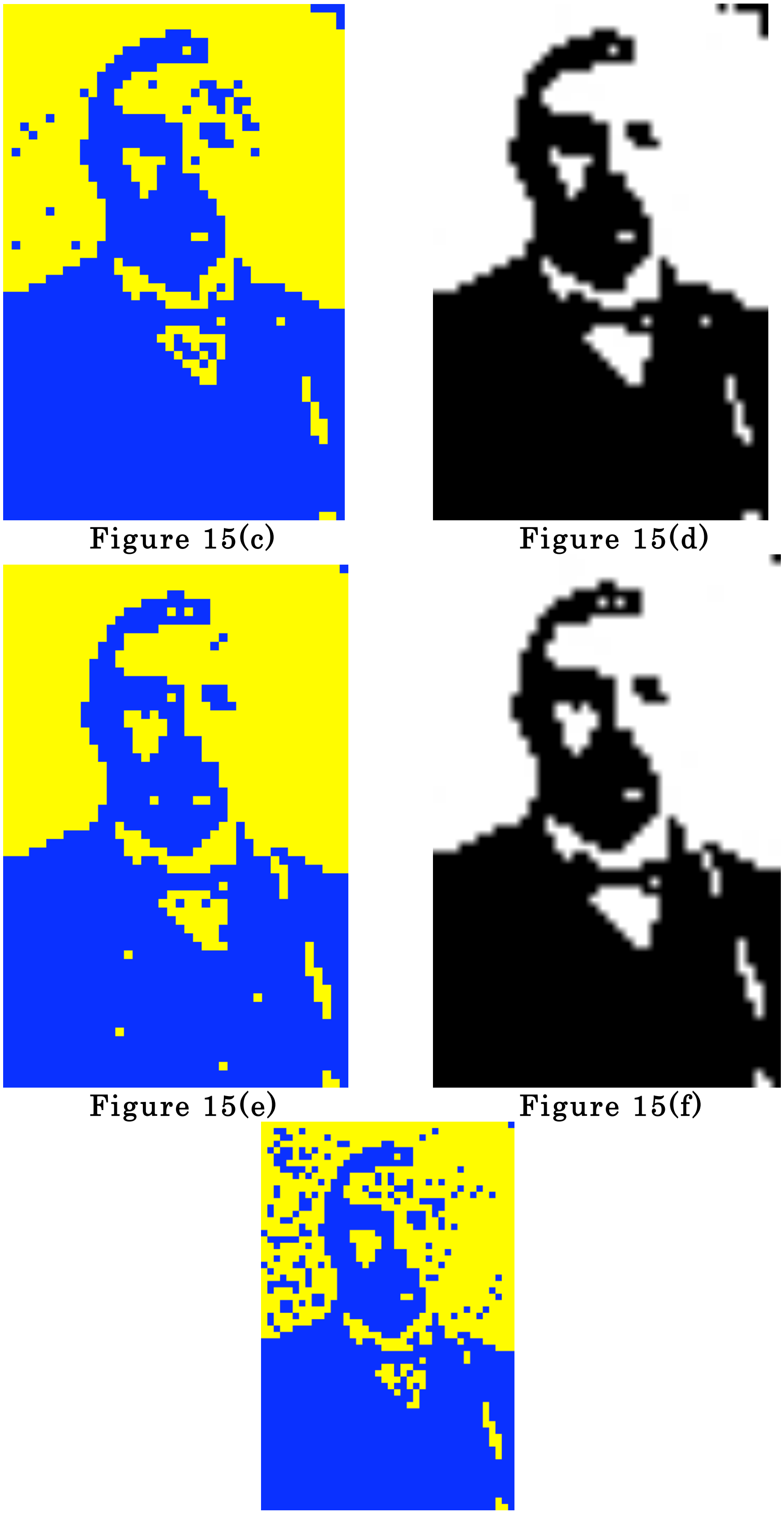


Figure 15(c)

Figure 15(d)

Figure 15(e)

Figure 15(f)

Figure 15(g)

**Figure 15 (c) – (g)**

Panel (c) shows the binary TPL output for the array at b = 2 and k = 10.

Panel (d) shows the Gimp output of the input image with a Threshold value of 136.

Panel (e) shows the binary TPL output for the array at b = 3 and k = 10.

Panel (f) shows the Gimp output of the input image with a Threshold value of 130.

Panel (g) shows the spotty binary TPL output for the array at b = 2 and k = 1.

SETJ arrays can also generate complex patterns that one might expect from cellular automata. Figure 16 shows some examples of patterns that have emerged from simple geometric bias voltage inputs.

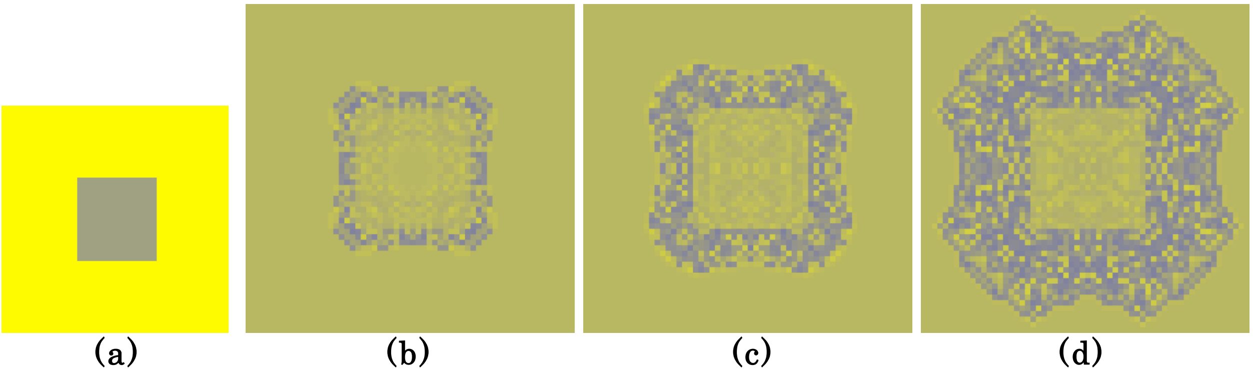


**Figure 16**

Panel (a) shows the input voltage for the array (same as Figure 14(a)). Panels (b) through (d) show the continuous TPL output evolution for $\tau$ = 1000, 2000, 3486 respectively.

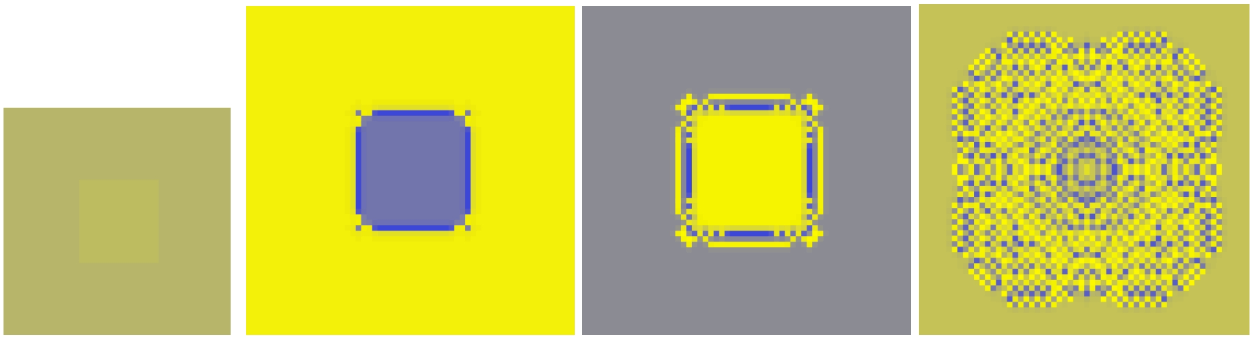


**Figure 17**

Panel (a) shows the input voltage for the array. Panels (b) through (d) show the continuous TPL output evolution for $\tau$ = 50, 100, and 257 respectively.

Figure 17 closes this section with some interesting behavior observed using the checkerboard bias voltage distribution and a continuous TPL output function.

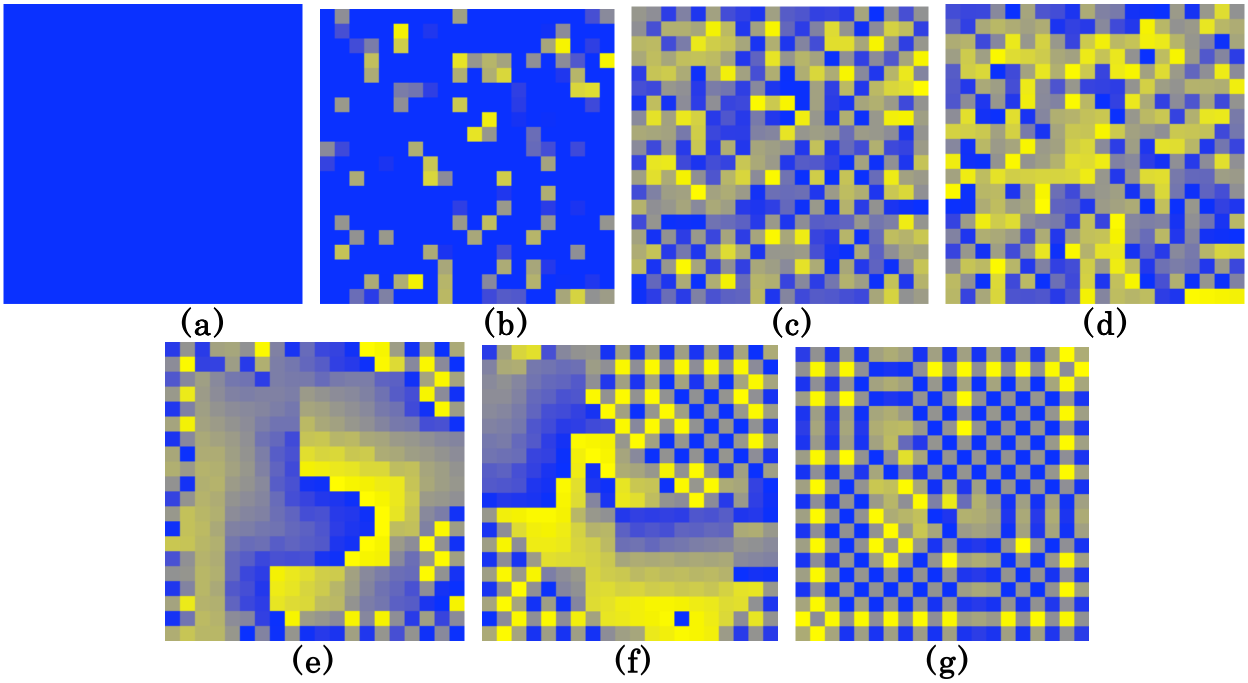


**Figure 17**

Panels (a) through (g) show the continuous TPL output function at the normalized times of $\tau$ = 1, 25, 50, 100, 1000, 2000, and 3120. This simulation was run with $b = 2$, $\gamma = 0.1$, $\varepsilon = 0.1$, and $k = 0.1$.

*Conclusion*

Single electron tunneling junction circuits provide an excellent setting to explore nonlinear dynamics. While an isolated SETJ circuit displays its own interesting dynamics, coupled SETJs are spatially extended nonlinear systems that can generate complex behavior and can potentially process information in a useful way. A deep understanding of the nonlinear interaction of these SETJ arrays could allow us to create new electronic devices that are not only smaller, but also smarter than current technology. Although the examples given here are a small subset of the possible behaviors exhibited by SETJ arrays, I hope to have conveyed some interesting possibilities.

The core programs I have written for this project should provide a good foundation for future exploration of SETJ arrays. *TPL_CNN.py* provides a general infrastructure for creating and manipulating arbitrarily large MxN SETJ arrays. Suggested improvements for this program include faster algorithms, a more realistic SETJ model that includes noise, temperature, and other non-ideal effects, and arbitrarily connected arrays that could exhibit small-world connectivity, higher dimensionality, or modularized functionality.

*References*

[1] Tao Yang, Richard A. Kiehl, and Leon O. Chua, “Tunneling Phase Logic Cellular Nonlinear Networks,” *International Journal of Bifurcation and Chaos*, Vol. 11, No. 12(2001)2895–2911

[2] Nikolai A. Nezlobin and Richard A. Kiehl, “Phase-locking in a nonuniform array of tunnel junctions,” *Journal of Applied Physics*, Volume 94, Number 10 15 November 2003

[3] Andrew Kilinga Kikombo, Tetsuya Hirose, Tetsuya Asai, Yoshihito Amemiya, “Non-linear phenomena in electronic systems consisting of coupled single-electron oscillators,” *Chaos, Solitons and Fractals* 37 (2008) 100–107

[4] T. Yang and L.O.Chua, "Nonlinear dynamics of driven single-electron tunneling junctions," Int. J*. Bifurcation and Chaos* 10, 1091 May 2000

[5] R. A. Kiehl, T. Yang, and L.O.Chua, "Tunneling-Phase Logic Based Cellular Nonlinear Networks," ECCTD’01 - European Conference on Circuit Theory and Design, August 28-31, 2001, Espoo, Finland