Memristor Based Chaotic Circuits

Bharathwaj Muthuswamy
Milwaukee School of Engineering
S-342, Fred Loocke Engineering Center
Milwaukee, WI 53202
Pracheta Kokate
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA, 94720, USA
muthuswamy@msoe.edu[1]
prachetakokate@berkeley.edu

***Abstract*—Ever since its physical fabrication in 2008, the memristor has shown promise in the fields of nano-electronics, computer logic and neuromorphic computers. Taking advantage of the circuit properties of the memristor, this paper proposes memristor based chaotic circuits. For the first time, memristor based chaotic circuits have been derived from the canonical Chua's circuit. These circuits present opportunities for developing applications under the constraints of scalability and low power. They also provide a memristor based framework for secure communications with chaos.**

***Index Terms*—nonlinear dynamics, chaos, memristor, chaotic circuit, Chua's circuit, Lyapunov exponents**

I. Introduction

The memristor was postulated as the fourth circuit element by Leon O. Chua in 1971 [1]. It thus took its place along side the rest of the more familiar circuit elements such as the resistor, capacitor, and inductor. The common thread that binds these four elements together as the four basic elements of circuit theory is the fact that the characteristics of these elements relate the four variables in electrical engineering (voltage, current, flux and charge) intimately. Fig. 1 shows this relationship graphically [2]. The memristor is a two terminal element,

[1] Author for correspondance. Also affiliated with the Nonlinear Electronics Laboratory, University of California, Berkeley

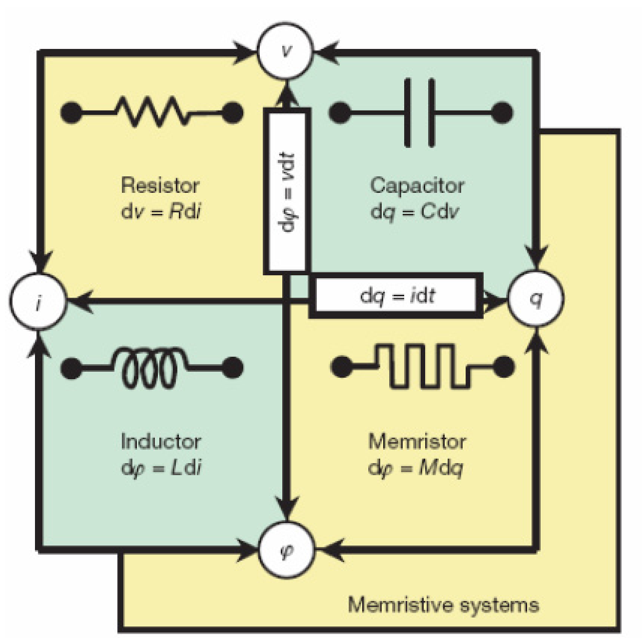


Fig. 1. The Four Basic Circuit Elements [2]

in which the magnetic flux (ϕ) between the terminals is a function of the electric charge that passes through the device [2]. The memristor M used in this paper is a flux controlled memristor that is characterized by its incremental menductance [2] function $W(\phi)$ describing the flux-dependent rate of change of charge:

$$W(\phi) = \frac{dq(\phi)}{d\phi} \tag{1}$$

The relationship between the voltage across ($v(t)$) and the current through ($i(t)$) the memristor is thus given by:

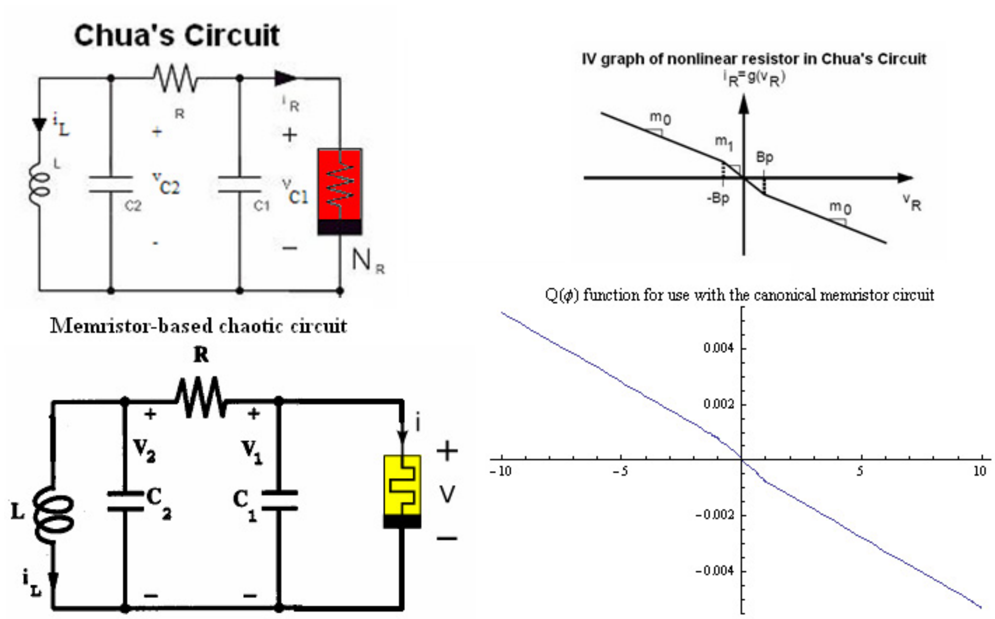
$$i(t) = W(\phi(t))v(t) \tag{2}$$

Memristor is an acronym for *memory-resistor* because in Eq. 2, since $W(\phi(t)) = W(\int v(t))$, the integral operator on the menductance function means the function *remembers* the past history of voltage values. Of course, if $W(\phi(t)) = W(\int v(t)) = constant$, a memristor is simply a resistor. For over thirty years, the memristor was not significant in circuit theory. In 2008, Williams et. al. [2] fabricated a solid state implementation of the memristor and thereby cemented its place as the 4th circuit element. They used two titanium dioxide films, with varying resistance which is dependent on how much charge has been passed through it in a particular direction. As a result of this realization, it is possible to have nonvolatile memory on a nano scale.

Until recently, electrical and electronics theory have been focusing on linear elements. However, the works of Chua et. al. have brought the study of nonlinear electronics to the forefront, proving that study of nonlinearity can hold very promising applications especially in the area of secure communications [3]. High frequency chaotic oscillators have immense potential for applications in secure communication [4]. One possibility for obtaining high frequency chaotic circuits is to use nano-scale devices [5], like the memristor.

One of the first memristor based chaotic circuits have been proposed by Itoh and Chua [6]. However, their paper uses a passive nonlinearity based on the memristor characteristics obtained by Williams et. al.. Hence, their circuits are not suitable for secure communications because active nonlinearities are essential for high signal-to-noise ratio (SNR) [7]. The canonical Chua's circuit uses an active nonlinearity for obtaining chaos, and hence it has found a variety of applications ever since inception in 1984 in secure communications [8]. Therefore, a natural question to ask is: can we obtain a memristor based chaotic circuit from the classical Chua's circuit? This paper answers yes to the question posed above. We obtain a memristor based chaotic circuit by simply increasing the dimensionality of the canonical Chua's circuit. We define simply increasing as adding a memristor with the same nonlinearity as the canonical Chua's circuit. This circuit is the subject of section II. In section III, we simplify the canonical memristor based chaotic circuit by removing one of the elements *a'la* Barboza and Chua [9]. We derive the menductance nonlinearity by using the constraint of hyperbolic equilibrium points (condition 1 of Shilnikov's Theorem [10]). Then in Section IV, we notice that unlike Barboza and Chua's three dimensional chaotic circuit, we can eliminate one of the negative elements in the four dimensional memristor based chaotic circuit and still obtain chaos. In Section V, Lyapunov exponents are computed for all three circuits, this provides empirical evidence of chaos. We use two *different* methods to compute the Lyapunov exponents for consistency checks. Section VI illustrates how we can easily extend the canonical memristor based chaotic circuit to higher dimensions and explains why this is important to secure communications. The paper concludes with a summary

TABLE 1
IN ORDER TO OBTAIN THE MEMRISTOR-BASED CHAOTIC CIRCUIT, WE REPLACED THE NONLINEAR RESISTOR IN CHUA'S CIRCUIT WITH A MEMRISTOR.



of the key points and discusses future work.

II. Canonical Memristor based chaotic circuit

As stated in the introduction section, our first circuit dimensionally extends the canonical Chua's circuit. Refering to Table 1, we derived the memristor-based chaotic circuit by simply replacing the nonlinear resistor in Chua's circuit with a flux-controlled memristor. Below are the equations governing our memristor-based chaotic circuit:

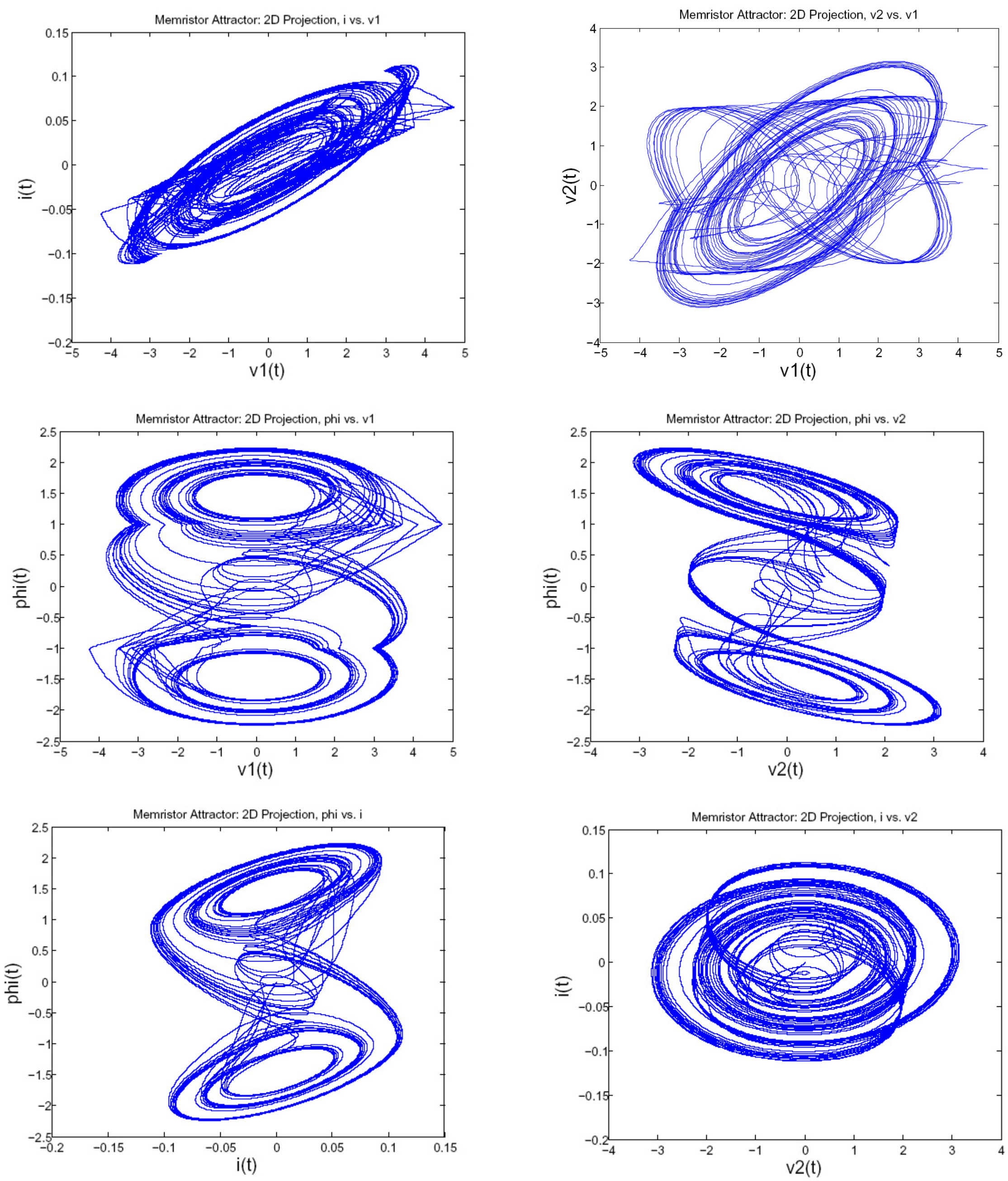
$$\begin{aligned}
\frac{d\phi}{dt} &= v_1(t) \\
\frac{dv_1(t)}{dt} &= \frac{1}{C_1}\left(\frac{v_2(t)-v_1(t)}{R} - W(\phi(t))\cdot v_1(t)\right) \\
\frac{dv_2(t)}{dt} &= \frac{1}{C_2}\left(\frac{v_1(t)-v_2(t)}{R} - i_L(t)\right) \\
\frac{di_L(t)}{dt} &= \frac{v_2(t)}{L}
\end{aligned} \tag{3}$$

The intuitive justification for dimensional extension is that an active nonlinearity is very important for obtaining a chaotic circuit. The dimensional extension not only preserves the active nonlinearity, it also introduces another nonlinearity in terms of the product $(W(\phi(t))v_1(t))$ in the equation above. These two nonlinearities should combine to give rise to chaos, as we observed. The $Q(\phi)$ function is obtained from the canonical Chua's circuit:

$$\begin{aligned}
Q(\phi) &= -0.5\cdot 10^{-3}\cdot\phi + \frac{-0.8\cdot 10^{-3}+0.5\cdot 10^{-3}}{2} \\
&\quad \cdot (|(\phi+1)| - |(\phi-1)|)
\end{aligned} \tag{4}$$

The menductance function that is obtained from the

TABLE 2
ATTRACTORS FROM THE STATE-SCALED CANONICAL MEMRISTOR-BASED CIRCUIT



$Q(\phi)$ function is:

$$W(\phi) = \frac{dq_m(\phi)}{d\phi} = \begin{cases} -0.5 \cdot 10^{-3} & \phi \leq -1 \\ -0.8 \cdot 10^{-3} & -1 < \phi < 1 \\ -0.5 \cdot 10^{-3} & \phi \geq 1 \end{cases} \quad (5)$$

Choosing the parameters as $C_1 = 5.5nF$, $C_2 = 49.5nF$, $L = 7.07mH$, $R = 1428\Omega$ and setting the initial conditions to $\phi(0) = 0$, $v_1(0) = 0$, $v_2(0) = 0$, $i(0) = 0.1$ we can see that this dimensionally extended circuit indeed generates chaotic behaviour. Appendix A has the simulation code. The simulation results in values for the states that are far beyond what is physically realizable. The system states can be rescaled to the appropriate (constrainted) values for $v_1(t)$ and $v_2(t)$. Refer to Appendix B for the rescaled system equations. Table 2 shows the attractors obtained from the rescaled canonical memristor-based chaotic circuit.

In the next section, we try to answer the question of whether we can reduce the number of elements in this circuit and still obtain chaos. Fewer elements in a circuit mean a more compact form for implementation on an integrated circuit. Since one of the greatest advantages of a memristor is a reduction in space (since the device functions at the nanoscale level), it is desirable to exploit this property.

III. Four Element Memristor based chaotic circuit

We simplified the circuit from the previous section *a'la* Barboza and Chua's four-element chaotic circuit. This circuit is significant since it is the *simplest* possible circuit in terms of the number of elements and it also displays bifurcation phenomenon not seen in the canonical Chua's circuit. Fig. 2 and Fig. 3 shows the four Element Chua's circuit and its realization respectively. Fig. 4 shows our version of the four-element Chua's circuit in which the nonlinear resistor is replaced by the memristor. Fig. 5 shows this circuit reduced down to the basic circuit elements. In order to choose the menductance for the memristor M in Fig. 5 we used the hyperbolic equilibrium point constraint from Shilnikov's theorem [10]. The resulting function is shown in Fig. 6. The system equations for Fig. 5 are:

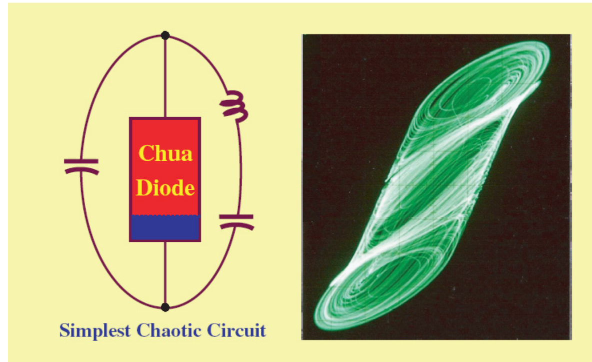


Fig. 2. The simplest Chua's circuit and its typical attractor [9]

$$\begin{aligned} \frac{d\phi}{dt} &= v_1 \\ i - W(\phi)v_1 &= C_1\frac{dv_1}{dt} \\ \kappa L_1\frac{di}{dt} &= v_1 - v_2 \\ i &= \frac{C_2}{\kappa}\frac{dv_2}{dt} \end{aligned} \qquad (6)$$

The parameters for (6) are: $C_1 = 33\ nF$, $C_2 = 100\ nF$, $L_1 = 10\ mH$ and $\kappa = 8.33$. In (6), we are free to pick $W(\phi)$. Using MATLAB, we get the results shown in Fig. 7 and Fig. 8. Appendix C has the MATLAB code used to obtain these results. One surprising result we obtained from this circuit is the fact that it can be

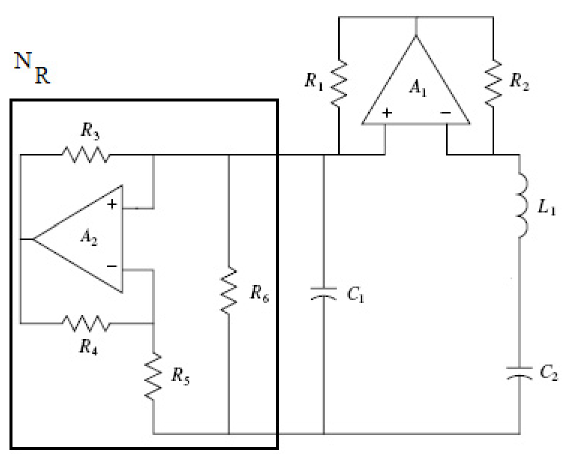


Fig. 3. A realization of the four-element Chua's circuit [9]

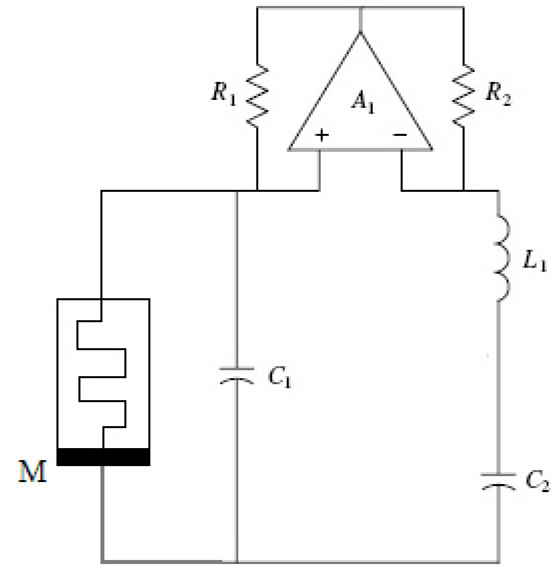


Fig. 4. Four-element memristor-based chaotic circuit

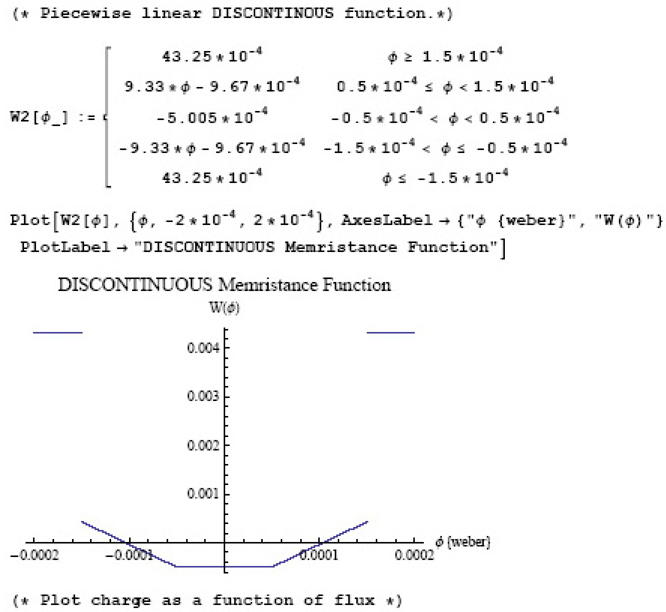


Fig. 6. Memristance function $W(\phi)$ as defined in Mathematica.

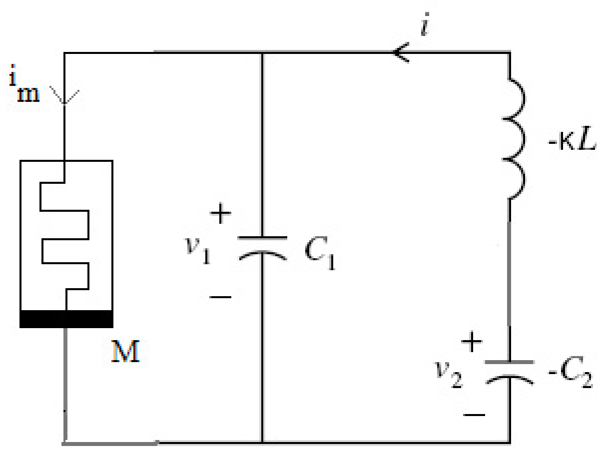


Fig. 5. Four-element memristor-based chaotic circuit showing only the basic circuit elements. The effect of op-amp A_1 from Fig. 4 is the set $-\kappa$.

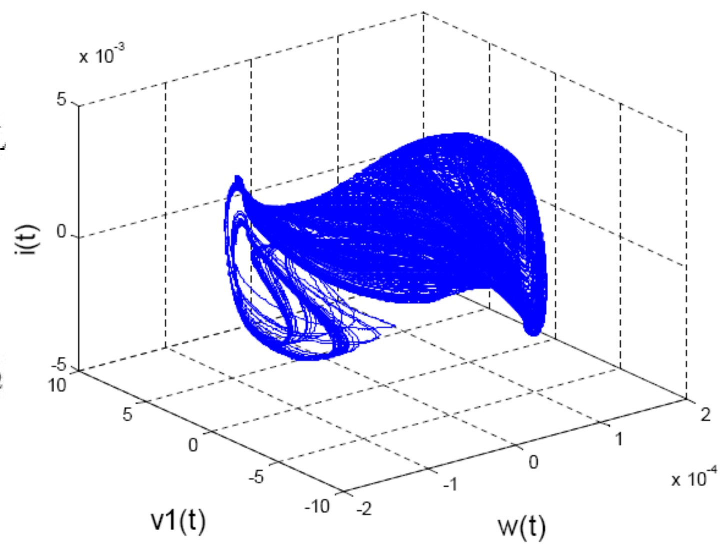


Fig. 7. 3D attractor from the four-element memristor-based chaotic circuit.

simplified even further. The next section describes how this may be possible and suggests how the simplification may be advantageous.

IV. Four Element Memristor based chaotic circuit with one negative element

Unlike Barboza and Chua's four element chaotic circuit, we discovered that our circuit requires only one negative element. Hence, we can simplify the circuit from the previous section even further. Specifically, in the circuit shown in Fig. 5, if we let C_2 be the only negative element

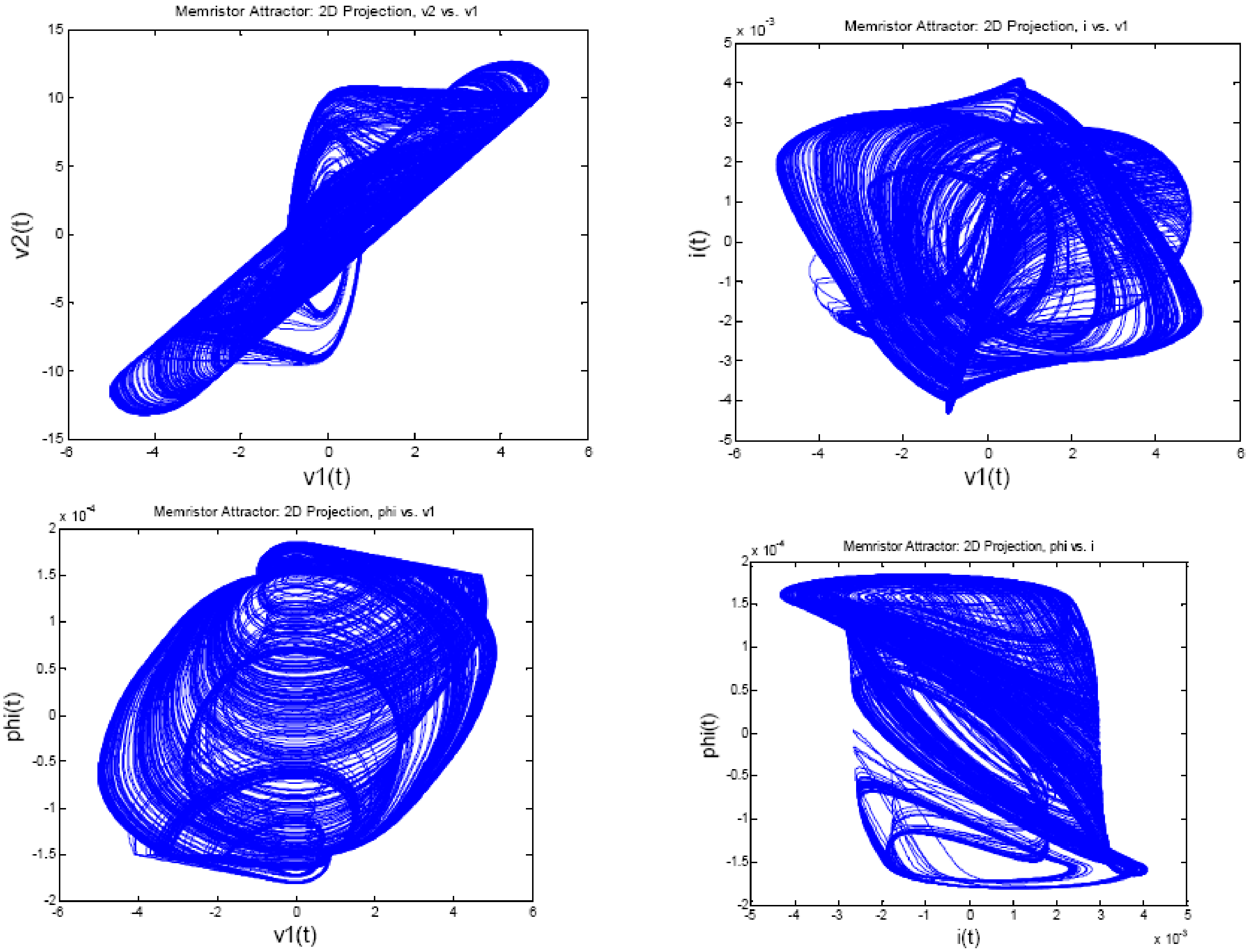


Fig. 8. 2D Projections of the attractor from the four-element memristor-based chaotic circuit.

and set κ to 1, we still get chaotic behaviour. System equations are:

$$\begin{aligned} \frac{d\phi}{dt} &= v_1 \\ i - W(\phi)v_1 &= C_1\frac{dv_1}{dt} \\ \kappa L_1\frac{di}{dt} &= v_1 - v_2 \\ i &= \frac{-C_2}{\kappa}\frac{dv_2}{dt} \end{aligned} \tag{7}$$

The other parameters, initial conditions and the menductance function ($W(\phi)$) are the same as the previous section. The simulation results are shown in Fig. 9 and Fig. 10, MATLAB simulation code is given in Appendix D. Having a single negative element is advantageous because we have reduced the number of active elements in the circuit. This leads to a reduction in power consumption. Although the above circuit and the two circuit(s) from the previous section(s) seem to exhibit chaotic attractors via simulation, a strong empirical indicator of chaos are Lyapunov exponents [10]. Lyapunov exponents characterize the rate of separation of infinitesimally close trajectories in state-space [10]. The rate of separation can be different for different orientations of the initial

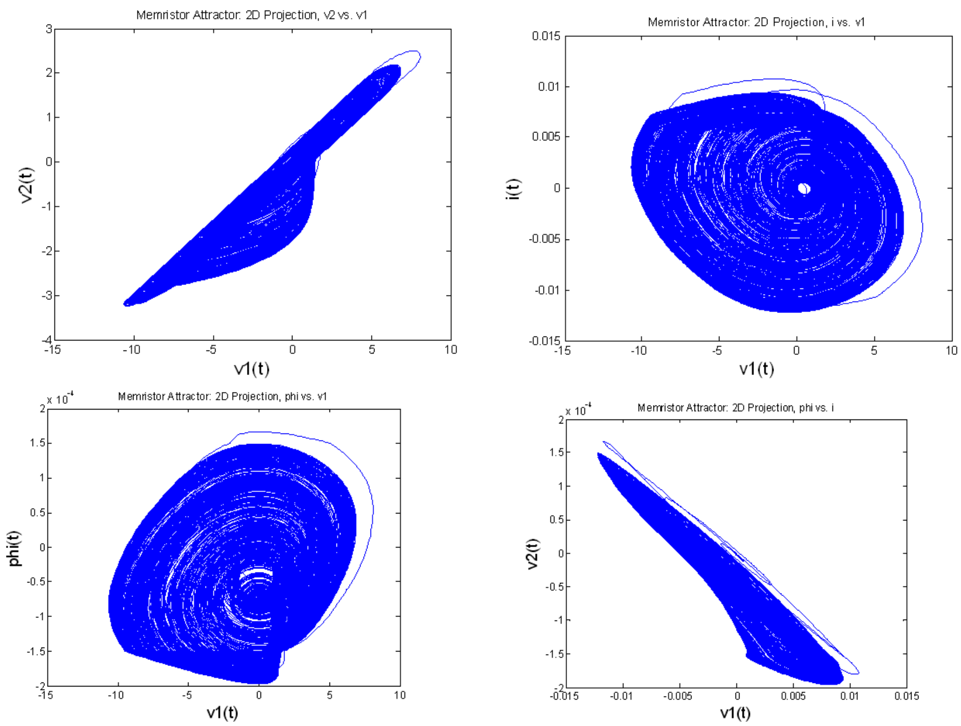


Fig. 10. 2D Projections of the attractor from the four-element memristor-based chaotic circuit with only one negative element.

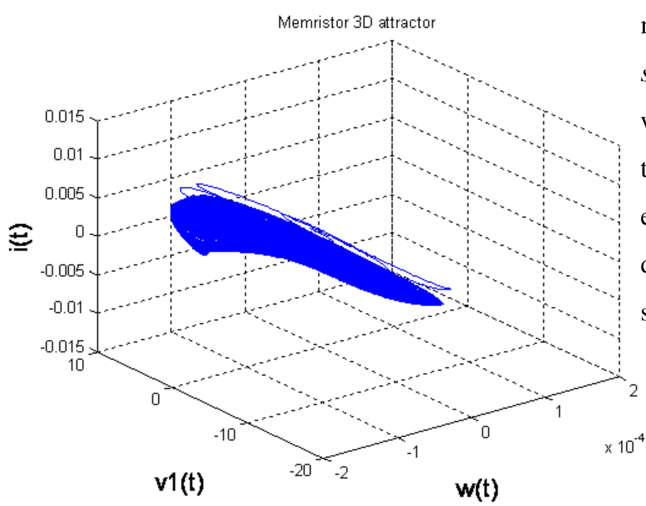


Fig. 9. 3D attractor from the four-element memristor-based chaotic circuit with only one negative element.

separation vector, hence the number of Lyapunov exponents is equal to the number of dimensions in phase space. If we have a positive Lyapunov exponent, that means we have an expanding direction. However, if the *sum* of the Lyapunov exponents is negative, that means we have contracting volumes in phase space. These two seemingly contradicting properties of the Lyapunov exponents are indications of chaotic behaviour in the dynamical system. Lyapunov exponents for our three systems are computed in the next section.

V. Lyapunov exponent calculations

Before computing the Lyapunov exponents, the time scales for the circuits are scaled to the order of seconds by: $\tau = \frac{t}{\sqrt{|L_1 C_2|}}$. This is necessary because the Lyapunov exponent algorithms numerically converge for these time scales. At lower time scales, we need really small step sizes for the ode solvers. This causes huge round-off errors.

As mentioned earlier, we use two independent meth-

TABLE 3
SUMMARY OF LYAPUNOV EXPONENTS

Circuit	QR Method	Time Series Method
canonical memristor-based chaotic circuit	0.085, 0, -0.0003, -0.668	0.086,0,0.0005,-0.672
four-element memristor-based chaotic circuit	0.11, 0, 0, -1.16	0.1, 0, 0, -1.2
four-element memristor-based chaotic circuit with only negative C_2	0.1, 0, 0, -0.7	0.1, 0, 0, -0.7

ods to estimate the Lyapunov exponents: the QR method from [11] and the time-series method from [12]. The LET toolbox from [13] and the Lyapunov time series toolbox from [14] have been used to estimate the exponents. Table 3 summarizes our results. Appendix E has the MATLAB code used for obtaining the values in Table 3. Let us analyze each row in the Table 3.

1) For the canonical memristor based chaotic circuit , we notice that we have four Lyapunov exponents. This alludes to the possibility of hyperchaos. However, in the circuit presented in this paper, hyperchaos seems to be absent. Although the time series method indicates a second positive Lyapunov exponent of 0.0005, this is probably numerical error and this Lyapunov exponent may tend towards zero as $t \to \infty$.
2) The four-element Chua's circuit with a memristor has one positive Lyapunov exponent, indicating the presence of chaos [14].
3) The four-element circuit with only a negative capacitance also gives to rise to chaotic behaviour because of the positive Lyapunov exponent [14]. However, Barboza and Chua's four-element circuit without the memristor does not seem to have chaotic behavior for the same set of C_1, L_1, C_2 and κ values. No matter what N_R function we choose for the Barboza circuit, we do not seem to get hyperbolic saddle equilibria if we have only one negative element. This difference between the Barboza-Chua circuit and the memristor circuit warrants further study.

Also note that the sum of the Lyapunov exponents is negative for all the circuits. This implies that volumes contract in phase space, however the positive Lyapunov exponent indicates an expanding trajectory. This implies that trajectories eventually converge to a fractal structure, namely, the chaotic attractor. The penultimate section in this paper suggests a possible application of the memristor based chaotic circuits to secure communication. We present a five-dimensional memristor based chaotic circuit that is obtained from the canonical four-dimensional memristor based chaotic circuit by the addition of an inductor and suggest why these circuits may hold promise for secure communication.

VI. Dimensionally extending canonical Memristor based chaotic circuit and application to secure communications

Fig. 11 shows that if we add inductor L_1 (highlighted in red) to the canonical memristor based chaotic circuit and set its value to 180 mH we still obtain chaos (all the other parameters, nonlinearity and initial conditions remain the same). The equations describing the five dimensional circuit are:

$$\begin{aligned} \frac{d\phi}{dt} &= v_1(t) \\ \frac{dv_1(t)}{dt} &= \frac{1}{C_1}\left(i_{L1} - W(\phi)\cdot v_1\right) \\ \frac{dv_2(t)}{dt} &= \frac{1}{C_2}\left(-i_{L1} - i_{L2}\right) \end{aligned} \quad (8)$$

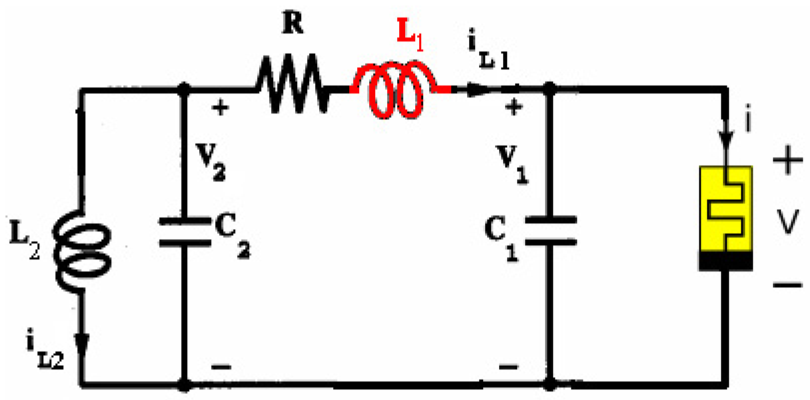


Fig. 11. Note that the addition of the inductor L_1 results in a five dimensional circuit. We can obtain chaos for an inductor value of 180 mH.

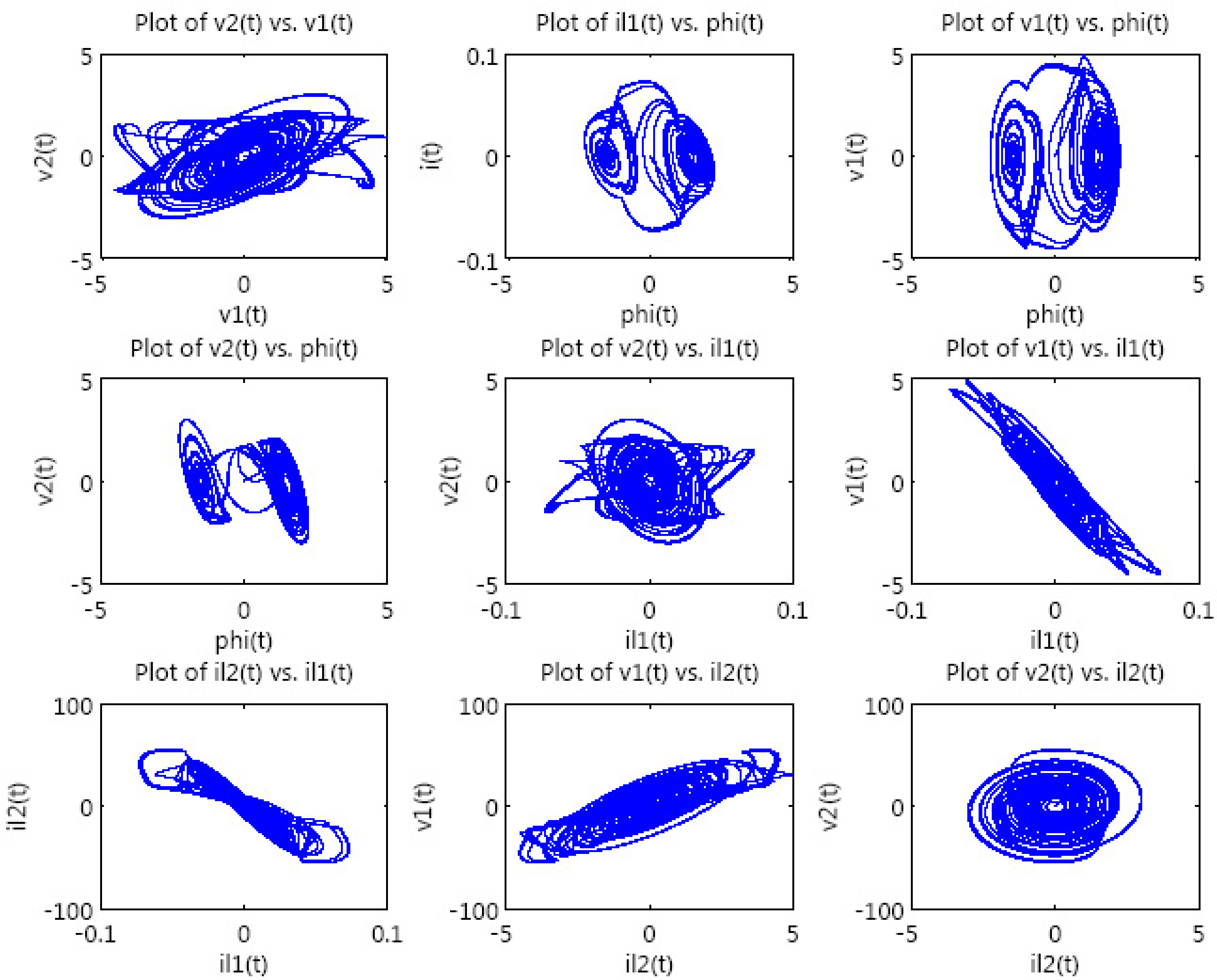


Fig. 12. Attractors obtained from the five dimensional circuit. Note that state scaling has already been incorporated.

$$\begin{aligned}\frac{di_{L1}(t)}{dt} &= \frac{1}{L_1}(v_2 - v_1 - i_{L1}\cdot R)\\ \frac{di_{L2}(t)}{dt} &= \frac{v_2}{L_2}\end{aligned}$$

Fig. 12 shows the attractors obtained from the rescaled canonical memristor-based chaotic circuit. Note that MATLAB simulation code for the circuit above has not been given since the MATLAB code can be easily extrapolated from the other appendices. The Lyapunov exponents that were computed for the circuit above are 0,0.22,0,-2.99,-3.05 (from LET toolbox) and 0,0.22,0,-3.38,-1.85 (from the time-series method). There is close agreement between the two methods. We also have one positive Lyapunov exponent and the sum of the exponents is negative. Hence, there is empirical evidence of chaos.

The point to note here is the ease of extending the four-dimensional system to a higher dimension. We simply added another element to the circuit and just had to tune its value. Moreover, as pointed out in literature [15], higher dimensional systems are suitable for secure communication because their attractors do not have an easily identifiable structure unlike lower dimensional chaotic systems. This directs the attention of secure communication to higher-dimensional systems [16], [17]. Hence, an interesting question that warrants further study is whether even higher dimensional (6th order, 7th order,...) circuits can be obtained from this five dimensional circuit.

VII. Conclusions

This paper has discussed three possible memristor-based chaotic circuits and presented a possible fourth candidate. Specifically, we have discovered:

1) There exists a memristor based chaotic circuit that can be obtained from the canonical Chua's circuit by simple "dimension extension"'.
2) There exists a simplification of the memristor based chaotic circuit from (1) above that has only four elements (we eliminated the resistor *a'la* Barboza and Chua).
3) We can simplify the four element memristor based chaotic circuit to use only one negative element.
4) We can extend the four dimensional memristor based chaotic circuit to five dimensions. This increase in dimensionality has implications for designing better secure communication systems.

These circuits warrant a plethora of future research, a few selected topics are:

1) Determining the route to chaos (period doubling etc.) in these circuits.
2) Proving the circuits are chaotic rigorously by way of topological horseshoe theory.
3) A framework for implementing memristor-based chaotic circuits.
4) Exploring the possibility of hyperchaos [18], [19], [20] in these circuits.
5) Mathematically explaining the dimensional extension exhibited by these systems.
6) Studying applications to secure communications.

Acknowledgment

The authors would like to thank Prof. Pravin P. Varaiya, Prof. Leon O. Chua and Ferenc Kovac for their support and guidance.

Appendix A
MATLAB simulation code for Canonical Memristor Based Chaotic Circuit

The file below is called Canonical.m. The corresponding W.m is shown after Canonical.m

```
%% Memristor based chaotic Chua's circuit simulation
%% Bharathwaj Muthuswamy,
%% Pracheta Kokate
%% June 13th 2008 - July 13th 2008,
```

```
%% June 2009
%% mbharat@cory.eecs.berkeley.edu
%% Ref: Stephen Lynch,
%% Dynamical Systems with Applications
%% using MATLAB
clear;
%% MAKE SURE YOU PUT CODE BELOW ON A
%% SINGLE LINE!
dmemristor=inline('[y(2);1/5.5e-9*
((y(3)-y(2))/1428
- W(y)*y(2));1/49.5e-9*((y(2)-y(3))/1428
- y(4)); y(3)/7.07e-3]','t','y');

options = odeset('RelTol',1e-7,'AbsTol',
1e-7);
[t,ya]=ode45(dmemristor,[0 10e-3],
[0,0,0,0.1],options);


plot(ya(:,2),ya(:,4));
title('Memristor Attractor: 2D Projection,
 i vs. v1')
fsize=15;
xlabel('v1(t)','Fontsize',fsize);
ylabel('i(t)','Fontsize',fsize);
figure
plot(ya(:,2),ya(:,3))
title('Memristor Attractor: 2D Projection,
 v2 vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('v2(t)','Fontsize',fsize);
figure
plot(ya(:,2),ya(:,1))
title('Memristor Attractor: 2D Projection,
phi vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);
figure
plot(ya(:,3),ya(:,1))
title('Memristor Attractor: 2D Projection,
phi vs. v2')
xlabel('v2(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);

figure
plot(ya(:,4),ya(:,1))
title('Memristor Attractor: 2D Projection,
phi vs. i')
xlabel('i(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);

figure
plot(ya(:,3),ya(:,4))
title('Memristor Attractor: 2D Projection,
i vs. v2')
xlabel('v2(t)','Fontsize',fsize);
ylabel('i(t)','Fontsize',fsize);

figure
plot(t,ya(:,2))
hold
plot(t,ya(:,4),'r')
title('Memristor chaotic time series:
v1 (blue) and v2 (red)')
figure
plot(t,ya(:,1))
hold
plot(t,ya(:,3),'r')
title('Memristor chaotic time series:
w (blue) and i (red)')
%% 3d plot: flux, current and voltage
figure
plot3(ya(:,1),ya(:,2),ya(:,3));
grid on
xlabel('w(t)','Fontsize',fsize);
ylabel('v1(t)','Fontsize',fsize);
zlabel('i(t)','Fontsize',fsize);
title('Memristor 3D attractor');
%% Menductance function W.m
function r = W(y)
    if(y(1) <= -1)
        r = -0.5e-3;
    elseif((y(1) > -1) && (y(1) < 1))
        r = -0.8e-3;
    else
        r = -0.5e-3;
end
```

Appendix B
MATLAB simulation code for Rescaled Canonical Memristor Based Chaotic Circuit

Use the same W.m as the previous appendix.

```
%% Memristor based chaotic Chua's circuit
%% simulation
%% Bharathwaj Muthuswamy, Pracheta Kokate
%% June 13th 2008 - July 13th 2008,
%% June 2009
%% mbharat@cory.eecs.berkeley.edu
%% Ref: Stephen Lynch,
%% Dynamical Systems with Applications
%% using MATLAB
clear;
%% MAKE SURE YOU PUT CODE BELOW ON A SINGLE LINE!
dmemristor=inline('[y(2); 1/5.5e-9*((y(3)-y(2))/1428-
W(y)*y(2)); 1/49.5e-9*((y(2)-y(3))/1428 -
y(4)); y(3)/7.07e-3]','t','y');

 options = odeset('RelTol',1e-7,'AbsTol',1e-7);
[t,ya]=ode45(dmemristor,[0 10e-3],[0,0,0,0.1],options);
ya(:,1) = ya(:,1);
ya(:,2) = ya(:,2)./10000;
ya(:,3) = ya(:,3)./5000;
ya(:,4) = ya(:,4)./500;

plot(ya(:,2),ya(:,4));
title('Memristor Attractor: 2D Projection, i vs. v1')
fsize=15;
xlabel('v1(t)','Fontsize',fsize);
ylabel('i(t)','Fontsize',fsize);
figure
plot(ya(:,2),ya(:,3))
title('Memristor Attractor: 2D Projection, v2 vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('v2(t)','Fontsize',fsize);
figure
plot(ya(:,2),ya(:,1))
```

```
title('Memristor Attractor: 2D Projection,
 phi vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);
figure
plot(ya(:,3),ya(:,1))
title('Memristor Attractor: 2D Projection,
phi vs. v2')
xlabel('v2(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);

figure
plot(ya(:,4),ya(:,1))
title('Memristor Attractor: 2D Projection,
 phi vs. i')
xlabel('i(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);

figure
plot(ya(:,3),ya(:,4))
title('Memristor Attractor: 2D Projection,
 i vs. v2')
xlabel('v2(t)','Fontsize',fsize);
ylabel('i(t)','Fontsize',fsize);

figure
plot(t,ya(:,2))
hold
plot(t,ya(:,4),'r')
title('Memristor chaotic time series:
v1 (blue) and v2 (red)')
figure
plot(t,ya(:,1))
hold
plot(t,ya(:,3),'r')
title('Memristor chaotic time series:
 w (blue) and i (red)')
%% 3d plot: flux, current and voltage
figure
plot3(ya(:,1),ya(:,2),ya(:,3));
grid on
xlabel('w(t)','Fontsize',fsize);
ylabel('v1(t)','Fontsize',fsize);
zlabel('i(t)','Fontsize',fsize);
title('Memristor 3D attractor');
```

APPENDIX C
MATLAB SIMULATION CODE FOR FOUR ELEMENT MEMRISTOR BASED CHAOTIC CIRCUIT

There are two files: the ode solver (memristorAudio.m) and the memristance function (W.m). Shown below is memristorAudio.m:

```
%% Memristor based chaotic Chua's circuit
%% simulation
%% Bharathwaj Muthusway,
%% Pracheta Kokate
%% June 13th 2008 - July 13th 2008,
%% June 2009
%% mbharat@cory.eecs.berkeley.edu
%% Ref:  Stephen Lynch,
%% Dynamical Systems with Applications
%% using MATLAB
clear;
%% MAKE SURE YOU PUT CODE BELOW ON A
 %% SINGLE LINE!
dmemristor=inline('[y(2);(y(3)-
W(y)*y(2))/33e-9;
(y(2)-y(4))/(8.33*10e-3);
y(3)*(8.33/100e-9)]','t','y');
options = odeset('RelTol',1e-7,
'AbsTol',1e-7);
[t,ya]=ode45(dmemristor,[0 100e-3],
[0,0.1,0,0],options);
plot(ya(:,2),ya(:,4))
title('Memristor Attractor: 2D Projection,
v2 vs. v1')
fsize=15;
xlabel('v1(t)','Fontsize',fsize);
ylabel('v2(t)','Fontsize',fsize);
figure
plot(ya(:,2),ya(:,3))
title('Memristor Attractor: 2D Projection,
 i vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('i(t)','Fontsize',fsize);
figure
plot(ya(:,2),ya(:,1))
title('Memristor Attractor: 2D Projection,
 phi vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);
figure
plot(ya(:,3),ya(:,1))
title('Memristor Attractor: 2D Projection,
 phi vs. i')
xlabel('i(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);
figure
plot(t,ya(:,2))
hold
plot(t,ya(:,4),'r')
title('Memristor chaotic time series:
 v1 (blue) and v2 (red)')
figure
plot(t,ya(:,1))
hold
plot(t,ya(:,3),'r')
title('Memristor chaotic time series:
 w (blue) and i (red)')

%% 3d plot: flux, current and voltage
figure
plot3(ya(:,1),ya(:,2),ya(:,3));
grid on
xlabel('w(t)','Fontsize',fsize);
ylabel('v1(t)','Fontsize',fsize);
zlabel('i(t)','Fontsize',fsize);
title('Memristor 3D attractor');
```

Shown below is W.m:

```
%% Menductance functions for use with
%% memristorAudio.m and
%% memristorAudioSIMPLEST.m
%% Bharathwaj Muthusway,
%% Pracheta Kokate
%% July 17th 2008,
```

```
%% June 2009
%% mbharat@cory.eecs.berkeley.edu

function r=W(y)
    if y(1) <= -1.5e-4
        r = 43.25e-4;
    elseif y(1) > -1.5e-4 && y(1) <= -0.5e-4
        r = -9.33*y(1)-9.67e-4;
    elseif y(1) > -0.5e-4 && y(1) < 0.5e-4
        r = -5.005e-4;
    elseif y(1) >= 0.5e-4 && y(1) < 1.5e-4
        r = 9.33*y(1)-9.67e-4;
    else
        r = 43.25e-4;
    end
end
```

The circuit parameters above were chosen such that the circuit frequencies are in the audio range. To listen to sounds of chaos, use the MATLAB command:

```
soundsc(ya(:,1),44000)
```

APPENDIX D
MATLAB SIMULATION CODE FOR FOUR ELEMENT MEMRISTOR BASED CHAOTIC CIRCUIT WITH SINGLE NEGATIVE ELEMENT

The file below is called memristorAudioSIMPLEST.m, use the same W.m in the previous appendix.

```
%% Memristor based chaotic Chua's circuit
%% simulation
%% Bharathwaj Muthusway, Pracheta Kokate
%% June 13th 2008 - July 3rd 2008
%% June 2009
%% mbharat@cory.eecs.berkeley.edu
%% Ref:  Stephen Lynch,
%% Dynamical Systems with Applications
%% using MATLAB

clear;
%% MAKE SURE CODE BELOW IS ON A SINGLE LINE.
%% W(y) is the same function from the
%% previous appendix
dmemristor=inline('[y(2);(y(3)-W(y)*y(2))/33e-9;
(y(2)-y(4))/(1*-10e-3);
y(3)*(1/100e-9)]','t','y');

options = odeset('RelTol',1e-7,'AbsTol',
1e-7);
[t,ya]=ode45(dmemristor,[0 100e-3],
[0,0.1,0,0],options);

plot(ya(:,2),ya(:,4))
title('Memristor Attractor: 2D Projection,
v2 vs. v1')
fsize=15;
xlabel('v1(t)','Fontsize',fsize);
ylabel('v2(t)','Fontsize',fsize);

figure
plot(ya(:,2),ya(:,3))
title('Memristor Attractor: 2D Projection,
 i vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('i(t)','Fontsize',fsize);

figure
plot(ya(:,2),ya(:,1))
title('Memristor Attractor: 2D Projection,
phi vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);

figure
plot(ya(:,3),ya(:,1))
title('Memristor Attractor: 2D Projection,
 phi vs. i')
xlabel('i(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);

fsize=15;
xlabel('v1(t)','Fontsize',fsize);
ylabel('v2(t)','Fontsize',fsize);

figure
plot(t,ya(:,2))
hold
plot(t,ya(:,4),'r')
title('Memristor chaotic time series:
v1 (blue) and v2 (red)')

figure
plot(t,ya(:,1))
hold
plot(t,ya(:,3),'r')
title('Memristor chaotic time series:
 w (blue) and i (red)')

%% 3d plot: flux, current and voltage
figure
plot3(ya(:,1),ya(:,2),ya(:,3));
grid on
xlabel('w(t)','Fontsize',fsize);
ylabel('v1(t)','Fontsize',fsize);
zlabel('i(t)','Fontsize',fsize);
title('Memristor 3D attractor');
```

APPENDIX E
LYAPUNOV EXPONENT PROGRAMS

The Lyapunov exponent program for the four-element memristor-based chaotic circuit is shown below.

```
function OUT = fourElementMemristor(t,X)
%MEMRISTOR Model of memristor based four
%Element chaotic circuit

% Settings:
% ODEFUNCTION: fourElementMemristor
% Final Time:  1000, Step: 0.01,
% Relative & Absolute Tol: 1e-007
% No. of discarded transients: 100,
% update Lyapunov: 10
% Initial Conditions: 0 0.1 0 0,
% no. of linearized ODEs: 16

% The first 4 elements of the input data X
```

```
% correspond to the
% 4 state variables.  Restore them.
% The input data X is a 12-element vector
% in this case.
% Note: x is different from X
w = X(1); x = X(2);y = X(3);z = X(4);

%% MAKE SURE CODE IS ON A SINGLE LINE!
% Parameters.
L1 = 10e-3;C1=33e-9;C2=100e-9;k=8.33;

% ODE
dw = (sqrt(L1*C2))*x;
dx = (sqrt(L1*C2)/C1)*(y-fourElementW(w)*x);
% dy = (sqrt(L1*C2)/(k*L1))*(x-z);
% comment dy above and uncomment dy
% below for ONE negative element
% this single negative element chaotic
% circuit may be the simplest
% possible four dimensional and
% four-element chaotic circuit
% ALSO, YOU NEED TO CHANGE JACOBIAN!
k = 1;
dy = (sqrt(L1*C2)/(k*-L1))*(x-z);
% end uncomment code
dz = ((sqrt(L1*C2)*k)/C2)*y;

% Q is a 4 by 4 matrix, so it has 12
% elements.
% Since the input data is a column
% vector, rearrange the last 12
% elements of the input data in a
% square matrix.

Q = [X(5), X(9),  X(13),X(17);
     X(6), X(10), X(14),X(18);
     X(7), X(11), X(15),X(19);
     X(8), X(12), X(16),X(20)];

% Linearized system (Jacobian)
 J = [ 0 sqrt(L1*C2) 0 0;
        0 -(sqrt(L1*C2)/C1)*fourElementW(w)
        sqrt(L1*C2)/C1 0
        %0 (sqrt(L1*C2)/(k*L1))
        % 0 -(sqrt(L1*C2)/(k*L1))
        % replace row above with
        % row below for one negative element
        % memsristor circuit
        0 -(sqrt(L1*C2)/(k*L1)) 0
(sqrt(L1*C2)/(k*L1))
        0 0 ((sqrt(L1*C2)*k)/C2) 0];

% Multiply J by Q to form a variational
% equation
F = J*Q;

OUT = [dw;dx; dy; dz; F(:)];
end
```

The Lyapunov exponent program for the canonical memristor-based chaotic circuit is shown below.

```
function OUT = fourDMemristorCanonical(t,X)
% Lyapunov exponent computation for
% Four-D Canonical Memristor
% Settings:  ODEFUNCTION: fourdm
% Final Time:  10000, Step: 1,
% Relative & Absolute Tol: 1e-007
% No. of discarded transients: 100,
% update Lyapunov: 10
% Initial Conditions: 0 0 0 2e-5,
% no. of linearized ODEs: 16

% The first 4 elements of the input
% data X correspond to the
% 4 state variables.  Restore them.
% The input data X is a 12-element
% vector in this case.
p = X(1); q = X(2);r = X(3);s = X(4);
% time scaling
tau = 1/sqrt(7.07e-3*49.5e-9);
% ODE
dp = (q*10e3)/tau;
dq = (1/(tau*5.5e-9))*(r/(2*1428)-q/1428
-canonicalW(p)*q);
dr = (1/(tau*49.5e-9))*((2*q)/1428-r/1428
-s/10);
ds = (10*r)/(tau*7.07e-3);

% Q is a 4 by 4 matrix, so it has
% 12 elements.
% Since the input data is a column
% vector, rearrange
% the last 12 elements of the input
% data in a square matrix.

Q = [X(5), X(9),  X(13),X(17);
     X(6), X(10), X(14),X(18);
     X(7), X(11), X(15),X(19);
     X(8), X(12), X(16),X(20)];

% Linearized system (Jacobian)
  J = [ 0,10e3/tau,0,0;
        0,-(1/(tau*5.5e-9))*(1/1428+
        canonicalW(p)),
        (1/(tau*5.5e-9))*(1/(2*1428)),0;
        0,(1/(tau*49.5e-9))*(2/1428),
        -(1/(tau*49.5e-9))*(1/1428),
        -(1/(tau*49.5e-9))*(1/10);
         0,0,10/(tau*7.07e-3),0];

% Multiply J by Q to form a variational
%equation
F = J*Q;

OUT = [dp; dq; dr; ds;F(:)];
end
```

For the Time-Series method, we use the same programs above, but the call functions are different and are given below. First is the call function for the four-element memristor-based chaotic circuits followed by the call function for the canonical memristor-based chaotic circuit.

```
options = odeset('RelTol',1e-7,'AbsTol',1e-7);
[T,Res]=lyapunov(4,@fourElementMemristor,@ode45,
0,0.01,1000,[0 0.1 0 0],10);
```

```
plot(T,Res);
title('Dynamics of Lyapunov exponents');
xlabel('Time'); ylabel('Lyapunov exponents');
options = odeset('RelTol',1e-7,'AbsTol',1e-7);
[T,Res]=lyapunov(4,@fourDMemristorCanonical,
@ode45,options,0,1,10000,[0 0 0 2e-5],1);
plot(T,Res);
title('Dynamics of Lyapunov exponents');
xlabel('Time'); ylabel('Lyapunov exponents');
```

REFERENCES

[1] Leon O. Chua, "Memristor - The Missing Circuit Element", *IEEE Transactions on Circuit Theory*, vol. CAT-18, no. 5, pp. 507–519, 1971.

[2] D.B. Strukov, G.S Snider, G.R. Stewart, and R.S Williams, "The Missing Memristor Found", *Nature*, vol. 453, pp. 80–83, 2008.

[3] P. Stavroulakis, *Chaos Applications in Telecommunications*, CRC Press, 2006.

[4] P.K. Roy and A. Basuray, "A High Frequency Chaotic Signal Generator: A Demonstration Experiment", *American Journal of Physics*, vol. 71, pp. 34–37, 2003.

[5] K. S. Sudheer and M. Sabira, "Adaptive Function Projective Synchronization of two-cell Quantum-CNN Chaotic Oscillators with Uncertain Parameters", *Physics Letters A*, vol. 373, pp. 1847–1851, 2009.

[6] Makoto Itoh and Leon O. Chua, "Memristor Oscillators", *International Journal of Bifurcation and Chaos*, vol. 18, no. 11, pp. 3183 – 3206, November 2008.

[7] A.S. Elwakil and M.P. Kennedy, "Construction of Classes of Circuit-Independent Chaotic Oscillators Using Passive-Only Nonlinear Devices", *IEEE Transactions on Circuits and Systems - I*, vol. 48, no. 3, pp. 289–308, 2001.

[8] M. Hasler, M.P. Kennedy, and J. Schweizer, "Secure Communications Via Chua's Circuit", *International Symposium on Nonlinear Theory and its Applications*, , no. 4, pp. 87–92, 1993.

[9] Ruy Barboza and Leon O. Chua, "The Four-Element Chua's Circuit", *International Journal of Bifurcation and Chaos*, vol. 18, no. 4, pp. 943–955, 2008.

[10] Leon O. Chua, M. Komuro, and Takashi Matsumoto, "The Double Scroll Family", *IEEE Transactions on Circuits and Systems*, vol. 33, no. 11, pp. 1072–1118, 1986.

[11] J.P. Eckmann and D. Ruelle, "Ergodic Theory of Chaos and Strange Attractors", *Review of Modern Physics*, vol. 57, no. 3, pp. 617–656, July 1985.

[12] Alan Wolf, Jack B. Swift, Harry L. Swinney, and John A. Vastano, "Determining Lyapunov Exponents from a Time Series", *Physica 16D*, pp. 285–317, 1985.

[13] Steve Siu, "Lyapunov exponent toolbox", *http://www.mathworks.com/matlabcentral/fileexchange/*, June 2008.

[14] Vasiliy Govorukhin, "Lyapunov exponents for odes", *http://www.mathworks.com/matlabcentral/fileexchange/*, June 2008.

[15] Zhigang Li and Daolin Xu, "A Secure Communication Scheme Using Projective Chaos Synchronization", *Chaos, Solitons and Fractals*, vol. 22, no. 2, pp. 477–481, 2004.

[16] U. Parlitz L. Kocarev and T. Stojanovski, "An Application Of Synchronized Chaotic Dynamic Arrays", *Physical Letters A*, vol. 217, pp. 280–284, 1996.

[17] J.F. Heagy T.L. Carroll and L.M. Pecora, "Transforming Signals With Chaotic Synchronization", *Physical Review E*, vol. 54, pp. 4676–4680, 1996.

[18] Takashi Matsumoto, Leon O. Chua, and K. Kobayashi, "Hyperchaos: Laboratory Experiment and Numerical Confirmation", *IEEE Transactions on Circuits and Systems*, vol. 33, no. 11, pp. 1143–1147, 1986.

[19] Toschimichi Saito, "An Approach Toward Higher Dimensional Hysteresis Chaos Generators", *IEEE Transactions on Circuits and Systems*, vol. 37, no. 3, pp. 399–409, March 1990.

[20] T. Matsumoto, Leon O. Chua, and K. Kobayashi, "Hyperchaos: Laboratory Experiment and Numerical Confirmation", *IEEE Transactions on Circuits and Systems: Letters*, vol. 33, no. 11, pp. 1143–1147, 1986.