# Memristor Based Chaotic Circuits

*Bharathwaj Muthuswamy*

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2009-6
http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-6.html

January 15, 2009

Acknowledgement

Many thanks to Prof. Leon O. Chua for his support and guidance.

# Memristor Based Chaotic Circuits

Bharathwaj Muthuswamy
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720
Email: mbharat@cory.eecs.berkeley.edu.

**Abstract**

In this work, we present two memristor based chaotic circuits. These circuits are obtained by replacing the nonlinear resistor in the four element Chua's circuit with a memristor. Hence, these chaotic circuits use only the four basic circuit elements. Moreover, one of these circuits has only one negative element in addition to the nonlinearity. The four element Chua's circuit requires two negative elements in addition to the nonlinearity.

## I. Introduction

THE memristor was postulated as the fourth circuit element by Leon O. Chua in 1971 [2]. Williams et. al. [8] discovered the memristor in 2008 thereby cementing its place as the 4th circuit element. On the other hand, Barboza and Chua [1] proposed the simplest possible chaotic circuit, shown in Fig. 1.

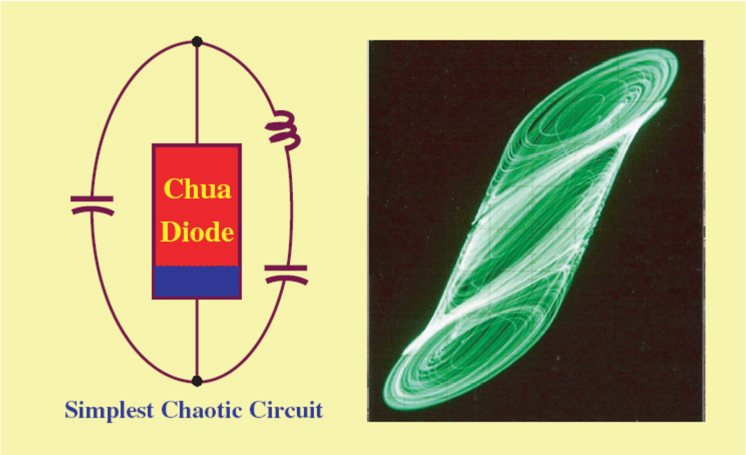


Fig. 1: The simplest Chua's circuit and its typical attractor [1]

A typical implementation of this circuit is shown in Fig. 2 [1]. $A_1$ is the op-amp for current inversion, this sets a negative $L_1$ and $C_2$. $N_R$ is the Chua diode.

Many thanks to Prof. Leon O. Chua for his support and guidance

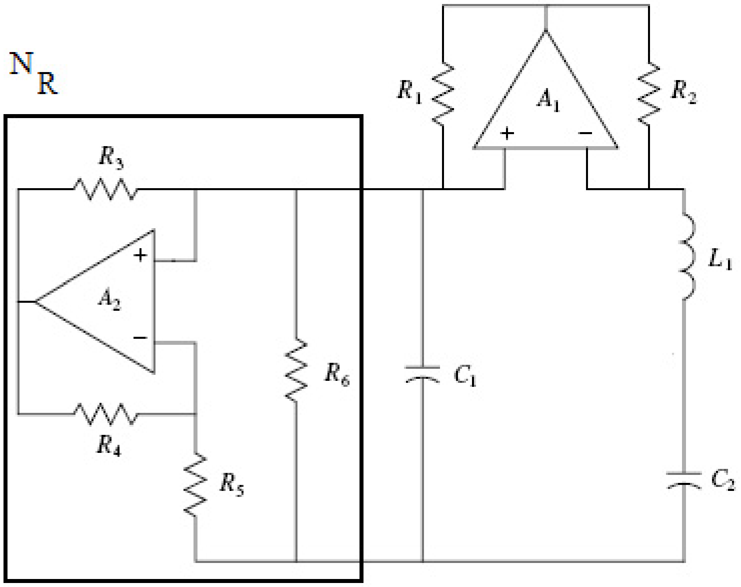


Fig. 2: A realization of the four element Chua's circuit [1]

It turns out that if the Chua diode is replaced by a memristor (with the appropriate nonlinearity), we still obtain chaos. Itoh and Chua [6] proposed some similar circuits, this work was done in parallel during Summer 2008. The work herein was proposed during the author's Qualifying Examination in August of 2008.

The organization of this report is: in the next section, we describe the first memristor based chaotic circuit along with simulation results. Next, the simpler single negative element memristor based chaotic circuit is proposed. Lyapunov exponents are computed for the two circuits via two different methods: the QR method from [4] and the time-series method from [9].

## II. Memristor chaotic circuit 1: Replacing nonlinear resistor $N_R$ with memristor

Fig. 3 shows the four-element Chua's circuit but with the nonlinear resistor replaced by the memristor. Fig. 4 shows this circuit reduced down to the basic circuit elements.

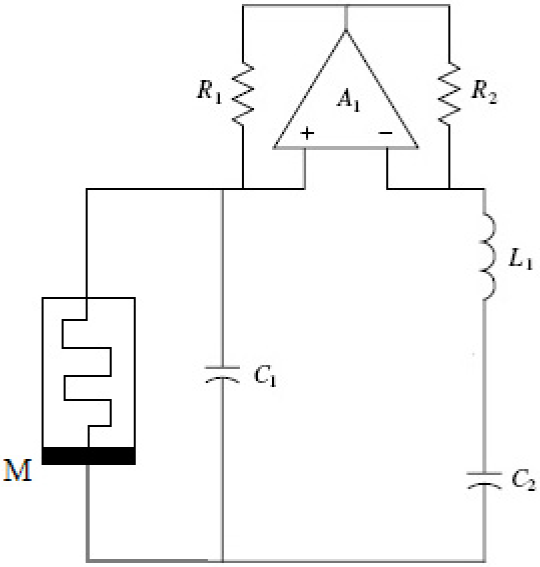


Fig. 3: Four Element Memristor based chaotic circuit

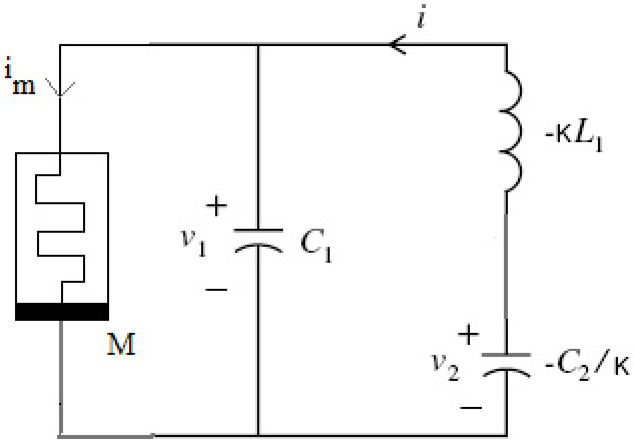


Fig. 4: Four Element Memristor based chaotic circuit showing only the basic circuit elements. The effect of op-amp $A_1$ from Fig. 3 is the set $-\kappa$.

The memristor M in Fig. 4 is a *flux controller memristor* whose driving-point characteristic [2] is given by:

$$i_m(t) = W(\phi(t))v_1(t) \tag{1}$$

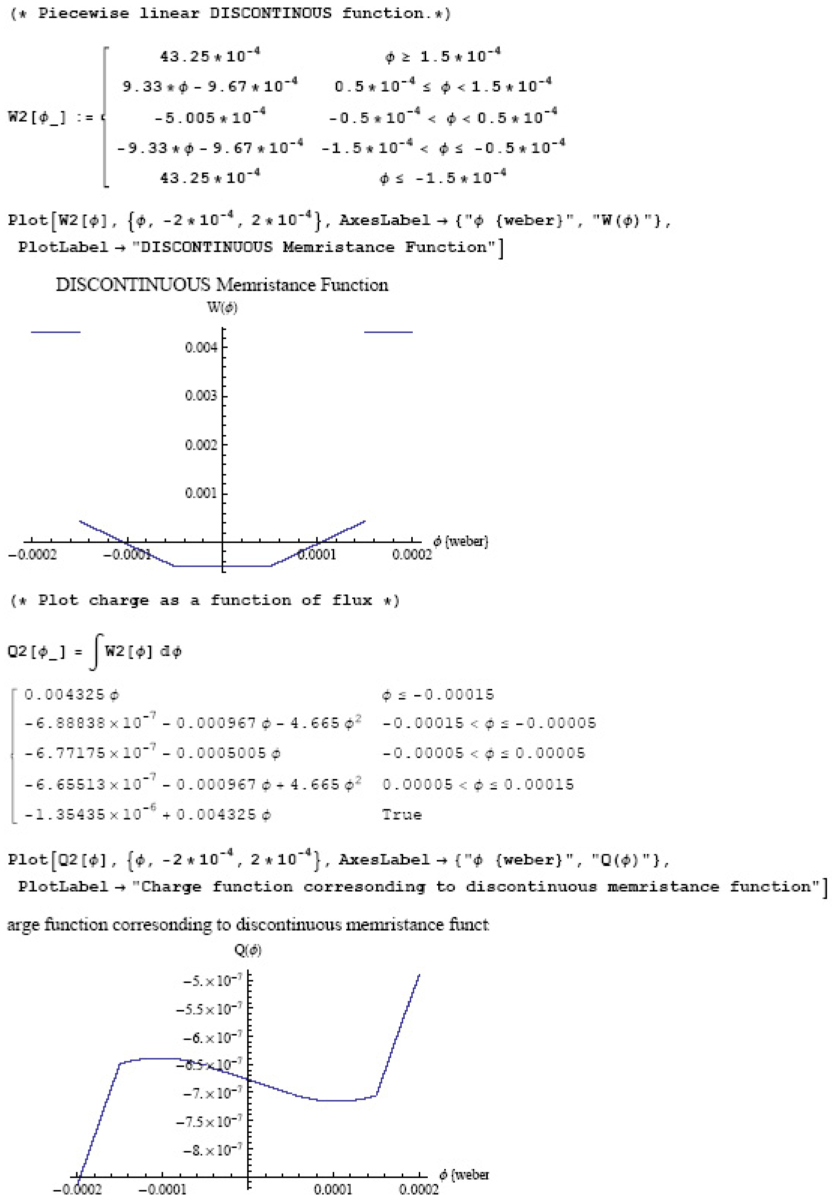
Here, $W(\phi(t))$ is called as the *incremental menductance*, defined as [2]:

$$W(\phi) = \frac{dq_m(\phi)}{d\phi} \tag{2}$$

Using (1) we can write the system equations for Fig. 4:

$$\begin{aligned} \frac{d\phi}{dt} &= v_1 \\ i - W(\phi)v_1 &= C_1\frac{dv_1}{dt} \\ \kappa L_1\frac{di}{dt} &= v_1 - v_2 \\ i &= \frac{C_2}{\kappa}\frac{dv_2}{dt} \end{aligned} \tag{3}$$

The parameters for (3) are: $C_1 = 33\ nF$, $C_2 = 100\ nF$, $L_1 = 10\ mH$ and $\kappa = 8.33$. In (3), we are free to pick $W(\phi)$. Using a simple piecewise linear $W(\phi)$ shown below, we get the results shown in the next section.



```
(* Piecewise linear DISCONTINOUS function.*)

             ⎡     43.25*10^-4               ϕ ≥ 1.5*10^-4
             ⎢  9.33*ϕ - 9.67*10^-4     0.5*10^-4 ≤ ϕ < 1.5*10^-4
W2[ϕ_] := ⎨     -5.005*10^-4            -0.5*10^-4 < ϕ < 0.5*10^-4
             ⎢ -9.33*ϕ - 9.67*10^-4    -1.5*10^-4 < ϕ ≤ -0.5*10^-4
             ⎣     43.25*10^-4               ϕ ≤ -1.5*10^-4

Plot[W2[ϕ], {ϕ, -2*10^-4, 2*10^-4}, AxesLabel → {"ϕ {weber}", "W(ϕ)"},
 PlotLabel → "DISCONTINUOUS Memristance Function"]
```

```
(* Plot charge as a function of flux *)

Q2[ϕ_] = ∫W2[ϕ] dϕ

⎡ 0.004325 ϕ                                 ϕ ≤ -0.00015
⎢ -6.88838 × 10^-7 - 0.000967 ϕ - 4.665 ϕ^2  -0.00015 < ϕ ≤ -0.00005
⎨ -6.77175 × 10^-7 - 0.0005005 ϕ             -0.00005 < ϕ ≤ 0.00005
⎢ -6.65513 × 10^-7 - 0.000967 ϕ + 4.665 ϕ^2  0.00005 < ϕ ≤ 0.00015
⎣ -1.35435 × 10^-6 + 0.004325 ϕ              True

Plot[Q2[ϕ], {ϕ, -2*10^-4, 2*10^-4}, AxesLabel → {"ϕ {weber}", "Q(ϕ)"},
 PlotLabel → "Charge function corresonding to discontinuous memristance function"]
```

Fig. 5: Memristance function $W(\phi)$ as defined in Mathematica. We also compute the $q(\phi)$ function.

## III. Memristor chaotic circuit 1: Simulation Results

Fig. 6 and Fig. 7 show the simulation results for the four element Memristor chaotic circuit. The appendix has the MATLAB code used to obtain these results.

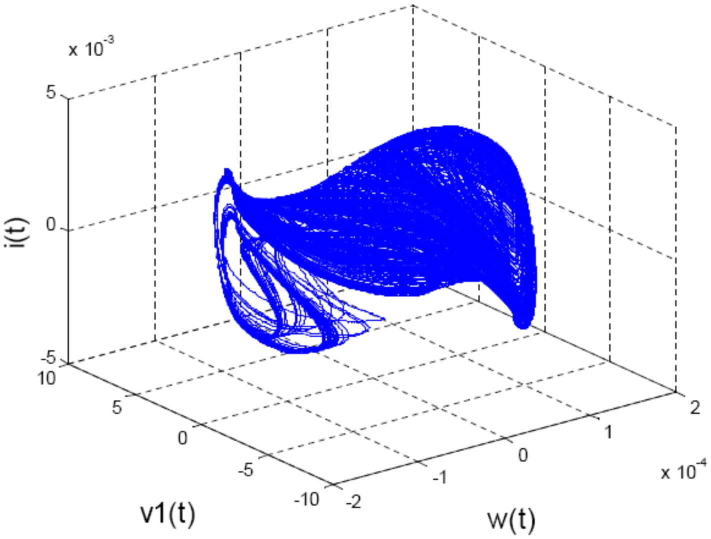


Fig. 6: 3D attractor from the four element Memristor chaotic circuit.

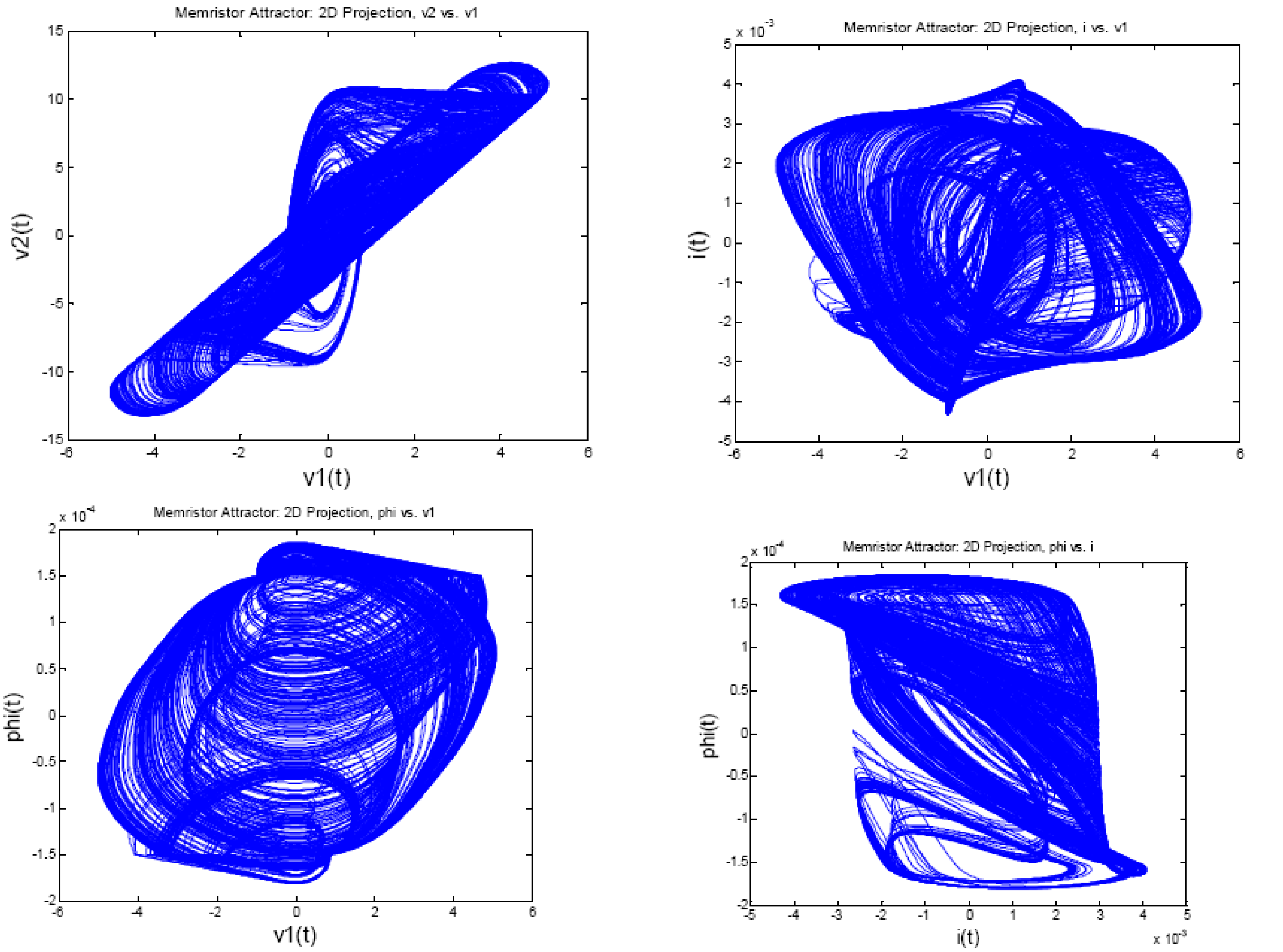


Fig. 7: 2D Projections of the attractor from the four element Memristor chaotic circuit.

## IV. Memristor chaotic circuit 2

In the circuit shown in Fig. 4, if we let $C_2$ be the only negative element and set $\kappa$ to 1, we still get chaotic behaviour. The simulation results are shown in the next section, MATLAB simulation code is given in the appendix.

## V. Memristor chaotic circuit 2: Simulation Results

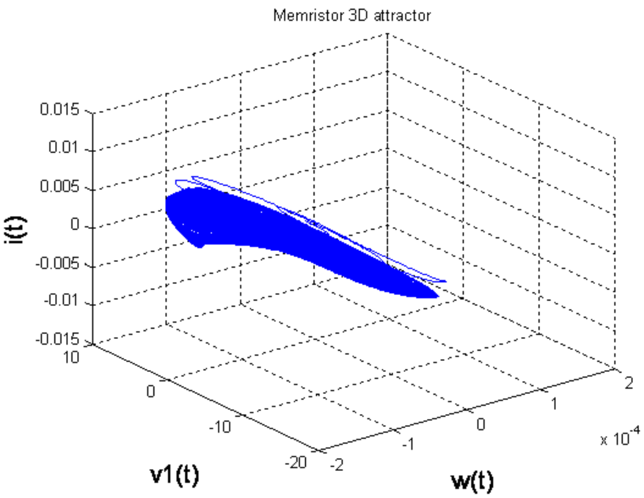


Fig. 8: 3D attractor from the four element Memristor chaotic circuit with only one negative element.

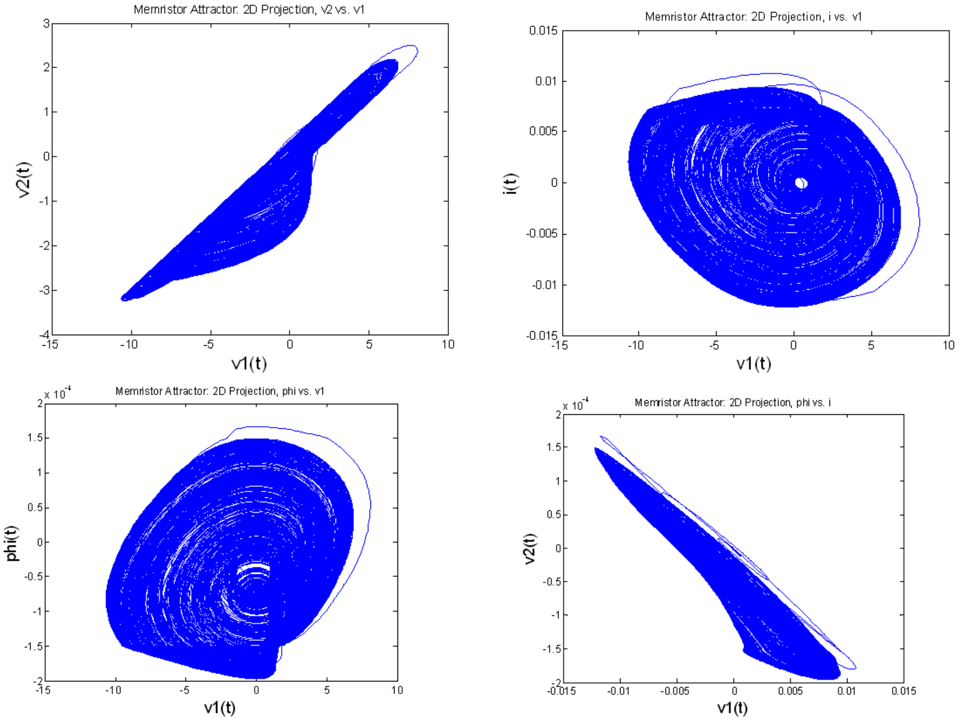


Fig. 9: 2D Projections of the attractor from the four element Memristor chaotic circuit with only one negative element.

## VI. Lyapunov exponent calculation

Before computing the Lyapunov exponents, the time scales for the first two circuits are scaled to the order of seconds by: $\tau = \sqrt{|L_1 C_2|}$. This is necessary because the Lyapunov exponent algorithms work best 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 methods to estimate the Lyapunov exponents: the QR method from [4] and the time-series method from [9]. The LET toolbox from [7] and the Lyapunov time series toolbox from [5] have been used to estimate the exponents. The table below summarizes our results. The Appendix has the MATLAB code used for obtaining the values in the table below.

TABLE I: Summary of Lyapunov exponents

| Circuit | QR Method | Time Series Method |
|---|---|---|
| Four Element | 0.11, 0, 0, -1.16 | 0.1, 0, 0, -1.2 |
| Four Element with only negative $C_2$ | 0.1, 0, 0, -0.7 | 0.1, 0, 0, -0.7 |

## VII. Conclusions

Let us analyze each row in the table above:

1) The four element chua's circuit with a memristor has one positive Lyapunov exponent, this indicates the presence of chaos [9].
2) The four element circuit with only a negative capacitance also gives to rise to chaotic behaviour because of the positive Lyapunov exponent [9]. However, Barboza and Chua's four-element circuit [1] without the memristor does not seem to have chaotic behavior for the same set of $C_1$, $L_1$, $C_2$ and $\kappa$ 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.

## VIII. Future Work

This paper has presented two possible memristor based chaotic circuits. Future work includes:

1) Determining the route to chaos (period doubling etc.) in these circuits [3].
2) Proving the circuits are chaotic rigorously by way of Shilnikov's theorem [3].
3) A rigorous bifurcation analysis.
4) A strategy for implementing these memristor-based chaotic circuits on an embedded platform. A method based on LABVIEW RT and LABVIEW FPGA is being investigated by the author.

## References

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

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

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

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

[5] V. Govorukhin. (2008, June) Lyapunov Exponents for ODEs. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=4628

[6] M. Itoh and L. O. Chua, "Memristor Oscillators," To appear in November 2008 issue of IJBC.

[7] S. Siu. (2008, June) Lyapunov Exponent Toolbox. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=233&objectType=File

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

[9] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov Exponents from a Time Series," *Physica 16D*, pp. 285–317, 1985.

## Appendix
## MATLAB simulation code for memristor chaotic circuit 1

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, June 13th 2008 - July 13th 2008
%% 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, July 17th 2008
%% 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
## MATLAB simulation code for memristor chaotic circuit 2

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, June 13th 2008 - July 3rd 2008
%% 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

### Lyapunov Exponent program for use with LET: Four Element Memristor Circuit(s)

Please note that the M-files below are configured for the single negative element circuit.

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

% COMPUTED LYAPUNOV EXPONENTS (TWO negative elements):
% 0.11,0,0,-1.1696
% COMPUTED LYAPUNOV EXPONENTS (ONE negative element):
% 0.1,0,0,-0.70
% 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);

% 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 fourElementW.m below is used by the M-file above

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

## Appendix

## Lyapunov Exponent program for use with Time-Series method: Four Element Memristor Circuit(s)

Use the same two programs as the previous appendix, but the call function is different and is given below:

```
[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’);
```