International Journal of Bifurcation and Chaos, Vol. 20, No. 5 (2010) 1335–1350

DOI: 10.1142/S0218127410026514

IMPLEMENTING MEMRISTOR BASED CHAOTIC CIRCUITS

BHARATHWAJ MUTHUSWAMY
Department of Electrical Engineering, Milwaukee School of Engineering, S342 Fred Loock Engineering Center, 1025 North Broadway, Milwaukee, Wisconsin 53202, USA
muthuswamy@msoe.edu

Received November 5, 2009; Revised January 15, 2010

This paper provides a practical implementation of a memristor based chaotic circuit. We realize a memristor using off-the-shelf components and then construct the memristor along with the associated chaotic circuit on a breadboard. The goal is to construct a physical chaotic circuit that employs the four fundamental circuit elements — the resistor, capacitor, inductor and the memristor. The central concept behind the memristor circuit is to use an analog integrator to obtain the electric flux across the memristor and then use the flux to obtain the memristor's characterstic function.

Keywords: Memristor; chaotic circuit; Chua's circuit.

1. Introduction

The memristor was postulated as the fourth circuit element by Leon O. Chua [1971]. 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. For over thirty years, the memristor was not significant in circuit theory. In 2008, Stan Williams and others [Strukov *et al.*, 2008] fabricated a solid state implementation of the memristor and thereby cemented its place as the fourth circuit element. Potential applications of such memristors span diverse fields ranging from nonvolatile memories on the nano-scale [Strukov *et al.*, 2008] to modeling neural networks [Persin & Di Ventra, 2009]. Since a memristor is a fundamental circuit element, circuit applications of memristors are also active topics of research [Di Ventra *et al.*, 2009]. Among these are applications of memristor based chaotic circuits, for example, Zhao and Wang [2009] proposed a memristor chaotic circuit for image encryption.

One of the first memristor based chaotic circuits was proposed by Itoh and Chua [2008]. Their paper uses a passive nonlinearity based on Williams' [Strukov *et al.*, 2008] memristor model. Muthuswamy and Kokate [2009] proposed other memristor based chaotic circuits. However, none of those papers suggest an implementation of a memristor. Since memristors are commercially unavailable (as of this writing), it would be very useful to have a circuit that emulates a memristor. Pershin and Di Ventra [2009] suggested a microcontroller emulation of a memristor. But the frequency range of this memristor is limited to approximately 50 Hz [Persin & Di Ventra, 2009]. Another approach is to use mutators [Chua, 1971]. This would require a circuit element (say a resistor) that already has the desired memristor characteristic (in the form of the resistor's i–v graph). The mutator would then

"mutate" this i–v graph into the desired q–ϕ characteristic of the memristor.

In contrast to the above two approaches, this paper proposes a simpler analog realization of a flux-controlled memristor. Note that this memristor is not an emulation but a physical device. In other words, we can seal our memristor in a black-box with only two external wires (along with three internal wires for power supply and ground). The circuit is also suitable for prototyping on a bread-board and has frequency content in the 0.5 KHz (audible) range.

This paper is organized as follows: we begin by discussing the fundamental theory behind memristors. Next, we give an overview of the memristor based chaotic circuit, followed by simulation results and the computation of Lyapunov exponents. This is followed by experimental results, then we illustrate period-doubling route to chaos. The paper concludes with a discussion of future work, acknowledgments, references and appendices.

2. Introduction to the Memristor

The memristor is a two-terminal element, in which the magnetic flux (ϕ) between the terminals is a function of the electric charge (q) that passes through the device [Chua, 1971]. The memristor used in this work is a flux controlled memristor that is characterized by its incremental memductance [Strukov *et al.*, 2008] function $W(\phi)$ describing the flux-dependent rate of change of charge:

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

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

$$i(t) \triangleq \frac{dq}{dt} = \frac{dq}{d\phi}\frac{d\phi}{dt} = W(\phi(t))v(t) \tag{2}$$

In Eq. (2), we used the chain rule to simplify dq/dt and then used the definition of $W(\phi)$ from Eq. (1). The circuit symbol for a memristor is shown in Fig. 1.

The justification for memristor as an acronym for memory-resistor should be clear: in Eq. (2), since $W(\phi(t)) = W(\int v(t))$, the integral operator on the memductance 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.

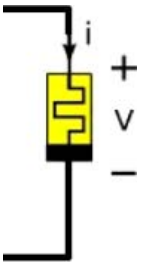


Fig. 1. The circuit symbol for a memristor.

3. Overview of the Memristor Based Chaotic Circuit and System Equations

To design the memristor based chaotic circuit, we started with Chua's circuit and simply replaced the Chua diode with a flux-controlled memristor [Muthuswamy & Kokate, 2009]. Refer to Fig. 2 for the circuit.

The equations for the memristor based chaotic circuit in Fig. 2 are:

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

The derivation for Eq. (3) is given in the appendix. $i(t)$ is defined via Eq. (2):

$$i(t) = W(\phi(t))v_1(t) = \frac{dq}{d\phi}v_1(t) \tag{4}$$

We chose a cubic nonlinearity for the q–ϕ function:

$$q(\phi) = \alpha\phi + \beta\phi^3 \tag{5}$$

The intuition is that smooth nonlinearities should be easier to implement. Also, Chua's circuit with a cubic nonlinearity does give rise to chaos [Zhong, 1994]. Therefore, the memductance function $W(\phi)$ is given by:

$$W(\phi) = \frac{dq}{d\phi} = \alpha + 3\beta\phi^2 \tag{6}$$

To obtain circuit parameters, we started with a set of parameters for Chua's circuit from [Zhong, 1994] that gives rise to chaos: $L = 18.91\,\text{mH}$, $C_2 = 78\,\text{nF}$, $R = 2.5\,\text{k}\Omega$ potentiometer, $C_1 = 7\,\text{nF}$. We then settled on realistic values close to the above: $L = 18\,\text{mH}$, $C_2 = 68\,\text{nF}$, $R = 2.5\,\text{k}\Omega$ potentiometer,

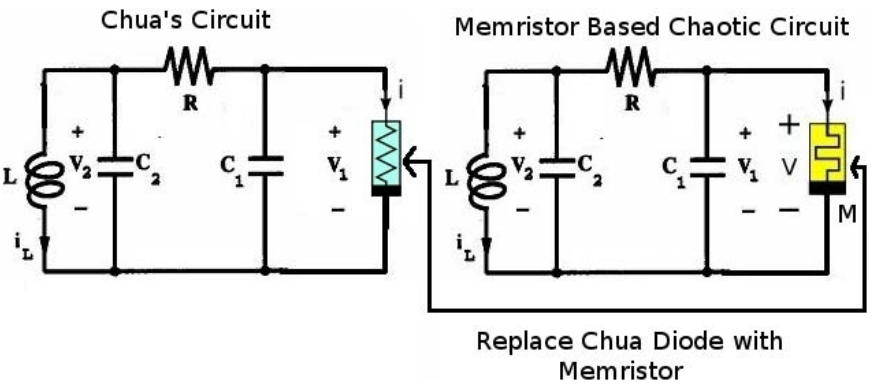


Fig. 2. The memristor chaotic circuit. It has been derived from Chua's circuit by replacing the Chua diode (colored blue) with the flux-controlled memristor M (colored yellow).

$C_1 = 6.8\,\text{nF}$. R is set to $2000\,\Omega$ to obtain the chaotic attractor. We let $\alpha = -0.667 \cdot 10^{-3}$ and $\beta = 0.029 \cdot 10^{-3}$ in Eq. (5), similar to values in [Zhong, 1994]. Before implementation, we simulated Eq. (3) with the q–ϕ function in Eq. (5) along with the parameter values above. The simulation is the subject of the next section.

4. Simulation Results

Simulating the system in the previous section using Mathematica, we get the results shown in Fig. 3. The relevant code is in the appendix, a variation of the code is also available online [Muthuswamy, 2009].

Notice the unrealistic values of voltage (in tens of thousands of volts) in Fig. 3. Therefore, our set of variables requires a rescaling. The new set of equations is shown below, the derivation is given in the appendix. Note that we have slightly abused our notation and reused the flux, voltage and current variables from Fig. 2 in the equation below. This has been done to emphasize the fact that our implementation realizes the memristor chaotic circuit from Fig. 2 despite the rescaling (more on this in the next section).

$$\begin{aligned}
\frac{d\phi}{dt} &= \frac{-v_1(t)}{\zeta} \\
\frac{dv_1(t)}{dt} &= \frac{1}{C_1}\left(\frac{v_2(t) - v_1(t)}{R} - W(\phi(t)) \cdot v_1(t)\right) \\
\frac{dv_2(t)}{dt} &= \frac{1}{C_2}\left(\frac{v_1(t) - v_2(t)}{R} - i_L(t)\right) \\
\frac{di_L(t)}{dt} &= \frac{v_2(t)}{L}
\end{aligned} \tag{7}$$

Notice that we have actually implemented a general *memristive system* [Chua & Kang, 1976],

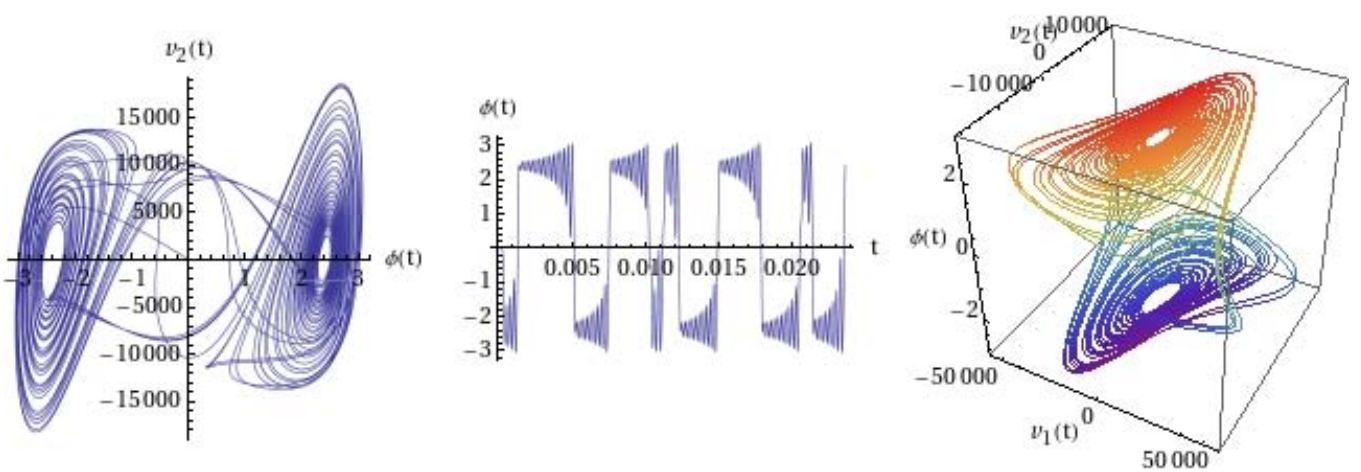


Fig. 3. The Mathematica demonstration output. The voltage values are in volts, the flux values in weber and the time in seconds.

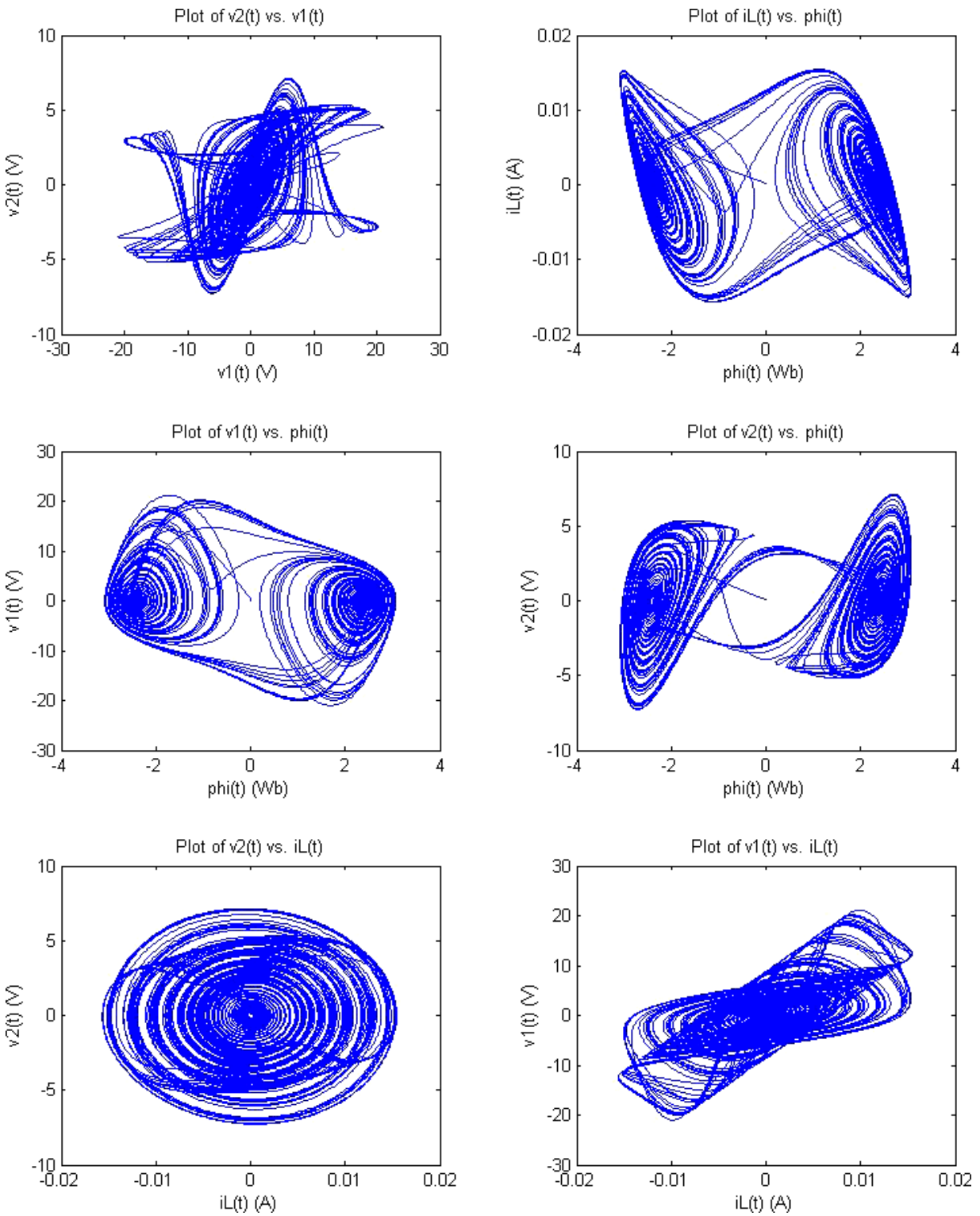


Fig. 4. Simulation results from MATLAB. We plot all the circuit state variables: (ϕ, v_1, v_2, i_L) to illustrate the different attractors generated by Eq. (7).

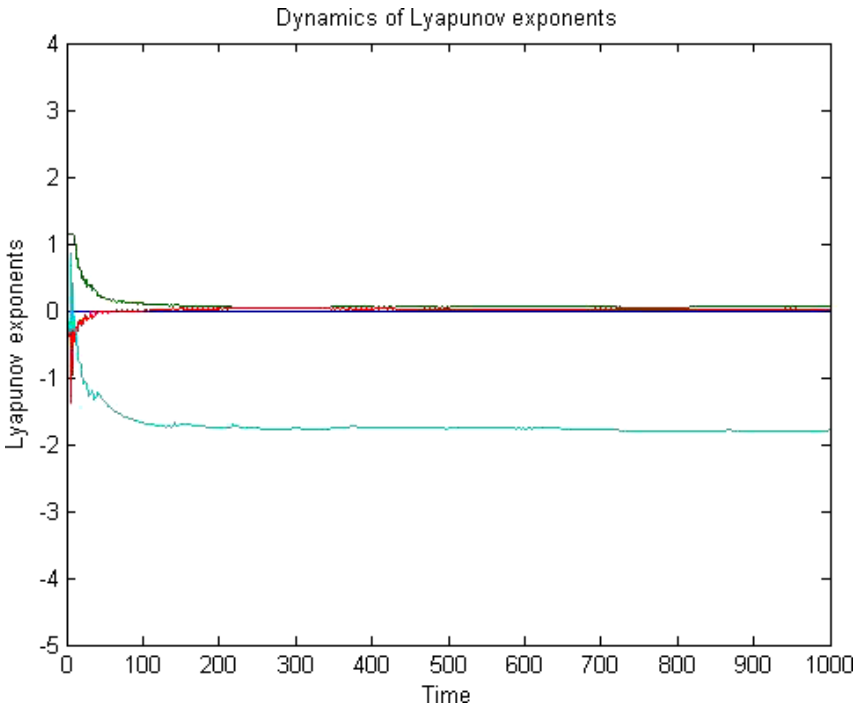


Fig. 5. A plot of the Lyapunov exponents as a function of time. Notice the quick convergence of the Lyapunov exponents program.

namely the voltage-controlled memristive one-port:

$$\left.\begin{aligned} \dot{x} &= f(x,v,t) \\ i &= G(x,v,t)v \end{aligned}\right\} \Rightarrow \begin{aligned} \dot{\phi} &= (-1/\zeta)\cdot v_1 \\ i &= (\alpha + 3\beta\phi^2)\cdot v_1 \end{aligned} \tag{8}$$

Therefore, we have $x = \phi$, $v = v_1$, $f(x,v,t) = -1/\zeta$ and $G(x,v,t) = W(\phi(t)) = \alpha + 3\beta\phi^2$. The parameter ζ in Eq. (7) is our rescaling factor, the negative sign is because of the inverting amplifier in the implementation. We will pick $\zeta = 8200\,\Omega \cdot 47 * 10^{-9}\,\text{nF}$, the exact reason for choosing these component values will be clear in the next section. We now simulate the rescaled system in Eq. (7) with MATLAB. We used a different simulation tool (as opposed to Mathematica) with a different integration method in order to reduce the effects of numerical error. The results are shown in Fig. 4, the simulation code is given in the appendix. Comparing Figs. 3 and 4, we see that rescaling has achieved the desired result of reducing the voltage values to a realistic range. Thus, in the next section we will implement Eq. (7).

To empirically confirm chaos, one can compute the Lyapunov exponents [Govorukhin, 2008]. Such exponents (using two different methods: the time series method [Wolf *et al.*, 1985] and the QR method [Eckmann & Ruelle, 1985]) computed for this circuit are $(0, 0.061, \approx 0, -1.79)$ and $(0, 0.05, \approx 0, -1.83)$. The two methods give exponents that are in close agreement. Notice that one of the exponents is positive and the sum of the exponents is negative indicating the presence of chaos [Muthuswamy & Kokate, 2009]. A sample MATLAB code for computing Lyapunov exponents is given in the appendix. The plot in Fig. 5 shows the Lyapunov exponents as a function of time.

5. Circuit Implementation and Experimental Results

The circuit schematic to realize Eq. (7) is shown in Fig. 6. The synthesis of the circuit is derived in the appendix. The dynamic equations for the circuit in Fig. 6 are:

$$\begin{aligned} \frac{dv_{15}}{dt} &= \frac{-v_1(t)}{R6\cdot C3} \\ \frac{dv_1(t)}{dt} &= \frac{1}{C_1}\left(\frac{v_2(t)-v_1(t)}{\text{Rpot}} - i_m(t)\right) \\ \frac{dv_2(t)}{dt} &= \frac{1}{C_2}\left(\frac{v_1(t)-v_2(t)}{\text{Rpot}} - i_L(t)\right) \\ \frac{di_L(t)}{dt} &= \frac{v_2(t)}{L_1} \end{aligned} \tag{9}$$

Notice that $\zeta = R6\cdot C3$. The fact that you have a negative sign in the first equation above is because

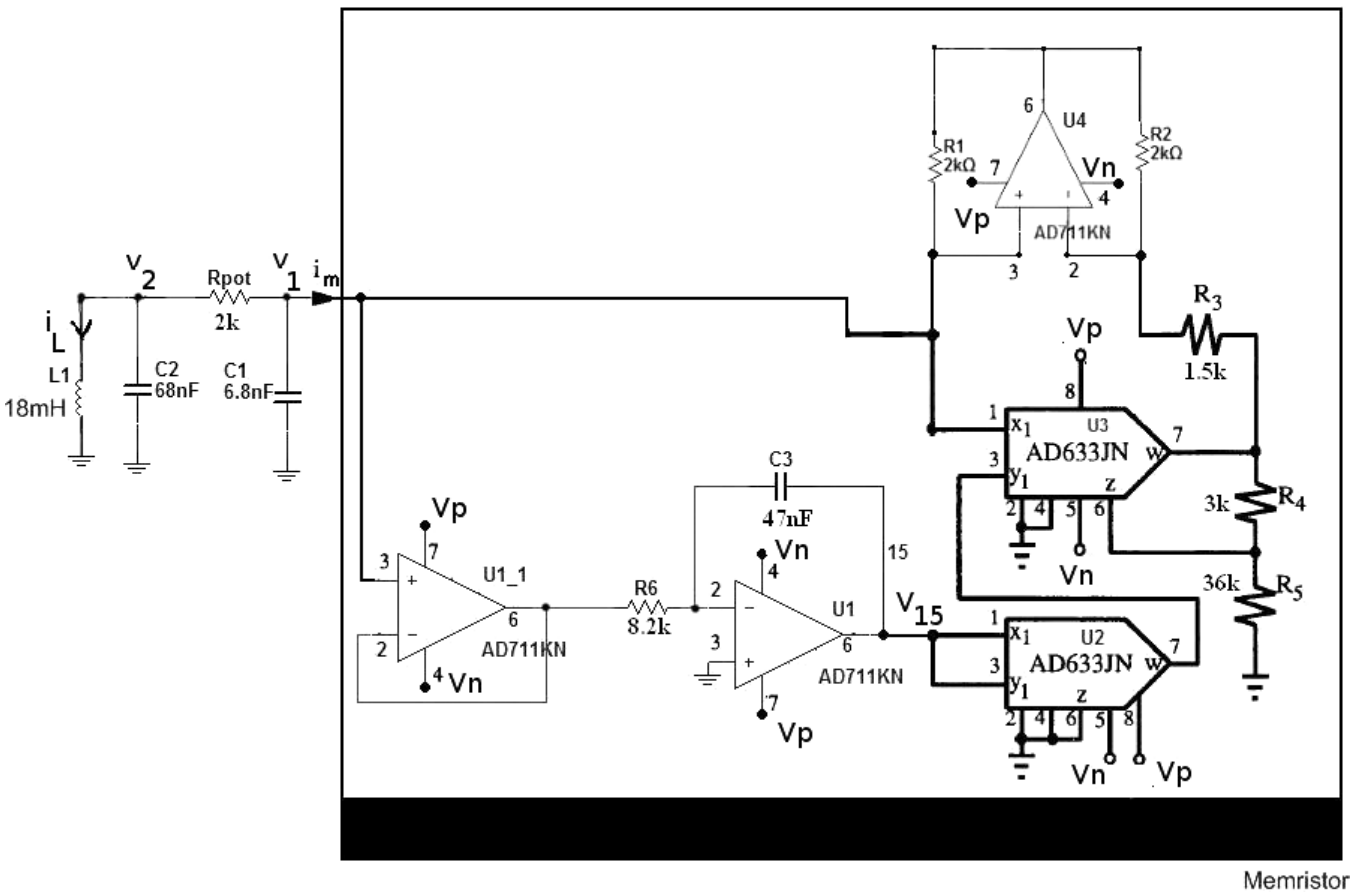


Fig. 6. The schematic of the memristor based chaotic circuit. The power supplies for the ICs are $V_p = +15\,\mathrm{V}$ and $V_n = -15\,\mathrm{V}$. We use an op-amp buffer (U1_1) to avoid loading effects. The circuit in the box above is our memristor realization.

of the inverting nature of the integrator. Thus, we set $R6 = 8.2\,\mathrm{k}\Omega$, $C3 = 47\,\mathrm{nF}$. The other parameters are as stated earlier: $L_1 = 18\,\mathrm{mH}$, $C_2 = 68\,\mathrm{nF}$, Rpot is set to 2000 Ω to obtain the attractor and $C_1 = 6.8\,\mathrm{nF}$. The circuit values corresponding to the memductace parameters α and β are shown in Fig. 6, the formulae relating α and β to the circuit parameters are:

$$\begin{aligned} \alpha &= \frac{-1}{R_3} \\ \beta &= \frac{1}{3}\left(\frac{R_4 + R_5}{R_3 \cdot R_4 \cdot 100}\right) \end{aligned} \tag{10}$$

The detailed derivations of Eqs. (9) and (10) are given in the final appendix. A picture of the breadboarded circuit, the various attractors, time-domain waveforms and their Fourier spectra (all from the physical circuit) are shown in Figs. 7–9. In order to demonstrate that we have actually synthesized a memristor, we experimentally determined the q–ϕ curve of our memristor. These results are detailed in the final appendix.

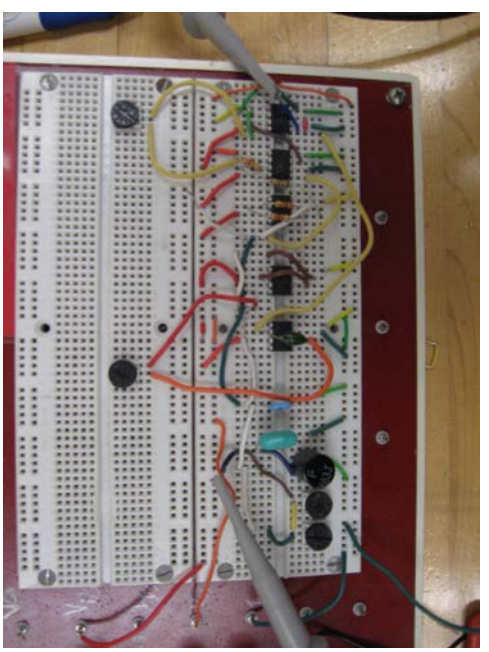


Fig. 7. The physical circuit. We used two potentiometers in series for Rpot from Fig. 6: 2 kΩ in series with a 500 Ω (the smaller pot is for finer control). We also used potentiometers for $R6$ (10 k) and $R5$ (50 k) in Fig. 6.

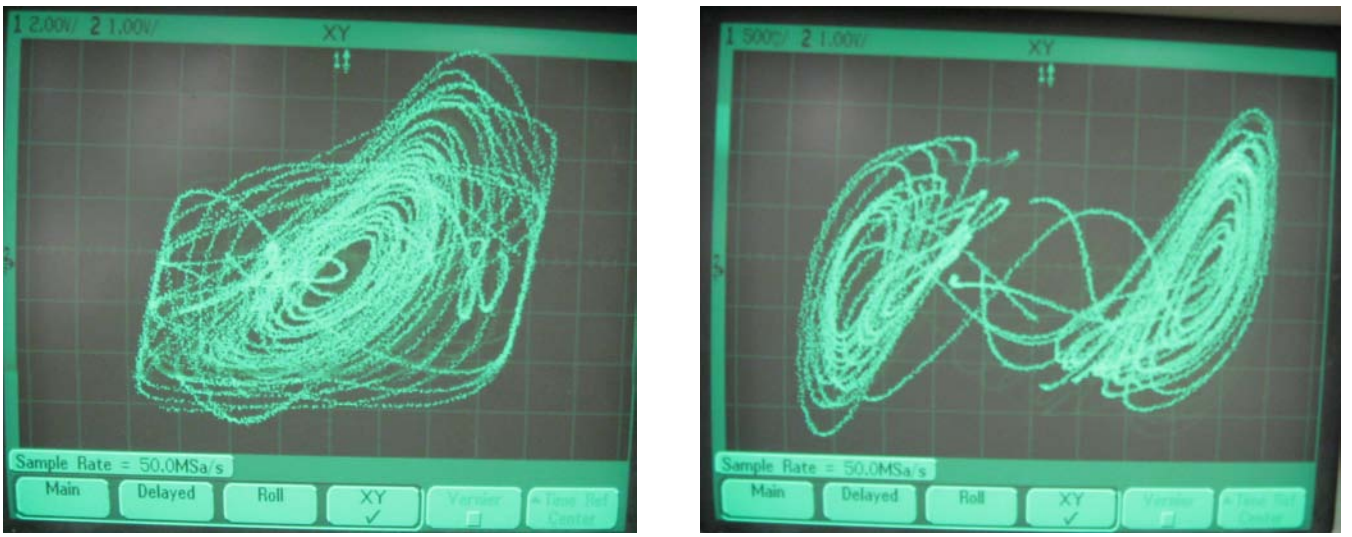


Fig. 8. Two attractors from the physical circuit. The plot on the left is $v_2(t)$ versus $v_1(t)$ and the plot on the right is $v_2(t)$ versus $\phi(t)$. The scales for the left plot are 2.00 V/div for X and 1.00 V/div for Y. The scales for the plot on the right are 0.5 V/div for X and 1.00 V/div for Y. Notice that the plot on the right is in excellent visual agreement with corresponding plot in the simulation (Fig. 4). The plot on the left has some distortion.

Two debugging hints about the physical circuit. First after breadboarding the circuit, you should check that you obtain limit-cycle behavior once you apply power and vary Rpot. If you instead observe DC signals on all variables, it means you have a wiring error. Second, you may have to add reset functionality and null offset circuitry to compensate for the nonidealities in the op-amp. A sign of this

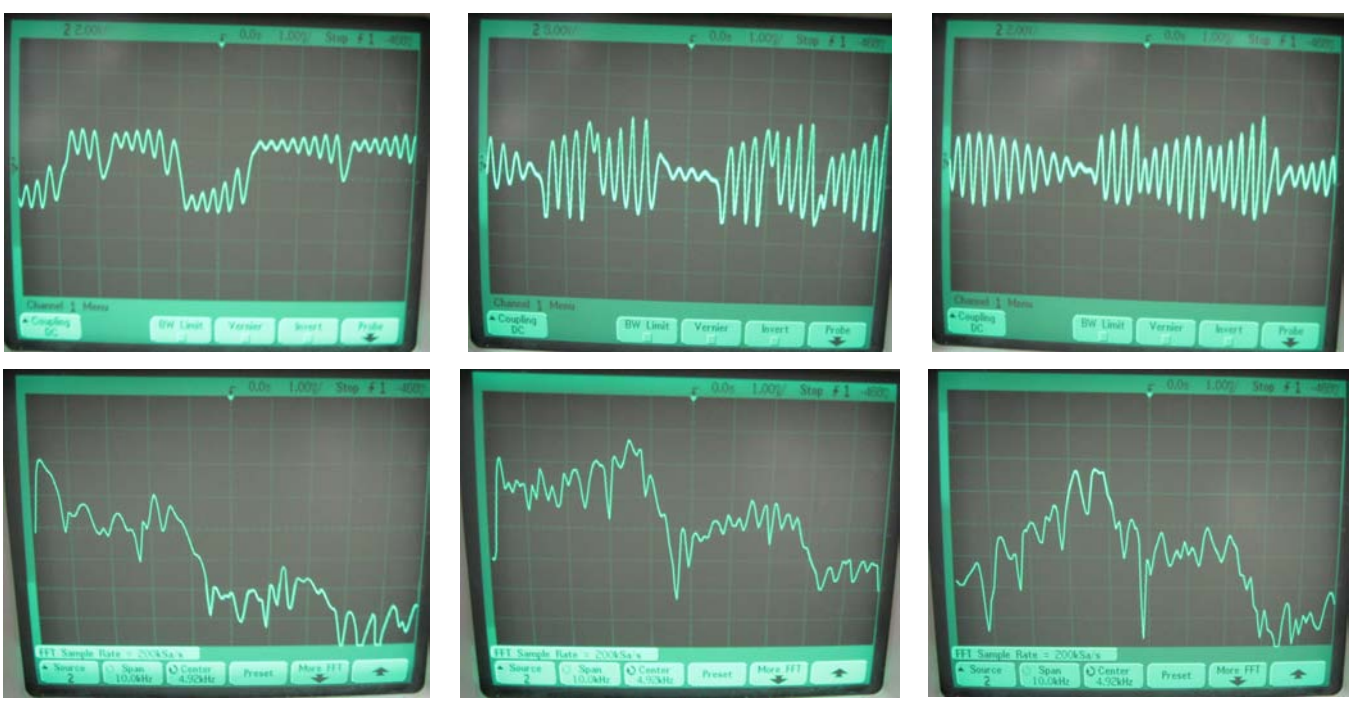


Fig. 9. Time-domain waveforms and their corresponding Fourier spectra. From left to right, first row is $\phi(t)$, $v_1(t)$ and $v_2(t)$ (time scales are in ms). The Y-axis scales are 2.00 V/div, 5.00 V/div and 2.00 V/div, respectively. The scopes have all been placed into "freeze" mode for capturing the time-domain waveforms. On the second row, from left to right, are the Fourier spectra of $\phi(t)$, $v_1(t)$ and $v_2(t)$, respectively. The center frequency for all spectra is approximately 4 KHz with a span of 10 KHz. The vertical scale is 10 dBV and a Hanning window has been used. Notice the wideband nature of the spectra, indicating chaos.

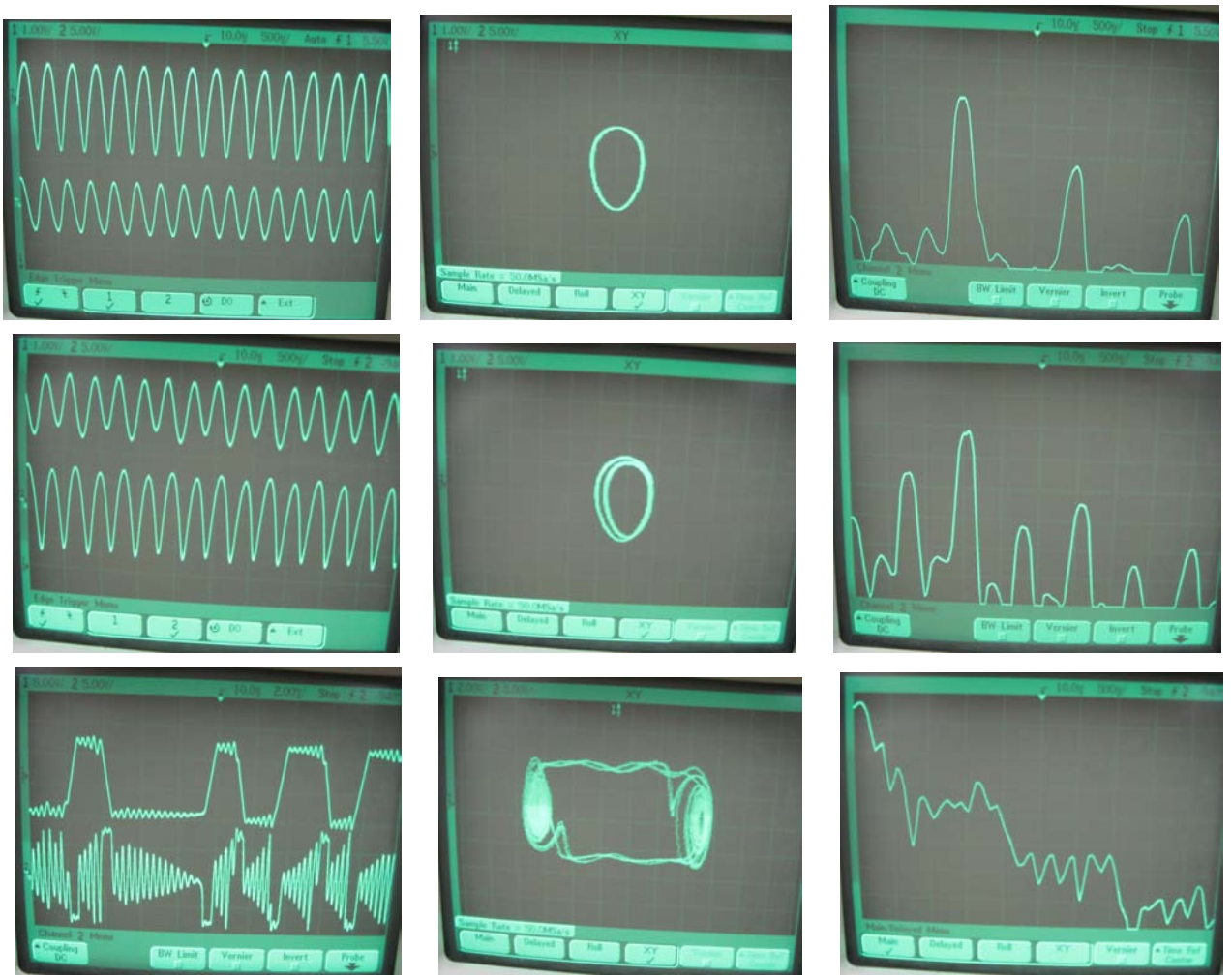


Fig. 10. Time-domain waveforms, phase plane plot and their corresponding Fourier spectra to illustrate period-doubling route to chaos. The waveforms are $\phi(t)$ and $v_1(t)$. X-axis is $\phi(t)$ and Y-axis is $v_1(t)$ for the phase plane plots. The bifurcation parameter from Fig. 6 is Rpot. Rpot = 2.2 k for period-1 limit cycle, Rpot = 2.08 k for period-2 limit cycle and Rpot = 2.0 k for the attractor. The scaling for the Fourier spectrum are the same as in Fig. 9. The scaling for the time-domain waveforms and the phase-plots are 1.00 V/div, 5.00 V/div for the limit cycles and 5.00 V/div for the attractor.

nonideality would be if you just observe transient chaotic attractors.

6. Illustration of Period-Doubling Route to Chaos

Figure 10 illusrates period-doubling chaos. In order to show the robustness of the circuit, we used a different set of parameters in the memductance function: $\alpha = -0.769 \cdot 10^{-3}$ and $\beta = 0.004 \cdot 10^{-3}$.

7. Conclusions and Future Work

In this paper, we showed how to build a memristor based chaotic circuit on a breadboard. The crux of the paper is the analog circuit realization of the memristor via operational amplifiers and analog multipliers. Based on the results from this paper, there are a plethora of opportunity for future work:

(1) It would be instructive to further experimentally characterize the q–ϕ relationship of the memristor design proposed in this paper. For example, the behavior of the q–ϕ curve as functions of α and β could be investigated since $q(\phi) \stackrel{\triangle}{=} \alpha\phi + \beta\phi^3$.
(2) It would be useful to have a memristor simulation framework (via PSPICE for instance). There has already been work done in this regard, refer to [Zhang *et al.*, 2009].
(3) Implement other memductance functions, for instance the classic piecewise linear memductance proposed in [Itoh & Chua, 2008].

(4) Provide rigorous proofs of chaos and bifurcation analysis for memristor chaotic circuits.

Acknowledgment

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

References

Chua, L. O. [1968] "Synthesis of new nonlinear network elements," *Proc. IEEE* **56**, 1325–1340.

Chua, L. O. [1971] "Memristor — The missing circuit element," *IEEE Trans. Circuit Th.* **CAT-18**, 507–519.

Chua, L. O. & Kang, S. M. [1976] "Memristive devices and systems," *Proc. IEEE* **64**, 209–223.

Chua, L. O., Komuro, M. & Matsumoto, T. [1986] "The double scroll family," *IEEE Trans. Circuits Syst.* **CAS-33**, 1073–1118.

Chua, L. O. [1993] "Global unfolding of Chua's circuit," *IEICE Trans. Fund. Electron. Commun. Comput. Sci. E* **76**, 704–734.

Di Ventra, M., Pershin, Y. V. & Chua, L. O. [2009] "Putting memory into circuit elements: Memristors, memcapacitors, and meminductors," *Proc. IEEE* **97**, 1371–1372.

Eckman, J. P. & Ruelle, D. [1985] "Ergodic theory of chaos and strange attractors," *Rev. Mod. Phys.* **57**, 617–656.

Govorukhin, V. [2008] "Calculation Lyapunov exponents for ODE," http://www.mathworks.com/matlabcentral/fileexchange/

Itoh, M. & Chua, L. O. [2008] "Memristor oscillators," *Int. J. Bifurcation and Chaos* **18**, 3183–3206.

Matsumoto, T., Chua, L. O. & Komuro, M. [1985] "The double scroll," *IEEE Trans. Circuits Syst.* **32**, 797–818.

Muthuswamy, B. [2009] "Memristor based chaotic system from the Wolfram demonstrations project," http://demonstrations.wolfram.com/MemristorBasedChaoticSystem/

Muthuswamy, B. & Kokate, P. P. [2009] "Memristor-based chaotic circuits," *IETE Techn. Rev.* **26**, 415–426.

Persin, Y. V. & Di Ventra, M. [2009a] "Experimental demonstration of associative memory with memristive Neural Networks," arXiv:0905.2935, http://arxiv.org.

Persin, Y. V. & Di Ventra, M. [2009b] "Practical approach to programmable analog circuits with memristors," arXiv:0908.3162, http://arxiv.org.

Roy, P. K. & Basuray, A. [2003] "A high frequency chaotic signal generator: A demonstration experiment," *Amer. J. Phys.* **71**, 34–37.

Siu, S. [2008] "Lyapunov exponent toolbox," http://www.mathworks.com/matlabcentral/fileexchange/

Strukov, D. B., Snider, G. S., Stewart, G. R. & Williams, R. S. [2008] "The missing memristor found," *Nature* **453**, 80–83.

Wolf, A., Swift, J. B., Swinney, H. L. & Vastano, J. A. [1985] "Determining Lyapunov exponents from a time series," *Physica D* **16**, 285–317.

Zhang, Y., Zhang, X. & Yu, J. [2009] "Approximate SPICE model for memristor," available via IEEEXplore.

Zhao-Hui, L. & Wang, H. [2009] "Image encryption based on chaos with PWL memristor in Chua's circuit," *Int. Conf. Commun. Circuits Syst.*, pp. 964–968.

Zhong, G. [1994] "Implementation of Chua's circuit with a cubic nonlinearity," *IEEE Trans. Circuits Syst.* **41**, 934–941.

Appendix A

Derivation of Memristor Based Chaotic Circuit Equations

In this appendix, we derive Eq. (3). Note that we have not drawn some of the circuit variables like currents in Fig. 2 to avoid clutter. We use the passive sign convention for all currents and voltages. Now, the voltage across the memristor in Fig. 2 is defined by:

$$\boxed{\frac{d\phi}{dt} \triangleq v_1(t)} \tag{A.1}$$

The current through the capacitor C_1 can be written in terms of the current through resistor R (i_R, flowing from v_2 to v_1) and the current through the memristor M (i) via KCL:

$$i_R = C_1 \frac{dv_1}{dt} + i \tag{A.2}$$

Note that we have used the current–voltage relationship for a linear capacitor in the equation. Simplifying the equation above using the passive sign convention and Ohm's law we get:

$$\boxed{\frac{dv_1(t)}{dt} = \frac{1}{C_1}\left(\frac{v_2(t) - v_1(t)}{R} - i(t)\right)} \tag{A.3}$$

Similarly, applying KCL to the node connecting the inductor L, capacitor C_2 and resistor R, we get:

$$\boxed{\frac{dv_2(t)}{dt} = \frac{1}{C_2}\left(\frac{v_1(t) - v_2(t)}{R} - i_L(t)\right)} \tag{A.4}$$

Finally, applying the $i-v$ relationship definition of the inductor, we get:

$$\boxed{\frac{di_L(t)}{dt} = \frac{v_2(t)}{L}} \tag{A.5}$$

Appendix B

Mathematica Simulation Code

Mathematica demonstration code for Fig. 3 is shown below. Notice that we have a negative sign in the first equation. This is because the physical implementation has an inverting amplifier and the negative sign in the simulation equations makes the simulated attractor visually match the attractor from the physical circuit.

$$
\begin{aligned}
&Manipulate[[Module[\{sol = \{\phi, v_1, v_2, i\}/. \\
&\qquad Quiet[NDsolve[\\
&\qquad \{\phi'[t] == -v_1[t], \\
&v_1'[t] == \frac{1}{C_1}\left(\frac{v_2[t]-v_1[t]}{R} - \left(\left(-0.667 * 10^{-3} + 0.029 * 10^{-3} * \phi[t]^2 * 3\right) * v_1[t]\right)\right), \\
&v_2'[t] == \frac{1}{C_2}\left(\frac{v_1[t]-v_2[t]}{R} - i[t]\right), i'[t] == \frac{v_2[t]}{L}, \phi[0] == 0, \\
&v[1] = 0.11, v_2[0] = 0.11, i[0] == 0.0\}/.\{C_1 \to C1val, C_2 \to C2val, L \to Lval, \\
&R \to Rval\}, \{\phi, v_2, v_2, i\}, \{t, 0, tmax\}, MaxSteps \to Infinity]][[1]]\}, \\
&Grid[\{\{ParametricPlot[\{First[sol][t], First[Rest[Rest[sol]]][t]\}, \{t, 0, tmax\}, \\
&AxesLabel \to \{\text{“}\phi[t]\text{”}, \text{“}v_2[t]\text{”}\}, ImageSize \to \{150, 150\}, AspectRatio \to 1], \\
&PlotRange \to All, MaxRecursion \to 8, PerformanceGoal \to 8], \\
&Plot[\{First[sol][t], \{t, 0, tmax\}, AxesLabel \to \{\text{“}t\text{”}, \text{“}\phi[t]\text{”}\}, \\
&ImageSize \to \{150, 150\}, PlotRange \to All, MaxRecursion \to 8, \\
&PerformanceGoal \to 8], ParametricPlot3D[\{First[Rest[sol]][t], \\
&First[Rest[Rest[sol]]][t], First[sol][t]\}, \{t, 0, tmax\}, AxesLabel \to \{\text{“}v_1[t]\text{”}, \text{“}v_2[t]\text{”}, \\
&\text{“}\phi[t]\text{”}\}, ImageSize \to \{150, 150\}, BoxRatios \to \{1, 1, 1\}, PlotRange \to Full, \\
&MaxRecursion \to 8, PerformanceGoal \to 8, ColorFunction \to \text{“}Rainbow\text{”}]\}\}], \\
&Text[Style[\text{“}CircuitParameters\text{”}, FontSize \to Medium, FontWeight \to Bold]], \\
&\{\{Rval, 2.0 * 10^3, \text{“}R \in \{1800\,\Omega, 1801\,\Omega, \ldots, 2400\,\Omega\}\text{”}\}, 1.8 * 10^3, 2.4 * 10^3, 1\}, \\
&\{\{Lval, 18 * 10^{-3}, \text{“}L \in \{18\text{ mH}, 18.5\text{ mH}, \ldots, 24\text{ mH}\}\text{”}\}, 18 * 10^{-3}, 24 * 10^{-3}, 0.5 * 10^{-3}\}, \\
&\{\{C2val, 68 * 10^{-9}, \text{“}C_2 \in \{68\text{ nF}, 68.5\text{ nF}, \ldots, 74\text{ nF}\}\text{”}\}, 68 * 10^{-9}, 74 * 10^{-9}, 0.5 * 10^{-9}\}, \\
&\{\{C1val, 6.8 * 10^{-9}, \text{“}C_1 \in \{6.8\text{ nF}, 6.85\text{ nF}, \ldots, 7.4\text{ nF}\}\text{”}\}, 6.8 * 10^{-9}, 7.4 * 10^{-9}, \\
&0.05 * 10^{-9}\}, \{\{tmax, 0.01, \text{“}t \in \{5 * 10^{-5}, 10e-5, \ldots, 0.01\ sec, \ldots, 0.1\ sec\}\text{”}\}, \\
&5 * 10^{-5}, 0.1, 5 * 10^{-5}\}, Text[Style[\text{”}2D\ Phase\ Portrait,\ Time\ Domain \\
&Waveform\ and\ a\ 3D\ Phase\ Portrait\text{”}, \\
&FontSize \to Medium, Font\,Weight \to Bold]], Synchronous\,Updating \to False] \quad \text{(B.1)}
\end{aligned}
$$

Appendix C

Rescaling Circuit Equations

In this appendix, we will derive Eq. (7) from Eq. (3). First, consider a simulation of our system *without* rescaling using the MATLAB code in the next appendix. A plot of all the attractors from this simulation is in Fig. 11, from which we can see that we need to rescale the current and voltage variables from Eq. (3)

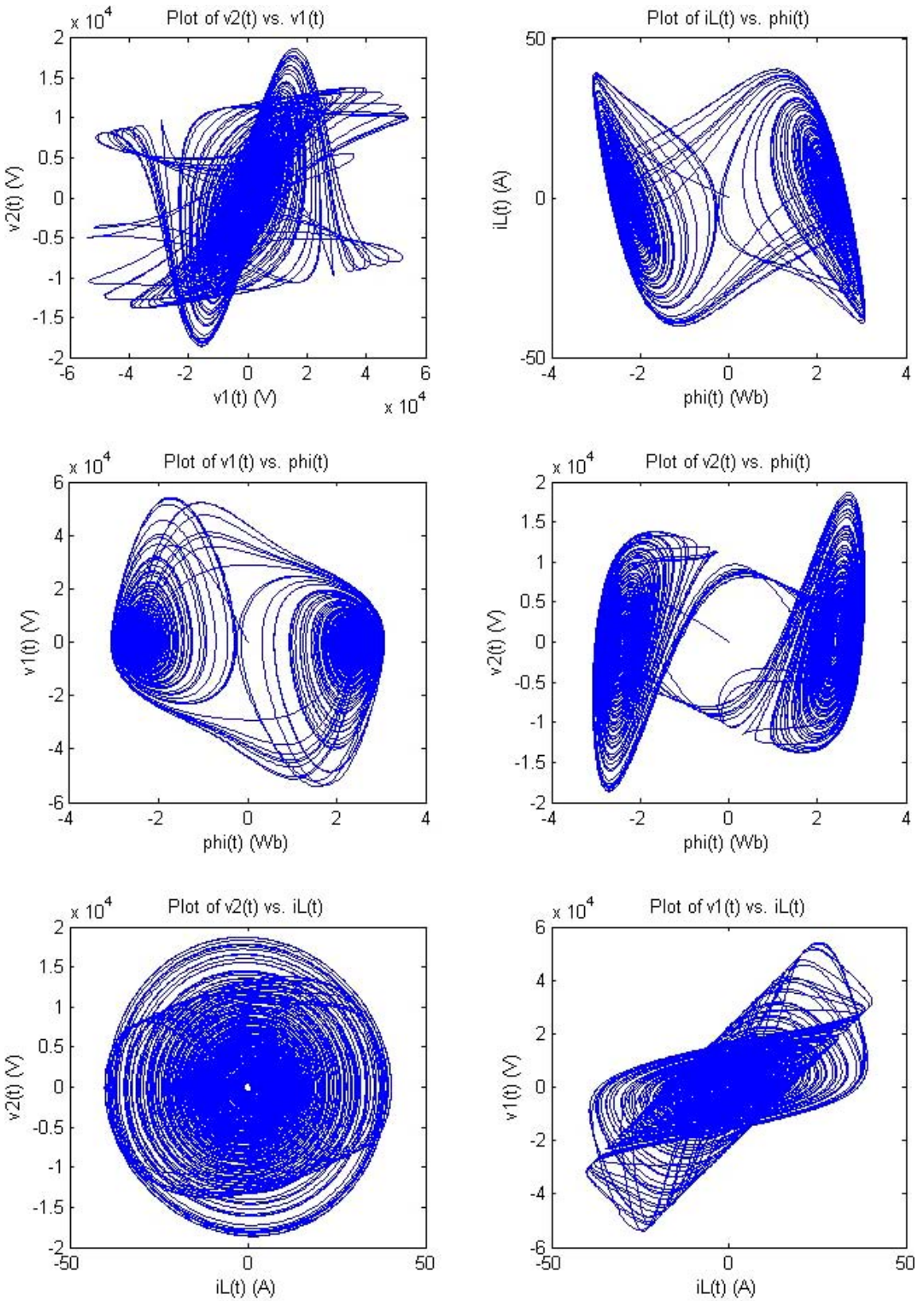


Fig. 11. Simulation results from MATLAB, the nonrescaled version.

as follows. We will leave the flux variable intact:

$$\begin{aligned} w &\triangleq \phi \\ x &\triangleq \frac{v_1}{\delta} \\ y &\triangleq \frac{v_2}{\delta} \\ z &\triangleq \frac{i_L}{\delta} \end{aligned} \tag{C.1}$$

Using the definitions above in Eq. (3) and simplifying, we will get:

$$\begin{aligned} \frac{dw}{dt} &= \delta x \\ \frac{dx}{dt} &= \frac{1}{C_1}\left(\frac{y-x}{R} - W(w)\cdot x\right) \\ \frac{dy}{dt} &= \frac{1}{C_2}\left(\frac{x-y}{R} - z\right) \\ \frac{dz}{dt} &= \frac{y}{L} \end{aligned} \tag{C.2}$$

Using $\delta \triangleq -1/\zeta$ in the equation above, we will get the desired result:

$$\begin{aligned} \frac{dw}{dt} &= \frac{-x}{\zeta} \\ \frac{dx}{dt} &= \frac{1}{C_1}\left(\frac{y-x}{R} - W(w)\cdot x\right) \\ \frac{dy}{dt} &= \frac{1}{C_2}\left(\frac{x-y}{R} - z\right) \\ \frac{dz}{dt} &= \frac{y}{L} \end{aligned} \tag{C.3}$$

Appendix D

MATLAB Simulation Code

MATLAB simulation code for Fig. 4

```
clear;
close all;

% **Make sure the dmemristor equality
    below is on ONE line!
% Cannot parametrize equality because
  we are using inline keyword
dmemristor = inline('[-(1/(8200*47e-9))
   *y(2);
(((y(3)-y(2))/2000)-quadraticW(y)*y(2))/
   6.8e-9;
((y(2)-y(3))/2000 - y(4))/68e-9;y(3)/
   18e-3]', 't','y');
options = odeset('RelTol',1e-7,'AbsTol',
          1e-7);

% **Make sure the declaration below is
    on ONE line!
[t,ya] = ode45(dmemristor,[0 50e-3],
[0,0.11,0.11,0],options);

subplot(3,2,1);
plot(ya(:,2),ya(:,3));
xlabel('v1(t) (V)');
ylabel('v2(t) (V)');
title('Plot of v2(t) vs. v1(t)');

subplot(3,2,2);
plot(ya(:,1),ya(:,4));
xlabel('phi(t) (Wb)');
ylabel('iL(t) (A)');
title('Plot of iL(t) vs. phi(t)');

subplot(3,2,3);
plot(ya(:,1),ya(:,2));
xlabel('phi(t) (Wb)');
ylabel('v1(t) (V)');
title('Plot of v1(t) vs. phi(t)');

subplot(3,2,4);
plot(ya(:,1),ya(:,3));
xlabel('phi(t) (Wb)');
ylabel('v2(t) (V)');
title('Plot of v2(t) vs. phi(t)');

subplot(3,2,5);
plot(ya(:,4),ya(:,3));
xlabel('iL(t) (A)');
ylabel('v2(t) (V)');
title('Plot of v2(t) vs. iL(t)');

subplot(3,2,6);
plot(ya(:,4),ya(:,2));
xlabel('iL(t) (A)');
ylabel('v1(t) (V)');
title('Plot of v1(t) vs. iL(t)');
```

The quadraticW.m file is shown below.

```
function r = quadraticW(y)
    r = -0.667e-3 + 0.029e-3*y(1).*y(1)*3;
end
```

Appendix E

Lyapunov Exponent Code for Time Series Method

This appendix gives the code necessary for computing Lyapunov exponents for our circuit based on the time series method [Wolf *et al.*, 1985]. The QR method [Eckmann & Ruelle, 1985] code [Siu, 2008] works similarly. Note that Lyapunov Exponents Toolbox from MATDS [Govorukhin, 2008] was used for computing the exponents. Also we have assumed $\zeta = -1$ for simplicity and time-scaled the ODEs for numerical stability.

```
function OUT = cubicMemristor(t,X)
%Memristor based chaotic circuit -
  implementation model

% Settings:
% ODEFUNCTION: cubicMemristor
% Final Time:  1000, Step: 0.01,
% update Lyapunov: 10
% Initial Conditions: 0 0.11 0.11 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 = 18e-3;C1=6.8e-9;C2=68e-9;R=2000;
tau = sqrt(L1*C2);

% ODE
dw = tau*x;
dx = (tau/C1)*(((y-x)/R)-quadraticW(w)*x);
dy = (tau/C2)*(((x-y)/R)-z);
dz = (tau*y)/L1;

% 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) about
  equilibrium state: (0,0,0,0)
 J = [ 0 tau 0 0;
       0 tau*(-1/(R*C1)-(quadraticW(w)/C1))
         tau/(R*C1) 0;
       0 tau/(R*C2) -tau/(R*C2) -tau/C2;
        0 0 tau/L1 0];

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

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

The quadraticW.m is the same as the previous appendix. The run_lyap.m (main script in MATLAB) is shown below.

```
[T,Res]=lyapunov(4,@cubicMemristor,@ode45,
        0,0.01,1000,[0 0.11 0.11 0],10);
plot(T,Res);
title('Dynamics of Lyapunov exponents');
xlabel('Time');
ylabel('Lyapunov exponents');
```

Appendix F

Synthesis of the Memristor Based Chaotic Circuit

In this appendix, we will derive Eq. (9). Specifically, we will concentrate on justifying that our analog circuit realizes a memristor, since deriving the nodal equations at nodes v_2 and v_1 in Fig. 6 have already been done in Appendix A. Consider op-amps $U1$ and U1_1 in Fig. 6. Since $U1$ is an integtrator op-amp, we have the first relationship from Eq. (9):

$$\boxed{\frac{dv_{15}}{dt} = \frac{-v_1(t)}{R6 \cdot C3}} \tag{F.1}$$

Now, we will derive the $i_m - v_1$ relationship in Fig. 6. The basic circuit to realize the current expression is the multiplier circuit in a feedback loop [Zhong, 1994]. A block diagram view of this circuit (along with the integrator) is shown in Fig. 12 (compare to Fig. 6).

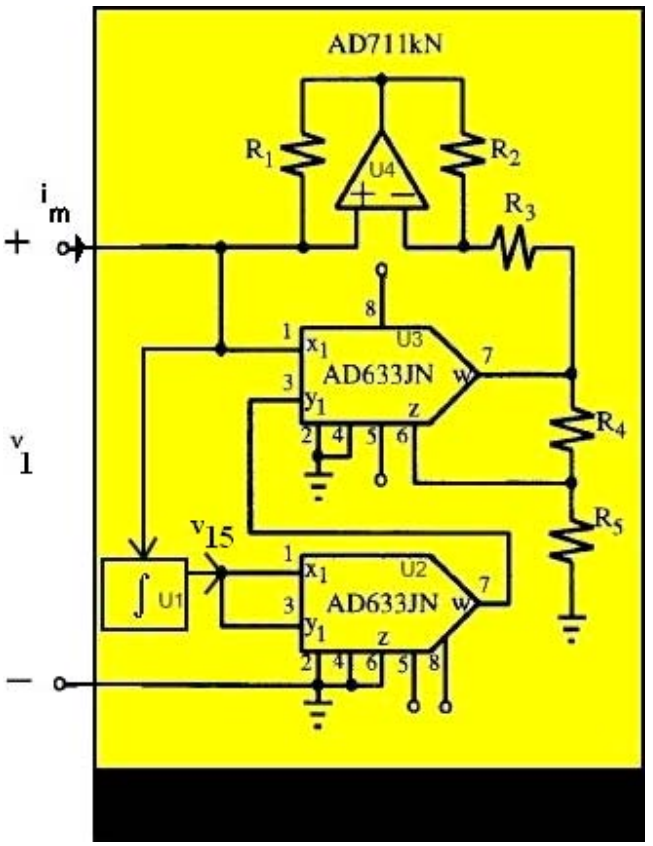


Fig. 12. Practical circuit for realizing a memristor. We have deliberately left out the power supplies to avoid clutter.

Based on Fig. 12, we can see that multiplier $U2$ implements:

$$v_{U2}(t) = \frac{v_{15}^2}{10} \tag{F.2}$$

The factor of 10 is inherent to the AD633, refer to its datasheet for further information. Multipler $U3$ in Fig. 12 implements [Zhong, 1994]:

$$v_{U3}(t) = v_{U2} \cdot v_1(t) \frac{(R_4 + R_5)}{10R_4} \tag{F.3}$$

Op-amp $U4$ in Fig. 12 is the current–inverter that implements [Zhong, 1994] (when $R_1 = R_2$):

$$i_m(t) = \frac{-v_1(t)}{R_3} + \frac{v_{U3}(t)}{R_3} \tag{F.4}$$

Substituting for $v_{U2}(t)$ from Eq. (F.2) into Eq. (F.3) and then substituting the result into Eq. (F.4) we get after simplification:

$$\boxed{i_m(t) = \left(\frac{-1}{R_3} + v_{15}^2 \cdot \frac{(R_4 + R_5)}{R_3 \cdot R_4 \cdot 100} \right) \cdot v_1(t)} \tag{F.5}$$

Now, consider our $W(\phi)$ obtained from $q(\phi)$ in Eq. (5):

$$W(\phi) = \alpha + \phi^2 \cdot (3\beta) \tag{F.6}$$

Using the definitions of the flux across and current through a flux-controlled memristor along with the required memductance function above, we have:

$$\begin{aligned} \frac{d\phi}{dt} &= v(t) \\ i(t) &= (\alpha + \phi^2 \cdot (3\beta)) \cdot v(t) \end{aligned} \tag{F.7}$$

Comparing Eqs. (F.1), (F.5) and (F.7) we see that our circuit does indeed realize a memristor. As stated earlier, we have synthesized a general memristive system.

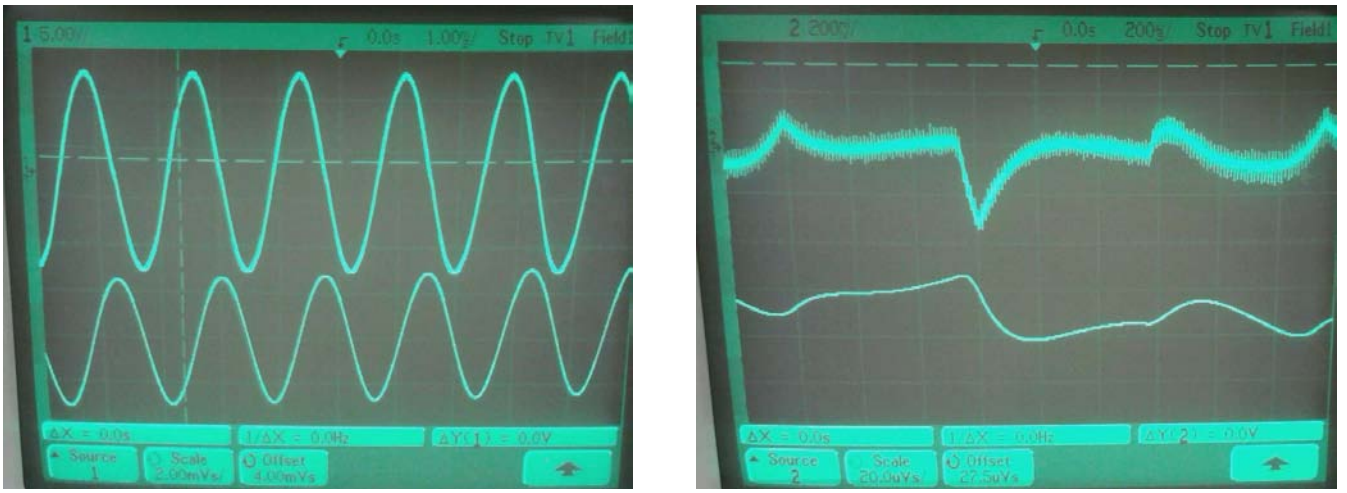
Comparing Eq. (F.5) and the second equality in Eq. (F.7) we have the following relationships for α and β in terms of the component values:

$$\begin{aligned} \alpha &= \frac{-1}{R_3} \\ \beta &= \frac{1}{3} \left(\frac{R_4 + R_5}{R_3 \cdot R_4 \cdot 100} \right) \end{aligned} \tag{F.8}$$

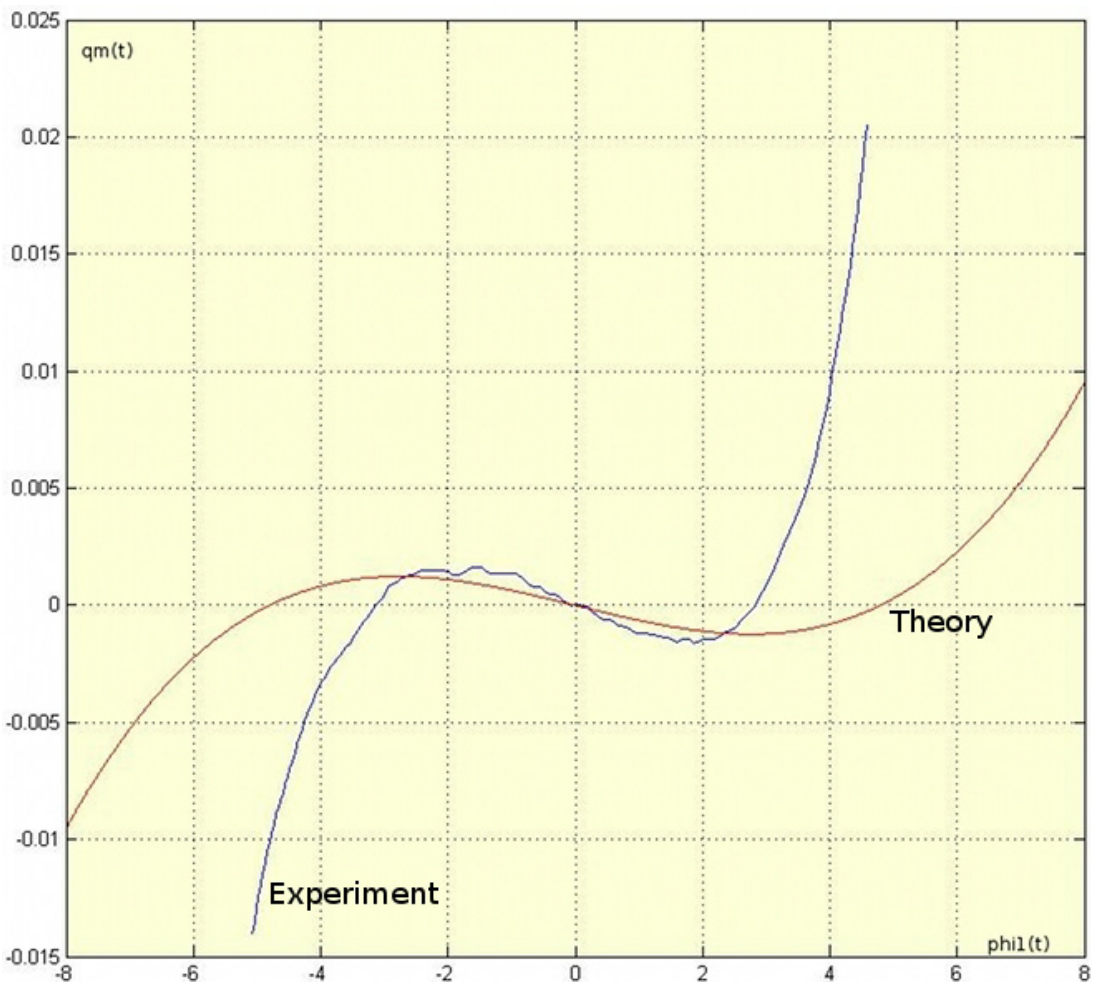
Recall that we chose $\alpha = -0.667 \cdot 10^{-3}$ and $\beta = 0.029 \cdot 10^{-3}$. Using these parameters, we get the following values for the components in Fig. 12: $R_3 = 1.5\,\text{k}$, $R_4 = 3\,\text{k}$, $R_5 = 36\,\text{k}$. Notice that we set $R_4 = 3\,\text{k}$ in Eq. (F.8) and then compute the other values. Also for the multiplier $R_4 > 1\,\text{k}$ and $R_5 < 100\,\text{k}$ (refer to the AD633 datasheet). The power supplies for all the ICs are $+15\,\text{V}$ (multiplier: pin eight, op-amp: pin seven) and $-15\,\text{V}$ (multiplier: pin five, op-amp: pin four).

To emphasize that we have a real memristor, we experimentally obtained the $i_m - v_1$ curve and hence the resulting $q_m - \phi_1$ curve for our memristor in Fig. 12. In order to obtain the memristor $q_m - \phi_1$ curve, we first applied a sinusoidal voltage of $9\,\text{V}$ amplitude with an offset of $0.15\,\text{V}$ at a frequency of $570\,\text{Hz}$ for $v_1(t)$ in Fig. 12.

Figure 13 illustrates the $q_m - \phi_1$ curves that we obtained. Using the integrator functionality of the oscilloscope, we obtained $\phi_1(t)$, the result is plotted in Fig. 13(a) ($v_1(t)$ is the top trace, $\phi_1(t)$ is the bottom trace). We measured the corresponding $i_m(t)$ using a current probe connected to the scope (top trace in Fig. 13(b)) and then integrated to obtain $q_m(t)$ (bottom trace in Fig. 13(b)). Notice the noisy nature of the current. The integration subsequently removes the high-frequency components.



(a) (b)



(c)

Fig. 13. Various memristor characteristic curves. The top plots are the time-domain waveforms of $v_1(t)$, $\phi_1(t)$ (on the left, starting at the top trace) and $i_m(t)$, $q_m(t)$ (on the right, starting at the top trace). The X-axis scale for the top-left plot is 1 ms/div, the Y-axis scale is 5.00 V/div. The X-axis scale for the top-right plot is 0.2 ms/div, the Y-axis scale is 0.2 V/div. The plot on the bottom shows the theoretical (red) and the measured (blue) $q_m - \phi_1$ curves.

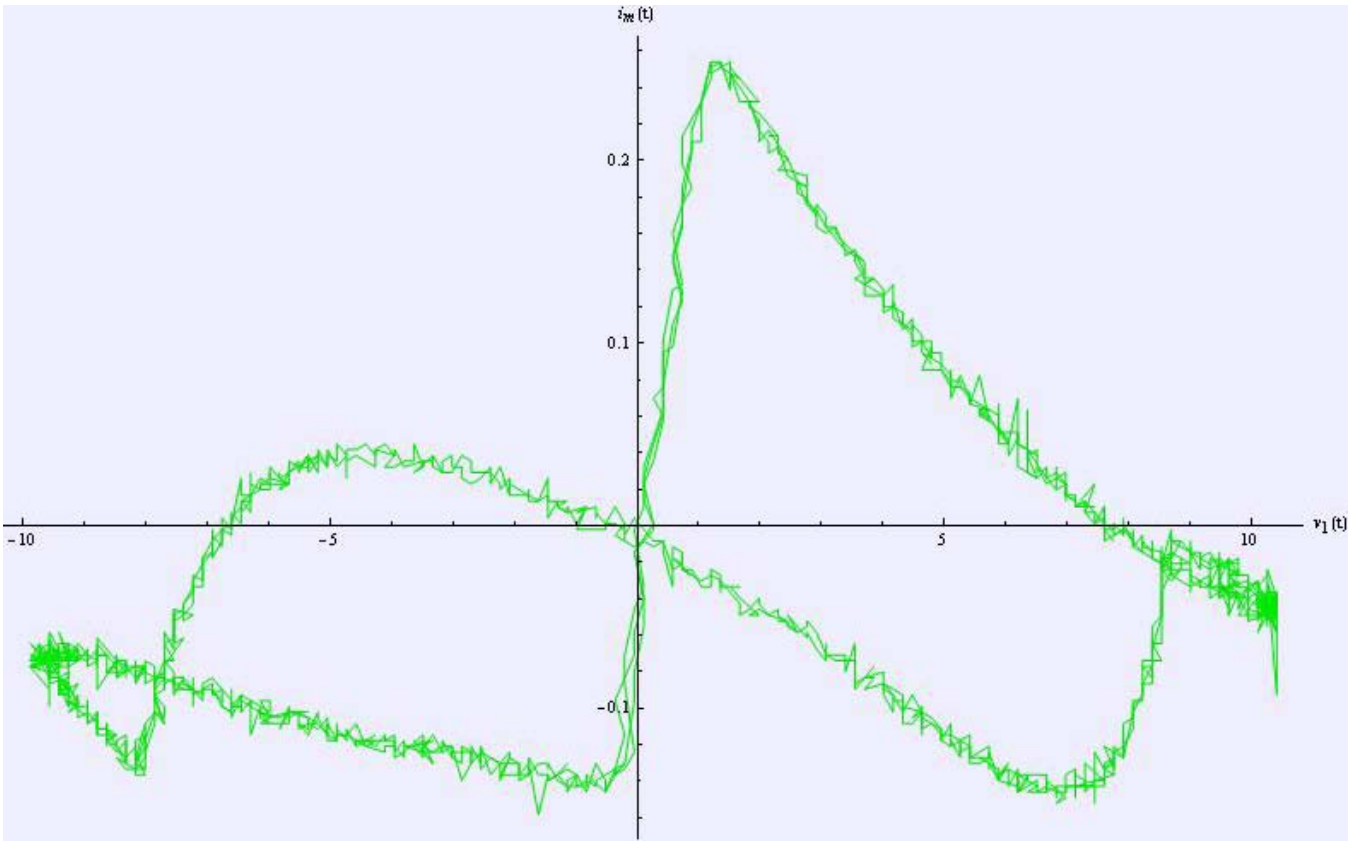


Fig. 14. The memristor $i_m - v_1$ graph. The noisy curves are due to the limited resolution of our data acquisition system. We also had to apply a slight offset to our curve to compensate for the physical integrator offsets. Notice the curve passes through the origin and we do have a pinched hysteresis loop.

Nevertheless, since the oscilloscope cannot plot integration data in XY mode, we had to use the computer to plot the $q_m - \phi_1$ curve. This is the last plot in Fig. 13. In order to obtain this plot we sampled the $i_m - v_1$ data into MATLAB since the oscilloscope's firmware would not allow us to sample the integrated data. We then used numerical trapezoidal integration to obtain $q_m - \phi_1$. We also had to offset the data to compensate for integrator offsets in the physical memristor and scale the data appropriately. Notice the close agreement between the theoretical (red) and experimental (blue) $q_m - \phi_1$ curves in the negative memductance region. Most importantly, the experimental curve also passes through the origin satisfying the generic properties specified in [Chua & Kang, 1976]. However there is marked difference between the theoretical and experimental curve in the positive memductance regions, this difference could be the study of future work.

Figure 14 shows the memristor i_m–v_1 curve. We applied a 10 V sinusoid with no offset voltage and a frequency of 570 Hz. Notice the pinched hysteresis loop, the fingerprint of a memristor [Strukov *et al.*, 2008; Chua & Kang, 1976].