Two Element Chaotic and Hyperchaotic Circuits

Bharathwaj Muthuswamy, Andrew Przybylski, Chris Feilbach, and Joerg Mossbrucker

Abstract The goals of this work are twofold: one is to illustrate the use of Field Programmable Gate Arrays (FPGAs) for emulating circuit elements with memory (memristors, memcapacitors and meminductors). The second goal is to use the FPGA emulation to realize two element chaotic and hyperchaotic circuits. Such circuits utilize fully nonlinear models of memory devices in series-parallel configuration.

1 Introduction

The memristor was postulated as the fourth fundamental 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 elements together as the four basic elements of circuit theory is the fact that the characteristics of these elements relate the four fundamental circuit variables (voltage, current, flux-linkage and charge).

For over 30 years, the memristor was not significant in circuit theory. However in 2008, Strukov et al. [12] from Hewlett-Packard labs announced that they had fabricated a solid state implementation of the memristor. Ever since their announcement, a variety of circuit applications of memristors have been developed; refer to [6] for examples and further references.

Chua and Kang [2] first extended the notion of the memristor to a general class of memristive systems. DiVentra et al. [4] incorporated capacitors and inductors into the notion of memory devices, as shown in Fig. 1. This "memory element"

B. Muthuswamy (✉) · A. Przybylski · C. Feilbach · J. Mossbrucker
Department of Electrical Engineering and Computer Science, Milwaukee School of Engineering, WI 53202, USA
e-mail: muthuswamy@msoe.edu; przybylskia@msoe.edu; feilbachc@msoe.edu; mossbruc@msoe.edu

S.G. Stavrinides et al. (eds.), *Chaos and Complex Systems*,
DOI 10.1007/978-3-642-33914-1_1,

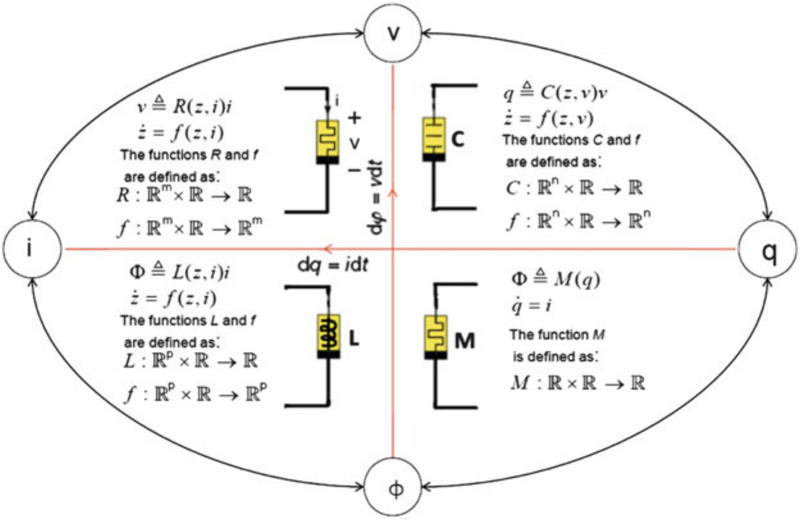

Fig. 1 We can generalize the four basic circuit elements to elements with memory. A resistor is a device that establishes a relation between current (i) and voltage (v), a memristor is a device whose resistance depends on a state variable (z). Similarly memcapacitors establish a memory relationship between charge (q) and voltage; meminductors establish a relationship between flux-linkage (ϕ) and current. Note that an ideal memristor is a special case of a general memristive system—the ideal memristor's internal state is simply the charge flowing through or the flux-linkage across the device

abstraction enables us to model nanoscale systems [7] where the dynamical properties of electrons and ions strongly depend on the history of the system.

In this work, we are concerned with utilizing memory elements to design two element chaotic and hyperchaotic circuits. We picked two elements because such circuits have been shown to correspond to models of physical systems [10].

Since memory elements are commercially unavailable as of this writing, we also show how one can use FPGAs to emulate such elements. Although microcontrollers have been used to emulate memory elements [8], FPGAs are an inherently parallel architecture and hence enable us to emulate memory devices at much higher frequencies than a sequential microcontroller. For instance, our FPGA emulator functions up to frequencies of 6 KHz. The micro controller emulation functions at a maximum frequency of 50 Hz [8]. Although an FPGA has been used to implement memristive chaotic circuits [9], our work is the first to emulate circuits with memcapacitors and meminductors.

This work is organized as follows: in Sect. 2 we review the principal ideas behind memory elements. In Sect. 3 we give an example of a two element chaotic circuit and in Sect. 4 we give an example of the two element hyperchaotic circuit. Section 5 discuss the FPGA physical emulation platform. Finally, we conclude with suggestions for future work.

In lieu of page constraints, we have focused on the main ideas in this work. For details regarding physical realization of memory elements using FPGAs (including the design source), please email the first author.

2 An Overview of Circuit Elements with Memory

Based on Fig. 1, we can generally define $z(t)$ to be a set of state variables describing the internal state of a system [2]. Let $x(t)$ and $y(t)$ be any two complementary constitutive variables (i.e., current, charge, voltage, or flux-linkage) denoting input and output of the system and g be a generalized response [4]. We can now define a general class of nth-order x-controlled memory devices as those described by Eqs. (1) and (2).

$$y(t) = g(z, x, t)x(t) \tag{1}$$

$$\dot{z} = f(z, x, t) \tag{2}$$

In Fig. 1, g is either $R(z, i)$ (memristance), $C(z, v)$ (memcapacitance), $L(z, i)$ (memductance) or $M(q)(q \equiv z$, memristance).

Practically speaking, different memory effects (namely, memristive, memcapacitive and/or meminductive features) could coexist in physical devices [10]. Hence we will utilize the fully nonlinear models of the memristor (Definition (1)), memcapacitor (Definition (2)) and meminductor (Definition (3)) from Riazza [10].

Definition 1. A fully nonlinear current-controller memristive system[1] is a device governed by the relations

$$v = \eta(z, i, t) \tag{3}$$

$$\dot{z} = f(z, i) \tag{4}$$

Systems in which the characteristics in Eq. (3) amount to $v = M(z, i)i$ describe the settings originally discussed by Chua and Kang [2]. Note also that the fully nonlinear form in Eqs. (3) and (4) makes it possible to accommodate physical devices that display memristive effects but whose characteristic does not arise as the time derivative of a $\phi - q$ relation, contrary to Chua's memristor [10].

Definition 2. A fully nonlinear voltage-controlled memcapacitor is governed by the relations

$$q = \omega(z, v, t) \tag{5}$$

$$\dot{z} = f(z, v) \tag{6}$$

[1] A voltage-controlled fully nonlinear memristive system is analogously defined.

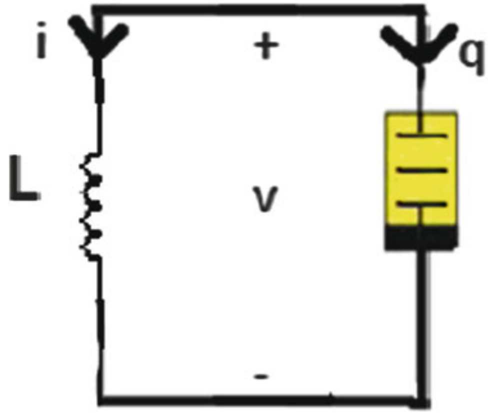

Fig. 2 An inductor in parallel with a fully nonlinear memcapacitor. The inductor provides one state variable whereas the memcapacitor provides two state variables. Hence we get a total of three state variables, the minimum required for chaotic behavior in a continuous time smooth dynamical system

Definition 3. A fully nonlinear current-controlled meminductor is governed by the relations

$$\phi = \theta(z, i, t) \tag{7}$$

$$\dot{z} = f(z, i) \tag{8}$$

Again, with $\omega(z, v, t) = C(z, v)v$ and $\theta(z, i, t) = L(z, i)i$, we get the systems proposed by Di Ventra et al. [4]. A physical instance of a fully nonlinear memcapacitor arising in a Josephson junction model is discussed in [10].

Now we will utilize these fully nonlinear devices to design chaotic and hyper chaotic circuits that utilize two elements in series-parallel configuration. This is possible because the internal state of a memory device need not be simply charge or flux-linkage. Rather, as mentioned earlier, it could be a n-dimensional set.

3 Two Element Chaotic Circuit

Consider the schematic shown in Fig. 2. This circuit is a modified version of the three element chaotic circuit from [5]. The memory device could be implemented on a FPGA and the inductor could be an external physical device. The interface to the FPGA is discussed in Sect. 5. Equations (9) through (11) describe the circuit. We have used x_C to denote the internal state of the memcapacitor.

$$\dot{x_C} = -v - \alpha x_C + v x_C \tag{9}$$

$$\dot{v} = \frac{-1}{C}\left(i + \beta(x_C^2 - 1)v\right) \tag{10}$$

$$\dot{i} = \frac{1}{L}v \tag{11}$$

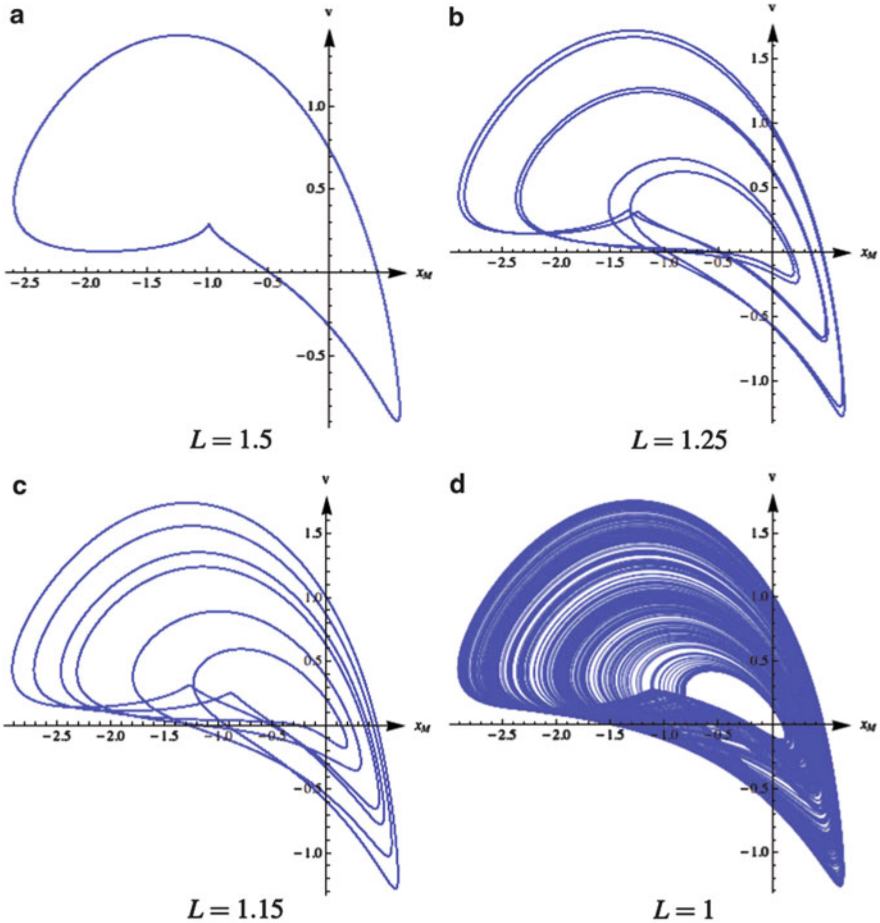

Fig. 3 Period-doubling route to chaos in the two element chaotic circuit. Note that we are using a dimensionless formulation of the equations, hence units are not specified. Parameter values are $\alpha = 0.6, \beta = 1.5, C = 3$. L is the bifurcation parameter. Initial conditions are $(0.1, 0, 0.1)$. The system is simulated for 10,000 time units, plot is only from 5,000 to 10,000 time units to minimize transient effects

The fully nonlinear memcapacitor is described by Eqs. (12) and (13).

$$q = Cv + \beta\left(\frac{x_C^3}{2} - x_C\right) \tag{12}$$

$$\dot{x_C} = -v - \alpha x_C + v x_C \tag{13}$$

Figure 3 shows a simulation of the period-doubling route to chaos. Mathematica simulation code is in appendix.

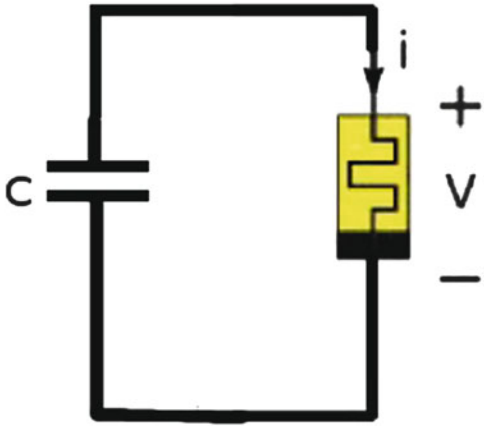

Fig. 4 A capacitor in parallel with a memristive device. This circuit models a higher order Rössler system [11]

The numerically calculated Lyapunov exponents are (0.029,0,–0.47) [5]. Note how we have one positive Lyapunov exponent and the sum of the Lyapunov exponents is negative. This indicates steady-state chaotic behavior.

4 Two Element Hyperchaotic Circuit

By simply increasing the number of internal state variables in the memory device, one could get hyperchaotic behavior. Consider the schematic shown in Fig. 4. Equations (14) through (17) describe the circuit.

$$\dot{x} = -y - z \tag{14}$$

$$\dot{y} = z + ay + v \tag{15}$$

$$\dot{z} = b + xz \tag{16}$$

$$\dot{v} = \frac{1}{C}(dv - ez) \tag{17}$$

The memristive system is described by Eqs. (18) through (21).

$$\dot{x} = -y - z \tag{18}$$

$$\dot{y} = z + ay + v \tag{19}$$

$$\dot{z} = b + xz \tag{20}$$

$$i = dv - ex \tag{21}$$

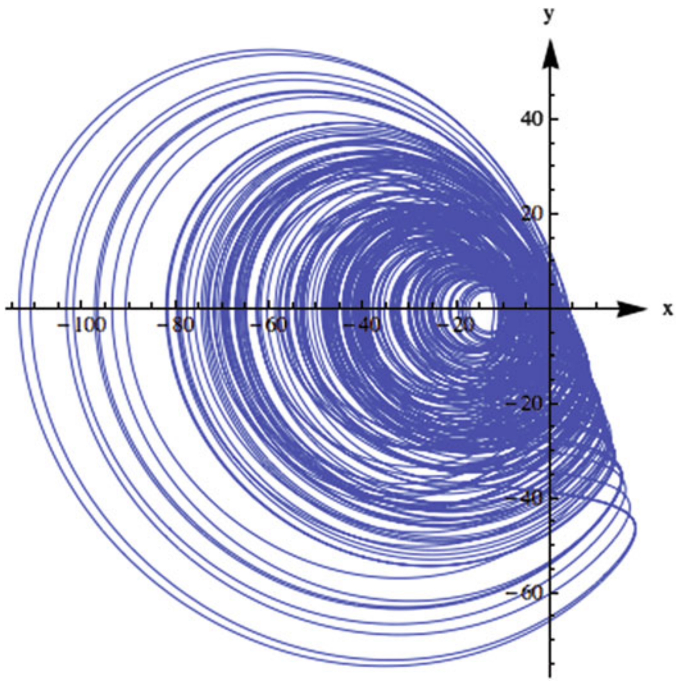

Fig. 5 The hyperchaotic attractor. Parameter values are $a = 0.25, b = 3, C = 1, d = 0.05, e = 0.5$. Initial conditions are (−6,0,0.5,14) [11]. Simulation is done for 10,000 time units, plot is only from 9,000 to 10,000 time units to minimize errors due to transient response

Figure 5 shows a phase plot from the simulation of the two element hyperchaotic circuit. We have omitted the simulation code and any bifurcation phenomenon since they are very similar to the two element chaotic circuit.

Lyapunov exponents are (0.1120, 0.0211,0,−24.9312) [11]. We now have two positive Lyapunov exponents but sum of the exponents is still negative. The two positive exponents indicate that stretching dynamics occur in two directions [11], indicating the presence of hyperchaos.

5 FPGA Based Physical Emulation Platform

Figure 6 shows the period-doubling route to chaos in a physical realization of the two element circuit. We are currently working on the physical realization of the hyperchaotic circuit.

The FPGA implements implements the nonlinear relations in Eqs. (3) through (8). Figure 6 shows the physical realization and the various analog sub-components that sample either the voltage or current into the FPGA. We utilize the circuit in

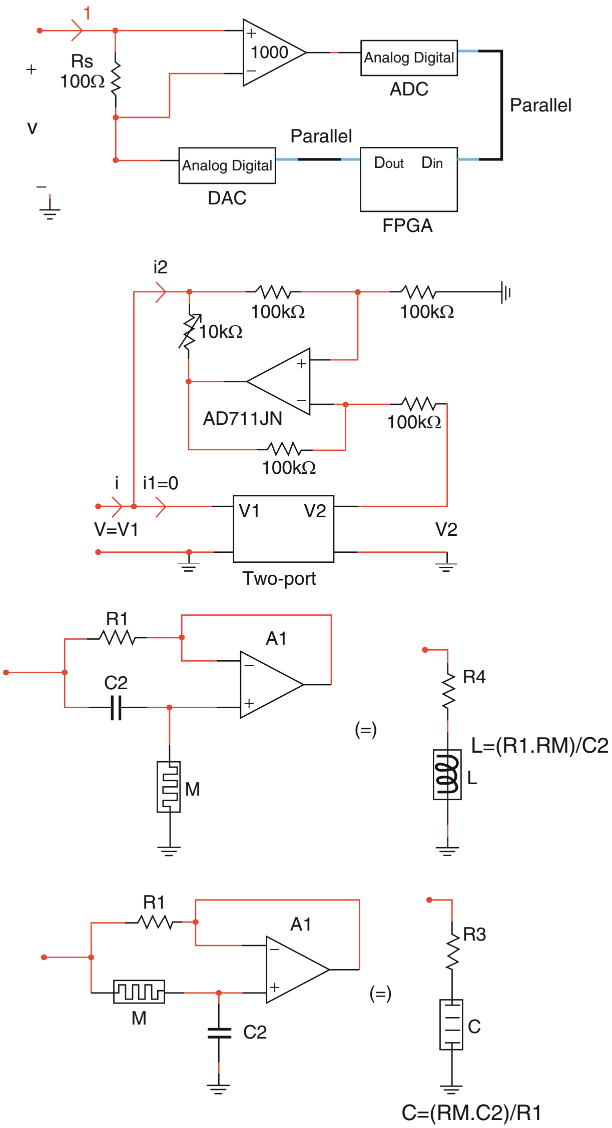

Fig. 6 The various components of the FPGA based memory devices emulator. The FPGA used was a Cyclone II, the development platform is a DE2 board from Terasic. All amplifiers are the AD711. The analog-to-digital converter (ADC) is the ADS7804 and the digital-to-analog converter (DAC) is the AD7245. 27-bit fixed point was used to emulate the memory devices on the FPGA. We used 27-bits because the Cyclone II FPGA has embedded 9-bit multipliers and hence the synthesis tool will be able to utilize the hardware multipliers instead of using FPGA logic elements for multiplication

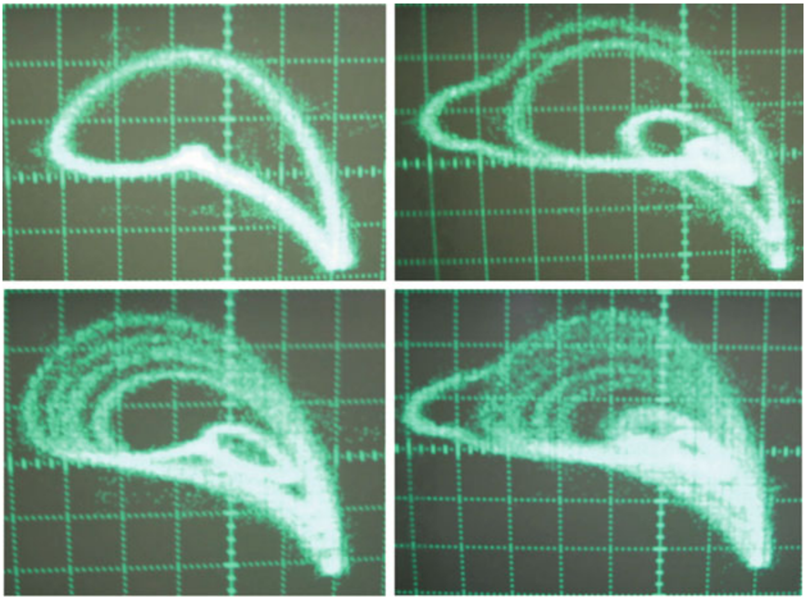

Fig. 7 Period-doubling route to chaos from the physical emulator

[3] along with the memristor to memcapacitor or meminductor converter proposed in [4]. Note the differential equations implemented on the FPGA may have to be amplitude scaled to match the analog subsystems.

Note that compared to a microcontroller emulation [8], the FPGA is not easy to "program". However the difficulty in using the FPGA is offset by the advantage gained in operational frequency of the memory device and the arbitrary size of the datapath.

6 Conclusions and Future Work

In this work, we have proposed the use of a mixed analog-digital emulation of memory devices. Currently, we are working on realizing the memory device characteristics for hyperchaos. We are also developing a robust user-interface for ease of FPGA development (Figs. 7 and 8).

We are also migrating our design to use sigma-delta and delta-sigma converters, instead of the current parallel ADCs and DACs. The reasoning is simplicity of interface. Nevertheless, sigma-delta and delta-sigma converters process one-bit at a time and hence the operational frequency of the circuit is reduced.

```
twoElementChaos =
 NDSolve[{i'[t] == v[t]/L, v'[t] == -1/C * (i[t] + β * (xM[t]^2 - 1) * v[t]), xM'[t] == -v[t] - α * xM[t] + v[t] * xM[t],
    i[0] == 0.1, v[0] == 0, xM[0] == 0.1} /. {L → 1, C → 3, β → 1.5, α → 0.6}, {i, v, xM},
   {t, 0, 10000}, MaxSteps → Infinity]

ParametricPlot[Evaluate[{xM[t], v[t]} /. twoElementChaos], {t, 5000, 10000},
 AxesLabel → {"x_M", "v"}, LabelStyle → Directive[Medium], AspectRatio → 1,
 AxesStyle → Arrowheads[0.05], PlotRange → All, PlotPoints → 10000]
```

Fig. 8 Mathematica 8 simulation code. NDSolve command simulates the differential equations and ParametricPlot command displays the phase plot

Acknowledgements Many thanks to Ferenc Kovac from the Electronic Support Group at the University of California, Berkeley for generous donation of hardware. Faculty travel grant from the Milwaukee School of Engineering enabled us to present our work at the fourth international chaos and complex systems symposium.

Appendix

Mathematica simulation code for two element chaotic circuit. Simulation parameters are for chaotic behavior.

References

1. Chua, L.O.: Memristor - the missing circuit element. IEEE Trans. Circ. Theor. **CAT-18**(5), 507–519 (1971)
2. Chua, L.O., Kang, S.M.: Memristive devices and systems. Proc. IEEE **64**(2), 209–223 (1976)
3. Chua, L.O., et al.: Synthesizing arbitrary driving-point and transfer characteristics. IEEE Trans. Circ. Syst. I: Fund. Theor. Appl. **45**(12), 1225–1232 (1998)
4. Di Ventra, M., Pershin, Y.V., Chua, L.O.: Putting memory into circuit elements: memristors, memcapacitors, and meminductors. Proc. IEEE **97**(8), 1371–1372 (2009)
5. Muthuswamy, B., Chua, L.O.: Simplest chaotic circuit. Int. J. Bifurcat. Chaos **20**(5), 1567–1580 (2010)
6. Pershin, Y.V., Di Ventra, M.: Experimental demonstration of associative memory with memristive neural networks. http://www.arxiv.org, arXiv:0905.2935 (2009)
7. Pershin, Y.V., Di Ventra, M.: Memory effects in complex materials and nanoscale systems. Adv. Phys. **60**(2), 145–227 (2011)
8. Pershin, Y.V., Di Ventra, M.: Practical approach to programmable analog circuits with memristors. arXiv:0908.3162 (2009). Available via DIALOG. http://www.arxiv.org. Cited 6 June 2012
9. Qi, A., et al.: Memristor oscillators and its FPGA implementation. Adv. Mater. Res. **383–390**, 6992–6997 (2011)
10. Riaza, R.: First order devices, hybrid memristors, and the frontiers of nonlinear circuit theory. arXiv:1010.0123v1 (2010). Available via DIALOG. http://www.arxiv.org. Cited 6 June 2012
11. Sprott, J.C.: Elegant Chaos, 1st edn. World Scientific, Singapore (2011)
12. Strukov, D.B., et al.: The missing memristor found. Nature **453**, 80–83 (2008)