Fractional Order Simple Chaotic Oscillator with Saturable Reactors and Its Engineering Applications

Karthikeyan Rajagopal
Center for Nonlinear Dynamics, Defence University, Ethiopia;
Department of Electronics Engineering, Durban University of Tech., South Africa,
Department of Electronics Engineering, Chennai Institute of Tech., India

Sajad Jafari
Biomedical Engineering Department, Amirkabir University of Technology, Tehran, Iran

Sezgin Kacar
Department of Electrical and Electronics Engineering, Faculty of Technology, Sakarya University of Applied Sciences, 54050 Serdivan, Sakarya, Turkey

Anitha Karthikeyan
Center for Nonlinear Dynamics, Defence University, Ethiopia

Akif Akgul
Department of Electrical and Electronics Engineering, Faculty of Technology, Sakarya University of Applied Sciences, 54050 Serdivan, Sakarya, Turkey

Corresponding author: akifakgul@ieee.org, aakgul@sakarya.edu.tr, aakgul@subu.edu.tr

Chaos is often observed in forced nonlinear electrical circuits with saturable reactors. Simplest of such a circuit is presented and investigated in both integer order and non-integer (fractional) order form. For numerical analysis of fractional order system, the Adam-Bashforth-Moulton method is used. Some dynamic properties of the novel system are investigated. For implementing the system in field programmable gate arrays (FPGA), the Adomian decomposition method is used. Power and resource utilization of the fractional order system for implementing in FPGA are presented. Also, in this work, a novel fractional based sound encryption application is implemented as a unique application.

KEYWORDS: Nonlinear systems, chaos, saturable reactors, fractional order systems, FPGA, sound encryption.
1. Introduction

A saturable reactor is a special form of inductor where the ferrite core can be forcefully saturated by a direct electric current. Once saturated, the inductance of the saturable reactor decreases and thus increases the flow of the AC current in the circuit. A simple periodically excited electrical circuit with only one saturated inductor can produce chaotic oscillations. However most of the dynamical analysis of this system are done with a 1-D approximated linear non autonomous equation which seems to be not very accurate today [8]. In addition, the saturated inductors reduce the size of the overall circuit compared to the non-saturated inductors [41]. A resonant circuit with two saturated inductors [40] and the existence of quasi-periodic invariable tori has been discussed in [44]. Bifurcation analysis of the two saturated inductor resonant circuit has been discussed with a three dimensional Duffing type equation in [54] considering the system as a non-autonomous type. Some other well-known simplest chaotic oscillators are also discussed in the literatures like the Chua’s oscillator [21], modified Wien bridge oscillator [27], simple 3D chaotic oscillator [31], Hartley oscillator [45], Vilnius oscillator [43], etc.

Fractional calculus has a history dating back to 17th century but it found its applications in science and engineering research only in the recent years [14, 16, 18, 39]. Many physical systems exhibit fractional order dynamics and thus fractional order control algorithms are achieving the attention of researchers [19, 31, 38, 47].

There are many engineering applications in the literature where chaotic systems have been used. In particular, Random Number Generators (RNGs) and encryption applications are the most common chaos-based studies. In such studies, widely, integer order chaotic systems are used. In addition, the use of fractional order chaotic systems increases the safety of such applications because it creates an extra parameter for each state variable. Because of this reason, fractional order systems are being used in recent years. In this study, a new RNG and sound encryption application based on a fractional order system has been implemented as a unique application.

Although more have been discussed about the two saturable reactor circuit, the dynamic properties of such systems have not been investigated in detail. In addition, the fractional order analysis of these systems reveals some interesting features and hence the dynamical analysis of integer and fractional order two saturable reactor circuit is derived.

2. Saturable Reactor Circuit (SRC)

Most of the chaotic circuits discussed in the literature have used nonlinear resistors but less have been reported about the use of nonlinear inductors or the saturable inductors. There have been chaotic autonomous circuits with Josephson element. The nonlinear inductors don’t have a negative resistance center region like the saturable inductors.

In this section, the nonlinear behavior of the two saturable inductor resonant circuit is investigated. Those systems have been discussed in [40, 54]. Various dynamical properties of them are derived in this part. Figure 1 shows the two saturable reactor circuit with two resistors, a capacitor, a periodic exciter, a DC potential and two saturable inductors.

Using Kirchhoff’s voltage and current laws,

\[
\begin{align*}
N \frac{d}{dt}(\phi_1 + \phi_2) + V_c &= E_c \sin(\omega t) \\
N \frac{d}{dt}(\phi_1 - \phi_2) + R_2 i_z &= E_2 \\
i_c &= C_i \frac{dV_c}{dt}
\end{align*}
\]
where $\phi_1, \phi_2$ are the flux across the saturable inductors $L_1, L_2$, $V_c$ is the voltage across the capacitor, $N$ is the number of turns of the inductor coil, $i_x$ is the current through the resistor $R_x$ and $i_y$ is the current through the capacitor. Using the method of simplification adopted in [54], the non-autonomous dimensionless model of the system (1) is derived as:

$$
\begin{align*}
\dot{x}(t) &= y(t) \\
\dot{y}(t) &= -a_1 y(t) - a_2 x(t)^3 - a_3 x(t) z(t)^2 + a_4 \cos(t) \\
\dot{z}(t) &= -a_5 x(t)^2 z(t) - a_6 z(t)^3 + a_7
\end{align*}
$$

where $a_1 = 0.043$, $a_2 = 0.125$, $a_3 = 0.375$, $a_4 = 0.22$, $a_5 = 0.01875$, $a_6 = 0.00625$, $a_7 = 0.03$ and the system shows chaotic behavior with initial conditions $[1, 1, 1, 1]$. Assuming $w = t$ and introducing a fourth state, the equation (1) can be made autonomous as:

$$
\begin{align*}
\dot{x}(t) &= y(t) \\
\dot{y}(t) &= -a_1 y(t) - a_2 x(t)^3 - a_3 x(t) z(t)^2 + a_4 \cos(w) \\
\dot{z}(t) &= -a_5 x(t)^2 z(t) - a_6 z(t)^3 + a_7 \\
\dot{w}(t) &= 1.
\end{align*}
$$

Even though the system shown in equation (3) is autonomous, some cannot ensure that the fast changing variable ($t$) and slow varying variables ($x(t), y(t), z(t)$) are fully decoupled and hence we adopt the methods of averaging as shown by [44, 54] to simplify the equation (3). By methods of averaging let us take

$$
\begin{align*}
x(t) &= x_s(t) \cos(t) + y_s(t) \sin(t) \\
y(t) &= -x_s(t) \sin(t) + y_s(t) \cos(t) \\
z(t) &= z_s(t) \\
w(t) &= w_s(t).
\end{align*}
$$

Using (4) in (3) and after some mathematical simplifications with equating sinusoidal and non-sinusoidal terms separately to zero, it can be solved as

$$
\begin{align*}
\dot{x}_s(t) &= -\frac{1}{2} \left[ b_1 x_s(t) + F y_s(t) \right] \\
\dot{y}_s(t) &= \frac{1}{2} \left[ F x_s(t) - b_1 y_s(t) + b_2 \right] \\
\dot{z}_s(t) &= b_3 - b_1 x_s(t)^2 z_s(t) - b_2 y_s(t)^2 z_s(t) - b_3 z_s(t)^3 \\
\dot{w}_s(t) &= 1,
\end{align*}
$$

where $F = 1 - b_1 \left[ x_s^2 + y_s^2 + 4 z_s^2 \right]$ and $b_1 = 0.045$, $b_2 = 0.22$, $b_3 = 0.03$, $b_4 = b_5 = 9.375e^{-3}$, $b_6 = 6.25e^{-3}$, $b_7 = 9.375e^{-2}$. It is clear that the first three states are decoupled from the fourth state making the system completely autonomous and solvable. If the third state $z_s = 0$, then the system becomes similar to a Duffing’s equation. Figure 2 shows the 2D phase portraits of the system (5).

**Figure 2**
2D phase portraits of the system (5)

The finite time Lyapunov exponents (LEs) of the system (5) are calculated using the Wolf’s algorithm [55] with the initial conditions as $[1, 1, 1, 1]$ and finite time for calculating the LEs taken as 20000s. The LEs are $L_1 = 0.00543$, $L_2 = 0$, $L_3 = -0.092$ and the fourth Lyapunov exponent of the system is always zero as the state depends on a constant.
To investigate the performance of the system with parameters, the bifurcation diagrams are plotted. Fixing all other parameters and varying $b_i = [0.015, 0.009]$, the bifurcation of the system (5) is studied. The initial condition for the first iteration is kept as $[1,1,1,1]$ and is changed at the end of every iteration to the last value of the state trajectories. As seen from Figure 3, the system takes a routine period doubling route to chaos and period halving exit from chaos. The chaotic region is seen for $b_i$ values in the range [0.034; 0.049].

**Figure 3**
Bifurcation of the system (5) with $b_i$

$$D^\alpha x = \frac{1}{2} [b_i x_a + F y_a]$$

$$D^\alpha y = \frac{1}{2} [F x_a - b_i y_z + b_i]$$

$$D^\alpha z_a = b_i - b_i x_a^2 z_s - b_i y_a z_a - b_i z_a^3$$

$$D^\alpha w_a = 1,$$

where $F = 1 - b_i [x_a^2 + y_a^2 + 4z_a^2]$ and $b_i = 0.045$, $b_a = 0.22$, $b_b = 0.03$, $b_c = 9.375e-3$, $b_d = 6.25e-3$, $b_e = 9.375e-2$.

There are three main approaches, namely, frequency-domain method [7], Adomian Decomposition Method (ADM) [36] and Adams-Bashforth-Moulton (ABM) algorithm [1, 10, 12, 17]. The method of Adams-Bashforth-Moulton (ABM) studied in this part [17] is used whose convergence and accuracy is more as discussed in [23].

Let us show a fractional system with order $q$ as

$$D^q x = f(t, x), \quad 0 \leq t \leq T,$$

where $x^k(0) = x_0^k$ for $k \in [0, n-1]$.

Equation (9) is similar to the Volterra integral equation [1] given by

$$x(t) = \sum_{k=0}^{n-1} x_0^k \frac{t^k}{k!} + \frac{1}{\Gamma(q)} \int_0^t \frac{f(\tau, x)}{(t-\tau)^{q-1}} d\tau,$$

where $h = \frac{T}{N}$, $t_n = nh$ where $h \in [0, N]$ and $\tau = \frac{\tau}{h}$.

The discrete form of (10) can be defined as

$$x(n+1) = \sum_{k=0}^{n-1} x_0^k \frac{t^k}{k!} + \frac{h^q}{\Gamma(q+1)} \sum_{j=0}^{n-1} a_{j,n+1} f(t_j, x_n(j)),$$

where

$$a_{j,n+1} = \begin{cases} n^{q+1} - (n-q) n+1, & j = 0 \\ (n-j+2)^{q+1} + (n-j)^{q+1} - 2(n-j+1)^{q+1}, & 1 \leq j \leq n \\ 1 & j = n+1 \end{cases}$$

$$x_n (n+1) = \sum_{k=0}^{n} x_0^k \frac{t^k}{k!} + \frac{1}{\Gamma(q)} \sum_{j=0}^{n} b_{j,n+1} f(t_j, x_n(j))$$

### 3. Fractional Order Saturable Reactor Circuit (FOSRC)

In this section, the fractional order saturable reactor circuit is derived using the Caputo fractional derivative [52] defined by

$$D^q x = J^{m-q} x^m, \quad (6)$$

where $m$ is the integer closest to $q$ with $m > q$ and $J^a$ is the $a^{th}$ order Reimann-Liouville integral operator given by

$$J^a y = \frac{1}{\Gamma(a)} \int_0^t (t-\tau)^{a-1} y(\tau) d\tau, \quad (7)$$

where $\Gamma$ is the gamma operator.

Using the $q^{th}$ order Caputo fractional derivative (6) and (7), the FOSRC system is derived as follows:
Let us show a fractional system with order \( q \) as

\[
b_{j,n+1} = \frac{h^q}{q} \left( (n - j + 1)^q - (n - j)^q \right).
\]

Using the above definitions (10) and (11), the FOSRC system can be defined as

\[
x_j(n+1) = x_j(0) + \frac{h^q}{\Gamma(q+2)} \sum_{j=0}^{\infty} \left[ \frac{1}{2} \left( h x'_j(j) + F x_j(j) \right) \right]
\]

\[
y_j(n+1) = y_j(0) + \frac{h^q}{\Gamma(q+2)} \sum_{j=0}^{\infty} \left[ \frac{1}{2} \left( h y'_j(j) + F y_j(j) \right) \right]
\]

\[
z_j(n+1) = z_j(0) + \frac{h^q}{\Gamma(q+2)} \sum_{j=0}^{\infty} \left[ \frac{1}{2} \left( h z'_j(j) + F z_j(j) \right) \right]
\]

where

\[
F_{n+1} = 1 - b_i \left[ x'_o(n+1)^2 + y'_o(n+1)^2 + 3(z'_o(n+1))^2 \right]
\]

\[
F_j = 1 - b_i \left[ x'_o(j)^2 + y'_o(j)^2 + 3(z'_o(j))^2 \right]
\]

and

\[
x_j(n+1) = x_j(0) + \frac{1}{\Gamma(q+2)} \sum_{j=0}^{\infty} \left[ \frac{1}{2} \left( h x'_j(j) + F x_j(j) \right) \right]
\]

\[
y_j(n+1) = y_j(0) + \frac{1}{\Gamma(q+2)} \sum_{j=0}^{\infty} \left[ \frac{1}{2} \left( h y'_j(j) + F y_j(j) \right) \right]
\]

\[
z_j(n+1) = z_j(0) + \frac{1}{\Gamma(q+2)} \sum_{j=0}^{\infty} \left[ \frac{1}{2} \left( h z'_j(j) + F z_j(j) \right) \right]
\]

\[
X_{j,0} = \begin{cases} n^{j+1} - (n - q)(n + 1)^{j+1} & \text{if } j = 0 \\ (n - j + 1)^{j+1} + (n - j)^{j+1} - 2(n - j + 1)^{j+1} & \text{if } 1 \leq j \leq n \\ 0 & \text{if } j = n + 1 \end{cases}
\]

\[
\theta_{j,i+1} = \frac{h^q}{q} \left( (n - j + 1)^q - (n - j)^q \right), \quad 0 \leq j \leq n
\]

for \( i = 1, 2, 3 \), \( j = 1, 2, 3 \),

where \( q \) takes the respective values of \( q_x, q_y, q_z \) depending on the state variables.

For the initial conditions \([1, 1, 1, 1]\) and commensurate fractional order \( q = 0.99 \), the 2D state portraits of the FOSRC system (12) are given in Figure 4.

**Figure 4** 2D phase portraits of the FOSRC system

To find the finite time Lyapunov exponents for the FOSRC system, the method of Wolf’s algorithm [55] is used as discussed in [26]. The LEs of the FOSRC system for fractional order \( q = 0.99 \) are calculated as \( L_1 = 0.00607, L_2 = 0, L_3 = -0.0875 \) and the fourth LE is zero as the state depends on a constant.

Figure 5 shows the bifurcation of the FOSRC system with fractional order \( q \) and as can be seen from the Figure, the FOSRC system shows two regions of chaotic oscillations for \( q \in [0.945, 0.95] \) and for \( q \in [0.985, 1] \) and takes period doubling route to chaos. To see the impact of parameter \( b_i \) on the FOSRC system, we derive the bifurcation plots as shown in Figure 6. The fractional order of the system is fixed at
\( q = 0.99 \). The FOSRC system shows chaotic region for \( 0.026 \leq b_1 \leq 0.044 \) and takes routing period doubling limit cycle route to chaos and period halving exit from chaos. For both the bifurcations shown in Figures 5 and 6, the initial conditions for the first iteration are kept at [1, 1, 1, 1] and, for every iteration, the initial conditions are reinitialized to the end values of the state variables. By comparing Figures 3 and 6, we could say that the bifurcation range of the parameter \( b_1 \) is more in fractional order than the integer order SRC system. This also proves that the fractional order dynamics show more complex behaviors than the integer order counter parts.

**4. Field Programmable Gate Array (FPGA) Implementation of FOSRC**

Many literatures have discussed about the implementation of integer order systems using FPGAs \([3, 9, 15, 35, 37, 42, 51, 53]\). Even though there are many literatures on the integer order FPGA implementations, less have been investigated on fractional order FPGA implementations. A FPGA implementation of fractional order chaotic systems with hidden oscillations have been implemented and the power efficiency analysis for various fractional orders were investigated in \([32, 47]\). FPGA implementation of a genetically optimized sliding mode control algorithm is investigated for synchronization of uncertain hyperchaotic systems \([38]\). Novel hyperchaotic systems, which show both self-excited and hidden attractors for choice of parameters are investigated with their fractional order form and a FPGA implementation is also presented \([19]\). A CFE Tustin approximation based FPGA implementation of various fractional order chaotic systems has been proposed in \([48]\). Grünwald-Letnikov (GL) differ-integral based fractional order analysis of chaotic system with its FPGA implementation is discussed in \([49]\).

Although for numerical analysis of the proposed FOSRC system Adams-Bashforth-Moulton (ABM) \([1, 10, 17]\) is used, for implementing the proposed FOSRC system in FPGA, the Adomian Decomposition Method (ADM) \([5, 14, 36, 56]\) is used. As because the ADM algorithm converges fast \([5, 14, 25, 56]\), the first 6 terms are used to get the solution of FONMCO system as in real cases. A time discretization method should be designed. That is to say, for a time interval of \( t_i \) (initial time) to \( t_f \) (final time), the interval is divided into \( (t_n, t_{n+1}) \) and the value of \( x(n+1) \) at time \( t_{n+1} \) is taken by applying \( x(n) \) at time \( t_n \) using the relation \( x(n + 1) = F(x(n)) \) \([11, 25]\).

As the FOSRC system mainly depends on the first three states and the fourth state \( (w_a) \) is already decoupled from the three states \( (x_a, y_a, z_a) \), the 3D model of the FOSRC system is used and the fractional order discrete form of the dimensionless state equations can be given as
Decomposition Method (ADM) \cite{5, 14, 36, 56} is used. FOSRC system in FPGA, the Adomian transformation is used, for implementing the proposed model of the FOSRC system is used and the fractional implementation of various fractional order chaotic systems is discussed in \cite{49}. A time discretization method should be designed. That is to say, for a time interval of $t_i$ (initial time) to $t_f$ (final time), the first 6 terms are used to get the solution of the system as in real cases. A time step $h = t_{n+1} - t_n$ and $\Gamma(\bullet)$ is the gamma function.

Let $A_i^n = x_n, A_2^n = y_n, A_3^n = z_n$. There are four types of nonlinear terms in the FOSRC system $(x_1^2 \cdot x_2 \cdot x_3^2)$, and the six Adomian polynomials can be derived as in Table 1.

\[
x_{n+1} = \sum_{j=0}^{6} A_j^n \frac{h^j}{\Gamma(j+1)}
\]
\[
y_{n+1} = \sum_{j=0}^{6} A_j^n \frac{h^j}{\Gamma(j+1)}
\]
\[
z_{n+1} = \sum_{j=0}^{6} A_j^n \frac{h^j}{\Gamma(j+1)},
\]

To increase the processing speed all the necessary static components like
\[
h^q, \quad \frac{\Gamma(2q+1)}{\Gamma^2(q+1)} h^{2q}, \quad \frac{\Gamma(3q+1)}{\Gamma(2q+1) \Gamma(q+1)} h^{3q}, \quad \frac{\Gamma(4q+1)}{\Gamma(3q+1) \Gamma(q+1)} h^{4q}, \quad \frac{\Gamma(5q+1)}{\Gamma(4q+1) \Gamma(q+1)} h^{5q}, \quad \frac{\Gamma(6q+1)}{\Gamma(5q+1) \Gamma(q+1)} h^{6q}
\]
are calculated before executing an iteration. The step size $h = 0.001$ and the commensurate fractional order for implementing the FOSRC in FPGA is taken as $q = 0.99$. Figure 7 shows the RTL schematics of the FOSRC system and Figure 8 shows the power consumption chart of the FPGA implemented on FPSRC system and the resources utilized are given in Table 2.

Table 1

<table>
<thead>
<tr>
<th>Nonlinear Component</th>
<th>Adomian polynomials</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(x_1^2 \cdot x_2 \cdot x_3^2)$</td>
<td>$A_1^n = (A_1^n)^2$ \n $A_2^n = 2 A_1^n A_1^n$ \n $A_3^n = 2 A_1^n A_1^n + (A_1^n)^2 \frac{\Gamma(2q+1)}{\Gamma^2(q+1)}$ \n $A_4^n = 2 A_1^n A_1^n + 2 A_1^n A_1^n + (A_1^n)^2 \frac{\Gamma(3q+1)}{\Gamma(2q+1) \Gamma(q+1)}$ \n $A_5^n = 2 A_1^n A_1^n + 2 A_1^n A_1^n + 2 A_1^n A_1^n \frac{\Gamma(4q+1)}{\Gamma(3q+1) \Gamma(q+1)}$ \n $A_6^n = 2 A_1^n A_1^n + 2 A_1^n A_1^n + 2 A_1^n A_1^n \frac{\Gamma(5q+1)}{\Gamma(4q+1) \Gamma(q+1)}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Nonlinear Component</th>
<th>Adomian polynomials</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(x_1^2)$</td>
<td>$A_1^n = (A_1^n)^2$ \n $A_2^n = 2 A_1^n A_1^n$ \n $A_3^n = 2 A_1^n A_1^n + (A_1^n)^2 \frac{\Gamma(2q+1)}{\Gamma^2(q+1)}$ \n $A_4^n = 2 A_1^n A_1^n + 2 A_1^n A_1^n + (A_1^n)^2 \frac{\Gamma(3q+1)}{\Gamma(2q+1) \Gamma(q+1)}$ \n $A_5^n = 2 A_1^n A_1^n + 2 A_1^n A_1^n + 2 A_1^n A_1^n \frac{\Gamma(4q+1)}{\Gamma(3q+1) \Gamma(q+1)}$ \n $A_6^n = 2 A_1^n A_1^n + 2 A_1^n A_1^n + 2 A_1^n A_1^n \frac{\Gamma(5q+1)}{\Gamma(4q+1) \Gamma(q+1)}$</td>
</tr>
</tbody>
</table>

| $(x_2 \cdot x_3^2)$ | $x_1 x_2 x_3$ | $A_1^n = (A_1^n)^2$ \n $A_2^n = 2 A_1^n A_1^n$ \n $A_3^n = 2 A_1^n A_1^n + (A_1^n)^2 \frac{\Gamma(2q+1)}{\Gamma^2(q+1)}$ \n $A_4^n = 2 A_1^n A_1^n + 2 A_1^n A_1^n + (A_1^n)^2 \frac{\Gamma(3q+1)}{\Gamma(2q+1) \Gamma(q+1)}$ \n $A_5^n = 2 A_1^n A_1^n + 2 A_1^n A_1^n + 2 A_1^n A_1^n \frac{\Gamma(4q+1)}{\Gamma(3q+1) \Gamma(q+1)}$ \n $A_6^n = 2 A_1^n A_1^n + 2 A_1^n A_1^n + 2 A_1^n A_1^n \frac{\Gamma(5q+1)}{\Gamma(4q+1) \Gamma(q+1)}$ |
**Figure 7**
RTL schematics of the FOSRC system implemented in FPGA

![RTL Schematics](image)

**Figure 8**
Power utilization of the FOSRC system implemented in FPGA

![Power Utilization](image)

**Table 2**
Resource utilization of the FOSRC system implemented in FPGA

<table>
<thead>
<tr>
<th>Resource</th>
<th>Utilization</th>
<th>Available</th>
<th>Utilization %</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUT</td>
<td>1549</td>
<td>101400</td>
<td>1.53</td>
</tr>
<tr>
<td>FF</td>
<td>192</td>
<td>202800</td>
<td>0.09</td>
</tr>
<tr>
<td>DSP</td>
<td>60</td>
<td>600</td>
<td>10.00</td>
</tr>
<tr>
<td>IO</td>
<td>97</td>
<td>285</td>
<td>34.04</td>
</tr>
<tr>
<td>BUFG</td>
<td>1</td>
<td>32</td>
<td>3.13</td>
</tr>
</tbody>
</table>

**Figure 9**
2D phase portraits of the FPGA implemented FOSRC system plotted in Simulink and processed in Kintex -7 FPGA development module with hardware software co-simulation

![2D Phase Portraits](image)
Figure 9 shows the 2D phase portraits of the FOSRC system plotted in Simulink and processed in Kintex 7 development board (Device=7k160t and Package=f-bg484 S) with hardware software cosimulations.

5. Sound Encryption Based on FOSRC System

There are many chaos-based engineering applications in the literature [4, 30, 33]. In previous studies, integer order chaotic systems are used in general [2, 34, 50]. In this section, sound encryption application is implemented as an application of FOSRC system. One of the most common techniques used in chaos-based encryption applications is to use a chaos-based Random Number Generator (RNG). For this reason, a RNG design based on the FOSRC system was first implemented.

5.1. FOSRC System Based RNG

One of the most basic methods used when performing chaos-based encryption applications is the use of RNGs designed using chaotic systems. These RNGs are implemented by number or bit arrays of numbers derived from the state variables of chaotic systems. The advantage of using chaotic systems here is that they can be computed depending on the system as well as the randomness of the generated sequences. This allows the correct decryption to be performed after encryption.

In this work, random sequences are obtained by using the x and y state variables of the FOSRC system given in Eq. (8). The FOSRC system was solved for the fractional order $q = 0.99$ and 0.05 step value using the Grunwald-Letnikov [24, 29] method. At each iteration, state variables are converted from float to binary and 16 bits are taken and random sequences are generated.

One of the most important criteria to show that a RNG really produces random numbers is that it successfully passes NIST-800-22 tests. For this, values of 1 to 0.01 (P-value) should be obtained from 15 different tests performed within the scope of NIST-800-22. Table 3 shows the test results of two different sequences of 1000000 bits obtained from the FOSRC system. One of these sequences is obtained directly from the variable x. The second random sequence is

<table>
<thead>
<tr>
<th>Statistical Tests</th>
<th>P-value (X)</th>
<th>P-value (XÅY)</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency (Monobit) Test</td>
<td>0,15157262850217</td>
<td>0,128510975637872</td>
<td>Successful</td>
</tr>
<tr>
<td>Block-Frequency Test</td>
<td>0,534351624443865</td>
<td>0,856056627487147</td>
<td>Successful</td>
</tr>
<tr>
<td>Cumulative-Sums Test</td>
<td>0,21479298193147</td>
<td>0,19747568823365</td>
<td>Successful</td>
</tr>
<tr>
<td>Runs Test</td>
<td>0,741356867016703</td>
<td>0,958281270363264</td>
<td>Successful</td>
</tr>
<tr>
<td>Longest-Run Test</td>
<td>0,327480986493886</td>
<td>0,415220465679507</td>
<td>Successful</td>
</tr>
<tr>
<td>Binary Matrix Rank Test</td>
<td>0,445738524821132</td>
<td>0,517421814495113</td>
<td>Successful</td>
</tr>
<tr>
<td>Discrete Fourier Transform Test</td>
<td>0,0967194894593149</td>
<td>0,125399812124251</td>
<td>Successful</td>
</tr>
<tr>
<td>Non-Overlapping Templates Test</td>
<td>0,315843707760675</td>
<td>0,105017005915441</td>
<td>Successful</td>
</tr>
<tr>
<td>Overlapping Templates Test</td>
<td>0,331533601051771</td>
<td>0,851302161188892</td>
<td>Successful</td>
</tr>
<tr>
<td>Maurer’s Universal Statistical Test</td>
<td>0,310552424542791</td>
<td>0,122236299085832</td>
<td>Successful</td>
</tr>
<tr>
<td>Approximate Entropy Test</td>
<td>0,286215734065522</td>
<td>0,260486225611139</td>
<td>Successful</td>
</tr>
<tr>
<td>Random-Excursions Test (x = -4)</td>
<td>0,046042297408132</td>
<td>0,799000245040072</td>
<td>Successful</td>
</tr>
<tr>
<td>Random-Excursions Variant Test (x = -9)</td>
<td>0,256078714085018</td>
<td>0,647208548194962</td>
<td>Successful</td>
</tr>
<tr>
<td>Serial Test-1</td>
<td>0,152884792081078</td>
<td>0,745304036764701</td>
<td>Successful</td>
</tr>
<tr>
<td>Serial Test-2</td>
<td>0,0493793622168328</td>
<td>0,482434651927462</td>
<td>Successful</td>
</tr>
<tr>
<td>Linear-Complexity Test</td>
<td>0,0877588569761234</td>
<td>0,551944470056125</td>
<td>Successful</td>
</tr>
</tbody>
</table>
5.2. Sound Encryption Based on Designed RNG

The block diagram of the sound encryption and decryption processes using the obtained FOSRC system based RNG is shown in Fig. 10. First, the values in the original sound file are converted from float to binary format. All the bits in the obtained binary numbers are XORed with a random sequence obtained from the RNG and encryption is performed. Finally, the encrypted bit sequences are converted back to the float values to obtain the encrypted sound. In the decoding process, the original sound can be obtained by giving the encrypted sound instead of the original sound and repeating the same operations. The original sound is shown in Fig. 11, the encrypted sound is shown in Fig. 12 and the decrypted sound is shown in Fig. 13.

When the figures are examined, there is no similarity between the original and the encrypted sound. In addition, the original sound can be obtained as a result of the decryption process (Fig. 13). Although the original and encrypted sound are not visually similar (Figures 11 and 12), the encryption must be resistant for any filtering process. For this, the original sound spectrum is shown in Fig. 14, the encrypted sound spectrum is shown in Fig. 15, and the decrypted sound spectrum is shown in Fig. 16. Original and decrypted sound spectra appear the same.
This indicates that there is no data loss during encryption and decryption. Encrypted sound spectrum shows a fairly homogeneous distribution. This shows that the encryption process is resistant to the filtering process and that a very good encryption is done.

6. Conclusions

A simple electrical circuit with two saturable reactors which gives chaotic oscillations has been proposed in this paper. Dynamical analysis of the proposed system like Lyapunov exponents and bifurcation analy-
sis have been investigated. A fractional order model of the SRC system has been derived and analyzed using Adam-Bashforth-Moulton method. Bifurcation of the FOSRC system with fractional order was derived and presented to show the existence of chaotic oscillations in FOSRC system with fractional order. For implementing the FOSRC system in FPGA, Adomian decomposition method was used and the RTL schematic of the FOSC systems was presented. Moreover, a new fractional-based sound encryption application has been successfully implemented with its some security analysis. In further research, the use of such saturable inductors in complex chaotic oscillators can be investigated. In addition, chaotic oscillators which use nonlinear resistors and memristors as the nonlinear element can be tested with such saturable inductors.

Acknowledgments
This work is supported by Sakarya University Scientific Research Projects Unit under Grants 2017-09-00-010.

References


