DIGITAL PROCESSOR IMPLEMENTATION OF CHAOTIC GENERATOR

BY

VICTOR GRIGORAŞ1,* and CARMEN GRIGORAŞ2,3

1Technical University “Gheorghe Asachi” of Iaşi
Faculty of Electronics, Telecommunications and Information Technology,
2Gr.T. Popa” University of Medicine and Pharmacy of Iaşi,
Faculty of Biomedical Engineering,
3Institute of Computer Science, Romanian Academy, Iaşi branch

Received: October 31, 2018
Accepted for publication: December 21, 2018

Abstract. Nonlinear systems with complex dynamical behaviour are well studied and electronic implementations are of great interest for measurement, communication and encryption applications. The present contribution aims at developing a digital implementation of a discrete-time chaotic generator, aiming applications in noise generation. The proposed generator is a nonlinear system designed based on an analog prototype submitted to a simple sampling process. The designed digital architecture is developed using a 32 bits ARM Cortex-M4 processor and its performance is studied from the dynamical and statistical points of view.

Keywords: nonlinear systems; chaotic dynamics; digital processor; signal sampling.

1. Introduction

Nonlinear systems perform different types of dynamical behaviors, starting from constant and periodic types up to complex quasi-periodic and

*Corresponding author: e-mail: carmen.grigoras@umfiasi.ro
chaotic ones. Chaotic systems aim applications in noise generation (Grigoraş, 2011a), binary random number calculation (Yalcın, 2004; Yang, 2004), signal encryption for secure communication (Kocarev, 2001; Grigoraş, 2011b), signal generators based on discrete components (Teodorescu, 2011) and other applications requiring complex dynamics. Analogue implementations of the studied nonlinear systems is aiming simple alternatives, (Piper, 2010; Li, 2015), but is affected by parameter sensitivity of the chaotic circuits. Digital implementations are also studied, (Mandal, 2017; Giakoumis, 2018), because of their higher implementation precision and parameter control.

The present contribution presents the performance of a programmable digital implementation for a chaotic nonlinear system, aiming applications in measurement and communications as a noise generator. The proposed discrete-time system is designed starting from a hysteretic fourth order analogue prototype submitted to direct sampling. The dynamic and statistic properties of the resulting system are analyzed and its state equations used to develop a digital processor implementation.

The following section is devoted to the design of the discrete-time nonlinear system, starting from the state equations of a hysteretic analogue prototype. Implementation results are presented in the third section, showing the dynamic and statistical properties of the implemented system. The obtained results highlight the efficiency of digital programmable implementation of the chaos generator, suggesting possible applications in signal processing, communication and biomedical fields. The final section draws resulting conclusions.

2. Discrete Nonlinear System Design

The proposed discrete-time nonlinear system, used to develop the programmed implementation, is a fourth order hysteretic generator, designed on the basis of an analogue prototype, introduced in Grigoraş, (2017), described by the state equations:

$$\begin{align*}
\frac{dx}{dt} &= \omega_0 y - z, \\
\frac{dy}{dt} &= -\omega_0 x + z, \\
\frac{dz}{dt} &= -x - y - \omega_0 z + \text{sat}(v), \\
\frac{dv}{dt} &= -G/\tau (-x) - v/\tau + Th \cdot G/\tau \cdot \text{sat}(v).
\end{align*}$$

(1)

The nonlinear algebraic function \(\text{sat}(v)\), denotes the saturation characteristic of the amplifier used to implement the hysteretic element:

$$\text{sat}(v) = \begin{cases} 
-1 & \text{if } v < -1, \\
v & \text{if } |v| < 1, \\
1 & \text{if } v > 1.
\end{cases}$$

(2)
The Schmitt trigger, used as a basic hysteretic element, is characterized by the differential equation, included in (1):

\[
\begin{align*}
\frac{dv}{dt} &= -\frac{v}{\tau} + Th \cdot \frac{G}{\tau} \cdot \text{sat}(v) - \frac{G}{\tau} \cdot u, \\
w &= \text{sat}(v).
\end{align*}
\] (3)

In eq. (3), \(u\) and \(w\) are the analogue input and output signals of the Schmitt trigger, \(G\) is the open loop gain and \(\tau\) the time constant of the operational amplifier in the implementation circuit and \(Th\) is the positive switching threshold scale factor of the Schmitt trigger. As presented in our previous contribution, equation (3) leads to a simple hysteretic behavior of the Schmitt trigger, the way depicted in Fig. 1, for a value of \(Th = 0.01\).

Fig. 1 – Inverting type Schmitt trigger hysteretic characteristic.

The previous analysis, developed for the analogue hysteretic prototype, highlights the fact that the system (1) is characterized by two equilibrium points that are unstable for positive values of the \(\omega_0\) parameter. By symbolic calculation, the analogue system is demonstrated to be dissipative for all positive values of the coefficients \(\omega_0\) and \(\tau\), while ergodicity and sensitivity properties, specific for chaotic systems, are justified by simulations, for large variation domains of the \(Th\) and \(\omega_0\) parameters.

The approach used in the following to obtain the discrete-time variant of the prototype system (1), is to replace analogue integrators with discrete-time accumulators, characterized by the equation:

\[w[k + 1] = w[k] + T \cdot e[k]\] (4)

where: \(w[k]\) denotes the state of the discrete-time accumulator and \(e[k]\) its discrete input signal. This approach leads to the approximation of the analogue time derivative with a discrete difference, as in:
In equations (4) and (5), $T$ denotes the discrete clock period of the resulting system, equal to the sampling time of the processed signals:

$$T = t_{k+1} - t_k, \quad k \in \mathbb{Z}.$$  \hfill (6)

Applying the proposed method, the resulting nonlinear discrete-time system is described by the difference state equations:

$$
\begin{align*}
\dot{x}[k+1] &= x[k] + T \left( \omega_0 \cdot y[k] - \omega_0 \cdot y[k] - z[k] \right), \\
y[k+1] &= y[k] - T \left( \omega_0 \cdot x[k] - z[k] \right), \\
z[k+1] &= z[k] - T \left( x[k] + y[k] + \omega_0 \cdot z[k] - b \cdot \text{sat}(v[k]) \right), \\
v[k+1] &= v[k] - T \left( G/\tau \cdot x[k] + v[k]/\tau - Th \cdot G/\tau \cdot \text{sat}(v[k]) \right).
\end{align*}
$$ \hfill (7)

Choosing a normalized clock period of maximum value $T = 0.025$, the time evolution of the state variables of the discrete system is non-periodic and unpredictable on large term, similar to the analogue system, as suggested in the results depicted in Fig. 2, where the three trajectories show the desired properties.

For the system parameter $\omega_0 = 2.21$, the state portrait of the discrete-time system follows a resembling profile to the analogue prototype, as shown in Fig. 3. Similar simulations, for different omega values, highlight the different attractors obtained in the analysis of the analogue system too.

![Fig. 2 – Time evolution of the state variables of the discrete system $x[k]$ (dot), $y[k]$ (circle) and $z[k]$ (star).]
3. Implementation Results

The proposed discrete nonlinear system has both dynamic and statistic properties that are attractive for noise or binary random number generation. Due to its discrete-time structure, the designed chaotic generator is open to the possibility of software controlled digital processor implementation.

For the digital implementation, we chose a processor of the ARM Cortex M4 type and an architecture using 32 bit values representation. This processor has sufficient computing power to perform the calculations imposed by the nonlinear algebraic functions comprised in the difference state equations of the nonlinear system to be implemented. The D/A converters, included in the microcontroller structure, have 12 bits conversion capability, reasonable for interfacing the tested digital system with the analogue world. Due to the high enough number representation in the chosen processor, the number of bits in the conversion structure leads to the dominating differences between floating point simulation results and the corresponding digital measurements.

The algorithm used in the program implementation is suggested by the system state equations and leads to the following pseudo code:

```
initialize (actual state vector);
while switch not pressed:
    compute state function (actual state vector);
    store result in (next state vector);
    transfer (next state vector) > (actual state vector);
```

The data structure needed for this simple algorithm is based upon two floating vectors with four components:

```
actual state vector;
next state vector;
```

Added to these vectors are the scalar system parameters:

```
T, \omega_0, Th, b, G, \tau;
```

The processor used in the tested implementation is STMicroelectronics STM32F334R8T6 and the generated signals were acquired with a Tektronix TDS2002B oscilloscope, digital results being saved and graphically represented on computer. Examples of the acquired state variables from the implemented system are presented in Figs. 3 and 4, for two different values of the parameter \omega_0, namely 2.21 and 1.1.
Subtracting the values of the state variables of the simulated and implemented discrete type system, we obtain the error values depicted in Fig. 5, with a maximum error value of $2 \times 10^{-4}$ and relatively uniform fill of the value range.

Statistically analyzing the designed discrete system, we can show results similar to the ones obtained in the analog case. The histograms depicted in Fig. 6, for similar system parameter $\omega_0 = 2.21$, highlight statistics of the first order with two to three peaks, for the $x[k]$, $y[k]$ and $z[k]$ state variables.

Errors in the simulated and implemented state variables histograms are also small, as suggested in the difference graph depicted in Fig. 7. The low order of magnitude maximum error validates the efficiency of use for the digital implementation of the chaotic generator.
Fig. 4 – Implemented attractor 2D projection for $\omega_0 = 1.1$.

Fig. 5 – Difference between $y[k]$ simulated and implemented state variables.
Fig. 6 – Hystograms for $x[k]$ (up), $y[k]$ (middle) and $z[k]$ (down) implemented state variables.
Fig. 7 – Average errors in histograms for the first three state variables: x, y, z.

4. Conclusion

The present contribution concentrates on designing a discrete chaotic generator starting from an analogue prototype. The proposed nonlinear system is developed based on a previously designed analogue generator that is converted to a discrete version by direct sampling thus using discrete accumulators as a replacement of analogue integrators. The previously developed analogue generator is closely followed in performance by the discrete counterpart in both dynamic and statistic performance. The resulting discrete system is digitally implemented by programming a 32 bit ARM Cortex M4 architecture processor. The programed structure implements the calculations given by the discrete-time state equations describing the generator. The resulting structure performances are compared to the more precise 64 bit floating point computer simulations. The small value errors obtained confirm the efficiency of the proposed implementation.

REFERENCES


### IMPLEMENTAREA CU PROCESOR NUMERIC A UNUI GENERATOR HAOTIC

(Rezumat)

Sistemele neliniare având comportari dinamice complexe sunt bine studiate, iar implementările electronice sunt de mare interes pentru aplicații în măsurări, comunicații și criptare. Prezentul articol vizează dezvoltarea unei implementări numerice pentru un generator haotic în timp discret, vizând aplicații la generarea zgomotului. Generatorul propus este un sistem neliniar, proiectat pe baza unui prototip analogic, supus unui proces de eşantionare simplă. Arhitectura numerică proiectată este dezvoltată folosind un processor. ARM Cortex-M4 de 32 de biți și performanțele sale sunt studiate din punct de vedere dinamic și statistic.