NÖHÜ Müh. Bilim. Derg. / NOHU J. Eng. Sci., Erken Görünüm / InPress
Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi
Niğde Ömer Halisdemir University Journal of Engineering Sciences
Araştırma makalesi / Research article
www.dergipark.org.tr/tr/pub/ngumuh / www.dergipark.org.tr/en/pub/ngumuh

A design template for globally stable nonlinear oscillators with multiple dynamics

Çoklu dinamiklere sahip global kararlı lineer olmayan osilatörler için bir tasarım şablonu

İsmail Öztürk[1,*]

[1] *Amasya University, Department of Electrical and Electronics Engineering, 05100, Amasya, Türkiye*

Abstract

In this study, a template is proposed for designing ordinary differential equation-based nonlinear oscillators. The template is a two-dimensional system with two control parameters and an energy function. By choosing the appropriate energy function, it is possible to obtain globally stable systems. These systems can be a gradient system, a Hamiltonian system, or a system with a stable limit cycle depending on the choice of control parameters. Hamiltonian and limit cycle cases can be used as a nonlinear oscillator for various applications. An example system is demonstrated by choosing a simple energy function and the obtained system is simulated to verify its dynamics. Hardware verification of the simulated system is performed with a field programmable gate array (FPGA) implementation.

Keywords: Nonlinear oscillator, Gradient system, Hamiltonian system, Limit cycle, FPGA

Öz

Bu çalışmada adi diferansiyel denklem tabanlı lineer olmayan osilatörlerin tasarımı için bir şablon önerilmektedir. Önerilen şablon iki adet kontrol parametresine ve bir adet enerji fonksiyonuna sahip iki boyutlu bir sistemdir. Uygun enerji fonksiyonu seçimi ile global olarak kararlı sistemler elde etmek mümkün olmaktadır. Bu sistemler farklı kontrol parametreleri için gradyan sistem, Hamilton sistemi veya kararlı bir limit çevrime sahip bir sistem olabilmektedir. Hamilton sistemi ve limit çevrim durumlarında sistem farklı uygulamalar için lineer olmayan osilatör olarak kullanılabilir. Basit bir enerji fonksiyonu seçimiyle örnek bir sistem elde edilip sistemin dinamikleri simülasyonlarla doğrulanmıştır. Simülasyonu yapılan sistemin donanımsal doğrulaması ise alanda programlanabilir kapı dizisi (FPGA) kullanılarak gerçekleştirilmiştir.

Anahtar kelimeler: Lineer olmayan osilatör, Gradyan sistem, Hamilton sistemi, Limit çevrim, FPGA

1 Introduction

Nonlinear oscillators play a fundamental part in many research areas and disciplines including engineering, robotics, chemistry, biology, physics, and mechanics [1–8]. For example, the well-known Van der Pol oscillator is used in many electrical circuits as a self-excitatory or forced oscillator [9, 10]. Oscillators like Van der Pol fall into the category of oscillators having a stable limit cycle [11]. A stable limit cycle is a periodic orbit that attracts nearby orbits causing fixed amplitude oscillations immune to external perturbations. Oscillators with limit cycles are widely used to model biological neuron models and many neuromorphic applications rely on them [12–15]. Central pattern generators and spiking neural networks are such applications [16, 17]. Besides neuronal activity, limit cycle oscillators are also used to model other biological phenomena like heartbeat signals and glycolysis [11, 18].

However, the design and analysis of limit cycles are not often easy due to their nonlinear properties. For example, the set of initial conditions that are attracted to the limit cycle is called the basin of attraction, and it is not always possible to find the basin of attraction explicitly. Most of the time, numerical methods are utilized to approximate the basin of attraction [19]. Knowing the basin of attraction is crucial for applications relying on limit cycle oscillators because initial conditions outside the basin of attraction or excessive external perturbations cause unwanted behavior instead of stable oscillations.

A limit cycle is called globally stable if its basin of attraction is the whole domain of the system (excluding isolated equilibria). Due to this property, globally stable limit cycles always produce stable oscillations regardless of the initial conditions or external perturbations. Therefore, globally stable limit cycle oscillators are more desirable for real-world applications. Again, it is not an easy task to design and analyze a globally stable limit cycle attractor.

One other type of nonlinear oscillators falls into the category of Hamiltonian systems [20]. Unlike limit cycle oscillators having a single periodic orbit, the phase space of Hamiltonian systems is full of different periodic orbits. Moreover, unlike a limit cycle, Hamiltonian orbits do not attract nearby orbits. In practice, a limit cycle oscillator produces a fixed amplitude and fixed frequency oscillation regardless of the initial condition. On the other hand, a Hamiltonian oscillator produces oscillations with different amplitude and frequency depending on the initial conditions.

* Sorumlu yazar / Corresponding author, e-posta / e-mail: ismail.ozturk@amasya.edu.tr (İ. Öztürk)
Geliş / Recieved: 25.03.2024 Kabul / Accepted: 25.07.2024 Yayımlanma / Published: 01.01.2021
doi: 10.28948/ngmuh.1458253

They can be used in applications or models where different oscillations are needed for different initial conditions. Simple harmonic oscillators are examples of Hamiltonian systems.

Another type of nonlinear systems is called gradient systems [21]. Unlike limit cycle systems or Hamiltonian systems, such systems do not allow the existence of any periodic orbits within their domain. Instead, orbits settle into one of the equilibria. Even though the nature of these systems does not allow periodic orbits, such non-oscillatory behavior is still desired in applications like nonlinear control [22].

Due to the wide application areas of nonlinear oscillators, researchers from various backgrounds may want to utilize a limit cycle oscillator, a Hamiltonian system, or a gradient system depending on their needs. However, as mentioned before, it is hard to analyze these systems especially if the researcher is not specialized in the mathematical analysis of dynamical systems. Therefore, researchers from different backgrounds may not properly find and use the correct system.

In this paper, a design template is presented to obtain two-dimensional ordinary differential equation-based systems that are capable of exhibiting all aforementioned behaviors. The obtained system can be a gradient system, a Hamiltonian system, or a system with a limit cycle by changing only two parameter values. Moreover, the single equilibrium of the gradient system and the limit cycle are globally stable, meaning that there is no need to calculate the basin of attractions. Using this template, researchers from different backgrounds may easily obtain gradient, Hamiltonian, and limit cycle behaviors on a single system without worrying about analytical details. Also, a field programmable gate array (FPGA) implementation is given to demonstrate the applicability of the obtained systems.

The organization of the paper is as follows. In Section 2, the design template is introduced, and an example system is obtained using this template. Afterward, the numerical simulations for different behaviors of the obtained system are given. In Section 3, the FPGA implementation details of the obtained system are presented. Finally, the conclusions are given in Section 4.

2 The design template and the numerical simulations

The equations of the design template are given in Equation (1).

$$\begin{aligned} \dot{x} &= -D_1E + \mu(D_1E + D_2E) + ay \\ \dot{y} &= -D_2E + \mu(D_2E - D_1E) - ax \end{aligned} \tag{1}$$

Here, $\mu \in [0,1]$ and $a \in \mathbb{R}$ are the parameter values. D_1 represents partial derivative with respect to x; D_2 represents partial derivative with respect to y. E is the abbreviation for $E(x,y)$ such that $E: \mathbb{R}^2 \to \mathbb{R}$ is a function which represents the energy of the solutions. Even though this energy function may not have actual physical meaning with potential and kinetic parts, it is convenient for analyzing the global behavior of the solutions. If $\dot{E} < 0$, it means that the solutions lose energy. In other terms, solutions must settle into an equilibrium. If $\dot{E} = 0$, it means that the solutions preserve energy which indicates the existence of periodic solutions.

The function E must be chosen according to the following conditions.

(1) E should have continuous partial derivatives.
(2) E should be non-negative ($E \geq 0$).
(3) $E = 0$ if and only if $x \in L$ or x is an equilibrium point. Here, L is a closed curve and it should also depend on parameter a.
(4) Closed curve L should only exist for $a > 0$.
(5) $D_1E.ay - D_2E.ax \leq 0$ for $a \in \mathbb{R}$.

As long as these conditions are satisfied, any function can be used to obtain a system with multiple dynamics. For the obtained systems, parameter μ introduces rotations to the phase plane as its value increases from zero. For $\mu = 1$, the systems become Hamiltonian regardless of the value of parameter a. On the other hand, parameter a introduces a limit cycle to the systems or makes the systems gradient. For $a < 0$, the systems are gradient, and the solutions settle into a globally stable equilibrium. For $a > 0$, the systems have a globally stable limit cycle. For $a = 0$, the system undergoes bifurcation for the appearance of the limit cycle.

These statements can be proven mathematically by investigating the $\dot{E}$ value in Equation (2).

$$\frac{dE}{dt} = D_1E.\dot{x} + D_2E.\dot{y} \tag{2}$$

For $0 \leq \mu < 1$, placing $\dot{x}$ and $\dot{y}$ expressions of Equation (1) into the Equation (2) yields $\dot{E} < 0$. This means that the solutions should asymptotically converge to the equilibrium if L does not exist ($a < 0$). If L exists ($a > 0$), then the solutions asymptotically converge to the closed curve L. Since closed curve L is not a set of equilibria of the system in Equation (1), the L should act as a stable limit cycle.

For $\mu = 1$, placing $\dot{x}$ and $\dot{y}$ into the Equation (2) yields $\dot{E} = 0$ making the system Hamiltonian. As a special case, for $\mu = 1$ and $a = 0$, the function E is exactly the same with the Hamiltonian function.

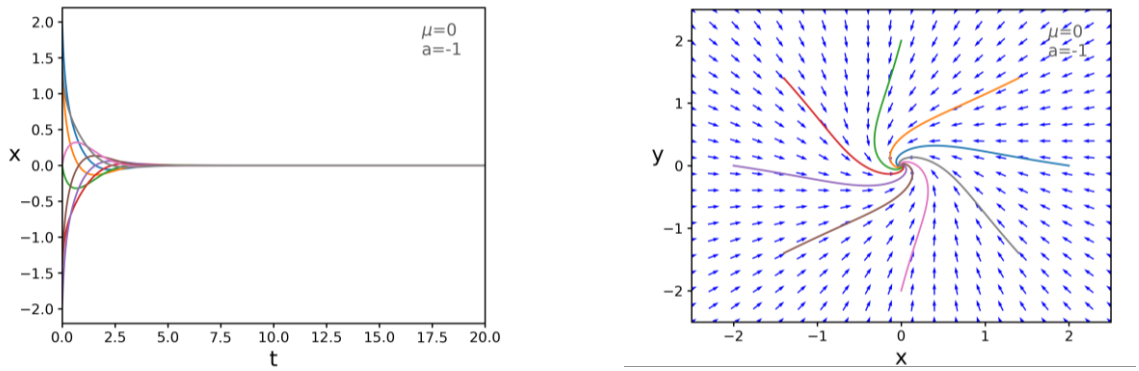
2.1 An example system

The simplest choice of the function E is given in Equation (3).

$$E(x,y) = (x^2 + y^2 - a)^2 \tag{3}$$

It is trivial to show that this function satisfies conditions (1) and (2). Notice that $E = 0$ if and only if $x^2 + y^2 - a = 0$. This is the equation of a circle with radius a. Therefore, the closed curve L of the function E corresponds to a circle with radius a. Hence, the function satisfies conditions (3) and (4). The final condition can be checked by using D_1E and D_2E values given in Equation (4) (constant coefficients are ignored since they only affect the speed of the solutions).

$$\begin{aligned} D_1E &= x(x^2 + y^2 - a) \\ D_2E &= y(x^2 + y^2 - a) \end{aligned} \tag{4}$$



(a) (b)

Figure 1. Gradient behavior for $a < 0$ and $\mu = 0$ (a) $x(t)$ graphs, (b) phase plane

Placing these expressions into the template equations gives us the system in Equation (5).

$$\begin{aligned} \dot{x} &= -x(x^2 + y^2 - a) + \mu(x + y)(x^2 + y^2 - a) \\ &\quad + ay \\ \dot{y} &= -y(x^2 + y^2 - a) + \mu(y - x)(x^2 + y^2 - a) \\ &\quad - ax \end{aligned} \tag{5}$$

Since the shape of the curve L is a circle, the limit cycle will be a circle in the phase plane. Therefore, the oscillations will be sinusoidal. By choosing another function with a different L, limit cycles with different shapes can be easily obtained.

2.2 *Gradient behavior*

The obtained system in Equation (5) should act as a gradient system for $a < 0$ and $\mu \neq 1$. In this case, the origin is a globally stable equilibrium, and all solutions should asymptotically converge to the origin regardless of the initial conditions.

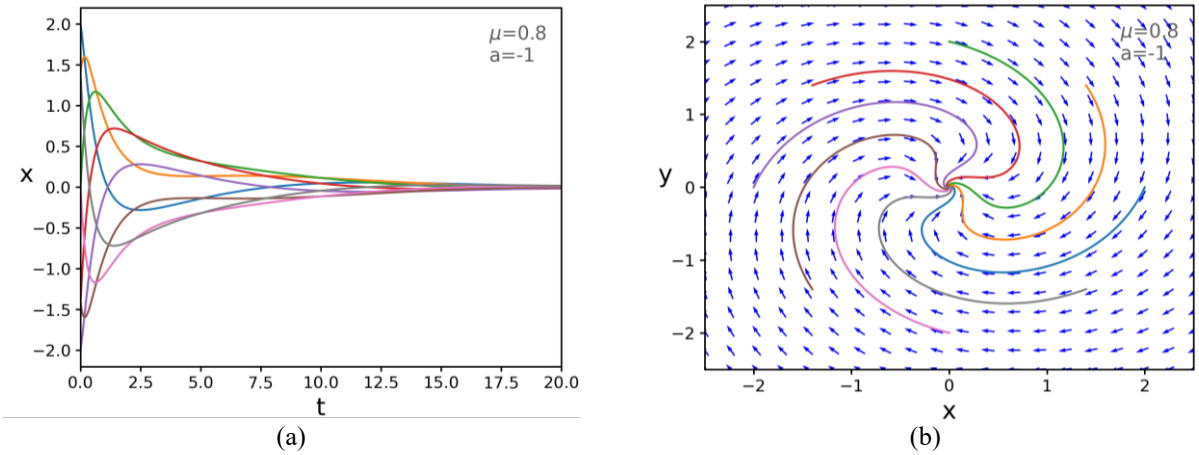
The simulation results for the gradient behavior are given in Figures 1 and 2. Various solutions are depicted in these figures for the initial conditions (2, 0), (1.4, 1.4), (0, 2), (-1.4, 1.4), (-2, 0), (-1.4, -1.4), (0, -2), and (1.4, -1.4). Figure 1 shows the solutions for $a = -1$ and $\mu = 0$. The solutions quickly converge to the origin as expected. Figure 2 demonstrates what happens if μ value is increased from zero. For $a = -1$ and $\mu = 0.8$ the phase plane in Figure 2b has more rotations than the phase plane in Figure 1b. Due to the increased rotations, the solutions converge to zero more slowly as seen in Figure 2a.

2.3 *Hamiltonian behavior*

The obtained system in Equation (5) should act as a Hamiltonian system for $\mu = 1$ regardless of the value of parameter a. In this case, each initial condition should cause a separate periodic solution. Therefore, the phase plane of the system should be full of closed orbits.

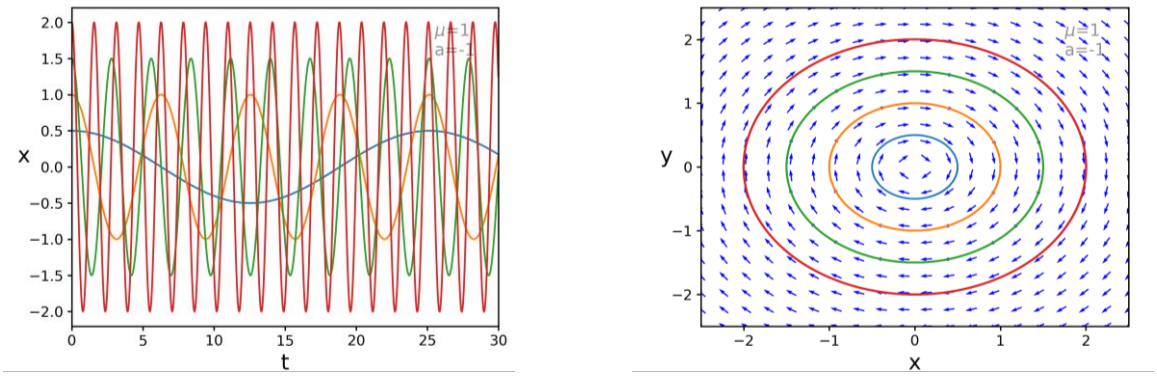
The simulation results for Hamiltonian behavior are given in Figure 3. The solutions are calculated for the initial conditions (0.5, 0), (1, 0), (1.5, 0), and (2, 0). As seen from Figure 3a each initial condition yields a different oscillation with different amplitude and frequency. Therefore, each initial condition has its own closed orbit in the phase plane. This can be seen in Figure 3b.

Changing the parameter a has no effect on the solutions as long as $\mu = 1$. Hence, the same figures are obtained for different a values. However, it must be noted that the parameter a may change the shape of the closed orbits for other function E choices.



(a) (b)

Figure 2. Gradient behavior for $a < 0$ and $\mu = 0.8$ (a) $x(t)$ graphs, (b) phase plane



(a) (b)

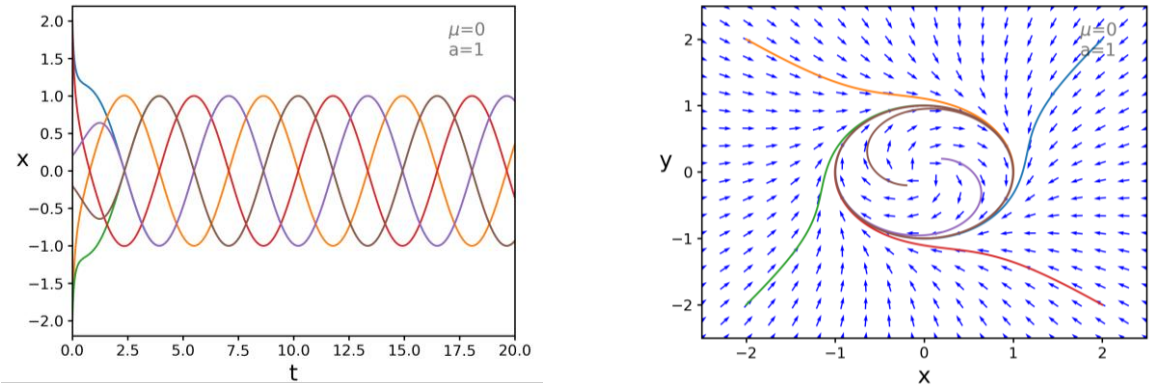
Figure 3. Hamiltonian behavior for $a = -1$ and $\mu = 1$ (a) $x(t)$ graphs, (b) phase plane

2.4 Limit cycle behavior

The obtained system in Equation (5) should act as a limit cycle system for $a > 0$ and $\mu \neq 1$. In this case, a globally stable limit cycle appears causing stable oscillations with constant amplitude and frequency regardless of the initial conditions (except the origin which is an unstable equilibrium). Due to the particular choice of the function E in Equation (3), the amplitude of the oscillations should be $\sqrt{a}$. This may not be the case for other energy functions.

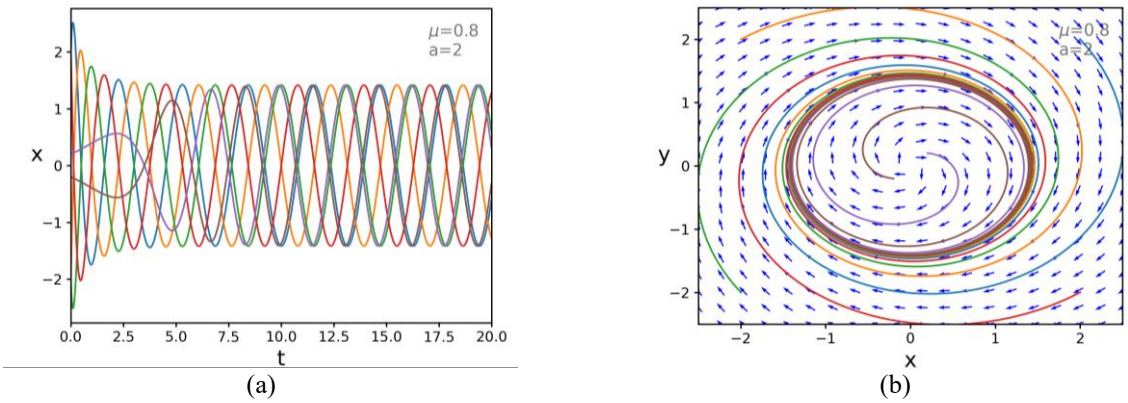
The simulation results for the limit cycle behavior are given in Figures 4 and 5. The simulations are performed for the initial conditions (2, 2), (-2, 2), (-2, -2), (2, -2), (0.2, 0.2), and (-0.2, -0.2). Figure 4 shows the solutions for $a = 1$ and $\mu = 0$. As seen from Figure 4a, all initial conditions settle into the stable limit cycle after some time. This can be seen more clearly from the phase plot given in Figure 4b.

Increasing the parameter a increases the amplitude of the oscillations. Increasing the parameter μ increases the rotations in the phase plane. This just prolongs the settling time, but solutions eventually settle into the limit cycle. Both of these situations are demonstrated in Figure 5 for parameter values $a = 2$ and $\mu = 0.8$.



(a) (b)

Figure 4. Limit cycle behavior for $a = 1$ and $\mu = 0$ (a) $x(t)$ graphs, (b) phase plane



(a) (b)

Figure 5. Limit cycle behavior for $a = 2$ and $\mu = 0.8$ (a) $x(t)$ graphs, (b) phase plane

3 FPGA implementation

Field programmable gate array (FPGA) is a reconfigurable digital integrated circuit. It can be reconfigured for different hardware designs, so it is very suitable for rapid prototyping. For this reason, the system in Equation (5) is implemented using an FPGA chip.

Since FPGA is a digital hardware, the system in Equation (5) should be discretized using an integration method. Euler's method is widely preferred for this purpose due to its implementation simplicity [23]. Even though this method is reliable for gradient and limit cycle systems, it is not suitable for Hamiltonian systems. Since each initial condition yields a different oscillation in Hamiltonian systems, integration errors cause a drift in the solutions. This drift constantly changes the amplitude of the oscillations resulting in a completely different behavior. The integration errors of Euler's method are too high for Hamiltonian systems. Therefore, this method cannot be used for our purposes.

Usually, symplectic integrators are preferred for the discretization of Hamiltonian systems [24]. However, these methods are either defined for separable Hamiltonian systems or they are implicit methods. The Hamiltonian case of the proposed template is not separable. Hence, easier and explicit separable symplectic methods cannot be used. On the other hand, other implicit symplectic methods involve finding roots in each calculation step. This is hard to implement on FPGA and calculations would consume many precious clock cycles. For these reasons, a non-symplectic but better integration method will be used for the discretization.

Explicit midpoint is another discretization method that involves two steps for each calculation [25]. This method is slightly more complicated than Euler's method, but it has far better performance with the Hamiltonian systems. As a matter of fact, all of the simulations in Section 2 are performed using the explicit midpoint method. However, it is still not a symplectic integrator, so its long-term stability is problematic with the Hamiltonian systems. To remedy this, additional hardware is designed to keep track of the integration errors when the parameter μ is set to 1. With this additional hardware, when the solutions come near the initial conditions, the difference from the initial conditions is calculated. When the difference is below some small threshold value, the next values of the solutions are reset to the initial conditions. This way, the oscillations do not drift away due to the accumulating integration errors, and long term stability is achieved. For all practical purposes, this is not different than using a symplectic integrator.

The explicit midpoint method is described in Equation (6).

$$\begin{aligned} k_1 &= f(t_n, u_n) \\ k_2 &= f(t_n + \Delta t/2\,, u_n + (\Delta t/2)k_1) \\ u_{n+1} &= u_n + \Delta t k_2 \end{aligned} \tag{6}$$

Here, $f: \mathbb{R}^2 \to \mathbb{R}^2$ represents the right-hand side of the system in Equation (5), and $u = (x, y)$ represents points in $\mathbb{R}^2$. As it is seen, it requires two steps for calculations. Instead of using similar hardware for each step, an integrator is designed which uses the same hardware for each step. This way, the area usage is reduced. The integrator design is given in Figure 6. This integrator uses two clock cycles for the calculations. The output is connected to D flip flops, which are enabled every two clock cycles with an activator. This allows outputs to be updated after the calculations finished. In the first clock cycle, Δt_in input takes $\Delta t/2$ value; f_in input takes u_n value. In the second clock cycle, Δt_in input takes Δt value; f_in input takes the output of the previous calculation stored in a separate register. This design reduces the area consumption, which is a very important design consideration for FPGAs.

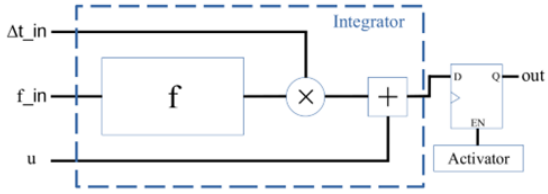


Figure 6. The integrator hardware

The block diagram of the FPGA implementation is given in Figure 7. The implementation consists of three main blocks. The leftmost block is the PLL (Phase Locked Loop) for clock generation. It feeds 1 MHz digital clock signal to other blocks. The middle block is the VHDL (Very High-Speed Integrated Circuit Hardware Description Language) implementation of the integrator in Figure 6. The integration is performed using 32-bit fixed-point arithmetic (Q7.24). Even though it seems that Equation (5) has many terms, the expression $(x^2 + y^2 - a)$ is repeated. So, only one hardware is used to calculate this value. The step size Δt is chosen as 0.01.

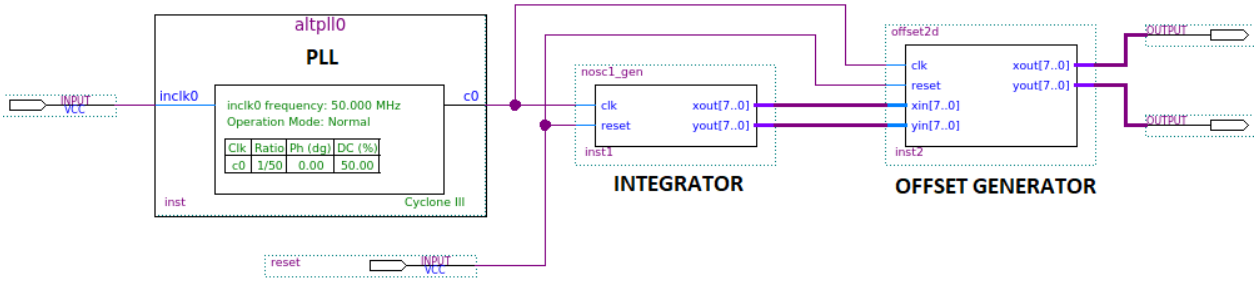


Figure 7. The block diagram of the FPGA implementation

The integrator block has two outputs representing x and y values of the Equation (5). These outputs are connected to the offset block for D/A (digital to analog) conversion. Since the D/A converters are 8-bit, 32-bit calculations are converted to 8-bit when sent to the offset block. The offset block simply adds an offset value to these 8-bit values. This offset causes generation of 2.5V DC offset voltage at the output of the D/A converters. This offset voltage is necessary because the D/A converters operate by a +5V single supply voltage. The offset voltage allows signals to swing around 2.5V.

The experimental setup is shown in Figure 8. The implementation is performed on Altera DE0 development board which contains Cyclone III FPGA chip (EP3C16F484C6). This FPGA has limited resources, but the design can easily fit inside such a low-cost FPGA. The PCB (printed circuit board) in the lower left corner of the figure contains two D/A converter chips. The outputs of the D/A convertors are observed with a two-channel oscilloscope. The implementation results are summarized in Table 1. Since the Hamiltonian case requires additional hardware for the reset operation, the results for the Hamiltonian case are given separately.

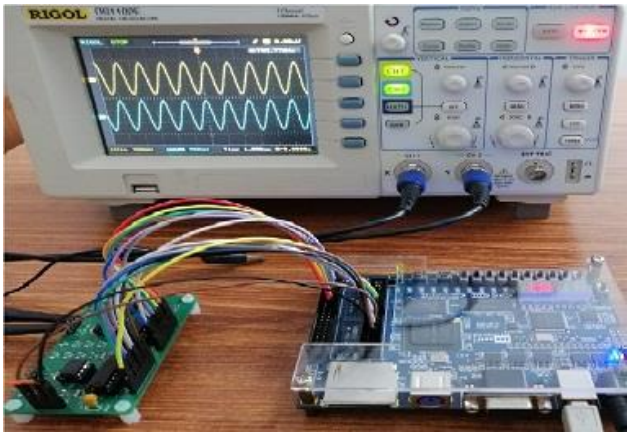


Figure 8. The experimental setup

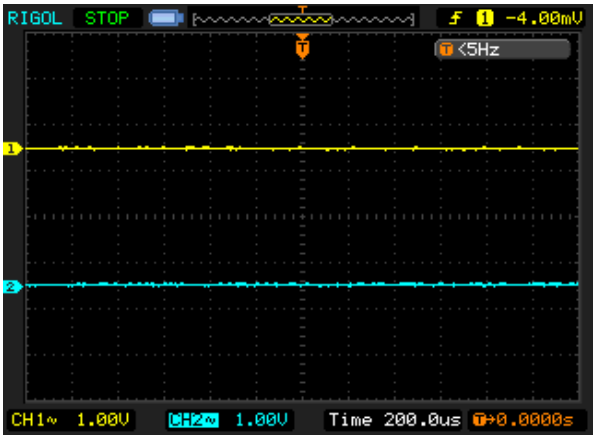
Table 1. FPGA implementation results

Parameters	Hamiltonian	Gradient / Limit Cycle
Area Consumption (Logic Elements)	2645 (17%)	2259 (15%)
Memory Usage (Flip Flop)	179 (1%)	145 (1%)
Multiplier Usage	64 (57%)	64 (57%)
Maximum Frequency	11.75 MHz	12.27 MHz

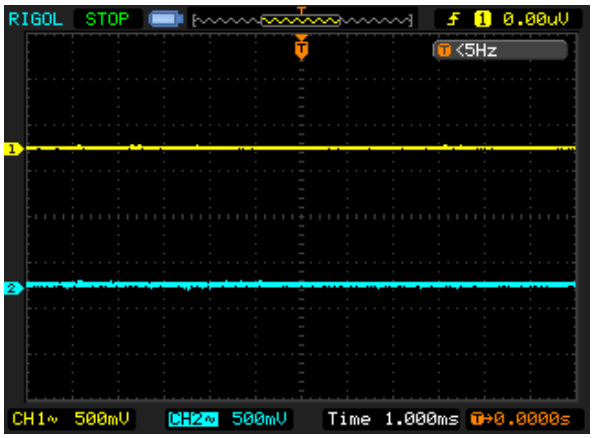
The oscilloscope views of the gradient case for the parameter values $\mu = 0$ and $a = -1$ are presented in Figure 9. For all views, the upper signal is the channel 1 (CH1) of the oscilloscope and it corresponds to the x values. The lower signal, on the other hand, is the channel 2 (CH2) of the oscilloscope and it corresponds to the y values.

The result for the initial condition (1.4, -1.4) is seen in Figure 9a. Similarly, the result for the initial condition (2, 1.4) is seen in Figure 9b. Since convergence is too fast, it is not possible to observe the convergence from the initial conditions to zero using the oscilloscope. Therefore, the gradient case observation in both figures are straight lines. As predicted, the orbits settle into the equilibrium regardless of the initial conditions.

The oscilloscope views of the Hamiltonian case are given in Figure 10 for the parameter values $\mu = 1$ and $a = -1$. In this figure results for different initial conditions are given. Figures 10a and 10b are the results for the initial condition (0.5, 0). On the other hand, Figures 10c and 10d are the results for the initial condition (1, 0). As it is seen from Figures 10a and 10c, the amplitude and the frequency of the oscillation are increased for the latter initial condition. The exact values of the frequency are written at the top right corner of Figures 10a and 10c. As expected, different initial conditions yield an oscillation with different frequency and amplitude. This is the expected behavior for the Hamiltonian case. Also, it should be noted that the oscillations are stable, and they do not drift away with time due to the integration errors.



(a)



(b)

Figure 9. Oscilloscope view of the gradient behavior outputs: (a) for the initial condition (1.4, -1.4); (b) for the initial condition (2, 1.4)

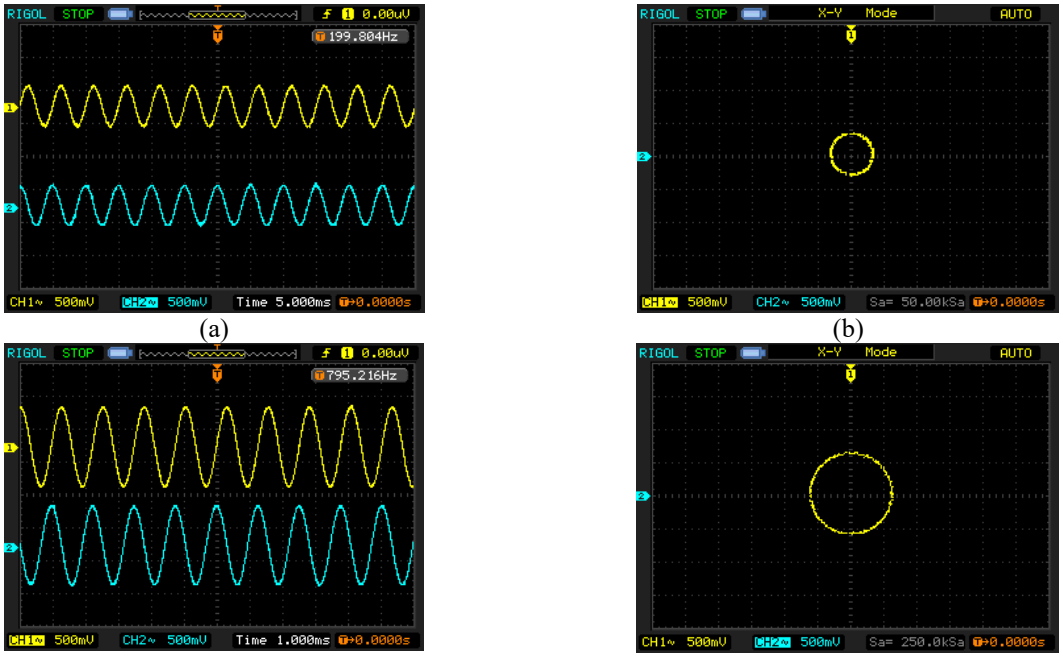


Figure 10. Oscilloscope views of the Hamiltonian behavior outputs: (a) and (b) are for the initial condition (0.5, 0); (c) and (d) are for the initial condition (1, 0)

Finally, the oscilloscope views of the limit cycle case are given in Figure 11 and Figure 12. The results for the parameter values $\mu = 0$, $a = 1$, and the initial condition (2, -2) are presented in Figures 11a and 11b. The results for the initial condition (0.2, -0.2) under the same parameter values are shown in Figures 11c and 11d. As it is seen, different initial conditions yield the same oscillation and changing the initial conditions does not change the frequency or the amplitude of the oscillation.

For the limit cycle case, oscillations with different amplitude and frequency can be obtained by changing the parameter a. Figure 12 demonstrates this case for the parameter values $\mu = 0.8$, $a = 2$ and the initial condition (0.2, 0.2). As it is seen, the amplitude and the frequency of the oscillation are different from the results in Figure 11.

The FPGA implementation confirms the numerical simulations in Section 2. All behaviors of the obtained systems can be successfully used in practical applications.

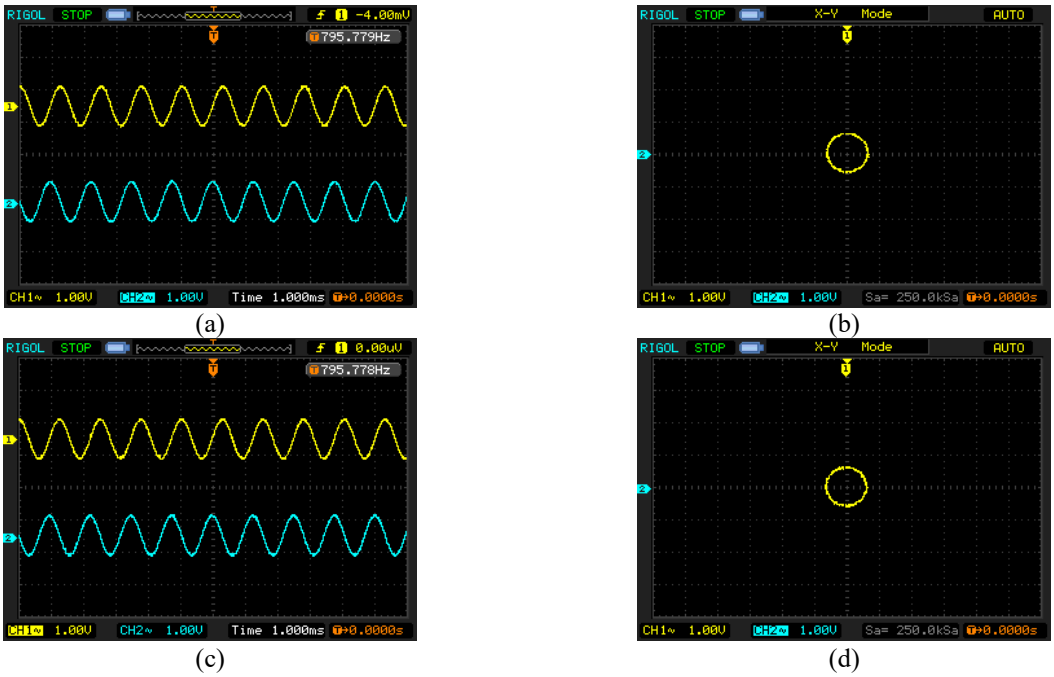
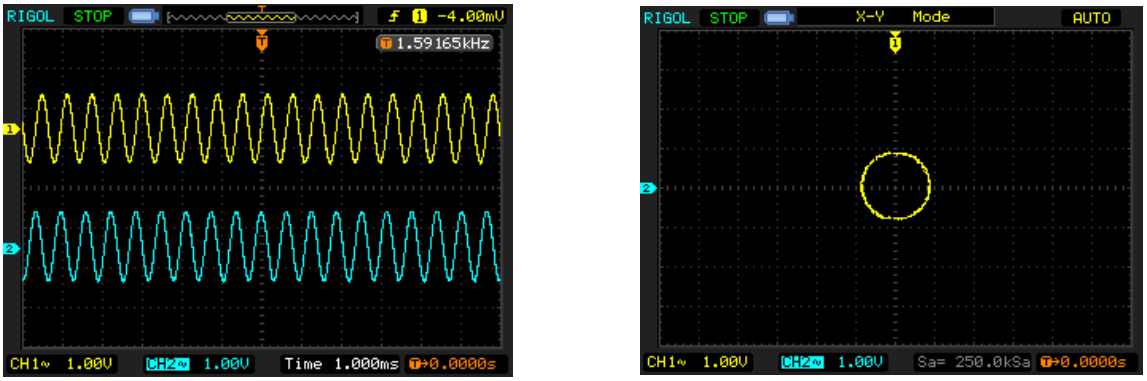


Figure 11. Oscilloscope views of the limit cycle behavior outputs: (a) and (b) are for the initial condition (2, -2); (c) and (d) are for the initial condition (0.2, -0.2)



(a) (b)

Figure 12. Oscilloscope views of the limit cycle behavior outputs for the parameter values $\mu = 0.8$ and $a = 2$

4 Conclusions

The proposed design template can help researchers to design their own nonlinear oscillators without delving into the details of the dynamical system analysis. This is especially helpful because nonlinear oscillators are needed in many disciplines, but researchers may not be too familiar with dynamical systems. The obtained systems are globally stable, so they can be used without concerning about their basin of attractions. The obtained systems can exhibit three main behaviors of dynamical systems namely gradient, Hamiltonian, and limit cycle behavior. Therefore, rich dynamics are observed with the same system. Oscillators with different shapes can be obtained by choosing different energy functions. The amplitude and the frequency of the oscillations can be adjusted by changing the parameter values or the initial conditions depending on the chosen behavior. Also, the FPGA implementation shows that the obtained systems can be used in practical applications. Especially, the solution presented in Section 3 for the long-term stability problem of the Hamiltonian systems can be helpful to the practical applications of Hamiltonian systems.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Similarity rate (iThenticate): 9 %

References

[1] P. Kakou, S. K. Gupta, and O. Barry, A nonlinear analysis of a Duffing oscillator with a nonlinear electromagnetic vibration absorber–inerter for concurrent vibration mitigation and energy harvesting. Nonlinear Dynamics, 112(8), 5847–5862, 2024. https://doi.org/10.1007/s11071-023-09163-6.

[2] W. Tian, T. Zhao, and Z. Yang, Supersonic meta-plate with tunable-stiffness nonlinear oscillators for nonlinear flutter suppression. International Journal of Mechanical Sciences, 229107533, 2022. https://doi.org/10.1016/j.ijmecsci.2022.107533.

[3] Y. Zhu, Y. Wu, Q. Liu, T. Guo, R. Qin, and J. Hui, A backward control based on σ-Hopf oscillator with decoupled parameters for smooth locomotion of bio-inspired legged robot. Robotics and Autonomous Systems, 106165–178, 2018. https://doi.org/10.1016/j.robot.2018.05.009.

[4] A. J. Adéchinan, Y. J. F. Kpomahou, L. A. Hinvi, and C. H. Miwadinou, Chaos, coexisting attractors and chaos control in a nonlinear dissipative chemical oscillator. Chinese Journal of Physics, 772684–2697, 2022. https://doi.org/10.1016/j.cjph.2022.03.052.

[5] F. Morán and A. Goldbeter, Onset of birhythmicity in a regulated biochemical system. Biophysical Chemistry, 20(1–2), 149–156, 1984. https://doi.org/10.1016/0301-4622(84)80014-9.

[6] A. Goldbeter, Computational approaches to cellular rhythms. Nature, 420(6912), 238–245, 2002. https://doi.org/10.1038/nature01259.

[7] H. G. Mayr and K. H. Schatten, Nonlinear oscillators in space physics. Journal of Atmospheric and Solar-Terrestrial Physics, 7444–50, 2012. https://doi.org/10.1016/j.jastp.2011.09.008.

[8] X. Huang and Q. Cao, A novel nonlinear oscillator consisting torsional springs and rigid rods. International Journal of Non-Linear Mechanics, 161104684, 2024. https://doi.org/10.1016/j.ijnonlinmec.2024.104684.

[9] N. Inaba, H. Okazaki, and H. Ito, Nested mixed-mode oscillations in the forced van der Pol oscillator. Communications in Nonlinear Science and Numerical Simulation, 133107932, 2024. https://doi.org/10.1016/j.cnsns.2024.107932.

[10] L. Zhu, Dynamics of switching van der Pol circuits. Nonlinear Dynamics, 87(2), 1217–1234, 2017. https://doi.org/10.1007/s11071-016-3111-8.

[11] A. Jenkins, Self-oscillation. Physics Reports, 525(2), 167–222, 2013. https://doi.org/10.1016/j.physrep.2012.10.007.

[12] E. M. Izhikevich, Simple model of spiking neurons. IEEE Transactions on Neural Networks, 14(6), 1569–1572, 2003. https://doi.org/10.1109/TNN.2003.820440.

[13] N. Korkmaz, İ. Öztürk, and R. Kılıç, Modeling, simulation, and implementation issues of CPGs for neuromorphic engineering applications. Computer Applications in Engineering Education, 26(4), 782–803, 2018. https://doi.org/10.1002/cae.21972.

[14] N. Dahasert, İ. Öztürk, and R. Kılıç, Experimental realizations of the HR neuron model with programmable hardware and synchronization applications. Nonlinear Dynamics, 70(4), 2343–2358, 2012. https://doi.org/10.1007/s11071-012-0618-5.

[15] N. Korkmaz, İ. Öztürk, and R. Kılıç, Multiple perspectives on the hardware implementations of biological neuron models and programmable design aspects. Turkish Journal of Electrical Engineering and Computer Sciences, 24(3), 1729–1746, 2016. https://doi.org/10.3906/elk-1309-5.

[16] Yu. A. Tsybina, S. Yu. Gordleeva, A. I. Zharinov, I. A. Kastalskiy, A. V. Ermolaeva, A. E. Hramov, and V. B. Kazantsev, Toward biomorphic robotics: A review on swimming central pattern generators. Chaos, Solitons & Fractals, 165112864, 2022. https://doi.org/10.1016/j.chaos.2022.112864.

[17] N. K. Kasabov, Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence, Springer Series on Bio and Neurosystems, Springer, Berlin, Heidelberg, 2019.

[18] E. E. Sel'kov, Self-oscillations in glycolysis 1. A simple kinetic model. European Journal of Biochemistry, 4(1), 79–86, 1968. https://doi.org/10.1111/j.1432-1033.1968.tb00175.x.

[19] P. Belardinelli and S. Lenci, A first parallel programming approach in basins of attraction computation. International Journal of Non-Linear Mechanics, 8076–81, 2016. https://doi.org/10.1016/j.ijnonlinmec.2015.10.016.

[20] J. K. Moser, Lectures on Hamiltonian systems. In Hamiltonian Dynamical Systems, CRC Press, 1987.

[21] S. H. Strogatz, Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering, 2nd Edn., CRC Press, 2019.

[22] J. Cortés, A. J. van der Schaft, and P. E. Crouch, Gradient realization of nonlinear control systems. IFAC Proceedings Volumes, 36(2), 63–68, 2003. https://doi.org/10.1016/S1474-6670(17)38868-7.

[23] İ. Öztürk and R. Kılıç, A novel method for producing pseudo random numbers from differential equation-based chaotic systems. Nonlinear Dynamics, 80(3), 1147–1157, 2015. https://doi.org/10.1007/s11071-015-1932-5.

[24] P. J. Channell and C. Scovel, Symplectic integration of Hamiltonian systems. Nonlinearity, 3(2), 231, 1990. https://doi.org/10.1088/0951-7715/3/2/001.

[25] J. Sundnes, Solving ordinary differential equations in python, Springer Nature, 2024.