

**OTONOM OLMAYAN BİR KAOTİK SİSTEM
VE UYGULAMASI**

**YÜKSEK LİSANS TEZİ
Müh. Serkan SEMİZ
504021203**

**Tezin Enstitüye Verildiği Tarih : 2 Mayıs 2005
Tezin Savunulduğu Tarih : 1 Haziran 2005**

**Tez Danışmanı : Doç.Dr. Serdar ÖZOĞUZ
Diğer Jüri Üyeleri Doç.Dr. Ece Olcay GÜNEŞ
Yrd.Doç.Dr. Osman Kaan EROL**

HAZİRAN 2005

ÖNSÖZ

Bu tez çalışması sırasında yardımlarını esirgemeyen Doç. Dr. Serdar Özoğuz'a ve desteklerinden dolayı aileme teşekkürlerimi sunarım.

Haziran 2005

Serkan SEMİZ

İÇİNDEKİLER

KISALTMALAR	iv
TABLO LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÖZET	ix
SUMMARY	x
1. GİRİŞ	1
1.1 Kaosun Tanımı	1
1.2 Uygulama Alanları	1
1.2.1 Rasgele Sayı Üretimi	2
1.2.2 Şifreleme	3
1.2.3 Mesaj Gizleme	4
1.2.4 Kaotik Haberleşme	5
1.2.4.1 Verici Yapıları	6
1.2.4.2 Alıcı Yapıları	7
2. OTONOM OLMAYAN KAOTİK OSİLATÖRÜN İNCELENMESİ	9
2.1 Otonom Olmayan Kaotik Osilatör	9
2.2 Kaotik Osilatörün Farklı Lineer Olmayan Fonksiyonlar ile Gerçeklenmesi	13
2.3 Otonom Olmayan Kaotik Osilatörün Duffing Osilatörü ile Benzerliği	17
2.4 Kaotik Osilatörün Melnikov Metodu ile İncelenmesi	19
2.5 Dallanma Diyagramları	22
3. OTONOM OLMAYAN KAOTİK OSİLATÖRÜN GERÇEKLENMESİ	25
3.1 OTA Yapısı	25
3.2 Osilatör Yapısı	29
3.3 Kaotik Osilatör	31
4. KAOTİK OSİLATÖR İLE RASGELE SAYI ÜRETİLMESİ	36
4.1 Rasgele Sayı Üreteçleri	36
4.2 Rasgelelik Testleri	37
4.3 Otonom Kaotik Osilatör ile Rasgele Sayı Üretimi	39
4.4 Otonom Olmayan Kaotik Osilatör ile Rasgele Sayı Üretimi	42
5. SONUÇLAR	51
KAYNAKLAR	53
ÖZGEÇMİŞ	55

KISALTMALAR

ASCII	: American Standard Code for Information Interchange
CSK	: Chaos Shift Keying
COOK	: Chaotic On-Off Keying
DCSK	: Differential Chaos Shift Keying
TR	: Transmitted Reference
SR	: Stored Reference
OTA	: Operational Transconductance Amplifier
CMOS	: Complementary Metal Oxide Semiconductor
NIST	: National Institute of Standards and Technology
FIPS	: Federal Information Processing Standards
SPICE	: Simulation Program with Integrated Circuit Emphasis

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 3.1. OTA yapısının transistor boyutları	28
Tablo 3.2. Kaotik osilatörde kullanılan eleman değerleri.....	32
Tablo 3.3. Arzu edilen g_m değerlerini ayarlamak için gerekli kontrol akımları	33
Tablo 4.1. Koşu sayılarının içinde bulunması gerekli aralıklar.....	38
Tablo 4.2. Nümerik analizler ile elde edilen dizilerin rasgelelik test sonuçları	46
Tablo 4.3. SPICE simülasyonu ile elde edilen dizinin rasgelelik test sonuçları	50

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1 : Kaotik Rasgele Sayı Üreteci Blok Diyagramı.....	2
Şekil 1.2 : Kaotik çekicinin (V_{min} , V_{max}) aralığının S adet bölüme ayrılması.....	3
Şekil 1.3 : İşaret Maskeleye Yapısı.....	4
Şekil 1.4 : Kaos haberleşmesinde kullanılan (a) CSK, (b) COOK, (c) DCSK yapıları.....	6
Şekil 1.5 : Korelasyon-temelli CSK alıcısı.....	8
Şekil 1.6 : DCSK alıcısı.....	8
Şekil 1.7 : COOK alıcısı.....	8
Şekil 2.1 : İki kapasite elemanına sahip ikinci dereceden sinüs osilatörü yapısı.....	9
Şekil 2.2 : Kaotik osilatör yapısı.....	10
Şekil 2.3 : X durum değişkeninin ölçeklenmiş zamana göre değişimi.....	11
Şekil 2.4 : $(\varepsilon, n, \beta, \varphi)=(0.05, 1.4, 0.44, 0.14)$ parametre değerleri için elde edilen çekicinin (a) X-Y düzlemine izdüşümü (b) 3-boyutlu çizimi.....	12
Şekil 2.5 : Uygulanan farklı tipte lineer olmayan fonksiyonlar (a) işaret fonksiyonu, (b) $\tanh(10x)$, (c) $x-x^3$, (d) $x-x^5$	13
Şekil 2.6 : $(\varepsilon, n, \beta, \varphi)=(0.05, 1.4, 0.44, 0.14)$ parametre seti için (2.6) denkleminde lineer olmayan fonksiyon olarak $\tanh(10x)$ kullanıldığında elde edilen kaotik çekici.....	14
Şekil 2.7 : $(\varepsilon, n, \beta, \varphi)=(0.05, 0.2, 0.3, 0.1)$ parametre seti için (2.6) denkleminde lineer olmayan fonksiyon olarak $x-x^3$ kullanıldığında elde edilen kaotik çekici.....	14
Şekil 2.8 : $(\varepsilon, n, \beta, \varphi)=(0.05, 0.2, 0.3, 0.1)$ parametre seti için (2.6) denkleminde lineer olmayan fonksiyon olarak $x-x^5$ kullanıldığında elde edilen kaotik çekici.....	15
Şekil 2.9 : Uygulanan farklı tipte lineer olmayan fonksiyonlar (a) 3 basamaklı işaret fonksiyonu, (b) 5 basamaklı işaret fonksiyonu...	15
Şekil 2.10 : $(\varepsilon, n, \beta, \varphi)=(0.05, 1, 0.3, 0.2)$ parametre seti için (2.6) denkleminde lineer olmayan fonksiyon olarak 3 basamaklı işaret fonksiyonu kullanıldığında elde edilen kaotik çekici.....	16
Şekil 2.11 : $(\varepsilon, n, \beta, \varphi)=(0.05, 1, 0.3, 0.2)$ parametre seti için (2.6) denkleminde lineer olmayan fonksiyon olarak 5 basamaklı işaret fonksiyonu kullanıldığında elde edilen kaotik çekici.....	16
Şekil 2.12 : Otonom olmayan kaotik osilatörün Duffing Osilatörüne	

dönüştürülmesi.....	17
...	
Şekil 2.13 : Duffing Osilatöründen elde edilen kaotik çekici.....	18
Şekil 2.14 : Otonom olmayan kaotik osilatörün van der Pol Osilatörüne dönüştürülmesi.....	18
.....	
Şekil 2.15 : Lineer olmayan fonksiyon olarak $\tanh(10x)$ kullanıldığı durum için Melnikov Fonksiyonunun nümerik çözümü.....	20
Şekil 2.16 : Lineer olmayan fonksiyon olarak 3 basamaklı işaret fonksiyonu kullanıldığı durum için Melnikov Fonksiyonunun nümerik çözümü.....	21
...	
Şekil 2.17 : $(\varepsilon, \beta, \varphi)=(0.05, 0.44, 0.14)$ değerleri için elde edilen, n parametresinin $[0.9, 1.1]$ aralığındaki dallanma diyagramı.....	22
Şekil 2.18 : $(\varepsilon, \beta, \varphi)=(0.05, 0.44, 0.14)$ değerleri için elde edilen, n parametresinin $[0.91, 0.95]$ aralığındaki dallanma diyagramı.....	23
Şekil 2.19 : $(n, \beta, \varphi)=(1.4, 0.44, 0.14)$ değerleri için elde edilen, ε parametresinin $[0.064, 0.071]$ aralığındaki dallanma diyagramı...	23
Şekil 2.20 : $(\varepsilon, \beta, n)=(0.05, 0.44, 1.4)$ değerleri için elde edilen, φ parametresinin $[0.18, 0.25]$ aralığındaki dallanma diyagramı.....	24
Şekil 3.1 Tasarımda kullanılan OTA yapısı.....	25
Şekil 3.2 : OTA'nın çıkış akımının giriş gerilimi ile değişimi ($I_A=48.5\mu A$).....	26
.	
Şekil 3.3 : OTA'nın geçiş iletkenliğinin frekans cevabı ($I_A=48.5\mu A$).....	27
Şekil 3.4 : OTA'nın giriş terminallerinin direncinin frekans ile değişimi ($I_A=48.5\mu A$).....	27
.	
Şekil 3.5 : OTA'nın çıkış terminalinin direncinin frekans ile değişimi ($I_A=48.5\mu A$).....	27
.	
Şekil 3.6: : OTA eğiminin farklı I_A kontrol akımı değerleri için frekans cevabı.....	28
Şekil 3.7 : 4OTA2C osilatör topolojisi.....	29
Şekil 3.8 : 4OTA2C osilatör topolojisinde V_{C1} 'in zamanla değişimi.....	30
Şekil 3.9 : 4OTA2C osilatör topolojisinde V_{C1} 'in Fourier analizi.....	30
Şekil 3.10 : 4OTA2C osilatör topolojisinde sınırlandırılmış V_{C1} 'in zamanla değişimi.....	31
.	
Şekil 3.11 : Kaotik osilatör devresi.....	31
Şekil 3.12 : Geribesleme yapısı.....	32
Şekil 3.13 : V_{C1} geriliminin ve geribesleme akımının zamanla değişimi.....	32
Şekil 3.14 : V_{C1} geriliminin zamanla değişimi	33

Şekil 3.15 : V_{C2} geriliminin zamanla değişimi.....	34
Şekil 3.16 : V_{C2} 'nin V_{C1} 'e karşı çizdirilmesiyle elde edilen kaotik çekici.....	34
Şekil 3.17 : $f_P=2\text{MHz}$ ve $C=1\text{pF}$ için V_{C2} 'nin V_{C1} 'e karşı çizdirilmesiyle elde edilen kaotik çekici.....	34
Şekil 4.1 : Otonom kaotik osilatör ile rasgele sayı üretimi.....	39
Şekil 4.2 : Otonom kaotik osilatörün ürettiği çift sarmallı çekici.....	40
Şekil 4.3 : Otonom kaotik osilatörün ürettiği çekicinin alt uzayları.....	40
Şekil 4.4 : Bir örnek dizi için Von Neumann Metodu ile elde edilen çıkış dizisi.....	41
...	
Şekil 4.5 : Otonom olmayan kaotik osilatörün $\sin \phi \tau = 0$ için Poincaré kesiti.....	42
...	
Şekil 4.6 : X durum değişkeninin bir sonraki periyottaki değerinin şimdiki değerine karşı çizdirilmesiyle elde edilen dönüşüm.....	43
Şekil 4.7 : X durum değişkeninin, saat işaretinin yükselen kenarında örneklenen 150 bin değerinin dağılımını gösteren histogram.....	44
Şekil 4.8 : Poker Testinin sonucunun kullanılan normalize eşik değeri ile değişimi.....	46
...	
Şekil 4.9 : Otonom olmayan kaotik osilatör ile rasgele sayı üretimi.....	46
Şekil 4.10 : Otonom olmayan kaotik sistem ile gerçekleştirilen rasgele sayı üretici yapısı (a) kaotik osilatör, (b) karşılaştırıcı, (c) örnekleyici.....	47
...	
Şekil 4.11 : Rasgele sayı üretici yapıda (a) V_{C1} işareti (b) σ işareti (c) σ işaretinin saat işaretinin yükselen kenarlarında örneklenmesiyle elde edilen σ_i işareti.....	48
Şekil 4.12 : Poker testi sonucunun kullanılan eşik gerilimi ile değişimi.....	50

OTONOM OLMAYAN BİR KAOTİK SİSTEM VE UYGULAMASI

ÖZET

Deterministik bir sistemin ilk koşullara yüksek duyarlılığın sonucu olarak gösterdiği periyodik olmayan davranışları ifade eden kaos, birçok bilimsel disiplinde araştırmacıların karşısına çıkan bir olgudur. Lineer olmayan sistemlerin çoğu ve kaotik sistemlerin neredeyse tamamı kapalı formda çözümlere sahip değildir. Bilgisayar ile gerçekleştirilen nümerik analizler kaotik sistemlerin davranışını daha iyi anlamamıza yardımcı olur. Son yıllarda bilgisayarların hızlanması ile nümerik analiz kolaylaşmış ve bu durum kaos araştırmalarına hız kazandırmıştır. Elektronik ve haberleşme alanında da kaos önemli uygulamalarda kullanılır hale gelmiştir. Bunlara örnek olarak rasgele sayı üretimi, şifreleme, mesaj gizleme ve kaotik modülasyon gösterilebilir. Bu uygulamalarda temel olarak kaotik işaretlerin geniş bantlı yapılarından ve karmaşıklıkları nedeniyle uzun zaman kestirimlerinin zor oluşundan faydalanılmaktadır.

Tezin ilk bölümünde kaosu tanımladıktan sonra uygulama alanları için örnekler verilmiştir. İkinci dereceden bir osilatöre saat işareti ve lineer olmayan geribesleme eklenmesiyle kaotik bir osilatör elde edilebilmektedir. İkinci bölümde bu prensipten hareketle oluşturulan kaotik osilatör yapısı incelenmiştir. Otonom olmayan osilatörün gösterdiği kaotik davranış Melnikov Metodu ile doğrulanmış ve dallanma diyagramları çizilerek sistemin zengin dinamik yapısı gözler önüne serilmiştir. Ayrıca osilatörde kullanılabilecek farklı nonlineer fonksiyon tipleri incelenmiştir. Üçüncü bölümde tümleştirilmeye uygun şekilde tasarlanan otonom olmayan yüksek frekanslı kaotik osilatörün detayları verilmiştir. Dördüncü bölümde kaotik süreçlerden rasgele sayı üretilmesi konusu açıklanmış, tasarlanan kaotik osilatörü kullanarak gerçekleştirilebilecek bir rasgele sayı üretici yapısı önerilmiş ve tasarım detayları aktarılmıştır. Son olarak beşinci bölümde teze ilişkin sonuçlar verilmiş ve osilatör ile ilgili bazı önerilerde bulunulmuştur.

A NON-AUTONOMOUS CHAOTIC SYSTEM and its APPLICATION

SUMMARY

Chaos is the aperiodic behavior of a deterministic system, which depends sensitively on the initial conditions. This phenomenon is encountered by scientists in many disciplines. Few nonlinear systems and almost no chaotic systems possess closed-form solutions. Computer-based numeric analyses help researchers gain a better understanding of such chaotic behavior. Advances in computers have made numerical analysis easier, which has accelerated the studies in the chaos field. Chaos has proved to be useful in many important applications in electronics and telecommunications. Random number generation, encryption, message hiding and chaotic modulation are examples for such applications in which the broad-band nature and the complex behavior of chaotic signals are utilized.

In the first chapter of this thesis, chaos is defined and examples are given for chaotic applications. Using pulse excitations and self feedback, any given second-order sinusoidal oscillator with two capacitors can be converted into a chaotic oscillator. The chaotic oscillator obtained by applying this circuit design method is examined in the second chapter. The chaotic behavior of the non-autonomous structure is verified using the Melnikov Method and the rich dynamics of the system are illustrated with bifurcation diagrams. Different types of nonlinear functions that can be used in the circuit are experimented. In the third chapter, the details of the high frequency chaos oscillator to be realized in integrated circuit is given. In the fourth chapter, the random number generation technique using chaotic processes is explained, a circuit for random number generation using the designed non-autonomous chaos oscillator is proposed and the details of the design are revealed. Finally in the fifth chapter, the conclusions of this thesis is given and some suggestions are made for the oscillator topology.

1. GİRİŞ

Kaos, birçok alanda rastlanılan bir olgudur. Uygulamaları astronomi, biyoloji, kimya, mühendislik, matematik, meteoroloji, fizik ve hatta sosyal bilimlerde araştırmacıların karşısına çıkmaktadır [1-3]. Son yıllarda kaos araştırmalarında yaşanan ilerlemeler büyük ölçüde bilgisayarların gelişmesi sonucunda ucuz işlem gücünün bilim adamlarının hizmetine sunulması ile gerçekleşmiştir. Lineer olmayan sistemlerin çoğunun ve kaotik sistemlerin neredeyse tamamının kapalı formda çözümlerine ulaşamaz. Bu noktada bilgisayarların hızlı ve kolay şekilde gerçekleştirdikleri nümerik analizler devreye girer. Önceden sadece büyük bir bilgisayar tesisine giriş yetkisi bulunan bilim adamlarıyla sınırlı olan nümerik analiz yapabilme olanağı, günümüzde kişisel bilgisayara sahip herkes tarafından ulaşılabilir hale gelmiştir. Ayrık zaman temelli bilgisayarların sürekli zaman sistemleri simülasyonlarında kaçınılmaz hatalar yaptıkları bilinmektedir ancak teorik çalışmalar ile doğrulanarak kullanılan simülasyonlar, araştırmacılar için çok faydalı araçlar haline gelmiştir [4].

1.1 Kaosun Tanımı

Kaos, deterministik bir sistemin ilk koşullara yüksek duyarlılığın sonucu olarak gösterdiği periyodik olmayan uzun-dönem davranışlardır. Zaman sonsuza giderken sistemin izlediği yol sabit noktalara veya yörüngelere oturmaz. Sistemdeki düzensiz davranışlar rasgele girişlerden değil, sistemin lineer olmayan yapısından ileri gelir. Başlangıç koşullarına gösterilen yüksek duyarlılık nedeniyle yakın yörüngeler birbirinden hızla uzaklaşır. Bu da sistemin uzun dönemli davranışının kestirimini imkansız hale getirir [1].

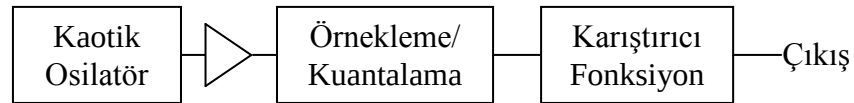
1.2 Uygulama Alanları

Kaotik davranış gösteren osilatörler birçok alanda kullanılmaya başlanmıştır. Bunlara örnek olarak rasgele sayı üretimi, şifreleme, mesaj gizleme ve kaotik haberleşme teknikleri gösterilebilir [5-11].

1.2.1 Rasgele Sayı Üretimi

Günümüzde rasgele sayılara, dolayısıyla da rasgele sayı üreteçlerine olan ihtiyaç gittikçe artmaktadır. Güvenlik uygulamaları ve simülasyonlarda kullanılmak üzere tekrarlanamaz ve önceden söylenemez rasgele sayı dizileri oluşturulması büyük öneme sahiptir. Rasgele anahtarlar üreten güvenlik sistemleri, elektronik devre tasarımında kullanılan Monte Carlo simülasyonları ve hava tahmin simülasyonları bu uygulamalara örnek olarak gösterilebilir [5,6]. Rasgele sayılar çoğunlukla çekirdek (seed) adı verilen bir başlangıç değerine bazı sayısal işlemler uygulanmasıyla üretilmektedir. Aynı çekirdek kullanıldığında rasgele sayı üretme işlemi tekrarlanabilir bir hal alır ve çıkışı gerçek anlamda rasgele olmaz.

Kaotik bir süreçten yararlanılarak tekrar edilemeyen değerlere ulaşmak mümkündür. Bu şekilde, bir kaotik osilatörün değerlerinin örneklenmesi ve kuantalanması prensibiyle oluşturulan yapı, rasgele sayı üretici olarak kullanılabilir (Şekil 1.1). Kaotik sürecin mevcut durumunun tahmin edilmesi mümkün olmayacağından çekirdek de aynı şekilde tahmin edilemez olacaktır. Her rasgele sayının kendine özel bir çekirdeğe sahip olması ise çıkışın gerçek anlamda tekrar-edilemez olmasını sağlar. Üretilen bit dizisindeki korelasyonun daha da azaltılması için, örneklemeyle elde edilen değerlere bir karıştırıcı fonksiyon uygulanabilir. Bütün bunların tek bir kırmık üzerinde üretilmesi, sistem parametrelerinin çok azının dışarıdan gözlemlenebilir olması nedeniyle, güvenliği önemli ölçüde arttıracaktır. Bu kırmığın sıradan bir bilgisayarın alt sistemi olarak kullanılması ise, bilgisayarda yapılacak simülasyonlar için kolayca rasgele sayı üretebilmeyi mümkün kılar.



Şekil 1.1: Kaotik Rasgele Sayı Üretici Blok Diyagramı

Kaotik osilatörden örnekleme ile elde edilen değerler 1 veya 0 şeklinde kodlanır. Kaotik sürece uygulanan örnekleme işlemi osilatörün temel frekansına göre çok küçükse, örneklerin korelasyonu çok düşük olacaktır. Daha yüksek çıkış hızı istendiğinde, daha hızlı örnekleme ihtiyacı duyulur. Bu durumda bile düşük korelasyonlu değerler elde etmek için uygun karıştırıcı fonksiyonlar kullanılabilir. Ayrıca karıştırıcı fonksiyonlar, kuantalama sonrasında ortaya çıkabilecek bitler arası dengesizliğin giderilmesini de mümkün kılmaktadır [7].

1.2.2 Şifreleme

Günümüzde bilginin güvenli şekilde bir noktadan diğerine iletilmesi amacıyla da kaotik süreçlerden yararlanılmaktadır. Mesajın ASCII karakterlerinden oluştuğunu düşünelim. Kaotik çekici $\varepsilon = (V_{\max} - V_{\min}) / S$ genişliğinde S adet bölüme ayrılır (Şekil 1.2). $V_{\max}$ ve $V_{\min}$ değerleri çekicinin tümünü kapsayabileceği gibi, çekicinin bir kısmını içine alacak şekilde de yerleştirilebilir.

Alfabe Elemanı	Bölüm Numarası	Aralık Pozisyonu
*	S	$X_{\min} + S\varepsilon$
@	S-1	$X_{\min} + (S-1)\varepsilon$
#	S-2	$X_{\min} + (S-2)\varepsilon$
\$	S-3	$X_{\min} + (S-3)\varepsilon$
	•	$X_{\min} + (S-4)\varepsilon$
	•	
b	4	$X_{\min} + 4\varepsilon$
a	3	$X_{\min} + 3\varepsilon$
/	2	$X_{\min} + 2\varepsilon$
%	1	$X_{\min} + \varepsilon$
		$X_{\min}$

Şekil 1.2: Kaotik çekicinin ($V_{\min}$, $V_{\max}$) aralığının S adet bölüme ayrılması

Örneğin $X_{n+1} = bX_n(1 - X_n)$ lojistik dönüşümünde (logistic map) b parametresi sistemin kaotik davranış göstermesini sağlayacak şekilde seçilsin. Şifrelenmesi istenilen karakter için, sistemin bir X_0 başlangıç durumundan başlayarak ilişkilendirilmiş kısma gelmesi için gerekli iterasyon sayısı, bu karakterin şifrelenmiş halini oluşturur. Şifrenin çözülebilmesi için S aralıkları, X_0 başlangıç koşulu ve b kontrol parametresi bilinmelidir. Şifrelenecek her karakter için kullanılacak başlangıç durumu, bir önceki karakterin şifrelenmesinde son iterasyon ile gelinen noktadır.

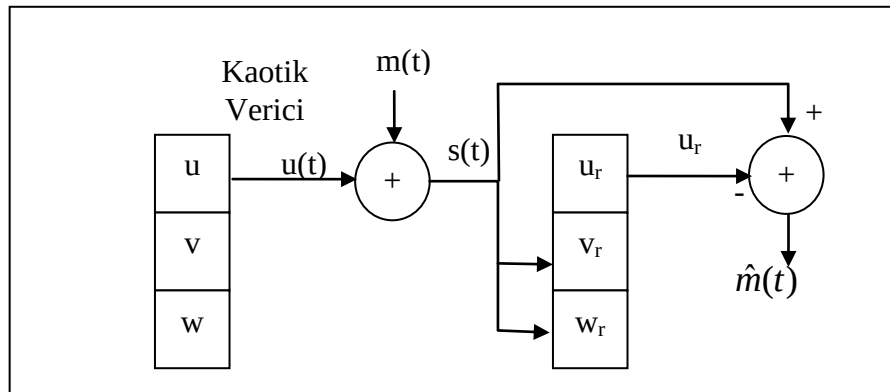
Bu algoritmada dikkat edilmesi gereken nokta, çekicinin bir başlangıç durumundan herhangi bir aralığa gelebilmesi için gerekli iterasyon sayısının 65532 altında kalmasıdır. Bu şekilde üretilen şifre, sadece bir byte veri kullanılarak iletebilir. Örnek olarak, $b=3.8$ değeri için çekicinin (0.2, 0.8) aralığı 256 bölüme ayrılmıştır. Her bir $\varepsilon=0.0023475$ genişliğindeki aralık, ASCII alfabesinin 256 karakterinden birine karşı düşürülmüştür. Bu değerler için 65532 iterasyon uzunluğunda bir yörüngenin, 256 aralığın her birinden en az 60 kez geçtiği belirtilmektedir [8].

1.2.3 Mesaj Gizleme

Kaotik sistemlerin senkronizasyon özelliğinden faydalanarak, haberleşmede mesaj işaretlerinin kaotik işaretler içine gizlenebileceğini gösterilmiştir. Bir kaotik sistem, sürücü ve yanıt alt-sistemleri olmak üzere iki kısma ayrılabilir. Yanıt alt-sisteminin bütün Lyapunov üstelleri negatif olduğunda bu iki alt-sistem arasında nümerik senkronizasyon gerçekleştiği belirtilmiştir [9]. Örneğin Lorenz Sistemi bu şekilde iki kısma ayrılabilir ve bu kısımlardan her biri, herhangi bir başlangıç koşulundan başlayarak senkron çalışır hale gelebilir. (1.1) denkleminde Lorenz Sistemi, durum değişkenleri ölçeklenmiş şekilde verilmektedir.

$$\begin{aligned}\dot{u} &= \sigma(v - u) \\ \dot{v} &= ru - v - 20uv \\ \dot{w} &= 5uv - bw\end{aligned}\quad (1.1)$$

İşaret maskeleye işleminde, vericide $m(t)$ işaretine gürültüyü andıran maskeleye işareti eklenir (Şekil 1.3). İletilen işaret $s(t)=m(t)+u(t)$ biçimindedir. Alıcı yapı verici ile senkronize çalışmaya başladığında $u_r(t) \cong u(t)$ olur. Alıcı tarafta senkronizasyondan yararlanılarak oluşturulan maskeleye işaretinin toplam işareten çıkarılmasıyla $\hat{m}(t) = s(t) - u_r(t)$ işareti elde edilir.



Şekil 1.3: İşaret Maskeleye Yapısı

Maskeleye metodu kendi başına bir kodlama veya modülasyon yapmaz çünkü fiziksel kanal için uygun bir işaret sağlamamaktadır. Bu yüzden $s(t)$, iletimden önce bir modülatöre gönderilmelidir. Ayrıca maskeleyenin uygulanabilmesi için, mesaj işaretinin güç seviyesinin, $u(t)$ işaretinin güç seviyesinden önemli ölçüde küçük olması gerekmektedir. 20 dB'lik bir fark ile tatminkar sonuçlar elde edilebileceği belirtilmiştir [10].

1.2.4 Kaotik Haberleşme

Kaotik haberleşme iletişim araştırmalarında oldukça yeni bir konudur. Bu konuyu ilginç hale getiren, kaotik davranışın karmaşık oluşu fakat basit dinamik sistemler ile elde edilebilmesidir. Kaotik işaretler düzensiz, periyodik olmayan, korelasyonsuz, geniş bantlı işaretler olup uzun süreli davranışlarının kestirimi imkansızdır. Bu özellikleri ile, dağınık-spektrumlu haberleşme ve şifreli haberleşme uygulamalarının gereksinimlerini karşılamaktadırlar.

Dağınık-spektrumlu haberleşmenin incelenmesindeki temel fikir, bilgi işaretinin dar bantlı işaretleri etkileyen faktörlere karşı daha az duyarlı hale getirilebileceğidir. Bu bozucu faktörlere örnek olarak filtreleme etkilerini ve periyodik işaretlerle sınırlı frekans aralığında gerçekleştirilecek girişimi gösterebiliriz. Ayrıca işaretin gücünü geniş bir frekans aralığına yaymak, işareti gürültü eşiğine yakın seviyelerde iletmeyi mümkün kılmaktadır. Gürültü eşiği seviyesinde gerçekleştirilen iletişimin bir gizlilik sağlayacağı da gerçektir. Aynı frekans aralığının birden fazla kullanıcı tarafından paylaşıldığı uygulamalarda da spektral dağıtma işleminden yararlanılmaktadır. Bu özellik ayrıca, kullanılan frekans bandındaki iletim gücünün sınırlı olduğu durumlarda kullanışlı olmaktadır.

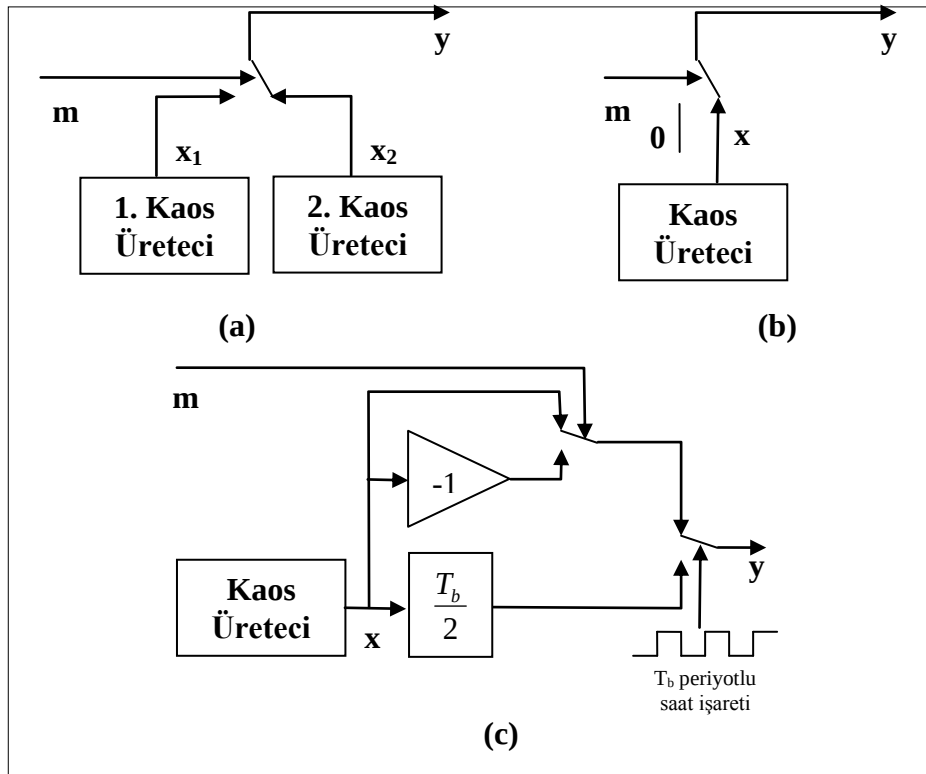
Haberleşme alanında kaosun kullanım alanları, kaotik davranışın şu üç temel karakteristik özelliği göz önüne alınarak incelenebilir:

- **Geniş-Bant Özelliği:** Kaotik işaretler yapıları itibariyle periyodik olmayan ve sürekli spektruma sahip işaretlerdir. Çoğunlukla spektrumları geniş bir aralıkta kayda değer güçte işaretler barındırmaktadır. Bu yüzden kaotik işaretler için geniş bantlı adlandırması uygun olmaktadır. Haberleşmede geniş bantlı işaretler, dar-bant işaretlerde etkili olan kanal kusurlarına karşı kullanılmaktadır.
- **Karmaşıklık Özelliği:** Kaotik işaretler karmaşık ve düzensiz yapıya sahiptir. Başlangıç değerlerindeki ufak bir sapma sonucu kaos üretici çok farklı bir yörüngeye sahip işaretler üretecektir. Bu yüzden üreticinin yapısının belirlenmesi ve ürettiği işaretin uzun zaman aralıklarında kestiriminin yapılması çok zordur.
- **Ortogonalite Özelliği:** Kaotik işaretler periyodik değildir ve çok hızlı bir şekilde kaybolan otokorelasyon fonksiyonlarına sahiptirler. Farklı veya aynı üreticilerden elde edilen farklı ilk koşullara sahip işaretler korelasyonsuz, yani ortogonal kabul edilir.

Kaosun dağınık-spektrumlu haberleşme alanında uygulanması kanal kodlaması ve modülasyon konularını kapsar. Kanal kodlaması ve modülasyon işlemleri birlikte, mesaj işaretinin fiziksel kanalda iletmeye uygun hale getirilmesini sağlayacak tek bir işlem olarak tanımlanabilir [11].

1.2.4.1 Verici Yapıları

Kanal kodlaması için statik ve dinamik kodlayıcılar mevcuttur. Statik kodlayıcılar belleksiz olup, bir andaki kodlama daha önceki kodlamalardan bağımsızdır. Bu metotta taşıyıcı işaret, mesajdan bağımsız bir kaotik kaynak tarafından sağlanır.



Şekil 1.4: Kaos haberleşmesinde kullanılan (a) CSK, (b) COOK, (c) DCSK yapıları

Kaotik haberleşme tekniklerine gösterilebilecek ilk örnek kaos kaydırmalı anahtarlama (chaos shift keying, CSK) yapısıdır. CSK bir sayısal modülasyon yöntemidir. N seviyeli bir mesaj işaretinin anlık değerine göre, N farklı karakteristik özelliğe sahip kaos üreticilerinden birinin işareti olan $x_i(t)$ ($i = 1, \dots, N$) kanaldan iletilir (Şekil 1.4a). Modüle edilmiş işaret şu şekildedir:

$$y(t) = \begin{cases} x_1(t), & m(t) = m_1 \\ x_2(t), & m(t) = m_2 \\ \dots & \\ x_N(t), & m(t) = m_N \end{cases} \quad (1.2)$$

CSK'nın özel bir durumu kaotik aç-kapa anahtarlama'sıdır (chaotic on-off keying, COOK). Bu durumda, ikilik tabandaki mesaj işaretinin değerine göre açılıp kapanan bir kaos üretici mevcuttur (Şekil 1.4b). İletilen işaretin ifadesi aşağıdaki gibidir:

$$y(t) = \begin{cases} 0, & m(t) = m_1 \\ x(t), & m(t) = m_2 \end{cases} \quad (1.3)$$

Klasik haberleşme yöntemlerinin yanı sıra kaos haberleşmesinde de kullanılan referans iletimi (transmitted reference method, TR) yaklaşımında referans farklı bir kanaldan gönderilmektedir. Bu farklı kanal farklı bir iletken, frekans bandı veya zaman aralığı olabilir. Böylelikle referansın alıcıda tekrar üretilmesi gerekmez. Saklanmış referans yönteminde (stored reference method, SR) ise referans alıcıda saklıdır veya oluşturulur. Bu yöntemde iletim kapasitesinin yarısı referans iletimi için kullanılmadığından aynı kaynaklar ile iki kat hız elde edilebilmektedir [11].

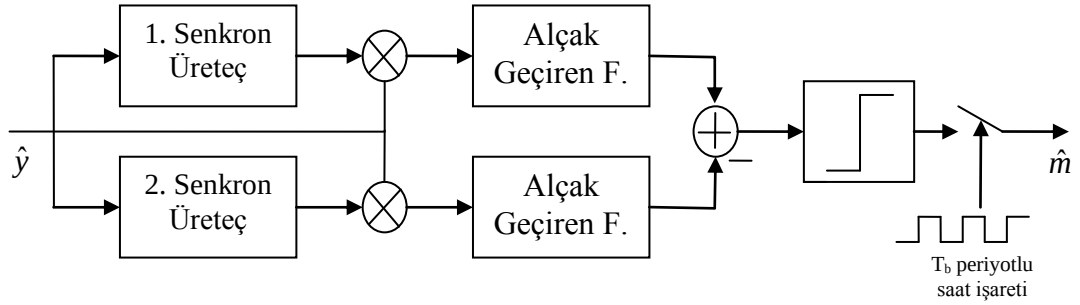
Mesaj taşıyıcısı yeniden üretilmesi kolay olmayan karmaşık bir işaret ise referans iletimi tekniği kullanılabilir. Bu teknik diferansiyel kaos kaydırmalı anahtarlama ile (differential chaos shift keying, DCSK) gerçekleşir (Şekil 1.4c). DCSK'da iki farklı kanal, zaman bölümlenmesi ile oluşturulur. Her mesaj sembolü için önce referans işaret, ardından da mesaj işaretini taşıyan modüle edilmiş referans iletilir. Sembol süresi T_b olan bir iki-tabanlı mesaj akışında, iletilen işaret şu şekilde ifade edilebilir:

$$y(t) = \begin{cases} x(t), & kT_b \leq t \leq \frac{2k+1}{2}T_b \\ x\left(t - \frac{T_b}{2}\right), & \frac{2k+1}{2}T_b \leq t \leq (k+1)T_b, \quad \forall k \in \mathbb{Z} \\ -x\left(t - \frac{T_b}{2}\right), & \frac{2k+1}{2}T_b \leq t \leq (k+1)T_b \\ m(t) = m_1 \\ -x\left(t - \frac{T_b}{2}\right), & \frac{2k+1}{2}T_b \leq t \leq (k+1)T_b \\ m(t) = m_2 \end{cases} \quad (1.4)$$

1.2.4.2 Alıcı Yapıları

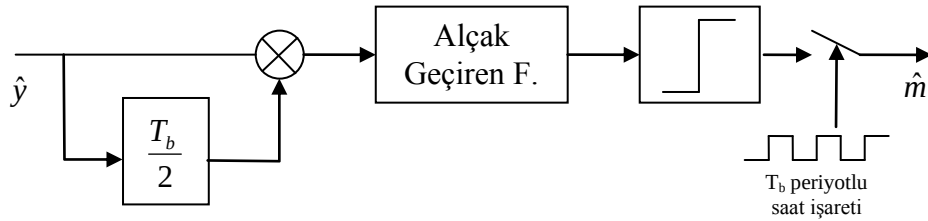
Kaotik haberleşmede kullanılan alıcı, alınan mesajı bir referans işareti kullanmak yada istatistik incelemesini yapmak suretiyle demodüle eder. Referans işaret korelasyon alıcılarında ve kaotik maskelemede çıkarma işlemi için kullanılır. Referanslar, vericideki üretece senkronize çalışan yerel üreteç tarafından sağlanabilir. Aksi halde, referans iletimi metodu ile mesajı taşıyan işaretin yanında referansın da alıcıya gönderilmesi gerekecektir.

Şekil 1.5’de CSK korelasyon alıcısı için bir örnek verilmiştir. Alıcıda bulunan iki adet senkronize olabilen sistem, vericideki kaotik sistemlerle eşleşir. Bunlardan biri kendi işareti olan x_i ile uyarılırsa, senkronize olarak korelasyonlu $\hat{x}_i$ işaretini üretir. Diğer sistem ise senkronize olmaz ve daha az korelasyonlu bir işaret üretir. Korelasyon değerlerinden büyük olan, iletilen mesajın sembolünü belirler.



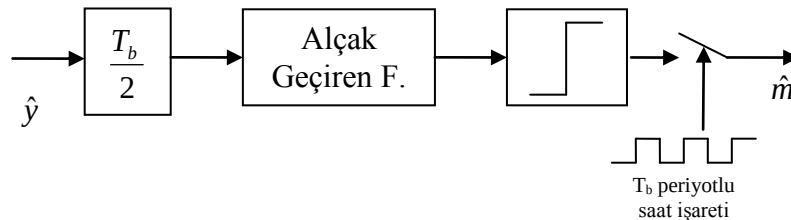
Şekil 1.5: Korelasyon-temelli CSK alıcısı

Şekil 1.6’da DCSK için korelasyon alıcısı yapısı gösterilmiştir. Referans, alınan işaretin T_b kadar geciktirilmesi ile elde edilir.



Şekil 1.6: DCSK alıcısı

Eğer alınan işaret mesaj işaretine birebir bağımlılık gösteriyorsa, işaretin istatistiksel incelemesini yapmak mümkündür. COOK yapısındaki aç kapa anahtarlama, alınan işaretin ortalama gücünde belirgin etki yapar. Bu sayede alıcıda, alınan güç ile öngörülen güç değeri karşılaştırılarak demodülasyon işlemi gerçekleştirilir [11].



Şekil 1.7: COOK alıcısı

2. OTONOM OLMAYAN KAOTİK OSİLATÖRÜN İNCELENMESİ

Bir devrede kaotik davranış görülebilmesi için devre en azından üçüncü dereceden olmalı ve bir lineer olmayan eleman bulunmalıdır. İkinci dereceden bir osilatöre eklenecek saat işareti ve lineer olmayan yapı ile de bu minimum şartlar sağlanabilir. 2. Bölüm’de bu prensip ile tasarlanan bir otonom olmayan kaotik osilatör yapısı incelenmiştir. Osilatörün kaotik oluşu nümerik hesaplamalar ile doğrulanmıştır. Devre parametrelerinin analizi sonucunda osilatörün zengin dinamik yapısını gösteren dallanma diyagramlarına ulaşılmıştır.

2.1 Otonom Olmayan Kaotik Osilatör

Günümüzde analog ve sayısal devreler aynı kırmık üstünde gerçekleştirilmektedir. Bu durum, kırmık üstünde çoğu zaman bir saat işaretinin mevcut olduğunu gösterir. Bu saat işareti ve bir lineer olmayan geri besleme kullanılarak iki kapasite elemanı bulunan herhangi bir ikinci dereceden sinüs osilatörünün kaotik osilatöre dönüştürülebileceği gösterilmiştir [12].



Şekil 2.1: İki kapasite elemanına sahip ikinci dereceden sinüs osilatörü yapısı

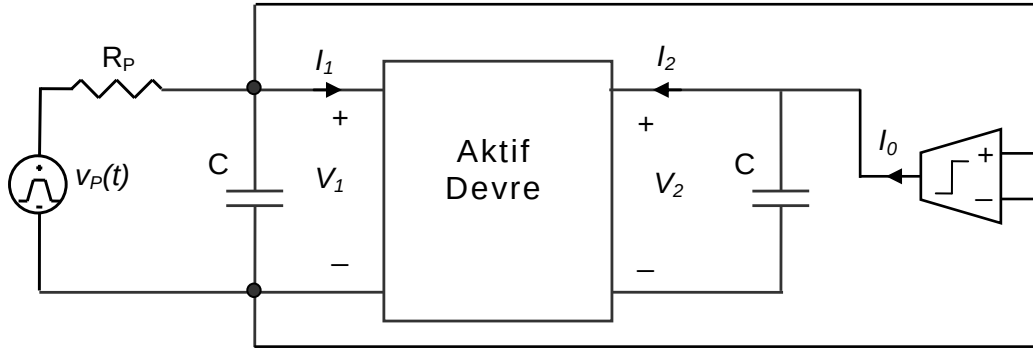
Şekil 2.1’de gösterilen yapıdaki aktif devrenin kısa-devre gösterimi ise şu şekildedir:

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (2.1)$$

$V_{1,2} = V_{C1,2}$ ve $I_{1,2} = -C_{1,2} \frac{dV_{1,2}}{dt}$ eşitliklerinin yerine konulması ve kapasitelerin eşit alınması sonucunda (2.2) denkleminde gösterilen durum denklemleri elde edilir.

$$\begin{aligned}\dot{V}_{C1} &= \frac{1}{C} [-g_{11}V_{C1} - g_{12}V_{C2}] \\ \dot{V}_{C2} &= \frac{1}{C} [-g_{21}V_{C1} - g_{22}V_{C2}]\end{aligned}\quad (2.2)$$

Devrenin ideal bir sinüs osilatörü olarak çalışması için $g_{11} = -g_{22}$ şartı sağlanmalıdır. Ayrıca $g_{12}g_{21}/g_{22}^2 = -(1+n)$ ve n bir sistem parametresi olmak üzere, osilatörün salınım frekansı $\omega_0 = (\sqrt{n}g_{22})/C$ olarak yazılabilir.



Şekil 2.2: Kaotik osilatör yapısı

Bir kaotik osilatör elde etmek için sinüs osilatörüne bir periyodik darbe kaynağı olan $V_p(t)$ ve lineer olmayan geribesleme eklendiğinde devre Şekil 2.2’de gösterilen duruma gelir. Geribesleme, V_{C1} gerilimi tarafından kontrol edilen, çıkış akımı I_o olan ve doymalı çalışan bir geçiş iletkenliği kuvvetlendiricisi ile gerçekleştirilmektedir. Yeni devre için durum denklemleri şu şekildedir:

$$\begin{aligned}\dot{V}_{C1} &= \frac{1}{C} \left[-\left(g_{11} + \frac{1}{R_p} \right) V_{C1} - g_{12}V_{C2} \right] + \frac{1}{R_p C} V_p(t) \\ \dot{V}_{C2} &= \frac{1}{C} [-g_{21}V_{C1} - g_{22}V_{C2}] + \frac{I_o}{C}\end{aligned}\quad (2.3)$$

$V_p(t) = V_p \operatorname{sgn}(\sin(\omega_p t))$, ω_p ise darbe kaynağının frekansdır. I_o ise şu şekilde ifade edilebilir:

$$I_o = I_{sat} \operatorname{sgn}(V_1) = \begin{cases} I_{sat}, & V_1 \geq 0 \\ -I_{sat}, & V_1 < 0. \end{cases}\quad (2.4)$$

Burada I_{sat} geçiş iletkenliği kuvvetlendiricisinin doyma akımını belirtmektedir.

$$X = -V_{C1}g_{22}^2 / I_{sat}g_{12} \quad (2.5a)$$

$$Y = V_{C2}g_{22} / I_{sat} \quad (2.5b)$$

$$\varepsilon = 1 / g_{22}R_p \quad (2.5c)$$

$$\beta = -V_p g_{22} / I_{sat}g_{12}R_p \quad (2.5d)$$

$$\phi = \omega_p C / g_{22} \quad (2.5e)$$

$$\tau = t(g_{22} / C) \quad (2.5f)$$

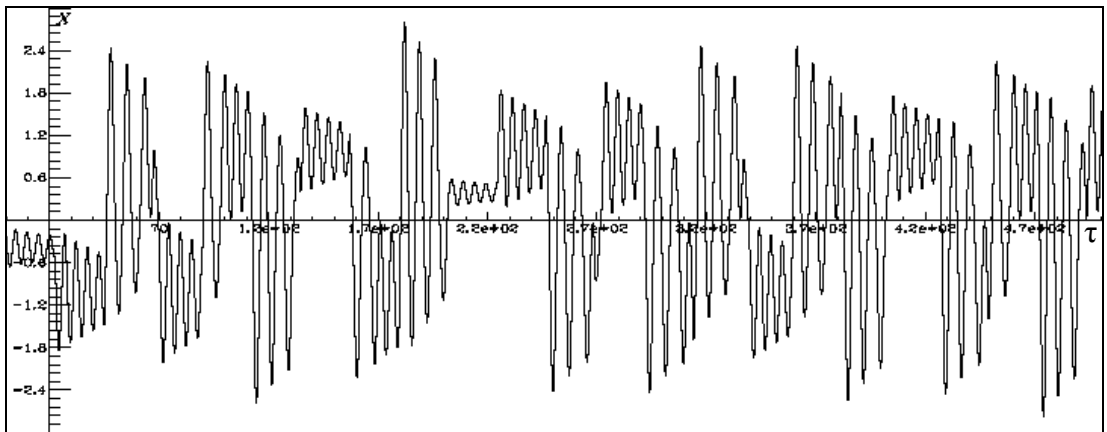
(2.5) denkleminde verilen değişken dönüşümleri sonucu elde edilen normalleştirilmiş diferansiyel denklem sistemi (2.6) denkleminde görülmektedir. Bahsedilen denklem sistemi, Bölüm 3’de gerçekleştirilecek kaotik osilatörün matematiksel ifadesi olup, bu tez çalışmasının temelini oluşturmaktadır.

$$\dot{X} = (1 - \varepsilon)X + Y + \beta p(\tau) \quad (2.6a)$$

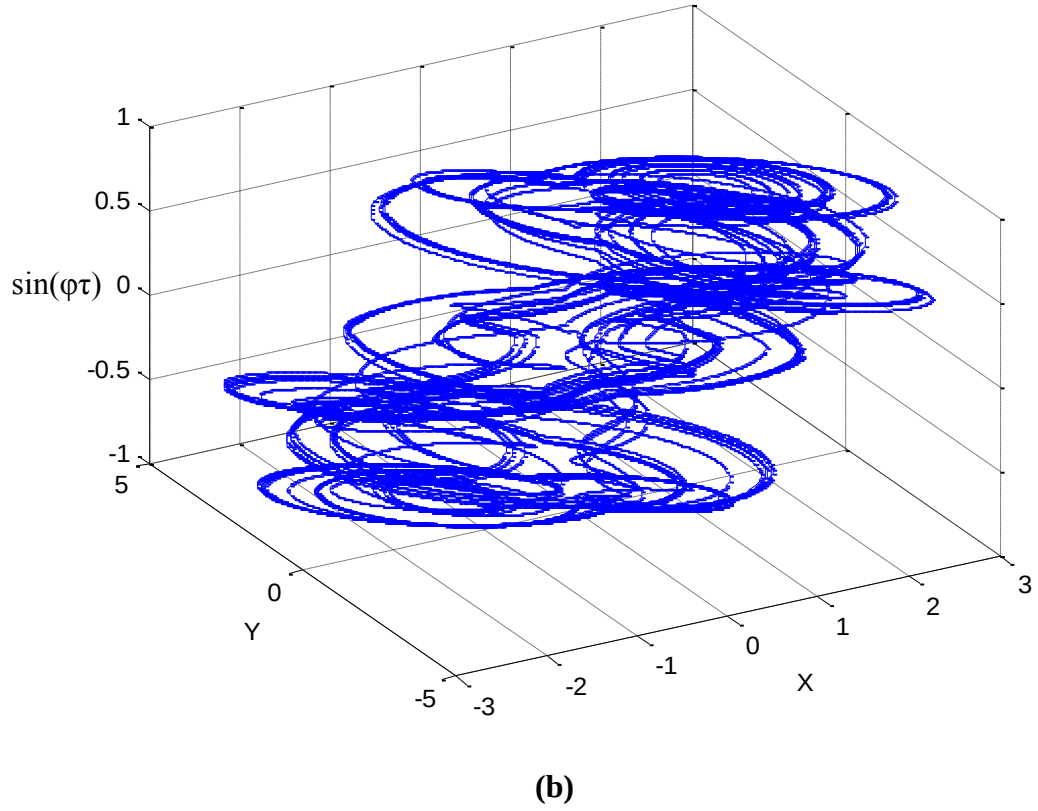
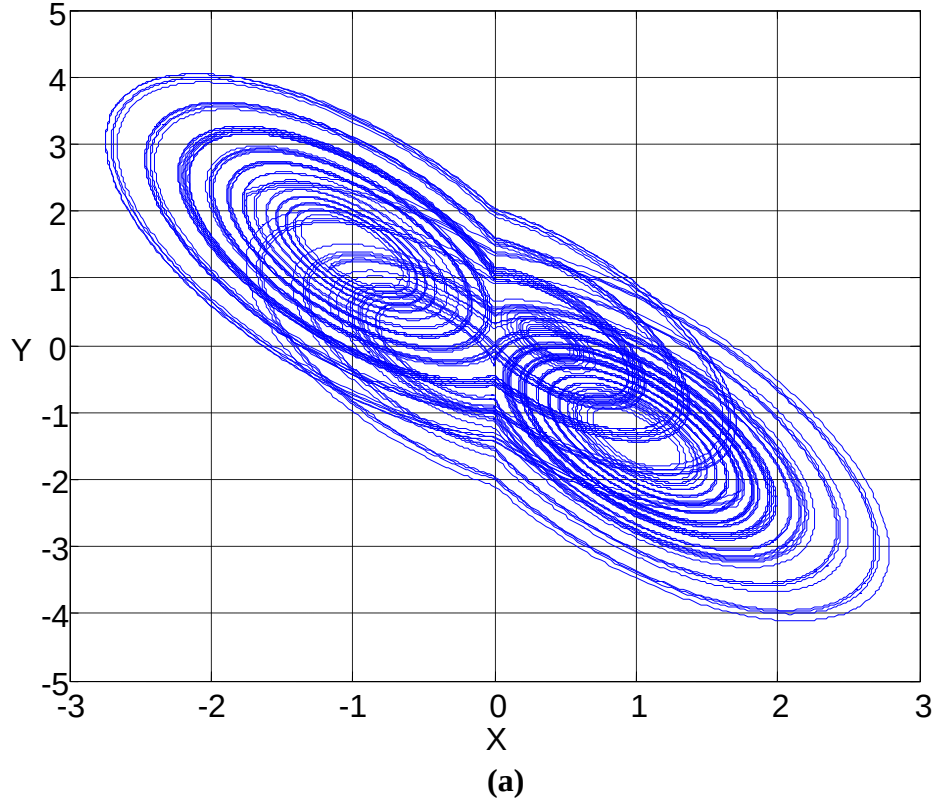
$$\dot{Y} = -(1 + n)X - Y + \text{sgn}(X)$$

$$p(\tau) = \text{sgn}(\sin \phi \tau) = \begin{cases} 1, & \sin \phi \tau \geq 0 \\ -1, & \sin \phi \tau < 0 \end{cases}, \quad \text{sgn}(X) = \begin{cases} 1, & X \geq 0 \\ -1, & X < 0 \end{cases} \quad (2.6b)$$

Bu diferansiyel denklem sistemi belli parametre aralıkları için kaotik davranış sergiler. $\varepsilon=0.05$, $n=1.4$, $\beta=0.44$ ve $\phi=0.14$ değerleri için nümerik analiz gerçekleştirilmiştir. Bu parametre değerleri için X durum değişkeninin ölçeklenmiş zamana göre değişimi Şekil 2.3’de, elde edilen kaotik çekici ise Şekil 2.4’de gösterilmiştir.



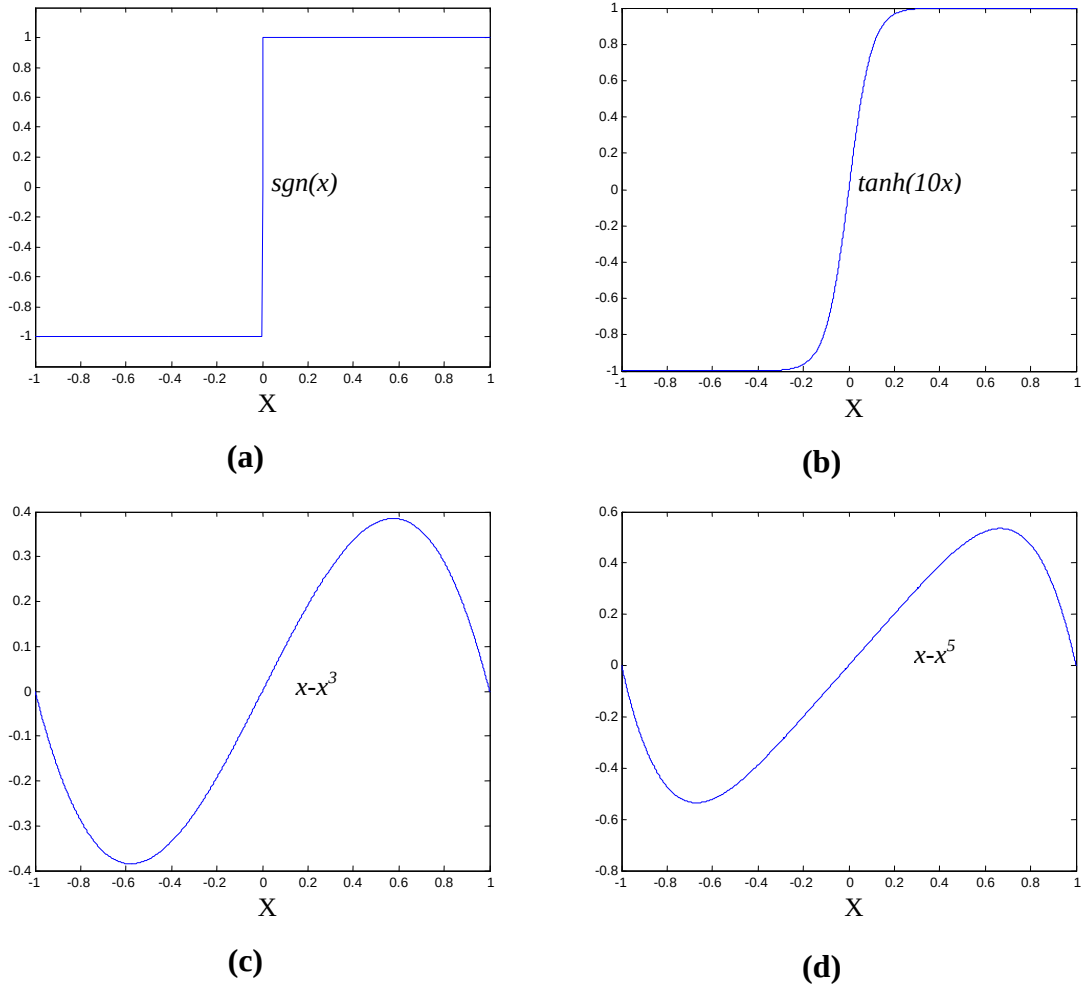
Şekil 2.3: X durum değişkeninin ölçeklenmiş zamana göre değişimi



Şekil 2.4: $(\varepsilon, n, \beta, \varphi)=(0.05, 1.4, 0.44, 0.14)$ parametre değerleri için elde edilen çekicinin (a) X-Y düzlemine izdüşümü (b) 3-boyutlu çizimi

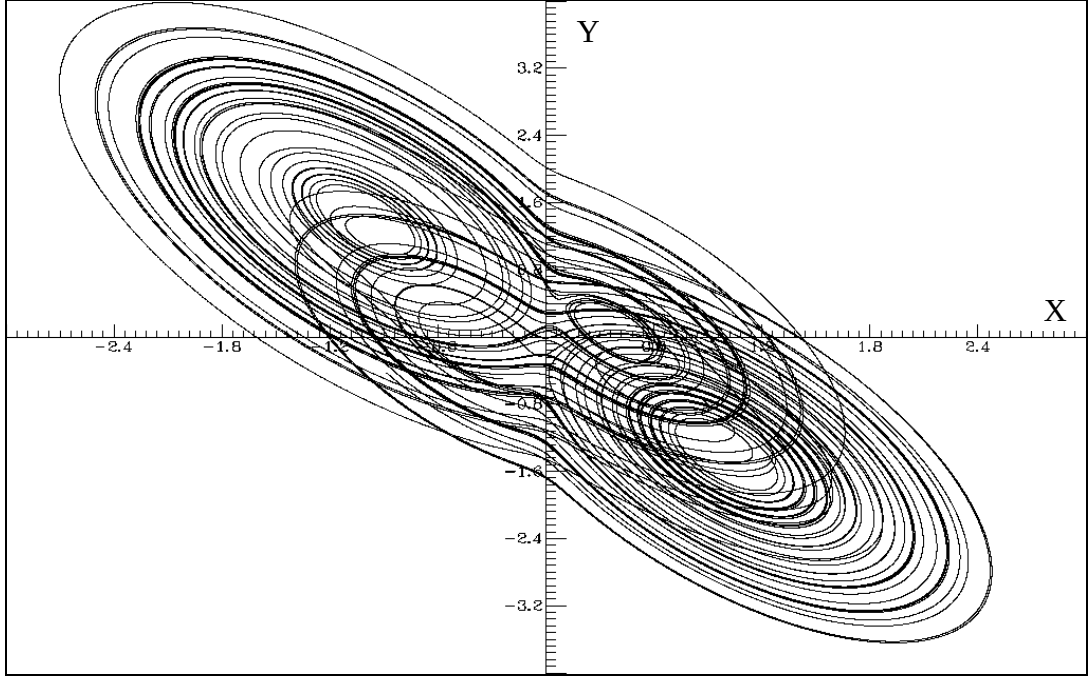
2.2 Kaotik Osilatörün Farklı Lineer Olmayan Fonksiyonlar ile Gerçeklenmesi

Lineer olmayan geribeslemede işaret fonksiyonu yerine $\tanh(10x)$, p tek sayı olmak üzere $x-x^p$ gibi farklı lineer olmayan fonksiyonlar kullanıldığında da kaos elde edilebildiği belirtilmiştir [12]. Şekil 2.5’de uygulanan lineer olmayan fonksiyonlar gösterilmiştir. Şekil 2.6, 2.7 ve 2.8’de ise, bu lineer olmayan fonksiyonların kullanılmasıyla elde edilen kaotik çekiciler verilmektedir.

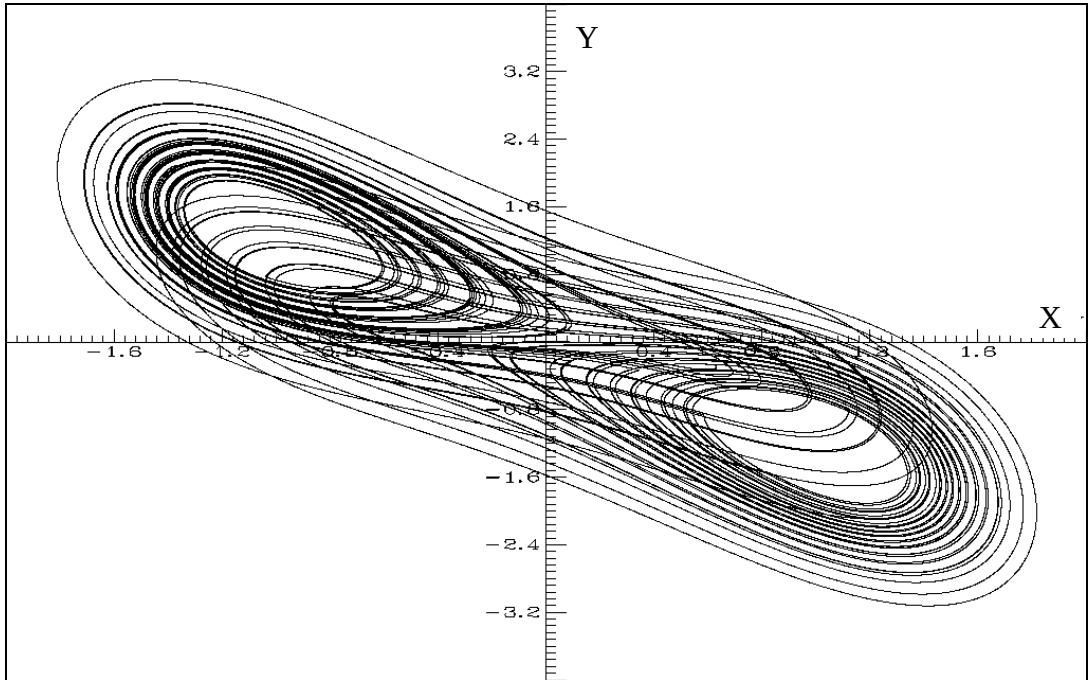


Şekil 2.5: Uygulanan farklı tipte lineer olmayan fonksiyonlar (a) işaret fonksiyonu, (b) $\tanh(10x)$, (c) $x-x^3$, (d) $x-x^5$

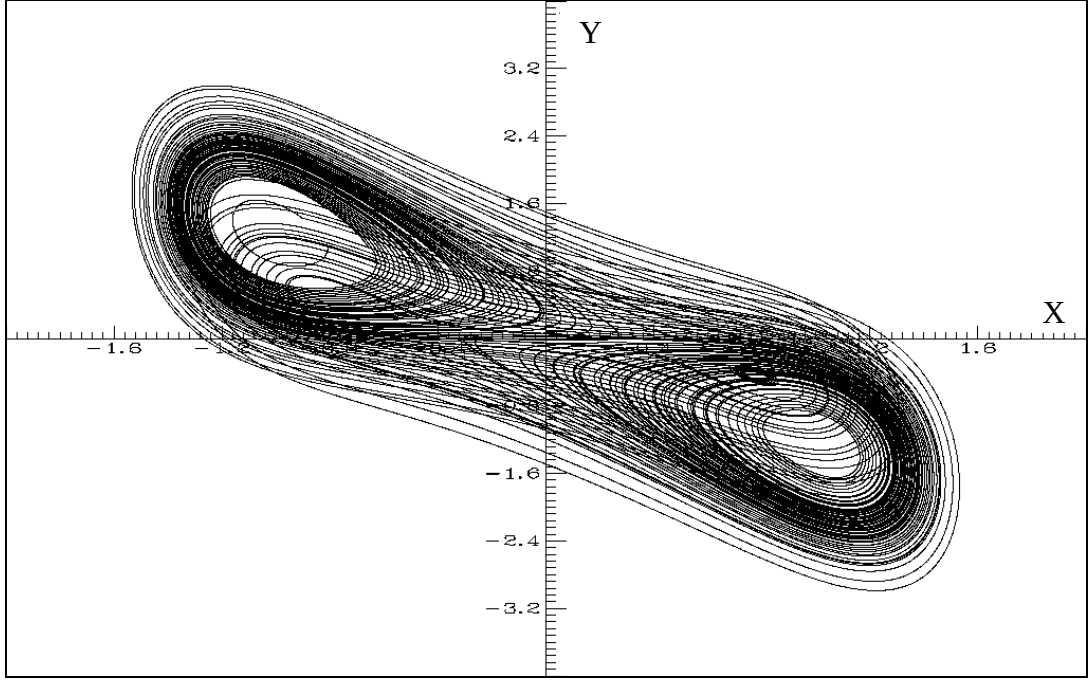
$\tanh(10x)$ fonksiyonu işaret fonksiyonuna yeterli yakınlıkta sonuç veren sürekli bir fonksiyondur. Şekil 2.6’da gösterilen, $\tanh(10x)$ kullanılmasıyla elde edilen çekicinin Şekil 2.4a’da verilen kaotik çekici ile benzerliği görülmektedir. Şekil 2.7 ve 2.8’deki çekiciler ise, görüntü itibariyle farklılaşmış olsalar da yine 4 sarmallı çekici yapısı korunmaktadır.



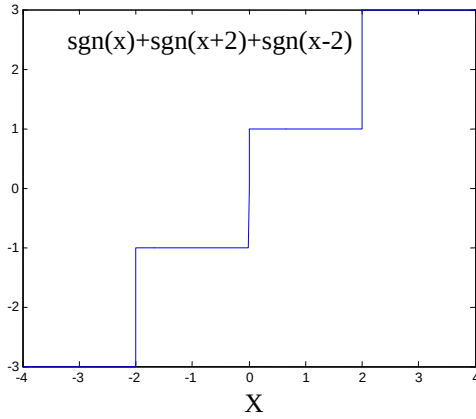
Şekil 2.6: $(\varepsilon, n, \beta, \varphi)=(0.05, 1.4, 0.44, 0.14)$ parametre seti için (2.6) denkleminde lineer olmayan fonksiyon olarak $\tanh(10x)$ kullanıldığında elde edilen kaotik çekici



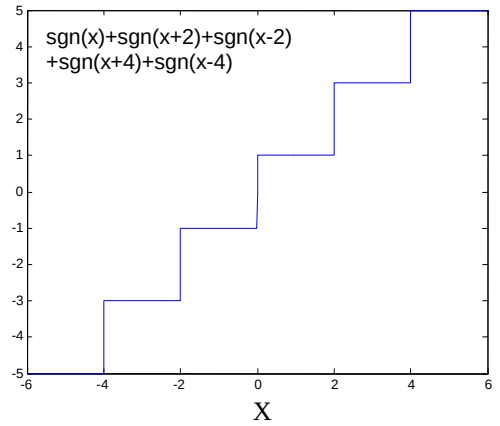
Şekil 2.7: $(\varepsilon, n, \beta, \varphi)=(0.05, 0.2, 0.3, 0.1)$ parametre seti için (2.6) denkleminde lineer olmayan fonksiyon olarak $x-x^3$ kullanıldığında elde edilen kaotik çekici



Şekil 2.8: $(\varepsilon, n, \beta, \varphi)=(0.05, 0.2, 0.3, 0.1)$ parametre seti için (2.6) denkleminde lineer olmayan fonksiyon olarak $x-x^5$ kullanıldığında elde edilen kaotik çekici



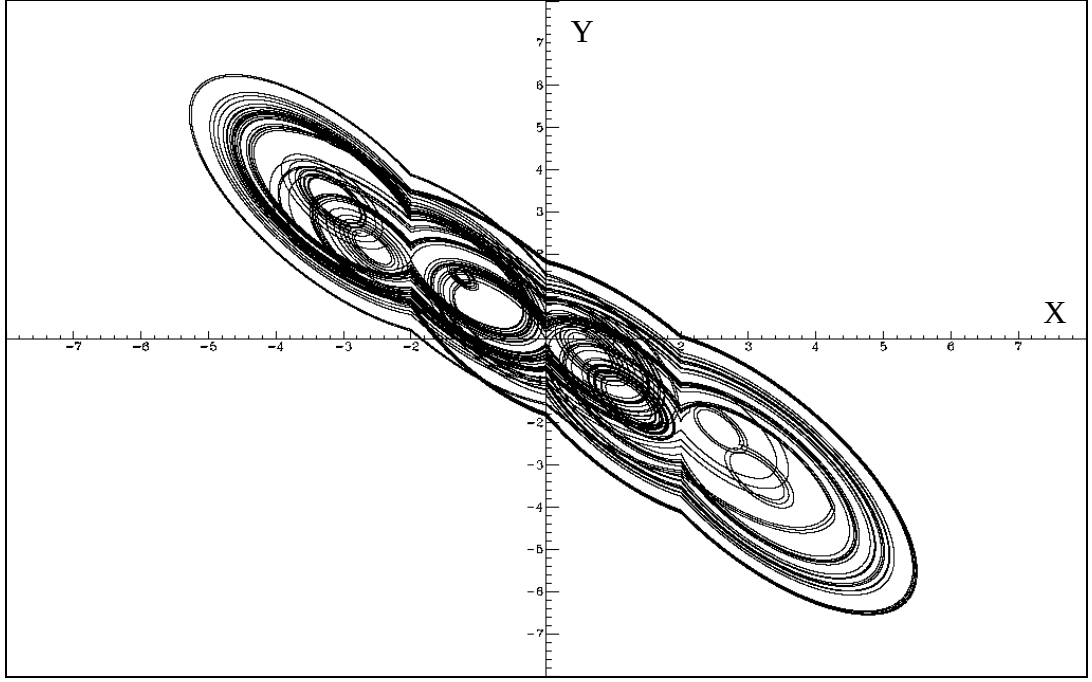
(a)



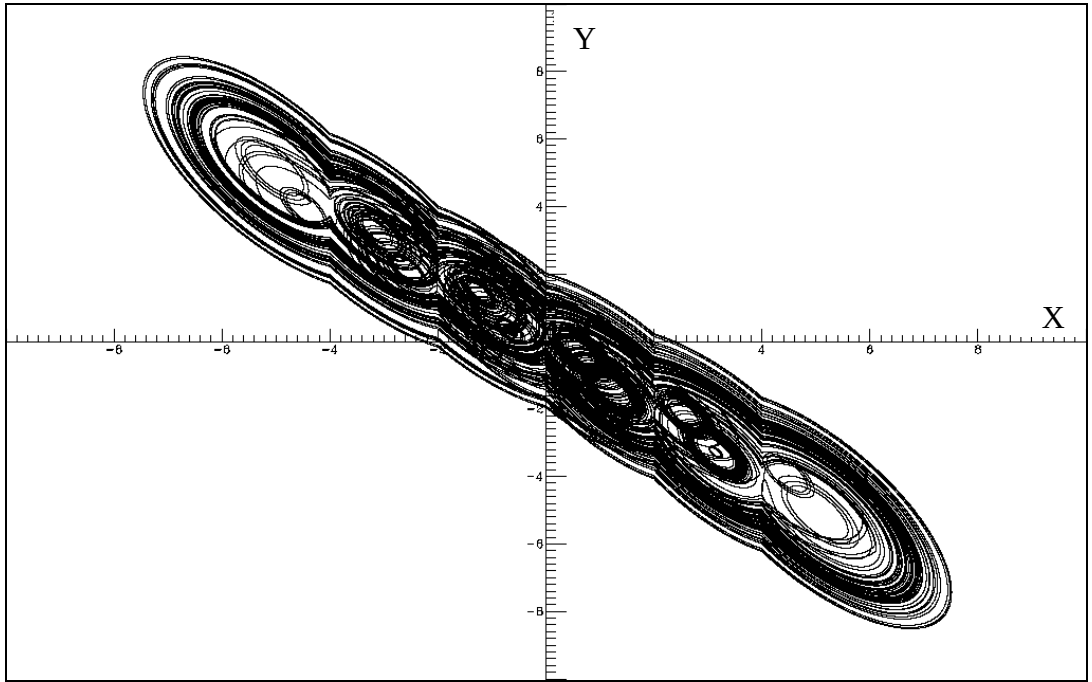
(b)

Şekil 2.9: Uygulanan farklı tipte lineer olmayan fonksiyonlar (a) 3 basamaklı işaret fonksiyonu, (b) 5 basamaklı işaret fonksiyonu

İşaret fonksiyonunun Şekil 2.9'da gösterildiği gibi daha yüksek katlı şekilde uygulanması, kaotik çekicinin daha çok sayıda sarmal içermesine neden olmaktadır. Bu yolla elde edilen kaotik çekiciler Şekil 2.10 ve 2.11'de verilmiştir. k basamaklı işaret fonksiyonu için $2(k+1)$ adet sarmala sahip çekici elde edilmiştir.



Şekil 2.10: $(\varepsilon, n, \beta, \varphi)=(0.05, 1, 0.3, 0.2)$ parametre seti için (2.6) denkleminde lineer olmayan fonksiyon olarak 3 basamaklı işaret fonksiyonu kullanıldığında elde edilen kaotik çekici



Şekil 2.11: $(\varepsilon, n, \beta, \varphi)=(0.05, 1, 0.3, 0.2)$ parametre seti için (2.6) denkleminde lineer olmayan fonksiyon olarak 5 basamaklı işaret fonksiyonu kullanıldığında elde edilen kaotik çekici

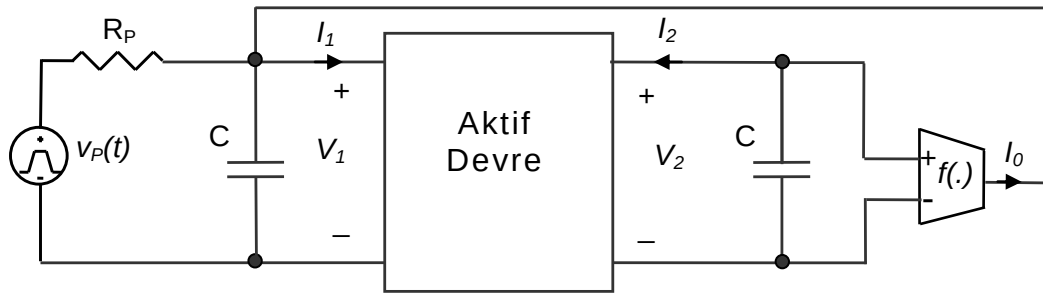
2.3 Otonom Olmayan Kaotik Osilatörün Duffing Osilatörü ile Benzerliği

Lineer olmayan dinamik sistemlerin temel özelliklerinin tümünü yansıtan en basit örneklerden birisi Duffing Osilatörü'dür. Bu sistem, van der Pol osilatörü ile birlikte, lineer olmayan osilasyon araştırmalarında en sık rastlanan örnek haline gelmiştir [13]. Duffing osilatörüne ilişkin denklemler

$$\frac{d^2x}{dt^2} + x + x^3 = f \cos(\omega t) \quad (2.7a)$$

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -dx_2 - x_1 - x_1^3 + f \cos(2\pi x_3) \\ \dot{x}_3 &= \omega / 2\pi \end{aligned} \quad (2.7b)$$

şeklinde verilebilir. Burada x_1 , x_2 ve x_3 faz uzayındaki koordinatları belirtmektedir. (2.6) denklemi ile belirtilen lineer olmayan sisteme uygulanan uyarma kaynağı sinüs işareti olarak alınıp, geribesleme tipi ve bağlandığı kapı değiştirilirse Duffing Osilatörü yapısının elde edileceği kolaylıkla görülebilir [12].



Şekil 2.12: Otonom olmayan kaotik osilatörün Duffing Osilatörüne dönüştürülmesi

Şekil 2.12'deki yapıdan elde edilen durum denklemleri şu şekildedir:

$$\begin{aligned} \dot{V}_{C1} &= \frac{1}{C} \left[- \left(g_{11} + \frac{1}{R_p} \right) V_{C1} - g_{12} V_{C2} \right] + \frac{1}{R_p C} V_p(t) + \frac{I_0}{C} \\ \dot{V}_{C2} &= \frac{1}{C} [- g_{21} V_{C1} - g_{22} V_{C2}] \end{aligned} \quad (2.8)$$

(2.5) denklemiyle belirtilen değişken dönüşümlerinin yapılması sonucu elde edilen sistem, $\alpha = g_{22} / g_{12}$ olmak üzere şu denklemlerle ifade edilebilir:

$$\begin{aligned} \dot{X} &= (1 - \varepsilon)X + Y + \beta p(\tau) + \alpha \operatorname{sgn}(Y) \\ \dot{Y} &= -(1 + n)X - Y \end{aligned} \quad (2.9)$$

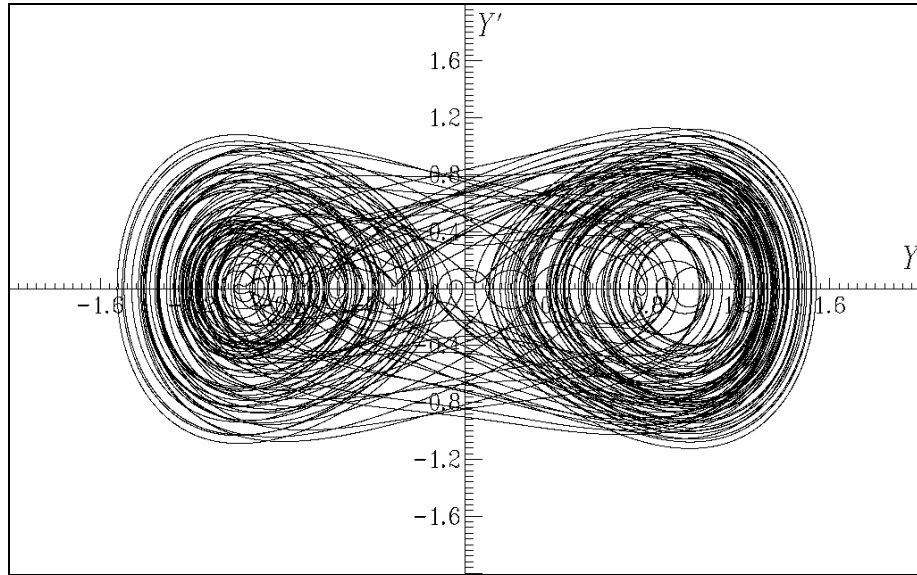
Lineer olmayan bileşen $f(Y)$ olarak genelleştirildikten sonra (2.9) denkleminin kapalı şekilde yazılmasıyla ortaya çıkan ifade şu şekildedir:

$$Y'' + \varepsilon Y' + (\varepsilon + n)Y + \alpha(1+n)f(Y) = -\beta(1+n)p(\tau) \quad (2.10)$$

Bu denklemde sürücü olarak kare dalga yerine sinüzoidal bir kaynak kullanılması ve lineer olmayan geri beslemenin $f(Y) = Y^3 - Y$ ile gerçekleştirilmesi sonucu Duffing Osilatörü elde edilir. $n = 1$ ve $\alpha = \beta = 1/2$ alınırsa (2.10) denklemi şu hale gelir:

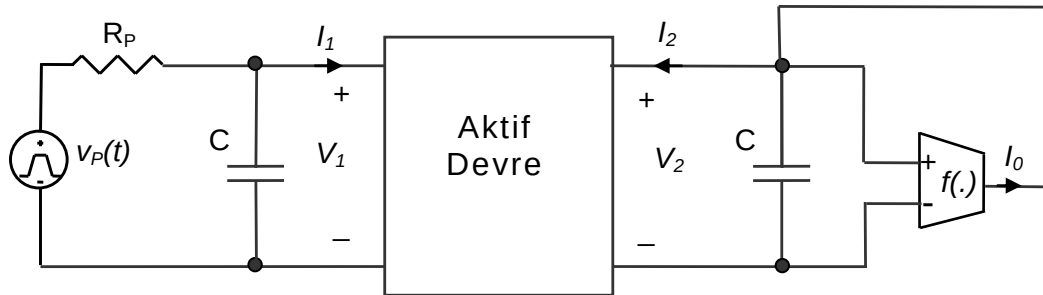
$$Y'' + \varepsilon Y' + \varepsilon Y + Y^3 = -A \sin \phi \tau \quad (2.11)$$

$(\varepsilon, \phi) = (0.01, 0.2)$ alınarak elde edilen Duffing Çekicisi Şekil 2.13'de gösterilmiştir.



Şekil 2.13: Duffing Osilatöründen elde edilen kaotik çekici

Diğer yandan, Şekil 2.12'de geribesleme akımının birinci kapı yerine ikinci kapıya verilmesi ve uygun parametre seçimi ile van der Pol osilatörüne yakın davranış gösteren bir devre yapısı elde edilebildiği ifade edilmiştir (Şekil 2.14).



Şekil 2.14: Otonom olmayan kaotik osilatörün van der Pol Osilatörüne dönüştürülmesi

2.4 Kaotik Osilatörün Melnikov Metodu ile İncelenmesi

Kaotik davranışı anlamak ve doğrulamak için mevcut az sayıda yöntemlerden biri Melnikov Metodu'dur. Bu yöntemin ana fikri, bir diferansiyel denklemin etkilenmiş (perturbed) çözümlerinin elde edilmesi için etkilenmemiş (unperturbed) durumun çözümlerinden yararlanılmasıdır [14]. Diferansiyel denklemimizi, bir Hamilton vektör alanı ve Hamilton olması şart olmayan küçük bir etkilenmenin (perturbasyon) toplamı olarak yazabildiğimizi varsayalım.

$$\begin{aligned} \dot{x} &= f(x) + \varepsilon g(x, t) \\ x &= \begin{pmatrix} u \\ v \end{pmatrix} \in R^2 \end{aligned} \quad (2.12)$$

Burada $f(x)$, R^2 üzerinde tanımlı Hamilton bir vektör alanı ve $\varepsilon g(x, t)$ Hamilton olması şart olmayan bir etkilenmeyi göstermektedir. Verilen fonksiyonların yeterince yüksek mertebeden türevleri mevcut ve $g(x, t)$ T-periyodiktir. Etkilenmemiş sistem $f_1 = \partial H / \partial v$ ve $f_2 = -\partial H / \partial u$ sağlanacak şekilde Hamilton olmalıdır. Kaotik davranışın görülmesi için $\varepsilon = 0$ durumunda sistemin p_0 eyer noktasını içeren bir homoklinik yörüngesi olmalı ve bu yörünge üstünde hesaplanan Melnikov Fonksiyonu $M(\tau_0)$ 'nin $\tau_0 \in [0, T]$ aralığında sıfır değerinden geçmesi gerekmektedir. Melnikov Fonksiyonu

$$M(\tau_0) = \int_{-\infty}^{\infty} f^0(\tau) \wedge g^0(\tau + \tau_0) d\tau \quad (2.13)$$

şeklinde tanımlıdır. Burada kullanılan $\wedge$ işlemi, a ve b vektörlerinin $a_1 b_2 - a_2 b_1$ şeklinde çarpımını göstermektedir. (2.6) denkleminde verilen sistemin Melnikov Metodu kullanımı için yeniden düzenlenmiş hali şu şekildedir:

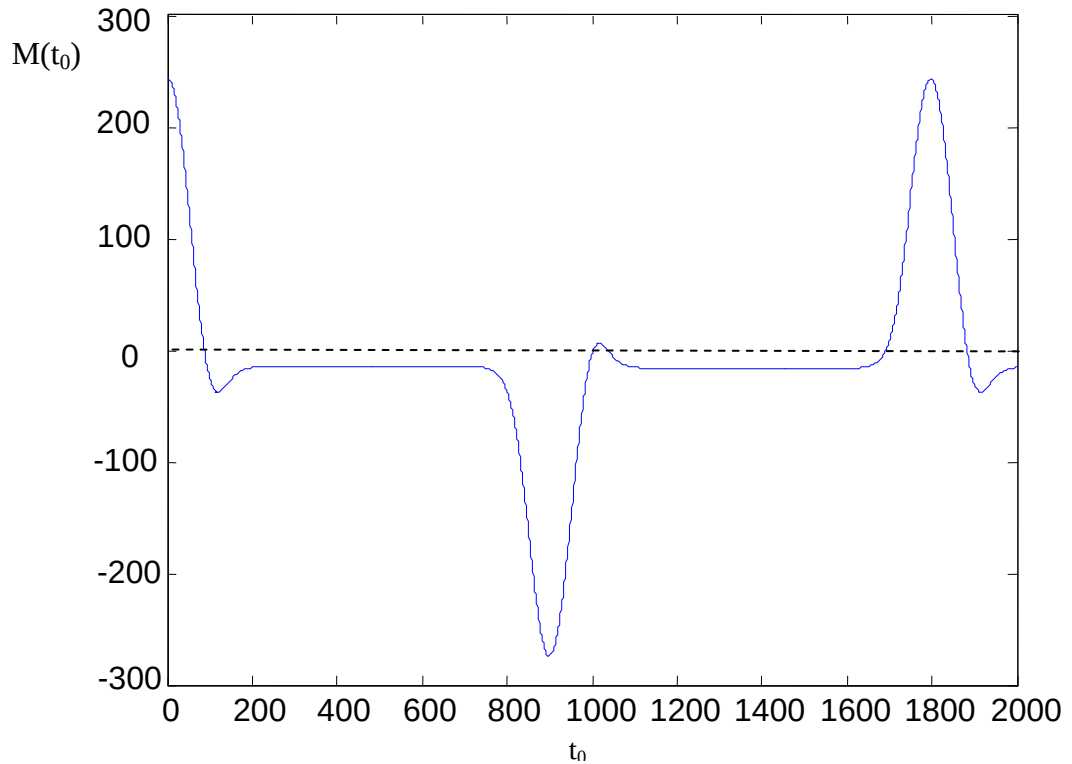
$$\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ -(1+n) & -1 \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} + \begin{pmatrix} \varepsilon(-X + (\beta/\varepsilon)p(t)) \\ \text{sgn}(X) \end{pmatrix} \quad (2.14)$$

Bu metodun kullanılabilmesi için sistemdeki yumuşak geçişe sahip olmayan fonksiyonların yerine, k büyük bir sayı olmak üzere, $\tanh(kX)$ fonksiyonu kullanılabilir. Bu yaklaşıklık ile $p(\tau) = \tanh(10 \sin \phi \tau)$ ve $\text{sgn}(X)$ yerine $\tanh(10X)$ alındığında elde edilen Melnikov Fonksiyonu (2.15) denkleminde verilmiştir.

$$M(\tau_0) = \int_{-\infty}^{\infty} [(1+n)X + Y - \tanh(10X)] \cdot \left[-X + \frac{\beta}{\varepsilon} \tanh(10 \sin \phi(\tau + \tau_0)) \right] d\tau \quad (2.15)$$

Nümerik hesaplamalar sonucunda Melnikov Fonksiyonu'nun $\tau_0 \in [0, T]$ için sıfırdan geçtiği görülmüştür (Şekil 2.15) ve sistemin kaotik davranış gösterdiği doğrulanmıştır. Nümerik hesaplamada kullanılan MATLAB kodu aşağıda verilmiştir:

```
e=0.05;beta=0.44;phi=0.14;n=1.4;
h=0:0.1:160; range=2000;
[t,x]=ode45('sist',[h],[0.45956500222544E-80 0.88814413736146E-80]);
plot(x(:,1),x(:,2))
for i=1:range
    t0=i*10/400;
    f1=-(1+n)*x(:,1)-x(:,2)+tanh(10*(x(:,1)));
    f2=x(:,1)-beta/e*tanh(10*sin(phi*(t+t0)));
    m(i)=sum(f1.*f2);end
plot(m)
function [y]=sist(t,y)
x=y; n=1.4;
y(1)=x(1)+x(2);
y(2)=-(1+n)*x(1)-x(2)+tanh(10*(x(1)));
```

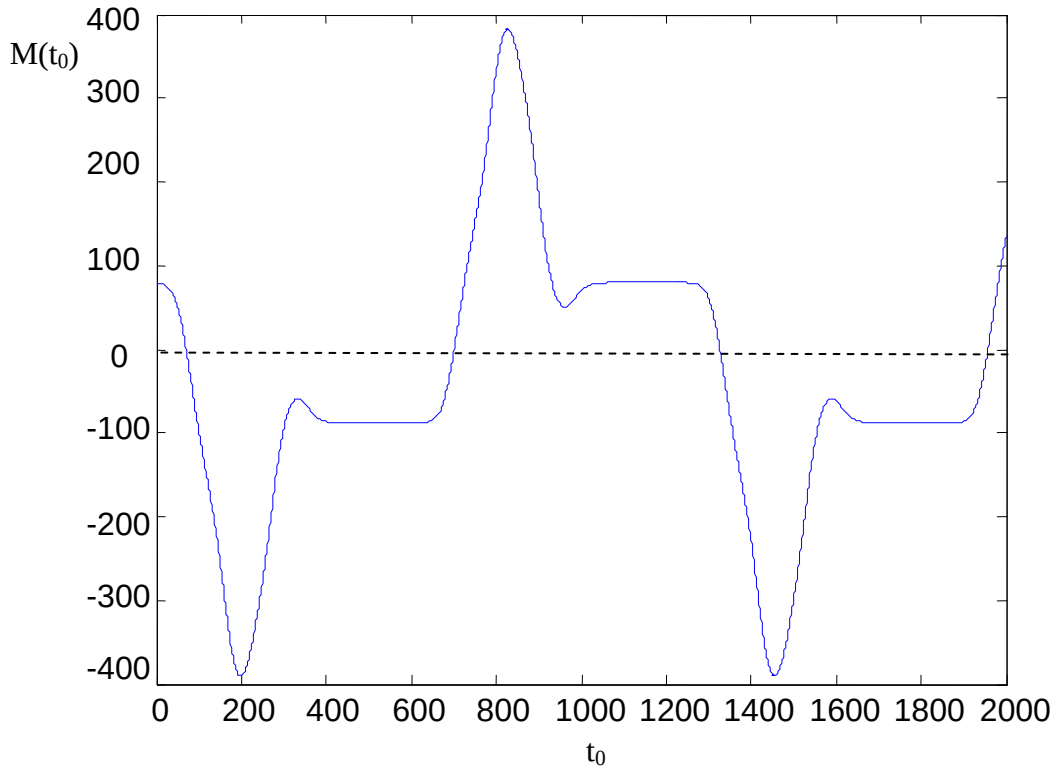


Şekil 2.15: Lineer olmayan fonksiyon olarak $\tanh(10x)$ kullanıldığı durum için Melnikov Fonksiyonunun nümerik çözümü

Benzer şekilde, lineer olmayan fonksiyon olarak 3 basamaklı işaret fonksiyonu uygulandığı durumda elde edilen kaotik çekici (Şekil 2.10) için Melnikov Metodu ile inceleme yapılmıştır. Bu amaçla kullanılan MATLAB kodu aşağıda verilmiştir. Şekil 2.16'da görüldüğü gibi, Melnikov Fonksiyonu sıfırdan geçmektedir ve bu durum için de kaotik davranış sergilendiği doğrulanmış olur.

```
e=0.05; beta=0.3; phi=0.2; n=1; h=0:0.1:155.6; range=2000;
[t,x]=ode45('sistx',[h],[4.472135954999579E-80 8.944271909999159E-80]);
figure(1); plot(x(:,1),x(:,2))
for i=1:range
    t0=i*10/400;
    f1=-(1+n)*x(:,1)-x(:,2) + tanh(10*(x(:,1))) + tanh(10*(x(:,1)-2)) +
    tanh(10*(x(:,1)+2));
    f2=x(:,1)-beta/e*tanh(10*sin(phi*(t+t0)));
    m(i)=sum(f1.*f2); end
figure(2); plot(m)

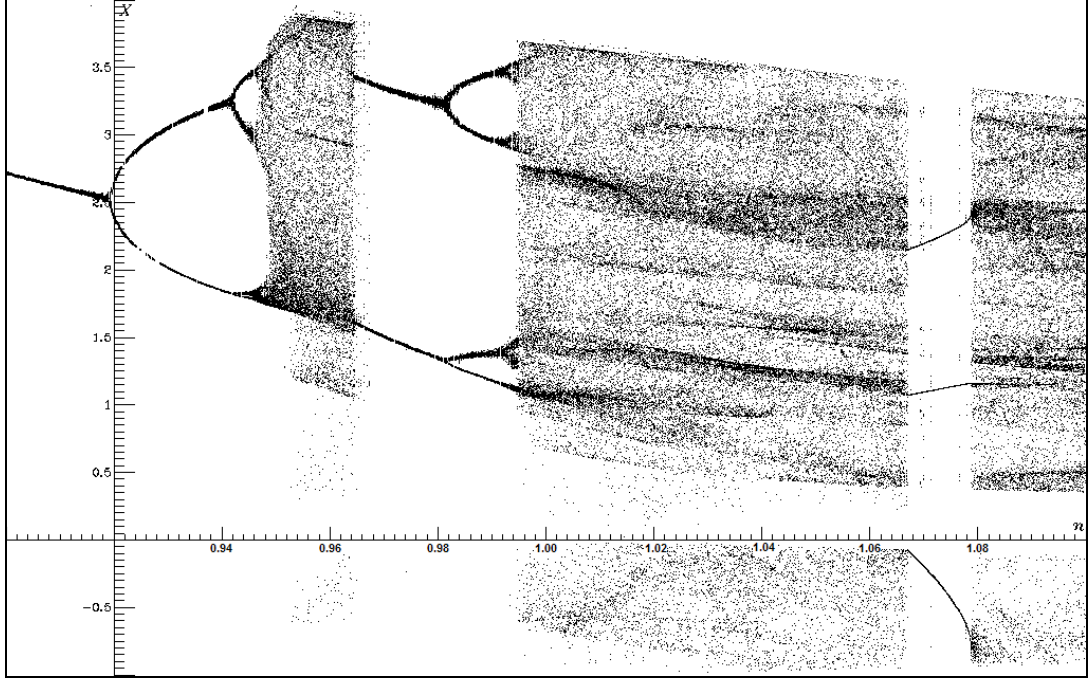
function [y]=sistx(t,y)
x=y;n=1.4;
y(1)=x(1)+x(2);
y(2)=-(1+n)*x(1)-x(2)+tanh(10*(x(1)))+tanh(10*(x(1)+1))+tanh(10*(x(1)-2));
```



Şekil 2.16: Lineer olmayan fonksiyon olarak 3 basamaklı işaret fonksiyonu kullanıldığı durum için Melnikov Fonksiyonunun nümerik çözümü

2.5 Dallanma Diyagramları

$x = f(x, \alpha)$ şeklinde n . mertebeden, α parametresine sahip bir sürekli zaman sistemini ele alalım. α küçük oranlarda değıştikçe sistemin limit kümesinin pozisyonunda da küçük değışimler oluşur. α 'daki ufak bir değışiklik limit kümede niteliksel bir değışime yol açıyorsa, bu değışime dallanma, bu değışimin gerçekleştiği α değerine ise dallanma değeri denir. Limit kümenin α parametresine karşı çizimiyle de dallanma diyagramına ulaşılır.



Şekil 2.17: $(\varepsilon, \beta, \varphi)=(0.05, 0.44, 0.14)$ değerleri için elde edilen, n parametresinin $[0.9, 1.1]$ aralığındaki dallanma diyagramı

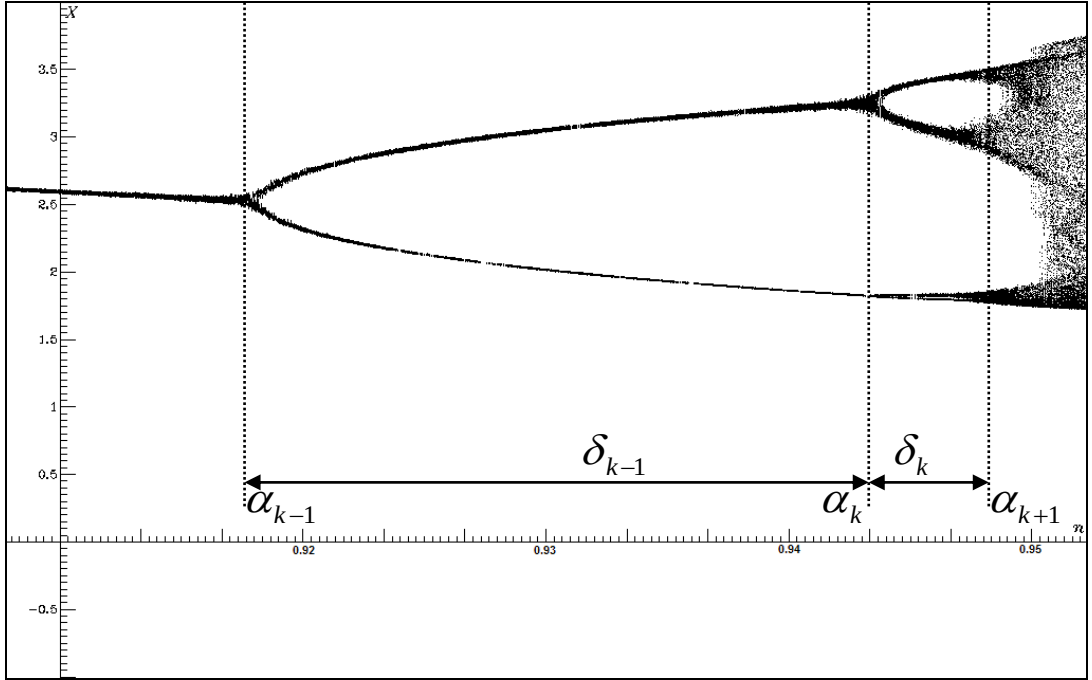
n parametresi için oluşturulan dallanma diyagramında (Şekil 2.17) kaosa ulaşma sürecindeki periyot katlanmaları göze çarpmaktadır. Osilatör ilk başta periyodik işaret üretirken, bir süre sonra periyot ikiye, daha sonra dörde çıkmaktadır. Bu şekilde periyot katlanmalarının sürmesiyle osilatör kaosa girer.

α_k sistemin 2^k periyodundan 2^{k+1} periyoduna geçtiği dallanma değeri olmak üzere:

$$\delta = \lim_{k \rightarrow \infty} \frac{\alpha_k - \alpha_{k-1}}{\alpha_{k+1} - \alpha_k} = \lim_{k \rightarrow \infty} \frac{\delta_{k-1}}{\delta_k} \quad (2.16)$$

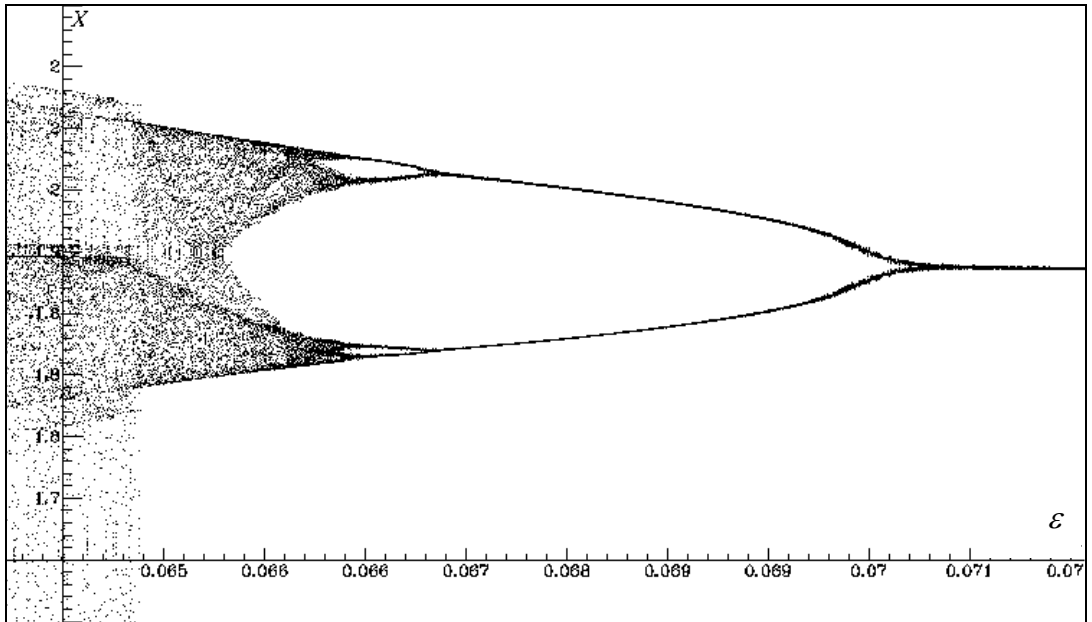
değerine Feigenbaum Sayısı adı verilir ve yaklaşık değeri 4.669'dur. Bu sabit, farklı kaotik sistemlerde görülmektedir ve bu yüzden evrensellik niteliği kazanmıştır [15].

Şekil 2.18’de görülen dallanma diyagramından hesaplanan δ değeri de 4.75 olup beklendiği gibi Feigenbaum Sayısına yakın çıkmıştır.



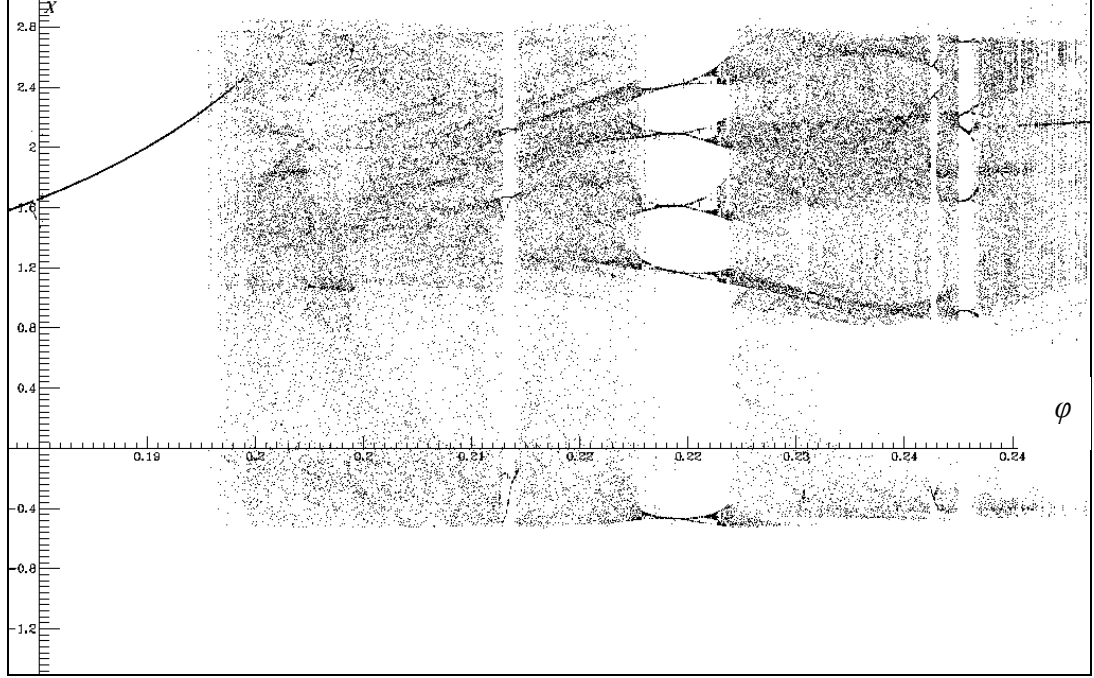
Şekil 2.18: $(\varepsilon, \beta, \varphi)=(0.05, 0.44, 0.14)$ değerleri için elde edilen, n parametresinin $[0.91, 0.95]$ aralığındaki dallanma diyagramı

Şekil 2.19’da ε parametresinin $[0.064, 0.071]$ aralığındaki dallanma diyagramı verilmiştir. Kaotik bölgenin içindeyken ε ’nin büyümesi ile periyot yarılanmaları (undoubling) meydana gelmiş ve sonunda periyodik işaret üretilmeye başlanmıştır.



Şekil 2.19: $(n, \beta, \varphi)=(1.4, 0.44, 0.14)$ değerleri için elde edilen, ε parametresinin $[0.064, 0.071]$ aralığındaki dallanma diyagramı

Kaotik osilatör φ parametresi için incelendiğinde Şekil 2.20'deki dallanma diyagramı elde edilir. Bu dallanma diyagramında da yine oldukça zengin bir dinamik yapı bulunmaktadır.

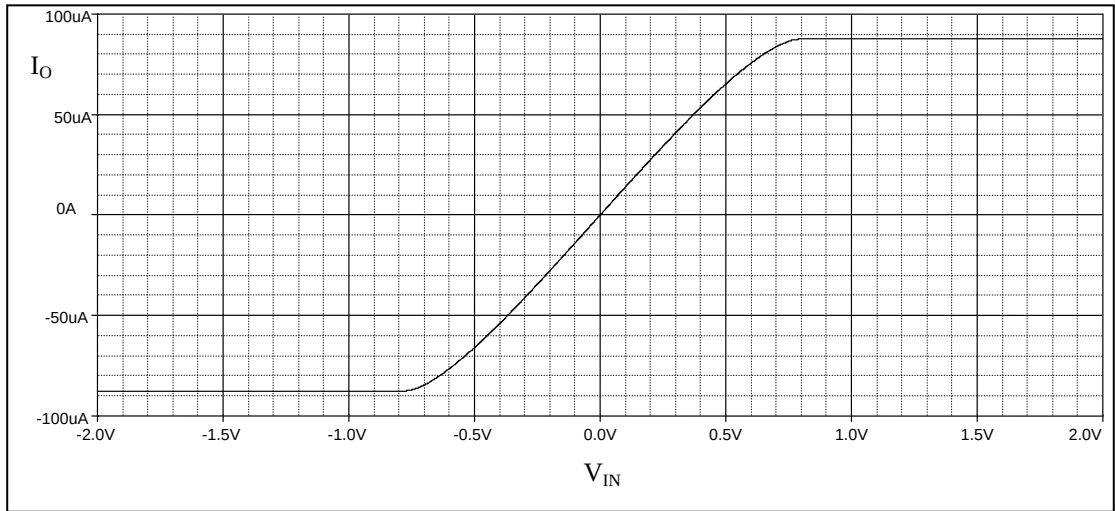


Şekil 2.20: $(\varepsilon, \beta, n)=(0.05, 0.44, 1.4)$ değerleri için elde edilen, φ parametresinin $[0.18, 0.25]$ aralığındaki dallanma diyagramı

Şekil 3.1’deki yapıda T1-T2 transistorları ile oluşturulmuş fark kuvvetlendiricisi, diyot bağlı T3-T4 ve T7-T8 transistorlarını sürmektedir. Bu transistorlarının akımları, transistor geometrileri ile belirlenen B çarpanı ile çarpılıp çıkışa aktarılmaktadır. OTA’nın geçiş iletkenliği şu şekilde hesaplanmaktadır:

$$g_m = B \sqrt{k'_n \cdot I_A \cdot \left(\frac{W}{L}\right)_1} \quad (3.1)$$

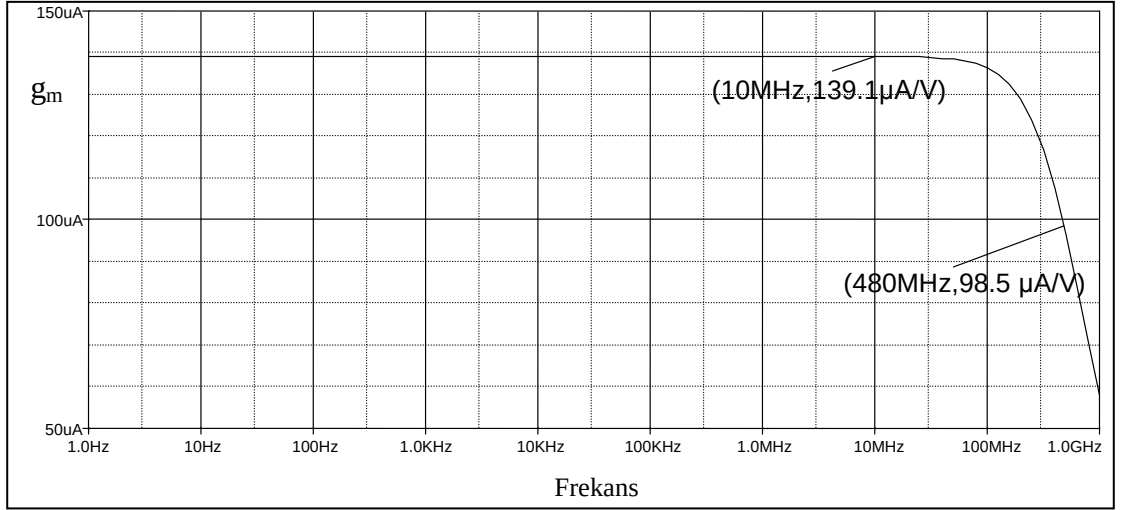
Şekil 3.2’de $I_A=48.5\mu A$ için OTA’nın çıkış akımının giriş gerilimi ile değişimi gösterilmiştir. OTA $\pm 800mV$ ’dan büyük giriş gerilimleri için doymaya girmektedir. OTA’nın maksimum çıkış akımı $\pm 87.8\mu A$ ’dır. OTA geçiş iletkenliğini gösteren eğim, $\pm 400mV$ ’dan sonra lineerliğini kaybetmektedir. Tasarlanan devrede gerilimler $250mV$ ’dan küçük tutulduğundan, OTA’nın lineer bölgede çalışması sağlanmıştır.



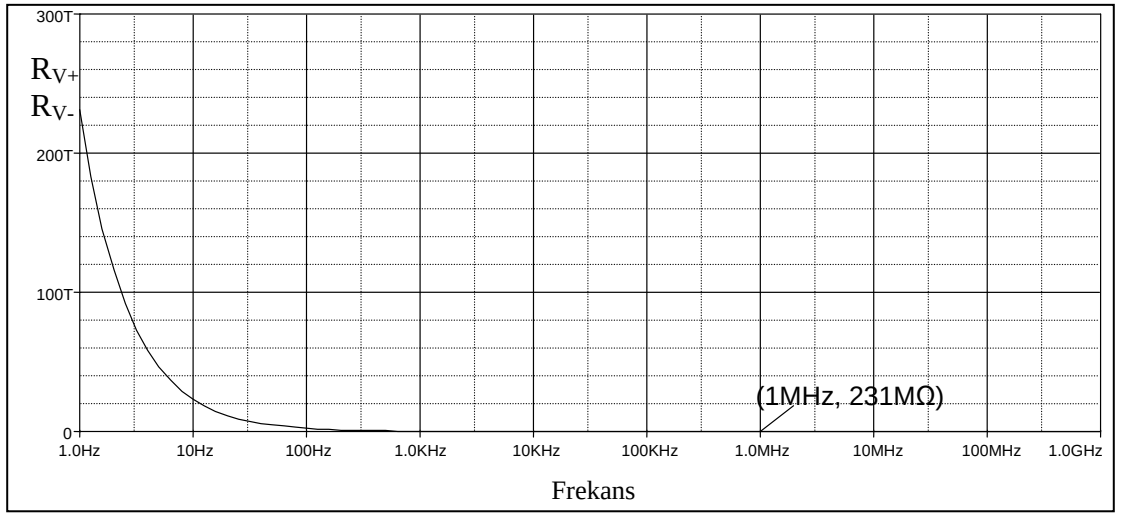
Şekil 3.2: OTA’nın çıkış akımının giriş gerilimi ile değişimi ($I_A=48.5\mu A$)

Tasarlanan devrede kullanılan kontrol akımlarından biri olan $I_A=48.5\mu A$ için OTA’nın geçiş iletkenliğinin frekans cevabı Şekil 3.3’de verilmiştir. Düşük frekanslarda OTA’nın geçiş iletkenliği $139.1 \mu A/V$ olup 3-dB bandgenişiği 480 MHz’dir.

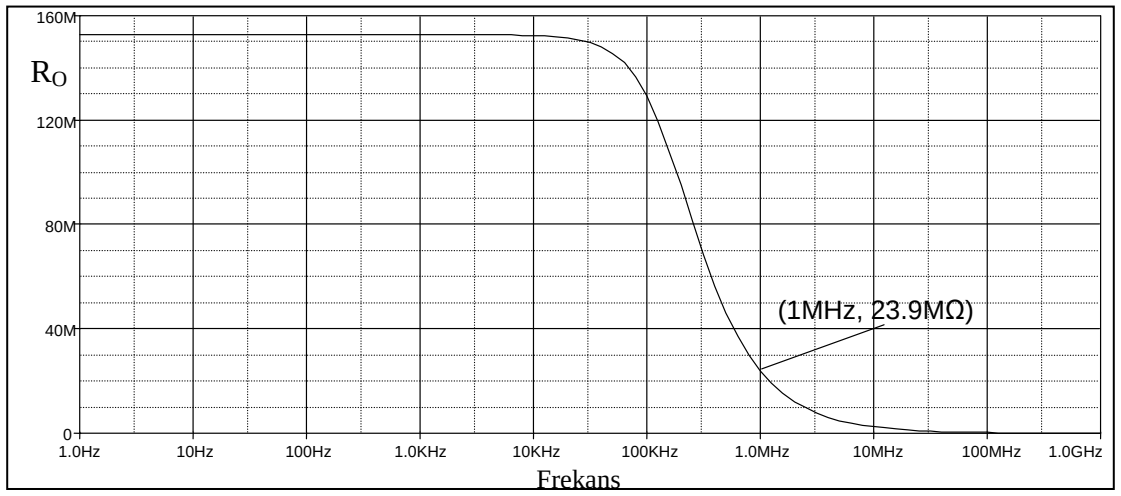
Devrenin yüksek frekansta çalışması amaçlandığından, OTA elemanının girişlerinin ve çıkışının bu frekanslarda göstereceği direnç büyük öneme sahiptir. Giriş kapılarının dirençlerinin frekans ile değişimi Şekil 3.4’de verilmiştir. Çıkış direnci için yapılan analiz ise Şekil 3.5’de gösterilmektedir. 10MHz altı frekanslar için terminal dirençleri $10M\Omega$ üstünde kalmaktadır, bu da tasarlanan yapı için yeterlidir.



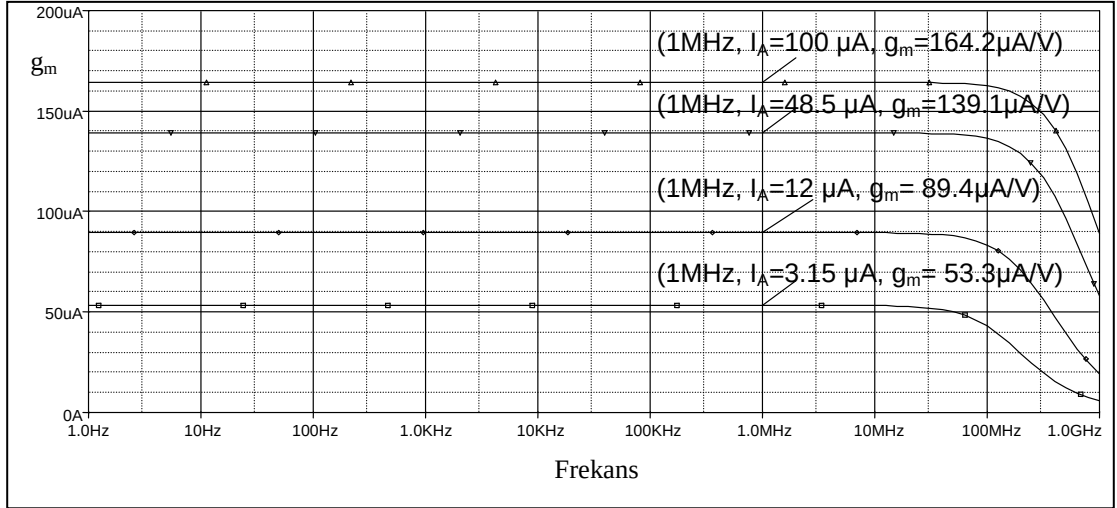
Şekil 3.3: OTA'nın geçiş iletkenliğinin frekans cevabı ($I_A=48.5\mu A$)



Şekil 3.4: OTA'nın giriş terminallerinin direncinin frekans ile değişimi ($I_A=48.5\mu A$)



Şekil 3.5: OTA'nın çıkış terminalinin direncinin frekans ile değişimi ($I_A=48.5\mu A$)



Şekil 3.6: OTA eğiminin farklı I_A kontrol akımı değerleri için frekans cevabı

OTA eğiminin I_A kontrol akımı ile değişimi Şekil 3.6’da gösterilmiştir. OTA yapısında besleme gerilimleri $V_{DD} = 2.5V$ ve $-V_{SS} = -2.5V$ ’dur. Kullanılan transistorların boyutları ise Tablo 3.1’de listelenmiştir.

Tablo 3.1: OTA yapısının transistor boyutları

Transistor	L [μm]	W [μm]	Transistor	L [μm]	W [μm]
T1	0.5	1	T8	0.5	5
T2	0.5	1	T9	0.5	10
T3	0.5	5	T10	0.5	10
T4	0.5	5	T11	0.5	10
T5	0.5	10	T12	0.5	10
T6	0.5	10	T13	0.5	10
T7	0.5	5	T14	0.5	10

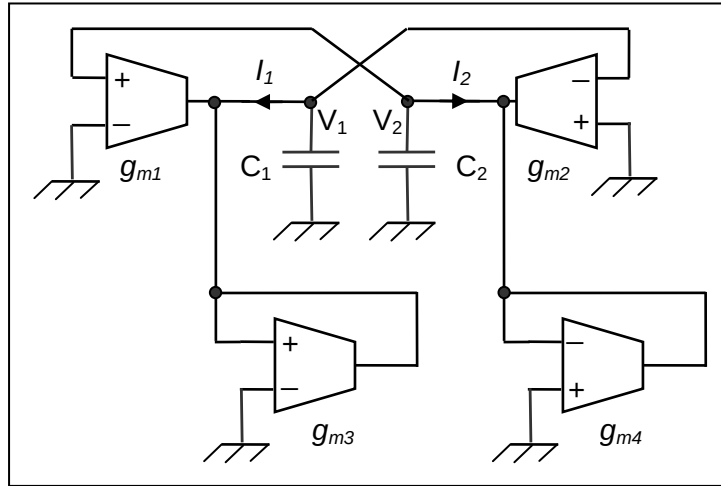
Simülasyonlarda kullanılan 0.5 μm CMOS proses parametreleri ise şu şekildedir:

```
.MODEL NT NMOS LEVEL=3
UO=460.5 TOX=1.0E-8 TPG=1 VTO=.62 JS=1.8E-6 XJ=.15E-6 RS=417
RSH=2.73 LD=0.04E-6 ETA=0 VMAX=130E3 NSUB=1.71E17 PB=.761
PHI=0.905 THETA=0.129 GAMMA=0.69 KAPPA=0.1 AF=1 WD=.11E-6
CJ=76.4E-5 MJ=0.357 CJSW=5.68E-10 MJSW=.302 CGSO=1.38E-10
CGDO=1.38E-10 CGBO=3.45E-10 KF=3.07E-28 DELTA=0.42 NFS=1.2E11

.MODEL PT PMOS LEVEL=3
UO=100 TOX=1E-8 TPG=1 VTO=-.58 JS=.38E-6 XJ=0.1E-6 RS=886
RSH=1.81LD=0.03E-6 ETA=0 VMAX=113E3 NSUB=2.08E17 PB=.911
PHI=0.905 THETA=0.120 GAMMA=0.76 KAPPA=2 AF=1 WD=.14E-6
CJ=85E-5 MJ=0.429 CJSW=4.67E-10 MJSW=.631 CGSO=1.38E-10
CGDO=1.38E-10 CGBO=3.45E-10 KF=1.08E-29 DELTA=0.81 NFS=0.52E11
```

3.2 Osilatör Yapısı

Kaotik osilatörün çekirdeğini oluşturacak osilatör için CMOS OTA-C yapısı seçilmiştir. Bu yapıda sadece geçiş iletkenliği kuvvetlendiricileri (OTA) ve kondansatörlerin mevcut olması önemli faydalar sağlamaktadır. Endüktans elemanı kullanımı gerektirmeyen bu osilatör topolojisinde OTA elemanlarının eğimleri tasarım parametresi olarak kullanılabilir. Osilatörün karakteristik denklemi $s^2 - bs + \Omega_0^2 = 0$ şeklindedir. b ve Ω_0 değerleri, OTA'ların geçiş iletkenlikleri ve kapasitelerin büyüklüğüne bağlıdır [16]. $b < 0$ olması durumunda osilatör sönümlü bir salınım gerçekleştirir. $b > 0$ için özdeğerler $j\omega$ ekseninin sağına geçmektedir. Aktif elemanların nonlineerlikleri nedeniyle çözümler limit çevrime oturur. Tasarlanan kaotik osilatörde kullanılan 4OTA2C osilatör yapısı Şekil 3.7'de gösterilmiştir.



Şekil 3.7: 4OTA2C osilatör topolojisi

Kapasite gerilimleri ve akımları Şekil 3.7'de gösterilen aktif devrenin kısa devre gösteriliminde yerine konulduğunda

$$\begin{bmatrix} -C_1 \dot{V}_{C1} \\ -C_2 \dot{V}_{C2} \end{bmatrix} = \begin{bmatrix} -g_{m3} & -g_{m1} \\ g_{m2} & g_{m4} \end{bmatrix} \begin{bmatrix} V_{C1} \\ V_{C2} \end{bmatrix} \quad (3.2)$$

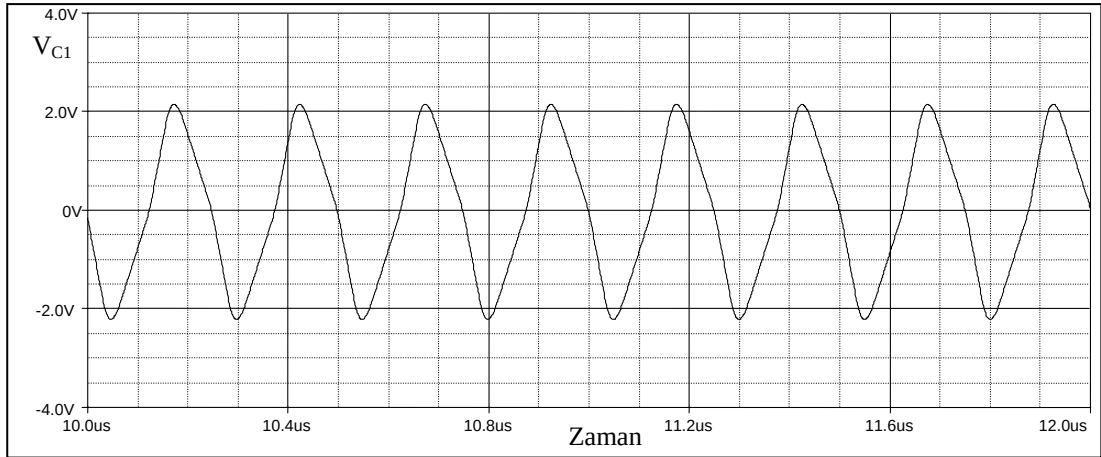
elde edilir. Buradan elde edilen karakteristik denklem şu şekildedir:

$$s^2 - \left(\frac{g_{m3}}{C_1} - \frac{g_{m4}}{C_2} \right) s + \frac{g_{m1}g_{m2} - g_{m3}g_{m4}}{C_1C_2} = 0 \quad (3.3)$$

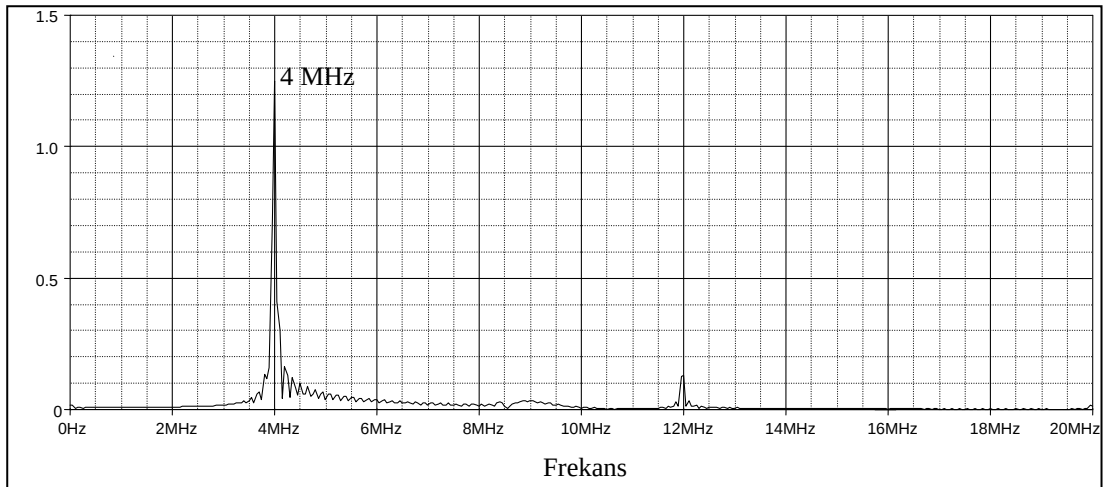
Osilasyon için $g_{m3}/C_1 = g_{m4}/C_2$ şartı sağlanmalıdır. Osilasyon frekansı ise

$$f_0 = \frac{1}{2\pi} \sqrt{\frac{g_{m1}g_{m2} - g_{m3}g_{m4}}{C_1C_2}} \quad (3.4)$$

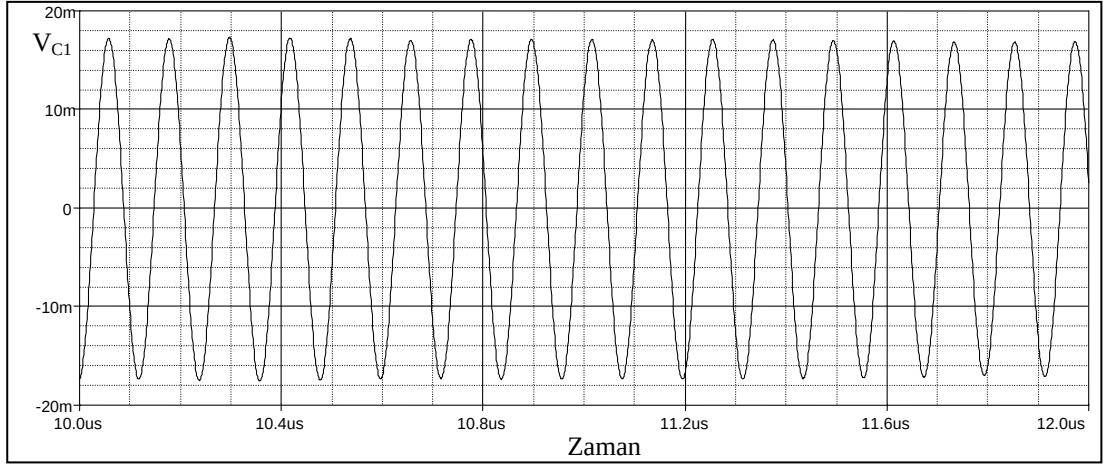
ifadesiyle bulunabilir. $g_{m1,m2}=139.1 \mu\text{A/V}$, $g_{m3,m4}=89.4 \mu\text{A/V}$ ve $C_{1,2}=2\text{pF}$ değerleri ile elde edilen 4 MHz frekanslı salınım Şekil 3.8’de, bu işaretin Fourier analizi ise Şekil 3.9’da verilmektedir. Verilen eleman değerleri için (3.4) denklemi ile salınım frekansı hesaplandığında $f_0=8.4 \text{ MHz}$ elde edilmektedir ancak yaklaşık $\pm 2\text{V}$ seviyelerinde salınım yapan devrenin frekansı, OTA elemanlarının yükselme eğimleri nedeniyle sınırlanmaktadır. g_{m4} değeri az miktar artırılarak salınımın besleme gerilimi seviyelerine yaklaşması önlendiğinde, (3.4) denklemiyle örtüşen sonuçlara ulaşılmaktadır. $g_{m4}=92 \mu\text{A/V}$ alınarak salınım genliğinin sınırlandırılmasıyla elde edilen 8.3 MHz frekanslı işaret Şekil 3.10’da verilmiştir.



Şekil 3.8: 4OTA2C osilatör topolojisiinde V_{C1} ’in zamanla değişimi



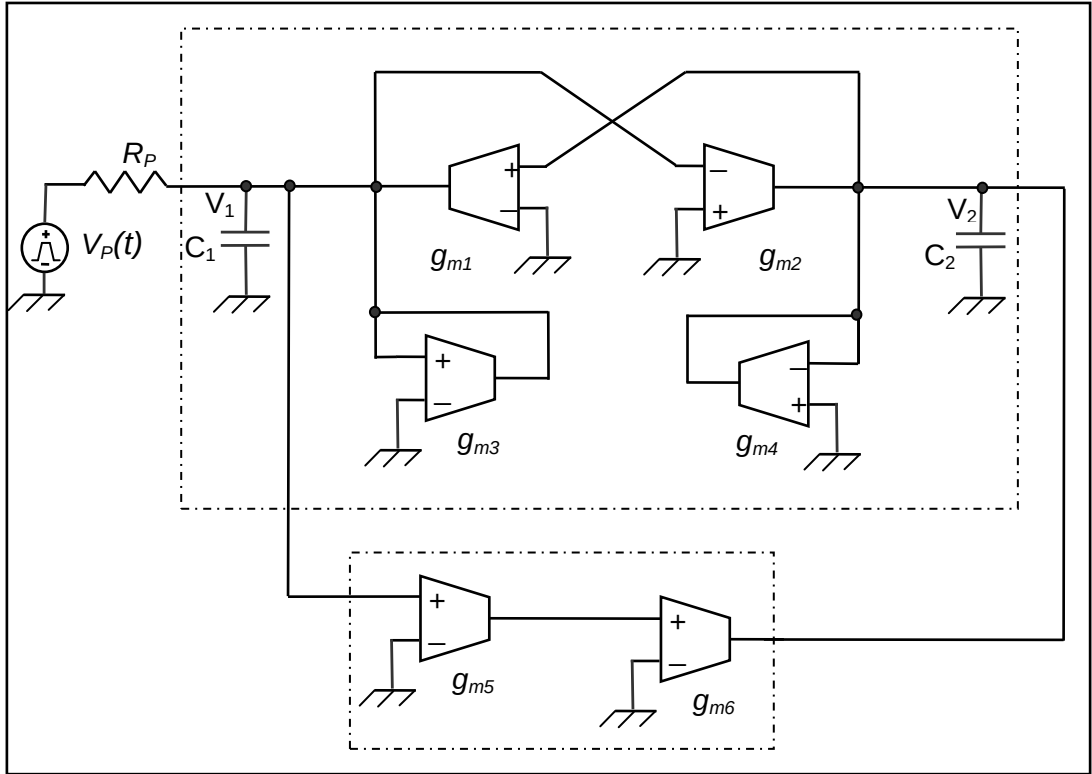
Şekil 3.9: 4OTA2C osilatör topolojisiinde V_{C1} ’in Fourier analizi



Şekil 3.10: 4OTA2C osilatör topolojisiinde sınırlandırılmış V_{C1} 'in zamanla değişimi

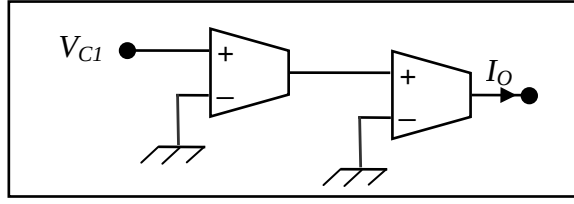
3.3 Kaotik Osilatör

Osilatöre yapılan eklemeler sonucu edilen kaotik osilatör Şekil 3.11'de verilmiştir. Kaotik osilatörün durum denklemleri ise (3.5) denkleminde verilmiştir.



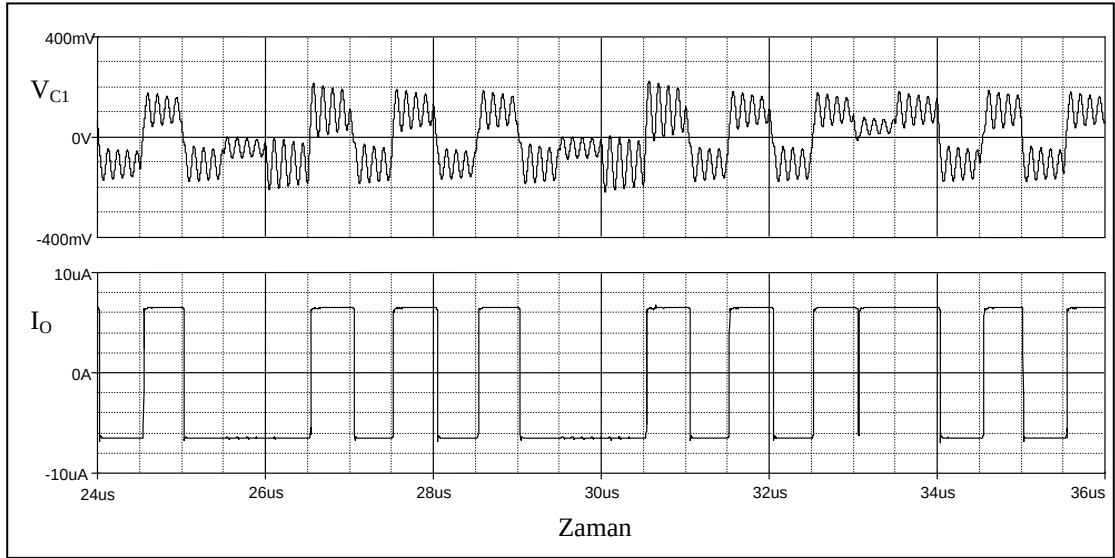
Şekil 3.11: Kaotik osilatör devresi

$$\begin{aligned} \dot{V}_{C1} &= \frac{1}{C} \left[\left(g_{m3} - \frac{1}{R_p} \right) V_{C1} + g_{m1} V_{C2} \right] + \frac{1}{R_p C} V_p(t) \\ \dot{V}_{C2} &= \frac{1}{C} [-g_{m2} V_{C1} - g_{m4} V_{C2}] + \frac{I_O}{C} \end{aligned} \quad (3.5)$$



Şekil 3.12: Geribesleme yapısı

Devrede (2.4) denklemiyle ifade edilen geribesleme işaretini üreten yapı Şekil 3.12’de gösterilmiştir. Bu yapı iki adet OTA’dan oluşmaktadır. OTA kullanılarak geribeslemenin yüksek hızlarda çalışabilmesi sağlanmıştır. Birinci OTA kazancı yükseltme amacı taşımaktadır. Doyma akımı devredeki I_{sat} akımına eşit olacak şekilde ayarlanan ikinci OTA ise, birinci kata uygulanan gerilimin pozitif ya da negatif olması durumuna göre I_{sat} ya da $-I_{sat}$ akımını akıtır. Şekil 3.13’de V_{C1} geriliminin zamanla değişimi ve bu gerilim değerlerine karşı düşen geribesleme akımı gösterilmiştir.



Şekil 3.13: V_{C1} geriliminin ve geribesleme akımının zamanla değişimi

Şekil 3.11’de gösterilen kaotik osilatörün eleman değerleri, osilatörün parametrelerini $(\epsilon, n, \beta, \varphi)=(0.05, 1.4, 0.44, 0.14)$ yapacak şekilde hesaplanmıştır. Hesaplanan bu değerler Tablo 3.2 de listelenmiştir.

Tablo 3.2: Kaotik osilatörde kullanılan eleman değerleri

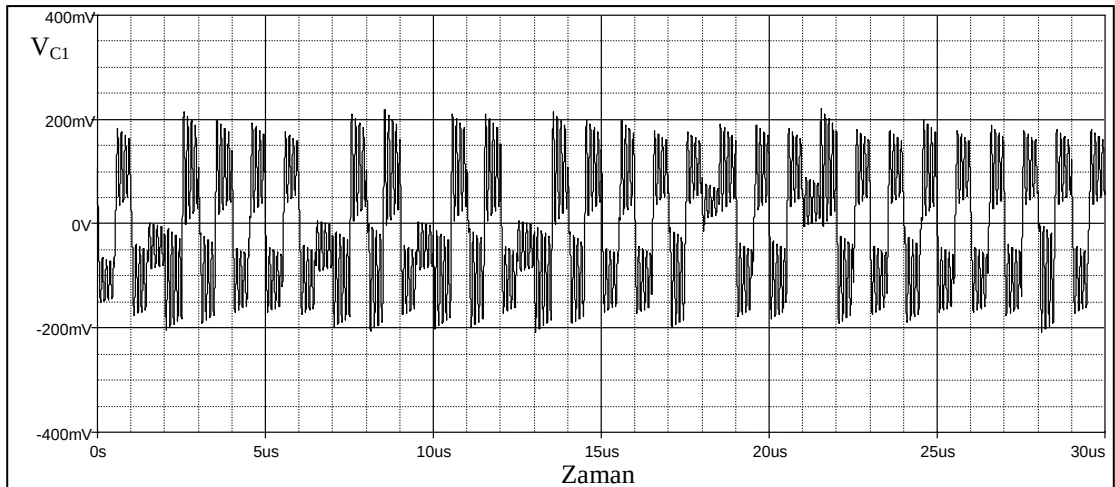
Eleman	V_P	f_P	C_1, C_2	R_P	g_{m1}, g_{m2}	g_{m3}, g_{m4}	I_{sat}
Değeri	1 V	1 MHz	2 pF	224 k Ω	139.1 μ A/V	89.4 μ A/V	6.5 μ A

OTA elemanların geiş iletkenliklerinin istenilen seviyeye getirilmesi için Tablo 3.3’de verilen kontrol akım deęerleri kullanılmıřtır.

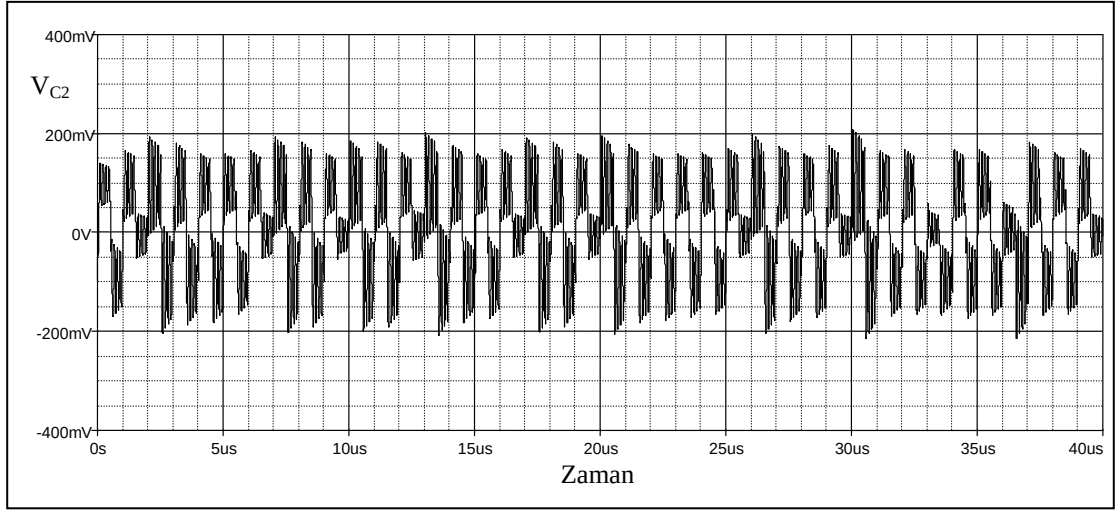
Tablo 3.3: Arzu edilen g_m deęerlerini ayarlamak için gerekli kontrol akımları

	$g_m (\mu A/V)$	$I_A (\mu A)$
OTA 1	139.1	48.5
OTA 2	139.1	48.5
OTA 3	89.4	12
OTA 4	89.4	12
OTA 5	164.2	100
OTA 6	53.3	3.15

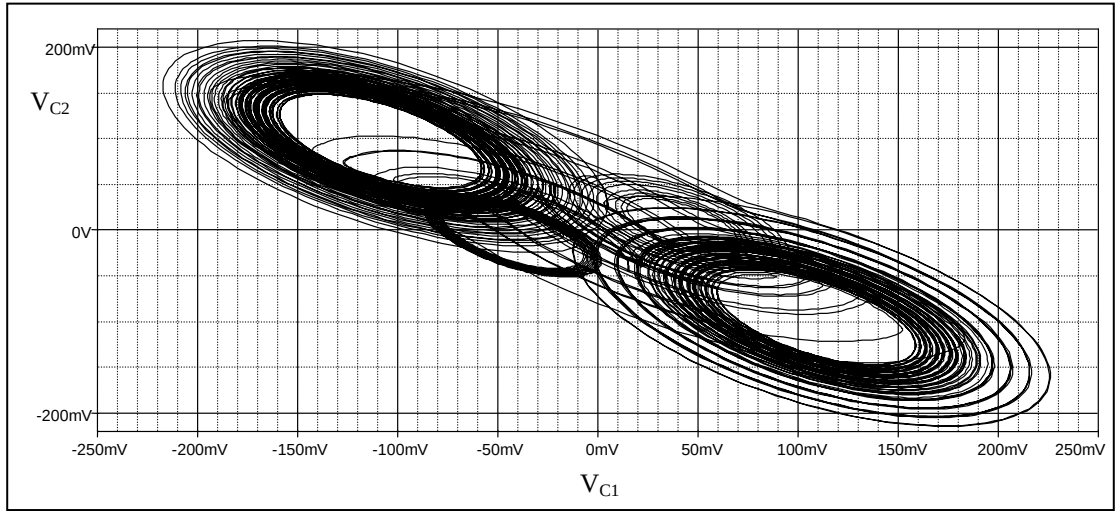
Tasarlanan kaotik osilatörün simülasyonu ile elde edilen, V_{C1} ve V_{C2} gerilimlerinin zamanla deęişim grafikleri sırasıyla Şekil 3.14 ve 3.15’de verilmektedir. Şekil 3.16’da ise bu iki gerilimin birbirine karşı çizdirilmesi ile oluşan kaotik çekici bulunmaktadır. V_{C1} geriliminin grafięi incelendięinde 4 farklı gerilim seviyesinde meydana gelen kümelenme dikkati çekmektedir. Bu kümelenmeler, kaotik çekicini sarmallarına karşı gelmektedir. Elde edilen kaotik çekici, nümerik analiz ile elde edilen yapıyla örtüşmektedir. Kaotik devrelerin bilgisayar simülasyonları yapılırken bilgisayarlarda deęerlerin sonlu hassasiyete sahip olduęu unutulmamalıdır. Devredeki gerilim ve akımların yuvarlanması ile oluşacak hata, bu noktadan sonraki yörüngenin ideal yörüngeden sapmasına yol açar. Bu durum, kaotik davranışın bilgisayar simülasyonunda tam olarak gözlemlenememesine neden olmaktadır.



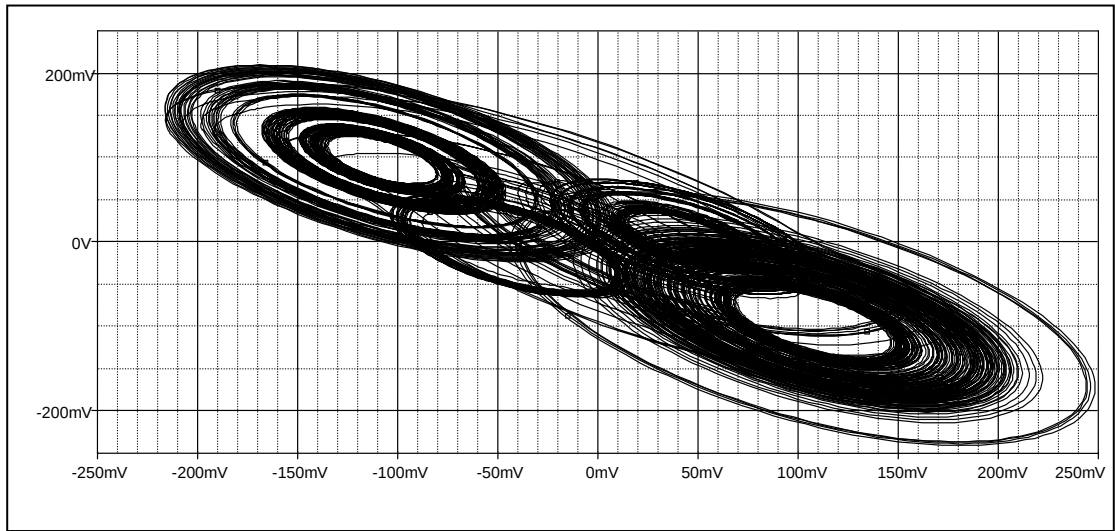
Şekil 3.14: V_{C1} geriliminin zamanla deęişimi



Şekil 3.15: V_{C2} geriliminin zamanla değişimi



Şekil 3.16: V_{C2} 'nin V_{C1} 'e karşı çizdirilmesiyle elde edilen kaotik çekici



Şekil 3.17: $f_p=2\text{MHz}$ ve $C=1\text{pF}$ için V_{C2} 'nin V_{C1} 'e karşı çizdirilmesiyle elde edilen kaotik çekici

Önerilen devrenin önemli bir özelliği de, devrede f_p ve C değerleri çarpımı sabit kaldığı sürece Şekil 3.16’da gösterilen kaotik çekici yapısının elde edilebilmesidir. Devreye uygulanan darbe kaynağının frekansının 2 katına çıkarılması ve kapasite değerlerinin yarıya düşürülmesi ile elde edilen kaotik çekici Şekil 3.17’de gösterilmiştir. Kapasitenin büyütülmesi ile aynı çekiciyi daha düşük frekanslı kaynakla gerçeklemek de mümkündür. Tasarımındaki bu esneklik sayesinde devre, sadece kapasite değerleri değiştirilerek, üzerinde kullanılacağı sistemde mevcut kaynak frekansına uyum sağlayacak hale getirilebilir. Böylelikle belirli bir frekansına bağımlılık ortadan kaldırılmış olur. Kullanılacak kaynak frekansının alt sınırını, tümdevre yapısında gerçekleştirilecek maksimum kapasite değeri belirler. Üst sınır ise devrede kullanılan OTA elemanlarının limitleri tarafından belirlenmektedir.

4. KAOTİK OSİLATÖR İLE RASGELE SAYI ÜRETİLMESİ

Kaos, Edward N. Lorenz tarafından “deterministik olan ancak deterministik gibi görünmeyen davranış” olarak tanımlanmıştır. Bir rasgele olaylar dizisinde, gerçekleşmesi mümkün bütün olaylar bir sonraki denemede gerçekleşebilir. Bu olayların gerçekleşme ihtimali de, bundan sonraki herhangi bir zamanda gerçekleşme ihtimalleri ile aynıdır. Bir deterministik dizide ise, bir sonraki olayın ne olacağı bellidir ve bu kesin kurallara bağlıdır [17].

Bir kaotik osilatörden rasgele sayı üretilmesi, osilatörden belli anlarda alınan örneklerin kuantalanması ve çıkışta bu bit dizisine belirli düzeltici algoritmaların uygulanması ile mümkün olmaktadır. Bu bölümde öncelikle dizilerin rasgeleliğini test etmek için kullanılan yöntemlerden bahsedilmiş, daha sonra otonom kaotik osilatör ile rasgele sayı üretiminin yöntemi anlatılmıştır. Otonom olmayan kaotik osilatör ile yüksek hızlı rasgele sayı üretimi için nasıl bir yöntem uygulandığı gösterilmiş ve bu yöntemle tasarlanan devre verilmiştir. Bu devre yapısında, Bölüm 3’de detayları verilen kaotik osilatör kullanılmıştır.

4.1 Rasgele Sayı Üreteçleri

Rasgele sayı üreteçleri üç temel gruba ayrılabilir. Bunlar gerçek rasgele sayı üreteçleri, sözde rasgele sayı üreteçleri ve hibrit rasgele sayı üreteçleridir. Bir gerçek rasgele sayı üretici, bir yarıiletkenin ısı gürültüsü yada bir osilatörün frekans kararsızlığı gibi öngörülemeyen doğal süreçleri ölçerek çalışır. Bu süreçler deterministik olmayan veya kaotik süreçlerdir. Bu tip üreteçler, yapılarından ötürü donanım-temelli üreteçler olarak da adlandırılmaktadır. Sözde rasgele sayı üreteçleri ise bir çekirdekten yola çıkarak çıkışlar üretirler. Çok hızlı çalışmalarına rağmen çıkışlarının entropisinin çekirdeğin entropisini geçmesi imkansızdır. Bu tip bir yapıda, kullanılan çekirdeğin bir gerçek rasgele sayı üretici tarafından üretilmesiyle hibrit rasgele sayı üretici ortaya çıkar. Çekirdeği üretiminde tuş basımı, bilgisayar faresi hareketi veya sabit diskin erişim süresi gibi kullanıcı etkileşimlerinden de yararlanılabilir [18,19].

4.2 Rasgelelik Testleri

Amerikan Ulusal Standart ve Teknoloji Enstitüsü (NIST) tarafından hazırlanan Federal Bilgi İşleme Standartları (FIPS) 140-1’de bir rasgele sayı üreticinin çıkışıını sınamak amacıyla uygulanması tavsiye edilen testler belirtilmiştir [20]. Üreteçten elde edilen 20 bin bitlik bir sayı dizisine uygulanması gereken testler ve bu testleri gerçekleyen MATLAB kodları aşağıda verilmiştir.

Tek-Bit Testi (Monobit Test): 20 bin bit içerisinde 1’lerin sayısı X olarak gösterilsin. $9654 < X < 10346$ sağlanıyorsa bit dizisi testi geçer.

```
function [sifir, bir, test_basarili]=tekbit(a)
sifir=0; bir=0; test_basarili=0;
for i=1:20000
    if a(i)==0 sifir=sifir+1;
    elseif a(i)==1 bir=bir+1; end end
if bir>9654 & bir<10346 test_basarili=1; end
```

Poker Testi: 20 bin bitlik dizi 5 bin adet 4 bitlik kümelere ayrılır. Her bir 4-bit değerin görülme sıklığı sayılır. $f(i)$, 4-bit değerin görülme sıklığını göstermek üzere

$$X = (16/5000) \times \left(\sum_{i=0}^{15} [f(i)]^2 \right) - 5000 \quad (4.1)$$

denkleminde $1.03 < X < 57.4$ sağlanıyorsa test başarılıdır.

```
function [x, test_basarili]=poker(a)
bulunma_sayisi(16)=0; kareler_toplami=0; test_basarili=0;
for i=1:4:20000
    hex=0;
    for j=0:3 hex=hex+a(i+j)*2^(3-j); end
    bulunma_sayisi(hex+1)=bulunma_sayisi(hex+1)+1; end
for i=1:16
    kareler_toplami=kareler_toplami+bulunma_sayisi(i)^2; end
x=16/5000*kareler_toplami-5000;
if x>1.03 & x<57.4 test_basarili=1; end
```

Koşu Testleri (Run Tests): Bir koşu, ardı ardına gelen sıfırların yada birlerin oluşturduğu dizidir. İncelenen dizide mevcut koşuların tümü sayılır ve kaydedilir. Eğer koşu sayılarının tümü tabloda verilen aralıklar içerisinde kalıyorsa test başarılıdır. Bu test için 6'dan uzun koşuların sayısı, 6'lık koşuların sayısına eklenir.

Tablo 4.1: Koşu sayılarının içinde bulunması gerekli aralıklar

Koşu Uzunluğu	İstenen Aralık
1	2267-2733
2	1079-1421
3	502-748
4	223-402
5	90-223
6+	90-223

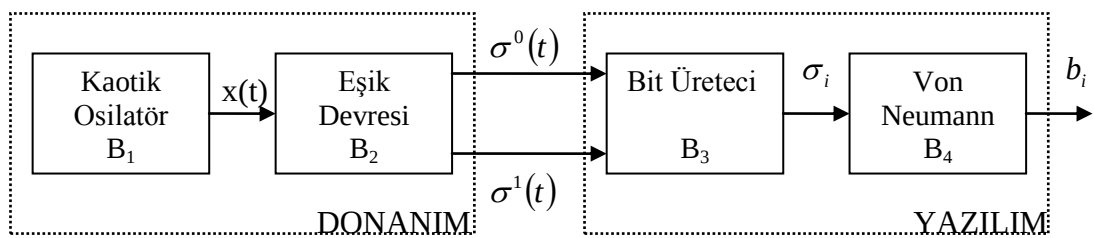
```
function [kosu0, kosu1, test_basarili]=kosu(a)
kosu0(6)=0; kosu1(6)=0; kosu=0; test_basarili=0;
for i=1:20000
    if a(i)==0 kosu=kosu+1;
    else if kosu>6 kosu0(6)=kosu0(6)+1;
        else if kosu~=0 kosu0(kosu)=kosu0(kosu)+1; end end
        kosu=0; end end
kosu=0;
for i=1:20000
    if a(i)==1 kosu=kosu+1;
    else if kosu>6 kosu1(6)=kosu1(6)+1;
        else if kosu~=0 kosu1(kosu)=kosu1(kosu)+1; end end
        kosu=0; end end
if kosu0(1)<2733 & kosu0(1)>2267 & kosu0(2)<1421 & kosu0(2)>1079 &
kosu0(3)<748 & kosu0(3)>502 & kosu0(4)<402 & kosu0(4)>223 &
kosu0(5)<223 & kosu0(5)>90 & kosu0(6)<223 & kosu0(6)>90 & kosu1(1)<2733
& kosu1(1)>2267 & kosu1(2)<1421 & kosu1(2)>1079 & kosu1(3)<748 &
kosu1(3)>502 & kosu1(4)<402 & kosu1(4)>223 & kosu1(5)<223 & kosu1(5)>90
& kosu1(6)<223 & kosu1(6)>90 test_basarili=1; end
```


Uzun Koşu Testi (Long Run Test): Bir uzun koşu, 34 veya daha uzun bir koşu olarak tanımlanır. 20,000 bitlik bir dizide eğer hiç uzun koşu yoksa test başarılıdır.

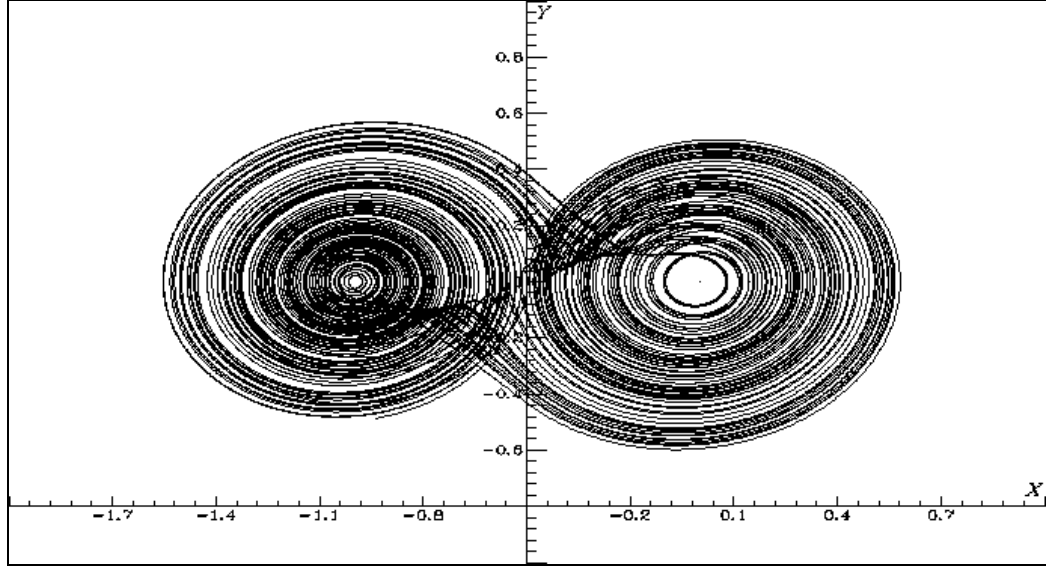
```
function [en_uzun, test_basarili]=uzunkosu(a)
suanki_bit=a(1); kosu=0; en_uzun=1; test_basarili=0;
for i=2:20000
    sonraki_bit=a(i);
    if suanki_bit==sonraki_bit kosu=kosu+1;
        if kosu>en_uzun en_uzun=kosu; end
    else kosu=1; end
    suanki_bit=sonraki_bit; end
if en_uzun<34 test_basarili=1; end
```

4.3 Otonom Kaotik Osilatör ile Rasgele Sayı Üretimi

Otonom bir sürekli-zaman kaotik sistemin gerçek rasgele sayı üreticisi olarak kullanılabildiği literatürde gösterilmiştir [18]. Önerilen üreteç donanım ve yazılım olmak üzere iki kısımdan oluşmaktadır (Şekil 4.1). Donanım kısmında kaotik osilatör ve eşik devresi bulunmaktadır. Kullanılan kaotik osilatör üçüncü dereceden otonom bir sistem olup çift sarmallıdır (Şekil 4.2). Osilatör bloğu B_1 olarak adlandırılmıştır. Osilatör tarafından üretilen çekicinin durum uzayı, B_2 bloğunda 3 alt uzaya ayrılır. Bu alt uzaylardan iki tanesi çekicinin sarmalları ile belirlenmiştir. Üçüncü alt uzay ise diğer ikisi arasındaki bölgeyi kapsamaktadır. B_3 bloğuyla gösterilen bit üretici, çekicinin alt uzaylar arasındaki geçişlerini kodlar. B_2 ve B_3 blokları birlikte tersi alınamaz (non-invertible) bir fonksiyon gerçeklemektedirler. Kaotik işaretin durum uzayında örneklenmesi, zaman açısından düzensiz örneklemeler anlamına gelmektedir. Bu da istenmeyen kişilerin osilatörün sonraki davranışların kestirmesini güçleştirmektedir. Son blok olan B_4 ise, bit üreticinin çıkışında bitler arasında oluşan dengesizliği ortadan kaldırır. Bunu Von Neumann Metodu ile gerçeklemektedir.



Şekil 4.1: Otonom kaotik osilatör ile rasgele sayı üretimi

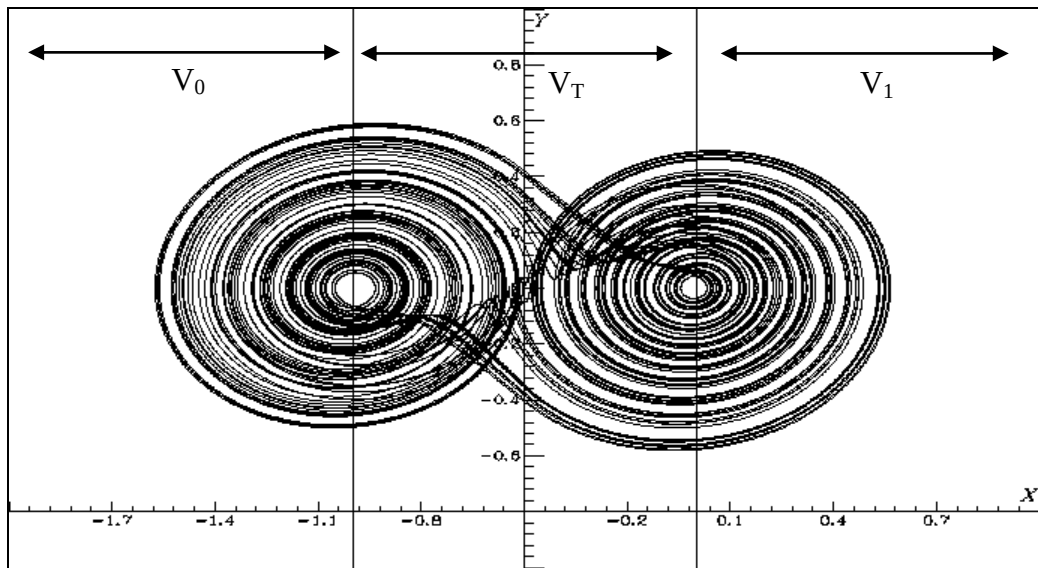


Şekil 4.2: Otonom kaotik osilatörün ürettiği çift sarmallı çekici

Kaotik bir işaretten bit dizisi üretmenin en kolay yolu bir eşik değerine göre kodlama yapmaktır. Eşikten küçük değerler 0, büyük değerler ise 1 olarak kodlanır.

$$\sigma_c(x) = \begin{cases} 0, & x < c \\ 1, & x \geq c \end{cases} \quad (4.2)$$

Bu şekilde elde edilen diziye kaotik bit dizisi adı verilir. Bu rasgele sayı üretici yapısında ise çekicinin 3 alt uzaya ayrılması amacıyla 2 eşik değeri kullanılmıştır. Bu alt uzaylar $V_0 = \{(x, y, z) | x \leq c_2\}$, $V_T = \{(x, y, z) | c_2 < x < c_1\}$, $V_1 = \{(x, y, z) | x \geq c_1\}$ ile tanımlıdır. $c_1=0$ ve $c_2=-1$ için bu alt uzaylar Şekil 4.3’de gösterilmiştir.



Şekil 4.3: Otonom kaotik osilatörün ürettiği çekicinin alt uzayları

B₂ bloğunun çıkış fonksiyonları olan $\sigma^0(t)$ ve $\sigma^1(t)$ şu şekilde tanımlanmıştır:

$$\begin{aligned}\sigma^1(x(t)) &= \begin{cases} 0, & x(t) < c_1 \\ 1, & x(t) \geq c_1 \end{cases} \\ \sigma^0(x(t)) &= \begin{cases} 0, & x(t) > c_2 \\ 1, & x(t) \leq c_2 \end{cases}\end{aligned}\quad (4.3)$$

2. bloğun çıkışından alınan işaretler bilgisayara aktarılmış ve bundan sonraki işlemler yazılım ile gerçekleştirilmiştir. Yazılım kısmının ilk parçası olan B₃ bloğu

$$\sigma_i(\sigma_0, \sigma_1) = \begin{cases} 0, & \sigma^0 = 0, \sigma^1 :_0 \uparrow^1 \\ 1, & \sigma^1 = 0, \sigma^0 :_0 \uparrow^1 \end{cases}\quad (4.4)$$

fonksiyonunu gerçeklemektedir. Burada $_0 \uparrow^1$ yükselen kenarı ifade eder. Diğer bir anlatımla, V_T alt uzayından V_1 alt uzayına sıçrayışta 1, V_T alt uzayından V_0 alt uzayına geçiş sırasında ise 0 biti üretilmektedir. Yapının son bloğu olan B₄'de ise çarpıklık düzeltme adı verilen teknik gerçekleştirilmektedir. Bu teknik ile bit üreticinin çıkışında oluşturulacak bitlerin dağılımları düzenlenir. Von Neumann Metodu ile 01 bit çifti 0'a, 10 bit çifti ise 1'e dönüştürülür. Geri kalan bit çiftleri ise kullanılmaz. Bu metod, kolay gerçekleştirilmesi ve bit sayısını çok fazla azaltmaması nedeniyle tercih edilir. Metodun matematiksel ifadesi (4.5) denklemi ile verilmiştir.

$$b_i(\sigma_i, \sigma_{i-1}) = \begin{cases} 0, & \sigma_i = 0 \wedge \sigma_{i-1} = 1 \\ 1, & \sigma_i = 1 \wedge \sigma_{i-1} = 0 \end{cases}\quad (4.5)$$

Von Neumann Metodu ile örnek bir diziden üretilen çıkış, Şekil 4.4'de gösterilmiştir.

Giriş:	0	1	0	0	1	1	0	1	1	0	1	0
Çıkış:		1	Çıkış Yok	Çıkış Yok			1		0		0	

Şekil 4.4: Bir örnek dizi için Von Neumann Metodu ile elde edilen çıkış dizisi

Önerilen otonom kaotik osilatör tabanlı gerçek rasgele sayı üreticinin, tüm denemelerde Bölüm 4.2'de anlatılan FIPS-140-1 testlerini geçtiği belirtilmiştir [18].

4.4 Otonom Olmayan Kaotik Osilatör ile Rasgele Sayı Üretimi

Otonom bir kaotik sistem ile rasgele sayı üretimi Bölüm 4.3’de anlatılmıştır. Benzer şekilde otonom olmayan kaotik sistem ile rasgele sayı üretilebilmesi amacıyla zamanın belli anlarında örnekleme yapılması düşünülmüştür. Bu amaçla Poincaré dönüşümünden yararlanılmıştır. n . dereceden T periyotlu bir sürekli zaman sistemi, Poincaré dönüşümü (Poincaré map) adı verilen $(n-1)$. derece ayırık-zaman sistemine dönüştürülebilir. Poincaré dönüşümü bu şekilde sürekli zaman ve ayırık zaman sistemlerini birbirine bağlamaktadır. Poincaré kesiti şu şekildedir:

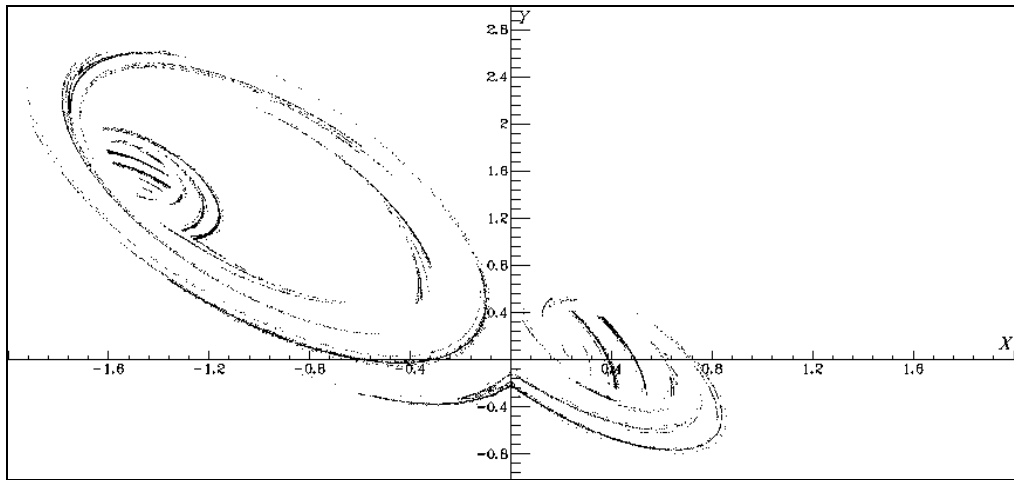
$$\Sigma := \{(x, \theta) \in R^n \times S^1 : \theta = \theta_0\} \quad (4.6)$$

Burada $S^1 := [0, 2\pi)$ ve $\theta(t) = \left(\frac{2\pi}{T}t\right) \bmod 2\pi$ ’dir. Her T saniyede yörünge Σ ile kesişir. Ortaya çıkan dönüşüm şu şekilde tanımlıdır:

$$P_N(x) := \phi_{t_0} + T(x, t_0) \quad (4.7)$$

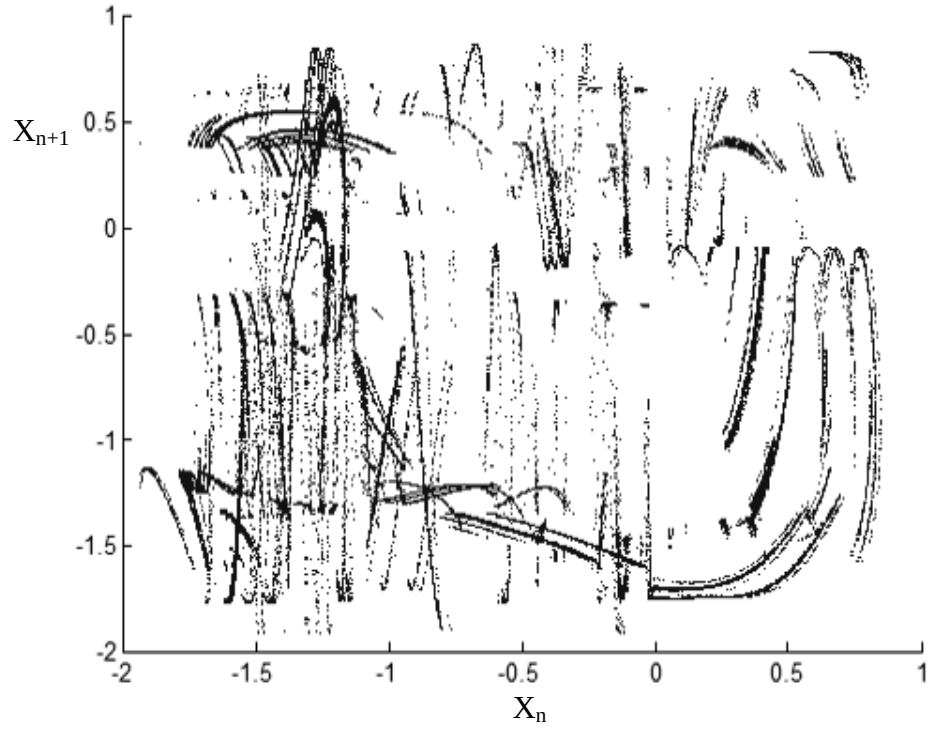
P_N tersi alınabilir (invertible) bir dönüşüm olup, bir yörünge her T saniyede örneklenmesi olarak düşünülebilir [4].

Şekil 4.5’de otonom olmayan kaotik osilatörü $\sin \phi\tau = 0$ anlarında örnekleyerek elde edilen Poincaré dönüşümü verilmiştir. Örnekleme yapıldığı $\sin \phi\tau = 0$ anları, sisteme uygulanan saat işaretinin yükselen kenarlarına karşı gelir. Bu durumun devredeki karşılığı, osilatörün çıkışının, saat işaretinin yükselen kenarı ile tetiklenen bir örnekleyici kullanılarak örneklenmesidir.



Şekil 4.5: Otonom olmayan kaotik osilatörün $\sin \phi\tau = 0$ için Poincaré kesiti

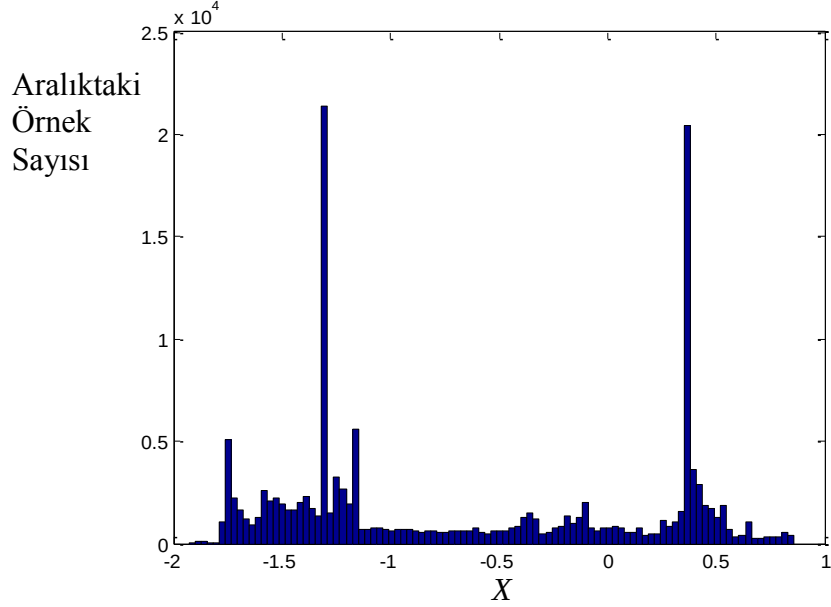
Elde edilen bu dönüşüm tersi alınabilir. Bu dönüşümdeki noktalardan sadece X değerlerinin dikkate alınıp Y değerlerinin kullanılmaması suretiyle dönüşümün tersi alınamaz hale gelmesi sağlanır. Bu şekilde üretilen bitlere bakılarak sistemin daha önceki durumlarına ulaşılması olasılığı engellenmiş olur. X durum değişkeninin bir sonraki durumunun şimdiki durumuna karşı çizdirilmesiyle elde edilen dönüşüm Şekil 4.6’da verilmiştir. Buradan da görüldüğü gibi, bir X durumuna birçok farklı noktadan ulaşılabilir. Bu nedenle de sistemin geriye doğru incelenmesi imkansız olacaktır.



Şekil 4.6: X durum değişkeninin bir sonraki periyottaki değerinin şimdiki değerine karşı çizdirilmesiyle elde edilen dönüşüm

(2.6) denklemi ile verilen kaotik sistemden rasgele sayı üretmek için denklemin nümerik analizi yapılmış ve saat işaretinin yükselen kenarlarında ($\sin \phi \tau = 0$ olduğu noktalarda) X durum değişkeninin 150 bin değeri kaydedilmiştir. Bu değerlerin dağılımı Şekil 4.7’deki histogramda gösterilmektedir. Eşik değerini belirlerken bu histogramdan yararlanılmıştır. Histogramın ağırlık merkezi $X = -0.06$ değeridir. Üreteç yapısında tek bir eşik değeri kullanılması amaçlanmıştır. Bu sayede karşılaştırmayı gerçekleştirmek basitleşecektir. (4.2) denklemiyle gösterilen şekilde kaotik çekici iki alt uzaya ayrılmış ve kodlama bu şekilde gerçekleştirilmiştir.

Ağırlık merkezine yakın bir eşik değeri seçilmesi ile üretilen bitlerin dağılımının düzgün olması sağlanır. Bu şekilde, Von Neumann Metodu uygulanması sonucunda kullanılmayacak bit sayısı düşer ve rasgele sayı üreticinin çıkış hızı yükselir.



Şekil 4.7: X durum değişkeninin, saat işaretinin yükselen kenarında örneklenen 150 bin değerinin dağılımını gösteren histogram

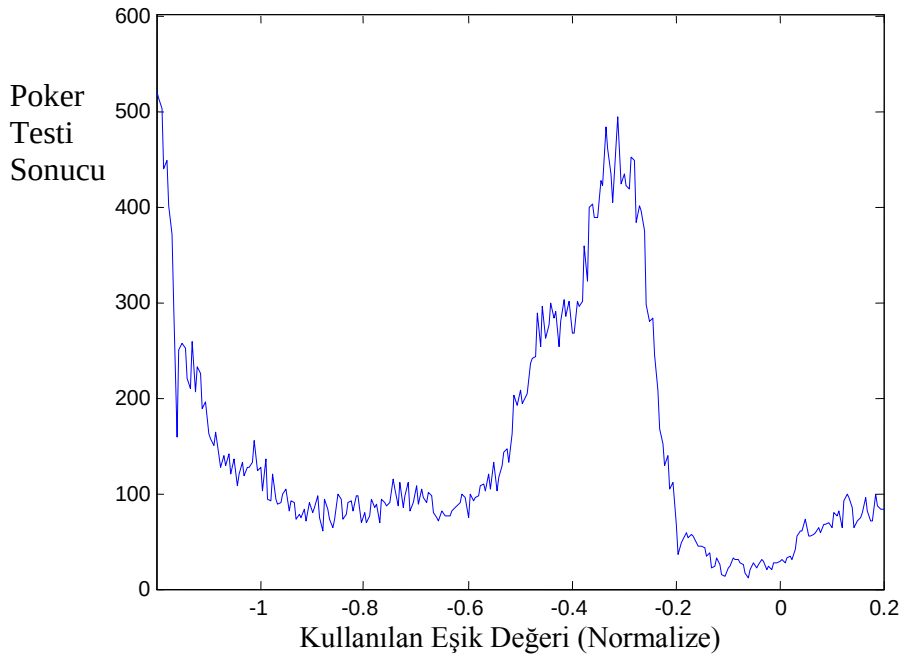
Elde edilen değerler bir eşik değeriyle karşılaştırılarak bit dizisi oluşturulmuş ve bu bit dizisine Von Neumann Metodu uygulanarak bitlerin dağılımındaki düzensizlik ortadan kaldırılmıştır. Von Neumann Metodu için yazılan MATLAB kodu aşağıda verilmiştir:

```
function [b, kullanılan_bit_sayisi]=neumann(a)
b(20000)=0; j=1;
for i=1:2:length(a)
    bit1=a(i);
    bit2=a(i+1);
    if bit1+bit2==1
        b(j)=bit1;
        j=j+1; end
    if j==20000 break end end
if j<20000 error('hata: giris yeterince uzun degil') end
kullanilan_bit_sayisi=i;
```

Üretilen bit dizilerinin rasgeleliği değişik eşik gerilimleri için test edilmiştir. Rasgelelik için Bölüm 4.2’de bahsedilen Poker Testinin sonuçları öncelikle dikkate alınmıştır. Bu incelemede kullanılan MATLAB kodu aşağıda verilmiştir:

```
fid=fopen('degerler.txt');
[a,count]=fscanf(fid, '%f',150000);
hist(a,100);
i=1;
for k=-1.2:0.005:0.2
    bit_dizisi=esik(a,k);
    rasgele_dizi=neumann(bit_dizisi);
    x(i)=poker(rasgele_dizi);
    i=i+1; end
plot([-1.2:0.005:0.2],x)
```

Poker Testinin sonucunun kullanılan eşik değeri ile değişimi Şekil 4.8’de verilmiştir. Elde edilen minimum değer 11.13 olup, bu değer için kullanılan eşik -0.06’dır. Bu değer için karşılaştırmacıda kullanılması gereken gerilim 6.79mV’dur. Bunun yerine eşik değeri olarak $X=0$ kullanılması durumunda, poker testi 28.2 değerini vermektedir. Bu değer arzu edilen sınırlar içerisinde ve devrede gerçekleşme açısından kolaylık sağlamaktadır.

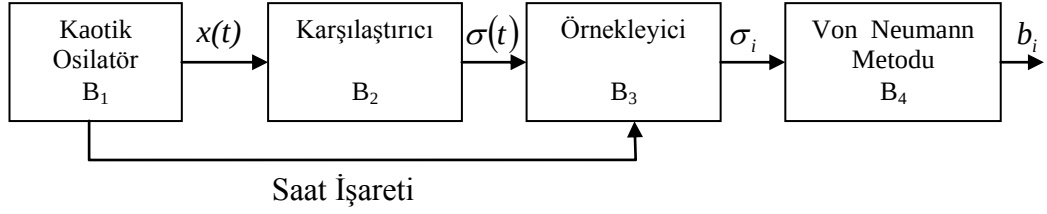


Şekil 4.8: Poker Testinin sonucunun kullanılan normalize eşik değeri ile değişimi

Rasgele sayı üreticinin nümerik analizi ile oluşturulan 20,000 bitlik dizilere rasgelelik testleri uygulanmış ve elde edilen sonuçlar Tablo 4.2’de verilmiştir. Farklı başlangıç koşullarından başlatılan simülasyonlar sonucunda üretilen diziler bütün denemelerde rasgelelik testlerini başarıyla geçmiştir.

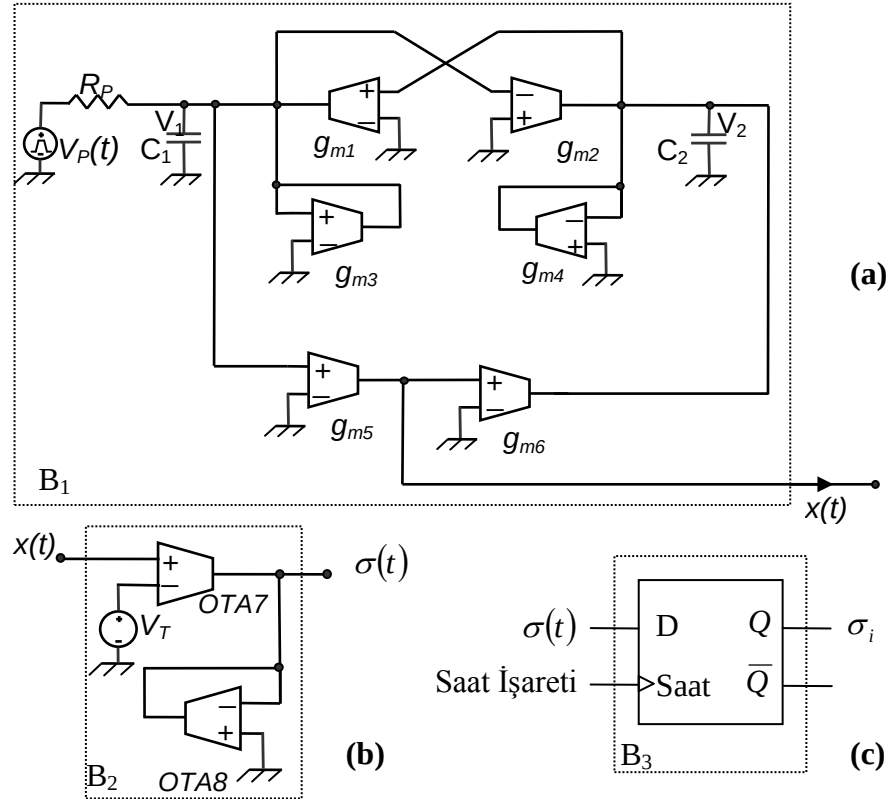
Tablo 4.2: Nümerik analizler ile elde edilen dizilerin rasgelelik test sonuçları

Dizi Numarası	Tek-Bit Testi	Poker Testi	Koşu Testleri		Uzun Koşu Testi
1	9989	28.19	S ₁ =2416 S ₂ =1156 S ₃ =697 S ₄ =355 S ₅ =124 S ₆₊ =166	B ₁ =2388 B ₂ =1194 B ₃ =713 B ₄ =320 B ₅ =139 B ₆₊ =160	Uzun Koşu Yok
Test Geçildi?	Evet	Evet	Evet		Evet
2	1007	11.47	S ₁ =2427 S ₂ =1205 S ₃ =660 S ₄ =323 S ₅ =172 S ₆₊ =144	B ₁ =2447 B ₂ =1166 B ₃ =712 B ₄ =300 B ₅ =143 B ₆₊ =163	Uzun Koşu Yok
Test Geçildi?	Evet	Evet	Evet		Evet
3	10031	18.82	S ₁ =2429 S ₂ =1148 S ₃ =729 S ₄ =282 S ₅ =154 S ₆₊ =166	B ₁ =2349 B ₂ =1263 B ₃ =667 B ₄ =325 B ₅ =147 B ₆₊ =158	Uzun Koşu Yok
Test Geçildi?	Evet	Evet	Evet		Evet
4	9810	32.97	S ₁ =2412 S ₂ =1197 S ₃ =699 S ₄ =324 S ₅ =163 S ₆₊ =169	B ₁ =2553 B ₂ =1115 B ₃ =691 B ₄ =330 B ₅ =136 B ₆₊ =140	Uzun Koşu Yok
Test Geçildi?	Evet	Evet	Evet		Evet
5	9963	30.31	S ₁ =2434 S ₂ =1208 S ₃ =663 S ₄ =328 S ₅ =141 S ₆₊ =166	B ₁ =2464 B ₂ =1164 B ₃ =711 B ₄ =294 B ₅ =144 B ₆₊ =164	Uzun Koşu Yok
Test Geçildi?	Evet	Evet	Evet		Evet



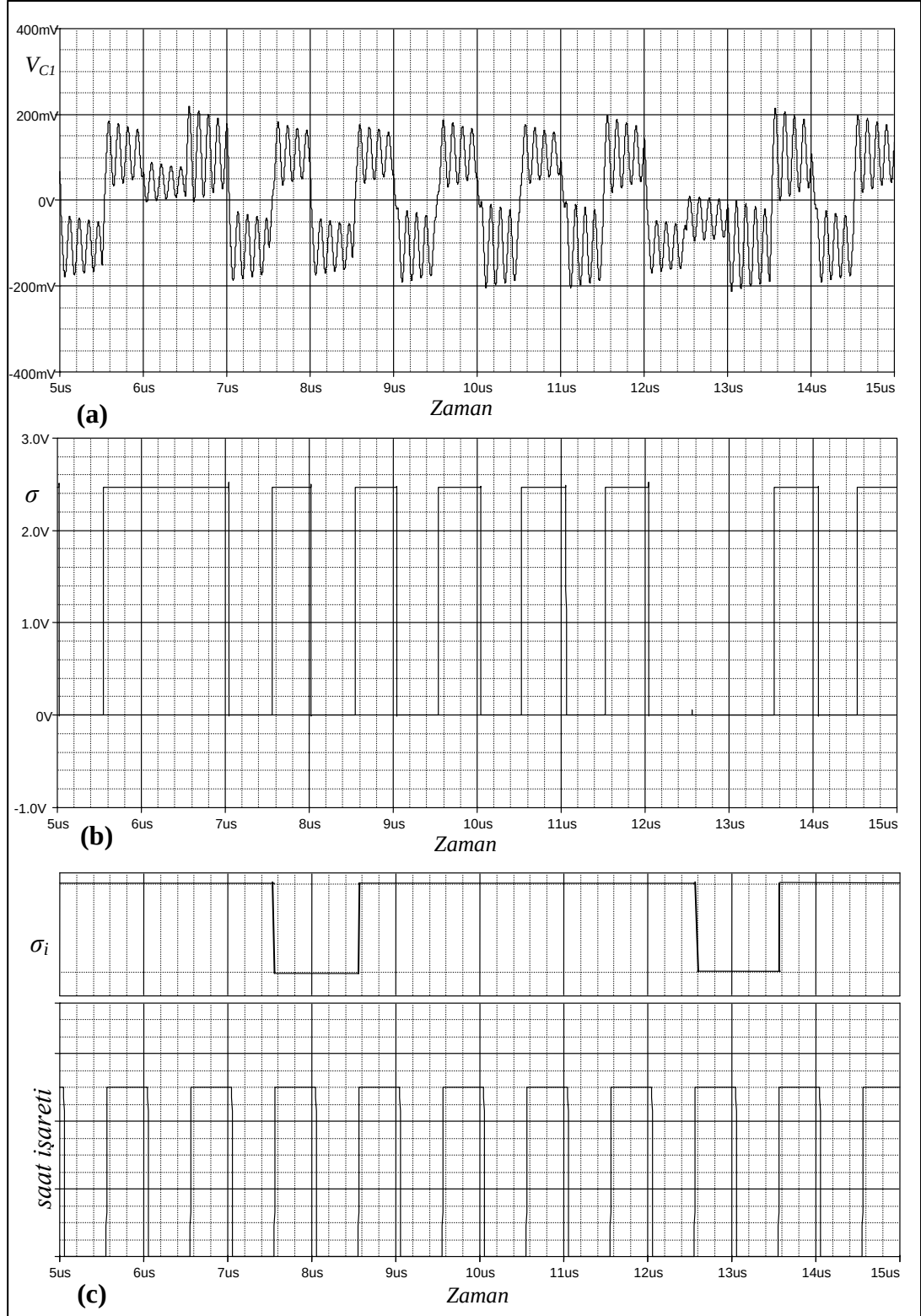
Şekil 4.9: Otonom olmayan kaotik osilatör ile rasgele sayı üretimi

4 temel bloktan meydana gelen rasgele sayı üretici yapısı Şekil 4.9’da gösterilmiştir. Bu blokların devre yapıları ise Şekil 4.10’da verilmektedir. İlk blok olan B_1 , Bölüm 3’de incelenen, (2.6) denklemiyle tanımlanmış kaotik osilatör ile gerçekleştirilmektedir. Osilatörün V_{C1} gerilimi, B_2 bloğunda V_T gerilimi ile karşılaştırılır. Bu blok iki OTA elemanı ile gerçekleştirilmiştir. OTA 7 ile belirtilen OTA yapısında besleme gerilimi olarak $V_{DD}=2.5V$ ve $V_{SS}=0V$ kullanılmıştır. Bu sayede $\sigma(t)$ işareti, doğrudan bir lojik kapıya bağlanmaya uygun şekilde üretilmektedir. B_3 bloğu basit bir D tipi flip-flop elemanından oluşmaktadır. $\sigma(t)$ işareti saat işaretinin yükselen kenarlarında örneklenerek bit dizisi oluşturulur. Bu bit dizisine Von Neumann Metodu donanım yada yazılım tabanlı uygulanarak bitlerdeki dengesizlik giderilir ve rasgele sayı üretimi gerçekleştirilmiş olur.



Şekil 4.10: Otonom olmayan kaotik sistem ile gerçekleştirilen rasgele sayı üretici yapısı (a) kaotik osilatör, (b) karşılaştırıcı, (c) örnekleyici

Tasarlanan rasgele sayı üreticinin işleyişini göstermek amacıyla, sırasıyla V_{C1} gerilimi, karşılaştırma sonrası elde edilen $\sigma(t)$ işareti, üretilen bitler ve örnekleyiciye uygulanan saat işareti Şekil 4.11’de verilmiştir.

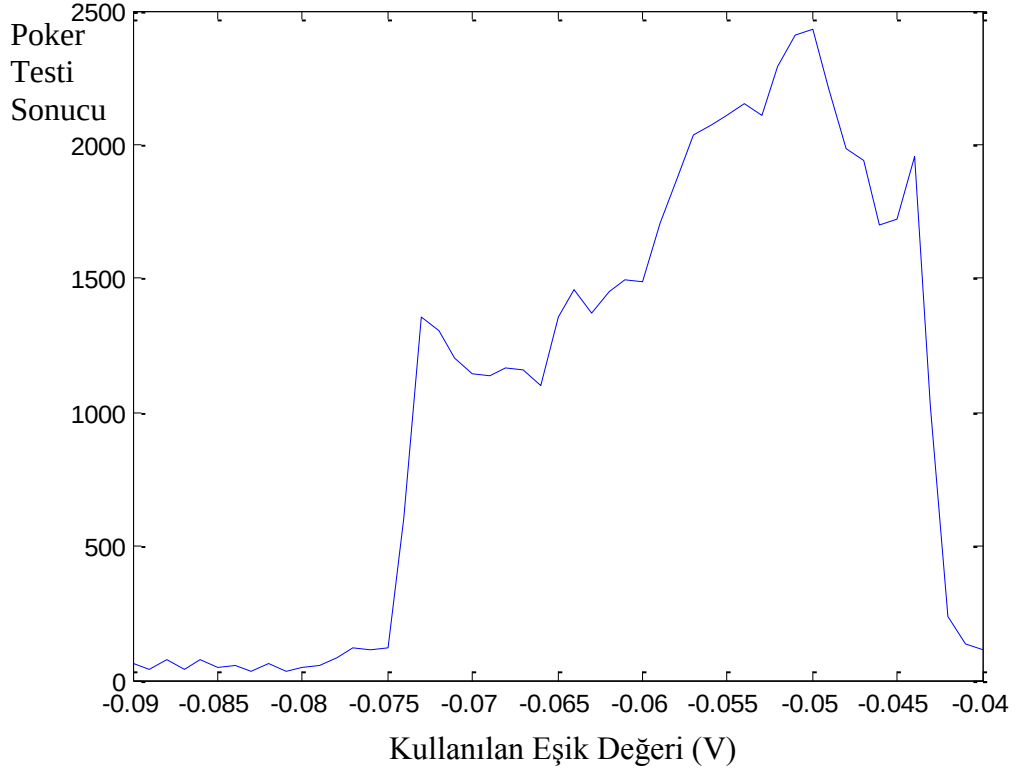


Şekil 4.11: Rasgele sayı üretici yapıda (a) V_{C1} işareti (b) σ işareti (c) σ işaretinin saat işaretinin yükselen kenarlarında örneklenmesiyle elde edilen σ_i işareti

Şekil 4.10’da gösterilen devre, SPICE programı kullanılarak 4µs maksimum zaman aralıklarıyla 200ms boyunca çalıştırılmış ve V_{C1} geriliminin değerleri kaydedilmiştir. Elde edilen 1.98GB büyüklüğündeki veri içinden, periyodun belli anlarındaki değerleri çıkarmak için aşağıda verilen C kodu kullanılmıştır:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
main(int argc, char *argv[]){
    const FILE *inPtr, *outPtr;
    char *satir, *ptr;
    float zaman, deger, peryot_carpani=0, peryot=1E-6, gecikme,
    tolerans=1E-8;
    if(argc!=3){
        printf("Kullanım: Data dosya_adi gecikme(us)\n");
        while (!kbhit());
        exit();}
    if((inPtr=fopen(argv[1], "r"))==NULL){
        printf("Dosya Acilamadi.\n");
        while (!kbhit());
        exit();}
    else printf("Dosya Acildi.\n");
    if((outPtr=fopen("c:\\data\\cikis.txt", "w"))==NULL){
        printf("Cikis Olusturulamadi.\n");
        while (!kbhit());
        exit();}
    else printf("Cikis Dosyasi Olusturuldu.\n");
    gecikme=atof(argv[2])*1E-6;
    fgets(satir, 100, inPtr);
    while((strstr(satir, "#C")==NULL)) fgets(satir, 100, inPtr);
    ptr=strtok(satir, "#C ");
    zaman=atof(ptr);
    while(!feof(inPtr)){
        fgets(satir, 100, inPtr);
        if((strstr(satir, "#C")!=NULL)){
            ptr=strtok(satir, "#C ");
            zaman=atof(ptr);}
        else{
            ptr=strtok(satir, ".");
            deger=atof(ptr);
            if(fabs(zaman- (peryot_carpani*peryot+gecikme))
            <=tolerans){
                fprintf(outPtr, "%f\t", deger);
                peryot_carpani+=1;}}}
    fclose(inPtr);
    fclose(outPtr);
    return 0;}
```

0.25 μ s gecikme değeri için elde edilen değerler ile elde edilen rasgele diziler, farklı eşik değerleri için incelenmiştir. Poker testi sonucunun kullanılan eşik değeri ile değişimi Şekil 4.12’de gösterilmektedir.



Şekil 4.12: Poker testi sonucunun kullanılan eşik gerilimi ile değişimi

Bu inceleme sonucunda en iyi rasgelelik değerlerini veren eşik değeri -82mV olarak belirlenmiştir. Bu değer için poker testi $X=23.31$ sonucunu vermektedir. Tablo 4.3’de görüldüğü gibi, $V_T=-82\text{mV}$ için elde edilen dizi, uygulanan rasgelelik testlerini geçmektedir.

Tablo 4.3: SPICE simülasyonu ile elde edilen dizinin rasgelelik test sonuçları

Eşik Gerilimi	Tek-Bit Testi	Poker Testi	Koşu Testleri	Uzun Koşu Testi
-82mV	9891	23.31	$S_1=2461$ $B_1=2480$ $S_2=1339$ $B_2=1348$ $S_3=677$ $B_3=669$ $S_4=313$ $B_4=333$ $S_5=146$ $B_5=120$ $S_{6+}=140$ $B_{6+}=117$	Uzun Koşu Yok
Test Geçildi?	Evet	Evet	Evet	Evet

5. SONUÇLAR

Gelişen bilgisayar teknolojisi sayesinde kaosun nümerik analiz ile incelenmesi kolaylaşmış ve bu durum kaos araştırmalarında önemli atılımların gerçekleşmesine yol açmıştır. Kaotik osilatörler birçok alanda kullanılmaya başlanmıştır. Bölüm 1’de bu uygulama alanları için rasgele sayı üretimi, şifreleme, mesaj gizleme ve kaotik haberleşme örnekleri verilmiştir.

İkinci dereceden bir osilatöre saat işareti ve lineer olmayan geribesleme eklenmesiyle kaotik bir osilatör elde edilebilmektedir. Bu prensip Bölüm 2’de incelenmiş ve bu yolla elde edilen kaotik çekici verilmiştir. Ayrıca, farklı lineer olmayan fonksiyonlar için sistem denklemlerinin ne tür davranışlar ürettiği incelenmiştir. Çok basamaklı işaret fonksiyonlarının kullanılması sayesinde, çok sayıda sarmala sahip yeni kaotik çekicilerin elde edilebildiği gösterilmiştir. Otonom olmayan kaotik osilatörün ufak değişiklikler ile kaos araştırmalarında büyük öneme sahip olan Duffing Osilatörü’ne nasıl dönüştürülebildiği açıklanmıştır. Kaotik davranışın araştırılmasında mevcut az sayıda yöntemden biri olan Melnikov Metodu kullanılarak diferansiyel denklem sisteminin tek basamaklı ve 3 basamaklı işaret fonksiyonu için kaotik olduğu gösterilmiştir. Bu denklem sisteminden elde edilen dallanma diyagramları ile sistemin zengin dinamik yapısı gözler önüne serilmiştir. Periyot dallanması ile kaosa giren sistemlerde görülen Feigenbaum Sayısının incelenen kaotik osilatörde de mevcut olduğu gözlemlenmiştir.

Bölüm 3’de tümleştirmeye uygun şekilde tasarlanan otonom olmayan kaotik osilatörün detayları verilmiştir. Kaotik osilatörün çekirdeği CMOS 4OTA2C yapısı ile gerçekleştirilmiştir. Bu yapıda sadece OTA ve kapasite elemanları kullanılmaktadır. OTA eğimleri kontrol akımları ile değiştirilerek devre parametreleri arzu edilen şekilde ayarlanmıştır. Devrede kaskod OTA yapısı kullanılarak yüksek frekanslarda çalışma mümkün kılınmıştır. Yapılan simülasyonlarda OTA’nın kapı dirençlerinin 10MHz’e kadar $10M\Omega$ üstünde kaldığı görülmüştür. Devredeki lineer olmayan geribeslemenin gerçekleşmesinde 2 adet OTA kullanılmıştır. Birinci OTA kazancı yükseltme amacı taşımaktadır. İkinci OTA ise, doyma akımı uygun şekilde

ayarlanarak, istenilen geribesleme akımını devreye aktarmaktadır. Geribeslemenin OTA ile gerçekleşmesi sayesinde devrenin çalışma hızının, geribeslemenin yavaş kalması nedeniyle sınırlanması önlenmiştir. Geribesleme akımının, V_{C1} geriliminin işaret değiştirmesine tepki süresi 20ns'den küçüktür. Devrenin SPICE simülasyonu ile elde edilen kaotik çekici yapısı, diferansiyel denklem sisteminin nümerik analizi ile elde edilen çekici ile birebir örtüşmektedir. Devreye uygulanan darbe kaynağının frekansı ile devredeki kapasite elemanı değerinin çarpımı sabit kaldığı sürece aynı kaotik çekiciye ulaşılmaktadır. Bu sayede, sadece kapasite değeri değiştirilerek istenen kaynak frekansı ile kaotik davranış gösterilmesi sağlanabilmektedir.

Giriş bölümünde bahsedilen kaotik osilatör uygulama alanlarından rasgele sayı üretimi konusu Bölüm 4'de detaylı olarak incelenmiş ve Bölüm 3'de verilen kaotik osilatör kullanılarak bir rasgele sayı üretici tasarlanmıştır. Öncelikle otonom kaotik osilatör ile oluşturulan rasgele sayı üretici incelenmiştir. Üretcin rasgeleliğini doğrulamak için kullanılan testler açıklanmış ve bu testleri gerçekleyen MATLAB kodları verilmiştir. Otonom olmayan kaotik osilatör ile rasgele sayı üretiminde Poincaré dönüşümü kullanılarak sistem 2. dereceden ayırık zaman sistemine dönüştürülmüştür. Bu dönüşüm sonrasında iki durum değişkeninden sadece birinin rasgele sayı üretiminde kullanılması ile sistemin tersi alınamaz hale gelmesi sağlanmıştır. Diferansiyel denklem sisteminin nümerik analizi ile saat işaretinin yükselen kenarlarında alınan örneklerin histogramı elde edilmiş ve bu histogram yardımıyla kuantalama için kullanılacak uygun eşik değeri belirlenmiştir. Farklı başlangıç koşullarıyla oluşturulan diziler testleri başarıyla geçmiştir. Tasarlanan üretcin SPICE simülasyonu ile elde edilen dizilerin uygun eşik seçimi sonrasında rasgelelik testlerini geçtiği gösterilmiştir. Tasarlanan devre yapısı ile 200kbps hızında rasgele bit üretilebilmektedir. Kaynak frekansının yükseltilmesi yada osilatör parametrelerinin değiştirilmesiyle bu çıkış hızını daha da yükseltmek mümkündür.

KAYNAKLAR

- [1] **Strogatz, S. H.**, 1994. Nonlinear Dynamics and Chaos, Addison-Wesley Publishing Company, United States of America.
- [2] **Lecar, M., Franklin, F., Holman, M. and Murray, N.**, 2001. Chaos in the Solar System, Annual Review of Astronomy and Astrophysics, **39**, 581-631.
- [3] **Goldberger, A. L., Rigney, D. R. and West, B. J.**, 1990. Chaos and Fractals in Human Physiology, Scientific American , **262 (2)** , 42-49.
- [4] **Parker, T. S., and Chua, L. O.**, 1989. Practical Numerical Algorithms for Chaotic Systems, Springer-Verlag, New York.
- [5] **Bratley, P., Fox, B. L. and Schrage, L. E.**, 1987. A Guide to Simulation, Springer-Verlag, New York.
- [6] **Jacoboni, C. and Lugli, P.**, 1989. The Monte Carlo Method for Semiconductor Device Simulation, Springer-Verlag, Wein.
- [7] **Johansson, A. J. and Floberg, H.**, 1999. Random Number Generation by Chaotic Double Scroll Oscillator on Chip, IEEE International Symposium on Circuits and Systems, June 1999, 407-409.
- [8] **Baptista, M. S.**, 1998. Cryptography with Chaos, Physics Letters A, **240**, 50-54.
- [9] **Pecora, L. M. and Carroll, T. L.**, 1990. Synchronization in Chaotic Systems, Physical Review Letters, **64**, 821-824.
- [10] **Cuomo, K. M., Oppenheim, A. V.**, 1993. Circuit Implementation of Synchronized Chaos with Applications to Communications, Physical Review Letters, **71**, 65-68.
- [11] **Abel, A. and Schwarz, W.**, 2002. Chaos Communications – Principles, Schemes, and System Analysis, Proceedings of the IEEE, **90 (5)**, 691–710.
- [12] **Özoğuz, S. and Elwakil, A. S.**, 2004. On the Realization of Circuit-Independent Nonautonomous Pulse-Excited Chaotic Oscillator Circuits, IEEE Transactions on Circuits and Systems-II: Express Briefs, **51 (10)**, 552-556.

- [13] **Parlitz, U. and Lauterborn, W.**, 1985. Superstructure in the bifurcation set of the Duffing equation, *Physics Letters A*, **107 (8)**, 351-355 .
- [14] **Guckenheimer, J. and Holmes, P.**, 1990. *Nonlinear Oscillations Dynamical Systems and Bifurcations of Vector Fields*, Springer Verlag, New York.
- [15] **Peitgen, H. O., Jürgens, H. and Saupe, D.**, 1992. *Chaos and Fractals: New Frontiers of Science*, Springer-Verlag, New York.
- [16] **Kuntman, H. H.**, 1997. *Analog MOS Tümdevre Tekniği*, İstanbul Teknik Üniversitesi Elektrik-Elektronik Fakültesi Ofset Baskı Atölyesi, İstanbul.
- [17] **Lorenz, E. N.**, 1995. *The Essence of Chaos*, University of Washington Press, Seattle.
- [18] **Yalcin, M. E., Suykens, J. A. K. and Vandewalle, J.**, 2004. True Random Bit Generation from a Double Scroll Attractor , *IEEE Transactions on Circuits and Systems-I*, **51 (7)**, 1395-1404.
- [19] **Kocarev, L. and Jakimoski, G.**, 2003. Pseudorandom Bits Generated by Chaotic Maps, *IEEE Transactions on Circuits and Systems I*, **50 (1)**, 123-126.
- [20] **FIPS 140-1**, 1994. *Security Requirements for Cryptographic Modules*, National Institute of Standards and Technology, United States of America.

ÖZGEÇMİŞ

Serkan Semiz 08.05.1981 Kocaeli doğumludur. İlk, orta ve lise eğitimini Kocaeli’de tamamlamıştır. 1998 yılında girdiği İstanbul Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü’nden 2002 yılında mezun olmuştur. Aynı yıl İstanbul Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü Elektronik Mühendisliği programında yüksek lisans eğitimine başlamıştır.