

**T.C.  
ERCIYES ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**KAOTİK SİSTEM TABANLI SÖZDE RASTGELE SAYI  
ÜRETEÇLERİNİN TASARIMI VE  
GERÇEKLEŞTİRİLMESİ  
(Yüksek Lisans Tezi)**

**Hazırlayan  
İsmail ÖZTÜRK**

**Danışman  
Prof. Dr. Recai KILIÇ**

**Temmuz 2013  
KAYSERİ**

## BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

İsmail ÖZTÜRK

İmza : 

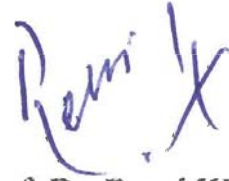
## YÖNERGEYE UYGUNLUK

**“Kaotik Sistem Tabanlı Sözde Rastgele Sayı Üreteçlerinin Tasarımı ve Gerçekleştirilmesi”** adlı Yüksek Lisans tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ne uygun olarak hazırlanmıştır.



İsmail ÖZTÜRK

Tezi Hazırlayan



Prof. Dr. Recai KILIÇ

Danışman



Prof. Dr. Necmi TAŞPINAR

Elektrik - Elektronik Mühendisliği ABD Başkanı

Prof. Dr. Recai KILIÇ danışmanlığında **İsmail ÖZTÜRK** tarafından hazırlanan “**Kaotik Sistem Tabanlı Sözde Rastgele Sayı Üreteçlerinin Tasarımı ve Gerçekleştirilmesi**” adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü **Elektrik-Elektronik Mühendisliği** Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

19.07.2013

**JÜRİ:**

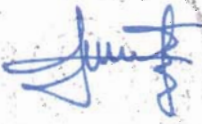
Başkan: Prof. Dr. Kenan DANIŞMAN

Üye : Prof. Dr. Recai KILIÇ

Üye : Prof. Dr. Derviş KARABOĞA

**ONAY:**

Bu tezin kabulü, Enstitü Yönetim Kurulunun 26/07/2013 tarih ve 2013/32-16 sayılı kararı ile onaylanmıştır.

26.07.2013  


Prof. Dr. Kazım KEŞLİOĞLU  
Enstitü Müdürü

## TEŞEKKÜR

Bu tezin yazımına gelene kadar geçen uzun süreçte ve sonrasında tecrübesi ve bilgi birikimiyle bana sürekli yol gösteren tez danışmanım Prof. Dr. Recai KILIÇ'a öncelikli olarak teşekkürlerimi sunarım.

Çalışmalarım boyunca yardımlarını esirgemeyen Arş. Gör. Erdem ARSLAN'a, Arş. Gör. Nimet KORKMAZ'a, son ana kadar beni yalnız bırakmayan Arş. Gör. Selim SOYLU ile Arş. Gör. Ökkeş Gökalp SÖKMEN'e ve adını burada sayamadığım tüm çalışma arkadaşlarıma teşekkürü bir borç bilirim.

Laboratuvar ve bilgisayar desteği sağlayarak çok uzun süre gerektiren simülasyonları kısa sürede yapmamı sağlayan Yrd. Doç. Dr. Selami PARMAKSIZOĞLU ve Yrd. Doç. Dr. Rıfat KURBAN'a da ayrıca teşekkürlerimi sunarım.

Son olarak beni bu güne kadar hep desteklemiş olan, bu süreçte de desteklerini esirgemeyen ve yoğunlaşan çalışmalarım sırasında sabır gösteren aileme en içten teşekkürlerimi sunar; kardeşim Figen ÖZTÜRK'e de ayrıca teşekkür ederim.

İsmail ÖZTÜRK  
Kayseri, Temmuz 2013

# **KAOTİK SİSTEM TABANLI SÖZDE RASTGELE SAYI ÜRETEÇLERİNİN TASARIMI VE GERÇEKLEŞTİRİLMESİ**

**İsmail ÖZTÜRK**

**Erciyes Üniversitesi, Fen Bilimleri Enstitüsü**

**Yüksek Lisans Tezi, Temmuz 2013**

**Tez Danışmanı: Prof. Dr. Recai KILIÇ**

## **ÖZET**

Kaos teorisi ile kriptografi arasındaki yakın ilişki neticesinde pek çok kaos tabanlı sözde rastgele sayı üretici literatüre sunulmuştur. Bu üreteçler genel olarak simetrik ve açık anahtarlamalı kriptografide akış ya da blok şifreleyici olarak kullanılmak üzere önerilmişlerdir. Kaotik haritalar, ayrık zamanlı yapıları ve kolay gerçekleştirim olanakları sayesinde bu üreteçlerin yapısında sıklıkla tercih edilmiştir. Fakat bu tür rastgele sayı üreteçlerinin donanım gerçekleştirmeleri kaos teorisinin tüm özelliklerini taşımamaktadır. Sonlu çözünürlük altında kısa periyotların oluşması ve farklı başlangıç şartları için birebir aynı periyotların elde edilmesi gibi problemlerle karşılaşmaktadır. Bu problemlerin çözümü için bit çözünürlüğünü arttırma, kaotik haritaları kaskat bağlama ve pertürbasyon gibi çeşitli çözümler önerilmiştir. Bu sorunların göz ardı edildiği ve herhangi bir çözümün sunulmadığı bir kaos tabanlı rastgele sayı üretici tasarımının donanımsal olarak gerçekleştirimi beklenenin aksine oldukça güvensiz ve verimsiz olacaktır.

Bu tez çalışmasında tüm bu unsurlar dikkate alınarak kaos tabanlı bir rastgele sayı üreticinin ne tür özellikleri sağlaması gerektiği, hangi tür haritaların iyi bir tercih olacağı, belirtilen problemlerin çözümü için hangi metodun uygulanması gerektiği gibi konular tartışılmaktadır. Elde edilen sonuçlara bağlı olarak literatürdeki üreteçler arasından gereksinimleri en iyi şekilde karşılayanlardan birinin FPGA (Field Programmable Gate Array) kullanılarak donanımsal gerçekleştirimi ve görüntü şifrelemede kullanımı sunulmuştur. Son olarak ise sürekli zamanlı kaotik sistemlerin ayrıklaştırılmasıyla elde edilmiş bir sözde rastgele sayı üretici tasarımı önerilmiş ve kaotik haritaların kullanıldığı üreteçlere göre avantaj ve dezavantajları tartışılmıştır.

**Anahtar Kelimeler:** Kaos, Kriptografi, Kaotik Haritalar, Sözde Rastgele Sayı Üreteçleri, FPGA.

# **DESIGN AND IMPLEMENTATION OF CHAOTIC SYSTEM BASED PSEUDO RANDOM NUMBER GENERATORS**

**İsmail ÖZTÜRK**

**Erciyes University, Graduate School of Natural and Applied Sciences**

**M. Sc. Thesis, July 2013**

**Thesis Supervisor: Prof. Recai KILIÇ**

## **ABSTRACT**

Many chaos based pseudo random number generators are presented into the literature due to the close relationship between chaos and cryptography. These generators are usually proposed to be used as stream or block ciphers in symmetric and public key cryptography. Mostly, chaotic maps are preferred in the structures of these generators because of their discrete time nature and easy implementation possibilities. However, hardware implementations of such random number generators do not possess all the aspects of chaos theory. Under finite precision, problems such as occurrence of short cycles and obtaining identical cycles for different initial conditions are encountered. Different remedies like increasing bit resolution, cascading multiple chaotic maps and perturbation are proposed for the solution of these problems. Implementation of a chaos based random number generator design which ignores these problems and does not propose a solution will be quite unsecure and inefficient contrary to expectations.

In this thesis, with considering all these facts, issues such as what properties a chaos based random number generator should provide, which type of chaotic maps would be a good choice, which method should be applied for solution of specified problems are discussed. Depending on the obtained results, one of the generators that satisfy the requirements to the best is implemented as hardware using FPGA (Field Programmable Gate Array) and its usage in an image encryption application is presented. Finally, design of a pseudo random number generator obtained by discretization of continuous time chaotic systems are proposed and its advantages and disadvantages according to the generators that use chaotic maps are discussed.

**Keywords:** Chaos, Cryptography, Chaotic Maps, Pseudo Random Number Generators, FPGA.

## İÇİNDEKİLER

### KAOTİK SİSTEM TABANLI SÖZDE RASTGELE SAYI ÜRETEÇLERİNİN TASARIMI VE GERÇEKLEŞTİRİLMESİ

|                               |       |
|-------------------------------|-------|
| BİLİMSEL ETİĞE UYGUNLUK ..... | ii    |
| YÖNERGEYE UYGUNLUK .....      | iii   |
| KABUL VE ONAY .....           | iv    |
| TEŞEKKÜR .....                | v     |
| ÖZET .....                    | vi    |
| ABSTRACT .....                | vii   |
| İÇİNDEKİLER .....             | viii  |
| TABLolar LİSTESİ .....        | xii   |
| ŞEKİLLER LİSTESİ .....        | xiii  |
| KISALTMALAR VE SİMGELER ..... | xvi   |
| <br>GİRİŞ .....               | <br>1 |

## 1. BÖLÜM GENEL BİLGİLER

|                                                             |   |
|-------------------------------------------------------------|---|
| 1.1. Gerçek Rastgele Sayı Üreteçleri (GRSÜ) .....           | 3 |
| 1.1.1. Kaotik Sistem Tabanlı GRSÜ'ler .....                 | 6 |
| 1.2. Sözde Rastgele Sayı Üreteçleri (SRSÜ) .....            | 6 |
| 1.2.1. Doğrusal Eşleşik Üreteçler .....                     | 6 |
| 1.2.2. Doğrusal Geribeslemeli Kaydırmalı Kaydediciler ..... | 7 |
| 1.2.3. Kaotik Sistem Tabanlı SRSÜ'ler .....                 | 8 |
| 1.3. Hibrit Rastgele Sayı Üreteçleri (HRSÜ) .....           | 9 |
| 1.4. İstatistiki Rastgelelik Testleri .....                 | 9 |

## 2. BÖLÜM

### KAOTİK HARİTALAR

|                                                                                |           |
|--------------------------------------------------------------------------------|-----------|
| <b>2.1. Kaotik Haritaların Genel Özellikleri .....</b>                         | <b>12</b> |
| 2.1.1. Kaotik Haritaların Yörüngeleri .....                                    | 13        |
| 2.1.2. Kaotik Haritalarda Lyapunov Üstelleri .....                             | 14        |
| 2.1.3. Kaotik Haritaların Tanım Aralığı .....                                  | 17        |
| 2.1.4. Rastgele Benzeri Çıkış Serisi .....                                     | 18        |
| 2.1.5. Feigenbaum Sabiti .....                                                 | 20        |
| 2.1.6. Fraktal Boyut .....                                                     | 21        |
| 2.1.7. Ergodiklik ve Topolojik Harmanlama .....                                | 23        |
| <b>2.2. Mevcut Kaotik Haritalar .....</b>                                      | <b>24</b> |
| 2.2.1. Lojistik Harita .....                                                   | 24        |
| 2.2.2. Tent (Çadır) Harita .....                                               | 25        |
| 2.2.3. Parçalı Doğrusal Kaotik Harita (PDKH) .....                             | 26        |
| 2.2.4. Henon Haritası .....                                                    | 28        |
| <b>2.3. Kaotik Haritaların Rastgele Sayı Üretici Olarak Kullanılması .....</b> | <b>29</b> |
| 2.3.1. Kaos ve Kriptoloji İlişkisi .....                                       | 29        |
| 2.3.2. Kaotik Sistem Tabanlı Kriptografik Şifreleyiciler .....                 | 30        |

## 3. BÖLÜM

### KAOTİK SİSTEM TABANLI SÖZDE RASTGELE SAYI ÜRETEÇLERİNİN GERÇEKLEŞTİRİMLERİ

|                                                                                   |           |
|-----------------------------------------------------------------------------------|-----------|
| <b>3.1. Sözde Kaos ve Sözde Kaotik Yörüngeler .....</b>                           | <b>34</b> |
| <b>3.2. Sözde Kaosun Özellikleri .....</b>                                        | <b>35</b> |
| 3.2.1. Yörüngeler .....                                                           | 36        |
| 3.2.2. Lyapunov Üstelleri .....                                                   | 37        |
| 3.2.3. Ergodiklik ve Topolojik Harmanlama .....                                   | 37        |
| <b>3.3. Kaos Tabanlı SRSÜ'lerin Gerçekleştirim Problemleri ve Çözümleri .....</b> | <b>38</b> |
| 3.3.1. Bit Çözünürlüğünü Arttırmak .....                                          | 39        |
| 3.3.2. Kaotik Haritaları Kaskat Bağlamak .....                                    | 39        |

|                                                                                  |    |
|----------------------------------------------------------------------------------|----|
| 3.3.3. Kaotik Haritaları Pertürbe Etmek .....                                    | 40 |
| 3.4. Periyot Oluşum Süreci ve Üretilen Periyotların Korelasyon Özellikleri ..... | 41 |
| 3.4.1. Periyot Uzunluklarının Nümerik Olarak Hesaplanması .....                  | 43 |
| 3.4.2. Sabit Noktalı Aritmetik ve Yuvarlama Türleri .....                        | 44 |
| 3.4.3. Farklı Çözünürlük ve Yuvarlama Türleri için Periyot Uzunlukları ..        | 45 |
| 3.4.4. Eşit Uzunluklu Periyotların Benzerlik Açısından Kıyaslanması .....        | 48 |
| 3.4.5. Transient Uzunluklarının Benzerlik Açısından Kıyaslanması .....           | 52 |
| 3.5. Sözde Kaotik Yörüngelerin Kriptosistemlerde Kullanımı .....                 | 53 |
| 3.5.1. Pertürbasyonsuz Eşleştirilmiş Kaotik Sistem .....                         | 53 |
| 3.5.2. Pertürbasyonlu Eşleştirilmiş Kaotik Sistem .....                          | 57 |

#### 4. BÖLÜM

##### PERTÜRBASYONA DAYALI KAOTİK SÖZDE RASTGELE SAYI ÜRETECİNİN DONANIM GERÇEKLEŞTİRİMİ VE ŞİFRELEMEDE KULLANIMI

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| 4.1. Pertürbasyona Dayalı Kaotik Sözde Rastgele Sayı Üretici .....           | 59 |
| 4.1.1. Pertürbasyon İçin Kullanılan SRSÜ'ler .....                           | 60 |
| 4.1.2. Pertürbasyon Aralığı ve Pertürbe Edilecek Bit Uzunluğu .....          | 61 |
| 4.1.3. Sistemin Periyot Uzunluğu .....                                       | 62 |
| 4.2. PDKH'nın Donanım Gerçekleştirimi .....                                  | 62 |
| 4.3. Eşleştirilmiş Kaotik Sistemin Donanım Gerçekleştirimi .....             | 64 |
| 4.4. Eşleştirilmiş Kaotik Sistemin Görüntü Şifrelemede Kullanımı .....       | 65 |
| 4.5. Görüntü Şifreleme Uygulamasının Donanımsal Olarak Gerçekleştirimi ..... | 68 |

#### 5. BÖLÜM

##### SÜREKLİ ZAMANLI KAOTİK SİSTEMLERE DAYALI KARIŞIK MODLU SÖZDE RASTGELE SAYI ÜRETECİ TASARIMI VE UYGULAMASI

|                                                                 |    |
|-----------------------------------------------------------------|----|
| 5.1 Sürekli Zamanlı Kaotik Sistemlerin Ayırıklaştırılması ..... | 71 |
| 5.2 Karışık Modlu Kaotik Sistem .....                           | 74 |
| 5.3 Lorenz Ailesinin Karışık Modlu Olarak Kullanılması .....    | 75 |
| 5.4 Karışık Modlu Sözde Rastgele Sayı Üretici Tasarımı .....    | 79 |

|                                                                                       |           |
|---------------------------------------------------------------------------------------|-----------|
| <b>5.5 Karışık Modlu Sözde Rastgele Sayı Üreticinin Donanım Gerçekleştirimi .....</b> | <b>82</b> |
| <b>5.6 Karışık Modlu Üreticinin Görüntü Şifrelemede Kullanımı .....</b>               | <b>83</b> |
| <b>5.7 Karışık Modlu Üreticinin Avantaj ve Dezavantajları .....</b>                   | <b>85</b> |

## 6. BÖLÜM

### SONUÇ VE ÖNERİLER

|                                |            |
|--------------------------------|------------|
| <b>Sonuç ve Öneriler .....</b> | <b>87</b>  |
| <b>KAYNAKLAR .....</b>         | <b>90</b>  |
| <b>ÖZGEÇMİŞ.....</b>           | <b>101</b> |

## TABLolar LİSTESİ

|            |                                                                                                                           |    |
|------------|---------------------------------------------------------------------------------------------------------------------------|----|
| Tablo 3.1. | Farklı bit çözünürlükleri ve yuvarlama türleri için elde edilen periyot uzunlukları .....                                 | 47 |
| Tablo 3.2. | 44 bit çözünürlüklü gerçekleştirim için periyotların karşılaşılma sayıları ve bunlar içerisindeki farklı periyotlar ..... | 50 |
| Tablo 3.3. | Eşleştirilmiş kaotik sistemin NIST SP 800-22 rastgelelik testi sonuçları ....                                             | 55 |
| Tablo 5.1. | Karışık modlu sözde rastgele sayı üretici için NIST rastgelelik testi sonuçları .....                                     | 82 |
| Tablo 5.2. | Sentezleme sonuçlarının karşılaştırılması .....                                                                           | 83 |

## ŞEKİLLER LİSTESİ

|             |                                                                                                                                       |    |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------|----|
| Şekil 1.1.  | Basit bir GRSÜ'nün blok yapısı .....                                                                                                  | 3  |
| Şekil 1.2.  | Jitter olayı [15] .....                                                                                                               | 5  |
| Şekil 1.3.  | Halka osilatörler kullanarak jitter tabanlı rastgele sayı üretimi [15] .....                                                          | 5  |
| Şekil 1.4.  | Tipik bir doğrusal geribeslemeli kaydırmalı kaydedici yapısı .....                                                                    | 7  |
| Şekil 1.5.  | HRSÜ'nün temel yapısı .....                                                                                                           | 9  |
| Şekil 2.1.  | Kaotik yörüngeler, a) Ayrık zamanlı kaotik harita için, b) Sürekli zamanlı kaotik sistem için .....                                   | 13 |
| Şekil 2.2.  | Kaotik haritalarda yörünge oluşumunun örümcek ağı (cobweb) çizimi ile gösterilmesi .....                                              | 14 |
| Şekil 2.3.  | Birbirine çok yakın iki noktanın kaotik fonksiyon ile değişimi .....                                                                  | 15 |
| Şekil 2.4.  | Birbirine çok yakın iki kaotik yörünge zamanla değişimi .....                                                                         | 17 |
| Şekil 2.5.  | Bernoulli haritasında esnetme ve katlama işlemi .....                                                                                 | 18 |
| Şekil 2.6.  | Periyot katlama bifürkasyonu .....                                                                                                    | 21 |
| Şekil 2.7.  | Menger Süngeri .....                                                                                                                  | 22 |
| Şekil 2.8.  | Tekdüze dağılımlı kaotik bir haritanın histogramı .....                                                                               | 24 |
| Şekil 2.9.  | Lojistik haritanın çekişi .....                                                                                                       | 25 |
| Şekil 2.10. | $k = 3.9$ için lojistik haritanın çıkış serisi dağılımı .....                                                                         | 25 |
| Şekil 2.11. | Tent haritanın çekişi .....                                                                                                           | 26 |
| Şekil 2.12. | Tent haritanın bifürkasyon diyagramı .....                                                                                            | 26 |
| Şekil 2.13. | PDKH'nın $p = 0.2$ için kaotik çekişi .....                                                                                           | 27 |
| Şekil 2.14. | Parçalı doğrusal kaotik haritanın (PDKH) bifürkasyon diyagramı .....                                                                  | 27 |
| Şekil 2.15. | Henon haritasının kaotik çekişi .....                                                                                                 | 28 |
| Şekil 2.16. | Henon haritasının $a$ parametresine göre bifürkasyon diyagramı .....                                                                  | 29 |
| Şekil 2.17. | Kaotik sistem tabanlı bir akış şifreleyicinin blok diyagramı .....                                                                    | 32 |
| Şekil 3.1.  | Sözde kaotik yörünge yapısı [62] .....                                                                                                | 36 |
| Şekil 3.2.  | İki tane kaotik haritanın kaskat bağlanması [95] .....                                                                                | 40 |
| Şekil 3.3.  | Lojistik harita gerçekleştiriminin farklı çözünürlükler için a) maksimum periyot uzunlukları, b) maksimum transient uzunlukları ..... | 45 |
| Şekil 3.4.  | İki sözde kaotik yörünge için a) baştan itibaren, b) $\tau_1 + 1$ ve $\tau_2 + 1$ noktalarından itibaren farkı .....                  | 48 |
| Şekil 3.5.  | İki sözde kaotik yörünge için kros-korelasyonu .....                                                                                  | 49 |

|             |                                                                                                                                       |    |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------|----|
| Şekil 3.6.  | İki sözde yörünge'nin transient bölgesinde kesişmesi [108] .....                                                                      | 50 |
| Şekil 3.7.  | İki sözde yörünge'nin periyot bölgesinde kesişmesi .....                                                                              | 51 |
| Şekil 3.8.  | Elde edilen sözde yörüngelerin transient uzunluklarının histogramı .....                                                              | 52 |
| Şekil 3.9.  | Eşleştirilmiş kaotik sistem [34] .....                                                                                                | 54 |
| Şekil 3.10. | Periyodu aynı olan iki sözde yörünge karşılaştırıldığında eşdeğer yapı .....                                                          | 55 |
| Şekil 3.11. | 56 bit çözünürlüklü gerçekleştirim için a) kaotik haritaların çıkışının kros-korelasyonu, b) çıkış bit dizisinin otokorelasyonu ..... | 56 |
| Şekil 4.1.  | Pertürbasyona dayalı eşleştirilmiş kaotik sistem [29] .....                                                                           | 60 |
| Şekil 4.2.  | PDKH çıkış serisi için elde edilen TISEAN sonucu .....                                                                                | 61 |
| Şekil 4.3.  | PDKH'nın FPGA program yapısının blok gösterimi .....                                                                                  | 63 |
| Şekil 4.4.  | PDKH'nın donanım gerçekleştiriminin a) zaman domenindeki, b) faz domenindeki osiloskop görüntüleri .....                              | 63 |
| Şekil 4.5.  | Eşleştirilmiş kaotik sistemin çıkışının osiloskop görüntüsü .....                                                                     | 64 |
| Şekil 4.6.  | $k_i$ çıkış serisi için elde edilmiş, a) otokorelasyon, b) kros-korelasyon bağıntıları .....                                          | 65 |
| Şekil 4.7.  | Çift port RAM'in kullanıldığı şifreleme mekanizmasının Simulink modeli .....                                                          | 66 |
| Şekil 4.8.  | Sistem başlar başlamaz yapılacak şifrelemede a) 0.03 saniye, b) 0.05 saniye, c) 0.09 saniye sonraki çıkış görüntüleri .....           | 67 |
| Şekil 4.9.  | a) Orijinal görüntü, b) orijinal görüntünün histogramı, c) şifrelenmiş görüntü, d) şifrelenmiş görüntünün histogramı .....            | 68 |
| Şekil 4.10. | Görüntü şifreleme uygulaması için deney düzeneği .....                                                                                | 69 |
| Şekil 4.11. | Donanım gerçekleştirimi sonucu elde edilen a) şifreleme yokkenki ekran görüntüsü, b) şifreleme aktifkenki ekran görüntüsü .....       | 70 |
| Şekil 5.1.  | Denklem (5.8) ile verilen kaotik sistemin a) İlk 100000 adım için simülasyonu, b) 75000 adımdan sonraki simülasyonu .....             | 73 |
| Şekil 5.2.  | Karışık modlu kaotik devre .....                                                                                                      | 74 |
| Şekil 5.3.  | Chua sistemine ait $(x,y,z) = (3.5, -3.5, 0)$ başlangıç şartı için elde edilen yörünge .....                                          | 76 |
| Şekil 5.4.  | Lorenz sisteminin kaotik çekişi .....                                                                                                 | 77 |
| Şekil 5.5.  | Chen sisteminin kaotik çekişi .....                                                                                                   | 78 |
| Şekil 5.6.  | Lü sisteminin a) $c = 13$ için, b) $c = 28$ için kaotik çekişi .....                                                                  | 78 |
| Şekil 5.7.  | Karışık modlu kullanılacak sistem için a) $r = 4.5$ için, b) $r = 7$ için elde edilen çeker görüntüleri .....                         | 79 |

|             |                                                                                                                               |    |
|-------------|-------------------------------------------------------------------------------------------------------------------------------|----|
| Şekil 5.8.  | Sözde rastgele sayı üreticinin Matlab modeli .....                                                                            | 80 |
| Şekil 5.9.  | Üretilen sözde rastgele sayıların histogramı .....                                                                            | 81 |
| Şekil 5.10. | Karışık modlu çeker yapısı .....                                                                                              | 82 |
| Şekil 5.11. | Karışık modlu sözde rastgele sayı üreticinin çıkışının osiloskop görüntüsü<br>.....                                           | 83 |
| Şekil 5.12. | Karışık modlu üreticinin çıkış serisi için elde edilmiş a) otokorelasyon, b)<br>kros-korelasyon bağıntıları .....             | 84 |
| Şekil 5.13. | a) Orijinal görüntü, b) orijinal görüntünün histogramı, c) şifrelenmiş<br>görüntü, d) şifrelenmiş görüntünün histogramı ..... | 85 |
| Şekil 5.14. | Donanım gerçekleştirimi sonucu elde edilen a) orijinal ekran görüntüsü, b)<br>şifrelenmiş ekran görüntüsü .....               | 86 |

## KISALTMALAR VE SİMGELER

|            |                                                                   |
|------------|-------------------------------------------------------------------|
| GRSÜ       | Gerçek Rastgele Sayı Üreteci                                      |
| SRSÜ       | Sözde Rastgele Sayı Üreteci                                       |
| HRSÜ       | Hibrit Rastgele Sayı Üreteci                                      |
| CMOS       | Komplementer Metal-oksit Yarıiletken                              |
| NIST       | National Institute of Standards and Technology                    |
| FFT        | Hızlı Fourier Dönüşümü                                            |
| FPGA       | Field Programmable Gate Array                                     |
| PDKH       | Parçalı Doğrusal Kaotik Harita                                    |
| SNR        | Sinyal Gürültü Oranı                                              |
| LFSR       | Lineer Geribeslemeli Kaydırmalı Kaydedici                         |
| IEEE       | Institute of Electrical and Electronics Engineers                 |
| LCG        | Lineer Eşleşik Üreteç                                             |
| VHDL       | Very High Speed Integrated Circuits Hardware Description Language |
| Mbps       | Megabit per Second                                                |
| Gbps       | Gigabit per Second                                                |
| RAM        | Random Access Memory                                              |
| SIPO       | Serial in Parallel Out                                            |
| VGA        | Video Graphics Array                                              |
| PLL        | Phase Locked Loop                                                 |
| AES        | Advanced Encryption Standart                                      |
| DES        | Data Encryption Standart                                          |
| $\lambda$  | Lyapunov Üsteli                                                   |
| $\delta$   | Feigenbaum Sabiti                                                 |
| $D_{KY}$   | Kaplan-Yorke Boyutu                                               |
| $\delta$   | Birim İmpuls                                                      |
| $\tau$     | Transient                                                         |
| $T$        | Periyot                                                           |
| $\Delta$   | Pertürbasyon Aralığı                                              |
| $\Delta_i$ | Anahtarlama Aralığı                                               |

## GİRİŞ

Rastgele sayı üretimi, kriptografik uygulamalar için temel bir gereksinim olmasının yanı sıra, Monte Carlo metodunun kullanıldığı bilimsel araştırmalarda, istatistiksel örneklemede, bilgisayar simülasyonlarında ve günlük hayattaki şans oyunlarında sıklıkla kullanılmaktadır. Bu uygulamalardaki gereksinimleri karşılamak üzere sürekli daha hızlı ve daha güvenli rastgele sayı üreteçleri modellenmeye çalışılmış ve gelişen teknolojiyle birlikte yeni üreteçler pratik uygulamalarda kullanıma geçerken, değişime ayak uyduramayanlar zamanla terk edilmiştir. Bu süreç içerisinde gelişimini sürdüren kaos teorisi ise teorik altyapısının olgunlaşmasından sonra, asla kendini tekrar etmeyen rastgele çıkışlı kaos üreteçlerinin elektronik devre olarak donanımsal gerçekleştirimlerinin mümkün olduğunun gösterilmesiyle birlikte rastgele sayı üretiminde kullanılmak üzere araştırmacıların yoğun ilgisini çekmiş ve bu konuda yoğun bir şekilde çalışmaların ortaya konulmasına neden olmuştur.

Kaotik sistemlerin rastgele çıkışları, iki farklı kaotik devrenin senkronize edilebileceğinin anlaşılmasından sonra, analog gerçekleştirimlere dayalı olarak güvenli analog haberleşme sağlamak için kullanılmış ve bu bağlamda pek çok model önerilerek çeşitli donanım uygulamaları gerçekleştirilmiştir. Sonraları analog sistemlerdeki bu rastgelelik özelliğinin dijital sistemlerde de kullanılabileceği düşüncesinin yaygınlaşmasıyla beraber özellikle ayrık zamanlı olan kaotik haritalara dayalı kaotik sistemlerin rastgele sayı üretiminde kullanılmasına yönelik çalışmalar ortaya konmuştur. Bu çalışmalar, kaos ile kriptografi arasında çok yakın bir ilişki olduğunu göstermiş ve kaotik sistem tabanlı rastgele sayı üreteçlerinin hem kompleks çıkışları hem de gerçekleştirim kolaylıkları nedeniyle kriptografik uygulamalarda mükemmel performans göstereceği düşünülmüştür. Bu bağlamda çok fazla miktarda kaotik sistem tabanlı rastgele sayı üretici literatüre sunulmuş olup bunlar çoğunlukla akış veya blok

şifreleyici olarak kriptografik uygulamalarda kullanılmak üzere önerilmiştir. Fakat teorik olarak mükemmel güvenliğe sahip olacağı düşünülen bu üreteçlerin pek çoğunun aslında sunuldukları kadar güvenli olmadıklarının gösterilmesiyle bu konuda bir takım problemlerin aşılmasının gerektiği ortaya çıkmıştır. Günümüze kadar bu problemlerin bazıları çözülmüş olsa da bazıları hala çözülmemiş olup bunların uygulamalarda güvenlik açığı oluşturup oluşturmayacağı teorik çalışmalardan ziyade bir takım mühendislik çözümlerine bağlı hale gelir olmuştur. Yine de, bu konudaki çalışmalar tüm canlılığıyla devam etmekte olup her geçen gün yeni bir kaos tabanlı rastgele sayı üretici literatüre sunulurken bu konuda yeni patentler alınmaktadır.

Bu tez çalışması literatürde mevcut olan kaos tabanlı rastgele sayı üreteçlerinin nasıl özelliklere sahip olması gerektiğini, bunların ne tür kaotik sistem kullanması gerektiğini ve kriptografik uygulamalarda kullanılmak üzere nelerin yapılması gerektiğini ve nelerin yapılmaması gerektiğini detaylıca irdelemektedir. Bu bağlamda 1. Bölüm’de rastgele sayı üretimi ve kaos tabanlı rastgele sayı üretimi hakkında genel bilgiler verilmektedir. 2. Bölüm’de ise kaos tabanlı rastgele sayı üretiminde kaotik üreteç olarak kullanılan kaotik haritalar genel özellikleriyle tanıtılmış olup rastgele sayı üretiminde ne tür kaotik haritaların kullanılması gerektiğinden bahsedilmiştir. Bu bölümün sonunda da kaos ile kriptografi arasındaki yakın ilişki açıklanmış ve kaotik sistemlerin kriptografik uygulamalarda nasıl kullanılacağına değinilmiştir. Kaotik haritaların dijital sistemler üzerinde nasıl gerçekleştirildiği ve bu esnada karşılaşılan problemlerin neler olduğu ise 3. Bölüm’de ele alınmıştır. Bir takım unsurlar göz önünde bulundurulmadığında güvenli olduğu varsayılan bir rastgele sayı üreticinin aslında ne kadar güvensiz olabileceği yine bu bölümde tartışılmıştır. Elde edilen sonuçlara bağlı olarak bu konudaki gereksinimleri en iyi şekilde karşılayan ve literatürdeki muadillerine nazaran öne çıkan bir kaotik rastgele sayı üreticinin FPGA (Field Programmable Gate Array) kullanılarak donanımsal olarak gerçekleştirimi ve akış şifreleyici olarak görüntü şifrelemede kullanımı 4. Bölüm’de sunulmaktadır. 5. Bölüm’de ise literatürdeki çoğu çalışmanın aksine kaotik haritalar yerine sürekli zamanlı kaotik sistemlerin ayrıklaştırılmasına dayanan ve yeni bir tasarımla rastgele sayı üreten kaotik bir üreteç önerilecek ve bu modelin kaotik harita tabanlı üreteçlere karşı avantaj ve dezavantajları tartışılacaktır. Son olarak 6. Bölüm’de sonuç ve öneriler özetlenmektedir.

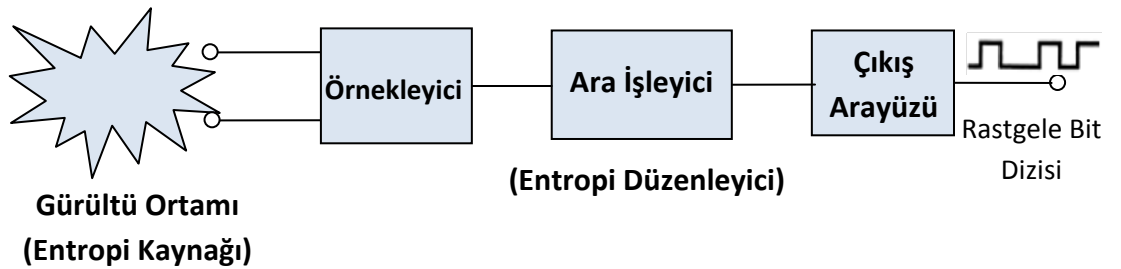
## 1. BÖLÜM

### GENEL BİLGİLER

Monte Carlo metotlarını gerektiren uygulamalarda, şans oyunlarında ve kriptografide kullanılmak üzere çeşitli rastgele sayı üreteçleri (RSÜ) veya rastgele bit üreteçleri (RBÜ) literatüre sunulmuştur. Bunlar genel olarak Gerçek Rastgele Sayı Üreteçleri (GRSÜ), Sözde Rastlantısal Sayı Üreteçleri (SRSÜ) ve Hibrit Rastgele Sayı Üreteçleri (HRSÜ) olmak üzere üç ana başlık etrafında toplanmaktadır [1-4].

#### 1.1. Gerçek Rastgele Sayı Üreteçleri (GRSÜ)

GRSÜ'ler fiziksel bir ortamdan alınan gürültü işaretlerini temel alarak rastgele sayı üreten üreteçlerdir [5]. Rastgelelik kaynağı herhangi bir matematiksel modele veya algoritmaya bağlı olmadığı için üretilen sayı dizisinin de gerçek veya fiziksel olarak rastgele olduğu varsayılır. Temel bir GRSÜ'nün blok yapısı Şekil 1.1'de görülmektedir [6].



Şekil 1.1. Basit bir GRSÜ'nün blok yapısı.

Burada gürültü kaynağının elde edildiği gürültü ortamı termal gürültü, yarı-iletken elemanların gürültüsü, atmosferik basınç değişimleri, dijital sistemlerde clock

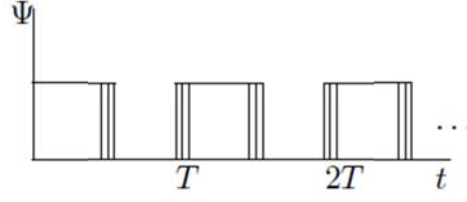
frekansındaki bozulmalar (jitter), radyoaktivite, neon lambası deşarjı veya herhangi bir beyaz gürültü kaynağı olabilir [5,7,8]. Bu gürültü ortamına aynı zamanda entropi kaynağı adı verilir. Gerçek rastgele sayı üretiminde bu gürültü ortamından alınan sinyaller kuvvetlendirilir ve örneklenmek suretiyle dijitize edilir. Dijitize edilen sinyallerin çoğu için gürültü kaynağının bant aralığı istenilen aralıkta olmaz ve belli değerlerde kutuplanır. Yani çıkışta uzun süreli 1 ve 0 değerleri görülür. Bu durumu gidermek ve gürültü kaynağının entropisini arttırmak için çıkışı belli bir aralığa yayacak olan ara bir işleme ünitesi (entropi düzenleyici) de isteğe bağlı olarak kullanılabilir. Von neumann ve XOR düzenleyicileri en bilinen ve kullanılan ara işleme ünitelerindendir. İşlemin en sonunda da düzenlenmiş rastgele bit dizileri bir çıkış arayüzü ile harici ortama aktarılır.

Literatürde farklı gürültü kaynaklarının kullanıldığı pek çok GRSÜ tasarımı sunulmuştur. Bu tasarımlar genel olarak dört teknikten faydalanmaktadırlar [6]:

- Gürültü kaynağının kuvvetlendirilerek doğrudan kullanılması
- Yarıkararlılık özelliğinin kullanılması
- Jitter olayının kullanılması
- Kaotik sistemlerin kullanılması

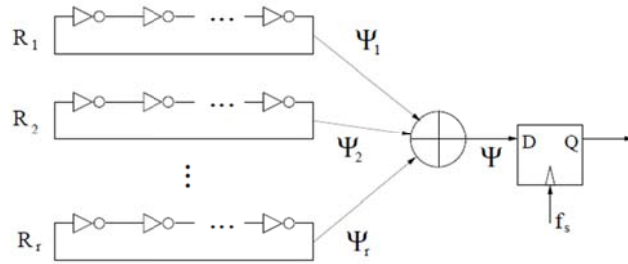
Gürültü kaynağının doğrudan kuvvetlendirildiği çalışmalara [9-11] örnek verilebilir. Bunlar beyaz gürültü kullanarak yüksek hızda bit dizileri üretebilse de harici etkilere açıktır ve iyi incelenmiş bir gürültü kaynağına ihtiyaç duymaktadırlar.

Yarıkararlılık flip-flop gibi yarıkararlı devre elemanlarının düzgün olmayan çalışma moduna alınmasıyla elde edilir. Bu moddayken eleman ne yüksek ne de düşük voltaj seviyesi gösterir ve kararsız osilasyonlar sergiler. Ardından da bu seviyelerden birine sabitlenir. Bu rastgelelik çeşitli çalışmalarda entropi kaynağı olarak sunulmuştur [12,13]. Bunların en önemli özelliği daha kompakt yapıda olmaları yüzünden harici etkilerden daha az etkilenmeleridir.



Şekil 1.2. Jitter olayı [15].

Jitter olayı ise Şekil 1.2’de görüldüğü üzere, dijital bir sistemdeki clock sinyalinin yükselme ve düşme zamanının hassas bir netlikte olmayıp idealden uzak rastgele bir şekilde değişmesidir. Bu durum da gene dijital sistemler üzerinde kullanılabilecek bir entropi kaynağı teşkil etmektedir. Fakat bu durum CMOS devrelerdeki termal etkiler yüzünden olduğundan entropisi düşüktür. Bu nedenle, çeşitli yöntemlerle jitter olayının daha da belirgin hale gelmesi yani mevcut entropinin artırılması sağlanır. Bu yöntemlerden en bilineni çift osilatör tabanlı yöntemidir. Bunlar bir hızlı bir de yavaş osilatörün bir D flip-flop ile örneklenmesi prensibine dayalı çalışır [14]. Buna örnek olarak, Sunar ve arkadaşları tarafından teorik altyapısı sunulan çalışma gösterilebilir [15]. Bu çalışmada halka osilatörler kullanarak jitter gürültüsünün entropisini arttırdıktan sonra bu osilatörlerin gürültü kaynağı olarak kullanıldığı bir GRSÜ tasarlanmıştır. Ardından bir başka çalışmada da bu yapı donanımsal olarak gerçekleştirilmiştir [16]. Bu tür tasarımların temel mantığı Şekil 1.3’de gösterilmektedir.



Şekil 1.3. Halka osilatörler kullanarak jitter tabanlı rastgele sayı üretimi [15].

Burada R’ler halka osilatörleri ve  $\Psi$  değerleri de çıkış sinyallerini temsil etmektedir. Çıkış sinyallerinin XOR’lanarak daha yavaş bir frekanstaki  $f_s$  sinyaliyle D flip-flop ile örneklenmesi flip-flop’un çıkışında gerçek bir rastgele dizinin oluşmasına neden olur.

Burada entropi kaynağı halka osilatörler olup bu yapı da jitter etkilerinden faydalanmak için yavaş ve hızlı osilatörlerin eşleştirilmesi mantığını kullanır.

### 1.1.1. Kaotik Sistem Tabanlı GRSÜ'ler

GRSÜ tasarımlarında kullanılan dördüncü teknik kaotik sistemlerin kullanılmasıdır. Nonlineer dinamiğin bir kolu olan kaos teorisi düşük boyutlu sistemlerin oldukça kompleks ve tahmin edilmesi zor olan davranışlar sergileyebileceğini göstermiştir. Bu da bu tür sistemlerin GRSÜ tasarımlarında gürültü kaynağı olarak kullanılabileceği fikrini akla getirmiştir. Bu bağlamda diferensiyel denklem tabanlı sürekli zamanlı kaotik sistemler ve ayrık zamanlı kaotik haritalar gürültü kaynağı olarak kullanılmaktadır. Bunların en önemli avantajı entropilerinin çeşitli yöntemlerle iyileştirilebilmesi ve ideal forma yakın hale getirilebilmesidir. Oysaki fiziksel ortamdan alınan gürültülerin entropisi dış etkenlerden oldukça etkilenmekte ve her zaman için kaliteli bir rastgelelik sağlayamamaktadır. Literatürde bu konuda yapılmış detaylı çalışmalar ve farklı kaotik sistemler kullanılarak elde edilmiş GRSÜ tasarımları bulunmaktadır [17-22].

### 1.2. Sözde Rastgele Sayı Üreteçleri (SRSÜ)

SRSÜ'ler deterministik yapıda olup çıkışları bir önceki değere bağlı olan üreteçlerdir. Yani tamamen matematiksel modellerle ifade edilebilir ve öngörülebilirler. Bunların çıkışları belirli bir süre boyunca rastgele gibi görünse de aslında bu rastgele dizi periyodik olarak kendisini tekrar etmektedir ve aynı başlangıç şartları (seed) için aynı bit dizisi üretilmektedir. Bu nedenle kullanımlarında bu hususlar dikkatlice göz önünde bulundurulmalıdır. Öte yandan, SRSÜ kullanımının literatürde tercih edilmesinin nedeni GRSÜ'lere göre donanımsal olarak çok daha kolay gerçekleştirilebilmeleri ve daha yüksek hızları sunabilmeleridir. Doğrusal eşleşik üreteçler ve doğrusal geribeslemeli kaydırmalı kaydediciler en çok bilinen ve kullanılan rastgele sayı üreteçlerinin başında gelmektedir [2,5,7].

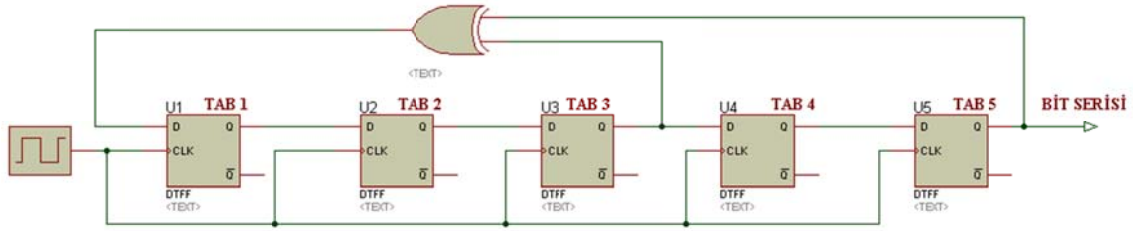
#### 1.2.1. Doğrusal Eşleşik Üreteçler

Bu tür üreteçler Denklem (1.1)'deki genelleştirilmiş denklem yardımıyla ifade edilmektedirler:

$$X_{i+1} = aX_i + b \pmod{m} \quad (1.1)$$

Burada ( $a, b$  ve  $m$ ) üreticinin kontrol parametreleri olup anahtar olarak kullanılmaktadır.  $X_0$  ise başlangıç değeridir. Eğer bu parametreler düzgün ayarlanırsa bu üreticinin çıkışı  $[0, m-1]$  aralığındaki tüm tamsayı değerleri görülmeden kendini tekrar etmeyecektir. Bu tür periyotlar  $m$ -serili ya da maksimum serili olarak adlandırılır. Sözde rastgele sayı üreticileri kullanılırken çıkış periyodunun bu şekilde maksimum olması istenir. Örneğin, yukarıdaki denklem için parametre değerleri  $a = 5$ ,  $b = 3$ ,  $m = 16$  ve  $X_0 = 1$  seçilirse  $\{1, 8, 11, 10, 5, 12, 15, 14, 9, 0, 3, 2, 13, 4, 7, 6, \dots, 1, 8, 11, 10, 5, \dots\}$  şeklinde bir dizi elde edilir [23]. Bu tür üreticiler daha çok yazılımsal olarak gerçekleştirilmeye uygundur.

### 1.2.2. Doğrusal Geribeslemeli Kaydırmalı Kaydediciler



Şekil 1.4. Tipik bir doğrusal geribeslemeli kaydırmalı kaydedici yapısı.

Bu tür üreticilerin temel yapısı Şekil 1.4'te görülmektedir. Şekilden de görülebileceği üzere bu tür yapılar sırasıyla dizilmiş kaydırmalı kaydediciler (register) ve bir geribesleme biriminden oluşmaktadır. Geribesleme birimi için farklı matematiksel fonksiyonlar kullanılabilmekte olup en sık kullanılan işlem XOR işlemidir. Bu tür fonksiyonların genel yapısı Denklem (1.2) ile ifade edilmektedir.

$$a(t) = c_1 a(t-1) \oplus c_2 a(t-2) \oplus \dots \oplus c_{n-1} a(t-n+1) \oplus a(t-n) \quad (1.2)$$

Bu üreticinin bit bazında işlem yaptığı için  $c_i \in \{0,1\}$  olmak zorundadır. Buna göre kaydırmalı kaydedicilerin mevcut durumu yukarıdaki geribesleme fonksiyonuna aktarılmakta ve buna göre bir fonksiyon çıkışı oluşturulmaktadır. Ardından kaydediciler bir bit kaydırılınca en soldaki bite bu fonksiyon değeri yazılmakta, en sağdaki bit ise

çıkışa verilmektedir. Bu yöntemle çıkışta oldukça iyi karakteristiklere sahip bir rastgele bit dizisi elde edilmektedir [7, 23].

Şekil 1.4'te görüldüğü üzere her bir kaydedici tab olarak adlandırılmaktadır. Toplam tab sayısı  $o$  kaydırmalı kaydedicinin mertebesini ( $L$ ) vermektedir. Farklı tablaların XOR işlemine tutularak geribesleme yapılması ile farklı periyotlar oluşur. Yalnızca  $m$ -serili kaydırmalı kaydediciler için periyot uzunluğu  $2^L - 1$  şeklinde maksimum olmaktadır. Periyot uzunluğunu  $m$ -serili yani maksimum yapan tab geribeslemesi değerleri ise önceden hesaplanıp tablolar halinde sunulmaktadır. Şekil 1.4'teki 5 bitlik lineer geribeslemeli kaydırmalı kaydedici için  $m$ -serisi veren tablolar sırasıyla (5,3,1)'dir. Dolayısıyla şekildeki kaydırmalı kaydedici  $m$ -serili olup mertebesi  $L = 5$  olduğundan periyodu  $2^5 - 1 = 31$  olacaktır. 0 hariç farklı başlangıç şartları için bu durum test edilebilir. Başlangıç şartı sıfır olursa çıkış daima sıfır olur. Bu tür üreteçler lineer eşleşik üreteçlerin aksine donanım gerçekleştirmelerine daha uygundurlar.

### 1.2.3. Kaotik Sistem Tabanlı SRSÜ'ler

Kaotik sistem tabanlı SRSÜ'ler sözde rastgele bit dizileri üretmek için kaotik sistemlerin kompleks dinamiklerinden faydalanan üreteçlerdir. Birinci dereceden nonlineer fark eşitliklerinden elde edilen kaotik SRSÜ tasarımında kullanılabileceği fikri ilk kez 1982 yılında Oishi ve Inoue tarafından ortaya konuldu [24]. Uzunca bir süre geçtikten sonra 1993 yılında Lin ve Chua kaos modunda çalışan ikinci dereceden dijital filtreler kullanarak bir SRSÜ tasarladılar [25]. 1996 yılında ise Andrecut lojistik harita kullanarak bir SRSÜ oluşturdu ve bunu mevcut eşleşik rastgele sayı üreteçleri ile karşılaştırdı [26]. 2001 yılında Kolesov ve arkadaşları ayrık zamanlı kaotik sinyallere dayanan bir SRSÜ sundular [27]. Daha sonra, Kocarev ve Stojanovski bir boyutlu parçalı lineer kaotik haritaların SRSÜ olarak kullanımlarını incelediler [18,28]. Li ve arkadaşları ise teorik bir analiz yaptılar ve parçalı lineer kaotik haritaların iyi bir korelasyona sahip olmaları, yüksek lineer komplekslikleri ve uzun çevrimleri gibi özellikleriyle mükemmel kriptografik özelliklere sahip olduğu sonucuna vardılar [29]. Fakat aynı zamanda tek bir kaotik sistem kullanımının kaotik sistem hakkında bilgi sızdırabileceği tehlikesine karşı potansiyel olarak güvenli olmadığı kanısına vardılar ve çözüm olarak da eşleştirilmiş kısmi parçalı kaotik harita sistemini önerdiler. Bu sistem birbirinden bağımsız çalışan iki kaotik harita çıkışının karşılaştırılmasıyla rastgele bir

bit dizisi üretmektedir. 2003 yılına gelindiğinde Kocarev ve Jakimoski kaotik haritaların SRSÜ yapısında nasıl farklı şekillerde kullanılabileceğini irdelediler ve yeni bir kaos tabanlı SRSÜ oluşturdular [30]. Daha sonra Huaping ve arkadaşları sözde rastlantısal sayı üretimi için eşleştirilmiş kaotik harita kafesinden yararlandılar [31]. 2005 yılında ise Liu, modifiye edilmiş lojistik harita tabanlı yeni bir SRSÜ tasarımını ortaya koydu [32]. 2006 yılında Wang ve arkadaşları z-lojistik harita tabanlı, sonlu hesaplama çözünürlüğü ile rastgele bit dizisi üreten SRSÜ tasarımını sundular [33]. 2007 yılında ise Ergün ve Özoğuz otonom olmayan kaotik bir sistemin Poincare haritasının istatistiksel testlerden başarıyla geçebileceğini göstermiştir [22]. Patidar ve arkadaşları ise 2009 yılında standart lojistik harita ile gerçekleştirdikleri SRSÜ'nün kapsamlı analizlerini ve istatistiksel testlerini yapmışlardır [34]. 2012 yılına baktığımızda ise bu konudaki çalışmaların görüntü şifrelemede kullanılmak üzere önerildiklerini görmekteyiz [35,36]. Bu konuda yapılan çalışmalar yeni tasarımlar ve donanım çözümleriyle devam etmekte ve canlılığını korumaktadır.

### 1.3. Hibrit Rastgele Sayı Üreteçleri (HRSÜ)

Hibrit rastgele sayı üreteçleri hem GRSÜ hem de SRSÜ yapılarını beraber kullanan üreteçlerdir. Temel bir HRSÜ yapısı Şekil 1.5'de görülmektedir. Buna göre GRSÜ yapısı SRSÜ tarafından üretilen rastgele diziler için başlangıç koşullarının (seed) seçiminde kullanılmaktadır. Bu sayede rastgele başlangıç koşullarından tetiklendiği için daha güvenilir ve SRSÜ'nün bütün avantajlarından yararlanabileceğimiz bir rastgele sayı üretici elde edilmiş olur [37].



Şekil 1.5. HRSÜ'nün temel yapısı.

### 1.4. İstatistiki Rastgelelik Testleri

Rastgele sayı üreteçlerinin güvenilirliği veya performansı istatistiki rastgelelik testleri yardımıyla değerlendirilmektedir. Literatürde en sık kullanılan rastgelelik testleri NIST SP 800-22, DIEHARD ve FIPS 140-2'dir [38-40]. Bu tür testler iki tane hipoteze

dayanmaktadır. Sıfır hipotezine göre ( $H_0$ ) incelenen bit dizisi rastgeledir. Diğer hipoteze göre ( $H_1$ ) ise bit dizisi rastgele değildir. Testler sonucun rastgele olduğunu söylemek yerine sıfır hipotezinin belli bir olasılıkla (p-değeri) inkar edilemeyeceğini ifade eder. Eğer p-değeri belli bir anlamlılık düzeyinin (significance level) altındaysa sıfır hipotezi tamamen geçersiz sayılır. Yani incelenen bit dizisi rastgele değildir. Bu bakımdan, iyi bir rastgele sayı üreticini yüksek p-değerleriyle tüm testlerden geçmesi beklenir.

NIST SP 800-22 istatistiki rastgelelik testleri literatürde en sık kullanılan test olup 15 farklı testten oluşmaktadır. Bu teste tabi tutulacak olan bit uzunluğu en az  $10^8$  olmalıdır. Genelde farklı başlangıç şartları için elde edilen bit dizilerinin birleşimi test için kullanılır. İçerdiği 15 farklı test şu şekildedir:

1. Frekans testi
2. Blok frekans testi
3. Kümülatif toplamalar testi
4. Yinelemeler testi
5. En uzun yinelemeler testi
6. Rank testi
7. Ayırık Fourier dönüşümü (FFT) testi
8. Örtüşmeyen şablon eşleştirme testi
9. Örtüşen şablon eşleştirme testi
10. Evrensel test
11. Yaklaşık entropi testi
12. Rastgele sapmalar testi
13. Değişken rastgele sapmalar testi
14. Seri test
15. Lineer komplekslik testi

Frekans testi, bit dizisindeki 0 ve 1'lerin oranına bakar. Bu oran hemen hemen aynı olmak zorundadır. Eğer bu test geçilmiyorsa diğer tüm testler uygulanmadan sıfır hipotezi reddedilir. Blok frekans testi ise M bitlik blok içerisindeki 1 değerlerinin oranının M/2 olup olmadığına bakar. M=1 için frekans testinin aynısıdır. Kümülatif toplamalar testinde 0 değerleri -1 ile değiştirilir ve bit dizisinin alt dizileri buna göre toplanarak toplamın rastgele bir sayı için olması gereken toplamdan ne kadar uzak veya

ne kadar yakın olduğunu belirler. Yinelemeler testi bit dizisindeki ardışık 1 ve 0 yinelemelerinin hangi uzunlukta olduğunu inceler. Böylelikle 0 ve 1 arasındaki geçiş sıklığının rastgele bir sayıda olması gerektiği gibi olup olmadığına bakar. En uzun yinelemeler testi ise aynı yinelemelerin maksimum uzunluğunu inceler. Rank testinde matrise dönüştürülen alt dizilerin ranklarının teorik olarak beklenen değerden sapması incelenir. FFT testinde ise Fourier dönüşümünde olmaması gereken periyodik tepe değerlerinin varlığı incelenir. Örtüşen ve örtüşmeyen şablon eşleştirme testlerinde  $m$  bitlik bir şablon seçilerek bit kaydırma metoduyla bu şablonun bulunduğu durumlara bakılır. Bu testler periyodik olmayan paternlerin varlığını inceler. Evrensel test ise bit başına entropiyi hesaba katarak bit dizisi içinde anahtar belirlenip belirlenemeyeceğine bakar. Kriptografik uygulamaların güvenliği açısından önemlidir. Yaklaşık entropi testi  $m$  ve  $m+1$  uzunluğunda iki ardışık bloğun karşılaşılma frekansının olması gerekene göre inceler. Rastgele ve değişken rastgele sapmalar testleri kümülatif toplam testindeki gibi 0'ların -1 yapılarak bit değerlerinin toplanmasına dayanır. Farklı paternler için elde edilecek değerlerin beklenen değerlerden sapmaları incelenir. Seri testte  $2^m$  adet  $m$  bitlik paternin tüm dizi içerisinde bulunma olasılıklarına bakılır. Elde edilen olasılıkların tekdüze dağılımlı bir çıkış için eşit olması beklenir. Lineer komplekslik testi ise üretilen bit dizisinin ne kadarlık bir lineer geribeslemeli kaydırmalı kaydedici ile üretilebileceğini inceler. Rastgele bir sayı için bu değer çok büyük olmalı. Kısa bir lineer kaydırmalı kaydediciyle ifade edilebilen bit dizileri rastgele sayılamaz [38,41].

## 2. BÖLÜM

### KAOTİK HARİTALAR

Dinamik sistemler ya diferansiyel denklemlerle ya da fark denklemleriyle ifade edilirler. Fark denklemleri ile ifade edilen sistemlere kısaca harita adı verilir ve bunlar ayrık zaman dilimlerinde faz uzayının farklı bölgelerini aşağıdaki genelleştirilmiş denkleme göre haritalarlar:

$$x_{n+1} = f(x_n) \quad (2.1)$$

Bu haritalama, sürekli yörüngeler oluşturmaktan ziyade faz uzayının bir yerinden bir başka yerine sıçrayan noktalar meydana getirir. Kaotik haritalarda ise bu noktalar faz uzayında asla kesişmez ve buna rağmen belirli bir alanda sınırlı kalırlar. Bu bölümde kaotik haritalar niteliksel ve niceliksel özellikleriyle incelenecek ve farklı kaotik harita modelleri tanıtılacaktır.

#### 2.1. Kaotik Haritaların Genel Özellikleri

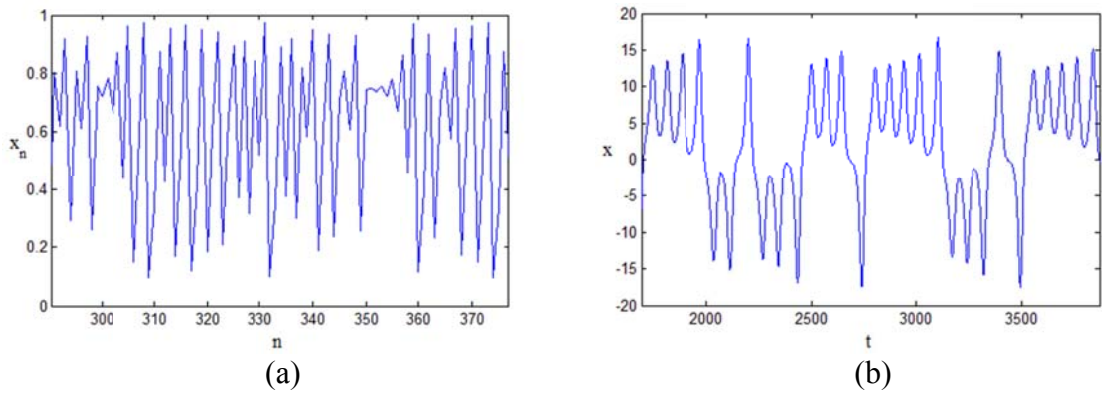
Kaotik haritalar diferansiyel denklem takımlarıyla tanımlanan sürekli zamanlı kaosu tüm özelliklerini taşırlar. Fakat sürekli zamanlı kaosu meydana gelebilmesi için Poincare-Bendixson teoremine göre en az üç boyut gerekirken, kaotik haritalar için böyle bir kısıtlama yoktur [42]. Kaotik haritalar bir boyutta bile kompleks kaotik davranış sergileyebilirler. Bu sayede, çok basit fark denklemleri kaosu gerçekleştirimi için cezbedici olanaklar sunmaktadır. Bu sürekli zamanlı kaos ile ayrık zamanlı kaosu en önemli farkıdır. Bir diğer önemli fark ise sonraki kısımlarda açıklanacak olan periyot katlama olayında evrensel bir sabitin bulunmasıdır. Bunun dışında, sürekli zamanlı

kaosun tüm özelliklerini bünyelerinde barındırırlar. Bu niceliksel ve niteliksel özellikler şu şekilde sıralanabilir:

- Sıçramalı yörüngeler
- Başlangıç şartlarına hassas bağıllık
- Pozitif Lyapunov Üsteli
- Sınırlı bir alanda tanımlılık
- Kendini tekrar etmeyen rastgele benzeri çıkış üretme
- Periyot kırılmalarının Feigenbaum sabiti ile öngörülebilmesi
- Fraktal boyuta sahip olma
- Ergodik olma ve topolojik harmanlama yapma

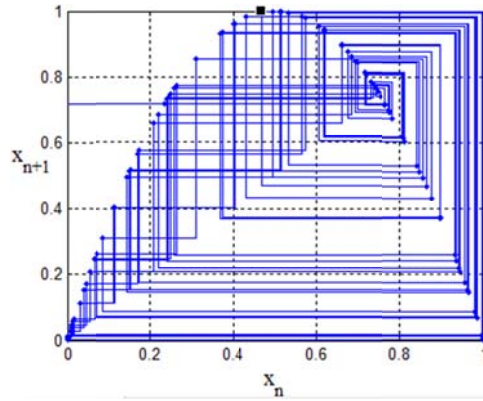
### 2.1.1. Kaotik Haritaların Yörüngeleri

Kaotik haritaların yörüngeleri iterasyon işlemi, yani fark denkleminin elde edilen her nokta için yeniden uygulanması gibi basit bir prensiple hesaplanır. İntegrasyon gibi karmaşık hesaplamalara gerek duymaz. Bu yönüyle bilgisayar gibi ayrık zamanlı sistemler üzerinde hesaplanmaları oldukça kolaydır. Bilgisayar simülasyonu ile hesaplanmış örnek bir kaotik harita yörüngesi Şekil 2.1a'da görülmektedir. Şekilden anlaşılacağı üzere yörünge her iterasyonda farklı bir noktaya sıçramaktadır. Bu durum, Şekil 2.1b'de görülen sürekli zamanlı bir kaotik sistemin yörüngesinden oldukça farklıdır.



Şekil 2.1. Kaotik yörüngeler, a) Ayrık zamanlı kaotik harita için, b) Sürekli zamanlı kaotik sistem için.

Bu iki yörünge arasındaki farklılığın temel nedeni denklemlerin çözülme yöntemidir. Sürekli zamanlı kaotik sistemlerde akış sürekli olduğu için çözümler de sürekli bir yapı göstermektedir. Kaotik haritaların iterasyonu ise bir noktadan diğerine sıçrama şeklinde geçişlere neden olmaktadır. Bu sıçramaların iterasyonla oluşum mekanizması Şekil 2.2’de görülen örümcek ağı (cobweb) çizimiyle daha iyi anlaşılmaktadır.



Şekil 2.2. Kaotik haritalarda yörünge oluşumunun örümcek ağı (cobweb) çizimi ile gösterilmesi.

Bu çizime göre, iterasyonun başladığı noktadan  $x=y$  eksenine bir doğru çizilir. Daha sonra, bu noktadan fark denkleminde göre iterasyon gerçekleştirilerek sonraki noktaya bir doğru çizilir. Sonra, elde edilen yeni noktadan  $x=y$  eksenine tekrar çizim yapılarak bu işlem tekrar ettirilir. Elde edilen şekil kaotik haritalardaki sıçrama olayının daha iyi kavranmasını sağlar.

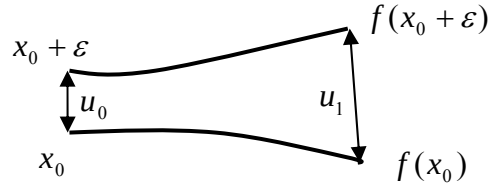
Örümcek ağı (cobweb) çizimi sırasında  $x=y$  eksenini dışında kalan kırılmaların oluşturduğu noktaların birleşimi o kaotik haritanın çekerini verir. Çeker, tıpkı sürekli zamanlı kaotik sistemlerde olduğu gibi ayrık zamanlı kaosta da yörüngelerin faz uzayında oluşturduğu sınırlı bölgenin şekline denir. Tüm yörüngeler bu bölge üzerine çekilir.

### 2.1.2. Kaotik Haritalarda Lyapunov Üstelleri

Lyapunov üstelleri kaos için niceliksel bir ölçüm olanağı sunar. Kısaca, birbirine oldukça yakın iki başlangıç noktasının iterasyonla birlikte birbirlerinden ne kadar uzaklaşacağını eksponansiyel olarak ölçüsüdür. Ayrık zamanlı veya sürekli zamanlı

fark etmeksizin tüm kaotik sistemlerde pozitif Lyapunov üstellerinin olması beklenir [43]. Pozitif Lyapunov üsteli, bu iki nokta arasındaki uzaklığın sürekli arması anlamına gelirken; negatif Lyapunov üsteli, bu iki noktanın birbirine yaklaştığını ifade eder. Dolayısıyla, pozitif Lyapunov üstelleri, tahmin edilebileceği üzere aynı zamanda başlangıç şartlarına hassas bağıllığın da olmazsa olmazıdır. Negatif ve sıfır Lyapunov üsteli olan sistemler ya sönümlenir ya da periyodik osilasyonlar gösterirler. Kaos gibi kompleks bir davranış sergileyemezler.

Lyapunov üstellerini daha iyi ifade edebilmek için birbirinden  $\varepsilon$  kadar uzaklıkta olan iki tane noktayı göz önüne alalım. Bu noktalar  $f(x_n)$  şeklinde tanımlı bir kaotik fonksiyon ile itere edilsin. Noktalar arasındaki  $u$  uzaklığının yeni durumu Şekil 2.3’de görüldüğü gibi olacaktır.



Şekil 2.3. Birbirine çok yakın iki noktanın kaotik fonksiyon ile değişimi.

İlk noktalar arasındaki uzaklık  $u_0$  ve sonraki uzaklık  $u_1$  olsun. Bu uzaklıklar arasında  $u_1 \approx e^\lambda u_0$  gibi bir eşitlik kurulabilir. Buradaki  $\lambda$  üsteli Lyapunov üsteli olarak adlandırılır. Bu üstelin matematiksel olarak nasıl hesaplandığını açıklamak için bu noktalar arasındaki uzaklığı  $u(x, y)$  cinsinden yazalım [45]:

$$u(f(x_0), f(x_0 + \varepsilon)) = e^\lambda u(x_0, x_0 + \varepsilon) \quad (2.2)$$

Bu denklemi  $n$  iterasyon için genelleştirirsek,

$$u(f^n(x_0), f^n(x_0 + \varepsilon)) = e^{\lambda n} u(x_0, x_0 + \varepsilon) \quad (2.3)$$

bağıntısını elde ederiz. Lyapunov üsteli,  $n \rightarrow \infty$  ve  $\varepsilon \rightarrow 0$  için tanımlı olup denklemde  $\lambda$  ' yı sola çekersek aşağıdaki bağıntıyı elde ederiz:

$$\lambda = \lim_{n \rightarrow \infty} \lim_{\varepsilon \rightarrow 0} \frac{1}{n} \ln \left| \frac{u(f^n(x_0), f^n(x_0 + \varepsilon))}{\varepsilon} \right| \quad (2.4)$$

Denklem (2.4)' teki logaritmik kısmın içi ve  $\varepsilon \rightarrow 0$  için alınan limit ifadesi aslında Denklem (2.5) ile verilen türev ifadesine eşdeğerdir:

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \ln \left| \frac{df^n(x_0)}{dx} \right| \quad (2.5)$$

Denklem (2.5)' e zincir kuralı uyguladığımız zaman da Denklem (2.6) ve Denklem (2.7)' yi elde ederiz.

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \ln \left| \frac{df(x_{n-1})}{dx} \cdot \frac{df(x_{n-2})}{dx} \dots \dots \frac{df(x_0)}{dx} \right| \quad (2.6)$$

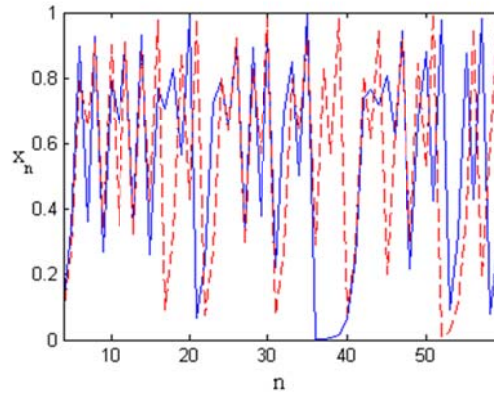
$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \ln \left| \prod_{n=0}^{n-1} \frac{df(x_n)}{dx} \right| \quad (2.7)$$

Sonuç olarak Denklem (2.7)'de aşağıdaki gibi yazılarak Lyapunov üstellerinin hesaplanması için genelleştirilmiş bir ifade elde edilir [43]:

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{n=0}^{n-1} \ln \left| \frac{df(x_n)}{dx} \right| \quad (2.8)$$

Denklem (2.8) ile bir boyutlu sistemlerin Lyapunov üstelleri doğrudan hesaplanabilmesine rağmen, 2 veya daha fazla boyutlu sistemlerde her boyut için vektörel olarak ayrı Lyapunov üstelleri olacağından ve her iterasyon için bu vektörlerin normali alınarak yeniden hesaplanması gerekeceğinden, çok boyutlu sistemlere yönelik farklı hesaplama metodları geliştirilmiştir [44].

Pozitif Lyapunov üstellerinin kaotik sistemlerdeki önemini vurgulamak için Şekil 2.4’ü inceleyelim. Bu şekilde sürekli çizgi ile gösterilen yörünge  $x_0 = 0.12$  başlangıç noktası için bir boyutlu bir kaotik haritanın çıkış serisini temsil etmektedir. Kesikli çizgi ile gösterilen yörünge ise aynı kaotik haritanın  $x_0 = 0.123$  başlangıç noktası için elde edilen çıkış serisidir. Şekil 2.4’ten görülebileceği üzere iki nokta arasında başlangıçta çok küçük bir fark olmasına rağmen, bu fark her iterasyonda Lyapunov üstelinin pozitif olması nedeniyle eksponansiyel olarak artmaktadır ve bir süre sonra bu iki yörünge birbirinden tamamen farklılaşmaktadır. Bu durum, kaotik sistemler için Lorenz’in ortaya sunduğu ünlü “kelebek etkisi” teoreminden başka bir şey değildir [46].



Şekil 2.4. Birbirine çok yakın iki kaotik yörüngeyi zamanla değişimi.

### 2.1.3. Kaotik Haritaların Tanım Aralığı

Önceki bölümde kaotik bir sistemin pozitif bir Lyapunov üsteline sahip olması gerektiğinden bahsedilmişti. Şimdiyse bunu göz önünde bulundurarak Denklem (2.9) ile verilen sistemi inceleyelim.

$$x_{n+1} = 2x_n \quad (2.9)$$

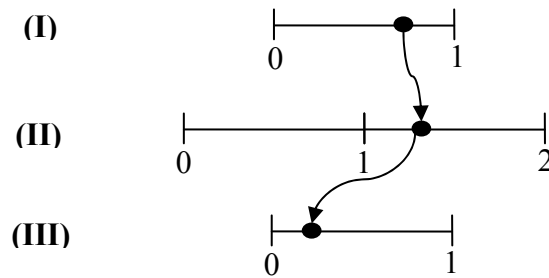
Bu sistemin Lyapunov üsteli, Denklem (2.8) kullanılarak  $\lambda = 2$  olarak hesaplanmaktadır. Lyapunov üsteli sıfırdan büyük olmasına rağmen bu sistem, kaotik bir davranış göstermek yerine pozitif başlangıç noktaları için artı sonsuza ve negatif başlangıç noktaları için de eksi sonsuza ıraksamaktadır. Dolayısıyla, buradan anlaşılabileceği üzere, bir sistemin pozitif Lyapunov üsteline sahip olması o sistemin kaotik olması için yeter şart değildir. Kaotik bir sistemin pozitif bir Lyapunov üstelinin yanı sıra sınırlı bir

alanda tanımlı olması gerekmektedir. Bu ise “esnetme ve katlama” adı verilen mekanizma sayesinde sağlanır. Buna göre kaotik yörüngeler pozitif Lyapunov üstelleri boyunca birbirlerinden uzaklaştırılarak “esnetilirken”, diğer taraftan faz uzayının farklı bölgelerine doğru “katlanarak” sınırlı bir alanda tutulurlar.

Bu bağlamda, Denklem (2.9) ile ifade edilen sistem incelendiğinde bu sistemin pozitif Lyapunov üsteli ile esnetme kriterini yerine getirdiğini fakat herhangi bir katlama mekanizmasının olmadığını görülmektedir. Bu sisteme katlama mekanizması sağlamak için her iterasyonda mod alma işlemi yapılabilir. Bu durumda, sistem Denklem (2.10)’daki gibi olacaktır. Bu haliyle sistem kaotik hale gelip (0,1) aralığında tanımlı kalmaktadır. Elde edilen yeni sisteme Bernoulli haritası adı verilmektedir [47].

$$x_{n+1} = 2x_n \mod(1) \quad (2.10)$$

Katlama işlemi ve sınırlı alanda tanımlılık Şekil 2.5’teki gösterim ile daha rahat anlaşılabilir. Şekil 2.5’teki (I) durumunda harita (0,1) aralığında bir noktada tanımlıdır. (II) durumunda ise bu başlangıç noktası 2 ile çarpılarak esnetilir ve tanım aralığı (0,2) olur. (III) durumunda ise 1 tabanında mod alınır ve tanım aralığı katlanarak tekrar (0,1) yapılır. Böylece bir iterasyon tamamlanmış olur ve başlangıç noktası da tanım aralığı içerisinde farklı bir noktaya haritalanmıştır. Kaotik haritanın tanım fonksiyonuna  $f(x_n)$  dersek, bu fonksiyonun tanım aralığı  $f : (0,1) \rightarrow (0,1)$  şeklinde ifade edilir.



Şekil 2.5. Bernoulli haritasında esnetme ve katlama işlemi.

#### 2.1.4. Rastgele Benzeri Çıkış Serisi

Daha önceki kısımlarda pozitif Lyapunov üstellerinin başlangıç şartlarına hassas bağımlılık sağladığını ve çok küçük pertürbasyonlar için bile yörüngeleri birbirinden

tamamen ayırdığını görmüştük. Aynı durumu kaotik sistemin uzun dönem davranışı için değerlendirirsek, sistem deterministik olmasına rağmen herhangi bir andaki değerlerden yola çıkarak sistemin davranışı hakkında uzun vadeli bir öngöründe bulunmanın imkansız olacağı sonucuna varırız [47]. Bu bağlamda kaotik bir haritanın çıkış serisinin  $x_0, x_1, x_2 \dots$  şeklinde yani  $\{x_i\}_{i=0}^N$  olduğunu varsayalım. Bu durumda,  $N \rightarrow \infty$  için bu seri uzun vadede tahmin edilemeyen gürültü benzeri bir işaret olacaktır. Başka başlangıç şartları için elde edilen diğer çıkış serileri de aynı nedenden dolayı birbirlerinden tamamen farklı olacaktır. Bunun dışında kaotik davranış periyodik olmadığı için çıkış serileri yani yörüngeler asla kendini tekrar edip periyodik olmayacaktır.

Kaotik sistemlerin bu özelliği nedeniyle çıkışları rastgele benzeri olarak adlandırılır ve bu özellikleri rastgele sayı üretmek için kullanılmalarının başlıca nedenidir. Çıkışlarına tamamen rastgele demek yerine rastgele benzeri denilmesinin temel nedeni ise rastgeleliğe neden olan temel faktörün aslında deterministik bir yapıda olmasıdır. Yani, aynı başlangıç şartlarına sahip iki kaotik sistem aynı denklemlerle ifade ediliyor ve her iki denklemin de tüm parametreleri biliniyorsa, her iki kaotik sistem çıkışı da rastgele benzeri olmasına rağmen birebir aynı olacaktır. Bu rastgele, yani stokastik süreçler için geçerli bir durum değildir. Örneğin, madeni parayı havaya atarak yazı veya tura gelmesine bakmak tamamen stokastik bir süreçtir [43].

Kaotik sistemlerin çıkış serileri kaotik olduğu müddetçe çıkışı rastgele benzeri olsa da bir kaotik sistemin tüm çıkışları rastgelelik özelliği gösterir denilemez. Çünkü bazı başlangıç şartları kaotik çıkış göstermek yerine sabit bir noktaya doğru yakınsayabilir ya da doğrudan ilk değeri değişmeksizin kalabilir. Bu durum dinamik sistemlerde var olan sabit noktaların gene dinamik sistem olan kaotik haritalarda da mevcut olması yüzündendir.  $f$  fonksiyonu ile tanımlı bir dinamik sistemdeki sabit noktalar  $f(x^*) = x^*$  şartını sağlayan tüm noktalardır [43]. Eğer kaotik sistemin yörüngelerinden biri bu noktalardan birinden başlar ya da bu noktalara düşerse bir daha bu noktadan çıkamaz; sabit değerde kalırlar. Denklem (2.10) ile verilen Bernoulli haritasının tanım aralığının  $[0,1]$  yerine  $(0,1)$  olmasının temel nedeni budur.  $x_0 = 0$  için çıkış serisi  $\{x_i\}_{i=0}^N = 0,0,0\dots$  şeklinde olacaktır. Dolayısıyla 0 noktası Bernoulli haritası için sabit noktadır. Aynı şekilde  $x_0 = 1$  olsaydı çıkış serisi  $\{x_i\}_{i=0}^N = 1,0,0\dots$  olacaktı. Tanım aralığı bu noktaları içermemesine rağmen tanım aralığı içerisindeki yörüngeler bu noktalara yakınsayabilir.

Mesela  $x_0 = 0.5$  için çıkış serisi  $\{x_i\}_{i=0}^N = 0.5, 1, 0, 0, 0, \dots$  şeklinde ikinci iterasyonda sabit noktaya girilirken,  $x_0 = 0.25$  için aynı şekilde üçüncü iterasyonda sabit noktaya girilir. Sonuç olarak, bir kaotik sistemin yörüngeleri rastgele özellik gösterse de bu durum mevcut tüm başlangıç şartları için geçerli olmak zorunda değildir diyebiliriz.

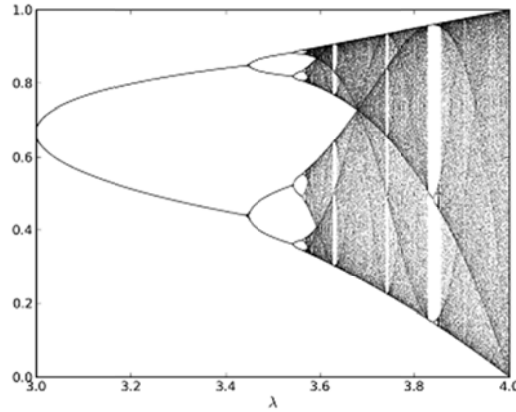
### 2.1.5. Feigenbaum Sabiti

Feigenbaum sabiti tanıtılmadan önce dinamik sistemlerdeki bifürkasyon teorisinden kısaca söz etmek gerekir. Bifürkasyon teorisi, kısaca dinamik bir sistemin kontrol parametreleri ile davranış değişimini inceleyen bir disiplindir [42]. Sistemin topolojik yapısının ya da niceliksel özelliklerinin değişmesi olayına bifürkasyon adı verilir. Bifürkasyon teorisinde yerel ve global olmak üzere iki tür genel bifürkasyon tipi bulunmaktadır. Yerel bifürkasyonlar dinamik sistemlerin sabit noktalarının oluşup yok olması veya bunların kararlılığının değişmesini incelerken global bifürkasyonlar dinamik sistemin tanım aralığının tümüne etki eden değişimleri inceler. Yerel bifürkasyonlar aşağıdaki gibi sıralanabilir [42]:

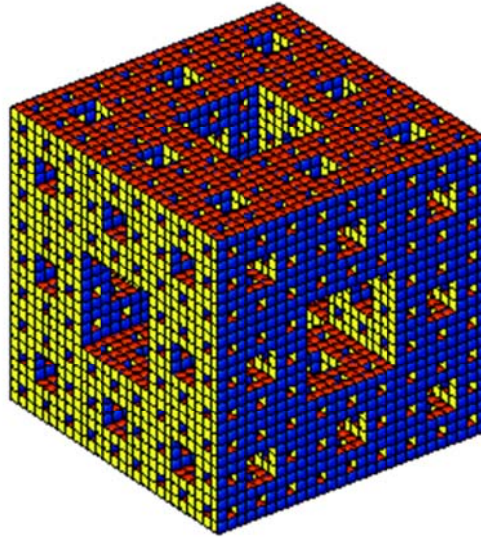
- Eyer-düğüm Bifürkasyonu
- Pitchfork Bifürkasyonu
- Transkritik Bifürkasyon
- Hopf Bifürkasyonu
- Periyot Katlama Bifürkasyonu

Bu bifürkasyon türlerinden periyot katlama bifürkasyonu kaotik haritalarda sıklıkla görülür. Kısaca, dinamik sistemin kontrol parametresi ile beraber çıkış serisinin periyodunun da 2,4,8... şeklinde 2'nin katı olacak biçimde sürekli artması durumudur. Bu artış her periyodik noktanın da kendi içerisinde 2 farklı periyoda ayrılarak çatallaşmasıyla gerçekleşir [43]. Periyotlar belli bir değerden sonra sonsuza doğru ıraksar ve kaos gözlemlenir. Bu durum Şekil 2.6' da verilen örnek bifürkasyon diyagramında görülebilir. 1978 yılında Mitchell J. Feigenbaum farklı kaotik haritalardaki periyot katlama bifürkasyonlarını incelerken tüm 1 boyutlu kaotik haritalarda periyot katlamanın olduğu parametre değerleri arasındaki uzaklık açısından sabit bir oranın olduğunu fark etti [48]. Bu oran Denklem (2.11) ile hesaplanmakta ve 4.669201609... gibi sabit bir değer almaktadır [43]:

$$\delta = \frac{A_n - A_{n-1}}{A_{n+1} - A_n} = 4.669201609... \quad (2.11)$$



cisimlerin aksine farklı oran ve büyüklüklerle kendini sürekli tekrar eden cisimlerdir. Bu özellikleri “kendine benzerlik” olarak adlandırılır. Ayrıca faz uzayındaki boyutları da tamsayı değil kesirlidir. Bu kesirli boyuta da fraktal boyut adı verilir. Fraktal cisimleri açıklayıcı en iyi örneklerden biri Menger süngeridir. Menger süngerini bir küp alınarak bu küpün her yüzünün 9 küpe bölünmesi ve ortada kalan küpün çıkarılması işleminin geri kalan tüm küpler için sonsuza kadar tekrarlanması işlemiyle elde edilir [50]. Menger süngerini Şekil 2.7’ de görülmektedir. Bu küpün yüzey alanı sonsuza giderken hacmi sıfıra yaklaşmaktadır. Küpün herhangi bir kısmı büyütülerek incelenecek olursa küpün tümüne benzediği görülecektir.



Şekil 2.7. Menger Süngerini.

Kaotik sistemlerin faz uzayında sınırlı alan kaplamasına rağmen yörüngelerinin aperiodyk ve birbirinden tamamen farklı olmasını sağlayan kaotik çekirlerinin fraktal yapı göstermesidir. Kaotik sistemlerde sürekli tekrarlanan katlama işlemi kendine benzerliğe ve dolayısıyla faz uzayında fraktal bir çekere neden olur [46].

Kaotik bir sistemin çekirinin fraktal boyutunu hesaplamak için farklı matematiksel modeller önerilmiştir. Bunlar sırasıyla kutu sayma boyutu, korelasyon boyutu, Kaplan-Yorke boyutudur [43,47]. Bunlar genel olarak farklı yöntemlere dayansa da birbirlerine yakın sonuçlar verirler ve beklenen sonuçlar arasında kullanılan metoda bağlı olarak büyüklük küçüklük ilişkileri bulunur [51]. Bunlar arasında en sık kullanılanlardan biri

Lyapunov üstellerinden kolayca hesaplamayı sağladığı için aynı zamanda Lyapunov boyutu olarak da adlandırılan Kaplan-Yorke boyutudur [52]. Kaplan-Yorke boyutu  $D_{KY}$  şeklinde ifade edilir ve Denklem (2.13) ile tanımlanır.

$$D_{KY} = k + \frac{\lambda_1 + \lambda_2 + \dots + \lambda_k}{|\lambda_{k+1}|} \quad (2.13)$$

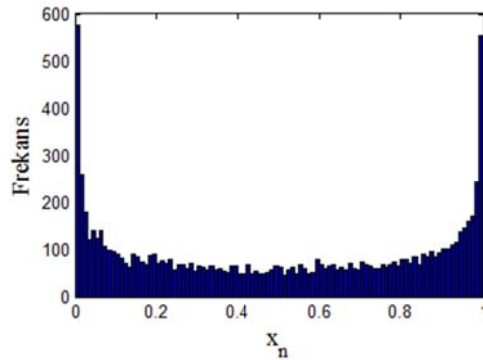
Burada  $\lambda$  ile ifade edilen değerler  $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_n$  şartını sağlayacak şekilde sıralanmış Lyapunov üstelleridir.  $k$  değeri ise  $\lambda_1 + \lambda_2 + \dots + \lambda_k > 0$  şartını sağlayan en büyük sayı değeridir.

### 2.1.7. Ergodiklik ve Topolojik Harmanlama

Ergodik teori, yörüngelerinin zamana göre ortalaması uzaya göre ortalamasına eşit olan dinamik sistemleri inceleyen bir matematik branşıdır [53]. Bu koşulu sağlayan dinamik sistemler ergodik olarak adlandırılır. Ergodik bir sistemin zaman ortalaması başlangıç koşullarından bağımsızdır [54]. Yani sistemin istatistiki özellikleri farklı başlangıç koşulları için değişmeyecektir.

Literatürde incelenen çoğu kaotik haritanın ergodik özellik gösterdiği ispatlanmış durumdadır [55]. Kaotik haritalardaki ergodik özellik, kaotik haritaların başlangıç koşulları fark etmeksizin aynı istatistiki ve olasılıksal özellikleri göstereceği anlamına gelir. Mesela, ergodik bir kaotik haritanın bir başlangıç şartı için hesaplanmış olan dağılım fonksiyonu diğer tüm başlangıç şartları için de birebir aynı olmalıdır. Bu çıkışın tekdüze olmasının istenildiği rastgele sayı üretimi için olmazsa olmaz bir gerekliliktir. Çıkış serisinde tekdüze dağılım gösteren bir kaotik haritanın histogramı Şekil 2.8’de görülmektedir. Bu istenilen dağılım tüm başlangıç koşulları için aynı kalacaktır.

Ergodik yörüngeler aynı zamanda topolojik harmanlamaya neden olurlar. Topolojik harmanlama, çeker üzerinde herhangi bir yerden başlayan bir yörüngenin çeker üzerindeki diğer tüm noktalara gelişigüzel bir şekilde yaklaşacağı anlamına gelir. Harmanlama çekerin iki ya da daha fazla parçaya ayrılmasını önler [56]. Ayrıca girişteki küçük bir sapmanın tüm çekere etkiyeceğini garanti altına alır. Ergodiklik ve topolojik harmanlama sonraki bölümlerde görüleceği üzere kriptografik olarak güvenli bir rastgele sayı üretici için gereklidir.



Şekil 2.8. Tekdüze dağılımlı kaotik bir haritanın histogramı.

## 2.2. Mevcut Kaotik Haritalar

Literatürde en sık kullanılan ve incelenen haritaların başında bir boyutlu lojistik harita gelmektedir [46]. Sıklıkla kullanılan bir diğer bir boyutlu harita ise parçalı doğrusal nonlinearliğe sahip olan tent (çadır) haritadır [47]. Bir başka parçalı doğrusal harita ise literatürde parçalı doğrusal kaotik harita (PDKH) olarak incelenmekte olup rastgele sayı üretiminde sıklıkla tercih edilmektedir [57]. İki boyutlu kaotik harita olarak ise Henon haritası detaylı bir biçimde çalışılmaktadır [46].

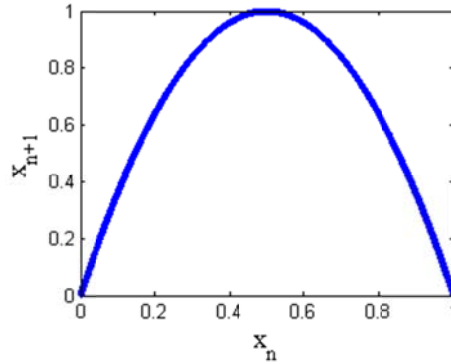
### 2.2.1. Lojistik Harita

Lojistik harita Denklem (2.14) ile ifade edilen, (0,1) aralığında tanımlı, bir boyutlu kaotik bir haritadır.

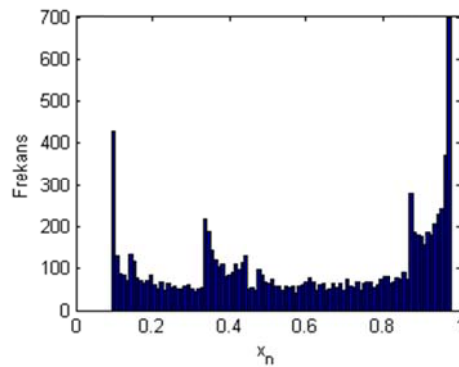
$$x_{n+1} = kx_n(1 - x_n) \quad (2.14)$$

Burada  $k$  parametresi bifürkasyon parametresi olup  $3.57 \leq k \leq 4$  için sistem kaotik davranış gösterir. Şekil 2.6'da görülen periyot katlamalı bifürkasyon diyagramı aslında  $k$  parametresine bağlı olarak lojistik haritaya aittir. Lojistik haritanın çekeri ise Şekil 2.9'daki gibidir. Lyapunov üsteli 10000 iterasyon için  $\lambda = 0.693125...$  şeklinde hesaplanmış olup  $k = 4$  için çıkışı Şekil 2.8'deki gibi tekdüze dağılım göstermektedir. Fakat  $k$  parametresinin farklı değerleri için dağılım tekdüzelikten çıkıp bozulmaktadır. Örnek olarak,  $k = 3.9$  için elde edilen dağılım Şekil 2.10'daki gibidir. Bu rastgele sayı

üretiminde istenmeyen bir durumdur. Bu nedenle, bu harita rastgele sayı üretiminde kullanılacaksa  $k$  parametresi 4 veya 4' e çok yakın seçilmelidir [34,58].



Şekil 2.9. Lojistik haritanın çeker.



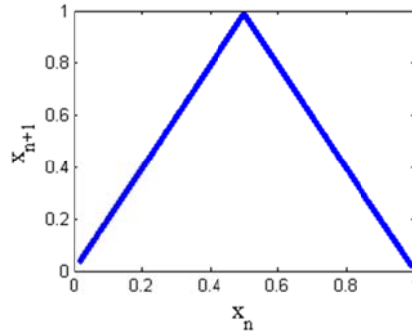
Şekil 2.10.  $k = 3.9$  için lojistik haritanın çıkış serisi dağılımı.

### 2.2.2. Tent (Çadır) Harita

Tent harita, adını Şekil 2.11' de görülen ve çadıra benzeyen üçgen şeklindeki çekerinden almaktadır. Tent harita parçalı doğrusal nonlineerliğe sahip kaotik sistemler sınıfına dahildir. Fark denklemleri şu şekildedir:

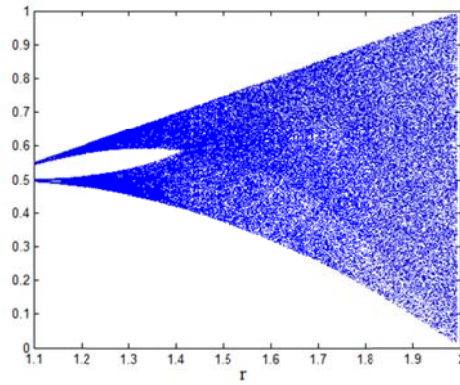
$$x_{n+1} = \begin{cases} rx_n, & 0 \leq x_n \leq 1/r \\ r(1 - x_n), & x_n \geq 1/r \end{cases} \quad (2.15)$$

Tent harita,  $(0,1)$  aralığında tanımlı olup  $1 > r > 2$  için kaotik davranış sergiler. Şekil 2.12'den görülebileceği üzere periyot katlama bifürkasyonu göstermez. Burada, diğer



Şekil 2.11. Tent haritanın çakeri.

parçalı doğrusal sistemlerde de gözlemlenen sınır çarpışması (border collision) bifürkasyonu görülür [59,61].

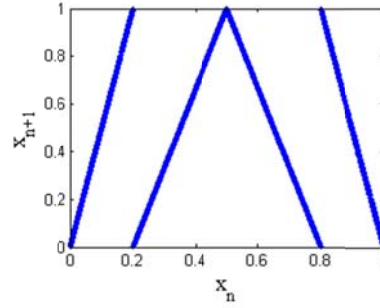


Şekil 2.12. Tent haritanın bifürkasyon diyagramı.

### 2.2.3. Parçalı Doğrusal Kaotik Harita (PDKH)

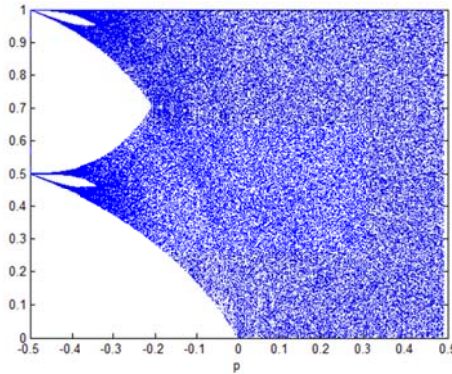
Bu kısımda verilecek olan parçalı doğrusal harita [60]'da önerilmiş olup [57]'de rastgele sayı üretiminde kullanılmak için tüm istatistiki özellikleri detaylıca incelenmiştir. Bu nedenle rastgele sayı üretiminde sıklıkla tercih edilmektedir [62]. Fark denklemi, Denklem (2.16)' da verilmiştir:

$$F(x_n, p) = \begin{cases} x_n / p, & x_n \in [0, p) \\ (x_n - p) / (0.5 - p), & x_n \in [p, 1/2] \\ F(1 - x_n, p), & x_n \in [1/2, 1] \end{cases} \quad (2.16)$$



Şekil 2.13. PDKH'nın  $p = 0.2$  için kaotik çekeri.

PDKH,  $0 < p < 1/2$  için kaotik davranış sergilemektedir.  $p = 0.2$  için çizdirilmiş kaotik çekeri Şekil 2.13'te gösterilmektedir. PDKH da tıpkı tent harita gibi periyot katlama bifürkasyonu göstermez. Bifürkasyon diyagramı Şekil 2.14'de görüldüğü gibidir. Bu haritanın bifürkasyonu  $p \in (0, 0.5)$  için oldukça düzgün bir yayılım göstermekte ve tekdüze dağılımlı çıkış serilerine neden olmaktadır. Bu şekilde, geniş bir parametre aralığında kaotik davranışın gözlemlenebildiği kaosa dayanıklı (robust) kaos adı verilir. Dayanıklı kaos, kaos tabanlı rastgele sayıcı üreteçlerine dayalı kriptografide, kaotik sistemin geniş parametre hassasiyetinin anahtar uzunluğunu arttırması nedeniyle istenilen bir özelliktir [63,64].



Şekil 2.14. Parçalı doğrusal kaotik haritanın (PDKH) bifürkasyon diyagramı.

Lojistik haritadaki kaos ile dayanıklı kaosu karşılaştıracak olursak, periyot katlama bifürkasyonuna dayalı lojistik haritada kaotik bölgeler arasında periyodik pencereler bulunduğu için lojistik haritanın bifürkasyon parametresinin dayanıklı kaos sağlayamayacağını ve dolayısıyla kaos tabanlı kriptografik uygulamalarda anahtar olarak kullanılamayacağını görürüz. Aksine daha önce belirttiğimiz üzere lojistik

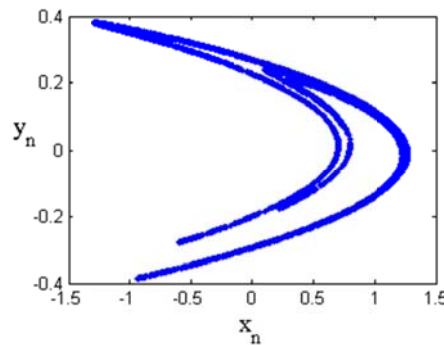
haritanın bifürkasyon parametresi sadece 4 ve 4'ün çok yakınındayken istenilen çıkış şartlarını sağlar. Bu ayrıcalıklı özellikleri nedeniyle PDKH kaos tabanlı rastgele sayı üretiminde ve kriptografide oldukça önemli bir haritadır.

#### 2.2.4. Henon Haritası

Şimdiye kadar incelenen kaotik haritaların tümü bir boyutluydu. Kaotik haritaların bir boyutlu olması gerçekleştirim kolaylığı ve düşük kaynak kullanımı nedeniyle doğal olarak daha avantajlıdır. Fakat iki ve daha fazla boyutlu haritalar da literatürde mevcuttur. İki boyutlu Henon haritası bu tür yüksek boyutlu haritalar için en çok incelenen haritalardan biridir [46,47]. Henon haritası Denklem (2.17) ile ifade edilmektedir:

$$\begin{aligned} x_{n+1} &= 1 + y_n - ax_n^2 \\ y_{n+1} &= bx_n \end{aligned} \quad (2.17)$$

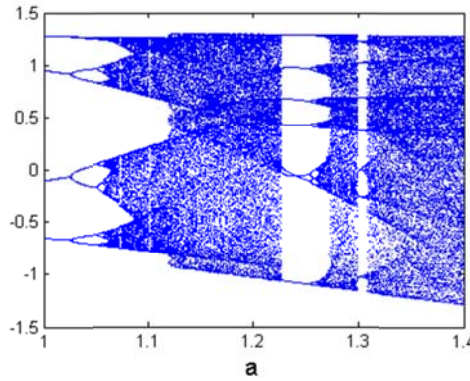
Kaos parametreleri tipik olarak  $a = 1.4$  ve  $b = 0.3$  olarak seçilir. Bu değerler için elde edilmiş olan kaotik çeki Şekil 2.15' de verilmiştir. İki boyutlu bir kaotik sistem olduğu için iki farklı Lyapunov üsteli bulunmaktadır. Bunlar sırasıyla  $\lambda_1 = 0.419$  ve  $\lambda_2 = -1.623$  olarak hesaplanmıştır. Kaplan-Yorke boyutu ise Denklem (2.13) kullanılarak  $D_{KY} = 1.2583$  olarak bulunmuştur. Buradan görüldüğü üzere çekiğin fraktal boyutu beklenildiği gibi topolojik boyuttan daha düşük ve kesirlidir.



Şekil 2.15. Henon haritasının kaotik çeki.

Henon haritasının  $b = 0.3$  değerinde sabitken  $a$  parametresi değiştirilerek elde edilmiş olan bifürkasyon diyagramı Şekil 2.16'daki gibi elde edilmiştir. Buradan görüleceği

üzere Henon haritası da tıpkı lojistik harita gibi periyot katlama bifürkasyonu göstermektedir. Bu nedenle dayanıklı kaos özelliği sergileyememektedir. Ayrıca bifürkasyon diyagramının yayılımı incelendiğinde bu aralıktaki hiçbir parametre değerinde tekdüze çıkış serisi sağlanamayacağı dolayısıyla bu haritanın bu haliyle rastgele sayı üretimine elverişli olmadığı görülmektedir.



Şekil 2.16. Henon haritasının  $a$  parametresine göre bifürkasyon diyagramı.

### 2.3. Kaotik Haritaların Rastgele Sayı Üretici Olarak Kullanılması

Daha önce bahsedilen kaotik haritaların rastgele benzeri çıkışı, farklı platformlarda aynı başlangıç koşulları için aynı rastgele çıkış dizisinin elde edilmesinin istendiği sözde rastgele sayı üretimi için idealdir. Fakat hangi tür kaotik haritanın hangi amaçla rastgele sayı üretici olarak kullanılacağı oldukça önem arz etmektedir. Mesela, kaotik sistem tabanlı rastgele sayı üretici Monte Carlo simülasyonları veya bilgisayar oyunları gibi güvenliğin önemli olmadığı alanlarda kullanılacaksa hangi tür kaotik sistemin kullanılacağı çok da önemli değildir. Fakat kaotik sistem tabanlı rastgele sayı üretici, üst düzey güvenliğin talep edildiği kriptografik uygulamalarda kullanılacaksa kullanılan kaotik sistemin tüm özellikleri dikkatle incelenmelidir. Bu çalışmada daha çok kriptolojik uygulamalarda kullanılmaya elverişli kaotik sistem tabanlı rastgele sayı üreticileri üzerinde durulacaktır.

#### 2.3.1. Kaos ve Kriptoloji İlişkisi

Claude Shannon'un, modern kriptografinin teorik temellerinden biri olan [65]'deki çalışmasında ortaya koymuş olduğu üzere kriptografik bir sistemin istatistikî analizlere karşı dirençli olabilmesi için yayılma (diffusion) ve karıştırma (confusion) özelliklerini

bünyesinde bulundurması gerekmektedir. Yayılma, şifrelenmemiş metin üzerinde gerçekleşen bir değişimin şifrelenmiş metin üzerinde birden fazla karakter için değişiklik göstermesi veya tam tersi şifrelenmiş metin üzerinde yapılacak bir değişikliğin şifrelenmemiş metin üzerindeki birden fazla karakterde değişime neden olması olayıdır. Bu sayede, istatistiksel ataklar için çok daha fazla şifrelenmiş metnin elde bulunması zorunlu hale gelecektir. Karıştırma ise kısaca anahtarın şifrelenmiş metin ile basit bir ilişkisinin olmaması durumudur. Yani şifrelenmiş metin üzerinden anahtar kolayca bulunamamalıdır [66,67].

Bu iki özellik ile kaos arasındaki ilişki ilk kez [68]'de detaylıca irdelenmiştir. Buna göre kaotik sistemlerdeki ergodiklik, yani kaotik sistemlerin çıkış serisinin istatistiki özelliklerinin başlangıç şartlarından bağımsız oluşu, karıştırma özelliğini karşılamaktadır. Bu sayede, kaotik sistem çıkışından yola çıkılarak anahtara yani başlangıç şartlarına ulaşamayacaktır. Benzer şekilde, topolojik harmanlama ve başlangıç şartlarına hassas bağımlılık yayılım özelliğine karşılık gelmektedir. Başlangıç şartları anahtar olarak kullanıldığında anahtarda yapılacak küçük bir değişiklik başlangıç şartlarına hassas bağımlılık nedeniyle şifrelenmiş metinde köklü değişimlere yol açacaktır. Topolojik harmanlama ise bu değişimin tüm şifrelenmiş metne etkiyeceğini garanti altına almaktadır. Buradan görüleceği üzere kaos ile kriptografik arasında çok yakın bir ilişki vardır.

### **2.3.2. Kaotik Sistem Tabanlı Kriptografik Şifreleyiciler**

Kriptografik şifreleme genel olarak simetrik anahtarlı şifreleme (symmetric-key cryptography) ve açık anahtarlı şifreleme (public-key cryptography) olmak üzere ikiye ayrılır. Simetrik anahtarlama şifrelemede alıcı ve gönderici için şifreleme ve deşifreleme aynı anahtarla sağlanmakta ve bu anahtar alıcı ve gönderici arasında paylaşılmaktadır. AES (Advanced Encryption Standart) ve DES (Data Encryption Standart) gibi şifreleme sistemleri bu metoda dayalıdır. Fakat bu tür şifrelemelerde paylaşılan anahtar üçüncü kişilerin eline geçerse tüm şifrelenmiş metin çözülebilir. Bu nedenle anahtarın taraflar arsında paylaşılması özellikle tarafların birbirinden uzakta olduğu durumlarda güvenlik açığına neden olmaktadır. Bu dezavantajı gidermek için 1970'lerde açık anahtarlı şifreleme metodu önerilmiştir. Bu yöntemde ise açık ve gizli olmak üzere iki tür anahtar kullanılır. Gizli anahtarın sahibi bir açık anahtar oluşturur.

Bu açık anahtar herkesin kullanımına açıktır. Daha sonra, karşı taraf bu açık anahtarı kullanarak şifrelenmiş metin oluşturur. Bu şifrelenmiş metin sadece gizli anahtarın sahibi tarafından çözülebilir. Bu sayede önceki yöntemin dezavantajı olan anahtar paylaşımı önlenmiş olur. Fakat bu yöntemin de dezavantajları vardır. Öncelikle, şifreleme ve deşifreleme için iki farklı şifre kullanıldığı için işlem ilk yönteme kıyasla daha yavaştır. İkinci olarak da kötü niyetli kişiler açık şifreyi kendilerinininki ile değiştirebilirler. RSA şifreleme sistemi bu yöntemi kullanan şifreleyicilere örnek verilebilir [66].

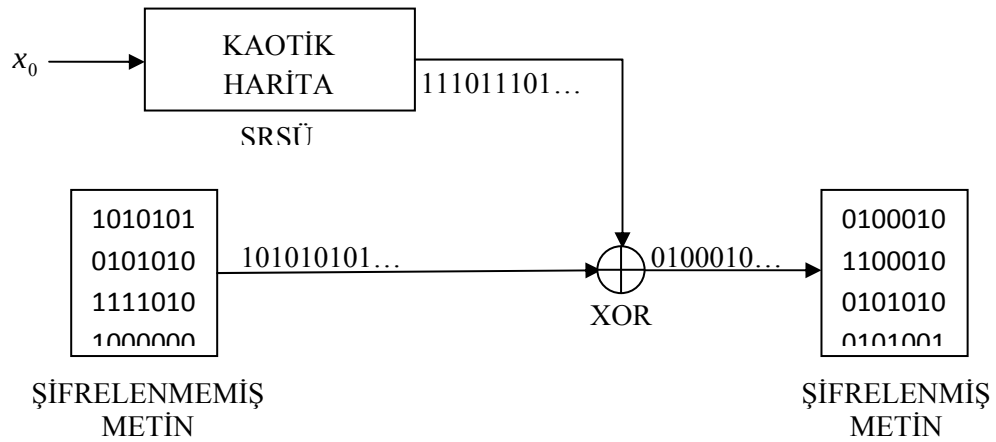
Simetrik anahtarlı şifreleme yönteminde akış şifreleyici (stream cipher) ve blok şifreleyici (block cipher) olmak üzere iki şifreleyici türü kullanılır. Akış şifreleyiciler bilgiyi seri olarak bit veya karakter parçalarıyla şifrelerken, blok şifreleyiciler büyük bit bloklarını algoritmaya sokarak şifreleme işlemi yaparlar [67].

Kaos tabanlı kriptografi ise genel olarak analog ve dijital olmak üzere ikiye ayrılmıştır. Analog tabanlı kaotik şifreleyici sistemler diferansiyel denklemlerle tanımlanan iki kaotik sistemin senkronizasyonunun sağlanabileceği gösterildikten sonra güvenli analog haberleşmede kullanılmak üzere önerilmişlerdir [69]. Fakat bu tür sistemlerdeki senkronizasyon yüzünden haberleşme esnasında veri sızdırılabileceği çeşitli çalışmalarla gösterilmiş ve önerildikleri kadar güvenli olmadıkları ortaya çıkmıştır [70-74]. Bu konu hakkındaki çalışmalar devam etmektedir.

Dijital tabanlı kaotik şifreleyici sistemler ise ayrık zamanlı kaotik haritaların senkronizasyona gerek olmaksızın sözde rastgele sayı üretmesine dayanan kriptografik sistemlerdir [75,76]. Bunlardan çoğu simetrik anahtarlı şifrelemede akış veya blok şifreleyici olarak kullanılmak üzere önerilmişlerdir. Açık anahtarlı şifrelemeli yöntemi kullanan nispeten daha az kaos tabanlı şifreleyici bulunmaktadır [77].

Kaotik bir haritanın rastgele benzeri çıkışı doğrudan sözde rastgele sayı üretimi için kullanılabilir. Bir akış şifreleyici ise aslında bir sözde rastgele sayı üreticinin (SRSÜ) çıkışının seri bit dizileri ile XOR gibi bir işlemle şifrelenmesi işlemi olduğu için bu tür haritalar aynı zamanda birer akış şifreleyicidir. Kaotik sistem tabanlı bir akış şifreleyicinin blok diyagramı Şekil 2.17’de verilmiştir. Buna göre, akış şifreleme şifrelenmemiş metnin kaotik haritanın sözde rastgele çıkışı ile XOR’lanmasıyla elde

edilmektedir.  $x_0$  başlangıç şartı hem rastgele sayı üreticisinin çekirdek (seed) değeri hem de şifreleme sisteminin gizli anahtarının parçasıdır.



Şekil 2.17. Kaotik sistem tabanlı bir akış şifreleyicinin blok diyagramı.

Şekil 2.17'deki bir diğer önemli husus kaotik haritanın çıkışından tanım aralığı dahilinde sürekli değerli bir çıkış yerine bit dizisi alınmasıdır. Bu durum farklı tekniklerle sağlanır. Bu tekniklerden biri kaotik sistemin sürekli değerlerinden yani yörüngesinden kısmi bitler çıkartmaktır [77-79]. Diğer bir teknik ise haritanın tanım aralığını parçalara bölerek her bir parçaya bir değer atamaktır [80,81]. Bu tekniğin özel durumu tanım aralığını ikiye bölerek bu değerler içerisindeki çıkışlara karşılık 0 ve 1 şeklinde bit üretmektir.

Burada kullanılacak olan harita türü için literatürde Lojistik harita, Henon haritası gibi periyot katlamalı ve dayanıklı kaotik olmayan seçenekler de önerilmiştir [78,82,83]. Fakat haritaları incelerken belirttiğimiz üzere bu tür haritaların tekdüze dağılım gösterdiği kaotik parametre değerleri oldukça sınırlıdır ve kaotik davranışlar arasında periyodik pencereler bulunur. Bunun dışında, rastgele sayı üretici olarak kullanılacak bir sistemin tanım aralığı konvansiyonel olarak (0,1) olmalıdır [84]. Yine bu özellik de sadece sınırlı parametre değerleri için geçerlidir. Bu durum çerçevesinde baktığımızda periyot katlamalı haritalar tercih edildiği zaman kontrol parametrelerinin anahtar olarak kullanılması istenmeyen aralıkta veya kaotik olmayan çıkışlara neden olmaktadır. Sadece başlangıç şartlarının anahtar olarak kullanılması ise anahtar uzunluğunu sınırlayan ve istenmeyen bir faktördür [85].

Lojistik haritayı rastgele sayı üretici olarak kullanan ve bu durumu dikkate almayan kaos tabanlı bir kriptografik sistemin kırılmasına örnek olarak [58] verilebilir. Bahsedilen çalışmada, kontrol parametrelerinin tekdüze olmayan dağılımlarından faydalanarak anahtarın tümü çıkarılabilmektedir. Bu örnek göz önüne alındığında, periyot kırılma bifürkasyonu gösterebilir veya göstermez bir kaotik haritanın çıkışı tekdüze dağılım göstermiyorsa elde edilecek olan sözde rastgele sayılarla yapılacak bir şifrelemenin güvenli olmayacağı sonucu ortaya çıkmaktadır. Çünkü tekdüze olmayan bir çıkış için yapılacak bir histogram analizi o kaotik haritanın kontrol parametrelerinin ne olduğunu verecektir ve kontrol parametresi kriptografik anahtarın bir parçasıysa anahtar bilgisi çıkarılmış olacaktır. Bu durum karıştırma şartını sağlamayan bir güvenlik açığı teşkil etmektedir.

Tüm bu unsurlar dikkate alındığında kriptografik uygulamalarda kullanılacak kaotik haritaların arada periyodik pencereler olmaksızın geniş parametre aralığında kaotik davranış (dayanıklı kaos) göstermesi ve çıkış dağılımının tüm bu aralıkta tekdüze olması gerektiği sonucu çıkarılmaktadır. İncelemiş olduğumuz kaotik haritalardan yalnızca PDKH'nın bu şartların hepsini sağladığı görülmektedir. Dolayısıyla, Şekil 2.17'de kullanılacak olan kaotik harita da PDKH gibi bir harita olmalıdır.

### 3. BÖLÜM

## KAOTİK SİSTEM TABANLI SÖZDE RASTGELE SAYI ÜRETEÇLERİNİN GERÇEKLEŞTİRİMLERİ

Önceki bölümde kaotik haritaların sözde rastgele sayı üretiminde kullanımının teorik olarak mümkün olduğundan ve bu yönüyle kaos ile kriptoloji arasında yakın bir ilişki bulunduğundan bahsetmiştik. Bu bölümde ise kaotik haritaların donanımsal veya yazılımsal pratik gerçekleştirimlerinden bahsedilecek ve bunların gerçek kriptografik uygulamalar için uygun olup olmadıkları incelenecektir.

#### 3.1. Sözde Kaos ve Sözde Kaotik Yörüngeler

Kaotik haritaların ayrık zamanlı yapıları nedeniyle dijital platformlarda sözde rastgele sayı üretmek ve SRSÜ kullanan kriptografik uygulamalarda kullanılmak için ideal olduklarından bahsetmiştik. Fakat dijital yapıların tümünün aynı zamanda sonlu çözünürlüğe sahip olması, sürekli değerli uzayda tanımlanmış olan kaotik yapıların teorik altyapısının geçerliliğini ortadan kaldırmakta ve kaotik haritaları öngörülenden bambaşka davranışlar içerisine sokmaktadır.

Bu durumu açıklamak için biri sürekli değerli uzayda diğeri ise ayrık değerli uzayda yani sonlu çözünürlükte tanımlı iki tane lojistik haritayı ele alalım. Ayrık değerli uzayda tanımlı haritanın çözünürlüğü noktadan sonra iki basamak olsun. Her iki harita da 0.1 değeriyle iterasyona başlatılsın. Sürekli değerli uzaydaki harita için yapılacak iterasyonların sonucu  $\{0.1, 0.36, 0.9216, 0.28901376..., 0.821939226..., 0.585420538..., \dots\}$  şeklinde kendini asla tekrarlamayacak biçimde kaotik davranış sergiler ve elde edilen çıkış serisi kaotik bir yörünge oluşturur. Ayrık değerli haritada aynı iterasyonu yaptığımızda elde edilecek çıkış serisi  $\{0.1, 0.36, 0.92, 0.29, 0.82, 0.59,$

0.96, 0.15, 0.51, 0.1, 0.36, ...} şeklinde sonlu elemanlı bir seri olacak ve yuvarlama hataları yüzünden periyodik olarak kendini sürekli tekrar edecektir [86]. İncelenen ikinci seride görüldüğü üzere yuvarlama hataları başlangıç şartı olan 0.1' in 9. iterasyonda tekrar elde edilmesini sağlamıştır. Böylelikle seri bu noktadan itibaren kendini tekrar etmiştir. Oysa ki kaos, faz uzayında kendini asla kesmeyen fraktal bir çekere ve sonsuz uzunlukta rastgele benzeri periyotlara sahip olarak tanımlanmıştı. Bu nedenle bu periyodik karakteristik gösteren ayırık değerli haritanın çıkışını kaotik olarak nitelendiremeyiz. Dolayısıyla bu tür sonlu çözünürlüklü sistemler üzerinde kaosun gerçekleşmesi mümkün değildir.

Yine de kaos teorisinin gelişiminin sonlu çözünürlüğe sahip bilgisayar teknolojisindeki gelişmelere paralel ilerlediği ve bilgisayarlar üzerinde yapılan simülasyonlara bağlı olduğu yadsınamaz bir gerçektir [87]. Bu durum adeta bir ikilem oluşturmaktadır. Dijital platformlar üzerinde kaosun gerçekleştirilemeyeceğini bilen bilim adamları, bu ikilemi gidermek ve bu tür bilgisayar simülasyonlarının geçerliliğini teyit etmek için gölgeleme lemmasını (shadowing lemma) ortaya atmış ve sıklıkla kullanmışlardır [88,89]. Gölgeleme lemması, belli şartlar altında kaotik bir sistemin numerik simülasyonu ile elde edilecek yörüngeleri çok yakın bir şekilde takip eden yani onu gölgeleyen gerçek bir kaotik yörüngenin bulunduğunu öngörür [88]. Fakat bu simülasyonların sonucu gerçek kaotik davranışları ne kadar yakın yansıtırsa yansıtın sonuçlar sadece sonlu zaman aralığı için geçerli olup eninde sonunda yuvarlama hataları nedeniyle periyodik hale gelerek gerçek kaotik davranıştan uzaklaşacaktır. Bu nedenle, teorik olarak ortaya konmuş kaos ve kaotik sistemlerin sınırlı çözünürlüğe sahip dijital platformlarda gerçekleştirimi arasındaki farkı belirtmek için ilki gerçek kaos olarak adlandırılırken ikincisi sözde kaos olarak adlandırılır. Benzer şekilde gerçek kaosun yörüngeleri gerçek yörünge şeklinde ifade edilirken, sözde kaosun yörüngeleri sözde yörünge olarak adlandırılır [62].

### 3.2. Sözde Kaosun Özellikleri

Sözde kaotik bir sistem için yuvarlama hataları ve sonlu çözünürlük gerçek kaos için geçerli olan pek çok özelliği tanımsız hale getirmekte veya geçersiz kılmaktadır. Bunu daha iyi açıklamak için sürekli değerli, ayırık zamanlı kaotik bir sistemin Denklem (3.1)' deki matematiksel ifadesini göz önüne alalım:

$$x_{n+1} = f(x_n) \quad (3.1)$$

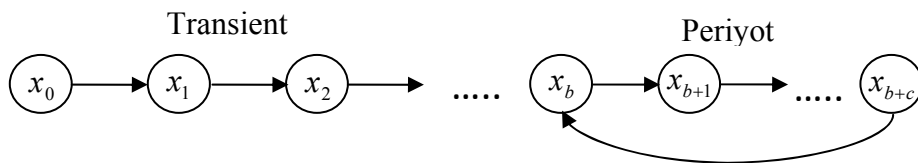
Önceki bölümde gösterilmiş olduğu üzere bu denklem teorik olarak kaosu bütün özelliklerini taşımakta ve  $\{x_i\}_{i=0}^{\infty}$  şeklinde sonsuz elemanlı kaotik bir yörüngeye neden olmaktadır. Bu ayrık zamanlı ifadenin, bilgisayar veya benzeri bir dijital platformda birebir tanımlı olduğu şekilde gerçekleştirildiğini düşünelim. Gerçekleştirmeyi yapan kişi bu denklemi gerçekleştirdiğini düşünse bile gerçekleştirilen sistemin matematiksel ifadesi aslında Denklem (3.2)'deki gibi olacaktır:

$$p_{n+1} = f(p_n) + \varepsilon_n \quad (3.2)$$

Burada,  $\varepsilon_n$  her bir iterasyondaki yuvarlama hataları olup hem yuvarlama algoritmasına hem de bit çözünürlüğüne bağlı olarak değişmektedir [88].  $p_n$  ise artık  $\{p_i\}_{i=0}^N$  şeklindeki sonlu kardinaliteye sahip sözde yörüngeyi elemanlarını temsil etmektedir. Bu açıdan bakıldığında iki denklemin birbirinden tamamen farklı olduğu görülmektedir. Netice itibarıyla özellikleri de farklı olacaktır. Şimdi, önceki bölümde gerçek kaos için verilen özellikleri ayrık değerli gerçekleştirmeler, yani sözde kaos için irdelleyelim.

### 3.2.1. Yörüngeler

Sözde kaotik sistemlerin yörüngeleri daha önce belirttiğimiz üzere periyodik davranış göstermektedir. Yörüngeler önce gerçek kaotik yörüngelerle yakın sonuç vermekte ve daha sonra yuvarlama hataları yüzünden gerçek kaotik yörüngeden tamamen farklılaşmaktadır. Bir noktadan sonra ise bu hatalar yüzünden kendini tekrar etmektedir. Bu durum literatürde Şekil 3.1'deki gibi gösterilmektedir [62]. Buna göre sözde yörüngeyi kendisini tekrar etmeden önceki kısmı transient ve kendini sürekli tekrar eden kısmı periyot olarak adlandırılmaktadır.



Şekil 3.1. Sözde kaotik yörünge yapısı [62].

Buna göre  $b$  değeri transient uzunluğuyken,  $c$  de periyot uzunluğunu vermektedir. Daha önce örnek vermiş olduğumuz iki basamaklı çözünürlüğe sahip sözde yörünge ayrıca  $b=0$  özel durumuna örnek teşkil etmektedir. Benzer şekilde  $c=0$  özel durumu da sözde yörünge'nin sabit noktaya düşmesi durumunda mümkündür. Bunların önemi, sonraki kısımlarda inceleyeceğimiz sözde rastgele sayı üreteçlerinin gerçekleştirimlerinde detaylıca vurgulanacaktır.

### 3.2.2. Lyapunov Üstelleri

Lyapunov üstelleri, birbirine oldukça yakın iki nokta arasındaki uzaklığın yörünge sonsuza giderken hesaplanan ortalama eksponansiyel değeri olarak açıklanmıştır. Tahmin edilebileceği üzere periyodik bir sistemde tüm yörüngeler arasındaki fark sürekli korunacağı için ortalama ıraksama miktarı sıfır olacaktır. Bu nedenle aslında periyodik olan sözde kaotik sistemler için Lyapunov üstelleri daima sıfır olmak zorundadır [90]. Fakat bu durum dijital sistemler üzerinde Lyapunov üstellerinin hesaplanamayacağı anlamına gelmemektedir. Aslında, özellikle çok boyutlu kaotik sistemler başta olmak üzere pek çok kaotik sistem için Lyapunov üstellerinin hesaplanması ancak numerik olarak yapılabilir ve bu tür hesaplamalar için farklı teknikler geliştirilmiştir [91].

Bu durumu açıklamak için Denklem (2.8)'deki ifade tekrar göz önünde bulundurulmalıdır. Bu denklemdaki Lyapunov üsteli hesabı iterasyon  $n \rightarrow \infty$  için geçerliydi. Fakat bilgisayar üzerindeki numerik analizlerde  $n \ll \infty$  şeklinde sınırlı iterasyon için de teorik hesaplanan değere yakın sonuçlar elde edilmektedir. Tabi bu durumun sadece sınırlı iterasyon için geçerli olduğu, periyodik tekrarlar işleme dahil oldukça sonucun teorik değerden uzaklaşıp sıfıra yaklaşacağı unutulmamalıdır.

### 3.2.3. Ergodiklik ve Topolojik Harmanlama

Ergodik teori sadece sürekli değerli sistemler için tanımlanmıştır. Öngörüleemeyen yuvarlama hataları ve sonlu çözünürlüğe sahip bir sistemde ergodik özelliğin ne olacağını açıklayan bir teori bulunmamaktadır [62]. Dolayısıyla sözde kaosun ergodik olduğundan bahsedilemez. Benzer şekilde topolojik harmanlama da periyodiklikle birlikte yok olmaktadır. Topolojik harmanlama, çeker üzerindeki herhangi bir noktadan başlayan yörünge'nin çeker üzerindeki tüm noktalara gelişigüzel yaklaşacağı anlamına

gelmekteydi. Fakat sonlu değerli uzayda bu durum farklılaşmakta ve ileride göreceğimiz üzere sözde yörüngeler çeker üzerinde sadece belli noktalara yaklaşmaktadırlar.

### 3.3. Kaos Tabanlı SRSÜ'lerin Gerçekleştirim Problemleri ve Çözümleri

Teorik olarak gerçek kaos ile sözde rastgele sayı üretimi ve kriptografik arasında yakın bir ilişki olsa da kaos tabanlı SRSÜ'lerin pratik uygulamalar için gerçekleştirmelerinde kaosun artık elde edilememesi ve beklenen özelliklerin sağlanamaması rastgele sayı üretiminde ciddi handikaplara neden olmaktadır. Bu durumu göz ardı edip teorik olarak ortaya konulmuş gerçek kaosun özelliklerinden yola çıkarak bir rastgele sayı üretici tasarlanıp donanımsal olarak gerçekleştirilirse elde edilecek üreteç pratik uygulamalar için yeterli uzunlukta sözde rastgele sayı üretemeyeceği gibi kriptografik olarak da tamamen güvensiz olacaktır. Nitekim, kriptografik uygulamalarda kullanılmak üzere literatüre sunulmuş ilk kaos tabanlı SRSÜ tasarımında sonlu çözünürlüğün etkisi tamamen göz ardı edilmiş, buna rağmen sunulan tasarımın kriptografik olarak güvenli özelliklere sahip olduğu iddia edilmişti [92]. Fakat sonradan [93]'de, bu yapının Şekil 3.1'e uygun biçimde periyodik çıkışlar göstermesi ve tek hassasiyetli kayan nokta kullanılarak bilgisayar üzerinde gerçekleştirildiğinde periyot uzunluklarının da ortalama 2993 gibi çok kısa bir değer olarak bulunması yüzünden hiçbir ciddi kriptografik uygulamada kullanılmayacağı açıkça gösterilmiştir.

Bunun dışında yuvarlama hataları ve sonlu çözünürlük gerçekleştirimi için genel geçer bir teoremin bulunmaması, oluşacak periyotların uzunluğunun ne olacağı veya yuvarlama hatalarının tam olarak nerede gerçekleşeceğine dair bir fikir vermemektedir. Aynı nedenden dolayı ergodiklik, harmanlama, Lyapunov üstelleri gibi özelliklerin durumu da açıklığa kavuşturulmuş değildir. Bu daha önce belirtilmiş olan kaos ile kriptografi arasında bulunan ergodiklik-harmanlama ve yayılma-karıştırma ilişkisinin de pratik gerçekleştirimden itibaren şüpheye düşmesine neden olmaktadır.

Bunlar göz önüne alındığında sonraki çalışmalarda teorik altyapıdan ziyade mühendislik çözümleriyle ortalama periyot uzunluklarının artırılması amaçlanmış ve kısa periyot sorununun çözümüne odaklanılmıştır. Ortalama periyot uzunluğu arttırmak için önerilen çözümler ana hatlarıyla bit çözünürlüğünü arttırmak, kaotik haritaları kaskat bağlamak

ve kaotik haritaları belli zaman aralıklarıyla pertürbasyona uğratmak şeklinde kategorilere ayrılabilir [62].

### 3.3.1. Bit Çözünürlüğünü Arttırmak

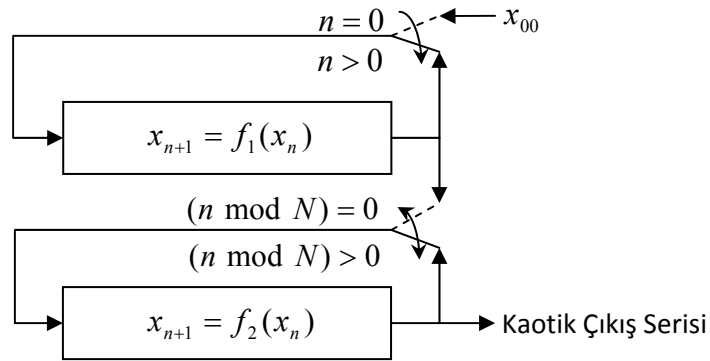
Kaotik sistemlerin gerçekleştirimlerinde bit çözünürlüğünün artırılmasının ortalama periyot uzunluğunu eksponansiyel olarak arttırdığı deneysel olarak gözlemlenmiş ve kısa periyotları gidermek için önerilmiştir [94]. Bu yapılabilecek en basit çözüm olmakla birlikte iki tane dezavantajı vardır. Birincisi, iyi bir sözde rastgele sayı üreticinin ciddi uygulamalarda kullanılabilmesi için periyot uzunluğunun  $10^9$  mertebesinde olması gerekir ki bunu sağlayabilecek bit çözünürlüğü bilgisayar üstünde sağlanamaması mümkün olduğu gibi harici donanım kullanıldığında da çok fazla kaynak tüketimine neden olup hızı düşürmektedir. İkinci olarak da ortalama periyot uzunluğunu arttırmak kısa periyotların varlığını yok etmemektedir ve bu durum uygulamalar için hala handikap yaratmaktadır. Kullanılan bir başlangıç şartı için periyot uzunluğunun ne olup olmayacağı hakkında fikir sahibi olmamak ve bu tür kısa periyotların mevcudiyeti bu çözümün de ciddi kriptografik uygulamalar için elverişli olmadığı sonucunu vermektedir.

### 3.3.2. Kaotik Haritaları Kaskat Bağlamak

Bu çözüm Heidari-Bateni ve McGillem tarafından önerilmiştir [95]. Önerilen yöntemde iki veya daha fazla kaotik harita kaskat bağlanarak her N iterasyondan sonra bunların birbirlerini set etmesi ve baştan başlatması prensibine göre çalışır. Bu sayede ortalama periyot uzunluğu etkin bir şekilde uzatılabilir. Fakat ne kadar çok kaotik harita kullanılırsa gerçekleştirim algoritması o kadar kompleks hale gelecek ve kaynak tüketimi artarken hız azalacaktır. Ayrıca haritaların bu şekilde bağlanması çıkışın tekdüzeliğini de ortadan kaldırmaktadır [62].

Kaskat bağlanmış iki tane kaotik haritanın yapısı Şekil 3.2’de görülmektedir. Buradan görüldüğü üzere başlangıç anında ilk kaotik haritaya  $x_{00}$  ilk değeri atanmaktadır. Ardından her N iterasyonda ikinci haritaya bağlı anahtar yön değiştirerek birinci haritanın o anki çıkışını ikinci haritanın başlangıç değeri olarak atamaktadır. Böylece ikinci harita bu yeni değerden itibaren iterasyona başlayıp her N iterasyon sonunda

anahtar tekrar yer değıştirince yeniden set edilmektedir. Seçilecek olan  $f_1$  ve  $f_2$  kaotik fonksiyonları birebir aynı olabileceğı gibi, aynı tür kaotik haritanın farklı kontrol parametrelerine sahip fonksiyonları ve hatta bambaşka kaotik haritalara ait fonksiyonlar da olabilir. Bu yöntemle ikinci haritanın çıkışına aynı şekilde kaotik harita bağlanmasıyla kat sayısı istenildiğı kadar arttırılabilir.



Şekil 3.2. İki tane kaotik haritanın kaskat bağlanması [95].

### 3.3.3. Kaotik Haritaları Pertürbe Etmek

Bu yöntemde kaotik haritayı belli zaman aralıklarında ( $\Delta$ ) bir başka sözde rastgele sayı üretici ile pertürbe ederek periyot uzunluğunu arttırmak amaçlanır [96,97]. Pertürbasyon için kullanılacak olan sözde rastgele sayı üretici genellikle kolay gerçekleştirim imkanı ve tekdüze çıkış sağlayan m-serili lineer geribeslemeli kaydırmalı kaydedici (LFSR) olarak seçilir. m-serili LFSR'nin mertebesi L bit olarak seçilirse çıkış periyodu ( $2^L - 1$ ) olacaktır. Bu üreticin kaç bitinin kaotik haritayı pertürbe etmek için kullanılacağı ise genel olarak Denklem (3.3) ile hesaplanacak olan sinyal gürültü oranı (SNR) ile belirlenir. Buna göre SNR 1'den oldukça büyük olmalıdır [96].

$$SNR = 10 \log_{10} \left( \frac{\text{Kaotik Sinyalin Maksimum Genliği}}{\text{Pertürbasyon Sinyalinin Maksimum Genliği}} \right) \quad (3.3)$$

Pertürbasyon işlemi ise seçilen bit sayısı uyarınca LFSR' den alınan bitlerin kaotik sistemin son bitleri ile XOR' lanmasıyla gerçekleştirilir. Bu sayede aslında  $\Delta$  zaman aralıklarıyla kaotik sistemin son değerine çok yakın bir başlangıç noktasından

başlatılarak periyot kısmının tekrar edilmemesi sağlanır [97]. Böylece de periyot uzunluğu  $T$ , Denklem (3.4) ile verilen biçimde arttırılmış olur.

$$T = \sigma \Delta (2^L - 1) \quad (3.4)$$

Burada,  $\sigma$  pozitif bir tamsayı olup kaotik sistem ve gerçekleştirim türüne bağlı olarak sürekli değişmektedir. Fakat tüm kaotik sistemler için  $\sigma$  en az 1 olabileceği için pertürbasyona dayalı kaotik haritalar için minimum periyot değeri Denklem (3.5)'teki gibi olacaktır.

$$T_{\min} = \Delta (2^L - 1) \quad (3.5)$$

Görüldüğü üzere kaotik sistemleri pertürbe ederek sistemin minimum periyodu teorik olarak hesaplanabilmekte ve düşük bit çözünürlüğüne rağmen kullanılan diğer SRSÜ'ye bağlı olarak çok daha yüksek periyot uzunluğu elde edilebilmektedir. Bunun yanı sıra, kaotik sistemin tekdüzelik gibi özellikleri de bozunmaksızın korunmaktadır [62]. Bu bağlamda, bu yöntemin kısa periyotları gidermek için kullanılabilecek en iyi yöntem olduğu söylenebilir.

### 3.4. Periyot Oluşum Süreci ve Üretilen Periyotların Korelasyon Özellikleri

Literatürde kısa periyot problemini bu bölümde bahsedilen metotlarla çözmüş olan pek çok farklı sözde rastgele sayı üretici ve kaos tabanlı kriptosistem modeli bulunmaktadır [29,34,98-103]. Bunlar her ne kadar uzun periyotlara sahip olsa da, bu sistemlerin uzun periyotlara sahip olması kriptografik olarak güvenli oldukları anlamına gelmemektedir. Çünkü bu tür sistemlerin kriptografik olarak güvenli olup olmadığının test edilebileceği genelleştirilmiş bir metot bulunmamaktadır. Dolayısıyla, bu tür kaos tabanlı kriptosistemlerin güvenilirliklerini doğrulamak için kaotik sistem çıkışına istatistiki rastgelelik testleri uygulanmakta ve önerilen modelin uzun periyotlar, sıfıra yakın kros-korelasyon ve  $\delta$  benzeri otokorelasyon gibi özelliklere sahip olduğu vurgulanmaktadır. Ayrıca kaotik haritaların başlangıç şartlarına hassas bağıllık özelliğine atıf yapılarak başlangıç koşulları genellikle kriptosistemin anahtarı veya anahtar parçası olarak kullanılmakta ve gerçekleştirim detayları bir kriptosisteme göre yüzeysel olarak ele alınmaktadır [85]. Mesela, çoğu kaos tabanlı rastgele sayı üreticinde yuvarlama hatalarının etkisi bilinse bile hangi tür yuvarlamanın kullanıldığı veya kullanılması

gerektiği açıklanmamaktadır. Bunun dışında, önerilen çoğu sistem için nümerik simülasyonlar çift hassasiyetli kayan noktalı aritmetik için verilmektedir. Bu ise sadece yazılımsal bilgisayar gerçekleştirmeleri için uygun olup sabit noktalı aritmetiğin tercih edildiği diğer elektronik donanım çözümlerinde yapılacak gerçekleştirmeler açısından net bir fikir vermemektedir. Tüm bunlar göz önüne alındığında kaos tabanlı rastgele sayı üretiminin kriptografide kullanımını konusunda halen ciddi sıkıntıların mevcut olduğu görülmektedir. Bu nedenle kaotik sistemlerin periyotlarının oluşum süreci ve bu periyotlar arasındaki korelasyon ilişkileri hassasiyetle ele alınmalıdır.

Literatürde çok fazla sayıda kaos tabanlı rastgele sayı üretici kriptografik uygulamalarda kullanılmak üzere önerilmiş olsa da nispeten az sayıda çalışmada periyot uzunluklarının nümerik olarak analizi yapılmış [93,94,104-107] ve çok daha az çalışmada periyot oluşum sürecinin teorik altyapısı sunulmaya çalışılmıştır [108-111]. [108]'de kaotik bir haritanın ayırık değerli uzayda gerçekleştiriminin ortalama periyot uzunluğu Denklem (3.6)'daki gibi hesaplanmıştır:

$$T \sim \varepsilon^{-D/2} \quad (3.6)$$

Burada  $\varepsilon$  mevcut çözünürlük ve  $D$  ise kaotik haritanın korelasyon boyutudur. Fakat periyot uzunluklarını nümerik olarak inceleyen sonraki çalışmalarda görüldüğü üzere farklı başlangıç değerleri için gerçekleştirimlerde farklı çözünürlükler için sürekli karşılaşılma olasılıkları yüksek olan baskın periyot uzunlukları elde edilmektedir [105,106]. Bu nedenle de teorik öngörü ile simülasyon sonuçları uyuşmamaktadır. Nümerik simülasyonların yapıldığı bu çalışmalar karşılaşılan bu baskın periyot uzunluklarının aynı olup olmadığını, yani başlangıç şartlarına hassasiyetin kaybolup kaybolmadığını incelemeksizin diğer çalışmalarda olduğu gibi kısa periyot sorununa odaklanmakta ve periyot uzunluklarının artırılması için metotlar önermektedirler [106, 107]. Fakat bu sürekli karşılaşılan baskın periyotların aslında aynı değerleri içeren seriler olup olmadıkları tartışılmamaktadır. Oysa ki [108]'de ortalama periyot uzunluklarının teorik hesabı dışında iki farklı sözde yörünge için yuvarlama hataları yüzünden kesişerek aynı periyot içerisine düşebileceğinden de bahsedilmektedir ki bu sonuç başlangıç şartlarını anahtar olarak kullanan kaos tabanlı kriptosistemler için ayrıca önemlidir. Bu nedenle bu durumu açıklığa kavuşturmak için öncelikle periyot uzunluklarının nümerik analizleri yapılarak baskın periyotlar ve karşılaşılma sıklıkları

farklı yuvarlama türleri için incelenecek, ardından bu sonuçlar ışığında periyotların oluşum ve kesişim süreçlerinin nasıl oluştuğu korelasyon ilişkileri ışığında belirlenecek ve en sonunda ise elde edilen sonuçların kriptosistemler üzerindeki etkisi yorumlanacaktır. Bu tez çalışmasında kaotik sistem tabanlı SRSÜ'lerin donanım gerçekleştirimlerine ağırlık verildiği için tüm nümerik analizler kayan noktalı aritmetik yerine donanım gerçekleştirimlerine uygun olan sabit noktalı aritmetik kullanılarak gerçekleştirilecektir.

### 3.4.1. Periyot Uzunluklarının Nümerik Olarak Hesaplanması

Periyot uzunluklarının nümerik analizinin yapıldığı ilk çalışma olan [93]'de periyot uzunluklarının belirlenmesi için basit bir algorithmadan faydalanılmıştır. Buna göre periyot uzunluğunu belirlemek için bilgisayar üzerinde aynı kaotik haritanın biri diğerinin iki katı hızda olacak şekilde iki farklı iterasyonu alınır. Bu iterasyonlar bir kez eşitlendikten sonra eşitliğin sağlandığı değer tekrar yakalanana kadar haritanın iterasyonu gerçekleştirilir ve bu esnada bir sayaç iterasyon miktarını sayar. Sayacın son değeri periyot uzunluğunu verir. Bu algoritma oldukça basit olmakla beraber bilgisayar üzerinde hafıza kullanımı oldukça azdır. Bunun yanı sıra periyot uzunluklarının nümerik analizinin yapıldığı sonraki çalışmalarda da kullanılmıştır [106]. Fakat bu algoritma ile transient uzunluklarının ne olduğu bulunamamaktadır. Dolayısıyla da tüm sözde yörünge hakkında fikir vermeksizin sadece periyot uzunluğunu hesaplamada faydalıdır.

Bu durum göz önüne alındığı için bu çalışmada periyot uzunluklarının hesaplanması için hafıza kullanımına dayalı farklı bir algoritma takip edilmiştir. Buna göre ilk önce kaotik harita gerçekleştiriminin  $N$  adet iterasyonu gerçekleştirilip elde edilen değerler hafıza üzerine kaydedilmektedir. Bu sayede elde edilen seri  $\{x_i\}_{i=1}^N$  olsun. Hafızaya yazılı seri arasından son değer kendisi hariç tutulmak üzere son değere eşit olan elemanlar aranır. Eğer hiç eşit eleman bulunamazsa başa dönülüp  $N$  değeri artırılır ve işlemler tekrarlanır. Buna göre eşit elemanlardan oluşan alt küme  $S = \{x_{r_1}, x_{r_2}, \dots, x_{r_N}\}$  olsun. Buna göre, periyot uzunluğu bu alt kümenin ardışık elemanlarının indisleri farkına  $(r_2 - r_1)$  eşit olacaktır. Bu sözde yörüngenin transient kısmı ise Denklem (3.7) ile verilen işlem gerçekleştirilerek bulunabilir.

$$\{x_i\}_{i=1}^{r1} \setminus \{x_i\}_{i=r1}^{r2-1} \quad (3.7)$$

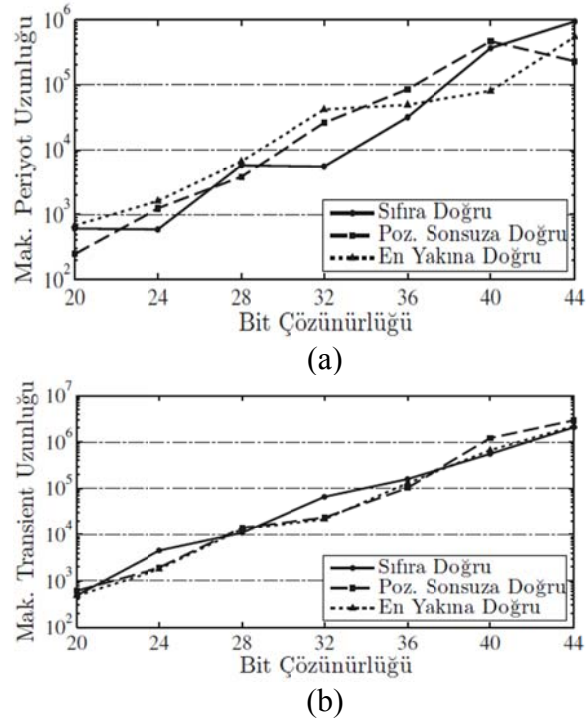
### 3.4.2. Sabit Noktalı Aritmetik ve Yuvarlama Türleri

Sabit noktalı nümerik notasyon küsuratlı kısmı ifade etmek için sabit bit uzunluğunun kullanıldığı notasyon türüdür. Bir sabit noktalı sayı ile eşit bit uzunluğunun kullanıldığı kayan noktalı nümerik notasyon, sabit noktaliya göre çok daha geniş bir hassasiyet aralığı sunsa da sabit noktalı notasyon hem daha hızlı hem de kaynak gereksinimi daha azdır. Ayrıca kayan noktalı aritmetikte yapılan işlemlerde küsuratlı kısmın hassasiyetinin sürekli değişmesi kaotik haritaların gerçekleştirimleri açısından tekdüzeliği daha da bozan ve teorik hesaplamaları ayrıca güçleştiren bir durumdur [106]. Bu bakımdan, kaotik sistem tabanlı SRSÜ' lerin donanım gerçekleştirimleri için sabit noktalı aritmetiğin kullanılması daha avantajlıdır.

Sabit noktalı aritmetik için kayan noktalı aritmetikte olduğu gibi yuvarlama türleri açısından ortaya konmuş bir standart bulunmamaktadır. Kayan noktalı aritmetikte pozitif sonsuza doğru yuvarlama, negatif sonsuza doğru yuvarlama, sıfıra doğru yuvarlama, en yakına doğru yuvarlama gibi IEEE 754 standardıyla genelleştirilmiş olan yuvarlama türleri bulunmaktadır [112]. Matlab yazılımı, sabit noktalı aritmetik için buna benzer yuvarlama türleri seçenekleri sunmaktadır. Her ne kadar bu yuvarlama türleri açık kaynak kodlu olarak sunulmasa da yaptıkları işlem bakımından yukarıda kayan noktalı aritmetik için bahsedilen yuvarlama türleri ile eşdeğer sonuç vermektedirler. Bu bağlamda pozitif sonsuza doğru yuvarlamada yuvarlama işlemleri sürekli daha büyük olan sayıya doğru gerçekleştirilirken, negatif sonsuza doğru yuvarlamada tam tersi daha küçük olan tamsayıya doğru gerçekleştirilir. Sıfıra doğru yuvarlamada ise pozitif sayılar için daha küçük tamsayıya doğru yuvarlama yapılırken, negatif sayılar için daha büyük tamsayıya doğru yuvarlama yapılır. En yakına doğru yuvarlamada ise yuvarlama en yakın tamsayıya doğru gerçekleştirilir. Bu bakımdan kümülatif hatanın en az olduğu yuvarlama türüdür. Bu yuvarlama türünde sayı iki tamsayıya eşit uzaklıktaysa tercihe bağlı olarak tek veya çift sayıya doğru yuvarlama yapılır. Birazdan göreceğimiz üzere her yuvarlama türü kaotik haritaların gerçekleştirimlerinde farklı bir etkiye neden olmaktadır.

### 3.4.3. Farklı Çözünürlük ve Yuvarlama Türleri için Periyot Uzunlukları

Bu kısımda yapılacak olan nümerik analizlerde literatürde en çok incelenmiş kaotik haritalardan biri olması münasebetiyle lojistik haritanın sabit noktalı aritmetik kullanılarak Matlab üzerinde farklı yuvarlama türleri için gerçekleştirimi ele alınmıştır. Her bit çözünürlüğünde tekdüze dağılım gösteren 1000 adet farklı rastgele başlangıç şartı için sözde yörüngeler hesaplanmış olup transient ve periyot uzunlukları daha önce açıklanan algoritma yardımıyla bulunmuştur. Sabit noktaları elimine etmek için kontrol parametresi 3.99 olarak seçilmiştir. Lojistik haritanın farklı yuvarlama türleri ve çözünürlükler için hesaplanmış olan maksimum transient ve periyot uzunlukları Şekil 3.3'de görüldüğü gibi elde edilmiştir.



Şekil 3.3. Lojistik harita gerçeştiriminin farklı çözünürlükler için a) maksimum periyot uzunlukları, b) maksimum transient uzunlukları.

Elde edilen sonuçlar daha önce [94]'de bulunan bit çözünürlüğü arttıkça periyot uzunluklarının eksponansiyel olarak artması sonucuyla paraleldir. Fakat farklı yuvarlama türleri açısından bakıldığında gerçeştirimlerde kullanılacak yuvarlama türünün oldukça önemli olduğu görülmektedir. Bit çözünürlüğünü arttırmak periyot uzunluğu bakımından her zaman olumlu sonuçlar vermemekle birlikte belli bir yuvarlama türünün diğerlerinden daha iyi olduğu sonucu da çıkarılamamaktadır. Bu

yüzden belirli bir bit çözünürlüğü için hangi yuvarlama türünün daha iyi olduğuna bakılmalıdır. Yani beklenenin aksine, en yakına yuvarlama kümülatif hatanın en az olduğu ve en çok kaynak gerektiren yuvarlama türü olmasına karşın [113], her zaman için en iyi sonuçları vermemektedir. Mesela, bit çözünürlüğü 40 bit iken gerçekleştirim açısından daha az kaynak gerektiren diğer basit metotlar daha uzun periyotlar sağlamaktadır. Benzer şekilde, yuvarlama türü sıfıra doğruyken çözünürlüğün 28 bitten 32 bite çıkarılması maksimum periyodu arttırmamıştır. Bu nedenle bu yuvarlama türünün kullanıldığı bir gerçekleştirimde 28 yerine 32 bit kullanmak performansı arttırmayıp sadece daha fazla kaynak tüketimine neden olacaktır.

Transient uzunluğu değişimi ise, tüm yuvarlama türleri için nispeten daha düzgün bir artış göstermektedir. Transient uzunlukları sonradan gösterileceği üzere farklı başlangıç şartları için çok geniş bir dağılım gösterdiği için, maksimum transient uzunluğunun hangi yuvarlama türü veya bit çözünürlüğünde daha fazla olacağı tek başına bir anlam ifade etmemektedir. Bu durum, periyot oluşum süreci incelenirken detaylıca açıklanacaktır. Fakat şimdi elde ettiğimiz sonuçları sadece maksimum periyotlar için değil farklı başlangıç şartları için irdeleyelim. 1000 adet tekdüze dağılımlı rastgele başlangıç şartı için farklı bit çözünürlüğü ve yuvarlama türleriyle elde edilen periyot uzunlukları ve bunların 1000 adet başlangıç şartı arasında karşılaşılma olasılıkları Tablo 3.1’de özetlenmiştir.

Tablo 3.1 incelendiği zaman farklı başlangıç şartları için daima sınırlı sayıda periyot uzunluğunun elde edildiği görülmektedir. Bu sınırlı periyotlar arasından da en az biri baskın bir karşılaşılma olasılığını elinde bulundurmaktadır. Bu sonuçlar [105,106]’da sunulan daha önceki bulguları doğrular niteliktedir. Mesela sıfıra doğru yuvarlamanın kullanıldığı 20 bit çözünürlüklü gerçekleştirimde uzunlukları 24 ve 610 olmak üzere sadece iki tane periyoda rastlanmıştır. Bunların karşılaşılma olasılığı ise sırasıyla 0.012 ve 0.998’dir. Yani her 1000 farklı başlangıç şartının 998’inde periyot uzunluğu 610 olarak bulunmaktadır. Şimdi burada odaklanılması gereken asıl konu bu periyotların uzunluğunu arttırmaya çalışmaktan ziyade farklı başlangıç şartları için elde edilen bu periyotların içerdikleri değer bakımından da birebir aynı olup olmadığının incelenmesi olmalıdır. Keza uzunluğu aynı olan periyotlar içerdikleri elemanlar bakımından da birebir aynıysa, bu durumda başlangıç şartlarına hassas bağımlılık ortadan kalkmış olur

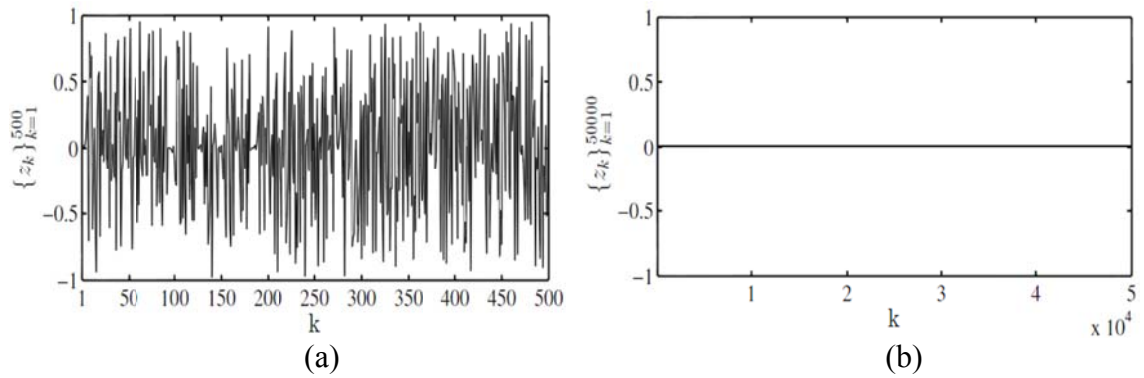
ve kriptografik uygulamalarda istenen yayılma (diffusion) özelliği de artık sağlanamaz. Bu husus elde edilecek sonuçlara bağlı olarak tekrar yorumlanacaktır.

Tablo 3.1. Farklı bit çözünürlükleri ve yuvarlama türleri için elde edilen periyot uzunlukları.

| Yuvarlama Türü     | Bit Çözünürlüğü | Periyot Uzunlukları ve Olasılıkları                                                                     |
|--------------------|-----------------|---------------------------------------------------------------------------------------------------------|
| Sıfıra Doğru       | 20              | 24(0.012), 610(0.998)                                                                                   |
|                    | 24              | 162(0.033), 588(0.967)                                                                                  |
|                    | 28              | 250(0.03), 4216(0.323), 5760(0.674)                                                                     |
|                    | 32              | 357(0.001), 468(0.001), 1334(0.001), 5001(0.02), 5541(0.997)                                            |
|                    | 36              | 111(0.001), 31810(0.999)                                                                                |
|                    | 40              | 21545(0.001), 366400(0.999)                                                                             |
|                    | 44              | 28770(0.007), 119945(0.04), 200944(0.005), 518132(0.023), 606897(0.079), 936866(0.846)                  |
| Poz. Sonsuza Doğru | 20              | 103(0.059), 140(0.078), 184(0.789), 246(0.074)                                                          |
|                    | 24              | 4(0.001), 15(0.018), 103(0.004), 251(0.017), 516(0.356), 722(0.138), 1249(0.466)                        |
|                    | 28              | 236(0.027), 778(0.002), 1134(0.015), 1671(0.55), 3350(0.252), 3824(0.154)                               |
|                    | 32              | 7(0.001), 54(0.001), 540(0.001), 562(0.001), 1111(0.001), 2138(0.009), 26167(0.986)                     |
|                    | 36              | 70(0.004), 4265(0.006), 18306(0.053), 85679(0.937)                                                      |
|                    | 40              | 10394(0.003), 68624(0.025), 465466(0.972)                                                               |
|                    | 44              | 40800(0.017), 66561(0.616), 90441(0.002), 116130(0.249), 229862(0.116)                                  |
| En Yakına Doğru    | 20              | 7(0.026), 21(0.011), 24(0.005), 119(0.07), 679(0.888)                                                   |
|                    | 24              | 15(0.002), 152(0.08), 257(0.092), 787(0.457), 1619(0.369)                                               |
|                    | 28              | 130(0.002), 636(0.642), 658(0.026), 6672(0.33)                                                          |
|                    | 32              | 652(0.001), 1117(0.001), 1130(0.003), 6078(0.113), 9439(0.073), 41999(0.809)                            |
|                    | 36              | 592(0.033), 1236(0.001), 2377(0.001), 4689(0.083), 8555(0.026), 28791(0.267), 29304(0.08), 49356(0.509) |
|                    | 40              | 1771(0.023), 1981(0.002), 40460(0.018), 43763(0.012), 43862(0.032), 80728(0.913)                        |
|                    | 44              | 87658(0.661), 120832(0.003), 130804(0.003), 269961(0.185), 316317(0.025), 545807(0.123)                 |

### 3.4.4. Eşit Uzunluklu Periyotların Benzerlik Açısından Kıyaslanması

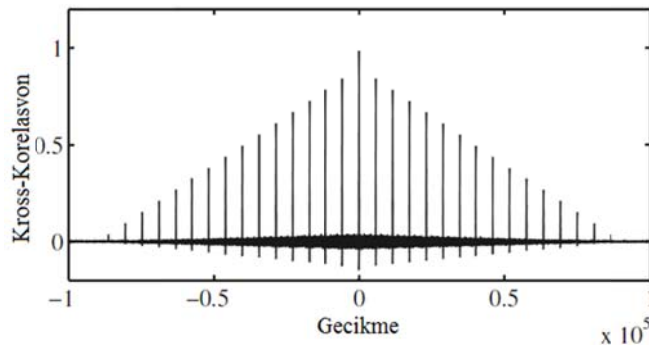
Bir kaotik haritanın başlangıç şartlarına hassas bağımlı olduğu pozitif Lyapunov üstellerine bakılarak teorik olarak ispatlanabilir. Fakat önceki kısımlarda gördüğümüz üzere bu teori ayrık değerli gerçekleştirimlerde pek faydalı bilgi vermemekteydi. Bu durumda ayrık değerli uzaydaki sözde kaotik yörüngelerin başlangıç şartlarına hassas bağımlı olup olmadığını nasıl ispat edebiliriz? Literatürde ayrık değerli gerçekleştirimlerin periyot uzunluğu üzerindeki etkileri göz önüne alınsa da bu soru genellikle ihmal edilmekte ve sözde kaosun da gerçek kaos gibi başlangıç şartlarına hassas bağımlılık özelliğini koruduğu varsayılmaktadır. Misal olarak [103]'de önerilen modelin başlangıç şartlarına hassas bağımlı olduğunu göstermek için kaotik sistemin birbirine çok yakın iki başlangıç noktasından başlayan sözde yörüngelerinin belli iterasyon boyunca farkı çizdirilmiştir. Benzer şekilde biz de 28 bit çözünürlüklü sabit noktalı lojistik harita gerçekleştirimimiz için  $x_1 = 0.53188916949292575$  ve  $y_1 = 0.50439786009276666$  noktalarından başlayarak sıfıra doğru yuvarlama metoduyla iki farklı sözde yörünge hesaplayıp bunların farkını çizdirirsek Şekil 3.4a' daki gibi bir şekil elde ederiz. Şekilden tıpkı [103]'deki sonuçta olduğu gibi iki sözde yörüngenin tamamen farklı ve başlangıç şartlarına hassas bağımlılığın mevcut olduğu izlenimi uyanmaktadır.



Şekil 3.4. İki sözde kaotik yörüngenin a) baştan itibaren, b)  $\tau_1 + 1$  ve  $\tau_2 + 1$  noktalarından itibaren farkı.

Fakat algoritmamızı kullanarak bu iki yörüngenin transient uzunluklarını hesap edip sözde yörüngeleri bu açıdan incelediğimizde ortaya bambaşka bir tablo çıkmaktadır. Elde ettiğimiz yörüngeler  $\{x_k\}_{k=1}^N$ ,  $\{y_k\}_{k=1}^N$  ve bunların transient uzunlukları sırasıyla  $\tau_1$ ,

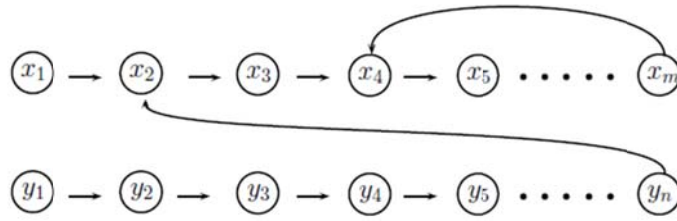
$\tau_2$  olsun. Eğer bu serilerin  $\tau_1 + 1$  ve  $\tau_2 + 1$  noktalarından başlayarak farkı çizdirilecek olursa, bu sefer Şekil 3.4b'yi elde ederiz. Görüldüğü üzere iki farklı sözde yörüngenin periyot kısımları arasında hiçbir fark yoktur. Eğer iki sözde yörünge böyle ortak elemanlar içeriyorsa, bu durum iki yörünge arasında korelasyon ilişkisi olması gerektiği anlamına gelir. Nitekim bu iki sözde yörüngenin kros-korelasyonu çizdirildiğinde Şekil 3.5 elde edilir. Bu kros-korelasyon, rastgele sayı üreteçlerinde istendiği türde sıfıra yakın olmak yerine güçlü korelasyonların bulunduğunu işaret etmektedir.



Şekil 3.5. İki sözde kaotik yörüngenin kros-korelasyonu.

İncelenen sözde kaotik yörüngeler için  $\tau_1 = 9736$ ,  $\tau_2 = 9748$  ve periyot uzunlukları ortak 5760 olarak bulunmuştu. Şekil 3.5'te en yüksek korelasyon değeri -12 değerinde konuşlanmış olup ilk sözde yörüngenin ikinciye göre  $\tau_1 - \tau_2$  kadar önde olduğu anlamına gelmektedir. Diğer tepe değerleri arasındaki gecikme farkı ise her ardışık değer için sabit olup 5760'a yani sözde yörüngelerin periyot uzunluğuna eşittir. Buna göre, bu tür korelasyon ilişkisi gösteren farklı sözde yörüngeler üzerinden periyot uzunluğu ve hangisinin periyodik bölgeye önce girdiği bilgisi çıkarılabilir.

İki sözde yörüngenin periyodik bölgelerinin aynı elemanları içermesi ve periyot uzunluğunun eşit olmasının nedeni Grebogi'nin çalışmasında iki yörüngenin yuvarlama hataları nedeniyle kesişmesi olarak izah edilmiştir [108]. Bu durum Şekil 3.6'da özetlenmiştir. Buna göre, ikinci yörünge yuvarlama hataları nedeniyle birincinin transient bölgesiyle kesişmekte ve bu noktadan itibaren birinci yörüngeyle aynı değerleri takip edip aynı periyot uzunluğunu paylaşmaktadır. Dolayısıyla da başlangıç şartlarına hassas bağımlılık bu kesişim noktasından itibaren geçersiz kalmaktadır.



Şekil 3.6. İki sözde yörüngenin transient bölgesinde kesişmesi [108].

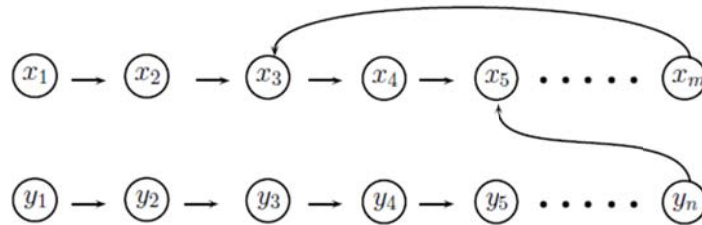
Daha önce Tablo 3.1’de göstermiş olduğumuz üzere farklı başlangıç şartları için sürekli belli uzunluklu periyotlarla karşılaşıyor ve bunlardan en az biri karşılaşımla olasılığı bakımından baskınlık gösteriyordu. Şimdi sorulması gereken asıl soru bu eşit uzunluklu periyotları veren başlangıç şartlarının kaçının bu şekilde transient bölgesinde kesiştiği olmalıdır. Yörüngeler için transient ve periyot uzunluklarını bildiğimiz için eşit uzunluklu periyotların ilk değerlerini kontrol etmemiz bu şekilde kesişen periyotların sayısını verecektir. Çünkü ilk değerleri aynı olan periyotların diğer tüm elemanları da aynı olmak zorundadır. Bu şekilde bir inceleme için 44 bitlik gerçekleştirimlerdeki periyotlar göz önünde bulundurulmuştur. Sonuçlar Tablo 3.2’de görüldüğü gibidir.

Tablo 3.2. 44 bit çözünürlüklü gerçekleştirim için periyotların karşılaşımla sayıları ve bunlar içerisindeki farklı periyotlar.

| Yuvarlama Türü     | Periyot Uzunluğu | Karşılaşılma | Farklı Periyotlar |
|--------------------|------------------|--------------|-------------------|
| Sıfıra Doğru       | 28770            | 7            | 1                 |
|                    | 119945           | 40           | 7                 |
|                    | 200944           | 5            | 4                 |
|                    | 518132           | 23           | 13                |
|                    | 606897           | 79           | 15                |
|                    | 936866           | 846          | 32                |
| Poz. Sonsuza Doğru | 40800            | 17           | 4                 |
|                    | 66561            | 616          | 3                 |
|                    | 90441            | 2            | 2                 |
|                    | 116130           | 249          | 7                 |
|                    | 229862           | 116          | 7                 |
| En Yakına Doğru    | 87658            | 661          | 5                 |
|                    | 120832           | 3            | 2                 |
|                    | 130834           | 3            | 1                 |
|                    | 269961           | 185          | 17                |
|                    | 316317           | 25           | 10                |
|                    | 545807           | 123          | 22                |

Tablo 3.2 incelendiğinde eşit uzunluklu periyotların hepsinin birebir aynı olmadığı görülmektedir. Mesela sıfıra doğru yuvarlamanın kullanıldığı gerçekleştirimde 28770 uzunluklu periyotla 7 kez karşılaşmıştır ve bu 7 periyot içinde 1 farklı periyot bulunmaktadır. Yani hepsi aslında birebir aynıdır. Fakat hemen alttaki 119945 uzunluklu periyoda baktığımızda, bununla farklı başlangıç şartları için toplamda 40 kez karşılaşıldığını ve bu 40 eşit uzunluklu periyodun 7 farklı periyottan meydana geldiği görülmektedir. Yani tüm eşit uzunluklu periyotlar birebir aynı değildir.

Bu aynı uzunlukta olup da ilk değerleri farklı olan periyotların korelasyon ilişkileri incelendiğinde Şekil 3.5'te elde ettiğimiz korelasyon ilişkisine benzer sonuçlar elde edilmektedir. Bu sonuçlara göre, eşit uzunluklu periyotların ilk değerleri aynı olmasa da bunların birbirlerinin belli aralıklarla ötelenmiş versiyonları oldukları görülmektedir. Bu da akla kesişimlerin periyot bölgesinde olduğu ikinci bir periyot oluşum sürecinin varlığını getirmektedir. Bu kesişim türü Şekil 3.7'de gösterilmektedir. Buna göre, ikinci sözde yörünge yuvarlama hataları yüzünden doğrudan birinci yörünge periyot bölgesiyle kesişmektedir.



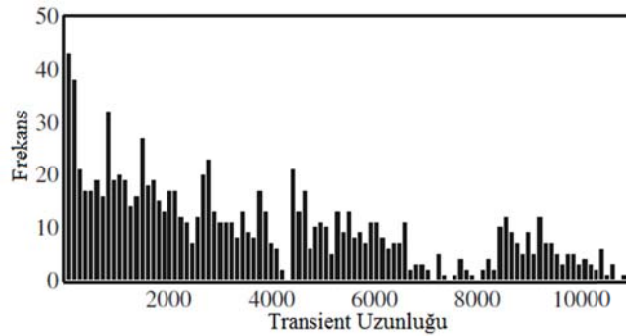
Şekil 3.7. İki sözde yörünge periyot bölgesinde kesişmesi.

Elde edilen sonuçlara göre transient kesişimi ve periyot kesişimi olmak üzere iki tür periyot kesişiminin mevcut olduğu ve bunların aynı değerleri takip eden eşit uzunluklu periyotlara neden oldukları görülmektedir. Tablo 3.2'den görüldüğü üzere ilk değeri farklı olan nispeten az sayıda eşit uzunluklu periyot olmasına karşın çoğu periyodun ilk değeri aynı olmaktadır. Bu da periyot kesişiminin, transient kesişimine kıyasla daha az gerçekleştiği anlamına gelmektedir. Netice itibarıyla hem transient hem de periyot kesişimleri başlangıç şartlarına hassas bağımlılığı ciddi derecede zedelemektedir. Tablo 3.1'de farklı başlangıç şartları için elde edilen sonuçları farklı başlangıç şartları açısından bu yeni perspektifle değerlendirdiğimizde kaotik bir sistemin ayrık değerli gerçekleştirmelerinin çıkışının çok da zengin bir çeşitlilik göstermediği görülmektedir.

Bu kısımda elde edilen sonuçların diğer haritalar için de geçerli olup olmadığına bakmak için farklı bifürkasyon türleri ve boyutlara sahip olan PDKH, tent harita ve Henon haritası için benzer şekilde hem kayan noktalı hem de sabit noktalı gerçekleştirmeler yapılmış olup sonuçta tıpkı lojistik haritada olduğu gibi farklı başlangıç şartları için Şekil 3.5'teki gibi korelasyon ilişkisi gösteren sözde yörüngeler elde edilmiştir. Bu da transient ve periyot kesişimlerinin matematiksel modelden bağımsız olarak sınırlı çözünürlük ve yuvarlama hatalarına bağlı bir olay olduğunu göstermektedir.

#### 3.4.5. Transient Uzunluklarının Benzerlik Açısından Kıyaslanması

Sözde kaotik yörüngelerin transient bölgeleri transient kesişimleri nedeniyle benzerlik gösterip aynı değerleri takip etmektedir. Dolayısıyla korelasyon ilişkileri çoğu zaman transient bölgesi içerisinde başlamaktadır. Örneğin daha önce ele aldığımız 28 bit çözünürlüklü lojistik harita gerçekleştirmelerinde transient uzunlukları  $\tau_1 = 9736$  ve  $\tau_2 = 9748$  olmasına rağmen kesişim transient içerisindeki 1343. ve 1355. iterasyonlarda gerçekleşmektedir. Buradan görüldüğü üzere, başlangıç şartlarına hassas bağımlılık transient kesişimlerinde daha periyot bölgesine girmeden kaybolmaktadır.



Şekil 3.8. Elde edilen sözde yörüngelerin transient uzunluklarının histogramı.

Transient kesişimlerinden önceki kısımların başlangıç şartlarına ne kadar bağlı olduğunu gözlemlemek için transient uzunluklarını incelememiz gerekir. Şekil 3.8'de ele almış olduğumuz 28 bit çözünürlüklü gerçekleştirim için transient uzunluklarının histogramı verilmiştir. Buna göre transient uzunlukları periyot uzunluklarının aksine belli değerlerde gruplanmak yerine geniş bir dağılım göstermektedir. Bu da transient

bölgelerinin transient kesişimlerinden önceki kısımlarının zengin bir çeşitlilik gösterip başlangıç şartlarına hassas bağımlılık koşulunu sağladığı anlamına gelmektedir.

Buna rağmen transient kesişimlerinin nerede olacağının bilinmemesi, hangi transient uzunluklarının ne kadar korelasyon ilişkisi gösterdiğini belirlemeyi oldukça zorlaştırmaktadır. Şekil 3.8 için baktığımızda transient uzunluklarının büyük çoğunluğunun 2000 değerinin altında olduğu görülmektedir. Bu da transient kesişimlerinin çoğunun bu aralıkta olduğu anlamına gelmektedir. Yani gerçek kaostaki başlangıç şartlarına hassas bağımlılık koşulunu sağlayabilecek kısım, kesin belirlenememekle birlikte genel olarak periyot uzunluğuna göre oldukça kısa kalmaktadır.

### 3.5. Sözde Kaotik Yörüngelerin Kriptosistemlerde Kullanımı

Transient ve periyot kesişimlerinin başlangıç şartlarına hassas bağımlılığı ortadan kaldırması, kriptosistemlerde olması istenen yayılma özelliğini de zedelemektedir. Eğer bu durum göz ardı edilip başlangıç şartları anahtar olarak kullanılırsa ciddi güvenlik açıkları oluşur. Şimdi bu kısımda ne gibi güvenlik açıklarının oluşacağı literatürde sunulmuş simetrik anahtarlı akış şifreleyici modelleri örnek verilerek açıklanacaktır.

İncelenecek olan ilk model [34]'de sunulmuş olup kaotik haritaların eşleştirilmesine dayanan bir akış şifreleyicidir. Kaotik haritaların eşleştirilmesi, iki kaotik haritanın çıkışlarının alınıp bir fonksiyona bağlı olarak karşılaştırılması suretiyle bit üretimi prensibine göre gerçekleştirilir. Bu şekilde iki kaotik harita kullanılmasının nedeni kriptosistemin güvenliğini daha da arttırmak içindir. [34]'de bu sistemin çift hassasiyetli kayan noktalı aritmetik kullanılarak yapılan gerçekleştiriminin NIST SP 800-22 istatistiki rastgelelik testi sonuçları verilerek sistemin mükemmel kriptografik özelliklere sahip olduğu vurgulanmaktadır. İncelenecek olan diğer model ise gene eşleştirilmiş haritalara dayanan fakat farklı olarak pertürbasyon uygulanan bir kriptosistemdir [29].

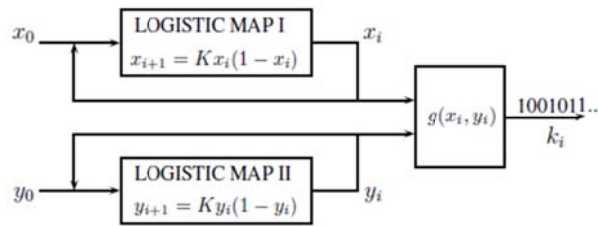
#### 3.5.1. Pertürbasyonsuz Eşleştirilmiş Kaotik Sistem

Bu kaotik sistemin blok diyagramı Şekil 3.9'da görüldüğü gibidir. İki tane lojistik haritanın çıkışının  $g(.)$  fonksiyonuna göre karşılaştırılmasıyla bit dizisi üretir ve bu

şekilde akış şifreleyici olarak kullanılır.  $x_0$  ve  $y_0$  başlangıç şartları anahtar olarak kullanılmaktadır. Kaotik harita olarak lojistik harita tercih edildiği için  $x_0, y_0 \in (0,1)$  ve  $K$  parametresi 2. Bölüm’de görmüş olduğumuz üzere 4 veya 4’e çok yakın olmalıdır. Nitekim [34]’de de  $K$  parametreleri eşit ve 4.0 olarak alınmıştır.  $g(.)$  fonksiyonu ise Denklem (3.8)’deki gibi seçilmiştir:

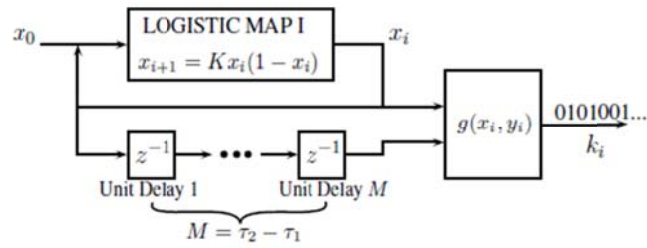
$$g(x_i, y_i) = \begin{cases} 1, & x_i > y_i \\ 0, & x_i \leq y_i \end{cases} \quad (3.8)$$

Karşılaştırma fonksiyonu yüzünden,  $K$  parametresi haritalar için aynı seçildiği müddetçe  $x_0 \neq y_0$  olmalıdır. Aksi takdirde çıkış sürekli sıfır olur.



Şekil 3.9. Eşleştirilmiş kaotik sistem [34].

Normalde eşleştirme işlemi ve farklı başlangıç şartlarının sistemi komplike bir hale getirerek bilgi sızdırmaya çalışan kişilere karşı güvenli bir şifreleme sağlaması gerekir. Önceki kısımda elde ettiğimiz sonuçlara dayalı olarak bu sistemin güvenilirliğini gözden geçirelim. Bunun için bu kaotik kriptosistemin sabit noktalı gerçekleştirmesini yaptığımızı ve kaotik haritalar üzerinde önceki kısımda incelediğimiz periyot uzunlukları  $\tau_1 = 9736$  ve  $\tau_2 = 9748$  olan sözde yörüngeleri oluşturduğumuzu varsayalım. Bu sözde yörüngeler arasında transient kesişimi bulunmaktaydı ve bu kesişim sırasıyla 1343. ve 1355. iterasyonlarda gerçekleşmekteydi. Bu nedenle, bu sözde yörüngeleri oluşturan başlangıç şartları için kesişimden itibaren kaotik kriptosistem artık iki farklı kaotik haritayı karşılaştırmayıp aynı kaotik haritayı geciktirilmiş haliyle karşılaştırmaktadır. Bu durum Şekil 3.10’da gösterilmekte olup şekildeki bu eşdeğer yapı kesişimden sonraki bit dizilerini orijinal sistemle aynı şekilde oluşturabilmektedir. Bu eşdeğer yapıyla sadece kesişimden önceki değerler için üretilecek çıkış dizisi aynı şekilde üretilmemektedir.



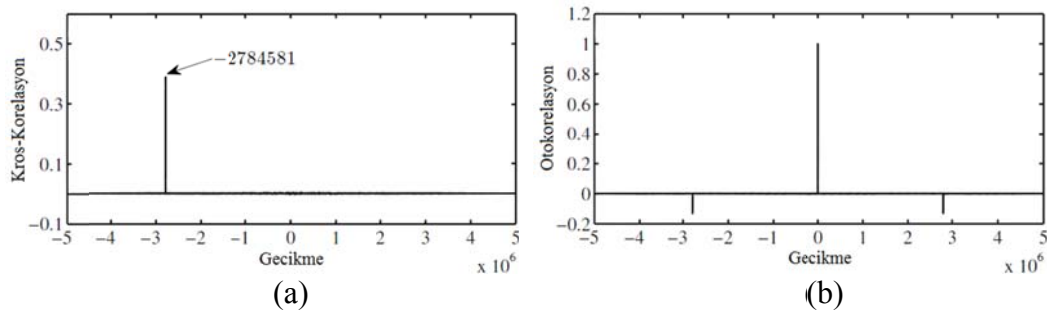
Şekil 3.10. Periyodu aynı olan iki sözde yörünge karşılaştırıldığında eşdeğer yapı.

Sonlu çözünürlük gerçekleştirimleri için sadece sonlu sayıda periyot uzunluğuyla karşılaşılacağı ve bunlar arasından en az biri baskın bir karşılaşılma olasılığına sahip olacağı için çıkışlar sürekli korelasyon gösterecek ve Şekil 3.10'daki eşdeğer yapı geçerli olacaktır. Bu durumun ne kadar dezavantajlı olduğunu göstermek için bu kriptosistemin sabit noktalı aritmetik kullanılarak simülasyonu yapılmış ve tıpkı [34]'de olduğu gibi NIST rastgelelik testleri yapılmıştır. Ancak çözünürlük 56 bite çıkarıldığında önerilen bu sistem tüm NIST testlerini geçebilmiştir. NIST test sonuçları Tablo 3.3'de gösterilmektedir. NIST testi için her biri  $10^6$  bit uzunluğunda olan 1000 farklı sözde yörünge kullanılmıştır.

Tablo 3.3. Eşleştirilmiş kaotik sistemin NIST SP 800-22 rastgelelik testi sonuçları.

| No.      | Test                         | Oran  | P-değeri | Sonuç |
|----------|------------------------------|-------|----------|-------|
| 1        | Frekans                      | 0.985 | 0.616305 | Geçti |
| 2        | Blok frekans                 | 0.992 | 0.729870 | Geçti |
| 3..4     | Kümülatif toplamlar          | 0.985 | 0.501939 | Geçti |
| 5        | Yinelemeler                  | 0.986 | 0.932333 | Geçti |
| 6        | En uzun yinelemeler          | 0.984 | 0.524101 | Geçti |
| 7        | Rank                         | 0.992 | 0.160805 | Geçti |
| 8        | FFT                          | 0.984 | 0.433590 | Geçti |
| 9..156   | Örtüşmeyen şablon eşleştirme | 0.990 | 0.487663 | Geçti |
| 157      | Örtüşen şablon eşleştirme    | 0.990 | 0.715679 | Geçti |
| 158      | Evrensel                     | 0.986 | 0.560545 | Geçti |
| 159      | Yaklaşık entropi             | 0.990 | 0.183547 | Geçti |
| 160..167 | Rastgele sapmalar            | 0.991 | 0.507904 | Geçti |
| 168..185 | Değişken rastgele sapmalar   | 0.991 | 0.460763 | Geçti |
| 186..187 | Seri                         | 0.993 | 0.481654 | Geçti |
| 188      | Lineer komplekslik           | 0.986 | 0.177628 | Geçti |

Şekil 3.3a'daki eksponansiyel artışı göz önüne aldığımızda, bu çözünürlük için periyot uzunlukları pratik hesaplamalara izin vermeyecek raddede uzun olmalıdır. Bu yüzden periyot uzunlukları hakkında herhangi bir ön bilgi olmaksızın doğrudan bu gerçekleştirimin korelasyon ilişkilerine bakılmıştır. Birkaç denemeden sonra Şekil 3.11a'daki gibi korelasyon gösteren sözde periyotlar elde edilmiştir. Bu periyotlar için başlangıç şartları sırasıyla  $x_0 = 0.53079755300897269$  ve  $y_0 = 0.77916723010201117$  olarak alınmıştır. Korelasyonun tepe yaptığı değer -2784581 olup birincinin ikinciye göre 2784581 iterasyon kadar önde olduğu anlamına gelmektedir. Bu başlangıç şartları için üretilecek olan bit dizisinin otokorelasyonu ise Şekil 3.11b'de görüldüğü gibidir. İdeal bir kriptosistemde bu otokorelasyonun  $\delta$  benzeri olması istenirken, buradaki otokorelasyon  $\mp 2784581$  değerlerinde konuşlanmış olan iki adet negatif korelasyon değeriyle idealden ayrılmaktadır.



Şekil 3.11. 56 bit çözünürlüklü gerçekleştirim için a) kaotik haritaların çıkışının kros-korelasyonu, b) çıkış bit dizisinin otokorelasyonu.

Bu kriptosistemin simetrik anahtarlı akış şifreleyici olarak kullanıldığını varsayalım. Kerckhoff prensibine göre bir kriptosistemin tüm gerçekleştirim detaylarının herkes tarafından bilindiği varsayılmalıdır [114]. Bu nedenle de yukarıda bahsedilen sonuçlar şifrelenmiş metinden bilgi sızdırmak isteyen kişilerce de bilinmektedir. Dolayısıyla, şifrelenmiş metinden bilgi sızdırmak isteyen bir kişi korelatif özellik gösteren sözde periyotların varlığından yararlanarak Şekil 3.10'daki eşdeğer yapı ile şifrelemede kullanılan bit dizisinin kesişimden sonrasını tekrar üreterek şifrelenmiş metnin büyük kısmını deşifre edebilir. Tek yapması gereken bu tür periyotların karşılaştırıldığını varsaymak ve aradaki gecikme miktarını bulmaya çalışmaktır. Sadece sonlu sayıda periyot uzunluğuyla karşılaşıldığı ve bunlardan en az birinin baskın bir karşılaşılma olasılığı olduğu göz önüne alındığında, şifrelemede kullanılan bit dizisini tekrar

oluşturmanın o kadar da zor olmadığı görülmektedir. Ayrıca, bu gerçekleştirim için  $2^{112} - 2^{56}$  farklı anahtarın mevcut olduğu düşünüldüğünde, anahtarın ne olduğunu belirlemeye çalışmadan metnin bir kısmının deşifre edilebilmesi kriptografik sistemlerde istenen yayılma özelliğinin amacına ters bir durum yaratmaktadır.

Bu sistemin önerildiği makalede [34] sistemin mükemmel kriptografik özelliklere sahip olduğunun belirtilmesi, başlangıç şartlarına hassas bağımlılık özelliğinin korunduğunun varsayılması ve sistemin tüm NIST rastgelelik testlerinden başarıyla geçmesi yüzündendir. Bu bölümde görmüş olduğumuz üzere sözde kaotik yörüngelerde başlangıç şartlarına hassas bağımlılık sadece yörüngelerin ilk kısımları için geçerlidir. Tablo 3.3'ten görüldüğü üzere de bir sistemin tüm NIST testlerini geçmesi o sistemin mükemmel kriptografik özelliklere sahip olduğu anlamına gelmemektedir. NIST testlerinin bu şekilde yanıltıcı sonuçlar vermesi periyotların çok uzaması yüzündendir. Bu tür testlerde ancak ilk  $10^6$  bit gibi sınırlı sayıda çıkışın testi yapılabildiğinden, bu gibi uzun periyotlu gerçekleştirimlerdeki dezavantajlar yapılan test tarafından yakalanamamaktadır. Bu nedenle, eğer bir kaotik sistem düşük bit çözünürlükleri için istatistiki testleri geçemiyorsa, incelenen bit uzunluğunu değiştirmeden çözünürlüğün artırılması sistemi iyileştirmemekte, sadece testin çıkıştaki eksiklikleri yakalamasını engellemektedir denilebilir. [34]'de yapılan hata da budur. Benzer şekilde 56 bitlik gerçekleştirim için korelasyon bağıntılarını 2784581'den daha düşük gecikmeler için inceleseydik idealde istendiği gibi sıfıra yakın kros-korelasyon ve  $\delta$  benzeri otokorelasyon elde ederdik ki bu sonuçlar da yanıltıcı olurdu.

### 3.5.2. Pertürbasyonlu Eşleştirilmiş Kaotik Sistem

Bu kriptosistem ise Şekil 3.9'daki yapının pertürbasyonlu hali olup [29]'da sunulmuştur. Bu sistem kaotik harita olarak PDKH'yı (Parçalı Doğrusal Kaotik Harita) kullanmakta ve bunları  $\Delta$  zaman aralıklarıyla pertürbe etmektedir. Burada sadece güvenlik açısından üzerinde durulacak olup detaylı incelenmesi ve donanımsal olarak gerçekleştirimi 4. Bölüm'de sunulmaktadır.

Bu kriptosistemin güvenilirliği  $\Delta$  pertürbasyon aralığına bağlıdır. Pertürbasyon aralığı ne kadar uzun olursa bir başlangıç şartı için transient ve periyot kesişimleri de o kadar artacak ve korelasyonlar bu sefer farklı pertürbasyonlar için geçerli olacaktır.

Pertürbasyonun önerildiği ilk çalışmalarda  $\Delta$  zaman aralığının ne olması gerektiğiyle ilgili net bir açıklama bulunmamaktadır [96,97]. [96]'da bu aralığın ortalama periyot uzunluğundan daha kısa tutularak kısa periyotların ortaya çıkmasını engelleyecek uzunlukta seçilmesi önerilmiştir. [97]'de ise bu aralığın sözde yörüngeleri transient bölgesi içerisinde tutacak uzunlukta olması önerilmiştir. Oysa ki pertürbasyon aralığının ortalama periyot uzunluğu civarında tutulması, pertürbasyonlar arasında ciddi korelasyonların oluşmasına neden olacak kadar uzun olacaktır. Bu aralığı yörüngeleri transient bölgesi içerisinde tutacak şekilde seçmek ise çok net bir ifade olmamaktadır. Çünkü Şekil 3.8'de gösterildiği üzere transient uzunlukları çok büyük farklılıklar göstermekte ve ne olacağı önceden tahmin edilememektedir. Ayrıca pertürbasyon aralığını bu şekilde ayarlasak bile bu sefer transient kesişimleri korelasyonlara neden olacaktır. Dolayısıyla korelasyon ilişkilerini en aza indirip sistemi başlangıç şartlarına hassas bağımlı hale getirmek için pertürbasyon aralığı mümkün olduğunca kısa tutulmalıdır denilebilir.

Bu açıdan baktığımızda kaotik haritaları pertürbe etmenin sadece ortalama periyotları arttırmak dışında korelasyon ilişkilerini de azaltarak sistemi güvenli hale getirdiği sonucuna varılabilir. Yine de her pertürbasyon işlemi için elde edilecek sözde kaotik yörüngenin aktif şekilde kullanılan kısmının periyodun kendine göre çok kısa bir parçası olacak olması verimlilik açısından soru işaretleri oluşturmaktadır.

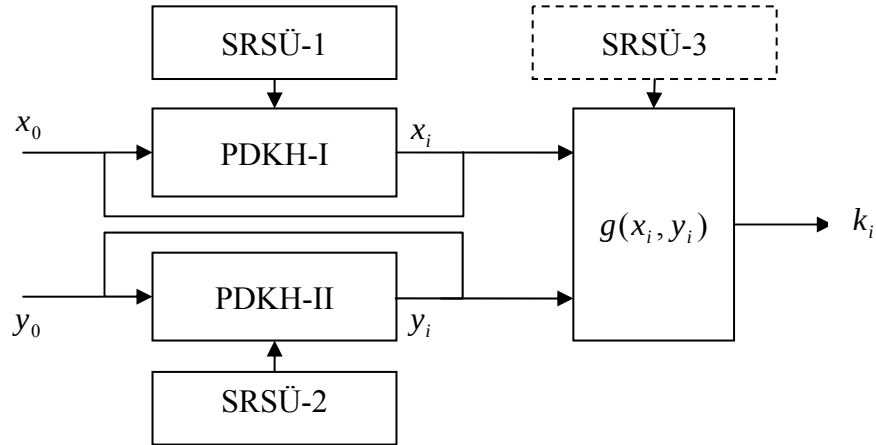
## 4. BÖLÜM

### PERTÜRBASYONA DAYALI KAOTİK SÖZDE RASTGELE SAYI ÜRETECİNİN DONANIM GERÇEKLEŞTİRİMİ VE ŞİFRELEMEDE KULLANIMI

Önceki bölümlerde kaos tabanlı sözde rastgele sayı üreteçlerinde tercih edilmesi gereken kaotik haritanın nasıl özelliklere sahip olması gerektiği, bu tür rastgele sayı üreteçlerinin gerçekleştirimlerinde karşılaşılan problemler ile bunların nasıl çözüleceği ve gerçekleştirimlerde nelere dikkat edilmesi gerektiği gibi hususlar açıklanmıştı. Bu bölümde ise tüm bu sonuçlar göz önünde bulundurularak mevcut kaos tabanlı SRSÜ'ler arasından seçilmiş olan en iyi özelliklere sahip üretcin donanım gerçekleştirimi yapılacak ve görüntü şifrelemede kullanımı gösterilecektir.

#### 4.1. Pertürbasyona Dayalı Kaotik Sözde Rastgele Sayı Üretci

Bu bölümde incelenecek olan rastgele sayı üretci önceki bölümde incelenmiş olan eşleştirilmiş kaotik sistemin pertürbasyona dayalı hali olup [29]'da önerilmiştir. Bu modelde kaotik harita olarak dayanıklı (robust) kaos özelliği gösteren PDKH (Parçalı Doğrusal Kaotik Harita) seçilmiştir. Bu özellikleriyle kaotik sistemin dijital gerçekleştirimlerindeki problemin sistemin güvenilirliğine olan negatif etkileri en aza indirilmeye çalışılmıştır. Eşleştirilmiş kaotik sistem tabanlı SRSÜ'nün blok diyagramı Şekil 4.1'de gösterilmektedir. Buna göre, farklı sözde rastgele sayı üreteçleri ile pertürbe edilen iki PDKH'nın çıkışı  $g(.)$  fonksiyonu ile karşılaştırılarak sözde rastgele bit üretimi gerçekleştirilmektedir.  $g(.)$  karşılaştırma fonksiyonu Denklem (4.1)'deki gibidir.



Şekil 4.1. Pertürbasyona dayalı eşleştirilmiş kaotik sistem [29].

$$g(x_i, y_i) = \begin{cases} 1 & , x_i > y_i \\ Yok & , x_i = y_i \\ 0 & , x_i < y_i \end{cases} \quad (4.1)$$

Kullanılan karşılaştırma fonksiyonunda haritaların çıkışı aynı olduğu zaman çıkışın verilmemesi istenmektedir. Böylelikle haritaların birebir aynı parametrelere sahip olduğu zaman çıkışın 1 veya 0'a takılıp kalması engellenmek istenmektedir. Fakat dijital sistemlerde bu şekilde çıkışın olmaması mümkün değildir. Bunun için eşitlik durumunda çıkışın ne olacağına karar verecek üçüncü bir SRSÜ Şekil 4.1'te görüldüğü gibi kullanılmaktadır. İlk iki SRSÜ ise sadece  $\Delta$  zaman aralıklarıyla kaotik haritaları pertürbe etmek için kullanılmaktadır.

#### 4.1.1. Pertürbasyon İçin Kullanılan SRSÜ'ler

Pertürbasyon için kullanılacak SRSÜ'ler hem gerçekleştirmeyi kolay hem de uzun periyotlara sahip olmalı. Bu bakımdan lineer geribeslemeli kaydırmalı kaydedici (LFSR) ve lineer eşleşik üreteçler (LCG) pertürbasyon için ideal rastgele sayı üreteçleridir. Fakat donanım gerçekleştirmeyi yapacağımız için donanım gerçekleştirmeye uygun olan lineer geribeslemeli kaydırmalı kaydedicilerin kullanılması daha avantajlıdır. Nitekim lineer eşleşik üreteçler yazılım gerçekleştirmeye daha uygundur. Seçilecek lineer geribeslemeli kaydırmalı kaydedicilerin mertebesi sırasıyla  $L_1$  ve  $L_2$  olsun. Bunlar m-serili seçilmeleri gerektiğinden periyotları  $2^{L_1} - 1$  ve  $2^{L_2} - 1$  olacaktır. Aynı başlangıç şartları için pertürbasyonun aynı olmamasını garanti

altına almak için de  $L_1 \neq L_2$  olmalıdır. Buradaki gerçeştirimde  $L_1 = 32$  ve  $L_2 = 33$  alınacaktır. Bu üreteçlerin başlangıç şartlarının sistemin anahtarının bir parçası olarak kullanılacağını da belirlemede fayda var.

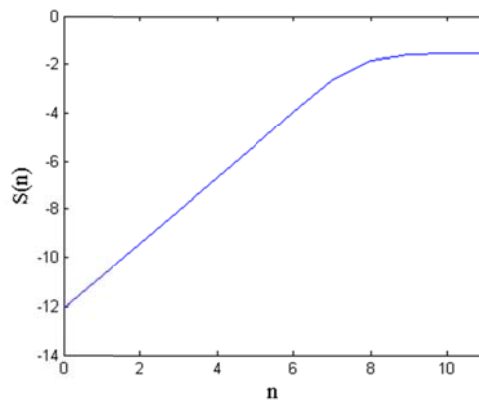
#### 4.1.2. Pertürbasyon Aralığı ve Pertürbe Edilecek Bit Uzunluğu

Kaotik haritalar için pertürbasyon aralığı sırasıyla  $\Delta_1$  ve  $\Delta_2$  olsun. Tıpkı lineer geribeslemeli kaydırmalı kaydedicilerin mertebelerinde olduğu gibi bu değerler de kaotik haritaların aynı başlangıç şartları için aynı çıkışı üretmesini engellemek için farklı ( $\Delta_1 \neq \Delta_2$ ) seçilmelidir. Bunun yanı sıra bu aralıkların transient kesişimlerini minimuma indirmek için mümkün olduğunca küçük tutulması gerektiğinden bahsetmiştik. Bu modelin önerildiği çalışmada alınan  $\Delta_1 = 99$  ve  $\Delta_2 = 101$  değerleri bu bakımdan oldukça uygundur [29].

Pertürbe edilecek bit uzunluğunun ( $l$ ) seçimi için daha önce sinyal gürültü oranının kullanıldığını bahsetmiştik. [29]'da ise farklı olarak bu uzunluğun hesaplanması için Denklem (4.2)'deki bağıntı önerilmiştir:

$$l \geq \lambda \log_2 e \quad (4.2)$$

Burada  $\lambda$  PDKH'nın Lyapunov üsteli olup bu ifadenin tercih edilmesinin nedeni haritanın karakteristik özelliklerini pertürbasyonlarda da sürdürmek içindir. Bunun yanı sıra kullanılan bit çözünürlüğü  $n$  olmak üzere  $l \ll n$  şartı da sağlanmalıdır.



Şekil 4.2. PDKH çıkış serisi için elde edilen TISEAN sonucu.

PDKH'nın Lyapunov üstellerinin hesaplanması için Denklem (2.8)'in kullanılması haritanın parçalı yapısı ve içerdiği bölme ifadeleri yüzünden pratik olmayacaktır. Daha pratik bir çözüm olarak zaman serisi analizi kullanılmıştır. Zaman serisi analiziyle kaotik bir haritanın sadece kısa bir çıkış serisine bakılarak Lyapunov üstelinin hesaplanması mümkündür [115]. Aynı şekilde PDKH'nın Lyapunov üstelini çıkış serisinden hesaplamak için literatürde sıklıkla tercih edilen TISEAN programı kullanılmıştır [116]. Elde edilen sonuç Şekil 4.2'de görülmektedir. Lyapunov üsteli bu şekildeki grafiğin eğimi olup  $\lambda = 1.35642$  şeklinde hesaplanmıştır. Buna göre pertürbe edilecek bit uzunluğu  $l \geq 1.95$  olmalıdır.

#### 4.1.3. Sistemin Periyot Uzunluğu

Her bir kaotik harita için periyot uzunluğu pertürbasyon nedeniyle sırasıyla  $\sigma_1 \Delta_1 (2^{L_1} - 1)$  ve  $\sigma_2 \Delta_2 (2^{L_2} - 1)$  şeklindedir. Sistemin  $k_i$  çıkışının periyodu ise şu şekilde olacaktır:

$$\text{ekok}(\sigma_1 \Delta_1 (2^{L_1} - 1), \sigma_2 \Delta_2 (2^{L_2} - 1)) \quad (4.3)$$

$\Delta_1, \Delta_2$  ile  $L_1, L_2$  aralarında asal seçilirse periyot uzunluğu Denklem (4.4)'teki gibi olacaktır [29]:

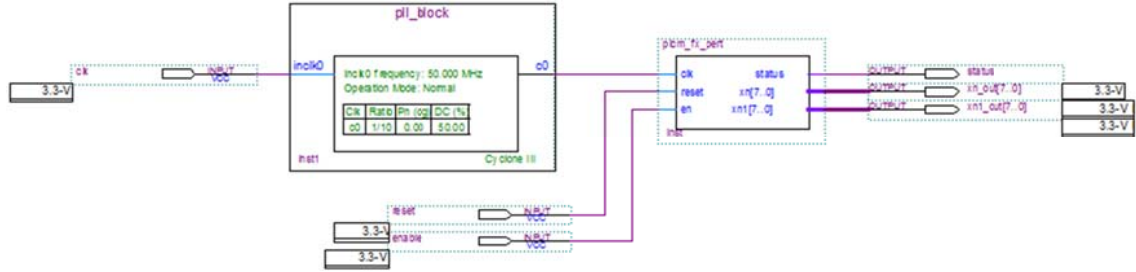
$$\text{ekok}(\sigma_1, \sigma_1) \Delta_1 \Delta_2 (2^{L_1} - 1)(2^{L_2} - 1) \approx \text{ekok}(\sigma_1, \sigma_1) \Delta_1 \Delta_2 2^{L_1 + L_2} \quad (4.4)$$

Burada  $\sigma_1$  ve  $\sigma_2$  değerleri her başlangıç şartı için değişeceğinden dolayı bunları 1 kabul ederek minimum bir periyot uzunluğu hesaplamakta fayda vardır.

#### 4.2. PDKH'nın Donanım Gerçekleştirimi

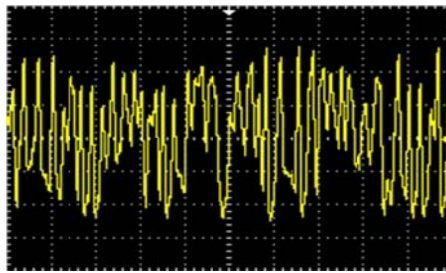
Tek bir kaotik haritanın gerçekleştirimi çıkış verisinin maksimum hızının belirlenmesi açısından oldukça önemlidir. Donanım gerçekleştirimi için yeniden programlanabilme özelliğiyle tasarım esnekliği ve hızlı prototiplendirme gibi özellikleri sunan FPGA (Field Programmable Gate Array) tercih edilmiştir. FPGA platformu olarak da Altera DE0 bordu üzerinde bulunan Cyclone III EP3C16F484C6 entegresi kullanılmıştır. FPGA'in programlanması içinse VHDL (Very High Speed Integrated Circuits

Hardware Description Language) dilinden faydalanılmıştır. Programlama için kurulan devre düzeneğinin blok diyagramı Şekil 4.3’de görülmektedir.

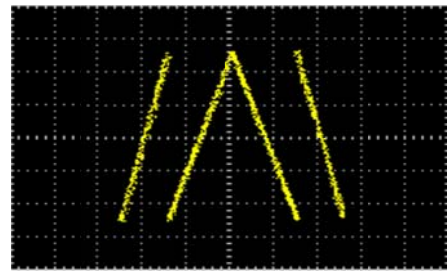


Şekil 4.3. PDKH’nın FPGA program yapısının blok gösterimi.

Gerçekleştirim için 32 bit sabit noktalı aritmetik kullanılmıştır. Pertürbasyon içinse mertebesi 32 ve tablaları sırasıyla 32, 30, 26, 25 olan m-serili lineer geribeslemeli kaydırmalı kaydediciden faydalanılmıştır. Pertürbasyon aralığı  $\Delta = 99$  ve pertürbe edilecek bit uzunluğu  $l = 4$  bit olarak seçilmiştir. PDKH’nın kontrol parametresi ve başlangıç değeri 0.19 olarak alınmıştır. Şekil 4.3’den de görülebileceği üzere haritanın çıkışının 8 bitlik kısmının hem kendisi hem de 1 birim geciktirilmiş hali FPGA’in çıkış pinlerine verilmektedir. Daha sonra bu çıkışlar alınarak 8 bitlik iki farklı dijital analog dönüştürücüye (D/A) verilmek suretiyle osiloskop üzerinde gözlem yapılmıştır. Zaman ve faz domeninde görülen çıkışlar Şekil 4.4’de verilmiştir. Zaman domeninde sadece geciktirilmemiş çıkış gösterilirken faz domeni ise hem geciktirilmiş çıkış hem de geciktirilmemiş çıkışa göre çizdirilmektedir. Görüldüğü üzere çıkışlar önceki bölümlerde simülasyonunu yapmış olduğumuz PDKH’nın çıkışıyla uyuşmaktadır. FPGA’in programlanması için kullanılan Quartus II editörünün sentez sonuçlarına göre bir tane PDKH’nın pertürbasyonlu gerçekleştirimi için 15408 lojik element içerisinde



(a)

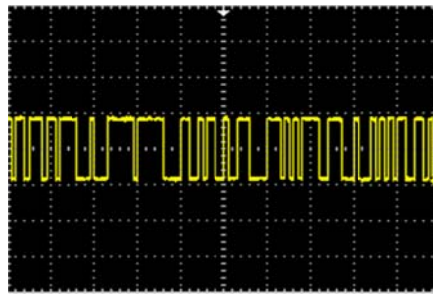


(b)

Şekil 4.4. PDKH’nın donanım gerçekleştiriminin a) zaman domenindeki, b) faz domenindeki osiloskop görüntüleri (Volt/div: 1V; Time/div: 2.5  $\mu$ s).

sadece 2823'ü (%18) kullanılmış ve sistemin maksimum çalışma frekansı 5.6 MHz olarak hesaplanmıştır. Model bu frekansın üstüne çıktıkça çıkış bozulacak ve hatalı hale gelecektir.

Bu çalışma frekansının bu kadar düşük olmasının nedeni PDKH yapısında bulunan bölme işlemidir. Sabit noktalı aritmetiğin kullanıldığı işlemlerde çarpma işlemi gömülü çarpıcılarla hızlandırıldığı için genelde bölme işlemi pek tercih edilmez. Bunun bir diğer nedeni bölme sonucu elde edilecek bit uzunluğunun sayıların tamsayı ve kesirli kısımlarının farkından oluşması ve bu işlemin bit uzunluğu içerisindeki ölçeklendirmeyi değiştirmesidir. Bu durumu Q notasyonu ile örnekleyelim. Q notasyonu sabit noktalı sayıların tamsayı ve kesirli kısımlarının bit uzunluğunu ifade etmek için kullanılır. QT.K ifadesinde T, tamsayı bit uzunluğunu ve K ise kesirli kısmın bit uzunluğunu verir. Buna göre Q18.12 sayısını Q10.2 sayısına bölersek bölüm sonucu elde edilecek sayı Q8.10 olacaktır. Buradaki tasarımda ise 32 bit sabit noktalı aritmetik kullanılırken bunlardan sadece 1 bit tamsayı ifade etmek için kullanılmış, geri kalan 31 bit kesirli kısmı ifade etmek için kullanılmıştır (Q1.31). İki tane Q1.31 sayısını böldüğümüzde ise bölüm Q32.0' a dönüştürülecek ve sonuç hatalı olacaktır. Bunu engellemek için bölen ifade bölümden önce Q2.62'ye dönüştürülmektedir. Bu da işlemler arasında ekstra zaman harcanmasına neden olmaktadır. Tüm bu açıdan baktığımızda bölme işleminin mevcudiyetinin bu model için dezavantaj yarattığı görülmektedir.

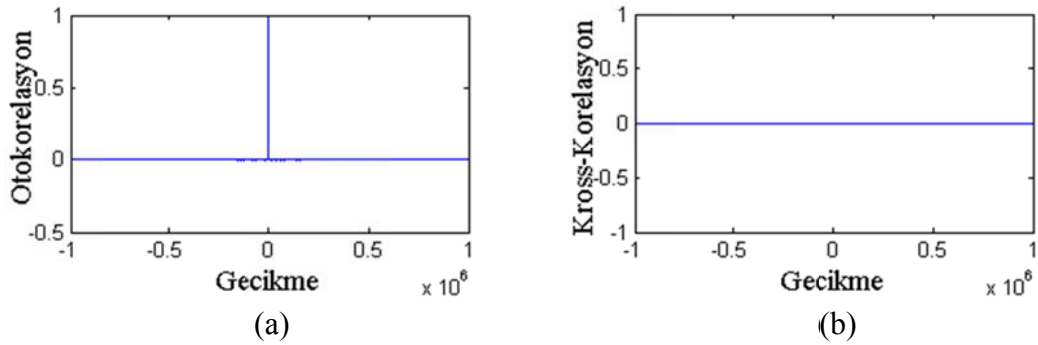


Şekil 4.5. Eşleştirilmiş kaotik sistemin çıkışının osiloskop görüntüsü (Volt/div: 2V; Time/div: 100  $\mu$ s).

#### 4.3. Eşleştirilmiş Kaotik Sistemin Donanım Gerçekleştirimi

Eşleştirilmiş kaotik sistemin gerçekleştirimi için önceki kısımdaki PDKH alınmış ve pertürbasyon aralığı  $\Delta = 101$  ve pertürbe edilen bit uzunluğu yine  $l = 4$  bit olan bir başka PDKH ile Denklem (4.1)'deki fonksiyonun VHDL gerçekleştirimi kullanılarak

model elde edilmiştir. İkinci haritanın pertürbasyon için kullanılan lineer geribeslemeli kaydırmalı kaydedicisinin mertebesi tabloları sırasıyla 33, 32, 29, 27 olmak üzere 33 olarak alınmış ve eşitlik durumunda çıkışın belirlenmesi için mertebesi yine 33 olan üçüncü bir lineer geribeslemeli kaydedici kullanılmıştır. FPGA gerçekleştirimi sonucu çıkışta elde edilen  $k_i$  bit dizisinin osiloskopla elde edilmiş görüntüsü Şekil 4.5'te verilmiştir. Yüksek frekans yüzünden osiloskop görüntüsünde glitch oluşmaması için görüntü alınırken çalışma frekansı düşük tutulmuştur.  $k_i$  çıkış serisinin  $10^6$  bit için elde edilmiş olan kross-korelasyon ve otokorelasyon ifadeleri ise Şekil 4.6'da verilmiştir. Pertürbasyon aralığı küçük tutulduğu için herhangi bir korelasyon ilişkisi beklenmemektedir. Bu bakımdan sistem oldukça güvenlidir denilebilir.



Şekil 4.6.  $k_i$  çıkış serisi için elde edilmiş, a) otokorelasyon, b) kros-korelasyon bağıntıları.

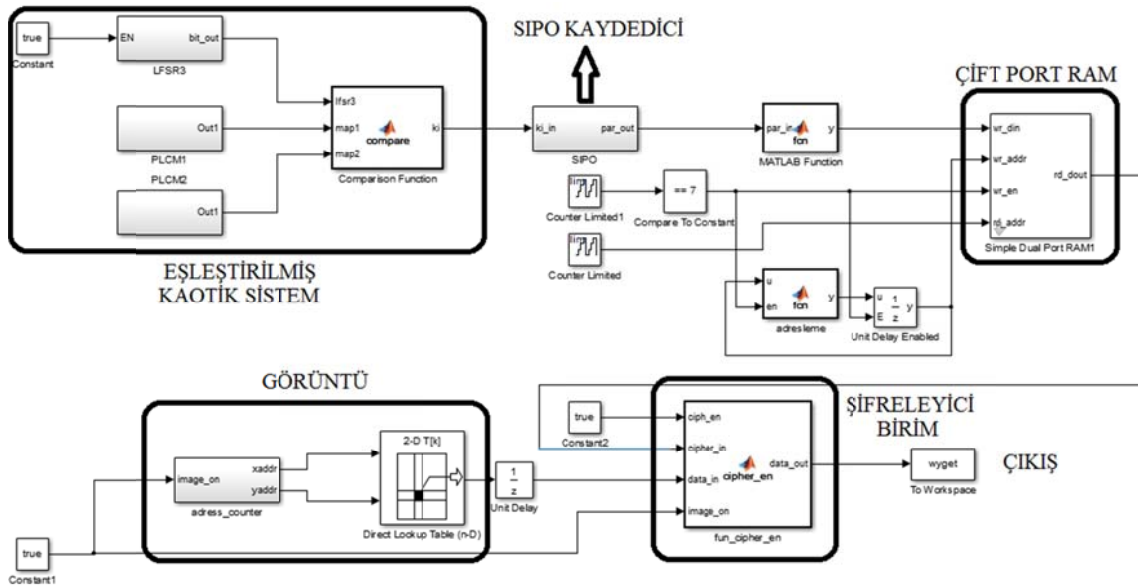
Sentezleme sonucu elde edilen verilere göre mevcut kaynaklar arasından 5741 adet lojik element (%37) kullanılmıştır ki bu beklenildiği üzere tek bir haritanın gerçekleştiriminin iki katından biraz daha fazladır. Maksimum çalışma frekansı ise 5.13 MHz olarak hesaplanmış olup biraz daha düşmüştür. Bu çalışma frekansı için çıkış verisi yaklaşık 5 Mbps hızında olup çıkışın minimum periyodu Denklem (4.4) kullanılarak  $3.69 \times 10^{-23}$  şeklinde hesaplanmaktadır.

#### 4.4. Eşleştirilmiş Kaotik Sistemin Görüntü Şifrelemede Kullanımı

Eşleştirilmiş kaotik sistem bu haliyle aynı zamanda bir akış şifreleyicidir. Dolayısıyla simetrik anahtarlı şifreleme uygulamalarında doğrudan kullanılabilir. Anahtar uzunluğu PDKH'nın başlangıç şartları ve  $p$  kontrol parametreleri ile pertürbasyon için kullanılan lineer geribeslemeli kaydırmalı kaydedicilerin başlangıç şartlarından oluşmaktadır. Bu

bakımdan toplam anahtar uzunluğu maksimum 194 bite kadar çıkmaktadır. Bu pek çok modern uygulama için yeterli bir uzunluktur. Önceki bölümlerde ifade etmiş olduğumuz üzere simetrik şifrelemede bu anahtar hem alıcı hem de gönderici tarafından bilinmeli ve başkasıyla paylaşılmamalıdır.

Görüntü şifreleme uygulaması için  $k_i$  çıkış serisi 256x256'lık gri tonlamalı bir görüntünün piksel değerlerini XOR işlemiyle şifrelemek için kullanılacaktır. Tüm gerçekleştirim işlemleri yine FPGA platformu üzerinde yapılacak olup görüntü ve şifrelenmiş hali bir RGB monitör aracılığıyla gözlemlenecektir. Ekran görüntüsü için çözünürlüğü 1024x768, yatay tarama frekansı 48.36 kHz, dikey tarama frekansı 60 Hz olan XGA standardı kullanılacaktır. Bu standart için görüntüler 8 bitlik paketler halinde 65 MHz'lık bir frekansla monitöre aktarılmalıdır. Bu ise gri tonlamalı görüntüler için 520 Mbps'lık ve renkli görüntüler için 1.56 Gbps'lık veri çıkışı anlamına gelmektedir. Oysa ki 5 Mbps'lık veri çıkış hızıyla eşleştirilmiş kaotik sistemin bu verileri eş zamanlı şifrelemesi mümkün değildir. Dolayısıyla bu kriptosistemi kullanarak görüntü şifrelemek için 65536x8 bitlik çift portlu bir RAM (Random Access Memory) üretilecek rastgele sayılar için tampon bellek görevi görecektir. Bu yapının Matlab Simulink modeli Şekil 4.7'de gösterilmektedir.



Şekil 4.7. Çift port RAM'in kullanıldığı şifreleme mekanizmasının Simulink modeli.

Şekil 4.7'deki model kullanılarak tampon belleğin tüm değerleri sıfırlandıktan sonra bir SIPO (Serial In Parallel Out) kaydedici (register) ile 8 bitlik paralel verilere dönüştürülen sözde rastgele sayılar her 8 saat çevriminde bir, baştan itibaren bu RAM'in adres satırlarına yazılır. RAM çift portlu olduğu için yazma işleminden bağımsız olarak içindeki veriler de okunabilmekte ve şifrelemede kullanılabilir. Yazma işleminin her 8 saat çevriminde bir yapılmasının nedeni ise SIPO kaydediciden alınan verilerin birbirinin kaydırılmış halleri olmasını engellemek içindir. Bu kısıtlama nedeniyle sistem çalıştırıldıktan ancak 65536x8 saat çevriminden sonra tam anlamıyla 256x256'lık bir görüntüyü şifreleyebilecek düzeye erişecektir. Bu ise 5 MHz çalışma frekansı olan kriptosistem için 0.105 saniye geçmesi demektir. Bu esnada iletilen veri 65 MHz'lik bir frekansa tabi olacağı için yaklaşık 52 Mbit'lik verinin tam olarak şifrelenmeden iletileceği anlamına gelmektedir. Bu 0.105 saniyelik dilim içerisinde yapılacak şifrelemenin çıkışta zamanla nasıl değişeceği Şekil 4.8'de görülmektedir.



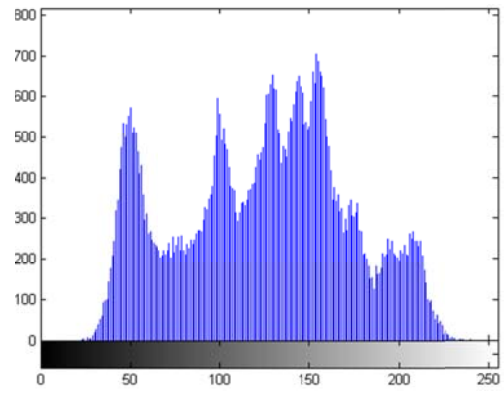
Şekil 4.8. Sistem başlar başlamaz yapılacak şifrelemede a) 0.03 saniye, b) 0.05 saniye, c) 0.09 saniye sonraki çıkış görüntüleri.

Dolayısıyla bu model ile yapılacak şifrelemelerde sistemi çalıştırıp tampon belleğe yazmayı başlatacak bir aktivasyon sinyalinin yanında şifrelemeyi daha sonra ayrıca başlatacak bir başka aktivasyon sinyalinin bulunması gerekmektedir. Şekil 4.7'deki şifreleyici birim bu aktivasyon sinyallerine bağlı olarak RAM'den okunan veriler ile görüntü verilerini XOR işlemi ile şifrelemekte veya şifreleme için aktivasyon sinyali gelmiyorsa görüntü verilerini monitöre olduğu gibi iletmektedir. Bu model kullanılarak yapılan şifreleme sonucu elde edilecek şifrelenmiş görüntü ve orijinal görüntü ile şifrelenmiş görüntünün histogramları Şekil 4.9'da gösterilmektedir. Buna göre görüntü tam olarak şifrelendiğinde şifrelenmiş görüntü orijinal görüntüyü tamamen

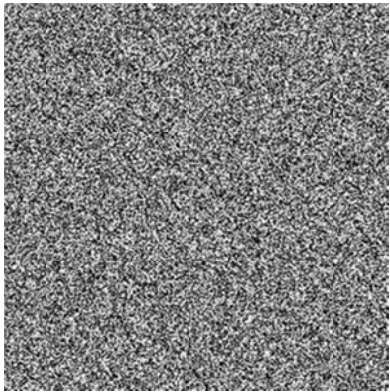
maskelemekte ve histogramı da orijinal görüntüden farklı olarak tekdüze dağılım göstermektedir. Bu sonuçlar görüntü şifrelemede tam olarak istenen bir durumdur. Bu sayede şifrelenmiş veri şifrelenen görüntü hakkında hiçbir ipucu vermemektedir. Rastgele sayı üreteçlerindeki tekdüze dağılım işte bu nedenle oldukça önem arz etmektedir. Eğer şifrelemede kullanılan rastgele sayılar tekdüze dağılım göstermeseydi orijinal görüntünün bazı kısımları maskelenemeyecekti ve histogramı analiziyle orijinal görüntü hakkında fikir sahibi olunabilecekti. Fakat tekdüze bir dağılımda bu duruma imkan tanınmamaktadır.



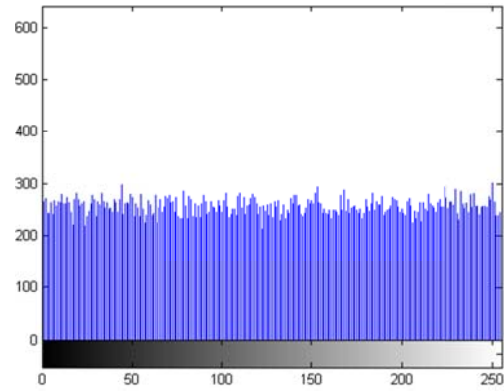
(a)



(b)



(c)



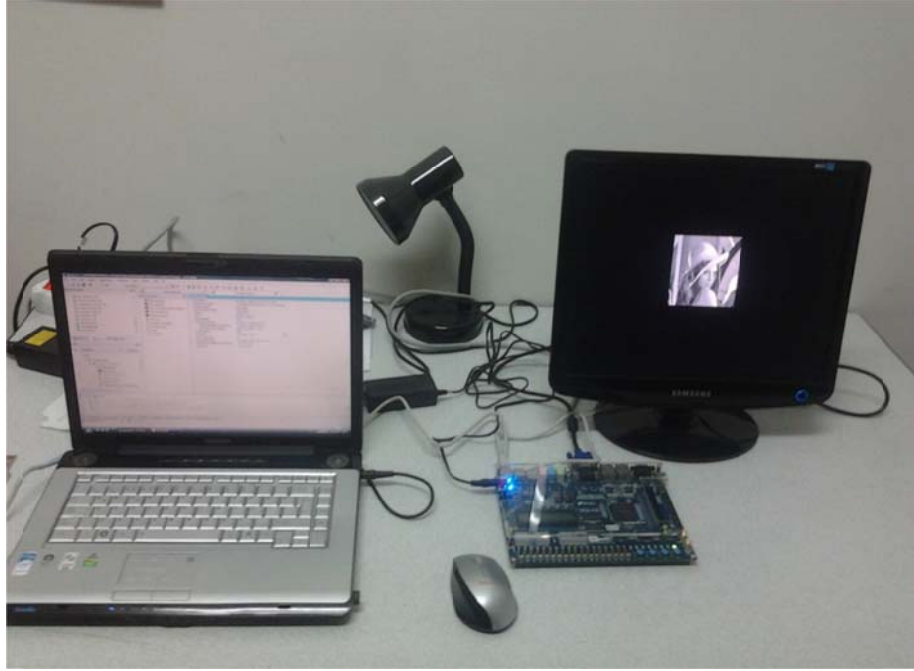
(d)

Şekil 4.9. a) Orijinal görüntü, b) orijinal görüntünün histogramı, c) şifrelenmiş görüntü, d) şifrelenmiş görüntünün histogramı.

#### 4.5. Görüntü Şifreleme Uygulamasının Donanımsal Olarak Gerçekleştirimi

Çift port RAM'in kullanıldığı şifreleme mekanizmasının donanımsal olarak gerçekleştirilmesi için önceki kısımlarda olduğu gibi yine FPGA platformu tercih edilmiştir. Fakat önceki kısımlarda kullanılan Altera DE0 bordu üzerindeki Cyclone III bu tasarım için yetersiz kaldığından dolayı Altera DE2 bordu üzerindeki Cyclone IV

EP4CE115F29C7 FPGA çipi kullanılmıştır. Önceki kısımda anlatılan şifreleme mekanizmasına ek olarak RGB monitörde XGA standardında görüntü elde etmek için ayrıca bir VGA kontrolörü tasarlanmış olup yine FPGA üzerinde gerçekleştirilmiştir. RGB monitöre sinyal göndermek için Altera DE2 boardu üzerinde bulunan VGA konektörü kullanılmıştır. Deney düzeneği Şekil 4.10’da görüldüğü gibidir.



Şekil 4.10. Görüntü şifreleme uygulaması için deney düzeneği.

Sistemi ve şifrelemeyi başlatmak için kullanılacak olan iki aktivasyon sinyali Altera DE2 boardu üzerinde bulunan anahtarlara atanmış olup sistem bunlarla kontrol edilmektedir. Şifreleme yokken monitör görüntüsü Şekil 4.11a’daki gibidir. Şifreleme yapıldığında ise ekran görüntüsü Şekil 4.11b’deki gibi olmaktadır. Şifreleme aktifken tampon bellek akış şifreleyicinin çıkışı ile sürekli güncellendiğinden monitördeki şifreli görüntü de sürekli değişmektedir. Akış şifreleyici için 5 MHz ve VGA kontrolü ile şifreleme uygulaması için 65 MHz olmak üzere iki farklı saat frekansı kullanılmıştır. Bu frekansları üretmek için FPGA çipi içerisindeki PLL (Phase Locked Loop) bloğundan faydalanılmıştır. Sentezleme sonrası elde edilen sonuçlara göre tüm uygulama için 44,316 lojik element kullanılmıştır. Bu ise toplam 114,480 mevcut lojik elementin %39’una karşılık gelmektedir. Bu sayının akış şifreleyici için gereken lojik element sayısının çok üstünde olmasının nedeni VGA kontrolü için sentezlenen diğer sistem

parçalarının olmasındandır. Uygulama için 524,288 hafıza biti kullanılmış olup bu ise mevcut hafıza kaynaklarının %13'üdür.



(a)



(b)

Şekil 4.11. Donanım gerçekleştirimi sonucu elde edilen a) şifreleme yokkenki ekran görüntüsü, b) şifreleme aktifkenki ekran görüntüsü.

## 5. BÖLÜM

### SÜREKLİ ZAMANLI KAOTİK SİSTEMLERE DAYALI KARIŞIK MODLU SÖZDE RASTGELE SAYI ÜRETECİ TASARIMI VE UYGULAMASI

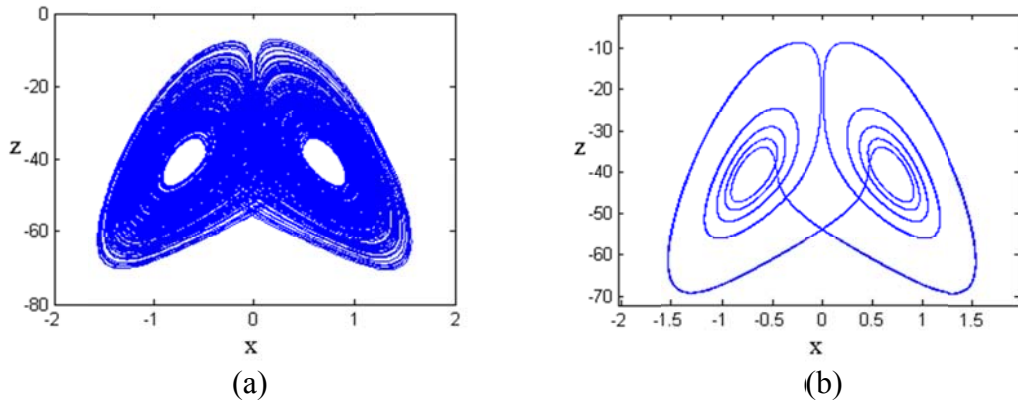
Önceki bölümlerde ifade etmiş olduğumuz üzere kaotik haritalar ayrık zamanlı yapıları nedeniyle dijital gerçekleştirmelere oldukça uygundu. Bunun yanı sıra, tek boyutta dahi kaotik davranış gösterebildiklerinden gerçekleştirmeleri de oldukça basitti. Tüm bu özellikleri göz önüne aldığımızda doğal olarak sözde rastgele sayı üretiminde kaotik harita kullanmanın oldukça avantajlı olacağı düşünülmektedir. Fakat görmüş olduğumuz üzere pratikte pek çok sorunla karşılaşmakta; iyi bir sözde rastgele sayı üreticinin taşınması gereken özellikleri sağlamak adına tasarımlar karmaşılaşmakta ve çıkış hızları düşük olmaktadır. Bu bakımdan bu bölümde, çok boyutlu yapılarıyla daha farklı dinamiklere imkan tanıyabilen sürekli zamanlı kaotik sistemlerin ayrıklaştırılmasıyla elde edilecek sözde rastgele sayı üreticilerinin tasarımları üzerinde durulacak ve kaotik haritaların bu tür üreticilere göre gerçekten de daha avantajlı olup olmadıkları donanım gerçekleştirmesi ve performans açısından karşılaştırmaları yapılmak suretiyle incelenecektir.

#### 5.1. Sürekli Zamanlı Kaotik Sistemlerin Ayrıklaştırılması

Sürekli zamanlı kaotik sistemler kaotik haritaların aksine diferansiyel denklemlerle ifade edilirler. Dolayısıyla kaotik haritaların dijital gerçekleştirmeleri sadece uzaysal ayrıklaştırmaya tabi tutulurken, sürekli zamanlı kaotik sistemlerin dijital gerçekleştirmeleri hem zamansal hem de uzaysal ayrıklaştırma gerektirir. Uzaysal ayrıklaştırmanın kaosu ortadan kaldırması, kısa ve aynı periyotlara neden olması gibi

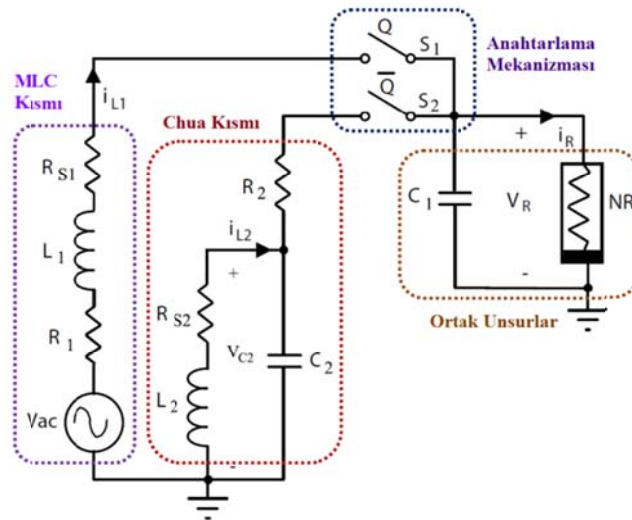
olaylar bu tür sistemlerin gerçekleştiriminde de mevcutken, bunun yanı sıra zamansal ayrıklaştırma olarak kullanılacak numerik integrasyonun getirmiş olduğu hatalarla da baş edilmek zorunda kalınmaktadır. Bu duruma verilebilecek en iyi örnek [117]'deki çalışmada ele alınmıştır. Bu çalışmada ele alınan sürekli zamanlı kaotik sistemin dijital ortamda 4. dereceden Runge-Kutta integrasyon metodu kullanılarak gerçekleştirimleri yapılmış ve farklı sonuçlar elde edilmiştir. Bit çözünürlüğü 32 bit olarak alındığında çift bantlı çeker gözlemlenirken, bit çözünürlüğü 80 bite çıkarıldığında çeker ortadan kaybolmuş ve kısa periyotlu bir sonuç elde edilmiştir. Oysaki teorik olarak bu kaotik sistemin çeker tek bantlı olup bu sonuçların hiçbirisi doğru sonucu vermemektedir. Bu durum 3. Bölüm'de bahsetmiş olduğumuz gölgeleme lemmasının geçerliliğini de şüpheye düşürmektedir. Dolayısıyla sürekli zamanlı kaotik sistemlerin numerik integrasyonunun sistemin orijinal davranışını yansıtmaması, karşılaşılan ilk sorundur denilebilir. [117]'de bu sorunun çözümü için numerik integrasyonu yapılan sisteme kontrollü bir şekilde tekdüze dağılımlı gürültü injekte edilmesi önerilmiştir. Bu kaotik haritalara uygulanan pertürbasyonla aynı şekilde bir uygulamadır.

Karşılaşılan diğer sorun ise sonlu çözünürlük yüzünden tıpkı kaotik haritalarda olduğu gibi periyodik yörüngelerin elde edilmesidir. Literatürde sürekli zamanlı kaotik sistemler için bu durum detaylıca ele alınmamıştır. Sürekli zamanlı kaotik sistemleri kullanan az sayıdaki sözde rastgele sayı üreteçleri numerik simülasyon sonuçlarını ispat olarak verip numerik integrasyon ile doğrudan gerçekleştirim yoluna gitmiştir [118-120].



Şekil 5.1. Denklem (5.8) ile verilen kaotik sistemin a) İlk 100000 adım için simülasyonu, b) 75000 adımdan sonraki simülasyonu.

Sürekli zamanlı kaotik sistemler için transient veya periyot uzunluklarının 3. Bölüm'deki algoritma ile hesaplanması çok boyutlu olmaları nedeniyle hem daha fazla işlem gerektirmekte hem de sözde yörüngeler daha uzun olacağından hesaplama süresi pratik olmayacak kadar uzun olmaktadır. Bu nedenle periyotların oluşma süreci simülasyon yapıp faz portresinin farklı değerler için incelenmesiyle ele alınmıştır. Şekil 5.1a'da Denklem (5.8) ile ifade edilen kaotik sistemin, başlangıç anından itibaren 100000 adım boyunca 0.01 adım aralığıyla Euler integrasyon metodu kullanılarak yapılmış simülasyon sonucu görülmektedir. Bu şekil oldukça kaotik görünse de yanıltıcıdır. Eğer bu grafiği 75000'ninci adımdan sonra çizdirirsek Şekil 5.1b'deki gibi periyodik bir sonuç elde ederiz. Yani kaotik haritalarda transient cevaptan sonra nasıl periyodik sonuç elde ediliyorsa, sürekli zamanlı kaotik sistemin ayrıklaştırılması sonucu bir süre sonra kararlı periyotlar elde edilmektedir. Bu kararlı periyotların ne kadar uzun olacağı ve ne zaman oluşacağı tıpkı kaotik haritalarda olduğu gibi şu an için hesaplanamamaktadır. Fakat yapılan denemelerde Runge-Kutta gibi daha hassas integrasyon metodları kullanılınca bu değerlerin oldukça arttığı gözlemlenmiştir. Sözde yörüngelerin yuvarlama hataları nedeniyle kararlı hale gelmesi hakkında kapsamlı bir çalışma [121]'de ele alınmıştır. Aynı çalışmada bu durumun giderilmesi için tıpkı [117]'de olduğu gibi sisteme gürültü eklenmesi önerilmiştir ki bu da yine pertürbasyondur. Bu bölümdeki sözde rastgele sayı üretimi tasarımında ise pertürbasyon metodu kullanılmayacaktır. Onun yerine anahtarlamaya dayanan karışık modlu kaotik sistemler yardımıyla yukarıdaki problemler giderilmeye çalışılacaktır.



Şekil 5.2 Karışık modlu kaotik devre.

## 5.2. Karışık Modlu Kaotik Sistem

Karışık modlu kaotik sistem ilk olarak [122]'de önerilmiştir. Bu sistem analog elemanlarla gerçekleştirilmiş olup Chua ve MLC devrelerinin anahtarlanması prensibine dayalı olarak çalışmaktadır. Bu anahtarlama sayesinde çıkış sırasıyla otonom ve otonom olmayan özellik göstererek güvenli haberleşmede kullanılmak üzere daha kompleks bir hal almaktadır. Karışık modlu kaotik sistemin analog devre gerçekleştirimi Şekil 5.2'de görülmektedir. Bu yapı içerisindeki Chua sistemi aşağıdaki denklemlerle ifade edilmektedir:

$$\begin{aligned}\frac{dx}{dt} &= \alpha(y - x - f(x)) \\ \frac{dy}{dt} &= x - y + z \\ \frac{dz}{dt} &= -\beta y\end{aligned}\tag{5.1}$$

$f(x)$  fonksiyonu ise Denklem (5.2)'deki gibi alınmaktadır.

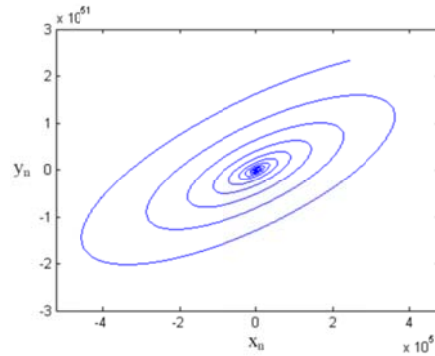
$$f(x) = m_0 x + 0.5(m_1 - m_0) \left[ |x + B_{p1}| - |x - B_{p1}| \right]\tag{5.2}$$

$m_0 = -0.68$ ,  $m_1 = -1.27$ ,  $B_{p1} = 1$ ,  $\alpha = 10$  ve  $\beta = 14.87$  değerleri için bu sistem kaotik davranış göstermektedir. MLC sistemi ise Denklem (5.3) ile ifade edilmektedir.

$$\begin{aligned}\frac{dx}{dt} &= y - f(x) \\ \frac{dy}{dt} &= -\beta x - \beta y + F \sin(\omega t)\end{aligned}\tag{5.3}$$

Bu sistem için  $f(x)$  nonlineerliği Denklem (5.2) ile aynı olup  $\beta=1$ ,  $F=0.25$ ,  $m_0 = -0.68$ ,  $m_1 = -1.27$ ,  $B_{p1} = 1$  ve  $\omega=0.6$  değerleri için kaotik davranış sergilemektedir. Bu iki yapıyı kullanarak karışık modlu kaotik devrenin dijital gerçekleştirimini yaparsak her anahtarlama diğer sistemin etkin olması sağlanacağından pertürbasyon olmaksızın bu iki sistemin de kararlı yörüngeler oluşturmasını önlemiş oluruz. Fakat bunun için bir sistemin son değerinin diğerinin başlangıç şartı olarak kullanılması gerekmektedir. Aksi takdirde her anahtarlama aynı değerleri baştan itibaren tekrar ve tekrar elde etmiş oluruz. Karışık modlu devrede kullanılan Chua sistemi bu tür bir gerçekleştirim için

uygun değildir. Çünkü Chua sistemi sadece sınırlı sayıdaki başlangıç şartları için kaotik davranış sergilerken çoğu başlangıç şartı için sadece sonsuza ıraksamaktadır. Hangi başlangıç şartları için sonsuza ıraksama olacağı ve hangileri için kaotik davranış sergileyeceği net sınırlarla belirlenmemiştir [123]. Chua sistemine ait  $(x,y,z) = (3.5, -3.5, 0)$  başlangıç şartı için spiral yaparak sonsuza ıraksayan bir yörünge Şekil 5.3’de görülmektedir. Bu nedenle, MLC devresinin farklı değerleri Chua sisteminin başlangıç şartı olarak kullanıldığında sistemin kaostan çıkması ve periyodik davranış göstermesi mümkündür. Dolayısıyla sözde rastgele sayı üretiminde karışık modlu kullanılmak üzere bir başka kaotik sistem seçilmelidir. Disipatif hacim değişimi sağlayan ve tüm başlangıç şartları için yakalama bölgesine (trapping region) sahip olan Lorenz sistemi bu iş için idealdir. Yakalama bölgesi matematiksel olarak tanımlanmış bir bölgedir [124]. Buna göre bu bölgede başlayan tüm başlangıç şartları bu bölgede sınırlı kalır. Lorenz sistemi bu özellikleriyle hemen hemen tüm başlangıç şartları için belli parametre değerlerinde kaotik davranış sergileyebilir.



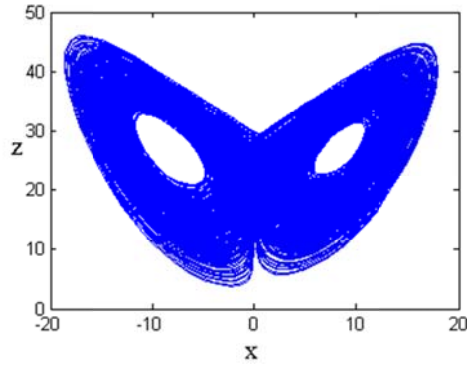
Şekil 5.3. Chua sistemine ait  $(x,y,z) = (3.5, -3.5, 0)$  başlangıç şartı için elde edilen yörünge.

### 5.3. Lorenz Ailesinin Karışık Modlu Olarak Kullanılması

Lorenz sistemi 1963 yılında Edward Lorenz tarafından ortaya konmuş olup aşağıdaki eşitliklerle ifade edilmektedir:

$$\begin{aligned} \frac{dx}{dt} &= -\sigma(x - y) \\ \frac{dy}{dt} &= Rx - y - xz \\ \frac{dz}{dt} &= xy - bz \end{aligned} \quad (5.4)$$

Sistem  $\sigma = 10$ ,  $R = 28$ ,  $b = 8/3$  değerleri için kaotik davranış göstermekte olup kaotik çekeri Şekil 5.4'teki gibidir. 1996 yılında Vanecek ve Celikovsky bu sistemi Denklem (5.5)'teki gibi lineer ve nonlinear parçalara bölünmüş bir şekilde genelleştirilmiştir. Bu genelleştirilmiş ifade için  $a_{12} \cdot a_{21} > 0$  şartını sağlayan her ifade Lorenz benzeri bir kaotik çeker üretmektedir [125].



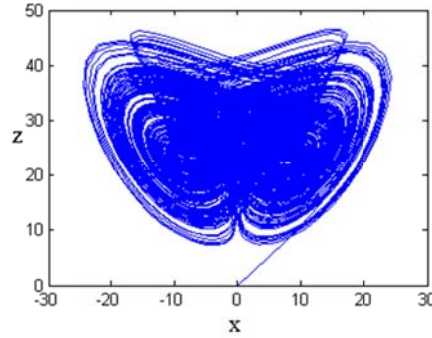
Şekil 5.4. Lorenz sisteminin kaotik çekeri.

$$\begin{bmatrix} dx/dt \\ dy/dt \\ dz/dt \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + x \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (5.5)$$

1999 yılında ise Chen, Denklem (5.6) ile tanımlanan yeni bir kaotik çeker buldu [126]. Bu sistemin denklem takımları Lorenz sistemine oldukça benzese de elde edilen kaotik çeker Şekil 5.5'te görüldüğü gibi olup topolojik olarak Lorenz sisteminden farklıdır [127]. Kaotik parametre değerleri  $a = 35$ ,  $b = 3$  ve  $c = 28$  şeklinde alınmaktadır.

$$\begin{aligned} \frac{dx}{dt} &= a(y - x) \\ \frac{dy}{dt} &= (c - a)x + cy - xz \\ \frac{dz}{dt} &= xy - bz \end{aligned} \quad (5.6)$$

Chen sistemi, Denklem (5.5) ile verilen kanonik ifadeye uyarlandığında Chen sisteminin  $a_{12} \cdot a_{21} < 0$  şeklinde Lorenz ailesinde yeni bir sınıfa dahil olduğunu görmekteyiz.

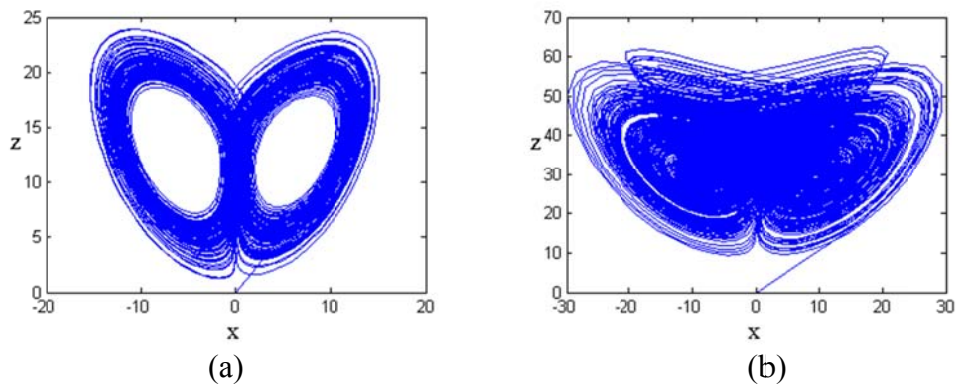


Şekil 5.5. Chen sisteminin kaotik çekeri.

Bu da akla  $a_{12} \cdot a_{21} = 0$  şartını sağlayan bu iki sınıf arasında bir geçiş formu olması gerektiğini akla getirmektedir. Nitekim 2002 yılında Jinhu Lü bu şartı sağlayan yeni bir kaotik sistem geliştirmiştir [128]. Bu sistemin diferansiyel denklem takımları Denklem (5.7)'de görüldüğü gibidir.

$$\begin{aligned} \frac{dx}{dt} &= a(y - x) \\ \frac{dy}{dt} &= cy - xz \\ \frac{dz}{dt} &= xy - bz \end{aligned} \quad (5.7)$$

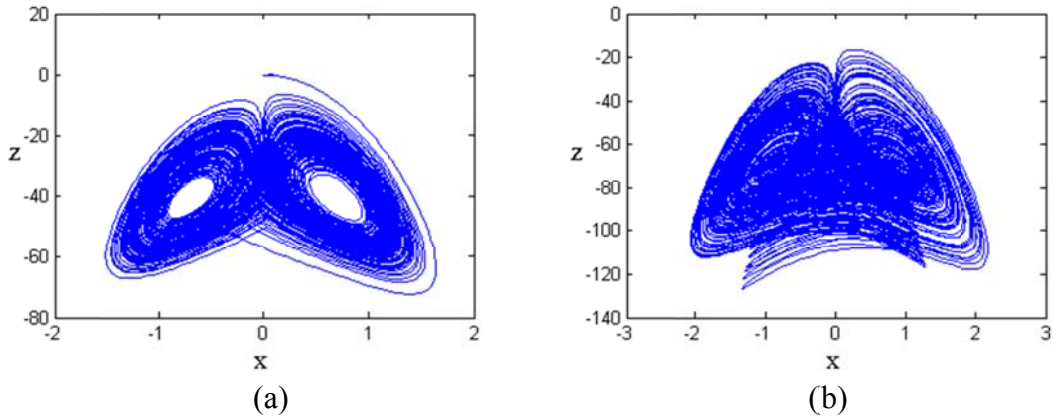
Bu sistem için  $a = 36$ ,  $b = 3$  şeklinde sabit tutulup  $c$  parametresi değiştirildikçe hem Lorenz benzeri hem de Chen benzeri kaotik bir çeker elde edilebilmektedir. Buna göre,  $12.7 < c < 17$  için Lorenz benzeri bir çeker elde edilirken,  $23 < c < 28.5$  için Chen benzeri bir çeker elde edilmektedir.



Şekil 5.6. Lü sisteminin a)  $c = 13$  için, b)  $c = 28$  için kaotik çekeri.

Lorenz ve Chen benzeri sistemler topolojik olarak birbirlerinden farklı olduklarından Lü sistemi gibi  $a_{12} \cdot a_{21} = 0$  şartını sağlayan sistemler sadece parametre değeri değiştirilerek karışık modlu olarak kullanılabilirler. Bu durum sözde rastgele sayı üretici olarak karışık modlu yapıyı kullandığımızda ayrıca avantajlıdır. Çünkü kaynak tüketimi açısından farklı sistemleri gerçekleştirerek kaynak harcamak yerine sadece parametre değişimiyle oldukça tasarruflu bir tasarım elde ederiz. Sürekli zamanlı kaotik sistemlerin ayrıklaştırılmasının kaotik haritaların gerçekleştirimine göre daha fazla kaynak gerektirdiğini göz önüne aldığımızda, sürekli zamanlı sistemlerin bu tür zengin dinamikler sergileyebilmesinin bu dezavantajı giderdiğini söyleyebiliriz. Bu bakımdan Lü sistemi bu bölümde tasarlayacağımız karışık modlu sözde rastgele sayı üretici için oldukça uygundur. Fakat bu tasarım için Lü sistemine oldukça benzeyen kendi geliştirmiş olduğumuz bir kaotik sistemi kullanacağız. Bu sistem Denklem (5.8) ile ifade edilmekte olup tıpkı Lü sistemi gibi hem Lorenz benzeri hem de Chen benzeri kaotik çekerler oluşturma kabiliyetine sahiptir. Yapısındaki karesel ifade nedeniyle Denklem (5.5) ile verilen genelleştirilmiş ifade ile gösterilememekle birlikte aynı tür Pitchfork bifürkasyonu gösterdiği için bu sınıftaki çekerlerden çok da farklı değildir.

$$\begin{aligned}\frac{dx}{dt} &= -9x + y \\ \frac{dy}{dt} &= ry + xz \\ \frac{dz}{dt} &= -y^2 - z\end{aligned}\tag{5.8}$$



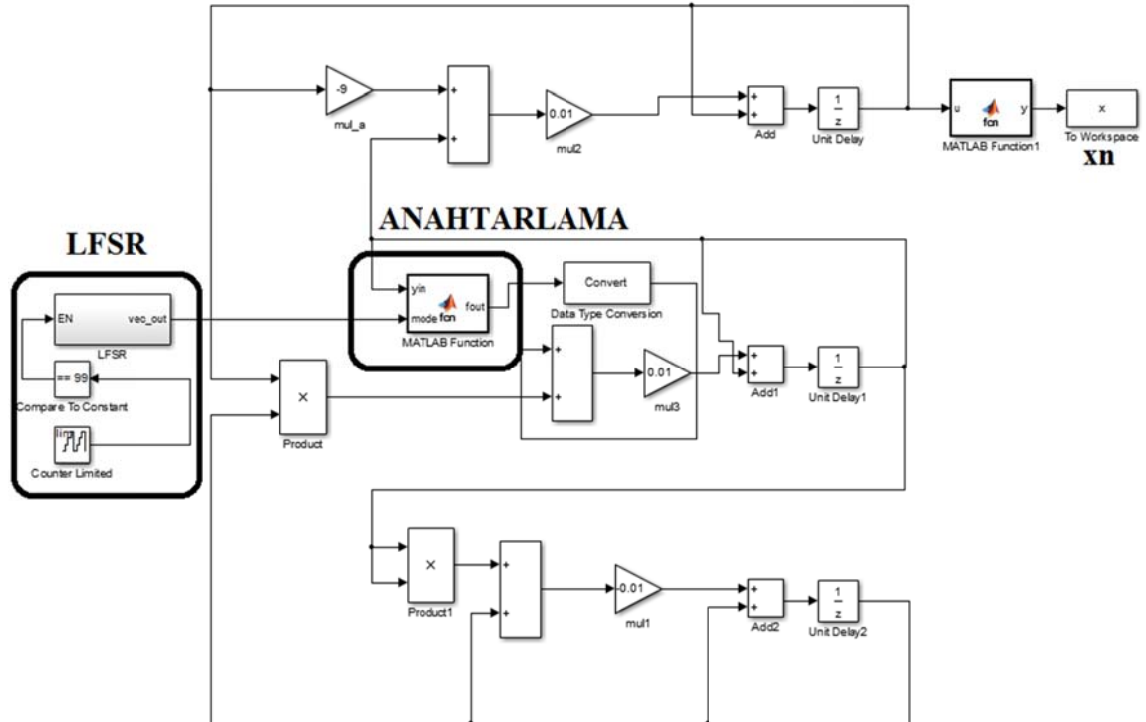
Şekil 5.7. Karışık modlu kullanılacak sistem için a)  $r = 4.5$  için, b)  $r = 7$  için elde edilen çeker görüntüleri.

Basitlik açısından bu sistemin bir kontrol parametresi -9 değerine sabitlenmiş olup sadece  $r$  parametresi değiştirilerek Lorenz ve Chen benzeri çıkışlar elde edilmektedir. Buna göre,  $r = 4.5$  için Lorenz benzeri kaotik çeker elde edilirken  $r = 7$  için Chen benzeri bir kaotik çeker elde edilmektedir. Elde edilen çeker görüntüleri Şekil 5.7’de görüldüğü gibidir.

#### 5.4. Karışık Modlu Sözde Rastgele Sayı Üretici Tasarımı

Karışık modlu sözde rastgele sayı üretici olarak kullanmak için Denklem (5.8) ile verilen sistemin ayrıklaştırılması gerekmektedir. Bunun için en basit olan ve en az kaynak tüketimi gerektiren ileri Euler ayrıklaştırma metodundan faydalanılmıştır. Elde edilen yeni denklem takımı Denklem (5.9)’da görüldüğü gibidir. Adım aralığı  $\Delta h = 0.01$  olarak alınmıştır.

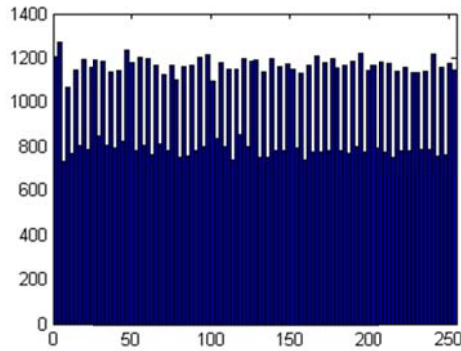
$$\begin{aligned} x_{n+1} &= x_n - (9x_n - y_n)\Delta h \\ y_{n+1} &= y_n + (ry_n + x_n z_n)\Delta h \\ z_{n+1} &= z_n - (y_n^2 + z_n)\Delta h \end{aligned} \quad (5.9)$$



Şekil 5.8. Sözde rastgele sayı üreticinin Matlab modeli.

Sistem ayrıklaştırıldıktan sonra tek yapılması gereken  $r$  parametresini 4.5 ve 7 değerleri arasında değiştirerek Lorenz ve Chen benzeri çıkışların sürekli değiştirilmesini ve birinin son değerlerinin diğerinin başlangıç şartı olarak kullanılmasını sağlamaktır. Bunu yapmak için m-serili bir lineer geribeslemeli kaydırmalı kaydediciden faydalanılmıştır. Eğer bu sözde rastgele sayı üreticinin çıkışı 1 ise  $r = 4.5$  ve çıkışı 0 ise  $r = 7$  olarak alınmaktadır. Lineer geribeslemeli kaydırmalı kaydedicinin kaç iterasyonda bir güncelleneceği ise  $\Delta i$  değeriyle ifade edilmektedir. Tasarımın Matlab Simulink gerçekleştirimi Şekil 5.8’de görüldüğü gibidir.

Bu gerçekleştirim için Q44.32 sabit noktalı aritmetik kullanılmış olup lineer geribeslemeli kaydırmalı kaydedicinin mertebesi 32 ve  $\Delta i = 100$  olarak seçilmiştir. Sözde rastgele sayı elde etmek için  $x_n$  çıkışının son 8 biti kullanılmıştır. Elde edilen sözde rastgele sayıların histogramı Şekil 5.9’daki gibidir.



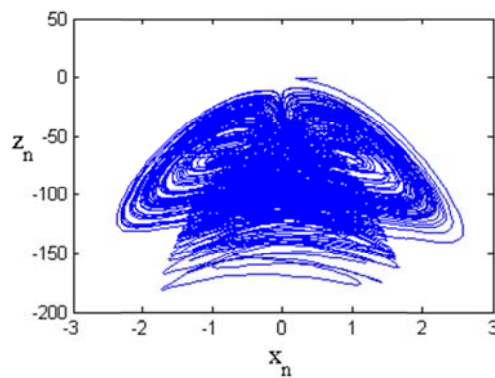
Şekil 5.9. Üretilen sözde rastgele sayıların histogramı.

Elde edilen bu histogramı ideal bir tekdüze dağılım göstermemektedir. Bunun nedeni ise anahtarlamalar sırasında elde edilen çekerlerin uzayda farklı bölgeleri kaplamalarıdır. Bu değişiklik karşılaşılan sayıların her anahtarlamada farklı frekanslarda fakat kendi içerisinde tekdüze dağılmasına neden olmaktadır. Bu rastgele sayılar frekans testini geçtiği için bu şekildeki bir dağılım büyük bir dezavantaj yaratmamaktadır. 1000 farklı başlangıç artı için elde edilen  $10^6$  bitlik bit serileri için uygulanmış olan NIST SP 800-22 rastgelelik testi sonuçları Tablo 5.1’de özetlenmiştir. Buna göre, tasarlanan sözde rastgele sayı üretici tüm testleri geçmektedir. Tablo 5.1’deki NIST testi sonuçlarında birden fazla sonucu olan testler için ortalama değerler verilmiştir. Yapılan testlerde rastgele sapmalar ve değişken rastgele sapmalar testleri dışındaki tüm testler

için minimum geçme oranı 0.980'dir. Rastgele sapmalar ve değişken rastgele sapmalar testleri için minimum geçme oranı 0.977 olarak hesaplanmıştır. Karışık modlu yapının anahtarlanması sonucu  $x_n$  ve  $z_n$  değerlerinin oluşturmuş olduğu çeker Şekil 5.10'da sunulmuştur. Elde edilen yeni çeker Şekil 5.7'deki Lorenz ve Chen benzeri çekelerin birleşimidir. Görüldüğü üzere anahtarlama faz uzayındaki dinamiklerin daha kompleks bir hal almasına neden olmuştur.

Tablo 5.1. Karışık modlu sözde rastgele sayı üretici için NIST rastgelelik testi sonuçları.

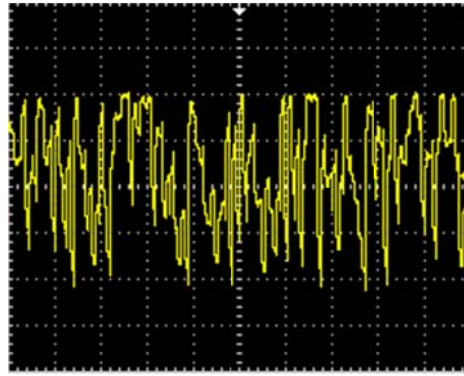
| No.      | Test                         | Oran  | P-değeri | Sonuç |
|----------|------------------------------|-------|----------|-------|
| 1        | Frekans                      | 0.990 | 0.299736 | Geçti |
| 2        | Blok frekans                 | 0.995 | 0.880145 | Geçti |
| 3..4     | Kümülatif toplamlar          | 0.991 | 0.764268 | Geçti |
| 5        | Yinelemeler                  | 0.989 | 0.641284 | Geçti |
| 6        | En uzun yinelemeler          | 0.985 | 0.747898 | Geçti |
| 7        | Rank                         | 0.991 | 0.834308 | Geçti |
| 8        | FFT                          | 0.991 | 0.829047 | Geçti |
| 9..156   | Örtüşmeyen şablon eşleştirme | 0.990 | 0.487812 | Geçti |
| 157      | Örtüşen şablon eşleştirme    | 0.992 | 0.630872 | Geçti |
| 158      | Evrensel                     | 0.987 | 0.128132 | Geçti |
| 159      | Yaklaşık entropi             | 0.987 | 0.174728 | Geçti |
| 160..167 | Rastgele sapmalar            | 0.990 | 0.634945 | Geçti |
| 168..185 | Değişken rastgele sapmalar   | 0.992 | 0.461929 | Geçti |
| 186..187 | Seri                         | 0.993 | 0.289845 | Geçti |
| 188      | Lineer komplekslik           | 0.988 | 0.836048 | Geçti |



Şekil 5.10. Karışık modlu çeker yapısı.

### 5.5. Karışık Modlu Sözde Rastgele Sayı Üreticinin Donanım Gerçekleştirimi

Bu sözde rastgele sayı üreticini de karşılaştırma yapabilmek açısından Altera DE0 bordu üzerindeki Cyclone III EP3C16F484C6 FPGA elemanı üzerinde önceki bölümde eşleştirilmiş kaotik sistem için yapılabenzer şekilde gerçekleştirdik. 8 bitlik rastgele çıkışın D/A dönüştürücü kullanılarak elde edilen osiloskop görüntüsü Şekil 5.11’de verilmiştir.



Şekil 5.11. Karışık modlu sözde rastgele sayı üreticinin çıkışının osiloskop görüntüsü.

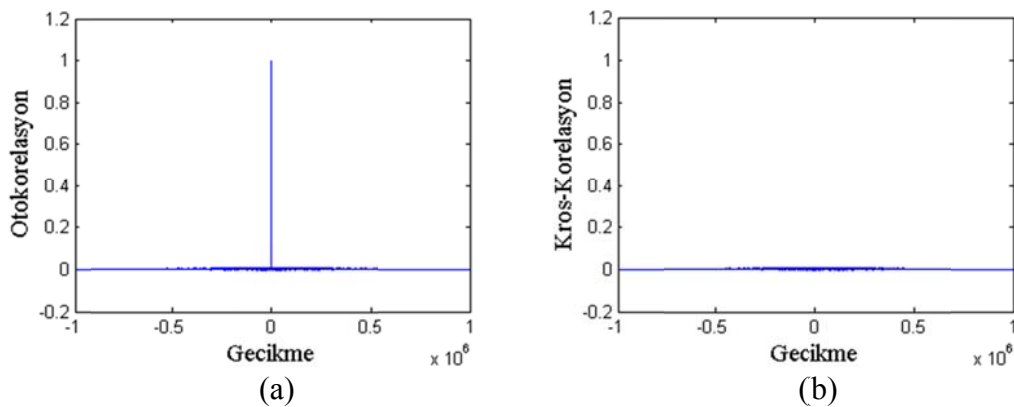
Tablo 5.2. Sentezleme sonuçlarının karşılaştırılması.

| SRSÜ                        | Toplam Bit Uzunluğu | Kaynak Tüketimi (LE) | Mak. Çalışma Frekansı | Veri Çıkış Hızı |
|-----------------------------|---------------------|----------------------|-----------------------|-----------------|
| Eşleştirilmiş Kaotik Sistem | 32                  | 5741 (%37)           | 5.13 MHz              | 5 Mbps          |
| Karışık Modlu SRSÜ          | 44                  | 1826 (%12)           | 30.98 MHz             | 247 Mbps        |

Sentezleme sonucu elde edilen değerler ise karşılaştırmalı biçimde eşleştirilmiş kaotik sistemin sentez sonuçlarıyla birlikte Tablo 5.2’de özetlenmiştir. Elde edilen sonuçlara göre karışık modlu üreteç beklenenin aksine kaotik harita kullanılan eşleştirilmiş kaotik sistemden daha az kaynak gerektirmektedir. Bu durumun iki nedeni vardır. Birincisi, eşleştirilmiş kaotik sistemde kullanılan bölme operasyonunun çarpmaya göre hem daha fazla kaynak gerektirmesidir. İkincisi ise, haritanın sonraki değerlerinin tahmin edilememesini sağlamak için iki tane kaotik haritanın kullanılmasıdır. Oysaki karışık modlu üreteçte tek bir sistemin gerçekleştirimi kullanılmıştır. Rastgele parametre değişimleri sistemin tüm dinamiğini her anahtarlamayla değiştirdiğinden karışık modlu

yapıda sonraki değerlerin tahmin edilebilmesi zaten oldukça zordur. Maksimum çalışma frekanslarının bu kadar farklı çıkmasının nedeni ise yine bölme işleminin yavaş olması yüzündendir. Karışık modlu üreteçte aynı anda 8 bit üretilebildiğinden veri çıkış hızı eşleştirilmiş kaotik sistemin kat kat üstündedir.

Karışık modlu rastgele sayı üreticinin 8 bitlik sözde rastgele çıkışının korelasyon bağıntıları Şekil 5.12’de verilmektedir. Buna göre üreteç ideal korelasyon özellikleri göstermektedir. Zaten minimum periyodu pertürbasyonlu sistemlere benzer şekilde  $\Delta i \cdot (2^L - 1)$  olacağından incelenen ilk  $10^6$  bit için herhangi bir korelasyon ilişkisi bulmak zordur. Bu periyot uzunluğu, gerçekleştirilen sistem için  $4.29 \times 10^{11}$  olup pek çok uygulama için yeterlidir ve  $\Delta i$  arttırılmak suretiyle daha da arttırılabilir.



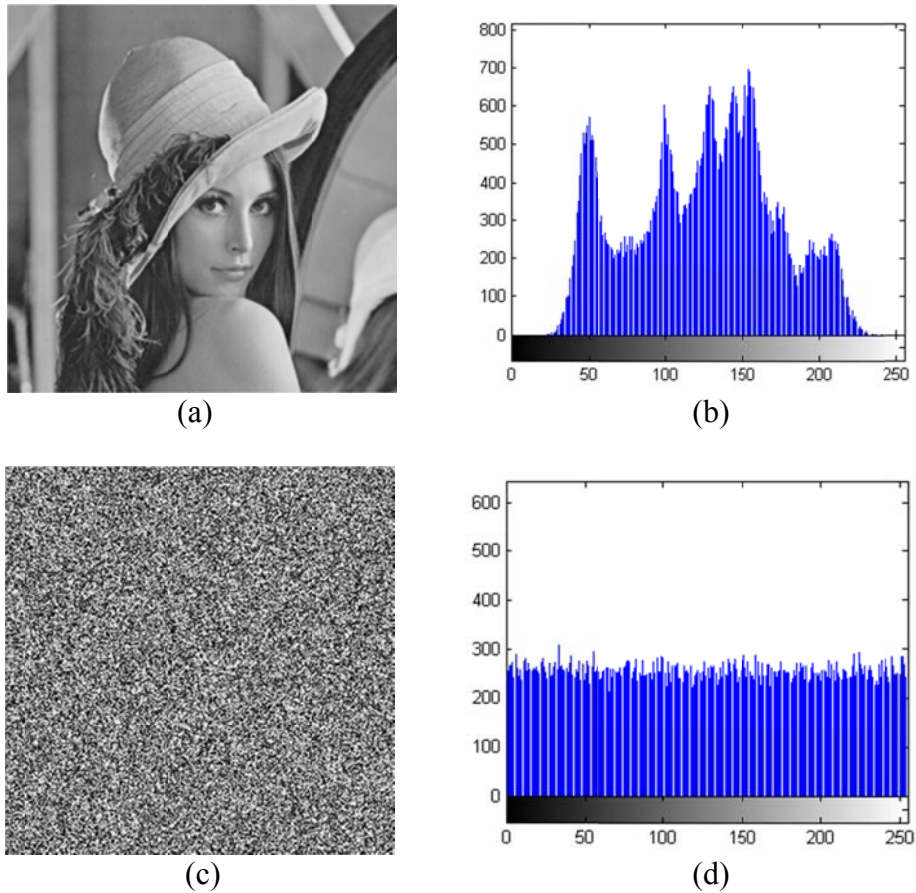
Şekil 5.12. Karışık modlu üreticinin çıkış serisi için elde edilmiş a) otokorelasyon, b) kros-korelasyon bağıntıları.

## 5.6. Karışık Modlu Üreticinin Görüntü Şifrelemede Kullanımı

Karışık modlu sözde rastgele sayı üreticinin veri çıkış hızı yüksek olduğu için önceki bölümde yapmış olduğumuz gibi tampon bellek kullanmaksızın doğrudan şifreleme yapabiliriz. Bu tür şifreleme Şekil 2.17’de gösterilmişti. Yapılacak şifreleme uygulaması yine simetrik anahtarlı olmalı. Veri yolu olarak 44 bit kullandığımız için kaotik sistemin ve lineer kaydırmalı kaydedicinin başlangıç koşulları anahtar olarak kullanıldığında toplam 164 bitlik bir anahtar uzunluğu elde ederiz.

Yapılacak olan şifrelemede önceki bölümdeki gibi 256x256 gri tonlamalı bir görüntü şifrelenecek olsa da bu üreteç için  $y_n$  ve  $z_n$  değişkenlerinden elde edilecek 8 bitlik

çıkışlarla rahatlıkla renkli (RGB) görüntüler de şifrelenebilir. Bu durumda veri çıkış hızı 743 Mbps'ye kadar çıkacaktır. Fakat, maksimum çalışma frekansı 30.98 MHz olduğundan ancak 25 MHz saat frekansında çalışan 640x480 çözünürlüklü VGA standardında görüntüleme yapabiliriz. Bu üreticinin Matlab Simulink modeli kullanılarak yapılan şifreleme sonucu elde edilen şifrelenmiş görüntü ve histogramı Şekil 5.13'de görülmektedir. Buna göre, bu üreteçle yapılan şifrelemede de şifrelenmiş görüntünün histogramı orijinal görüntü hakkında fikir vermeyecek şekilde tekdüze bir dağılım göstermektedir.



Şekil 5.13. a) Orijinal görüntü, b) orijinal görüntünün histogramı, c) şifrelenmiş görüntü, d) şifrelenmiş görüntünün histogramı.

Altera DE2 bordu üzerindeki Cyclone IV EP4CE115F29C7 FPGA çipinin VHDL ile programlanması ile yapılan donanım gerçekleştirimi sonucu RGB monitörde elde edilen görüntüler ise Şekil 5.14'de verilmiştir. FPGA gerçekleştirimi için 114,480 lojik elementten 40,069'u (%35) kullanılmıştır. Tampon bellek kullanılmadığı için hafıza kullanımı olmamıştır. Görüntüyü saklamak için yine lojik elementlerden

faaydalanılmıřtır. Saat frekansı ise bahsettiđimiz üzere 640x480 VGA standardına uygun řekilde 25 MHz olarak alınmıřtır. B ylelikle karıřık modlu s zde rastgele sayı  retecinden alınan 200 Mbps’lik veri, RGB monit r n kırmızı, yeřil ve mavi giriřlerine Altera DE2 bordu  zerinde bulunan 8 bitlik D/A d n řt r c  yardımıyla aynı anda uygulanmıř ve ekranda gri tonlamalı g r nt  elde edilmiřtir. Yine bordun  st ndeki anahtarlar yardımıyla řifrelemenin aktif ya da pasif olması kontrol altına alınmıřtır.



(a)



(b)

řekil 5.14. Donanım ger ekleřtirmesi sonucu elde edilen a) orijinal ekran g r nt s , b) řifrelenmiř ekran g r nt s .

### 5.7. Karıřık Modlu  retec n Avantaj ve Dezavantajları

Karıřık modlu s zde rastgele sayı  retec  s rekli zamanlı kaotik sistemlerin ayrıklařtırılmasına dayalı olmasına rađmen zengin dinamikleri sayesinde orijinalinde ayrık zamanlı olan kaotik haritalarla  retilmiř s zde rastgele sayı  rete lerine nazaran daha hızlı  alıřıp ger ekleřtirim i in daha az kaynak gerektirmektedir.  zellikle  ok boyutlu olması nedeniyle i erisinden  ıkarılabilecek veri olduk a artmaktadır. Tek bir parametre deđiřimiyle sistemin t m  zelliklerinin deđiřmesi ve bu deđiřimin ise rastgele anahtarlamalarla kontrol altına alınması, sistemin eřleřtirme olmaksızın dođrudan kullanılabilecek řekilde g venli olmasını sađlamaktadır. Bunun yanı sıra, bu anahtarlama iřlemi iki sistemden birinin son deđerinin diđerinin bařlangı  řartı olarak kullanılması nedeniyle s zde y r ngelerin kararlı hale gelmesini engellemekte ve integrasyon hatalarının propogasyonunu  nlemektedir. Bu nedenle bu iřlem pert rbasyona bir alternatif olarak da d ř n lebilir.

Tüm bunlara rağmen üretcin eksik kaldığı noktalar da bulunmaktadır. Üreteç NIST testlerini geçmesine rağmen ideal bir tekdüze dağılım göstermemektedir. Anahtarlama sonucu tanım aralığının değişmesi bunun başlıca nedenidir. Bu ise ileride parametre değişimiyle hemen hemen aynı tanım aralığında tanımlı iki farklı kaotik çekerin elde edildiği bir tasarımla giderilebilir. Bunun dışında bu üretcin kriptografik olarak güvenli olduğunu ispatlamak için sonraki bit testini (next bit test) geçtiği ispat edilmelidir.

## 6. BÖLÜM

### SONUÇ VE ÖNERİLER

Bu tez çalışmasında öncelikli olarak kaotik sistem tabanlı sözde rastgele sayı üretimi üzerinde yoğunlaşmıştır. Bu bağlamda ayrık zamanlı olması itibariyle dijital gerçekleştirmelere ve dolayısıyla sözde rastgele sayı üretimine uygun olan kaotik haritalar rastgele sayı üretimine yatkınlık bakımından incelenmiştir. Sonuç olarak sınır çarpışması bifürkasyonuna sahip olan ve dayanıklı kaos özelliği gösteren haritaların sözde rastgele sayı üretimi için ideal oldukları belirlenmiştir. Bunun yanı sıra kaosun başlangıç şartlarına hassas bağıllık, harmanlama ve ergodiklik gibi özellikleri ile modern kriptografik sistemlerde aranan yayılma ve karıştırma şartlarını sağladığı belirtilmiş ve bu nedenle kaos ile kriptografi arasında yakın bir ilişki olduğundan bahsedilmiştir. Örnek olarak kaotik sistem tabanlı bir akış şifreleyici ele alınmıştır.

Sonraki kısımda ise kaosun sadece sürekli uzayda tanımlı olması nedeniyle ayrık değerli uzayda yapılacak gerçekleştirimlerin pek çok probleme neden olduğu gösterilmiş ve sözde kaos kavramı tanıtılmıştır. Karşılaşılan gerçekleştirim problemleri genel olarak matematiksel bir teorinin eksikliği, kısa periyotlar ve farklı başlangıç şartları için birebir aynı periyotların elde edilmesi şeklinde gruplandırılabilir. Bu problemlerin çözümü için kullanılan mühendislik metotlarından sadece pertürbasyonun problemlere en iyi çözümü getirdiği görülmüştür. Çünkü transient ve periyot kesişimleri sözde kaotik yörüngelerin birebir aynı olmasına neden olmakta ve farklı başlangıç şartları için sürekli aynı periyotların elde edilmesi uzun periyotların mevcudiyetini anlamsızlaştırmaktadır. Bunun yerine çok küçük aralıklarla yapılacak pertürbasyonlarla hem bu kesişimler önlenirken hem de çıkıştaki bit serisinin çok uzun iterasyonlar boyunca kendini tekrar etmeyeceği garanti altına alınmaktadır.

Kaotik sistem tabanlı rastgele sayı üreticilerinin gerçekleştirmeleri detaylıca irdelendikten sonra pertürbasyon kullanan literatürdeki mevcut kaos tabanlı sözde rastgele sayı üreticileri arasından baştan beri ortaya koymuş olduğumuz gereksinimleri en iyi şekilde karşılayan bir üreticinin donanım gerçekleştirimi yapılmış ve görüntü şifrelemede kullanımı gösterilmiştir. Neticede bu üreticinin dayanıklı kaos sağlamak için faydalandığı bölme işlemlerinin donanım gerçekleştirmelerini büyük ölçüde zedelediği ve sistemi yavaşlattığı görülmüştür. Bu durumun üstesinden gelmek için üretilen sözde rastgele sayılar bir tampon bellekte saklanıp o şekilde kullanılmıştır.

Daha sonra ise literatürdeki genel eğilimin aksine sürekli zamanlı kaotik sistemlerin ayrıklaştırılmasıyla elde edilen sözde rastgele sayı üreticilerinin tasarımı üzerinde durulmuştur. Bu tür gerçekleştirimlerde, kaotik sistemlerdeki gerçekleştirim problemlerinin yanı sıra bir de nümerik integrasyondaki hatalar nedeniyle meydana gelecek problemlerle karşılaşmaktadır. Bu problemler nedeniyle kaotik sistemlerin dijital gerçekleştirimleri orijinal sistemden farklı türde davranış gösterebilmekte ve yörüngeler kararlı hale gelerek kısa periyotlar oluşturmaktadırlar. Literatürde bulunan önceki çalışmalarda bu problemlerin üstesinden gelmek için de pertürbasyon uygulamak önerilmiştir. Fakat bu tez çalışmasında farklı bir yol izlenerek karışık modlu yapılarla bu sorunlar giderilmeye çalışılmıştır. Karışık modlu yapı orijinalde iki kaotik sistemin anahtarlanması metoduna dayanmaktadır. Buradaki uygulamada ise bir sistemin iki farklı kaotik durumunun anahtarlanmasında kullanılarak oldukça fazla kaynak tasarrufu elde edilmiştir. Nitekim bu durum donanım gerçekleştirimine yansımış ve beklenenin aksine kaotik harita kullanan pertürbasyonlu sözde rastgele sayı üreticisine nazaran çok daha yüksek veri çıkış hızları daha az kaynak tüketilerek elde edilebilmiştir. Karışık modlu anahtarlama kullanılarak da hem yörüngelerin kararlı hale gelmesi önlenmiş hem de nümerik integrasyon hatalarının zamanla propogasyonunun önüne geçilmiştir. Bu da yukarıda bahsedilen sorunların çözümü için karışık modlu yapının pertürbasyona bir alternatif teşkil etmesine neden olmaktadır.

Son olarak oluşturmuş olduğumuz çerçeveye genel olarak baktığımızda halen kaos tabanlı rastgele sayı üreticilerinin pek çok eksiği olduğunu görmekteyiz. Bunun başlıca nedeni gerçekleştirimler için halihazırda mevcut bir matematiksel teoremin olmayışdır. Bu da bugüne kadar sayısız kaos tabanlı rastgele sayı üretici önerilmesine ve bu konuda patentler alınmasına rağmen, bu tür üreticilerin gerçekleştirimini şüpheye düşürmekte ve

bunların genel geçer ticari bir uygulamada kullanımlarının önüne geçmektedir. Gerçekleştirimlerde sonlu çözünürlük gibi etkilerin sonuçlarını tamamen göz ardı eden ve tamamen varsayımsal olarak kurulmuş rastgele sayı üreticilerinin çokluğu da bu konuda yardımcı olmamaktadır. Dolayısıyla, kaotik sistem tabanlı sözde rastgele sayı üretimi hakkında daha fazla çalışma yapılması gerekmektedir.

## KAYNAKLAR

1. Menezes, A., Oorschot, P., Vanstone S., 1996. Handbook of Applied Cryptography. CRC Press, Florida, 816 pp.
2. Trappe, W., Washington, L. C., 2006. Introduction to Cryptography with Coding Theory. Pearson, New Jersey, 592 pp.
3. Gentle, J. E., 2003. Random Number Generation and Monte Carlo Methods. Springer, New York, 398 pp.
4. Levy, P. S., Lemeshow, S., 1991. Simple random sampling. **Sampling of Populations: Methods and Applications, 4** : 43-81.
5. Schneier, B., 1996. Applied Cryptography. Wiley, New York, 758 pp.
6. Bucci, M., Luzzi, R., 2008. Fully digital random bit generators for cryptographic applications. **IEEE Trans. Circuits and Syst. I**, **55** (3) : 861-875.
7. Dalkıran, İ., 2010. Yapay Sinir Ağı Tabanlı Kaotik İşaret Üretici Tasarımı ve Uygulamaları. Erciyes Üniversitesi, Doktora Tezi, Kayseri, 116 s.
8. Maher, D. P., Rance, R. J., 1999. Random number generators founded on signal and information theory, pp. 219-230. In: Cryptographic Hardware and Embedded Systems (Eds: Ç.K. Koç, C. Paar). Springer, Berlin.
9. Holman, W.T., Connelly, J.A., Dowlatabadi, A.B., 1997. An integrated analog/digital random noise source. **IEEE Trans. Circuits and Syst. I**, **44** (6) : 521-528.
10. Bagini, V., Bucci, M., 1999. A design of reliable true random number generator for cryptographic applications, pp. 204-218. In: Cryptographic Hardware and Embedded Systems (Eds: Ç.K. Koç, C. Paar). Springer, Berlin.
11. Bucci, M., Germani, L., Luzzi, R., Tommasino, P., Trifiletti, A., Varanonuovo, M., 2003. A high-speed IC random-number source for smartcard microcontrollers. **IEEE Trans. Circuits and Syst. I**, **50** (11) : 1373-1380.
12. Tokunaga, C., Blaauw, D., Mudge, T., 2008. True random number generator with a metastability-based quality control. **IEEE Journal of Solid-State Circuits**, **43** (1) : 78-85.
13. Majzoobi M., Koushanfar F., Devadas S., 2011. FPGA-Based true random number generation using circuit metastability with adaptive feedback control,

- pp. 17-32. *CHES 2011, Cryptographic Hardware and Embedded Systems, Sep. 28 - Oct. 1, Nara, Japan.*
14. Petrie, C.S., Connelly, J.A., 1996. Modeling and simulation of oscillator-based random number generators, pp. 324-327. *ISCAS'96, IEEE International Symposium on Circuits and Systems, May 12, Atlanta, Georgia.*
  15. Sunar, B., Martin, W.J., Stinson, D.R., 2007. A provably secure true random number generator with built-in tolerance to active attacks. **IEEE Trans. Computers**, **56** (1) : 109-119.
  16. Schellekens D., Preneel B., Verbauwhede I., 2006. FPGA vendor agnostic true random number generator, pp. 1-6. *FPL'06, International Conference on Field Programmable Logic and Applications, August 28-30, Madrid, Spain.*
  17. Yalcin, M.E., Suykens, J.A.K., Vandewalle, J., 2004. True random bit generation from a double-scroll attractor. **IEEE Trans. Circuits and Syst. I**, **51** (7) : 1395-1404.
  18. Stojanovski, T., Kocarev, L., 2001. Chaos-based random number generators-part I: analysis. **IEEE Trans. Circuits and Syst. I**, **48** (3) : 281-288.
  19. Addabbo, T., Alioto, M., Fort A., Rocchi, S., Vignoli, V., 2006. A feedback strategy to improve the entropy of a chaos-based random bit generator. **IEEE Trans. Circuits and Syst. I**, **53** (2) : 326-337.
  20. Callegari S., Rovatti R., Setti G., 2005. Embeddable ADC-based true random number generator for cryptographic applications exploiting nonlinear signal processing and chaos. **IEEE Trans. Signal Processing**, **53** (2) : 793-805.
  21. Degaldo-Restituto, M., Medeiro, F., Rodriguez-Vazquez, A., 1993. Nonlinear switched-current CMOS IC for random signal generation. **Electronics Letters**, **29** (25) : 2190-2191.
  22. Ergün, S., Özoğuz, S., 2007. Truly random number generators based on a non-autonomous chaotic oscillator. **International Journal of Electronics and Communications**, **61** (4) : 235-242.
  23. Zeng, K., Yang, C.H., Wei, D.Y., Rao, T., 1991. Pseudorandom bit generators in stream-cipher cryptography. **Computer**, **24** (2) : 8-17.
  24. Oishi, S., Inoue, H., 1982. Pseudo-random number generators and chaos. **Transactions of the Institute of Electronics and Communication Engineers of Japan E**, **65** : 534-541.

25. Lin, T., Chua, L. O., 1993. New class of pseudo-random number generator based on chaos in digital filters. **International Journal of Circuit Theory and Applications**, **21** : 473-480.
26. Andrecut, M., 1998. Logistic map as a random number generator. **International Journal of Modern Physics B**, **12** : 921-930.
27. Kolesov, V. V., Belyaev, R. V., Voronov, G. M., 2001. A digital random-number generator based on the chaotic signal algorithm. **Journal of Communications Technology and Electronics**, **46** (11) : 1258-1263.
28. Stojanovski, T., Pihl, J., Kocarev, L., 2001. Chaos based random number generators - Part II: Practical realization. **IEEE Trans. Circuits and Syst. I**, **48** (3) : 382-385.
29. Shujun, Li., Xuanqin, M., Yuanlong, C., 2001. Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography, pp. 316-329. *INDOCRYPT 2001, Second International Conference on Cryptology, Dec. 16-20, 2001, Chennai, India.*
30. Kocarev, L., Jakimoski, G., 2003. Pseudorandom bits generated by chaotic maps. **IEEE Trans. Circuits and Syst. I**, **50** (1) : 123-126.
31. Huaping, L., Wang, S., Gang, H., 2004. Pseudo-random number generator based on coupled map lattices. **International Journal of Modern Physics B**, **18** : 2409-2414.
32. Liu, J., 2005. Design of a chaotic random sequence and its application. **Computer Engineering**, **31** : 150-152.
33. Wang, L., Wang, F.P., Wang, Z.J., 2006. Novel chaos-based pseudo-random number generator. **Acta Physica Sinica**, **55** : 3964-3968.
34. Patidar, V., Sud, K. K., Pareek, N. K., 2009. A pseudo random bit generator based on chaotic logistic map and its statistical testing. **Informatica**, **33** (4) : 441-452.
35. Mohammad Seyedzadeh, S., Mirzakuchaki, S., 2012. A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. **Signal processing**, **92** (5) : 1202-1215.
36. Kanso, A., Ghebleh, M., 2012. A novel image encryption algorithm based on a 3D chaotic map. **Communications in Nonlinear Science and Numerical Simulation**, **17** (7) : 2943-2959.

37. Tarsa, I.G., Budariu, G.D., Grozea, C., 2010. Study on a true random number generator design for FPGA, pp. 461-464. *COMM 2010, 8th International Conference on Communications, June 10-12, Bucharest, Romania.*
38. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., 2001. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Tech. rep., NIST special publication 800-22.
39. Marsaglia, G., 1995. The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness. (Web page: <http://www.stat.fsu.edu/pub/diehard/>), (Date accessed: July 2013).
40. National Institute of Standard and Technology, 2001. FIPS Pub 140-2, Security Requirements for Cryptographic Modules. (Web page: <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>), (Date accessed: July 2013).
41. Erat, M., 2008. FPGA Tabanlı, PCI Arayüzlü, Gerçek Zamanlı Rasgele Sayı Üretici Test Sistemi ve Uygulamaları. Erciyes Üniversitesi, Doktora Tezi, Kayseri, 97 s.
42. Strogatz, S., 1994. *Nonlinear Dynamics and Chaos*. Perseus Books, New York, 498 pp.
43. Alligood, K. T., Sauer, T. D., Yorke, J. A., 1996. *Chaos: An Introduction to Dynamical Systems*. Springer, New York, 603 pp.
44. Geist, K., Parlitz, U., Lauterborn, W., 1990. Comparison of different methods for computing lyapunov exponents. **Progress of Theoretical Physics**, **83** (5) : 875-893.
45. Banarjee, S., 2009. Lecture series on chaos, fractals and dynamical systems, lecture-24: Lyapunov exponents. (Web page: [http://utubersity.com/?page\\_id=895&tubepress\\_page=2](http://utubersity.com/?page_id=895&tubepress_page=2)), (Date accessed: May 2013).
46. Williams, G. P., 1997. *Chaos Theory Tamed*. Joseph Henry Press, Washington D.C., 406 pp.
47. Hilborn, R. C., 2000. *Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers*. Oxford University Press, New York, 643 pp.
48. Feigenbaum, M. J., 1978. Quantitative universality for a class of nonlinear transformations. **J. Stat. Phys.**, **19** : 25-52.

49. Mandelbrot, B., 1967. How long is the coast of Britain? Statistical self-similarity and fractional dimension. **Science, New Series**, **156** (3775) : 636-638.
50. Peitken, H., Jürgens, H., Saupe, D., 2004. Chaos and Fractals New Frontiers of Science. Springer, New York, 864 pp.
51. Chlouverakis, K. E., Sprott, J. C., 2005. A comparison of correlation and lyapunov dimensions. **Physica D**, **200** (1) : 156-164.
52. Kaplan, J., Yorke, J., 1979. Chaotic behavior of multidimensional difference equations, pp. 204-227. In: Functional Differential Equations and Approximation of Fixed Points, Lecture Notes in Mathematics (Eds: H.O. Peitgen, H.O. Walther). Springer, Berlin.
53. Eckmann, J. P., Ruelle, D., 1985. Ergodic Theory. **Reviews of Modern Physics**, **57** (3) : 617-656.
54. Petersen, K., 1989. Chaos and Fractals New Frontiers of Science. Cambridge University Press, New York, 329 pp.
55. Schuster, H. G., Just, W., 2005. Deterministic Chaos an Introduction. Wiley, Weinheim, 287 pp.
56. Sprott, J. C., 1989. Strange Attractors: Creating Patterns in Chaos. M&T Books, New York, 426 pp.
57. Li, S., Li, Q., Li, W., Mou, X., Cai, Y., 2001. Statistical properties of digital piecewise linear chaotic maps and their roles in cryptography and pseudo-random coding, pp. 205-221. In: Cryptography and Coding (Eds: B. Honary). Springer, Berlin.
58. Alvarez, G., Montoya, F., Romera, M., Pastor, G., 2003. Cryptanalysis of a discrete chaotic cryptosystem using external key. **Physics Letters A**, **319** (3) : 334-339.
59. Gardini, L., Tramontana, F., Avrutin, V., Schanz, M., 2010. Border-collision bifurcations in 1d piecewise-linear maps and Leonov's approach. **International Journal of Bifurcation and Chaos**, **20** (10) : 3085-3104.
60. Zhou, H., Ling, X., 1997. Problems with the chaotic inverse system encryption approach. **IEEE Trans. Circuits and Systems I**, **44** (3) : 268-271.
61. Nusse, H. E., Yorke, J. A., 1992. Border-collision bifurcations including "period two to period three" for piecewise smooth systems. **Physica D: Nonlinear Phenomena**, **57** (1) : 39-57.

62. Li, S., Chen, G., Mou, X., 2005. On the dynamical degradation of digital piecewise linear chaotic maps. **International Journal of Bifurcation and Chaos**, **15** (10) : 3119-3151.
63. Banerjee, S., Yorke, J. A., Grebogi, C., 1998. Robust Chaos. **Physical Review Letters**, **80** (14) : 3049-3052.
64. Nagaraj, N., Shastry, M. C., Vaidya, P. G., 2008. Increasing average period lengths by switching of robust chaos maps in finite precision. **The European Physical Journal Special Topics**, **165** (1) : 73-83.
65. Shannon, C. E., 1949. Communication theory of secrecy systems. **Bell Systems Technical Journal**, **28** (4) : 656-715.
66. Trappe, W., Washington, L. C., 2006. Introduction to Cryptography with Coding Theory. Pearson Prentice Hall, New Jersey, 577 pp.
67. Menezes, A. J., Van Oorschot, P. C., Vanstone, S. A., 1997. Handbook of Applied Cryptography. CRC Press, New York, 780 pp.
68. Kocarev, L., Jakimoski, G., Stojanovski, T., Parlitz, U., 1998. From chaotic maps to encryption schemes, pp. 514-517. *ISCAS'98, IEEE International Symposium on Circuits and Systems, May. 31- June 3, Monterey, CA, USA.*
69. Carroll, T. L., Pecora, L. M., 1991. Synchronizing chaotic circuits. **IEEE Trans. Circuits and Systems I**, **38** (4) : 453-456.
70. Guojie, H., Zhengjin, F., Ruiling, M., 2003. Chosen ciphertext attack on chaos communication based on chaotic synchronization. **IEEE Trans. Circuits and Systems I**, **50** (2) : 275-279.
71. Short, K. M., 1997. Signal extraction from chaotic communications. **International Journal of Bifurcation and Chaos**, **7** (7) : 1579-1597.
72. Zhou, C., Chen, T., 1997. Extracting information masked by chaos and contaminated with noise: Some considerations on the security of communication approaches using chaos. **Physics Letters A**, **234** (6) : 429-435.
73. Alvarez, G., Montoya, F., Pastor, G., Romera, M., 2004. Breaking a secure communication scheme based on the phase synchronization of chaotic systems. **Chaos**, **14** : 274-278.
74. Alvarez, G., Montoya, F., Romera, M., Pastor, G., 2004. Breaking two secure communication systems based on chaotic masking. **IEEE Trans. Circuits and Systems II**, **51** (10) : 505-506.

75. Stojanovski, T., Kocarev, L., 2001. Chaos-based random number generators–Part I: Analysis. **IEEE Trans. Circuits and Systems I**, **48** (3) : 281-288.
76. Stojanovski, T., Phil, J., Kocarev, L., 2001. Chaos-based random number generators–Part II: Practical realization. **IEEE Trans. Circuits and Systems I**, **48** (3) : 382-385.
77. Li, S., 2003. Analyses and New Designs of Digital Chaotic Ciphers. Xi'an Jiaotong University, Doktora Tezi, Shaanxi, China, 193 s.
78. Pareek, N. K., Patidar, V., Sud, K. K., 2003. Discrete chaotic cryptography using external key. **Physics Letters A**, **309** (1-2) : 75-82.
79. Zhang, H., Wang, H., Chen, W., 2002. Oversampled chaotic binary sequences with good security. **J. Circuits, Systems and Computers**, **11** (2) : 173-185.
80. Zhou, H., Ling, X., 1997. Generating chaotic secure sequences with desired statistical properties and high security. **International Journal of Bifurcation and Chaos**, **7** (1) : 205-213.
81. Bernstein, G. M., Lieberman, M. A., 1990. Secure random number generation using chaotic circuits. **IEEE Trans. Circuits and Systems**, **37** (9) : 1157-1164.
82. Bianco, M. E., Reed, D. A., 1991. Encryption system based on chaos theory. US Patent No. 5048086, 1991.
83. Forre, R., 1991. The henon attractor as a keystream generator, pp. 76-81. EuroCrypt'91, Advances in Cryptology, April 8-11, Brighton, UK.
84. Gentle, J. E., 2003. Random Number Generation and Monte Carlo Methods. Springer, New York, 381 pp.
85. Alvarez, G., Li, S., 2006. Some basic cryptographic requirements for chaos-based cryptosystems. **International Journal of Bifurcation and Chaos**, **16** (8) : 2129-2151.
86. Tsonis, A. A., 1991. The effect of truncation and round-off on computer simulated chaotic trajectories. **Computers Math. Applic.**, **21** (8) : 93-94.
87. Gleick, J., 1997. Kaos. Tübitak Yayınları, Ankara, 412 pp.
88. Hammel, M. S., Yorke, J. A., Grebogi, C., 1987. Do numerical orbits of chaotic dynamical processes represent true orbits?. **Journal of Complexity**, **3** (2) : 136-145.

89. Anosov, D. V., 1967. Geodesic flows and closed Riemannian manifolds with negative curvature. **Trudy Mat. Inst. Steklov**, **90** : 3-210.
90. Kocarev, L., 2001. Chaos-based cryptography: a brief overview. **IEEE Circuits and Systems Magazine**, **1** (3) : 6-21.
91. Sprott, J. C., 2003. Chaos and Time-Series Analysis, Oxford University Press, USA, 507 pp.
92. Matthews, R., 1989. On the derivation of a chaotic encryption algorithm. **Cryptologia**, **13** (1) : 29-42.
93. Wheeler, D. D., 1989. Problems with chaotic cryptosystems. **Cryptologia**, **13** (3) : 243-250.
94. Wheeler, D. D., Matthews, R., 1991. Supercomputer investigations of a chaotic encryption algorithm. **Cryptologia**, **15** (2) : 140-152.
95. Heidari-Bateni, G., McGillem, C., 1994. A chaotic direct-sequence spread-spectrum communication system. **IEEE Transactions on Communications**, **42** (234) : 1524-1527.
96. Tao, S., Ruli, W., Yixun, Y., 1998. Perturbance-based algorithm to expand cycle length of chaotic key stream. **Electronics Letters**, **34** (9) : 873-874.
97. Cernak, J. 1996. Digital generators of chaos. **Physics letters A**, **214** (3) : 151-160.
98. Chen, S. L., Hwang, T., Chang, S. M., Lin, W. W., 2010. A fast digital chaotic generator for secure communication. **International Journal of Bifurcation and Chaos**, **20** (12) : 3969-3987.
99. Lee, P. H., Pei, S. C., Chen, Y. Y., 2003. Generating chaotic stream ciphers using chaotic systems. **Chinese Journal of Physics**, **41** (6) : 559-581.
100. Lü, H., Wang, S., Li, X., Tang, G., Kuang, J., Ye, W., Hu, G., 2004. A new spatiotemporally chaotic cryptosystem and its security and performance analyses. **Chaos: An Interdisciplinary Journal of Nonlinear Science**, **14** (3) : 617-629.
101. Kanso, A., Smaoui, N., 2009. Logistic chaotic maps for binary numbers generations. **Chaos, Solitons & Fractals**, **40** (5) : 2557-2568.
102. Sun, F., Liu, S., Li, Z., Lü, Z., 2008. A novel image encryption scheme based on spatial chaos map. **Chaos, Solitons & Fractals**, **38** (3) : 631-640.

103. Xiang, T., Liao, X., Tang, G., Chen, Y., Wong, K. W., 2006. A novel block cryptosystem based on iterating a chaotic map. **Physics Letters A**, **349** (1) : 109-115.
104. Keller, J., Wiese, H., 2007. Period lengths of chaotic pseudo-random number generators, pp. 7-11. *Proceedings of the Fourth IASTED International Conference on Communication, Feb. 12-14, Innsbruck, Austria.*
105. Bogdan, C., Charge, P., Fournier-Purnaret, D., 2007. Behaviour of chaotic sequences under a finite representation and its cryptographic applications. *IEEE Workshop on Nonlinear Maps and Applications, Toulouse, France.*
106. Wang, S., Liu, W., Lu, H., Kuang, J., Hu, G., 2004. Periodicity of chaotic trajectories in realizations of finite computer precisions and its implication in chaos communications. **International Journal of Modern Physics B**, **18** : 2617-2622.
107. Nagaraj, N., Shastry, M. C., Vaidya, P. G., 2008. Increasing average period lengths by switching of robust chaos maps in finite precision. **The European Physical Journal Special Topics**, **165** (1) : 73-85.
108. Grebogi, C., Ott, E., Yorke, J. A., 1988. Roundoff-induced periodicity and the correlation dimension of chaotic attractors. **Physical Review A**, **38** : 3688-3692.
109. Beck, C., Roepstor, G., 1987. Effects of phase space discretization on the long-time behavior of dynamical systems. **Physica D: Nonlinear Phenomena**, **25** (1) : 173-180.
110. Beck, C., 1989. Scaling behavior of random maps. **Physics Letters A**, **136** (3) : 121-125.
111. Bastolla, U., Parisi, G., 1997. Attraction basins in discretized maps. **Journal of Physics A: Mathematical and General**, **30** (11) : 3757-3769.
112. Santoro, M. R., Bewick, G., Horowitz, M. A., 1989. Rounding algorithms for IEEE multipliers, pp. 176-183. *Proceedings of 9th Symposium on Computer Arithmetic, Sept. 6-8, Santa Monica, USA.*
113. Kavvadias, N., Masselos, K., 2012. Design of fixed-point rounding operators for the VHDL-2008 standard, pp. 1-8. *DASIP 2012, Design and Architectures for Signal and Image Processing, October 23-25, Karlsruhe, Germany.*

114. Stinson, D. R., 2006. *Cryptography: Theory and Practice*. CRC Press, USA, 616 pp.
115. Rosenstein, M. T., Collins, J. J., De Luca, C. J., 1993. A practical method for calculating largest Lyapunov exponents from small data sets. **Physica D: Nonlinear Phenomena**, **65** (1) : 117-134.
116. Hegger, R., Kantz, H., Schreiber, T., 1999. Practical implementation of nonlinear time series methods: The TISEAN package. **Chaos**, **9** (2): 413-435.
117. Fryska, S. T., Zohdy, M. A., 1992. Computer dynamics and shadowing of chaotic orbits. **Physics Letters A**, **166** (5): 340-346.
118. Guan, Z., Huang, F., Guan, W., 2005. Chaos-based image encryption algorithm. **Physics Letters A**, **346** (1): 153-157.
119. Sadoudi, S., Azzaz, M. S., Djeddou, M., Benssalah, M., 2009. An FPGA real-time implementation of the Chen's chaotic system for securing chaotic communications. **International Journal of Nonlinear Science**, **7** (4): 467-474.
120. Sadoudi, S., Azzaz, M. S., 2009. Hardware implementation of the Rössler chaotic system for securing chaotic communication, pp. 1-5. *SETIT 2009, 5<sup>th</sup> International Conference: Sciences of Electronic, Technologies of Information and Telecommunication, March 22-26, Tunisia*.
121. Blank, M., 1994. Pathologies generated by round-off in dynamical systems. **Physica D: Nonlinear Phenomena**, **78** (1) : 93-114.
122. Kılıç, R., Alçı, M., Tokmakçı, M., 2000. Mixed-mode chaotic circuit. **Electronics Letters**, **36** (2) : 103-104.
123. Matsumoto, T., Chua, L., Komuro, M., 1985. The double scroll. **Circuits and Systems**, **32** (8) : 797-818.
124. Robinson, R. C., 2013. *An Introduction to Dynamical Systems: Continuous and Discrete*. American Mathematical Society, USA, 760 pp.
125. Vanecek, A., Celikovsky, S., 1996. *Control Systems: From Linear Analysis to Synthesis of Chaos*. Prentice Hall, New Jersey, 277 pp.
126. Chen, G., Ueta, T., 1999. Yet another chaotic attractor. **International Journal of Bifurcation and Chaos**, **9** (7) : 1465-1466.
127. Ueta, T., Chen, G., 2000. Bifurcation analysis of Chen's equation. **International Journal of Bifurcation and Chaos**, **10** (8) : 1917-1931.

128. Lü, J., Chen, G., 2002. A new chaotic attractor coined. **International Journal of Bifurcation and Chaos**, **12** (3) : 659-661.

## ÖZGEÇMİŞ

### KİŞİSEL BİLGİLER

Adı, Soyadı: İsmail ÖZTÜRK

Uyruğu: Türkiye (TC)

Doğum Tarihi ve Yeri: 13 Temmuz 1985, Burdur

Medeni Durumu: Bekâr

Tel: +90 352 207 66 66 - 32237

Fax: +90 352 207 66 66

email: [ismailozturk@erciyes.edu.tr](mailto:ismailozturk@erciyes.edu.tr)

Yazışma Adresi: Erciyes Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü, 38039, Melikgazi/KAYSERİ

### EĞİTİM

| Derece | Kurum                        | Mezuniyet Tarihi |
|--------|------------------------------|------------------|
| Lisans | EÜ M.F. Elektrik-Elektronik  | 2011             |
| Lise   | Maltepe Askeri Lisesi, İzmir | 2004             |

### İŞ DENEYİMLERİ

| Yıl         | Kurum                                                       | Görev               |
|-------------|-------------------------------------------------------------|---------------------|
| 2012- Halen | EÜ Mühendislik Fakültesi<br>Elektrik-Elektronik Müh. Bölümü | Araştırma Görevlisi |

### YABANCI DİL

İngilizce

### YAYINLAR

1. Dahasert, N., Öztürk, İ., Kılıç, R., 2012. Izhikevich nöron modelinin alan programlanabilir elemanlarla gerçekleştirimi. *20. Sinyal işleme ve İletişim Uygulamaları Sempozyumu (SIU2012), Nisan 18-20, 2012, Fethiye, Antalya.*
2. Dahasert, N., Öztürk, İ., Kılıç, R., 2012. Experimental Realizations of the HR Neuron Model with Programmable Hardware and Synchronization Applications. **Nonlinear Dynamics**, **70**(4): 2343-2358.
3. Günay, E., Kılıç, R., Dahasert, N., Öztürk, İ., 2012. Merkezi desen üreteçleri için donanımsal çözümler. *Elektrik-Elektronik ve Bilgisayar Mühendisliği Sempozyumu (ELECO 2012), 29 Kasım - 01 Aralık 2012, Bursa.*