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

KAOS TABANLI KRİPTOSİSTEMLERİN YENİ TASARIM YAKLAŞIMLARI İLE DİJİTAL OLARAK GERÇEKLEŞTİRİMLERİ

Hazırlayan
İsmail ÖZTÜRK

Danışman
Prof. Dr. Recai KILIÇ

Doktora Tezi

Şubat 2019
KAYSERİ

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

KAOS TABANLI KRİPTOSİSTEMLERİN YENİ TASARIM YAKLAŞIMLARI İLE DİJİTAL OLARAK GERÇEKLEŞTİRİMLERİ

(Doktora Tezi)

Hazırlayan
İsmail ÖZTÜRK

Danışman
Prof. Dr. Recai KILIÇ

Şubat 2019
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

"Kaos Tabanlı Kriptosistemlerin Yeni Tasarım Yaklaşımları ile Dijital Olarak Gerçekleştirimleri" adlı Doktora tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi' ne uygun olarak hazırlanmıştır.

Hazırlayan
İsmail ÖZTÜRK

Danışman
Prof. Dr. Recai KILIÇ

Elektrik-Elektronik Mühendisliği ABD Başkanı

Prof. Dr. Necmi TAŞPINAR

Prof. Dr. Recai KILIÇ danışmanlığında **İsmail ÖZTÜRK** tarafından hazırlanan **"Kaos Tabanlı Kriptosistemlerin Yeni Tasarım Yaklaşımları ile Dijital Olarak Gerçekleştirimleri"** adlı bu çalışma jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü **Elektrik-Elektronik Mühendisliği** Anabilim Dalında **Doktora** tezi olarak kabul edilmiştir.

22/02/2019

JÜRİ:

Danışman : Prof. Dr. Recai KILIÇ .....

Üye : Prof. Dr. Kenan DANIŞMAN .....

Üye : Prof. Dr. Veysel ASLANTAŞ .....

Üye : Prof. Dr. Müştak E. YALÇIN .....

Üye : Doç. Dr. Serhan YAMAÇLI .....

ONAY:

Bu tezin kabulü Enstitü Yönetim Kurulunun ..26/02/2019 tarih ve 2019/16-04 ..sayılı kararı ile onaylanmıştır.

26/02/2019

Prof. Dr. Mehmet AKKURT

Enstitü Müdürü

TEŞEKKÜR

Bana çalışmalarım süresince her türlü yardımı ve fedakârlığı sağlayan, pek çok açıdan bana yol göstererek kariyerimin şekillenmesinde büyük rol oynayan danışmanım sayın Prof. Dr. Recai KILIÇ'a teşekkür ederim.

Ayrıca, bana olan inançları ve verdikleri koşulsuz destek için aileme de teşekkür ederim.

İsmail ÖZTÜRK

Şubat 2019, KAYSERİ

KAOS TABANLI KRİPTOSİSTEMLERİN YENİ TASARIM YAKLAŞIMLARI İLE DİJİTAL OLARAK GERÇEKLEŞTİRİMLERİ

İsmail ÖZTÜRK

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Doktora Tezi, Şubat 2019
Danışman: Prof. Dr. Recai KILIÇ

ÖZET

Kaos dinamik sistemlerin bir davranış çeşididir. Bu davranışta dinamik sistemin çıkışı periyodik değildir ve çıkış rastgele bir işarete benzemektedir. Buna rağmen, çıkış deterministik denklemlerle tamamen öngörülebilmektedir. Ayrıca, kaotik bir sistemin bu tür çıkışları başlangıç şartlarına hassas bağlı olup farklı başlangıç şartları istatistiksel olarak benzer özellikler sergilerler. Kaotik sistemlerin bu özellikleri onları kriptografik uygulamalarda kullanılmak için cazip hale getirmektedir. Bu nedenle literatürde "kaos tabanlı kriptografi" adında yeni bir çalışma alanı doğmuştur. Fakat kaos tabanlı kriptografi ciddi sorunlar içermektedir. En temel sorun ise kaosun dijital platformlarda gerçekleştirimlerinde yaşanmaktadır. Kaotik sistemlerin reel sayılar kümesinde tanımlanmasına karşın dijital platformlarda reel sayıları kullanmak olanaksızdır. Bu ise kaotik sistemlerin dijital platformlarda gerçekleştirimlerinde hatalı ve periyodik çıkışların üretilmesine neden olmaktadır. Bu nedenle, mevcut kaos tabanlı kriptosistemler ya gerçekleştirilmesi mümkün olmayan marjinal tasarımlardır, ya da araştırmacıların çoğunun bu durum dikkate almamasından dolayı güvenli değillerdir. Dolayısıyla, kaos tabanlı kriptosistemlerin dijital olarak gerçekleştirilmelerine yönelik yeni tasarım yaklaşımlarına ihtiyaç vardır. Bu tez çalışmasında bu açığı kapatmak üzere kaotik sistemlerin dijital olarak gerçekleştirilmesi için yeni bir metot geliştirilmiştir. Bu metot ile kaotik sistemlerin gerçek periyodik ve gerçek kaotik yörüngelerinin elde edilmesi sağlanmıştır. Dahası, üretilen bu yörüngelerin kaos tabanlı kriptosistemlerde kullanılabilmesi için yeni bir tasarım yaklaşımı önerilmiştir. Gerçekleştirimler yazılım ve FPGA (Alan Programlanabilir Kapı Dizileri) donanımı üzerinde yapılmıştır.

Anahtar Kelimeler: Kaos, Kriptografi, FPGA, Dijital gerçekleştirim.

DIGITAL REALIZATIONS OF CHAOS-BASED CRYPTOSYSTEMS WITH NEW DESIGN APPROACHES

İsmail ÖZTÜRK

Erciyes University, Graduate School of Natural and Applied Sciences
PhD Thesis, February 2019
Supervisor: Prof. Dr. Recai KILIÇ

ABSTRACT

Chaos is one of the behavior types of dynamical systems. In this behavior, the output of the dynamical system is non-periodic, and the output seems a random signal. Nevertheless, the output can be completely determined by deterministic equations. Besides, the outputs of chaotic systems have sensitive dependence on initial conditions and different initial conditions exhibit statistically similar properties. Such properties of chaotic systems make them appealing for cryptographic applications. For this reason, a new field of study called "chaos-based cryptography" has been born in the literature. But, chaos-based cryptography has serious problems. The most fundamental problem is about realizations of chaos on digital platforms. Although the chaotic systems are defined on real number sets, it is not possible to utilize real numbers on digital platforms. This causes the production of erroneous and periodic outputs on digital realizations of chaotic systems. Therefore, existing chaos-based cryptosystems are either marginal designs that are not possible to realize, or they are unsecure because most of the researchers ignore this fact. Accordingly, new design approaches are needed for digital realizations of chaos-based cryptosystems. In this thesis, in order to close this deficiency, a new method is developed for digital realizations of chaotic systems. The obtention of true periodic and true chaotic orbits is made possible by using this method. Moreover, a new design approach is proposed in order to utilize these orbits in chaos-based cryptosystems. The realizations are performed on software and FPGA (Field Programmable Gate Array) hardware.

Keywords: Chaos, Cryptography, FPGA, Digital implementation.

İÇİNDEKİLER

KAOS TABANLI KRİPTOSİSTEMLERİN YENİ TASARIM YAKLAŞIMLARI İLE DİJİTAL OLARAK GERÇEKLEŞTİRİMLERİ

KISALTMALAR

Sembol	Açıklama
FPGA	Alan programlanabilir kapı dizileri (ing. field programmable gate array)
b.ş.h.b.	Başlangıç şartlarına hassas bağlılık
IDCS	Tam sayılar kümesinde tanımlı kaotik sistem (ing. integer domain chaotic system)
LSB	En az anlamlı bit (ing. least significant bit)
MSB	En çok anlamlı bit (ing. most significant bit)
İKKH	İkili kaydırmalı kaotik harita
*n*B	*n*-boyutlu
RSÜ	Rastgele sayı üreteci
SRSÜ	Sözde rastgele sayı üreteci
GRSÜ	Gerçek rastgele sayı üreteci
ÇBBH	Çok boyutlu baker haritası
ODE	Adi diferansiyel denklem (ing. ordinary differential equation)
LFSR	Lineer geribeslemeli kaydırmalı kaydedici (ing. linear feedback shift register)
PLL	Faz kilitlemeli döngü (ing. phase locked loop)
obeb	Ortak bölenlerin en büyüğü
HDL	Donanım tanımlama dili (ing. hardware description language)
VHDL	İng. very high speed integrated circuit HDL
RTL	İng. register transfer level
D/A	Dijital analog
CH1	Kanal 1
CH2	Kanal 2
MHz	Megahertz
kş.	-e karşı
EKS	Eşleştirilmiş kaotik sistem
M/D	Mevcut değil
NIST	Ulusal standartlar ve teknoloji enstitüsü (ABD)
G-LFSR	Genelleştirilmiş LFSR

M-İKKH	Modifiye İKKH
Mbps	Saniye başına megabit

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

GİRİŞ

Kaos fenomeni keşfedilmeden önce dinamik sistemlerin bilinen davranış şekilleri şu şekilde sıralanmaktaydı: (i) sabit duruma yakınsama; (ii) osilasyon (periyodik davranış); (iii) yarı-periyodik davranış (düzensiz periyotlarla osilasyon). Kaos dinamik sistemlerin en son keşfedilmiş olan davranış türüdür. Lorenz tarafından kaos fenomeni hakkında doğrudan bu konuya odaklı yayınlanan ilk makale (daha "kaos" ifadesi bu davranışı adlandırmak için kullanılmadan önce) "deterministik periyodik olmayan akış" (ing. "deterministic nonperiodic flow") olarak adlandırılmıştı [1]. Bu başlık kaotik davranış gösteren sistemlerin özelliklerinin çoğunu iyi bir şekilde özetlemektedir. Buna göre, kaotik sistemlerin çıkışı öncelikle periyodik olmamalıdır. Periyodik olmayan kaotik çıkışlar bu özelliklerinden dolayı rastgele bir işarete benzemektedir. Kaotik çıkışların (diğer bir adıyla yörüngelerin), rastgele yerine rastgele benzeri olarak nitelenmesinin nedeni de yine bu başlıkta saklıdır. Normalde rastgele işaretlerin hiçbir formül ya da bağıntı tarafından öngörülememesi gerekmektedir. Fakat kaotik yörüngeler deterministik olarak yani belli matematiksel eşitlikler ile tanımlanmaktadır. Bu bağlamda, kaotik yörüngeler rastgele sinyaller gibi periyodik olmasalar da rastgele sinyallerin aksine, kaotik yörüngelerin sonraki durumları mevcut durumdan yola çıkarak tamamen tahmin edilebilir yapıdadır.

Bu özellikler dışında, kaosun diğer karakteristik özellikleri başlangıç şartlarına hassas bağlılık ve pozitif Lyapunov üstelleri olarak adlandırılmaktadır [2]. Başlangıç şartlarına hassas bağlılık (b.ş.h.b.) popüler kültürde "kelebek etkisi" adıyla şöhret kazanmıştır. Basitçe, bu özellik birbirine çok yakın iki yörüngenin belli bir süre birbirinden oldukça uzaklaşacağı veya yakın başlangıç şartları için kaotik sistemlerin çıkışlarının bambaşka olacağı anlamına gelmektedir. Lyapunov üstelleri ise yörüngeler arasındaki ıraksama miktarının matematiksel bir ölçüsü olup her kaotik sistemin en az bir pozitif Lyapunov üsteline sahip olması beklenmektedir [2]. Tabi bunlar dışında kaotik sistemlerin sahip

olması gereken en önemli özellik (ya da daha doğru bir tabirle gereksinim) ise nonlineerliktir.

Davranışı birtakım eşitliklerle belirlenebilen deterministik yapıya sahip olmasına rağmen, periyodik olmayan rastgele benzeri çıkışlara sahip kaotik sistemlerin ilk akla gelen uygulama alanları doğal olarak rastgele sayı üretimi ve bilgi güvenliğidir. Yapılan çalışmalar, kaosun b.ş.h.b, ergodiklik (benzer istatistiki özelliklere sahip olma) gibi özelliklerinin kriptografide bulunan karıştırma (confusion) ve yayılma (diffusion) gereksinimlerini birebir karşıladığını ortaya koymuştur [3, 4]. Dolayısıyla, araştırmacıların Lorenz ile ciddi bir araştırma konusu haline gelmiş olan kaos teorisi ile kriptografi arasındaki yakın ilişkiyi keşfetmesi çok uzun sürmemiştir [5–7]. Böylelikle, "kaos tabanlı kriptografi" adı verilen yeni bir çalışma alanı doğmuştur [8].

Günümüzde, rastgele sayı üreteçleri, akış ve blok şifreleyici, kriptografik özet (hash) fonksiyonu, anahtar paylaşma protokolü, fiziksel klonlanamaz fonksiyon vb. yapılarda kullanılmak üzere pek çok kaos tabanlı kriptosistem önerilmiştir [9–16]. Kaos tabanlı kriptosistemlerin ilk kullanım alanı analog güvenli haberleşme olmuştur. Bu tür uygulamalar, Leon Chua tarafından sürekli zamanlı kaotik sistemlerin analog devre elemanları ile gerçekleştirilebileceklerinin ve ardından Carroll ile Pecora tarafından sürekli zamanlı iki kaotik sistemin çıkışının senkronize edilebileceğinin gösterilmesinden sonra yaygınlaşmaya başlamıştır [17–19]. Analog güvenli haberleşmede sürekli zamanlı iki kaotik sistem analog elektronik devreler ile gerçekleştirilir. Ardından, devrelerden birinin çıkışı analog haberleşmede kullanılan analog sinyaller üzerine bindirilerek bu sinyallerinin üçüncü kişiler tarafından ele geçirildiğinde bu kişilerin gürültüden farksız bir sinyalle karşılaşmaları sağlanır. Haberleşme kanalı üzerinden bu birleştirilmiş sinyallerin yanı sıra bir senkronizasyon sinyali de gönderilir ve bu senkronizasyon sinyali alıcıda bulunan ikinci kaotik devreyi senkronize etmek için kullanılır. Alıcı kısmında senkronizasyon işlemi yapıldıktan sonra aynı kaotik işaret üretilir ve şifrelenmiş analog sinyalden bu senkronize olmuş sinyal çıkarıldığında geriye orijinal mesaj kalır ve alıcı mesaja güvenli bir şekilde ulaşmış olur [6]. Kaotik maskeleme olarak adlandırılan bu metoda sonraki çalışmalarda daha farklı eklemeler yapılsa da kaos tabanlı güvenli analog haberleşmenin temel prensibi bu şekildedir [20–24]. Fakat, sonraki yıllarda analog kaotik haberleşme hakkında önerilen bu tür yapıların çoğunun kriptografik açıdan güvenli olmadığı gösterilmiştir [25–30]. Bu nedenle, günümüzde kaos tabanlı analog haberleşme

hakkında yapılan yayınlar doksanlı yıllara göre epey azalmış ve kaos tabanlı kriptografi hakkındaki çalışmalar daha çok kaotik sistemlerin dijital olarak kriptografik uygulamalarda kullanılmasına yoğunlaşmıştır.

Kaotik sistemlerin dijital olarak kriptografide kullanımına yönelik ilk tasarım Matthews tarafından önerilmiştir [5]. Önerilen bu metotta, kaos tabanlı analog haberleşmede olduğunun aksine ayrık zamanlı kaotik haritalar (ing. map) kullanılmıştır. Kriptosistem kayan noktalı aritmetik ile kaotik sistemin yörüngelerinin hesaplanmasına ve hesaplanan yörüngeleri kullanarak modulo gibi basit aritmetik işlemlerle açık metnin şifrelenmesi ve deşifrelenmesi prensibine göre çalışmaktadır. Anahtar olarak ise, kaotik haritanın başlangıç şartı kullanılmıştır. Kaotik sistemlerin b.ş.h.b. özelliği yüzünden anahtar çok az değişikliğe uğrasa bile farklı yörüngelerin üretileceği düşünülerek sistem güvenli olarak kabul edilmiştir. Fakat, hemen sonra Wheeler tarafından gösterildiği üzere önerilen kriptosistem güvenli değildir [31]. Wheeler'ın gösterdiği üzere dijital sistemlerdeki sonlu çözünürlük üretilen yörüngelerin periyodik olmasına neden olmakta ve önerilen kriptosistemin kısa periyotları ise kriptosistemin anahtar olmaksızın kolayca kırılabilmesine neden olmaktadır. Sonraki çalışmalar bu kısa periyotların uzatılmasına yönelik mühendislik çözümleri olmuştur [32]. Önerilen ilk gelen çözüm bit çözünürlüğünü arttırmak olmuştur [33]. Fakat, kayan noktalı aritmetik kullanıldığında sadece iki genel standart bulunmakta ve sabit noktalı aritmetikte ise bit çözünürlüğü ile performans ve kaynak kullanımı arasında ters bir orantı bulunmaktadır. Dolayısıyla, aynı sorunu gidermek üzere kaotik sistemlerin art arda bağlandığı kaskatlı yapılar önerilmiştir [34]. Bu metotta da aynı işi yapmak için çok sayıda kaotik sistemin kullanılması gerekliliği gerçekleştirim bakımından verimsiz olup, farklı kaotik sistemlerin art arda bağlanmasında sonraki sistemin istenmeyen dinamikler sergileme ihtimali bulunmaktadır. Bu nedenle, kaotik sisteme belli aralıklarla gürültü enjekte etme prensibine dayalı pertürbasyon metodu geliştirilmiş olup, önerilen bu metotlar arasında en iyisi olduğu gösterilmiştir [32, 35, 36]. Öztürk ve Kılıç tarafından yapılan bir çalışmada ise sistem parametrelerinin anahtarlanması suretiyle aynı sorun giderilmeye çalışılmış ve pertürbasyon metoduna bir alternatif olarak sunulmuştur [37]. Fakat, yine Öztürk ve Kılıç tarafından gösterildiği üzere kısa periyotlar kaotik sistemlerin dijital gerçekleştirimlerindeki tek problem değildir [38]. Bu çalışmada gösterildiği üzere, üretilen yörüngeler belli bir transient kısmından sonra kendini tekrar eden döngülere

düşmektedir. Fakat, yörüngeler arasındaki kesişimler yüzünden farklı başlangıç şartları için bu periyodik döngülerin çoğu birbirinin aynısı olmaktadır. Bu ise b.ş.h.b. özelliğine tamamen karşı olup, bazı kaos tabanlı kriptosistemlerde anahtar bilinmeksizin orijinal verinin elde edilmesine neden olmaktadır [38]. İstenmeyen bu durumun önüne geçebilmek adına ya pertürbasyon metodunun gürültü ekleme aralığı düşük tutulmalı ya da [37]'deki parametre anahtarlama metodu kullanılmalıdır.

Bu çalışmaların göstermiş olduğu üzere, konvansiyonel sabit veya kayan noktalı dijital gerçekleştirimler ile üretilen yörüngeler, periyodik oldukları için kesinlikle kaotik olarak adlandırılamazlar. Dahası, hesaplamalar sırasında meydana gelen yuvarlama hataları ve sonlu çözünürlük yüzünden, üretilen yörüngeler orijinal matematiksel modelin yörüngeleri değildir. Bu nedenle, bu yolla üretilen yörüngeler [32]'de sözde yörüngeler olarak adlandırılmaktadır. Ayrıca, kaosun kriptografide kullanılmasında rol oynayan b.ş.h.b. özelliği dijital gerçekleştirimde uzun vadede korunmazken, diğer gerçekleştirimlerin ne derece korunduğunu inceleyen araştırmalar da bulunmamaktadır. Bu tür araştırmaların yapılamamasının sebebi ise yuvarlama hatalarının takibinin mümkün olmamasından kaynaklanmaktadır. Bunlar dışında, dijital gerçekleştirim ile orijinal matematiksel model arasındaki en temel fark tanım kümelerinin değişik olmasıdır. Orijinal kaotik model sonsuz sayı içeren reel sayılar kümesinde tanımlanırken, dijital platformlarda sadece sonlu bitle ifade edilen sonlu çoklukta sayılar mevcuttur. Dahası, sayılar sonlu bitlerle ifade edildiği için dijital platformlarda reel sayıların hiçbiri yoktur. Bu ise neden kaotik sistemlerin konvansiyonel sabit veya kayan noktalı aritmetik ile gerçekleştirilemeyeceğinin en önemli göstergesidir.

Literatüre sunulmuş olan pek çok kaos tabanlı kriptosistem yine reel sayılar kümesi baz alınarak tanımlanmıştır. Bu bakımdan, bu tür kriptosistemler gerçekleştirilmesi mümkün olmayan marjinal metotlar olarak görülmektedirler [8]. Maalesef, araştırmacıların çoğu bu durumu ihmal etmekte; dijital platformlarda kaosun özelliklerinin aynen korunduğunu ve üretilen yörüngelerin de kaotik yörüngeler olduğunu varsayan çalışmalar yapmaktadırlar [39–44]. Bu tür çalışmaların sonraki analizlerinde ise sıklıkla güvenli olmadıkları ortaya çıkmaktadır [4, 45–49]. Önerilen kriptosistemde, üretilen yörüngelerin periyodik sözde yörüngeler olduğu bilinse ve yukarıda bahsedilen metotlarla kısa periyotlar, yuvarlama hatalarının etkileri vb. dezavantajlar giderilse bile, bu tür kriptosistemler yine de klasik kriptografide kullanılmak için cazip yaklaşımlar değildirler.

Bunun en temel nedeni, klasik kriptografinin tam sayılara ve basit bitsel işlemlere dayanmasıdır. Bu bakımdan, reel sayılar kümesinde tanımlı kaotik sistemlerin sabit veya kayan noktalı aritmetik ile gerçekleştirimlerinin klasik kriptografide hiçbir karşılığı yoktur. Ayrıca başka problemler de vardır ve bunlar şu şekilde sıralanabilir: (i) Üretilen sözde yörüngelerin periyot uzunluğunun ne olduğu belli değildir; (ii) Orijinal matematiksel model ile sonlu çözünürlük ve yuvarlama hataları altında yapılan gerçekleştirim arasındaki ilişki net bir şekilde belli değildir; (iii) Aynı gerçekleştirim platforma ve yuvarlama türüne göre değişiklik gösterebilir ve bu ise şifreleme / deşifreleme işleminin farklı platformlarda farklı sonuçlar vermesine neden olabilir; (iv) Sabit ve kayan noktalı aritmetik işlemlerin donanım ile gerçekleştirimi zordur.

Bahsedilen bu nedenler yüzünden kaotik sistemlerin dijital olarak gerçekleştirimlerinde sabit ve kayan noktalı gerçekleştirimler dışında yeni yaklaşımlara ihtiyaç vardır. Şimdiye kadar, literatürde bunu sağlayabilen tek bir metot önerilmiştir. Önerilen bu metot tam sayılar kümesinde tanımlı kaotik sistemlere (IDCS) dayanmaktadır [50, 51]. Basitçe özetlemek gerekirse IDCS'ler bir veya daha fazla rastgele diziyi giriş olarak alır ve ardından bunları bitsel lojik işlemlerle karıştırırlar. IDCS'lerin kaotik olarak adlandırılmasının nedeni ise kaotik özellik gösterdiklerinin Devaney'in kaos tanımına göre ispatlanabilmesidir [50, 52]. Şimdiye kadar farklı türlerde fakat sınırlı diyebileceğimiz sayıda IDCS'ler önerilmiştir [53–55]. IDCS'ler hakkında yapılan çalışmaların en önemlisi [51]'de yapılmış olup, bu çalışmada her boyut için genelleştirilmiş bir IDCS önerilmiş ve bunun dijital gerçekleştirimleri farklı boyutlar için yapılmıştır.

Bu tez çalışmasında ise kaotik sistemlerin dijital gerçekleştirimlerinde kullanılmak üzere konvansiyonel sabit veya kayan noktalı aritmetiğe bağlı olmayan ve basit bitsel işlemlerle gerçekleştirilebilecek yeni tasarım yaklaşımlarının geliştirilmesi amaçlanmaktadır. Bu kapsamda, IDCS yaklaşımından farklı olarak, literatürde mevcut olan klasik kaotik sistemlerin bazılarının tamamen dijital olarak gerçekleştirilebilmesini mümkün kılan ve bu tür sistemlerin gerçek (hatasız) kaotik ve gerçek periyodik yörüngelerinin üretilmesini sağlayan yeni bir metot geliştirilmiştir. Bu geliştirilen metot "LSB (en az anlamlı bit) uzatma metodu" olarak adlandırılmıştır [56]. LSB uzatma metodu ancak ikili kaydırmalı kaotik haritalar (İKKH) ile kullanılabilmekte olup İKKH'lar yalnızca kaydırma, NOT, XOR gibi basit lojik işlemlerle gerçekleştirilebilen literatürdeki mevcut haritaların özel

bir türüdür. Bunların "ikili kaydırmalı" olarak adlandırılmalarının nedeni ise tüm parametre değerlerinin ikinin katı olması ve bu tür çarpımların ikili kaydırmalar ile ifade edilebilmesi yüzündendir. Bernoulli, tent ve baker haritaları İKKH'lara birer örnektir [56]. Bu tür haritaların konvansiyonel sabit veya kayan noktalı gerçekleştirimleri bu ikili kaydırmalar yüzünden daima sıfıra yakınsamaktadır [57]. LSB uzatma metodunda sabit veya kayan noktalı aritmetik kullanılmaz. Sabit bit uzunluğunda bir hafıza birimi alınır ve İKKH'ya karşılık gelen bitsel lojik işlemler bu hafıza birimine uygulanır. Bu işlemler sonucunda hafıza biriminin LSB bitlerinden bazıları ikili kaydırmalar yüzünden sıfır olur. LSB uzatma metodunda ise bu sıfır ile dolan LSB bitleri rastgele bitler ile doldurularak bir nevi çözünürlük uzatma işlemi yapılmış olunur. LSB uzatma adı da yapılan bu işlemlerden gelmektedir. Bu sayede üretilen yörüngeler asla sıfıra yakınsamazlar ve yuvarlama hataları mevcut olmadığı için orijinal matematiksel modele ait yörüngeler elde edilir. Sonraki bölümlerde gösterileceği üzere, eğer bitler bir sözde rastgele sayı üretecinden (SRSÜ) elde ediliyorsa, üretilen yörüngeler kaotik sistemin gerçek periyodik yörüngeleri; eğer bitler bir gerçek rastgele sayı üretecinden (GRSÜ) elde ediliyorsa, üretilen yörüngeler kaotik sistemin gerçek kaotik yörüngeleri olmaktadır. LSB uzatma metodunun diğer bir önemli özelliği ise İKKH'lar dışındaki diğer kaotik haritaların yörüngelerinin de topolojik konjuge fonksiyonları yardımıyla elde edilebilmesidir. Böyle bir ilişki kullanılarak lojistik harita adı verilen kaotik sistemin gerçek yörüngeleri elde edilmiştir. Lojistik haritanın literatürde en çok çalışılan kaotik sistemlerden birisi olmasına karşın tamamen dijital bir gerçekleştirim ile gerçek yörüngelerinin elde edilmesi bildiğimiz kadarıyla literatürde bir ilktir.

Tezin sonraki kısımlarında ise normalde 2-boyutlu (2B) olan baker haritasının tüm boyutlar için genelleştirilmesi yapılmıştır. Bunun yapılma nedeni [51]'de her boyut için genelleştirilmiş olan IDCS'ye karşılık, her boyut için genelleştirilmiş bir İKKH önererek LSB uzatma metodunu IDCS metoduna karşı bir alternatif olarak sunabilmektir. Bu amaçla, 2B baker haritası her boyut için genelleştirilmiş ve önerilen bu genelleştirilmiş harita "çok boyutlu baker haritası" (ÇBBH) olarak adlandırılmıştır. ÇBBH'nın her boyut için kaotik olduğu Devaney'in kaos tanımına göre matematiksel olarak ispatlanmıştır. Dahası, bu haritanın her boyut için sadece iki temel yapı bloğu kullanılarak dijital olarak gerçekleştirilebileceği gösterilmiştir. Bu temel bloklar kullanılarak ÇBBH'nın farklı

boyutlar için hem yazılımsal olarak hem de FPGA (Alan Programlanabilir Kapı Dizileri) donanımı üzerinde gerçekleştirimleri yapılmıştır.

Son olarak ise LSB uzatma metodu ile üretilen gerçek periyodik ve gerçek kaotik yörüngelerin kaos tabanlı kriptografide kullanımı hakkında kıyaslamalar yapılmış ve bu tez çalışmasında üretilen gerçek periyodik yörüngelerin kullanılması önerilmiştir. Fakat, yapılan incelemelerde LSB uzatma metodu ile üretilen gerçek periyodik yörüngelerin doğrudan kullanılabilmesinin mümkün olmadığı fark edilmiş ve bu tür yörüngeleri kriptografide kullanabilmek için yeni tasarımlara ihtiyaç olduğu görülmüştür. Bu amaçla, tezin son bölümünde LSB uzatma metodu ile tamamen dijital olarak üretilen gerçek periyodik yörüngelerin kaos tabanlı kriptografide kullanılabilmesi için yeni bir tasarım önerilmiştir. Önerilen bu tasarımın kriptografik açıdan güvenli bir SRSÜ veya akış şifreleyici olarak kullanılabileceği istatistiksel testler yardımıyla gösterilmiştir. Bu yapının donanım gerçekleştirimi de yine FPGA üzerinde yapılmıştır. Bu çalışma, literatürde daha önce önerilmiş fakat sözde yörüngeler kullanıldığı için güvenli olmayan diğer kaos tabanlı kriptosistem tasarımlarının güvenli hale getirilebilmesi için temel teşkil etmektedir.

Tezin organizasyonu şu şekildedir: Birinci bölümde tez çalışmasında kullanılan temel kavramlar kısaca açıklanmakta ve kaos tabanlı kriptografide kullanılan temel yapıların detayları verilmektedir. Dinamik sistemler ve kaos hakkındaki bazı temel matematiksel tanımlar ve teoremler de yine bu bölümde verilmektedir. Son olarak, bu tez çalışması sırasında keşfedilen yeni bir sürekli zamanlı kaotik sistemin analizi ve elektronik devre gerçekleştirimi de kaotik sistemlere ve verilen kavramların kullanımına örnek teşkil etmesi amacıyla bu bölümde sunulmaktadır. İkinci bölümde ise LSB uzatma metodu ve İKKH'lar tanıtılmaktadır. Yine bu bölümde, LSB uzatma metodu ile bazı İKKH'lar ve bunların topolojik konjugelerine ait gerçek periyodik ve gerçek kaotik yörüngelerinin yazılımsal olarak ve FPGA donanımı üzerinde nasıl elde edileceği gösterilmektedir. ÇBBH'nın genelleştirilmesi ve önerilen bu haritanın tüm boyutlar için kaotik olduğunun matematiksel ispatı üçüncü bölümde verilmektedir. Aynı bölüm içerisinde, ÇBBH'nın her boyut için iki temel yapı bloğu kullanılarak gerçekleştirilebileceği gösterilmekte ve bu temel bloklar ile yazılımsal ve donanımsal gerçekleştirimlerin nasıl yapılacağı gösterilmektedir. Dördüncü bölümde LSB uzatma metodu ile üretilen gerçek periyodik yörüngelerin kaos tabanlı kriptografide kullanılabilmesi için yeni bir tasarım yaklaşımı

önerilmektedir. Son olarak, tez çalışmasından çıkarılan sonuçlar ve elde edilen sonuçlar hakkındaki değerlendirmeler beşinci bölümde sunulmakta ve gelecekte yapılacak çalışmalar hakkında öngörüler de yine aynı bölümde verilmektedir.

1. BÖLÜM

TEMEL KAVRAMLAR VE TANIMLAR

1.1. Dinamik Sistemler ve Kaos

1.1.1. Bir Boyutlu Haritalar

Bu tez çalışmasında, ayrık zamanlı dinamik sistemler, İngilizce "map" ifadesinin Türkçe karşılığı olan "harita" şeklinde adlandırılmaktadır. Türkçe literatürde bu ifade aynı zamanda "fonksiyon" veya "dönüşüm" olarak da çevrilmektedir. Fakat, "map" denilen sistemler alelade fonksiyonlar veya dönüşümler olmayıp özel olarak tanım ve hedef kümesi aynı olan fonksiyonlardır. Bu bağlamda, bu tez çalışmasında, anlam kaybına mahal vermemek adına bu tür yapılara "harita" denilmektedir. Buna göre, haritalar Tanım 1.1.1'deki gibi tanımlanmaktadırlar.

Tanım 1.1.1. Genel olarak tanım kümesi hedef kümesi ile aynı olan; $A \subset \mathbb{R}^n$ olmak üzere $f: A \to A$ şeklindeki ayrık zamanlı dönüşümlere özel olarak **harita** adı verilmektedir.

Bir dinamik sistem bir kurala bağlı olarak şimdiki durumu önceki durum cinsinden ifade eden ve olası tüm durumların kümesini içeren matematiksel bir modeldir. Ayrık zamanlı dinamik sistemler $x_n = f(x_{n-1})$ şeklindeki haritalarla ifade edilmektedir. Burada x_n şimdiki durumu ifade ederken; $f(x_{n-1})$ önceki durumlara bağlı kuralı temsil etmektedir.

Tanım 1.1.2. $f: A \to A$ olmak üzere $\{x, f(x), f^2(x), \dots, f^k(x), \dots\}$ kümesine x noktasının *f* haritası altındaki **yörünge**si denilmekte ve x noktası **başlangıç şartı** olarak adlandırılmaktadır. $f(p) = p$ şartını sağlayan noktalara ise **sabit nokta** adı verilmektedir.

Tanım 1.1.3. $f: A \to A$ olmak üzere $f^k(p_i) = p_i$ şartını sağlayan k noktanın her biri **periyot-k noktası** olarak adlandırılmaktadır. Başlangıç şartı p_i olan yörüngeler ise **periyot-k yörünge** olarak adlandırılmaktadır.

Tanım 1.1.3'ten görülebileceği üzere sabit noktalar aynı zamanda periyot-1 nokta olarak da adlandırılabilir.

Teorem 1.1.4. x bir sabit nokta olmak üzere $|f'(x)| < 1$ için x kararlı ve $|f'(x)| > 1$ için x kararsızdır. Bir periyot-k yörünge içinse $\left|\prod_{i=1}^{k} f'(x_i)\right| < 1$ için yörünge kararlı; $\left|\prod_{i=1}^{k} f'(x_i)\right| > 1$ için yörünge kararsızdır [2].

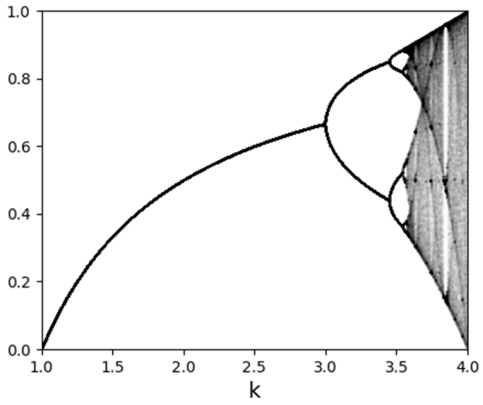
Kararlı periyot-k yörüngeler yakın yörüngeleri çekerken kararsız yörüngeler yakın yörüngeleri iterler.

Tanım 1.1.5. Kararlı bir sabit veya periyodik noktaya yakınsayan tüm başlangıç şartlarının kümesine **çekim havzası** adı verilmektedir.

Literatürde en çok incelenen haritalardan biri olan "lojistik harita" bir boyutlu haritalara verilebilecek en iyi örneklerden biridir. Lojistik harita $G: [0,1] \rightarrow [0,1]$ Eşitlik (1.1)'deki gibi tanımlanmaktadır.

$$G(x) = kx(1-x) \tag{1.1}$$

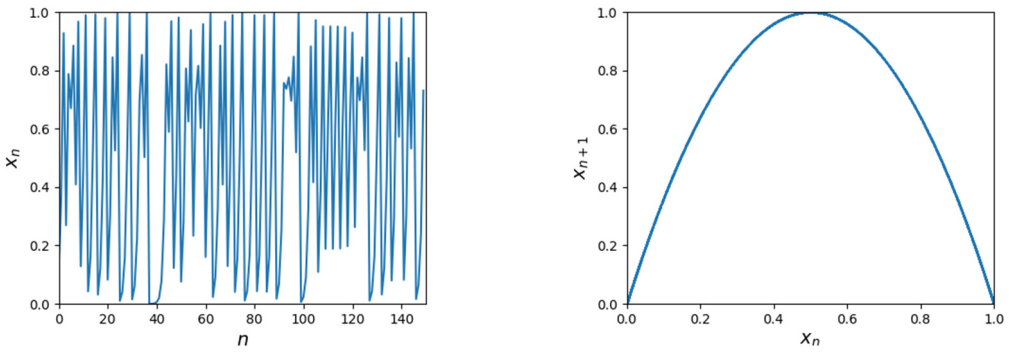
Lojistik harita farklı parametre değerleri için farklı davranışlar sergilemektedir. Dinamik sistemlerin değişen parametrelere karşı değişen özellikleri dallanma (bifürkasyon) diyagramları ile incelenir. Lojistik harita için k parametresinin değişimine göre elde edilen dallanma diyagramı Şekil 1.1'de görülmektedir.



Şekil 1.1 Lojistik haritanın dallanma diyagramı

Şekil 1.1'deki dallanma diyagramında her k değeri için elde edilen yörüngelerin almış olduğu değerler çizilmektedir. Buna göre, $k \in [1,3]$ için lojistik haritanın yörüngeleri tek bir sabit noktaya yakınsamaktadır. $k = 3$ değerinden sonra ise yörüngeler kararlı bir periyot-2 yörüngeye yakınsamakta; daha sonra sırasıyla periyot-4, periyot-8 derken periyotlar bir anda artmaktadır. Bu olaya periyot katlama dallanması adı verilmektedir [2]. Periyotların ani bir şekilde arttığı ve yörüngelerin tüm aralıkları doldurduğu bölgeler haritanın kaotik olduğu bölgelerdir. Dikkat edilirse, kaotik bölgeler arasında boşluğa neden olan periyodik davranışlar bulunmaktadır. Kaotik bölgeler arasındaki bu tür periyodik bölgelere "periyodik pencere" adı verilmektedir. Kaotik davranışın elde edildiği periyot katlama dallanması dışında çok daha farklı dallanma çeşitleri mevcuttur [2].

Dallanma diyagramından görülebileceği üzere lojistik harita $k = 4$ için kaotiktir. Bu parametre değeri için lojistik haritaya ait çeker yapısı ve yörüngelerden biri Şekil 1.2'de verilmektedir. Çeker dediğimiz şekiller yörünge değerlerinin faz uzayındaki çizimleridir. Kararlı bir kaotik sistemin çekim havzasındaki tüm başlangıç şartları bu yapıya yakınsamaktadır. İleriki kısımlarda gösterilmekte olduğu üzere her kaotik sistemin farklı bir çeker yapısı olup bunlar genelde fraktal boyutludur.



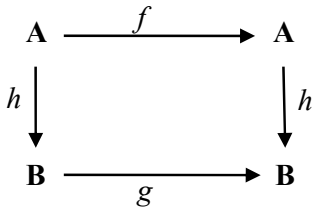
(a) (b)

Şekil 1.2 Lojistik haritaya ait (a) bir yörünge; (b) çeker yapısı

1.1.2. Homeomorfizm ve Sembolik Dinamikler

Tanım 1.1.6. (X, T_X) ve (Y, T_Y) topolojik uzayları arasındaki $f: X \to Y$ şeklindeki bir fonksiyon: (i) birebir ve örten; (ii) sürekli ve (iii) tersi de sürekli ise **homeomorfizm** olarak adlandırılır.

$f: A \to A$ ve $g: B \to B$ haritaları arasında $h: A \to B$ şeklinde bir homeomorfizm kurulabiliyorsa f *ve* g haritaları **topolojik konjuge** haritalar; h fonksiyonu ise **konjuge fonksiyonu** olarak adlandırılır. Bu homeomorfizm neticesinde dönüşümler arasında ortaya çıkan ilişki Şekil 1.3'teki gibi özetlenebilir.



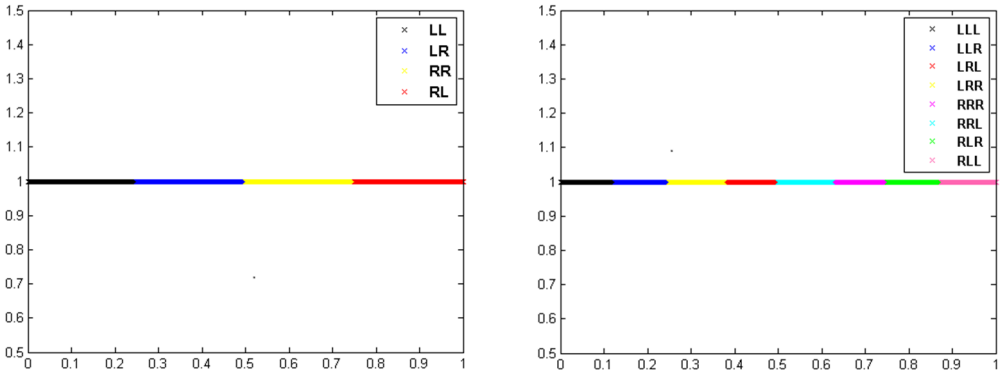
Şekil 1.3 Konjuge fonksiyonlar arasındaki ilişki

Dolayısıyla, $f(x) = (h^{-1} \circ g \circ h)(x)$ ve genelleştirirsek $f^k(x) = (h^{-1} \circ g^k \circ h)(x)$ olduğunu görürüz. Buna göre, eğer $f^k(x)$ periyodikse $g^k(x)$ de periyodiktir; eğer $f^k(x)$ kararlıysa $g^k(x)$ de kararlıdır vs. Yani, topolojik özellikler korunur. Bu şekilde bir ilişki karmaşık bir haritayı daha basit bir haritayı ele alarak incelemeye imkan vermenin yanı sıra mevcut dinamikleri daha soyut bir uzayda sembolik dinamikler ile ele almayı da mümkün kılmaktadır.

Sembolik dinamikleri anlamak için Eşitlik (1.2) verilen ve $T: [0,1] \to [0,1]$ şeklinde ifade edilen çadır haritasını (tent map) ele alalım.

$$T(x) = \begin{cases} 2x, & x \in \left[0, \frac{1}{2}\right] \\ 2(1-x), & x \in \left[\frac{1}{2}, 1\right] \end{cases} \qquad (1.2)$$

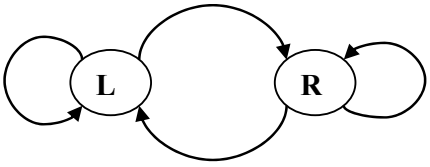
Eğer $x \in [0,1/2]$ ise bu noktayı L sembolü ile ve $x \in [1/2,1]$ ise bu noktayı R sembolü ile ifade edebiliriz. Buna göre $\{T^k(x)\}$ yörüngesi $s_i \in \{L, R\}$ olmak üzere $s_1 s_2 \dots s_k$ şeklinde sembolik bir güzergahla temsil edilebilir. Tabi bu güzergah başka x değerleri için de aynı olacaktır. Şekil 1.4'te farklı başlangıç şartları için hesaplama yapıldığında sembolik güzergahların tanım kümesi üzerinde kapladıkları yerler ve boyutları görünmektedir.



(a) (b)

Şekil 1.4 (a) $k = 2$ için sembolik dinamiklerin kapladıkları yerler; (b) $k = 3$ için sembolik dinamiklerin kapladıkları yerler

Buna göre $s_1 s_2 \dots s_k$ ile ifade edilebilen tüm başlangıç şartları tanım kümesi üzerinde uzunluğu $1/2^k$ ile ifade edilen aralıklara karşılık gelmektedir. Bu da $k \to \infty$ için sembolik güzergahların uzunluğu sıfır olan tek bir noktaya karşılık gelebileceğini göstermektedir. Olası güzergahlar incelendiğinde ise Şekil 1.5'teki gibi bir geçiş diyagramının mümkün olduğu görülmektedir. Buna göre her türlü sembolik güzergah kombinasyonu mümkün olup; her farklı kombinasyon $[0,1]$ üzerinde farklı bir noktaya karşılık gelmektedir.



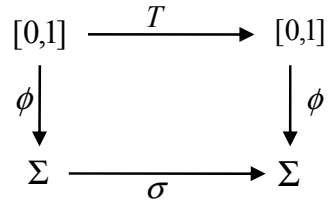
Şekil 1.5 Çadır haritası için geçiş diyagramı

Geçiş diyagramına göre oluşturulabilecek olası tüm sembolik güzergahların kümesi Σ olsun. Buna göre $\phi: [0,1] \to \Sigma$ şeklinde birim aralıktaki her noktayı karşılık gelen sembolik güzergaha gönderen bir ϕ fonksiyonu bulunabilir.

Tanım 1.1.7. $\sigma: \Sigma \to \Sigma$ ve $s_i \in L, R$ olmak üzere $\sigma(s_1 s_2 \dots s_k \dots) = s_2 \dots s_k \dots$ fonksiyonuna **kaydırma fonksiyonu** adı verilir.

Sembolik güzergahlara uygulanan kaydırma fonksiyonunun birim aralıkta T haritasına karşılık geldiği aşikardır. Bu da akla aslında ϕ fonksiyonunun birim aralık ile Σ kümeleri arasında bir homeomorfizm bulunduğu fikrini getirir. Nitekim Şekil 1.6 incelendiğinde T

ile σ'nin konjuge fonksiyonlar olduğu görülür. Buna göre, çadır haritasını incelemek için kaydırma fonksiyonunun Σ kümesi üzerindeki davranışını incelemek yeterlidir.



Şekil 1.6 Birim aralık ve Σ arasındaki homeomorfizm

Dolayısıyla sembolik dinamiklerin incelenmesi söz konusu sistem hakkında pek çok bilgi vermektedir. Örneğin $LRLRLRLR \ldots \equiv \overline{LR}$ şeklindeki bir güzergah $\sigma^2(\overline{LR}) = \overline{LR}$ olduğu için periyot-2 bir noktanın varlığına işaret etmektedir.

1.1.3. Başlangıç Şartlarına Hassas Bağlılık ve Periyodik Yörüngeler

Tanım 1.1.8. $f: A \to A$ dinamik sistemi için $x_0 \in A$ noktasının herhangi bir δ komşuluğunda bulunacak bir x noktası ve pozitif bir k tamsayı değeri için $|f^k(x) - f^k(x_0)| \geq \varepsilon$ şartını sağlayan bir $\varepsilon > 0$ değeri bulunabiliyorsa x_0 noktası başlangıç şartlarına hassas bağlıdır denilir. Eğer her $x_0 \in A$ için yukarıdaki şart sağlanıyorsa f **başlangıç şartlarına hassas bağlı** olarak nitelendirilir.

Bu kısımda örnek olması açısından Eşitlik (1.2) ile verilen çadır haritasının başlangıç şartlarına hassas bağlı olduğu sembolik dinamikler yardımıyla gösterilecektir. Başlangıç şartlarına hassas bağlılık kaotik dinamiklerin bir sonucu olmakla beraber kaotik olmayan dinamiklerde de karşılaşılabilmektedir.

Önerme 1.1.9. Eşitlik (1.2) ile verilen T: $[0,1] \to [0,1]$ çadır haritası başlangıç şartlarına hassas bağlıdır.

İspat: Önceki kısımda gördüğümüz üzere ilk k sembolik güzergahlarının ilk k sembolü aynı olan başlangıç şartları $1/2^k$'lık bir aralık içerisinde bulunur. Buna göre, verilen her δ komşuluğunda ilk k sembolü aynı olan ve $1/2^k < \delta$ şartını sağlayan başka bir x noktası bulunacaktır. x_0'ın güzergahı $s_1 \ldots s_k s_{k+1} s_{k+2} \ldots$ olsun. Bu durumda x noktasının güzergahını

$$s^c = \begin{cases} R, & s = L \\ L, & s = R \end{cases}$$

olmak üzere $s_1 \dots s_k s_{k+1}^c s_{k+2}^c \dots$ şeklinde seçebiliriz. $\varepsilon = 1/4$ olsun. Bu durumda $T^k(x_0)$'ın ilk iki sembolü $T^k(x)$'in ilk iki sembolünün komplementi olacaktır. Şekil 1.4 (a)'da görüldüğü üzere ilk iki sembolü birbirinin komplementi olan noktalar arasında uzunluğu $1/4$ olan başka aralıklar bulunmaktadır. Bu yüzden $|f^k(x) - f^k(x_0)| > 1/4$'dür. ■

Sembolik dinamiklerin yardımıyla bir dinamik sistemin hangi periyodik yörüngelere sahip olduğu belirlenebilir.

Önerme 1.1.10. Çadır haritası T her periyot için periyodik yörüngelere sahip olup bu periyodik yörüngeler kararsızdır.

İspat: Geçiş diyagramından gördüğümüz üzere her sembolik güzergah kombinasyonu mümkündür. Buna göre, her k değeri için $\phi(x) = \overline{s_1 \dots s_k}$ şartını sağlayan bir $x \in [0,1]$ noktası bulunmaktadır. $\sigma^k\big(\phi(x)\big) = \phi(x)$ olduğu için $\phi(x)$ noktası σ fonksiyonu için bir periyot-*k* noktası olacaktır. Homeomorfizm nedeniyle bu T'nin de her periyot için periyodik bir yörüngeye sahip olacağı anlamına gelmektedir. Her $x \in [0,1]\backslash\{1/2\}$ için $|T'(x)| = 2$ olması nedeniyle de Teorem 1.1.4'e göre bu periyodik yörüngeler kararsızdır. ■

Görüldüğü üzere bir kaotik sistemin çıkışlarının tamamı kaotik değildir. Bazı başlangıç şartları için kaotik sistemin çıkışları da periyodik olmaktadır. İleride gösterileceği üzere bir kaotik sistem sayılabilir sonsuzlukta periyodik yörünge içermekte ve bu periyodik yörüngeler de tanım kümesi içinde yoğun olmaktadır. Kaotik yörüngeler ise sayılamaz sonsuzluktadır. Matematikte sayılabilir sonsuzluk sayılamaz sonsuza kıyasla ihmal edilebilir büyüklükte olduğundan ve periyodik yörüngelerin kararsız olmasından dolayı kaotik sistemin periyodik yörüngeleri genellikle dikkate alınmaz. Fakat, bu tez çalışmasında periyodik yörüngelerin eldesi için bir metot geliştirildiğinden ve bu periyodik yörüngelerin kaos tabanlı kriptografide kullanımı önerildiğinden, periyodik yörüngeler üzerinde de durulmaktadır.

1.1.4. Çok Boyutlu Haritalar ve Eyer Noktalarının Manifoldları

Bir boyutlu haritalarda sadece kararlı ve kararsız sabit noktalar varken çok boyutlu haritaları incelediğimizde eyer noktası adı verilen başka bir tür sabit noktanın varlığı ile karşılaşırız. En az bir yönde yakın noktaları çekerken en az bir noktada iten sabit noktalara eyer noktası adı verilir. Lineer bir haritanın yakın noktaları çektiği yön o haritanın özvektörleri yardımıyla bulunur ve bu yönde kararlıdır denilir. Lineer olmayan bir haritanın kararlı veya kararsız olduğu yönler ise sabit noktalar etrafındaki lineerleştirme neticesinde belirlenir.

Tanım 1.1.11. $A \subset \mathbb{R}^n$, $f: A \to A$ ve λ_i lineer dönüşümün özdeğerleri olmak üzere $\lambda_i \neq 1$ şartı sağlanıyorsa f **hiperbolik harita** olarak adlandırılır.

Tanım 1.1.12. $A \subset \mathbb{R}^n$, $f: A \to A$ olmak üzere $\mathbf{p}$ noktası sabit nokta olsun. Eğer $Df(\mathbf{p})$'nin özdeğerlerinin hiçbiri bire eşit değilse $\mathbf{p}$ noktası **hiperbolik** olarak adlandırılır.

Çok boyutlu haritalarda kararlılıktan daha çok eyer noktalarının analizi önem kazanmaktadır. Bir eyer noktasının kararlı ve kararsız yönleri çok boyutlu tanım kümesi üzerinde manifold adı verilen bir alt kümenin ortaya çıkmasına neden olur. Kabaca, n boyutlu bir kümede lokal olarak kümenin $n-k$ elemanı diğer k elemanın bir fonksiyonu şeklinde ifade edilebiliyorsa bu kümeye n boyutlu uzaya gömülü k boyutlu manifold adı verilir [58].

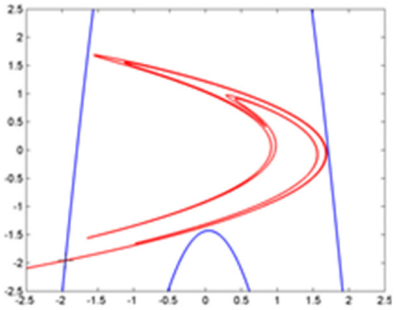
Tanım 1.1.13. Bir $\mathbf{p}$ eyer noktası için kararlı ve kararsız manifoldlar sırasıyla $W^s(\mathbf{p})$ *ve* $W^u(\mathbf{p})$ sembolleriyle ifade edilirler. Buna göre, $W^s(\mathbf{p}) = \mathbf{x}: \lim_{k\to\infty} f^k(\mathbf{x}) = \mathbf{p}$ *ve* $W^u(\mathbf{p}) = \mathbf{x}: \lim_{k\to\infty} f^{-k}(\mathbf{x}) = \mathbf{p}$ şeklinde tanımlanır.

Çok boyutlu kaotik haritalara örnek olarak Eşitlik (1.3) ile ifade edilen Henon haritası verilmektedir.

$$f\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a - x^2 + by \\ x \end{pmatrix} \tag{1.3}$$

$a = 1.28$ ve $b = -0.3$ için [59]'da verilen algoritma yardımıyla elde edilen kararlı ve kararsız manifoldlar Şekil 1.7'de görülmektedir. Şekilde kırmızı ile gösterilen kararsız,

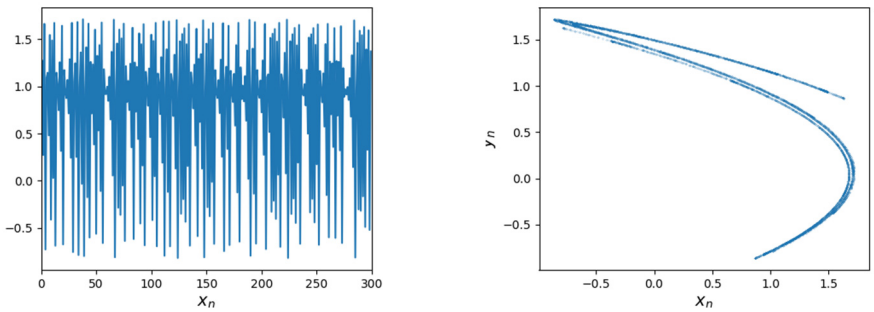
mavi ile gösterilen manifold kararlıdır. Beklenildiği üzere kararlı ve kararsız manifoldlar sadece eyer noktası olan $(-2,-2)$ noktasında kesişirler.



Şekil 1.7 $a = 1.28$ ve $b = -0.3$ için Henon haritasının kararlı ve kararsız manifoldları

Eğer kararlı ve kararsız manifoldlar eyer noktası dışında başka bir noktada kesişiyorsa bu noktalar "homoklinik nokta" olarak adlandırılır. Bu tür bir kesişim ilginç dinamiklerin ortaya çıkmasına neden olur. Tanım gereği homoklinik nokta hem kararlı hem de kararsız manifoldun bir üyesidir. Yani ileri yönde itere edilirken **p** noktasına yakınsarken, geri yönde itere edildiğinde de **p** noktasına yakınsamaktadır. Ayrıca kararlı ve kararsız manifoldlar f altında değişmez oldukları için bu tür noktalar itere edildiklerinde de her iki manifoldun elemanı olmaya devam etmek zorundadırlar. Bu ise tek bir homoklinik noktanın varlığının sonsuz homoklinik noktanın varlığı anlamına gelmesine neden olmaktadır [60]. Bu ise homoklinik noktalar etrafındaki kompleks davranışın durum uzayının büyük bir bölümüne yayılmasına neden olmaktadır [2]. Dahası sonraki kısımlarda gösterildiği üzere homoklinik noktaların varlığı Smale'in at nalı dinamiklerinin ve dolayısıyla kaosun habercisidir [61].

Henon haritasının $a = 2.12$ ve $b = -0.3$ değerleri için x boyutundaki bir yörüngesi ve çekeri Şekil 1.8'de gösterilmektedir.



(a) (b)

Şekil 1.8 Henon haritasına ait (a) x boyutunda bir yörünge; (b) çeker yapısı

1.1.5. Lyapunov Üstelleri

Lyapunov üstelleri bir dinamik sisteme ait birbirine çok yakın iki yörünge arasındaki ıraksama miktarının eksponansiyel bir ölçüsüdür. Pozitif Lyapunov üsteli yakın yörüngelerin ıraksadığı anlamına gelirken, negatif Lyapunov üstelleri yakın yörüngelerin yakınsadığı anlamına gelir. Bu sayı ne kadar büyükse, yakınsama veya ıraksama da o ölçüde büyük olur. Çok boyutlu kaotik sistemde her boyut için bir Lyapunov üsteli hesaplanır ve bir kaotik sistem en az bir adet pozitif Lyapunov üsteline sahip olmalıdır. Birden fazla boyutta pozitif Lyapunov üsteline sahip bir kaotik sistem "hiperkaotik" olarak nitelendirilir. Çok boyutlu kaotik haritalarda en az bir Lyapunov üsteli negatif olmalıdır. Bu da çok boyutlu kaotik sistemlerde bazı yönlerde tanım kümesi esnetilirken bazı yönlerde tanım kümesinin katlandığı anlamına gelir. Lyapunov üstellerinin matematiksel tanımı aşağıdaki gibidir.

Tanım 1.1.14. f haritası $\mathbb{R}^m$ üzerinde tanımlı düzgün (smooth) bir fonksiyon ve N, m-boyutlu bir küre olsun. J_n sembolü, $\mathbf{v}_0$ başlangıç şartı için n-inci iterasyonun Jacobian matrisi olarak ifade edilirse, $J_n = Df^n(\mathbf{v}_0)$ olur. Bu durumda, $J_n N$ m-boyutlu bir elipsoid olacaktır. r_k^n değeri $J_n N$ elipsoidinin k-nıncı ortogonal eksen büyüklüğü olmak üzere, $\mathbf{v}_0$ yörüngesine ait k-nıncı Lyapunov üsteli λ_k (eğer limit mevcutsa) aşağıdaki gibi ifade edilir:

$$\lambda_k = \ln\left(\lim_{n\to\infty}(r_k^n)^{1/n}\right)$$

Çoğu zaman Lyapunov üstelleri için analitik bir değer bulmak imkansızdır. Bu nedenle, Lyapunov üstellerini hesaplamak için bir takım numerik metotlar geliştirilmiştir [62, 63]. Bu tez çalışmasında ÇBBH'nın Lyapunov üstelleri analitik olarak bulunurken, diğer durumlarda numerik metotlara başvurulmuştur.

1.1.6. Kaos

Bilim insanları, 19. yüzyıla kadar atomlar gibi aşırı sayıdaki cisimlerin etkileşimi dışındaki dinamikleri bir denge noktasına sönümleyen ve periyodik veya kuasiperiyodik davranış gösterenler olarak ikiye ayırıyorlardı. 19. yüzyılda Poincare ve Maxwell gibi matematikçilerin çalışmaları bu tür kategorilere sokulamayacak üçüncü bir davranışın olması gerektiğini ortaya koydu. Fakat bu üçüncü davranış çeşidi 20. yüzyılın ortalarına

kadar bilim insanları arasında ilgi görmedi [2]. 1963 yılında ise Lorenz tarafından yapılan bir çalışma ile diferansiyel denklemlerin başlangıç şartlarına hassas bağlı, sınırlı fakat periyodik olmayan davranışlar sergileyebileceği gösterildi [1]. Önerilen diferansiyel denklem sistemi, atmosferik olayları modelleyen karmaşık denklemlerin oldukça sadeleştirilmiş bir haliydi. Buna rağmen, bu sadeleştirilmiş denklemlerin bile bu tür karmaşık dinamikler sergileyebiliyor olması atmosferik tahminlerin uzun vadede bir işe yaramayacak olması anlamına geliyordu. İlk olarak "deterministik periyodik olmayan akış" olarak adlandırılan bu yeni tür davranış, sonradan Yorke tarafından kısaca "kaos" olarak adlandırıldı [64]. Zamanla bu üçüncü tür davranış bilim insanları arasında rağbet görmeye başladı ve çeşitli matematikçiler tarafından matematiksel temelleri ortaya konuldu. Günümüze kadar kaos adı verilen bu dinamik davranışın farklı tanımları ortaya konuldu [52, 64, 65]. Literatürde en çok kullanılan kaos tanımı Devaney tarafından ortaya konulmuş olup Tanım 1.1.15'deki gibidir [52].

Tanım 1.1.15. (X, d, f) metrik bir uzay olmak üzere $f: X \to X$ haritası için, (i) f geçişli (transitive); (ii) f'nin periyodik noktaları X içerisinde yoğun; (iii) f başlangıç şartlarına hassas bağlı ise f **kaotik** olarak adlandırılır.

[66] ve [67]'de gösterildiği üzere, eğer X sonsuz ise (i) ve (ii)'nin varlığı (iii)'ü gerektirmektedir. Bu nedenle, başlangıç şartlarına hassas bağlılık, kelebek etkisi gibi ifadelerle kaosun temel özelliği olsa da, aslında bu tanım için X sonsuz olduğu müddetçe matematiksel olarak gereksizdir [68]. Bu özelliğin çıkarılması bize daha etkin bir kaos tanımı bırakır, çünkü (i) ve (ii) numaralı özellikler topolojik özellikler olup homeomorfizm altında korunurken (iii) numaralı özellik metrik bir özellik olup homeomorfizm altında her zaman korunmaz [66]. Oysaki, önceki kısımlarda gösterildiği üzere kaotik dinamiklerin analizinde homeomorfizm önemli bir yer tutmaktadır.

Sonraki tanımlar Tanım 1.1.15'de kullanılan bazı terimleri açıklamaktadır.

Tanım 1.1.16. $A \subset B$ kümesi her $x \in B$ ve $\varepsilon > 0$ için $N_\varepsilon(x) \cap A \neq \emptyset$ şartını sağlıyorsa A kümesi B kümesi içerisinde **yoğun**dur denilir. Burada $N_\varepsilon(x)$, x noktasının ε komşuluğudur.

Tanım 1.1.17. X'in boş olmayan herhangi iki açık alt kümesi U ve V için $f^k(U) \cap V \neq \emptyset$ şartını sağlayan bir k değeri bulunabiliyorsa f **geçişli**dir denilir.

Bahsetmiş olduğumuz üzere kaosun farklı tanımları da mevcuttur. Mesela, [2]'de kaotik yörüngeler Lyapunov üstelleri kullanılarak aşağıdaki gibi tanımlanmaktadır.

Tanım 1.1.18. $\{\mathbf{v_0}, \mathbf{v_1}, \ldots, \mathbf{v_k}, \ldots\}$ f haritasının sınırlı bir yörüngesi olmak üzere aşağıdaki şartlar sağlanıyorsa yörünge **kaotik**tir denir: (i) Yörünge asimptotik olarak periyodik değil; (ii) hiçbir Lyapunov sayısı tam olarak bire eşit değil; (iii) maksimum Lyapunov sayısı birden büyük [2].

Bir başka kaos tanımı örneği olarak Smale at nalı özelliği gösteren kaotik sistemler gösterilebilir [61]. Smale at nalı davranışı gösteren kaotik sistemlerin sahip olması gereken karakteristikler Teorem 1.1.19'da verilmektedir [69]. Bu teoreme göre bir kaotik sistem sayılabilir sonsuzlukta periyodik yörüngelere sahipken, sayılamaz sonsuzlukta kaotik yörüngelere sahip olmalıdır.

Teorem 1.1.19. Eğer f Smale at nalı davranışına sahipse, f (i) yoğun bir yörüngeye sahiptir; (ii) sayılabilir çoklukta kararsız periyodik yörüngelere sahiptir ve her periyot uzunluğu için bir periyodik yörünge mevcuttur; (iii) sayılamaz çoklukta kaotik yörüngelere sahiptir [69].

1.1.7. Fraktal Çekerler ve Fraktal Boyut

Fraktal kümeler en basit tabiriyle kendine benzer bir yapı gösteren kümelerdir. Fraktaller genel olarak itere fonksiyon sistemleri ile ifade edilirler [70, 71]. İtere fonksiyon sistemleri dolgun (complete) bir metrik uzayda tanımlı büzüşme (contraction) fonksiyonları kümesidir. Bir $f: X \to X$ büzüşme fonksiyonu d metriği ve her $x, y \in X$ için $0 \leq s < 1$ olmak üzere $d\big(f(x), f(y)\big) \leq sd(x, y)$ şartını sağlayan fonksiyonlardır. İtere fonksiyon sistemleri, iterasyonlar ilerledikçe bir kümeye yakınsar. Başlangıç şartları bu kümeye yakınsadığı için bu nihai küme "çeker" olarak adlandırılır. Ayrıca, elde edilen son yapı fraktal olduğu için boyutu da tam sayı değil kesirli olmaktadır. Teorem 1.1.20 bu bahsettiğimiz özellikleri matematiksel olarak formüle etmektedir.

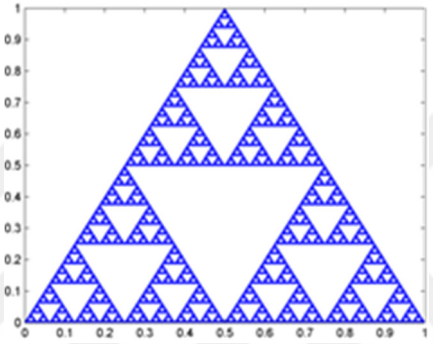
Teorem 1.1.20. $\{X; w_1, \ldots w_N\}$, (X, d) metrik uzayında tanımlı ve büzüşme sabiti s olan itere fonksiyon sistemi olsun. Bu durumda, $H(X)$, X'in boş olmayan tüm kompakt alt kümelerinin uzayı, d_H Hausdorff metriği ve $w_n \in (H(X), d_H)$ olmak üzere

$$W(B) = \bigcup_{n=1}^{N} w_n(B)$$

ile ifade edilen $W: H(X) \to H(X)$ fonksiyonu $(H(X), d_H)$ üzerinde büzüşme sabiti s olan bir büzüşme fonksiyonudur. Dahası, her $B \in H(X)$ için

$$\lim_{n\to\infty} W^n(B) = A$$

şartını sağlayan değişmez ve tek bir A kümesi vardır ve bu küme itere fonksiyon sisteminin çekeri olarak adlandırılır [71].



Şekil 1.9 İtere fonksiyon sisteminin fraktal çekeri

Teorem 1.1.20 fraktal yapıları oluşturmada oldukça önem arz etmektedir. Öncelikle herhangi bir boş olmayan kompakt alt küme için W'nin iterasyonları A'ya yakınsayacaktır. Tek bir noktanın kümesi kompakt olduğu için başlangıçta tek bir nokta seçmemiz mümkündür. İkinci olarak, bu A kümesi değişmez olduğu için $W(A) = A$ şartını sağlayacaktır. Yani iterasyonlar neticesinde daima aynı çekeri elde etmemiz gerekmektedir. Eşitlik (1.4) ile verilen itere fonksiyon sistemlerine Teorem 1.1.20 uygulandığında başlangıç kümesi tek bir nokta alınarak Şekil 1.9'daki fraktal çeker elde edilmiştir.

$$\begin{aligned} w_1(\mathbf{x}) &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ w_2(\mathbf{x}) &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1/2 \\ 0 \end{bmatrix} \\ w_3(\mathbf{x}) &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1/4 \\ 1/2 \end{bmatrix} \end{aligned} \tag{1.4}$$

Şekil 1.9'daki fraktal literatürde Sierpinski fraktali olarak adlandırılmaktadır. İtere fonksiyon sistemi büzüşme fonksiyonları olduğu için iterasyonlar sonsuza giderken başlangıçtaki hacmin git gide küçülerek sıfıra düşmesi beklenir. Bu durum çekerin boş

küme veya tek bir nokta olması gerektiği anlamına gelmemektedir; eğer itere fonksiyon sistemi 2 boyutlu bir uzayda tanımlıysa çeker kümesinin 2-boyutlu hacminin sıfır olması gerektiği anlamına gelmektedir. Fakat, Şekil 1.9'dan görülebileceği üzere elde edilen çekerin boyutu 1'den büyük olmalıdır. Bu ise bize boyutun 1 ile 2 arasında olması gerektiğini göstermektedir ve şeklin boyutu kesirli olmalıdır. Bu tür fraktallerin boyutunu hesaplamak için çeşitli teknikler geliştirilmiştir. Bunlardan biri kutu sayma boyutu olarak adlandırılmaktadır. Kutu sayma boyutunun temeli fraktalleri oluşturan kuralları uygularken ortaya çıkan boş olmayan ve tekparça (connected) kümeleri kenar uzunluğu ε olan $N(\varepsilon)$ tane kutu ile kaplamaya dayanır. Buna göre kutu sayma boyutu D_B aşağıdaki gibi hesaplanır [2].

$$D_B = \lim_{\varepsilon \to \infty} \frac{ln(\,N(\varepsilon))}{ln(\,1/\varepsilon)} \tag{1.5}$$

Örnek olarak Şekil 1.9'daki fraktal için $b_n = 1/2^n$ ve $N(b_n) = 3^n$ olduğundan, Eşitlik (1.5)'e göre fraktalin boyutu $\ln 3 \,/\, \ln 2 = 1.585$ olup yukarıda bahsedildiği gibi 2'den küçük fakat 1'den de büyüktür.

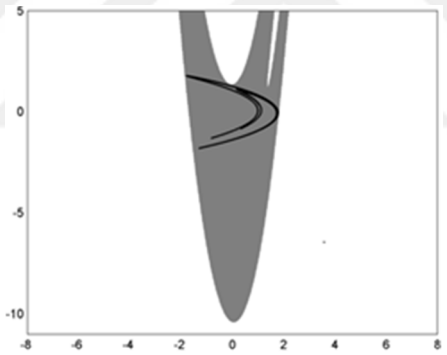
1.1.8. Smale At Nalı Haritası ve Kaotik Çekerler

Smale at nalı haritası 2 boyutlu metrik uzaylarda kaosun elde edilmesi için bir şablon sunmaktadır [2, 61, 69]. Bu şablonda 2 boyutlu bir kare ele alınarak f haritasının bir yönde iterken diğer yönde çektiği varsayılır. Bu nedenle her iterasyonda hedef kümesi uzun bir şerite dönüştürülür ve orijinal karenin üzerine katlanır. Bu işlemde f haritasının terslendirilebilir olduğu varsayılır ve aynı işlem ters fonksiyonla da uygulanır. Ters fonksiyonda kararlı ve karasız yönlerin ters çevrileceğine dikkat etmek gerekir. Bu işlem harita ve tersi için sonsuz kez uygulandığında şeritler bir çizgi halini alır; yatay ve dikey çizgilerin kesişimi bize birtakım noktalar verir. Bu işlem sonucunda elde edilecek noktaların boş bir küme olamayacağı "iç içe geçmiş aralıklar teoremi"nden (nested interval property) ispatlanır [72].

Bölüm 1.1.2'de yapıldığı gibi bu işlem sembolik dinamiklerle ifade edilebilir. Farklı olarak, haritanın tersi alınabiliyor olması dolayısıyla sembollerin negatif iterasyonlar için de $\ldots s_{-k} \ldots s_{-1} \bullet s_0 s_1 \ldots s_k \ldots$ şeklinde ifade edilebiliyor olması söylenebilir. Burada negatif ve pozitif semboller arasındaki nokta şimdiki durumu ifade etmektedir. Ayrıca

kaydırma fonksiyonu σ da terslendirilebilir olacaktır. Geri kalan homeomorfizmin detayları Bölüm 1.1.2'dekine benzer.

At nalı haritası sonucu elde edilen bu kesişim noktalarının en önemli özelliği f ve f^{-1} altında değişmez olan bir küme oluşturmalarıdır. Bu kümenin kaotik olduğu [69]'de gösterilmektedir. Ayrıca aynı kurallarla elde edildiği ve 2 boyutlu boyutu sıfır olduğu için gerçekten de yapısı önceki kısımda ifade edilen fraktal çekerlere benzer. Dolayısıyla bu değişmez kaotik küme kaotik çeker olarak adlandırılır. Fraktal çekerlerden farklı olarak kaotik çekerler tüm başlangıç şartlarını çekmez; diğer lineer olmayan dinamikler gibi bir çekim havzasına sahiptir. Bu nedenle kaotik çekerler bulunurken çekim havzası üzerindeki bir nokta başlangıç koşulu olarak alınır ve yörünge değişmez olana kadar itere edilir (pratikte ise yörüngenin bir bölgede toplanması beklenir). Örnek olarak, Eşitlik (1.3) ile verilen Henon haritasının $a = 1.4$ ve $b = 0.3$ için elde edilmiş kaotik çekeri ve çekerin çekim havzası Şekil 1.10'da verilmektedir.



Şekil 1.10 Henon haritasının kaotik çekerinin çekim havzası

Bir çekim havzasına sahip olması nedeniyle kaotik çeker global olarak kararlıdır. Fakat çeker üzerindeki yörüngeler kararsız olduğu için lokal olarak kararsızdır denilir. Kaotik çeker de tıpkı Şekil 1.9'daki fraktal gibi 2 boyutlu hacmi sıfır olan bir yapıdadır. Dolayısıyla bir fraktal boyuta sahip olması gerekmektedir. Bu boyutu hesaplamak için farklı yöntemler kullanılsa da Lyapunov üstellerini kullanmak en kolay yöntemdir [73]. Bu yöntem Kaplan-Yorke boyutu olarak adlandırılır ve h_i sistemin Lyapunov üstelleri olmak üzere Eşitlik (1.6) kullanılarak hesaplanır [74].

$$D_{KY} = p + \frac{1}{|h_{p+1}|}\sum_{i=1}^{p} h_i \qquad (1.6)$$

Burada p tam sayısı, $\sum_{i=1}^{p} h_i \geq 0$ şartını sağlayan en büyük değerdir. Bu boyut ifadesinin temeli tamamen bir önceki kısımda görmüş olduğumuz kutu sayma boyutuna dayanmaktadır. Henon haritası için Lyapunov üstelleri $h_1 = 0.42$ ve $h_2 = -1.62$ olarak hesaplanmıştır. Dolayısıyla, Henon haritasının fraktal boyutu yaklaşık olarak $D_{KY} = 1.2593$'tür.

1.1.9. Sürekli Zamanlı Kaotik Sistemler

Sürekli zamanlı kaotik sistemler $\mathbf{x} \in \mathbb{R}^n$ olmak üzere $\dot{\mathbf{x}} = f(\mathbf{x})$ şeklinde diferansiyel denklem takımlarıyla ifade edilen sistemlerdir. Bu tür sistemlerin çözümleri yine yörünge veya akış olarak adlandırılır ve $\mathbf{v_0}$ başlangıç şartı için $\psi(t, \mathbf{v_0})$ ile gösterilir. Burada t zaman değerini belirtmektedir. Bu tür sürekli zamanlı yörüngelerin sabit T aralıklarındaki değerleri $\psi(t_0 + nT, \mathbf{v_0})$ şeklinde alınırsa, yörüngeler tıpkı önceki kısımlardaki haritaların yörüngeleri gibi değerlendirilebilir. Dolayısıyla, önceki kısımlardaki sonuç ve tanımlar sürekli zamanlı sistemlere de uygulanabilir. Sürekli zamanlı sistemlerin en büyük farkı kaotik davranışların otonom lineer olmayan diferansiyel denklemlerle en az üç boyutta elde edilebilmesidir. Bu tür denklemler için bir boyutta elde edilebilecek davranışların sınırlı olduğu barizken, Poincare-Bendixson teoremi de iki boyutta yalnızca kararlı duruma yakınsama, periyodiklik veya yarı-periyodiklik davranışlarının elde edilebileceğini öngörmektedir [75]. Bu nedenle, otonom diferansiyel denklem takımlarıyla kaosun görülebilmesi için en az üç boyuta ihtiyaç vardır.

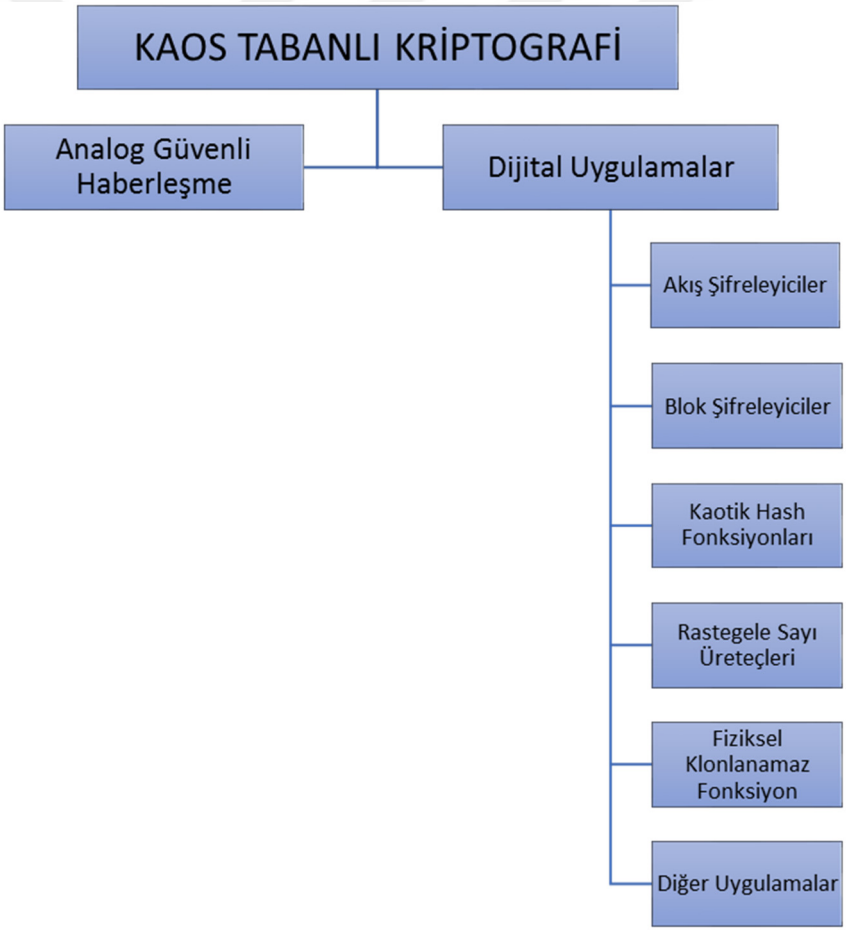
Sürekli zamanlı kaotik sistemlerin özellikleri ve analizi, bu tez çalışması sırasında keşfedilmiş yeni bir adi diferansiyel denklem (ODE) tabanlı kaotik sistem üzerinden kapsamlı bir şekilde Bölüm 1.3'de verilmektedir.

1.2. Kaos Tabanlı Kriptografi

1.2.1. Kaos Tabanlı Kriptografinin Kullanım Alanları

Kaos tabanlı kriptografi kaotik sistemlerin bilgi güvenliğinde kullanıldığı bir çalışma alanıdır. Şekil 1.11'den görülebileceği üzere kaos tabanlı kriptosistemler analog güvenli haberleşme ve dijital uygulamalar adı altında iki genel kategoriye ayrılmaktadır. Giriş bölümünde bahsedildiği üzere, analog güvenli haberleşmede sürekli zamanlı iki kaotik sistem analog elektronik devreler ile gerçekleştirilir. Ardından, devrelerden birinin çıkışı

analog haberleşmede kullanılan analog sinyaller üzerine bindirilerek bu sinyallerinin üçüncü kişiler tarafından ele geçirildiğinde bu kişilerin gürültüden farksız bir sinyalle karşılaşmaları sağlanır. Haberleşme kanalı üzerinden bu birleştirilmiş sinyallerin yanı sıra bir senkronizasyon sinyali de gönderilir ve bu senkronizasyon sinyali alıcıda bulunan ikinci kaotik devreyi senkronize etmek için kullanılır. Alıcı kısmında senkronizasyon işlemi yapıldıktan sonra aynı kaotik işaret üretilir ve şifrelenmiş analog sinyalden bu senkronize olmuş sinyal çıkarıldığında geriye orijinal mesaj kalır ve alıcı mesaja güvenli bir şekilde ulaşmış olur [6]. Bu metoda sonradan farklı değişiklikler önerilse de kaos tabanlı analog güvenli haberleşmenin temeli kısaca bu prensibe dayanmaktadır [20–24]. Bu tez çalışmasında, güvenli analog haberleşme üzerinde durulmayacaktır.



Şekil 1.11 Kaos tabanlı kriptosistemlerin sınıflandırılması

Şekil 1.11'den görülebileceği üzere kaos tabanlı kriptosistemlerin dijital uygulamaları çok farklı amaçlar için kullanılmaktadır [9–16]. Blok şifreleyiciler 64 veya 128 bit gibi bit bloklarını ürettikleri aynı uzunluktaki gizli anahtarlarla şifrelerler. Blok şifreleyicilerin genel işleyiş prensibi aşağıdaki fonksiyon yardımıyla ifade edilir:

$$x_{n+1} = E(x_n, z) \tag{1.7}$$

Eşitlik (1.7)'de x_n şifrelenecek veriyi ve z gizli anahtarı temsil ederken x_0 şifrelenecek veri bloğu anlamına gelmektedir.

Akış şifreleyiciler temel olarak sözde rastgele sayı üreteçleri gibi bir bit dizisi (s) üreten kriptosistemlerdir. Akış (stream) şifreleyicilerin işleyiş prensibi çok daha basit olup Eşitlik (1.8)'deki gibi ifade edilir [76].

$$y_i = e_s(x_i) \equiv x_i + s_i \pmod 2 \tag{1.8}$$

Burada e_s şifreleme fonksiyonunu, x_i şifrelenecek biti, s_i akış bitini temsil etmektedir. Eğer deşifreleme fonksiyonunu d_s ile ifade edersek, deşifreleme için gereken fonksiyonun aslında şifreleme için kullanılan fonksiyonla aynı olduğu görülmektedir:

$$\begin{aligned} d_s(y_i) &\equiv y_i + s_i \pmod 2 \\ &\equiv (x_i + s_i) + s_i \pmod 2 \\ &\equiv x_i + 2s_i \pmod 2 \\ &\equiv x_i \pmod 2 \end{aligned}$$

Bu özellik akış şifreleyicilerin en büyük avantajlarındandır.

Kriptografik hash (özet) fonksiyonlarının genel işleyiş prensibi Eşitlik (1.9)'daki gibidir.

$$h = H(m) \tag{1.9}$$

Kriptografik hash fonksiyonu (H), m ile ifade edilen bir veriyi alarak h ile temsil edilen sabit uzunluklu bir hash dizisi döndürür [77]. Hash fonksiyonları veri güvenliğini arttırmada ve iletilen veri üzerinde müdahale olup olmadığının kontrolünde kullanılır. Genellikle bir veri akış veya blok şifreleyici ile şifrelenmeden önce hash fonksiyonu kullanılarak farklı değerlere dönüştürülür.

Rastgele sayı üreteçleri (RSÜ) ise adı üzerinde rastgele sayılar veya bit dizileri üretmek için kullanılır. Genel olarak, sözde rastgele sayı üreteçleri (SRSÜ) ve gerçek rastgele sayı üreteçleri (GRSÜ) olarak ikiye ayrılırlar. GRSÜ'ler fiziksel bir entropi kaynağını gürültü olarak kullanarak kendini asla tekrar etmeyen rastgele bit dizileri üretirler. SRSÜ'ler ise deterministik yöntemlerle kendini uzun periyotlarla tekrar eden ve aynı anahtarla aynı bit

dizilerini üreten RSÜ'leridir. Her RSÜ kriptografik uygulamalarda kullanılamaz. RSÜ'lerin kriptografik uygulamalarda kullanılabilmesi için NIST SP 800-22, Diehard, TestU01 gibi bazı istatistiksel testleri geçmesi gerekmektedir [78–80]. Bunlar arasından literatürde en geçerli olan test NIST SP 800-22 testleri olup bu testleri geçen RSÜ'ler kripografik olarak güvenli RSÜ'ler olarak adlandırılır. Fakat, bu testlerin sonuçları nihai değildir ve bu testleri geçen RSÜ'lerin de güvenli olamayabileceği göz önünde bulundurulmalıdır.

Fiziksel klonlanamaz fonksiyonlar ise son dönemde ortaya çıkmış kriptografik uygulamalardan biridir. Gömülü sistemlerde söz konusu elektronik platformun dijital olarak kimliğinin belirlenmesi ve kopyalanmasının önüne geçilmesi amacıyla kullanılmaktadır [81].

Bu bahsedilen uygulamalar dışında kaos tabanlı kriptografinin anahtar eşleştirme protokolü gibi çok yaygın olmayan diğer uygulamaları da mevcuttur [14].

1.2.2. Mevcut Kaos Tabanlı Kriptosistemler

Literatürde pek çok kaos tabanlı akış şifreleyici bulunmaktadır [5, 11, 82–86]. Bunların çoğunluğu e_s fonksiyonu olarak kaotik haritaları temel almaktadır. Rastgele bitleri ise kaotik haritaların yörüngelerinden doğrudan alarak veya çeşitli eşik değerlerini kullanarak üretmektedirler [87]. Bunun yanı sıra gizli anahtar olarak kaotik haritanın başlangıç şartlarını ve/veya kontrol parametrelerini kullanmaktadırlar. Her ne kadar her çalışma oldukça güvenli olarak sunulsa da literatürdeki pek çok çalışma da bu tür kriptosistemlerin aslında güvenli olmadığını ortaya koymuştur [38, 88, 89]. Belirlenen güvelik problemlerini aşmak için kullanılan kaotik haritayı ikiye çıkarma [90, 91], veya gizli anahtar olarak parametre kullanımının açıklarını kapatmak için dayanıklı (robust) kaotik haritalar kullanma [11] gibi metotlar önerilse de asıl sorun sürekli uzayda tanımlı kaotik sistemlerin ayrık değerli uzayda gerçekleştiriminden kaynaklanmaktadır [32, 38]. Bunun yanı sıra akış şifreleyicilerin çoğunluğu kriptografik uygulamalarda olması gereken temel gereksinimleri karşılayamamaktadır [4]. Kaotik sistem tabanlı kriptografi içerisinde diğer kriptografik algoritmalarda standartlaşmış güvenlik ve performans inceleme kriterleri bulunmadığı için bu sistemlerin hiçbiri tam olarak güvenli kabul edilememektedir [8]. Bu tür şifreleyiciler arasından [11] ve [37] gibi pek çok sorunu göz önünde bulundurmuş olan çalışmalar olsa da aynı nedenden dolayı henüz bunların

gerçekte ne kadar güvenilir olduğu ya da diğer kriptografik eşdeğerlerine oranla performanslarının nasıl olduğu henüz belirlenememiştir.

Bahsedilen bu sorunlar aynen geçerli olmakla birlikte literatüre sunulmuş kaos tabanlı blok şifreleyicilerin sayısı da bir hayli fazladır [92–96]. Yapılan bu çalışmalara rağmen teorik olarak kabul gören veya ticari uygulamalarda kullanılan blok şifreleyicilerin hiçbiri kaos veya dinamik sistemleri kullanmamaktadır [97]. Bunun en önemli nedeni sürekli değerli kaotik sistemlerin ayrıklaştırılmasında yaşanan sorunlardan ziyade blok şifreleyicilerin hızlı çalışması gerekliliğinden kaynaklanmaktadır. Geleneksel blok şifreleyiciler ikili sistemde veya sınırlı bitlerle ifade edilen uzayda tanımlı fonksiyonlardır ve bu nedenle donanım gerçekleştirimlerine oldukça uygundurlar. Buna rağmen günümüzün artan iletişim gereksinimlerine göre bu tür blok şifreleyiciler bile pek çok adımdan oluştukları için yavaş kalabilmektedir. Kaos tabanlı blok şifreleyicilerde ise donanım gerçekleştirimi hemen hemen hiç göz önünde bulundurulmamaktadır. Oysaki lineer olmayan çarpımlar, sözde kaotik hesaplamaların epey bir bit sayısı gereksinimi olması gibi unsurlar geleneksel kriptosistemlere göre kaos veya dinamik sistem tabanlı kriptosistemleri hiç de cazip kılmamaktadır [97]. Yapılan çalışmaların çoğunluğunun bu durumu göz ardı ederek oldukça donanım gerektiren, yavaş veyahut nasıl gerçekleştirileceği net ifade edilmemiş yeni tasarımlarla ortaya çıkmış olması bu durumun düzelmesine yardımcı olmamaktadır.

Donanım gereksinimi ve hızdan muzdarip olma durumu kaos tabanlı hash fonksiyonlarında geçerlidir. Hash fonksiyonu olarak kaotik sistem kullanan literatürdeki ilk yayında lojistik harita kullanılmıştır [98]. Sonraki çalışmalarda ise farklı haritalar kullanılarak veya farklı tasarımlar önerilerek daha yüksek performanslı yapılar önerilmiştir [13, 50, 99, 100]. İncelenen bu çalışmalarda kayan noktalı aritmetik veya çok sayıda kaotik haritanın donanım gereksinimi veya hız gözetilmeksizin kullanıldığı görülmüştür. Oysaki hash fonksiyonları sıklıkla diğer şifreleyicilerle beraber kullanıldığı ve internet iletişiminde yaygın olduğu için oldukça hızlı olmak zorundadır. MD5 ve SHA-1 gibi yaygın algoritmalarda potansiyel güvenlik açıkları görülmesine rağmen istenen hız için bu algoritmaların halen kullanıldığı göz önünde bulundurulduğunda [76], incelenen kaos tabanlı hash fonksiyonlarının da ticari uygulamalarda tercih edileceğini söylemek güçtür.

Kaos tabanlı SRSÜ'ler ile akış şifreleyiciler arasında yakın bir ilişki vardır. Dolayısıyla, akış şifreleyiciler için bahsedilenler kaos tabanlı SRSÜ'ler için de geçerlidir. Kaos tabanlı kriptografide belki de en az sorunlu olan uygulamalar ise kaos tabanlı GRSÜ tasarımlarına aittir. Bunlar, temel olarak analog devre olarak üretilen kaotik osilatörlerden çeşitli metotlarla bit üretilmesi metoduna dayanmaktadırlar. Üretilen bitler ise çoğunlukla dijital platformlarda kullanılacak olan anahtarların üretiminde kullanılmaktadır. Örnek olarak, [10]'da bir kaotik osilatörün çıkışlarının bir eşik detektörü ile kıyaslanması sonucu bit dizisi üretilmektedir. Bu tür uygulamalarda bit üretim hızı nispeten düşük olsa da entegre tabanlı gerçekleştirimlerde oldukça yüksek hızlara ulaşmak mümkün olmaktadır [101, 102].

Burada bahsedilen yapıların yanı sıra, özellikle son dönemlerde DNA şifrelemeyle kaos tabanlı kriptosistemleri birleştiren yayınlar ortaya konulmuştur [103]. Bu tür yayınlar kaos tabanlı blok şifreleyicilerdeki güvenlik zafiyetlerini DNA şifrelemeyi de işin içine katarak aşmayı amaçlamaktadırlar. DNA şifrelemede 00, 11 gibi komplement bitleri Adenin-Timin gibi komplement olan DNA bileşenleri ile ifade etmeye dayanır. Kaos tabanlı şifreleyicilerle bunu birleştirmenin güvenliği ne kadar arttırdığı net olarak ortaya konamasa da bu işlemlerin gerçekleştirimi zorlaştırdığı bu yayınlarda da belirtilmektedir [103].

1.2.3. Kaotik Sistemlerin Dijital Gerçekleştirimleri

Kaotik sistemin analog devre üzerinde gerçekleştirildiği GRSÜ tasarımları hariç diğer tüm tasarımlarda, kaos tabanlı kriptosistemlerin dijital uygulamaları, kullanılan kaotik sistemin dijital platform üzerinde gerçekleştirilmesi prensibine dayanmaktadır. Fakat, kaotik sistemlerin konvansiyonel sabit veya kayan noktalı dijital gerçekleştirimlerinin matematiksel olarak kaotik olmasının imkânı yoktur. Çünkü, Bölüm 1.1'de gösterilmekte olduğu üzere kaotik bir sistem sayılabilir sonsuzlukta kararsız periyodik yörüngeye ve sayılamaz sonsuzlukta kaotik yörüngeye sahip olmalıdır. Oysaki dijital sistemlerde sayılar sadece sınırlı sayıda rasyonel sayı ile ifade edilebilmektedir. Bu rasyonel sayılar ise ikili tabanda küsuratı periyodik olmayan rasyonel sayılardır. Bu nedenle matematiksel olarak deterministik bir şekilde tanımlanan kaotik haritaların dijital platformlarda gerçekleştirilmesi için yeterli çoklukta sayı bulunmamaktadır. Eğer irrasyonel sayıları bir

şekilde dijital olarak ifade edebilsek bile bunlar hafıza üzerinde sınırlı sayıda olacakları için periyodik olmayan kaotik bir yörüngeyi ifade etmekte yetersiz kalacaklardır.

Bu durumun en güzel örneği Eşitlik (1.2) ile verilen ve kaotik olduğu literatürde iyi bilinen çadır haritası T'dir. İkili tabanda sonlu sayıda bit ile ifade edilen başlangıç şartları (rasyonel sayılar) için T altındaki yörüngeler homeomorfizm altında kaydırma haritasının gerçekleştirdiği kaydırma işlemleri yüzünden sıfıra yakınsamak zorundadır. Fakat, dijital bir platformda elimizdeki tüm sayılar bu tür rasyonel sayılardır. Dolayısıyla, T'nin tüm dijital gerçekleştirimleri için yörüngelerin tümü sıfıra yakınsamak zorundadır. Sonuç olarak bu tür gerçekleştirimlerde elimizde hiçbir kaotik davranış yoktur.

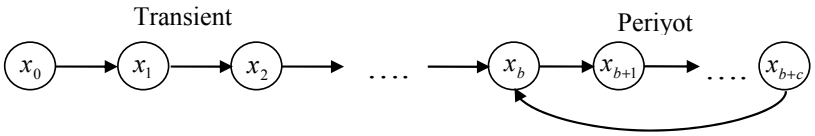
Başka haritaların gerçekleştirimi T gibi tamamen bir noktaya yakınsamasa bile orijinal tasarlanan haritayla alakasız olacaktır. Çünkü, bu tür haritalar sonsuz bir metrik uzayda tanımlı olmasına rağmen dijital platformlar sonlu bir uzay verir ve kaos burada tanımsızdır.

Buna rağmen, pek çok kaotik sistemin simülasyonları dijital platformlarda yapılmaktadır. Ancak, burada yapılan kaotik sistemleri dijital olarak gerçekleştirmek ya da kaosun varlığını ispatlamak değil, kaotik davranışların bir tür gösterimini yapmaktır. Aksi düşünülerek tasarlanan kaos tabanlı kriptosistemlerin matematiksel olarak kaos tabanlı olduklarını söylemek imkansızdır.

1.2.4. Kaos Tabanlı Kriptosistemlerdeki Problemler

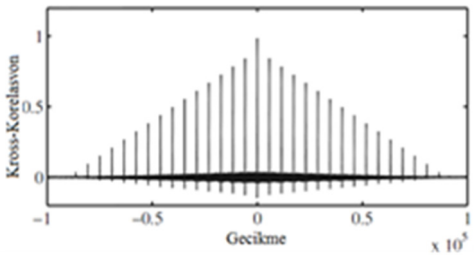
Bir önceki kısımda gösterildiği üzere, kaotik sistemlerin sahip olduğu başlangıç şartlarına hassas bağlılık, rastgele benzeri çıkış üretme, ergodiklik ve topolojik karışma gibi temel özellikler sadece sürekli değerli uzayda tanımlanmıştır. Kaos tabanlı kriptosistemler ise dijital ortamlarda gerçekleştirildikleri için bu özelliklerin hangilerinin korunup korunmadığı ciddi bir soru işaretidir. Nitekim kaotik sistemlerin kriptografik uygulamalarda kullanılmasını öneren ilk çalışmada bu hususlar dikkate alınmamıştır [5]. Sonraki bir çalışmada ise [5]'de önerilen yapının bilgisayar ortamındaki sonlu çözünürlükte gerçekleştirildiğinde rastgele benzeri bir çıkış üretmek yerine, üretilen yörüngelerin yuvarlama hataları nedeniyle periyodik çıkışlar ürettiği ve bu periyotların da modern kriptografik uygulamalar için çok kısa olduğu gösterilmiştir [31]. Bu şekilde

üretilen periyodik yörüngeler Şekil 1.12'deki gibidir. Buna göre, belli bir transient bölgesinden geçildikten sonra yörüngeler periyodik hale gelmektedir.



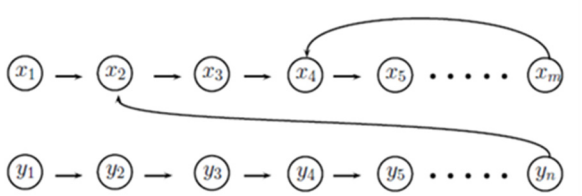
Şekil 1.12 Kaotik sistemlerin sayısal ortamlarda gerçekleştirilmesi sonucu oluşan periyodik yörüngeler

Dahası, [32]'de kısa periyotlu yörüngeler üretilme sorununun yanı sıra sonlu çözünürlük gerçekleştirimlerinin dinamik degradasyona neden olarak ergodiklik ve topolojik karışma gibi özellikleri de bozdukları gösterilmiştir. Yine sonlu çözünürlükteki gerçekleştirimlerde, başlangıç şartlarına hassas bağlılığın üretilen çıkışların sadece transient bölgesinde etkin olduğu ve çıkışın periyodik kısımlarının farklı başlangıç şartları için büyük oranda aynı olduğu [38]'de gösterilmektedir. Örnek olarak, Eşitlik (1.1)'de verilen lojistik harita, kayan noktalı aritmetik ile sonlu çözünürlük altında gerçekleştirildiğinde iki farklı yörüngenin korelasyon özellikleri Şekil 1.13'deki gibi olmaktadır.



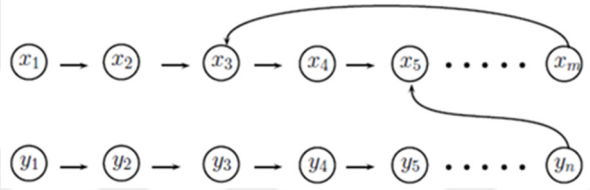
Şekil 1.13 Sonlu çözünürlükte lojistik harita tarafından üretilen yörüngelerin kros-korelasyonu [104]

Bu şekilden görüldüğü üzere, farklı yörüngeler arasında güçlü birer korelasyon ilişkisi bulunmaktadır. Bunun nedeni ise üretilen farklı yörüngelerin periyodik kısımlarının benzer olmasıdır. Bu durum, iki yörünge özelinde düşünülecek olursa; ikinci yörünge yuvarlama hataları nedeniyle birincinin transient bölgesiyle kesişmekte ve bu noktadan itibaren birinci yörüngeyle aynı değerleri takip edip aynı periyot uzunluğunu paylaşmaktadır [105]. Şekil 1.14 bu durumu tasvir etmektedir.



Şekil 1.14 İki yörüngenin sonlu çözünürlükte transient bölgesinde kesişmesi [105]

Bu transient kesişimlerine ilave olarak yine [38]'de yapılan çalışmada iki yörüngenin periyot bölgesi kesişimlerinin de söz konusu olduğu tespit edilmiştir. Periyot kesişimi gösterimi Şekil 1 .15'te verilmektedir. Başlangıç şartlarına hassas bağımlılık bu kesişim noktalarından itibaren geçersiz kalmaktadır. Bu durumu göz ardı eden kaos tabanlı kriptosistemlerde güvenlik zaaflarının olduğu ise [38]'deki çalışmanın devamında gösterilmektedir.



Şekil 1.15 İki yörüngenin sonlu çözünürlükte periyot bölgesinde kesişmesi [38]

Bu nedenle, sürekli değerli uzayda tanımlı kaosun sahip olduğu özelliklerin sayısal gerçekleştirimlerde de korunacağının varsayımı ile kaos tabanlı kriptosistem tasarımlarının önerilmesi hatalı bir yaklaşım olmaktadır. Buna rağmen, literatürdeki pek çok çalışma bu tür sorunları göz ardı etmekte ve gerçekleştirim detaylarını oldukça yüzeysel bir biçimde sunmaktadır [4]. Dolayısıyla, önerilen kaos tabanlı kriptosistemlerin birçoğu altyapıları itibarı ile güvensiz olmaktadır. Örneğin, başlangıç şartlarına hassas bağlılık özelliğinin dijital gerçekleştirimlerde de korunacağını kabul eden ve kaotik sistemin başlangıç şartını gizli anahtar olarak kullanan [91]'deki kriptosistemin gizli anahtar bilinmeksizin kırılabileceği [38]'de gösterilmektedir.

Önerilen kaos tabanlı kriptosistemlerde bulunan diğer bir eksiklik ise önerilen yapıların kriptanalizlerinin yapılmamasıdır. Çoğu kaos tabanlı kriptosistem literatüre sunulurken, temel kriptanaliz teknikleri ile güvenliği kontrol edilmediği için yukarıda bahsedilen sorunların neden olduğu açıklar ilk başta belli olmamaktadır. Fakat bu sistemlere temel kriptanaliz teknikleri uygulandığında tasarımdan kaynaklanan güvensizliklere sahip oldukları ortaya çıkmaktadır [88, 89, 106].

Kaos tabanlı kriptosistemlerde sonlu çözünürlüğün neden olduğu sorunlar dışında, kriptosistemin güvenliğini zedeleyebilecek bir takım tasarımdan kaynaklı eksiklikler de bulunmaktadır. Bunlar sadece tek bir kaotik sistemin kullanılması, yavaş şifreleme hızı, kompleks bir gerçekleştirime sahip olma ve gerçekleştirim için çok fazla donanım kaynağı kullanma şeklinde sıralanabilir [87].

Ayrıca, üretilen yörüngelerin periyodik sözde yörüngeler olduğu bilinse ve bundan kaynaklı birtakım dezavantajlar giderilse bile, kaos tabanlı kriptosistemler yine de klasik kriptografide kullanılmak için cazip yaklaşımlar olmamaktadırlar. Bunun en temel nedeni, klasik kriptografinin tam sayılara ve basit bitsel işlemlere dayanmasıdır. Bu bakımdan, reel sayılar kümesinde tanımlı kaotik sistemlerin sabit veya kayan noktalı aritmetik ile gerçekleştirimlerinin klasik kriptografide karşılığı bulunmamaktadır. Bunun yanı sıra klasik kriptografi açısından istenmeyecek bazı dezavantajlar daha vardır. Bunlar şu şekilde sıralanabilir: (i) Üretilen sözde yörüngelerin periyot uzunluğunun ne olduğu belli değildir; (ii) Orijinal matematiksel model ile sonlu çözünürlük ve yuvarlama hataları altında yapılan gerçekleştirim arasındaki ilişki net bir şekilde belli değildir; (iii) Aynı gerçekleştirim platforma ve yuvarlama türüne göre değişiklik gösterebilir ve bu ise şifreleme / deşifreleme işleminin farklı platformlarda farklı sonuçlar vermesine neden olabilir; (iv) Sabit ve kayan noktalı aritmetik işlemlerin donanım ile gerçekleştirimi zordur. Bu bahsedilen nedenlerden ötürü kaos tabanlı kriptosistemlerin gerçekleştirimi için yeni tasarım yaklaşımlarına ihtiyaç bulunmaktadır.

1.3. Yeni Bir Sürekli Zamanlı Kaotik Sistemin Analizi ve Devre Gerçekleştirimi

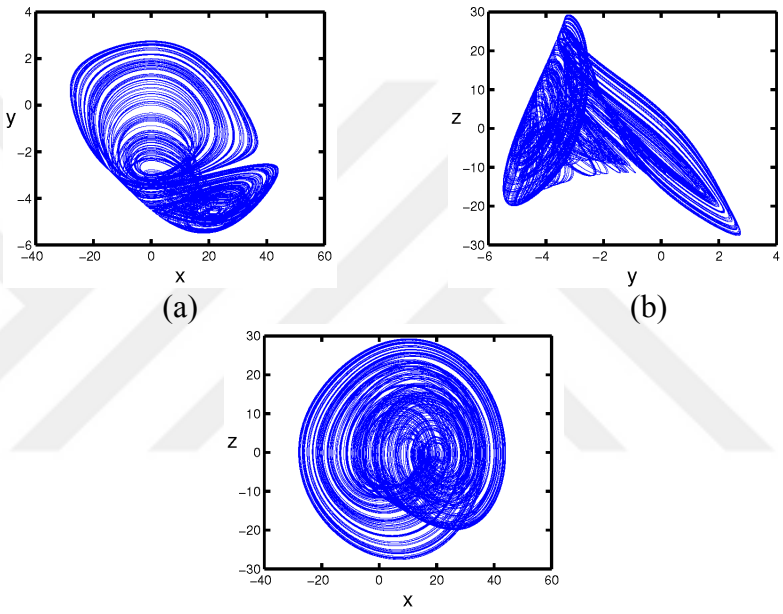
Bu bölümde, bu tez çalışması sırasında keşfedilmiş olan ODE tabanlı yeni bir kaotik sistem üzerinden bu tür sistemlerin analizlerinin ve devre gerçekleştiriminin nasıl yapılacağı gösterilmektedir. Bölüm 1.1'de verilen kavram ve tanımların nasıl kullanıldığı da uygulamalı olarak bu bölümde gösterilmektedir.

1.3.1. Önerilen Kaotik Sistem

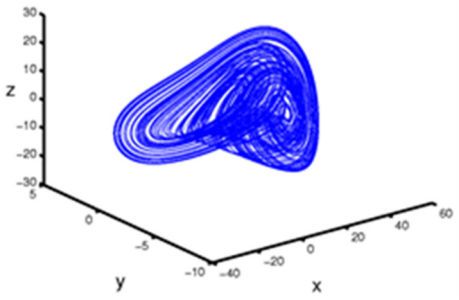
Önerilen kaotik sistem üç boyutlu adi diferansiyel denklem (ODE) takımı ile Eşitlik (1.10)'daki gibi ifade edilmektedir [107].

$$\begin{aligned} \dot{x} &= y - az \\ \dot{y} &= -x - bx^2 y \\ \dot{z} &= c(x - y^2) \end{aligned} \tag{1.10}$$

Sistemin elektronik devre gerçekleştiriminde standart eleman değerlerinin kullanılmasının mümkün olması için parametre değerleri $a = 10$, $b = 0.01$ ve $c = 20/3$ şeklinde seçilmiştir. Seçilen bu değerler için elde edilen çeker yapıları Şekil 1.16 ve Şekil 1.17'de görüldüğü gibidir.



Şekil 1.16 Kaotik çekerin faz görüntüsü: (a) x-y düzlemi; (b) y-z düzlemi; (c) x-z düzlemi



Şekil 1.17 Kaotik çekerin üç boyutlu görüntüsü

1.3.2. Denge Noktalarının Analizi

Eşitlik (1.10)'un iki tane denge noktasının olduğunun bulunması oldukça kolaydır. Bu denge noktalarından biri orijin $\vec{O}$ olup diğer denge noktası Eşitlik (1.11)'deki gibidir.

$$\vec{E} = \left(\frac{1}{b^{2/3}}, \frac{-1}{b^{1/3}}, \frac{-1}{ab^{1/3}}\right) \tag{1.11}$$

Bu denge noktalarının kararlılık analizini yapabilmek için Eşitlik (1.10)'a ait Jacobian matrisinin incelenmesi gerekmektedir. Jacobian matrisi Eşitlik (1.12)'deki gibi bulunmaktadır.

$$J = \begin{bmatrix} 0 & 1 & -a \\ -1 - 2bxy & -bx^2 & 0 \\ c & -2cy & 0 \end{bmatrix} \tag{1.12}$$

Jacobian matrisi kullanarak denge noktaları $\vec{O}$ ve $\vec{E}$ etrafında yapılacak lokal lineerleştirme neticesinde $\vec{\lambda}_O$ ve $\vec{\lambda}_E$ özdeğerleri elde edilmektedir. Yukarıda verilen standart parametre değerleri için bu özdeğerler sırasıyla $\vec{\lambda}_O = (0, \quad 8.226i, \quad -8.226i)$ ve $\vec{\lambda}_E = (2.1441 + 9.968i, \quad 2.1441 - 9.968i, \quad -8.9297)$ olarak bulunmaktadır. Buna göre orijin hiperbolik olmayan bir denge noktası iken $\vec{E}$ hiperbolik bir eyer-fokus noktasıdır. Eyer-focus noktasında lokal olarak sistem kompleks düzlemde spiral bir manifolda; tek boyutlu bir yol boyunca da bir diğer manifolda sahiptir [108]. Orijin hiperbolik olmadığı için bu noktaya Lyapunov kararlılık teoremi uygulanamaz [2]. Dolayısıyla bu nokta etrafında sistemin nasıl davranacağı klasik analiz metotlarıyla belirlenemez. Fakat $\vec{E}$ noktasının kararlılığı Lyapunov kararlılık teoremine göre belirlenebildiği için bu nokta eyer noktası olarak adlandırılmaktadır. $\vec{\lambda}_E$ incelendiğinde $\vec{E}$'nin kompleks düzlem üzerindeki manifoldunun kararsız olduğu; buna karşın diğer tek boyutlu manifoldunun kararlı olduğu görülmektedir. Buna göre, $\vec{E}$ çevresindeki çözümler bir boyutlu manifold boyunca $\vec{E}$'ye doğru çekilirken kompleks düzlem boyunca spiral yaparak $\vec{E}$ dışına itilirler.

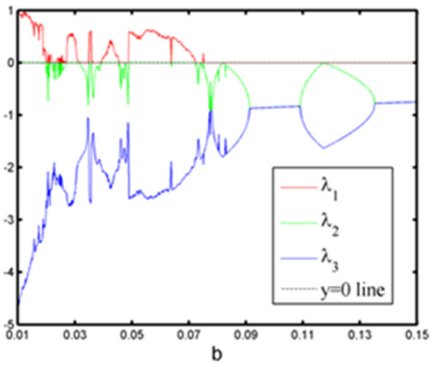
$\vec{\lambda}_O$ için genel bir formülün bulunması oldukça kolay olup Eşitlik (1.13)'deki gibi bulunmaktadır. Görüldüğü üzere tüm parametre değerleri için özdeğerin bir bileşeni sürekli sıfır olmaktadır. Bu ise tüm parametre değerleri için orijinin hiperbolik olmayan

bir denge noktası olduğu anlamına gelmektedir. Yani sistem tüm parametre değerleri için kararlılığı Lyapunov kararlılık teoremi ile belirlenemeyen hiperbolik olmayan bir denge noktasına sahip olacaktır.

$$\vec{\lambda}_o = (0, \quad i\sqrt{1+ac} \quad -i\sqrt{1+ac}) \tag{1.13}$$

1.3.3. Lyapunov Üstellerinin Analizi

Kaosun en bilinen göstergelerinden biri pozitif Lyapunov üstellerinin varlığıdır. Lyapunov üstelleri birbirine yakın yörüngelerin bir doğrultu boyunca birbirlerinden uzaklaşmalarının eksponansiyel bir ölçüdür. Bölüm 1.1'den hatırlanacağı üzere tüm kaotik sistemler en az bir pozitif Lyapunov üsteline sahip olmalıdır. Lyapunov üstellerinin analitik olarak bulunması çok basit sistemler dışında mümkün değildir. Bu nedenle Lyapunov üstellerinin hesaplanmasında birtakım algoritmalardan faydalanılır. Bu çalışmada, Lyapunov üstellerini hesaplamak için değişken eşitlikler (variational equations) metodu kullanılmaktadır [109]. Buna göre, standart parametre değerleri için Lyapunov üstelleri $\{0.9645, 0, -4.6991\}$ olarak bulunmuştur. Görüldüğü üzere bu parametre değerleri için sistem pozitif bir Lyapunov üsteline sahip olduğu için kaotiktir. Bu çalışma boyunca a ve c paramtereleri sabit tutulurken b parametresi değiştirilecektir. Buna göre, değişen b parametresi için hesaplanan Lyapunov üstelleri Şekil 1.18'de görülmektedir. Sistem pozitif Lyapunov üstelinin olduğu parametre değerleri için kaotiktir.

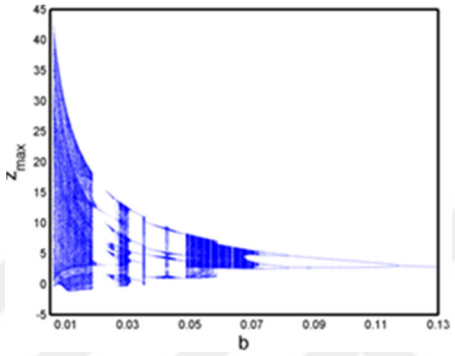


Şekil 1.18 Değişen b parametresine karşılık Lyapunov üstellerinin değişimi

1.3.4. Dallanma Analizi

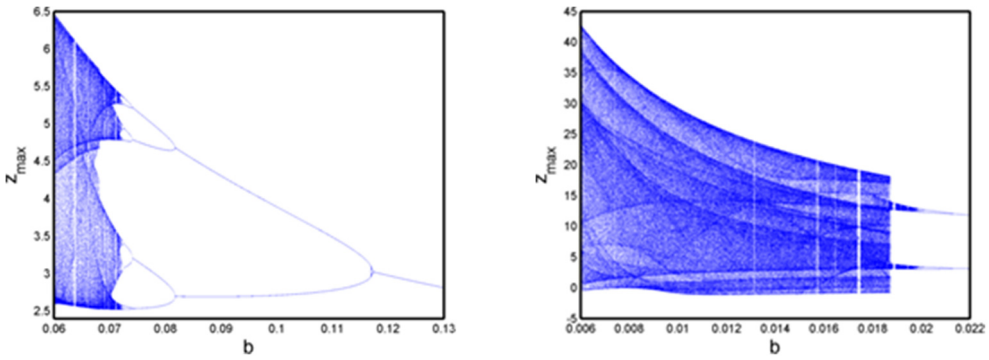
Sürekli zamanlı sistemlerin dallanmalarını incelemenin bir yolu ayrıklaştırma yaparak ayrık zamanlı sistemin simülasyonlarının yapılmasıdır. Sürekli zamanlı sistemleri

ayrıklaştırmak için kullanılan metotlardan biri ise lokal ekstremum noktalarının kaydedilerek bunların ayrık zamanlı bir sistemin yörüngesi olarak kullanılmasıdır. Bu amaç doğrultusunda [1]'deki çalışmaya benzer olarak z ekseni boyunca lokal maksimumların kümesi olan z_{max} değerleri incelenmiştir. z_{max} değerleri için elde edilen dallanma diyagramı Şekil 1.19'da görüldüğü gibidir. Sistemin kaotik olduğu bölgeler (standart değer olan $b = 0.01$ dahil omak üzere) şekilden açıkça görülmektedir.



Şekil 1.19 Değişen b parametresi için elde edilen dallanma diyagramı

Şekil 1.19'un $b \in [0.06,0.13]$ aralığındaki daha yakın bir görüntüsü ise Şekil 1.20 (a)'da daha net görülmektedir. Bu şekilden görülebileceği üzere önerilen sistem klasik periyot katlama senaryosu ile kaosa gitmektedir. Standart parametre değeri olan $b = 0.01$' den hemen önceki periyot katlama dallanmaları ise Şekil 1.20 (b)'de verilmektedir.

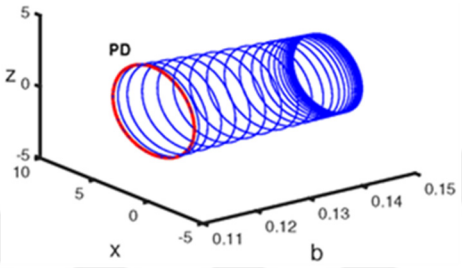


(a) (b)

Şekil 1.20 Farklı değerler için dallanma diyagramı: (a) $b \in [0.06,0.13]$ için; (b) $b \in [0.006,0.022]$ için

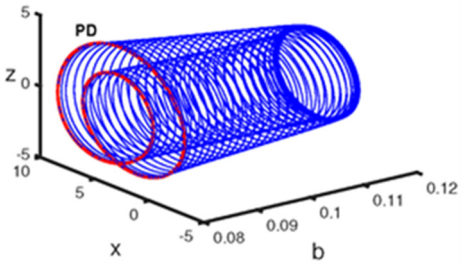
Yukarıdaki dallanma diyagramlarının doğruluğunu teyit etmek için Matcont süreklilik analizi programından faydalanılmıştır [110]. Bu programı kullanarak kararlı bir limit çevrim seçildikten sonra değişen parametre değerleri için söz konusu limit çevrimin

sürekliliği takip edilebilir ve Floquet çarpanları hesaplanarak dallanmaların gerçekleştiği parametre değerleri bulunabilir. Floquet çarpanları limit çevrimlerin kararlılığının bir ölçüsü olup, sürekli zamanlı sistemler için bir Floquent çarpanının Öklidyen normu daima bire eşit olup diğer Floquet çarpanlarının Öklidyen normları birden küçük ise limit çevrim kararlıdır [2]. Periyot katlama noktalarında ise bir diğer Floquet çarpanının Öklidyen normu da bire eşit hale gelmektedir. Matcont yazılımı ile $b = 0.15$ noktasında seçilmiş olan kararlı limit çevrimin süreklilik analizi Şekil 1.21'de gösterildiği gibi elde edilmiştir.



Şekil 1.21 $b = 0.15$' teki kararlı limit çevrimin Matcont ile süreklilik analizi

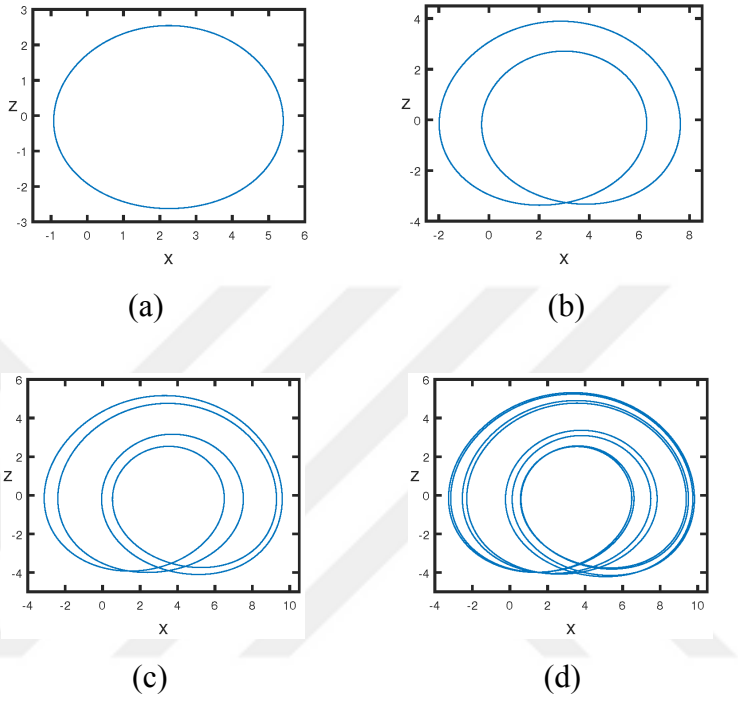
PD ile etiketlenen $b = 0.117238$ değerinde iki tane Floquet çarpanının Öklidyen normu bire eşit hale gelmekte ve Matcont periyot katlama dallanması tespit etmektedir. Bu parametre değerinden sonra söz konusu limit çevrim kararsız olmakta ve periyodu öncekinin iki katı olan yeni bir kararlı limit çevrim oluşmaktadır. Bu yeni oluşan periyot-2 limit çevrimin süreklilik analizi ise Şekil 1.22'de gösterildiği gibi elde edilmiştir.



Şekil 1.22 $b = 0.117238$ noktasında oluşan periyot-2 limit çevrimin süreklilik analizi

Bu süreklilik analizi sonrasında $b = 0.0817725$ değerinde bir diğer periyot katlama dallanması yakalanmaktadır. Yine aynı şekilde bu parametre değerinden sonra periyot-2 limit çevrim kararsız hale gelmekte ve yeni bir periyot-4 limit çevrim elde edilmektedir. Benzer analizler sonucunda periyot katlama dallanmalarının olduğu yerler periyot-8 için $b = 0.07412$, periyot-16 için $b = 0.07255$ olarak bulunmuştur. Görüldüğü üzere periyot katlamalar git gide sıklaşmakta ve periyot katlamalar arasındaki farklar bir Feigenbaum

sabitine yakınsamaktadır [111]. Bu şekilde sonsuz tane kararsız periyodik yörünge meydana gelip neticede sistem kaosa gitmektedir. Bu klasik periyot katlama senaryosudur. Bu değerler arasındaki farklı periyotlara sahip periyodik yörüngeler Şekil 1.23'de görüldüğü gibidir.



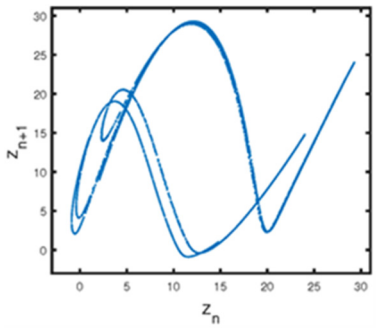
Şekil 1.23 Farklı b değerleri için elde edilmiş periyodik yörüngeler: (a) $b = 0.15$ için periyot-1 yörünge; (b) $b = 0.1$ için periyot-2 yörünge; (c) $b = 0.075$ için periyot-4 yörünge; (d) $b = 0.073$ için periyot-8 yörünge

Yine benzer biçimde Şekil 1.20 (b)'deki periyot katlama dallanmalarının yerleri sırasıyla $b = 0.02132$ ve $b = 0.02013$ olarak bulunmuştur. Farklı olarak başlangıç değeri olan $b = 0.022$' de kararlı limit çevrim periyot-2 yörünge olup yukarıdaki noktalarda sırasıyla periyot-4 ve periyot-8 yörüngeler oluşmaktadır. Yine aynı şekilde periyot katlama senaryosu ile sistem kaosa gitmektedir. Şekil 1.20 (b)'de yine bu senaryoda görülebilen periyodik pencerelerin varlığı da görülmektedir.

1.3.5. Poincare Kesiti ve z_{max} Değerlerinin Ruelle Grafiği

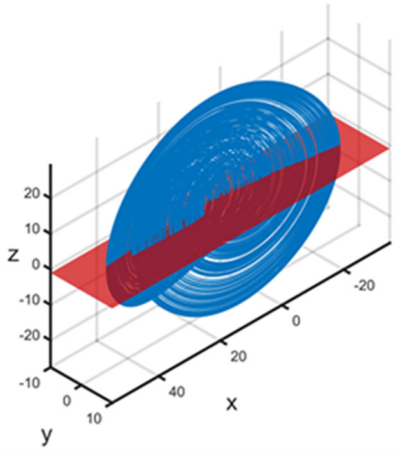
Dallanma analizlerinde incelemiş olduğumuz z_{max} değerlerinin periyodik olup olmadığının incelenmesi için z_{max} değerlerinin Ruelle grafiği incelenebilir. İlgili grafik Şekil 1.24'te gösterilmektedir. Buna göre z_{max} değerleri tıpkı kaotik bir harita gibi bir çeker oluşturmaktadır. Oluşan çeker oldukça kompleks bir yapıda için bu kaotik harita

için kapalı formda bir ifade elde etmek mümkün görünmemektedir. Yine de bu grafik sistemin kaotik davranışının ayrık zamandaki bir göstergesidir.

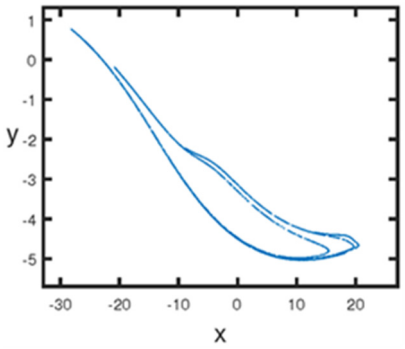


Şekil 1.24 z_{max} değerlerinin Ruelle grafiği

Poincare kesiti ise n boyutlu uzayda tanımlı sürekli zamanlı yörüngelerin üzerinden transversal olarak geçtiği $n-1$ boyutlu bir düzlemdir [2]. Bu kesit ile sürekli zamanlar tıpkı z_{max} incelemesindeki gibi ayrıklaştırılır. Çoğu zaman bu ayrıklaştırılmış dinamikleri analiz etmek sürekli zamanlı sistemin kendisini analiz etmekten daha kolaydır. Önerilen sistem için Poincare kesiti olarak $z=-1$ düzlemi seçilmiştir. Şekil 1.25'den görülebileceği üzere bu düzlem yörüngelerin akışına transversaldir. Yüzey üzerinden geçen yörüngeler için z' nin azalan değerleri kaydedildiğinde ise Şekil 1.26'daki gibi bir Poincare kesiti elde edilmektedir.

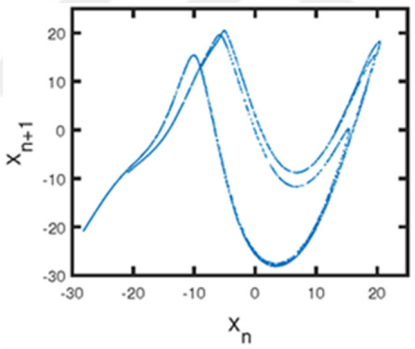


Şekil 1.25 Kaotik çeker ve seçilen düzlem



Şekil 1.26 Poincare kesiti

Tek başına Poincare kesitinin incelenmesi yeterli değildir. Genellikle Poincare kesitinin Ruelle grafiği çıkarılarak bu grafikten Poincare haritasının nasıl bir davranışa sahip olduğu incelenir. Şekil 1.26'daki Poincare kesitinin x eksenindeki Ruelle grafiği Şekil 1.27'de görüldüğü gibidir. Bu şekle göre Poincare haritası kesinlikle kompleks bir davranış göstermekte olup elde edilen grafik kaotik bir çekerdir. Dolayısıyla Poincare analizi de sistemin kaotik olduğunu desteklemektedir.



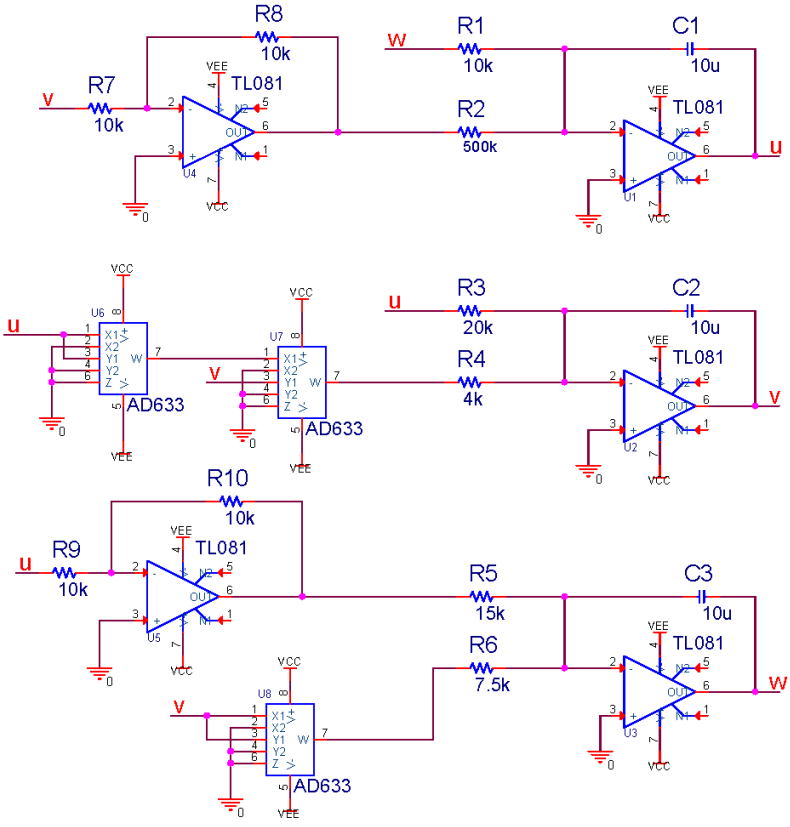
Şekil 1.27 Poincare kesitinin Ruelle grafiği

1.3.6. Önerilen Sistemin Elektronik Devre Gerçekleştirimi

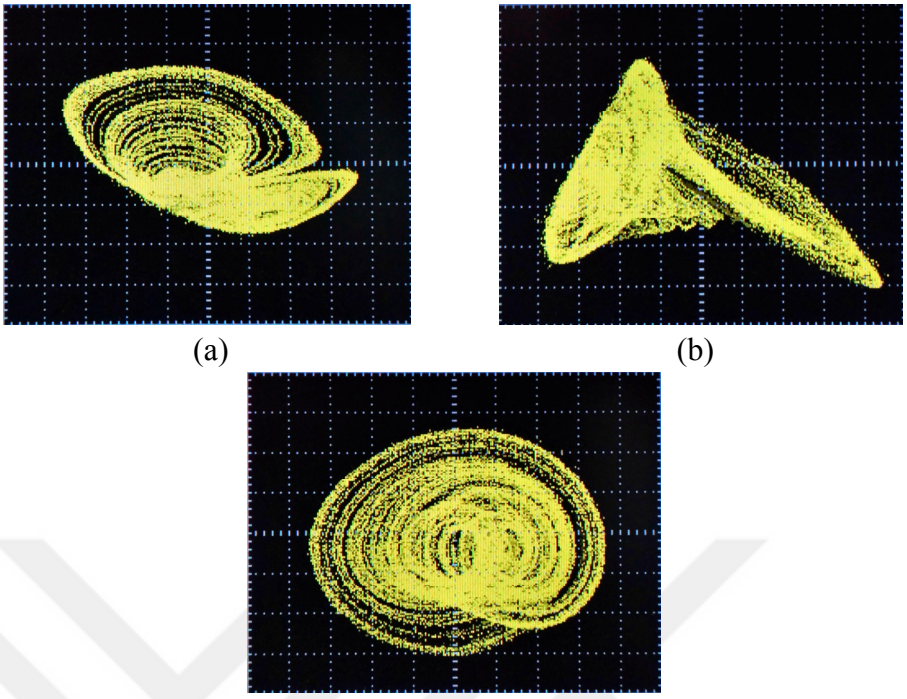
Eşitlik (1.10) ile verilen eşitlikler boyutsuzdur yani diferansiyel denklem sisteminin yörünge değerlerinin fiziksel bir karşılığı yoktur. Fakat, bu sistemin elektronik devre gerçekleştirimini yapmak için yörünge değerleri volt cinsinden ifade edilmelidir ve analog bir devrede volt değerleri besleme gerilimleriyle sınırlıdır. Dolayısıyla, Eşitlik (1.10)'un çözümleri bu besleme gerilimleri arasında olmalıdır. Kurulacak devrede besleme gerilimleri $\mp 12V$ ile sınırlandırılacaktır. Buna uygun olarak, Eşitlik (1.10) şu dönüşümler kullanılarak skalalandırılmıştır: $u = x/5$, $v = y$, $w = z/5$. Skalalandırma sonucu elde edilen yeni sistem Eşitlik (1.14)'teki gibi elde edilmektedir.

$$\begin{aligned} \dot{u} &= 0.2v - aw \\ \dot{v} &= -5u - 25bu^2\mathrm{v} \\ \dot{w} &= c(u - 0.2v^2) \end{aligned} \tag{1.14}$$

Eşitlik (1.14)'ün devre gerçekleştirimi için TL081 opamp ve AD633 analog çarpıcı kullanılmıştır. Kurulan devre yapısı Şekil 1.28'de görüldüğü gibidir. Bu devre yapısının elektronik devre olarak kurulması neticesinde elde edilmiş osiloskop görüntüleri ise Şekil 1.29'da verilmektedir. Görüldüğü üzere osiloskopta elde edilen görüntüler daha önce elde edilmiş olan sonuçlarla uyumludur. Ayrıca, elektronik devre gerçekleştirimi ile kaotik istemin fiziksel olarak gerçekleştirimi yapılmış olup, böylelikle kaotik sistemin sadece simülasyonlarla elde edilen bir sonuç olmadığı gösterilmiştir. Önerilen kaotik sistemin bu fiziksel gerçekleştirimi GRSÜ ile rastgele bit üretimi veya güvenli analog haberleşme gibi uygulamalarda kullanılabilir.



Şekil 1.28 Elektronik devre gerçekleştiriminin şematiği



(a) (b)

(c)

Şekil 1.29 Deney düzeneğinden elde edilen osiloskop görüntüleri: (a) x-y düzlemi, (b) y-z düzlemi, (c) x-z düzlemi

2. BÖLÜM

KAOTİK SİSTEMLERİN GERÇEK YÖRÜNGELERİNİN LSB UZATMA METODU İLE DİJİTAL OLARAK ÜRETİLMESİ

Daha önceki bölümde sunulduğu üzere dijital platformlarda konvansiyonel aritmetik işlemleri kullanarak kaotik bir işaret elde etmek mümkün değildir. Fakat, literatürde bu konu üzerinde çalışmalar yapan araştırmacıların birçoğu bu konuyu ihmal ederek kaotik olmayan (sözde) işaretleri kaotikmiş gibi varsayarak kriptografik sistemler gibi güvenliğin üst düzeyde olması gereken tasarımlarda kullanmaktadır [39–44]. Oysaki, daha önceki bölümde belirtildiği üzere bu şekilde üretilen kaotik yörüngeler orijinal kaotik yörüngeler değildir [32, 38, 112]**.** Bu durum, hem önerilen kriptografik sistemin güvenilirliğini zedelemekte, hem de kaotik sistemlerin farklı disiplinlerdeki araştırmacılar arasında yeteri kadar anlaşılmadığı izlenimini uyandırmaktadır. Bu nedenle gerek eğitim gerek de uygulama safhalarında gerçek kaotik yörüngelerin dijital platformlar üzerinde üretilmesi hususuna ağırlık verilmesi gerekmektedir.

Literatürde gerçek kaotik yörüngelerin üretilmesi hakkındaki çalışmalar nispeten az sayıdadır. Fakat bu çalışmaların ortak noktası sadece belli tür kaotik haritalar üzerinde yoğunlaşmalarıdır. Bunun nedeni ise konvansiyonel dijital aritmetik işlemlerde mevcut olan yuvarlama hatalarının önüne geçebilmektir. Eğer bir kaotik haritanın tüm parametreleri ikinin katıysa ve diğer tüm işlemler yuvarlamaya neden olmuyorsa bu tür haritaları "ikili kaydırmalı kaotik haritalar" (İKKH) olarak adlandırdık. Tanımı gereği bir İKKH yuvarlama hatalarına maruz kalmaz. Fakat, konvansiyonel dijital aritmetik işlemleri kullanarak bu yapıların benzetimini yapmak da mümkün değildir. Çünkü bu tür kaotik sistemlerin sabit veya kayan noktalı gerçekleştirimleri ikili kaydırmalar (binary shift) yüzünden daima sıfıra yakınsamaktadır [57]. Literatürde bu tür haritalar kullanılırken gerçekleştirimin sıfıra yakınsamaması için bazı çalışmalarda harita modifiye edilmektedir [113–117]. Diğer bazı çalışmalarda ise ikinin katı olan parametre

değerlerinden biri çok az değiştirilerek kullanılmakta veya ikinin katı olmayan başka bir parametre değeri seçilmektedir [118–122]. Fakat, her iki durumda da yörüngelerin eldesi yuvarlama hatalarının hesaplamalara eklenmesiyle mümkün olmaktadır ve bu nedenle üretilen yörüngeler daha önce bahsetmiş olduğumuz sözde yörüngelerdir.

Bu tür simülasyon sonuçlarının sıfıra yakınsamasının nedeni bir önceki bölümde ifade edildiği üzere dijital platformlar üzerinde irrasyonel sayıların ifade edilememesidir. Bu nedenle, gerçek yörüngelerin üretilmesi hakkındaki çalışmalar benzer İKKH'ların yörüngelerini elde etmek için irrasyonel sayıları dolaylı olarak kullanmaya dayanmaktadır. Örneğin, [123]'de yapılan çalışmada Bernoulli haritasının gerçek yörüngelerini elde edebilmek için başlangıç şartlarının hesaplanabilir sayılar olarak seçilmesi önerilmiştir. Hesaplanabilir sayılar, sonlu adımlarla virgülden sonraki basamakları hesaplanabilen sayılardır [124]. Bu yöntem ile teorik olarak gerçek yörüngeler hesaplanabilir olsa da pratikte çok başarılı görünmemektedir. Öncelikle hesaplanabilir bir sayının basamaklarını belirlemek için bir algoritma gerekmektedir ve bu nedenle tüm hesaplanabilir sayıların kümesi sayılabilir olmak zorundadır. Oysaki sayılabilir çokluktaki bir küme matematiksel olarak ihmal edilebilir bir kümedir. Dahası sadece π ve e gibi bilinen hesaplanabilir sayılar için bu tür algoritmalar mevcut olup artan basamaklarla beraber algoritmaların doğruluğu azalırken, yapılan işlemlerin karmaşıklığı ise artmaktadır [125]. Bu nedenle, bu yöntem sadece belli başlangıç koşullarına sahip yörüngelerin sınırlı sayıdaki iterasyonları için kullanılabilir. Ayrıca, Bernoulli haritası dışında farklı aritmetik işlemleri de içeren İKKH'lar için kullanılmaz.

Yakın zamanda yapılan bir çalışmada ise 1-boyutlu kısmi doğrusal kaotik haritaların gerçek yörüngelerini hesaplamak için lineer fraksiyonel haritalara dayalı bir metot önerilmiştir [126]**.** Bu metotta, kaotik haritalar lineer fraksiyonel haritalarla temsil edilmekte ve yörünge ise kübik polinomların reel kökü olarak ifade edilmektedir. Bu yöntemde ise lineer fraksiyonel haritaların katsayıları çok çabuk büyümekte ve her iterasyonda kübik polinomun kökünün bulunması gerekmektedir. Bu nedenle bu yöntem de sadece sınırlı uzunluktaki gerçek yörüngenin hesaplanması için kullanılabilir ve yavaştır. Dolayısıyla, donanım gerçekleştirimleri için uygun değildir. Ayrıca, bu yöntemle ifade edilebilen irrasyonel sayılar cebirsel olarak adlandırılır ve cebirsel sayılar tüm irrasyonel sayılara kıyasla sayılabilir sonsuzlukta bir küme meydana getirirler. Yani, irrasyonel sayıların neredeyse tümü cebirsel değil transandantal sayılardır [127].

Bahsedilen bu metotların kısıtlamaları ve pratik gerçekleştirimlere uygun olmaması yüzünden araştırmacılar kaotik sistemlerin dijital gerçekleştirimlerinde bu metotlara başvurmadan doğrudan sabit veya kayan noktalı aritmetiği kullanmaktadır. Netice olarak da yukarıda verilen örneklerde olduğu gibi sözde yörüngeler hatalı bir şekilde gerçek kaotik yörüngelermiş gibi kullanılmaktadır [39–44]. Bu durumun önüne geçebilmek adına, kaotik sistemlerin gerçek yörüngelerinin dijital olarak gerçekleştirimi için daha basit metotlara ihtiyaç vardır. Gerçek kaotik yörüngelerin dijital olarak üretilmesi için daha basit ve doğrudan bir yöntem ise ikili kaydırmalar neticesinde en az anlamlı bitlerde (LSB) oluşan sıfırları rastgele veya tekrar eden bit paternleriyle değiştirmektir. Literatürde buna benzer bir uygulama yapan az sayıda çalışma mevcuttur [128–130]. [128] ve [129]'da kaydırmalı bir kaydedicinin (shift register) kaydedici kısmı Bernoulli haritasının bir yaklaştırımı olarak kabul edilerek gerçek periyodik yörüngeler üretilmektedir. [130]'de is lineer geribeslemeli bir kaydedicinin (LFSR) kaydedici kısmı Bernoulli haritasının bir yaklaştırımı olarak kabul edilmektedir. Bu devreye ek lojik devreler eklenerek çadır haritası gibi farklı İKKH'lar elde edilmektedir.

Bahsedilen çalışmaların hepsinde gerçek periyodik yörüngeler üretilerek bu yörüngelerin sınırlı çözünürlükteki kısımları kaydedilmektedir; gerçek kaotik yörüngeler üretilmemektedir. Bu bölümde yaptığımız çalışmada ise yukarıda kullanılan metotlar genelleştirilerek farklı İKKH'ların gerçek kaotik ve periyodik yörüngelerinin elde edilmesi için hem yazılımsal hem de donanımsal gerçekleştirimlere uygun algoritmalar verilmektedir. Genelleştirilen bu metot LSB uzatma metodu olarak adlandırılmıştır. Bunun yanı sıra, topolojik konjugeler kurularak lojistik ve Chebyshev haritaları gibi İKKH olmayan kaotik haritaların da gerçek periyodik ve kaotik yörüngeleri elde edilmektedir. Bildiğimiz kadarıyla bu tür kaotik haritaların gerçek yörüngelerinin elde edilmesi bakımından bu çalışma bir ilktir [56]. Özellikle lojistik haritanın en çok çalışılan kaotik haritalardan biri olmasına karşın gerçek yörüngelerinin daha önce dijital olarak elde edilmemiş olması açısından da yapılan bu çalışma ilgi çekicidir.

2.1. İkili Kaydırmalı Kaotik Haritalar

Yukarıda bahsetmiş olduğumuz üzere ikili kaydırmalı kaotik haritalar (İKKH) tüm parametreleri ikinin katı olan ve yuvarlama hatalarına neden olmayan aritmetik işlemlerle gerçekleştirilebilen harita türleridir. Bu tür haritaların parametre değerleri ikinin katı

olduğu için bu parametrelerle çarpma veya bölme işlemi ikili kaydırmalara denk gelmektedir. Dolayısıyla, bu haritalarla yapılacak işlemleri daha iyi anlayabilmek adına öncelikle sayıların ikili olarak nasıl temsil edildiğini anlamak gerekir. Bu bağlamda, birim aralıkta tanımlı $x \in [0,1]$ sayısını inceleyelim. Bu sayı $b_i \in \{0,1\}$ olmak üzere ikili olarak $x = (0.b_1 b_2 \dots b_n \dots)_2$ şeklinde ifade edilir ve bu gösterim tarzının Tablo 2.1'de gösterildiği üzere olası üç çeşidi bulunmaktadır.

Tablo 2.1 Birim aralıktaki sayıların ikili gösterimleri

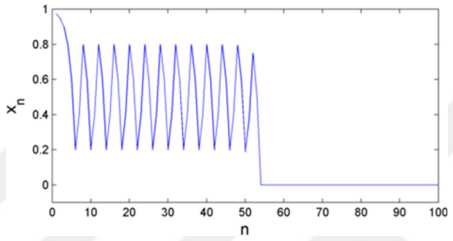
Tip	İkili Gösterim	Açıklama
i	$x = (0.b_1 b_2 \dots b_N)_2$	Sayı sonlu bitlerle ifade edilir.
ii	$x = (0.b_1 \dots b_k b_{k+1} \dots b_{k+N-1} b_k \dots)_2 = \left(0.b_1 \dots \overline{b_k b_{k+1} \dots b_{k+N-1}}\right)_2$	Sayı sonsuz bitle ifade edilir ve *N* bit kendini tekrar eder.
iii	$x = (0.b_1 b_2 b_3 \dots)_2$	Sayı sonsuz bitle ifade edilir ve bitler kendini tekrar etmez.

Tablo 2.1'de tip (i) ve (ii) ile ifade edilen sayılar rasyonel sayılara karşılık gelirken, tip (iii) ile ifade edilen sayılar irrasyonel sayılara karşılık gelmektedir [131]. İkinin katı olan sayılarla çarpıldığında bu gösterim ya sağa ya da sola kayar. Şimdi bu gösterim tarzı kullanılarak İKKH'ları inceleyebiliriz. İlk olarak inceleyeceğimiz İKKH, Bernoulli haritası $S: [0,1) \to [0,1)$ olup Eşitlik (2.1)'de verilmektedir.

$$S(x) = 2x \ (mod\ 1) = \begin{cases} 2x, & 0 \le x < 1/2 \\ 2x - 1, & 1/2 \le x < 1 \end{cases} \tag{2.1}$$

Her $x \in (0,1) - \{1/2\}$ için $|S'(x)| = 2$ olduğundan dolayı, süreksizlik noktası olan $1/2$'den geçmeyen her periyodik olmayan yörünge için Lyapunov üsteli $ln2$ olacaktır. Bu ise, $1/2$ noktası sabit nokta olan sıfıra haritalandığından dolayı, tüm periyodik olmayan yörüngelerin kaotik olduğunu göstermektedir. Ayrıca, S haritası ikili tabanda sola kaydırmaya eşdeğer olduğundan $S(0.b_1 b_2 b_3 \dots)_2 = (0.b_2 b_3 b_4 \dots)_2$ olacaktır. Bu da Tablo 2.1'deki tip (i) sayılara ait yörüngelerin tümünün eninde sonunda sıfıra yakınsayacağı anlamına gelir. Tip (ii) sayıların yörüngeleri ise transient kısmından sonra periyodu *N* olan yörüngeler olacaktır. Son olarak, yalnızca tip (iii) sayılar periyodik

olmayan kaotik yörüngeleri verecektir. Bu durum, neden Bernoulli haritasının sabit veya kayan noktalı gerçekleştirimlerinin sıfıra yakınsamak zorunda olduğunu açıklamaktadır: Dijital bir platformda yalnızca tip (i) sayılar mevcuttur ve bu sayılar sonlu bir hafızada bitler şeklinde saklanır. Bu haritanın ikili kaydırmaları neticesinde hafıza elemanındaki MSB biti atılırken LSB biti sıfır ile doldurulur. Böylelikle belli bir iterasyondan sonra hafıza elemanı tamamen sıfırlarla dolar ve yörünge sıfıra yakınsamış olur. Bernoulli haritasının tipik bir bilgisayar simülasyonu örneği Şekil 2.1'de görülmektedir. Beklendiği üzere simülasyon bir süre sonra sıfıra yakınsamaktadır.



Şekil 2.1 Bernoulli haritasının tipik bir simülasyonu

Eşitlik (1.2)'de verilmiş olan çadır haritası T de bir diğer İKKH'dır. Süreksizlik noktası dışında Bernoulli haritası için geçerli olan tüm sonuçlar çadır haritası için de geçerlidir. Eğer $x < 1/2$ ise, harita tıpkı Bernoulli haritası gibi sayıyı sola kaydırır. Eğer $x \geq 1/2$ ise, sayı öncelikle $(1 - x)$ işlemi ile sıfıra yakınlaştırılır ve sonra sola kaydırılır. Buradan görüleceği üzere, tip (i) sayılar yine sıfıra yakınsarken, tip (ii) sayılar periyodik ve tip (iii) sayılar kaotik olacaktır. Dolayısıyla, bu haritanın da konvansiyonel dijital gerçekleştirimleri sıfıra yakınsamak zorundadır.

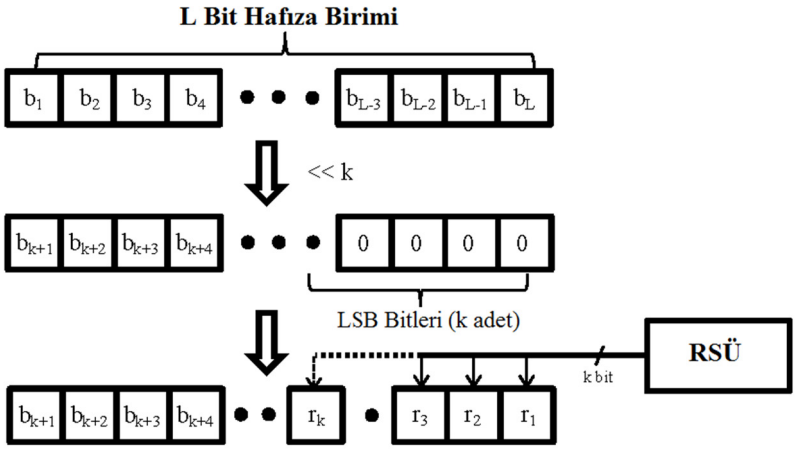
Son olarak inceleyeceğimiz İKKH, öncekilerin aksine 2 boyutlu (2B) olup baker (fırıncı) haritası olarak adlandırılmaktadır. I^2: $[0,1) \times [0,1)$ olmak üzere, baker haritası $B: I^2 \rightarrow I^2$ Eşitlik (2.2)'deki gibi ifade edilmektedir.

$$B(x,y) = \begin{cases} (2x, y/2), & 0 \leq x < 1/2 \\ (2x-1, y/2+1/2), & 1/2 \leq x < 1 \end{cases} \quad (2.2)$$

Dikkat edilirse, harita x boyutunda y değerlerinden bağımsız olarak Bernoulli haritasının aynısıdır. Aynı zamanda, y boyutundaki değerler de x değerlerine bağlı olarak belirlenir. Dolayısıyla, Bernoulli haritasına ait olan tüm özellikler 2B için genelleştirilmiş olup elde ettiğimiz önceki sonuçlar baker haritası için de aynen geçerlidir.

2.2. LSB Uzatma Metodu ve Dijital Gerçekleştirim Algoritmaları

Görmüş olduğumuz üzere sonlu çözünürlük yüzünden incelemiş olduğumuz haritaların hepsinin yörüngeleri sıfıra yakınsamaktadır. Bu çalışmada önerdiğimiz metot dijital bir platformda kaydırma işlemleri sonunda hafıza elemanının sıfır ile dolan LSB bitlerinin rastgele veya [128]'deki gibi kendini tekrar eden bit dizileri ile doldurulması prensibine dayanmaktadır. Metodun genel işleyiş prensibi Şekil 2.2'deki gibidir.



Şekil 2.2 LSB uzatma metodunun işleyiş prensibi

Şekil 2.2'de görülebileceği üzere LSB uzatma metoduna ait tüm işlemler L bitlik hafıza birimi üzerinde yapılmaktadır. Bu L bitlik hafıza birimi $x = (0.b_1b_2 \dots b_L \dots)_2$ şeklindeki bir sayının ilk L bitini temsil etmektedir. Bu şekilde bir kullanım başlangıç şartının ve yörüngenin mevcut değerinin 2^{-L}'lik bir çözünürlük ile belirlenebilmesini sağlamaktadır. Bu metot ile hafıza birimi üzerinde yapılan kaydırmalar ve birtakım işlemler neticesinde sondaki k adet LSB bitinin sıfır ile dolduğunu varsayalım. Bu durumda, bu LSB bitleri bir rastgele sayı üreteci (RSÜ) veya kendini tekrar eden bit paternleri ile doldurulur. Bu sayede, eğer RSÜ bir SRSÜ (Sözde Rastgele Sayı Üreteci) ise veya kendini tekrar eden bit paternleri kullanılıyorsa başlangıç şartı tip (ii) olan bir yörünge elde edilir. Eğer bir GRSÜ (Gerçek Rastgele Sayı Üreteci) kullanılıyorsa, başlangıç şartı tip (iii) olan bir yörünge elde edilir. Her iki durum için de çoğu zaman başlangıç şartının net değerini bilmek her zaman için mümkün olmayabilir. İşlemler yalnızca L bitlik bir hafıza birimi üzerinde gerçekleştirildiğinden, RSÜ'nün bit dizisi başlangıç şartına eşit olmak zorunda değildir. Bu gibi durumlarda elde edilen yörüngeye karşılık gelen bir başlangıç şartının olduğu varsayımı yapılır ve bu başlangıç şartının da ancak ilk L biti tam olarak bilinebilir.

LSB uzatma metodundaki en önemli kısım ise hafıza birimi üzerinde yapılacak olan işlemleri belirlemektir. Bahsetmiş olduğumuz üzere, İKKH gerçekleştirimleri yuvarlama hatalarına neden olacak herhangi bir aritmetik işleme sahip olmamalıdır. Bu nedenle bu tür gerçekleştirimler için basit lojik işlemlere dayalı algoritmalar gerekmektedir. Bu amaç doğrultusunda önceki kısımda incelenmiş İKKH'ların her biri için bir dijital gerçekleştirim algoritması hazırlanmıştır. Bernoulli haritasının dijital gerçekleştirimi için hazırlanmış olan algoritma Algoritma 2.1'de verilmektedir.

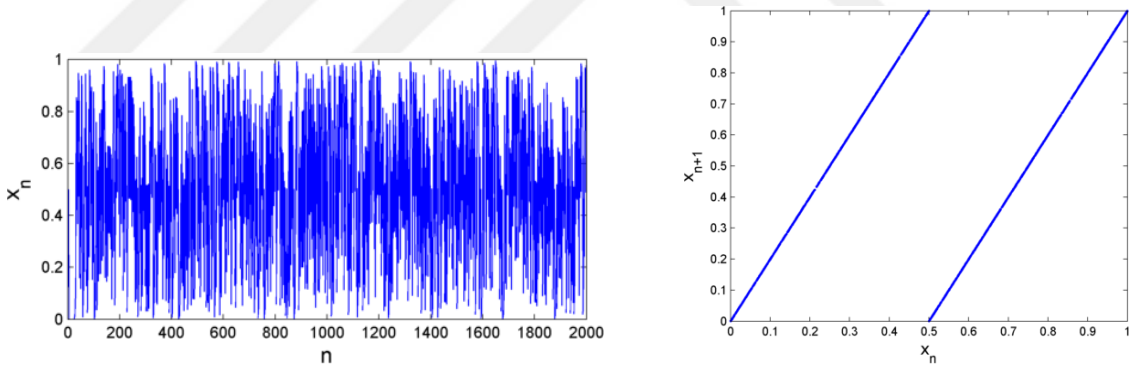
Algoritma 2.1 Bernoulli haritasının LSB uzatma metodu ile gerçekleştirimi

```
function Bernoulli(initval,N)

input: initval  // [0,1) arasında kayan noktalı bir sayı
       N  // pozitif bir tam sayı
output: orbit // 1xN boyutunda bir tam sayı dizisi
variable: memunit // L bit işaretsiz (unsigned) tam sayı

begin function
       orbit[1]=initval
       memunit := floor(2^L*initval) // L bit işaretsiz tam sayıya dönüştür
       for i=1 to N-1
              memunit := memunit << 1
              if rand() == 1
                     memunit := memunit bitwise or 1
              end if
              orbit[i+1] := 2^-L*double(memunit)
       end for
end function
```

Algoritma 2.1'de *L* bit hafıza birimi "memunit" değişkeni olarak adlandırılmıştır ve *L* bit işaretsiz (unsigned) tam sayıya karşılık gelmektedir. Tüm işlemler bu tam sayı üzerinde gerçekleştirilmektedir. *L* platforma bağlı olarak 8, 16, 32, 64 vs. olabilir. Hafıza biriminin başlangıç değeri "initval" ve yapılacak iterasyon miktarı "N" kullanıcı tarafından girilir. Girilen bu başlangıç değeri üretilecek yörüngenin asıl başlangıç değerinin ilk *L* bitine karşılık gelmektedir. Başlangıç şartının geri kalan kısmı RSÜ tarafından belirlenecektir. Eşitlik (2.1)'deki Bernoulli haritası ikili tabanda tek bir sola kaydırma işleminden ibarettir ve bu işlem "<< 1" ile sembolize edilir. "rand()" fonksiyonu ise her iterasyonda RSÜ tarafından üretilen bitin değerini döndüren bir fonksiyonu temsil etmektedir. "floor()" fonksiyonu kayan noktalı bir sayının küsuratlı kısmını atmak için kullanılırken, "double"

fonksiyonu bir sayının kayan noktalı türe çevrilmesi için kullanılmaktadır. Önerilen algoritmada LSB bitinin yeni değeri "bitwise or" işlemi ile belirlense de bu amaçla diğer herhangi bir yöntem kullanılabilir.

Algoritma 2.1 MATLAB üzerinde MATLAB'in kendi içindeki SRSÜ'nün RSÜ olarak kullanılmasıyla dijital olarak gerçekleştirilmiştir. Gerçekleştirimde $L = 32$ olarak alınmıştır. SRSÜ kullanıldığı için üretilen yörüngenin başlangıç şartı tip (ii) bir sayıya karşılık gelmektedir. Dolayısıyla, üretilen yörünge periyodik bir yörünge olmak zorundadır. Hesaplamalar esnasında yuvarlama hataları veya başka hiçbir hata hesaba dahil olmadığı için üretilen yörünge orijinal matematiksel model tarafından öngörülen "gerçek" yörüngelerdendir. Daha net ifade etmek gerekirse, yörünge periyodik olduğu için üretilen yörünge Bernoulli haritasının gerçek periyodik yörüngesidir. Birinci bölümden hatırlanacağı üzere, bir kaotik sistem sayılabilir sonsuzlukta periyodik yörüngeye sahiptir. Üretilen yörünge ve bu yörüngeye ait çeker yapısı Şekil 2.3'de görüldüğü gibidir.



(a) (b)

Şekil 2.3 (a) Bernoulli haritasının gerçek periyodik yörüngesi; (b) Gerçek periyodik yörüngeye ait çeker yapısı

MATLAB'in kendi içindeki SRSÜ "Mersenne twister" olarak adlandırılmaktadır ve bu üretecin periyodu (2^{19937}-1) olduğundan üretilen gerçek periyodik yörüngenin de periyodu aynı olacaktır. Her ne kadar bu periyot değeri çok büyük gibi görünse de Sharkovskii teoremine göre Bernoulli haritası (periyot-3 yörüngeye sahip olduğundan) akla gelebilecek her periyot değeri için bir periyodik yörüngeye sahiptir [64, 132].

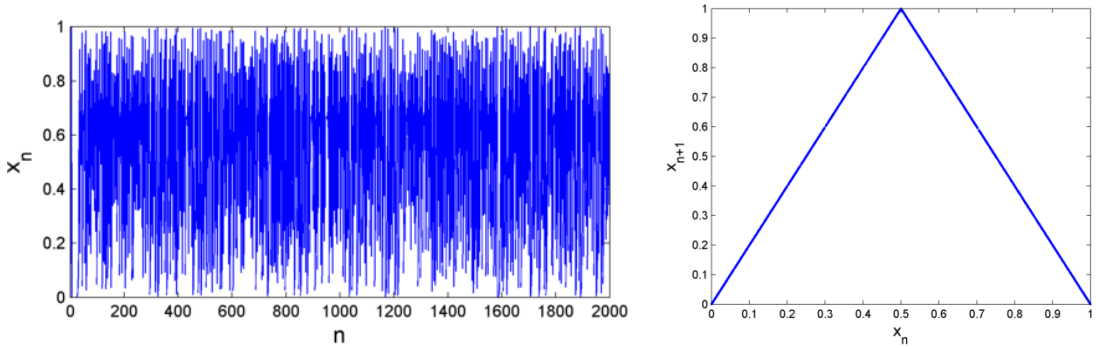
Çadır haritası için hazırlanmış olan dijital gerçekleştirim algoritması Algoritma 2.2'de verilmektedir. Algoritma 2.2 Bernoulli için hazırlanmış olanla temel olarak aynıdır. Tek fark Eşitlik (1.2)'deki $(1 - x)$ işleminin gerçekleştirimindedir. Normalde dijital

platformlarda $(1 - x)$ işlemi x sayısının ikinin tümleyeni alınarak gerçekleştirilir. Fakat, LSB uzatma metodunda sondaki bit sürekli uzatıldığı için bu işlem gerçekleştirilirken hafıza biriminin sadece tümleyenini ("bitwise NOT") almak yeterlidir. Bu şekilde tasarlanmış olan dijital gerçekleştirim algoritması Algoritma 2.2'deki gibidir.

Algoritma 2.2 Çadır haritasının LSB uzatma metodu ile gerçekleştirimi

```
function Tent(initval,N)

input: initval  // [0,1) arasında kayan noktalı bir sayı
       N  // pozitif bir tam sayı
output: orbit // 1xN boyutunda bir tam sayı dizisi
variable: memunit // L bit işaretsiz (unsigned) tam sayı
constant: compval := 2^(L-1)  // L bit işaretsiz (unsigned) tam sayı

begin function
       orbit[1]=initval
       memunit := floor(2^L*initval) // L bit işaretsiz tam sayıya dönüştür
       for i=1 to N-1
              if memunit < compval
                     memunit := memunit << 1
              else
                     memunit := bitwise not memunit
                     memunit := memunit << 1
              end if
              if rand() == 1
                     memunit := memunit bitwise or 1
              end if
              orbit[i+1] := 2^-L*double(memunit)
       end for
end function
```

Algoritma 2.2'deki bir diğer farklılık ise "compval" sabitinin kullanılmasıdır. Bu sabit Eşitlik (1.2)'deki $1/2$'den küçük veya büyük olma durumlarını ayırt etmek için kullanılmaktadır. Görülebileceği üzere algoritma sadece tümleyen alma ve kaydırma gibi basit lojik işlemlere dayanmaktadır. Bu nedenle de üretilen yörüngeler yine gerçek yörüngeler olmalıdır. Ayrıca, RSÜ olarak GRSÜ kullanılırsa periyodik yörüngelerin elde edileceği bir kez daha görülmektedir. Buna göre, $L = 32$ için Algoritma 2.2'nin yine MATLAB üzerindeki gerçekleştirimi sonucu elde edilmiş gerçek periyodik yörüngesi ve bu yörüngeye ait çeker yapısı Şekil 2.4'te verilmiştir.



(a) (b)

Şekil 2.4 (a) Çadır haritasının gerçek periyodik yörüngesi; (b) Gerçek periyodik yörüngeye ait çeker yapısı

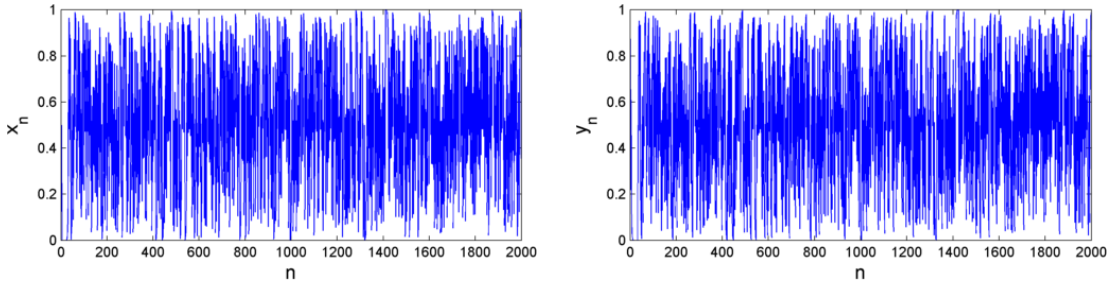
Son olarak, 2B baker haritasının LSB uzatma metoduna dayalı dijital gerçekleştirim algoritması Algoritma 2.3'te verilmiştir. Bu sefer harita iki boyutlu olduğu için iki adet hafıza birimi ("memunit") kullanılmıştır. Eşitlik (2.2)'den görülebileceği üzere baker haritasının x boyutu Bernoulli haritası ile aynıdır. Bu nedenle, Algoritma 2.3'teki x boyutu gerçekleştirimi de Algoritma 2.1'dekiyle aynıdır. Yine aynı eşitliklerden görülebileceği üzere y boyutundaki gerçekleştirim x boyutundaki değerlere bağlı olarak sağa kaydırma (">> 1") işlemi yapılması prensibine dayanmaktadır. Eğer $x > 1/2$ ise y değeri sağa kaydırılmaktadır. Aksi takdirde ise $y + 1$ değeri sağa kaydırılmaktadır. $y + 1$ değerinin sağa kaydırılması ise y'nin sağa kaydırılmasından sonra MSB bitinin 1 yapılması işlemiyle aynıdır. Dolayısıyla, Algoritma 2.3'te bu bağıntılar kullanılmıştır. MSB bitinin 1 yapılması "memunit2" ile "compval" sabitinin "bitwise or" işlemine tabi tutulmasıyla sağlanmıştır.

Algoritma 2.3 Baker haritasının LSB uzatma metodu ile gerçekleştirimi

```
function Baker(initval,N)

input: initval  // [0,1) arasındaki kayan noktalı sayıların 1x2'lik dizisi
       N  // pozitif bir tam sayı
output: orbit // kayan noktalı sayılara ait 2xN'lik bir dizi
variable: memunit1, memunit2 // L bit işaretsiz tam sayılar
constant: compval := 2^(L-1)  // L bit işaretsiz tam sayı

begin function
       orbit[1][1]=initval[1]
       orbit[2][1]=initval[2]
       memunit1 := floor(2^L*initval[1]) // L bit işaretsiz tam sayıya dönüştür
```

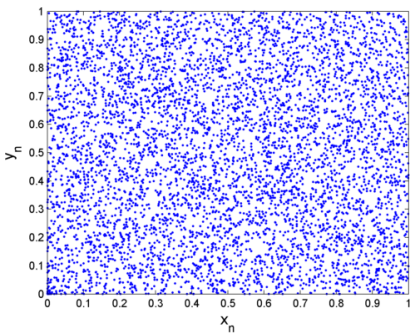
```
        memunit2 := floor(2^L*initval[2]) // L bit işaretsiz tam sayıya dönüştür
        for i=1 to N-1
                // y-dimension:
                memunit2 := memunit2 >> 1
                if memunit1 >= compval
                        memunit2 := memunit2 bitwise or compval
                end if
                // x-dimension:
                memunit1 := memunit1 << 1
                if rand() == 1
                        memunit1 := memunit1 bitwise or 1
                end if
                orbit[1][i+1] := 2^-L*double(memunit1)
                orbit[2][i+1] := 2^-L*double(memunit2)
        end for
end function
```

Algoritma 2.3'ün de $L = 32$ için MATLAB gerçekleştirimi yapılmıştır. Algoritma yine basit lojik işlemlere dayanmaktadır. SRSÜ kullanıldığı için de yine gerçek periyodik yörüngeler elde edilmiştir. Elde edilen gerçek periyodik yörüngeye ait görüntüler Şekil 2.5'de verilirken, bu gerçek periyodik yörüngenin çekeri Şekil 2.6'da gösterilmektedir.



(a) (b)

Şekil 2.5 Baker haritasının gerçek periyodik yörüngesi: (a) x boyutu; (b) y boyutu



Şekil 2.6 Baker haritasının periyodik yörüngesine ait çeker yapısı

Bu tür İKKH'ların gerçek kaotik yörüngelerini üretebilmek için tip (iii) sayılara, yani RSÜ olarak GRSÜ kullanımına ihtiyaç bulunmaktadır. Dijital platformlarda gerçek rastgele sayı üretebilmek içinse bir takım özel donanımlara ihtiyaç bulunmaktadır. Bu nedenle, gerçek kaotik yörüngelerin üretimi Bölüm 2.4'te donanım ile yapılacaktır. Nitekim, bu kısımda verilen tüm algoritmalar basit lojik işlemlere dayalı olduklarından bunlar donanım gerçekleştirimlerine de son derece uygundurlar ve aynı şekilde basitçe donanım üzerinde de gerçekleştirilebilirler.

2.3. Diğer Haritaların Gerçek Yörüngelerinin Topolojik Konjuge ile Üretilmesi

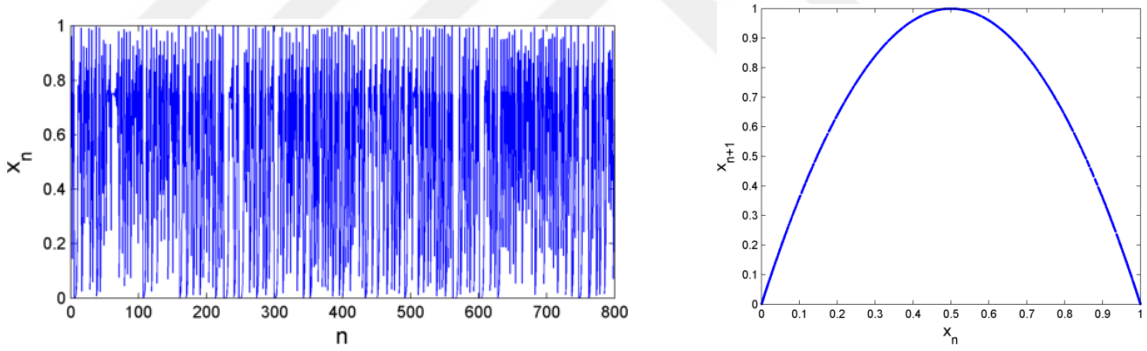
Topolojik konjuge kavramı Bölüm 1.1'de açıklanmıştı. Kısaca tekrar etmek gerekirse, $f: A \to A$ ve $g: B \to B$ olmak üzere $h \circ f = g \circ h$ şartını sağlayan bir $h: A \to B$ homeomorfizmi elde edilebiliyorsa f ve g konjugedir denilir. Eğer, $h: A \to B$ sürekli fonksiyonu birebir, örten ve tersi de sürekli ise homeomorfizm olarak adlandırılır. Bu tanımdan h'ın birebir olma şartı çıkarılırsa, f ve g yarı-konjugedir denilir. Bu durumda h artık terslendirilebilir olmasa da $h \circ f^k = g^k \circ h$ şartı sağlandığından iki fonksiyonun yörüngeleri yine aynı özellikleri gösterecektir. Bu kısımda, İKKH ve diğer haritalar arasındaki topolojik konjuge veya yarı-konjuge bağıntılarından yararlanarak İKKH olmayan haritaların da gerçek yörüngeleri üretilmektedir. Normalde bu tür haritaların dijital gerçekleştirimlerinde sonlu çözünürlük ve yuvarlama hataları yüzünden gerçek yörüngeler elde edilemese de LSB uzatma metodu ile üretilen gerçek yörüngelere konjuge veya yarı-konjuge bağıntıları uygulandığında bu tür haritaların da gerçek yörüngelerinin eldesi mümkün olmaktadır. Bu LSB uzatma metodunun güçlü yanlarından biridir.

2.3.1. Lojistik Haritanın gerçek Yörüngelerinin Elde Edilmesi

İlk olarak Eşitlik (1.1)'de $G: [0,1] \to [0,1]$ ile verilen lojistik haritanın $k = 4$ parametre değeri için gerçek yörüngeleri elde edilecektir. Bu harita ile İKKH olan ve Eşitlik (1.2)'de verilen çadır haritası arasında Eşitlik (2.3)'deki h fonksiyonu kullanılarak bir konjuge ilişkisi kurulabileceği bilinmektedir [2].

$$h(x) = \frac{(1 - cos(\pi x))}{2} \tag{2.3}$$

Lojistik haritanın bir gerçek yörüngesini hesaplayabilmek için öncelikle çadır haritasının $\{x_i\}_{i=1}^{N}$ şeklinde bir gerçek yörüngesi Algoritma 2.2 kullanılarak bulunmalıdır. Daha sonra, $h \circ T^k = G^k \circ h$ bağıntısı kullanılarak lojistik haritanın yörüngesi $\{h(x_i)\}_{i=1}^{N}$ şeklinde hesaplanabilir. Bu dolaylı hesaplamadaki limitasyonlar hafıza biriminin çözünürlüğü olan L ve h fonksiyonunun değerinin hesaplanmasındaki hatalardır. Fakat, bunlar sadece yörünge değerlerinin temsil edilmesinde geçerli olup, asıl iterasyon hesabına etki etmemektedirler. Bu tür hatalar mevcut hesaplama sürecine etki edip sonraki hesaplamaları değiştirmediği için de elde edilen yörünge yine gerçek yörünge olmaktadır. Topolojik konjuge bağıntısının özelliğinden dolayı, eğer İKKH'nın yörüngesi kaotikse, elde edilen yörünge de kaotik olacaktır. Aynı şekilde, eğer İKKH'nın yörüngesi periyodikse, elde edilen yörünge de periyodik olacaktır. Buna göre, önceki kısımda çadır haritası için üretilen gerçek periyodik yörüngeye Eşitlik (2.3)'teki konjuge fonksiyonu uygulandığında, lojistik haritaya ait Şekil 2.7'deki gerçek periyodik yörünge ve çeker yapısı elde edilmiştir.

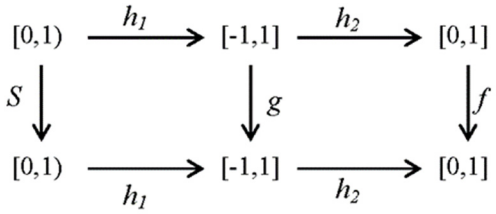


(a) (b)

Şekil 2.7 a) Lojistik haritanın gerçek periyodik yörüngesi; (b) Gerçek periyodik yörüngeye ait çeker yapısı

2.3.2. Chebyshev Haritalarının Gerçek Yörüngelerinin Elde Edilmesi

Başka haritaların gerçek yörüngeleri de benzer şekilde elde edilebilir. Bir başka örnek olarak Bernoulli haritası $h_1(x) = cos(2\pi x)$ ve $h_2(x) = (x+1)/2$ konjuge fonksiyonları ile birlikte kullanılmıştır. Oluşturulan konjuge bağıntıları Şekil 2.8'da özetlenmektedir.



Şekil 2.8 Oluşturulan konjuge bağıntıları

Bu bağıntılarda $g(x) = 2x^2 - 1$ için $h_1 \circ S = g \circ h_1$ şartının sağlandığını göstermek kolaydır. h_1'in sürekli olmasına rağmen birebir olmaması h_1'i yarı-konjuge yapmaktadır. Diğer taraftan, $f(x) = 4x^2 - 4x + 1$ için $h_2 \circ g = f \circ h_2$'dir ve h_2'nin homeomorfizm olmasından dolayı bu ikinci bağıntı bir konjuge bağıntısıdır. Dikkat edilirse, $g(x) = 2x^2 - 1$ bir Chebyshev polinomudur. Chebyshev polinomları Tanım 2.3.1'deki gibi tanımlanmaktadır.

Tanım 2.3.1. $p_n(cos(\theta)) = cos(n\theta)$ şartını sağlayan bir p_n polinomu birinci tür n'inci Chebyshev polinomu olarak adlandırılır [133].

Buna göre, eğer aynı konjuge fonksiyonları S^2 ile kullanılırsa, bu sefer $g_2(x) = 8x^4 - 8x^2 + 1$ ve $f_2(x) = 64x^4 - 128x^3 + 80x^2 - 16x + 1$ fonksiyonları elde edilmektedir. Dikkat edilirse g_2 de bir Chebyshev polinomudur. Önerme 2.3.2'nin öngörmüş olduğu üzere, bu durum bir tesadüf değildir. Bu önermenin sonucu olarak, Şekil 2.8'deki konjuge yapısında S yerine S^n seçildiğinde de konjuge bağıntıları geçerliliğini korumakta ve g_n ile ifade edilen yeni Chebyshev haritaları ile S^n arasında topolojik konjuge oluşmaktadır.

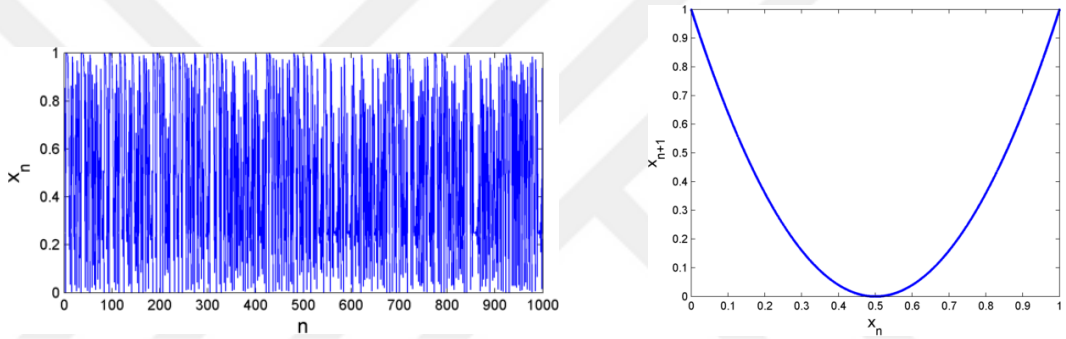
Önerme 2.3.2. $h_1(x) = cos(2\pi x)$ fonksiyonu S^n ve birinci tür 2^n'inci Chebyshev polinomları (g_n) arasında bir yarı-konjuge oluşturur.

İspat: Her n için $h_1(S^n)(x) = cos(2^{n+1}\pi x)$ şartı sağlanmaktadır. Buna göre, her n için yarım açı formülünü kullanarak $h_1 \circ S^n = g_n \circ h_1$ şartını sağlayan bir g_n bulabiliriz. Fakat, bu durumda $cos(2^n 2\pi x) = g_n(cos(2\pi x))$ şartı sağlanıyor olacağından, Tanım 2.3.1'e göre g_n birinci tür 2^n'inci Chebyshev polinomudur. ■

Böylelikle, her biri İKKH olan S^n haritaları ve g_n Chebyshev fonksiyonları arasında genelleştirilmiş bir bağıntı kurulmuş oldu. Bu nedenle, her g_n fonksiyonu aslında kaotik birer harita olmak zorundadır. Normalde h_2 fonksiyonunu kullanmadan bunların da gerçek yörüngelerini tıpkı lojistik haritanınkiler gibi elde edebiliriz. Fakat, g_n

haritalarının tanım kümesi $[-1,1]$ aralığında olup negatif sayıları donanım gerçekleştiriminde ifade etmek zordur. Bu yüzden, her g_n fonksiyonuna h_2 uygulanarak tanım aralığı $[0,1]$ olan yeni f_n haritaları elde edilmektedir.

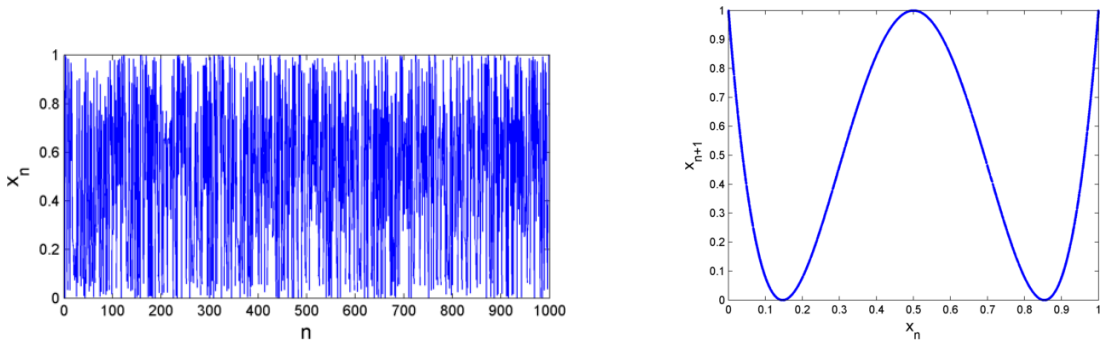
Buna göre, f_1 haritası en başta bulmuş olduğumuz $f(x) = 4x^2 - 4x + 1$ haritasına karşılık gelmektedir. Bu haritanın gerçek yörüngelerini elde etmek için öncelikle Bernoulli haritası S'nin $\{x_i\}_{i=1}^{N}$ şeklindeki bir gerçek yörüngesi Algoritma 2.1 yardımıyla hesaplanmalıdır. Ardından, f_1'in gerçek yörüngesi $\{h_2(h_1(x_i))\}_{i=1}^{N}$ işlemleri ile bulunur. Bu haritanın MATLAB gerçekleştirimi sonucu elde edilmiş olan gerçek periyodik yörüngesi ve bu yörüngeye ait çeker yapısı Şekil 2.9'da görüldüğü gibidir.



(a) (b)

Şekil 2.9 a) f_1 haritasının gerçek periyodik yörüngesi; (b) Gerçek periyodik yörüngeye ait çeker yapısı

Bu kurmuş olduğumuz genelleme ile yukarıdakine benzer şekilde herhangi bir f_n Chebyshev haritasının gerçek yörüngelerini elde etmek mümkündür. Mesela, $f_2(x) = 64x^4 - 128x^3 + 80x^2 - 16x + 1$ haritasının gerçek yörüngelerini elde etmek için öncelikle LSB uzatma metoduyla S^2'nin gerçek yörüngesinin üretilmesi gerekmektedir. Bu ise Algoritma 2.1'in uygun bir şekilde modifiye edilmesiyle (kaydırmanın iki bite çıkarılmasıyla) yapılabilir. Ardından, yine h_1 ve h_2 konjuge fonksiyonlarının uygulanmasıyla f_2 haritasının gerçek yörüngesi elde edilir. Bu şekilde yapılan MATLAB gerçekleştirimi sonucu elde edilen bir gerçek periyodik yörünge de gösterilmektedir.



(a) (b)

Şekil 2.10 a) f_2 haritasının gerçek periyodik yörüngesi; (b) Gerçek periyodik yörüngeye ait çeker yapısı

2.3.3. Gerçek Yörünge Üretiminde Yarı-Konjuge Kullanmanın Limitasyonları

Konjuge bağıntısı kullanımında herhangi bir problem olmamakla birlikte, gerçek periyodik yörüngeler yarı-konjuge altında üretiliyorsa farklı yörüngeler arasındaki periyot uzunlukları aynen korunmayabilir. Bunun nedeni yarı-konjuge bağıntıda birebir olma şartının mevcut olmamasıdır. Eğer iki yörünge arasında birebir ilişki bulunmazsa bir haritanın farklı değerleri diğer haritada aynı değere karşılık gelebilir. Örnek olarak bir önceki kısımdaki h_1 ile kurulmuş olan yarı-konjuge bağıntısını inceleyelim. h_1 fonksiyonu ikiye-bir fonksiyon olduğu için beklenen davranış yarı-konjuge ile elde edilen bazı periyodik yörüngelerin periyotlarının yarıya inmesidir. Yarıya inmeye neden olan başlangıç şartları $cos(2^{n+1}\pi x) = cos(2\pi x)$ eşitliğinin çözülmesiyle bulunabilir. Bu eşitlik bir Chebyshev haritasının n-inci iterasyonunun kendini tekrar edeceğini ifade etmektedir. Bu eşitliğin genel çözümleri $n > 0$ ve $k \in \mathbb{Z}$ olmak üzere, $x = k/(2^n - 1)$ ve $x = k/(2^n + 1)$'dir. Örnek olarak $k = 3$ ve $n = 2$ seçildiğinde, elde edilen başlangıç şartlarından biri $x_1 = 3/5 = (0.\overline{1001})_2$ olmaktadır. Eğer bu başlangıç şartına uygun olarak Algoritma 2.1'deki rand() fonksiyonu {1,0,0,1} dizisini tekrar ederek döndürürse, elde edilecek yörüngenin periyot uzunluğu 4 olacaktır. Fakat, $\{h_2(h_1(x_i))\}_{i=1}^{N}$ işlemi uygulanarak elde edilen yörüngenin periyot uzunluğu 2'dir.

2.4. Gerçek Yörüngelerin Donanım Üzerinde Üretilmesi

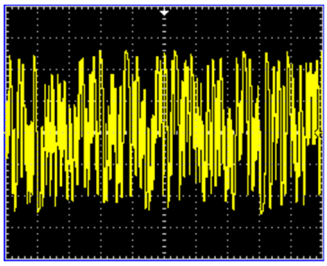
Daha önce bahsetmiş olduğumuz üzere önerilen algoritmalar basit lojik işlemlere dayalı olduklarından donanım gerçekleştirimlerine son derece uygundurlar. Ayrıca, gerçek kaotik yörüngeleri üretmek için bir GRSÜ'ne ve GRSÜ içinse özel donanımlara ihtiyaç

vardır. Bu nedenle, bir önceki kısımda verilen algoritmaların dijital donanım üzerinde gerçekleştirimleri bu bölümde yapılmaktadır. Öncelikle SRSÜ kullanılarak FPGA üzerinde gerçek periyodik yörüngeler üretilmekte ve sonrasında bir GRSÜ tasarımı yine FPGA üzerinde gerçekleştirilip LSB uzatma metodu ile gerçek kaotik yörüngeler üretilmektedir.

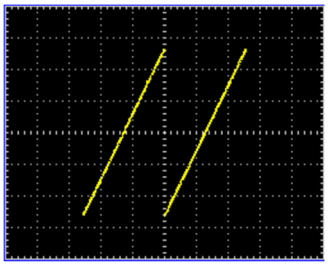
Bu gerçekleştirimlerde FPGA olarak Altera DE0 geliştirme bordu kullanılmıştır. Bu bord üzerinde bulunan Cyclone III FPGA elemanı 15408 lojik bloğa, 4 PLL'ye ve 484 G/Ç pinine sahiptir. Gerçekleştirimlerde VHDL (Very High Speed Integrated Circuit Hardware Description Language) donanım tanımlama dili kullanılmıştır.

2.4.1. Gerçek Periyodik Yörünge Üretimi

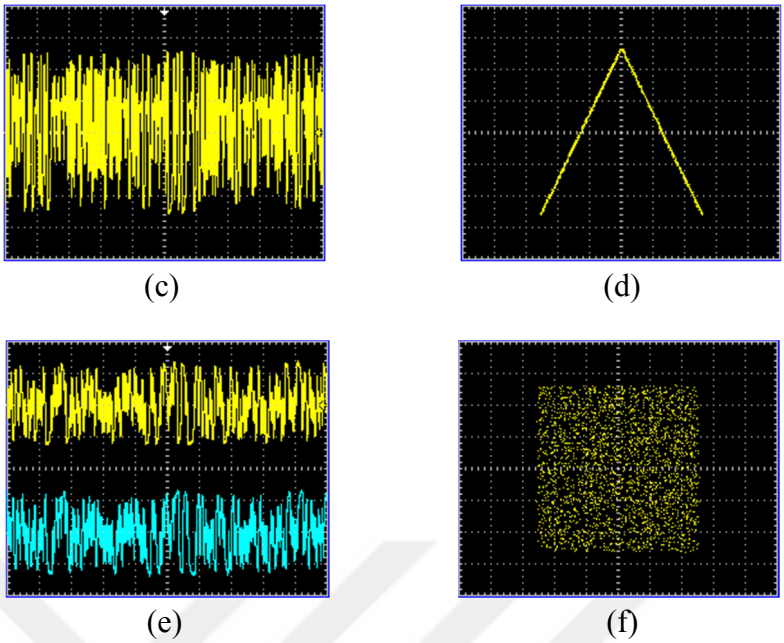
Önerilen algoritmalar XOR, NOT, kaydırma gibi basit lojik işlemlere dayalı olduğundan, bunları VHDL ile gerçekleştirmek oldukça kolaydır. VHDL ile hafıza biriminin uzunluğu L de istenildiği gibi seçilebilmektedir. Ayrıca, donanım gerçekleştiriminde yazılım gerçekleştirimlerinden farklı olarak yörünge değerleri saklanmamakta; değerler her iterasyonda çıkış bitlerine verilmektedir. Bu çıkışlar ise bir D/A dönüştürücüye verilerek osiloskop üzerinde yörüngelerin görüntülenmesi sağlanmıştır. Çıkışlar verilirken 1B haritalar için yörüngenin bir saat darbesi geciktirilmiş hali de çıkış bitlerine yönlendirilmiştir. Bu sayede çeker yapılarının da osiloskop üzerinde görüntülenmesi sağlanmıştır. SRSÜ olarak donanımda çok kolay gerçekleştirilebilmesi nedeniyle lineer geribeslemeli kaydırmalı kaydedici (LFSR) kullanılmıştır. LFSR'lerin uzunluğu 128 bit olarak seçilmiştir. Dolayısıyla, periyodik yörüngelerin her birinin periyodu 2^{128}-1'dir. Önceki kısımda verilen algoritmalar ile yapılan FPGA gerçekleştirimleri neticesinde elde edilen osiloskop görüntüleri Şekil 2.11'de verilmektedir. Elde edilen görüntüler önceki sonuçlarla uyumludur.



(a)



(b)

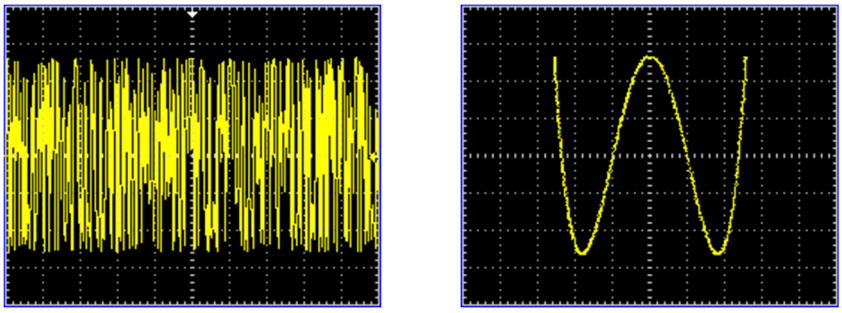


(c) (d)

(e) (f)

Şekil 2.11 Osiloskop görüntüleri: (a)-(b) Bernoulli haritası için gerçek periyodik yörünge ve çeker yapısı; (c)-(d) tent haritası için gerçek periyodik yörünge ve çeker yapısı; (e)-(f) baker haritasının yörüngeleri ve çeker yapısı

Topolojik konjuge ile gerçek yörüngeleri elde etmek içinse trigonometrik hesaplamaların dijital platform üzerinde yapılmasına ihtiyaç bulunmaktadır. Bu amaçla, h ve h_1 konjuge fonksiyonlarının gerektirdiği trigonometrik hesaplar FPGA üzerinde CORDIC (Coordinate Rotation Digital Computer) algoritması kullanılarak yapılmıştır. Hesaplamalar esnasında Q6.32 şeklinde sabit noktalı aritmetik kullanılmıştır. LSB metodundaki L=32 bitlik hafıza elemanının Q6.32'ye çevrimi ise hafıza elemanının bit değerlerinin sabit noktalı sayıların 32 bitlik kesirli kısmına atanmasıyla sağlanmıştır. 6 bitlik tam sayı kısmı ise sıfır bitleriyle doldurulmuştur. Gerçekleştirim olarak ise $f_2(x) = 64x^4 - 128x^3 + 80x^2 - 16x + 1$ haritasının gerçek periyodik yörüngesi S^2 ile konjuge kurularak hesaplanmıştır. Önceki kısımlarda gösterilmiş olduğu üzere S^2 haritasının LSB uzatma metodu ile gerçekleştirimi her iterasyonda iki tane rastgele bite ihtiyaç duymaktadır. Bu nedenle, gerçekleştirim esnasında farklı başlangıç şartlarına sahip iki adet LFSR kullanılmıştır. Elde edilen osiloskop görüntüleri Şekil 2.12'deki gibidir.

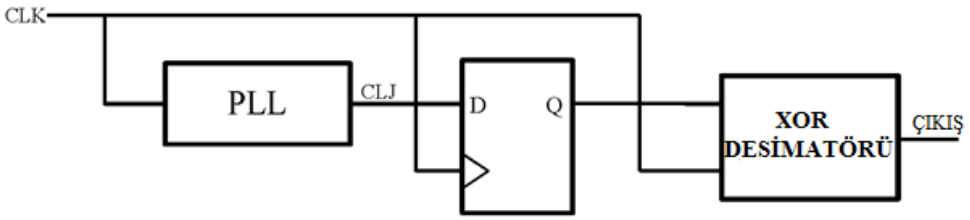


(a) (b)

Şekil 2.12 f_2 haritasının donanım gerçekleştiriminin osiloskop görüntüleri: (a) gerçek periyodik yörünge; (b) çeker yapısı

2.4.2. Gerçek Kaotik Yörünge Üretimi

Gerçek kaotik yörünge elde etmek için tek yapılması gereken önceki donanım gerçekleştirimlerindeki LFSR'leri bir GRSÜ ile değiştirmektir. Gerçek kaotik yörünge üretiminde kullanılacak olan gerçek rastgele sayı üreteci [134]'de sunulmuş bir üreteç olup yapısı Şekil 2.13'de gösterildiği gibidir. Üreteç PLL yapısındaki jitter kaymalarını entropi kaynağı olarak kullandığı için FPGA üzerinde tamamen dijital olarak gerçekleştirime oldukça uygundur.



Şekil 2.13 PLL tabanlı gerçek rastgele sayı üreteci [134]

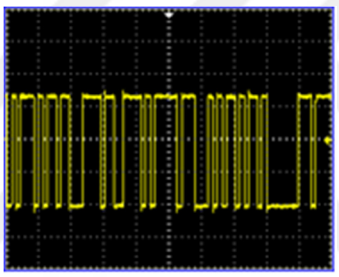
Şekil 2.13'den görülebileceği üzere üreteç bir adet PLL, 1 adet D flip-flop ve 1 adet XOR desimatöründen oluşmaktadır. Burada CLJ sinyali jittere sahip clock sinyali iken CLK sistem clock sinyalidir. XOR desimatörü flip-flopun çıkışını K_d kez örnekler ve bu örneklere XOR işlemi uygular. Bu yapıya desimatör adı verilmesinin nedeni çıkış veri hızını K_d oranında bölmesi nedeniyledir.

Verilen bu GRSÜ yapısının istenilen rastgele çıkışı sağlayabilmesi için bir takım şartların sağlanması gerekmektedir. K_m ve K_d sırasıyla PLL'nin çarpma ve bölme katsayıları

olmak üzere K_d tek olmalı ve $\text{obeb}(K_m, K_d) = 1$ şartı sağlanmalıdır. Ayrıca, T_{CLK} sistem clock sinyalinin periyot değeri olmak üzere

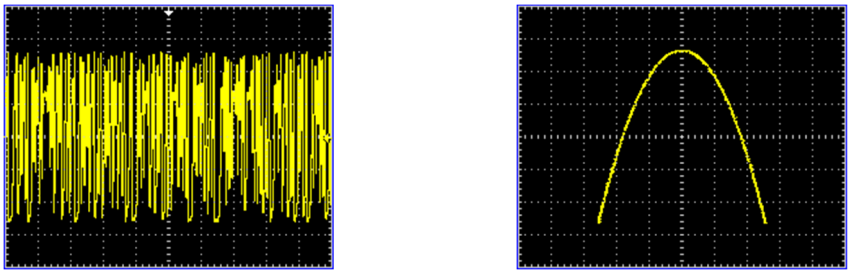
$$\max(\Delta T_{min}) = T_{CLK} \frac{\text{obeb}(2K_m, K_d)}{4K_m}$$

için $\sigma_{JIT} > \max(\Delta T_{min})$ şartı sağlanmalıdır. Çoğu Altera FPGA'leri için σ_{JIT} değeri tipik olarak 15 ps'den daha büyüktür [134]. Gerçekleştirim sırasında bu şartlar göz önünde bulundurularak $f_{CLK} = 1/T_{CLK} = 200$ MHz, $K_m = 206$, $K_d = 273$ olarak seçilmiş ve Cyclone III çipi içerisinde bulunan gömülü PLL yapıları kullanılmıştır. Gerçekleştirilen üretecin veri çıkış hızı 732.6 Kb/s olup üretilen rastgele bitler Şekil 2.14'de gösterilmektedir.



Şekil 2.14 Gerçek rastgele sayı üreteci gerçekleştiriminin çıkışının osiloskop görüntüsü

Bu rastgele sayı üreteci ile çadır haritasının gerçek kaotik yörüngeleri Algoritma 2 yardımıyla üretilmiş olup çıkışlar Şekil 2.11 (c) ve (d)'dekilerle (görünürde) aynıdır. Fakat, bu sefer üretilen kaotik yörüngedir ve periyodik değildir. Önceki donanım gerçekleştirimlerden farklı olarak lojistik haritanın da gerçek kaotik yörüngeleri bu GRSÜ yapısı yardımıyla elde edilmiştir. Gerçekleştirim detayları tamamen önceki gerçekleştirimlerle aynıdır. Öncelikle çadır haritası Algoritma 2.2 kullanılarak ve rastgele bitler GRSÜ'den alınarak VHDL ile gerçekleştirilmiş; sonrasında da konjuge fonksiyonu h'ın CORDIC algoritması ile gerçekleştirilmesiyle sabit noktalı (Q6.32) aritmetik ile lojistik haritaya ait gerçek kaotik yörünge elde edilmiştir. Lojistik haritaya ait gerçek kaotik yörünge ve çeker yapısının osiloskop görüntüsü Şekil 2.15'de verilmektedir. Bildiğimiz kadarıyla, bu lojistik haritanın gerçek kaotik yörüngesinin tamamen dijital olarak üretildiği ilk çalışmadır.



(a) (b)

Şekil 2.15 Lojistik haritanın donanım gerçekleştiriminin osiloskop görüntüleri: (a) gerçek kaotik yörünge; (b) çeker yapısı

Her bir FPGA gerçekleştirimine ait kaynak tüketimi ve maksimum çalışma frekansı değerleri de görüldüğü gibidir. Buradan görülebileceği üzere doğrudan algoritmalar kullanılarak yapılan üretimler son derece düşük kaynak tüketimine sahip olup yüksek hızlarda çalışabilmektedir. Fakat, CORDIC algoritması ve sabit noktalı aritmetiğin kullanılmasını gerektiren topolojik konjuge yardımıyla yapılan üretimlerde kaynak tüketimi bir anda artmakta ve hız düşmektedir.

Tablo 2.2 FPGA gerçekleştirim sonuçları

Model	Alan Kullanımı (Lojik Blok)	Maks. Çalışma Frekansı
Bernoulli haritası (LFSR)	199 (% 1)	404.37 MHz
Çadır haritası (LFSR)	219 (% 1)	206.1 MHz
Baker haritası (LFSR)	199 (% 1)	500 MHz
f_2 haritası (LFSR)	2525 (% 16)	43.47 MHz
GRSÜ	311 (% 2)	254.97 MHz
Lojistik harita (GRSÜ)	2634 (% 17)	42.01 MHz

3. BÖLÜM

BAKER HARİTASININ ÇOK BOYUTLU GENELLEŞTİRİMİ VE LSB UZATMA METODU İLE GERÇEKLEŞTİRİMİ

Kaos ile kriptografi arasında görünürde yakın bir ilişki olsa da pratik uygulamalardaki sıkıntılar nedeniyle kaos tabanlı kriptosistemler klasik kriptografiyle ilgilenen araştırmacılar tarafından rağbet görmemektedir. Bunun (dijital gerçekleştirim problemleri dışındaki) en temel nedeni, klasik kriptografinin tam sayılara ve basit bitsel işlemlere dayanmasına karşın kaos tabanlı kriptografinin reel sayılar kümesinde tanımlı sistemlerin sabit veya kayan noktalı aritmetikle gerçekleştirimlerine dayanmasıdır. Oysa ki, reel sayılar kümesinde tanımlı kaotik sistemlerin sabit veya kayan noktalı aritmetik ile gerçekleştirimlerinin klasik kriptografide hiçbir karşılığı yoktur. Dolayısıyla, klasik kriptografinin bugüne kadar geliştirilmiş olan metodolojisini kaos tabanlı kriptografiye uyarlamak oldukça zor olmaktadır. Bunun yanı sıra, kaos tabanlı kriptografi alanındaki araştırmacılar güvenliliğin incelenmesi hususunda klasik kriptografi ile uğraşan araştırmacılar kadar titiz davranmamaktadır. Klasik kriptosistemler en ince detayına kadar düşünülmüş tasarımlar olurken, kaos tabanlı kriptosistemlerde pek çok husus iyi düşünülmemekte veya göz ardı edilmektedir [4]. Mesela, kaos tabanlı kriptografide kaotik sistemin ürettiği sözde yörüngelerin periyot uzunluğunun ne olduğu belli değildir. Böyle bir belirsizlik ciddi kriptosistemler için kabul edilebilir bir durum değildir. Ayrıca, orijinal matematiksel model ile sonlu çözünürlük ve yuvarlama hataları altında yapılan gerçekleştirim arasındaki ilişki de net değildir. Gerçekleştirim detayları standarda bağlanmadığından ve sabit / kayan noktalı aritmetiğin farklı implementasyonlarının mümkün olmasından dolayı, şifreleme / deşifreleme işleminin farklı platformlarda aynı sonucu vereceğinin de bir garantisi yoktur. Bunlara ek olarak, klasik kriptografik sistemler doğrudan donanım üzerinde gerçekleştirilebilirken, kaos tabanlı kriptosistemleri donanım üzerinde gerçekleştirebilmek için öncelikle kayan veya sabit noktalı aritmetiğin donanımda bir gerçekleştiriminin yapılması şarttır.

Bahsedilen bu nedenler yüzünden kaotik sistemlerin dijital gerçekleştirimlerinde kullanılmak üzere konvansiyonel sabit veya kayan noktalı aritmetiğe bağlı olmayan ve tıpkı klasik kriptosistemler gibi basit bitsel işlemlerle gerçekleştirilebilecek yeni tasarım yaklaşımlarının geliştirilmesi gerekmektedir. Şimdiye kadar, literatürde bu koşulu sağlayabilen tek bir metot önerilmiştir. Önerilen bu metot, tam sayılar kümesinde tanımlı kaotik sistemlere (IDCS) dayanmaktadır [50, 51, 53–55]. IDCS'ler bir veya daha fazla rastgele tam sayı dizisini giriş olarak alan ve ardından bunları bitsel lojik işlemlerle karıştıran haritalardır. Bu tür tam sayılar kümesinde tanımlı haritaların Devaney tanımına göre kaotik oldukları ispatlanmıştır [50].

İkinci bölümde önerilmekte olan LSB uzatma metodu klasik kaotik sistemlerin bazılarının gerçek yörüngelerinin dijital platformlarda kolayca elde edilmesini sağlamaktadır. Yine aynı bölümde bahsedildiği üzere, LSB uzatma metodu dışındaki diğer gerçek yörünge üretim metotları pratik uygulamalarda kullanılmaya ve donanım gerçekleştirimlerine uygun değildirler. Ayrıca, LSB uzatma metodu da basit bitsel işlemlere dayalı bir metot olup sabit veya kayan noktalı aritmetik kullanılmaksızın gerçekleştirilebilmektedir. Bu bakımdan, LSB uzatma metodu da tıpkı IDCS metodu gibi kaos tabanlı kriptosistemlerdeki basit lojik işlemlerle gerçekleştirilebilme ve gerçek yörüngelerin üretilmesi sorunlarına çözüm getirebilecek durumdadır.

Fakat, LSB uzatma metodu ve IDCS kıyaslandığında birtakım farklılıklar olduğu görülmektedir. Dikkat edilirse IDCS metodu kaotik sistemlerin gerçek yörüngelerinin üretilmesi hakkındaki ikinci bölümde değinilmemişti. Bunun nedeni, IDCS metodunun literatürde mevcut olan reel sayılar kümesinde tanımlı klasik kaotik sistemlerin gerçekleştirimlerinden ziyade, tam sayılar kümesinde tanımlı yeni tür kaotik sistemlerin önerilmesine dayanmasıdır. Bu bakımdan, IDCS metodu konvansiyonel olmayan yeni kaotik sistemlerin gerçekleştirimine dayalıyken, LSB uzatma metodu klasik kaotik sistemlerin konvansiyonel olmayan yollarla üretimine dayanmaktadır. Bunun dışında, [51]'de tüm boyutlar için geçerli bir IDCS önerilmiştir. Ancak, LSB uzatma metodu İKKH'lara dayalı olup literatürde her boyut için tanımlı bir İKKH bulunmamaktadır.

Dolayısıyla, LSB uzatma metodunun kaos tabanlı kriptosistem uygulamalarında IDCS metoduna bir alternatif olabilmesi için her boyut için geçerli bir İKKH tanımlanmalıdır. Bu amaçla, bu bölümde yapılan çalışmada, normalde 2 boyutlu (2B) olan ve Eşitlik (2.2)

ile verilen baker haritası her boyut için genelleştirilerek, elde edilen yapı "çok boyutlu baker haritası" (ÇBBH) olarak adlandırılmıştır. ÇBBH'nın her boyut için kaotik olduğu Devaney'in kaos tanımına göre matematiksel olarak ispatlanmış ve ÇBBH'nin çeşitli özellikleri analiz edilmiştir. Yine matematiksel olarak ÇBBH'nın her boyut için lojik işlemlerden ibaret sadece iki tür temel yapı bloğu ile gerçekleştirilebileceği gösterilmiştir. ÇBBH'na ait gerçekleştirimler önerilen bloklar yardımı ile hem yazılım hem de FPGA donanımı üzerinde yapılmış; ÇBBH'na ait farklı boyutlardaki gerçek periyodik ve gerçek kaotik yörüngeler elde edilmiştir. Bu önerilen temel yapı blokları ile dijital olarak istenilen boyutta kaotik işaret üretmek çok kolay olmakta ve yukarıda bahsedilmiş olan sorunlara bir çözüm getirilmektedir.

3.1. Çok Boyutlu Baker Haritası

Bu kısımda ÇBBH tanıtılacak olup ÇBBH'nın kaotik olduğu Devaney tanımına göre ispatlanacaktır [52]. Literatürde pek çok kaos tanımı vardır ve bunların her biri bazı avantaj ve dezavantajlara sahiptir [135]. Bu çalışmada Devaney tanımının kullanılmasının nedeni, tanımın topolojik özelliklere dayalı olması ve bunun neticesinde tanım ile Bölüm 1.1'de bahsedilen sembolik dinamikler arasında bağ kurulabilmesidir. Bölüm 1.1'den hatırlanacağı üzere, sembolik dinamikleri incelemek, kaotik sistemin kendisini incelemekten daha kolay olmaktadır.

Literatürde daha önce baker haritasının sadece 3B için bir genelleştirmesi [136]'da yapılmış olup, burada yapılan her boyut için genelleştirme ve bunun matematiksel ispatının yapılması literatürde bir ilktir. Ayrıca, ÇBBH'nın çeker yapısı ve Lyapunov üstelleri gibi birtakım özellikleri de bu kısımda analiz edilmektedir.

3.1.1. Baker Haritasının Her Boyut İçin Genelleştirilmesi

ÇBBH'nın eşitlikleri verilmeden önce β_m fonksiyonu tanıtılmalıdır. Bu fonksiyon bir tam sayının bit değerlerini döndürmekte olup Tanım 3.1.1'deki gibi tanımlanmaktadır.

Tanım 3.1.1. $\beta_m(k,n)\colon \mathbb{N} \times \mathbb{N}^+ \to \{0,1\}$, k tam sayısının $m-1$ bitlik ikili gösteriminin (MSB biti 1 olmak üzere) n'inci bitini döndüren bir fonksiyon olsun. Buna göre, eğer $k = (b_1 b_2 \dots b_n \dots)_2$ ise $\beta_m(k,n) = b_n$ olacaktır. Burada m hep boyutu temsil ettiğinden, bu fonksiyon kısaca $\beta(k,n)$ ile ifade edilecektir.

Bu çalışmada ÇBBH $F: D \to D$ ile temsil edilecek olup, ÇBBH'nın tanım kümesi Eşitlik (3.1)'deki gibi ifade edilmektedir.

$$D = \{ (x_1, x_2, \dots, x_m) \mid x_i \in [0,1),\ i \in \{1,2, \dots, m\} \} \tag{3.1}$$

ÇBBH'nın kendisi ise Eşitlik (3.2)'deki gibi tanımlanmaktadır.

$$F(\mathbf{x}) = \begin{cases} \mathrm{T}_1(\mathbf{x}), & \mathbf{x} \in C_0 \\ \mathrm{T}_2(\mathbf{x}), & \mathbf{x} \in C_1 \\ \vdots & \vdots \\ T_{2^{m-1}-1}(\mathbf{x}), & \mathbf{x} \in C_{2^{m-1}-1} \end{cases} \tag{3.2}$$

Burada $\mathbf{x} = (x_1, x_2, \dots, x_m)$'dir ve $k \in \{0,1, \dots, 2^{m-1} - 1\}$ olmak üzere T_k,

$$\begin{aligned} T_k(\mathbf{x}) &= A\mathbf{x} - \boldsymbol{p}_k \\ &= \begin{bmatrix} 2 & 0 & \cdots & 0 & 0 \\ 0 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 2 & 0 \\ 0 & 0 & \cdots & 0 & 1/2^{m-1} \end{bmatrix}_{m \times m} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m-1} \\ x_m \end{bmatrix} - \begin{bmatrix} \beta(k,1) \\ \beta(k,2) \\ \vdots \\ \beta(k,m-1) \\ -k/2^{m-1} \end{bmatrix} \end{aligned}$$

şeklindedir. Ayrıca, burada A köşegen elemanları son eleman hariç 2 olan bir köşegen matris ve $m \geq 2$ ÇBBH'nın boyutudur. C_k kümeleri ise Eşitlik (2.3)'deki gibi tanımlanır.

$$C_k = \left\{ \mathbf{x} \in D \mid \frac{\beta(k,i)}{2} \leq x_i < \frac{\beta(k,i)+1}{2},\ i \in \{1, \dots, m-1\} \right\} \tag{3.3}$$

Bu ÇBBH genelleştirimi $m = 2$ için Eşitlik (2.2)'deki baker haritasına karşılık gelmektedir. $m = 3$ durumu ise tam olarak [136]'de verilen 3B baker haritasına karşılık gelmemektedir. Bu durumda, ÇBBH ile [136]'daki harita arasında tanım kümesinin karıştırılma sırasına dayalı bir farklılık bulunmaktadır. Bildiğimiz kadarıyla, $m > 3$ için ÇBBH'nın literatürde bir karşılığı bulunmamaktadır. Ayrıca, normalde tanımsız olan $m = 1$ durumunda ÇBBH'nın Eşitlik (2.1) ile verilen Bernoulli haritasını verdiği düşünülürse ÇBBH tüm boyutlar için tanımlanmış olur.

3.1.2. Ters Harita

ÇBBH'nın tersini vermeden önce ÇBBH'nın terslendirilebilir olduğunun ispatlanması gerekmektedir. Bunun için F haritasının bijeksiyon olduğunun gösterilmesi gerekir.

Lemma 3.1.2. $F(C_k) = D_k = \left\{ \mathbf{x} \in D \mid \frac{k}{2^{m-1}} \leq x_m < \frac{k+1}{2^{m-1}} \right\}$.

İspat: $\mathbf{y} = F(C_k)$ olsun. Bu durumda, $\mathbf{y} = T_k(C_k)$'dir. Aynı zamanda, eğer $\mathbf{x} \in C_k$ ise, her $i \in \{1, \dots, m-1\}$ için $\frac{\beta(k,i)}{2} \leq x_i < \frac{\beta(k,i)+1}{2}$'dir. Her $i \neq m$ için $y_i = 2x_i - \beta(k,i)$ olduğundan aynı i değerleri için $0 \leq y_i < 1$'dir. Benzer şekilde, $y_m = \frac{x_m}{2^{m-1}} + \frac{k}{2^{m-1}}$ olduğundan $\frac{k}{2^{m-1}} \leq x_m < \frac{k+1}{2^{m-1}}$ olmalıdır. ■

Bu çalışma boyunca aksi belirtilmediği müddetçe $k \in \{0,1, \dots, 2^{m-1} - 1\}$ kabul edilecektir. Ayrıca, $F(\mathbf{x})$ bir $\mathbf{x}$ noktasının hedef kümesindeki değerini temsil ederken, $F(C_k)$ ise C_k kümesinin F altındaki dönüşümünü ifade etmektedir.

Lemma 3.1.3. $\bigcup_{k=0}^{2^{m-1}-1} C_k = \bigcup_{k=0}^{2^{m-1}-1} F(C_k) = D$.

İspat: $\bigcup_{k=0}^{2^{m-1}-1} F(C_k) = D$ eşitliği doğrudan Lemma 3.1.2'den gelmektedir. $\bigcup_{k=0}^{2^{m-1}-1} C_k = D$ ifadesinin doğruluğunu görebilmek için $\beta(k,i)$'nin $k \in \{0,1, \dots, 2^{m-1} - 1\}$ için hem 0 hem de 1'i en az bir kez döndürdüğünü fark etmek gerekir. Bu durumda, $\bigcup_{k=0}^{2^{m-1}-1} C_k = \{\mathbf{x} \in D \mid 0 \leq x_i < 1,\ i \in \{1, \dots, m-1\}\} = D$ olmalıdır. ■

Önerme 3.1.4. F bijeksiyondur.

İspat: Lemma 3.1.2 ve Lemma 3.1.3' ün sonucu olarak

$$F(D) = F\left(\bigcup_{k=0}^{2^{m-1}-1} C_k \right) = \bigcup_{k=0}^{2^{m-1}-1} F(C_k) = D$$

olmalıdır. Yani, F örtendir. F'nin birebir olduğunu gösterebilmek için $\mathbf{x}, \mathbf{y} \in D$ için $F(\mathbf{x}) = F(\mathbf{y})$ olduğunda $\mathbf{x} = \mathbf{y}$ olması gerektiği gösterilmelidir. $F(C_k) = T_k(C_k)$ kümelerinin ayrık olmasından dolayı $F(\mathbf{x}) = F(\mathbf{y})$ ancak $\mathbf{x}, \mathbf{y} \in C_k$ olduğunda mümkündür. Bu durumda, $T_k(\mathbf{x}) = T_k(\mathbf{y})$ olmalıdır. A'nın sütun vektörlerinin lineer bağımsız olması dolayısıyla $T_k(\mathbf{x}) = T_k(\mathbf{y})$ gerektiği gibi $\mathbf{x} = \mathbf{y}$ anlamına gelmektedir. ■

Önerme 3.1.4'ün bir sonucu olarak ters harita $F^{-1}: D \to D$ mevcut olmak zorundadır ve Eşitlik (3.4)'deki gibi tanımlanır.

$$F^{-1}(\mathbf{x}) = \{T_k^{-1}(\mathbf{x}), \quad \mathbf{x} \in D_k \tag{3.4}$$

Burada,

$$T_k^{-1}(\mathbf{x}) = A^{-1}\mathbf{x} - A^{-1}\mathbf{p}_k$$
$$= \begin{bmatrix} 1/2 & 0 & \cdots & 0 & 0 \\ 0 & 1/2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1/2 & 0 \\ 0 & 0 & \cdots & 0 & 2^{m-1} \end{bmatrix}_{m\times m} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m-1} \\ x_m \end{bmatrix} + \begin{bmatrix} \beta(k,1)/2 \\ \beta(k,2)/2 \\ \vdots \\ \beta(k,m-1)/2 \\ -k \end{bmatrix}$$

şeklindedir.

3.1.3. Çok Boyutlu Baker Haritasının Kaotik Özelliği

ÇBBH'nın Devaney'in kaos tanımına göre kaotik olduğu bu kısımda ispatlanacaktır. Devaney'in kaos tanımı ve burada kullanılan pek çok tanım Bölüm 1.1'de bulunmaktadır. Dolayısıyla, kullanılan tanımlar burada tekrar verilmeyecektir. Ayrıca, ÇBBH'nın kaotik olduğu ispatlanırken sonuca adım adım gidilecektir. Bu nedenle, ilk başta verilen önerme ve teoremler ilk başta pek anlam ifade etmeyebilir. Fakat, sonuca bu elde ettiğimiz önermeler ile gidilecektir. İspatlar içerisinde sıklıkla $A, B \subseteq D$ olmak üzere $F(A \cap B) = F(A) \cap F(B)$, $F^{-1}\big(F(A)\big) = A$ ve $F\big(F^{-1}(B)\big) = B$ eşitliklerine başvurulmaktadır. Bu eşitlikler genel olarak doğru olmasa da F'nin birebir ve örten olması dolayısıyla F için doğrudurlar.

Lemma 3.1.5. $S_J \in \{0,1,\dots,2^{m-1}-1\}$ olmak üzere

$$\bigcap_{j=0}^{N} F^{-j}\left(C_{S_j}\right) = \left\{ \mathbf{x} \in D \mid \sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}} \le x_i < \sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}} + \frac{1}{2^{N+1}},\ i \in \{1,\dots,m-1\} \right\}.$$

İspata geçmeden önce bazı açıklamalar yapılmalıdır. Öncelikle, F^{-N} ifadesi $F^{-1} \circ F^{-1} \dots \circ F^{-1}$'ye karşılık gelmektedir. Benzer şekilde F^N, $F \circ F \dots \circ F$'dir. Tabi $F^0\left(C_{S_j}\right)$, C_{S_j}'dir.

İspat: İspat tümevarıma (indüksiyon) dayalıdır. $N = 0$ durumu Eşitlik (3.3)'deki tanıma karşılık geldiği için doğrudur. Şimdi hipotezin $N > 0$ için doğru olduğunu varsayalım. Bu durumda,

$$F^{-1}\left(\bigcap_{j=0}^{N} F^{-j}\left(C_{S_j}\right)\right) = F^{-1}\left(\bigcup_{l=0}^{2^m-1}\left(\left(\bigcap_{j=0}^{N} F^{-j}\left(C_{S_j}\right)\right) \cap D_l\right)\right)$$

$$= \bigcup_{l=0}^{2^m-1} F^{-1}\left(\left(\bigcap_{j=0}^{N} F^{-j}\left(C_{S_j}\right)\right) \cap D_l\right)$$

olacaktır. Buna göre, eşitliğin sağ tarafı

$$= \bigcup_{l=0}^{2^m-1} T_l^{-1}\left(\left\{\mathbf{x} \in D \mid \sum_{j=0}^{N} \frac{\beta(S_j, i)}{2^{j+1}} \le x_i < \sum_{j=0}^{N} \frac{\beta(S_j, i)}{2^{j+1}} + \frac{1}{2^{N+1}}, i \in \{1, \dots, m-1\},\right.\right.$$

$$\left.\left.\frac{l}{2^{m-1}} \le x_m < \frac{l+1}{2^{m-1}}\right\}\right)$$

$$= \bigcup_{l=0}^{2^m-1} \left(\left\{\mathbf{x} \in D \mid \sum_{j=0}^{N} \frac{\beta(S_j, i)}{2^{j+2}} + \frac{\beta(l, i)}{2} \le x_i < \sum_{j=0}^{N} \frac{\beta(S_j, i)}{2^{j+2}} + \frac{1}{2^{N+2}} + \frac{\beta(l, i)}{2},\right.\right.$$

$$\left.\left. i \in \{1, \dots, m-1\}\right\}\right)$$

şeklindedir. Dikkat edilirse,

$$\sum_{j=0}^{N} \frac{1}{2^{j+2}} + \frac{1}{2^{N+2}} = \frac{1}{2}$$

olduğundan

$$\sum_{j=0}^{N} \frac{\beta(S_j, i)}{2^{j+2}} + \frac{\beta(l, i)}{2} \le x_i < \sum_{j=0}^{N} \frac{\beta(S_j, i)}{2^{j+2}} + \frac{1}{2^{N+2}} + \frac{\beta(l, i)}{2}$$

ifadesi aşağıdaki ifadeye eşdeğerdir:

$$\frac{\beta(l,i)}{2} \leq x_i < \frac{\beta(l,i)+1}{2}$$

Buna göre, $C_{S_{-1}}$ Eşitlik (3.3) kullanılarak tanımlanırsa $C_{S_{-1}} \cap F^{-1}\left(\bigcap_{j=0}^{N} F^{-j}\left(C_{S_j}\right)\right)$ ifadesinin ancak ve ancak $l = S_{-1}$ olursa boş olmadığı görülür. Böylece,

$$C_{S_{-1}} \cap F^{-1}\left(\bigcap_{j=0}^{N} F^{-j}\left(C_{S_j}\right)\right) = C_{S_{-1}} \cap \left(\bigcap_{j=0}^{N} F^{-j-1}\left(C_{S_j}\right)\right) = \bigcap_{j=-1}^{N} F^{-j-1}\left(C_{S_j}\right)$$

olur ve bu da aşağıdaki ifadeler eşittir:

$$= \left\{ \mathbf{x} \in D \mid \sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+2}} + \frac{\beta(S_{-1},i)}{2} \leq x_i < \sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+2}} + \frac{1}{2^{N+2}} + \frac{\beta(S_{-1},i)}{2}, \right.$$
$$\left. i \in \{1,\dots,m-1\} \right\}$$

$$= \left\{ \mathbf{x} \in D \mid \sum_{j=-1}^{N} \frac{\beta(S_j,i)}{2^{j+2}} \leq x_i < \sum_{j=-1}^{N} \frac{\beta(S_j,i)}{2^{j+2}} + \frac{1}{2^{N+2}},\ i \in \{1,\dots,m-1\} \right\}$$

Elde edilen bu eşitlikte j değişkeni değiştirilirse aşağıdaki eşitliği elde etmiş oluruz:

$$\bigcap_{j=0}^{N+1} F^{-j}\left(C_{S_{j-1}}\right) = \left\{ \mathbf{x} \in D \mid \sum_{j=0}^{N+1} \frac{\beta(S_{j-1},i)}{2^{j+1}} \leq x_i < \sum_{j=0}^{N+1} \frac{\beta(S_{j-1},i)}{2^{j+1}} + \frac{1}{2^{N+2}}, \right.$$
$$\left. i \in \{1,\dots,m-1\} \right\}$$

Burada $C_{S_{j-1}}$ ifadesini C_{S_j} şeklinde yeniden etiketleyebiliriz ve böylelikle tümevarım adımı doğrulanmış olur. ■

Lemma 3.1.5'in bir sonucu da C_k kümeleri arasındaki herhangi bir geçişin mümkün olduğunu göstermesidir. Yani herhangi $S_0, S_1 \in \{0,\dots,2^{m-1}-1\}$ için $C_{S_0} \cap F^{-1}(C_{S_1})$ ifadesi boş değildir.

Lemma 3.1.6. $S_{-j} \in \{0,1,\dots,2^{m-1}-1\}$ olmak üzere,

$$\bigcap_{J=1}^{N} F^{j}\left(C_{S_{-j}}\right) = \left\{\mathbf{x} \in D \;\middle|\; \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} \le x_m < \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} + \frac{1}{2^{N(m-1)}}\right\}.$$

İspat: İspat tümevarıma (indüksiyon) dayalıdır. $N = 1$ durumu Lemma 3.1.2 nedeniyle doğrudur. Şimdi hipotezin $N > 1$ için doğru olduğunu varsayalım. Bu durumda, $F\left(\bigcap_{j=1}^{N} F^{j}\left(C_{S_{-j}}\right)\right)$ ifadesi aşağıdakilere eşit olacaktır:

$$= \left\{\mathbf{x} \in D \mid \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} \le x_m < \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} + \frac{1}{2^{N(m-1)}}\right\}$$

$$= F\left(\bigcup_{l=0}^{2^{m-1}-1} \left\{\mathbf{x} \in D \mid \frac{\beta(l,i)}{2} \le x_i < \frac{\beta(l,i)+1}{2},\ i \in \{1,\dots,m-1\}\right.\right.$$
$$\left.\left. \text{ve } \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} \le x_m < \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} + \frac{1}{2^{N(m-1)}}\right\}\right)$$

$$= \bigcup_{l=0}^{2^{m-1}-1} T_l\left(\left\{\mathbf{x} \in D \mid \frac{\beta(l,i)}{2} \le x_i < \frac{\beta(l,i)+1}{2},\ i \in \{1,\dots,m-1\}\right.\right.$$
$$\left.\left. \text{ve } \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} \le x_m < \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} + \frac{1}{2^{N(m-1)}}\right\}\right)$$

$$= \bigcup_{l=0}^{2^{m-1}-1} \left\{\mathbf{x} \in D \mid \sum_{j=1}^{N} \frac{S_{-j}}{2^{(j+1)(m-1)}} + \frac{l}{2^{m-1}} \le x_m \right.$$
$$\left. < \sum_{j=1}^{N} \frac{S_{-j}}{2^{(j+1)(m-1)}} + \frac{1}{2^{(j+1)(m-1)}} + \frac{l}{2^{m-1}}\right\}$$

Dikkat edilirse, eğer her $j \in \{1,\dots,N\}$ için $S_{-j} = 2^{m-1} - 1$ seçilirse,

$$\sum_{j=1}^{N} \frac{S_{-j}}{2^{(j+1)(m-1)}} + \frac{1}{2^{(j+1)(m-1)}} = \frac{1}{2^{m-1}}$$

olacaktır. Bu yüzden,

$$\sum_{j=1}^{N} \frac{S_{-j}}{2^{(j+1)(m-1)}} + \frac{l}{2^{m-1}} \leq x_m < \sum_{j=1}^{N} \frac{S_{-j}}{2^{(j+1)(m-1)}} + \frac{1}{2^{(N+1)(m-1)}} + \frac{l}{2^{m-1}}$$

eşitsizliği

$$\frac{l}{2^{m-1}} \leq x_m < \frac{l+1}{2^{m-1}}$$

eşitsizliği anlamına gelir. Dolayısıyla, $F\left(C_{S_0}\right) \cap F\left(\bigcap_{j=1}^{N} F^j\left(C_{S_{-j}}\right)\right)$ ifadesi ancak ve ancak $l = S_0$ olursa boş değildir. Bu yüzden,

$$F\left(C_{S_0}\right) \cap F\left(\bigcap_{j=1}^{N} F^j\left(C_{S_{-j}}\right)\right) = \bigcap_{j=0}^{N} F^{j+1}\left(C_{S_{-j}}\right)$$

ifadesi aşağıdakine eşittir:

$$= \left\{ \mathbf{x} \in D \mid \sum_{j=0}^{N} \frac{S_{-j}}{2^{(j+1)(m-1)}} \leq x_m < \sum_{j=0}^{N} \frac{S_{-j}}{2^{(j+1)(m-1)}} + \frac{1}{2^{(N+1)(m-1)}} \right\}$$

j değişkenini değiştirdiğimizde ise,

$$\bigcap_{J=1}^{N+1} F^j\left(C_{S_{1-j}}\right) = \left\{ \mathbf{x} \in D \mid \sum_{j=1}^{N+1} \frac{S_{1-j}}{2^{j(m-1)}} \leq x_m < \sum_{j=1}^{N+1} \frac{S_{1-j}}{2^{j(m-1)}} + \frac{1}{2^{(N+1)(m-1)}} \right\}$$

olduğunu görürüz. C_{S_j} ifadesini $C_{S_{j-1}}$ şeklinde etiketlediğimizde ise tümevarım adımını doğrulamış oluruz.

Lemma 3.1.5 ve 3.1.6, Önerme 3.1.7'yi ispatlamaktadır. Burada H_N, D'deki m-boyutlu bir hiper-dikdörtgeni temsil etmektedir.

Önerme 3.1.7. $H_N = \bigcap_{j=-N}^{N} F^{-j}\left(C_{S_j}\right)$ olsun. Bu durumda,

$$H_N = \Bigg\{ \mathbf{x} \in D \mid \sum_{j=0}^{N} \frac{\beta(S_j, i)}{2^{j+1}} \le x_i < \sum_{j=0}^{N} \frac{\beta(S_j, i)}{2^{j+1}} + \frac{1}{2^{N+1}},\ i \in \{1, \dots, m-1\} \text{ ve } \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} \le x_m < \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} + \frac{1}{2^{N(m-1)}} \Bigg\}.$$

Lemma 3.1.8. $U = \bigcap_{j=-N}^{N} F^{-j}\left(C_{S_j}\right)$ ve $V = \bigcap_{j=-M}^{M} F^{-j}\left(C_{S'_j}\right)$ olmak üzere, $j \in \{-N, \dots, N\}$ için eğer $M > N$ ve $S_j = S'_j$ ise, $U \supset V$'dir.

İspat: Önerme 3.1.7'yi kullandığımızda U ve V aşağıdaki gibi bulunur:

$$U = \Bigg\{ \mathbf{x} \in D \mid \sum_{j=0}^{N} \frac{\beta(S_j, i)}{2^{j+1}} \le x_i < \sum_{j=0}^{N} \frac{\beta(S_j, i)}{2^{j+1}} + \frac{1}{2^{N+1}},\ i \in \{1, \dots, m-1\} \text{ ve } \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} \le x_m < \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} + \frac{1}{2^{N(m-1)}} \Bigg\}$$

$$V = \Bigg\{ \mathbf{x} \in D \mid \sum_{j=0}^{N} \frac{\beta(S_j, i)}{2^{j+1}} + \sum_{j=N+1}^{M} \frac{\beta(S'_j, i)}{2^{j+1}} \le x_i < \sum_{j=0}^{N} \frac{\beta(S_j, i)}{2^{j+1}} + \sum_{j=N+1}^{M} \frac{\beta(S'_j, i)}{2^{j+1}} + \frac{1}{2^{M+1}},\ i \in \{1, \dots, m-1\} \text{ ve } \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} + \sum_{j=1}^{N} \frac{S'_{-j}}{2^{j(m-1)}} \le x_m < \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} + \sum_{j=N+1}^{M} \frac{S'_{-j}}{2^{j(m-1)}} + \frac{1}{2^{M(m-1)}} \Bigg\}$$

Dikkat edilirse, $\sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}} \le \sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}} + \sum_{j=N+1}^{M} \frac{\beta(S'_j,i)}{2^{j+1}}$ ve $\sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} \le \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} + \sum_{j=1}^{N} \frac{S'_{-j}}{2^{j(m-1)}}$'dir. Aynı zamanda, $\sum_{j=N+1}^{M} \frac{1}{2^{j+1}} + \frac{1}{2^{M+1}} = \frac{1}{2^{N+1}}$ ve $\sum_{j=N+1}^{M} \frac{2^{m-1}-1}{2^{j(m-1)}} + \frac{1}{2^{M(m-1)}} = \frac{1}{2^{N(m-1)}}$ olduğundan

$$\sum_{j=0}^{N}\frac{\beta(S_j,i)}{2^{j+1}}+\sum_{j=N+1}^{M}\frac{\beta(S'_j,i)}{2^{j+1}}+\frac{1}{2^{M+1}}\leq\sum_{j=0}^{N}\frac{\beta(S_j,i)}{2^{j+1}}+\frac{1}{2^{N+1}}$$

ve

$$\sum_{j=1}^{N}\frac{S_{-j}}{2^{j(m-1)}}+\sum_{j=N+1}^{M}\frac{S'_{-j}}{2^{j(m-1)}}+\frac{1}{2^{M(m-1)}}\leq\sum_{j=1}^{N}\frac{S_{-j}}{2^{j(m-1)}}+\frac{1}{2^{N(m-1)}}$$

eşitsizlikleri elde edilir. Bu ise $U\supseteq V$ anlamına gelir. Fakat, μ Lebesgue ölçümü olmak üzere, $\mu(U)>\mu(V)$ olduğundan bu ilişki $U\supset V$ şeklinde olmalıdır. ■

Lemma 3.1.9. A bir küme ve $\|\cdot\|$ Öklidyen norm olmak üzere $\operatorname{diam}(A)=\sup\{\,\|\mathbf{x}-\mathbf{y}\|\mid\mathbf{x},\mathbf{y}\in A\,\}$ olsun. Bu durumda,

$$\lim_{N\to\infty}\operatorname{diam}\left(H_N=\bigcap_{j=-N}^{N}F^{-j}\left(C_{S_j}\right)\right)=0$$

olur.

İspat: Sonuç doğrudan

$$\operatorname{diam}(H_N)=\sqrt{\sum_{i=1}^{m-1}\left(\frac{1}{2^{N+1}}\right)^2+\left(\frac{1}{2^{N(m-1)}}\right)^2}$$

eşitliğinden gelmektedir. ■

Lemma 3.1.8 ve 3.1.9 birlikte kullanıldığında aşağıdaki sonuç elde edilmektedir.

Önerme 3.1.10. $H_\infty=\bigcap_{j\in\mathbb{Z}}F^{-j}\left(C_{S_j}\right)$ ya D'de bir noktaya ya da boş kümeye karşılık gelmektedir.

Lemma 3.1.11. $S_j\in\{0,\dots,2^{m-1}-1\}$ ve $i\in\{1,\dots,m-1\}$ olmak üzere,

$$I_N=\bigcap_{j=0}^{N}\left[\sum_{j=0}^{N}\frac{\beta(S_j,i)}{2^{j+1}},\sum_{j=0}^{N}\frac{\beta(S_j,i)}{2^{j+1}}+\frac{1}{2^{N+1}}\right)$$

olsun. Her $j > N$ için $\beta(S_j, i) = 1$ şartını sağlayan bir $N \in \mathbb{N}$ değeri bulunamıyorsa, $\lim_{N\to\infty} I_N = I_\infty$ ifadesi $x_i = \sum_{j=0}^{\infty} \frac{\beta(S_j,i)}{2^{j+1}}$ şeklinde tek bir noktaya karşılık gelir.

İspat: Öncelikle $\lim_{N\to\infty} \sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}}$ limitinin var olduğu gösterilmelidir. $\sum_{j=0}^{N} \frac{1}{2^{j+1}}$ dizisi tarafından domine edildiğinden $\lim_{N\to\infty} \sum_{j=0}^{N} \frac{1}{2^{j+1}} = 1$ olduğundan, $\lim_{N\to\infty} \sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}}$ yakınsaktır ve limit mevcuttur.

Sonrasında hem $\left\{\lim_{N\to\infty} \sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}}\right\} \subseteq I_\infty$ hem de $\left\{\lim_{N\to\infty} \sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}}\right\} \supseteq I_\infty$ olduğunu göstermek ispat için yeterli olacaktır. Birinci durumda her $N \in \mathbb{N}$ için $\left\{\lim_{N\to\infty} \sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}}\right\} \in I_N$ olduğu gösterilmelidir. $\sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}}$'nin azalmayan bir dizi olmasından dolayı, her N için

$$\sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}} \leq \lim_{N\to\infty} \sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}}$$

olduğu aşikardır. Bu nedenle, her N için

$$\lim_{N\to\infty} \sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}} < \sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}} + \frac{1}{2^{N+1}}$$

olduğunu göstermek yeterlidir. Bu ise,

$$\lim_{M\to\infty} \sum_{j=N+1}^{M} \frac{\beta(S_j,i)}{2^{j+1}} < \frac{1}{2^{N+1}}$$

olduğunu göstermeye eşdeğerdir. Eğer her $j > N$ için $\beta(S_j, i) = 1$ olursa

$$\lim_{M\to\infty} \sum_{j=N+1}^{M} \frac{1}{2^{j+1}} = \frac{1}{2^{N+1}}$$

olur. Dolayısıyla, yukarıdaki eşitsizlik her $N \in \mathbb{N}$ için $\beta(S_j, i) = 0$ şartını sağlayan bir $j > N$ bulunabiliyorsa sağlanabilmektedir. Bu ise her $j > N$ için $\beta(S_j, i) = 1$ şartını

sağlayan bir $N \in \mathbb{N}$ değeri bulunamıyorsa demekle aynıdır ve hipotez tarafından bu şart sağlanmaktadır. Böylelikle, ilk durum ispatlanmaktadır.

İkinci durumda, $N \to \infty$ için $\mathrm{diam}(I_N) \to 0$ ve $I_{N+1} \subset I_N$ olması dolayısıyla I_∞ ya boştur ya da bir noktadır. I_∞ boşsa ispat tamamlanmış olur. Bu yüzden, $x \in I_\infty$ şeklinde bir noktanın olduğu varsayılacaktır. Fakat I_∞'un tanımına göre x, $\lim_{N\to\infty} \sum_{j=0}^{N} \frac{\beta(S_j,i)}{2^{j+1}}$'dan başka bir değer alamaz ve bu ikinci durumu ispatlar. ■

Lemma 3.1.12. $S_{-j} \in \{0, \dots, 2^{m-1} - 1\}$ olmak üzere,

$$I'_N = \bigcap_{j=1}^{N} \left[\sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}}, \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} + \frac{1}{2^{N(m-1)}} \right)$$

olsun. Her $j > N$ için $S_{-j} = 2^{m-1} - 1$ şartını sağlayan bir $N \in \mathbb{N}^+$ bulunamıyorsa, $\lim_{N\to\infty} I'_N = I'_\infty$ ifadesi $x_m = \sum_{j=1}^{\infty} \frac{S_{-j}}{2^{j(m-1)}}$ şeklinde tek bir noktaya karşılık gelmektedir.

İspat: $\max(S_{-j}) = 2^{m-1} - 1$ ve $\lim_{N\to\infty} \sum_{j=1}^{N} \frac{2^{m-1}-1}{2^{j(m-1)}} = 1$ olduğundan dolayı $\lim_{n\to\infty} \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}}$ yakınsar ve limit vardır. Dolayısıyla, $\left\{ \lim_{N\to\infty} \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} \right\} \subseteq I'_\infty$ ve $\left\{ \lim_{N\to\infty} \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} \right\} \supseteq I'_\infty$ olduğu gösterilmelidir. Birinci durumda, her $N \in \mathbb{N}$ için $\left\{ \lim_{N\to\infty} \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} \right\} \in I'_N$ olduğu gösterilmelidir. $\sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}}$ azalmayan bir dizi olduğundan, her N için

$$\sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} \leq \lim_{N\to\infty} \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}}$$

olduğu aşikardır. Bu yüzden, her $N \in \mathbb{N}$ için

$$\lim_{N\to\infty} \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} < \sum_{j=1}^{N} \frac{S_{-j}}{2^{j(m-1)}} + \frac{1}{2^{N(m-1)}}$$

olduğu gösterilmelidir. Bu ise,

$$\lim_{M\to\infty}\sum_{j=N+1}^{M}\frac{S_{-j}}{2^{j(m-1)}}<\frac{1}{2^{N(m-1)}}$$

olduğunu göstermeye eşdeğerdir. Eğer, her $j>N$ için $S_{-j}=2^{m-1}-1$ seçilirse

$$\lim_{M\to\infty}\sum_{j=N+1}^{M}\frac{2^{m-1}-1}{2^{j(m-1)}}=\frac{1}{2^{N(m-1)}}$$

olacaktır. Bu nedenle, yukarıdaki eşitsizlik her $N\in\mathbb{N}$ için, $S_{-j}\neq 2^{m-1}-1$ şartını sağlayan bir $j>N$ bulunabiliyorsa sağlanır. Bu şart hipotez tarafından saplandığından birinci durum doğrudur.

İkinci durum için $\left\{\lim_{N\to\infty}\sum_{j=1}^{N}\frac{S_{-j}}{2^{j(m-1)}}\right\}\supseteq I'_{\infty}$ olduğu gösterilmelidir. $\operatorname{diam}(I'_N)=\frac{1}{2^{N(m-1)}}$ olduğundan $\operatorname{diam}(I'_{\infty})=0$'dır. Bu I'_{∞}'un ya boş küme ya da bir nokta olduğunu gösterir. Eğer boş kümeyse ispat tamamlanmış olur. Bu yüzden, $x\in I'_{\infty}$ şeklinde bir nokta olduğunu varsayalım. I'_N tanımına göre x noktası $\lim_{N\to\infty}\sum_{j=1}^{N}\frac{S_{-j}}{2^{j(m-1)}}$'den başka bir nokta olamaz. Bu ise ikinci durumu doğrular. ■

Sonraki önerme Lemma 3.1.11 ve 3.1.12'nin doğrudan bir sonucudur.

Önerme 3.1.13. Her $i\in\{1,\dots,m-1\}$ ve her j > N için $\beta(S_j,i)=1$ şartını sağlayan bir $N\in\mathbb{N}$ değeri ve her $j>N$ için $S_{-j}=2^{m-1}-1$ şartını sağlayan bir $N\in\mathbb{N}^{+}$ değer bulunamıyorsa, $H_{\infty}=\bigcap_{j\in\mathbb{Z}}F^{-j}\left(C_{S_j}\right)$ ifadesi

$$\{\mathbf{x}\}=\left\{x_i=\sum_{j=0}^{\infty}\frac{\beta(S_j,i)}{2^{j+1}},\ i\in\{1,\dots,m-1\}\text{ ve }x_m=\sum_{j=1}^{\infty}\frac{S_{-j}}{2^{j(m-1)}}\right\}$$

ile ifade edilen tek bir noktaya karşılık gelir.

Şimdi ise $\bigcap_{j\in\mathbb{Z}}F^{-j}\left(C_{S_j}\right)$ ifadesinin belli şartlar altında D'de eşsiz bir noktaya karşılık geldiğini göstereceğiz. Bunu ispatlamak için öncelikle birim aralıktaki sayıların B tabanındaki açılımları hakkındaki aşağıdaki lemmaya ihtiyaç vardır.

Lemma 3.1.14. $x = \sum_{k=1}^{\infty} x_k /B_k$ ve $y = \sum_{k=1}^{\infty} y_k /B_k$, $x, y \in [0,1)$ şeklindeki iki sayının $B \geq 2$ tam sayı olmak üzere B tabanındaki açılımları olsun. Bariz durum olan her k için $x_k = y_k$ durumu dışında, x ve y ancak ve ancak aşağıdaki şartları sağlayan bir $\mathrm{N} \in \mathbb{N}^+$ bulunabiliyorsa eşit olur:

i. $k < N$ için $x_k = y_k$
ii. $k > N$ için $x_k = 0$ ve $y_k = B - 1$
iii. $x_N = y_N + 1$

Lemma 3.1.14 bilinen bir matematiksel ilişki olsa da bütünlük olması açısından ispatı da burada aşağıdaki gibi verilmektedir.

İspat: Farklı B açılımları için $x = y$ olduğunu varsayalım. Genelliği kaybetmeden, $N \in \mathbb{N}^+$ olmak üzere bir $k < N$ ve $x_N > y_N$ için $x_k = y_k$ olduğunu varsayalım. Bu durumda,

$$\begin{aligned} 0 &= x - y \\ &= \frac{x_N - y_N}{B^N} + \sum_{k=N+1}^{\infty} \frac{(x_k - y_k)}{B^k} \\ &\geq \frac{1}{B^N} + \sum_{k=N+1}^{\infty} \frac{(x_k - y_k)}{B^k} \\ &\geq \frac{1}{B^N} - (B-1) \sum_{k=N+1}^{\infty} \frac{1}{B^k} \\ &= \frac{1}{B^N} - \frac{1}{B^N} = 0 \end{aligned}$$

olur. Bu ise yukarıdaki eşitsizliklerin eşitlik olması gerektiği anlamına gelir. İlk eşitsizlik ancak ve ancak $x_N = y_N + 1$ olursa eşitlik olur. İkinci eşitsizlik ise ancak ve ancak $k > N$ için $x_k = 0$ ve $y_k = B - 1$ olursa eşitlik olur. Bu ise hipotezi her iki yön için ispatlar. ■

Önerme 3.1.15. $H_\infty = \bigcap_{j \in Z} F^{-j}\left(C_{S_j}\right)$ ifadesi Önerme 3.1.13'de verilen kısıtlamalar altında D'de eşsiz bir noktaya karşılık gelir.

İspat: Önerme 3.1.13'deki nokta formülüne dikkat edilirse $x_i = \sum_{j=0}^{\infty} \frac{\beta(S_j,i)}{2^{j+1}}$ ikili tabanda açılıma; $x_m = \sum_{j=1}^{\infty} \frac{\beta(S_{-j},i)}{2^{j(m-1)}}$ ise 2^{m-1} tabanında açılıma karşılık gelir. Ayrıca, aynı önermede verilen kısıtlamalar Lemma 3.1.14'teki şartlara karşılık gelmektedir. Dolayısıyla, H_∞ D'de eşsiz bir noktaya karşılık gelmelidir. ■

Bu önerme yardımıyla bir sembol uzayı tanımlayarak ÇBBH'nın dinamiklerini sembolik dinamikler yardımıyla inceleyebiliriz. Kullanacağımız sembol uzayı aşağıdaki gibi tanımlanmaktadır.

Tanım 3.1.16. Σ^*, $S_j \in \{0,1,\dots,2^{m-1}-1\}$ olmak üzere $(\dots S_{-2}S_{-1} \cdot S_0S_1S_2 \dots)$ şeklinde olup aşağıda verilen şartları sağlayan sonsuz dizilerin bir kümesidir:

i. Her $j < -N$ için $S_j = 2^{m-1}-1$ şartını sağlayan bir $N \in \mathbb{N}^+$ yoktur,
ii. Her $j > N$ için $\beta(S_j, i) = 1$ şartını sağlayan bir $i \in \{1,2,\dots,m-1\}$ ve $N \in \mathbb{N}$ yoktur.

Ayrıca, eğer $u \in \Sigma^*$ ise $u_{[-k,k]}$ ifadesi $(S_{-k} \dots S_{-1} \cdot S_oS_1 \dots S_k)$ şeklindeki dizi parçasına karşılık gelir. Bu nedenle, $u_{[-k,k]}$ ifadesi temel k terim olarak adlandırılmaktadır. Eğer $u_{[-k,k]} = v_{[-k,k]}$ ise, u ve v temel k terimi paylaşmaktadır denilir.

Yukarıdaki sembolik uzayda verilen semboller C_k tanımındaki k değerlerinin etiketleri olarak kabul edildiğinde, uzay tanımında verilen şartlar Önerme 3.1.13'de verilen kısıtlamalara karşılık geldiğinden, Σ^*'deki her dizi D'de eşsiz bir noktaya karşılık gelmelidir. Bu sembol dizilerinin gösterimindeki nokta, j değerlerinin negatiften pozitife geçişini belirtmektedir. Lemma 3.1.5'ten hatırlanacağı üzere, S_j sembolleri arasındaki herhangi bir geçiş mümkündür. Ancak, Tanım 3.1.16'da verilen şartları ihlal eden sembol dizilerinden kaçınılmalıdır.

Tanım 3.1.17. $u, v \in \Sigma^*$ ve $K = \{n \in \mathbb{N} \mid u_{[-n,n]} = v_{[-n,n]}\}$ olsun. $k = \max(K)$ olmak üzere,

$$d(u,v) = \begin{cases} 0, & u = v \\ 1, & K = \emptyset \\ \dfrac{1}{2^{k+1}}, & diğer \end{cases}$$

şeklindedir.

Sonraki önerme (Σ^*, d)'nin bir metrik uzay oluşturduğunu söylemektedir. $u, v, w \in \Sigma^*$ olmak üzere (Σ^*, d)'nin metrik uzay olabilmesi için $d(u,v) \geq 0$, $d(u,v) = 0 \Leftrightarrow u = v$, $d(u,v) = d(v,u)$ ve üçgen eşitsizliği $d(u,v) \leq d(u,w) + d(w,v)$ şartları sağlanmalıdır.

Önerme 3.1.18. (Σ^*, d) metrik bir uzaydır.

İspat: $u, v, w \in \Sigma^*$ olsun. Dikkat edilirse $d(u,v) \geq 0$, $d(u,v) = d(v,u)$ ve $d(u,v) = 0$ ancak ve ancak $u = v$ olursa doğrudur. Dolayısıyla üçgen eşitsizliği $d(u,v) \leq d(u,w) + d(w,v)$'nin doğru olduğu gösterilmelidir. Eğer $d(u,v) = 0$ ise ispat tamamlanmış olur. Eğer bir $k \in \mathbb{N}$ için $d(u,v) = 1$ ve $d(u,w) = 1/2^{k+1}$ ise, w ve v temel k terimi paylaşamaz. Çünkü bu durum mümkün olmayacak şekilde u ve v'nin de temel k terimi paylaşacağı anlamına gelir. Dolayısıyla, $d(w,v) = 1$ üçgen eşitsizliğini karşılar. Son olarak, mümkün olan üçüncü durumu inceleyelim. Bu durumda, bir $k \in \mathbb{N}$ için $d(u,v) = 1/2^{k+1}$ olmaktadır ve üçgen eşitsizliğinin eşdeğeri olan $d(u,v) - d(u,w) \leq d(w,v)$ ispatlanacaktır. $d(u,w) = 1/2^{k'+1} < d(u,v) = 1/2^{k+1}$ olsun. Aksi takdirde ispat tamamlanmış olur. Bu durum için $k' > k$ olmak üzere u ve w temel k' terimi; u ve v temel k terimi paylaşmaktadır. Bu yüzden v ve w aynı zamanda temel k terimi paylaşmalıdır ve $d(w,v) = 1/2^{k+1}$'dir. Böylece son olası durum da ispatlanmış olur. ■

Bölüm 1.1'de verilen kaydırma haritasının buradaki iki yönlü semboliklere uygulanışı Tanım 3.1.19'da verilmektedir.

Tanım 3.1.19. Kaydırma fonksiyonu σ aşağıdaki gibidir:

$$\sigma(\dots S_{-2}S_{-1} \cdot S_0 S_1 S_2 \dots) = (\dots S_{-1}S_0 \cdot S_1 S_2 S_3 \dots)$$

Dikkat edilirse, $u \in \Sigma^*$ için $\sigma(u) \in \Sigma^*$'dir. Yani, σ Σ^*'yı Σ^*'ya haritalar.

H_∞ kullanılarak D'deki her noktaya karşılık bir sembol dizisi atayabiliriz. Bu ilişki Tanım 3.1.20'deki ϕ fonksiyonu ile sağlanmaktadır.

Tanım 3.1.20. ϕ: $D \to \Sigma^*$ sembolik dizi döndüren aşağıdaki gibi bir fonksiyondur:

$$\phi(\mathbf{x}) = \phi\left(\bigcap_{j\in\mathbb{Z}} F^{-j}\left(C_{S_j}\right)\right) = (\dots S_{-1} \cdot S_0 S_1 \dots)$$

Lemma 3.1.21. ϕ bijeksiyondur.

İspat: Önerme 3.1.15'e göre her $\bigcap_{j\in\mathbb{Z}} F^{-j}\left(C_{S_j}\right)$ verilen kısıtlamalar altında eşsizdir. Bu yüzden, her $\phi\left(\bigcap_{j\in\mathbb{Z}} F^{-j}\left(C_{S_j}\right)\right) \in \Sigma^*$ sembolik dizisi D'de eşsiz bir noktaya karşılık gelir ve ϕ birebirdir. Aynı zamanda, Σ^* uzayına ait Tanım 3.1.16'daki kısıtlamalar Önerme 3.1.13'deki kısıtlamalara karşılık geldiğinden ϕ örten olmalıdır. ■

ϕ bijeksiyon olduğu için terslendirilebilir bir fonksiyondur. Şimdiki amacımız ϕ'yi konjuge fonksiyon olarak kullanarak ÇBBH F ve kaydırma fonksiyonu σ arasında bir topolojik konjuge oluşturmaktır. Bunun için öncelikle ϕ ve tersinin sürekli olduğu gösterilmelidir. Farklı metrik uzaylar arasındaki noktasal süreklilik şu şekilde formüle edilmektedir: Her $\varepsilon > 0$ için, $\mathbf{y} \in D$ noktalarında $\|\mathbf{x}-\mathbf{y}\| < \delta$ olduğunda $d\big(\phi(\mathbf{x}),\phi(\mathbf{y})\big) < \varepsilon$ şartını sağlayan bir $\delta > 0$ bulunabiliyorsa, ϕ fonksiyonu $\mathbf{x} \in D$ noktasında süreklidir denilir. Ters fonksiyon için de tanım buna benzerdir.

Lemma 3.1.22. ϕ süreklidir.

İspat: Herhangi bir $\mathbf{x} \in D$ noktası $\{\mathbf{x}\} = \bigcap_{j\in\mathbb{Z}} F^{-j}\left(C_{S_j}\right)$ şeklinde temsil edilebilir. $1/2^{N+1} < \varepsilon$ olacak şekilde bir $N \in \mathbb{N}$ ve $\|\mathbf{x}-\mathbf{y}\| < \delta$ şartını sağlayan her $\mathbf{y} \in D$ noktasının $\bigcap_{j=-N}^{N} F^{-j}\left(C_{S_j}\right)$ içinde kalmasını sağlayacak bir δ seçelim. Bu durumda, $\phi(\mathbf{x})$ ve $\phi(\mathbf{y})$'nin temel N terimi aynı olmalıdır. Bu yüzden, $\|\mathbf{x}-\mathbf{y}\| < \delta$ olduğunda $d\big(\phi(\mathbf{x}),\phi(\mathbf{y})\big) \le 1/2^{N+1} < \varepsilon$ olacaktır. ■

Lemma 3.1.23. ϕ^{-1} süreklidir.

İspat: Verilen bir $u \in \Sigma^*$ için, m D'nin boyutu olmak üzere $\sqrt{\sum_{i=1}^{m-1}\left(\frac{1}{2^{N+1}}\right)^2 + \left(\frac{1}{2^{N(m-1)}}\right)^2} < \varepsilon$ şartını sağlayan bir $N \in \mathbb{N}$ seçelim ve $\delta = 1/2^{N+1}$ olsun. Bu durumda, $d(u,v) < \delta$ ifadesi u ve $v \in \Sigma^*$ dizilerinin aynı temel N terimi paylaşacağı anlamına gelecektir. Yani $\phi^{-1}(u)$ ve $\phi^{-1}(v)$ noktaları $\bigcap_{j=-N}^{N} F^{-j}\left(C_{S_j}\right)$ içerisinde kalmalıdır. Dolayısıyla, gerektiği gibi

$$\|\phi^{-1}(u)-\phi^{-1}(v)\| \le \operatorname{diam}\left(\bigcap_{j=-N}^{N} F^{-j}\left(C_{S_j}\right)\right) = \sqrt{\sum_{i=1}^{m-1}\left(\frac{1}{2^{N+1}}\right)^2+\left(\frac{1}{2^{N(m-1)}}\right)^2} < \varepsilon$$

olmalıdır. ■

Şimdi topolojik konjuge bağıntısı Önerme 3.1.24 ile kurulabilir.

Önerme 3.1.24. σ ve F konjuge fonksiyonu ϕ altında topolojik olarak konjugedir.

İspat: Lemma 3.1.21, 3.1.22 ve 3.1.23 birlikte kullanıldığında ϕ'nin bir homeomorfizm olduğu görülmektedir. Dolayısıyla, $\phi \circ F = \sigma \circ \phi$ olduğu gösterilmelidir. Bu ilişki ise aşağıdaki gibi bulunmaktadır:

$$\begin{aligned}
\phi\big(F(x)\big) &= \phi\left(F\left(\bigcap_{j\in Z} F^{-j}\left(C_{S_j}\right)\right)\right)\\
&= \phi\left(\bigcap_{j\in Z} F^{-j+1}\left(C_{S_j}\right)\right)\\
&= (\dots S_{-1}S_0 \cdot S_1S_2 \dots)\\
&= \sigma(\dots S_{-2}S_{-1} \cdot S_0S_1 \dots)\\
&= \sigma\big(\phi(x)\big). \blacksquare
\end{aligned}$$

Bölüm 1.1'de verilen Devaney'in kaos tanımının (Tanım 1.1.15) ilk iki şartı topolojik özelliklerdir. Bu nedenle, bu özellikler topolojik konjuge altında korunur [66]. Bundan sonra verilecek sonuçlar σ'nın Devaney tanımındaki ilk iki şartı sağladığını gösterecektir. Önerme 3.1.24 ile verilen topolojik konjuge ise, ÇBBH F'nin de bu özelliklere sahip olacağını garanti altına alacaktır. Bu özellikleri gösterebilmek için öncelikle [67]'de verilen aşağıdaki teoreme ihtiyaç vardır.

Teorem 3.1.25. $f: X \to X$ bir harita olsun. Eğer X'in izole noktaları yoksa, yoğun bir yörüngenin varlığı topolojik geçişlilik anlamına gelir [67].

Önerme 3.1.26. $\sigma: \Sigma^* \to \Sigma^*$ topolojik geçişlidir.

İspat: $u = (S_j)$ aşağıdaki gibi bir sembol dizisi olsun:

$$u = \cdots S_{-1} \cdot 0\ 1 \ldots (2^{m-1}-1)\ 00\ 01 \ldots (2^{m-1}-1)(2^{m-1}-1)\ 000\ 001 \ldots$$

Burada $(S_j)_{j<0}$ Tanım 3.1.16'ya uygun herhangi bir dizi olabilir. $(S_j)_{j>0}$ dizisi ise $l \geq 1$ için l uzunluklu her bit bloğu kombinasyonuna yukarıdaki gibi sahiptir. Bu tanımlamaya göre her $N > 0$ için $S_k = 0$ şartını sağlayan bir $k > N$ vardır. Dolayısıyla, Tanım 3.1.16'ya göre $u \in \Sigma^*$'dir. Dikkat edilirse $\sigma^n(u)$ bir $n > 0$ değeri için herhangi bir $v \in \Sigma^*$ ile ilk k terimi paylaşacaktır. Bu yüzden, her $\varepsilon > 0$ ve $v \in \Sigma^*$ için $d(\sigma^n(u), v) = 1/2^{k+1} < \varepsilon$ şartını sağlayan bir n seçebiliriz. Yani $u \in \Sigma^*$'nin σ altında yoğun bir yörüngesi vardır.

Şimdiyse Σ^*'nın izole noktaları olmadığını göstereceğiz. $u \in \Sigma^*$'nın her δ komşuluğu için u ile aynı temel k terimi paylaşan başka bir $v \in \Sigma^*$ bulabiliriz. Bu durumda, $d(u, v) = 1/2^{k+1} < \varepsilon$ olacaktır. $(k+1)$'inci sembolü farklı olan bir dizi seçmek bunun için yeterli olacaktır. Bu Σ^*'nın izole noktalara sahip olmadığını gösterir ve Teorem 3.1.25'e göre σ topolojik geçişlidir. ■

Tanım 3.1.27. σ'nın periyodik noktaları kendini her iki yönde tekrar eden sembol dizileridir. Örneğin, $m = 3$ ve $u = (\ldots 210210 \cdot 210210 \ldots)$ olsun. Bu tür tekrarlanan sembolleri Bölüm 1.1'dekine benzer şekilde $(\overline{210 \cdot 210})$ ile göstereceğiz. Dikkat edilirse, bu örnek için $\sigma^3(u) = u$ olmaktadır. Bu yüzden, u noktası σ'nın bir periyot-3 noktasıdır. Eğer $u \in \Sigma^*$ σ'nın bir periyodik noktası ise, her $n \in \mathbb{Z}$ için $\sigma^n(u)$ da σ'nın bir periyodik noktasıdır. Genel olarak, σ'nın periyodik noktaları $u = (\overline{u_0 \ldots u_k \cdot u_0 \ldots u_k})$ şartını sağlayan noktalar olarak tanımlanmaktadır.

Önerme 3.1.28. σ'nın periyodik noktaları Σ^*'da yoğundur.

İspat: $u = (\ldots u_{-k} \ldots u_{-1} \cdot u_0 \ldots u_k \ldots)$ Σ^*'daki herhangi bir nokta olsun. Verilen bir $\varepsilon > 0$ için $1/2^{k+1} < \varepsilon$ olacak şekilde bir $k \in \mathbb{N}$ seçelim. Aynı zamanda, $v = (\overline{u_{-k} \ldots u_0 \ldots u_k 0 \cdot u_{-k} \ldots u_0 \ldots u_k 0})$ olacak şekilde bir v dizisi seçelim. Dikkat edilirse $v_{-1} = v_{2k+1} = 0$'dır. Tekrar eden sıfırların dizide bulunması v'nin Tanım 3.1.16'daki şartları yerine getireceğini garantiler. Bu nedenle $v \in \Sigma^*$'dir. Buna göre, $\sigma^k(v) \in \Sigma^*$ bir periyodik noktadır. Görülebileceği üzere de $\sigma^k(v)$ ve u (en az) temel k terimi paylaşmaktadır. Dolayısıyla, $d(\sigma^k(v), u) \leq 1/2^{k+1} < \varepsilon$'dur. Böylelikle, Tanım 1.1.16'ya göre istenen sonuç elde edilmiş olur. ■

Topolojik konjuge nedeniyle, Önerme 3.1.26 ve Önerme 3.1.28'de elde edilmiş olan sonuçlar aynı zamanda ÇBBH F için de geçerlidir. Devaney'in kaos tanımındaki (Tanım 1.1.15) son şart olan başlangıç şartlarına hassas bağlılık (b.ş.h.b.) ise metrik bir özellik olup topolojik konjuge altında korunmaz. Fakat, bir sürekli f fonksiyonu için tanım kümesi sonlu değilse, birinci ve ikinci şartlar b.ş.h.b. şartını kendiliğinden sağlamaktadır [66]. Yani bahsedilen durumlar için b.ş.h.b. şartı gereksizdir. Bizim durumumuzda ise F sürekli olmadığından bu durumun F için de geçerli olup olmadığı net değildir. Dolayısıyla, çalışmanın eksiksiz olması adına b.ş.h.b. özelliği burada F için ayrıca ispatlanacaktır. b.ş.h.b. Tanım 1.1.8'de tanımlanmaktadır.

Önerme 3.1.29. F b.ş.h.b. özelliği gösterir.

İspat: $\mathbf{x}$ D'deki herhangi bir nokta olsun. Daha önceki sonuçlardan elde ettiğimiz üzere $\mathbf{x}$ noktası $\bigcap_{j\in\mathbb{Z}} F^{-j}\left(C_{S_j}\right)$ ile temsil edilebilir. Dikkat edilirse, $\bigcap_{j\in\mathbb{Z}} F^{-j}\left(C_{S_j}\right)$ ifadesi $\left\{\mathbf{x}\in D \mid \mathbf{x}\in C_{S_0}, F(\mathbf{x})\in C_{S_1}, \dots, F^N(\mathbf{x})\in C_{S_N}\right\}$ ifadesine eşdeğerdir. Dolayısıyla, $\{\mathbf{x}\} = \bigcap_{j\in\mathbb{Z}} F^{-j}\left(C_{S_j}\right)$'nin ileri yönlü bir güzergahı $\phi^+(\mathbf{x}) = S_0S_1 \dots S_N$ şeklindeki bir sembol dizisi ile ifade edilebilir. Bu durumda, $\|\mathbf{y}-\mathbf{x}\| < \operatorname{diam}\left(\bigcap_{j=-N}^{N} F^{-j}\left(C_{S_j}\right)\right)$ şartını sağlayan herhangi bir $\mathbf{y}\in D$ noktası N'inci terime kadar aynı ileri güzergaha sahip olacaktır. $\mathbf{x}$'in herhangi bir δ komşuluğu N_δ için $\operatorname{diam}\left(\bigcap_{j=-N}^{N} F^{-j}\left(C_{S_j}\right)\right) = \sqrt{\sum_{i=1}^{m-1}\left(\frac{1}{2^{N+1}}\right)^2 + \left(\frac{1}{2^{N(m-1)}}\right)^2} < \delta$ şartını sağlayan bir $N\in\mathbb{N}$ seçelim. Bu ileri güzergahları $\mathbf{x}$'in ileri güzergahıyla ilk N terimi aynı olan $\mathbf{y}\in N_\delta(\mathbf{x})$ noktaları bulabilmemizi garanti altına alır. $\mathbf{x}$ ve $\mathbf{y}\in N_\delta(\mathbf{x})$'nin ileri güzergahları sırasıyla $\phi^+(\mathbf{x}) = S_0S_1 \dots S_NS_{N+1}S_{N+2} \dots$ ve $\phi^+(\mathbf{y}) = S_0'S_1' \dots S_N'S_{N+1}'S_{N+2}' \dots$ olsun. Aynı zamanda, $i\in\{1,2,\dots,m-1\}$ olmak üzere $\left(\beta(S_{N+1},i)\,\beta(S_{N+2},i)\right)_2$ ifadesi bir n_i doğal sayısının ikili tabanda gösterimi olsun. $n_i' = (n_i+2) \bmod 4$ olmak üzere öyle bir $\mathbf{y}$ seçelim ki $\beta(S_{N+1}',i) = \beta(n_i',1)$ ve $\beta(S_{N+2}',i) = \beta(n_i',2)$ olsun. Böyle bir $\mathbf{y}\in N_\delta(\mathbf{x})$ noktasını C_{S_j}'ler arasında herhangi bir geçiş mümkün olduğu için seçebiliriz. Böylelikle $F^N(\mathbf{x})$ ve $F^N(\mathbf{y})$ noktalarının ileri güzergahları sırasıyla $\phi^+\left(F^N(\mathbf{x})\right) = S_{N+1}S_{N+2} \dots$ ve $\phi^+\left(F^N(\mathbf{y})\right) = S_{N+1}'S_{N+2}' \dots$ olacaktır. Lemma 3.1.5'e göre, bu güzergahlar sırasıyla aşağıda verilen kümelerin alt kümelerine ait noktalara karşılık gelir:

$$\left\{\mathbf{z} \in D \mid \frac{\beta(S_{N+1}, i)}{2} + \frac{\beta(S_{N+2}, i)}{4} \leq z_i < \frac{\beta(S_{N+1}, i)}{2} + \frac{\beta(S_{N+2}, i)}{4} + \frac{1}{4},\right.$$
$$\left. i \in \{1, \ldots, m-1\}\right\}$$
$$\left\{\mathbf{w} \in D \mid \frac{\beta(S'_{N+1}, i)}{2} + \frac{\beta(S'_{N+2}, i)}{4} \leq w_i < \frac{\beta(S'_{N+1}, i)}{2} + \frac{\beta(S'_{N+2}, i)}{4} + \frac{1}{4},\right.$$
$$\left. i \in \{1, \ldots, m-1\}\right\}$$

Yukarıdaki S'_{N+1} ve S'_{N+2} seçimleri $|z_i - w_i| > 1/4$ olacağını garanti altına alır. Böylelikle,

$$\|F^N(\mathbf{x}) - F^N(\mathbf{y})\| > \sqrt{(m-1)\left(\frac{1}{4}\right)^2} = \frac{\sqrt{m-1}}{4}$$

olur. Bu yüzden, Tanım 1.1.8'e göre $\varepsilon = \frac{\sqrt{m-1}}{4}$ seçimi sonuç için yeterlidir. ■

Önerme 3.1.29 Devaney'in kaos tanımındaki son şartı da sağlamış oldu. Hatırlanacağı üzere diğer iki şart, Önerme 3.1.26 ve Önerme 3.1.28 ile sağlanmıştı. Böylece, bu çalışmanın temel teoremi olan Teorem 3.1.30 ispatlanmıştır. Hatırlanacağı üzere, $m = 1$ durumunda F Bernoulli haritası olarak tanımlanırsa, ÇBBH her boyut için kaotik olacaktır.

Teorem 3.1.30. $F: D \to D$ Devaney'in kaos tanımına göre her $m \geq 2$ için kaotiktir.

3.1.4. Çok Boyutlu Baker Haritasının Çekeri

Bu kısımda, F'nin ölçüm korumalı bir harita olduğu gösterilecektir. $\mathcal{L}(\mathbb{R}^m)$ (ya da kısaca $\mathcal{L}$) $\mathbb{R}^m$'deki tüm Lebesgue ölçülebilir kümelerin bir koleksiyonu olmak üzere $\mu: \mathcal{L}(R^m) \to [0, \infty)$ Lebesgue ölçümü olsun. Bahsettiğimiz sonuca ulaşabilmek için öncelikle aşağıdaki teoreme ihtiyacımız vardır.

Teorem 3.1.31. $\mathbf{a} \in \mathbb{R}^m$ bir sabit olsun. $T: \mathbb{R}^m \to \mathbb{R}^m$ terslendirilebilir bir lineer dönüşüm olmak üzere $\tau_{\mathbf{a}}: \mathbb{R}^m \to \mathbb{R}^m$ ifadesi $\tau_{\mathbf{a}}(x) = \mathbf{x} + \mathbf{a}$ şeklinde olsun. Buna göre,

i. Eğer $E \in \mathcal{L}$ ise, $\tau_{\mathbf{a}} \in \mathcal{L}$ ve $T(E) \in \mathcal{L}$'dir.
ii. Eğer $E \in \mathcal{L}$ ise, $\mu\big(\tau_{\mathbf{a}}(E)\big) = \mu(E)$ ve $\mu\big(T(E)\big) = |\det(T)|\mu(E)$'dir [137].

Sonuç 3.1.32. T terslendirilebilir bir matris olmak üzere $T_{\mathbf{a}}\colon \mathbb{R}^m \to \mathbb{R}^m$ ifadesi $T_{\mathbf{a}}(\mathbf{x}) = T\mathbf{x} + \mathbf{a}$ şeklinde bir afin dönüşüm olsun. Bu durumda, her $E \in \mathcal{L}$ için $T_{\mathbf{a}}(E) \in \mathcal{L}$ olup $\mu\big(T_{\mathbf{a}}(E)\big) = |\det(T)|\mu(E)$'dir.

İspat: Dikkat edilirse, $T(\mathbf{x}) = T\mathbf{x}$ olmak üzere $T_{\mathbf{a}} = \tau_{\mathbf{a}} \circ T$ şeklindedir. Teorem 3.1.31 (i)'e göre, eğer $E \in \mathcal{L}$ ise $T(E) \in \mathcal{L}$ olmalıdır. Benzer şekilde, eğer $T(E) \in \mathcal{L}$ ise $T_{\mathbf{a}}(E) = \tau_{\mathbf{a}}\big(T(E)\big) \in \mathcal{L}$ olmalıdır. Dolayısıyla, Teorem 3.1.31 (ii) kullanılarak

$$\mu\big(T_{\mathbf{a}}(E)\big) = \mu\Big(\tau_{\mathbf{a}}\big(T(E)\big)\Big) = \mu\big(T(E)\big) = |\det(T)|\mu(E)$$

ilişkisi kurulabilir. ■

Önerme 3.1.33. F ölçüm korumalı bir haritadır. Yani, $E_k \subseteq D_k$ ölçülebilir kümeler olmak üzere her $E = \bigcup_{k=0}^{2^{m-1}-1} E_k$ için $\mu\big(F^{-1}(E)\big) = \mu(E)$'dir.

İspat: Ölçülebilir kümelerin birleşimi de ölçülebilir olduğundan E de ölçülebilir olacaktır. Ayrıca, D_k kümeleri ayrık olduğundan $E_k \subseteq D_k$ kümeleri de ayrıktır. Bu ilişkileri kullandığımızda

$$\mu(E) = \mu\left(\bigcup_{k=0}^{2^{m-1}-1} E_k\right) = \sum_{k=0}^{2^{m-1}-1} \mu(E_k)$$

sonucuna ulaşırız. Sonuç 3.1.32'ye göre $T_k^{-1}(E_k)$ ölçülebilir olup $\mu\big(T_k^{-1}(E_k)\big) = |\det(A^{-1})|\mu(E_k)$ olacaktır. $|\det(A^{-1})| = 1$ olduğundan bu ilişki

$$\sum_{k=0}^{2^{m-1}-1} \mu(E_k) = \sum_{k=0}^{2^{m-1}-1} \mu\big(T_k^{-1}(E_k)\big)$$

şeklindedir. Dikkat edilirse $T_k^{-1}(D_k) = C_k$ olup C_k kümeleri ayrıktır. Bu nedenle, $T_k^{-1}(S_k) \subseteq T_k^{-1}(D_k)$ kümeleri de ayrık olmalıdır. Böylelikle,

$$\sum_{k=0}^{2^{m-1}-1} \mu\big(T_k^{-1}(E_k)\big) = \mu\left(\bigcup_{k=0}^{2^{m-1}-1} T_k^{-1}(E_k)\right)$$

olur. Fakat,

$$\bigcup_{k=0}^{2^{m-1}-1} T_k^{-1}(E_k) = \bigcup_{k=0}^{2^{m-1}-1} F^{-1}(E_k) = F^{-1}\left(\bigcup_{k=0}^{2^{m-1}-1} E_k\right) = F^{-1}(E)$$

olmalıdır. Sonuç olarak, $\mu(E) = \mu\left(\bigcup_{k=0}^{2^{m-1}-1} T_k^{-1}(E_k)\right) = \mu\left(F^{-1}(E)\right)$'dir. ■

Önerme 3.1.33 kümelerin hacminin (yüksek boyutlu m-hacimlerinin) F altında değişmediğini ifade etmektedir. Buna göre, bir küme verildiğinde, F ile ne kadar dönüşüme sokulursa sokulsun, bu küme daha düşük boyutlu bir çeker yapısına yakınsayamayacaktır. Dolayısıyla, F'nin fraktal boyutlu bir garip (strange) çekeri olamaz ve F'nin yörüngeleri faz uzayında tüm D uzayını doldurmalıdır.

3.1.5. Lyapunov Üstelleri ve Periyodik Noktalar

Dikkat edilirse, $\mathbf{c}_i$ sabit terimler ve $n \in \mathbb{N}^+$ olmak üzere $F^n(\mathbf{x}) = A^n\mathbf{x} + \sum_{i=1}^{n} \mathbf{c}_i$ şeklindedir. Burada, $E = \bigcup_{i=1}^{m-1}\{\, \mathbf{x} \in D \mid x_i = 1/2 \,\}$ olmak üzere her $F^n(x) \in D \setminus E$ için

$$J_n = DF^n(\mathbf{x}) = A^n = \begin{bmatrix} 2^n & 0 & \cdots & 0 \\ 0 & 2^n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 2^{-n(m-1)} \end{bmatrix}$$

olur. $F^n(\mathbf{x}) \in E$ için J_n süreksizlik yüzünden tanımlı değildir. Tanım 1.1.14'teki gibi N bir m boyutlu küre olsun. A^n'in ortogonal bir matris olmasından dolayı, J_nN asal eksenleri standart birim vektörler yönünde olan bir elipsoid olmalıdır. $k \in \{1, \dots, m\}$ olmak üzere r_k^n ifadesi J_nN elipsoidinin k'nıncı asal ekseninin uzunluğunu temsil etsin. Bu durumda, her $i \in \{1, \dots, m-1\}$ için $r_i^n = 2^n$ ve $r_m^n = 2^{-n(m-1)}$ olacaktır. Sonuç olarak Lyapunov üstelleri, $i \in \{1, \dots, m-1\}$ için

$$\lambda_i = \ln\left(\lim_{n\to\infty} (2^n)^{1/n}\right) = \ln 2$$

ve

$$\lambda_m = \ln\left(\lim_{n\to\infty} \left(2^{-n(m-1)}\right)^n\right) = -(m-1)\ln 2$$

şeklinde olmalıdır. $\mu(E) = 0$ olmasından dolayı, hemen hemen her yörünge için F'nin $m-1$ pozitif Lyapunov üsteli ve bir tane de negatif Lyapunov üsteli olmalıdır. Bu bakımdan, ÇBBH F hiperkaotik bir haritadır [138].

Sonraki önerme F'nin periyodik noktalarının türünü belirlemektedir. Bu bilgi sonraki kısımlardaki pratik gerçekleştirim uygulamaları açısından önem taşımaktadır.

Önerme 3.1.34. F'nin periyodik noktaları rasyonel sayılardır.

İspat: $u = (\dots S_{-1} \cdot S_0 S_1 \dots)$ Σ^*'daki bir sembol dizisi olmak üzere Önerme 3.1.13'e göre bu sembol dizisine karşılık gelen başlangıç şartı

$$\mathbf{x} = \begin{pmatrix} \sum_{j=0}^{\infty} \frac{\beta(S_j, 1)}{2^{j+1}} \\ \vdots \\ \sum_{j=0}^{\infty} \frac{\beta(S_j, m-1)}{2^{j+1}} \\ \sum_{j=1}^{\infty} \frac{S_{-j}}{2^{j(m-1)}} \end{pmatrix}$$

şeklindedir. Önerme 3.1.24'teki topolojik konjuge yüzünden, eğer u dizisi σ'nın bir periyot-k noktası ise $\mathbf{x}$ de F'nin bir periyot-k noktası olmalıdır. Tanım 3.1.27'den bildiğimiz üzere σ'nın periyodik noktaları kendini her iki yönde tekrar eden dizilerdir. Eğer u böyle bir diziyse $\beta(S_j, i)$ değeri j ile birlikte periyodik olarak kendini tekrar etmeli. Bu nedenle, her x_i'nin ikili tabanda açılımları tekrar eden rakamlardan oluşmalıdır. Bu tür ikili açılımlar ise, ikinci bölümden bildiğimiz üzere rasyonel sayılara karşılık gelmektedir. x_m'in 2^{m-1} tabanındaki açılımı için de benzer bir argüman kurulduğunda x_m'in de rasyonel sayı olması gerektiği sonucuna varılır. ■

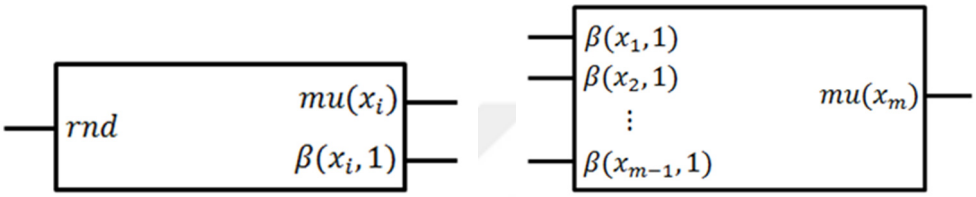
3.2. Çok Boyutlu Baker Haritasının Dijital Gerçekleştirimleri

Bu kısımda ÇBBH'nın LSB uzatma metodu ile dijital gerçekleştirimleri verilmektedir. ÇBBH'nın bir İKKH olmasından dolayı bu harita basit lojik işlemler ve kaydırmalar ile gerçekleştirilebilmektedir. Aşağıda gösterilmekte olduğu üzere, ÇBBH'nin bu gerçekleştirimleri her $m \geq 2$ boyutu için sadece iki temel yapı bloğu kullanılarak yapılabilmektedir. Bu iki temel yapı bloğu kullanılarak yapılan gerçekleştirimlerin Eşitlik

(3.2) ile verilen F haritasına karşılık geldiği de yine bu kısımda matematiksel olarak gösterilmektedir. Bu bloklar kullanılarak yapılan yazılım gerçekleştirimi $m = 3$ durumu içinken; FPGA üzerinde yapılan donanım gerçekleştirimi $m = 4$ durumu içindir.

3.2.1. Temel Yapı Blokları ve Yazılım Gerçekleştirimi

Önerilen temel yapı blokları sırasıyla x_i bloğu ve x_m bloğu olarak adlandırılmaktadır. Bu bloklar Şekil 3.1'de görüldüğü gibidir.



(a) (b)

Şekil 3.1 (a) x_i bloğu; (b) x_m bloğu

Bu bloklar sabit L uzunluklu bir kaydedici veya LSB uzatma metodunda mu ile ifade edilen bir hafıza birimi içeren lojik bloklardır. Elbette, bu hafıza birimi bir saat darbesi ile güncellenmektedir. Fakat, gösterim kolaylığı olması açısından saat darbesi girişi Şekil 3.1'de ihmal edilmiştir. Önceki bölümden hatırlanacağı üzere hafıza biriminin boyutu mimariye ve dijital platforma bağlı olarak 16, 32 veya 64 olabilir. Fakat, m boyutuna bağlı olarak $L \gg m - 1$ seçilmelidir. $mu(x_i)$ ve $mu(x_m)$ çıkışları hafıza biriminin çıkışlarını ifade etmektedir. Şekildeki β fonksiyonu ise Tanım 3.1.1'de verilen fonksiyonun aynısıdır. Son olarak, x_i bloğundaki rnd girişi LSB uzatma metodu için gerekli olan rastgele bit girişidir.

x_i bloğu hafıza birimini sadece sola kaydırmaktadır. Bu sola kaydırma işleminden sonra ise LSB bitinin yeni değeri rnd girişine göre belirlenmektedir. $i \in \{1, \dots, m-1\}$ olmak üzere, $x_i(n)$ ve $x_m(n)$ sırasıyla x_i ve x_m'in n iterasyon sonundaki değerlerini temsil etsin. Bu durumda, Eşitlik (3.2)'ye göre bir $k \in \{0, \dots, 2^{m-1} - 1\}$ değeri için $x_i(n+1) = 2x_i(n) - \beta(k, i)$ olacaktır. Ancak, $\beta(k, i) \in \{0,1\}$ olmasından dolayı bu eşitlik k değerinden bağımsız olarak x_i bloğunda yapılan tek bir sola kaydırmaya karşılık gelmektedir. LSB bitini rnd ile değiştirmek bu eşitliği bozmamakta; sadece LSB uzatma metoduna uygun olarak çözünürlüğü arttırmaktadır. Gerçekleştirimde x_i bloğundan $m - 1$ tane kullanmak gerekmektedir.

Diğer taraftan, dijital gerçekleştirim için tek bir x_m bloğu yeterlidir. Şekil 3.1 (b)'den görülebileceği üzere x_m bloğunun $m-1$ girişi vardır. x_i bloğunun çıkışları bu girişlere bağlanmaktadır. Her iterasyonda, x_m bloğu kendi hafıza birimini $m-1$ bit sağa kaydırmaktadır. Sonrasında, her giriş $\beta(mu(x_m), i) = \beta(x_i, 1)$ ilişkisi kullanılarak hafıza biriminin belli değerlerine atanır. x_m bloğunun bu çalışma prensibi ile Eşitlik (3.2) arasındaki ilk bakışta anlaşılamayabilir. Bu ilişkiyi görebilmek için öncelikle bir $k \in \{0, \dots, 2^{m-1}-1\}$ değeri için Eşitlik (3.2)'ye göre $x_m(n+1) = (x_m(n)+k)/2^{m-1}$ olduğuna dikkat edilmelidir. Bu işlem $mu(x_m)$ hafıza birimini sağa $m-1$ kaydırdıktan sonra, elde edilen değere $k/2^{m-1}$ eklemekle eşdeğerdir. Burada, $mu(x_m)$'i $k/2^{m-1}$ ile toplamak yerine $\beta(mu(x_m), i) = \beta(k, i)$ ataması yapılabilir. Ayrıca Eşitlik (3.3) kullanılarak şu matematiksel ilişkiler elde edilmektedir:

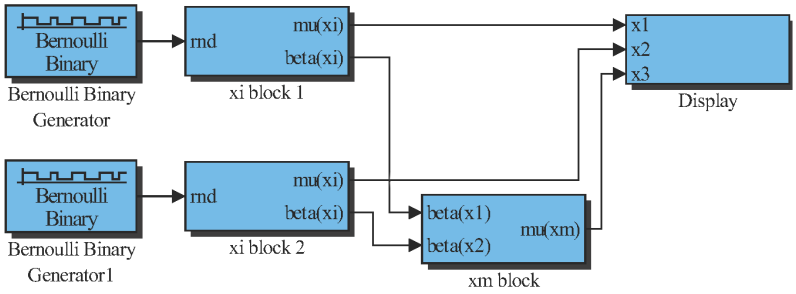
i. $\beta(x_i, 1) = 0 \Leftrightarrow x_i < 1/2 \Leftrightarrow \beta(k, i) = 0$
ii. $\beta(x_i, 1) = 1 \Leftrightarrow x_i \geq 1/2 \Leftrightarrow \beta(k, i) = 1$

Bu iki ilişkiyi birleştirdiğimiz zaman $\beta(k, i) = \beta(x_i, 1)$ olduğunu görürüz ve bu da $\beta(mu(x_m), i) = \beta(k, i) = \beta(x_i, 1)$ sonucunu verir.

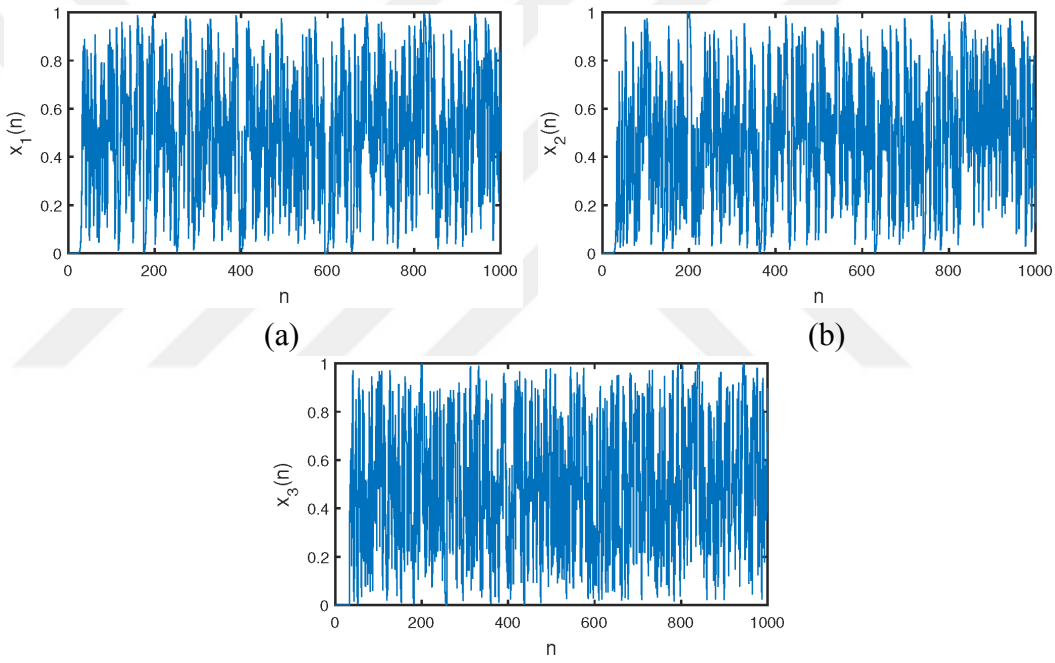
Özetlemek gerekirse, Eşitlik (3.2) ile verilen ÇBBH yalnızca iki temel yapı bloğu kullanılarak dijital olarak gerçekleştirilebilir. Dijital gerçekleştirim için $m-1$ tane x_i bloğuna ve bir tane de x_m bloğuna ihtiyaç vardır. Her iki temel yapı bloğu da sadece lojik kaydırma ve bit atama işlemlerinden ibarettir. Dolayısıyla, hiçbir aritmetik işleme gereksinim duyulmamaktadır. Bit atama işlemleri herhangi bir metot kullanılarak yapılabilir. Örneğin, belli bir bit dizisi ile XOR işlemi ile maskeleme yapmak bunun için bir seçenektir.

$m = 3$ ve $L = 32$ için MATLAB gerçekleştirimi Şekil 3.2'de görülmektedir. Görülebileceği üzere iki x_i bloğu ve bir x_m bloğu kullanılmıştır. LSB uzatma metodu için gerekli olan rastgele bitler iki tane MATLAB'e ait SRSÜ yapısı kullanılarak elde edilmiştir. Buradaki "Display" bloğu üretilen yörüngeleri görüntülemek için kullanılmaktadır. Bu görüntüleme esnasında hafıza biriminin değeri $[0,1)$ arasında L bitlik kesirli sayı olarak alınmıştır. Üretilen yörüngeler Şekil 3.3'de verilmektedir. Bu yörüngelere karşılık gelen çeker yapısı ise Şekil 3.4'de görüldüğü gibidir. Önceki teorik

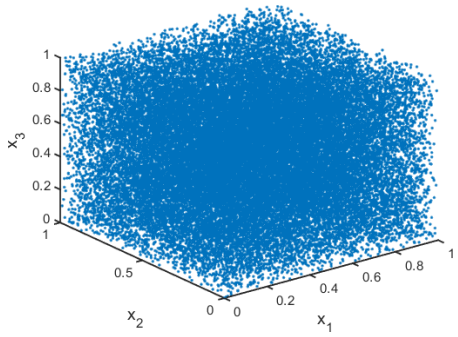
sonuçlardan beklediğimiz üzere yörüngeler tüm faz uzayını doldurmakta ve bir çeker yapısı oluşmamaktadır.



Şekil 3.2 3-boyutlu baker haritasının MATLAB gerçekleştirimi



Şekil 3.3 MATLAB gerçekleştirimi neticesinde elde edilen yörüngeler: (a) x_1; (b) x_2; (c) x_3



Şekil 3.4 MATLAB gerçekleştirimine ait faz uzayı

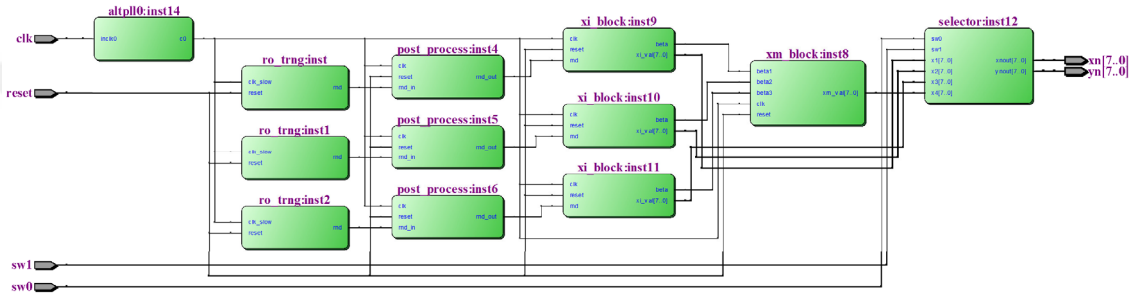
LSB uzatma metodunda SRSÜ'lerin kullanılmasından dolayı üretilen yörüngelerin başlangıç şartları ikili tabanda L bitten sonra SRSÜ'nün periyodu ile kendini takip eden sayılara karşılık gelmektedir. Önerme 3.1.34'ten bu tür sayıların ÇBBH'nın periyodik noktalarına karşılık geldiğini biliyoruz. Dolayısıyla, SRSÜ ile üretilen bu tür yörüngeler gerçek periyodik yörüngelerdir. Diğer taraftan, eğer GRSÜ'ler kullanılmış olsaydı yörüngelerin başlangıç şartları kendini asla tekrar etmeyen ikili tabandaki sayılar, yani irrasyonel sayılar olacaktır. Bu durumda, yine Önerme 3.1.34'e göre, üretilen yörüngeler gerçek kaotik yörüngeler olmalıdır. Dijital GRSÜ'ler ancak FPGA gibi donanımlar üzerinde kurulabildiklerinden gerçek kaotik yörünge üretimi sonraki kısımda yapılmaktadır.

3.2.2. FPGA Gerçekleştirimi

Basit bitsel işlemlere dayalı olmalarından dolayı, temel yapı blokları herhangi bir dijital platform üzerinde kolaylıkla gerçekleştirilebilir. Bu çalışmada dijital platform olarak bir kez daha DE0 geliştirme bordu üzerindeki Cyclone III FPGA çipi kullanılmıştır. FPGA gerçekleştirimi ile gerçek kaotik yörüngeleri elde edebilmek içinse yukarıda bahsettiğimiz üzere GRSÜ yapılarına ihtiyaç vardır. GRSÜ yapısı olarak [139]'da önerilen halka osilatör tabanlı yapı kullanılmıştır. Bu GRSÜ tasarımında birden fazla halka osilatörler kullanılmakta ve bu osilatörlerin çıkışı bir D flip-flopa bağlanmaktadır. Daha sonra flip-flopların çıkışı XOR ile birbirine bağlanmakta ve XOR kapısının çıkışı da son bir D flip-flopa bağlanmaktadır. Bu tasarıma göre, rastgele sayılar üretebilmek için 25 halka osilatör kullanmak yeterlidir [139]. Bu nedenle, FPGA gerçekleştiriminde her birinde 13 adet terslendirici bulunan 25 halka osilatör kullanılmıştır.

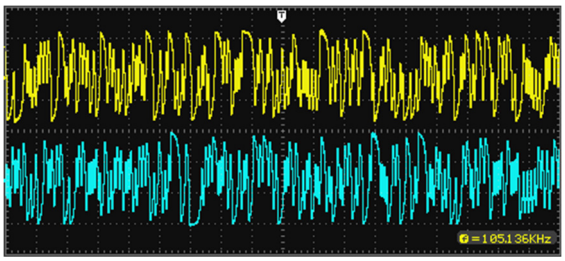
Tüm halka osilatörler, x_i blokları ve x_m blokları VHDL kullanılarak tanımlanmıştır. Gerçekleştirimde $m = 4$ ve $L = 32$ olduğundan üç tane x_i bloğu ve bir tane x_m bloğu kullanılmıştır. Yapılan FPGA gerçekleştirimine ait RTL (Register Transfer Level) şematiği Şekil 3.5'deki gibidir. Şekilde GRSÜ'ler "ro_trng" ile etiketlenmiş olup bu şekilden görülebileceği üzere üç tane halka osilatör tabanlı GRSÜ kullanılmıştır. GRSÜ'lerden hemen sonraki "post_process" blokları rastgele bitlerde oluşabilecek olası sapmaları (bias) gideren basit XOR düzelticileridir [134]. Bu doğrultuculardan sonra rastgele sayılar x_i bloklarına bağlanmaktadır. x_i ve x_m bloklarının ve halka osilatörlerdeki flip-flopların çalışması için gerekli olan sistem saati FPGA içerisine

gömülü PLL (“altpll0” bloğu) yardımıyla üretilmektedir. x_i ve x_m bloklarının sadece en çok anlamlı 8 biti çıkışa verilmekte ve bu çıkışlar “selector” adı verilen bloğa bağlanmaktadır. Bu blok $sw0$ ve $sw1$ bitlerinin değerine göre dört girişinden sadece ikisini çıkışa vermektedir. $sw0$ ve $sw1$ bitleri ise DE0 bordu üzerindeki iki anahtarla kontrol edilecek şekilde ayarlanmıştır. Böylelikle, 8 bitlik hafıza birimi değerlerinden ikisi seçilerek iki adet D/A dönüştürücüye verilmiş ve bu D/A dönüştürücü çıkışları da iki kanallı bir osiloskop yardımıyla gözlenmiştir. Bu sayede dört boyuttaki çıkışın da gözlemlenmesi (ikişer ikişer) mümkün olabilmiştir.

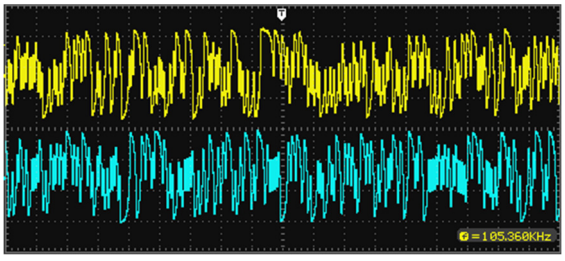


Şekil 3.5 FPGA gerçekleştirimine ait RTL şematiği

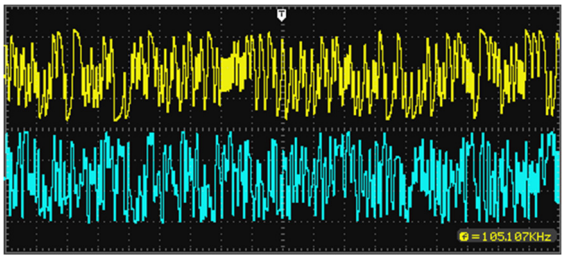
Osiloskobun iki kanalına ait (CH1 ve CH2) 8 bit çözünürlüklü yörünge görüntüleri Şekil 3.6’da verilmektedir. Bu yörüngelerin oluşturduğu faz uzayının osiloskop üzerindeki 2B izdüşümleri (CH1 kş. CH2) ise Şekil 3.7’de görüldüğü gibi elde edilmektedir. Beklendiği üzere yine bir çeker yapısı elde edilmemektedir. Dolayısıyla, gözlemlenen osiloskop görüntüleri önceki sonuçlarla uyumludur.



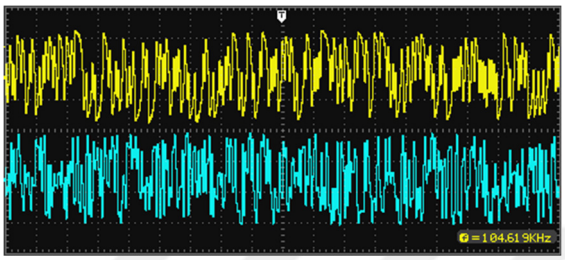
(a)



(b)

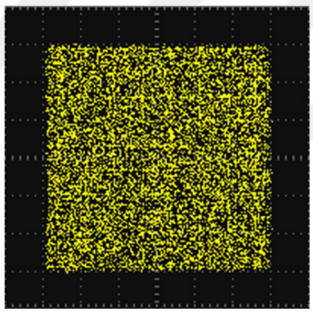


(c)

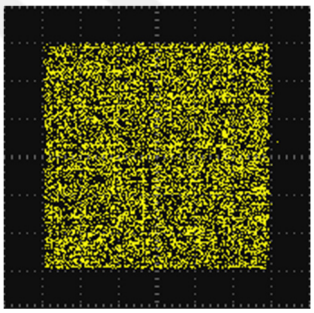


(d)

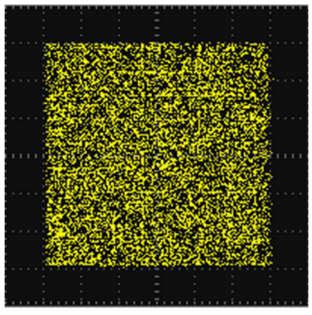
Şekil 3.6 D/A dönüştürücü çıkışlarına ait osiloskop görüntüleri (CH1 ve CH2): (a) x_1 ve x_2; (b) x_1 ve x_3; (c) x_1 ve x_4; (d) x_3 ve x_4;



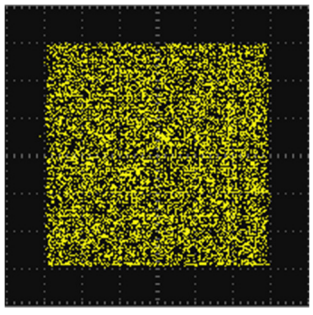
(a)



(b)



(c)



(d)

Şekil 3.7 Osiloskop faz görüntüleri (CH1 kş. CH2): (a) x_1 kş. x_2; (b) x_1 kş. x_3; (c) x_1 kş. x_4; (d) x_3 kş. x_4;

FPGA gerçekleştirimi toplamda 1047 lojik blok kullanmaktadır ve bu toplam kaynakların yüzde 7'sine karşılık gelmektedir. Kaynakların çoğu halka osilatör tabanlı GRSÜ gerçekleştiriminde kullanılmaktadır. Aynı gerçekleştirim LFSR'ler kullanılarak

yapıldığında kaynak tüketiminin sadece 365 olduğu (toplam kaynakların yüzde 2'si) görülmüştür. Fakat, LFSR kullanıldığında elde edilen yörüngelerin kaotik olmadığı; gerçek periyodik yörüngeler olduğu unutulmamalıdır. Gerçekleştirim için FPGA'in maksimum çalışma frekansı 255.23 MHz olarak belirlense de uygulamada 1 MHz'lik bir sistem saati kullanılmıştır. FPGA gerçekleştirim sonuçları Tablo 3.1'de özetlenmektedir.

Tablo 3.1 FPGA gerçekleştirim sonuçları

Model	Alan Kullanımı (Lojik Blok)	Maks. Çalışma Frekansı
4B baker haritası (GRSÜ)	1047 (% 7)	255.23 MHz
4B baker haritası (LFSR)	365 (% 2)	315.06 MHz
GRSÜ	333 (% 2)	305.81 MHz

4. BÖLÜM

KAOS TABANLI KRİPTOGRAFİDE GERÇEK PERİYODİK YÖRÜNGELERİN KULLANILMASI

Daha önceki bölümlerde gösterilmiş olduğu üzere kaos tabanlı kriptografide kaotik sistemlerin dijital gerçekleştirimleri ciddi sorunlara neden olmaktadır. Bunun yanı sıra, reel sayılar kümesinde tanımlı sistemlerin sabit veya kayan noktalı aritmetik ile sonlu çözünürlüklü dijital platformlarda kullanılması, tam sayılar kümesinde tanımlı klasik kriptografiye uygun bir uygulama tarzı olmamaktadır. Hem kaotik sistemlerin dijital gerçekleştirim problemlerine çözüm getirmek, hem de basit bitsel işlemler ile kaotik yörüngeler üreterek klasik kriptografiye daha uygun bir tasarım seçeneği sunabilmek adına, önceki bölümlerde gerçek periyodik ve gerçek kaotik yörüngelerin üretilmesi için yeni tasarım metotları önerilmiştir. Şimdi karşımıza çıkan soru ise, üretilen bu yörüngelerin hangisinin kaos tabanlı kriptografide kullanılacağıyla alakalıdır.

İlk bakışta, gerçek kaotik yörüngelerin kullanılması bariz seçenek olarak görülebilir. Fakat, LSB uzatma metodunda kullanılan GRSÜ yapısı nedeniyle bir platformda üretilen kaotik yörüngenin aynısını başka bir platformda üretebilmek mümkün değildir. Yine de iki farklı platformda üretilen gerçek kaotik yörüngeleri senkronize etmek mümkündür. Mesela, [51]'de iki farklı platform üzerinde IDCS metodu ile üretilen iki farklı kaotik yörünge senkronize edilerek, basit bir şifreleme ve deşifreleme uygulaması yapılmıştır. Bu tür bir uygulama, sürekli zamanlı kaotik sistemlerin elektronik devre gerçekleştirimlerine dayalı analog güvenli haberleşme uygulamalarına oldukça benzerdir [6, 21, 23]. Ancak, daha önce bahsedildiği üzere, farklı kriptosistemler arasında senkronizasyon verisinin iletilme gerekliliği ciddi güvenlik problemlerine neden olmaktadır [140–142]. Bu nedenle, bu tür senkronizasyona dayalı analog güvenli haberleşmeye dair yapılan çalışmalar geçmiş yıllarda olduğu kadar canlı değildir. Ayrıca, modern kriptografik uygulamalarda senkronizasyona ihtiyaç duymadan sadece anahtar

güvenliğine dayalı uygulamalar yapılmaktadır. Bu tür bir senkronizasyon uygulamasına, modern kriptografide ne kadar ihtiyaç olduğu da bu bakımdan bir soru işaretidir.

Diğer taraftan, gerçek periyodik yörüngeler arada herhangi bir senkronizasyon sinyaline ihtiyaç duymaksızın kullanılabilmektedir. Ayrıca, çok yüksek periyot uzunluğuna sahip gerçek periyodik yörüngeler, pratik uygulamalar için gerçek kaotik yörüngelerden neredeyse farksız olacaktır. Dahası, LSB uzatma metodunda kullanılan SRSÜ'lerin başlangıç değerleri anahtar olarak kullanıldığında, aynı anahtar ile farklı dijital platformlarda aynı gerçek periyodik yörüngeyi üretmek mümkündür. Bu modern kriptografik uygulamalara daha uygun bir yaklaşım tarzıdır.

Bunun yanı sıra, LSB uzatma metodu ile periyot uzunluğu net olarak belli gerçek periyodik yörüngeleri kullanmak, periyot uzunluğu belirsiz olan sabit veya kayan noktalı aritmetik ile üretilmiş hatalı sözde kaotik yörüngeleri kullanmaktan daha güvenilirdir. Ayrıca, bu çalışma esnasında hatalı yörüngelerin kullanılması sırasında yörüngelerin yuvarlama hataları nedeniyle tanım aralığı dışına çıkması gibi bir problem daha keşfedilmiş olup, bu problemden de bu bölümde bahsedilecektir. Yuvarlama hatalarına maruz kalmadığından, gerçek periyodik yörüngeler bu tür bir problemden de muzdarip değildir.

Dahası, ikinci bölümde gösterildiği üzere, literatürdeki pek çok İKKH ve İKKH olmayan lojistik harita, Chebyshev haritası gibi diğer haritaların gerçek periyodik yörüngeleri üretilebilmektedir. Üçüncü bölümde önerilen ÇBBH ile de istenilen her boyut için gerçek periyodik yörüngeleri üretmek mümkündür. Bu bakımdan, gerçek periyodik yörüngelerin üretilmesi ve önceden önerilmiş kaos tabanlı kriptosistemlerdeki kaotik sistemlerin sabit / kayan noktalı gerçekleştirimleri yerine, bunların LSB uzatma metodu ile yapılan gerçekleştirimlerinin kullanılması için elimizde bolca seçenek mevcuttur. Ayrıca, LSB uzatma metodunun bilinen kaotik haritaları gerçekleştirmeye imkan tanımasından dolayı, IDCS metodunun aksine, bizim önerdiğimiz metot literatüre önceden sunulmuş olan kaos tabanlı kriptosistemlere kolayca entegre edilebilir. Dolayısıyla, kaos tabanlı kriptografide gerçek periyodik yörüngelerin kullanılması oldukça cazip bir fikirdir.

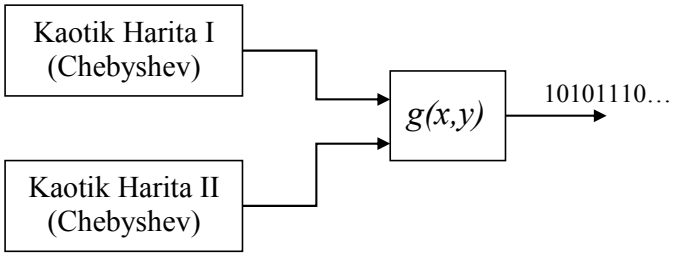
Fakat, LSB uzatma metodu ile üretilmiş gerçek periyodik yörüngelerin kaos tabanlı kriptografide doğrudan kullanımı ciddi bir probleme sahiptir. Dikkat edilirse, SRSÜ'lerin anahtar olarak kullanılacak başlangıç değerleri kaotik sistemin asıl başlangıç şartıyla aynı

değildir. Bu nedenle, bazı anahtar değerleri kriptografide ihtiyaç duyulan yayılma (diffusion) şartını sağlamayabilir [3, 4]. Bir başka deyişle, anahtarda yapılacak küçük bir değişiklik korelasyonu yüksek olan yörüngelerin üretilmesine neden olabilir. Sonraki kısımlarda böyle bir duruma örnek verilmektedir. Bu tür istenmeyen durumların önüne geçmek için gerçek periyodik yörüngelerin kaos tabanlı kriptografide kullanımında yeni tasarım yaklaşımlarına ihtiyaç vardır.

Bu çalışmada, öncelikle mevcut kaos tabanlı kriptosistem tasarımlarında yaygın olarak kullanılan sabit / kayan noktalı aritmetiğe dayalı sözde kaotik yörüngeler ile LSB uzatma metoduna dayalı gerçek periyodik yörüngeler karşılaştırılmaktadır. Bu bağlamda, sözde kaotik yörüngeler kullanan bir kaos tabanlı kriptosistem incelenmekte ve bu tür kriptosistemlerin, önceki bölümlerde bahsedilen sorunlar dışında, daha önce literatürde bilinmeyen başka sorunlar da sergileyebileceği gösterilmektedir. Sonrasında, aynı tasarımda sadece sabit / kayan noktalı harita gerçekleştirimlerinin LSB uzatma metodu ile yapılacak gerçekleştirimlerle değiştirilmesinin hem bahsedilen bu sorunu engellediği hem de ciddi güvenlik artışına neden olduğu gösterilmektedir. Son olarak ise, yukarıda bahsedilen yayılma sorununu göz önünde bulunduran yeni bir gerçek periyodik yörünge üretim tasarımı önerilmekte ve bu yeni tasarım ilk başta incelemiş olduğumuz kaos tabanlı kriptosisteme uyarlanmaktadır. Böylelikle, LSB uzatma metodu ile üretilmiş olan gerçek periyodik yörüngeler ilk kez bir kaos tabanlı kriptosisteme entegre edilmiş olmaktadır.

4.1. Kaos Tabanlı Temel Bir Kriptosistemin İncelenmesi

Bu kısımda literatürde çok kullanılan temel bir kaos tabanlı kriptosistem incelenecektir. İncelenecek kriptosistem iki tane kaotik haritanın çıkışının karşılaştırılması suretiyle çıkış bitleri üretmektedir [11, 91]. Bu kriptosistem SRSÜ veya akış şifreleyici olarak kullanılabilmekte olup Şekil 4.1'de görüldüğü gibidir.



Şekil 4.1 Eşleştirilmiş kaotik sistem (EKS)

İki kaotik haritanın karşılaştırılmasına dayalı olduğu için bu kriptosistem "eşleştirilmiş kaotik sistem" veya kısaca EKS olarak adlandırılmaktadır [11]. Birinci haritanın çıkışı x, ikincininki y olarak adlandırıldığında, Eşitlik (4.1)'deki karşılaştırma fonksiyonu g ile bit üretilmektedir.

$$g(x,y) = \begin{cases} 1, & x < y \\ 0, & \text{diğer} \end{cases} \tag{4.1}$$

Kaos tabanlı kriptografi için [4]'te ortaya koyulmuş olan standartlara göre, bu kriptosistemde herhangi bir 1B kaotik harita kullanılabilmelidir. Buradaki uygulama için Eşitlik (4.2) ile verilen Chebyshev haritası kullanılmıştır.

$$f(x) = 8x^4 - 8x^2 + 1 \tag{4.2}$$

Kriptosistemde literatürde sıklıkla başvurulduğu üzere sabit veya kayan noktalı aritmetikle sözde yörüngelerin üretildiği varsayılacaktır. Daha önceki bölümlerde bahsedildiği üzere, kaotik sistemlerin sabit veya kayan noktalı aritmetik ile yapılan gerçekleştirimlerinin kaotik olması mümkün değildir. Bu nedenle, bu tür bu tür gerçekleştirimlere dayalı kaotik sistemlerin kriptografide kullanılması pek çok soruna yol açmaktadır [32]. Mesela, [38]'de gösterildiği üzere bu tür gerçekleştirimler sonucu üretilen yörüngeler sadece transient kısmında orijinal bir çıkış vermekte; periyodik bölgeye düştükten sonra yörüngeler birbirlerinin aynısı olmaktadır. Bu ise, EKS akış şifreleyici olarak kullanıldığında şifrelenmiş veriden orijinal verinin anahtar bilinmeksizin elde edilebilmesine neden olmaktadır [38]. Ayrıca, üretilen bu tür yörüngelerin periyodunu önceden belirlemek olanaksızdır. Böyle bir belirsizlik altında EKS'yi güvenliğin üst düzeyde olması istenen kriptografik uygulamalarda kullanmak imkansızdır. Bu dezavantajlar dışında literatürde bahsedilmeyen başka dezavantajlar daha doktora çalışması sırasında fark edilmiş olup bunlardan az sonra bahsedilecektir.

Öncelikle, EKS'nin kayan noktalı aritmetik gerçekleştirimi kullanılarak 1000 adet 10^6 uzunluğunda rastgele bit dizisi üretilmiş ve NIST rastgelelik testi ile test edilmiştir [143]. Test için standart parametre değerleri kullanılmıştır. Tablo 4.1'deki sonuçlardan da görülebileceği üzere üretilen sözde rastgele bitler tüm testleri başarı ile geçmektedir. Fakat [38]'de gösterildiği üzere bu bit uzunlukları başka üreteçler için standart olarak

kabul edilse bile kaos tabanlı rastgele sayı üreteçlerinin dezavantajları daha sonraki bitlerde ortaya çıkabilmektedir.

Tablo 4.1 Kayan noktalı EKS için NIST rastgelelik testi sonuçları ($1000x10^6$ bit)

Test	Oran	P-değeri	Sonuç
Frekans	0.9889	0.633932	Geçti
Blok frekans	0.9919	0.234200	Geçti
Kümülatif toplamlar	0.9869	0.927386	Geçti
Kümülatif toplamlar	0.9858	0.401379	Geçti
Yinelemeler	0.9899	0.955190	Geçti
En uzun yinelemeler	0.9829	0.861526	Geçti
Rank	0.9889	0.617145	Geçti
FFT	0.9909	0.056412	Geçti
Örtüşmeyen şablon eşleştirme	0.9909	0.382443	Geçti
Örtüşen şablon eşleştirme	0.9940	0.062384	Geçti
Evrensel	0.9909	0.071947	Geçti
Yaklaşık entropi	0.9929	0.833418	Geçti
Rastgele sapmalar	0.9904	0.503583	Geçti
Değişken rastgele sapmalar	0.9904	0.598556	Geçti
Seri	0.9949	0.249414	Geçti
Seri	0.9929	0.755134	Geçti
Lineer komplekslik	0.9909	0.640307	Geçti

Dolayısıyla, Tablo 4.1'deki sonuçlar kaos tabanlı üreteçler için güvenilir değildir. Nitekim, aynı testler $5x10^7$ uzunluklu 100 tane bit dizisi için gerçekleştirildiğinde Tablo 4.2'deki gibi sonuçlar elde edilmiştir. Bu sonuçlardan görüldüğü üzere konvansiyonel kaos tabanlı üreteç altı tane testten kalmıştır.

Tablo 4.2 Kayan noktalı EKS için NIST rastgelelik testi sonuçları ($100x5x10^7$ bit)

Test	Oran	P-değeri	Sonuç
Frekans	0.9895	0.103890	Geçti
Blok frekans	0.9789	0.023368	Geçti
Kümülatif toplamlar	0.9894	0.167699	Geçti
Kümülatif toplamlar	1.0000	0.005062	Geçti
Yinelemeler	0.9468	0.602458	**Kaldı**
En uzun yinelemeler	0.9789	0.090936	Geçti
Rank	0.9895	0.918572	Geçti
FFT	M/D	M/D	**Kaldı**
Örtüşmeyen şablon eşleştirme	0.9895	0.972458	Geçti

Örtüşen şablon eşleştirme	0.7579	0.000000	**Kaldı**
Evrensel	0.9895	0.458035	Geçti
Yaklaşık entropi	0.9149	0.000017	**Kaldı**
Rastgele sapmalar	1.0000	0.579479	Geçti
Değişken rastgele sapmalar	0.9792	0.625552	Geçti
Seri	0.8925	0.479268	**Kaldı**
Seri	0.9043	0.087929	**Kaldı**
Lineer komplekslik	1.0000	0.324180	Geçti

Bu durumun nedenleri daha önceki bölümlerde bahsetmiş olduğumuz üzere aşağıdaki gibi özetlenebilir:

i. Dijital gerçekleştirimin neden olduğu degredasyon (dijital sistemlerde kaosun olmaması) [32].
ii. Transientten sonra periyodik bölgelerin aynı olması [38].
iii. Kısa periyotlar [31].

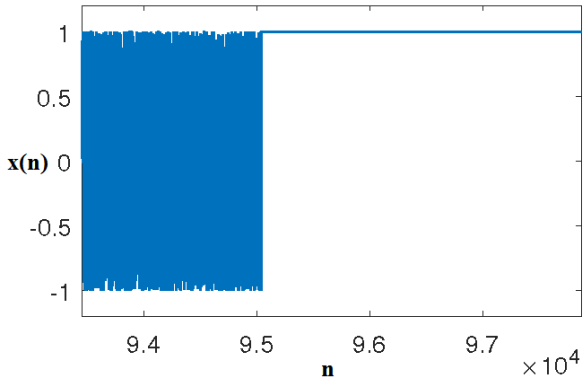
Doktora çalışmaları sırada EKS'de istenmeyen çıkışların meydana gelmesine neden olan iki yeni davranış daha gözlemlenmiştir. Bunlar ise aşağıdaki gibi sıralanmaktadır:

iv. Yuvarlama hataları nedeniyle yörüngenin sabit noktaya düşmesi.
v. Yuvarlama hataları nedeniyle yörüngenin tanım aralığı dışına çıkması.

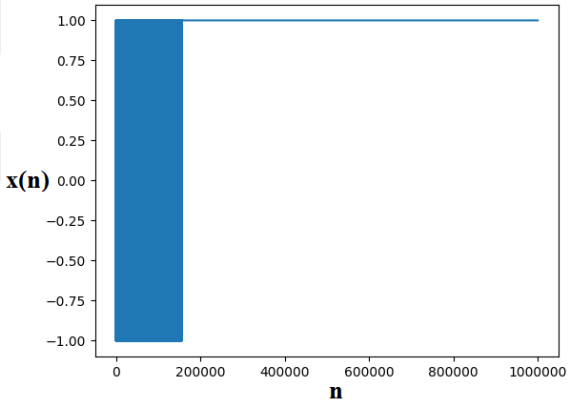
Sabit veya kayan noktalı aritmetikle yapılan gerçekleştirimde yörüngenin sabit noktaya düşmesi MATLAB ve Python ile yapılan gerçekleştirimlerde görülürken, birim aralığın dışına çıkma olayı C ile yapılan gerçekleştirimlerde gözlemlenmiştir. Bu ise farklı dillerle yapılan gerçekleştirimlerde bile farklı davranışların elde edilebileceğini göstermektedir. Bu ise kriptografik uygulamalarda kesinlikle istenmeyecek bir durumdur.

Eşitlik (4.2)'deki Chebyshev haritasının MATLAB gerçekleştiriminde 10^6 uzunluğundaki her 1000 yörüngeden ortalama 8 tanesinin bir sabit noktaya düştüğü; C gerçekleştirimlerinde ise ortalama 4 yörüngenin tanım aralığı dışına çıktığı belirlenmiştir. Bu ise Şekil 4.1'deki EKS'nin belli iterasyonlardan sonra sadece 1 veya 0'lardan oluşan bit blokları üretmesine neden olmaktadır. Dolayısıyla, bu yapıyı kullanacak olan kriptosistemlerde çok ciddi güvenlik açıklarının olacağı aşikardır. MATLAB gerçekleştiriminde sabit noktaya düşen bir yörünge örneği Şekil 4.2'de görülmektedir. Şekilden görüldüğü üzere yörünge 950000'ninci iterasyondan sonra sabit noktaya

düşmektedir. Olayın bu kadar uzun iterasyonlardan sonra meydana gelmesi Tablo 4.1'deki rastgelelik testinin neden bunu tespit edemediğini açıklamaktadır. Aynı durumun Python gerçekleştirimi örneği ise Şekil 4.3'deki gibidir.

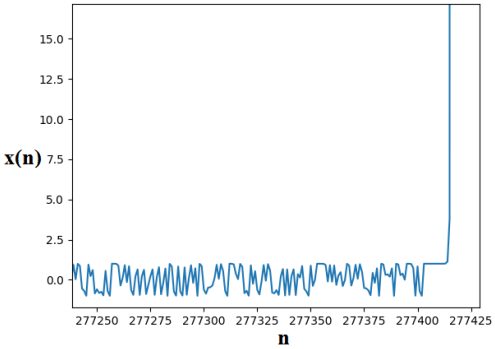


Şekil 4.2 Chebyshev haritasının MATLAB gerçekleştiriminin sabit noktaya düşmesi



Şekil 4.3 Chebyshev haritasının Python gerçekleştiriminin sabit noktaya düşmesi

C gerçekleştiriminde tanım aralığı dışına taşan bir yörüngenin örneği ise Şekil 4.4'de görülmektedir. Bu tür yörüngeler kısa sürede sonsuza gitmekte olduklarından yukarıdaki gibi güvenlik açığına neden olmak dışında çeşitli program hatalarına da neden olabilme ihtimalleri vardır.



Şekil 4.4 Chebyshev haritasının C gerçekleştiriminin tanım aralığı dışına taşması

4.2. LSB Uzatma Metodu ile Gerçek Periyodik Yörüngelerin Kullanılması

Bu kısımda ise yukarıdaki konvansiyonel metot yerine ikinci bölümde sunulan LSB uzatma metodu kullanılacaktır. Bu bağlamda, orijinal kaotik sisteme ait olmayan hatalı periyodik çıkışlar yerine orijinal kaotik sisteme ait olan gerçek periyodik yörüngeler kullanılacaktır. Bunun için yapılması gereken işlemler şu şekilde sıralanabilir:

i. İKKH gerçekleştirimi için kaydırma, terslendirme vb. temel lojik işlemlere dayalı bir algoritma kurulur (daha önce önerilen algoritmalardan biri seçilebilir) veya gerekli boyuta bağlı olarak ÇBBH kullanılır.
ii. Eğer üretilmek istenen kaotik sistem İKKH değilse, istenen kaotik sistem ve bir İKKH arasında topolojik konjuge veya yarı-konjuge fonksiyonu belirlenir.
iii. SRSÜ'nün ilk değerleri gizli anahtar olarak kullanılarak üretilen gerçek periyodik yörüngeler kaos tabanlı kriptosistemler içerisinde kullanılır.

Buna göre, Şekil 4.1'de verilen EKS yapısındaki kaotik haritalar kayan noktalı aritmetik yerine bu sefer LSB uzatma metodu ile gerçekleştirilmiştir. LSB uzatma metodunda 128 bitlik LFSR kullanılmış ve $L = 32$ olarak seçilmiştir. Eşitlik (4.2)'deki Chebyshev haritasının gerçek periyodik yörüngelerinin LSB uzatma metodu ile nasıl üretileceği ikinci bölümde anlatılmaktadır (g_2 haritası).

Bu yeni EKS yapısı için de standart 1000x10^6 bitlik NIST rastgelelik testleri yapılmıştır. Tablo 4.3'de görüldüğü üzere LSB uzatma metoduna dayalı EKS de tüm testleri geçmektedir. Fakat, önerilen yapının asıl üstün tarafı bit dizilerinin her birinin uzunluğu 5x10^7 bite çıkarıldığında görülmektedir. 100 adet 5x10^7 uzunluklu bit dizisi için yapılan NIST rastgelelik sonuçları Tablo 4.4'de verilmektedir. Görülebileceği üzere LSB uzatma metodu kullanıldığında sonuçlar belirgin şekilde iyileşmektedir. [37]'de belirtildiği üzere "Örtüşen Şablon Eşleştirme" testi bu bit uzunluğu için yanlış sonuçlar verdiğinden üreteç sadece "Örtüşmeyen Şablon Eşleştirme" testlerinin birkaçından kalmıştır. Örtüşmeyen şablon testlerinden birkaçından kalınmasının nedeni ise 100 olan bit dizisi adedinin test için az gelmesinden kaynaklanmış olabilir. Fakat, test için kullanılan bit sayısının halihazırda çok fazla olması bit dizlerinin sayısını daha da arttırmaya imkan vermemektedir. Sonuç olarak, tek bir tasarım metodu değişikliği ile sonuçların bu denli iyileşmesi önerilen metodun oldukça güçlü bir yanıdır.

Tablo 4.3 LSB uzatma metoduna dayalı EKS için NIST rastgelelik testi sonuçları ($1000x10^6$ bit)

Test	Oran	P-değeri	Sonuç
Frekans	0.9880	0.745908	Geçti
Blok frekans	0.9930	0.310049	Geçti
Kümülatif toplamlar	0.9880	0.302657	Geçti
Kümülatif toplamlar	0.9880	0.968128	Geçti
Yinelemeler	0.9920	0.981940	Geçti
En uzun yinelemeler	0.9870	0.291091	Geçti
Rank	0.9850	0.301194	Geçti
FFT	0.9880	0.589341	Geçti
Örtüşmeyen şablon eşleştirme	0.9860	0.073872	Geçti
Örtüşen şablon eşleştirme	0.9900	0.406499	Geçti
Evrensel	0.9930	0.259616	Geçti
Yaklaşık entropi	0.9830	0.072964	Geçti
Rastgele sapmalar	0.9876	0.115281	Geçti
Değişken rastgele sapmalar	0.9969	0.489204	Geçti
Seri	0.9890	0.877083	Geçti
Seri	0.9960	0.411840	Geçti
Lineer komplekslik	0.9950	0.304126	Geçti

Tablo 4.4 LSB uzatma metoduna dayalı EKS için NIST rastgelelik testi sonuçları ($100x5x10^7$ bit)

Test	Oran	P-değeri	Sonuç
Frekans	0.9900	0.883171	Geçti
Blok frekans	1.0000	0.334538	Geçti
Kümülatif toplamlar	1.0000	0.262249	Geçti
Kümülatif toplamlar	0.9900	0.304126	Geçti
Yinelemeler	0.9900	0.946308	Geçti
En uzun yinelemeler	0.9900	0.275709	Geçti
Rank	0.9900	0.137282	Geçti
FFT	0.9900	0.851383	Geçti
Örtüşmeyen şablon eşleştirme	0.9600	0.366918	Geçti*
Örtüşen şablon eşleştirme	0.9000	0.000000	**Kaldı**
Evrensel	0.9900	0.637119	Geçti
Yaklaşık entropi	0.9900	0.350485	Geçti
Rastgele sapmalar	0.9897	0.143505	Geçti
Değişken rastgele sapmalar	1.0000	0.772760	Geçti
Seri	0.9900	0.719747	Geçti
Seri	0.9900	0.996335	Geçti

Lineer komplekslik	0.9900	0.964295	Geçti

NIST rastgelelik testleriyle nicel olarak ispatlanmış güvenlik artışının dışında LSB uzatma metoduna dayalı kaotik harita gerçekleştirimini kullanmanın daha pek çok avantajı vardır. Öncelikle, artık orijinal matematiksel model tarafından tanımlı yörüngeler kullanılmaktadır. Davranışı hiçbir modelle net olarak belirlenemeyen, sürekli hesaplama hatalarına maruz kalan sözde yörüngeler kullanılmamaktadır. Sabit veya kayan noktalı aritmetikle üretilen sözde yörüngelerin aksine, bu metotla üretilen yörüngelerin periyodu net olarak bellidir. Ayrıca, sözde yörüngelerin periyotları kriptografik uygulamalar için kısa kalabilirken, LSB uzatma metodunda periyotlar istenilen uzunlukta seçilebilir.

Bunun dışında, sözde yörüngeler arasındaki kesişimler yüzünden konvansiyonel yöntemle üretilen yörüngelerden sadece birazı eşsiz olmaktadır; üretilen periyotların çoğu birbirinin aynısıdır [38]. Bunun EKS'de güvenlik açıklarına neden olduğu yine [38]'de gösterilmiştir. Ancak, LSB uzatma metodunda hesaplama hatalarına bağlı kesişimler olmadığından transientten sonra kesişen yörüngeler yoktur. Dolayısıyla, LSB uzatma metoduna dayalı EKS [38]'de gösterilen güvenlik açığından da muzdarip değildir.

Konvansiyonel gerçekleştirime dayalı kaotik haritaların sergilediği diğer istenmeyen davranışlar da burada yörüngelerin sabit noktaya düşmesi ve yörüngelerin tanım aralığı dışına çıkması şeklinde belirlenmişti. Yuvarlama hatalarının mevcut olmaması nedeniyle LSB uzatma metodu ile üretilmiş yörüngeler asla bu iki istenmeyen davranışı sergilemez. Yine bu istenmeyen davranışlar incelenirken farklı programlama dilleri ile yapılan gerçekleştirimlerde bu hataların farklılık gösterdiği de görülmüştür. Bu da sabit veya kayan noktalı aritmetiğin nasıl uygulandığına göre kaos tabanlı kriptosistemlerin farklı davranışlar sergileyebileceği anlamına gelmektedir. Yani, sabit veya kayan noktalı aritmetiğe dayalı bir kaos tabanlı kriptosistem ile bir makinede yapılan şifreleme, bir başka makinede deşifrelenemeyebilir. Konvansiyonel dijital gerçekleştirim ile LSB uzatma metoduna dayalı gerçekleştirimler için bu yapmış olduğumuz kıyaslamalar Tablo 4.5'de özetlenmektedir. Tablo 4.5'den görülebileceği üzere LSB uzatma metodunun kaos tabanlı kriptosistemlerde kullanımı pek çok probleme çözüm getirmektedir.

Tablo 4.5 Sabit / Kayan noktalı gerçekleştirim ile LSB uzatma metoduna dayalı gerçekleştirimin kıyaslanması

Sabit / Kayan Noktalı Gerçekleştirim	**LSB Uzatma Metoduna Dayalı Gerçekleştirim**
- Üretilen yörüngeler orijinal sisteme ait değildir.	- Üretilen yörüngeler orijinal sistemin gerçek periyodik yörüngeleridir.
- Çıkışın periyodik olmasına rağmen çıkışın periyodunu önceden bilmek mümkün değildir.	- Çıkışın periyodu net olarak bellidir.
- Transientten sonra kısa periyotlar vardır.	- Periyotlar istenilen uzunlukta seçilebilir.
- Transient kısmından sonra kesişimler yüzünden sadece sınırlı sayıda farklı periyodik yörünge mevcuttur.	- Yörüngelerin transient uzunluğu hafıza birimi uzunluğu kadardır; yakın yörüngeler birbirleriyle kesişmez.
- Yuvarlama hataları nedeniyle yörünge sabit noktaya düşebilir.	- Yuvarlama hatası yoktur; istenmedikçe yörüngeler sabit noktaya düşmez.
- Yuvarlama hataları nedeniyle yörünge tanım aralığı dışına çıkabilir.	- Yuvarlama hatası yoktur; ayrıca yörüngeler gerçek yörünge olduğundan kesinlikle tanım aralığı dışına çıkamaz.
- Farklı programlama dilleri (MATLAB, Python, C) ile farklı davranışlar gözlemlenebilmektedir. Bu ise aynı başlangıç şartları için farklı işaretler üretilebileceği anlamına gelir.	- İKKH algoritmalarının tamamen lojik işlemlere bağlı olmasından dolayı gerçekleştirim programlama dilinden bağımsızdır. Sadece konjuge fonksiyonun gerçekleştirimi farklılık yaratabilir.

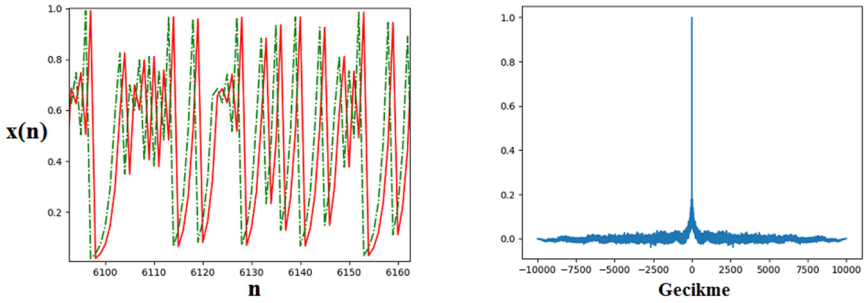
Tabi sabit veya konvansiyonel aritmetikle yapılan gerçekleştirimlere kıyasla, LSB uzatma metodunun da bazı dezavantajları vardır. Öncelikle, İKKH'ların parametre değerlerinin ikinin katı olması gerekliliği ve konjuge fonksiyonlarının sabit parametre değerleri için kurulabilmesinden dolayı LSB uzatma metoduna dayalı kriptosistemlerde parametre değeri anahtar olarak kullanılmaz. Fakat, parametre değerlerinin anahtar olduğu kaos tabanlı kriptosistemler kolayca kırıldığı için zaten önerilmemektedir [88]. İkinci dezavantaj olarak da İKKH olmayan her kaotik sistem için bir konjuge fonksiyonu

bulmanın mümkün olmaması gösterilebilir. Farklı haritalar arasında topolojik konjuge fonksiyonu bulmak çok da kolay değildir.

Her ne kadar bu kısımda LSB uzatma metodunun konvansiyonel sabit / kayan noktalı aritmetiğe göre avantajlarının çok fazla olduğu gösterilmiş olsa da EKS'nin LSB uzatma metodu ile bu şekilde kullanımı yine de kriptrografik olarak güvenli değildir. LSB uzatma metoduna dayalı kaotik haritaların kaos tabanlı kriptosistemlerde kullanılması için biraz daha farklı yaklaşımlara ihtiyaç vardır. Bunun nedeni ve kullanılacak olan yeni tasarım yaklaşımı sonraki kısmın konusudur.

4.3. Gerçek Periyodik Yörüngelerin Kullanılması İçin Yeni Bir Tasarım Yaklaşımı

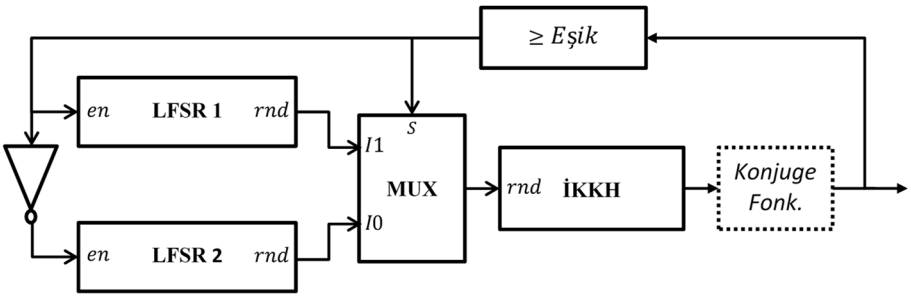
Bir önceki kısımda kaos tabanlı bir kriptosistemde bulunan kaotik harita doğrudan LSB uzatma metoduna göre gerçekleştirilerek kullanılmıştı. Bu şekilde kaotik haritanın gerçek periyodik yörüngeleri kullanılmıştı. Böyle doğrudan bir kullanım pek çok sorunu çözse de kriptografik olarak güvenli değildir. Bunun nedenini görebilmek için benzer başlangıç şartları için LSB uzatma metoduna göre gerçekleştirilmiş iki kaotik haritanın yörüngelerini inceleyelim. Tek bir LFSR ile gerçekleştirilebildiğinden Eşitlik (1.2) ile verilen çadır haritasını göz önünde bulunduralım. LFSR'nin başlangıç durumunun sadece birinci biti bir (diğerleri sıfır) yapıldığında üretilen gerçek periyodik yörünge ile LFSR'nin sadece ikinci biti bir yapıldığında elde edilen yörünge Şekil 4.5 (a)'da görülmektedir. Görüldüğü üzere elde edilen yörüngeler iterasyon ne kadar artarsa artsın birbirlerinin bir birim kaydırılmış versiyonlarıdır. Dolayısıyla, bu iki yörünge arasındaki korelasyon Şekil 4.5 (b)'de görüldüğü gibi yüksek olmaktadır. LFSR'nin çalışma prensibi göz önüne alındığında bu durum normaldir. Fakat, kaos tabanlı kriptosistemlerde, klasik kriptografinin yayılma gereksinimi b.ş.h.b. özelliği ile sağlanmakta olup böyle bir davranışın olmaması beklenir.



(a) (b)

Şekil 4.5 LFSR'nin yakın başlangıç durumları için elde edilen: (a) çadır haritası yörüngeleri; (b) yörüngelerin korelasyonu

Bu tür bir davranışın gözlemlenmesinin nedeni LFSR'nin başlangıç durumu ile kaotik haritanın başlangıç şartının aynı şey olmamasıdır. İkinci bölümden hatırlanacağı üzere LSB uzatma metodu ile üretilen yörüngenin başlangıç şartını istenildiği gibi ayarlamak olanaksızdır. Başlangıç şartı üretilen rastgele bitlere bağlı olarak oluşmaktadır. Dolayısıyla, yakın anahtar değerleri için farklı çıkışların oluşmasını sağlayarak klasik kriptografinin bir gereksinimi olan yayılma şartını sağlayan yeni bir tasarıma ihtiyaç vardır. Gerçek periyodik yörüngelerin kaos tabanlı kriptografide kullanılması için geliştirilmiş bu tür yeni bir tasarım Şekil 4.6'da verilmektedir.



Şekil 4.6 Önerilen yeni tasarım yaklaşımı (M-İKKH yapısı)

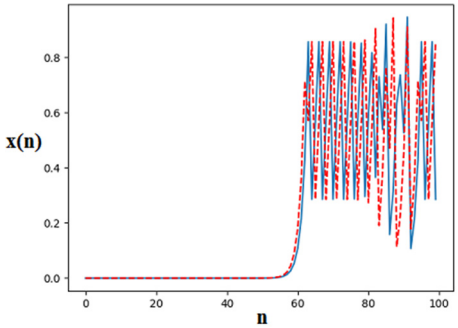
Önerilen tasarım yaklaşımımda İKKH artık tek başına kullanılmamaktadır; Şekil 4.6'da görüleceği üzere diğer pek çok lojik blokla birlikte kullanılmakta olup bu yeni yapı "Modifiye İKKH yapısı" ya da kısaca "M-İKKH yapısı" olarak adlandırılmaktadır. M-İKKH yapısındaki İKKH'nın çıkışı opsiyonel olarak bir konjuge fonksiyona bağlanmaktadır. İkinci bölümden hatırlanacağı üzere topolojik konjuge fonksiyonları İKKH olmayan diğer kaotik haritaların yörüngelerini elde etmek için kullanılmaktadır. Eğer bir İKKH doğrudan kullanılacaksa bu bloğa ihtiyaç yoktur. Sonraki "Eşik" bloğu giriş değerine göre lojik 1 veya 0 çıkışı üreten bir bloktur. $[a, b]$ aralığında tanımlı bir

haritanın yörünge değeri x kabul edildiğinde, tipik bir eşik fonksiyonu $c \in [a, b]$ eşik değeri olmak üzere Eşitlik (4.3)'teki gibi davranmalıdır.

$$f(x) = \begin{cases} 1, & x \geq c \\ 0, & x < c \end{cases} \tag{4.3}$$

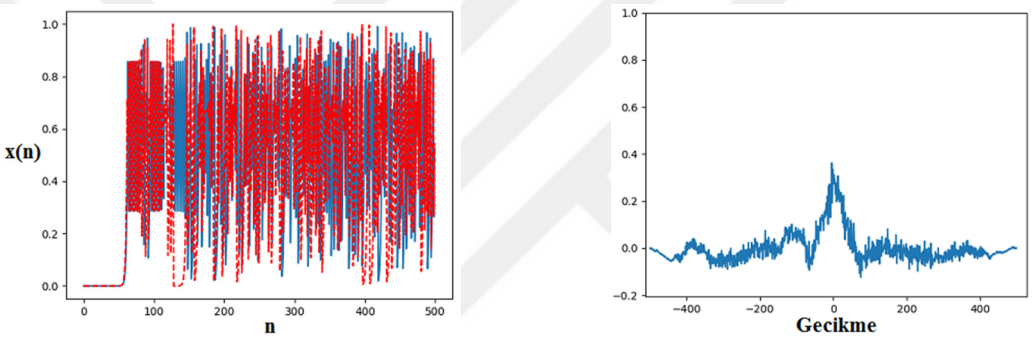
Görüldüğü üzere eşik fonksiyonunun çıkışı "MUX" ile gösterilen bir çoklayıcının (multiplexer) kontrol bitine bağlanmıştır. Bu çoklayıcı eğer S biti 0 ise çıkışa $I0$ girişini; eğer S biti 1 ise çıkışa $I1$ girişini vermektedir. Aynı zamanda, bu eşik fonksiyonu çıkışı LFSR'lerin "*en*" ile gösterilen aktifleştirme uçlarına bağlanmaktadır. Yalnız, ikinci LFSR'ye bağlanırken bu işaretin değili alınmaktadır. Böylece, eşik fonksiyonunun değeri 1 iken "LFSR 1" aktifleştirilmekte ve bu LFSR'nin rastgele bit çıkışı İKKH'yı beslemek üzere çoklayıcı tarafından seçilmektedir. Bu durumda, "LFSR 2" aktif olmayacağı için çalışmayacak ve önceki durumunu aynen muhafaza edecektir. Eşik fonksiyonu 0 çıkışını verdiğindeyse bunun tam tersi olacak ve rastgele bit bu sefer "LFSR 2"den alınacaktır. M-İKKH yapısı ile normalde kullanılması gereken LFSR sayısı iki katına çıksa da yayılma şartını sağlayabilmek için bu gereklidir.

Şimdi yine çadır haritası kullanarak Şekil 4.6'deki M-İKKH yapısını Python ile kuralım. Konjuge fonksiyon olmaksızın $L = 32$ ve $c = 0.25$ olsun. M-İKKH yapısı için LFSR'lerin her ikisinin başlangıç durumunda sadece birinci biti 1 yapıp diğerlerini sıfır yapalım. Bu durumda, hafiza biriminin değeri en başta 0 yapıldığından ilk iterasyonda sadece LFSR 2 kaydırılacaktır. Buna göre, LFSR 1'in yine ilk bitini 1 yapıp LFSR 2'nin bu sefer ikinci bitini 1 yaparsak İKKH'yı doğrudan kullandığımız duruma benzer anahtar değerleri elde etmiş oluruz. Bu anahtar değerleri için iki farklı yörünge üretip ilk 100 iterasyonu incelediğimizde Şekil 4.7'deki yörüngeleri elde ederiz.



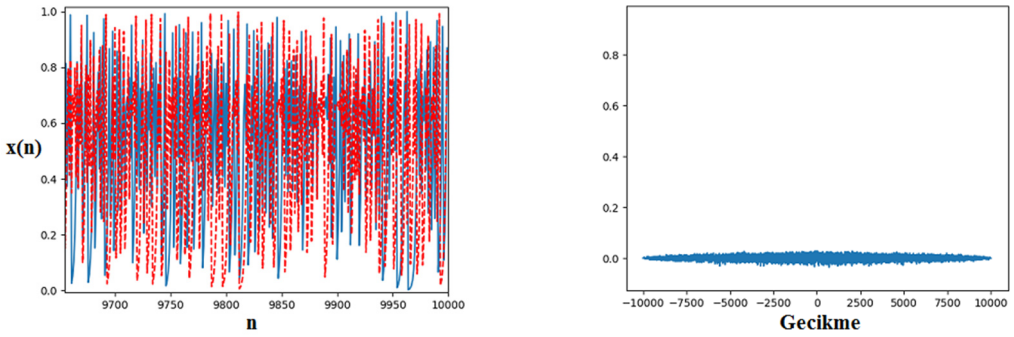
Şekil 4.7 M-İKKH ile yakın anahtar değerleri için üretilmiş iki yörüngenin ilk 100 iterasyonu

Şekilden görülebileceği üzere hafıza birimi rastgele bitlerle dolana kadar sıfıra yakın bir sonuç elde edilmektedir. Sonrasında ise beklendiği üzere aralarında 1 iterasyonluk gecikme olan iki yörünge tıpkı İKKH'nın doğrudan kullanımında olduğu gibi elde edilmektedir. Fakat, kısa bir süre sonra bu durum değişmekte ve yörüngeler farklılaşmaya başlamaktadır. İlk 500 iterasyon için elde edilen yörüngeler ve korelasyon ilişkisi ise Şekil 4.8'de verilmektedir. Görülebileceği üzere baştaki yakın ilişki yüzünden yörüngeler arası korelasyon yüksek çıkmaktadır. Fakat, ilk 10000 iterasyon incelendiğinde Şekil 4.9'daki yörüngeler ve korelasyon elde edilmektedir. Buradan görülebileceği üzere, uzun iterasyonlar sonucunda anahtar değerleri yakın olan iki yörünge birbirinden tamamen alakasız hal almış ve buna bağlı olarak korelasyon ilişkisi sıfır seviyesine düşmüştür.



(a) (b)

Şekil 4.8 M-İKKH ile yakın anahtar değerleri için üretilmiş iki yörüngeye ait: (a) ilk 500 iterasyon; (b) ilk 500 iterasyonun korelasyon ilişkisi

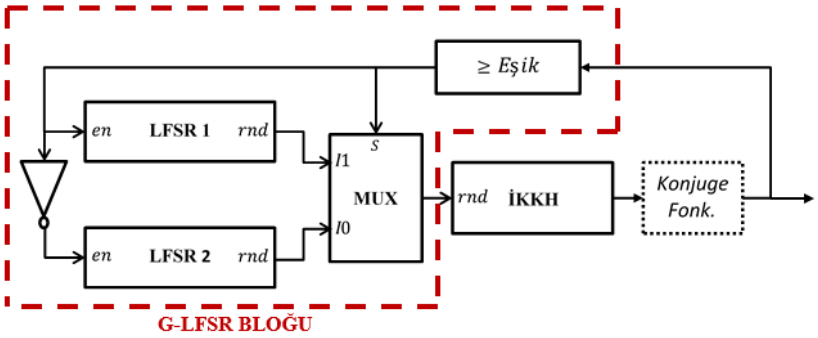


(a) (b)

Şekil 4.9 M-İKKH ile yakın anahtar değerleri için üretilmiş iki yörüngeye ait: (a) 10000 iterasyonun son kısmı; (b) ilk 10000 iterasyonun korelasyon ilişkisi

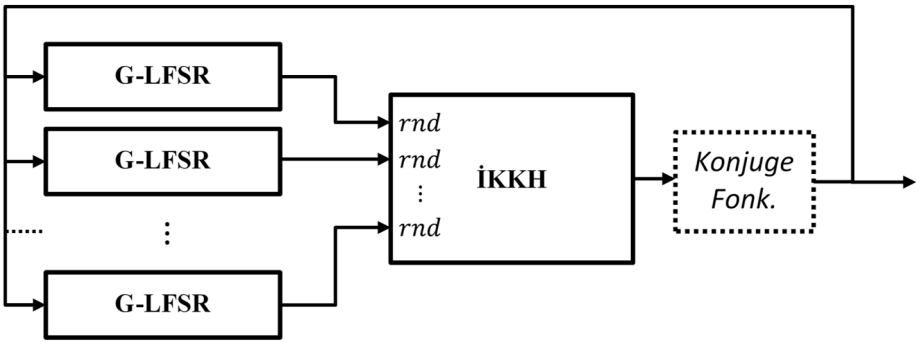
Birden fazla rastgele bitin gerektiği (Eşitlik (4.2)'deki Chebyshev haritası gibi) durumlar için Şekil 4.6'daki İKKH ve konjuge fonksiyon dışındaki tüm bloklar yinelenmelidir. Bu blok yapıları "genelleştirilmiş LFSR bloğu" ya da kısaca G-LFSR bloğu olarak adlandırılacaktır. G-LFSR bloğu Şekil 4.10'da kesikli çizgiler arasında kalan kısımdır.

Dolayısıyla, bu bloğun girişi eşik fonksiyonu; çıkışı ise çoklayıcı (MUX) tarafından verilen rastgele bittir.



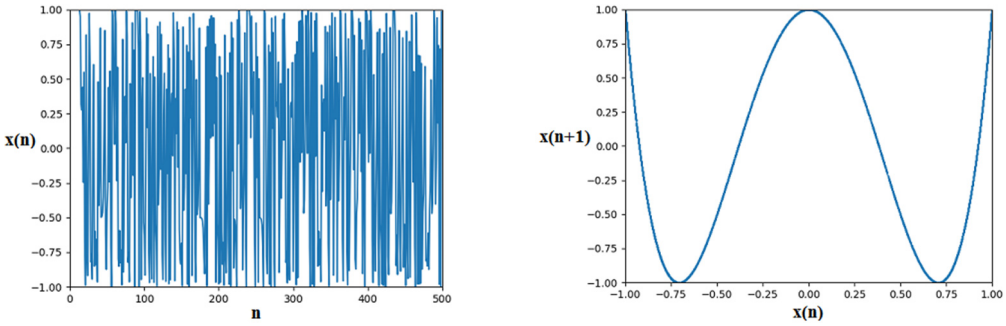
Şekil 4.10 M-İKKH yapısı içindeki G-LFSR bloğu

Buna göre, birden fazla rastgele bitin kullanıldığı M-İKKH yapısı Şekil 4.11'deki gibi olmaktadır. Bu yapıda her G-LFSR bloğunun farklı c eşik değerlerine sahip olması, farklı bloklar arasında farklı bit dizileri üretileceğini garanti altına alır. Eğer İKKH çok boyutluysa G-LFSR'lerin girişleri (yani eşik fonksiyonunun girişleri) herhangi bir boyuttan alınabilir.



Şekil 4.11 Birden fazla rastgele bitin kullanıldığı M-İKKH yapısı

Şimdi Şekil 4.11'deki yapı kullanılarak iki rastgele bitin gerektiği Eşitlik (4.2)'deki Chebyshev fonksiyonu M-İKKH tasarımı ile gerçekleştirilebilir. Bu durumda iki adet G-LFSR bloğu kullanmamız gerekmektedir. Ayrıca, ikinci bölümden hatırlanacağı üzere, bu harita için İKKH olarak Bernoulli haritası ve konjuge fonksiyonu olarak $h(x) = cos(2\pi x)$ kullanılmalıdır. Gerçekleştirim sırasında LFSR'ler 128 bit ve $L = 32$ olarak seçilmiştir. Bunun yanı sıra, G-LFSR'lerin eşik değerleri sırasıyla $c_1 = 0$ ve $c_2 = 0.5$ olarak seçilmiştir. Gerçekleştirim için Python dili kullanılmıştır. Üretilen gerçek periyodik yörünge ve yörüngeye ait çeker yapısı Şekil 4.12'de görüldüğü gibi elde edilmiştir. Çeker yapısı beklendiği gibidir.



(a) (b)

Şekil 4.12 M-İKKH ile yapılan Chebyshev gerçekleştiriminin: (a) bir yörüngesi; (b) çeker yapısı

Daha önce İKKH'ların doğrudan gerçekleştirimi ile Chebyshev haritasının kullanıldığı EKS kriptosisteminin rastgelelik testlerini vermiştik. Şimdi Şekil 4.1'deki EKS yapısını M-İKKH ile gerçekleştirilmiş Chebyshev haritaları yardımıyla kuracağız. Bunun için yapılması gereken tek şey yukarıda kurmuş olduğumuz M-İKKH yapılarını EKS içerisinde kaotik harita olarak kullanmaktır. Buna göre, M-İKKH tabanlı Chebyshev haritalarının kullanıldığı EKS kriptosistemi yine Python ile gerçekleştirilmiştir. Bu şekilde üretilen bit dizilerinin de NIST rastgelelik testleri yapılmıştır. 1000 adet 10^6 bit uzunluklu bit dizisine ait test sonuçları Tablo 4.6'daki gibidir. Bu standart değerler için M-İKKH tabanlı EKS tüm testlerden geçmiştir.

Tablo 4.6 M-İKKH bloklarına dayalı EKS için NIST rastgelelik testi sonuçları ($1000x10^6$ bit)

Test	Oran	P-değeri	Sonuç
Frekans	0.9930	0.790621	Geçti
Blok frekans	0.9890	0.048716	Geçti
Kümülatif toplamlar	0.9930	0.595549	Geçti
Kümülatif toplamlar	0.9920	0.088226	Geçti
Yinelemeler	0.9880	0.601776	Geçti
En uzun yinelemeler	0.9900	0.786830	Geçti
Rank	0.9910	0.205531	Geçti
FFT	0.9830	0.308561	Geçti
Örtüşmeyen şablon eşleştirme	0.9930	0.274341	Geçti
Örtüşen şablon eşleştirme	0.9900	0.424453	Geçti
Evrensel	0.9920	0.191687	Geçti
Yaklaşık entropi	0.9880	0.165340	Geçti
Rastgele sapmalar	0.9869	0.122910	Geçti
Değişken rastgele sapmalar	0.9885	0.910330	Geçti
Seri	0.9890	0.994403	Geçti

Seri	0.9900	0.136553	Geçti
Lineer komplekslik	0.9830	0.901166	Geçti

Daha önce bahsetmiş olduğumuz üzere, kaos tabanlı kriptografide dezavantajlar uzun iterasyonlar sonucu ortaya çıkabilmektedir. Bu nedenle, tıpkı İKKH'ların doğrudan kullanıldığı durumda yapmış olduğumuz gibi, uzun bit dizileri için de testler yapılmalıdır. Bu bağlamda, $5x10^7$ bit uzunluğunda 100 adet dizi için testler tekrarlanmıştır. Test sonuçları Tablo 4.7'de verilmektedir. Hatırlanacağı üzere, bu değerler için örtüşen şablon eşleştirme hatalı sonuçlar vermektedir [37]. Dolayısıyla, bu test dışında, gerçekleştirilen EKS yapısı daha önceki İKKH'nın doğrudan kullanıldığı gerçekleştirimde olduğu gibi sadece birkaç örtüşmeyen şablon testinden kalmıştır. Buna göre, tıpkı İKKH'nın doğrudan kullanıldığı durumda olduğu gibi, klasik kayan noktalı metoda göre çok daha iyi bir performans elde edilmiştir. Farklı olarak, M-İKKH yapıları sayesinde bu EKS tasarımı yakın anahtar değerleri için korelasyona sahip olmayan bit dizileri üretmektedir. Bu nedenle de İKKH'nın doğrudan kullanıldığı metodun aksine M-İKKH tabanlı EKS kriptografik uygulamalarda kullanılabilir.

Tablo 4.7 M-İKKH bloklarına dayalı EKS için NIST rastgelelik testi sonuçları ($100x5x10^7$ bit)

Test	**Oran**	**P-değeri**	**Sonuç**
Frekans	1.0000	0.779188	Geçti
Blok frekans	1.0000	0.883171	Geçti
Kümülatif toplamlar	1.0000	0.637119	Geçti
Kümülatif toplamlar	1.0000	0.739918	Geçti
Yinelemeler	0.9900	0.699313	Geçti
En uzun yinelemeler	0.9700	0.009535	Geçti
Rank	0.9900	0.955835	Geçti
FFT	0.9800	0.045675	Geçti
Örtüşmeyen şablon eşleştirme	1.0000	0.262249	Geçti*
Örtüşen şablon eşleştirme	0.8800	0.000000	**Kaldı**
Evrensel	0.9800	0.040108	Geçti
Yaklaşık entropi	0.9800	0.759756	Geçti
Rastgele sapmalar	0.9789	0.341565	Geçti
Değişken rastgele sapmalar	1.0000	0.260784	Geçti
Seri	1.0000	0.289667	Geçti
Seri	1.0000	0.066882	Geçti
Lineer komplekslik	0.9900	0.137282	Geçti

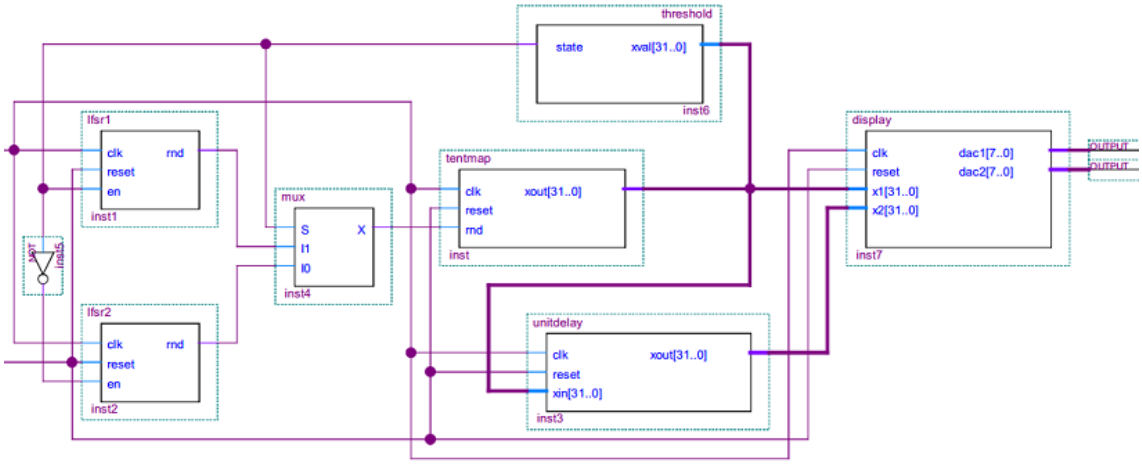
4.4. Önerilen Tasarımın Donanım Gerçekleştirimleri

Bu kısımda önce bir M-İKKH yapısının FPGA üzerinde donanım gerçekleştirimi yapılacaktır. Üretilen yörünge ve çeker yapısı osiloskop üzerinde gözlemlenecektir. Ardından ise, bu M-İKKH yapısı önceki kısımda olduğu gibi EKS yapısı içerisinde kullanılacak ve böylelikle M-İKKH tabanlı EKS kriptosisteminin de donanım gerçekleştirimi yapılmış olacaktır. Daha önceki bölümlerde olduğu gibi gerçekleştirim esnasında Altera DE0 geliştirme bordu ve üzerindeki Cyclone III FPGA çipi kullanılacaktır. Donanım tanımlamaları VHDL dili ile yapılmaktadır.

İkinci bölümden hatırlanacağı üzere LSB uzatma metoduna dayalı gerçekleştirimlerde en çok donanım kaynağı tüketimi konjuge fonksiyonların gerçekleştiriminde yaşanmaktadır. Tablo 2.2'den görülebileceği üzere İKKH'lar için gerçekleştirim %1-%2 kaynak tüketimine neden olurken konjuge fonksiyon gerektiren gerçekleştirimlerde bu oran %16-%17 seviyelerine çıkmaktadır. Ayrıca, konjuge fonksiyon gerçekleştirimi sırasında kullanılan sabit / kayan noktalı aritmetik, daha önce eleştirmiş olduğumuz üzere klasik kriptografide tercih edilmemektedir. Bu bağlamda, her ne kadar kullanımı konvansiyonel kaos tabanlı kriptosistem gerçekleştirimlerindeki gibi ciddi sorunlara neden olmasa da konjuge fonksiyonun M-İKKH yapısı içerisinde kullanılması pek de avantajlı görünmemektedir. Bu yüzden, buradaki donanım gerçekleştirimleri sırasında Şekil 4.6'daki konjuge bloğu kullanılmayacaktır. Böylelikle, gerçekleştirim tıpkı klasik kriptografide tercih edildiği üzere basit bitsel işlemlere dayalı olacak ve kaynak tüketimi minimuma indirgenecektir.

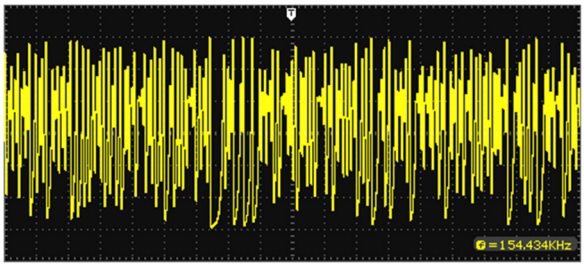
Donanım gerçekleştiriminde dikkat edilecek bir diğer husus ise eşik fonksiyonunun gerçekleştirilmesidir. Eğer $n < L$ bir doğal sayı olmak üzere eşik değerleri $1/2^n$'in pozitif lineer kombinasyonlarından seçilirse, büyüklük kıyaslaması sadece belli bitlerin kontrol edilmesiyle sağlanabilir. Örneğin, LSB uzatma metodunda $L = 32$ uzunluklu bir hafıza birimi kullanılsın. Eğer eşik değeri $c = 0.5 = 1/2$ olursa, x yörünge değeri olmak üzere $x \geq 0.5$ şartı ancak ve ancak hafıza biriminin ilk (MSB) biti 1 olursa sağlanacaktır. Yani, eşik fonksiyonunu gerçekleştirmek için ilk biti kontrol etmek yeterlidir. Benzer şekilde, eğer $c = 0.75 = 1/2^1 + 1/2^2$ olursa bu durumda hafıza biriminin hem birinci hem de ikinci biti kontrol edilmelidir.

Buna göre, İKKH olarak çadır haritasının kullanıldığı bir M-İKKH tasarımı Şekil 4.6'ya uygun olarak FPGA donanımı üzerinde kurulmuştur. Bahsettiğimiz üzere şekildeki konjuge bloğu kullanılmamıştır. Gerçekleştirimde her iki LFSR'nin uzunluğu 128 bit; $L = 32$ ve $c = 0.25$ olarak seçilmiştir. İKKH gerçekleştirimi Algoritma 2.2'ye uygun olarak yapılmıştır. VHDL ile oluşturulan FPGA şematiği Şekil 4.13'deki gibidir. Burada İKKH gerçekleştirimi "tentmap" bloğuna; eşik fonksiyonu gerçekleştirimi ise "threshold" bloğuna karşılık gelmektedir. Kullanılan "unitdelay" bloğu İKKH çıkışını bir saat darbesi kadar geciktirmek için kullanılmıştır. Bu sayede osiloskop ile çeker yapısının gözlemlenmesi mümkün olmuştur. "display" bloğu ise 32 bitlik İKKH çıkışı ve bunun geciktirilmiş halini 8 bite dönüştürüp D/A dönüştürücülere göndermek için kullanılmaktadır. Şematikte sistem saati "clk" ve "reset" girişleri boşta görünse de bu şeklin sayfaya sığması için böyle verilmiştir. Normalde bu girişler PLL'e ve bir giriş portuna bağlıdır. Bunlar dışında şematik yapısı Şekil 4.6'daki konfigürasyonla konjuge fonksiyonu bloğu hariç birebir aynıdır.

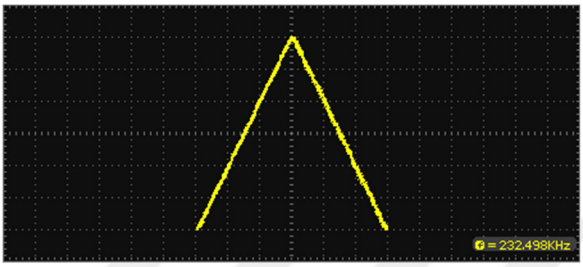


Şekil 4.13 Çadır haritasına ait M-İKKH yapısının FPGA gerçekleştirim şematiği

FPGA gerçekleştirimi neticesinde D/A dönüştürücülerin çıkışları osiloskop ile gözlemlendiğinde Şekil 4.14'deki görüntüler elde edilmiştir. Şekil 4.14 (a)'da İKKH çıkışının ilk 8 bitlik kısmının analog eşdeğeri görülmektedir. Bu tipik bir gerçek periyodik yörüngedir. Şekil 4.14 (b)'deki çeker yapısı ise ikinci bölümden hatırlanacağı üzere bu gerçek periyodik yörüngenin beklendiği üzere çadır haritasına ait olduğunu göstermektedir.



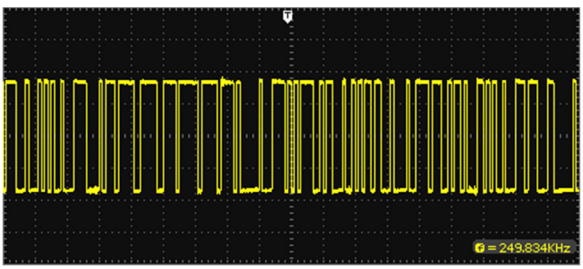
(a)



(b)

Şekil 4.14 Çadır haritasına ait M-İKKH yapısının FPGA gerçekleştiriminin osiloskop görüntüleri: (a) gerçek periyodik yörünge; (b) çeker yapısı

Son olarak ise bu M-İKKH yapıları EKS içindeki kaotik haritalar yerine kullanılarak EKS'nin bir FPGA gerçekleştirimi yapılacaktır. Bunun için yapılması gerekenler Şekil 4.1'deki kaotik haritalar yerine Şekil 4.13'deki gerçekleştirimin kullanılması ve yine Şekil 4.1'deki karşılaştırma fonksiyonunun VHDL ile ifade edilmesinden ibarettir. Karşılaştırma fonksiyonu g'yi VHDL ile gerçekleştirmek oldukça kolaydır. Bunun için tek yapılması gereken hafıza birimi değerlerini işaretsiz tam sayılar olarak kullanmaktır. Böylelikle iki farklı yörünge değeri tam sayılar olarak büyüklük açısından kıyaslanabilir. EKS gerçekleştirimi sonrası elde edilen rastgele bitler doğrudan DE0 bordu üzerindeki giriş / çıkış pinlerinden biri üzerinden alınarak osiloskop üzerinde gözlemlenmiştir. Elde edilen rastgele bitlerin osiloskop görüntüsü Şekil 4.15'de verilmektedir.



Şekil 4.15 M-İKKH tabanlı EKS'nin FPGA gerçekleştirimi çıkışının osiloskop görüntüsü

Gerçekleştirimde yine $L = 32$ kullanılmıştır. Eşik değerleri sırasıyla $c_1 = 0.25$ ve $c_2 = 0.75$ olarak seçilmiştir. Eşik değerinin gerçekleştiriminin nasıl yapılacağı daha önce

anlatılmıştı. Buna göre, ikinci eşik fonksiyonu bloğunda yapılan tek şey hafıza birimi değerlerinin 1'nci ve 2'nci bitlerini "ve kapısı" ile çıkışa vermektir. FPGA gerçekleştirim sonuçları Tablo 4.8'de verilmektedir. Beklenildiği üzere alan kullanımı oldukça düşüktür. Maksimum çalışma frekansları epey yüksek olsa da gözlemleme sırasında 1 MHz'lik bir sistem saati kullanılmıştır. Elde edilen maksimum çalışma frekansına göre M-İKKH tabanlı EKS'nin maksimum veri hızı 292.40 Mbps'dir.

Tablo 4.8 FPGA gerçekleştirim sonuçları

Model	Alan Kullanımı (Lojik Blok)	Maks. Çalışma Frekansı
M-İKKH	319 (% 2)	318.78 MHz
M-İKKH tabanlı EKS	629 (% 4)	292.40 MHz

5. BÖLÜM

TARTIŞMA, SONUÇ VE ÖNERİLER

Bu tez çalışmasında kaos tabanlı kriptosistemlerin dijital olarak gerçekleştirilebilmesi için yeni tasarım yaklaşımları önerilmiştir. Bu tasarım yaklaşımları sayesinde, normalde dijital olarak üretilemeyen gerçek kaotik yörüngelerin ve bir kaotik sistemin tanım kümesine yoğun bir şekilde dağılmış olan gerçek periyodik yörüngelerin dijital olarak üretilebilmesi mümkün olmuştur. Bunun yanı sıra, üretilen gerçek periyodik yörüngelerin literatürdeki mevcut kaos tabanlı kriptosistemlerde kullanılabilmesi, yine bu tez çalışmasında önerilen tasarımlarla sağlanmıştır.

Gerçek kaotik ve gerçek periyodik yörüngelerin dijital olarak üretilmesi ikinci bölümde önerilen ve LSB uzatma metodu olarak adlandırılan yöntem sayesinde olmuştur. LSB uzatma metodu, İKKH olarak adlandırılan kaotik haritalara dayanmaktadır. Bu tür kaotik haritalar, parametre değerleri ikinin katları olan ve basit lojik işlemlerle gerçekleştirilebilen literatürde mevcut haritalardır. Bu tür haritaların konvansiyonel sabit veya kayan noktalı gerçekleştirimleri daima sıfıra giderken, LSB uzatma metodu bu tür haritaların dijital gerçekleştirimine imkan tanımaktadır. Elbette, İKKH olarak sınıflandırılabilecek kaotik haritalar, literatürdeki haritaların sadece küçük bir kısmıdır. Bu dezavantajı gidermek adına, yine ikinci bölümde İKKH olmayan başka haritaların kaotik yörüngelerinin de LSB uzatma metoduyla üretilebilmesi için bir yöntem önerilmiştir. Bu yöntemde, gerçek yörüngelerinin üretilmesi istenen kaotik harita ile bir İKKH arasında topolojik konjuge ilişkisi kurulmaktadır. Sonrasında, elde edilen konjuge fonksiyon ile İKKH yörüngelerinden diğer haritanın yörüngeleri üretilmektedir. Örnek olarak, ikinci bölümde İKKH olmayan lojistik haritanın gerçek yörüngeleri üretilmiştir. Bildiğimiz kadarıyla, literatürde en çok incelenen kaotik haritalardan biri olmasına karşın bu haritanın gerçek yörüngelerinin tamamen dijital olarak elde edilmesi literatürde bir ilktir.

Bu bölümde yapılan gerçek yörünge üretimi kaos tabanlı kriptografi açısından son derece önemlidir. Kaos tabanlı kriptografinin dijital uygulamalarında yaygın olarak sabit veya kayan noktalı aritmetik kullanılmaktadır. Bu şekilde üretilen yörüngeler ise sonlu çözünürlük ve yuvarlama hataları yüzünden hem periyodik hem de hatalar yüzünden orijinal matematiksel modelle alakasız yörüngelerdir. Maalesef, bu konu hakkında çalışan araştırmacıların çoğu bu şekilde üretilmiş olan yörüngelerin kaotik olduğunu varsaymaktadır. Oysa ki bu tür yörüngeler kısa periyotlar bir kriptosistemde kullanıldıklarında, periyotlar arasındaki kesişimler, beklenmedik davranışlar gibi nedenlerden dolayı ciddi güvenlik açıklarına neden olmaktadır. Özellikle üretilen periyotlar arasındaki kesişimler tüm olası başlangıç şartları için sadece sınırlı sayıda eşsiz yörüngenin üretilmesine neden olmaktadır. Üretilen yörüngelerin çoğunun periyodik kısmı birbirinin aynı olmaktadır. Doğal olarak, böyle yörüngeler kriptografik açıdan kullanılmak için uygun değildir ve kaotik sistemlerin başlangıç şartlarına hassas bağlılık özelliği ile hedeflenen de bu değildir. Diğer taraftan, önerilen LSB uzatma metodu ile üretilen yörüngeler ise yuvarlama hatalarına maruz kalmamakta ve orijinal matematiksel model ile tanımlanmış gerçek yörüngeler üretmektedir. Bu nedenle sabit / kayan noktalı gerçekleştirim için bahsedilen bu dezavantajlar LSB uzatma metodu ile üretilen yörüngeler için geçerli değildir.

Ayrıca, LSB uzatma metoduna dayalı yörünge üretiminin sadece basit bitsel işlemlere dayalı olması bu metodun diğer bir önemli özelliğidir. Bu sayede LSB uzatma metodu dijital olarak hem yazılımda hem de donanımda kolayca gerçekleştirilebilmektedir. İkinci bölümde farklı İKKH'lar için bu tür gerçekleştirimlerin nasıl yapılacağına dair algoritmalar sunulmuştur. Bu algoritmalara benzer yöntemlerle kaotik haritaların dijital olarak gerçekleştirilmesi klasik kriptografi açısından da daha caziptir. Çünkü, klasik kriptografi tam sayılar kümesinde tanımlı sistemlere dayalıdır ve tam sayılar üzerinde yapılan işlemler basit bitsel işlemlerdir. Oysa ki, sabit / kayan noktalı aritmetikle reel sayılar kümesinde tanımlı sistemlerin dijital olarak gerçekleştiriminin klasik kriptografide bir karşılığı yoktur.

LSB uzatma metodunun kolayca gerçekleştirilebilmesinin diğer bir avantajı olarak, kaotik sistemlerin eğitimi sırasında araştırmacılar hızlıca bu metoda başvurulabilir. Eğitim sırasında sabit / kayan noktalı aritmetik kullanarak sözde (hatalı) yörüngeler üretmek yerine, verilen algoritmalardan biri ve tercihen konjuge fonksiyon kullanarak

gerçek periyodik veya gerçek kaotik yörüngelerin üretilmesi, kaotik sistemlerin dijital gerçekleştirimleri hakkında literatürde bulunan pek çok yanılgıyı ortadan kaldıracaktır. Bu sayede, sonsuz reel sayılar kümesinde tanımlı kaotik sistemlerin, sadece sonlu rasyonel sayılara sahip olan dijital platformlarda sabit / kayan noktalı aritmetik ile neden gerçekleştirilemeyeceği de daha net anlaşılacaktır. Ayrıca, kaotik haritaların periyodik yörüngelere de sahip olduğu gerçeği göz ardı edilmemiş olacaktır.

Literatüre baktığımızda, gerçek kaotik yörüngelerin dijital olarak üretilmesi hakkında LSB uzatma metodu dışında sadece tek bir metot olduğu görülmektedir. Bu metotta, tam sayılar kümesinde tanımlı yeni kaotik haritalar (IDCS) önerilmektedir. Bu haritalar bir GRSÜ tarafından üretilen tam sayıları almakta ve basit bitsel işlemlerle bunları dönüştürmektedir. LSB uzatma metodu ile kıyaslandığında, bu metot ile yeni kaotik haritalar önerilmesi gerekirken, LSB uzatma metodunun literatürde mevcut olan kaotik haritaların gerçekleştirimine imkan tanıdığı görülmektedir. Fakat, literatüre her boyut için tanımlanmış bir IDCS sunulmuşken, her boyut için tanımlı mevcut bir İKKH bulunmamaktadır. Bu ise, LSB uzatma metodunun IDCS metoduna göre bir dezavantajıdır. Bu dezavantajı gidermek adına, üçüncü bölümde literatürde mevcut olan 2B baker haritası her boyut için genelleştirilmiştir. Literatürde bu haritanın sadece 3B için bir genelleştirmesi mevcuttur. Bu bakımdan bu tür bir genelleştirme bir ilktir. Bu genelleştirmenin her boyut için kaotik olduğu Devaney'in kaos tanımına göre matematiksel olarak ispatlanmıştır. Ayrıca, birden büyük her boyut için bu haritanın sadece iki temel yapı bloğu ile gerçekleştirilebileceği yine matematiksel olarak gösterilmiştir. Bu temel yapı bloklarının da basit lojik işlemlerden ibaret olması daha önce bahsetmiş olduğumuz üzere kriptografik uygulamalar ve gerçekleştirim kolaylığı açısından caziptir. Bu bloklar kullanılarak farklı boyutlar için hem yazılım hem de FPGA ile donanım gerçekleştirimleri yapılmıştır. Dolayısıyla, diğer araştırmacılar da benzer şekilde iki temel yapı bloğunu kullanarak ihtiyaç duyulan boyutlarda gerçek periyodik veya gerçek kaotik yörüngeleri elde edebilir. Böylelikle, üçüncü bölüm sayesinde LSB uzatma metodu IDCS metoduna ciddi bir alternatif haline getirilmiştir.

LSB uzatma metodu ile üretilmiş yörüngelerin kaos tabanlı kriptosistemlerde kullanımı ise dördüncü bölümde incelenmiştir. IDCS metodu ile üretilen gerçek kaotik yörüngelerin kriptosistem olarak kullanılması için literatürde analog güvenli haberleşme yapılarına benzer şekilde senkronizasyon kullanılması önerilmektedir. Fakat, analog güvenli

haberleşme hakkındaki çalışmaların birçoğunda ortaya koyulduğu üzere, iletim ortamında bir senkronizasyon sinyalinin çeşitli şekillerde bulunması ciddi güvenlik açıklarına neden olmaktadır. Ayrıca, modern kriptografide bu tür bir senkronizasyon sinyali yerine açık veya gizli anahtarlı protokollerle şifreleme tercih edilmektedir. Bu nedenler yüzünden, gerçek kaotik yörüngelerin senkronizasyon ile şifreleme amacıyla kullanılması çok cazip görünmemektedir. Tıpkı IDCS metodunda olduğu gibi LSB uzatma metodunda da gerçek kaotik yörüngelerin üretilmesi için GRSÜ yapılarına ihtiyaç bulunmaktadır. Ancak, GRSÜ yardımıyla üretilmiş yörüngelerin senkronizasyon olmaksızın başka bir platformda aynen üretilmesi mümkün değildir. Dolayısıyla, bu tez çalışmasında kaos tabanlı kriptografide gerçek kaotik yörüngeler yerine gerçek periyodik yörüngelerin kullanılması savunulmaktadır. Gerçek periyodik yörüngeler LSB uzatma metodu ile herhangi bir platformda arada bir senkronizasyon sinyali olmaksızın üretilebilmektedir. Ayrıca, Sharkovskii teoremine göre bir kaotik sistem (çoğunlukla) akla gelebilecek her periyot uzunluğu için periyodik bir yörüngeye sahip olmalıdır. Bu bakımdan çok uzun periyot uzunluğuna sahip periyodik yörüngeler pratikte gerçek kaotik yörüngelerden farksız olacaktır.

Gerçek periyodik yörüngelerin kaos tabanlı kriptografide kullanımının, literatürde yaygın olarak tercih edilen sabit / kayan noktalı sözde yörünge kullanımına kıyasla pek çok avantajı vardır. Öncelikle kullanılan periyodik yörüngelerin periyot uzunluğu bellidir. Sözde yörüngelerde ise yörüngeyi önceden belirlemek imkansızdır. Sözde yörüngelerin kesişmeleri neticesi konvansiyonel metotla çok az eşsiz yörünge üretilirken gerçek periyodik yörüngelerde kesişim olmaz. Sözde yörüngeler yuvarlama hataları neticesinde tanım aralığı dışına çıkabilir veya sabit noktaya düşebilir. Gerçek periyodik yörüngelerde ise böyle davranışların görülmesi imkansızdır. Dolayısıyla, tüm bu bahsedilen dezavantajlar sözde yörüngelerin kullanıldığı kaos tabanlı kriptosistemlerde ciddi güvenlik risklerine neden olurken, gerçek periyodik yörüngeler kullanıldığında bu riskler elimine edilmektedir.

Fakat, gerçek periyodik yörüngelerin kaos tabanlı kriptografide doğrudan kullanımı da güvenlik riskleri yaratmaktadır. Bunun nedeni, anahtar olarak SRSÜ'lerin başlangıç durumu kullanıldığında kriptografideki yayılma şartının sağlanamamasıdır. Yani, yakın anahtar değerleri aynı yörüngenin geciktirilmiş versiyonlarını üretmektedir. Bunun önüne geçmek için dördüncü bölümde M-İKKH yapısı denilen yeni bir tasarım önerilmiştir. M-

İKKH'nın farklı durumlar için nasıl kullanılacağı bu bölümde detaylıca açıklanmıştır. Bu bölümde ilk kez bir M-İKKH yapısı literatüre önceden sunulmuş olan bir kaos tabanlı kriptosistem yapısı içerisinde kaotik harita olarak kullanılmıştır. Yani LSB uzatma metodu ile üretilmiş gerçek periyodik yörüngeler ilk kez bir kaos tabanlı kriptosistemde kullanılmıştır. Gösterildiği üzere, M-İKKH kullanımı yukarıda bahsetmiş olduğumuz dezavantajları ortadan kaldırmakta ve kullanıldığı kriptosistemin istatistiksel rastgelelik testlerini ciddi derecede iyileştirmektedir. M-İKKH yapısına dayalı kaos tabanlı kriptosistemin donanım gerçekleştirimlerinde dikkat edilmesi gereken hususlar da aynı bölümde irdelenmiş ve FPGA üzerinde donanım gerçekleştirimleri de ilk kez yapılmıştır.

M-İKKH yapıları ile gerçek periyodik yörüngelerin mevcut kaotik sistemler üzerinde kullanılması, yukarıda bahsettiğimiz literatürde bulunan diğer gerçek kaotik yörünge üretim metodu olan IDCS metoduna göre ciddi avantajlar sağlamaktadır. Çünkü, IDCS metodunun mevcut kaos tabanlı kriptosistemler içerisinde nasıl kullanılacağına dair bir çalışma literatürde mevcut değildir. Bunun yanı sıra, IDCS metodunun aksine M-İKKH yapılarının literatürde bilinen kaotik sistemlere ait yörüngeler üretmesi, kaos tabanlı kriptografi hakkında önceden yapılmış olan çalışmaların devamlılığı açısından da önerdiğimiz metodun bir avantajıdır. Keza, gösterildiği üzere M-İKKH yapıları mevcut kaos tabanlı kriptosistemlere kolayca entegre edilebilmektedir. Bu bağlamda, kaos tabanlı kriptografi alanında çalışan araştırmacılar M-İKKH yapılarına daha aşina olacaklarıdır.

Görüldüğü üzere, bu tez çalışması ile kaotik sistemlerin dijital gerçekleştirimlerinde bulunan pek çok açık kapatılmaktadır. Böylelikle, normalde gerçekleştirilmesi mümkün olmayan marjinal tasarımlar olarak düşünülen kaos tabanlı kriptosistemlerin hatasız bir şekilde gerçekleştirilebilir oldukları gösterilmektedir. Daha önce, bahsedilen gerçekleştirim zafiyetleri ve klasik kriptografide kullanılmayan tarzda gerçekleştirimler yüzünden kaos tabanlı kriptosistemler, klasik kriptografi alanında çalışan araştırmacılar tarafından ciddiye alınmamaktaydı. Bu tez çalışmasında önerilen tasarım yaklaşımları ile kaos tabanlı kriptografi ile klasik kriptografi arasındaki boşluk tamamen olmasa da önemli oranda kapatılmıştır. Bu yüzden, ileride kaos tabanlı kriptografinin klasik kriptografi alanında çalışan araştırmacılar tarafında da rağbet görmesi beklenmektedir.

Dolayısıyla, bu tez çalışmasından sonraki çalışmalar M-İKKH yapılarının literatüre önceden sunulmuş olan kaos tabanlı diğer kriptosistemler içerisinde kullanımına

odaklanmalıdır. Her ne kadar bu tez çalışmasında önerilen tasarım yaklaşımları kaos tabanlı kriptosistemlerin güvenli bir şekilde gerçekleştirimlerine imkan tanısa da kullanılan kriptosistemin kendisinin güvenliği de kriptanaliz metotlarıyla incelenmelidir. Maalesef, literatüre sunulan kaos tabanlı kriptosistemlerin dijital gerçekleştirimlerinin nasıl yapılacağına dikkat edilmediği gibi önerilen kriptosistemin güvenliği de klasik kriptografide bulunan yöntemlerle test edilmemektedir. Bu durum, kaos tabanlı kriptografi ile klasik kriptografi arasındaki boşluğun bir diğer nedenidir. Fakat, önceden sabit / kayan noktalı gerçekleştirim bu tür analizlere izin vermezken, basit bitsel işlemlere dayalı bizim önerdiğimiz tasarımlar bu tür analizlere daha uygun olacaktır. Dolayısıyla, kaos tabanlı kriptosistemlerin pratik gerçekleştirimlerinin yanı sıra bu kriptosistemlerin kriptanalizlerinin yapılmasına da ağırlık verilmelidir.

KAYNAKÇA

1. Lorenz, E. N., 1963. Deterministic nonperiodic flow. **Journal of the Atmospheric Sciences**, **20** (2): 130–141.
2. Alligood, K. T., Sauer, T., Yorke, J., 1996. Chaos An Introduction to Dynamical Systems. Springer-Verlag, New York, 603 pp.
3. Shannon, C. E., 1949. Communication theory of secrecy systems. **Bell System Technical Journal**, **28** (4): 656–715.
4. Alvarez, G., Li, S., 2006. Some basic cryptographic requirements for chaos-based cryptosystems. **International Journal of Bifurcation and Chaos**, **16** (08): 2129–2151.
5. Matthews, R., 1989. On the derivation of a chaotic encryption algorithm. **Cryptologia**, **13** (1): 29–42.
6. Cuomo, K. M., Oppenheim, A. V., Strogatz, S. H., 1993. Synchronization of lorenz-based chaotic circuits with applications to communications. **IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing**, **40** (10): 626–633.
7. Kocarev, L., Jakimoski, G., Stojanovski, T., Parlitz, U., From chaotic maps to encryption schemes, 514–517. ISCAS ’98. Proceedings of the 1998 IEEE International Symposium on Circuits and Systems.
8. Kocarev, L., 2001. Chaos-based cryptography: a brief overview. **IEEE Circuits and Systems Magazine**, **1** (3): 6–21.
9. Kocarev, L., Jakimoski, G., 2003. Pseudorandom bits generated by chaotic maps. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, **50** (1): 123–126.
10. Yalçın, M. E., Suykens, J. A. K., Vandewalle, J., 2004. True random bit generation from a double-scroll attractor. **IEEE Transactions on Circuits and Systems I: Regular Papers**, **51** (7): 1395–1404.
11. Shujun, L., Xuanqin, M., Yuanlong, C., 2001. Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography, 316–329. Progress in Cryptology—INDOCRYPT.

12. Jakimoski, G., Kocarev, L., 2001. Chaos and cryptography: block encryption ciphers based on chaotic maps. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, **48** (2): 163–169.

13. Xun Yi, 2005. Hash function based on chaotic tent maps. **IEEE Transactions on Circuits and Systems II: Express Briefs**, **52** (6): 354–357.

14. Xiao, D., Liao, X., Deng, S., 2007. A novel key agreement protocol based on chaotic maps. **Information Sciences**, **177** (4): 1136–1142.

15. Tuncer, T., 2016. The implementation of chaos-based puf designs in field programmable gate array. **Nonlinear Dynamics**, **86** (2): 975–986.

16. Kocarev, L., Lian, S., 2011. Chaos-based Cryptography: Theory, Algorithms and Applications. Springer Science & Business Media, 397 pp.

17. Matsumoto, T., 1984. A chaotic attractor from chua's circuit. **IEEE Transactions on Circuits and Systems**, **31** (12): 1055–1058.

18. Matsumoto, T., Chua, L., Komuro, M., 1985. The double scroll. **IEEE Transactions on Circuits and Systems**, **32** (8): 797–818.

19. Carroll, T. L., Pecora, L. M., 1991. Synchronizing chaotic circuits. **IEEE Transactions on Circuits and Systems**, **38** (4): 453–456.

20. Kılıç, R., Alçı, M., 2001. Mixed-mode chaotic circuit and its application to chaotic communication for transmission of analog signals. **International Journal of Bifurcation and Chaos**, **11** (02): 571–581.

21. Dedieu, H., Kennedy, M. P., Hasler, M., 1993. Chaos shift keying: modulation and demodulation of a chaotic carrier using self-synchronizing chua's circuits. **IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing**, **40** (10): 634–642.

22. Tao Yang, Chua, L. O., 1996. Secure communication via chaotic parameter modulation. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, **43** (9): 817–819.

23. Wu, C. W., Chua, L. O., 1993. A simple way to synchronize chaotic systems with applications to secure communication systems. **International Journal of Bifurcation and Chaos**, **03** (06): 1619–1627.

24. Yang, T., 2004. A survey of chaotic secure communication systems. **International Journal of Computational Cognition**, **2** (2): 81–130.

25. JinFeng, H., JingBo, G., 2008. Breaking a chaotic secure communication scheme. **Chaos: An Interdisciplinary Journal of Nonlinear Science**, **18** (1): 013121.

26. Álvarez, G., Li, S., 2004. Breaking network security based on synchronized chaos. **Computer Communications**, **27** (16): 1679–1681.

27. Li, S., Álvarez, G., Chen, G., Mou, X., 2005. Breaking a chaos-noise-based secure communication scheme. **Chaos: An Interdisciplinary Journal of Nonlinear Science**, **15** (1): 013703.

28. Álvarez, G., Montoya, F., Romera, M., Pastor, G., 2004. Breaking parameter modulated chaotic secure communication system. **Chaos, Solitons & Fractals**, **21** (4): 783–787.

29. Yang, T., Yang, L.-B., Yang, C.-M., 1998. Cryptanalyzing chaotic secure communications using return maps. **Physics Letters A**, **245** (6): 495–510.

30. Yang, T., Yang, L.-B., Yang, C.-M., 1998. Breaking chaotic secure communication using a spectrogram. **Physics Letters A**, **247** (1–2): 105–111.

31. Wheeler, D. D., 1989. Problems with chaotic cryptosystems. **Cryptologia**, **13** (3): 243–250.

32. 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.

33. Wheeler, D. D., Matthews, R. A. J., 1991. Supercomputer investigations of a chaotic encryption algorithm. **Cryptologia**, **15** (2): 140–152.

34. Heidari-Bateni, G., McGillem, C. D., 1994. A chaotic direct-sequence spread-spectrum communication system. **IEEE Transactions on Communications**, **42** (234): 1524–1527.

35. Fryska, S. T., Zohdy, M. A., 1992. Computer dynamics and shadowing of chaotic orbits. **Physics Letters A**, **166** (5–6): 340–346.

36. Blank, M., 1994. Pathologies generated by round-off in dynamical systems. **Physica D: Nonlinear Phenomena**, **78** (1–2): 93–114.

37. Öztürk, İ., Kılıç, R., 2015. A novel method for producing pseudo random numbers from differential equation-based chaotic systems. **Nonlinear Dynamics**, **80** (3): 1147–1157.

38. Öztürk, İ., Kiliç, R., 2014. Cycle lengths and correlation properties of finite precision chaotic maps. **International Journal of Bifurcation and Chaos**, **24** (09): 1450107.

39. Farajallah, M., El Assad, S., Deforges, O., 2016. Fast and secure chaos-based cryptosystem for images. **International Journal of Bifurcation and Chaos**, **26** (02): 1650021.

40. Kumar, M., Kumar, S., Budhiraja, R., Das, M. K., Singh, S., 2017. A cryptographic model based on logistic map and a 3-d matrix. **Journal of Information Security and Applications**, **32**: 47–58.

41. García-Martínez, M., Campos-Cantón, E., 2015. Pseudo-random bit generator based on multi-modal maps. **Nonlinear Dynamics**, **82** (4): 2119–2131.

42. Ullah, A., Jamal, S. S., Shah, T., 2017. A novel construction of substitution box using a combination of chaotic maps with improved chaotic range. **Nonlinear Dynamics**,: .

43. Wang, X., Liu, C., Xu, D., Liu, C., 2016. Image encryption scheme using chaos and simulated annealing algorithm. **Nonlinear Dynamics**, **84** (3): 1417–1429.

44. Huang, X., Ye, G., 2014. An efficient self-adaptive model for chaotic image encryption algorithm. **Communications in Nonlinear Science and Numerical Simulation**, **19** (12): 4094–4104.

45. Li, S., Mou, X., Cai, Y., Ji, Z., Zhang, J., 2003. On the security of a chaotic encryption scheme: problems with computerized chaos in finite computing precision. **Comput. Phys. Commun.**, **153** (1): 52–58.

46. Li, S., Mou, X., Yang, B. L., Ji, Z., Zhang, J., 2003. Problems with a probabilistic encryption scheme based on chaotic systems. **Int. J. Bifurcation Chaos**, **13** (10): 3063–3077.

47. Li, C., Li, S., Chen, G., Halang, W. A., 2009. Cryptanalysis of an image encryption scheme based on a compound chaotic sequence. **Image Vision Comput.**, **27** (8): 1035–1039.

48. Alvarez, G., Li, S., 2006. Breaking an encryption scheme based on chaotic baker map. **Phys. Lett. A**, **352** (1–2): 78–82.

49. Alvarez, G., Amigó, J. M., Arroyo, D., Li, S., 2011. Lessons learnt from the cryptanalysis of chaos-based ciphers, pp. 257–295. In: Chaos-Based Cryptography. Springer.

50. Bahi, J. M., Guyeux, C., 2010. Hash functions using chaotic iterations. **Journal of Algorithms & Computational Technology**, **4** (2): 167–181.

51. Wang, Q., Yu, S., Li, C., Lu, J., Fang, X., Guyeux, C., Bahi, J. M., 2016. Theoretical design and fpga-based implementation of higher-dimensional digital chaotic systems. **IEEE Transactions on Circuits and Systems I: Regular Papers**, **63** (3): 401–412.

52. Devaney, R., 2008. An introduction to chaotic dynamical systems. Westview press, 416 pp.

53. Bahi, J. M., Fang, X., Guyeux, C., Wang, Q., 2014. Suitability of chaotic iterations schemes using xorshift for security applications. **Journal of Network and Computer Applications**, **37**: 282–292.

54. Wang, Q.-X., Yu, S.-M., Guyeux, C., Bahi, J., Fang, X.-L., 2015. Study on a new chaotic bitwise dynamical system and its fpga implementation. **Chinese Physics B**, **24** (6): 060503.

55. Wang, Q., Yu, S., Guyeux, C., Bahi, J. M., Fang, X., 2014. Theoretical design and circuit implementation of integer domain chaotic systems. **International Journal of Bifurcation and Chaos**, **24** (10): 1450128.

56. Öztürk, İ., Kılıç, R., 2018. Digitally generating true orbits of binary shift chaotic maps and their conjugates. **Communications in Nonlinear Science and Numerical Simulation**, **62**: 395–408.

57. Li, S., 2004. When chaos meets computers. **arXiv preprint nlin/0405038**,: 7.

58. Hubbard, J. H. (John H., Hubbard, B. B., 2015. Vector Calculus, Linear Algebra, and Differential forms : A Unified Approach. Matrix Editions, 818 pp.

59. You, Z., Kostelich, E. J., Yorke, J. A., 1991. Calculating stable and unstable manifolds. **International Journal of Bifurcation and Chaos**, **01** (03): 605–623.

60. Ott, E., 2002. Chaos in Dynamical Systems. Cambridge University Press, Cambridge, 478 pp.

61. Smale, S., 1967. Differentiable dynamical systems. **Bulletin of the American Mathematical Society**, **73** (6): 747–818.

62. Wolf, A., Swift, J. B., Swinney, H. L., Vastano, J. A., 1985. Determining lyapunov exponents from a time series. **Physica D: Nonlinear Phenomena**, **16** (3): 285–317.

63. 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** (1–2): 117–134.

64. Li, T.-Y., Yorke, J. A., 1975. Period three implies chaos. **The American Mathematical Monthly**, **82** (10): 985.

65. Li, J., Ye, X. D., 2016. Recent development of chaos theory in topological dynamics. **Acta Mathematica Sinica**, **32** (1): 83–114.

66. Banks, J., Brooks, J., Cairns, G., Davis, G., Stacey, P., 1992. On devaney's definition of chaos. **The American Mathematical Monthly**, **99** (4): 332.

67. Silverman, S., 1992. On maps with dense orbits and the definition of chaos. **Rocky Mountain Journal of Mathematics**, **22** (1): 353–375.

68. Crannell, A., 1995. The role of transitivity in devaney's definition of chaos. **The American Mathematical Monthly**, **102** (9): 788–793.

69. Wiggins, S., 2003. Introduction to Applied Nonlinear Dynamical Systems and Chaos. Springer, 843 pp.

70. Barnsley, M. F., Demko, S., 1985. Iterated function systems and the global construction of fractals. **Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences**, **399** (1817): 243–275.

71. Barnsley, M. F., 1993. Fractals Everywhere. Academic Press, 531 pp.

72. Abbott, S., 2002. Understanding Analysis. Springer, New York, NY, 260 pp.

73. Grassberger, P., Procaccia, I., 1983. Measuring the strangeness of strange attractors. **Physica D: Nonlinear Phenomena**, **9** (1–2): 189–208.

74. Kaplan, J. L., Yorke, J. A., 1979. Chaotic behavior of multidimensional difference

equations, pp. 204–227. In: Functional Differential Equations and Approximation of Fixed Points. Springer.

75. Hartman, P., 2002. Ordinary Differential Equations. Society for Industrial and Applied Mathematics, 612 pp.

76. Paar, C., Pelzl, J., 2009. Understanding Cryptography : A Textbook for Students and Practitioners. Springer, 372 pp.

77. Preneel, B., 2010. Cryptographic hash functions. **European Transactions on Telecommunications**, **5** (4): 431–448.

78. Bassham, L. E., Rukhin, A. L., Soto, J., Nechvatal, J. R., Smid, M. E., Barker, E. B., Leigh, S. D., Levenson, M., Vangel, M., Banks, D. L., Heckert, N. A., Dray, J. F., Vo, S., 2010. SP 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications. National Institute of Standards & Technology.

79. L'Ecuyer, P., Simard, R., 2007. TestU01: a c library for empirical testing of random number generators. **ACM Transactions on Mathematical Software**, **33** (4): 22.

80. Marsaglia, G., DIEHARD: a battery of tests of randomness. https://web.archive.org/web/20160125103112/http://stat.fsu.edu/pub/diehard/.

81. Maes, R., 2013. Physically Unclonable Functions. Springer, Berlin, Heidelberg, 193 pp.

82. Habutsu, T., Nishio, Y., Sasase, I., Mori, S., 1990. A secret key cryptosystem using a chaotic map. **IEICE Transactions**, **E73** (7): 1041–1044.

83. Erdmann, D., Murphy, S., 1992. Hénon stream cipher. **Electronics Letters**, **28** (9): 893.

84. Kohda, T., Tsuneda, A., 1995. Chaotic bit sequences for stream cipher cryptography and their correlation functions, 86–97. Chaotic Circuits for Communication.

85. Kanso, A., 2010. Cipher systems based on controlled exact chaotic maps. **International Journal of Bifurcation and Chaos**, **20** (12): 4039–4053.

86. Lian, S., Sun, J., Wang, J., Wang, Z., 2007. A chaotic stream cipher and the usage in video protection. **Chaos, Solitons & Fractals**, **34** (3): 851–859.

87. Li, S., 2003. Analyses and New Designs of Digital Chaotic Ciphers. Xi'an Jiaotong

University, Doktora Tezi, 193 pp.

88. Álvarez, G., Montoya, F., Romera, M., Pastor, G., 2003. Cryptanalysis of a discrete chaotic cryptosystem using external key. **Physics Letters A**, **319** (3–4): 334–339.

89. Álvarez, G., Montoya, F., Romera, M., Pastor, G., 2003. Cryptanalysis of an ergodic chaotic cipher. **Physics Letters A**, **311** (2–3): 172–179.

90. Pareek, N. K., Patidar, V., Sud, K. K., 2005. Cryptography using multiple one-dimensional chaotic maps. **Communications in Nonlinear Science and Numerical Simulation**, **10** (7): 715–723.

91. Patidar, V., Sud, K., Pareek, N., 2009. A pseudo random bit generator based on chaotic logistic map and its statistical testing. **Journal of Informatica**, **33** (4): 441–452.

92. Scharinger, J., 1998. Fast encryption of image data using chaotic kolmogorov flows. **Journal of Electronic Imaging**, **7** (2): 318.

93. Lian, S., Sun, J., Wang, Z., 2005. A block cipher based on a suitable use of the chaotic standard map. **Chaos, Solitons & Fractals**, **26** (1): 117–129.

94. Armand Eyebe Fouda, J. S., Yves Effa, J., Sabat, S. L., Ali, M., 2014. A fast chaotic block cipher for image encryption. **Communications in Nonlinear Science and Numerical Simulation**, **19** (3): 578–588.

95. Hussain, I., Shah, T., Gondal, M. A., 2013. Application of s-box and chaotic map for image encryption. **Mathematical and Computer Modelling**, **57** (9–10): 2576–2579.

96. Xun Yi, Chik How Tan, Chee Kheong Siew, 2002. A new block cipher based on chaotic tent maps. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, **49** (12): 1826–1829.

97. Masuda, N., Jakimoski, G., Aihara, K., Kocarev, L., 2006. Chaotic block ciphers: from theory to practical algorithms. **IEEE Transactions on Circuits and Systems I: Regular Papers**, **53** (6): 1341–1352.

98. Wong, K.-W., 2003. A combined chaotic cryptographic and hashing scheme. **Physics Letters A**, **307** (5–6): 292–298.

99. Khan, M. K., Zhang, J., Wang, X., 2008. Chaotic hash-based fingerprint biometric

remote user authentication scheme on mobile devices. **Chaos, Solitons & Fractals**, **35** (3): 519–524.

100. Yang, H., Wong, K.-W., Liao, X., Wang, Y., Yang, D., 2009. One-way hash function construction based on chaotic map network. **Chaos, Solitons & Fractals**, **41** (5): 2566–2574.

101. Argyris, A., Deligiannidis, S., Pikasis, E., Bogris, A., Syvridis, D., 2010. Implementation of 140 gb/s true random bit generator based on a chaotic photonic integrated circuit. **Optics Express**, **18** (18): 18763.

102. Pareschi, F., Setti, G., Rovatti, R., 2010. Implementation and testing of high-speed cmos true random number generators based on chaotic systems. **IEEE Transactions on Circuits and Systems I: Regular Papers**, **57** (12): 3124–3137.

103. Liu, L., Zhang, Q., Wei, X., 2012. A rgb image encryption algorithm based on dna encoding and chaos map. **Computers & Electrical Engineering**, **38** (5): 1240–1248.

104. Öztürk, İ., 2013. Kaotik Sistem Tabanlı Sözde Rastgele Sayı Üreteçlerinin Tasarımı ve Gerçekleştirilmesi. Erciyes Üniversitesi, Yüksek Lisans Tezi, 101 pp.

105. Grebogi, C., Ott, E., Yorke, J. A., 1988. Roundoff-induced periodicity and the correlation dimension of chaotic attractors. **Physical Review A**, **38** (7): 3688–3692.

106. Arroyo, D., Li, C., Li, S., Alvarez, G., Halang, W. A., 2009. Cryptanalysis of an image encryption scheme based on a new total shuffling algorithm. **Chaos, Solitons & Fractals**, **41** (5): 2613–2616.

107. Öztürk, İ., Kılıç, R., 2018. Analysis and experimental verification of a new chaotic system with hyperbolic/non-hyperbolic equilibria. **Chaos, Solitons & Fractals**, **117**: 183–191.

108. Hirsch, M. W., Smale, S., Devaney, R. L., 2012. Differential Equations, Dynamical Systems, and an Introduction to Chaos. Academic Press, 415 pp.

109. Geist, K., Parlitz, U., Lauterborn, W., 1990. Comparison of different methods for computing lyapunov exponents. **Progress of Theoretical Physics**, **83** (5): 875–893.

110. Dhooge, A., Govaerts, W., Kuznetsov, Y. A., 2003. MATCONT: a matlab package for numerical bifurcation analysis of odes. **ACM Transactions on Mathematical Software**, **29** (2): 141–164.

111. Feigenbaum, M. J., 1978. Quantitative universality for a class of nonlinear transformations. **Journal of Statistical Physics**, **19** (1): 25–52.

112. 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** (17n19): 2617–2622.

113. Martínez-González, R. F., Díaz-Méndez, J. A., Palacios-Luengas, L., López-Hernández, J., Vázquez-Medina, R., 2016. A steganographic method using bernoulli's chaotic maps. **Computers & Electrical Engineering**, **54**: 435–449.

114. López-Hernández, J., Vázquez-Medina, R., Ortiz-Moctezuma, M. B., Del-Rio-Correa, J. L., 2012. Digital implementation of a pseudo-random noise generator using chaotic maps. **IFAC Proceedings Volumes**, **45** (12): 209–214.

115. Cao, L.-C., Luo, Y.-L., Qiu, S.-H., Liu, J.-X., 2015. A perturbation method to the tent map based on lyapunov exponent and its application. **Chinese Physics B**, **24** (10): 100501.

116. Martínez-González, R. F., Díaz-Méndez, J. A., Rojas-López, C. E., Vázquez-Medina, R., 2012. Digital noise generator based on bernoulli chaotic map. **Applied Mechanics and Materials**, **152–154**: 1869–1873.

117. Addabbo, T., Alioto, M., Fort, A., Rocchi, S., Vignoli, V., 2006. The digital tent map: performance analysis and optimized design as a low-complexity source of pseudorandom bits. **IEEE Transactions on Instrumentation and Measurement**, **55** (5): 1451–1458.

118. Wang, H., Song, B., Liu, Q., Pan, J., Ding, Q., 2014. FPGA design and applicable analysis of discrete chaotic maps. **International Journal of Bifurcation and Chaos**, **24** (04): 1450054.

119. Li, C., Luo, G., Qin, K., Li, C., 2017. An image encryption scheme based on chaotic tent map. **Nonlinear Dynamics**, **87** (1): 127–133.

120. Alvarez, E., Fernández, A., García, P., Jiménez, J., Marcano, A., 1999. New

approach to chaotic encryption. **Physics Letters A**, **263** (4–6): 373–375.

121. Ablay, G., 2016. Chaotic encryption based data transmission using delta and delta-sigma modulators. **International Journal of Applied Mathematics, Electronics and Computers**, **4** (Special Issue-1): 368–373.

122. Amin, M., Faragallah, O. S., Abd El-Latif, A. A., 2010. A chaotic block cipher algorithm for image cryptosystems. **Communications in Nonlinear Science and Numerical Simulation**, **15** (11): 3484–3497.

123. McCauley, J. L., Palmore, J. I., 1986. Computable chaotic orbits. **Physics Letters A**, **115** (9): 433–436.

124. Turing, A. M., 1937. On computable numbers, with an application to the entscheidungsproblem. **Proceedings of the London Mathematical Society**, **2** (1): 230–265.

125. Bailey, D., Borwein, P., Plouffe, S., 1997. On the rapid computation of various polylogarithmic constants. **Mathematics of Computation**, **66** (218): 903–913.

126. Saito, A., Ito, S., 2014. Computation of true chaotic orbits using cubic irrationals. **Physica D: Nonlinear Phenomena**, **268**: 100–105.

127. Niven, I., 1956. Irrational Numbers. The Mathematical Association of America, Washington, 163 pp.

128. Maggio, G., Galias, Z., 2002. Applications of symbolic dynamics to differential chaos shift keying. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, **49** (12): 1729–1735.

129. Maggio, G. M., Reggiani, L., 2005. Pseudo-chaotic communication method exploiting symbolic dynamics. US Patent No. 6,882,689.

130. Mozsary, A., Azzinari, L., Krol, K., Porra, V., 2001. Theoretical connection between pn-sequences and chaos makes simple fpga pseudo-chaos sources possible, 457–460. ICECS 2001. 8th IEEE International Conference on Electronics, Circuits and Systems.

131. Flake, G. W., 1998. The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation. MIT Press, 520 pp.

132. Sharkovskii, A. N., 1995. Coexistence of cycles of a continuous map of the line into

itself. **International Journal of Bifurcation and Chaos**, **05** (05): 1263–1273.

133. Mason, J. C., Handscomb, D. C., 2002. Chebyshev Polynomials. CRC Press, 360 pp.

134. Fischer, V., Drutarovský, M., 2003. True random number generator embedded in reconfigurable hardware, pp. 415–430. In: Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems.

135. Aulbach, B., Kieninger, B., 2001. On three definitions of chaos. **Nonlinear Dynamics and Systems Theory**, **1** (1): 23–37.

136. Mao, Y., Chen, G., Lian, S., 2004. A novel fast image encryption scheme based on 3d chaotic baker maps. **International Journal of Bifurcation and Chaos**, **14** (10): 3613–3624.

137. Folland, G., 2009. A Guide to Advanced Real Analysis. The Mathematical Association of America, Washington DC, 107 pp.

138. Baier, G., Klein, M., 1990. Maximum hyperchaos in generalized hénon maps. **Physics Letters A**, **151** (6–7): 281–284.

139. Wold, K., Tan, C. H., 2009. Analysis and enhancement of random number generator in fpga based on oscillator rings. **International Journal of Reconfigurable Computing**, **2009**: 1–8.

140. Short, K. M., 1997. Signal extraction from chaotic communications. **International Journal of Bifurcation and Chaos**, **07** (07): 1579–1597.

141. Li, S., Álvarez, G., Chen, G., 2005. Breaking a chaos-based secure communication scheme designed by an improved modulation method. **Chaos, Solitons & Fractals**, **25** (1): 109–120.

142. Li, S., Alvarez, G., Li, Z., Halang, W. A., 2007. Analog chaos-based secure communications and cryptanalysis: a brief survey. **arXiv preprint nlin/0710.5455**,: 6.

143. Sýs, M., Říha, Z., 2014. Faster randomness testing with the nist statistical test suite, pp. 272–284. In: Security, Privacy, and Applied Cryptography Engineering. Springer.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	: İsmail Öztürk
Uyruğu	: Türkiye (T.C)
Doğum Tarihi ve Yeri	: 13.07.1985 - Burdur
Medeni Durum	: Bekar
e-mail	: ismail.ozturk@amasya.edu.tr
Yazışma Adresi	: Amasya Üniversitesi, Teknoloji Fakültesi, A Blok 103 Nolu Oda, 05100, Merkez / AMASYA

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi, Elektrik-Elektronik Müh.	2013
Lisans	Erciyes Üniversitesi, Elektrik-Elektronik Müh.	2011
Lise	Maltepe Askeri Lisesi, İzmir	2004

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2018-Halen	Amasya Üniversitesi	Araştırma Gör.
2012-2018	Erciyes Üniversitesi	Araştırma Gör.

YABANCI DİL

İngilizce

SCI SCI-E KAPSAMINDAKİ YAYINLAR

1. Öztürk İ., Kılıç R., Digitally generating true orbits of binary shift chaotic maps and their conjugates, *Commun Nonlinear Sci Numer Simul*, vol. 62, pp. 395–408, 2018. (doi : 10.1016/j.cnsns.2018.02.039)

2. Korkmaz N., Öztürk İ., Kalınlı A., Kılıç R., A comparative study on determining nonlinear function parameters of the Izhikevich neuron model, *J Circuit Syst Comp*, vol. 27(10), pp. 1850164, 2018. (doi : 10.1142/S0218126618501645)

3. Korkmaz N., Öztürk İ., Kılıç R., Modeling, simulation and implementation issues of CPGs for neuromorphic engineering applications, *Comput Appl Eng Educ*, vol. 26(4), pp. 782-803, 2018. (doi : 10.1002/cae.21972)

4. Öztürk İ., Kılıç R., Analysis and experimental verification of a new chaotic system with hyperbolic/non-hyperbolic equilibria, *Chaos Soliton Fract*, vol. 117, pp. 183-191, 2018. (doi : 10.1016/j.chaos.2018.10.028)

5. Korkmaz N., Öztürk İ., Kılıç R., "Multiple perspectives on the hardware implementations of biological neuron models and programmable design aspects", *Turk J Elec Eng & Comp Sci*, vol. 24, pp. 1729-1746, 2016. (doi : 10.3906/elk-1309-5)

6. Korkmaz N., Öztürk İ., Kılıç R., "The investigation of chemical coupling in a HR neuron model with reconfigurable implementations", *Nonlinear Dyn*, vol. 86(3), pp. 1841-1854, 2016. (doi : 10.1007/s11071-016-2996-6)

7. Öztürk İ., Kılıç R., "A novel method for producing pseudo random numbers from differential equation-based chaotic systems", *Nonlinear Dyn*, vol. 80(3), pp. 1147-1157, 2015. (doi : 10.1007/s11071-015-1932-5)

8. Öztürk İ., Kılıç R., "Cycle lengths and correlation properties of finite precision chaotic maps", *Int J Bifurcation Chaos*, vol. 24(9), pp. 1450107, 2014. (doi : 10.1142/S0218127414501077)

9. Dahasert N., Öztürk İ., Kılıç R., "Experimental realizations of the HR neuron model with programmable hardware and synchronization applications", *Nonlinear Dyn*, vol. 70(4), pp. 2343–2358, 2012. (doi : 10.1007/s11071-012-0618-5)

ULUSAL /ULUSLARASI KONFERANSLARDA SUNULAN BİLDİRİLER

1. Öztürk İ., Kılıç R., "A new pseudo random number generator based on Chebyshev maps and parameter switching", 6th International Conference on Control Engineering & Information Technology (CEIT 2018), İstanbul, TÜRKİYE, 25-27 Ekim 2018

2. Korkmaz N., Kılıç R., Kalınlı A., Öztürk İ., "Parameter estimations for the modified Izhikevich neuron model with optimization methods", 10th Chaotic Modeling and Simulation International Conference, Barselona, İSPANYA, 30 Mayıs - 2 Haziran 2017, pp.70-70

3. Korkmaz N., Öztürk İ., Kalınlı A., Kılıç R., "Hardware verification: Determining the parameters of the modified Izhikevich neuron model with genetic algorithm", 10th International Conference on Electrical and Electronics Engineering (ELECO), Bursa, TÜRKIYE, 30 Kasım - 2 Aralık 2017, pp.1-1

4. Kılıç R., Korkmaz N., Öztürk İ., "A comparative study on experimental realizations of multiscroll chaos generators", The 9th Chaotic Modeling and Simulation International Conference, Londra, İNGİLTERE, 23-26 Mayıs 2016, pp.51-51

5. Arık S., Korkmaz N., Öztürk İ., Kılıç R., Günay E., "Reconfigurable implementations of PWL-based multiscroll chaos generator", 8th International Conference on Electrical and Electronics Engineering (ELECO), Bursa, TÜRKIYE, 28-30 Kasım 2013, pp.604-608

6. Günay E., Kılıç R., Dahasert N., Öztürk İ., "Merkezi desen üreteçleri için donanımsal çözümler", Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu, Bursa, TÜRKİYE, 29 Kasım-01 Aralık 2012, pp.332-335

7. Dahasert N., Öztürk İ., Kılıç R., "Izhikevich nöron modelinin alan programlanabilir elemanlarla gerçekleştirimi", 20. IEEE Sinyal İşleme ve Uygulamaları Kurultayı (SIU), Antalya, TÜRKİYE, 18-20 Nisan 2012