

**T. C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**KAOTİK SİSTEM TASARIMINDA
PROGRAMLANABİLİR ANALOG DİZİ
UYGULAMALARI**

**Tezi Hazırlayan
Fatma YILDIRIM DALKIRAN**

**Tezi Yöneten
Prof. Dr. Recai KILIÇ**

**Elektronik Mühendisliği Anabilim Dalı
Doktora Tezi**

**Mart 2011
KAYSERİ**

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

**KAOTİK SİSTEM TASARIMINDA
PROGRAMLANABİLİR ANALOG DİZİ
UYGULAMALARI
(Doktora Tezi)**

**Tezi Hazırlayan
Fatma YILDIRIM DALKIRAN**

**Tezi Yöneten
Prof. Dr. Recai KILIÇ**

**Bu çalışma; Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimi
tarafından FBT-07-11 kodlu proje ile desteklenmiştir.**

**Mart 2011
KAYSERİ**

Prof. Dr. Recai KILIÇ danışmanlığında **Fatma YILDIRIM DALKIRAN** tarafından hazırlanan **“Kaotik Sistem Tasarımında Programlanabilir Analog Dizi Uygulamaları”** adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilim Dalında **Doktora** tezi olarak kabul edilmiştir.

11/03/2011

JÜRİ:

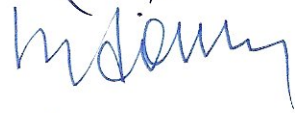
Başkan: Prof. Dr. Shahram MİNAEI



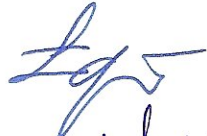
Üye : Prof. Dr. Recai KILIÇ



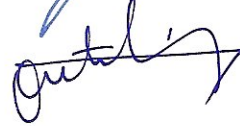
Üye : Prof. Dr. Kenan DANIŞMAN



Üye : Yrd. Doç. Dr. Enis GÜNAY



Üye : Yrd. Doç. Dr. Aytekin BAĞIŞ

**ONAY:**

Bu tezin kabulü, Enstitü Yönetim Kurulunun 29/03/2011 tarih ve 2011/13_02. sayılı kararı ile onaylanmıştır.

29/03/2011.


Prof. Dr. Necmettin MARAŞLI
Enstitü Müdürü

TEŐEKKÖR

Doktora tezimin gerekleŐmesinde bilgi ve yardımlarını esirgemeyerek uyarıları ile yol gÖsteren tez danıŐmanım Sayın Prof. Dr. Recai KILIÇ' a, tezin yÖnlendirilmesinde katkılarda bulunan hocalarım Prof. Dr. Kenan DANIŐMAN ve Prof. Dr. Mehmet Emin YÖKSEL' e alıŐmama dođrudan ya da dolaylı olarak katkılarda bulunmuŐ olan tÖm arkadaşlarıma ve deđerli hocalarıma teŐekkÖrÖ bor bilirim. alıŐmalarımda bana yorumlarıyla yardımcı olan ve bilgisayar baŐında ge saatlere kadar bana eŐlik eden eŐim İlker DALKIRAN' a ve bu tez alıŐması sÖresince her ne kadar kendisi benim alıŐmalarımın idrakinde olmasa da, bana her zaman gÖlen yÖzÖyle moral veren biricik kızım Elif' e ŐÖkranlarımı sunarım.

Bu tez alıŐması Erciyes Üniwersitesi Bilimsel AraŐtırma Projeleri Birimi tarafından “**FBT-07-11**” kodlu proje ile desteklenmiŐtir.

KAOTİK SİSTEM TASARIMINDA PROGRAMLANABİLİR ANALOG DİZİ UYGULAMALARI

Fatma YILDIRIM DALKIRAN
Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Doktora Tezi, Mart 2011
Tez Danışmanı: Prof. Dr. Recai KILIÇ

ÖZET

Kaotik işaret üreten osilatör devreleri, kaos tabanlı uygulamaların en önemli tasarım aşamasını oluşturmaktadır. Çünkü bu uygulamalarda kaotik dinamiklerinin karakteristik özelliklerinden yararlanmak ve bu özellikleri fiziki hayata geçirebilmek için öncelikle bu dinamiklerin pratik olarak üretilmesi gerekmektedir. Literatürde esnek mimarili kaotik işaret üreteç sayısının azlığından dolayı, bu konuda yeni ve farklı özelliklere sahip kaotik işaret üreteçlerinin literatüre kazandırılması gerekliliği vardır.

Son yıllarda elektronik devre tasarımında tasarım esnekliği sağlayan Analog Dizi Elemanlarına dayalı yeni devre tekniklerinin kaotik osilatör tasarımına da uygulanması, kaotik devre modellerini ve bu modellerin kullanıldığı uygulamaları çok daha esnek ve kullanışlı kılacaktır. Bu tez çalışmasında; kaos tabanlı uygulamalar için yeni ve programlanabilir kaotik işaret üreteç devre modellerinin deneysel olarak tasarlanması amaçlanmıştır. Bu amaç doğrultusunda Alan Programlanabilir Analog Dizi Elemanına (Field Programmable Analog Array, FPAA) dayalı devre düzenekleri kullanılarak analog devre tasarımına uygun, yeniden yapılandırılabilir ve programlanabilir kaotik işaret üreteç devre yapıları tasarlanmıştır. Ayrıca bu programlanabilir kaotik osilatör devrelerinin senkronizasyon uygulamaları da incelenmiştir. Tez çalışması kapsamında kaotik osilatörlerle ilgilenen araştırmacılar, lisans ve lisansüstü öğrenciler için faydalı bir eğitim materyali olan AnadigmCHAOS arayüzü geliştirilmiştir.

Altı bölümden oluşan tez çalışmasının ilk bölümünde FPAA' lara kısa bir giriş yapılmış, literatürde yer alan FPAA' lar kullanılarak gerçekleştirilen kaotik osilatörlerle ilgili çalışmalar incelenmiş ve tezin amacı verilmiştir.

İkinci bölümde FPAA' lar, FPAA' ların tarihsel gelişimi ve tez çalışmasında kullanılan FPAA ile ilgili detaylı bilgi verilmiştir.

Üçüncü bölümde, tez çalışması kapsamında programlanabilir ve yeniden yapılandırılabilir olarak tasarımı yapılan kaotik osilatörlerin klasik devre modelleri üzerinde durulmuş olup, FPAA' lı eşdeğerleri ile karşılaştırma yapılabilmesi için nümerik analizleri ve ayrık deneysel gerçekleştirmeleri yapılmıştır.

Dördüncü bölümde, bir önceki bölümde verilen kaotik osilatör devre modellerinin FPAA-tabanlı tasarım ve gerçekleştirmeleri ile bu yapıların senkronizasyon uygulamaları üzerinde durulmuştur.

Beşinci bölümde FPAA' nın yazılım programı olan *AnadigmDesigner 2* programından ve programda yer alan *AnadigmFilter* ve *AnadigmPID* yardımcı araçlarından bahsedilmiştir. Bu yardımcı araçlar arasında yer alabilecek kaotik sistem tasarımına dönük *AnadigmCHAOS* isimli yeni bir uygulama geliştirilmiştir.

Altıncı ve son bölümde elde edilen sonuçlar tartışılmış ve değerlendirilmiştir.

Anahtar Sözcükler: Kaos, Kaotik Sistem Modelleme, Kaotik Osilatör, FPAA, *AnadigmCHAOS* yazılımı.

PROGRAMMABLE ANALOG ARRAY APPLICATIONS IN CHAOTIC SYSTEM DESIGN

Fatma YILDIRIM DALKIRAN

Erciyes University, Graduate School of Natural and Applied Sciences

Ph. D. Thesis, March 2011

Thesis Supervisor: Prof. Dr. Recai KILIÇ

ABSTRACT

Chaos generators are the most important design step of chaos-based applications. Because of utilizing the characteristic features of chaotic dynamics and applying those features in physical life, firstly it is necessary to generate these chaotic dynamics practically in these applications. Due to the fact that there are not enough amounts of well-known chaotic circuits with flexible architecture in the literature, in this context, it is necessary that new chaos generators which have different specifications type should be described.

Of late years analogue array units provides design flexibility in the area of electronic circuit design. Analogue array units based new circuit design techniques which are applied on the design of chaotic oscillators make flexible and useful those chaotic circuits and the realizations of those models. In this thesis it is aimed that new and programmable chaotic signal generator models which can be used in chaos based applications are practically designed. For this purpose chaotic signal generators which are reconfigurable, programmable and suitable for analogue circuit design, are designed by use of Field Programmable Analog Array, (FPAA) based circuit modules. Beside the synchronization applications of those reprogrammable chaotic oscillator circuits are also investigated. In addition a user interface program named as AnadigmCHAOS and useful for researchers, undergraduate and graduate students interested in chaotic oscillators have been developed.

In the first section of this thesis consisting of six sections a brief information about FPAA's is given, the studies related to FPAA based chaotic oscillators are investigated and the aim of this thesis is explained.

In the second section, the detailed information about FPAA's, historical development and structure of FPAA's is given.

In the third section, classical circuit models of chaotic oscillators which have programmable and reconfigurable design technique are insisted on in the context of this thesis. In addition to compare the performance of equivalent FPAA-based designs numerical analysis and practical implementations of classical circuit models of chaotic oscillators have been made.

FPAA-based design and implementation of chaotic oscillators, mentioned in previous section, are presented in the fourth section. Besides synchronization applications of FPAA-based chaotic oscillators are also studied carefully.

In fifth section some information about AnadigmDesigner 2 that is software of FPAA and the subtools of this program that are *AnadigmFilter* and *AnadigmPID* are mentioned. A new software application that is related to chaotic system design, named as AnadigmCHAOS and can be considered among these subtools of AnadigmDesigner 2 is developed.

In the sixth and the final section the obtained results are discussed and evaluated.

Keywords: Chaos, Chaotic System Modeling, Chaotic Oscillators, FPAA, AnadigmCHAOS Software.

İÇİNDEKİLER

KABUL VE ONAY	i
TEŞEKKÜR	ii
ÖZET	iii
ABSTRACT	v
İÇİNDEKİLER	vii
KISALTMALAR LİSTESİ	xi
TABLolar LİSTESİ	xii
ŞEKİLLER LİSTESİ	xiv
1. BÖLÜM	
GİRİŞ	1
1.1. Giriş	1
1.2. Çalışmanın Amacı	4
2. BÖLÜM	
ALAN PROGRAMLANABİLİR ANALOG DİZİLER	6
2.1. Alan Programlanabilir Analog Dizi (FPAA)	6
2.2. FPAA' nın Tasarım Metodolojisi	7
2.2.1. Ayırık Zamanlı / Sürekli Zamanlı Çalışma	7
2.2.1.1. Anahtarlamalı Kapasitör Tekniği	8
2.2.2. Voltaj Modlu / Akım Modlu Çalışma	9
2.3. FPAA' nın Tarihsel Gelişimi	10
2.3.1. Bireysel Olarak Gerçekleştirilen FPAA Tasarımları	11
2.3.2. Ticari Olarak Gerçekleştirilen FPAA Tasarımları	13
2.4. FPAA' ların Uygulama Alanları	17
2.5. Tez Çalışmasında Kullanılan FPAA' nın Özellikleri	18
2.5.1. AN221K04-DVLP2 Uygulama Kartının Çoklu Kullanımı	20
2.6. FPAA' nın Bilgisayar Arayüz Programı: AnadigmDesigner 2	21

3. BÖLÜM	
ANALOG KAOTİK OSİLATÖRLER: MODELLEME, SİMÜLASYON VE TASARIM	38
3.1. Giriş	38
3.2. Kaos İle İlgili Kavramlar	39
3.2.1. Dallanma Diyagramı	39
3.2.2. Lyapunov Üstelleri	40
3.2.3. Zaman / Frekans Gösterimleri	41
3.2.4. Faz Düzlemi Gösterim	42
3.2.5. Lojistik Harita (Logistic Map)	43
3.2.6. Poincare Kesit Gösterimi	43
3.3. Elektronik Devrelerde Kaos ve Analog Kaos Üreteçleri	45
3.3.1. Analog Kaotik İşaret Üreteçler	45
3.3.1.1. Chua Osilatörü	47
3.3.1.1.1. Kaotik Chua Osilatörünün Donanım Gerçekleştirimi	51
3.3.1.2. Kaotik Colpitts Osilatörü	56
3.3.1.2.1. Kaotik Colpitts Osilatörün Donanım Gerçekleştirimi	57
3.3.1.3. Duffing Osilatör	59
3.3.1.3.1. Duffing Osilatörün Donanım Gerçekleştirimi	59
3.3.1.4. Kennedy Osilatör	61
3.3.1.4.1. Kennedy Osilatörün Donanım Gerçekleştirimi	62
3.3.1.5. Lorenz Sistemi	63
3.3.1.5.1. Lorenz Sisteminin Donanım Gerçekleştirimi	65
3.3.1.6. MLC (Murali-Lakshmanan-Chua) Osilatörü	68
3.3.1.6.1. MLC Osilatörün Donanım Gerçekleştirimi	69
3.3.1.7. Rössler Sistemi	70
3.3.1.7.1. Rössler Sisteminin Donanım Gerçekleştirimi	71
3.3.1.8. Sprot Sistemleri	74
3.3.1.8.1. Sprot Sisteminin Donanım Gerçekleştirimi	76
3.3.1.9. HSA-Tabanlı Chua Osilatörü	78
3.3.1.9.1. HSA-Tabanlı Chua Osilatörünün Donanım Gerçekleştirimi	79

4. BÖLÜM

FPAA-TABANLI KAOTİK OSİLATÖR TASARIMI VE UYGULAMALARI	82
4.1. Giriş	82
4.2. Programlanabilirlik ve Yeniden Yapılandırılabilirlik	82
4.3. Tasarım Prosedürü ve Deneysel Kurulum	83
4.4. FPAA-Tabanlı Kaotik Osilatör Tasarımları	85
4.4.1. FPAA-Tabanlı Chua Osilatörü	85
4.4.1.1. FPAA-Tabanlı Chua Devre Modeli-I	86
4.4.1.2. FPAA-Tabanlı Chua Devre Modeli-II	89
4.4.1.3. FPAA-Tabanlı Chua Devre Modeli-III	92
4.4.1.4. FPAA-Tabanlı Chua Devre Modeli-IV	96
4.4.2. FPAA-Tabanlı Kaotik Colpitts Osilatörü	97
4.4.3. FPAA-Tabanlı Duffing Osilatör	100
4.4.4. FPAA-Tabanlı Kennedy Osilatör	102
4.4.5. FPAA-Tabanlı Lorenz Sistemi	102
4.4.6. FPAA-Tabanlı MLC Osilatör	106
4.4.7. FPAA-Tabanlı Rössler Sistemi	108
4.4.8. FPAA-Tabanlı Sprot Systemlerinin Gerçekleştirimi	110
4.4.8.1. FPAA-Tabanlı Sprot System Modeli-I	110
4.4.8.2. FPAA-Tabanlı Sprot System Modeli-II	113
4.4.8.3. FPAA-Tabanlı Sprot System Modeli-III	115
4.4.8.4. FPAA-Tabanlı Sprot System Modeli-IV	115
4.4.8.5. FPAA-Tabanlı Sprot System Modeli-V	118
4.4.9. HSA-Tabanlı Kaotik Yapıların FPAA' da Gerçekleştirimi	118
4.4.9.1. HSA-Tabanlı Chua Devresinin FPAA' da Gerçekleştirimi	118
4.4.9.2. HSA-Tabanlı MLC Devresinin FPAA' da Gerçekleştirimi	122
4.4.10. FPAA-Tabanlı n -Çekerli Kaos Üreteçleri	124
4.5. FPAA-Tabanlı PLL Modeli	131
4.6. FPAA-Tabanlı Kaotik Sistemlerin Kullanıldığı Senkronizasyon Gerçekleştirimleri	133
4.6.1. FPAA-Tabanlı Chua Devresinin Kullanıldığı Senkronizasyon Gerçekleştirimleri	133

4.6.1.1.	FPAA-Tabanlı Chua Devresinin Kullanıldığı Kaskad Bağlantı Yöntemi İle Senkronizasyon Gerçekleştirimi	133
4.6.1.2.	FPAA-Tabanlı Chua Devresinin Kullanıldığı Pecora&Carroll Metodu İle Senkronizasyon Gerçekleştirimi	137
4.6.1.3.	FPAA-Tabanlı Chua Devresinin Kullanıldığı Kuplajlama Metodu İle Senkronizasyon Gerçekleştirimi	139
4.6.2.	FPAA-Tabanlı MLC Sisteminin Kullanıldığı Senkronizasyon Gerçekleştirmeleri	141
4.6.2.1.	FPAA-Tabanlı MLC Sisteminin Kullanıldığı Pecora&Carroll Metodu İle Senkronizasyon Gerçekleştirimi	143
4.6.2.2.	FPAA-Tabanlı MLC Sisteminin Kullanıldığı Tek Yönlü Kuplajlama Metodu İle Senkronizasyon Gerçekleştirimi	144
5. BÖLÜM		
	FPAA YAZILIM UYGULAMASI	147
5.1.	Giriş	147
5.2.	AnadigmDesigner 2 Arayüzü ve Yardımcı Araçları	148
5.2.1.	AnadigmDesigner 2 Arayüzünde Filtre Tasarım Aracı: <i>AnadigmFilter</i>	148
5.2.2.	AnadigmDesigner 2 Arayüzünde Kontrolör Tasarım Aracı: <i>AnadigmPID</i>	151
5.3.	AnadigmDesigner 2 Arayüzünde Kaotik Sistem Tasarım Aracı: <i>AnadigmCHAOS</i>	154
6. BÖLÜM		
	SONUÇ VE DEĞERLENDİRME	169
	KAYNAKLAR	176
	ÖZGEÇMİŞ	189

KISALTMALAR LİSTESİ

FPAA	Field Programable Analog Array
FPGA	Field Programable Gate Array
CAB	Configurable Analog Block
CAM	Configurable Analog Module
FPMA	Field Programmable Mixed-Signal Array
EPAC	Electrically Programmable Analog Circuit
TRAC	Totally Reconfigurable Analog Circuit
LUT	Look-Up Table
SAR	Successive-Approximation Register
ADC	Analog-Digital Converter
HSA	Hücresel Sinir Ağı
PLL	Phase Locked Loop
EDA	Electronic Design Automation
PID	Proportional Integral Derivative
GUI	Graphic User Interface
CFOA	Current Feedback Operational Amplifier
VNIC	Voltage Controlled Negative Impedance Converter
CCII	Second Generation Current Conveyor
VOA	Voltage Op-Amp
A/D	Analog/Dijital
D/A	Dijital/Analog
VLSI	Very Large Scale Integration

TABLOLAR LİSTESİ

	Sayfa No
Tablo 2.1. Bireysel Olarak Gerçekleştirilen FPAA Tasarımlarının Birbirleri İle Karşılaştırılması.	13
Tablo 2.2. Ticari Olarak Gerçekleştirilen FPAA Tasarımlarının Birbirleri İle Karşılaştırılması.	17
Tablo 3.1. Kaotik Chua Osilatörünü Modellemede En Çok Kullanılan Doğrusal Olmayan Fonksiyonlar.	47
Tablo 3.2. Sprot Systemlerini Modellemede Kullanılan Doğrusal Olmayan Fonksiyonlar.	74
Tablo 4.1. Chua Devresinin FPAA Gerçekleştirim Şemaları.	87
Tablo 4.2. Denklem (4.7) ile İfade Edilen FPAA-Tabanlı Chua Devre Modeli-I' in CAM Modülleri ve Parametre Değerleri.	90
Tablo 4.3. Denklem (4.2) ile İfade Edilen FPAA-Tabanlı Chua Devre Modeli-II' nin CAM Modülleri ve Parametre Değerleri.	92
Tablo 4.4. Denklem (4.3) ile İfade Edilen FPAA-Tabanlı Chua Devre Modeli-III' ün CAM Modülleri ve Parametre Değerleri.	94
Tablo 4.5. Denklem (4.4) ile İfade Edilen FPAA-Tabanlı Chua Devre Modeli-IV' ün CAM Modülleri ve Parametre Değerleri.	96
Tablo 4.6. Kaotik Osilatörlerden Colpitts Osilatörün, Duffing Osilatörün ve Kennedy Osilatörün FPAA Gerçekleştirim Şemaları.	99
Tablo 4.7. Kaotik Osilatörlerden Lorenz Sisteminin, MLC Osilatörünün ve Rössler Sisteminin FPAA Gerçekleştirim Şemaları.	105
Tablo 4.8. Sprot Systemlerinden Model-I, Model-II, Model-III ve Model-IV' ün FPAA Gerçekleştirim Şemaları.	112
Tablo 4.9. Sprot Systemlerinden Model-V' in FPAA Gerçekleştirim Şeması.	113
Tablo 4.10. FPAA-Tabanlı 3-Çekerli Kaos Üretecinin CAM Modülleri ve Parametre Değerleri.	127
Tablo 4.11. FPAA-Tabanlı İki-Çift Çekerli Kaos Üretecinin CAM Modülleri ve Parametre Değerleri.	129

Tablo 4.12. FPAA-Tabanlı 5-Çekerli Kaos Üretecinin CAM Modülleri ve Parametre Değerleri.	129
Tablo 4.13. FPAA-Tabanlı 7-Çekerli Kaos Üretecinin CAM Modülleri ve Parametre Değerleri.	130
Tablo 5.1. AnadigmCHAOS Arayüzünde “ <i>Chaotic Oscillator</i> ” Kısımından Chua Osilatörünün Seçimi İçin Hazırlanan Kodlar.	160
Tablo 5.2. AnadigmCHAOS Arayüzünde “ <i>Function Type</i> ” Kısımından <i>Trigonometric</i> Fonksiyonunun Seçimi İçin Hazırlanan Kodlar.	161
Tablo 5.3. AnadigmCHAOS Arayüzünde <i>TEST</i> İşlemi İçin Hazırlanan Kodlar.	162

ŞEKİLLER LİSTESİ

	Sayfa No
Şekil 2.1. FPAA' nın Blok Diyagramı.	7
Şekil 2.2. Direncin Yerine Kullanılan Anahtarlama Kapasitör Devresi.	8
Şekil 2.3. (a) İntegral Alıcı Devre, (b) İntegral Alıcı Devrenin Anahtarlama Kapasitör Devre Eşdeğeri.	10
Şekil 2.4. İntegral Alıcı Devrenin Anahtarlama Zamanlama İşaretinin Şeması.	10
Şekil 2.5. Lee ve Gulak Tarafından Sunulan Bi-quadratic Filtre [48].	12
Şekil 2.6. Johns Hopkins Üniversitesi' ndeki FPAA Araştırma Grubu Tarafından Ortaya Konulan Tasarımdaki Yapılandırılabilir Analog Modül Yapısı.	12
Şekil 2.7. Motorola MPAA020' nin, (a) FPAA Yapısı, (b) FPAA CAB Yapısı.	14
Şekil 2.8. Zetex Firmasının Ürettiği TRAC020' nin Dizi Yapısı.	15
Şekil 2.9. ispPAC10' nun İç Yapısı.	16
Şekil 2.10. İşaret İyileştirme ve İşleme Uygulamalarında FPAA' nın Kullanımı [50].	18
Şekil 2.11. Radyo Alıcı Uygulamasında FPAA' nın Kullanımı [30].	18
Şekil 2.12. (a)AN221E04 Tipi FPAA' nın Blok Diyagramı, (b) CAB' in İç Yapısı.	19
Şekil 2.13. AN221K04-DVLP2 Uygulama Kartı.	21
Şekil 2.14. AN221K04-DVLP2 Uygulama Kartının İkili Kullanımı.	21
Şekil 2.15. AnadigmDesigner 2' de Oluşturulan Bir Sistemin Yüklenmesi ve Doğru Atlama Ayarları İçin Gerekli Olan Düzenlemeler.	22
Şekil 2.16. AnadigmDesigner 2 Yazılımının Arayüz Gösterimi.	23
Şekil 2.17. AnadigmDesigner 2 Yazılımında Kullanılan CAM Listesi.	24
Şekil 2.18. FPAA' nın Anahtarlama İşaretinin Parametrelerinin Ayarlandığı Arayüz Gösterimi.	24
Şekil 2.19. <i>SUMFILTER</i> Bloğun, (a) FPAA Yazılımındaki Gösterimi, (b) Anahtarlama Kapasitör Devre Yapısı.	25

Şekil 2.20. Eviren, Evirmeyen ve Eviren Girişe Sahip <i>SUMFILTER</i> Bloğun (a) FPAA Yazılım Programındaki Gösterimi, (b) Anahtarlamaalı Kapasitör Yapısı, (c) Eşdeğer Devre Yapısı.	26
Şekil 2.21. <i>GAININV</i> Bloğun, (a) FPAA Yazılımındaki Gösterimi, (b) Anahtarlamaalı Kapasitör Devre Yapısı.	27
Şekil 2.22. <i>RECTIFIERHALF</i> Bloğun, (a) FPAA Yazılımındaki Gösterimi, (b) Anahtarlamaalı Kapasitör Devre Yapısı.	28
Şekil 2.23. <i>MULTIPLIER</i> Bloğun, (a) FPAA Yazılımındaki Gösterimi, (b) Anahtarlamaalı Kapasitör Devre Yapısı.	29
Şekil 2.24. <i>OSCILLATORSINE</i> Bloğun, (a) FPAA Yazılımındaki Gösterimi, (b) Anahtarlamaalı Kapasitör Devre Yapısı.	30
Şekil 2.25. <i>VOLTAGE</i> Bloğun FPAA Yazılımındaki Gösterimi.	30
Şekil 2.26. <i>SUMDIFF</i> Bloğun, (a) FPAA Yazılımındaki Gösterimi, (b) Anahtarlamaalı Kapasitör Devre Yapısı.	31
Şekil 2.27. <i>TRANSFER FUNCTION</i> Bloğun, (a) FPAA Yazılımındaki Gösterimi, (b) Anahtarlamaalı Kapasitör Devre Yapısı.	32
Şekil 2.28. <i>FILTERBILINEAR</i> Bloğun FPAA Yazılımındaki Gösterimi.	33
Şekil 2.29. <i>FILTERBILINEAR</i> Bloğun Alçak Geçiren Filtre Tipindeki Anahtarlamaalı Kapasitör Devre Yapısı.	33
Şekil 2.30. <i>FILTERBILINEAR</i> Bloğun Yüksek Geçiren Filtre Tipindeki Anahtarlamaalı Kapasitör Devre Yapısı.	34
Şekil 2.31. <i>FILTERBILINEAR</i> Bloğun Tüm Geçiren Filtre Tipindeki Anahtarlamaalı Kapasitör Devre Yapısı.	35
Şekil 2.32. <i>FILTERBILINEAR</i> Bloğun Kutup ve Sıfır Noktalarına Sahip Filtre Tipindeki Anahtarlamaalı Kapasitör Devre Yapısı.	36
Şekil 2.33. AnadigmDesigner 2 Yazılımının Simülasyon Özelliğini Gösteren Birinci Örnek Uygulamanın FPAA Yazılımındaki Gösterimi ve Simülatör Çıktısı.	37
Şekil 2.34. AnadigmDesigner 2 Yazılımının Simülasyon Özelliğini Gösteren İkinci Örnek Uygulamanın FPAA Yazılımındaki Gösterimi ve Simülatör Çıktısı.	37
Şekil 3.1. Rössler Sistemine Ait Dallanma Diyagramı.	40
Şekil 3.2. Başlangıç Şartlarına Üstel Bağımlılığın Şematik Gösterimi.	41

Şekil 3.3.	Rössler Sistemine Ait Lypunov Üstellerinin Çizimi.	41
Şekil 3.4.	Rössler Sisteminin Benzetiminden Elde Edilen $x(t)$ Dinamiğinin, (a) Zaman Domeninde Gösterimi, (b) Kaotik Frekans Spektrumu.	42
Şekil 3.5.	Rössler Sisteminin Benzetiminden Elde Edilen Faz Düzlemindeki Gösterimi.	43
Şekil 3.6.	Lojistik Map' in Dallanma Diyagramı.	44
Şekil 3.7.	Poincare Kesit Gösterimi.	44
Şekil 3.8.	Kaotik Chua Osilatörünün SIMULINK™ Modeli.	48
Şekil 3.9.	Kaotik Chua Osilatörünün SIMULINK™ Modelinden Elde Edilen Kaotik Çeker ve DC Karakteristik Sonuçları, (a) NR1 Bloğu Sistemde, (b) NR2 Bloğu Sistemde, (c) NR3 Bloğu Sistemde. Nümerik Analizde Kullanılan Sistem Parametreleri, $\alpha=10$ ve $\beta=14.87$ dir.	50
Şekil 3.10.	Trigonometrik Doğrusal Olmayan Fonksiyon Kullanılarak Gerçekleştirilen Kaotik Chua Osilatörünün SIMULINK™ Modeli.	51
Şekil 3.11.	Trigonometrik Doğrusal Olmayan Fonksiyon Kullanılarak Gerçekleştirilen Kaotik Chua Osilatörünün SIMULINK™ Analiz Sonuçlarındaki Kaotik Çeker Gösterimi ve DC Karakteristiği.	51
Şekil 3.12.	(a) Chua Osilatörünün Devre Şeması, (b) Chua Osilatöründeki Doğrusal Olmayan Direncin Üç Parçalı Doğrusal Karakteristiği.	52
Şekil 3.13.	Chua Osilatörünün Devre Şeması.	53
Şekil 3.14.	CFOA' nın Fonksiyonel Blok Diyagramı.	54
Şekil 3.15.	Chua Osilatörünün Devre Yapısındaki, (a) Doğrusal Olmayan Direnç, (b) İndüktör Simülatörü.	54
Şekil 3.16.	CFOA-Tabanlı İndüktörsüz Chua Sisteminin Deneysel Gerçekleştiriminden Elde Edilen, (a) Kaotik Devre Dinamikleri: Üstteki İşaret V_{C2} (1 V/div), Altteki İşaret V_{C1} (2 V/div), Time/div:2 ms/div, (b) $(V_{C2}-V_{C1})$ Düzlemindeki Çift Çeker Yapısı, x-Ekseni:500 mV, y-Ekseni:1 V.	55
Şekil 3.17.	Kaotik Colpitts Osilatörün SIMULINK™ Modeli.	57
Şekil 3.18.	Kaotik Colpitts Osilatörün Devre Şeması.	57

- Şekil 3.19. (i) Colpitts Osilatörün Donanım Gerçekleştiriminden Elde Edilen, (a) (i) Kaotik Devre Dinamikleri: Üstteki İşaret V_{C1} (2 V/div), Alttaki İşaret V_{C2} (1 V/div), (b) (i) $(V_{C1}-V_{C2})$ Düzlemindeki Çeker Gösterimi, (ii) Kaotik Colpitts Osilatörünün SIMULINK™ Modelinden Elde Edilen Nümerik Analiz Sonuçları, (a) (ii) Kaotik Dinamikleri, (b) (ii) $(x-z)$ Düzlemindeki Çeker Gösterimi. 58
- Şekil 3.20. Duffing Osilatörün SIMULINK™ Modeli. 59
- Şekil 3.21. Duffing Osilatörün İndüktörsüz Devre Gerçekleştirimi. 60
- Şekil 3.22. (i) Duffing Osilatörün Donanım Gerçekleştiriminden Elde Edilen, (a) (i) Kaotik Devre Dinamikleri: Üstteki İşaret V_{C1} (0.4 V/div), Alttaki İşaret V_{C2} (0.4 V/div), (b) (i) $(V_{C1}-V_{C2})$ Düzlemindeki Çeker Gösterimi, (ii) Duffing Osilatörün SIMULINK™ Modelinden Elde Edilen Nümerik Analiz Sonuçları, (a) (ii) Kaotik Dinamikleri, (b) (ii) $(x-y)$ Düzlemindeki Çeker Gösterimi. 61
- Şekil 3.23. Kennedy Osilatörün SIMULINK™ Modeli. 62
- Şekil 3.24. Kennedy Osilatörün Devre Şeması. 63
- Şekil 3.25. (i) Kennedy Osilatörün Donanımsal Gerçekleştiriminden Elde Edilen, (a) (i) Kaotik Devre Dinamikleri: Üstteki İşaret V_{C1} (4 V/div), Alttaki İşaret V_{C2} (2 V/div), (b) (ii) $(V_{C1}-V_{C2})$ Düzlemindeki Çeker Gösterimi, (ii) Kennedy Osilatörün SIMULINK™ ile Modellemesinden Elde Edilen Nümerik Analiz Sonuçları, (a) (ii) Kaotik Dinamikleri, (b) (ii) $(x-y)$ Düzlemindeki Çeker Gösterimi. 64
- Şekil 3.26. Lorenz Sistemin SIMULINK™ Modeli. 65
- Şekil 3.27. (a) Ölçeklendirilmiş Lorenz Sisteminin Devre Şeması. Analog Çarpıcılar M_1 ve M_2 , Akım Çıkış Elemanı Olarak Kullanılmışlardır, (b) Akım Çıkış Elemanı Olarak Kullanılan M_1 Analog Çarpıcının İç Yapısı. 66
- Şekil 3.28. (i) Lorenz Sisteminin Donanım Gerçekleştiriminden Elde Edilen, (a) (i) Kaotik Devre Dinamikleri: Üstteki İşaret V_u (2 V/div), Alttaki İşaret V_v (2 V/div), (b) (i) (V_u-V_v) Düzlemindeki Çeker 67

Gösterimi, (ii) Lorenz Sistemin SIMULINK™ Modelinden Elde Edilen Nümerik Analiz Sonuçları, (a) (ii) Kaotik Dinamikleri, (b) (ii) $(x-y)$ Düzlemindeki Çeker Gösterimi.	
Şekil 3.29. MLC Osilatörün SIMULINK™ Modeli.	68
Şekil 3.30. MLC Osilatörü.	69
Şekil 3.31. (i) MLC Osilatör Devresinin (a) (i) Kaotik Devre Dinamikleri: Üstteki İşaret V_{Cl} (2 V/div), Altındaki İşaret i_L (100 Ω ' luk Direnç İle) (500 mV/div), Time/div: 500 μ s/div, (b) (i) $(V_{Cl}-i_L)$ Düzlemindeki Çift Çeker Yapısı, X-Ekseni:1 V, Y-Ekseni:500 mV, (ii) MLC Osilatörün SIMULINK™ Modelinden Elde Edilen Nümerik Analiz Sonuçları, (a) (ii) Kaotik Dinamikleri, (b) (ii) $(x-y)$ Düzlemindeki Çeker Gösterimi.	70
Şekil 3.32. Rössler Sisteminin SIMULINK™ Modeli.	71
Şekil 3.33. Modifiye Edilen Rössler Sistemi.	72
Şekil 3.34. (i) Rössler Sisteminin Deneysel Çalışmadan Elde Edilen, (a) (i) Kaotik Devre Dinamikleri: Üstteki İşaret x (2 V/div), Altındaki İşaret y (2 V/div), (b) (i) $(x-y)$ Düzlemindeki Çeker Gösterimi, (ii) Rössler Sisteminin SIMULINK™ Modelinden Elde Edilen Nümerik Analiz Sonuçları, (a) (ii) Kaotik Dinamikleri, (b) (ii) $(x-y)$ Düzlemindeki Çeker Gösterimi.	73
Şekil 3.35. Sprot Systeminin SIMULINK™ Modeli.	75
Şekil 3.36. Sprot Systeminin (a) Devre Şeması, (b) Doğrusal Olmayan Fonksiyona Ait Devre Bloğu.	76
Şekil 3.37. Deneysel Olarak Gerçekleştirilen Sprot Systeminin, (a) Kaotik Devre Dinamikleri: Üstteki İşaret u (2 V/div), Altındaki İşaret v (200 mV/div), Time/div:2 ms/div, (b) $(u-v)$ Düzlemindeki Çift Çeker Yapısı, x-Ekseni:2 V, y-Ekseni:200 mV, (c) NR2 Bloğu Sistemde İken Sprot Systeminin SIMULINK™ Modellemesinden Elde Edilen $(u-v)$ Düzlemindeki Çeker Gösterimi.	77
Şekil 3.38. HSA Modelinde Hücre Bağlantı Şeması.	79
Şekil 3.39. HSA-Tabanlı Chua Osilatörünün SIMULINK™ Modeli.	79
Şekil 3.40. HSA-Tabanlı Chua Osilatörünün Devre Şeması.	80

Şekil 3.41. (i) HSA-Tabanlı Chua Osilatörünün Donanım Gerçekleştiriminden Elde Edilen, (a) (i) Kaotik Devre Dinamikleri: Üstteki İşaret x_1 (1 V/div), Alttaki İşaret x_2 (500 mV/div), Time/div:50 μ s/div, (b) (i) (x_1-x_2) Düzlemindeki Çift Çeker Yapısı, x-Ekseni:1 V, y-Ekseni:500 mV, (ii) HSA-Tabanlı Chua Osilatörünün SIMULINK™ Modelinden Elde Edilen Nümerik Analiz Sonuçları, (a) (ii) Kaotik Dinamikleri, (b) (ii) $(x-y)$ Düzlemindeki Çeker Gösterimi.	81
Şekil 4.1. FPAA-Tabanlı Tasarım ve Gerçekleştirim Diyagramı.	84
Şekil 4.2. FPAA-Tabanlı Gerçekleştirim Deneysel Kurulumu. Bu Kurulumda Anadigm Firmasının AN221E04 Tipi FPAA [6] Kullanılmıştır.	85
Şekil 4.3. <i>TRANSFER FUNCTION</i> Bloкта Elde Edilen Parçalı Doğrusal Fonksiyonun Grafiği.	89
Şekil 4.4. FPAA-Tabanlı Chua Devre Modeli-I' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (1 V/div), Alttaki İşaret y (0.4 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.88$ İken Elde Edilen Tek Çeker, (d) $\mu=0.8$ İken Elde Edilen Periyot 2 Davranışı.	91
Şekil 4.5. FPAA-Tabanlı Chua Devre Modeli-II' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (0.4 V/div), Alttaki İşaret y (0.4 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.833$ İken Elde Edilen Tek Çeker, (d) $\mu=0.708$ İken Elde Edilen Periyot 1 Davranışı.	93
Şekil 4.6. <i>TRANSFER FUNCTION</i> Bloкта Elde Edilen $[\tanh(.)]$ Fonksiyonun Grafiği.	94
Şekil 4.7. FPAA-Tabanlı Chua Devre Modeli-III' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (1 V/div), Alttaki İşaret y (0.4 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.92$ İken Elde Edilen Tek Çeker, (d) $\mu=0.825$ İken Elde Edilen Periyot 1	95

- Davranışı.
- Şekil 4.8. FPAA-Tabanlı Chua Devre Modeli-IV' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (0.4 V/div), Altındaki İşaret y (0.2 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.946$ İken Elde Edilen Tek Çeker, (d) $\mu=0.898$ İken Elde Edilen Periyot 1 Davranışı. 98
- Şekil 4.9. *TRANSFER FUNCTION* Bloкта Elde Edilen $f(z)$ Fonksiyonun Grafiği. 100
- Şekil 4.10. FPAA-Tabanlı Kaotik Colpitts Osilatörden, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (1 V/div), Altındaki İşaret y (1 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=5.3$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=7$ İken Elde Edilen Periyot 1 Davranışı. 101
- Şekil 4.11. FPAA-Tabanlı Duffing Osilatörden, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (1 V/div), Altındaki İşaret y (0.4 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=1.75$ İken Elde Edilen Limit Çevrim Davranışı. 103
- Şekil 4.12. FPAA-Tabanlı Kennedy Osilatörden, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (1 V/div), Altındaki İşaret y (0.4 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.62$ İken Elde Edilen Tek Çeker, (d) $\mu=0.57$ İken Elde Edilen Periyot 1 Davranışı. 104
- Şekil 4.13. FPAA-Tabanlı Lorenz Sisteminden, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (0.4 V/div), Altındaki İşaret y (0.4 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ ve $(x-z)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.828$ İken Elde Edilen Tek Çeker, (d) $\mu=0.825$ İken Elde Edilen Periyot 1 Davranışı. 107
- Şekil 4.14. FPAA-Tabanlı MLC Osilatörden, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (0.2 V/div), Altındaki

- İşaret y (0.2 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=1.3$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=1.35$ İken Elde Edilen Periyot 1 Davranışı.
- Şekil 4.15. FPAA-Tabanlı Rössler Osilatörden, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (0.4 V/div), Alttaki İşaret y (0.4 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=3$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=3.5$ İken Elde Edilen Periyot 1 Davranışı. 111
- Şekil 4.16. FPAA-Tabanlı Sprot System Modeli-I' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret u (0.2 V/div), Alttaki İşaret v (0.2V/div), (b) $\mu=1$ İken Elde Edilen $(u-v)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.875$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=0.75$ İken Elde Edilen Periyot 1 Davranışı. 114
- Şekil 4.17. FPAA-Tabanlı Sprot System Modeli-II' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret u (1 V/div), Alttaki İşaret v (1V/div), (b) $\mu=1$ İken Elde Edilen $(u-v)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.675$ İken Elde Edilen Periyot 5 Davranışı, (d) $\mu=0.65$ İken Elde Edilen Periyot 1 Davranışı. 116
- Şekil 4.18. FPAA-Tabanlı Sprot System Modeli-III' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret u (0.4 V/div), Alttaki İşaret v (0.4V/div), (b) $\mu=1$ İken Elde Edilen $(u-v)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.88$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=0.8$ İken Elde Edilen Periyot 1 Davranışı. 117
- Şekil 4.19. FPAA-Tabanlı Sprot System Modeli-IV' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret u (1 V/div), Alttaki İşaret v (0.4V/div), (b) $\mu=1$ İken Elde Edilen $(u-v)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.6$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=0.5$ İken Elde Edilen Periyot 1 Davranışı. 119

- Şekil 4.20. FPAA-Tabanlı Sprott Sistem Modeli-V' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret u (1 V/div), Alttaki İşaret v (0.4V/div), (b) $\mu=1$ İken Elde Edilen $(u-v)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.74$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=0.7$ İken Elde Edilen Periyot 1 Davranışı. 120
- Şekil 4.21. HSA-Tabanlı Chua Devresinin FPAA Gerçekleştirim Şeması. 121
- Şekil 4.22. FPAA Kullanılarak Gerçekleştirilen HSA-Tabanlı Chua Devresinin, (a) Kaotik Devre Dinamikleri: Üstteki İşaret x_1 (1 V/div), Alttaki İşaret x_2 (1V/div), (b) (x_1-x_2) Düzlemindeki Çeker Gösterimi. 122
- Şekil 4.23. HSA-Tabanlı MLC Devresinin FPAA Gerçekleştirim Şeması. 124
- Şekil 4.24. FPAA Kullanılarak Gerçekleştirilen HSA-Tabanlı MLC Devresinin, (a) Kaotik Devre Dinamikleri: Üstteki İşaret x_1 (0.2 V/div), Alttaki İşaret x_2 (0.2V/div), (b) (x_1-x_2) Düzlemindeki Çeker Gösterimi. 125
- Şekil 4.25. İki-Çift Çekerli Yapı İçin *TRANSFER FUNCTION* Bloкта Elde Edilen $h(x)$ Fonksiyonun Grafiği. 126
- Şekil 4.26. FPAA-Tabanlı n -Çekerli Kaos Üretici. 127
- Şekil 4.27. (a) FPAA-Tabanlı 3-Çekerli Kaos Üreticinin $(x-y)$ Düzlemindeki Çeker Gösterimi, (b) FPAA-Tabanlı İki-Çift Çekerli Kaos Üreticinin $(x-y)$ Düzlemindeki Çeker Gösterimi. 128
- Şekil 4.28. (a) FPAA-Tabanlı 5-Çekerli Kaos Üreticinin $(x-y)$ Düzlemindeki Çeker Gösterimi, (b) FPAA-Tabanlı 7-Çekerli Kaos Üreticinin $(x-y)$ Düzlemindeki Çeker Gösterimi. 130
- Şekil 4.29. Faz Kilitlemeli Döngü Şeması. 131
- Şekil 4.30. PLL Modelinin FPAA Gerçekleştirim Şeması. 132
- Şekil 4.31. FPAA-Tabanlı PLL Modelden, (a) $\mu=6.66$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret φ (1 V/div), Alttaki İşaret y (1 V/div), (b) $\mu=6.66$ İken Elde Edilen $(\varphi-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=5.55$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=5$ İken Elde Edilen Periyot 1 Davranışı. 134

Şekil 4.32. FPAA-Tabanlı Senkronizasyon Gerçekleştirmeleri İçin Deneysel Kurulum. Bu Kurulumlarda Anadigm Firmasının AN221E04 Tipi FPAA [6] Kullanılmıştır.	135
Şekil 4.33. Chua Devresinin Kullanıldığı Kaskad Bağlantı Yöntemi İle Yapılan Senkronizasyonda Kullanılan Blok Diyagram.	136
Şekil 4.34. FPAA-Tabanlı Chua Devresinin Kullanıldığı Kaskad Bağlantı Yöntemi İle Yapılan Senkronizasyon Gerçekleştirim Şeması.	136
Şekil 4.35. FPAA-Tabanlı Chua Devresinin Kullanıldığı Kaskad Bağlantı Yöntemi İle Yapılan Senkronizasyon Gerçekleştiriminden Elde Edilen (a) Kaotik Devre Dinamikleri: Üstteki İşaret x (1 V/div), Altındaki İşaret x'' (1 V/div), (b) $(x-x'')$ Gösterimleri.	137
Şekil 4.36. Chua Devresinin Kullanıldığı Pecora&Carroll Yöntemi İle Yapılan Senkronizasyonda Kullanılan Blok Diyagram.	138
Şekil 4.37. FPAA-Tabanlı Chua Devresinin Kullanıldığı Pecora&Carroll Yöntemi İle Yapılan Senkronizasyon Gerçekleştirim Şeması.	139
Şekil 4.38. FPAA-Tabanlı Chua Devresinin Kullanıldığı Pecora&Carroll Yöntemi İle Yapılan Senkronizasyon Gerçekleştiriminden Elde Edilen (a) Kaotik Devre Dinamikleri: Üstteki İşaret x (1 V/div), Altındaki İşaret x'' (1 V/div), (b) $(x-x'')$ Gösterimleri.	140
Şekil 4.39. FPAA-Tabanlı Chua Devresinin Kullanıldığı Kuplajlama Yöntemi İle Yapılan Senkronizasyon Gerçekleştirim Şeması.	141
Şekil 4.40. FPAA-Tabanlı Chua Devresinin Kullanıldığı Kuplajlama Yöntemi İle Yapılan Senkronizasyon Gerçekleştiriminden Elde Edilen, (a) $x(t)$ Kaotik Devre Dinamiği (b) $x'(t)$ Kaotik Devre Dinamiği, (c) $(x-x')$ Gösterimleri.	142
Şekil 4.41. FPAA-Tabanlı MLC Sisteminin Kullanıldığı Pecora&Carroll Yöntemi İle Yapılan Senkronizasyon Gerçekleştirim Şeması.	143
Şekil 4.42. FPAA-Tabanlı MLC Sisteminin Kullanıldığı Pecora&Carroll Yöntemi İle Yapılan Senkronizasyon Gerçekleştiriminden Elde Edilen (a) Kaotik Devre Dinamikleri: Üstteki İşaret y (0.2 V/div), Altındaki İşaret y' (0.2V/div), (b) $(y-y')$ Gösterimleri.	144
Şekil 4.43. MLC Sisteminin Kullanıldığı Tek Yönlü Kuplajlama Yöntemi	145

	İle Yapılan Senkronizasyonda Kullanılan Blok Diyagram.	
Şekil 4.44.	FPAA-Tabanlı MLC Sisteminin Kullanıldığı Tek Yönlü Kuplajlama Yöntemi İle Yapılan Senkronizasyon Gerçekleştirim Şeması.	146
Şekil 4.45.	FPAA-Tabanlı MLC Sisteminin Kullanıldığı Tek Yönlü Kuplajlama Yöntemi İle Yapılan Senkronizasyon Gerçekleştiriminden Elde Edilen (a) Kaotik Devre Dinamikleri: Üstteki İşaret x (0.2 V/div), Alttaki İşaret x' (0.2 V/div), (b) $(x-x')$ Gösterimleri.	146
Şekil 5.1.	AnadigmDesigner 2 Programında <i>AnadigmFilter</i> Yardımcı Aracının Seçimi.	149
Şekil 5.2.	<i>AnadigmFilter</i> Aracının Ekran Görünümü.	150
Şekil 5.3.	AnadigmFilter Aracında Tasarlanan Filtrenin AnadigmDesigner 2 Arayüz Ortamına Aktarılması.	151
Şekil 5.4.	Blok Diyagram Sekmesi Seçildiğindeki AnadigmPID Ekranı.	152
Şekil 5.5.	<i>AnadigmPID</i> Aracında Yer Alan, (a) Sistem Ekran Görünümü, (b) Filtre Ekran Görünümü, (c) Özellikler Ekran Görünümü.	153
Şekil 5.6.	Giriş Sekmesinde Ayarlamalar Yapılarak Oluşturulan PID Tasarımın Blok Diyagram Ekranı.	154
Şekil 5.7.	<i>AnadigmPID</i> ' de Tasarlanan Kontrolörün AnadigmDesigner 2' ye Aktarımı.	155
Şekil 5.8.	AnadigmCHAOS Yardımcı Aracın Arayüzü.	157
Şekil 5.9.	Örnek Olarak Trigonometrik Fonksiyona Sahip Chua Osilatörün Seçilmesi.	159
Şekil 5.10.	Arayüzde Trigonometrik Fonksiyona Sahip Chua Osilatörün Mevcut Parametreleri ile Nümerik Analizinden Elde Edilen Sonuçlar.	163
Şekil 5.11.	Arayüzde Trigonometrik Fonksiyona Sahip Chua Osilatördeki (k) Parametresi Değiştirilerek Yapılan Nümerik Analizden Elde Edilen Tek Çeker Kaotik Dinamikleri ve Çeker Gösterimi.	164
Şekil 5.12.	Seçilen Trigonometrik Fonksiyona Sahip Chua Osilatörün AnadigmDesigner 2' ye Aktarılması.	165

Şekil 5.13. Arayüzde Lorenz Osilatörün Mevcut Parametreleri ile Nümerik Analizinden Elde Edilen Sonuçlar.	166
Şekil 5.14. Arayüzde Lorenz Sistemindeki d Parametresi Değiştirilerek Yapılan Nümerik Analizden Elde Edilen Sonuçlar.	167
Şekil 5.15. Seçilen Lorenz Sisteminin AnadigmDesigner 2' ye Aktarılması.	168

1. BÖLÜM

GİRİŞ

1.1. Giriş

20. yüzyılın ilk yarısında yapılan çalışmalar neticesinde ortaya konan ilk yarı iletken transistör elektronik sistemlerin tasarımında yeni bir çığır açmıştır. 1980'lerden itibaren CMOS VLSI teknolojisi hızla gelişmiş ve bu teknoloji sayesinde, ayrık yarı iletken elemanlar yerine milyonlarca transistör küçük bir silikon tümdevre içerisine yerleştirilmiş ve bu tümdevreler elektronik devre tasarımında kullanılmıştır. Küçük hacmine rağmen içerisinde çok sayıda transistör barındıran bu tümdevreler, özellikle sayısal sistem tasarımında tasarımcılara büyük kolaylık sağlamıştır. Bugün gelinen noktada üretilen tümdevreler, daha ziyade sayısal sistem tasarımında kullanılırken analog sistem tasarımına yönelik tümdevre teknolojisi geriden gelmektedir. Bu durumda analog sistemlerin yerini dijital sistemlerin alması gerekliliği doğmuyor mu? Başka bir soru; analog tümdevre teknolojisinin sayısal sistem teknolojisine kıyasla geriden gelmesine rağmen neden analog tasarımlar halen daha büyük talep alıyorlar?

Bu sorunun cevabı aslında çok basittir: Her şeye rağmen gerçek dünyanın analog olmasıdır. Ses, ışık, sıcaklık, durum, hız, basınç vb. gibi fiziksel büyüklüklerin hepsi zamanla değişebilen “analog” büyüklüklerdir. Analog devreler, dijital devrelerle birlikte analog dünyanın işlenmesinde oldukça önemli bir rol oynar [1]. Buna ek olarak analog sistemler, düşük güçlü ve yüksek hızlı işaret işleme uygulamalarında dijital devrelerle bir rekabet içerisinde [2, 3]. Dijital devreler analog eşdeğerleriyle karşılaştırıldığında daha büyük esnekliğe, daha kolay tasarım süreçlerine ve doğal olarak dijital sistemlerle iyi bir uyuma sahiptir. Buna karşın dijital sistemler, birçok uygulamada sensör ve uyarıcı arabağlaşım devrelerine veya A/D veya D/A dönüştürücülere ihtiyaç duyarlar. Ayrıca dijital sistemler, daha karmaşık bir devre yapısına sahip olup daha büyük kırımlık alanına da ihtiyaç duyarlar. Daha büyük kırımlık alanı, analog eşdeğerlerine göre haliyle

daha yavaş bir çalışmayı ve daha fazla güç tüketimini de beraberinde getirmektedir. Öte yandan analog sistemlerde otomasyon esnekliğinin az olması, gerekli olan eleman çeşitliliği, analog sinyallerin değişik seviyelerde ve sürekli zamanda olması ve yüksek doğrulukta devre modellemesinin gerekliliği analog sistemler için dezavantaj teşkil etmektedir [2].

Analog sistemler için alan programlanabilir analog diziler (**Field Programmable Analog Array, FPAA**), dijital sistemler için alan programlanabilir kapı dizileri (**Field Programmable Gate Array, FPGA**) yukarıda bahsedilen problemleri ortadan kaldırmak için alternatif donanımlar olarak karşımıza çıkmaktadır. Esnek mimarileri, yüksek hata toleransları, sıcaklığa karşı kararlılıkları, devre karmaşasını ortadan kaldırmaları, yeniden programlanabilir olmaları, bu donanımların tercih edilmelerinde önemli etkenlerdir [4-6].

FPAA' lar, tasarım süreçleri uzun olan analog devrelerin tasarımını hızlandırmak ve ayrıca hem analog hem de hibrid (analog-dijital) devrelerin tasarımında kararlılık, doğruluk, düşük fiyat, hızlı prototip teknikleri sunduğu için programlanabilir analog sistemler üzerine yapılan araştırmaların kilit noktasını oluşturmuştur [7, 8]. Alan programlanabilir analog dizi teknolojisi ilk olarak 1980'li yıllarda ortaya çıkmıştır. Günümüzde FPAA' lar, FPGA' nın analog eşdeğeri olarak kabul edilmektedir [9-11].

FPAA' lar, temel analog elemanları içeren ve farklı analog fonksiyonları gerçekleştirmek için elektriksel olarak yapılandırılabilen yarı iletken tümdevrelerdir. FPAA' lar programlanabilirlik ve yeniden yapılandırılabilirlik özellikleri sayesinde tasarımcıya, donanım çalışırken bile tasarım üzerinde değişiklik yapma imkanı sunmaktadır. Diğer bir deyişle FPAA' lar bu özellikleri sayesinde herhangi bir analog devrenin prototipinin hızlı ve kolay bir şekilde gerçekleştirilmesini sağlamaktadır [12].

Literatürde FPAA' lar, bireysel ve ticari olarak gerçekleştirilen FPAA' lar olmak üzere ikiye ayrılmaktadır. Bireysel olarak gerçekleştirilen FPAA' lar daha çok işaret işleme uygulamalarında kullanılmışlardır [13-22]. Ticari olarak gerçekleştirilen FPAA' lar [6, 23-26], işaret işleme, filtre uygulamaları, radyo dalgalarıyla kimliklendirme, tıp, telekomünikasyon, kaotik osilatörler gibi birçok farklı alana uygulanmışlardır [6, 8, 10, 27-40]. Bu uygulama alanlarından kaotik osilatörler, kaos ve kaotik sistemlerin fiziksel

olarak pratik hayata geçirilmesinde önemli rol oynarlar. Kaotik işaret üreten devre modelleri, kaos tabanlı uygulamalarının en önemli tasarım aşamasını oluşturmaktadır. Çünkü bu uygulamalarda kaos dinamiklerinin özelliklerinden yararlanmak ve bu özellikleri fiziki hayata geçirebilmek için öncelikle bu dinamiklerin pratik olarak üretilmesi gerekmektedir. Literatürde özellikleri çok iyi bilinen ve avantajlı bir şekilde mühendislik uygulamalarında kullanılabilen kaotik işaret üreteçlerinin sayısı azdır. Bu konuda yeni ve özellikli kaotik işaret üreteçlerinin literatüre kazandırılması gerekliliği vardır. Özellikle son yıllarda elektronik devre tasarımında esnek tasarım ve otomasyon alanında kullanılan yeni devre tekniklerinin kaotik işaret üreteç tasarımına da uygulanması, kaotik devre modellerini çok daha esnek ve kullanışlı kılacaktır. Bu amaç doğrultusunda kaotik osilatör yapılarının deneysel olarak yeniden yapılandırılabilir ve programlanabilir tasarımını gerçekleştirmek için FPAA' lar uygun bir çözüm noktasıdır.

Literatürde FPAA' lar kullanılarak gerçekleştirilen çok az sayıda kaotik osilatör bulunmaktadır. FPAA' lar ile kaotik osilatör tasarımı ilk olarak Capanotte ve arkadaşları tarafından yapılmış olup, bu çalışmada HSA-tabanlı Chua devresinin tasarımının ve senkronizasyon uygulamasının FPAA üzerinde gerçekleştiriminden bahsetmişlerdir [32]. Sonraki çalışmada ise, Hulub ve arkadaşları 3x3' lük çekere sahip kaotik devrelerin FPAA' lar ile gerçekleştirimi ve senkronizasyonu üzerinde durmuşlardır [33]. Diğer bir çalışma da Arena ve arkadaşları tarafından ortaya konmuştur. Çalışmalarında farklı kaotik devrelerin senkronizasyonu ve bu devreler tarafından üretilen kaotik işaretlerin ayrıştırılması üzerinde durmuşlardır [34]. Bu deneysel çalışmada FPAA, farklı kaotik devreleri gerçeklemede kullanılmıştır. Literatürde kaotik osilatör devrelerinin FPAA ile gerçekleşmesi konusunda bu çalışmaların dışında bu tez çalışması esnasında yapılan çalışmalar bulunmaktadır. Bu çalışmalardan biri, Sprott tarafından ortaya konulan farklı doğrusal olmayan fonksiyonlara sahip kaotik sistemin FPAA' da gerçekleştirimi üzerinedir [35, 36]. Bir diğer çalışma otonom olmayan MLC devresinin FPAA üzerinde gerçekleştirimi [37] olup, başka bir çalışma da, Chua devresinin farklı doğrusal olmayan bloklar ile FPAA' da gerçekleştirimi üzerinedir [38]. En son çalışma ise hücreli sinir ağı tabanlı MLC devresinin FPAA üzerinde gerçekleştirimini içermektedir [39].

1.2. Tezin Amacı

Bu tez çalışmasının amacı, kaotik osilatör devrelerinin alan programlanabilir analog dizi elemanları kullanılarak tasarlanabileceğini göstermek ve bu sayede programlanabilir ve yeniden yapılandırılabilir kaotik osilatör yapılarını ortaya koymaktır. Ayrıca alan programlanabilir analog diziler kullanılarak gerçekleştirilen kaotik osilatörlerin birbirleri arasında senkronizasyonun sağlanabileceği düşüncesinin pratik olarak uygulanabilmesi için çalışmalar planlanmıştır. Bahsedilen hedefler doğrultusunda kaotik osilatörlerden literatürde üzerinde en çok çalışılanlarının, FPAA tabanlı olarak gerçekleştirilmesi amaçlanmıştır. FPAA' ların yeniden programlanabilmeleri, kısa süre içerisinde devre parametrelerinin değiştirilmesine ve kaotik devrelerin yeni parametre değerlerine uygun çıkış üretmesine imkan sağlamaktadır. Bunun yanı sıra FPAA tabanlı iki kaotik devre arasında senkronizasyon çalışmalarının yapılması planlanmıştır. Ayrıca kaotik osilatörlerle ilgilenen araştırmacıların, bilim adamlarının ve öğrencilerin kullanabileceği faydalı bir eğitim kaynağı olarak AnadigmCHAOS adlı bir arayüz uygulamasının geliştirilmesi düşünülmüştür.

Bu tez çalışmasının ikinci bölümünde alan programlanabilir analog diziler hakkında detaylı bilgi verilmiştir.

Üçüncü bölümde kaos ile ilgili kavramlar örneklerle ele alınmıştır. Daha sonra literatürde yer alan kaotik osilatör tiplerinden bahsedilmiştir. Ayrıca bu osilatör tiplerinin kaotik davranışları nümerik analiz yöntemi ile incelenmiş ve bu devrelerin ayrı elemanlarla deneysel gerçekleştirmeleri yapılmıştır.

Dördüncü bölümde, programlanabilirlik ve yeniden yapılandırılabilirlik üzerinde durulmuş, FPAA tasarım prosedüründen bahsedilmiştir. Bir önceki bölümde nümerik analizi yapılan kaotik osilatör devrelerinin FPAA-tabanlı tasarım ve gerçekleştirmeleri üzerinde durulmuştur. Tasarlanan devreler, elde edilen deneysel sonuçları ile beraber verilmiştir. Ayrıca sadece osilatör yapılarının değil bununla birlikte içerisinde osilatörlerin de olduğu daha karmaşık yapıların FPAA-tabanlı tasarımının mümkün olabileceğini göstermek için faz kilitlemeli döngü modelinin FPAA' da tasarımı incelenmiştir. Bunların yanı sıra FPAA-tabanlı kaotik osilatör devrelerinin senkronizasyon uygulamaları incelenmiş ve deneysel çalışmaları yapılmıştır.

Beşinci bölümde Anadigm firmasının FPAA tasarımı için geliştirilen AnadigmDesigner 2 programından ve bu programda yer alan *AnadigmFilter* ve *AnadigmPID* yardımcı araçlarından bahsedilmiştir. Bu yardımcı araçlar arasında yer alabilecek ve kaotik sistem tasarımıyla uğraşan araştırmacılara ve bu konuyla ilgili eğitim gören lisans ve lisansüstü öğrencilerine yönelik AnadigmCHAOS isimli bir yazılım uygulaması hazırlanmıştır. Bu uygulama kaotik osilatör tasarımı ile ilgili olup bu bölümde hazırlanan arayüz tanıtılmıştır.

Tezin altıncı ve son bölümünde ise elde edilen sonuçlar tartışılmıştır.

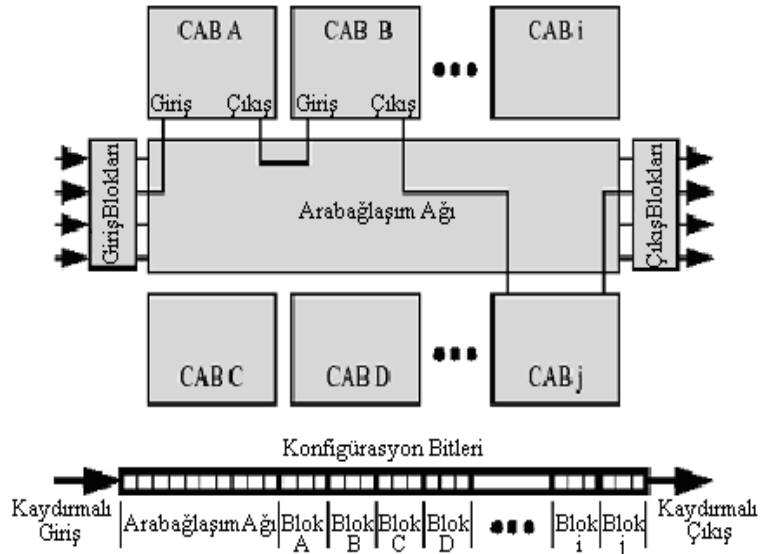
2. BÖLÜM

ALAN PROGRAMLANABİLİR ANALOG DİZİLER

2.1. Alan Programlanabilir Analog Dizi (FPAA)

Elektronik sistemler analog, sayısal veya hem analog hem de sayısal devrelerden oluşmaktadır. Alan programlanabilir dizilerin geliştirilmesiyle birlikte çok büyük ölçekli analog veya sayısal devrelerin tasarımları daha da kolaylaşmıştır. Bünyesinde hem analog hem de sayısal modüller bulunduran büyük bir sistemde, sayısal devrelerin programlanabilme özelliği alan programlanabilir kapı dizileri ile sağlanabilmektedir. Bununla birlikte tüm sistemin programlanabilmesi ve yeniden yapılandırılabilmesi için analog devrelerin de programlanmasına ihtiyaç duyulmaktadır. Analog devre tasarımının pahalı olması ve çok uzun zaman almasından dolayı bu faktörlerin tasarım maliyetlerine etkilerinin azaltılması gereksinimi de doğmaktadır. Bu noktada alan programlanabilir analog diziler (Field Programmable Analog Array, FPAA) iyi bir alternatif çözüm olarak ortaya çıkmıştır [6, 7, 14, 16, 20, 23-26, 30].

Şekil 2.1’ de görüldüğü gibi bir FPAA elemanı, yapılandırılabilir analog bloklardan (Configurable Analog Blocks, CAB) ve programlanabilir ara bağlaşım ağından oluşmaktadır. Konfigürasyon bit dizisi, kaydırmalı kaydedicide (shift register) saklanır. Bit dizisindeki bazı bitler, arabağlaşım ağının bağlanabilirliğini ayarlama da kullanılır. Bu bitlerden bazıları da yapılandırılabilir analog blokların fonksiyonelliğini programlamada kullanılırlar. Yapılandırılabilir analog bloklar, kuvvetlendirme, integral alma, toplama, filtreleme, karşılaştırma ve analog/dijital veya dijital/analog dönüştürme gibi işlemleri gerçekleştirmek için programlanabilirler [7, 41-43]. Ara bağlaşım ağı ise, yapılandırılabilir analog blokların içerisindeki ve birbiriyle olan bağlantılarını ve yapılandırılabilir analog blokların dış dünya ile olan bağlantılarını sağlar.



Şekil 2.1. FPAA' nın Blok Diyagramı.

2.2. FPAA' nın Tasarım Metodolojisi

Literatürde FPAA' nın tasarım metodolojisi ile ilgili çeşitli yaklaşımlar yer almaktadır. Bunlardan bazıları bu kısımda incelenecektir.

2.2.1. Ayırık Zamanlı / Sürekli Zamanlı Çalışma

FPAA' nın gerçekleştiriminde anahtar niteliğindeki seçim, ayırık zamanda ya da sürekli zamanda çalışmadır. Ayırık zamanlı yaklaşımlarda, anahtarlamalı kapasitör veya anahtarlamalı akım teknolojisi kullanılır. Bu yaklaşımda, programlanabilir elemanların VLSI gerçekleştirimleri için kırkımık üzerinde ayarlanabilir devre düzeneklerinin kullanımına gerek duyulmaz ve bu sebeple dijital kontrol için uygundur. Bununla birlikte örneklenmiş bilgi tekniklerinden bazıları, giriş işaretlerinin en az bir yarısının band sınırlı olmasını gerekli kılar ve bu durum için bozukluğu düzeltme (anti-aliasing) ve yeniden oluşturma (reconstruction) filtrelerinin kullanılması gerekecektir. Bu gereksinim, ayırık zamanlı FPAA devre gerçekleştiriminin band genişliğini sınırlamaktadır. Bu durum ise ayırık zamanlı yaklaşımın dezavantajıdır [3].

Sürekli zamanda çalışan FPAA' da, temel olarak harici veya geçiş iletkenliğine sahip elemanlar kullanılmaktadır. Hem eşik altı hem de doğrusal devrelerin sürekli zaman teknikleri programlanabilir analog devrelerde kullanılmaktadır. Sürekli zaman devre teknikleri band sınırlı giriş işaretlerine gerek duymaz, ama büyük bir dinamik sahada devre elemanlarının programlanabilir karmaşık gerçekleştirimlerine ihtiyaç duyar. Bu

da programlama esnekliğini azaltmaktadır [3]. Bu tez çalışmasında yer alan uygulama devrelerinde ayrı zamanlı çalışan, anahtarlama kapasitör tekniğine sahip FPAA tercih edilmiştir.

2.2.1.1. Anahtarlama Kapasitör Tekniği

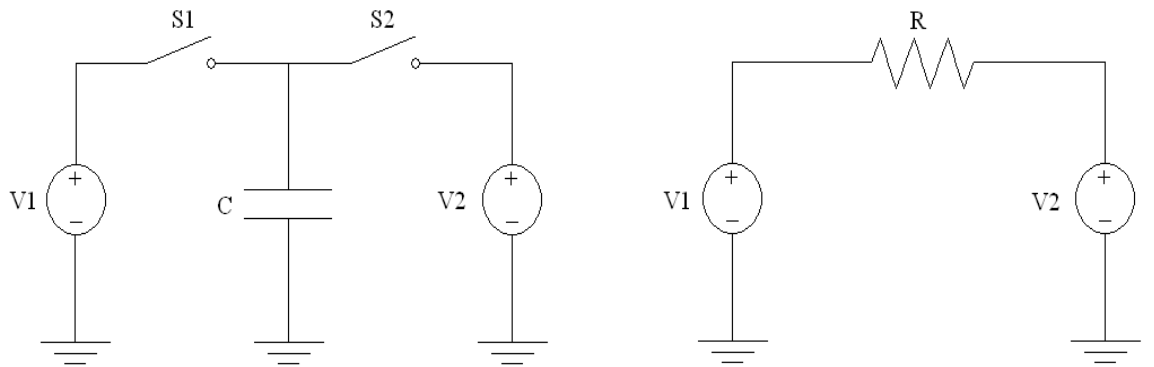
Anahtarlama kapasitör tekniği, iki devre düğümü arasında periyodik olarak anahtarlanan kapasitör tabanlı bir tekniktir. Periyodik olarak anahtarlanan kapasitör, aynı düğümler arasına bağlanan bir direncin yerine kullanılabilir. Kapasitörler, güç tüketmeme gibi avantajlara sahiptir. Bu teknik sayesinde kırkık üzerinde değişik analog devrelerin gerçekleştirimi yapılabilir. [42, 44-46].

Çalışma Prensipleri: Şekil 2.2’ de görüldüğü gibi anahtarlama kapasitör devresi, bir adet kapasitörden, iki adet V_1 ve V_2 gerilim kaynağından ve S_1 ve S_2 anahtarlarından oluşmaktadır. S_1 anahtarı kapalı (ON) ve S_2 anahtarı açık (OFF) iken C kapasitörü, girişe uygulanan V_1 voltajına şarj olur. C kapasitörü üzerindeki toplam şarj; $Q_1=C.V_1$ ’ dir. Anahtarların pozisyonu değiştirdiğinde yani S_1 açıldığında ve S_2 kapandığında C kapasitörü, uygulanan V_2 voltajına kadar deşarj olur. Bu durumda kapasitör üzerindeki toplam şarj; $Q_2=C.V_2$ olur [47].

Anahtarlama periyodu sırasında girişten çıkışa transfer edilen net şarj;

$$\Delta Q = Q_1 - Q_2 = C(V_1 - V_2) \quad (2.1)$$

şeklinde ifade edilir.



Şekil 2.2. Direncin Yerine Kullanılan Anahtarlama Kapasitör Devresi.

Girişten çıkışa akan akımın değeri,

$$I = \Delta Q / T = C(V_1 - V_2) / T = f_s C (V_1 - V_2) \quad (2.2)$$

olur. Bu ifade, direnç üzerinden akan I akımının eşdeğeridir ve değeri aşağıdaki gibi kolayca hesaplanabilir:

$$I = V_1 - V_2 / R \quad (2.3)$$

burada R direnci aşağıdaki gibi tanımlanabilir:

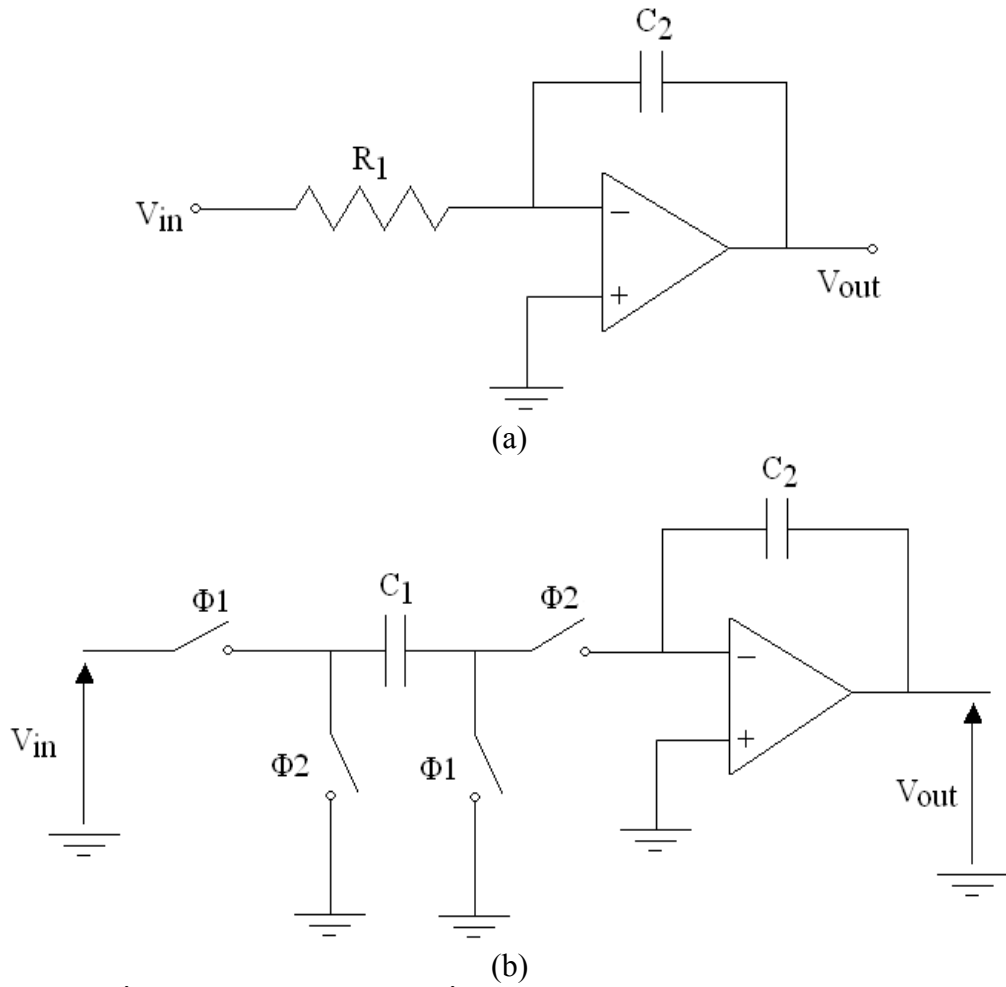
$$R = T / C = 1 / f_s C \quad (2.4)$$

C kapasitans değeri ve T anahtarlama periyodu veya f_s örnekleme frekansı modifiye edilerek istenilen direnç değeri elde edilebilir. Daha öncede bahsedildiği gibi sadece kapasitörler ile direnç gibi davranan bir devre elde edilebilir.

Şekil 2.2' deki devre şeması, op-ampları ile birlikte kullanıldığında değişik analog fonksiyonlar gerçekleştirilebilir. Analog fonksiyon gerçekleştirimi için Şekil 2.3(a)' da normal dirençli ve kapasitörlü integral alıcı devre verilirken bu devrenin eşdeğeri olan anahtarlama kapasitör tekniği kullanılarak gerçekleştirim Şekil 2.3(b)' de örnek olarak verilmiştir. Şekil 2.4' de anahtarlama işaret şemasında gösterilen Φ_1 ve Φ_2 , anahtar grubunun kapandığındaki fazları ifade etmektedir. Burada T , devrenin anahtarlama işareti olup anahtarlama işareti (master clock) olarak adlandırılmaktadır [47].

2.2.2. Voltaj Modlu / Akım Modlu Çalışma

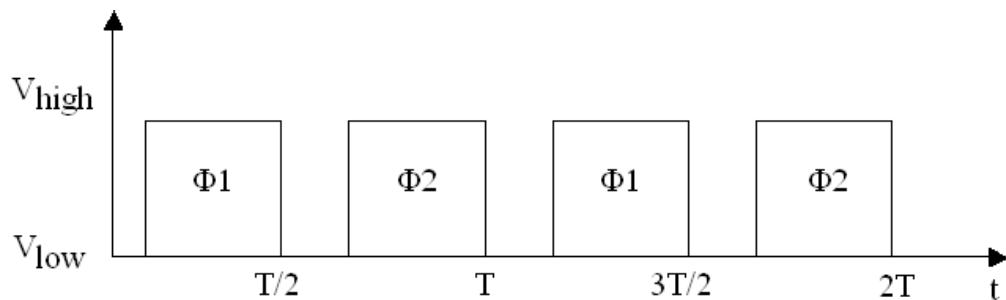
FPAA gerçekleştiriminde diğer bir önemli tasarım seçimi, işaret parametresi olarak voltaj ya da akımın kullanımıdır. Voltaj işaretleri, yüksek bir yayılıma (fanout) sahiptir ve voltaj modlu devre teknikleri çok gelişmiştir. Programlanabilir analog devrelerden bazıları voltaj modlu olarak geliştirilmiştir. Bununla beraber programlanabilir analog devrelerin bazı gerçekleştirimlerinde işaret parametresi olarak akım seçildiğinde akım modlu devre; daha basit gerçekleştirimlere, yüksek doğruluk ve yüksek band genişliği gibi avantajlara sahip olur [3]. Hem ayrık zamanlı hem de sürekli zamanlı metotlarda voltaj ve akım modları kullanılabilir iken ayrık zamanlı yaklaşımlarda voltaj modlu kullanım daha yaygındır.



Şekil 2.3. (a) İntegral Alıcı Devre, (b) İntegral Alıcı Devrenin Anahtarlamalı Kapasitör Devre Eşdeğeri.

2.3. FPAA' nın Tarihsel Gelişimi

Literatürde FPAA ile ilgili yapılan tasarımlar, bireysel ve ticari olmak üzere ikiye ayrılabilir. Bu kısımda, yapılan FPAA tasarımları incelenecek olup, bu tasarımların çeşitli kriterlere göre yapılan mukayeseleri tablo halinde verilecektir.



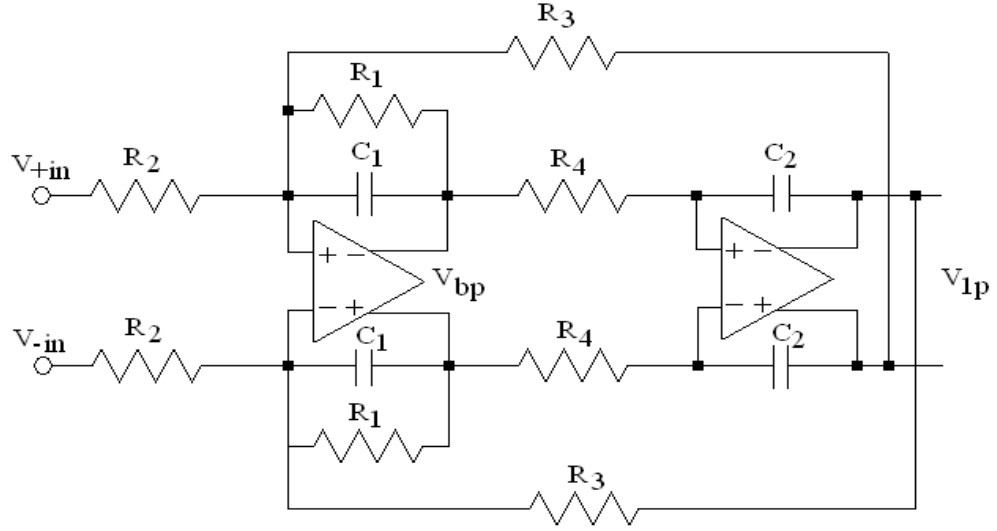
Şekil 2.4. İntegral Alıcı Devrenin Anahtarlamalı Zamanlama İşaretinin Şeması.

2.3.1. Bireysel Olarak Gerçekleştirilen FPAA Tasarımları

Literatürde ilk FPAA tasarımı, Massimo Sivilotti tarafından önerilmiş olup, analog sinir ağlarının sentezi ve testinde kullanılmıştır [1, 13]. Bu tasarımda anahtar elemanı olarak CMOS elemanı kullanılmıştır. CMOS' dan, hiyerarşik yönlendirme ağında (routing network) akım aynaları, diferansiyel çift gibi temel kaynakları bağlamada yararlanılmıştır [1]. Bir başka çalışma ise Lee ve Gulak tarafından eşik altı bölgede çalışan CMOS kullanılarak gerçekleştirilmiştir [14]. Bu tasarım, 100 KHz' den küçük bir çalışma frekansına sahiptir ve sinir ağlarının donanımsal gerçekleştirmelerinde kullanılmıştır. Bu çalışmalarını Toronto üniversitesinde görev yapan bir araştırma grubu ile sürdüren Lee ve Gulak, iki farklı FPAA mimarisi daha geliştirmiştir [15, 48]. Birinci tip FPAA, geçiş iletkenliği esas alınarak tasarlanmış, sürekli zaman CMOS tabanlı ve 100 KHz sahasında çalışabilen bir elemandır [15]. Bu FPAA' nın, akustik frekans sahasındaki işaret işleme uygulamalarında, Şekil 2.5' de gösterilen biquad filtre tasarımında kullanılması hedeflenmiştir. Bu FPAA tasarımındaki yapılandırılabilir analog bloklar, işlemsel kuvvetlendirici ve anahtarlanabilir kapasitör ihtiva etmektedir. Bu tasarımda ara bağlaşım ağındaki anahtarlar, MOSFET transistörler kullanılarak gerçekleştirilmiştir. MOSFET transistörler, programlanabilir açma/kapama anahtarı, polarite değişim anahtarı ve değişken direnç olarak kullanılmıştır. MOSFET transistörler dört bölgeyi gerçekleştirmekte de kullanılmıştır [48]. Diğer tip FPAA, akım taşıyıcı tabanlı olup, video frekans sahasındaki (10 MHz) uygulamalarda kullanılmıştır. Bu tasarımda her bir CAB, ikinci nesil akım taşıyıcıdan ve programlanabilir kapasitör ve direnç dizisinden oluşmaktadır [48].

FPAA üzerine çalışmalar, Toronto üniversitesindeki gruba ek olarak Portland State üniversitesindeki bir başka araştırma grubu tarafından da yapılmıştır. Bu grup, iki kutuplu (bipolar) akım-modlu, sürekli-zamanlı, OTA-tabanlı FPAA ortaya koymuştur [16, 17]. Bu FPAA tasarımlarının sürekli, bulanık ve çoklu değerli lojik uygulamalarda kullanılmasını hedeflenmiştir. Bu FPAA tasarımı, 10 MHz' e kadar olan frekans sahasında çalışmaktadır [49].

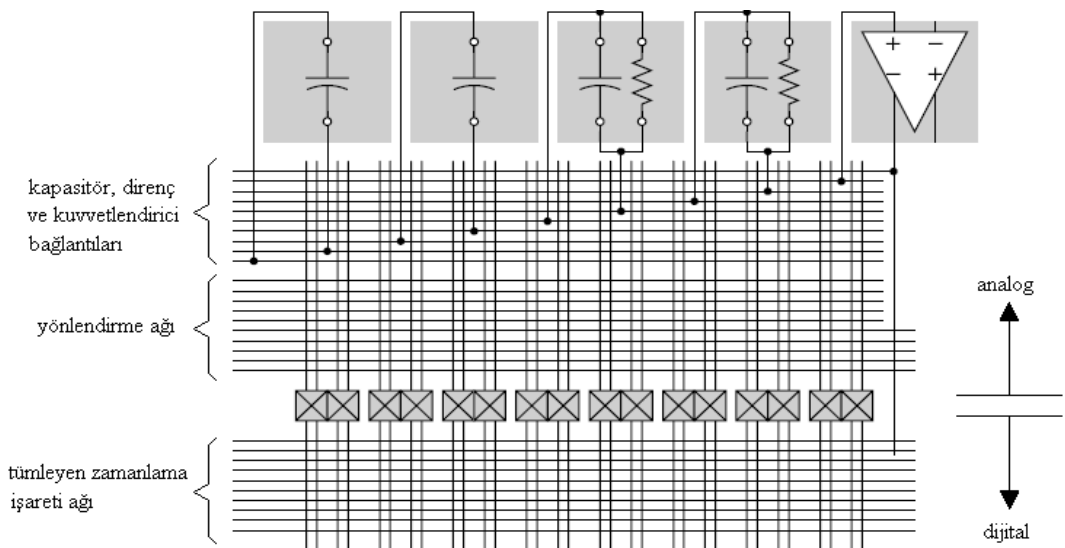
Yukarıda bahsedilen araştırma gruplarına ilaveten Johns Hopkins üniversitesinde de bir araştırma grubu, hem FPAA' yı hem de FPGA' yı içeren alan programlanabilir karışık işaretli dizi devreleri (Field Programmable Mixed-signal Array - FPMA) üzerine



Şekil 2.5. Lee ve Gulak Tarafından Sunulan Biquadratic Filtre [48].

çalışmıştır [18]. Bu devrenin analog modülü, işlemsel kuvvetlendiriciden, programlanabilir dirençten, programlanabilir kapasitör dizisinden ve CMOS anahtardan oluşmaktadır. Bu tasarımdaki yapılandırılabilir analog modülün yapısı, Şekil 2.6' da verilmiştir [49].

Bu kısımda FPAA tasarımlarından literatürde üzerinde en çok durulan bireysel gerçekleştirmelerinden bahsedilmiştir. Burada incelenen FPAA tasarımları ile beraber literatürde bireysel olarak gerçekleştirimi var olan diğer tasarımların özellikleri özet şeklinde Tablo 2.1' de yer almaktadır.



Şekil 2.6. Johns Hopkins Üniversitesi' ndeki FPAA Araştırma Grubu Tarafından Ortaya Konulan Tasarımdaki Yapılandırılabilir Analog Modül Yapısı.

Tablo 2.1. Bireysel Olarak Gerçekleştirilen FPAA Tasarımlarının Birbirleri İle Karşılaştırılması.

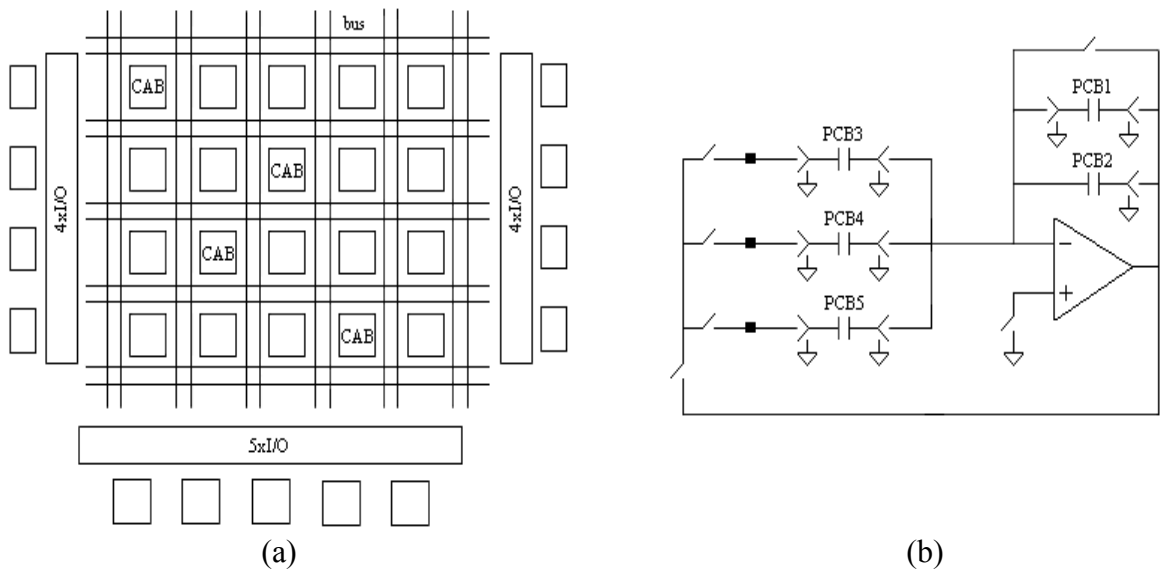
CAB	Teknik	Teknoloji	Band Genişliği	Kullanım Alanları
Sivilotti [13]	Eşik altı	CMOS	--	Analog sinir ağları yapılarının sentezi ve testi
Lee ve Gulak [14]	Eşik altı	CMOS	100 KHz	Sinir ağlarının donanımsal gerçekleştirimi
Lee ve Gulak [15] (Toronto)	Op-amp tabanlı	CMOS	100 KHz,	Analog işaret işleme uygulamaları
Lee ve Gulak [15] (Toronto)	İkinci nesil akım taşıyıcı tabanlı	CMOS	10 MHz	Analog işaret işleme uygulamaları
Pierzchala ve arkadaşları [16, 17] (Portland State)	Transistör tabanlı akım modlu	OTA	10 MHz	Lojik uygulamaları
Edwardz ve arkadaşları [18] (Johns Hopkins)	Op-amp tabanlı	CMOS	20 MHz	Uzay Uygulamaları
Kutuk ve Kang [19, 20]	Anahtarlama kapasitör	--	2KHz-260KHz	Sürekli işaret işleme uygulamaları
Pankiewicz [21]	OTA-tabanlı	OTA	Birkaç KHz' den birkaç MHz' e kadar	Sürekli işaret işleme uygulamaları
Permont ve arkadaşları [22]	İkinci nesil akım taşıyıcı tabanlı yaklaşım	CMOS	Birkaç KHz' den birkaç MHz' e kadar	Sürekli işaret işleme uygulamaları

2.3.2. Ticari Olarak Gerçekleştirilen FPAA Tasarımları

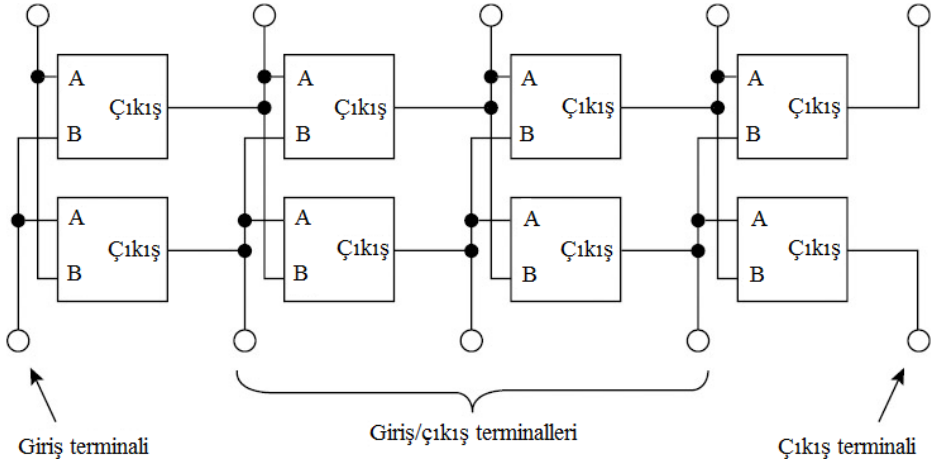
1995 yılında IMP Şirketi tarafından FPAA benzeri EPAC 50E10 ve daha sonra da EPAC 50E30 üretilmeye başlanmıştır [24]. Bu tasarımlar anahtarlama kapasitör tabanlı, ayırık zamanlı tasarımlardır. Bu elemanlar giriş modüllerinden, analog çarpıcıdan, programlanabilir kuvvetlendiriciden, iletim hattından ve çıkış modüllerinden oluşmaktadır. Bu devrenin programlanması için 200 bitlik bir bit dizisi kullanılmaktadır. Band genişliği, anahtarlama kapasitör tekniğinin kullanılmasından dolayı 125KHz (anahtarlama frekansı 1MHz) ile sınırlıdır. EPAC 50E10, işaret işleme

uygulamalarında, EPAC 50E30, görüntüleme uygulamalarında kullanılmışlardır [7, 48]. EPAC ailesi ticari olarak çok uzun soluklu olmamıştır.

1997 yılında Motorola şirketi, MPAA020 adlı CMOS anahtarlamalı kapasitör tabanlı bir FPAA tasarımı gerçekleştirmiştir [23]. Şekil 2.7(a)' da gösterilen bu FPAA yapısı, 4x5 CAB' leri içermektedir ve 619 bitlik bir bit dizisi kullanılarak programlanmaktadır. CAB hücre dizisinin üç tarafında 13 adet giriş/çıkış hücreleri bulunmaktadır [7, 48]. Şekil 2.7(b)' de açık devre şekli gösterilen CAB, bir adet işlemsel kuvvetlendiriciden, beş adet programlanabilir kapasitör dizisinden ve bir grup anahtardan oluşmaktadır. Kapasitör dizilerinin değerleri, kırkık üzerindeki hafızada saklanan dijital kontrol bitlerinden ayarlanabilmektedir. Kırkık üzerindeki yapılandırılabilir ara bağlaşım ağı, genel ve yerel ara bağlaşımardan oluşmaktadır. Şekil 2.7(a)' da gösterilen bu FPAA' da 11 çift hat bulunmaktadır. Bunlar CAB' lerin her bir satır ve sütunu boyunca yatay veya dikey olarak yer alırlar [12]. Bu FPAA' da, her bir CAB, birinci dereceden filtreleme fonksiyonunu gerçekleştirebilmektedir [48]. EPAC ailesinde olduğu gibi bu FPAA tasarımında da anahtarlama frekansı 1MHz, band genişliği, 200KHz' dir. MPAA020' nin programlanması, bilgisayarın seri portu aracılığıyla yapılmaktadır. Motorola, FPAA ve FPGA ürünlerinin üretimini tamamen durdurduğu için bu MPAA020' nin ticari ömrü de EPAC serisi gibi uzun olmamıştır [49].



Şekil 2.7. Motorola MPAA020' nin, (a) FPAA Yapısı, (b) FPAA CAB Yapısı.



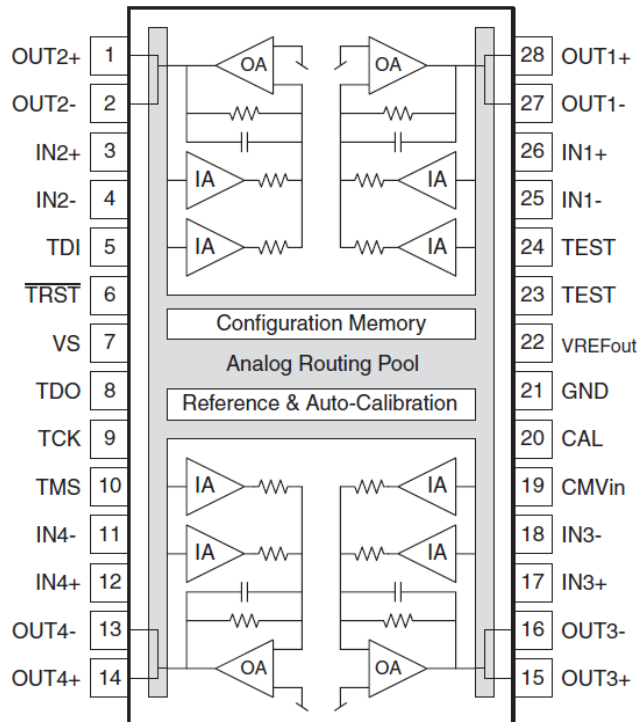
Şekil 2.8. Zetex Firmasının Ürettiği TRAC020' nin Dizi Yapısı.

Motorola ile aynı yıl Zetex firması, sürekli zaman analog programlanabilir bir eleman olan TRAC020' yi (Totally Reconfigurable Analog Circuit-TRAC) gerçekleştirmiştir [26]. Bu elemanın band genişliği 4MHz' e kadar çıkabilmektedir. TRAC serisi, 2x10 şeklinde olmak üzere toplam 20 adet CAB' den oluşmakta ve 60 bitlik bir bit dizisi kullanılarak programlanmaktadır. Her bir CAB, logaritma, anti logaritma, negatif alma, evirmeyen geçiş, toplama, işlemsel kuvvetlendirici, yarım dalga doğrultucu ve boş durum gibi bu sekiz fonksiyondan birini gerçekleştirebilmektedir [7, 48]. Burada evirmeyen geçiş, topolojik nedenlerden dolayı kullanılmış olup birim kazançlı kuvvetlendirici gibi dizide hiçbir değişiklik yapmaksızın hücre boyunca bir rota izler. Ara bağlantı ağı, Şekil 2.8' deki gibi CAB' ler arasına direkt bağlanmıştır.

1990' lı yılların sonlarında Lattice Semiconductor firması, ispPAC10, ispPAC20 ve ispPAC30 olarak adlandırdığı yeniden yapılandırılabilir analog devre ailesini sunmuştur [25]. ispPAC ailesi, aynı temel blok yapısına sahip olduğu için ispPAC10' un blok diyagramı örnek olarak Şekil 2.9' da verilmiştir. Şekilden de görüldüğü gibi ispPAC elemanı, işlemsel kuvvetlendirici, direnç ve kapasitör üçlüsünün yanı sıra programlanabilir enstrümantasyon kuvvetlendiricisini de içermektedir. Bu eleman hassas filtreleme, toplama/çıkarma, kazanç/zayıflatma ve integral alma gibi temel analog fonksiyonları gerçekleştirir [25, 49]. ispPAC ürünleri, uygulama alanlarına göre farklı band genişliklerine sahiptir. Mesela ispPAC10 ve ispPAC20' nin maksimum band genişliği, 100KHz iken ispPAC30' un 1.55MHz' dir. ispPAC ailesinin üyelerinin her biri, farklı uygulamalarda kullanılmaktadır. Bu uygulamalardan bazıları işaret işleme, kapalı döngü ve görüntüleme uygulamaları ve analog-dijital dönüştürücüler olarak

sayılabilir. Tüm ispPAC ailesi üyeleri, SDRAM konfigürasyon ötelemeli kaydedici (SDRAM configuration shift register) ve kırkım üzerindeki hafıza arasında konfigürasyon bitleri değiştirilerek anında tekrar yapılandırılabilir.

Günümüze kadar ticari olarak gelebilen FPAA tasarımı Anadigm firması tarafından gerçekleştirilmiştir [6]. Anadigm tarafından sunulan FPAA' lar, anahtarlamalı kapasitör tekniği tabanlıdır ve ikinci neslin üretimi devam etmektedir. Birinci nesil üretimi, AN10E40 olarak adlandırılmış olup, 4x5 CAB' lerden oluşmaktadır ve band genişliği, 250KHz' dir. Her bir CAB, kazanç, sınırlayıcı, karşılaştırıcı, doğrultucu, osilatör, örnekleme-tutma ve farklı tipteki filtreler gibi birçok fonksiyonu gerçekleştirebilmektedir [1, 49]. Anadigm firması tarafından üretilen II. nesil FPAA' lar, AN120E04, AN121E04, AN131E04, AN220E04, AN221E04 ve AN231E04 tipi FPAA' lardır [6]. Bu FPAA' lar, 2x2 CAB' lerden meydana gelmiş olup band genişliği, 2MHz' dir. Bu II. nesil FPAA' lardaki CAB' ler, birinci nesildeki CAB' ler gibi kazanç, sınırlayıcı, karşılaştırıcı, doğrultucu, osilatör, örnekleme-tutma, toplama-çıkarma, çarpma-bölme ve farklı tipteki filtreler gibi birçok fonksiyonu gerçekleştirebilmektedir. Bu tez çalışmasında kullanılan FPAA ailesinden AN221E04 ile ilgili detaylı bilgi Bölüm 2.5' de verilecektir.



Şekil 2.9. ispPAC10' nun İç Yapısı.

Tablo 2.2. Ticari Olarak Gerçekleştirilen FPAA Tasarımlarının Birbirleri İle Karşılaştırılması.

Üretici Firma	Teknoloji	Band Genişliği	Download	Teknik	Kullanıldığı Alanlar
IMP	CMOS	125 KHz, 150 KHz	Seri Port	Anahtarlamalı Kapasitör	İşaret iyileştirme ve görüntüleme uygulamaları
Motorola	CMOS	200 KHz	Seri Port	Anahtarlamalı Kapasitör	İşaret işleme uygulamaları
Zetex	Bipolar	4 MHz	Paralel Port	Sürekli zaman	İşaret işleme uygulamaları
Lattice	CMOS	100 KHz, 1.5 MHz	Seri Port	Sürekli zaman	İşaret işleme ve görüntüleme uygulamaları, filtre uygulamaları
Anadigm	CMOS	2 MHz	Seri Port	Anahtarlamalı Kapasitör	Filtre uygulamaları, RFID, işaret işleme, biyomedikal ile ilgili uygulamalar,

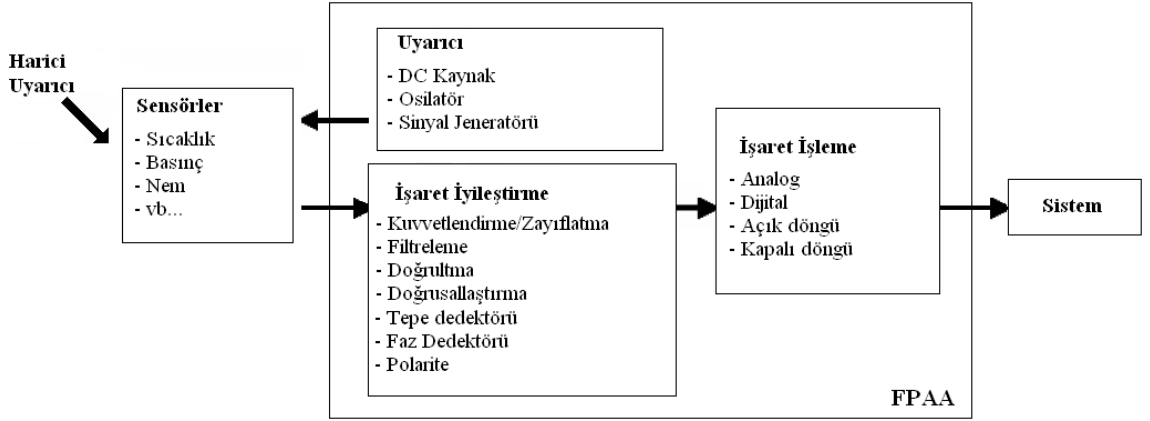
Buraya kadar ticari olarak gerçekleştirilen FPAA tasarımlarından bahsedilmiştir. Bu tasarımların birbirleri ile karşılaştırılmaları özet şeklinde Tablo 2.2' de yer almaktadır.

2.4. FPAA' ların Uygulama Alanları

FPAA' lar birçok farklı alanda kullanılmaktadır. Bunlar arasında;

- İşaret iyileştirme ve işleme [6, 10]
- Yapay sinir ağı gerçekleştirimi [27]
- Filtre uygulamaları [19, 28]
- Lojik uygulamaları [21]
- Adaptif kontrol [8]
- Radyo dalgalarıyla kimliklendirme (Radio Frequency Identification, RFID) [29]
- Telekomünikasyon [30]
- Tıp [31]
- Kaotik osilatör [32-39]
- Havacılık [6]
- Mühendislik eğitimi uygulamaları [40]

gibi alanlar sayılabilir. Bu uygulama alanlarından örnek olarak işaret iyileştirme ve işleme uygulamaları için FPAA kullanımını içeren bir blok yapı, Şekil 2.10' da verilmiş

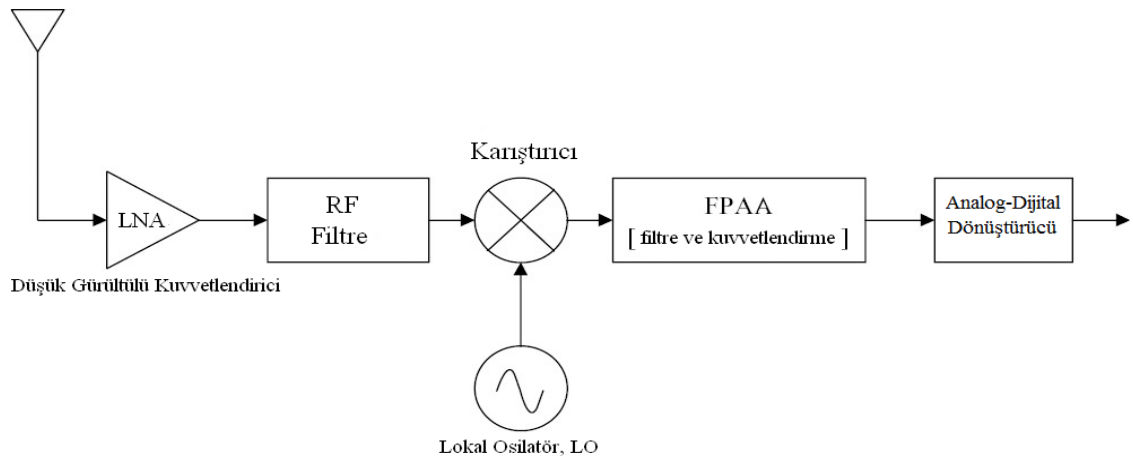


Şekil 2.10. İşaret İyileştirme ve İşleme Uygulamalarında FPAA' nın Kullanımı [50].

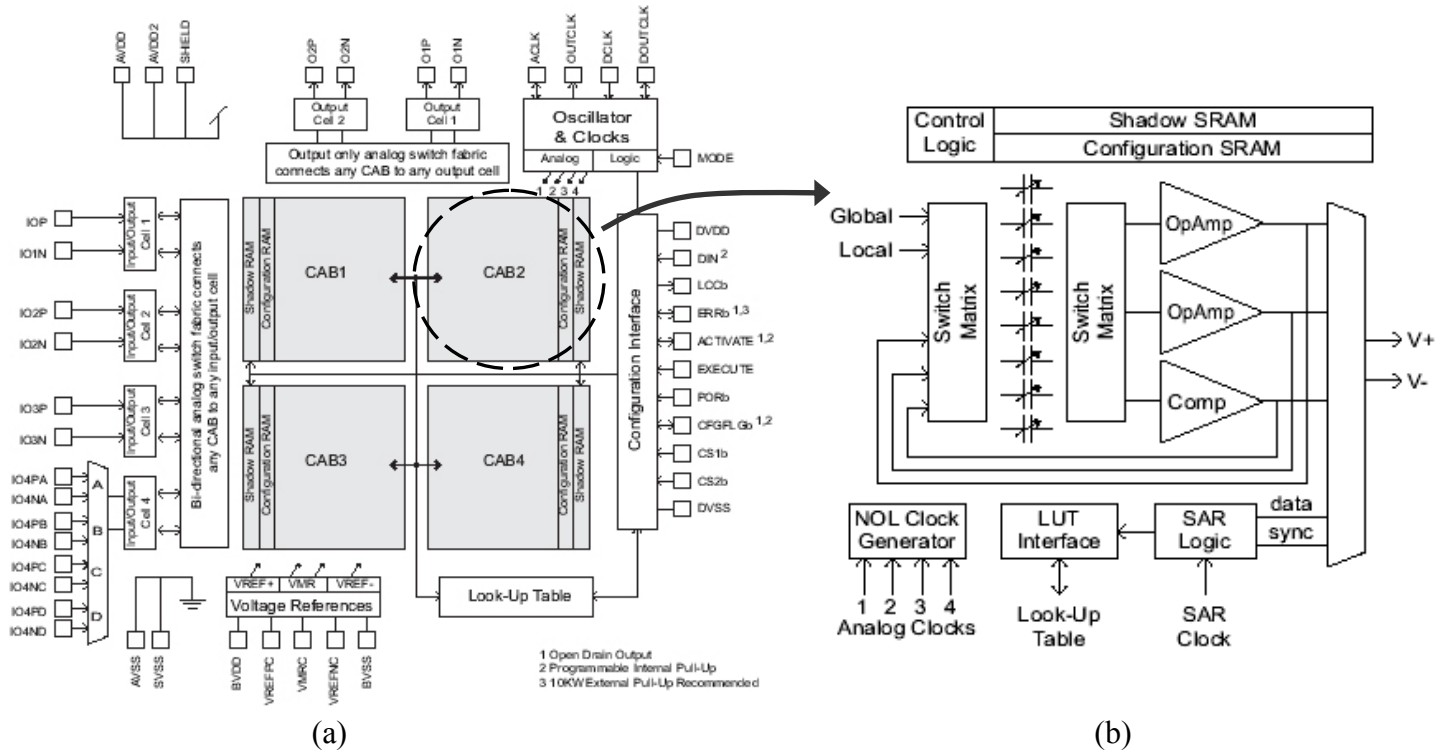
olup radyo alıcı uygulamasında FPAA kullanımını içeren blok yapı ise Şekil 2.11' de yer almaktadır.

2.5. Tez Çalışmasında Kullanılan FPAA' nın Özellikleri

Bu tez çalışmasında Anadigm firması tarafından üretilen anahtarlamalı kapasitör tekniğine dayalı, ayırık zamanda çalışan, II. nesil AN221E04 isimli FPAA kullanılmıştır. Şekil 2.12(a)' da AN221E04 isimli FPAA' nın blok diyagramı gösterilmektedir. AN221E04, 2x2 boyutundaki CAB' lerden ve harici giriş/çıkış bloklarından oluşmaktadır [6]. Bu FPAA, dört adet yapılandırılabilir giriş/çıkış biriminden ve iki adet çıkış biriminden oluşmakta olup aynı firmanın ürettiği I. nesil FPAA' lar ile karşılaştırıldığında daha yüksek band genişliği ve iyileştirilmiş bir işaret-gürültü oranı sunar. Ayrıca bu ürün, sensör çıkışının doğrusallaştırılması ve rastgele seçilmiş dalga formu sentezi gibi doğrusal olmayan fonksiyonlar için kullanılabilir.



Şekil 2.11. Radyo Alıcı Uygulamasında FPAA' nın Kullanımı [30].



Şekil 2.12. (a)AN221E04 Tipi FPAA' nın Blok Diyagramı, (b) CAB' in İç Yapısı.

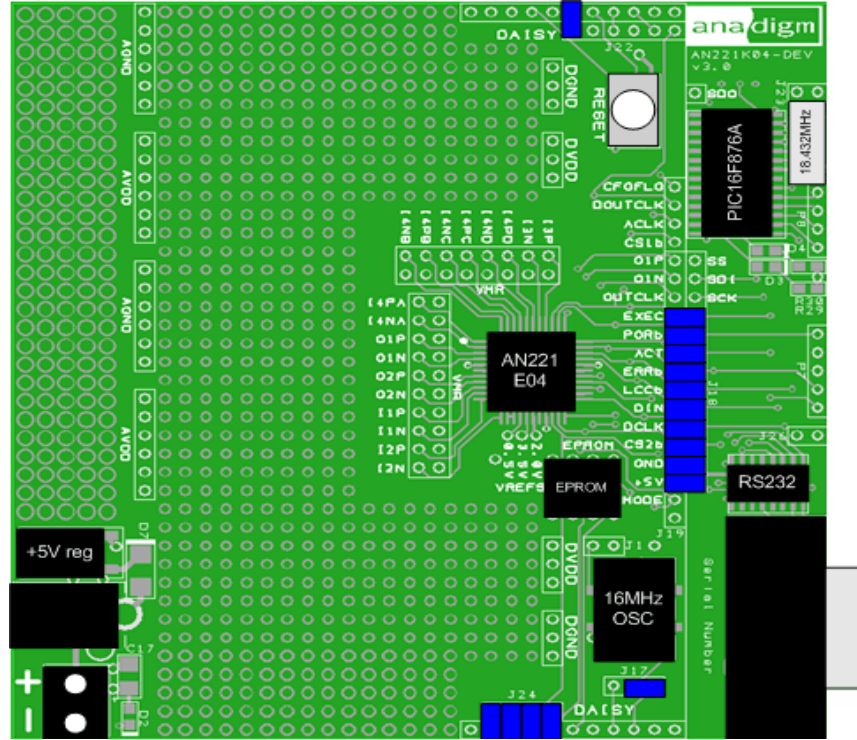
Tasarım sırasında kullanılan integral alıcı, türev alıcı, filtre, karşılaştırıcı gibi çeşitli analog fonksiyonlar CAB' lerin içerisinde programlanabilir bir şekilde gerçekleştirilebilir [41-43]. Şekil 2.12(b)' de gösterilen CAB blok diyagramı, kapasitör dizisine ve anahtar matrisine ek olarak bir karşılaştırıcı ve iki adet işlemsel kuvvetlendiriciye de sahiptir [51]. Ayrıca CAB içerisinde bulunan kontrol lojik, dış dünyadan gölge SRAM' e bilgi transfer eder ve konfigürasyon SRAM' e de bu bilgiyi kopyalar. CAB' deki anahtarlama matrisi, konfigürasyon SRAM' ler tarafından kontrol edilen ve devre bağlantıları için kullanılan statik ve dinamik anahtarlara sahiptir. CAB' in iç yapısında bulunan kapasitör dizisi, 8 adet eşit boyutta kapasitörü ihtiva eder. Bu kapasitörler, istenilen kapasitans değeri için programlanabilir [44, 47]. Her bir CAB' deki hafıza tablosu (Look Up Table, LUT) arayüzü, hafıza tablosuna erişmeyi sağlar. Ardışık yaklaşımlı kaydedici (Successive-Approximation Register, SAR) lojiji, analog-dijital dönüştürücü (Analog-Digital Converter, ADC) olarak kullanılmaktadır. Burada üretilen adres kodları, LUT' ta saklanan bilgiye erişimde kullanılır [44].

Bu tez çalışmasında AN221E04 isimli FPAA' nın içerisinde yer aldığı, Anadigm firmasının ürettiği AN221K04-DVLP2 isimli uygulama kartı kullanılmıştır. Şekil 2.13' de bu uygulama kartının görünümü yer almaktadır. Bu kart üzerinde AN221E04 isimli FPAA, 16 MHz' lik osilatör modülü, 16F8786A isimli PIC, 18.432 MHz' lik kristal osilatör ve EPROM yer almaktadır [6].

2.5.1. AN221K04-DVLP2 Uygulama Kartının Çoklu Kullanımı

AN221K04-DVLP2 isimli uygulama kartlarının, gerekli atlama (jumper) ayarları yapılmak kaydıyla, birbirlerine bağlanmasıyla çoklu kullanımı mümkündür. Şekil 2.14' de iki adet AN221K04-DVLP2 uygulama kartının gerekli bağlantıları yapılarak ikili kullanımı verilmiştir.

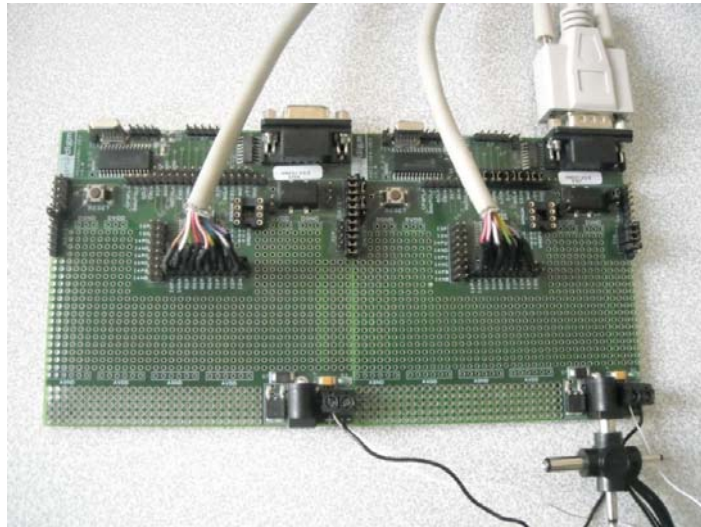
FPAA' nın programlanmasında kullanılan yazılım olan AnadigmDesigner 2' de üç farklı kart kullanılarak oluşturulan bir sistemin yüklenmesi ve yapılması için gerekli atlama ayarları, Şekil 2.15' de gösterilmiştir [6]. Uygulama kartlarında (+) ve (-) sembolleri ile gösterilen besleme girişleri, kablolar yardımıyla birleştirilir. Birleştirildikten sonra beslemenin uygulama kartlarından birine verilmesi yeterli olacaktır.



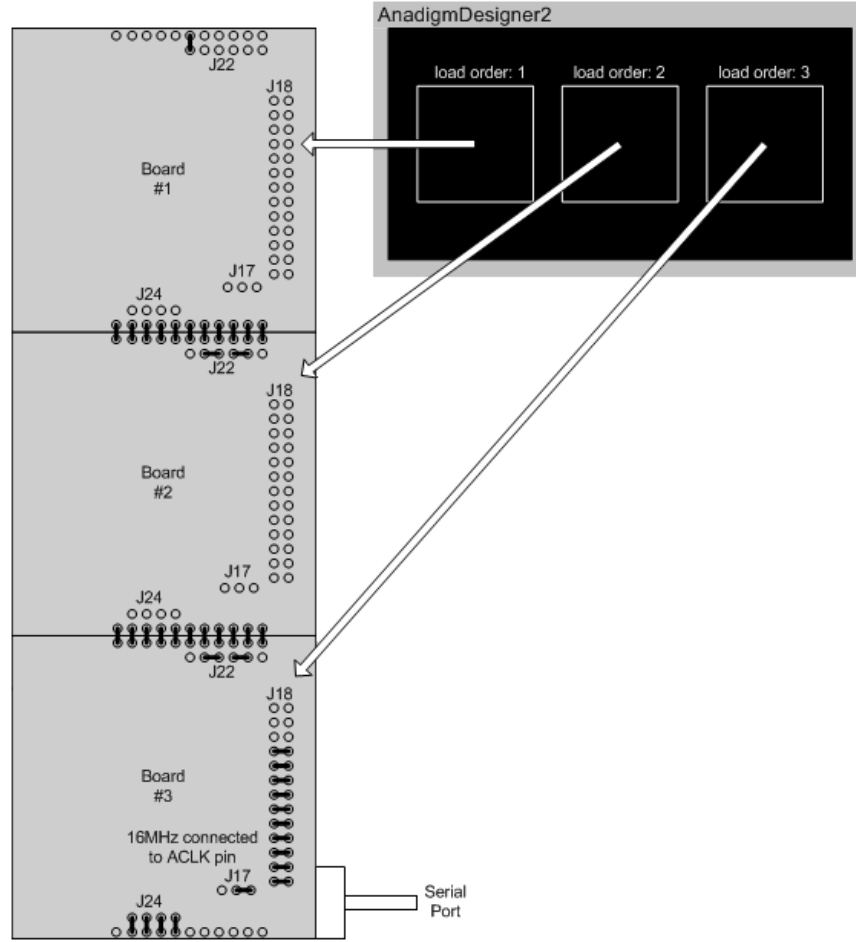
Şekil 2.13. AN221K04-DVLP2 Uygulama Kartı.

2.6. FPAA' nın Bilgisayar Arayüz Programı: AnadigmDesigner 2

AN221E04 tipi FPAA' nın programlanmasında kullanılan yazılım AnadigmDesigner 2' nin arayüz gösterimi Şekil 2.16' da verilmiştir. Bu yazılım, üretici firma tarafından önceden tanımlanan bloklar ve giriş/çıkış birimleri sayesinde tasarımcıya devre tasarımı sırasında kolaylık sağlar.

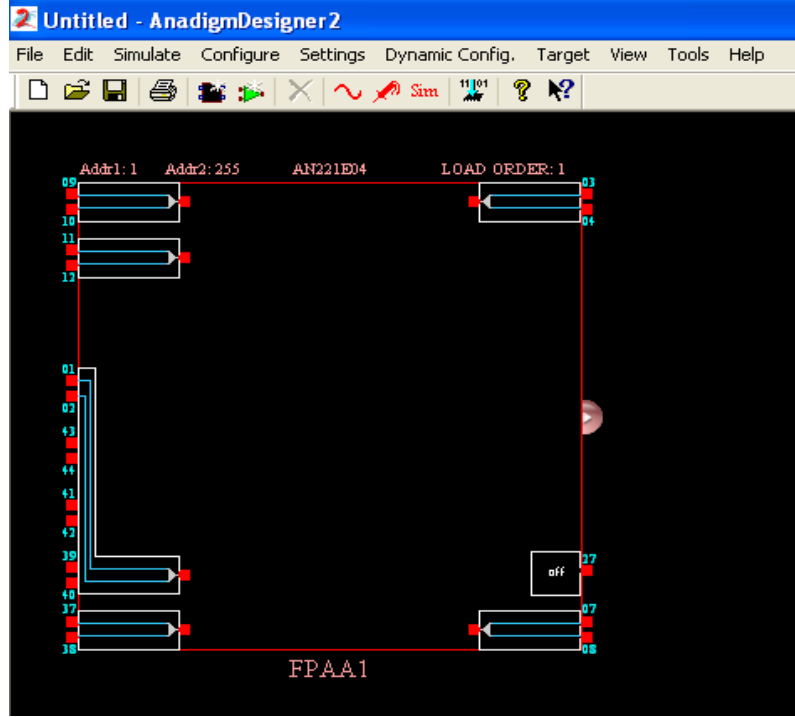


Şekil 2.14. AN221K04-DVLP2 Uygulama Kartının İkili Kullanımı.



Şekil 2.15. AnadigmDesigner 2’ de Oluşturulan Bir Sistemin Yüklenmesi ve Doğru Atlama Ayarları İçin Gerekli Olan Düzenlemeler.

Yazılımda tanımlanmış olan bu bloklara yapılandırılabilir analog modüller (Configurable Analog Modules, CAM) adı verilir ve bu modüller farklı fonksiyonları gerçekleştirirler. Bu fonksiyonlardan bazılarının yer aldığı CAM listesi, Şekil 2.17’ de gösterilmiştir. Tasarımcı, bu listeden seçtiği temel analog işlem bloklarını kullanarak kendi tasarımını gerçekleştirebilir. Bu bloklar ile ilgili bilgiler, AnadigmDesigner 2 yazılımının yardım dosyalarında verilmektedir. Bu analog bloklardan tez çalışmasında kullanılanlar ile filtre tiplerinden örnek bir blok olan *FILTERBILINEAR* aşağıda detaylı bir şekilde incelenmiştir. Yazılımda tasarıma geçmeden önce anahtarlama işaretine ait parametrelerin ayarlarının yapılması gerekmektedir. Çünkü her bir CAM, anahtarlama kapasitör devresinden oluşmakta olup, bu devrelerin fonksiyonlarını yerine getirebilmeleri analog anahtarlama işaretine ihtiyaç duyulmaktadır. Anahtarlama işaretinin parametre değerleri, CAM’ in etkili olabileceği frekans sahasıyla doğrudan



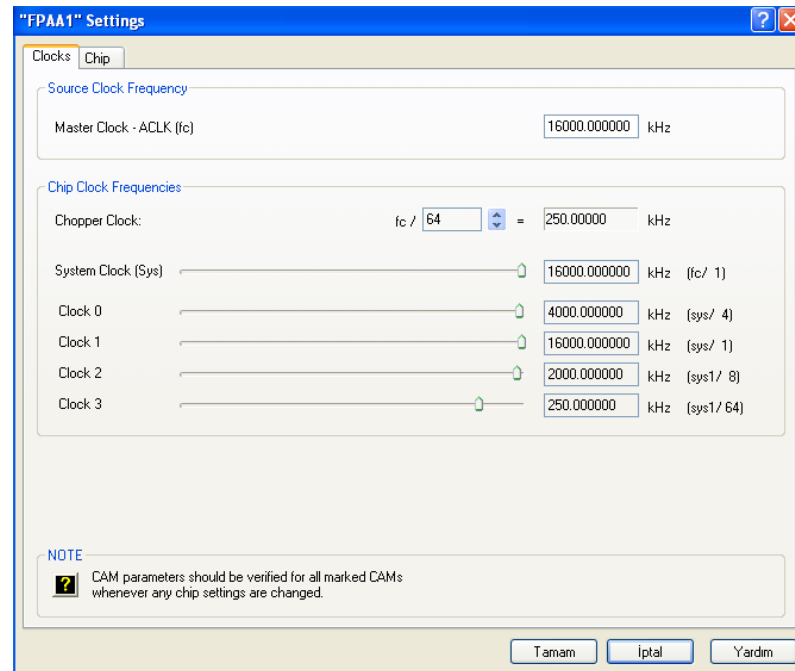
Şekil 2.16. AnadigmDesigner 2 Yazılımının Arayüz Gösterimi.

ilişkilidir. Örneğin 2MHz frekansı olan bir anahtarlama işareti, 4KHz ile 200KHz arasında çalışabilecek bir CAM' in varlığı anlamına gelir. Şekil 2.18' de FPAA' nın anahtarlama işaretinin parametrelerinin ayarlandığı arayüz gösterilmektedir. Bu arayüzde yer alan anahtarlama işareti (Master Clock, f_c), kırmık tarafından üretilebilecek analog işaretler için örnekleme oranını tayin eder ve frekans sınırlarını belirler. Ana anahtarlama işareti, FPAA tarafından işlenebilecek analog işaretlerin frekans sınırları ile belirlenir ve örnekleme oranı ile ayarlanır. AN221K04-DVLP2 uygulama kartında bulunan 16 MHz' lik osilatör modülünden dolayı f_c maksimum 16 MHz olabilmektedir. Kullanıcı eğer isterse bu frekansı azaltabilir ama artıramaz. Clocklar kısmındaki Chopper, System, Clock 0, Clock 1, Clock 2 ve Clock 3, ana anahtarlama işaret frekansının aşağı doğru bölünmesiyle türetilir. Chopper ve sistem clockları, ana anahtarlama işaretinden türetilirken, kalan diğer clocklar system clockundan türetilmektedir. AnadigmDesigner 2 yazılımı kullanıcıya yaptığı tasarımlar için zamanlama işaretlerini ayarlama kolaylığı sunmaktadır.

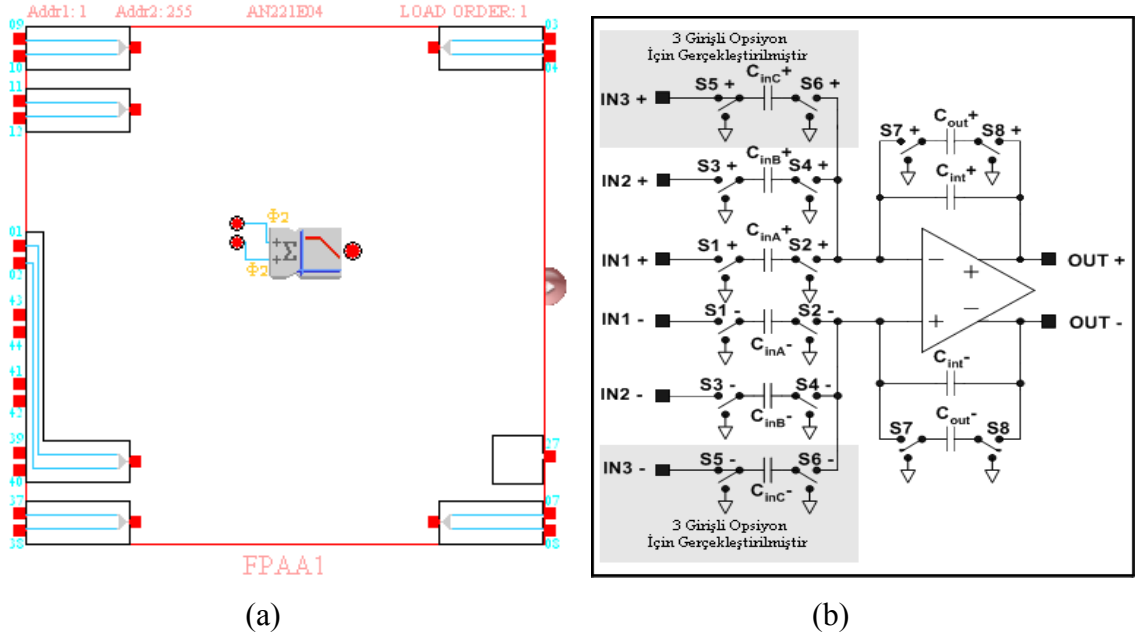
Bu tez çalışmasında kullanılan CAM modüllerinden biri olan *SUMFILTER* bloğu, Şekil 2.19' da gösterilmektedir. Tek kutuplu alçak geçiren filtreye de sahip olan bu blok, üç girişe kadar toplama yapabilmektedir. Ancak üçüncü girişin ihtiyaç halinde kullanıma

CAM	Description	Version	Approved
ADC-SAR	Analog to Digital Converter (SAR)	(*)	Yes
Comparator	Comparator	(*)	Yes
Differentiator	Inverting Differentiator	(*)	Yes
Divider	Divider	(*)	Yes
FilterBilinear	Bilinear Filter	(*)	Yes
FilterBiquad	Biquadratic Filter	(*)	Yes
FilterDCBlockLP	DC Blocking HPF with Optional LPF	0.0.7	No
FilterLowFreqBi...	Low Corner Frequency Bilinear LPF (External...	(*)	Yes*
FilterVoltageCo...	Voltage Controlled Filter	1.4.0	No
GainHalf	Half Cycle Gain Stage	(*)	Yes
GainHold	Half Cycle Inverting Gain Stage with Hold	(*)	Yes
GainInv	Inverting Gain Stage	(*)	Yes
GainLimiter	Gain Stage with Output Voltage Limiting	(*)	Yes*
GainPolarity	Gain Stage with Polarity Control	(*)	Yes
GainSwitch	Gain Stage with Switchable Inputs	(*)	Yes
GainVoltageCo...	Voltage Controlled Variable Gain Stage	(*)	Yes
Hold	Sample and Hold	(*)	Yes
HoldVoltageCo...	Voltage Controlled Sample and Hold	(*)	Yes
Integrator	Integrator	(*)	Yes
Multiplier	Multiplier	(*)	Yes
MultiplierFilterL...	Multiplier with Low Corner Frequency LPF (E...	(*)	Yes*
OscillatorSine	Sinewave Oscillator	(*)	Yes
PeakDetect2	Peak Detector	(*)	(*)
PeakDetectExt	Peak Detector (External Caps)	0.0.3	No
PeriodicWave	Arbitrary Periodic Waveform Generator	(*)	Yes
RectifierFilter	Rectifier with Low Pass Filter	(*)	Yes
RectifierHalf	Half Cycle Rectifier	(*)	Yes
RectifierHold	Half Cycle Inverting Rectifier with Hold	(*)	Yes
SquareRoot	Square Root	(*)	Yes
SumBiquad	Sum/Difference Stage with Biquadratic Filter	(*)	Yes
SumDiff	Half Cycle Sum/Difference Stage	(*)	Yes

Şekil 2.17. AnadigmDesigner 2 Yazılımında Kullanılan CAM Listesi.



Şekil 2.18. FPAA' nın Anahtarlama İşaretinin Parametrelerinin Ayarlandığı Arayüz Gösterimi.



Şekil 2.19. *SUMFILTER* Bloğun, (a) FPA A Yazılımındaki Gösterimi, (b) Anahtarlamalı Kapasitör Devre Yapısı.

sunulduğu özellikle belirtilmelidir. Girişler eviren veya evirmeyen giriş olup, kullanılan fonksiyona göre bu girişler, toplanır veya çıkartılır. Her giriş programlanabilir bir kazanca sahiptir. Giriş voltajlarının toplamı doğru bir çıkış üretmek için programlanabilir köşe frekansına sahip, tek kutuplu alçak geçiren filtreden geçirilir. Köşe frekansı, alçak, yüksek ve band geçiren filtrelerde frekans cevabının genliğinin, tepe değerinin 3 dB altına indiği noktalardır ve kesim frekansı olarak da adlandırılabilir. *SUMFILTER* bloğun çıkış fonksiyonu aşağıda verilmiştir [6].

$$V_{out}(s) = \frac{2\pi f_0 [\pm G_1 V_{input1}(s) \pm G_2 V_{input2}(s) \pm G_3 V_{input3}(s)]}{s + 2\pi f_0} \quad (2.5)$$

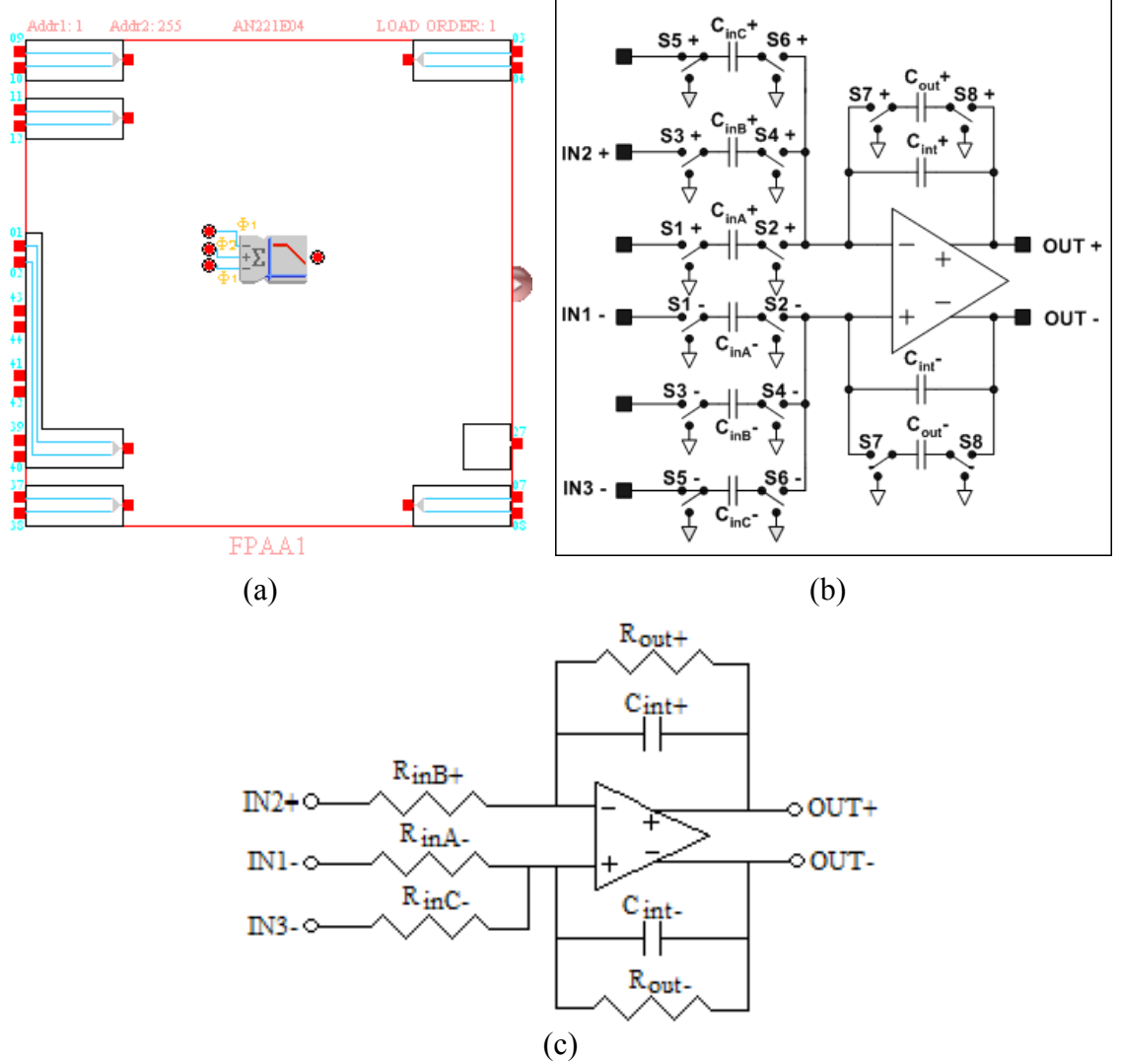
G_1 , G_2 , G_3 , girişlerin kazançlarını ifade ederken, V_{input1} , V_{input2} ve V_{input3} , girişlerdeki voltaj değerleridir. Her bir terimin işareti, CAM opsiyonundaki her giriş kolu için seçilen polariteye bağlıdır. Terimler, evirmeyen giriş için toplanırken eviren girişler için çıkartılır. Kapasitör değerleri, aşağıdaki bağıntılara göre belirlenir.

$$f_0 = \frac{f_c}{\pi} \frac{C_{out}}{2C_{int} + C_{out}} \quad (2.6)$$

$$G_1 = \frac{C_{inA}}{C_{out}}, \quad G_2 = \frac{C_{inB}}{C_{out}}, \quad G_3 = \frac{C_{inC}}{C_{out}} \quad (2.7)$$

Köşe frekans sınırları, örnekleme sırasında kullanılan anahtarlama işaretinin frekansı ile doğrusal olarak ilişkilidir. Mutlak sınırlar $f_c/1000$ ile $f_c/10$ arasındadır. Mutlak sınırlarla ilişkili olarak değişen kazanç değerleri de köşe frekans sınırlarının tayininde rol oynar.

Örnek olarak *SUMFILTER* bloğun nasıl çalıştığı burada izah edilmiştir. Bir adet eviren ve iki adet evirmeyen girişe sahip bu bloğun FPAA yazılım programındaki gösterimi Şekil 2.20(a)' da ve anahtarlama kapasitör devre yapısı Şekil 2.20(b)' de yer alırken eşdeğer devresi Şekil 2.20(c)' de yer almaktadır. Bu eşdeğer devre, *SUMFILTER* blok gibi hem toplama/çıkarma yapmakta hem de alçak geçiren filtre olarak girişindeki işaretleri süzmektedir.



Şekil 2.20. Eviren, Evirmeyen ve Eviren Girişe Sahip *SUMFILTER* Bloğun (a) FPAA Yazılım Programındaki Gösterimi, (b) Anahtarlama Kapasitör Yapısı, (c) Eşdeğer Devre Yapısı.

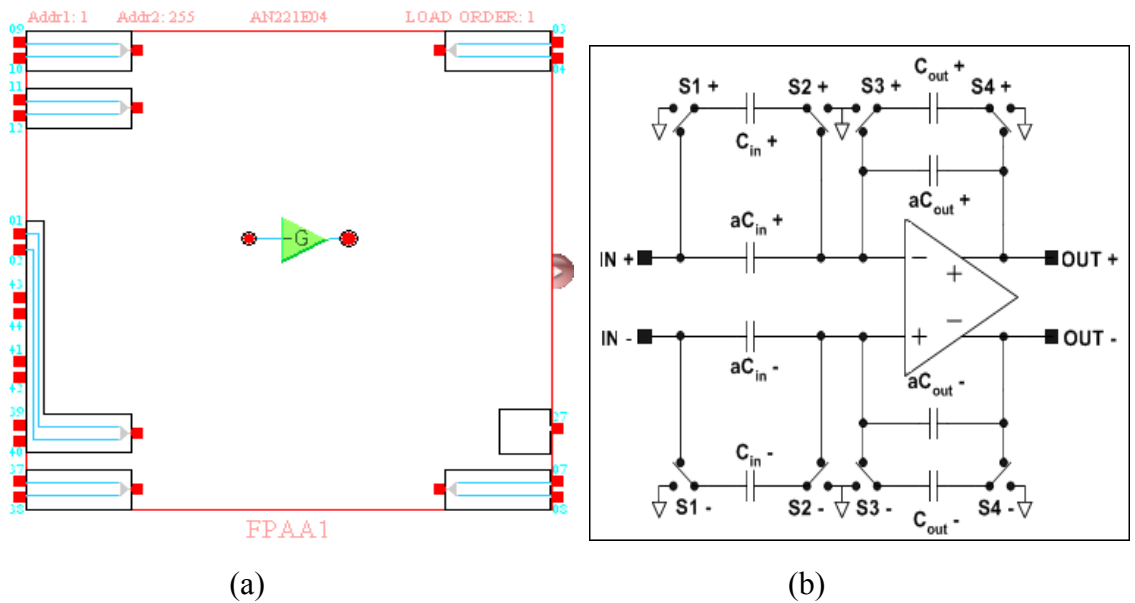
CAM'lerden bir diğeri, $GAININV$ bloğudur. Bu blok, eviren kazanç katı olup, FPAA gösterim şeması ve anahtarlama kapasitör devre yapısı Şekil 2.21' de verilmiştir. Bu blokta giriş voltajı, programlanabilir kazancın değeri ile ölçeklendirilir. Çıkış voltajı ile giriş voltajı arasında 180° faz farkı bulunmaktadır. Kazanç katı, sürekli bir giriş ve çıkışa sahiptir. Bu blok, 0.01~100 arasında ayarlanabilir kazanç imkanı sunar. $GAININV$ bloğuna ait transfer fonksiyonu aşağıda verilmiştir [6].

$$\frac{V_{out}(s)}{V_{in}(s)} = -G \quad (2.8)$$

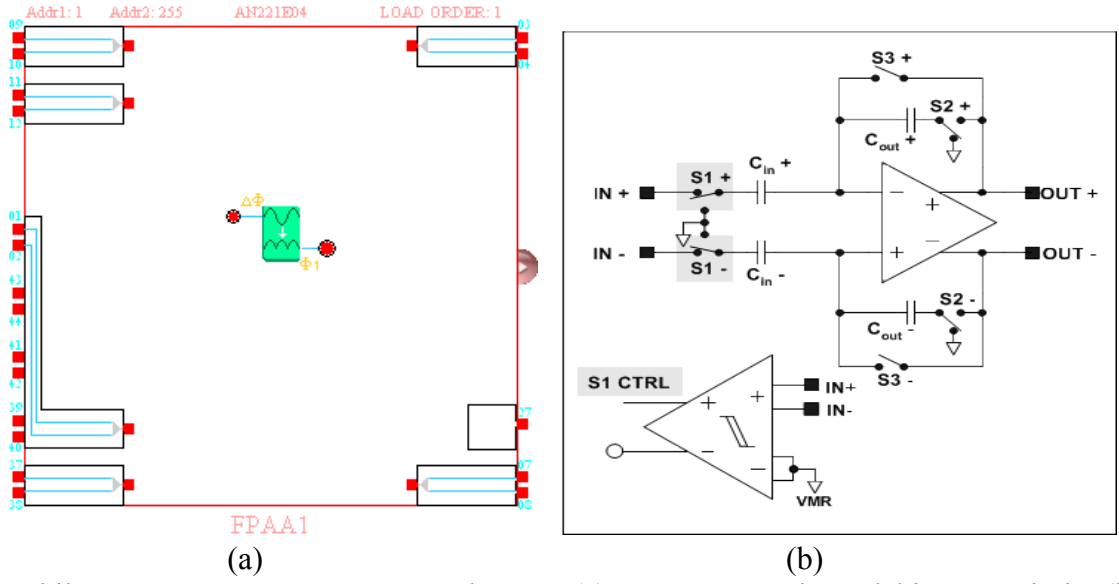
Bu blokta da kapasitör değerleri, Denklem (2.9)'daki ifadeye göre tayin edilir.

$$G = \frac{C_{in}}{C_{out}} \quad (2.9)$$

Diğer bir CAM modülü, $RECTIFIERHALF$ bloğudur ve FPAA gösterim şeması ile anahtarlama kapasitör eşdeğer devresi, Şekil 2.22' de verilmiştir. Bu blok, hem tam dalga doğrultucu bloğu hem de yarım dalga doğrultucu bloğu olarak kullanılabilir. Bloğun çıkışından giriş işareti ile aynı fazda ya da 180° faz farklı işaret almak mümkündür. Bu bloğun tam dalga doğrultucu versiyonu için çıkış fonksiyonu aşağıdaki gibidir [6]:



Şekil 2.21. $GAININV$ Bloğun, (a) FPAA Yazılımındaki Gösterimi, (b) Anahtarlama Kapasitör Devre Yapısı.



Şekil 2.22. *RECTIFIERHALF* Bloğun, (a) FPAAs Yazılımındaki Gösterimi, (b) Anahtarlamaalı Kapasitör Devre Yapısı.

$$V_{out} = \pm G |V_{in}| \quad (2.10)$$

Pozitif yarı dalgı doğrultucu için çıkış fonksiyonu;

$$V_{out} = \begin{cases} \pm G V_{input}, & V_{in} > 0 \\ 0, & V_{in} < 0 \end{cases} \quad (2.11)$$

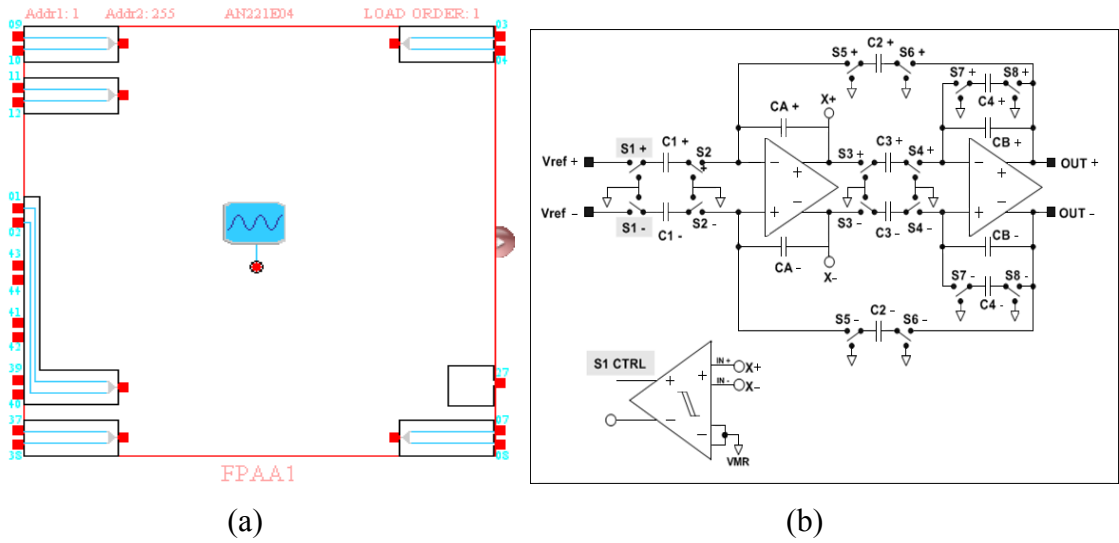
ile tanımlanırken negatif yarı dalgı doğrultucu için çıkış fonksiyonu aşağıdaki gibi tanımlıdır.

$$V_{out} = \begin{cases} 0, & V_{in} > 0 \\ \pm G V_{input}, & V_{in} < 0 \end{cases} \quad (2.12)$$

Yukarıdaki denklemlerdeki G , kazanç olup, önündeki \pm işareti, doğrultmanın eviren veya evirmeyen kuvvetlendirici ile yapıldığını belirtmektedir.

Şekil 2.23' de CAM modüllerinden *MULTIPLIER* bloğu gösterilmiş olup, bu blok çarpma işlemini gerçekleştirmektedir. X ve Y giriş işaretleri ve çarpım faktörü birbiriyle çarpılır. Bu bloğa ait çıkış fonksiyonu aşağıda verilmiştir [6].

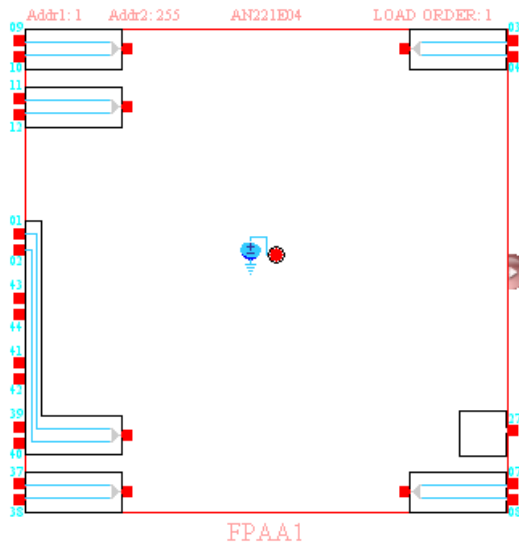
$$V_{out} = M \cdot V_X \cdot V_Y \quad (2.13)$$



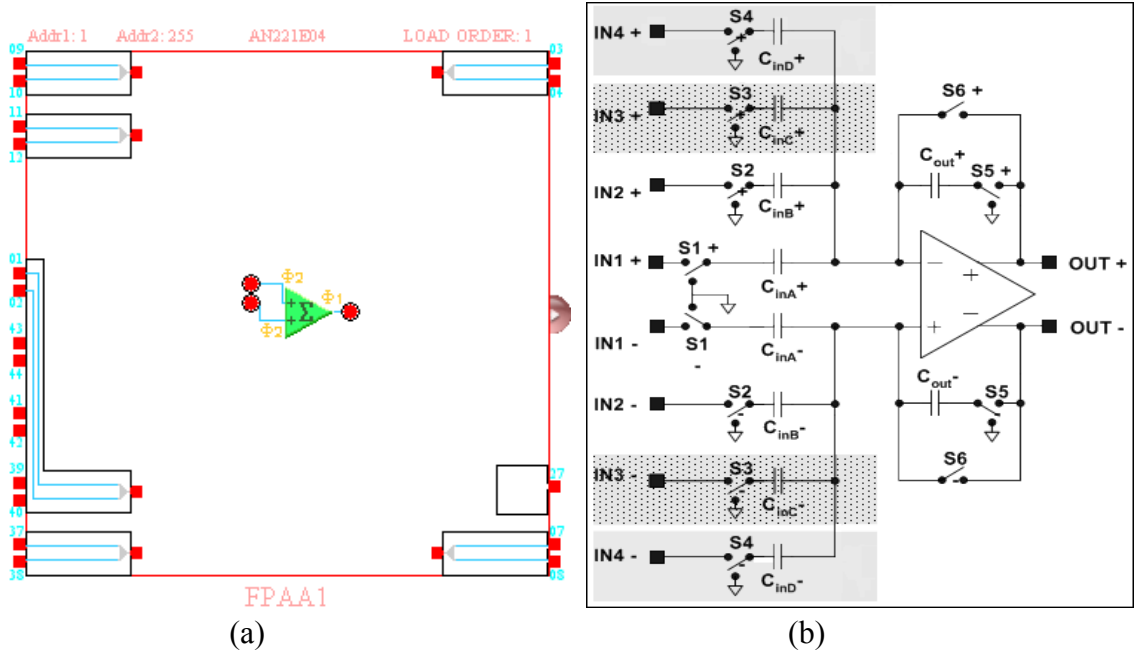
Şekil 2.24. *OSCILLATORSINE* Bloğun, (a) FPA A Yazılımındaki Gösterimi, (b) Anahtarlamalı Kapasitör Devre Yapısı.

Şekil 2.25' de AnadigmDesigner 2 yazılım programındaki gösterimi verilen *VOLTAGE* bloğu, $\pm 3V$ ' luk bir DC voltaj kaynağıdır.

SUMDIFF bloğu, CAM modüllerinden biri olup, dört girişe kadar yarım periyot toplama işlemini yapar. Hem toplama hem de fark alma işlemi transfer fonksiyonunda gerçekleştirildiği için girişler, ya evirendir ya da evirmeyendir. Bu bloğun her bir girişi programlanabilir bir girişe sahiptir. FPA A yazılım gösterimi ve anahtarlamalı kapasitör devre gösterim şeması Şekil 2.26' da verilen bu bloğa ait çıkış fonksiyonu aşağıda yer almaktadır [6].



Şekil 2.25. *VOLTAGE* Bloğun FPA A Yazılımındaki Gösterimi.



Şekil 2.26. *SUMDIFF* Bloğun, (a) FPAAs Yazılımındaki Gösterimi, (b) Anahtarlamalı Kapasitör Devre Yapısı.

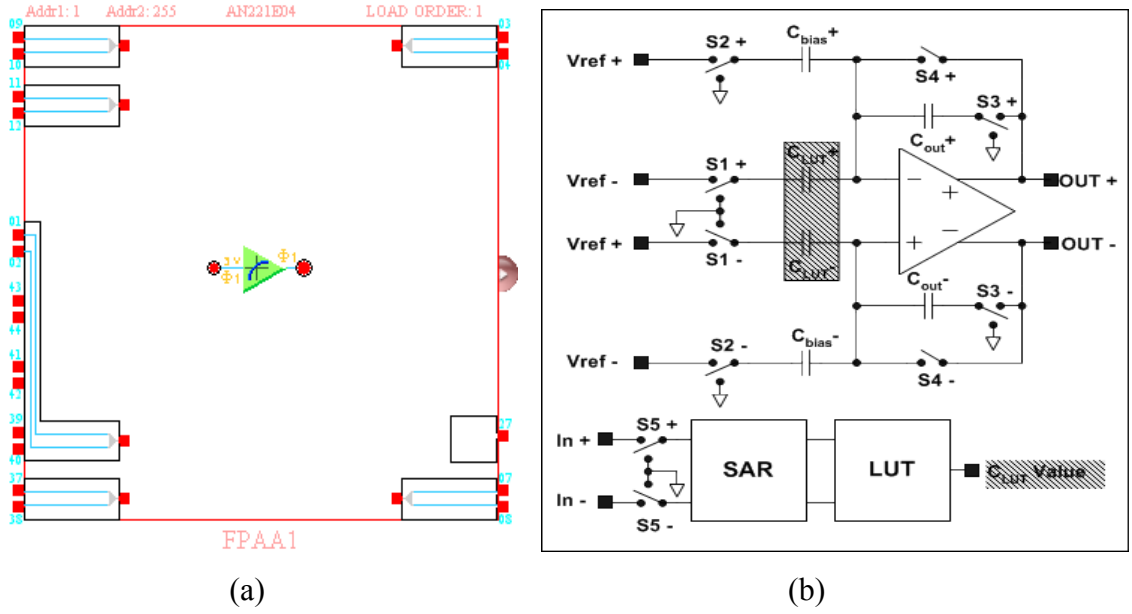
$$V_{out} = \pm G_1 V_{input1} \pm G_2 V_{input2} \pm G_3 V_{input3} \pm G_4 V_{input4} \quad (2.17)$$

G_1 , G_2 , G_3 ve G_4 , değişik giriş kollarının kazançlarını ifade ederken, V_{input1} , V_{input2} , V_{input3} ve V_{input4} , değişik giriş kollarındaki giriş voltajlarıdır. Denklemdaki üçüncü ve dördüncü terimler ihtiyaç halinde kullanılır. Her bir terimin işareti, CAM opsiyonundaki her giriş kolu için seçilen polariteye bağlıdır. Terimler, evirmeyen giriş için toplanır ve eviren girişler için çıkartılır. Kapasitör değerleri, aşağıdaki ifadelerle belirlenir.

$$G_1 = \frac{C_{inA}}{C_{out}}, \quad G_2 = \frac{C_{inB}}{C_{out}}, \quad G_3 = \frac{C_{inC}}{C_{out}}, \quad G_4 = \frac{C_{inD}}{C_{out}} \quad (2.18)$$

Kazancın üst sınırı anahtarlama işaretinin frekansının bir fonksiyonudur. Mutlak sınırlarla ilişkili olarak diğer kazanç değerleri, kazanç sınırının tayininde rol oynar.

Diğer bir CAM modülü de *TRANSFER FUNCTION* bloğudur. Bu blok, kullanıcı tarafından belirlenen LUT' a göre 256 adımlık bir çıkış voltajı üretir. Bu blok ile ilgili gösterimler Şekil 2.27' de verilmiş olup, bu bloğa ait çıkış fonksiyonu aşağıda yer almaktadır [6].



Şekil 2.27. *TRANSFER FUNCTION* Bloğun, (a) FPAAs Yazılımındaki Gösterimi, (b) Anahtarlamalı Kapasitör Devre Yapısı.

$$V_{out} = \frac{3(C_{LUT} - C_{bias})}{C_{out}} \quad (2.19)$$

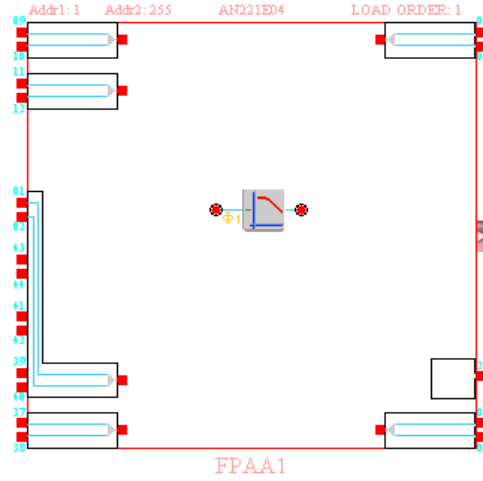
Bu blokta da kapasitör değerleri, aşağıda yer alan ifade ile tayin edilir.

$$C_{bias} = 128 \quad (2.20)$$

Tasarımda kullanılan voltaj transfer fonksiyonu, C_{out} değerini belirlemede ve C_{LUT} değerlerini LUT' a yüklemede kullanılır. Çalışma esnasında SAR, giriş voltajı tabanlı C_{LUT} ' a bu değerleri yazmak için LUT' u çalıştırır.

FILTERBILINEAR bloğu da CAM modüllerindeki filtre tiplerinden biridir ve FPAAs gösterimi Şekil 2.28' de verilmiştir. Bu blok, alçak geçiren, yüksek geçiren, tüm geçiren veya kutup ve sıfır noktalarına sahip olan tam periyotlu tek kutuplu bir filtredir. Bu filtre, programlanabilir köşe frekansına ve geçiş bandı kazancına sahiptir. Alçak geçiren filtrenin anahtarlamalı kapasitör devre modeli, Şekil 2.29' da verilmiş olup, transfer fonksiyonu aşağıda yer almaktadır [6].

$$\frac{V_{out}(s)}{V_{in}(s)} = \pm \frac{2\pi f_0 G}{s + 2\pi f_0} \quad (2.21)$$



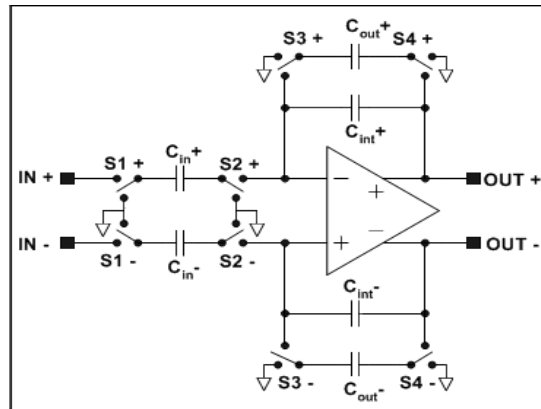
Şekil 2.28. *FILTERBILINEAR* Bloğun FPAA Yazılımındaki Gösterimi.

Burada G , geçiş bandı kazancıdır ve köşe frekansı f_0 , kazancın $-3+20\text{Log}G$ dB olduğu andaki frekanstır. Köşe frekansı ve kazanç ifadeleri Denklem (2.22)' de verilmektedir.

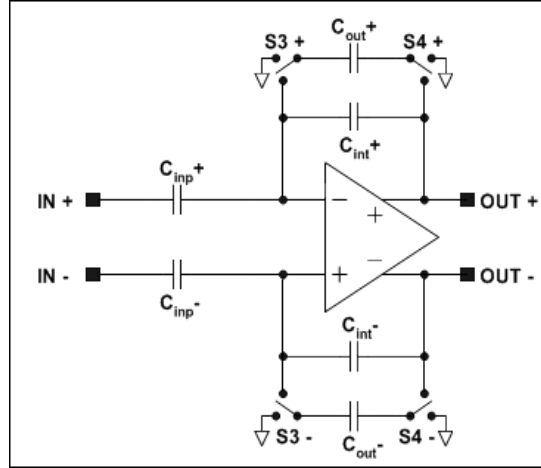
$$f_o = \frac{f_c}{\pi} \frac{C_{out}}{(2C_{int} + C_{out})}, \quad G = \frac{C_{in}}{C_{out}} \quad (2.22)$$

Köşe frekans sınırları, örnekleme sırasında kullanılan anahtarlama işaretinin frekansı ile doğrusal olarak ilişkilidir. Mutlak sınırlar $f_c/1000$ ile $f_c/10$ arasındadır. Mutlak sınırlarla ilişkili olarak değişen kazanç değerleri de köşe frekans sınırlarının tayininde rol oynar. Yüksek geçiren filtrede de bu sınırlar aynı şekildedir.

Yüksek geçiren filtre için anahtarlama kapasitör devre modeli, Şekil 2.30' da verilmiştir ve bu filtreye ait transfer fonksiyonu aşağıda yer almaktadır [6].



Şekil 2.29. *FILTERBILINEAR* Bloğun Alçak Geçiren Filtre Tipindeki Anahtarlama Kapasitör Devre Yapısı.



Şekil 2.30. *FILTERBILINEAR* Bloğun Yüksek Geçiren Filtre Tipindeki Anahtarlama Kapasitör Devre Yapısı.

$$\frac{V_{out}(s)}{V_{in}(s)} = -\frac{Gs}{s + 2\pi f_0} \quad (2.23)$$

Buradaki G ve f_0 , alçak geçiren filtrede ifade edildiği gibi geçiş bandı kazancıdır ve köşe frekansıdır. Bu filtrenin köşe frekans ve kazanç ifadeleri aşağıda verilmiştir.

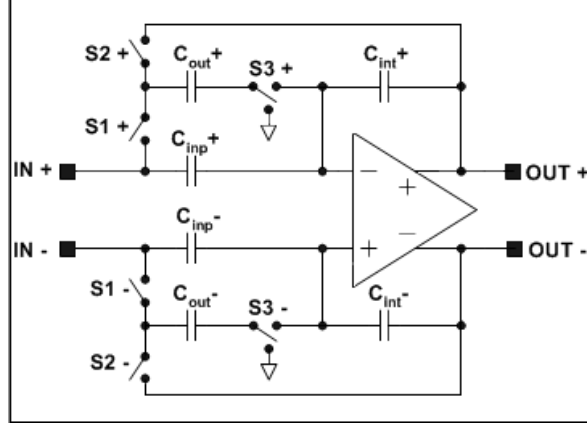
$$f_0 = \frac{f_c}{\pi} \frac{C_{out}}{(2C_{int} + C_{out})}, \quad G = \frac{2C_{inp}}{(2C_{int} + C_{out})} \quad (2.24)$$

Şekil 2.31' de anahtarlama kapasitör devre modeli gösterilen tüm geçiren filtrenin transfer fonksiyonu aşağıda verilmiştir [6].

$$\frac{V_{out}(s)}{V_{in}(s)} = -\frac{s - 2\pi f_0}{s + 2\pi f_0} \quad (2.25)$$

Burada köşe frekans f_0 , -90° faz kaydırma anındaki frekanstır. Bu filtre, çalışma frekans boyunca birim kazançta sahiptir. Bu filtrenin köşe frekans ifadesi aşağıda verilmiştir.

$$f_0 = \frac{f_c}{\pi} \frac{C_{out}}{(2C_{int} + C_{out})}, \quad C_{inp} = C_{int} \quad (2.26)$$



Şekil 2.31. *FILTERBILINEAR* Bloğun Tüm Geçiren Filtre Tipindeki Anahtarlamalı Kapasitör Devre Yapısı.

Tüm geçiren bu filtrede de köşe frekans sınırları, örnekleme sırasında kullanılan anahtarlama işaretinin frekansı ile doğrusal olarak ilişkili olup mutlak sınırlar $f_c / 1000$ ile $f_c / 10$ arasındadır. Bu filtre genelde bir işaret fazını kaydırmak için kullanılır. Denklem (2.27) ile verilen ifade, (f_{in}) işaret frekansında bir faz kaydırma için ihtiyaç duyulan f_0 köşe frekansının kullanıcı tarafından hesaplanmasını sağlar [6].

$$Faz\ kaydırma = 180 - \left[2 * \arctan\left(\frac{f_0}{f_{in}}\right) \right] \quad (2.27)$$

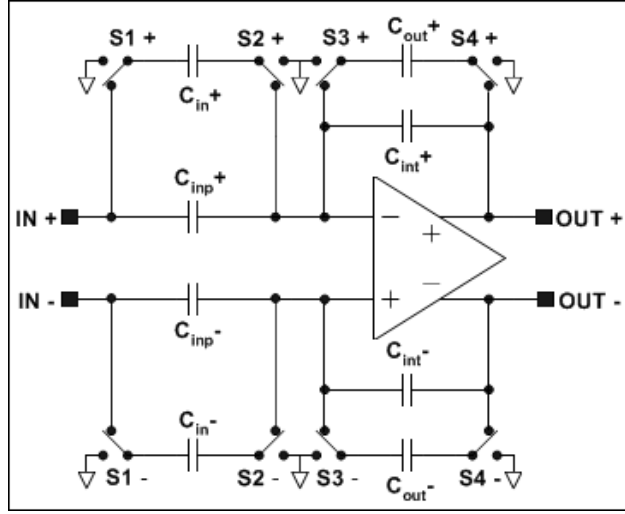
Bu denklem yeniden düzenlendiğinde [6];

$$f_0 = \tan\left[\frac{Faz\ kaydırma - 180}{2}\right] * f_{in} \quad (2.28)$$

elde edilir. Burada f_0 , filtre köşe frekansı ve f_{in} , işaret frekansıdır. Faz kaydırma sınırları, 90° ile 180° arasındadır.

Diğer bir kutup ve sıfır noktalarına sahip filtre, Şekil 2.32' de verilmiştir. Bu filtre tipi frekans bandındaki rezonans frekanslarını belirler. Sıfır noktası, sıfır empedans frekansını ve kutup noktası da yüksek empedans frekansını gösterir. Aşağıda bu filtre türüne ait transfer fonksiyonu yer almaktadır [6].

$$\frac{V_{out}(s)}{V_{in}(s)} = -\frac{G_H(s + 2\pi f_z)}{s + 2\pi f_p}, \quad G_L = \frac{f_z}{f_p} G_H \quad (2.29)$$



Şekil 2.32. *FILTERBILINEAR* Bloğun Kutup ve Sıfır Noktalarına Sahip Filtre Tipindeki Anahtarlama Kapasitör Devre Yapısı.

Burada G_L , DC kazanç, G_H , yüksek frekans kazancı, f_p , kutup frekansı ve f_z , sıfır frekansıdır. Sıfır frekansı, kutup frekansına göre büyük ya da küçük olabilir. Bu filtrenin kapasitörler ile frekans ve kazanç ilişkisi aşağıda verilmiştir.

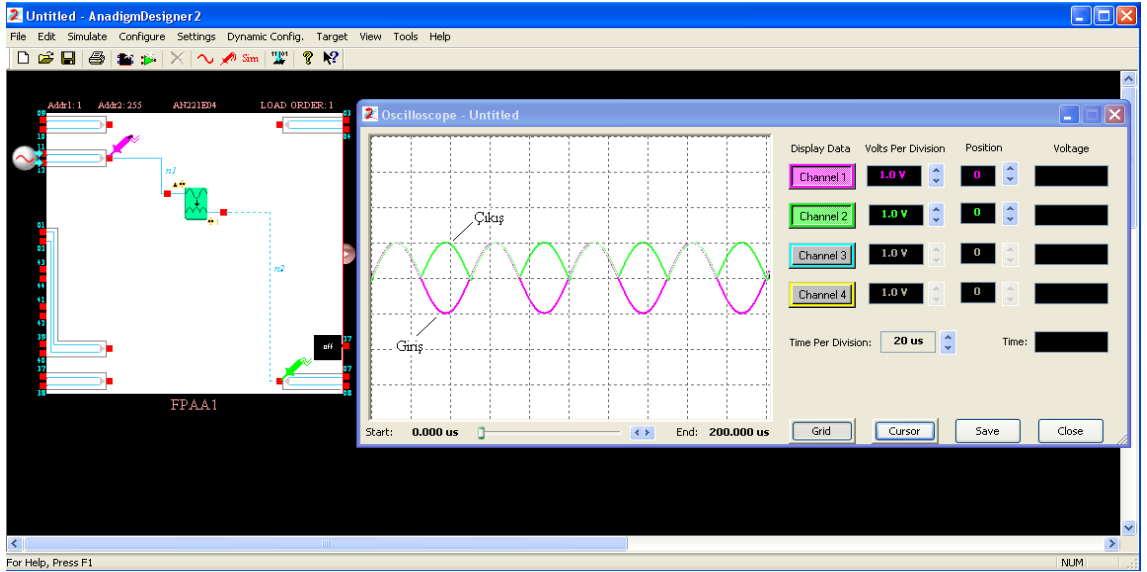
$$f_p = \frac{f_c}{\pi} \frac{C_{out}}{(2C_{int} + C_{out})}, \quad f_z = \frac{f_c}{\pi} \frac{C_{in}}{(2C_{inp} + C_{in})} \quad (2.30)$$

$$G_L = \frac{C_{in}}{C_{out}}, \quad G_H = \frac{(2C_{inp} + C_{in})}{(2C_{int} + C_{out})} \quad (2.31)$$

Kutup frekans sınırları ve sıfır frekans sınırları, örnekleme sırasında kullanılan anahtarlama işaretinin frekansı ile doğrusal olarak ilişkilidir. Mutlak sınırlar $f_c/1000$ ile $f_c/10$ arasındadır.

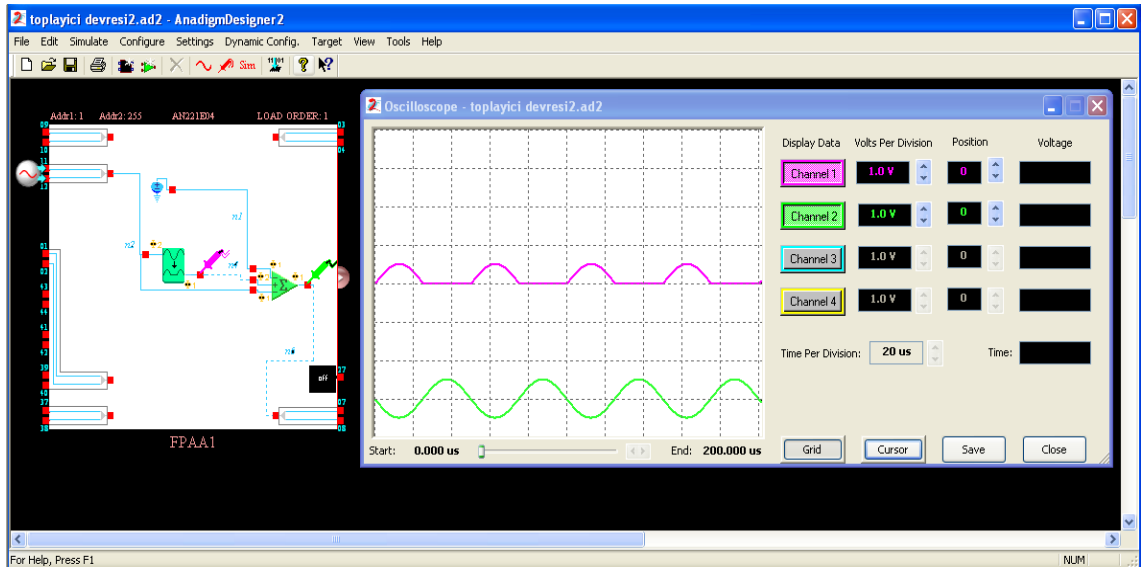
AnadigmDesigner 2 yazılımı, tasarımcının tasarladığı devreyi test etmesi için zaman domeni fonksiyonuna sahip bir simülasyon ihtiva etmektedir. Yazılımın bu özelliğinden yararlanılarak yukarıda anlatılan CAM modülleri veya kalan diğer modüllerin simülasyonu yapılabilir. Simülasyonun kullanıldığı iki örnek uygulama Şekil 2.33 ve 2.34' de yer almaktadır. Bu simülasyonla ilgili detaylı bilgi Bölüm 5.2' de verilecektir.

Birinci örnek uygulamada *RECTIFIERHALF* bloğu kullanılmış olup bu blok, tam dalga doğrultma işlemini gerçekleştirmektedir. Bu bloğun girişine tepe değeri 1 V ve frekansı 20 KHz olan sinüzoidal bir işaret uygulanmıştır. Simülasyon sonucunda elde edilen çıkış işareti giriş işareti ile eş zamanlı olarak çizdirilmiş olup Şekil 2.33' de verilmiştir.



Şekil 2.33. AnadigmDesigner 2 Yazılımının Simülasyon Özelliğini Gösteren Birinci Örnek Uygulamanın FPAA Yazılımındaki Gösterimi ve Simülatör Çıktısı.

İkinci örnek uygulamada *DC* ve *AC* işaretlerin toplanması/çıkarılması işlemi yapılmıştır. Bu uygulamada da *RECTIFIERHALF* bloğu kullanılmış olup bu blok sinüzoidal bir kaynaktan alınan işareti pozitif yarım dalga doğrultmaktadır. Doğrultulan işaret, *VOLTAGE* bloğundan alınan *DC* işaret ve sinüzoidal kaynaktan alınan işaret ile *SUMDIFF* bloğunda toplama/çıkarma işlemine tabii tutulmuştur. Pozitif yarım dalga doğrultulmuş işaret ve *SUMDIFF* bloğunun çıkışından elde edilen işaret Şekil 2.34' de verilmiştir.



Şekil 2.34. AnadigmDesigner 2 Yazılımının Simülasyon Özelliğini Gösteren İkinci Örnek Uygulamanın FPAA Yazılımındaki Gösterimi ve Simülatör Çıktısı.

3. BÖLÜM

ANALOG KAOTİK OSİLATÖRLER: MODELLEME, SİMÜLASYON VE TASARIM

3.1. Giriş

Kaos kelimesi Yunanca' dan türetilmiş olup, her şeyden önce var olan sonsuz boş uzay anlamına gelmektedir. Daha sonra Romalılar bu kavramı düzen ve uyum içeren, dünya mimarisinde orijinal işlenmemiş şekilsiz kütle olarak yorumladılar. Modern kullanımda kaos, düzensizliğin ve karışıklığın bir durumu olarak ifade edilmektedir [52].

Kaos teorisi, 1960' lı yıllarda Edward Lorenz tarafından hava tahmini yapılırken kelebek etkisinin ortaya çıkarılmasıyla incelenmeye başlanmıştır [53]. Lorenz, bilgisayarda uzun bir serinin çözümünü hesaplatmıştır. Hesaplama işlemi bittikten sonra yaptığı işlemlerin doğruluğunu kontrol ederken farkında olmadan giriş değerlerinde çok küçük değişiklikler yapmış ve yeniden hesaplatmıştır. Daha sonra bilgisayar çıktılarını gözden geçirdiğinde yeni bir sonuç kümesi elde ettiğini bulmuştur. Başlangıç koşullarındaki küçük farklılıkların sonuca etkisinin ihmal edilecek kadar küçük olduğunu düşünen Lorenz, düşüncelerinde yanılmıştır. Giriş değerlerindeki ihmal edilebilecek kadar küçük değişiklikler çok farklı sonuçların ortaya çıkmasına neden olmuştur. Bu durumdan dolayı Lorenz, uzun süreli hava tahminlerinin önceden tahmin edilmesinin mümkün olmadığını saptamıştır. Herhangi bir günde hava durumundaki küçük değişikliklerin, birkaç hafta sonra çok büyük değişikliklere yol açabildiğini ortaya çıkarmıştır. Kompleks bir sistemde bir yerdeki küçük bir değişikliğin sistemin başka bir yerinde büyük bir tepkiye neden olması şeklinde tanımlanabilen kelebek etkisi veya teoremi de rastlantı sonucunda bulunmasına rağmen kaos teorisinin başlangıcını oluşturmaktadır [54].

Günümüze kadar kaos ve kaotik olgu üzerine yaygın olarak matematikçiler, fizikçiler, biyolojiciler, tıpçılar, mühendisler çalışmışlar ve bugün de bu çalışmalar artan ivmeyle

devam etmektedir. Kaosun incelendiği başlıca alanlar arasında mühendislik, süreç kontrolü, haberleşme, işaret işleme, elektronik, tıbbi uygulamalar, yönetim ve finans alanları sayılabilir [55-57]. Kaosla ilgili yapılan çalışmaları daha da ileriye taşımak amacıyla matematiksel modeli ortaya konmuş bir kaotik sistemin, herhangi bir disiplin içerisinde hem varlığının tanımlanması hem de kontrollü bir şekilde kullanımı daha kolay olacaktır. Örneğin malzeme mühendisliğinde yeni materyallerin kullanıma sunulması, atomik düzeydeki kaotik hareketlerin matematiksel olarak modellenmesi ile mümkün olabilir. Bir diğer örnek insan vücudundan alınan EKG işaretlerindeki kaotik davranışlardır. Kalp krizi öncesi ortaya çıkan kaotik davranışların daha çabuk sezilmesi, belki de kalp krizinin ve devamında gelecek ölümlerin engellenmesi için bir araç olacaktır.

Bu bölümde ilk önce kaos ile ilgili kavramlardan kısaca bahsedilmiş, elektronik devrelerde kaos olayı üzerinde durulmuştur. Bu tez çalışmasında programlanabilir kaotik işaret üreteçlerin tasarımı ve gerçekleştirimi üzerinde durulduğu için bu programlanabilir üreteçler verilmeden önce literatürde var olan kaotik üreteçler, bu bölümde verilmiş ve bu üreteçlerin SIMULINK™ modellemeleri ile yapılan nümerik analizi ve ayrıık elemanlarla deneysel gerçekleştirimi de incelenmiştir.

3.2. Kaos İle İlgili Kavramlar

Matematiksel modele sahip bir sistemin kaotik olup olmadığını anlamak için dallanma (bifurcation) diyagramına, lojistik haritasına (logistic map), zaman, frekans ve faz gösterimlerine, Lyapunov üstellerine ve Poincare kesit gösterimine bakılabilir. Bu kavramlarla ilgili açıklayıcı bilgiler aşağıda verilmiştir.

3.2.1. Dallanma Diyagramı

Dinamik bir sistemde, sistemi matematiksel olarak tarif eden denklemlerin türevlerini sıfır yapan değerler, sistemin denge noktaları olarak tanımlanır. Bu dinamik sistemde, sistem parametreleri değiştirildiği zaman denge noktalarının kararlılığı, denge noktalarının sayısıyla birlikte değişebilir. Dinamik sistemdeki bu değişimler dallanma olarak ve bu değişimdeki parametreler de dallanma noktaları olarak adlandırılır [58]. Başlangıç koşullarına bağlı bir sistemin dallanma davranışı, kaos için farklı kollarda devam eder. Bu dallanma diyagramı, çoklu noktaya sahiptir. Periyodik bir sistemin

dallanma diyagramı ise bir noktaya sahiptir [59]. Kaotik sistemlerden Rössler sistemine ait dallanma diyagramı örnek olarak Şekil 3.1’ de verilmiştir. Bu dallanma diyagramı, Rössler sistemine ait d parametresinin 0 ile 2 arasında değişimi ile oluşturulmuştur.

3.2.2. Lyapunov Üstelleri

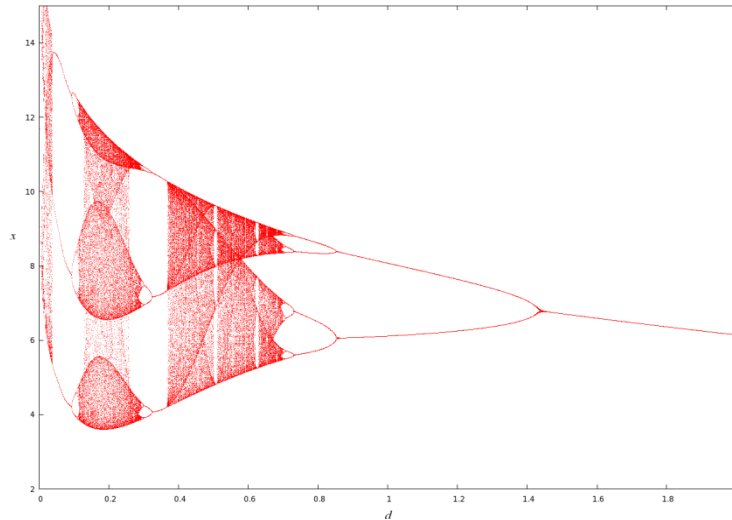
Başlangıç şartlarına üstel bağımlılığın anlamı, eğer $t = 0$ anında d_0 kadar küçük bir mesafeyle birbirinden ayrılmış iki başlangıç noktası alınırsa (t)’ nin artırılmasıyla bu noktalardan başlayan yörüngeler birbirinden üstel olarak uzaklaşacaktır. Bu durum grafiksel olarak Şekil 3.2’ de gösterilmiştir. İki yörünge arasındaki (d) mesafesinin (t)’ ye bağımlılığının, $t = t_0$ anındaki (d_0) başlangıç ayrışmasıyla birlikte, aşağıda verilen üstel bir fonksiyonla ifade edildiği kabul edilir [60].

$$d(t) = d_0(t_0)e^{\lambda t} \quad (3.1)$$

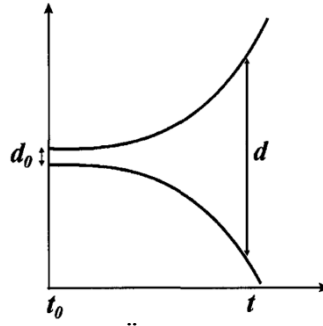
Denklemden yer alan λ sembolü; başlangıç şartlarında meydana gelen küçük değişimlere karşı hassasiyetin bir göstergesi olup, Lyapunov üsteli olarak isimlendirilir. Lyapunov üsteli aşağıdaki gibi tanımlanabilir [60].

$$\lambda = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \frac{d}{d_0} \quad (3.2)$$

Eğer $\lambda > 0$ ise, Şekil 3.2’ deki gibi yörüngeler birbirinden ayrı yönlerde doğru hareket ederler. Eğer $\lambda < 0$ ise yörüngeler birbirine yaklaşır [60]. Lyapunov üstellerinden en az



Şekil 3.1. Rössler Sistemine Ait Dallanma Diyagramı.

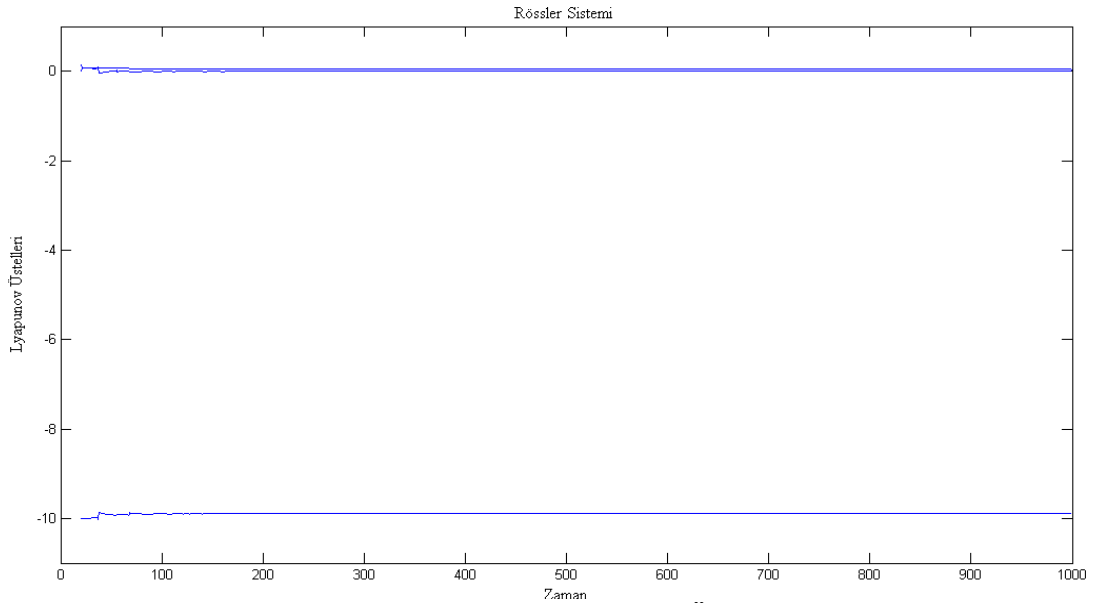


Şekil 3.2. Başlangıç Şartlarına Üstel Bağımlılığın Şematik Gösterimi.

birisi pozitif ise hareket kaotik olarak tanımlanır. Örnek olarak kaotik sistemlerden biri olan Rössler sistemine ait Lyapunov üstellerinin grafiksel gösterimi Şekil 3.3’ de verilmiştir. Şekilden de görüldüğü gibi bu sistemin Lyapunov üstelleri, -9.77, 0 ve 0.09 değerlerindedir.

3.2.3. Zaman / Frekans Gösterimleri

Dinamik bir sistemden elde edilen işaretlerin hem fiziksel olarak osiloskop ekranındaki, hem de benzetim veya nümerik analiz yapıldığında bilgisayar ekranındaki gösterimi, zaman domenindedir. Eğer sistem doğrusal bir davranış sergiliyorsa sistemden elde edilen işaretler periyodik olarak adlandırılır. Sistem davranışı, sistemin başlangıç şartları ile değişiyorsa ve sınırlı yenileyen davranışlar sergiliyorsa bu davranışlar, kaotik işaretlerdir. Frekans gösterimi ise sistemdeki işaretlerin zaman domeni yerine frekans

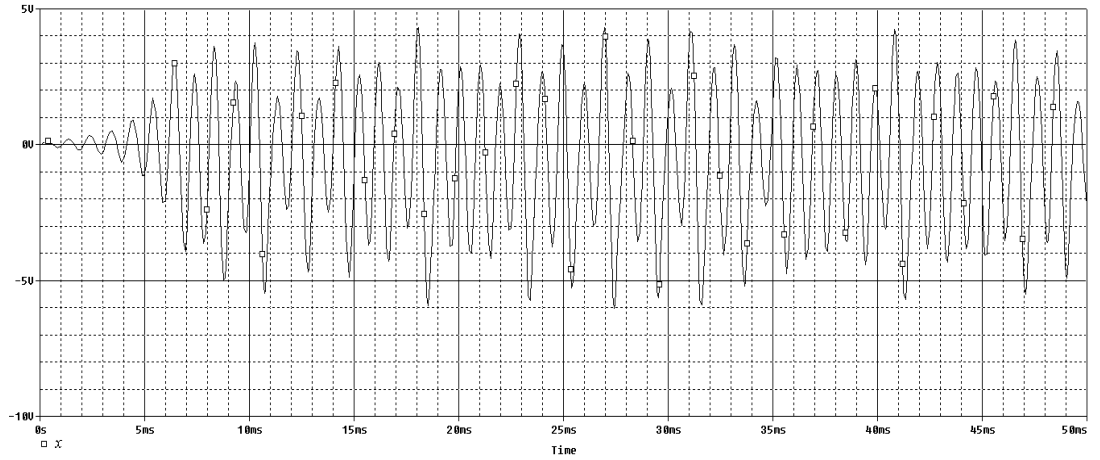


Şekil 3.3. Rössler Sistemine Ait Lyapunov Üstellerinin Çizimi.

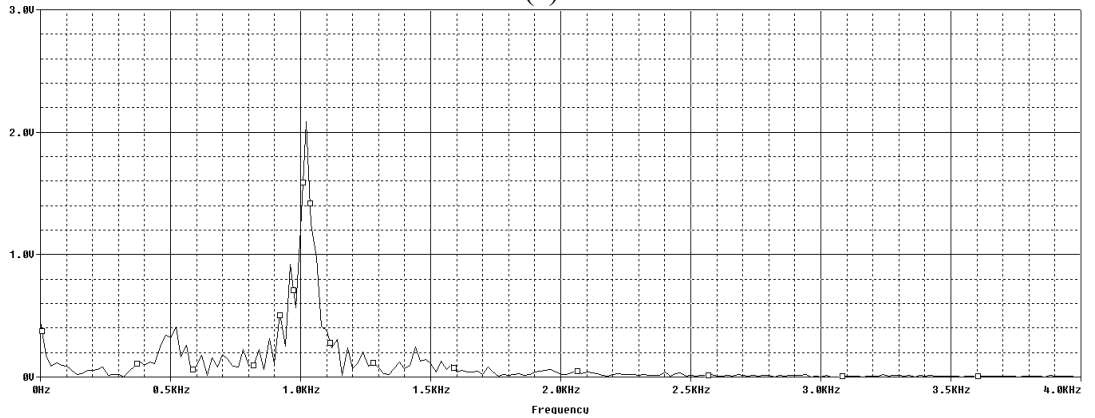
domeninde gösterimidir. Periyodik özelliğe sahip bir sistemin frekans cevabı, frekans ekseninde tek bir çizgi tayfi şeklinde olup, kaotik bir sistemin frekans cevabı, yayılı bir spektruma sahip olarak elde edilir. Şekil 3.4’ de zaman/frekans gösterimine örnek olarak Rössler sisteminin benzetiminden elde edilen sonuçlar verilmiştir.

3.2.4. Faz Düzlemi Gösterim

Faz düzlemi gösterimi, dinamik bir sistemden elde edilen zaman bağlı işaretlerin grafik olarak gösterimidir. İncelenen sistem doğrusal ise, sistemden elde edilen faz gösteriminin kapalı bir çevrim oluşturduğu yani sistemin periyodik hareket sergilediği görülür. İncelenen sistem kaotik bir sistem ise faz-uzay gösterimi düzensiz ve anlamsız bir şekle sahip olup, bu tip şekiller kaosun var olduğu anlamına gelmektedir. Faz düzlemi gösterimine örnek olarak Rössler sisteminin benzetiminden elde edilen çeker (attractor) gösterimi, Şekil 3.5’ de yer almaktadır.

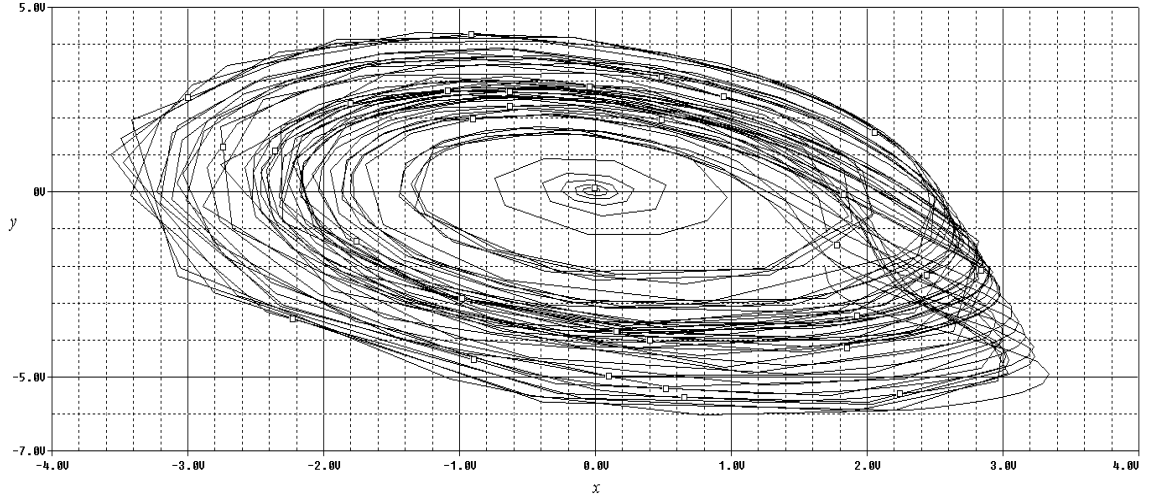


(a)



(b)

Şekil 3.4. Rössler Sisteminin Benzetiminden Elde Edilen $x(t)$ Dinamiğinin, (a) Zaman Domeninde Gösterimi, (b) Kaotik Frekans Spektrumu.



Şekil 3.5. Rössler Sisteminin Benzetiminden Elde Edilen Faz Düzlemindeki Gösterimi.

3.2.5. Lojistik Harita (Logistic Map)

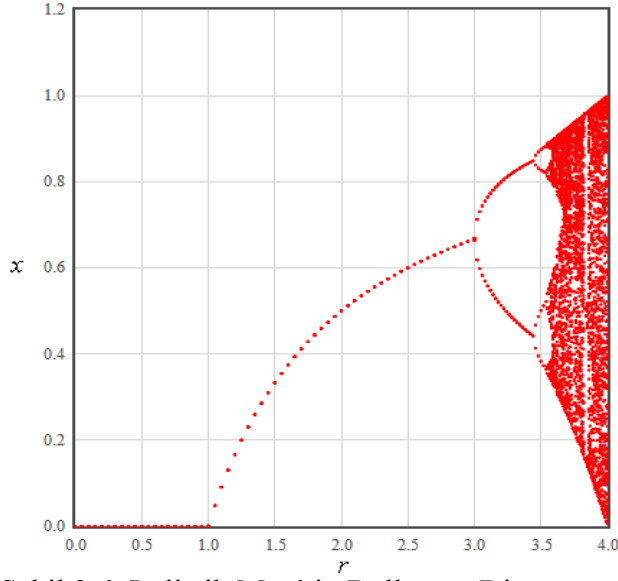
1976 yılında Robert May, bir makalesinde basit doğrusal olmayan haritaların, çok karmaşık dinamiklere sahip olabileceğinden bahsetmiştir [59]. May, lojistik harita olarak adlandırdığı bu haritayı, popülasyondaki böcek türlerinin yıllık değişimlerini göstermek için ekolojik bir model oluşturmada kullanmıştır. Bu modele göre $(n+1)$ ' inci yıldaki popülasyon, (n) ' inci yıldaki popülasyon ile aşağıdaki bağıntıyla matematiksel olarak ilişkilidir [61].

$$x_{n+1} = r x_n(1-x_n), \quad (3.3)$$

Burada (x) , $0 \leq x \leq 1$ ve (r) , $0 \leq r \leq 4$ ' dür. (r) , kontrol parametresi olup bu parametrenin değişimi ile Denklem (3.3) ile ifade edilen sistemin davranışı da değişmektedir. (r) parametresi $0 \leq r \leq 1$ arasında iken sistem $x=0$ durumundadır. (r) , $1 \leq r \leq 3$ arasında olduğunda (x) periyodik çalışma modunda olup $(r)=3$ değerinde, periyot-2 davranışını sergilemektedir. $3 \leq r \leq 3.57$ aralığında (x) ' in davranışı 4, 8, 16 gibi periyodik katlanmalar şeklinde olurken, $3.58 \leq r \leq 4$ arasında (x) , kaotik çalışma modundadır. (r) parametresine bağlı olarak sergilenen davranışlar Şekil 3.6' da verilmiştir [61].

3.2.6. Poincare Kesit Gösterimi

Henri Poincare tarafından kaotik davranışa sahip bir sistemin faz-uzay gösterimini daha anlaşılır hale getirmek için ortaya konan Poincare kesiti, dinamik bir sistemden enlemesine alınan bir kesittir [62]. Bu kesitteki faz noktalarına da Poincare mapping adı

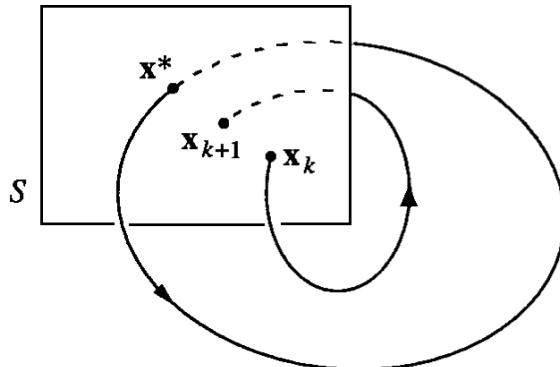


Şekil 3.6. Lojistik Map' in Dallanma Diyagramı.

verilmektedir. $\dot{x} = f(x)$ denkleminde sahip n -boyutlu bir sistem düşünölsün. Şekil 3.7' de gösterilen (S), n boyutlu sistemi enlemesine kesen ($n-1$) boyutlu kesit yüzeyi olup, Poincare gösterimi aşağıdaki gibi ifade edilebilir:

$$x_{k+1} = P(x_k) \quad (3.4)$$

P , Poincare gösterimidir ve $(x_k) \in S$ olup (k)' nıncı kesişimdir. x^* , P ' nin sabit noktası ve $P(x^*) = x^*$ durumunda (x^*) ' da başlayan eğri, (x^*) için T zamanı boyunca döner ve böylece $\dot{x} = f(x)$ orijinal sistem için kapalı bir eğri oluşmuş olur. Bu durumda sistem, periyodik bir sistem olarak yorumlanır [59]. Eğer sistem, kapalı bir eğri değil de belirli alanlarda yoğunlaşmış kümeler şeklinde noktalardan oluşmuş ise sistem kaotik sistem olarak adlandırılır.



Şekil 3.7. Poincare Kesit Gösterimi.

3.3. Elektronik Devrelerde Kaos ve Analog Kaos Üreteçleri

Elektronik devrelerde kaos olayının varlığı son 35-40 yıl içerisinde ortaya çıkmıştır. Daha önceleri de, gerçekleştirilen elektrik ve elektronik devrelerde doğrusal olmayan davranışlar gözlenmiş ama kavramsal olarak tanımı yapılamadığı için bu davranışlar, gürültü olarak nitelendirilmiştir. Buna en çarpıcı örnek, Hollandalı elektrik mühendisi ve fizikçisi Van der Pol' ün harici kaynakla sürülen bir neon lamba osilatörü üzerinde yaptığı deneysel çalışmadır. Van der Pol, 1927 yılında Nature Magazine adlı dergide [63] çıkan makalesinde, bu devreyle ilgili çalışmaları sırasında sıklıkla düzensiz bir davranış gözlemlediğini ve bu davranışı, gürültü olarak adlandırdığını ifade etmiştir. 1980' li yıllarda Van der Pol' un çalışması tekrar irdelenmiş ve gürültü olarak adlandırılan davranışların aslında kaotik bir devre dinamiği olduğu ortaya çıkarılmıştır [64, 65].

Kaos ilk olarak otonom olmayan devrelerden Van der Pol osilatörü, neon lamba osilatörü, zorlamalı dirençli osilatör devreleri ile direnç, indüktör ve diyottan oluşan osilatör devrelerinden elde edilmiştir [63-68]. Otonom devreler üzerine literatürde çok sayıda çalışma mevcuttur ve bu çalışmalarda ilgiyi en çok gören devreler, Chua osilatörü, Lorenz ve Rössler sistemleridir [69]. Chua osilatörünün, 1983 yılında Leon O. Chua tarafından geliştirilmesi neticesinde elektronik devrelerde kaosun varlığı hem matematiksel olarak hem benzetim neticesinde hem de deneysel olarak ilk defa ortaya konmuştur [70].

Son 30 yılda kaos ile ilgili yapılan çalışmalarda, basit RLC ve RC devreleri, anahtarlamalı kapasitör devreleri, faz kilitlemeli döngü (Phase Locked Loop, PLL) yapıları, güvenli haberleşme sistemleri, kaotik modülatör ve demodülatörler, adaptif filtreler, sayısal filtreler ve güç devreleri gibi pek çok elektronik devre ve sistem irdelenmiştir [55-57, 65, 68, 71-78]. Bu kısımda literatürde yer alan ve üzerinde en çok çalışılan analog kaotik işaret üreteçleri incelenmiş, SIMULINK™ modellemeleri ve ayrıık elemanlar kullanılarak deneysel gerçekleştirmeleri yapılmıştır.

3.3.1. Analog Kaotik İşaret Üreteçler

Kaotik işaret üreten devreler, kaos tabanlı uygulamaların en önemli tasarım aşamasını oluşturmaktadır. Çünkü herhangi bir uygulamada kaos dinamiklerinden yararlanmak için öncelikle bu dinamiklerin pratik olarak üretilmesi gerekmektedir. Bununla birlikte

literatürde özellikleri çok iyi bilinen ve avantajlı bir şekilde kullanılan kaotik işaret üreteçlerinin sayısı sınırlıdır.

Kaotik bir sistemin matematiksel modeli ve nümerik analizi, o sistemin fiziksel olarak gerçekleştirilmeden önce tasarım parametrelerinin önceden belirlenmesinde ve sistemin analizinde çok önemli bir rol oynamaktadır. Matematiksel modeli ortaya konan ve nümerik analizi başarıyla yapılan bir kaotik sistemin deneysel olarak da gerçekleştirimi kaotik işaret üreteç tasarım çalışmalarını ileriye taşır.

Literatürde çeşitli nümerik analiz yöntemleri bulunmakla birlikte, bu tez çalışmasında, MATLABTM/SIMULINKTM kullanılarak nümerik analizler yapılmıştır. SIMULINKTM, nümerik analiz özelliğinin yanı sıra grafiksel programlama özelliğine sahip olan, hem doğrusal hem de doğrusal olmayan sistemlerin modellenmesinde kullanılan etkili bir programdır. Grafiksel programlama özelliğine sahip SIMULINKTM, de analiz edilecek herhangi bir model, komut satırlarına ihtiyaç duyulmaksızın, basit blok yapılar ile oluşturulur. Bu sayede diferansiyel denklem takımına sahip bir kaotik sistem de, sistemi ifade eden diferansiyel denklemlerin komut satırı yerine görsel öğelerden oluşan uygun blokların kullanılmasıyla kolaylıkla modellenebilir, grafiksel olarak gerçekleştirilebilir ve elde edilen sonuçlar analiz edilebilir [79].

Aşağıdaki kısımlarda literatüre girmiş ve üzerinde en çok çalışma yapılan analog kaotik osilatör devreleri üzerinde durulmuştur. Öncelikle bu sistemlerin modelleme çalışmaları ve nümerik analizleri yapılmıştır. Bu modellenen ve analizleri yapılan sistemlerin fiziksel olarak da yapılabileceğini göstermek için incelenen osilatör devrelerinin ayrı elemanlarla deneysel gerçekleştirmeleri yapılmıştır. Bu deneysel gerçekleştirmelerde fiziksel büyüklükleri ve parazitik etkileri nedeniyle indüktör elemanı yerine aktif elemanlardan ve RC yapılardan oluşan indüktör simülatörleri kullanılmıştır. Ayrıca deneysel çalışmaları yapılacak sistemlerin nümerik analiz sonuçlarında sistem parametrelerinin maksimum değeri, devre besleme voltajının üzerinde ise o sisteme devre gerçekleştirimi yapılmadan önce ölçeklendirme işlemi uygulanmıştır. Bu işlemle devre parametrelerinin maksimum değerleri, besleme voltajından daha küçük değerlere gelmiş olup devre tasarımı ona göre yapılmıştır.

3.3.1.1. Chua Osilatörü

Bugüne kadar tasarlanan kaotik sistemler arasında Chua osilatörünün özel bir yeri vardır [69]. Basit bir devre yapısına sahip olması ve kaotik davranış sergilemesi dolayısıyla Chua osilatörü, kaos ve kaotik işaret uygulamalarında model yapı olarak kabul edilmektedir [80]. Chua osilatörü aşağıdaki durum denklemleri ile ifade edilmektedir:

$$\begin{aligned}\frac{dx}{d\tau} &= \alpha(y - x - f(x)) \\ \frac{dy}{d\tau} &= x - y + z \\ \frac{dz}{d\tau} &= -\beta y\end{aligned}\tag{3.5}$$

Burada x , y ve z , sistemin durum değişkenleri iken α ve β , sistem parametreleridir. $f(x)$, doğrusal olmayan fonksiyon olup, sistemin kaosa girmesinde önemli bir rol oynar. Kaotik Chua osilatörünün gerçekleştiriminde, parçalı doğrusal fonksiyon [81, 82], kübik fonksiyon [83, 84], parçalı ikinci dereceden fonksiyon [55, 85] ve trigonometrik fonksiyonlar [55, 86] gibi farklı doğrusal olmayan fonksiyonlar kullanılmıştır. Chua osilatörünün modellenmesinde en çok tercih edilen fonksiyonlar ve bu fonksiyonların parametre değerleri, Tablo 3.1' de yer almaktadır. Tabloda yer alan $NR1$, $NR2$ ve $NR3$, Chua osilatörünün SIMULINK™ modellemesinde sırasıyla Denklem (3.6), (3.7) ve (3.8)' i kullanarak oluşturulan doğrusal olmayan blokların adlarıdır.

Doğrusal ve doğrusal olmayan dinamik sistemlerin nümerik analizinde MATLAB™/SIMULINK™ analiz yöntemi aktif olarak kullanılmaktadır. SIMULINK™, grafiksel programlama özelliğine sahiptir. Chua osilatörüne ait

Tablo 3.1. Kaotik Chua Osilatörünü Modellemede En Çok Kullanılan Doğrusal Olmayan Fonksiyonlar.

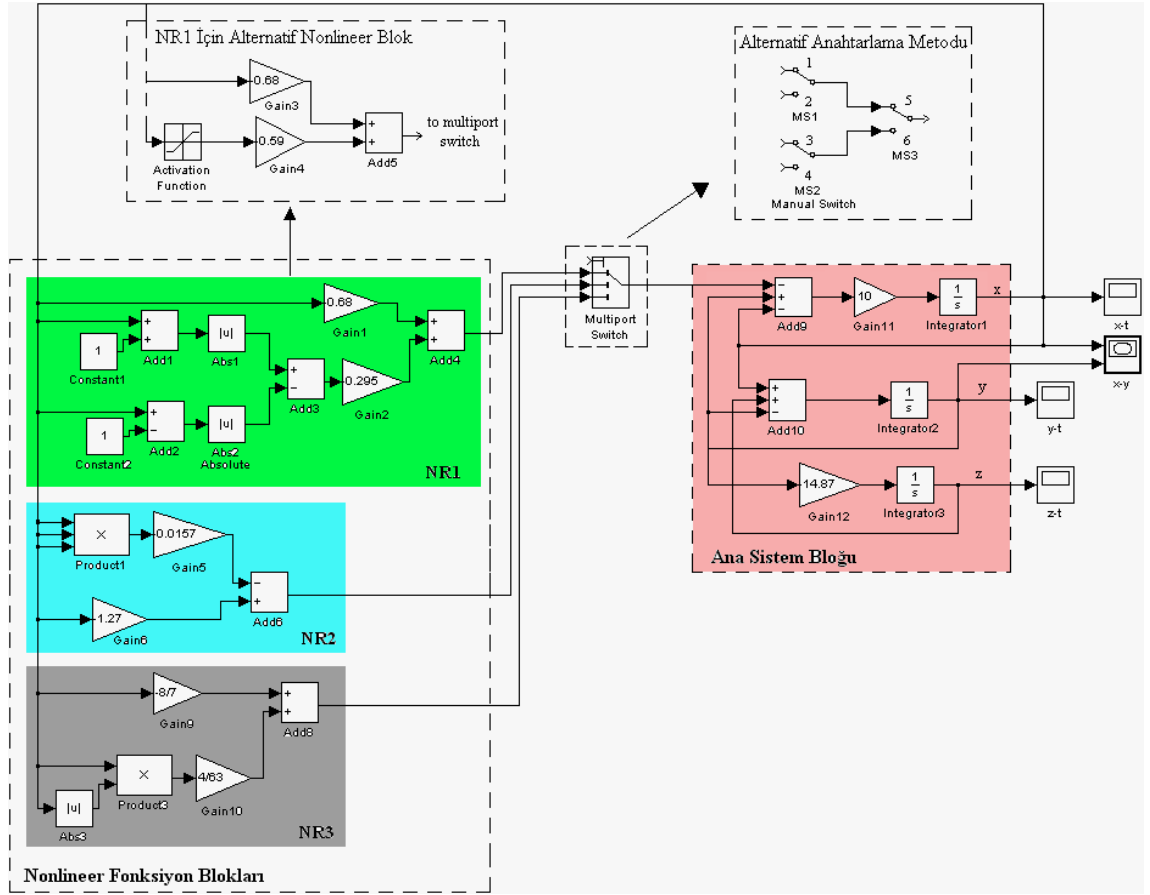
Doğrusal Olmayan Fonksiyon	Fonksiyon Parametreleri	SIMULINK™ Modelindeki Blok Adı
$f(x) = b x + 0.5(a - b)(x + c - x - c)$ (3.6)	$a = -1.27, b = -0.38, c = 1$	$NR1$
$f(x) = k_1 x - k_2 x^3$ (3.7)	$a = -1.27, b = -0.0157$	$NR2$
$f(x) = d_1 x + d_2 x x $ (3.8)	$d_1 = -8/7, d_2 = 4/63$	$NR3$

SIMULINK™ modeli, Şekil 3.8' de gösterilmektedir [79]. Bu model Denklem (3.5)-(3.8)' deki matematiksel tanımlamalara göre oluşturulmuştur. Şekilden görüldüğü gibi sistem, bir adet ana sistem bloğundan, bir adet çoklu anahtarlama mekanizmasından ve üç adet *NR1*, *NR2*, *NR3* olarak isimlendirilen doğrusal olmayan fonksiyon bloklarından oluşmaktadır. Buradaki çoklu anahtarlama mekanizması, ana sistem bloğu ile doğrusal olmayan fonksiyon blokları arasındaki bağlantıyı sağlamaktadır. Ana blok, Denklem (3.5)' i gerçekleyen integral alma bloklarından oluşmaktadır. Şekil 3.8' de *NR1*, *NR2* ve *NR3* ile gösterilen doğrusal olmayan fonksiyonlar, Tablo 3.1' de verilen denklemlerdeki matematiksel ifadelerle uygun yapı blokları kullanılarak gerçekleştirilmiştir. Tablo 3.1' de yer alan ve Denklem (3.6) ile tanımlanan *NR1*, parçalı doğrusal fonksiyon olup kazanç blokları, mutlak değer alma blokları, toplama/çıkarma bloğu ve sabit değer bloğu kullanılarak oluşturulmuştur. Ayrıca parçalı doğrusal fonksiyonunun alternatif bir gerçekleştirimi, Şekil 3.8' de gösterildiği gibi aktivasyon fonksiyon bloğu kullanılarak da yapılabilir.

NR2 olarak adlandırılan ve kübik doğrusal olmayan fonksiyon, Denklem (3.7) ile tanımlanmış olup kazanç bloklarından, çarpma bloğundan ve toplama/çıkarma bloğundan oluşmaktadır. *NR3* ise Denklem (3.8) ile ifade edilen parçalı ikinci dereceden doğrusal olmayan fonksiyondur. Bu fonksiyonun gerçekleştiriminde kazanç blokları, mutlak değer alma bloğu, çarpma bloğu ve toplama bloğu kullanılmıştır.

SIMULINK™ programı kullanılarak yapılan nümerik analizler neticesinde, kaotik Chua osilatörüne ait zaman domenindeki kaotik davranışlar ve $(x-y)$ düzlemindeki çift çeker gösterimleri elde edilmiştir. Bu sonuçlara ilaveten her bir doğrusal olmayan fonksiyon bloğunun DC karakteristiği de incelenmiştir. Nümerik analiz sonucunda elde edilen DC karakteristikler ve çeker gösterimleri Şekil 3.9' da verilmiştir. Şekilden görüldüğü gibi, Chua osilatörü farklı tipte doğrusal olmayan fonksiyonlar kullanılarak gerçekleştirilmesine rağmen devre davranışı orijinalliğini bozmamakta ve çeker yapısını oluşturmaktadır.

Kaotik Chua osilatörünü, trigonometrik formda doğrusal olmayan fonksiyon ile de gerçekleştirmek mümkündür. Bu gerçekleştirimde kullanılan denklem takımı aşağıda verilmiştir [86].



Şekil 3.8. Kaotik Chua Osilatörünün SIMULINK™ Modeli.

$$\frac{dx}{d\tau} = \alpha(0.63y - 0.63x - f(x))$$

$$\frac{dy}{d\tau} = 0.63x - 0.63y + z$$

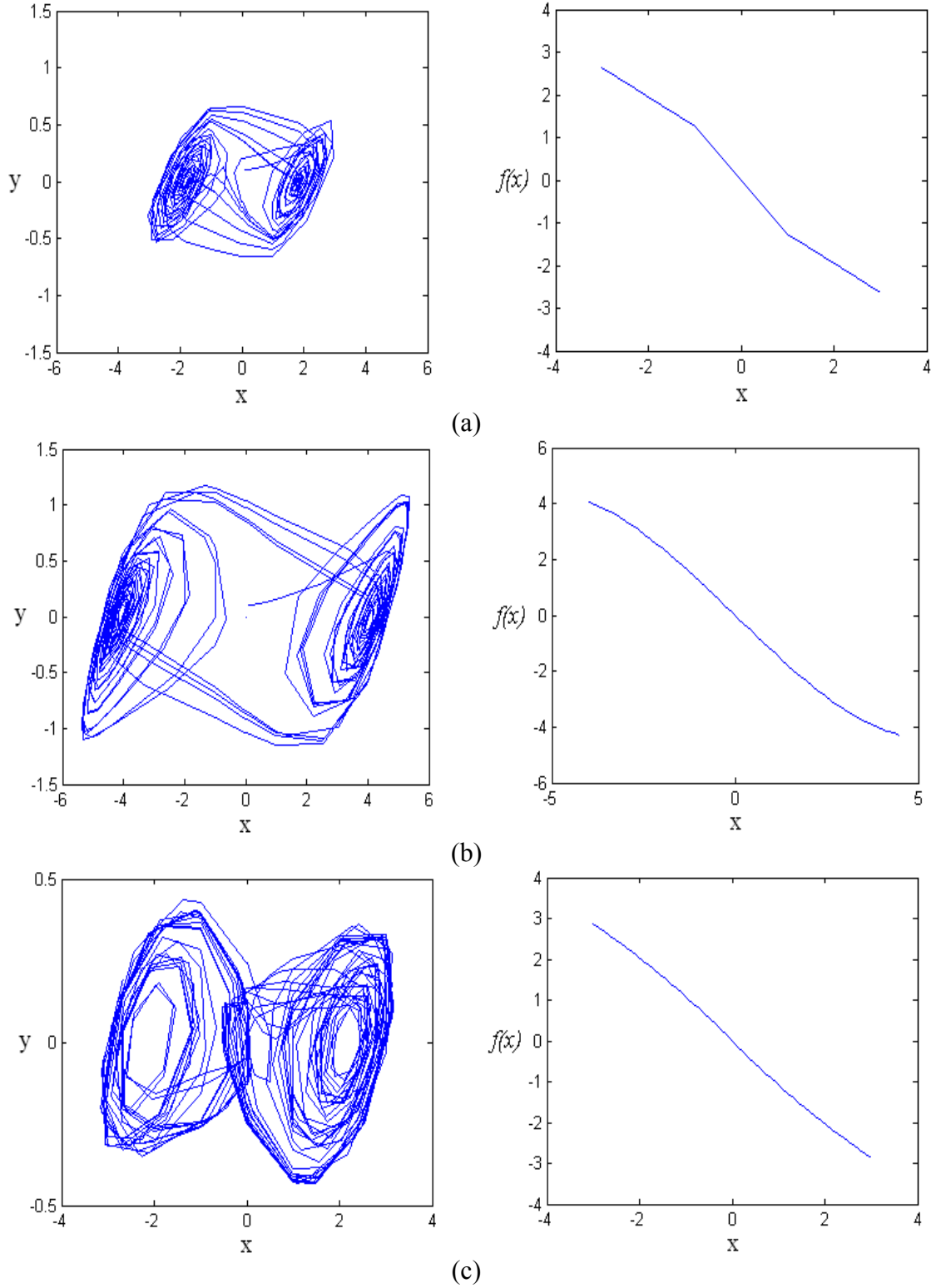
$$\frac{dz}{d\tau} = -\beta y$$

$$f(x) = -a \tanh(bx)$$

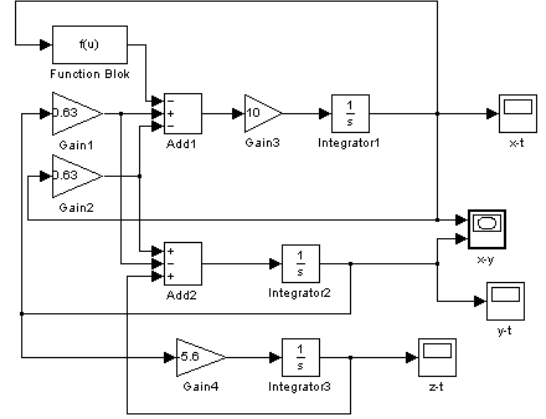
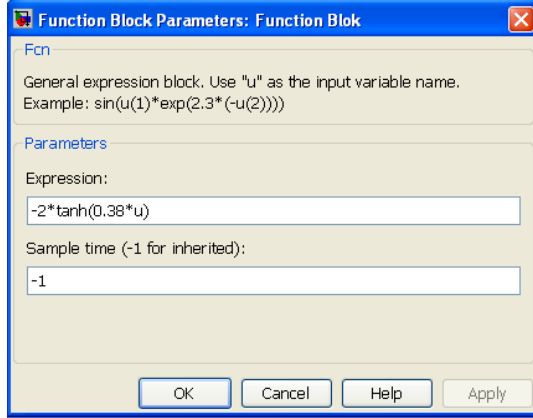
(3.9)

Denklem (3.9)' da verilen α , β , a ve b sistem parametreleri olup, sırasıyla 10, 5.6, 2 ve 0.38 değerlerine sahiptirler. Bu osilatörün SIMULINK™ ile nümerik analizi yapılabilir.

Bu doğrusal olmayan fonksiyona sahip Chua osilatöründeki x , y ve z ifadelerindeki katsayılar, $NR1$, $NR2$ ve $NR3$ ' ün kullanıldığı Chua osilatöründeki x , y ve z ifadelerindeki katsayı değerlerinden farklı olduğu için ayrı olarak incelenmiştir.



Şekil 3.9. Kaotik Chua Osilatörünün SIMULINK™ Modelinden Elde Edilen Kaotik Çeker ve DC Karakteristik Sonuçları, (a) NR1 Bloğu Sistemde, (b) NR2 Bloğu Sistemde, (c) NR3 Bloğu Sistemde. Nümerik Analizde Kullanılan Sistem Parametreleri, $\alpha=10$ ve $\beta=14.87$ dir.

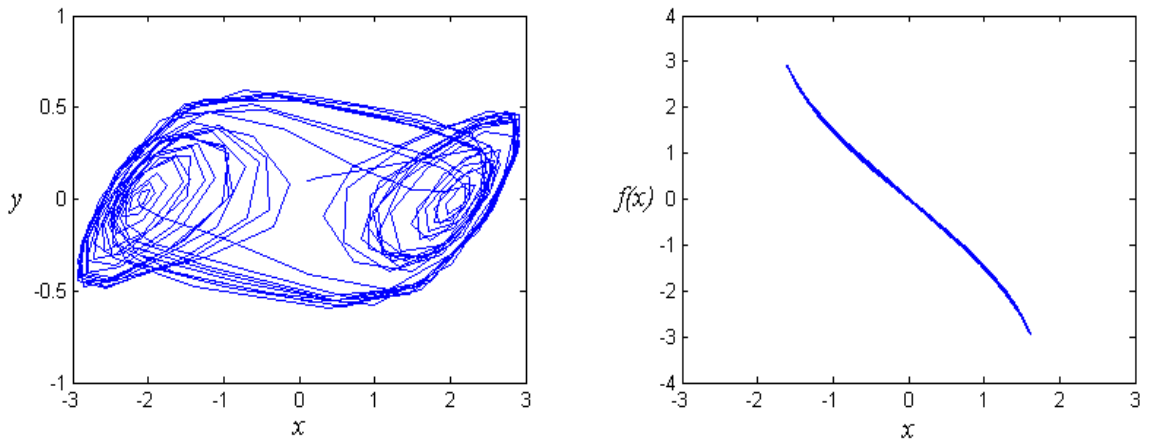


Şekil 3.10. Trigonometrik Doğrusal Olmayan Fonksiyon Kullanılarak Gerçekleştirilen Kaotik Chua Osilatörünün SIMULINK™ Modeli.

Trigonometrik doğrusal olmayan fonksiyonun analizinde FUNCTION bloğu kullanılmıştır. Bu blok sayesinde, SIMULINK™ kütüphanesinde var olmayan, standart matematiksel fonksiyonlardan farklı herhangi bir fonksiyon gerçekleştirilebilir. Bu sistemin SIMULINK™ modeli FUNCTION blok gösterimi ile birlikte Şekil 3.10' da verilmiştir. Bu modelleme kullanılarak elde edilen DC karakteristik ve çift çeker gösterimi Şekil 3.11' de yer almaktadır.

3.3.1.1.1. Kaotik Chua Osilatörünün Donanım Gerçekleştirimi

Kaotik bir sistemin matematiksel modelinden donanım gerçekleştirimi mümkündür. Bunu yapabilmek için sistem değişkenlerinin çalışma aralığının nümerik analiz sonuçlarına göre incelenmesi gerekmektedir. Eğer bu aralık, analog elektronik devrenin çalışma sınırlarından fazla ise sistem denklem takımı ölçeklendirme işlemine tabii

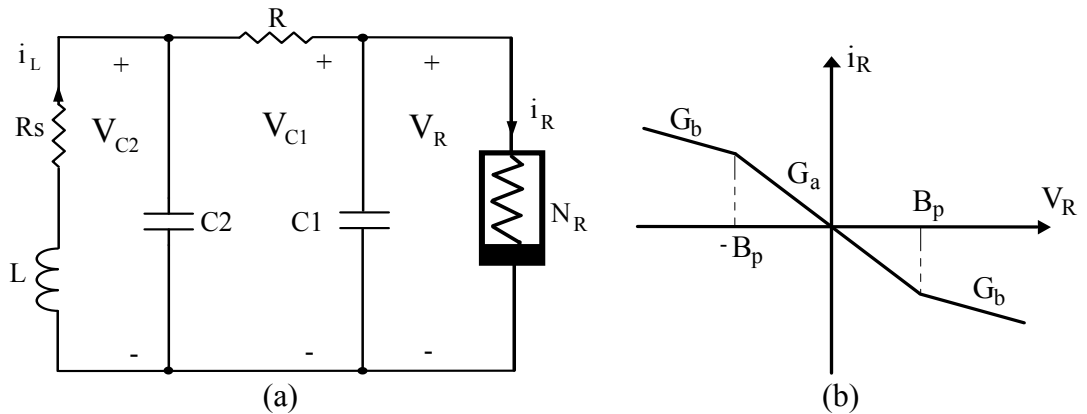


Şekil 3.11. Trigonometrik Doğrusal Olmayan Fonksiyon Kullanılarak Gerçekleştirilen Kaotik Chua Osilatörünün SIMULINK™ Analiz Sonuçlarındaki Kaotik Çeker Gösterimi ve DC Karakteristiği.

tutulur. Bu ölçeklendirme işleminin sonucunda da devre parametreleri belirlenir. Buna göre Chua osilatörünün Denklem (3.5) ile verilen denklem takımı dikkate alınarak analog devre gerçekleştirimi yapılabilir. Bu kısımda, örnek olarak literatürde yer alan ve üzerinde en çok çalışma yapılmış olan parçalı doğrusal fonksiyona sahip Chua osilatörü deneysel olarak gerçekleştirilmeye çalışılmıştır. Devre gerçekleştiriminde $\pm 9V$ ile kutuplanmış AD844 tipi akım geribeslemeli işlemsel kuvvetlendiriciler (Current Feedback Operational Amplifier, CFOA) kullanılmıştır. Chua osilatörünün Şekil 3.9’ da verilen nümerik analiz sonuçlarına göre durum değişkenlerinin maksimum değerleri $|x_{maksimum}|= 3$, $|y_{maksimum}|= 0.7$ ve $|z_{maksimum}|= 4.6$ ’ dır. Bu değerler işlemsel kuvvetlendiricinin doğrusal çalışma bölgesi sınırları içerisinde olduğundan ölçeklendirmeye gerek duyulmadan elektronik gerçekleştirimi yapılabilir. Şekil 3.12’ de devre şeması verilen Chua sistemi, üç adet enerji depolayıcı elemandan, bir adet doğrusal dirençten ve bir adet Chua diyodu olarak adlandırılan N_R doğrusal olmayan dirençten oluşmaktadır [70]. Chua devresi, aşağıdaki durum denklemleriyle tanımlanmaktadır ($V_{C1}=V_R$):

$$\begin{aligned} C_1 \frac{dV_R}{dt} &= \frac{1}{R} (V_{C2} - V_R) - f(V_R) \\ C_2 \frac{dV_{C2}}{dt} &= i_L - \frac{1}{R} (V_{C2} - V_R) \\ L \frac{di_L}{dt} &= -V_{C2} - i_L \cdot R_S \end{aligned} \quad (3.10)$$

Şekil 3.12(b)’ de grafiksel olarak gösterilen $f(V_R)$ parçalı doğrusal bir fonksiyon olup,



Şekil 3.12. (a) Chua Osilatörünün Devre Şeması, (b) Chua Osilatöründeki Doğrusal Olmayan Direncin Üç Parçalı Doğrusal Karakteristiği.

$$i_R = f(V_R) = G_b V_R + \frac{1}{2} \cdot (G_a - G_b) \times (|V_R + B_p| - |V_R - B_p|) \quad (3.11)$$

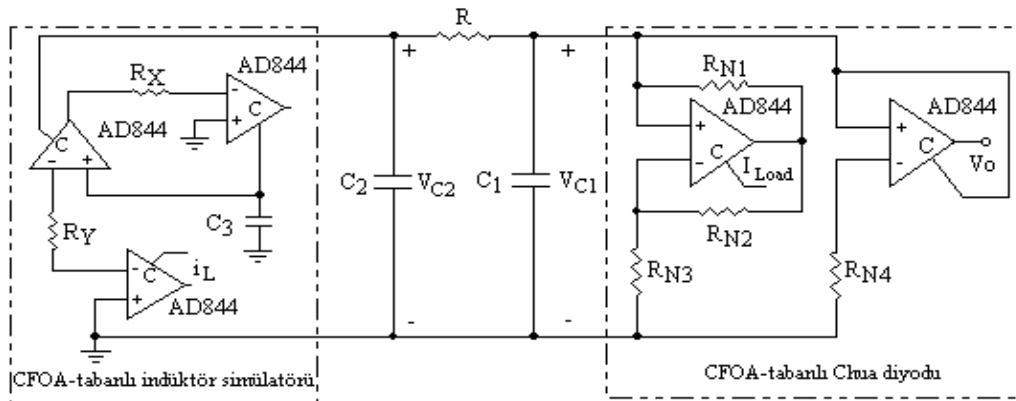
ile tanımlanır. Burada G_a ve G_b , sırasıyla iç ve dış bölgelerdeki eğimlerdir ve $\pm B_p$ ise, kırılma noktalarını sembolize etmektedir.

Bu denklemlerdeki değişkenlerin Denklem (3.5) ve (3.6)'daki karşılıkları aşağıda verilmiştir.

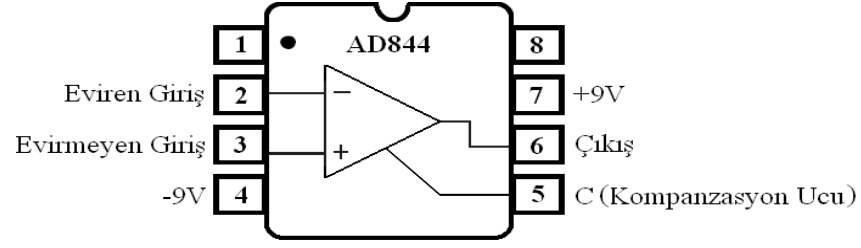
$$\begin{aligned} dV_R/dt = dV_{C1}/dt = dx/d\tau \quad dV_{C2}/dt = dy/d\tau \quad di_L/dt = dz/d\tau \\ 1/RC_1 = \alpha \quad 1/L = \beta \quad 1/RC_2 = 1 \quad R_s/L = 0 \\ f(V_R) = f(x) \quad G_a = a \quad G_b = b \quad B_p = c \end{aligned} \quad (3.12)$$

Chua osilatörünün elektronik gerçekleştirimi için devre şeması, Şekil 3.13' de verilen [87], CFOA-tabanlı doğrusal olmayan direnç [88] ve CFOA-tabanlı indüktör simülatörü [89] ile gerçekleştirilen yapı kullanılmıştır. Bu tasarımda doğrusal olmayan direnci gerçekleştirmek için iki adet AD844 tipi CFOA ve dört adet direnç kullanılmıştır. CFOA, yüksek hızlı analog işaret işleme uygulamaları için uygun analog yapı bloğudur. Bu blok, ikinci nesil akım taşıyıcı (Second Generation Current Conveyor, CCII) ile voltaj modlu kuvvetlendiriciyi içermektedir. CFOA, aşağıda verilen matris formatında tanımlanabilir [88].

$$\begin{bmatrix} V_x \\ I_y \\ I_c \\ V_o \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} I_x \\ V_y \\ V_c \\ I_o \end{bmatrix} \quad (3.13)$$



Şekil 3.13. Chua Osilatörünün Devre Şeması.

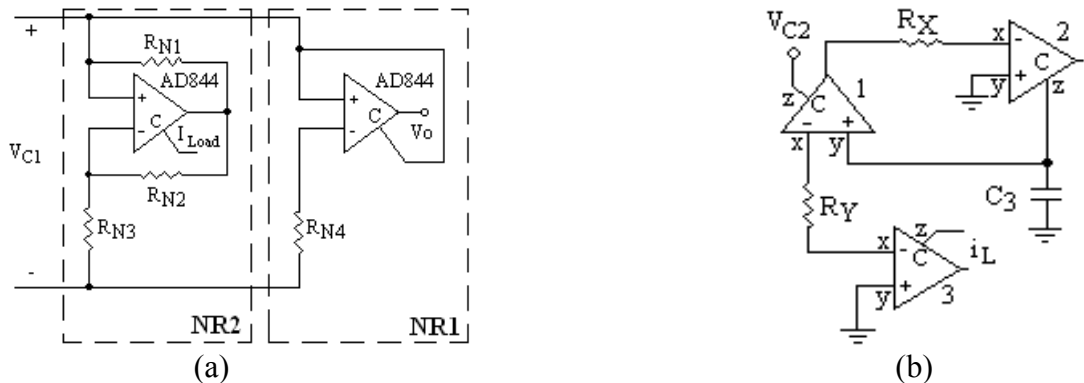


Şekil 3.14. CFOA'nın Fonksiyonel Blok Diyagramı.

Burada (x, y) eviren ve evirmeyen girişler olup, sırasıyla çok düşük ve çok yüksek giriş empedansı olarak ifade edilir. Akım ve voltaj çıkışları ise sırasıyla C ve O terminallerinden elde edilir. CFOA, hem voltaj modlu Op-amp (Voltage Op-Amp, VOA) gibi hem de CCII gibi çalışabilir ve CFOA'nın fonksiyonel blok diyagramı Şekil 3.14' de verilmiştir. Chua osilatörünün gerçekleştiriminde kullanılan CFOA-tabanlı doğrusal olmayan direnç, Şekil 3.15(a)' da görüldüğü gibi NR1 ve NR2 olarak adlandırılan iki adet voltaj kontrollü negatif empedans çeviricinin (Voltage Controlled Negative Impedance Converter, VNIC) paralel bağlanmasıyla oluşturulmuştur. CCII gibi bağlanan CFOA ile R_{N4} direnci, doğrusal NIC olarak çalışmaktadır. Diğer CFOA ve R_{N1} , R_{N2} ve R_{N3} dirençleri doğrusal olmayan VNIC olarak çalışırlar. Bu durumda CFOA, VOA gibi davranmaktadır. Bu doğrusal olmayan dirence ait Şekil 3.12(b)' de yer alan (V_R-i_R) grafiğindeki G_a ve G_b eğimleri ile B_p kırılma noktası denklemleri aşağıda verilmiştir.

$$G_a = -\frac{1}{R_{N4}} - \frac{R_{N2}}{R_{N1}R_{N3}} \quad G_b = -\frac{1}{R_{N4}} + \frac{1}{R_{N1}}$$

$$B_p = \frac{R_{N3}}{R_{N2} + R_{N3}} E_{sat} \quad (3.14)$$

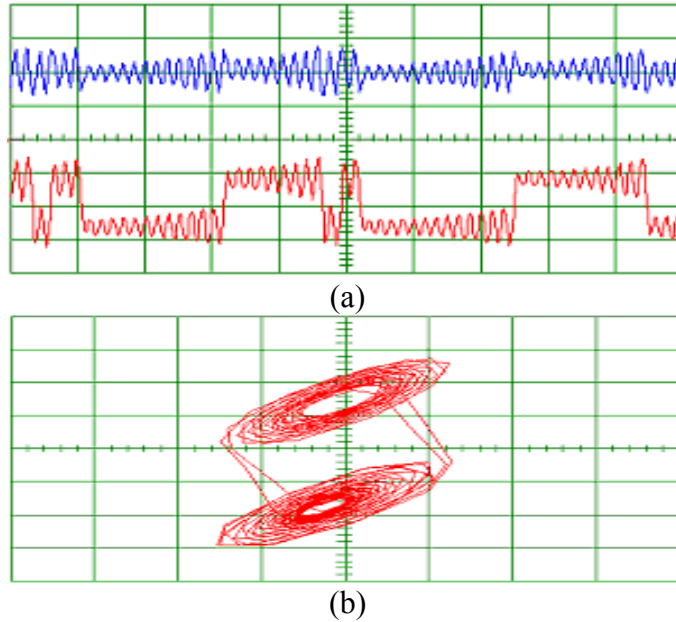


Şekil 3.15. Chua Osilatörünün Devre Yapısındaki, (a) Doğrusal Olmayan Direnç, (b) İndüktör Simülasyonu.

Şekil 3.15(b)' de görüldüğü gibi Chua osilatöründe indüktör simülatörünü gerçekleştirmek için üç adet AD844 tipi CFOA, iki adet direnç ve bir adet kapasitör kullanılmıştır [89]. Denklem (3.13)' deki CFOA matris formu dikkate alınarak yapılan indüktör simülatör analiz denklemleri ve eşdeğer indüktans denklemi aşağıda verilmiştir.

$$\begin{aligned}
 i_{x3} = i_{z3} = i_L & & V_{x1} = i_{x3}R_Y = i_L R_Y & & V_{z2} = V_{y1} = V_{x1} = \frac{i_{z2}}{sC_3} \\
 i_{z2} = sC_3 R_Y i_L & & i_{x2} = i_{z2} = \frac{V_{C2}}{R_X} & & sC_3 R_Y i_L = \frac{V_{C2}}{R_X} \\
 \frac{V_{C2}}{i_L} = sC_3 R_X R_Y & & L_{eq} = R_X R_Y C_3 & & (3.15)
 \end{aligned}$$

Bu deneysel çalışmada kullanılan devre parametre değerleri; $R_{N1}=R_{N2}=22 \text{ k}\Omega$, $R_{N3}=2 \text{ k}\Omega$ pot. (benzetim ve nümerik analiz çalışmalarında R_{N3} direnci $3.3 \text{ k}\Omega$ alınmıştır), $R_{N4}=2.2 \text{ k}\Omega$, $L=18 \text{ mH}$, $C_1=10 \text{ nF}$, $C_2=100 \text{ nF}$ ve $R=1.7 \text{ k}\Omega$ alınarak yapılmıştır [89]. CFOA-tabanlı indüktör simülatöründen $L=18 \text{ mH}$ ' yi elde etmek için, Denklem (3.15)' e göre $R_X=R_Y=1 \text{ k}\Omega$, $C_3=18 \text{ nF}$ olarak belirlenmiştir [72]. Deneysel çalışmadan elde edilen V_{C1} , V_{C2} kaotik devre dinamikleri, Şekil 3.16(a)' da, $(V_{C2}-V_{C1})$ düzlemindeki çift çeker yapısı ise Şekil 3.16(b)' de gösterilmiştir.



Şekil 3.16. CFOA-Tabanlı İndüktörsüz Chua Sisteminin Deneysel Gerçekleştiriminden Elde Edilen, (a) Kaotik Devre Dinamikleri: Üstteki İşaret V_{C2} (1 V/div), Altteki İşaret V_{C1} (2 V/div), Time/div:2 ms/div, (b) $(V_{C2}-V_{C1})$ Düzlemindeki Çift Çeker Yapısı, x-Ekseni:500 mV, y-Ekseni:1 V.

3.3.1.2. Kaotik Colpitts Osilatörü

Klasik Colpitts osilatörü normal şartlarda sinüzoidal işaret elde etmekte kullanılmaktadır. Ancak osilatörün devre parametrelerinde yapılan değişiklikler neticesinde kaotik davranış sergilediği kanıtlanmıştır. Colpitts osilatöründeki kaos olgusunu, ilk olarak Michael Peter Kennedy ortaya koymuş [90] olup, bu konu hakkında farklı çalışmalar literatürde yer almıştır [91-94]. Colpitts osilatörlerle ilgili yapılan ilk çalışmalarda birkaç 10 kHz' lerede çalışan BJT, parçalı doğrusal fonksiyon olarak modellenmiştir. BJT' nin çalışma frekansından dolayı kaotik Colpitts osilatörü de alçak çalışma frekansı ile sınırlanmıştır. Daha sonraki çalışmalarda bu frekans problemi devreye yapılan modifikasyonlarla ortadan kaldırılmıştır [93].

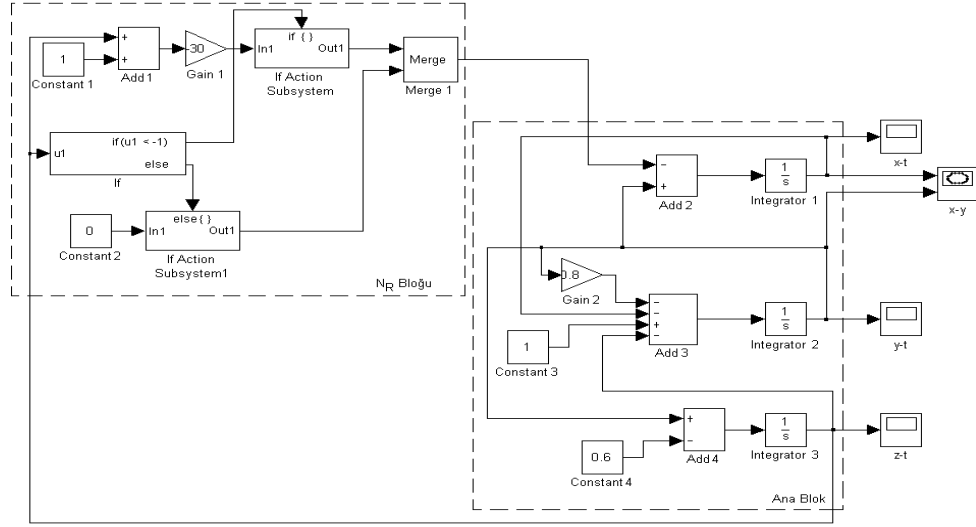
Colpitts osilatörün diferansiyel denklem takımı, aşağıda verilmiştir [94].

$$\begin{aligned}\frac{dx}{d\tau} &= y - f(z) \\ \frac{dy}{d\tau} &= d - x - b \cdot y - z \\ \frac{dz}{d\tau} &= y - k\end{aligned}\tag{3.16}$$

$$f(z) = \begin{cases} -a(z+1), & z < -1 \\ 0 & z > -1 \end{cases}$$

burada a , b , d ve k sistem parametreleri olup $a=30$, $b= 0.8$, $d=1$ ve $k= 0.6$ 'dır ve $f(z)$, doğrusal olmayan fonksiyondur.

Kaotik Colpitts osilatörünün SIMULINK™, de modellenerek nümerik analizi yapılabilir. SIMULINK™, bloklar çağırarak yapılan görsel bir analiz yöntemidir. Colpitts osilatörünün SIMILINK™, de modellenmesi sırasında Denklem (3.16)' da verilen denklemler kullanılmıştır. Denklem takımında yer alan x , y ve z durum değişkenleri, SIMULINK™ modelinde integral alma blokları, kazanç blokları, toplama/çıkarma blokları ve sabit değer blokları kullanılarak elde edilmiştir ve bu model Şekil 3.17' de verilmiştir.

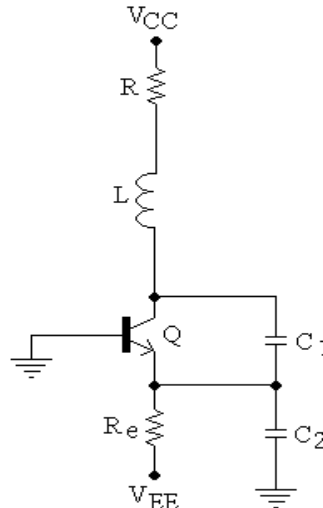


Şekil 3.17. Kaotik Colpitts Osilatörün SIMULINK™ Modeli.

Colpitts osilatörün denklem takımında yer alan $f(z)$ doğrusal olmayan fonksiyonu gerçekleştirmek için, toplama, sabit değer ve kazanç blokları kullanılmıştır. Bu blokların yanı sıra tek girişli *if* bloğu, $z < -1$ kriterine göre işlem yapmaktadır.

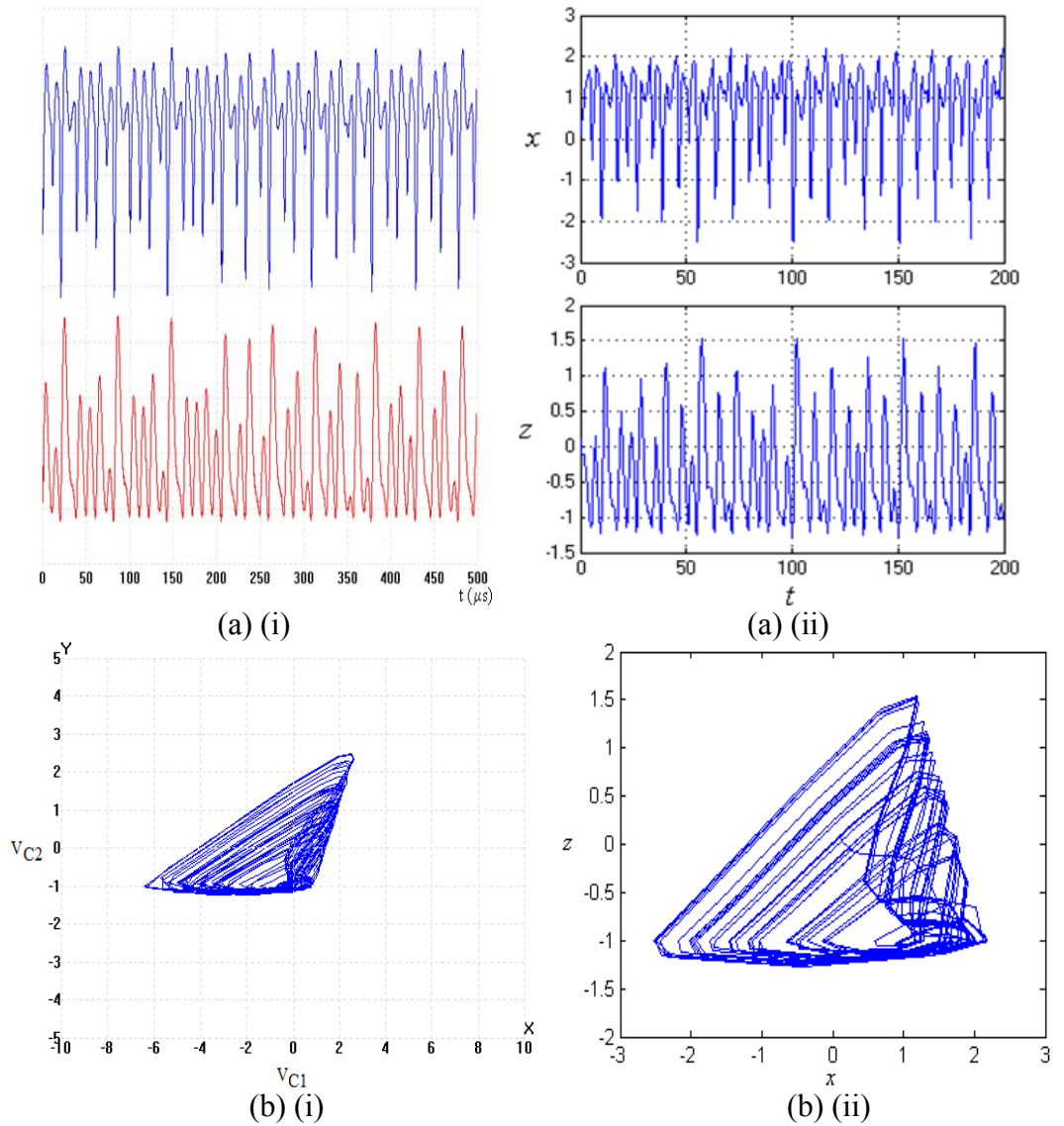
3.3.1.2.1. Kaotik Colpitts Osilatörün Donanım Gerçekleştirimi

Kaotik Colpitts osilatörü ayrıık elemanlarla deneysel ortamda gerçekleştirilebilir. Şekil 3.18’ de devre gerçekleştirim şeması verilen Colpitts osilatörde, doğrusal olmayan fonksiyonu gerçeklemek için tek BJT kullanılmıştır [91]. Devre gerçekleştiriminde kullanılan elemanların değerleri: $C_1=C_2=56$ nF, $R=30$ Ω , $V_{CC}=5$ V, $V_{EE}=-5$ V, $R_e=400$ Ω ve $L=98$ μ H’ dir. Devrede transistör olarak da 2N2222 tipi BJT kullanılmıştır.



Şekil 3.18. Kaotik Colpitts Osilatörün Devre Şeması.

Devre, ± 5 V ile beslendiğinden dolayı devreden elde edilecek dinamikler en fazla ± 5 V civarında olması gerekmektedir. Nümerik analiz sonuçlarına bakıldığında x , y ve z durum değişkenlerinin maksimum değerleri ± 5 V' dan küçük olduğu için Colpitts osilatörünün deneysel gerçekleştirimi, ölçeklendirme işlemine gerek kalmadan yapılmıştır. Yapılan deneysel çalışmadan elde edilen sonuçlar, Şekil 3.19' da nümerik analiz sonuçları ile beraber verilmiştir. Bu deneysel sonuçların nümerik analiz sonuçları ile örtüştüğü görülmüştür.



Şekil 3.19. (i) Colpitts Osilatörün Donanım Gerçekleştiriminden Elde Edilen, (a) (i) Kaotik Devre Dinamikleri: Üstteki İşaret V_{C1} (2 V/div), Altaki İşaret V_{C2} (1 V/div), (b) (i) ($V_{C1}-V_{C2}$) Düzlemindeki Çeker Gösterimi, (ii) Kaotik Colpitts Osilatörünün SIMULINK™ Modelinden Elde Edilen Nümerik Analiz Sonuçları, (a) (ii) Kaotik Dinamikleri, (b) (ii) ($x-z$) Düzlemindeki Çeker Gösterimi.

3.3.1.3. Duffing Osilatör

Duffing osilatör, ikinci dereceden diferansiyel denklem takımına sahip otonom olmayan kaotik sistemlerden birisidir [68, 95-98]. Bu sistem, başlangıç koşullarına ve harici AC işaretin genliğine ve frekansına göre çeşitli kaotik davranışlar sergilemektedir. Bu osilatöre ait durum denklemleri aşağıda verilmiştir [68]:

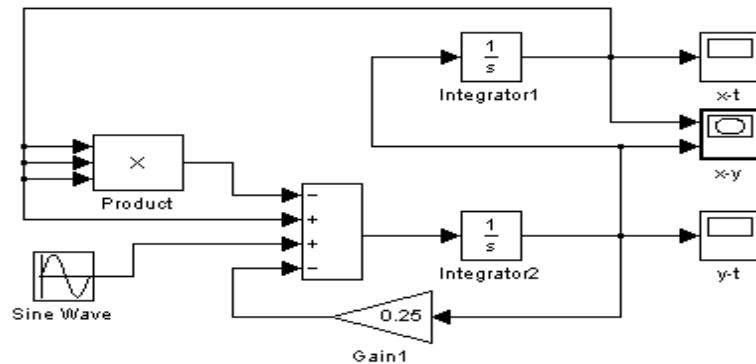
$$\begin{aligned} \frac{dx}{d\tau} &= y \\ \frac{dy}{d\tau} &= a \sin t - by - dx - kx^3 \end{aligned} \quad (3.17)$$

Denklemden yer alan katsayıların değerleri $a=0.3$, $b=0.25$, $d=-1$ ve $k=1$ şeklindedir.

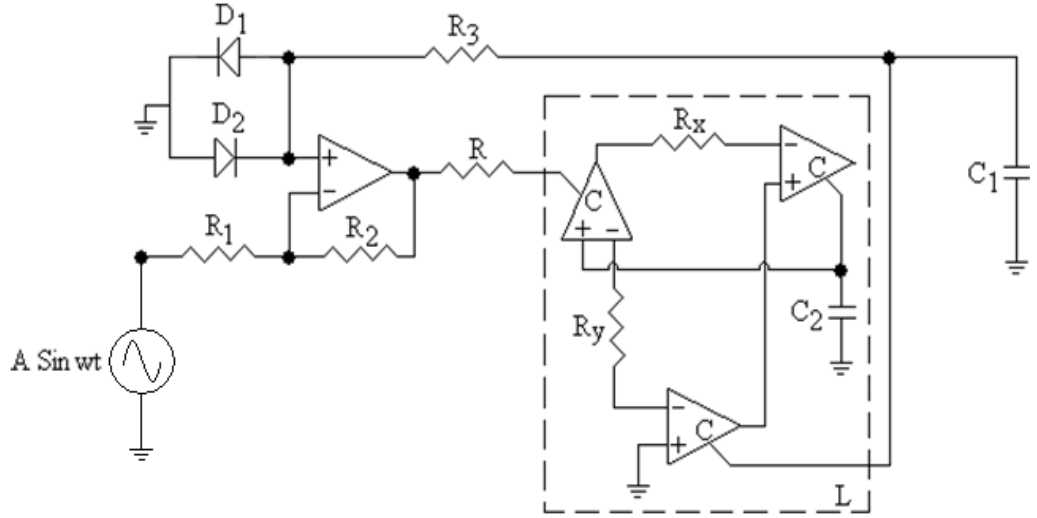
SIMULINK™ ortamında modellenen Duffing osilatör, Şekil 3.20’de verilmiştir. Bu modelin tasarımı sırasında toplama/çıkarma, çarpma, kazanç ve integral alma blokları kullanılmıştır. Denklem (3.17)’de yer alan ($\sin t$) ifadesini gerçeklemek için de sinüs dalga üreten farklı bir blok kullanılmıştır.

3.3.1.3.1. Duffing Osilatörün Donanım Gerçekleştirimi

Duffing osilatörün yapılan nümerik analiz sonuçlarına göre x ve y durum değişkenlerinin maksimum değerleri $\pm 2V$ ’dan küçüktür. Devrede kullanılan besleme voltajı $\pm 9V$ olacağı için osilatör durum değişkenleri ölçeklendirme işlemine gerek duymadan ayırık elemanlarla gerçekleştirilebilir. Otonom olmayan özelliğe sahip Duffing osilatörün devre gerçekleştirimi, Şekil 3.21’de verilmiştir. Literatürde yer alan Duffing osilatör devresinde indüktör elemanı bulunmaktadır [97]. Fiziksel büyüklük problemi, tam eleman değerinin elde edilememesi ve parazitik etkilere sahip olmasından



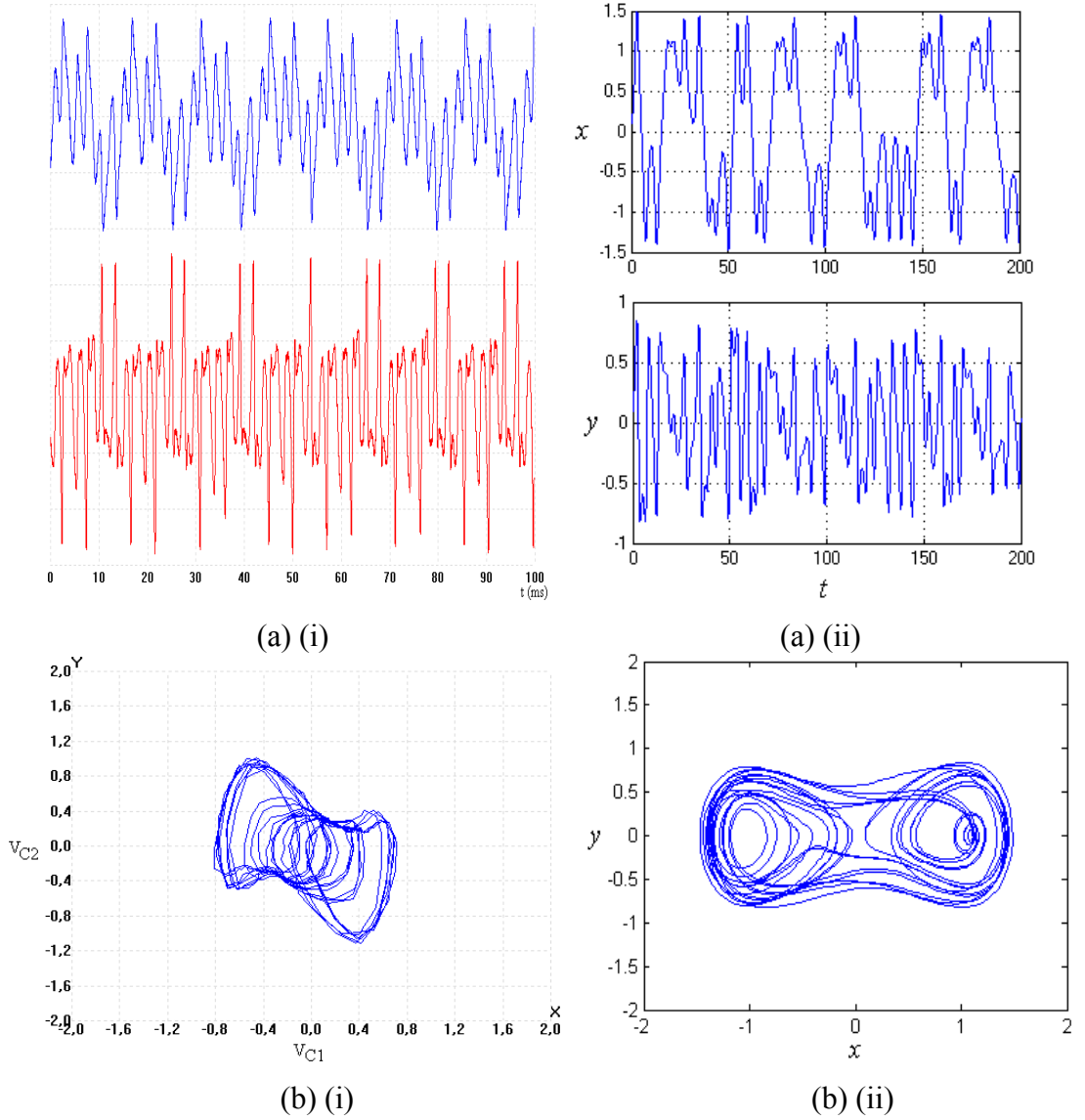
Şekil 3.20. Duffing Osilatörün SIMULINK™ Modeli.



Şekil 3.21. Duffing Osilatörün İndüktörsüz Devre Gerçekleştirimi.

dolayı devrede indüktör yerine aktif elemanlardan ve RC yapılardan oluşan indüktör simülatorü kullanılmıştır. Devredeki iki adet D_1 ve D_2 diyotu ve R_3 direncini içeren pozitif geri besleme döngüsü doğrusal olmayan davranışın kaynağıdır. Devredeki işlemsel kuvvetlendirici, hem harici olarak verilen sinüzoidal işareti kuvvetlendirme rolünü, hem de pozitif doğrusal olmayan geri besleme için kuvvetlendirme katı rolünü oynamaktadır [97].

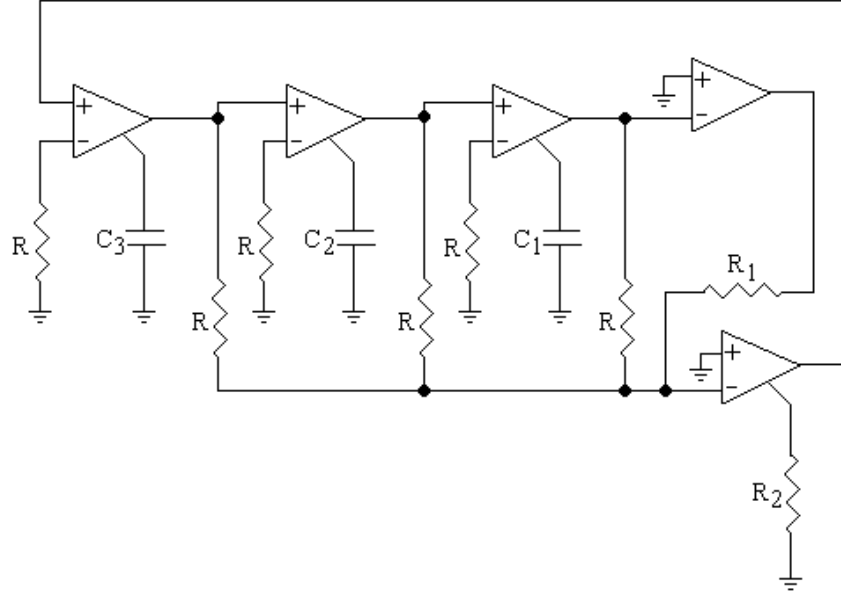
Bu osilatör devresinde kullanılan elemanlar ve değerleri: $R=20 \Omega$, $R_1=R_2=R_3=10 \text{ K}\Omega$, $C_1=470 \text{ nF}$ dır. Devrede kullanılan CFOA tabanlı indüktör simülatorünün parametreleri, Denklem (3.15)' e göre $L=19 \text{ mH}$ ' yi elde etmek için $R_x=R_y=1 \text{ k}\Omega$, $C_2=19 \text{ nF}$ olarak belirlenmiştir. Devrede indüktör simülatorünü gerçekleştirmek için AD844 işlemsel kuvvetlendiricileri kullanılmıştır. Devrede doğrusalsızlık için de, LM741 Op-Amp' ı ve 1N4001 tipi D_1 ve D_2 diyotları kullanılmıştır. Devrenin kaosa girmesinde önemli rol oynayan harici AC işaretin genliği $A=200 \text{ mV}$ ve frekansı $f=1.5 \text{ KHz}$ olarak belirlenmiştir. Genlik değeri azaldıkça devrenin davranışı kaotik işareten periyodik işarete dönüşmektedir. Devrenin deneysel çalışmalar sonucunda elde edilen kaotik davranışları ve çeker gösterimi Şekil 3.22' de verilmiştir. Bu sonuçların nümerik analiz sonuçlarına benzediği görülmüştür.



Şekil 3.22. (i) Duffing Osilatörün Donanım Gerçekleştiriminden Elde Edilen, (a) (i) Kaotik Devre Dinamikleri: Üstteki İşaret V_{C1} (0.4 V/div), Altaki İşaret V_{C2} (0.4 V/div), (b) (i) $(V_{C1}-V_{C2})$ Düzlemindeki Çeker Gösterimi, (ii) Duffing Osilatörün SIMULINK™ Modelinden Elde Edilen Nümerik Analiz Sonuçları, (a) (ii) Kaotik Dinamikleri, (b) (ii) $(x-y)$ Düzlemindeki Çeker Gösterimi.

3.3.1.4. Kennedy Osilatör

Michael Peter Kennedy ve arkadaşları tarafından literatüre sunulmuş olan bu osilatör, en temel kaotik davranışları sergilemektedir. Basit bir yapıya sahip olduğu için elektronik devre olarak da tasarlanması mümkün olan bu osilatöre ait diferansiyel denklem takımı aşağıda yer almaktadır [99, 100].



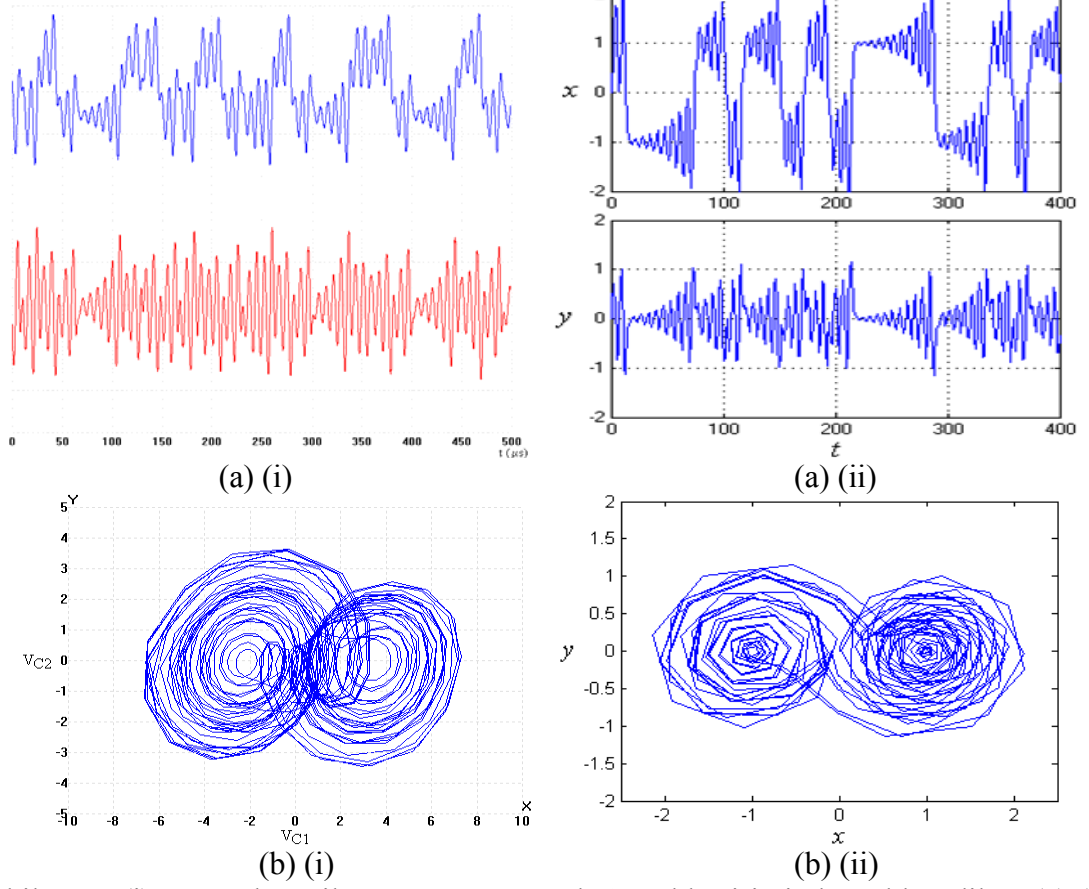
Şekil 3.24. Kennedy Osilatörün Devre Şeması.

toplayıcı katı da kullanılarak devre tamamlanmıştır. Eviren toplayıcının kazancı, 0.6 ile aynıdır ve bu kazanç, R_2 direncinin değişimi ile değişebilir. (*sgn*) doğrusalsızlığı devrede karşılaştırıcı gibi çalışan işlemsel kuvvetlendirici ile sağlanmaktadır [100]. Bu deneysel kurulumda kullanılan elemanların değerleri: $R=1 \text{ K}\Omega$, $C_1=C_2=C_3=1 \text{ nF}$, $R_1=5 \text{ K}\Omega$ pot. ve $R_2=1 \text{ K}\Omega$ pot.dur. R_1 ve R_2 dirençlerinin değişimine göre devre kaotik davranış sergilemektedir. $R_1=2.46 \text{ K}\Omega$ ve $R_2=0.59 \text{ K}\Omega$ değerlerinde devrenin kaosa girdiği görülmüştür. Şekil 3.25' de devrenin deneysel sonuçlardan elde edilen kaotik dinamikleri ve çeker yapısı gösterilmiştir. Şekilden görüldüğü gibi deneysel sonuçlar, SIMULINK™ nümerik analiz yöntem sonuçlarıyla örtüşmektedir.

3.3.1.5. Lorenz Sistemi

Lorenz sistemi, 1963 yılında Edward N. Lorenz tarafından literatüre sunulmuştur [53]. Bir meteorolog olan Lorenz hava olaylarını modellemek için 3. Dereceden doğrusal olmayan bir denklem takımını ortaya koymuştur. Kendi ismiyle anılan bu kaotik denklem takımı aslında Lorenz farkında olmasa da kaos ve kaotik sistemlerin başlangıcını oluşturmuştur.

Lorenz sistemine ait matematiksel ifadeler aşağıda yer almaktadır [101].

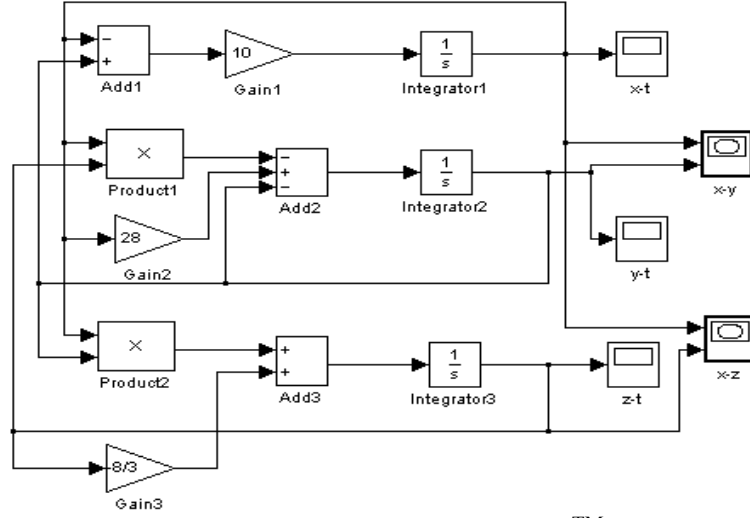


Şekil 3.25. (i) Kennedy Osilatörün Donanımsal Gerçekleştiriminden Elde Edilen, (a) (i) Kaotik Devre Dinamikleri: Üstteki İşaret V_{C1} (4 V/div), Alttaki İşaret V_{C2} (2 V/div), (b) (ii) ($V_{C1}-V_{C2}$) Düzlemindeki Çeker Gösterimi, (ii) Kennedy Osilatörün SIMULINK™ ile Modellemesinden Elde Edilen Nümerik Analiz Sonuçları, (a) (ii) Kaotik Dinamikleri, (b) (ii) ($x-y$) Düzlemindeki Çeker Gösterimi.

$$\begin{aligned}
 \frac{dx}{dt} &= -ax + ay \\
 \frac{dy}{dt} &= rx - y - xz \\
 \frac{dz}{dt} &= -bz + xy
 \end{aligned}
 \tag{3.19}$$

Sistemin kaotik davranış sergilemesi için sistem parametreleri $a=10$, $b=8/3$ ve $r=28$ olarak seçilmiştir.

Lorenz sistemi, grafiksel programlama özelliğine sahip SIMULINK™, de modellemesi yapıлып, nümerik analiz özelliği sayesinde de analiz sonuçlarının alınabileceği bir sistemdir. Bu sistemin SIMULINK™ modellemesi Şekil 3.26' da yer almaktadır. Bu modelin tasarımında x , y ve z durum değişkenleri integral alıcı blokların çıkışlarından



Şekil 3.26. Lorenz Sistemin SIMULINK™ Modeli.

elde edilmektedir. Denklem takımında yer alan xy ve xz çarpımları, çarpma blokları ile gerçekleştirilmiştir. Modelde kullanılan toplama/çıkarma ve kazanç blokları ise denklem takımında yer alan diğer işlemlerin gerçekleştirilmesi için kullanılmıştır.

3.3.1.5.1. Lorenz Sisteminin Donanım Gerçekleştirimi

Lorenz sistemi, kaostan ilgili çalışmaların temelini oluşturan ve gelişimini sağlayan temel osilatörlerden birisidir. Bu önemli sistem, atmosferik kararsızlığın basitleştirilmiş matematiksel modeli olarak geliştirilmiş olup basit devre topolojisinden türetilmemiştir [102]. Bu sisteme ait diferansiyel denklem takımı, Denklem (3.19) ile verilmiştir.

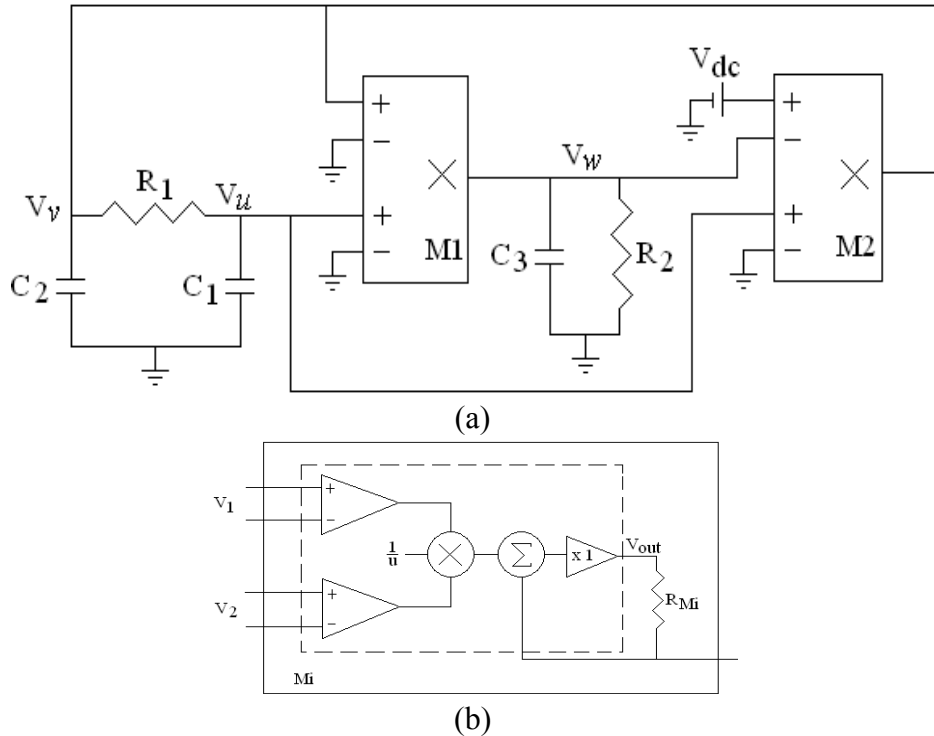
Sistemin önceki kısımda incelenen nümerik analiz sonuçlarına göre durum değişkenleri maksimum $|x_{maksimum}| = 20$, $|y_{maksimum}| = 27$ ve $|z_{maksimum}| = 48$ değerlerindedir. Elektronik devre gerçekleştiriminde ise devre, $\pm 15V$ ile beslenecektir. Bu durumda sistemin teknik olarak pratik gerçekleştirimi mümkün olamamaktadır. Bu problemi çözmek için sistem denklemlerine ölçeklendirme işlemi yapılacaktır. Yeni sistemin durum değişkenleri aşağıda verilmiştir.

$$u = \frac{x}{k}, \quad v = \frac{y}{k}, \quad w = \frac{z}{n} \quad (3.20)$$

Burada k ve n , ölçekleme faktörü olarak seçilmiştir. Ölçeklendirilen devrenin denklem takımı aşağıda verilmiştir.

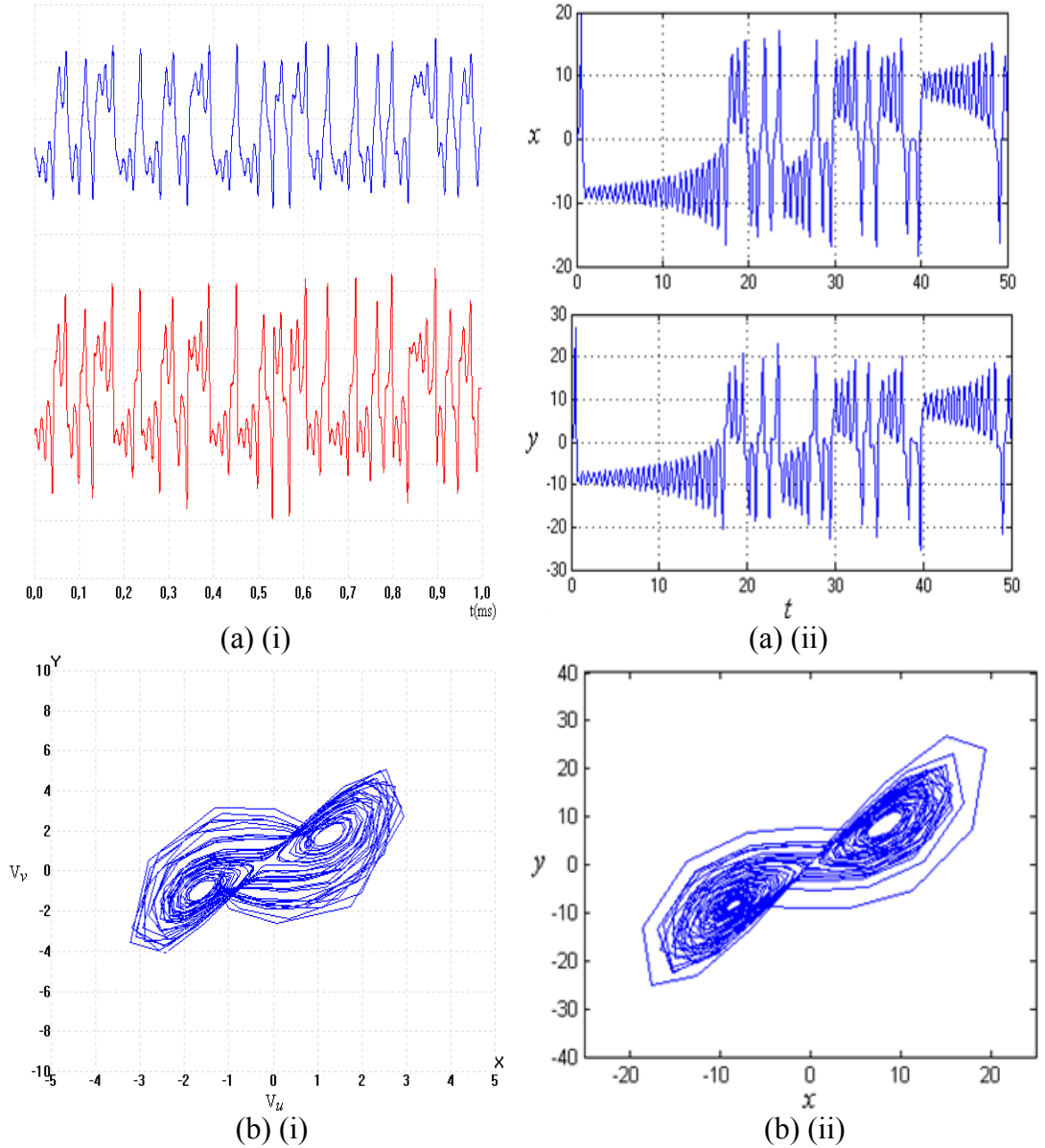
$$\begin{aligned}
\frac{du}{dt} &= -au + av \\
\frac{dv}{dt} &= ru - v - nuw \\
\frac{dw}{dt} &= -bw + \frac{k^2}{n}uv
\end{aligned}
\tag{3.21}$$

Burada k ve n sırasıyla 5.5 ve 11 seçilerek ölçeklendirme işlemi tamamlanmış olur. Bu ölçeklendirme işlemine göre gerçekleştirilen Lorenz sisteminin devre şeması Şekil 3.27' de verilmiştir [103]. Bu devre gerçekleştiriminde 2 adet çarpıcı elemanı, 3 adet kapasitör, 2 adet direnç ve dc voltaj kaynağı kullanılmıştır. Çarpıcı elemanı olarak AD633 tipi analog çarpıcı kullanılmıştır. Analog çarpıcı devreleri iki diferansiyel girişin çarpımı ile orantılı bir çıkış voltajı üretirler. Bununla birlikte üçüncü giriş voltajı çıkıştan önce çarpımla toplanır. Bir akım çıkışı elde etmek için kullanılan voltaj çıkışı ile toplanan giriş arasında dirençle sağlanan bağlantı Şekil 3.27(b)' de gösterilmiştir. V_1 ve V_2 voltajları, diferansiyel girişlerden sağlanmaktadır [103].



Şekil 3.27. (a) Ölçeklendirilmiş Lorenz Sisteminin Devre Şeması. Analog Çarpıcılar M_1 Ve M_2 , Akım Çıkış Elemanı Olarak Kullanılmışlardır, (b) Akım Çıkış Elemanı Olarak Kullanılan M_i Analog Çarpıcının İç Yapısı.

Ölçeklendirilmiş Lorenz sisteminin gerçekleştiriminde kullanılan elemanlar ve değerleri: $R_1=10\text{ K}\Omega$ pot., $R_2=5.1\text{ K}\Omega$, $R_{M1}=560\ \Omega$, $R_{M2}=56\ \Omega$, $C_1=200\text{ pF}$, $C_2=2\text{ nF}$, $C_3=820\text{ pF}$ ’dır. Devrede yer alan V_{dc} voltajı için, devrenin kaotik dinamikler üretmesi için 0-6 V arasında ayarlanabilir bir kaynak kullanılmıştır. Ölçeklendirilmiş Lorenz sisteminin elektronik gerçekleştiriminden elde edilen deneysel sonuçlar, Şekil 3.28’de yer almaktadır.



Şekil 3.28. (i) Lorenz Sisteminin Donanım Gerçekleştiriminden Elde Edilen, (a) (i) Kaotik Devre Dinamikleri: Üstteki İşaret V_u (2 V/div), Alttağı İşaret V_v (2 V/div), (b) (i) (V_u - V_v) Düzlemindeki Çeker Gösterimi, (ii) Lorenz Sistemin SIMULINK™ Modelinden Elde Edilen Nümerik Analiz Sonuçları, (a) (ii) Kaotik Dinamikleri, (b) (ii) (x - y) Düzlemindeki Çeker Gösterimi.

3.3.1.6. MLC (Murali-Lakshmanan-Chua) Osilatörü

Kaotik MLC osilatörü, kaotik yapılar arasında en dikkat çekici modellerden birisidir. Harici bir sinüzoidal kaynağa sahip olan bu yapı, literatürde yer alan otonom olmayan kaotik sistemlerden birisidir. Yapısı oldukça basit olan bu devre, 1994 yılında Murali, Lakshmanan ve Chua tarafından geliştirilmiştir [104]. Bu otonom olmayan MLC osilatörün boyutsuz formdaki diferansiyel denklemleri aşağıda yer almaktadır [105]:

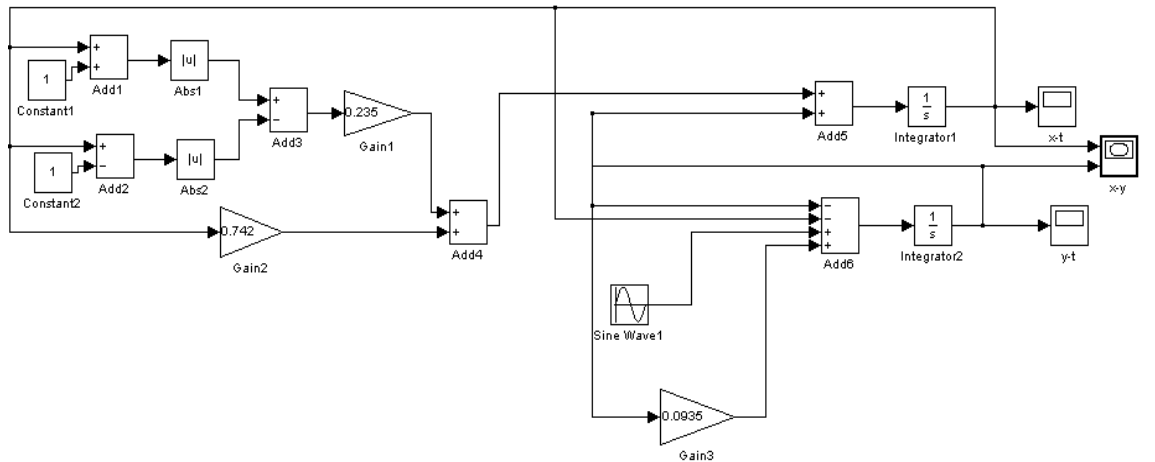
$$\begin{aligned} \frac{dx}{d\tau} &= y - f(x) \\ \frac{dy}{d\tau} &= -d.y - d.k.y - d.x + F\sin(wt) \end{aligned} \quad (3.22)$$

burada $d=1$, $k=0.0935$, $w=0.75$ ve $F=0.15$ olarak alınmıştır. $f(x)$, doğrusal olmayan fonksiyon olup aşağıdaki gibi tanımlanmaktadır:

$$f(x) = b.x + 0.5(a - b)(|x + c| - |x - c|) \quad (3.23)$$

burada $a=-1.02$, $b=-0.55$ ve $c=1$ 'dir.

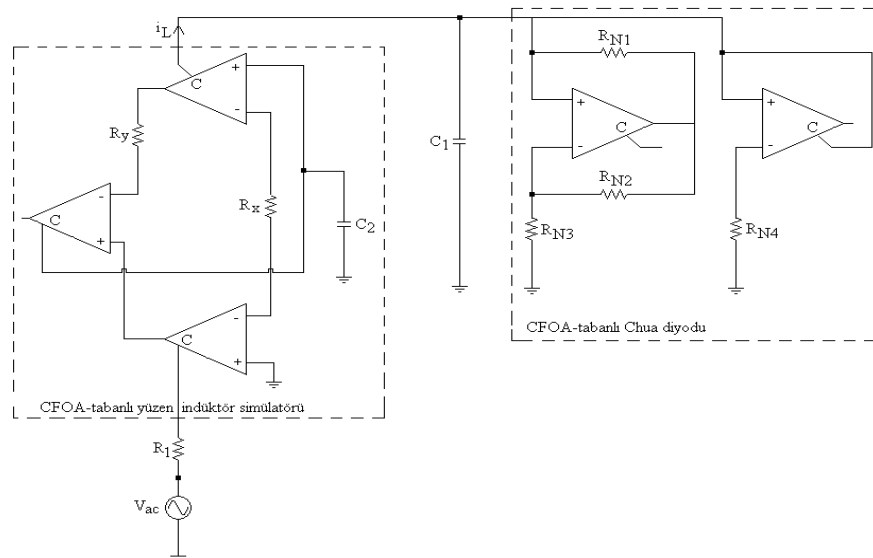
MLC osilatörünün Denklem (3.22) ve (3.23)' de yer alan matematiksel ifadeler kullanılarak oluşturulan SIMULINK™ modeli, Şekil 3.29' da verilmiştir. Bu modelde doğrusal olmayan fonksiyonu gerçekleştirmek için toplama/çıkarma, kazanç, sabit değer bloklarının yanı sıra mutlak değer alma blokları kullanılmıştır. Osilatör durum değişkenlerini elde etmek için ise integral alma, toplama/çıkarma, kazanç blokları ve $\sin(wt)$ ifadesini gerçekleştirmek için sinüs dalga bloğu kullanılmıştır.



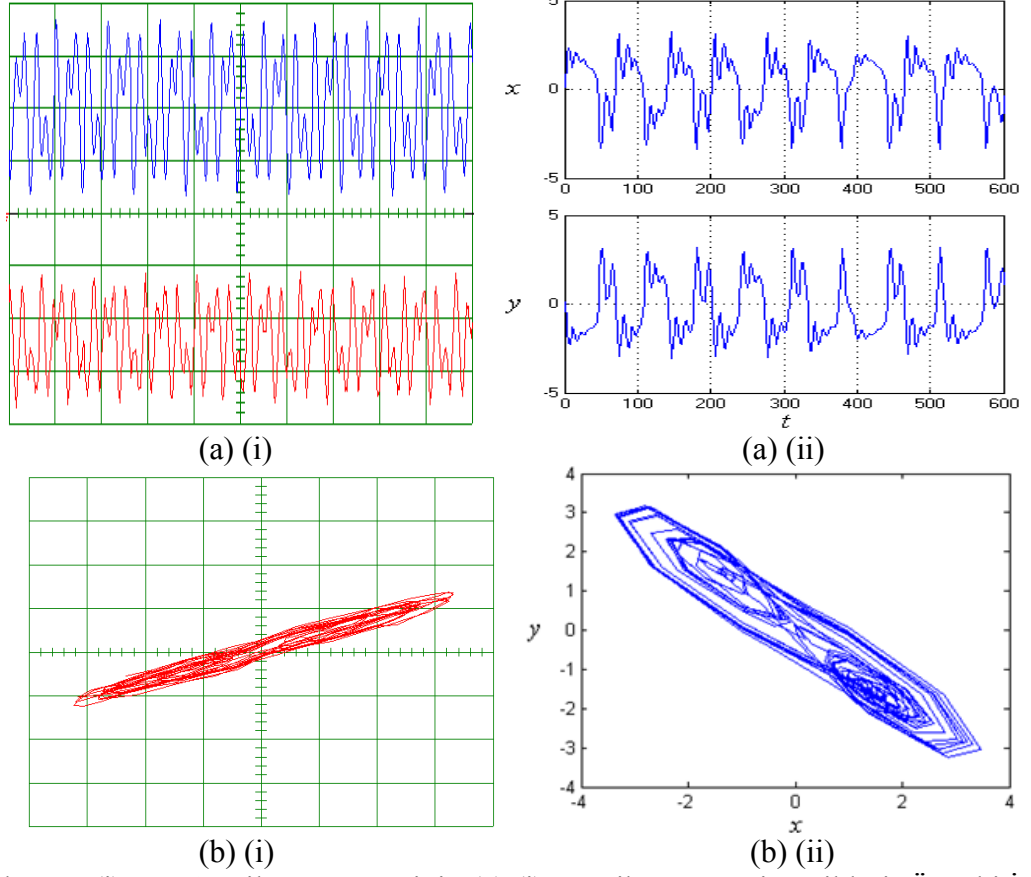
Şekil 3.29. MLC Osilatörün SIMULINK™ Modeli.

3.3.1.6.1. MLC Osilatörün Donanım Gerçekleştirimi

MLC osilatörü, otonom olmayan kaotik osilatörler arasında basit devre yapısına ve analiz kolaylığına sahiptir. Devre gerçekleştiriminde devre, $\pm 9V$ ile beslenecektir. MLC osilatörün SIMULINK™ modellemesinden elde edilen sonuçlara göre durum değişkenlerinin maksimum değerleri $|x_{maksimum}|= 3.5$ ve $|y_{maksimum}|= 3.3$ ' dür. Bu değerler devre besleme voltaj değerinin altında olduğu için ölçeklendirmeye gerek duyulmadan elektronik devre gerçekleştirimi yapılabilir. Bu osilatör gerçekleştiriminde Chua devresinde yer alan Chua diyodu kullanılmıştır. Bu yapının en önemli özelliği devrede harici bir sinüzoidal kaynağın olması ve kaynak genliğinin devrenin kaosa girmesindeki değişim parametresi olmasıdır. MLC osilatörün elektronik devre gerçekleştirimi Şekil 3.30' da verilmiştir [106, 107]. Bu gerçekleştirimde Chua diyodu, $\pm 9 V$ ile kutuplanmış iki adet AD844 tipi CFOA ve dört adet direnç kullanılarak gerçekleştirilmiştir. Bu dirençlerin değerleri: $R_{N1}=R_{N2}=22 k\Omega$, $R_{N3}=2 k\Omega$ pot. ve $R_{N4}=2.2 k\Omega$ ' dur. Devrede yer alan yüzen (floating) indüktör simülatörü, yüzen indüktör yerine kullanılmış olup, üç adet AD844 elemanı ile iki adet direnç ve bir adet kapasitörle gerçekleştirilmiştir [108]. Yüzen indüktör $L=18 mH$ ' yi elde etmek için indüktör simülatörünün parametreleri, Denklem (3.15)' e göre $R_x=R_y=1 k\Omega$, $C_2= 18 nF$ olarak belirlenmiştir. İndüktör simülatörü ve Chua diyodu dışında devrede yer alan diğer parametreler; $C_1 =10nF$, $R_1=1340\Omega$ olarak belirlenmiştir. Ayrıca devrede frekansı $f=8890 Hz$ ve genliği $A=0.15 V$ olan V_{ac} sinüzoidal kaynak da bulunmaktadır. Harici işaretin genliği (A), kaotik



Şekil 3.30. MLC Osilatörü.



Şekil 3.31. (i) MLC Osilatör Devresinin (a) (i) Kaotik Devre Dinamikleri: Üstteki İşaret V_{CI} (2 V/div), Altteki İşaret i_L (100 Ω ' luk Direnç İle) (500 mV/div), Time/div: 500 μ s/div, (b) (i) ($V_{CI}-i_L$) Düzlemindeki Çift Çeker Yapısı, X-Eksen:1 V, Y-Eksen:500 mV, (ii) MLC Osilatörün SIMULINK™ Modelinden Elde Edilen Nümerik Analiz Sonuçları, (a) (ii) Kaotik Dinamikleri, (b) (ii) ($x-y$) Düzlemindeki Çeker Gösterimi.

değişim parametresi olarak kullanılmaktadır. Bu genlik değerinin sıfırdan yukarıya doğru artırılmasıyla sistem, değişik dallanma ve kaos olayı sergilemektedir. Deneysel sonuçlardan elde edilen kaotik devre dinamikleri, Şekil 3.31(a) (i)' de gösterilirken, ($V_{CI}-i_L$) düzlemindeki çeker yapısı Şekil 3.31(b) (i)' de verilmektedir.

3.3.1.7. Rössler Sistemi

Otto Rössler, Lorenz sistemine çok benzeyen ve analizi daha kolay yapılan bir sistem tasarlamak istemiş ve bu amaçla yaptığı çalışmalar neticesinde 1970'li yıllarda Rössler sistemini ortaya koymuştur [109]. Bu sistem, Lorenz sistemine çok benzemekle birlikte daha basit yapıda olup, 3 adet diferansiyel denklemlerle ifade edilmektedir.

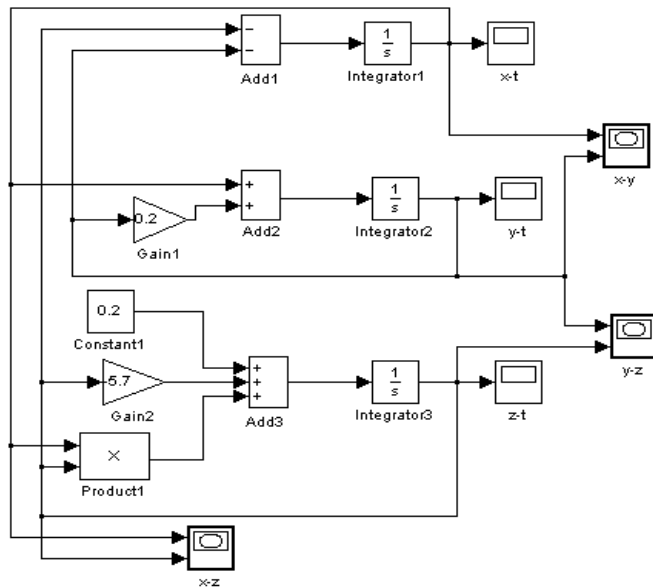
$$\begin{aligned}\frac{dx}{d\tau} &= -y - z \\ \frac{dy}{d\tau} &= x + k.y \\ \frac{dz}{d\tau} &= d + x.z - a.z\end{aligned}\quad (3.24)$$

Burada a , d ve k , sistem parametreleri olup, $a=5.7$, $d=k=0.2$ ' dir.

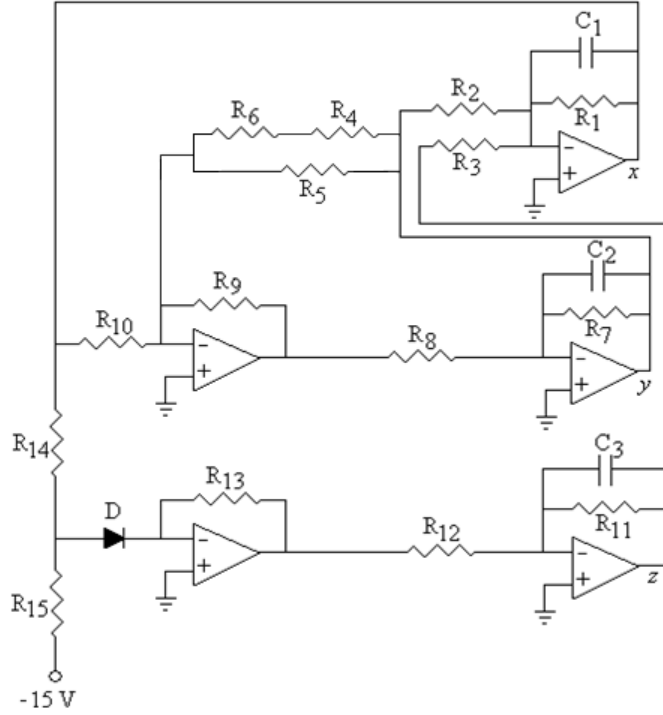
Şekil 3.32' de yer alan SIMULINK™ modeli ile Rössler sisteminin grafiksel programlama özelliğiyle analizi yapılabilir. Bu modelde sistem durum değişkenleri x , y ve z , integral alıcı blokların çıkışlarından elde edilmiştir. Sistemin denklem takımında yer alan işlemleri gerçekleştirmek için toplama, sabit değer, kazanç ve çarpma blokları kullanılmıştır.

3.3.1.7.1. Rössler Sisteminin Donanım Gerçekleştirimi

Literatürde Rössler sisteminin farklı elektronik devre gerçekleştirmeleri mevcuttur [110-113]. Bu devre gerçekleştirmeleri arasında Thomas L. Carroll tarafından ortaya konulan devre yapısında, Rössler sistemi bazı modifikasyonlar yapılarak gerçekleştirilmiştir [110] ve bu gerçekleştirilen sistemin devre şeması Şekil 3.33' de verilmiştir. Bu devre yapısında çarpım elemanı yoktur. Onun yerine doğrusalsızlık, bir adet diyot, üç adet direnç ve bir adet işlemsel kuvvetlendirici ile yapılan parçalı doğrusal fonksiyonla sağlanmıştır.



Şekil 3.32. Rössler Sisteminin SIMULINK™ Modeli.



Şekil 3.33. Modifiye Edilen Rössler Sistemi.

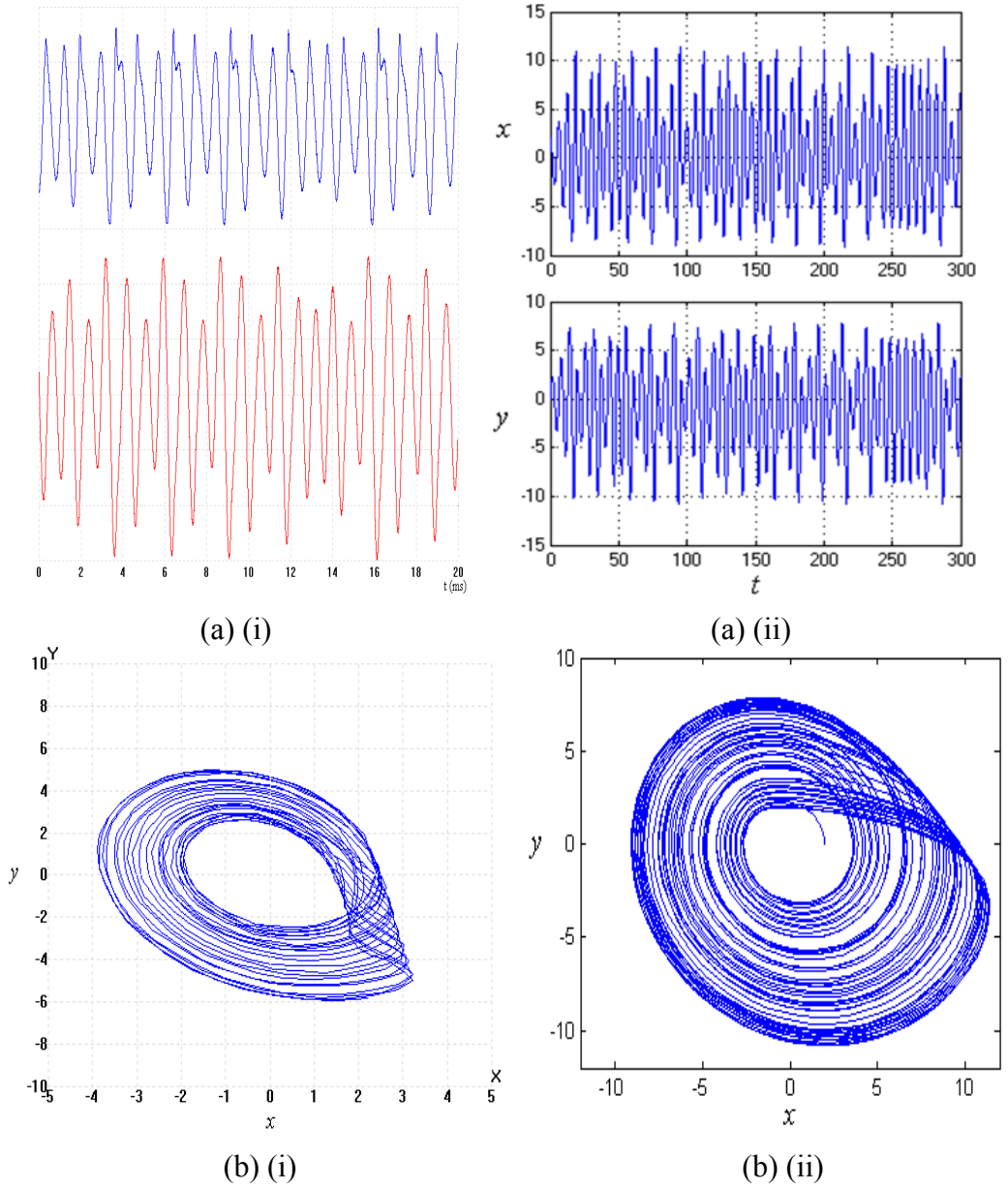
Burada diyodun anahtar işlevinden başka özel bir görevi yoktur. Bu yüzden (x) işareti, 3V' u geçtiği zaman işlemsel kuvvetlendirici saturasyona girer. Modifiye edilen devre denklemleri aşağıda verilmiştir.

$$\begin{aligned}
 \frac{dx}{d\tau} &= -\alpha(r.x + \beta.y + \delta.z) \\
 \frac{dy}{d\tau} &= -\alpha(-x + \gamma.y + 0,02.z) \\
 \frac{dz}{d\tau} &= -\alpha(-g(x) + z) \\
 g(x) &= \begin{cases} 0 & x \leq 3 \\ \mu.(x-3) & x > 3 \end{cases}
 \end{aligned} \tag{3.25}$$

Burada α zaman faktörü olup, $10^4 s^{-1}$ e eşittir. Denklemdaki diğer parametreler; $r=0.05$, $\beta=0.5$, $\delta=1$, $\gamma=R/R_6$ ve $\mu=15'$ dir. Burada $R=10 \text{ K}\Omega$ olup R_6 , $27\text{K}\Omega$ ile $200\text{K}\Omega$ arasında değiştirilen kontrol parametresidir.

Rössler sistemin devre gerçekleştiriminde UA741 tipi genel amaçlı işlemsel kuvvetlendiriciler kullanılmıştır. Devrede kullanılan diyot, 1N4001 tipi genel amaçlı diyottur. Devre, $\pm 15 \text{ V}$ ile beslenmiştir. Devrede kullanılan diğer elemanlar ve değerleri: $C_1=C_2=C_3=1 \text{ nF}$, $R_1=2 \text{ M}\Omega$, $R_2=200 \text{ K}\Omega$, $R_3=R_8=R_{11}=R_{12}=100 \text{ K}\Omega$, $R_4=47$

$K\Omega$, $R_5=75\text{ K}\Omega$, $R_6=200\text{ K}\Omega$ pot., $R_7=5\text{ M}\Omega$, $R_9=R_{10}=R_{14}=10\text{ K}\Omega$, $R_{13}=150\text{ K}\Omega$ ve $R_{15}=68\text{ K}\Omega$ ’ dur. Deneysel çalışmada $R_6=33.6\text{ K}\Omega$ iken devrenin kaotik davranış sergilediği görülmüştür. Deneysel çalışmadan elde edilen (x) ve (y) kaotik dinamikleri ve çeker gösterimleri Şekil 3.34’ de nümerik analiz sonuçları ile karşılaştırmalı verilmiştir. Şekilden deneysel sonuçlar ile SIMULINK™ sonuçlarının uyuştuğu görülmüştür.



Şekil 3.34. (i) Rössler Sisteminin Deneysel Çalışmadan Elde Edilen, (a) (i) Kaotik Devre Dinamikleri: Üstteki İşaret x (2 V/div), Altındaki İşaret y (2 V/div), (b) (i) (x - y) Düzlemindeki Çeker Gösterimi, (ii) Rössler Sisteminin SIMULINK™ Modelinden Elde Edilen Nümerik Analiz Sonuçları, (a) (ii) Kaotik Dinamikleri, (b) (ii) (x - y) Düzlemindeki Çeker Gösterimi.

3.3.1.8. Sprottt Sistemleri

Sprottt sistemleri, literatüre Julien Clinton Sprottt tarafından sunulmuş olup, ‘‘Jerk’’ denklemleri tabanlıdır [114]. Dięer bir deyişle sisteme ait kaotik dinamikler, birbirlerinin türevleri formundadır.

Sprottt sistemine ait durum denklemleri ařağıdaki gibi ifade edilmektedir:

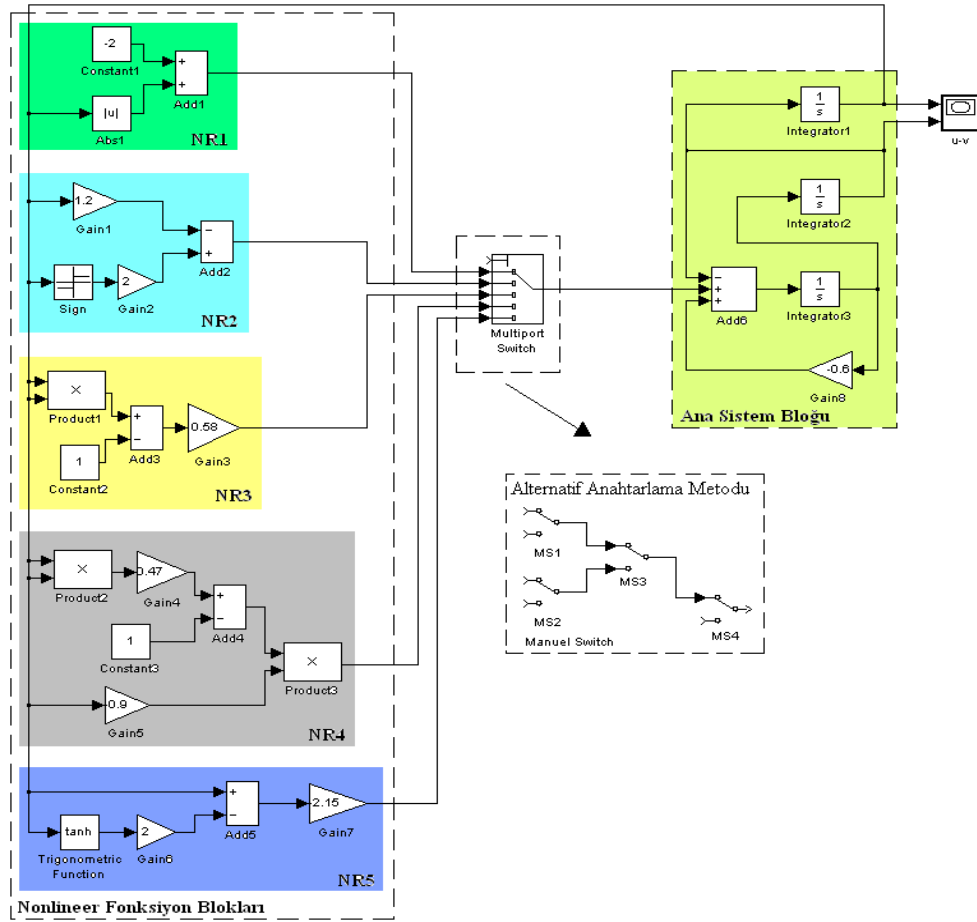
$$\begin{aligned}\frac{du}{d\tau} &= v \\ \frac{dv}{d\tau} &= w \\ \frac{dw}{d\tau} &= f(u) - pw - v\end{aligned}\tag{3.26}$$

Burada u , v ve w , sistemin durum deęişkenleri olup, p ise sistem parametresidir. $f(u)$, doğrusal olmayan fonksiyon olup, sistemin kaosa girmesinde önemli rol oynar. Bu kaotik sistem, farklı tipteki doğrusal olmayan fonksiyonlarla gerçekleştirilebilir [114, 115]. Tablo 3.2’ de bu sistemde kullanılabilen bazı doğrusal olmayan fonksiyonlar ve bu fonksiyonların parametre deęerleri yer almaktadır.

Sprottt sistemlerinin SIMULINK™ modellemesi, Şekil 3.35’ de yer almaktadır [79]. Bu model Denklem (3.26)- (3.31)’ deki matematiksel tanımlamalara göre oluşturulmuştur. Şekilden görüldüğü gibi sistem, Chua osilatöründeki gibi bir adet ana sistem bloęundan, bir adet çoklu anahtarlama mekanizmasından ve beş adet $NR1$, $NR2$, $NR3$, $NR4$ ve $NR5$

Tablo 3.2. Sprottt Sistemlerini Modellemede Kullanılan Doğrusal Olmayan Fonksiyonlar.

Doęrusal olmayan Fonksiyonlar	Fonksiyon Parametreleri	SIMULINK™ Modelindeki Blok Adı
$f(u) = u - b$ (3.27)	$b=2$	$NR1$
$f(u) = -b.u + c.\text{sgn}(u)$ (3.28)	$b=1.2, c=2$	$NR2$
$f(u) = b(u^2 / c - c)$ (3.29)	$b=0.58, c=1$	$NR3$
$f(u) = -b.u.(u^2 / c - 1)$ (3.30)	$b=0.9, 1/c=0.47$	$NR4$
$f(u) = -b(u - 2 \tanh(c.u) / c)$ (3.31)	$b=2.15, c=1$	$NR5$

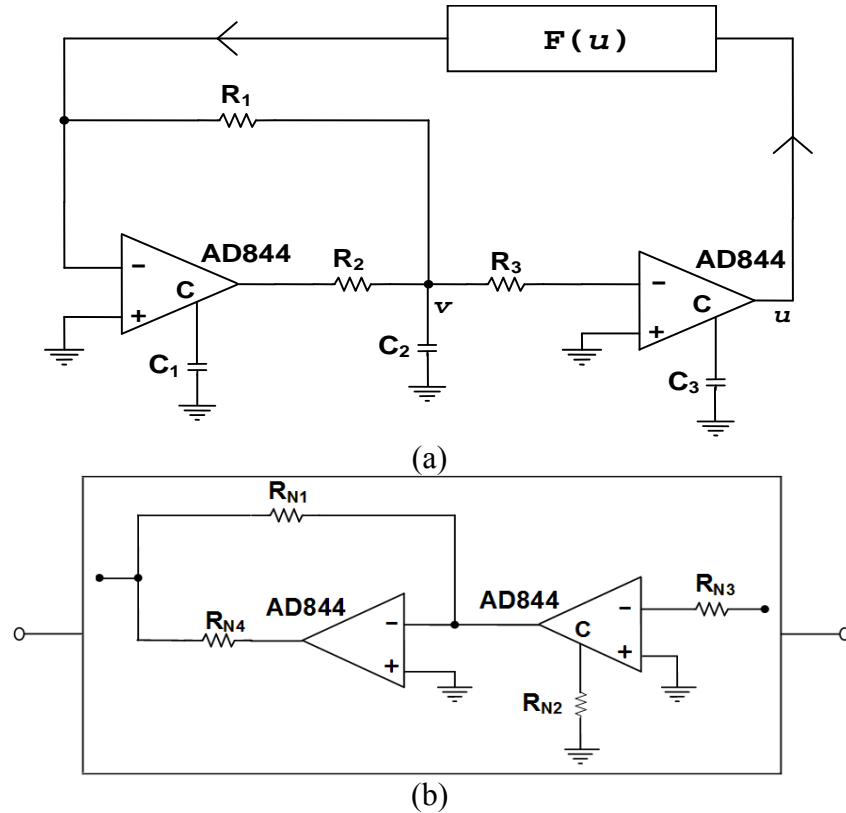


Şekil 3.35. Sprott Sisteminin SIMULINK™ Modeli.

olarak isimlendirilen doğrusal olmayan fonksiyon bloklarından oluşmaktadır. Buradaki çoklu anahtarlama mekanizması, ana sistem bloğu ile doğrusal olmayan fonksiyon blokları arasındaki bağlantıyı sağlamaktadır. Şekilden görüldüğü gibi sistemin SIMULINK™ modelinde ana sistem bloğu, integral alma, toplama/çıkarma ve kazanç bloklarından oluşmaktadır. *NR1*, *NR2*, *NR3*, *NR4* ve *NR5* diye isimlendirilen doğrusal olmayan fonksiyonlar, Tablo 3.2' de yer alan matematiksel ifadelerine uygun olarak gerçekleştirilmiştir. Bu modelde *NR1* için mutlak değer alma, toplama/çıkarma ve sabit değer blokları kullanılmış iken *NR2* için kazanç, toplama/çıkarma ve signum blokları kullanılmıştır. Buradaki signum bloğu, $sgn(u)$ ifadesini gerçeklemek için kullanılmıştır. *NR3*' ün yapısında çarpma, kazanç, toplama/çıkarma ve sabit değer blokları var iken *NR4*' ün yapısında sadece çarpma, kazanç, toplama/çıkarma blokları bulunmaktadır. *NR5* ise kazanç, toplama/çıkarma bloklarının yanı sıra (\tanh) ifadesini gerçekleştirmek için FUNCTION bloğu kullanılarak oluşturulmuştur.

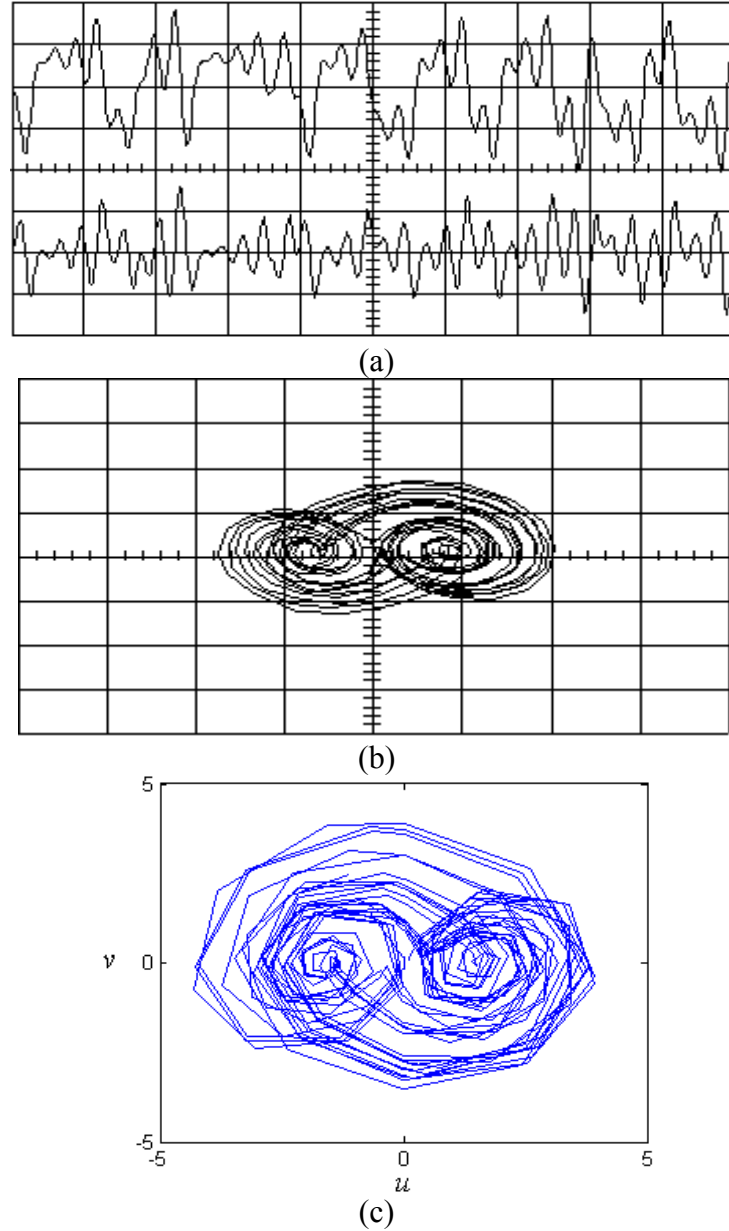
3.3.1.8.1. Sprot Systeminin Donanım Gerçekleştirimi

Farklı doğrusal olmayan fonksiyonlara sahip Sprot sistemleri ayrı elemanlar kullanılarak gerçekleştirilebilir [115-117]. Bunlar arasında Denklem (3.28) ile ifade edilen fonksiyona sahip Sprot sistemi bu kısımda deneysel olarak gerçekleştirilmeye çalışılmıştır. Deneysel gerçekleştirimde devre ± 9 V ile beslenecektir. Önceki kısımlarda yapılan nümerik analiz sonuçlarına göre $u_{\text{maksimum}} = \pm 4$, $v_{\text{maksimum}} = \pm 3.5$ ve $w_{\text{maksimum}} = \pm 3.5$ değerlerine sahiptir. Bu değerler, ölçeklendirme işlemine gerek olmadan elektronik devre gerçekleştiriminin yapılabileceğini göstermektedir. Bu sistemin ayrı elemanlarla devre gerçekleştirim şeması, Şekil 3.36' da verilmiştir [117]. Bu devre, CFOA' lar ile gerçekleştirilen basit RC osilatör devresidir. Denklem (3.26) ile ifade edilen sistemin durum değişkenlerinin gerçekleştirimi, Şekil 3.36(a)' da verilmiş olup bu gerçekleştirimde iki adet ± 9 V ile kutuplanan AD844 tipi CFOA, üç adet kapasitör ve üç adet direnç kullanılmıştır. Denklem (3.28) ile ifade edilen doğrusal olmayan fonksiyonun devre gerçekleştirimi, Şekil 3.36(b)' de yer alırken bu devrede iki adet ± 9 V ile kutuplanan AD844 tipi CFOA ve dört adet direnç kullanılmıştır [117].



Şekil 3.36. Sprot Sisteminin (a) Devre Şeması, (b) Doğrusal Olmayan Fonksiyona Ait Devre Bloğu.

Deneysel çalışmalarda Şekil 3.36(a)'daki devrenin parametreleri $R_1=R_2=R_3=1\text{ K}\Omega$, $C_1=0.02\text{ }\mu\text{F}$, $C_2=0.5\text{ }\mu\text{F}$ ve $C_3=0.01\text{ }\mu\text{F}$ 'dir. Şekil 3.36(b)'de verilen doğrusal olmayan fonksiyona ait bloğun parametre değerleri ise $R_{N1}=10\text{ K}\Omega$ pot, $R_{N2}=R_{N3}=1\text{ K}\Omega$, $R_{N4}=45\text{ K}\Omega$ olarak sabit alınmıştır. Bu deneysel çalışmadan elde edilen kaotik devre dinamikleri ve çeker gösterim sonuçları, sırasıyla Şekil 3.37(a) ve (b)'de verilirken, SIMULINK™



Şekil 3.37. Deneysel Olarak Gerçekleştirilen Sprott Sisteminin, (a) Kaotik Devre Dinamikleri: Üstteki İşaret u (2 V/div), Altındaki İşaret v (200 mV/div), Time/div:2 ms/div, (b) $(u-v)$ Düzlemindeki Çift Çeker Yapısı, x-Ekseni:2 V, y-Ekseni:200 mV, (c) NR2 Bloğu Sistemde İken Sprott Sisteminin SIMULINK™ Modellemesinden Elde Edilen $(u-v)$ Düzlemindeki Çeker Gösterimi.

analizinden elde edilen çeker gösterimi Şekil 3.37(c)' de yer almaktadır. Sonuçlar incelendiğinde deneysel sonuçlar ile nümerik analiz sonuçlarının uyduğu görülmüştür.

3.3.1.9. HSA-Tabanlı Chua Osilatörü

Hücrel Sinir Ağları (HSA), lineer olmayan sistemlerde de kullanılabilen bir ağ yapısıdır. HSA, bir biyolojik sinir ağı gibi gerçek zamanda işaret işleyen, geniş ölçekli, doğrusal olmayan bir analog devredir. En küçük yapı taşlarına *Hücre (cell)* adı verilmektedir [118]. Bu devre yapılarında en yakın komşular, birbirleriyle iletişim kurmakta ve mevcut bilgileri doğrudan değişebilmektedirler [119-121].

İlk olarak 1988 yılında Chua ve Yang [122] tarafından tanıtılan HSA ile ilgili birçok teorik ve deneysel çalışma literatürde yer almaktadır [123-127]. Bu çalışmalar arasında en çok ilgi çeken sistem, otonom Chua devresinin HSA ile gerçekleştirimidir. HSA (Hücrel Sinir Ağları)-tabanlı Chua devresi, Paolo Arena ve ekibi tarafından ortaya konulmuş olup, bu yapıda üç adet HSA hücresi uygun bir ara bağlantı ile birbirine bağlanmıştır [128]. Arena, oluşturduğu bu yapıya, "Durum Kontrollü HSA" adını vermiştir. Durum kontrollü yapı, HSA mimarisini daha esnek kılmakta olup, kaotik devrelerin gerçekleştirimini kolaylaştırmaktadır. HSA modelinde hücre bağlantılarının nasıl yapıldığını gösteren şema Şekil 3.38' de verilmiştir.

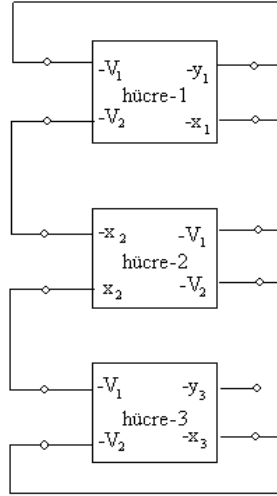
Arena tarafından ortaya konan HSA-tabanlı Chua devresinin durum denklemleri aşağıda yer almaktadır.

$$\begin{aligned}\dot{x}_1 &= -x_1 + 3.857y_1 - 1.5714x_1 + 9x_2 \\ \dot{x}_2 &= -x_2 + x_1 + x_3 \\ \dot{x}_3 &= -x_3 - 14.286x_2 + x_3\end{aligned}\tag{3.32}$$

burada y_1 , parçalı doğrusal fonksiyon olup, aşağıdaki gibi tanımlanmaktadır.

$$y_1 = 0.5(|x_1 + 1| - |x_1 - 1|)\tag{3.33}$$

Şekil 3.39' da HSA-tabanlı Chua osilatörünün SIMULINK™, de gerçekleştirilen modellenmesi verilmiştir. Bu modelde y_1 parçalı doğrusal fonksiyon, kazanç blokları, mutlak değer alma blokları, toplama/çıkarma bloğu ve sabit değer bloğu kullanılarak

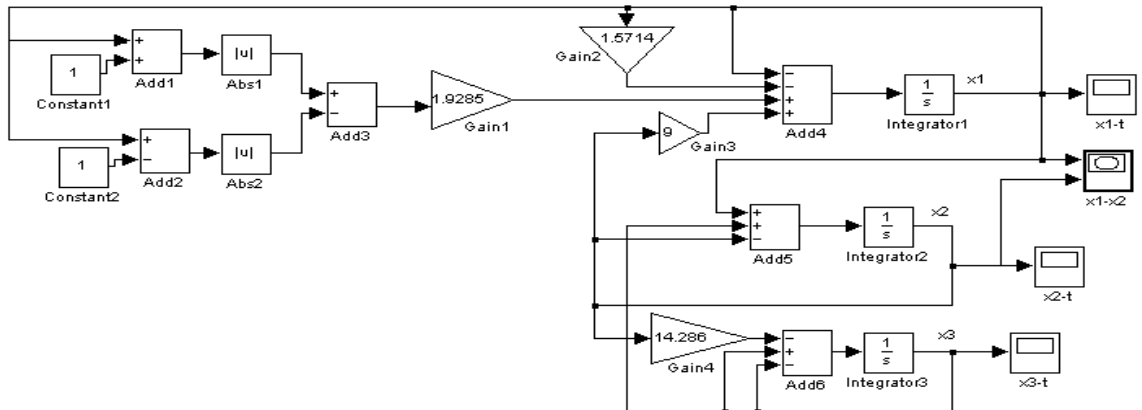


Şekil 3.38. HSA Modelinde Hücre Bağlantı Şeması.

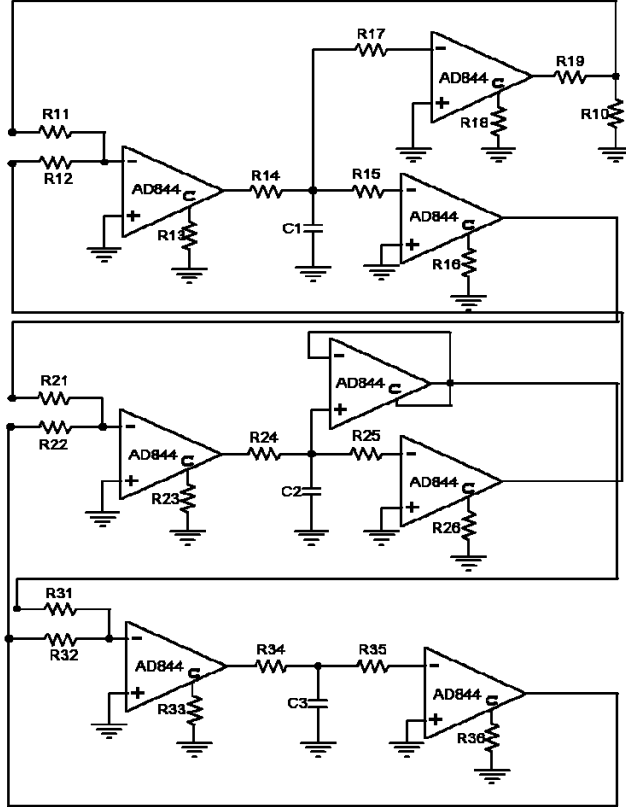
oluşturulmuştur. Osilatörün x_1 , x_2 ve x_3 durum değişkenleri ise integral alma, toplama/çıkarma ve kazanç blokları kullanılarak elde edilmiştir.

3.3.1.9.1. HSA-Tabanlı Chua Osilatörünün Donanım Gerçekleştirimi

HSA-tabanlı Chua osilatörünün nümerik analiz sonuçlarına göre sistem değişkenlerinin maksimum değerleri ± 3 V' dan düşük olduğu için devre denklemleri ölçeklendirilmeden elektronik devre gerçekleştirimi yapılabilir. Literatürde HSA-tabanlı Chua osilatörünün farklı donanım gerçekleştirmeleri bulunmaktadır [126, 128, 129]. Bu kısımda CFOA-tabanlı HSA devresi incelenmiş olup devre şeması Şekil 3.40' da verilmiştir [129]. Bu devrede kullanılan tüm işlemsel kuvvetlendiriciler, ± 9 V ile kutuplanmış AD844 tipi CFOA' dır. Devredeki osilatör durum değişkenleri x_1 , x_2 ve x_3 , sırasıyla C_1 , C_2 ve C_3 kondansatörlerinin üzerine düşen voltajdan elde edilmiştir. Devrede kullanılan



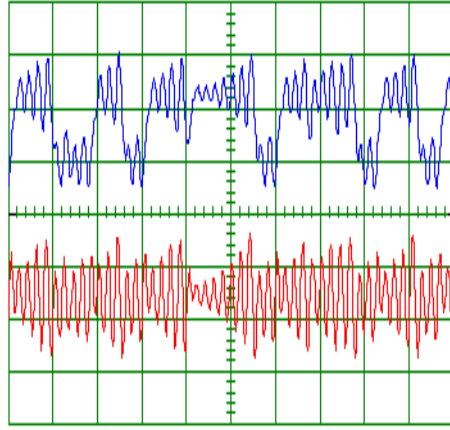
Şekil 3.39. HSA-Tabanlı Chua Osilatörünün SIMULINK™ Modeli.



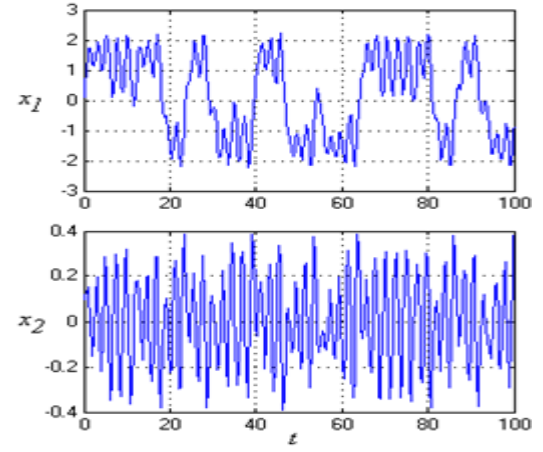
Şekil 3.40. HSA-Tabanlı Chua Osilatörünün Devre Şeması.

elemanların parametre değerleri: $R_{11}=13.2 \text{ k}\Omega$, $R_{12}=5.7 \text{ k}\Omega$, $R_{13}=20 \text{ k}\Omega$, $R_{14}=390 \Omega$, $R_{15}=100 \text{ k}\Omega$, $R_{16}=100 \text{ k}\Omega$, $R_{17}=30 \text{ k}\Omega$, $R_{18}=920 \text{ k}\Omega$, $R_{19}=15 \text{ k}\Omega$, $R_{20}=6 \text{ k}\Omega$, $R_{21}=R_{22}=R_{23}=R_{25}=R_{26}=100 \text{ k}\Omega$, $R_{24}=1 \text{ k}\Omega$, $R_{31}=R_{32}=R_{33}=R_{35}=R_{36}=100 \text{ k}\Omega$, $R_{34}=1 \text{ k}\Omega$, $R_{31}=7.8 \text{ k}\Omega$, $C_1=C_2=C_3=51 \text{ nF}$ dir.

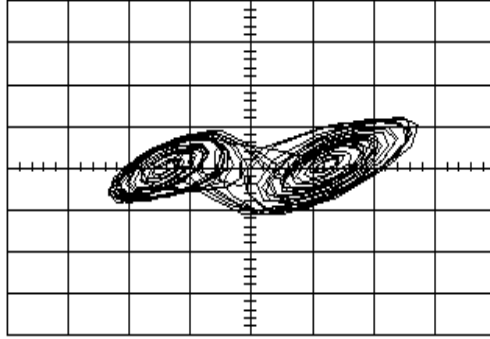
Bu deneysel çalışmadan elde edilen x_1 ve x_2 kaotik dinamikleri ve (x_1-x_2) düzlemindeki çeker gösterimi sırasıyla Şekil 3.41(a) ve (b)' de nümerik analiz sonuçları ile beraber verilmiştir. Deneysel sonuçlardan görülmektedir ki HSA-tabanlı Chua devresi, hem nümerik analiz sonuçları ile örtüşmekte hem de Chua osilatörünün orijinal kaotik davranışlarını sergileyebilmektedir.



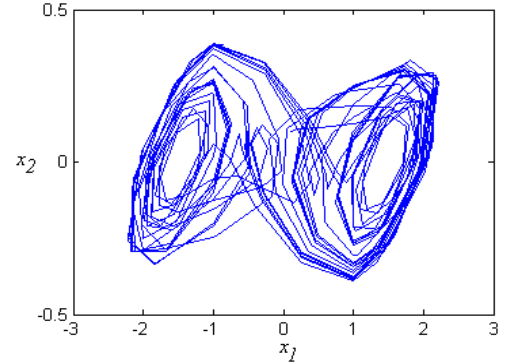
(a) (i)



(a) (ii)



(b) (i)



(b) (ii)

Şekil 3.41. (i) HSA-Tabanlı Chua Osilatörünün Donanım Gerçekleştiriminden Elde Edilen, (a) (i) Kaotik Devre Dinamikleri: Üstteki İşaret x_1 (1 V/div), Altteki İşaret x_2 (500 mV/div), Time/div:50 μ s/div, (b) (i) (x_1 - x_2) Düzlemindeki Çift Çeker Yapısı, x-Ekseni:1 V, y-Ekseni:500 mV, (ii) HSA-Tabanlı Chua Osilatörünün SIMULINK™ Modelinden Elde Edilen Nümerik Analiz Sonuçları, (a) (ii) Kaotik Dinamikleri, (b) (ii) (x - y) Düzlemindeki Çeker Gösterimi.

4. BÖLÜM

FPAA-TABANLI KAOTİK OSİLATÖR TASARIMI VE UYGULAMALARI

4.1. Giriş

Alan programlanabilir analog diziler, analog sistem tasarımında ayırık devre gerçekleştirmelerinin yerine alternatif olarak kullanılabilir. FPAA elemanlarının yazılımları, tasarımı yapılan devreler FPAA üzerine yüklendikten ve işlem yapmaya başladıktan sonra devre çalışırken bile devre parametrelerin kolayca değiştirilmesine ve yeni tasarımların daha kolay yapılmasına imkan sağlar. Bu tez çalışmasında kaotik osilatörlerin tasarımında FPAA' lar kullanılmıştır. Kaotik osilatör gerçekleştirmeleri sırasında bu elemanlar, işlemsel kuvvetlendiricilere, karşılaştırıcılara, analog çarpıcılara ve diğer ayırık elemanlara göre daha etkili, basit ve ekonomiktir. Bilindiği gibi kaotik sistemler, başlangıç şartlarına ve devre parametrelerine karşı aşırı hassas oldukları için bu değerlerde meydana gelebilecek en küçük değişimler devrenin çıkışı üzerinde çok büyük etkilere sahiptir. Bununla birlikte programlanabilme ve yeniden yapılandırılabilme açısından büyük bir esnekliğe sahip olan FPAA' lar, tasarım sürecinde devre parametrelerinin kolaylıkla modifiye edilmesine imkan sağladığı için devre çıkışındaki değişimlerin izlenmesini kolaylaştırmaktadır. Dolayısıyla bu bölümde kaotik osilatör devrelerinin FPAA-tabanlı gerçekleştirmeleri ve senkronizasyon uygulamaları üzerinde durulmuştur. Ayrıca model-tabanlı sistemlerin de FPAA' da gerçekleştirilebileceğini göstermek amacıyla faz kilitlemeli döngü (Phase Locked Loop, PLL) modelinin FPAA' da tasarımı incelenmiştir.

4.2. Programlanabilirlik ve Yeniden Yapılandırılabilirlik

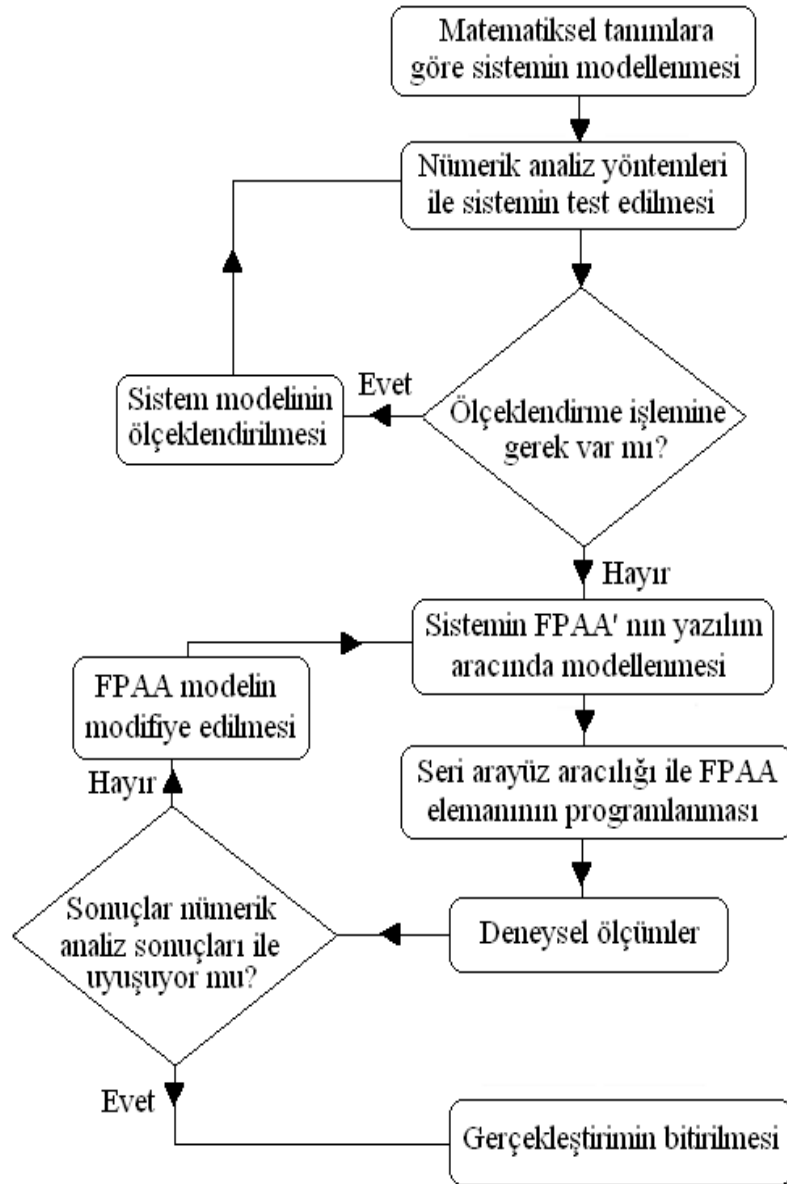
Analog sistemler, analog işaret işleme ve iyileştirme, endüstriyel kontrol, biyomedikal ölçümler gibi uygulamalarda dijital elektronikte arayüz oluşturmada önemli bir role sahiptir. Bunların yanı sıra analog çözümler, düşük doğruluklu, düşük güçlü ve yüksek

hızlı işaret işleme uygulamaları için dijital devrelerle rekabet içersindedir. Dijital tümdevrelerin en önemli özelliği, ürün maliyetinin ve ürünün tasarım ve gelişim zamanının azaltılmasıdır. Bunun için anahtar kavram, birçok sistemin tek donanımda yeniden yapılandırılabilirlik özelliği ile gerçekleştiriminin yapılabilmesidir. Programlanabilirlik ve yeniden yapılandırılabilirlik için alan programlanabilir dizilerin kullanımı tercih edilebilir. Programlanabilir dizilerin özellikleri arasında ihtiyaç duyulan tümdevreyi yapmak için hızlı prototip yapma, anında yeniden yapılandırma ve bilgisayar tabanlı tasarım araçlarının kullanılması sayılabilir [30]. Elektronik sistemler hem analog hem de dijital devre yapılarını içermektedir. Programlama, dijital devrelerde FPGA' ların kullanımı ile yapılabilmektedir. Ama tüm sistemi programlanabilir ve yeniden yapılandırılabilir yapmak için analog kısmında programlanabilir olması gerekmektedir. Tüm sistemin programlanabilir olma ihtiyacının yanında analog sistem tasarımının uzun zaman alması ve pahalı olmasından dolayı ortaya çıkan ihtiyaçların karşılanması amacıyla programlanabilir analog diziler üzerine çalışmalar yapılmış ve bu çalışmalar neticesinde FPAA' lar ortaya konmuştur [3, 30]. FPAA' lar, analog ve analog-dijital devreler için doğruluk, düşük fiyat, hızlı prototip teknikleri sunmaktadır. Bu özelliklerden dolayı programlanabilir ve yeniden yapılandırılabilir yapıya sahip kaotik sistemler, FPAA' lar kullanılarak elde edilmiştir. Aşağıdaki kısımlarda bu sistemler daha detaylı incelenecektir.

4.3. Tasarım Prosedürü ve Deneysel Kurulum

FPAA gerçekleştiriminin akış diyagramı Şekil 4.1' de görülmektedir. Şekilden görüldüğü gibi ele alınan sistem, FPAA üzerinde gerçekleştirilmeden önce nümerik analizi yapılarak test edilir. FPAA elemanı, $\pm 2V$ saturasyon seviyesine sahip olduğu için nümerik analiz sonucuna göre sistem, ihtiyaç duyulması halinde ölçeklendirme işlemine tabii tutulabilir. FPAA' nın yazılım aracında istenilen sistem modellendikten sonra bu model, seri port aracılığıyla FPAA uygulama kartına gönderilir. FPAA uygulama kartından alınan deneysel sonuçlar, nümerik analiz sonuçları ile ve/veya aynı sistemin ayrık elektronik gerçekleştiriminden elde edilen deneysel sonuçlarla karşılaştırılır. Eğer bu sonuçlar birbirlerine uyuyor ise gerçekleştirim tamamlanmış olur. Eğer uymuyorsa FPAA üzerindeki model, sonuçlar uyuşana kadar modifiye edilir.

Deneysel ölçümler, FPAA uygulama kartının giriş/çıkış terminalleri kullanılarak yapılır. Zaman domenindeki ve X-Y modundaki kaotik dinamikleri görüntülemek için Şekil 4.2' deki ölçüm sistemi kullanılmıştır. Bu sistem, herhangi bir elektronik devreden alınan elektriksel işaretlerin bilgisayar ekranında görüntülenmesine ve nümerik değerler dizisi şeklinde kaydedilmesine imkan sağlamaktadır [130].



Şekil 4.1. FPAA-Tabanlı Tasarım ve Gerçekleştirim Diyagramı.



Şekil 4.2. FPAA-Tabanlı Gerçekleştirimin Deneysel Kurulumu. Bu Kurulumda Anadigm Firmasının AN221E04 Tipi FPAA [6] Kullanılmıştır.

4.4. FPAA-Tabanlı Kaotik Osilatör Tasarımları

Bu kısımda, nümerik analizleri ve ayrık elemanlarla deneysel çalışmaları incelenen Bölüm 3.3.1.1-3.3.1.9 arasında yer alan kaotik osilatör yapılarının FPAA-tabanlı deneysel gerçekleştirmeleri üzerinde durulmuştur.

4.4.1. FPAA-Tabanlı Chua Osilatörü

Farklı doğrusal olmayan fonksiyonlara sahip Chua osilatörü, programlanabilirlik ve yeniden yapılandırılabilirlik açısından FPAA kullanılarak gerçekleştirilecek potansiyele sahip sistemlerdendir. Chua osilatörü, Bölüm 3.3.1.1’ de SIMULINK™ modellemesi ile incelenmiş ve donanım olarak gerçekleştirilebileceği parçalı doğrusal fonksiyon kullanılarak verilmiştir. Chua osilatörü diğer doğrusal olmayan fonksiyonlar kullanılarak da donanım gerçekleştirimi yapılabilir. Donanım gerçekleştirimi yapılırken farklı değerlere sahip direnç, kondansatör, indüktör elemanlarının yanı sıra değişik işlemsel kuvvetlendiricilerin, çarpıcıların kullanımına da ihtiyaç duyulmaktadır. Ayrıca eleman değerlerinin tam olarak ayarlanması problemi, kullanılacak işlemsel kuvvetlendiricilerin temin edilme sıkıntısı ve hepsinin deneysel gerçekleştirimi için ayrı ayrı geçen süre de ilave edilirse donanım gerçekleştirmelerinin ne kadar zor olduğu ortaya çıkmaktadır. Bu zorlukların üstesinden gelmek için programlanabilir ve yeniden yapılandırılabilir özelliğe sahip FPAA kullanılmıştır. FPAA’ nın yeniden yapılandırılabilirlik özelliği sayesinde tasarımı yapılan devreler FPAA üzerine

yüklendikten ve deneysel sonuçlar alındıktan sonra başka bir sistem FPAA' ya gönderilebilir ve çalışmalar aksamadan devam ettirilebilir. Ayrıca FPAA' da tasarlanan devre çalışırken bile devre parametrelerin kolayca değiştirilmesi FPAA' nın programlanabilme özelliğini göstermektedir. Bu kısımda programlanabilir ve yeniden yapılandırılabilir formda, farklı doğrusal olmayan fonksiyonları kullanarak FPAA-tabanlı Chua devre modellerinin gerçekleştirimi yapılmıştır [38]. Bu gerçekleştirim sayesinde yukarıda bahsedilen ayrık elemanlarla gerçekleştirim sıkıntılarından kurtulmuş olup programlanabilir yapıya sahip modeller ortaya konulmuş olur. Chua osilatörünün FPAA-tabanlı gerçekleştirimlerinde doğrusal olmayan fonksiyonlar olarak parçalı doğrusal fonksiyon [82], kübik fonksiyon [83, 84], parçalı ikinci dereceden fonksiyon [55, 85] ve trigonometrik fonksiyon [55, 86] kullanılmıştır. Chua osilatörünün durum denklemleri Bölüm 3' de Denklem (3.5)-(3.9)' da verilmiştir.

Bölüm 3.3.1.1' de verilen Chua osilatörünün nümerik analiz sonuçları dikkate alınarak FPAA tasarımları yapılmıştır. Daha önceden bahsedildiği gibi FPAA elemanı $\pm 2V$ saturasyon seviyesine sahiptir. Bu kısıtlama nedeniyle bazı tasarımlarda ölçeklendirme işlemi de yapılmıştır. Ölçeklendirme işleminden sonra, FPAA' nın yazılım aracı ile modellenen sistemler FPAA uygulama kartına gönderilmiştir. Ölçeklendirilen sistem denklemleri ve FPAA' nın yazılım aracı ile modellenen Chua sisteminin gerçekleştirim şemaları, Tablo 4.1' de yer almaktadır.

4.4.1.1. FPAA-Tabanlı Chua Devre Modeli-I

Chua devresinin FPAA-tabanlı gerçekleştirimi için ilk olarak doğrusal olmayan fonksiyonlar arasında bulunan parçalı doğrusal fonksiyon kullanılmıştır. Bu fonksiyon kullanılarak gerçekleştirilecek FPAA tasarımında önce ölçeklendirme işlemi yapılmıştır. Ölçeklendirme işleminden önce osilatörün nümerik analizi yapılır. Chua osilatörünün Denklem (3.5) ve (3.6) ile verilen denklem takımına göre Bölüm 3.3.1.1' de nümerik analizi yapılmış olup analiz sonucuna göre durum değişkenleri $|x|= 3V$, $|y|= 0.6V$ ve $|z|= 4V$ olduğu görülmüştür. Bu modelde ölçeklendirme işlemi için tüm durum değişkenleri 2' ye bölünerek yeniden düzenlenmiştir ve düzenlenen devre denklemleri de Tablo 4.1' de Denklem (4.1) ile verilmiştir. Tablo 4.1' deki parçalı doğrusal fonksiyon kullanılarak gerçekleştirilmek istenen FPAA-tabanlı gerçekleştirimde

Tablo 4.1. Chua Devresinin FPAA Gerçekleştirim Şemaları.

<p>MODEL-1</p> <p>[Doğrusal Olmayan Fonksiyon: Parçalı Doğrusal]</p> $x = 10\mu y - 3.2x + 2.95(x+1 - x-1)$ $\dot{y} = x - y + z$ $z = -14.87y$ <p>(4.1)</p>	
<p>MODEL-2</p> <p>[Doğrusal Olmayan Fonksiyon: Kübik Benzeri]</p> $\dot{x} = 2.4\mu y + 2.7x - 3.925x^3$ $\dot{y} = 4.167x - y + 7.083z$ $z = -2.099y$ <p>(4.2)</p>	
<p>MODEL-3</p> <p>[Doğrusal Olmayan Fonksiyon: Trigonometrik]</p> $\dot{x} = 6.3\mu y - 6.3x + 20 \tanh(0.38x)$ $\dot{y} = 0.63x - 0.63y + z$ $z = -5.6y$ <p>(4.3)</p>	
<p>MODEL_4</p> <p>[Doğrusal Olmayan Fonksiyon: Parçalı İkinci Dereceden Fonksiyon]</p> $\dot{x} = 3.03\mu y + 1.428x - 0.635x 3.3x $ $\dot{y} = 3.3x - y + 4z$ $z = -3.7175y$ <p>(4.4)</p>	

SUMFILTER blok, x , y ve z durum değişkenlerini elde etmek için kullanılmıştır. Bu bloğu kullanabilmek için bloğun Denklem (2.5) ile ifade edilen çıkış fonksiyonunun sağlanması gerekmektedir. Bu amaçla kaotik sisteme ait denklem takımına Laplace dönüşümü uygulanır. Bu durumda devre denklemleri aşağıdaki gibi tanımlanır.

$$\begin{aligned} s x(s) &= 10\mu y(s) - 3.2x(s) + 2.95(|x(s)+1| - |x(s)-1|) \\ s y(s) &= x(s) - y(s) + z(s) \\ s z(s) &= -14.87y(s) \end{aligned} \quad (4.5)$$

SUMFILTER bloğun çıkış fonksiyonunda yer alan $(s+2\pi f_0)$ ifadesini elde etmek için yukarıdaki her bir denklem için eşitliğin her iki tarafına sırasıyla $x(s)$, $y(s)$ ve $z(s)$ ifadeleri ilave edilir.

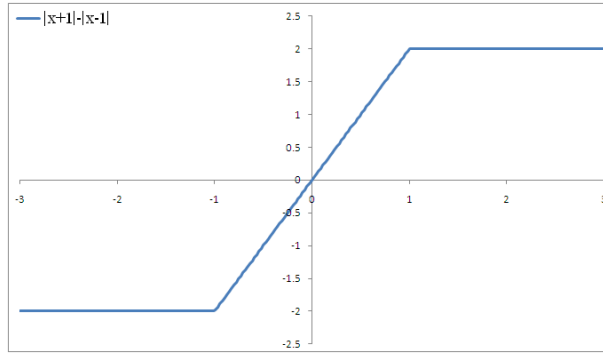
$$\begin{aligned} s x(s) + x(s) &= x(s) + 10\mu y(s) - 3.2x(s) + 2.95(|x(s)+1| - |x(s)-1|) \\ s y(s) + y(s) &= y(s) + x(s) - y(s) + z(s) \\ s z(s) + z(s) &= z(s) - 14.87y(s) \end{aligned} \quad (4.6)$$

Bu denklem takımı yeniden düzenlenerek aşağıdaki gibi ifade edilir.

$$\begin{aligned} (s+1)x(s) &= 10\mu y(s) - 2.2x(s) + 2.95(|x(s)+1| - |x(s)-1|) \\ (s+1)y(s) &= x(s) + z(s) \\ (s+1)z(s) &= z(s) - 14.87y(s) \end{aligned} \quad (4.7)$$

Denklem (4.7)'de yer alan $(s+1)$ ifadesi ile *SUMFILTER* bloğun Denklem (2.5)' de verilen çıkış fonksiyonundaki $(s+2\pi f_0)$ ifadesi birbirine eşitlenirse bu durumda f_0 köşe frekansı; $2\pi f_0=1$ ' den $f_0=0.159$ kHz olmuş olur.

Denklem (4.7)' ye oluşturulan FPAA gerçekleştirim şeması Tablo 4.1' de verilmiştir. Bu gerçekleştirim şemasında devre kazançları *SUMFILTER* blok kazançları tarafından gerçekleştirilmektedir. Kullanıcı tarafından belirlenen *TRANSFER FUNCTION* blok, parçalı-doğrusal fonksiyonu gerçekleştirmek için kullanılmıştır ve bu fonksiyonun grafiği Şekil 4.3' de verilmiştir. Burada $(|x+1|-|x-1|)$ fonksiyonu için hazırlanan dosya MICROSOFT Excel™ programında oluşturulmuş ve (.csv) olarak kaydedilmiştir. Kaydedilen bu dosya *TRANSFER FUNCTION* bloğa yüklenerek parçalı doğrusal fonksiyon gerçekleşmiştir. Bu modelde kullanılan bloklar ve bu blokların parametre



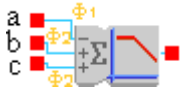
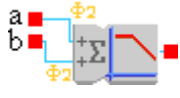
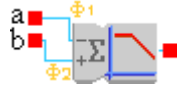

Şekil 4.3. *TRANSFER FUNCTION* Bloкта Elde Edilen Parçalı Doğrusal Fonksiyonun Grafiği.

değerleri Tablo 4.2’ de verilmiştir. Denklem (4.7)’ ye göre *SUMFILTER* blokların köşe frekansları, 0.159 kHz bulunmasına rağmen Chua osilatör modelinin FPAA gerçekleştirimi bu parametre değeriyle istenilen sonuçları vermemiştir. Köşe frekansı 0.8 kHz olarak değiştirildiğinde tasarlanan model kaotik davranış sergilemiştir. Bu model gerçekleştiriminde programlanabilirlik, Denklem (4.7)’ de yer alan (μ) parametresi ile yapılabilmektedir. (μ) parametre değişimi, FPAA içerisinde programlanabilirlik özelliği vasıtasıyla kolayca yapılabilmektedir. (μ) parametresi 1 iken devre orijinal Chua devresinin dinamiklerini üretmektedir. Devre çalışırken ($\mu=0.88$) olarak değiştirildiğinde devre tek çekerli davranış sergilemekte, ($\mu=0.8$) değeri verildiğinde devre davranışı periyot-2 işaretine dönüşmektedir. FPAA-tabanlı deneysel gerçekleştirim sonucunda elde edilen kaotik devre dinamikleri ve ($x-y$) düzlemindeki çeker gösterimleri, Şekil 4.4(a) ve (b)’ de verilmiş, (μ) parametresinin değiştirilmesiyle elde edilen tek çekerli ve periyot-2 davranışı sırasıyla Şekil 4.4(c) ve (d)’ de gösterilmiştir. Parçalı doğrusal fonksiyon kullanılan FPAA-tabanlı Chua devresinin kaotik dinamik ve çeker gösterim sonuçları, Chua devresinin Bölüm 3.3.1.1’ de verilen SIMULINKTM sonuçlarıyla ve ayırık elemanlarla tasarlanan donanım gerçekleştirimi sonuçları ile uyuşmaktadır.

4.4.1.2. FPAA-Tabanlı Chua Devre Modeli-II

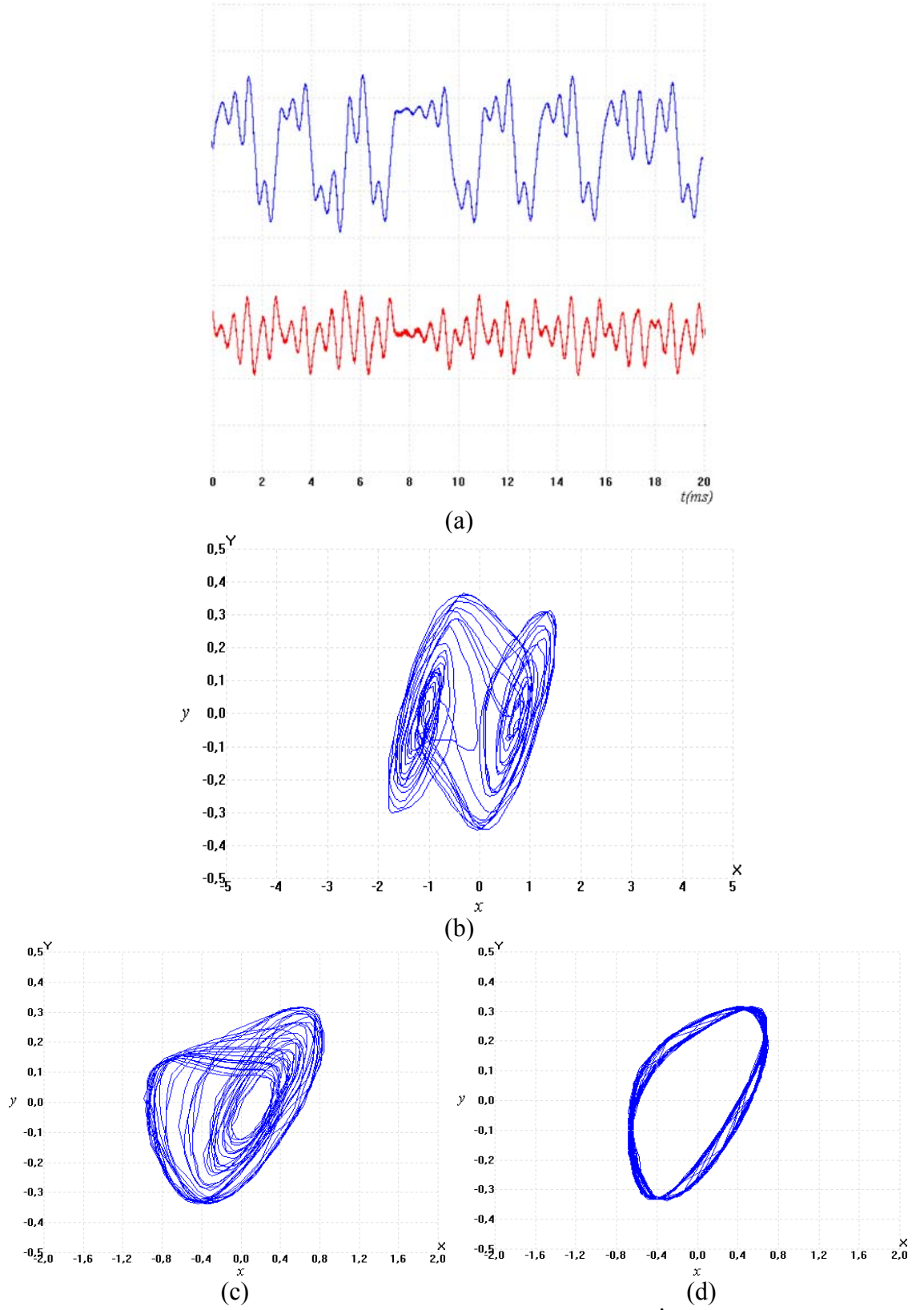
Yeniden yapılandırma potansiyelindeki sistem FPAA-tabanlı olarak gerçekleştirildikten ve deneysel sonuçlar alındıktan sonra, modifiye edilen sistem FPAA’ ya gönderilebilir ve çalışmalar aksamadan devam ettirilebilir. Bu kapsamda FPAA-tabanlı Chua devre modeli-I gerçekleştirdikten sonra Chua devre modeli-II ve diğerlerinin FPAA gerçekleştirimi üzerine çalışmalar devam ettirilmiştir.

Tablo 4.2. Denklem (4.7) ile İfade Edilen FPAA-Tabanlı Chua Devre Modeli-I' in CAM Modülleri ve Parametre Değerleri.

CAM Modülü	CAM No.su	Ayarlanan Parametresi	Parametre Değeri
 <p><i>SUMFILTER</i></p>	1	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı c Giriş Kazanç Katsayısı	0.8 2.2 2.95 10
 <p><i>SUMFILTER</i></p>	2	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.8 1 1
 <p><i>SUMFILTER</i></p>	3	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.8 9.9 1
 <p><i>TRANSFER FUNCTION</i></p>	4		

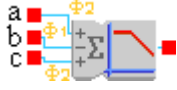
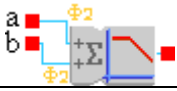

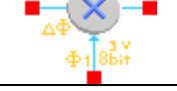
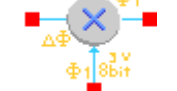
Denklem (4.2) ile ifade edilen matematiksel ifadeye göre oluşturulan Chua devre modeli-II' nin gerçekleştirim şeması Tablo 4.1' de verilmiştir. Bu gerçekleştirim şemasında ve bundan sonraki verilecek FPAA gerçekleştirim şemalarında *SUMFILTER* bloklar kullanıldığı için FPAA-tabanlı gerçekleştirimlerde, Chua devre modeli-I' de olduğu gibi kaotik sistemlere ait denklemlerin Laplace dönüşümleri yapılmıştır. Bu dönüşümlerdeki temel prensip aynı olduğu için denklem kalabalıklığına yol açmamak amacıyla Chua devre modeli-I için yapılan işlemlerin tekrar ifade edilmesi gereksiz görülmüştür.

Chua devre modeli-II' nin gerçekleştiriminde doğrusal olmayan fonksiyon olarak kübik benzeri doğrusal olmayan fonksiyon kullanılmıştır. Kübik benzeri doğrusal olmayan fonksiyon, *MULTIPLIER* bloklar kullanılarak elde edilmiştir. Sistem dinamikleri (x), (y) ve (z)' i elde etmek için, *SUMFILTER* blokları kullanılmıştır. Bu gerçekleştirimde kullanılan bloklar, parametre değerleri ile birlikte Tablo 4.3' de yer almaktadır. (μ) parametresi, Chua devre modeli-I' de olduğu gibi bu modelde de FPAA' nın programlanabilirlik özelliğini gösteren parametre olup sistemin farklı dinamikler sergilemesi için farklı değerlere ayarlanmıştır. (μ) parametresi 1 iken devre çift çeker sergilemekte, 0.833 iken devrede tek çeker oluşmakta ve 0.708 iken devre periyodik



Şekil 4.4. FPAA-Tabanlı Chua Devre Modeli-I' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (1 V/div), Alttağı İşaret y (0.4 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.88$ İken Elde Edilen Tek Çeker, (d) $\mu=0.8$ İken Elde Edilen Periyot 2 Davranışı.

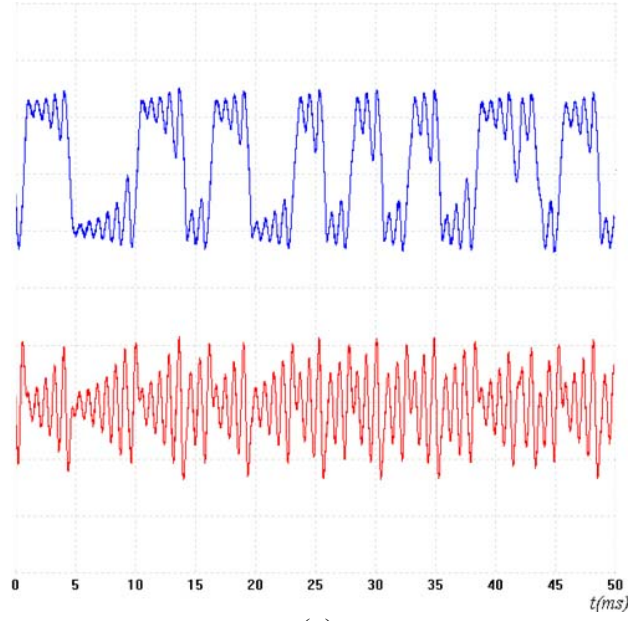
Tablo 4.3. Denklem (4.2) ile İfade Edilen FPAA-Tabanlı Chua Devre Modeli-II' nin CAM Modülleri ve Parametre Değerleri.

CAM Modülü	CAM No.su	Ayarlanan Parametresi	Parametre Değeri
 <i>SUMFILTER</i>	1	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı c Giriş Kazanç Katsayısı	0.4 3.7 3.925 2.4
 <i>SUMFILTER</i>	2	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.4 4.167 7.083
 <i>SUMFILTER</i>	3	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.4 1 2.099
 <i>MULTIPLIER</i>	4	Çarpım Faktörü	1
 <i>MULTIPLIER</i>	5	Çarpım Faktörü	1

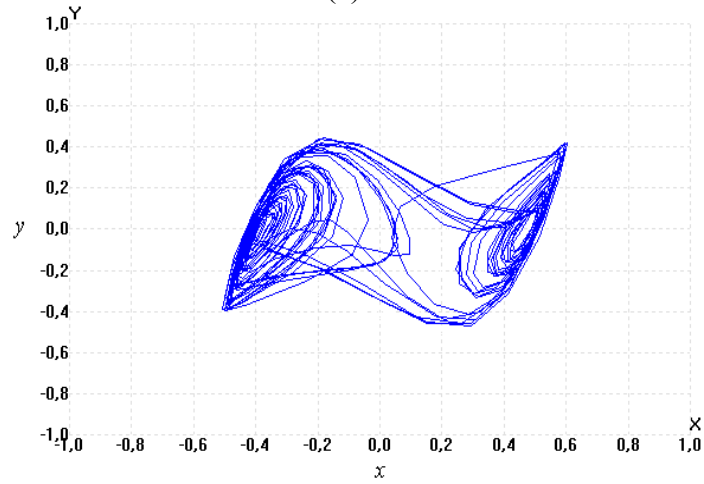
davranış sergilemektedir. Bu modele ait kaotik dinamikler ve çift çeker gösterimi Şekil 4.5(a) ve (b)' de gösterilirken (μ) katsayısının değişimi ile oluşan tek çeker ve periyot-1 gösterimleri Şekil 4.5(c) ve (d)' de verilmiştir.

4.4.1.3. FPAA-Tabanlı Chua Devre Modeli-III

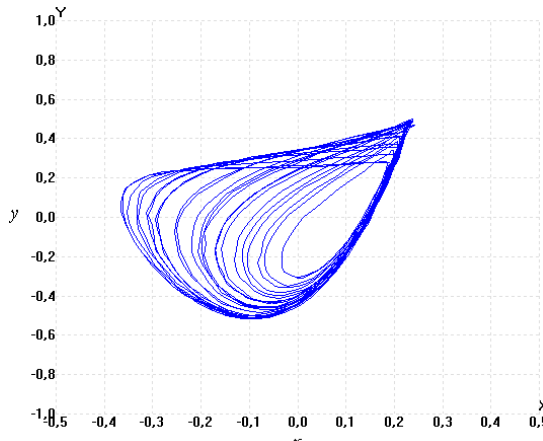
Chua devresinin FPAA-tabanlı gerçekleştirmelerinden üçüncüsü, Denklem (4.3)' de yer alan matematiksel modele göre yapılmış olup devre şeması da Tablo 4.1' de verilmiştir. Bu gerçekleştirimde $[tanh(.)]$ terimini içeren trigonometrik doğrusal olmayan fonksiyon kullanılmıştır. $[tanh(.)]$ terimini içeren trigonometrik fonksiyonu gerçekleştirmek için kullanıcı tarafından belirlenen *TRANSFER FUNCTION* blok kullanılmıştır. $[tanh(.)]$ terimine ait fonksiyon dosyası ExcelTM, de oluşturulur ve .csv uzantılı olarak kaydedilir. Daha sonra da bu dosya, *TRANSFER FUNCTION* bloğa yüklenir. ExcelTM, de oluşturulan $[tanh(.)]$ fonksiyonun grafiği Şekil 4.6' da verilmiştir. Bu modelde diğer modellerde olduğu gibi sistem dinamiklerini elde etmek için *SUMFILTER* blokları kullanılmıştır. Bu gerçekleştirimde kullanılan bloklar ve parametre değerleri Tablo 4.4' de yer almaktadır.



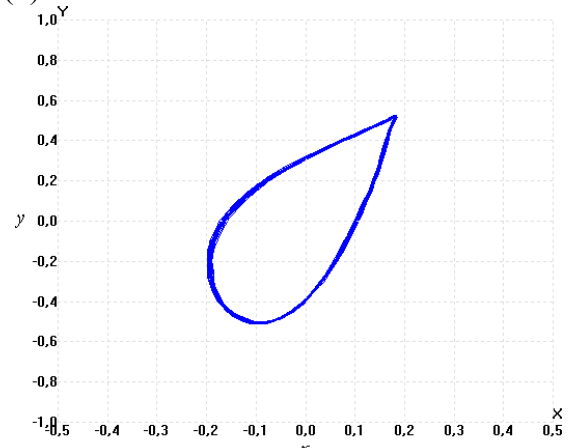
(a)



(b)

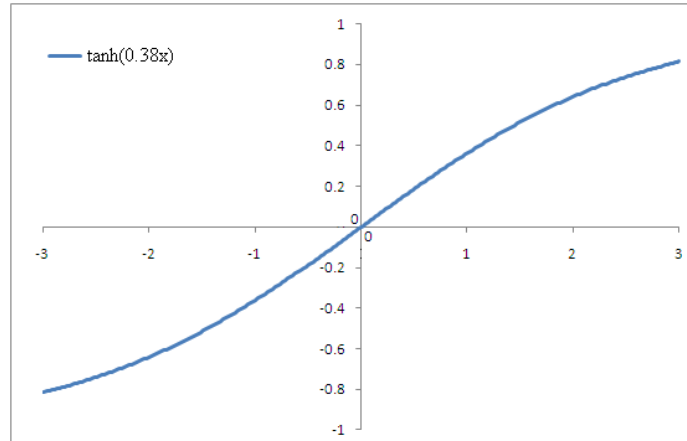


(c)



(d)

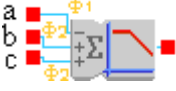
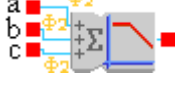


Şekil 4.5. FPAA-Tabanlı Chua Devre Modeli-II' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (0.4 V/div), Altteki İşaret y (0.4 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.833$ İken Elde Edilen Tek Çeker, (d) $\mu=0.708$ İken Elde Edilen Periyot 1 Davranışı.

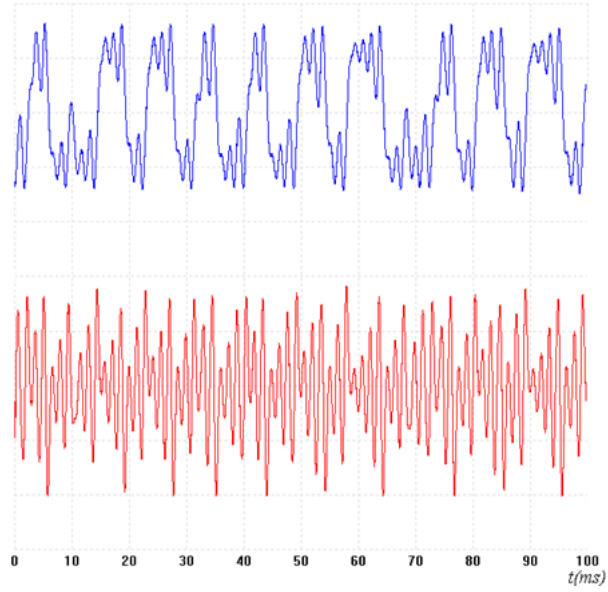


Şekil 4.6. *TRANSFER FUNCTION* Bloкта Elde Edilen [$\tanh(.)$] Fonksiyonun Grafiği.

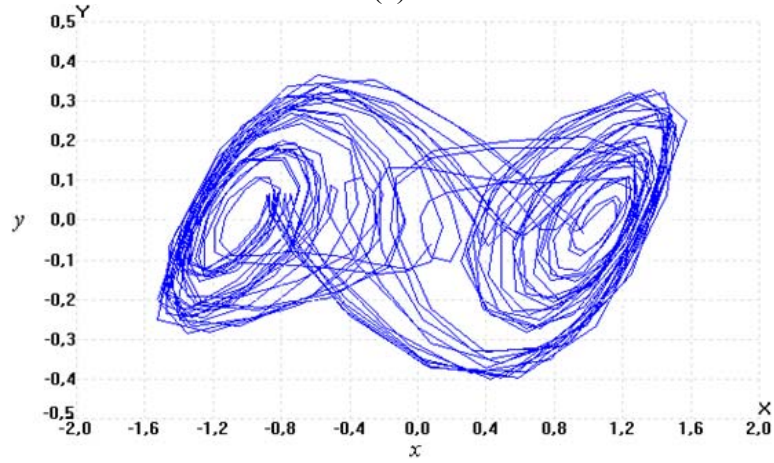
Bu gerçekleştirimde (μ) katsayısı, modelin değişik dinamikler sergilemesi için programlanabilir parametre olarak seçilmiştir. Modelde ($\mu=1$) iken sistem kaotik davranışlar sergilemektedir ve sergilediği bu davranışlar Şekil 4.7(a) ve (b)' de verilmiştir. Bu ayarlanabilir parametre 0.92 iken model tek çekerli davranış sergilerken 0.825 değeri için periyot-1 davranışına sahip olmaktadır ve bu davranışlar, sırasıyla Şekil 4.7(c) ve (d)' de gösterilmiştir.

Tablo 4.4. Denklem (4.3) ile İfade Edilen FPAA-Tabanlı Chua Devre Modeli-III' ün CAM Modülleri ve Parametre Değerleri.

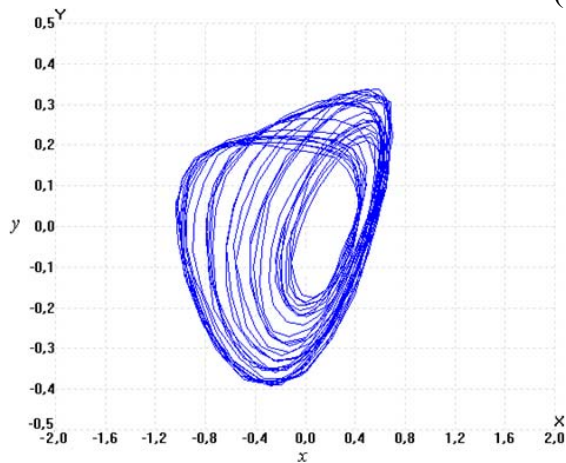
CAM Modülü	CAM No.su	Ayarlanan Parametresi	Parametre Değeri
<p><i>SUMFILTER</i></p> 	1	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı c Giriş Kazanç Katsayısı	0.4 5.3 20 6.3
<p><i>SUMFILTER</i></p> 	2	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı c Giriş Kazanç Katsayısı	0.4 0.37 0.9 1
<p><i>SUMFILTER</i></p> 	3	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.4 1 5.6
<p><i>TRANSFER FUNCTION</i></p> 	4		



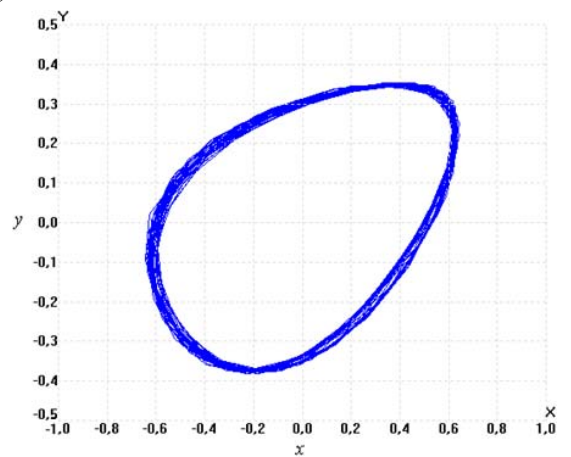
(a)



(b)



(c)



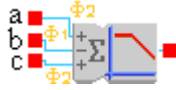
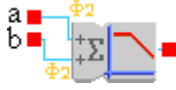


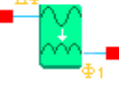
(d)

Şekil 4.7. FPAA-Tabanlı Chua Devre Modeli-III' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (1 V/div), Alttağı İşaret y (0.4 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.92$ İken Elde Edilen Tek Çeker, (d) $\mu=0.825$ İken Elde Edilen Periyot 1 Davranışı.

4.4.1.4. FPAA-Tabanlı Chua Devre Modeli-IV

Denklem takımı Denklem (4.4)' de yer alan parçalı ikinci dereceden doğrusal olmayan fonksiyon kullanılarak FPAA-tabanlı Chua devre modellerinden dördüncüsü gerçekleştirilmiştir. Bu gerçekleştirimde de sistem durum değişkenleri x , y ve z , diğer modellerde olduğu gibi *SUMFILTER* blokların çıkışlarından elde edilirken devre kazançları, bu blokların kazançları ile ayarlanmıştır. Bu gerçekleştirimde *SUMFILTER* blokların dışında *MULTIPLIER* ve *RECTIFIERHALF* bloklar kullanılmış olup bu bloklar, parçalı ikinci dereceden doğrusal olmayan fonksiyonu gerçekleştirimde kullanılmıştır. Doğrusal olmayan bu fonksiyon, kullanıcı tarafından belirlenen *TRANSFER FUNCTION* blok kullanılarak da gerçekleştirilebilir. FPAA programının esnekliğinin gösterilmesi amacıyla alternatif bir yaklaşım kullanılarak *MULTIPLIER* ve *RECTIFIERHALF* bloklar ile gerçekleştirim yapılmıştır. Denklem (4.4)' de yer alan $(|3.3x|)$ ifadesi *RECTIFIERHALF* blok ile elde edilirken $[x \cdot |3.3x|]$ işlemi için *MULTIPLIER* blok kullanılmıştır. Tablo 4.5' de bu modele ait FPAA blokları ve bu bloklarda kullanılan parametreler yer almaktadır.

Tablo 4.5. Denklem (4.4) ile İfade Edilen FPAA-Tabanlı Chua Devre Modeli-IV' ün CAM Modülleri ve Parametre Değerleri.

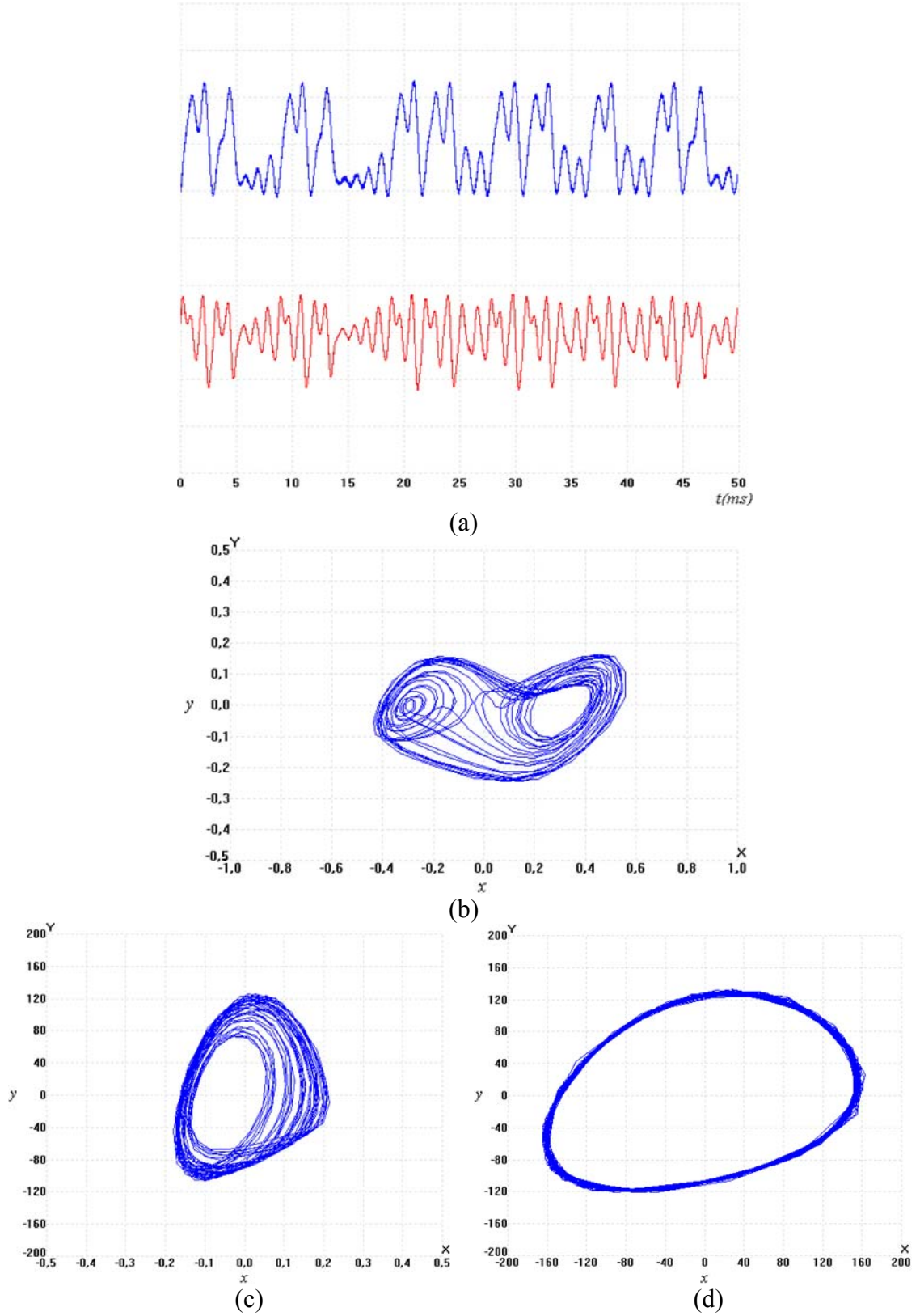
CAM Modülü	CAM No.su	Ayarlanan Parametresi	Parametre Değeri
 <i>SUMFILTER</i>	1	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı c Giriş Kazanç Katsayısı	0.4 2.46 0.692 3.53
 <i>SUMFILTER</i>	2	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.4 3.3 4
 <i>SUMFILTER</i>	3	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.4 3.7175 1
 <i>MULTIPLIER</i>	4	Çarpım Faktörü	1
 <i>RECTIFIERHALF</i>	5	Kazanç Katsayısı	3.3

Bu gerekleřtirimde ayarlanabilir parametre diđer modellerde olduđu gibi (μ)' dır. (μ) katsayısı 1 iken devre modelinden kaotik dinamikler elde edilmektedir. Bu katsayı 0.946 seildiđinde ise sistem tek eker oluřturmaktadır. Periyodik davranıř elde etmek iin katsayı deđerinin 0.898' e ayarlanması gerekmektedir. Bu FPAA-tabanlı kaotik modelden elde edilen deneysel sonular, Őekil 4.8' de gsterilmektedir.

4.4.2. FPAA-Tabanlı Kaotik Colpitts Osilatr

Blm 3.3.1.2' de detaylı Őekilde verilen kaotik Colpitts osilatrnn FPAA-tabanlı gerekleřtirmisi mmkndr. Bu gerekleřtirim iin kullanılan devre Őeması, Tablo 4.6' da verilmiř olup, Denklem (4.8)' e gre oluřturulmuřtur. Bu gerekleřtirimde $f(z)$ dođrusal olmayan fonksiyon, *TRANSFER FUNCTION* blok kullanılarak gereklenmiřtir. *TRANSFER FUNCTION* blok, kullanıcı tarafından hazırlanan (.csv) uzantılı dosyaların yklenmesi ile alıřan bir bloktur. Buradaki $f(z)$ dođrusal olmayan fonksiyon iin hazırlanan dosya MICROSOFT ExcelTM programında *if, else* komutları ile oluřturulmuř ve (.csv) olarak kaydedilmiřtir. Kaydedilen bu dosya *TRANSFER FUNCTION* blođa yklenerek dođrusal olmayan fonksiyon blođu gereklenmiřtir. Őekil 4.9' da *TRANSFER FUNCTION* blokta gereklenen $f(z)$ dođrusal olmayan fonksiyonun grafiđi verilmiřtir.

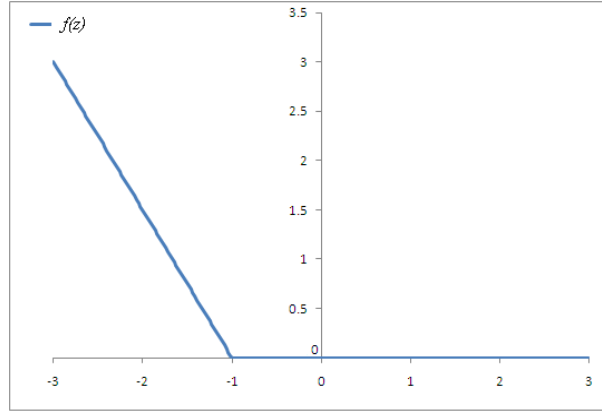
Denklem takımında dođrusal olmayan blođun dıřında kalan ifadeler iin *SUMFILTER*, *SUMDIFF* ve *VOLTAGE REFERENCE* blokları kullanılmıřtır. Bu bloklardan *SUMFILTER* blokları hem sistem dinamiklerinin elde edilmesinde hem de devre kazanlarının belirlenmesinde kullanılmıřlardır. Devreye ait denklem takımında yer alan (y) dinamiđini elde etmek iin 4 adet parametrenin toplama ve/veya ıkarma iřlemine sokulması gerekmektedir. *SUMFILTER* blođunun ise en fazla 3 adet giriři bulunmaktadır. Bu yzden toplama/ıkarma iřlemi iin *SUMDIFF* blođu kullanılmıřtır. Ayrıca kaotik Colpitts osilatrne ait hem (y) dinamiđinde hem de (z) dinamiđinde yer alan sabit deđerleri elde etmek iin de bir adet *VOLTAGE REFERENCE* blođu kullanılmıřtır. FPAA-tabanlı bu gerekleřtirimde parametre deđiřimi ile devrenin periyot-1, periyot-2 ve kaotik davranıř sergilediđi grlmřtir. Bu ayarlanabilir parametre, denklemde yer alan (μ) katsayıdır. Ayarlanabilen bu parametre, 1 iken devre kaotik davranıř sergilemekte olup, Őekil 4.10(a) ve (b)' de verilmiřtir. Devre, $\mu=5.3$



Şekil 4.8. FPAA-Tabanlı Chua Devre Modeli-IV' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (0.4 V/div), Alttaki İşaret y (0.2 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.946$ İken Elde Edilen Tek Çeker, (d) $\mu=0.898$ İken Elde Edilen Periyot 1 Davranışı.

Tablo 4.6. Kaotik Osilatörlerden Colpitts Osilatörün, Duffing Osilatörün ve Kennedy Osilatörün FPAAs Gerçekleştirim Şemaları.

<p style="text-align: center;">Colpitts Osilatör</p> $\dot{x} = y - f(z)$ $\dot{y} = 1 - x - 0.8\mu y - z \quad (4.8)$ $\dot{z} = y - 0.6$ $f(z) = \begin{cases} -30(z+1), & z < -1 \\ 0, & z > -1 \end{cases}$	
<p style="text-align: center;">Duffing Osilatör</p> $\dot{x} = \mu y$ $\dot{y} = 0.3 \text{Sint} - 0.25y + x - x^3 \quad (4.9)$	
<p style="text-align: center;">Kennedy Osilatör</p> $\dot{x} = \mu y$ $\dot{y} = z \quad (4.10)$ $\dot{z} = -0.6 [x + y + z - \text{sgn}(x)]$	

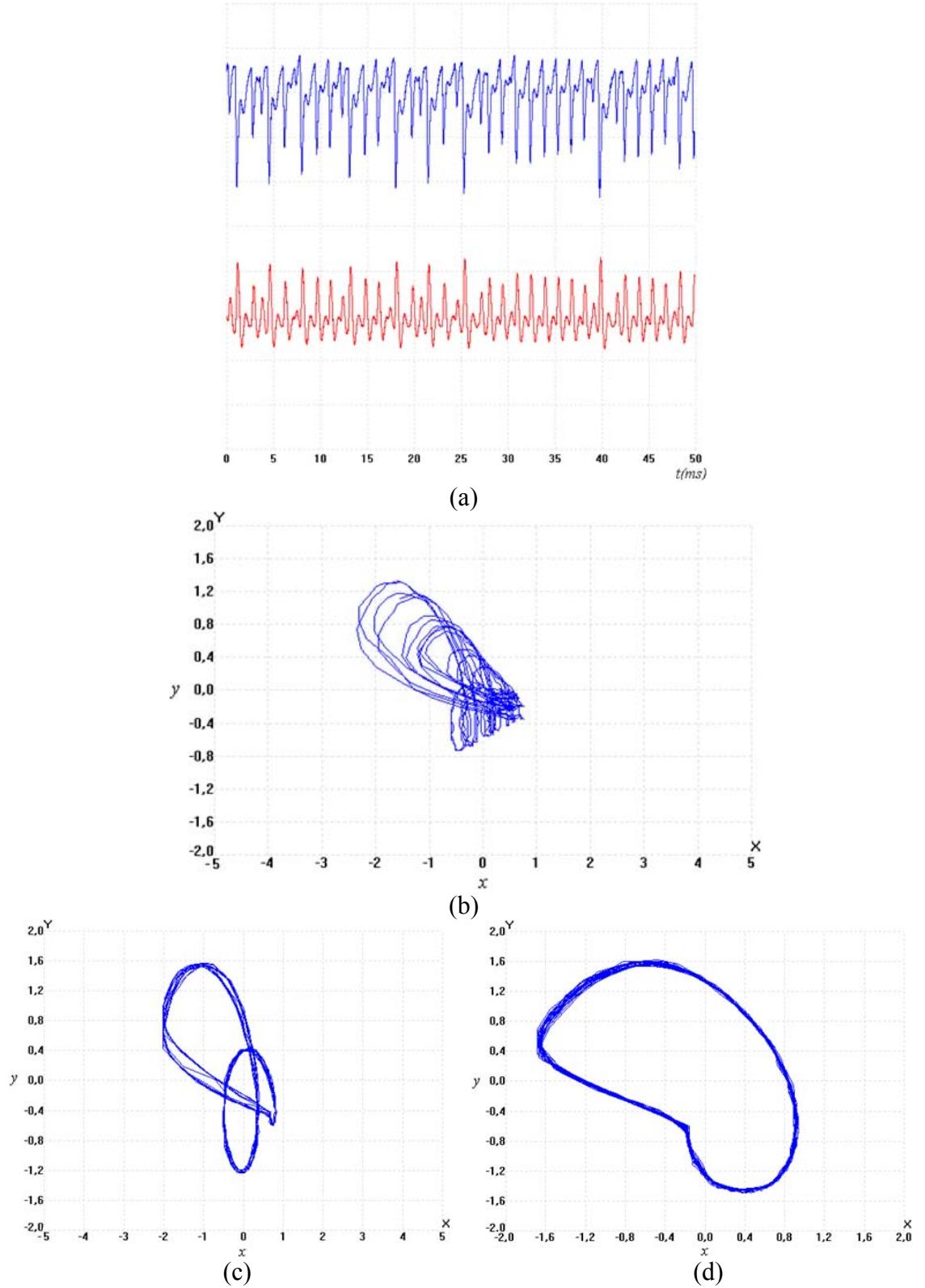


Şekil 4.9. *TRANSFER FUNCTION* Bloкта Elde Edilen $f(z)$ Fonksiyonun Grafiği.

iken periyot-2 davranışını ve $\mu=7$ iken periyot-1 davranışını sergilemektedir ve bu davranışların $(x-y)$ düzlemindeki deneysel sonuçları sırasıyla Şekil 4.10(c) ve (d)' de yer almaktadır. Bu deneysel sonuçların Bölüm 3.3.1.2' de verilen nümerik analiz sonuçları ile benzer olduğu görülmüştür.

4.4.3. FPAA-Tabanlı Duffing Osilatör


Harici bir sinüzoidal kaynağa sahip olan Duffing osilatörü, literatürde yer alan otonom olmayan kaotik sistemlerden birisidir. Denklem (4.9)' a göre FPAA gerçekleştirimi yapılan bu yapının FPAA şeması Tablo 4.6' da verilmiştir. Bu gerçekleştirimde sistem dinamiklerini elde etmek için *SUMFILTER* blokların yanı sıra *TRANSFER FUNCTION*, *SUMDIFF* ve *OSCILLATORSINE* blokları da kullanılmıştır. FPAA, dahili bir AC sinyal kaynağına sahip olduğu için Duffing osilatörün FPAA-tabanlı gerçekleştirimi sırasında harici bir kaynağa gereksinim duyulmaz. FPAA' nın içerisindeki blok *OSCILLATORSINE* bloğu olup, bu tasarımda $(Sint)$ ifadesini gerçekleştirmek için kullanılmıştır. Bu tez çalışmasında *OSCILLATORSINE* blok ile 0.6-10 KHz arasında değişen frekans değerleri ve 1.8 ila 4V arasında değişen genlik değerleri aralığında çalışılmıştır. Duffing osilatör gerçekleştiriminde frekans değeri 1 KHz ve genlik de 2 V seçilmiştir. Osilatörde kullanılan bir diğer blok olan *TRANSFER FUNCTION* blok, (x^3) ifadesini elde etmek için seçilmiştir. Devrede kullanılan *SUMDIFF* blok ise, (y) dinamiğini elde etmek amacıyla 4 adet parametrenin toplama/çıkarma işleminin yapılması için *SUMFILTER* bloğuna ek olarak kullanılmıştır.



Şekil 4.10. FPAAs-Tabanlı Kaotik Colpitts Osilatörden, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (1 V/div), Alttaki İşaret y (1 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=5.3$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=7$ İken Elde Edilen Periyot 1 Davranışı.

Bu osilatör devresinde de programlanabilirliği sağlayan parametre denklemdeki (μ) katsayısıdır. Bu katsayı 1 iken osilatör kaotik davranış sergilerken, 1.75 iken limit çevrim (limit cycle) davranışını sergilemektedir. Bu tasarımdan elde edilen tüm deneysel sonuçlar, Şekil 4.11' de yer almaktadır.

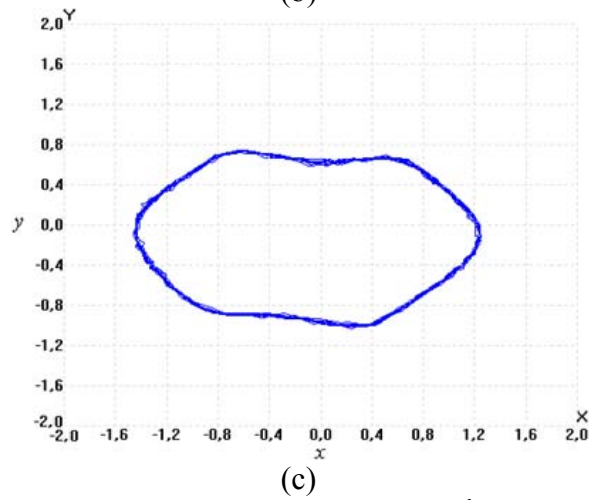
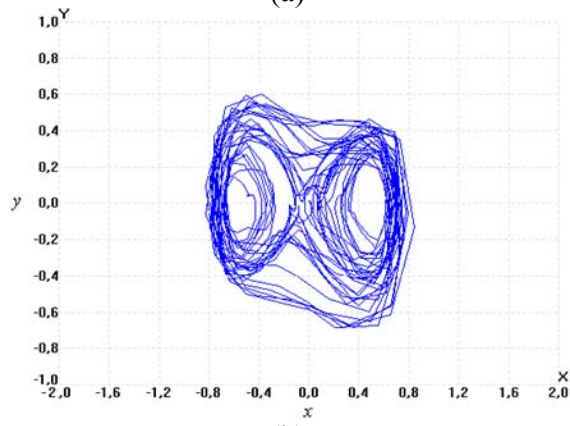
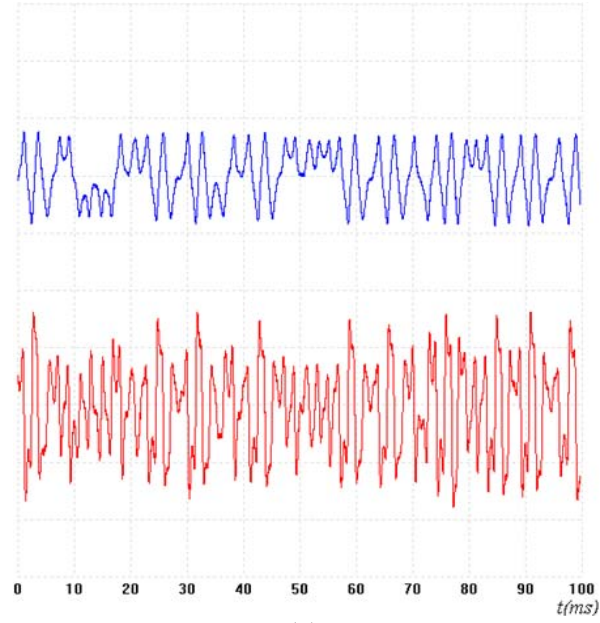
4.4.4. FPAA-Tabanlı Kennedy Osilatör

Kennedy osilatörün FPAA gerçekleştirim şeması Tablo 4.6' da verilmiş olup, denklem takımı Denklem (4.10)' da tanımlanmıştır. Tablodan görüldüğü gibi durum değişkenleri x , y ve z , *SUMFILTER* blokların çıkışlarından, devre kazançları da *SUMFILTER* blok kazançlarından elde edilmiştir. Denklem (4.10)' daki doğrusalsızlığı sağlayan $sgn(x)$ fonksiyonu, *GAIN LIMITER* bloğu kullanılarak elde edilmiştir.  *GAIN LIMITER* bloğu, girişine gelen işareti kuvvetlendirerek sınırlı bir çıkış değeri üretir. Bu blokta sürekli bir giriş ve çıkış vardır. Bu osilatör tasarımında bu bloğun girişine gelen ± 1 V aralığında değişen işarete karşı bir çıkış değeri üretilmiştir. Bu aralıkların dışında gelen işaretler ± 1 V' da sınırlanmıştır.

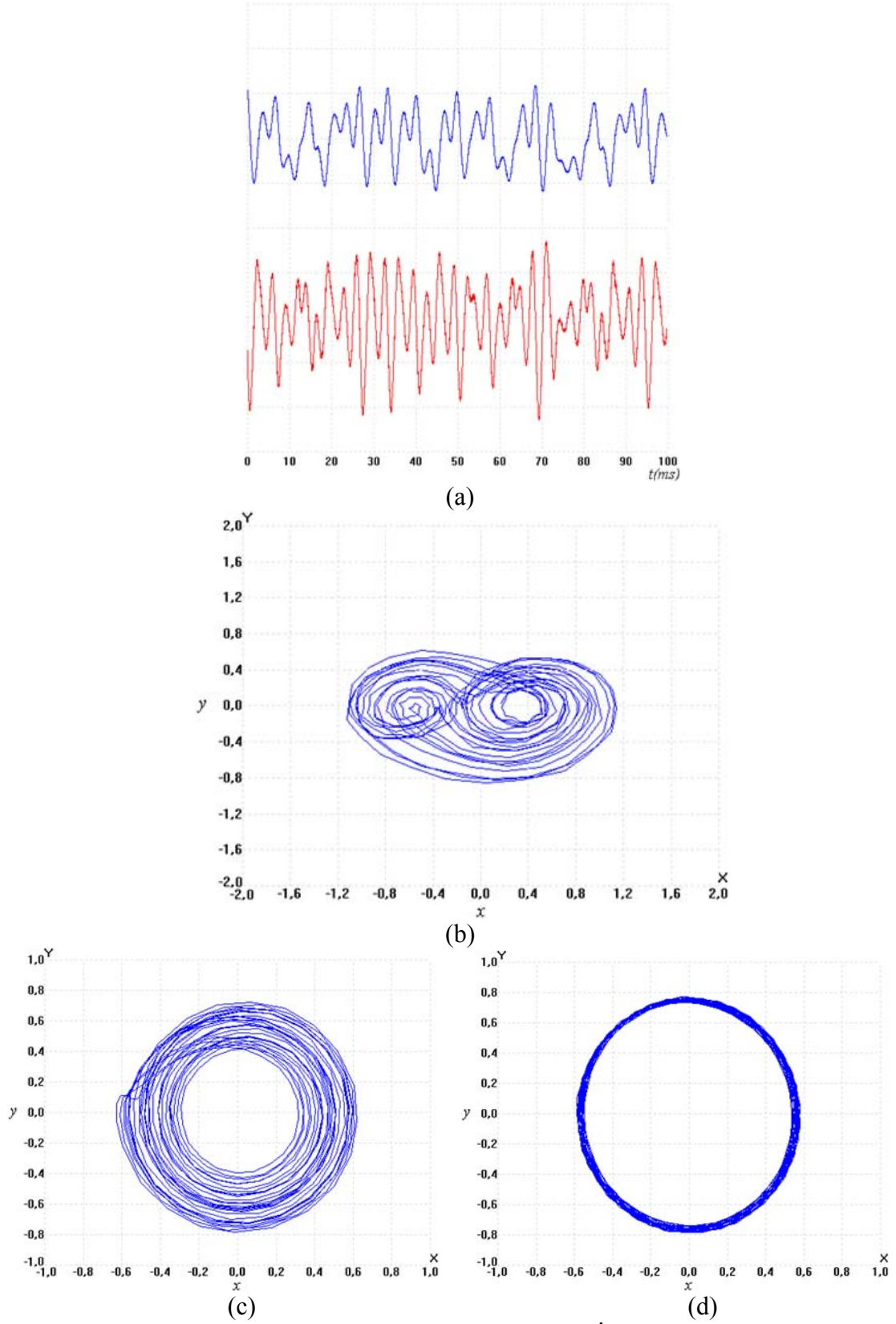
Kennedy osilatörün denklem takımında yer alan programlanabilirliği sağlayan (μ) katsayısı, devrenin periyodik işareten kaotik işarete kadar değişen davranış sergilemesini sağlamaktadır. $\mu=1$ iken, 0.62 iken ve 0.57 iken devrenin sergilediği davranışlar sırasıyla çift çeker, tek çeker ve periyot-1 şeklinde tanımlanmaktadır. Bu deneysel çalışmalardan elde edilen kaotik dinamikler ve çift çeker gösterim Şekil 4.12(a) ve (b)' de, tek çeker ve periyot-1 davranışları Şekil 4.12(c) ve (d)' de verilmiştir.

4.4.5. FPAA-Tabanlı Lorenz Sistemi

Lorenz sistemi yıllardır üzerinde birçok araştırmacının çalıştığı kaotik sistemlerden biridir. Bu sistemin FPAA gerçekleştirimi mümkün olup, Tablo 4.7' de verilmiştir. Bu gerçekleştirimde FPAA' nın ± 2 V saturasyon seviyesine sahip olmasından dolayı Bölüm 3.3.1.5' de Denklem (3.19) ile verilen devre denklemleri ölçeklendirme işlemine tabii tutulmuştur ve ölçeklendirilen devre denklemleri Denklem (4.11)' de verilmiştir.



Şekil 4.11. FPAA-Tabanlı Duffing Osilatörden, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (1 V/div), Altteki İşaret y (0.4 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=1.75$ İken Elde Edilen Limit Çevrim Davranışı.



Şekil 4.12. FPAA-Tabanlı Kennedy Osilatörden, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (1 V/div), Altteki İşaret y (0.4 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.62$ İken Elde Edilen Tek Çeker, (d) $\mu=0.57$ İken Elde Edilen Periyot 1 Davranışı.

Tablo 4.7. Kaotik Osilatörlerden Lorenz Sisteminin, MLC Osilatörünün ve Rössler Sisteminin FPAA Gerçekleştirim Şemaları.

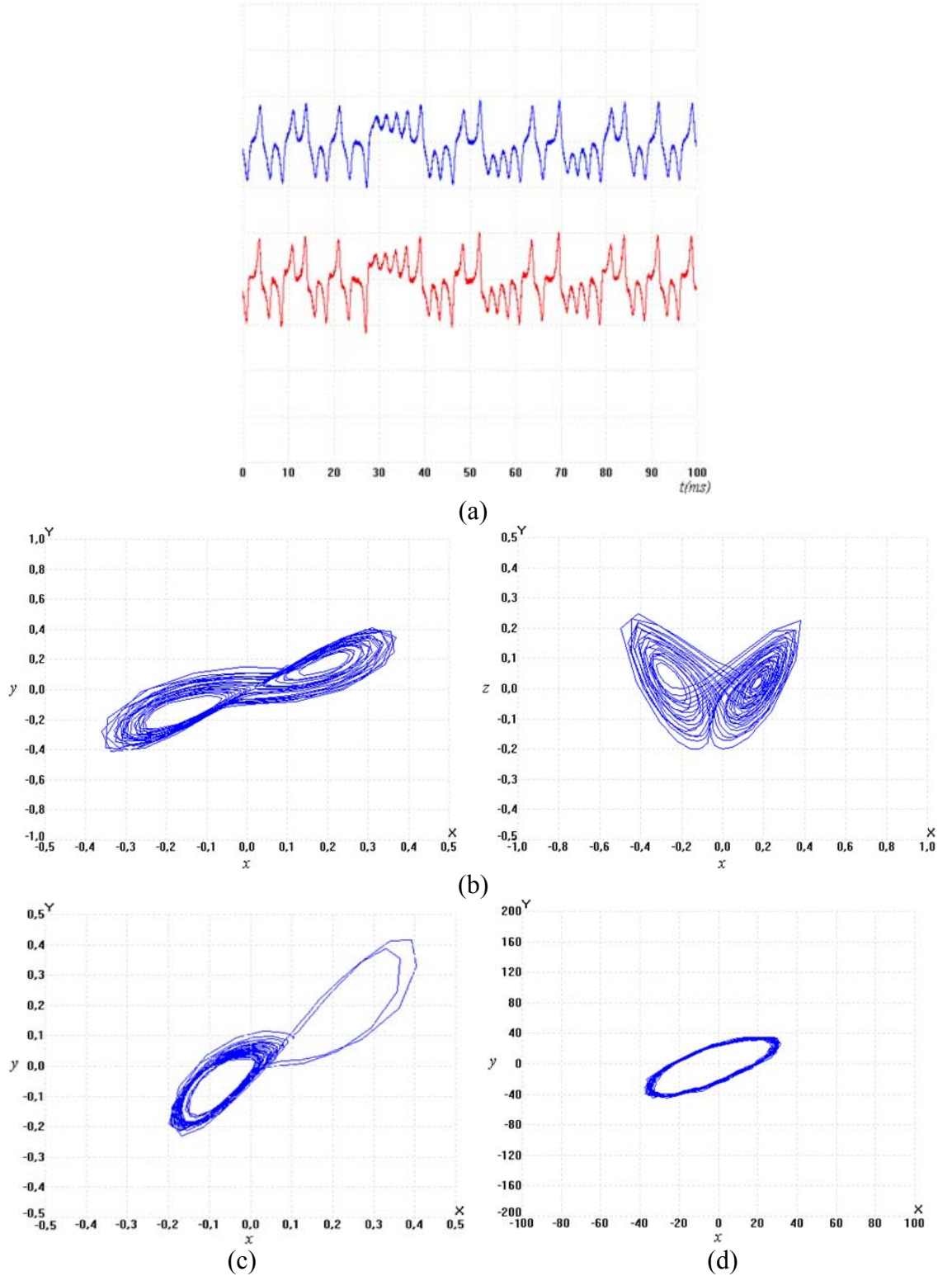
<p style="text-align: center;">Lorenz Sistemi</p> $\begin{aligned} \dot{x} &= -10x + 12.77y \\ \dot{y} &= -y - 35.22xz + 21.91\mu x \\ \dot{z} &= 2.67z + 9.2xy \end{aligned} \quad (4.11)$	
<p style="text-align: center;">MLC Osilatör</p> $\begin{aligned} \dot{x} &= y - f(x) \\ \dot{y} &= -y - x - \mu(vy - F \sin(\omega t)) \\ f(x) &= bx + 0.5(a-b)(x+1 - x-1) \end{aligned} \quad (4.12)$	
<p style="text-align: center;">Rössler Osilatör</p> $\begin{aligned} \dot{x} &= -y - 2\mu z \\ \dot{y} &= x + 0.2y \\ \dot{z} &= 0.01 + 10xz - 5.7z \end{aligned} \quad (4.13)$	

Lorenz sisteminin FPAA gerekleřtiriminde *SUMFILTER* bloklar, diđer gerekleřtirimlerde olduđu gibi sistem deđiřkenlerini elde etmek iin kullanılmıřtır. Devrede bu blođun dıřında *MULTIPLIER* bloklar ve *GAIN* bloklar kullanılmıřtır. Bu bloklardan *MULTIPLIER* bloklar, Denklem (4.11)' de yer alan $(x.z)$ ve $(x.y)$ ifadelerini elde etmek amacıyla kullanılmıřlardır. *SUMFILTER* bloklarında kazanç parametresi olarak en fazla 20 deđeri verilmektedir. Bu gerekleřtirimde ise 35.22 ve 21.91 katsayıları bulunmaktadır. Bu katsayıları elde etmek iin *SUMFILTER* kazanç blokları ile birlikte *GAININV* blođu kullanılmıřtır. Diđer bir *GAININV* blođu da z dinamiđini elde etmek iin *SUMFILTER* blođa ek olarak kullanılmıřtır. Bu gerekleřtirimde de devreden elde edilen iřaretin periyodik iřarettten kaotik iřarete deđiřimini sađlayan ayarlanabilir bir katsayı mevcuttur. Bu katsayı, denklem takımında yer alan (μ) katsayısıdır. Bu katsayı 1 iken devre kaotik davranıř sergilemektedir ve sergilenen bu davranıřlar, Őekil 4.13(a) ve (b)' de hem zaman domeninde hem de $(x-y)$ ve $(x-z)$ dzleminde verilmiřtir. Őekil 4.13(c) ve (d)' de ise devrenin tek eker ve periyodik davranıřları yer almaktadır. FPAA-tabanlı bu gerekleřtirimden elde edilen sonular, Blm 3.3.1.5' de verilen hem SIMULINK sonularıyla hem de ayrıık elemanlarla yapılan deneysel alıřma sonuları ile uyurmaktadır.

4.4.6. FPAA-Tabanlı MLC Osilatr

Otonom olmayan kaotik sistemler arasında yer alan MLC osilatr, en basit kaotik sistemlerden birisidir. Harici AC kaynađın frekans ve genliđini ieren sistem parametrelerine bađlı olarak kaotik davranıř sergileyen bu osilatr, programlama ve yeniden yapılandırılabilir gerekleřtirimler iin uygun yapılardan birisidir.

FPAA, dahili bir AC sinyal kaynađına sahip olduđu iin MLC osilatrn FPAA' de gerekleřtirimi sırasında harici bir kaynađa gereksinim duyulmaz ve MLC osilatrnn kaos modunda alıřmasında kullanılan genlik ve frekans parametreleri programlanabilir formda kolayca deđiřtirilebilir. Bu osilatrn durum denklemleri, Denklem (4.12)' de verilmiřtir [37]. Denklemde yer alan (x) ve (y) sistemin durum deđiřkenleri olurken v , F ve w , sistem parametreleridir ve bu parametreler, $v=0.0935$, $F=0.15$ ve $w=0.6$ ' dir. Denklemde dođrusal olmayan fonksiyon $f(x)$ ile ifade edilmiř olup, $a=-1.02$ ve $b=-0.55$ ' dir.



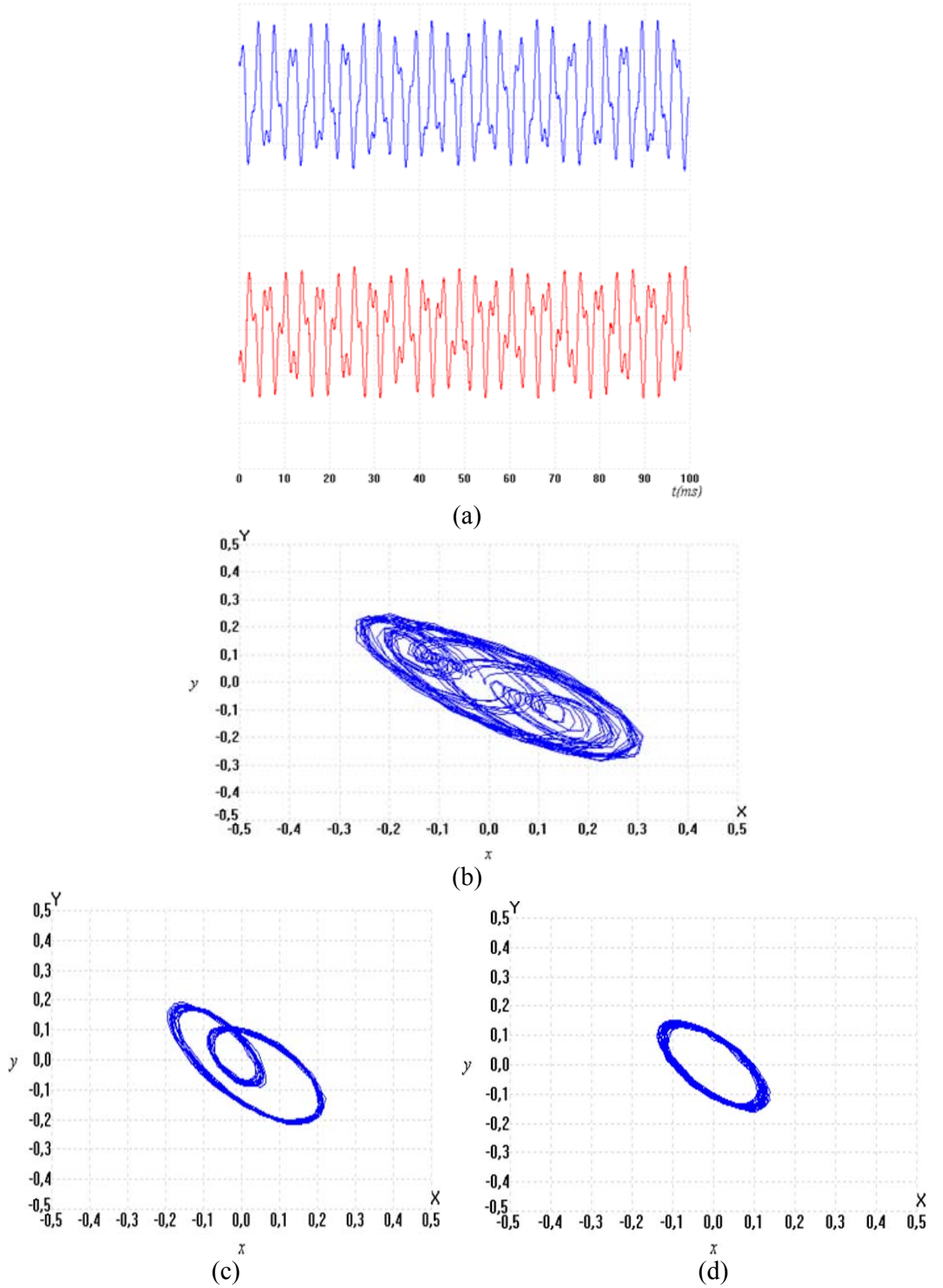
Şekil 4.13. FPAA-Tabanlı Lorenz Sisteminden, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (0.4 V/div), Altteki İşaret y (0.4 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ ve $(x-z)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.828$ İken Elde Edilen Tek Çeker, (d) $\mu=0.825$ İken Elde Edilen Periyot 1 Davranışı.

MLC osilatörün FPAA gerçekleştirim şeması Tablo 4.7' de verilmiştir. Bu gerçekleştirimde x ve y durum değişkenleri, *SUMFILTER* blokların çıkışlarından elde edilmektedir. Devre kazançları ise *SUMFILTER* ve *SUMDIFF* blokların kazançları ile gerçekleştirilmektedir. Denklem (4.12)' deki doğrusal olmayan fonksiyonda yer alan $(|x+1|-|x-1|)$ ifadesini gerçekleştirmek için *TRANSFER FUNCTION* blok kullanılmıştır. Bu bloğa yüklenen (.csv) uzantılı dosya hazırlanırken mutlak değerli ifadeler, ExcelTM, de (*abs*) komutu ile gerçekleştirilmiştir ve Şekil 4.3' de ExcelTM, de hazırlanan dosyadan elde edilen grafik verilmiştir. Bu FPAA-tabanlı gerçekleştirimde kullanılan *OSCILLATORSINE* bloğu ise, sinüs dalga işaret kaynağını elde etmek amacıyla kullanılmıştır. Devrenin denklem takımında yer alan (μ) parametresi, programlanabilirliği sağlayan parametre olarak tanımlanabilir. Bu parametre, programlanabilir formda değiştirilebilir ve bu değişime göre devre periyodik davranıştan kaotik davranışa kadar farklı davranışlar sergileyebilmektedir. (μ) parametresi 1 iken devre kaotik davranış sergilemekte olup, 1.3 değerinde periyot-2 davranışını, 1.35 iken de periyot-1 davranışını sergilemektedir. MLC devresinin FPAA-tabanlı gerçekleştiriminden elde edilen tüm deneysel sonuçlar, Şekil 4.14' de yer almaktadır. Bu sonuçların Bölüm 3.3.1.6' da verilen nümerik analiz sonuçları ve donanım gerçekleştirim sonuçları ile uyum içinde olduğu görülmüştür.

4.4.7. FPAA-Tabanlı Rössler Sistemi

Basit bir denklem takımına sahip olan Rössler sisteminin durum denklemleri ve nümerik sonuçları Bölüm 3.3.1.7' de verilmiştir. FPAA' nın $\pm 2V$ saturasyon voltajından dolayı bu sistem, ölçeklendirme işlemine tabii tutulmuştur. Bu sistemin FPAA gerçekleştiriminde kullanılan durum denklemleri Tablo 4.7' de Denklem (4.13) ile verilmiştir. Bu sistemin FPAA' da gerçekleştirimi de yine Tablo 4.7' de yer almaktadır.

Rössler sisteminin FPAA gerçekleştiriminde (x) , (y) ve (z) durum değişkenleri diğer gerçekleştirimlerde olduğu gibi *SUMFILTER* blokların çıkışlarından elde edilmiştir. Devre denklem takımında yer alan $(x.z)$ çarpımını elde etmek için *MULTIPLIER* bloğu kullanılmıştır. Denklemde kullanılan 0.01 sabit değeri de *VOLTAGE REFERENCE* blok ile iki adet *GAININV* blok kullanılarak sağlanmıştır.



Şekil 4.14. FPAA-Tabanlı MLC Osilatörden, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (0.2 V/div), Alttaki İşaret y (0.2 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=1.3$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=1.35$ İken Elde Edilen Periyot 1 Davranışı.

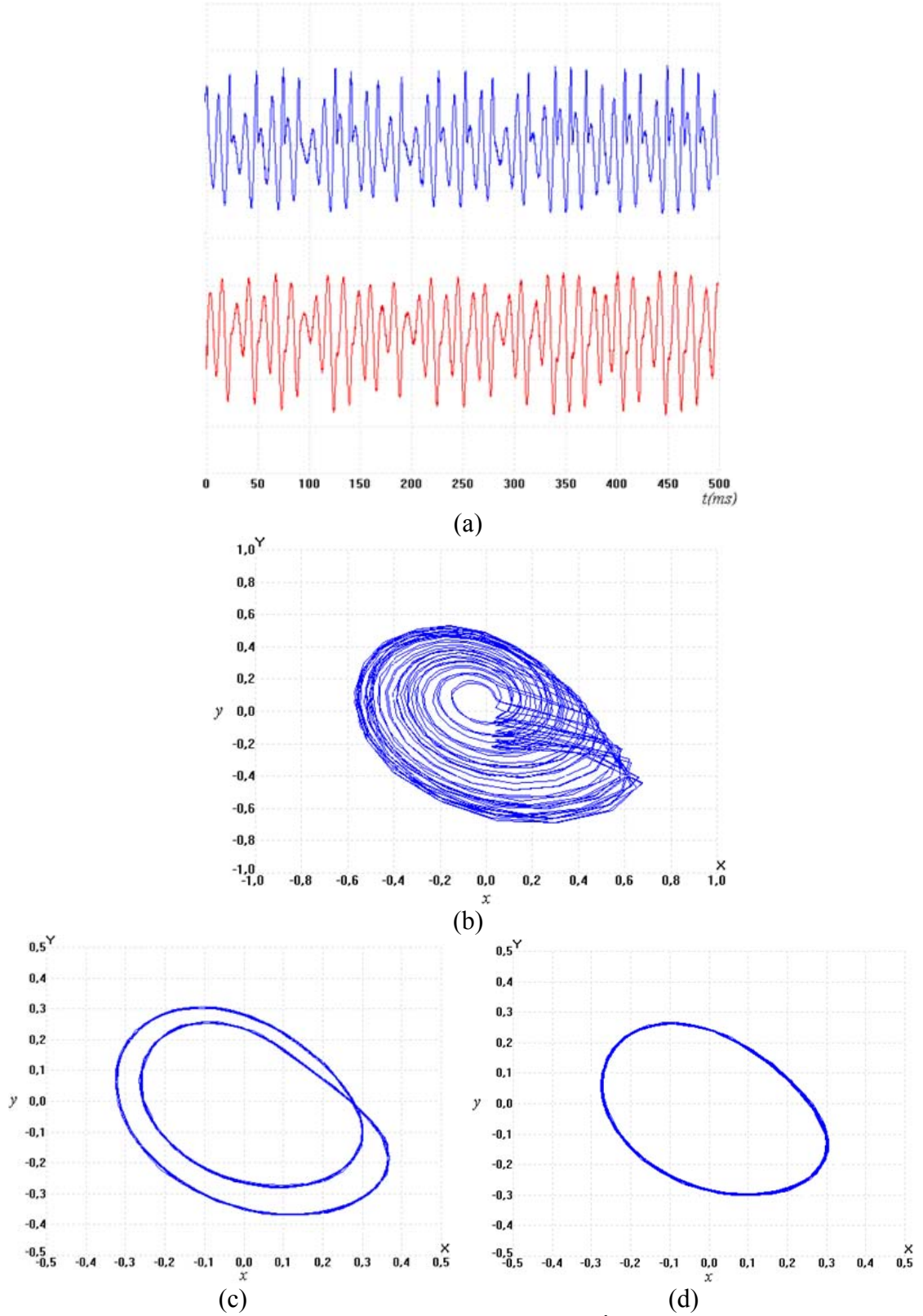
Bu gerekleřtirimde programlanabilirlięi (μ) parametresi saęlamaktadır. $\mu=1$ iken FPAA-tabanlı Rössler sistemi kendi orijinal davranıřını sergilemektedir. $\mu=3$ iken Rössler sisteminin davranıřı periyot-2' ye dőnüşürken $\mu=3.5$ iken periyot-1 davranıřı elde edilmektedir. (μ) parametresinin deęiřimine göre elde edilen deneysel sonular Őekil 4.15' de yer almaktadır.

4.4.8. FPAA-Tabanlı Sprot Systemlerinin Gerekleřtirimini

Bu kısımda Sprot sistemlerinin FPAA-tabanlı gerekleřtirimleri incelenmiřtir. Sprot sisteminin denklem takımı, Bölüm 3.3.1.8' de Denklem (3.26-3.31) ile verilmiř olup SIMULINK™ modellemesi yapılmıřtır. Bu sistem de Chua osilatöründe olduęu gibi farklı doęrusal olmayan fonksiyonlar kullanılarak ayrıık elemanlarla gerekleřtirilebilir. Bunlardan her birinin gerekleřtiriminin ok fazla zaman alması, kullanılacak elemanların tam deęerlerinin bulunmasının zor olması ve devrede kullanılacak Op-Amp' ların temininin sıkıntılı olması gibi dezavantajlardan dolayı analog devrelerin deneysel ortamda gerekleřtirimini bayaęı zahmetlidir. FPAA' lar programlanabilir ve yeniden yapılandırılabilir özellikleri sayesinde bu dezavantajları ortadan kaldırmaktadır. Bu kısımda farklı doęrusal olmayan fonksiyonlara sahip Sprot sisteminin FPAA-tabanlı gerekleřtirimleri incelenmiřtir [35, 36]. Bu sistem FPAA' da gerekleřtirilmeden önce Bölüm 3.3.1.8' de yapılan nümerik analiz sonularına göre öleklendirme iřlemine tabii tutulmuřtur. Öleklendirilmiř sistem denklemleri ve FPAA gerekleřtirim Őemaları, Tablo 4.8 ve 4.9' da verilmiřtir.

4.4.8.1. FPAA-Tabanlı Sprot System Modeli-I

Sprot sistem modeli-I, Denklem (4.14)' de yer alan matematiksel modele göre FPAA-tabanlı tasarlanmıřtır ve tasarım detayları Tablo 4.8' de gösterilmiřtir. Burada *SUMFILTER* blokları u , v ve w sistem deęiřkenlerinin elde edilmesini saęlamaktadır. Sistem kazançları da bu blokların kazançları tarafından saęlanır. Gerekleřtirimde kullanılan bir dięer blok, *VOLTAGE REFERENCE* blok olup, doęrusal olmayan fonksiyondaki sabit deęeri elde etmede kullanılmıřtır. Devre Őemasındaki dięer bloklar olan *RECTIFIERHALF* ve *SUMDIFF* blokları doęrusal olmayan fonksiyonda kalan dięer iřlemleri gerekleřtirmişlerdir.

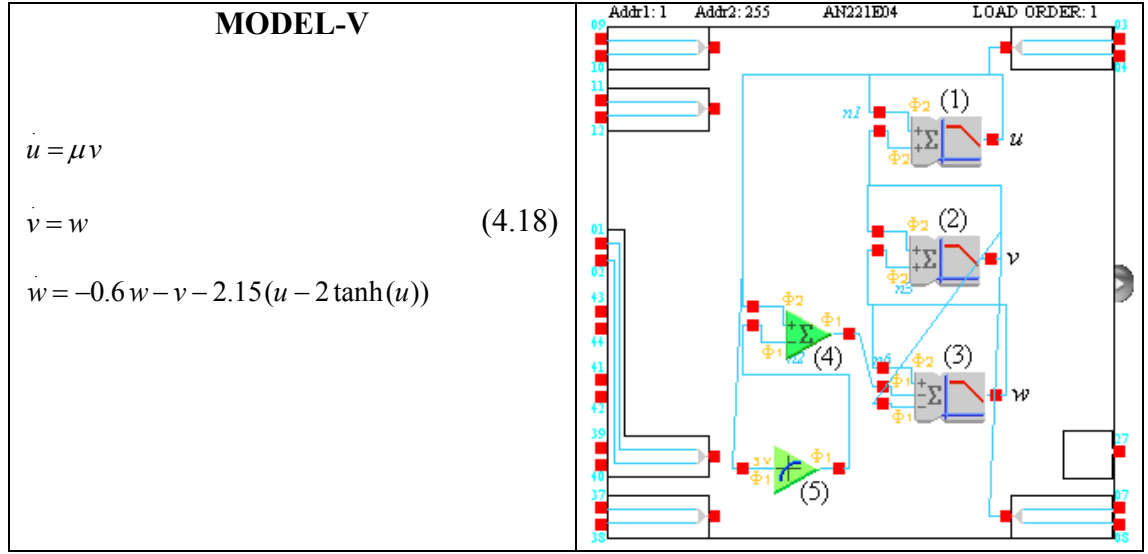


Şekil 4.15. FPAA-Tabanlı Rössler Osilatörden, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret x (0.4 V/div), Alttaki İşaret y (0.4 V/div), (b) $\mu=1$ İken Elde Edilen $(x-y)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=3$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=3.5$ İken Elde Edilen Periyot 1 Davranışı.

Tablo 4.8. Sprott Sistemlerinden Model-I, Model-II, Model-III ve Model-IV' ün FPAA Gerçekleştirim Şemaları.

<p style="text-align: center;">MODEL-I</p> $\dot{u} = 0.8 \mu v$ $\dot{v} = 0.875 w$ $\dot{w} = -0.6 w - 1.143 v + 1.143 u - 2.285$ <p style="text-align: right;">(4.14)</p>	
<p style="text-align: center;">MODEL-II</p> $\dot{u} = v$ $\dot{v} = w$ $\dot{w} = -0.6 w - v + 2 \operatorname{sgn}(u) - 1.2 \mu u$ <p style="text-align: right;">(4.15)</p>	
<p style="text-align: center;">MODEL-III</p> $\dot{u} = \mu v$ $\dot{v} = w$ $\dot{w} = -0.6 w - v - 0.58(1 - u^2)$ <p style="text-align: right;">(4.16)</p>	
<p style="text-align: center;">MODEL-IV</p> $\dot{u} = \mu v$ $\dot{v} = w$ $\dot{w} = -0.6 w - v - 0.9 u (0.47 u^2 - 1)$ <p style="text-align: right;">(4.17)</p>	

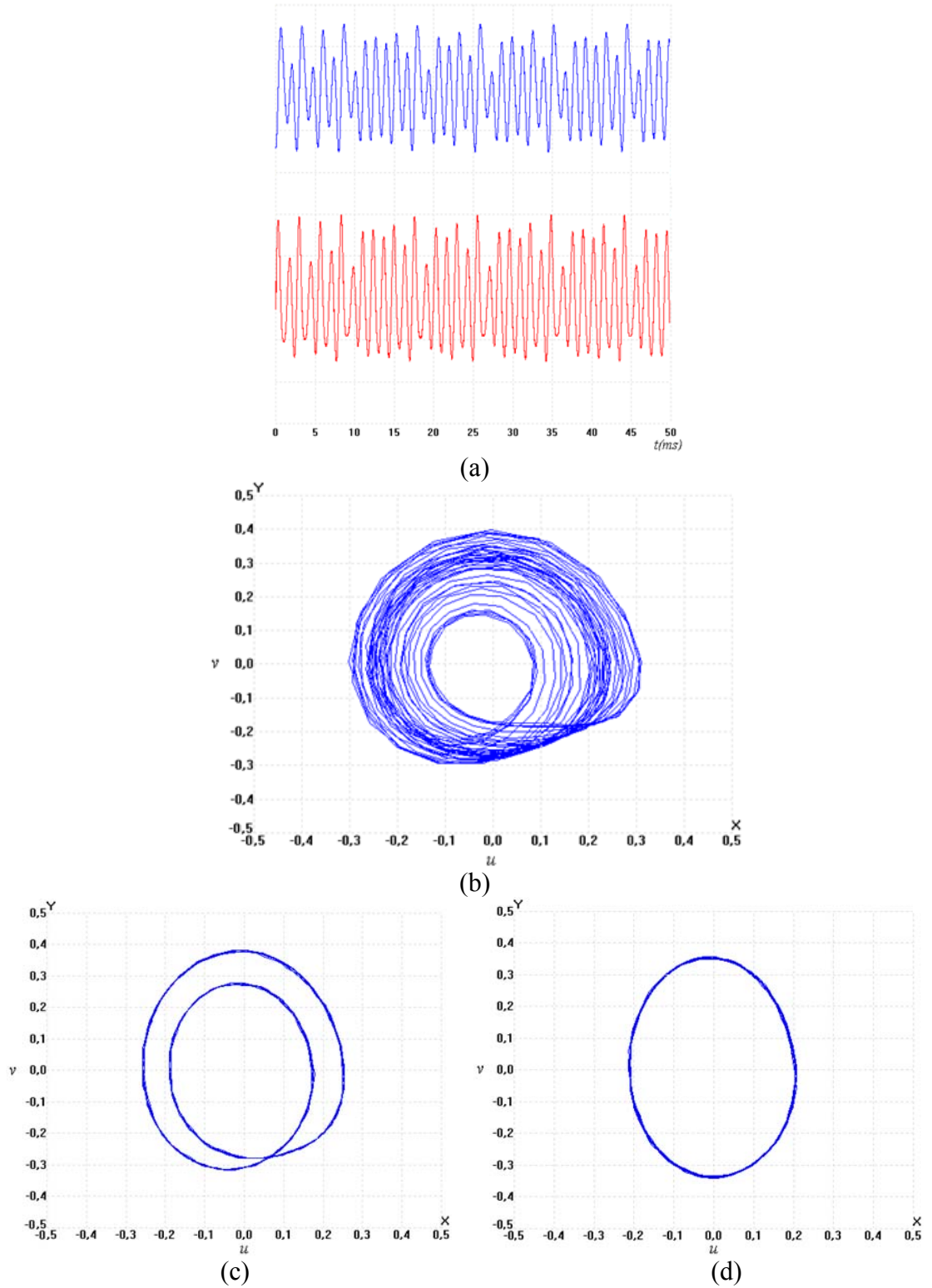
Tablo 4.9. Sprot Sistemlerinden Model-V' in FPAA Gerçekleştirim Şeması.



Bu gerçekleştirim için programlanabilir parametre (μ) parametresidir. (μ) parametresinin değişimi ile sistem değişik davranışlar sergilemektedir. $\mu=1$ iken sistem kaotik davranış sergilemekte olup Şekil 4.16(a)' da kaotik dinamikler, Şekil 4.16(b)' de ($u-v$) düzlemindeki çeker gösterimleri verilmiştir. $\mu=0.875$ iken sistem davranışı periyot-2 olurken $\mu=0.75$ iken davranış periyot-1' e dönüşmektedir ve bu sonuçlar Şekil 4.16(c) ve (d)' de yer almaktadır.

4.4.8.2. FPAA-Tabanlı Sprot Sistem Modeli-II

FPAA' nın yeniden yapılandırılabilirlik özelliği sayesinde tasarımı yapılan sistem FPAA üzerine yüklendikten ve deneysel sonuçlar alındıktan sonra başka bir sistem FPAA' ya gönderilebilir ve çalışmalar aksamadan devam ettirilebilir. Bu kapsamda FPAA-tabanlı Sprot sistem modeli-I' den sonra Sprot sistem modeli-II ve diğerlerinin FPAA gerçekleştirimi üzerine çalışmalar devam ettirilmiştir. İncelenen FPAA-tabanlı Sprot sistemlerinden ikincisi, parçalı doğrusal fonksiyon kullanılarak gerçekleştirilmiş olup matematiksel modeli Denklem (4.15) ile verilmiştir. Bu gerçekleştirimde *SUMFILTER*, *GAIN LIMITER* ve *SUMDIFF* blokları kullanılmıştır. *GAIN LIMITER* blok ile $[sgn(u)]$ terimini içeren parçalı doğrusal fonksiyon gerçekleştirilmiştir. *SUMDIFF* bloğu ise denklemde (w) dinamiği elde etmek için 4 adet toplama/çıkarma işleminde *SUMFILTER* bloğa ek olarak kullanılmıştır.



Şekil 4.16. FPAA-Tabanlı Sprott Sistem Modeli-I' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret u (0.2 V/div), Alttağı İşaret v (0.2V/div), (b) $\mu=1$ İken Elde Edilen $(u-v)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.875$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=0.75$ İken Elde Edilen Periyot 1 Davranışı.

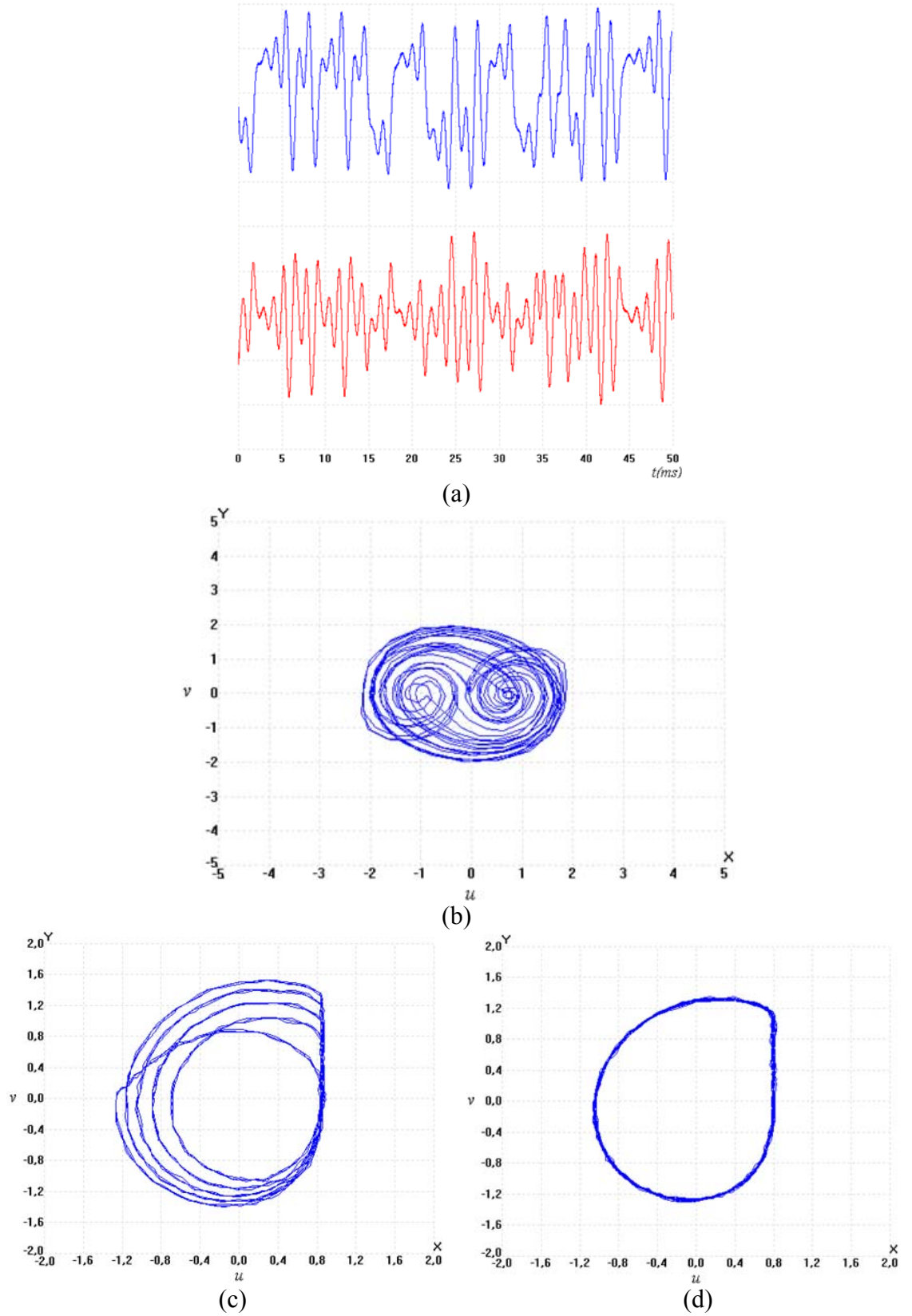
Sprott Sistem Modeli-II' nin Denklem (4.15)' de yer alan denklem takımındaki (μ), programlanabilirliği sağlayan parametre olup, bu parametrenin deęişimi ile sistem deęişik dinamikler sergileyebilmektedir. Sistem $\mu=1$ iken kaotik davranış sergilerken, $\mu=0.675$ iken periyot-5 ve $\mu=0.65$ iken periyot-1 davranışlarını sergilemektedir. Sistemin sergilediđi bu davranışların deneysel sonuçları, Şekil 4.17' de verilmiştir.

4.4.8.3. FPAA-Tabanlı Sprott Sistem Modeli-III

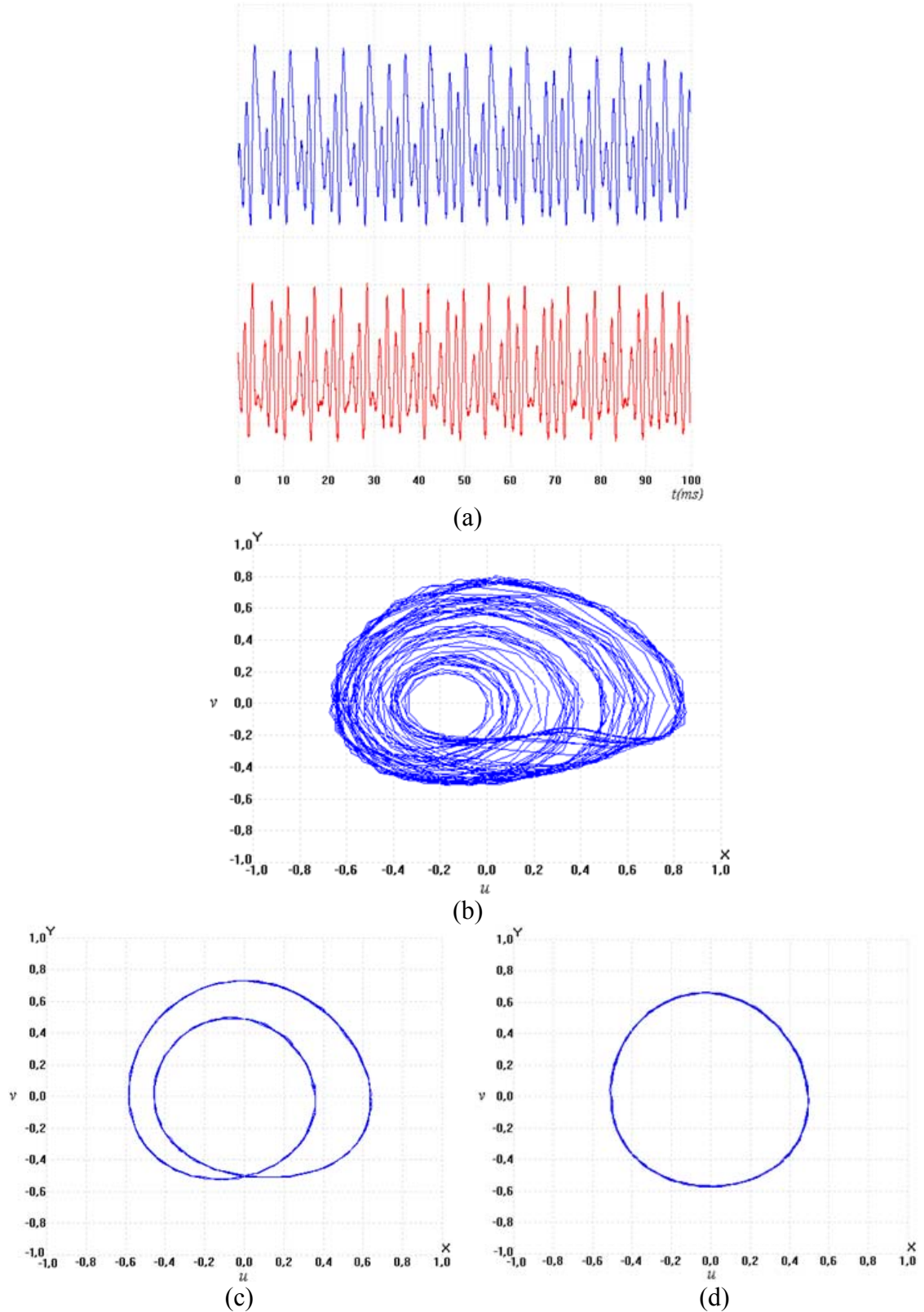
Sprott sistem modeli-III, Tablo 4.8' deki Denklem (4.16) ile tanımlanan matematiksel modele göre FPAA' da gerçekleştirilmiştir. Bu gerçekleştirimde *SUMFILTER* blokların yanı sıra *MULTIPLIER*, *SUMDIFF* ve *VOLTAGE REFERENCE* blokları kullanılmıştır. Denklemde yer alan $[0.58(1-u^2)]$ ifadesi sisteme ait doğrusal olmayan fonksiyon olup bu fonksiyon *MULTIPLIER*, *SUMDIFF* ve *VOLTAGE REFERENCE* blokları ile gerçekleştirilmiştir. Sistemde programlanabilirliği sağlayan parametre (μ)' dür. (μ)' nün deęeri 1 olduğunda sistem davranışı kaotik davranış olmaktadır. (μ)' nün deęeri 0.88 iken sistem periyot-2 davranışını sergilerken 0.8 olduğunda periyodik işarete dönüşmektedir. (μ)' nün deęişimine göre elde edilen sistem davranışları Şekil 4.18' de yer almaktadır.

4.4.8.4. FPAA-Tabanlı Sprott Sistem Modeli-IV

Denklem (4.17) ile matematiksel modeli verilen Sprott sistem modeli-IV' ün FPAA gerçekleştirimi, kübik benzeri doğrusal olmayan fonksiyon kullanılarak gerçekleştirilmiştir. Bir önceki gerçekleştirimde de kullanılan *MULTIPLIER*, *SUMDIFF* ve *VOLTAGE REFERENCE* blokları, bu gerçekleştirimde de kullanılmış olup, bu modelde $[0.9u(0.47u^2-1)]$ ifadesine sahip kübik benzeri doğrusal olmayan fonksiyonu gerçekleştirmişlerdir. Bu gerçekleştirimde önceki gerçekleştirimde kullanılan bloklara göre (u^3) ifadesini elde etmek amacıyla fazladan bir *MULTIPLIER* blok daha kullanılmıştır. Programlanabilir yapıya sahip bu sistem gerçekleştirilirken sistem davranışı periyodiklikten kaotikliğe doğru deęiştirdiđi görülmüştür. Bu deęişikliđi sağlayan parametre (μ) katsayıdır. $\mu=1$ iken sistem davranışı kaotik davranış olup



Şekil 4.17. FPAA-Tabanlı Sprott Sistem Modeli-II' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret u (1 V/div), Altındaki İşaret v (1V/div), (b) $\mu=1$ İken Elde Edilen $(u-v)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.675$ İken Elde Edilen Periyot 5 Davranışı, (d) $\mu=0.65$ İken Elde Edilen Periyot 1 Davranışı.



Şekil 4.18. FPAА-Tabanlı Sprot Sistem Modeli-III' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret u (0.4 V/div), Altta İşaret v (0.4V/div), (b) $\mu=1$ İken Elde Edilen $(u-v)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.88$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=0.8$ İken Elde Edilen Periyot 1 Davranışı.

zaman domenindeki kaotik davranışlar ve $(u-v)$ düzlemindeki çeker gösterimleri Şekil 4.19 (a) ve (b)' de yer almaktadır. $\mu=0.6$ ve 0.5 değerlerinde iken sistem davranışı sırasıyla periyot-2 ve periyot-1 şeklindedir ve bu değerlerdeki sistem davranışı Şekil 4.19(c) ve (d)' de gösterilmiştir.

4.4.8.5. FPAA-Tabanlı Sprot System Modeli-V

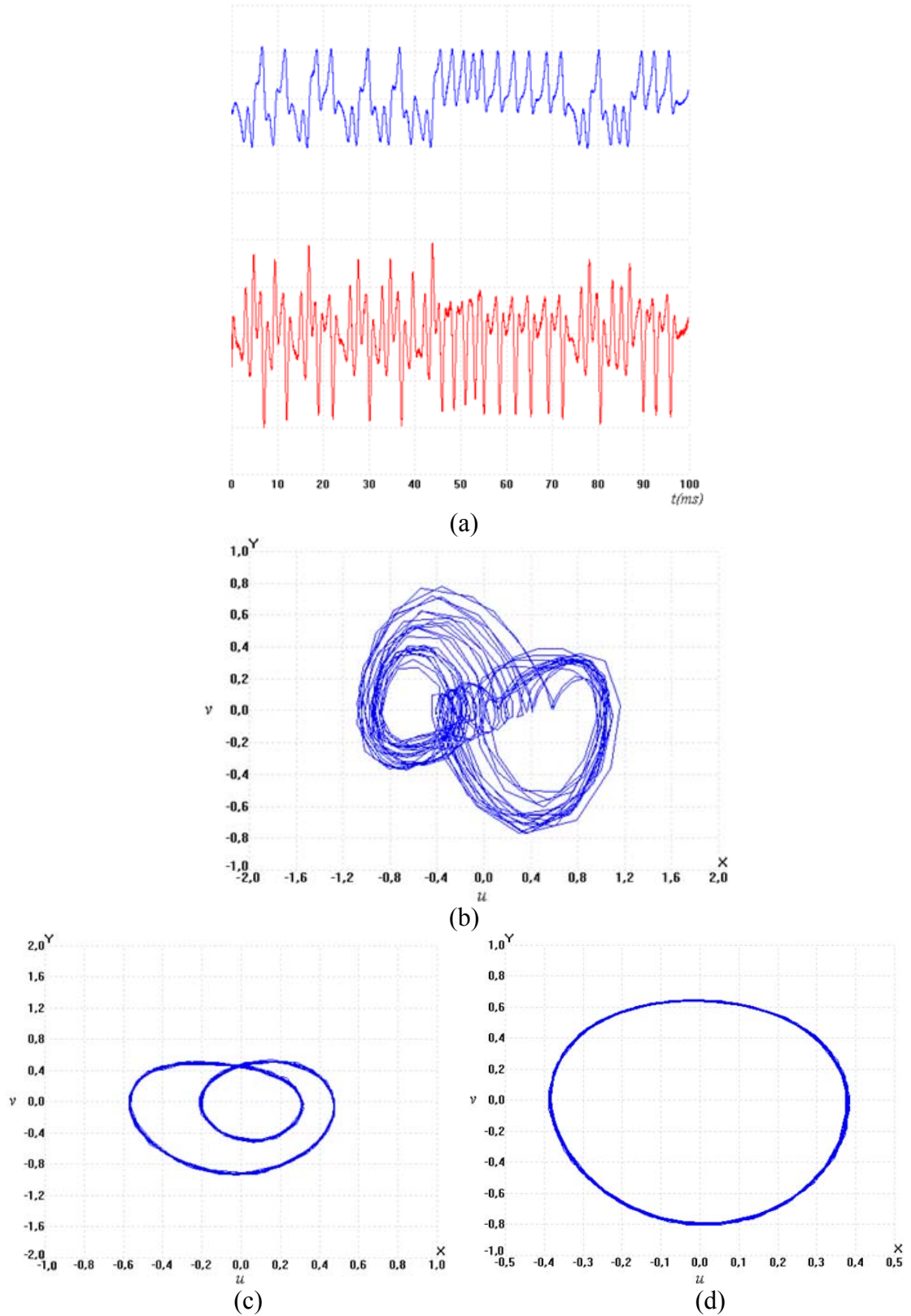
Sprot sistem modeli-V' in FPAA gerçekleştirim şeması Tablo 4.9' da verilmiş olup, Denklem (4.18) ile verilen matematiksel modele göre gerçekleştirilmiştir. Bu gerçekleştirim için doğrusal olmayan fonksiyon olarak trigonometrik fonksiyon kullanılmıştır. Bu modelin gerçekleştirilmesinde *SUMFILTER* bloğun yanında *TRANSFER FUNCTION* ve *SUMDIFF* blokları da kullanılmıştır. Önceki gerçekleştirimlerde olduğu gibi bu gerçekleştirimde de *SUMFILTER* bloğu sistem dinamiklerini elde etmede kullanılmıştır. Gerçekleştirimde kullanılan diğer bir blok *TRANSFER FUNCTION* bloğu $[\tanh(u)]$ terimini içeren trigonometrik fonksiyonu gerçekleştirmek için kullanılmıştır. *SUMDIFF* bloğu ise (w) dinamiğini elde etmek amacıyla dört adet parametrenin toplama/çıkarma işleminin yapılması için *SUMFILTER* bloğuna ek olarak kullanılmıştır. Bu gerçekleştirim için ayarlanabilir parametre olarak kullanılan (μ) parametresi 1 iken sistem davranışı kaotiktir. Sistem davranışı $\mu=0.74$ iken periyot-2, $\mu=0.7$ iken periyodiktir. (μ) ' nün değişimine göre değişen sistem davranışlarının deneysel sonuçları Şekil 4.20' de verilmiştir.

4.4.9. HSA-Tabanlı Kaotik Yapıların FPAA' da Gerçekleştirimi

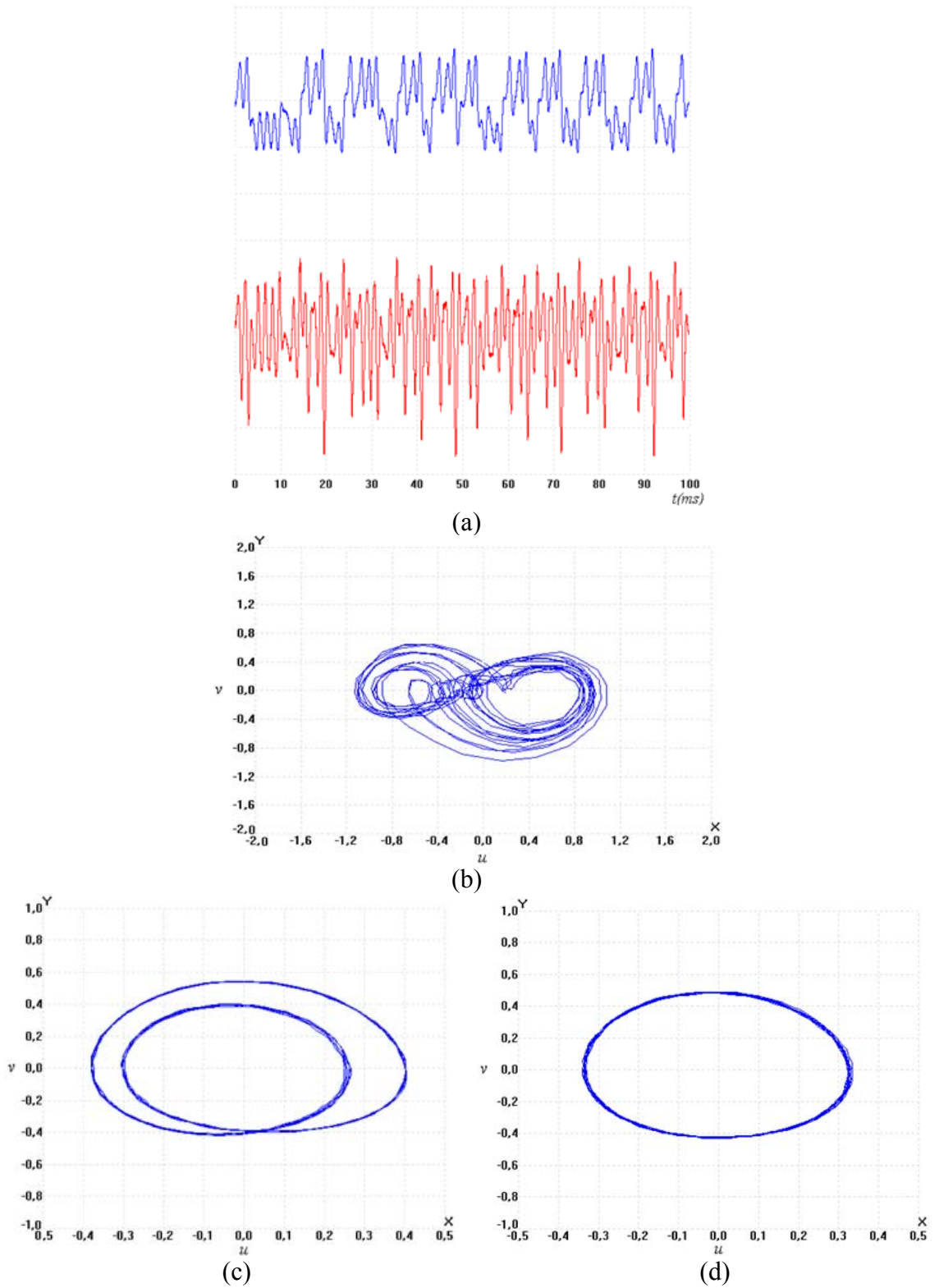
Bu kısımda hücrel sinir ağı (HSA)-tabanlı Chua ve MLC devrelerinin FPAA' da tasarımı üzerinde durulmuştur.

4.4.9.1. HSA-Tabanlı Chua Devresinin FPAA' da Gerçekleştirimi

HSA-tabanlı devrelerin programlanabilir formda FPAA gerçekleştirilebileceğini göstermek amacıyla bu kısımda HSA-tabanlı Chua devresi incelenmiştir. HSA-tabanlı Chua devresinin FPAA' da gerçekleştirimi yapılmadan önce Bölüm 3.3.1.9' da verilen nümerik analiz sonuçlarına bakıldığında devre denklemlerinin ölçeklendirilmesi gerektiği görülmüştür. Bu devre yapısı Caponetto ve ekibi tarafından da incelenmiş [32]



Şekil 4.19. FPAA-Tabanlı Sprott Sistem Modeli-IV' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret u (1 V/div), Alttağı İşaret v (0.4V/div), (b) $\mu=1$ İken Elde Edilen $(u-v)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.6$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=0.5$ İken Elde Edilen Periyot 1 Davranışı.



Şekil 4.20. FPAA-Tabanlı Sprott Sistem Modeli- V' den, (a) $\mu=1$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret u (1 V/div), Alttaki İşaret v (0.4V/div), (b) $\mu=1$ İken Elde Edilen $(u-v)$ Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=0.74$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=0.7$ İken Elde Edilen Periyot 1 Davranışı.

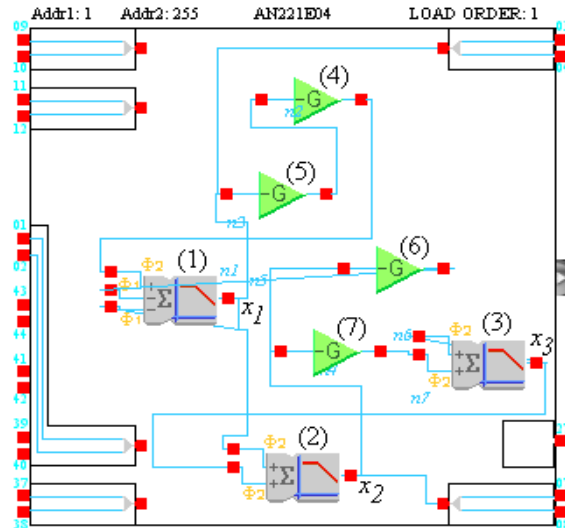
olup bu ekibin yapmış olduğu ölçeklendirme işlemi sunulan çalışmada açıkça ifade edilmemiş ve kullanılan parametre değerleri orijinal sisteme ait denkleme göre farklılık göstermiştir. Bu nedenle aşağıda verilen ölçeklendirilmiş devre denklemleri yeniden oluşturulmuş ve buna göre devre tasarımı yapılmıştır.

$$\begin{aligned}\dot{x}_1 &= -x_1 + 1.9285y_1 - 1.5714x_1 + 4.5x_2 \\ \dot{x}_2 &= -x_2 + 2x_1 + 2x_3 \\ \dot{x}_3 &= -x_3 - 7.143x_2 + x_3\end{aligned}\quad (4.19)$$

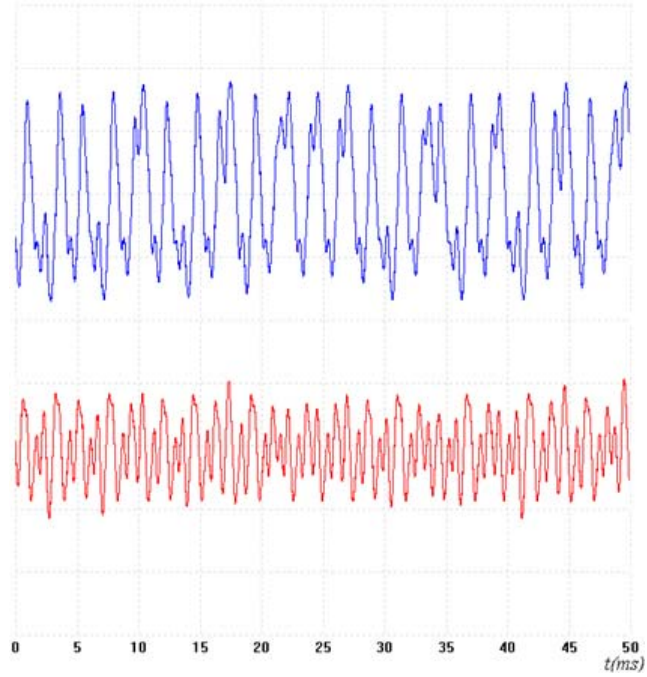
Burada y_1 , parçalı doğrusal fonksiyon olup, aşağıdaki gibi tanımlanmaktadır.

$$y_1 = 0.5(|x_1 + 1| - |x_1 - 1|) \quad (4.20)$$

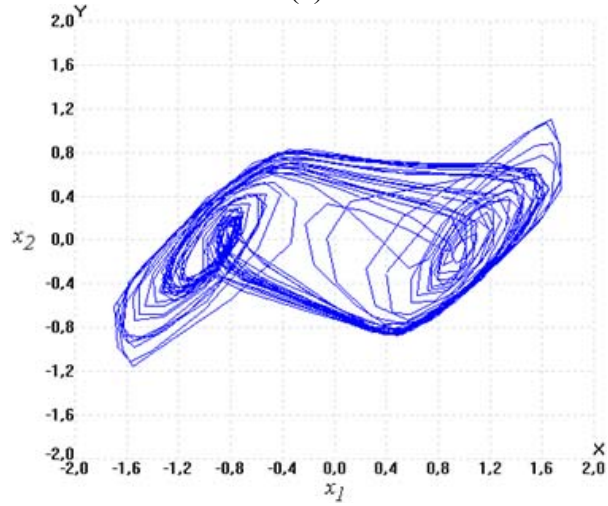
Ölçeklendirilmiş denklem takımına göre oluşturulan FPAA gerçekleştirim şeması Şekil 4.21' de verilmiştir. Şemadan görüldüğü üzere durum değişkenlerinden (x_1)' in elde edilmesinde *SUMFILTER* blokla beraber üç adet *GAININV* bloğu kullanılmıştır. *GAININV* bloklardan arka arkaya bağlanmış iki tanesi doğrusal olmayan fonksiyon (y_1)' i gerçekleştirmek için kullanılmıştır. Kalan diğer blok ise devre kazançlarının elde edilmesinde *SUMFILTER* bloğa ek olarak kullanılmıştır. (x_2) ve (x_3) dinamikleri *SUMFILTER* blokların çıkışlarından elde edilmektedir. Devrede kullanılan dördüncü *GAININV* bloğu ise (x_3) dinamiğinin elde edilmesi amacıyla devre kazancının ayarlanmasında kullanılmıştır. Bu gerçekleştirim sayesinde elde edilen devre dinamikleri ve bu dinamiklerin çeker gösterimleri ise Şekil 4.22' de yer almaktadır.



Şekil 4.21. HSA-Tabanlı Chua Devresinin FPAA Gerçekleştirim Şeması.



(a)



(b)

Şekil 4.22. FPAA Kullanılarak Gerçekleştirilen HSA-Tabanlı Chua Devresinin, (a) Kaotik Devre Dinamikleri: Üstteki İşaret x_1 (1 V/div), Altaki İşaret x_2 (1V/div), (b) (x_1-x_2) Düzlemindeki Çeker Gösterimi.

4.4.9.2. HSA-Tabanlı MLC Devresinin FPAA' da Gerçekleştirimi

Ayrık elemanlarla tasarlanmış ve nümerik analizleri yapılmış olan otonom olmayan MLC devresinin FPAA tasarımı Bölüm 4.4.6' da verilmişti. Burada ise HSA-tabanlı olarak tasarlanan MLC devresinin FPAA tasarımı incelenecektir. MLC devresinin durum denklemleri, HSA tabanlı gerçekleştirim için yeniden düzenlenmiştir. Düzenlenen durum denklemleri aşağıdaki verilmiştir.

$$\begin{aligned}\dot{x}_1 &= -x_1 + a_1 y_1 + s_{11} x_1 + s_{12} x_2 \\ \dot{x}_2 &= -x_2 + s_{21} x_1 + s_{22} x_2 + i_2\end{aligned}\quad (4.21)$$

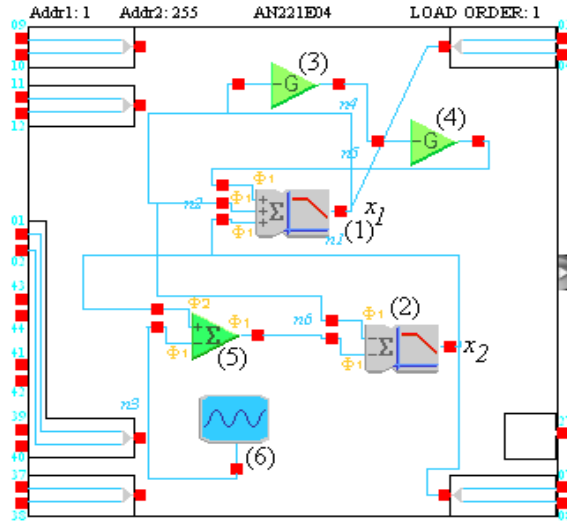
burada “ a ” ve “ s ” parametreleri, $a_1 = (b - a)$; $s_{11} = 1 - b$; $s_{12} = 1$; $s_{21} = -1$; $s_{22} = -\nu$ ve $i_2 = F \sin(\omega t)$ şeklinde seçildiğinde ve x_1 ve x_2 , Denklem (3.22)’deki x ve y ’ye eşit kabul edildiğinde orijinal MLC devresinin durum denklemleri elde edilmiş olur. HSA-tabanlı MLC devresinin kaotik davranış sergilemesi için $a_1 = 0.47$, $s_{11} = 1.55$, $s_{12} = 1$, $s_{21} = -1$, $s_{22} = -0.015$ olarak seçilmiştir.

Denklem (4.21) ile tanımlanan HSA-tabanlı MLC devre modelinin, FPAA-tabanlı gerçekleştiriminden önce nümerik analiz işlemi yapılmıştır ve MLC devresinin orijinal kaotik dinamiklerini elde etmek için Denklem (4.21)’deki devre parametreleri, nümerik analiz sonuçlarına göre modifiye edilmiştir. Modifiye edilen devre modeli, aşağıdaki durum denklemleri ile tanımlanmıştır [39].

$$\begin{aligned}\dot{x}_1 &= -x_1 + 0.47 y_1 + 1.55 x_1 + x_2 \\ \dot{x}_2 &= -x_2 - x_1 + \alpha(-0.0935 x_2 + f \sin(\omega t))\end{aligned}\quad (4.22)$$

Burada f , α ve ω , sırasıyla 0.15, 1 ve 0.6 olarak belirlenmiş olup (y_1), Denklem (4.20) ile ifade edilmiştir.

Denklem (4.22)’ye göre tasarlanan HSA-tabanlı MLC devresinin FPAA gerçekleştirim şeması Şekil 4.23’de verilmiştir. Bu şemada (x_1) ve (x_2) durum değişkenleri, *SUMFILTER* blokların çıkışlarından elde edilmiştir. Devre kazançları, *SUMFILTER* blokların ve *SUMDIFF* bloğun blok kazançları ayarlanarak sağlanmıştır. *OSCILLATORSINE* bloğu, sinüs dalga işaret kaynağı olarak kullanılmıştır. HSA-tabanlı MLC devre modelinin FPAA gerçekleştirimindeki *GAININV* blokları, HSA-tabanlı Chua devresinde olduğu gibi doğrusal olmayan çıkış fonksiyonu (y_1)’i gerçekleştirmek için kullanılmıştır. HSA-tabanlı MLC devresinin FPAA gerçekleştirimi için yapılan deneysel çalışmalardan elde edilen kaotik dinamikler ve (x_1 - x_2) düzlemindeki çeker gösterimi, Şekil 4.24’de yer almaktadır.



Şekil 4.23. HSA-Tabanlı MLC Devresinin FPAA Gerçekleştirim Şeması.

4.4.10. FPAA-Tabanlı n -Çekerli Kaos Üreteçleri

Literatürde üzerinde en çok çalışılan Chua devresinin farklı gerçekleştirmeleri bulunmaktadır. Bunlardan biri de Chua devresinin n -çekerli olarak gerçekleştirimidir. n -çekerli Chua devresinin durum denklemleri aşağıda yer almaktadır [131]:

$$\begin{aligned} \delta \frac{dx}{d\tau} &= \alpha(y - h(x)) \\ \frac{dy}{d\tau} &= \delta x - y + \mu z \\ \eta \frac{dz}{d\tau} &= -\beta y \end{aligned} \quad (4.23)$$

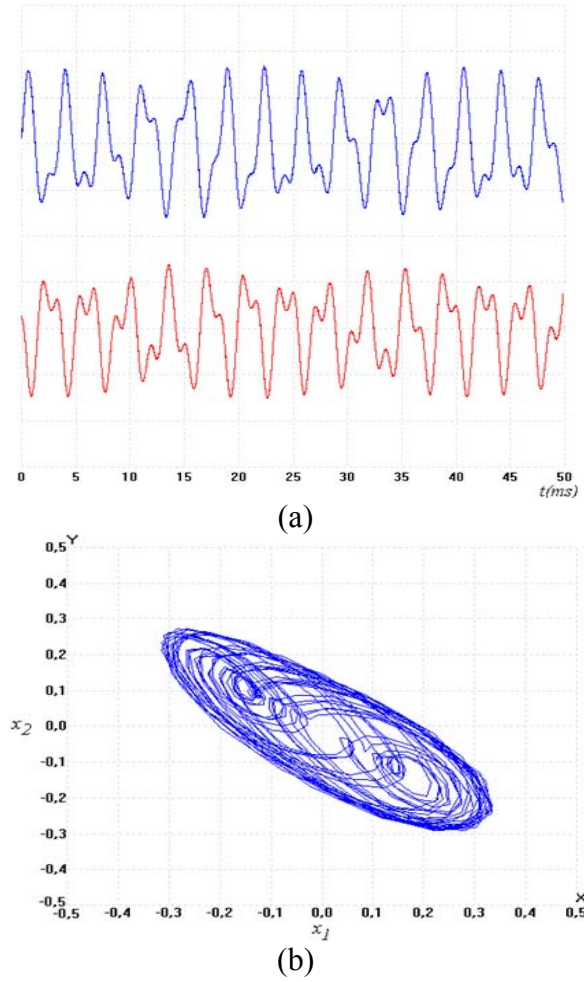
Burada (x) , (y) ve (z) , durum değişkenleri olup, $h(x)$ parçalı doğrusal karakteristiktir.

$$h(x) = m_{2q-1} \delta x + \frac{1}{2} \sum_{i=1}^{2q-1} (m_{i-1} - m_i) (|\delta x + c_i| - |\delta x - c_i|) \quad (4.24)$$

Burada q , doğal sayıdır. $q=1$, $\alpha=9$ ve $\beta=14.286$ olduğunda Chua devresinin davranışı olan çift çeker elde edilmektedir [18]. $m = [m_0; m_1; \dots; m_{2q-1}]$ ve $c = [c_1; c_2; \dots; c_{2q-1}]$ 'dir ve n -çekerli gösterimler için parametre değerleri aşağıda yer almaktadır [55, 131, 132]:

- $q=1$: çift çeker (2-çeker)

$$\begin{aligned} m &= [-1/7; +2/7] \\ c &= 1 \end{aligned} \quad (4.25)$$



Şekil 4.24. FPAA Kullanılarak Gerçekleştirilen HSA-Tabanlı MLC Devresinin, (a) Kaotik Devre Dinamikleri: Üstteki İşaret x_1 (0.2 V/div), Altteki İşaret x_2 (0.2V/div), (b) (x_1-x_2) Düzlemindeki Çeker Gösterimi.

- $q=2$: iki-çift çeker (4-çeker)

$$m=[-1/7; +2/7; -4/7; +2/7] \quad (4.26)$$

$$c=[1; 2.15; 3.6]$$

- $q=2$: üç-çeker

$$m=[+0.9/7; -3/7; +3.5/7; -2.4/7] \quad (4.27)$$

$$c=[1; 2.15; 4]$$

- $q=3$: beş-çeker

$$m=[+0.9/7; -3/7; +3.5/7; -2.7/7; +4/7; -2.4/7] \quad (4.28)$$

$$c=[1; 2.15; 3.6; 6.2; 9]$$

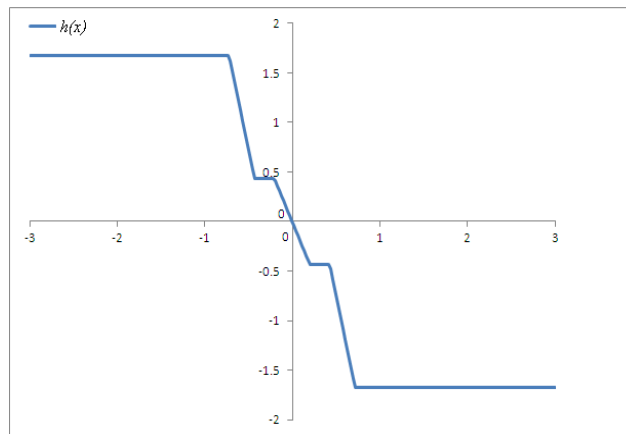
- $q=3$: yedi-çeker

$$m=[+0.9/7; -3/7; +3.5/7; -2.4/7; +2.52/7; -1.68/7; +2.52/7; -1.68/7] \quad (4.29)$$

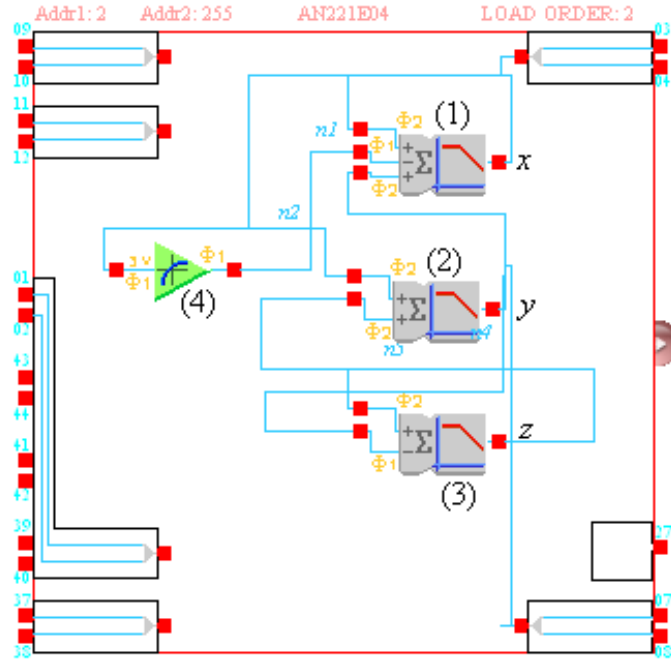
$$c=[1; 2.15; 3.6; 6.2; 9; 14; 25]$$

n -çekerli kaos üreticinin ilk olarak nümerik analizi yapılmış olup analiz sonuçlarına göre ölçeklendirme işlemi yapılmıştır. Ölçeklendirme işleminde kullanılan parametreler, Denklem (4.23) ve (4.24)' de (δ) ve (η) ile ifade edilmiştir. Yeniden yapılandırılabilir özelliğe sahip FPAA' da n -çekerli kaos üreticinin ölçeklendirilmiş devre denklemleri ile gerçekleştirimi yapılmıştır. Tüm gerçekleştirimlerde üç adet *SUMFILTER* blokla bir adet *TRANSFER FUNCTION* blok kullanılmıştır. *SUMFILTER* bloklar, durum değişkenleri (x), (y) ve (z)' i elde etmede kullanılmışlardır. Diğer blok olan *TRANSFER FUNCTION* blok ise, $h(x)$ doğrusal olmayan fonksiyonu gerçekleştirmekte kullanılmıştır. Burada örnek olarak iki-çift çekerli yapı gerçekleştirmek için *TRANSFER FUNCTION* bloğa yüklenecek $h(x)$ doğrusal olmayan fonksiyona ait dosya, ExcelTM, de hazırlanmıştır ve hazırlanan dosyanın grafiği Şekil 4.25' de verilmiştir. Bu grafikten görüleceği üzere fonksiyon 4 noktada kırılmaya uğramakta olup, sistemin iki-çift çekerli yapı sergileyebileceğini göstermektedir. n -çekerli kaos üretici için kullanılan FPAA gerçekleştirim şeması Şekil 4.26' da verilmiştir.

3-çekerli kaos üretici için Denklem (4.23) ve (4.24)' de yer alan (δ) ve (η) değerleri 1 olarak belirlenmiştir. Bu değerlere göre sistem kazançları *SUMFILTER* bloğa kaydedilir. Ayrıca bu değerlerle $h(x)$ doğrusal olmayan fonksiyon ExcelTM, de (*abs*) komutları ile oluşturulur ve oluşturulan dosya (*.csv*) olarak kaydedilir. Kaydedilen bu dosya *TRANSFER FUNCTION* bloğa yüklenir. FPAA-tabanlı 3-çekerli tasarımda kullanılan bloklar parametre değerleri ile beraber Tablo 4.10' da yer almaktadır. Bu gerçekleştirimden elde edilen deneysel sonuçlar Şekil 4.27(a)' da verilmiştir.

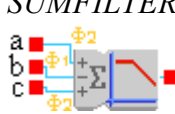





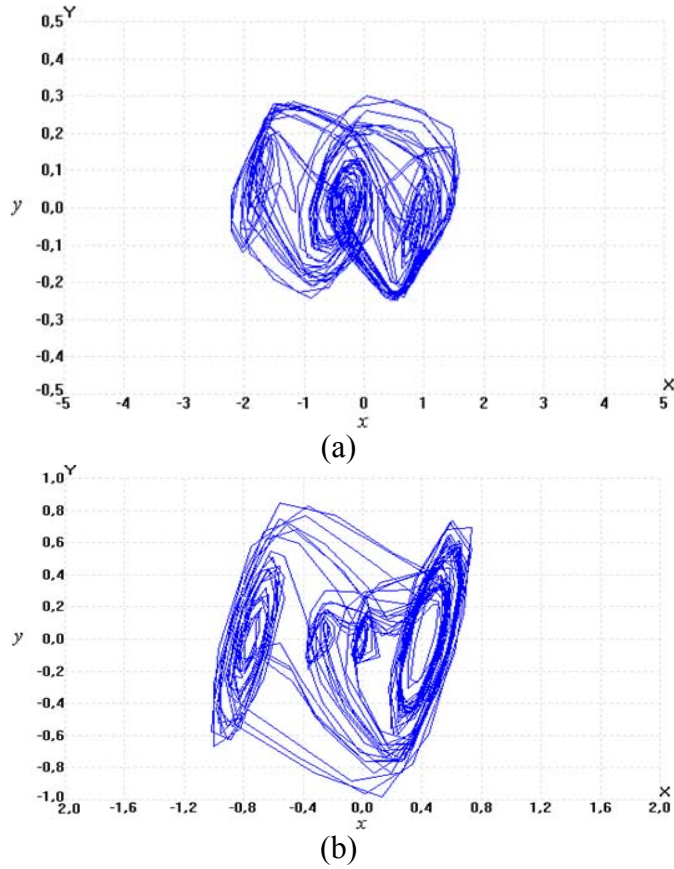
Şekil 4.25. İki-Çift Çekerli Yapı İçin *TRANSFER FUNCTION* Blokta Elde Edilen $h(x)$ Fonksiyonun Grafiği.

Şekil 4.26. FPAA-Tabanlı n -Çekerli Kaos Üretici.

2-çift çekerli (4-çekerli) yapıyı elde etmek için Denklem (4.23) ve (4.24)' deki (δ) ve (η) ifadeleri, $\delta=5$ ve $\eta=7$ olarak belirlenmiştir. Bu parametrelere göre yapılan ölçeklendirme işleminden sonra gerçekleştirilen FPAA' daki tasarım şeması Şekil 4.26' da yer almaktadır. Bu tasarım önceki tasarıma göre farklı (δ) ve (η) değerlerine sahip

Tablo 4.10. FPAA-Tabanlı 3-Çekerli Kaos Üretcinin CAM Modülleri ve Parametre Değerleri.

CAM Modülü	CAM No.su	Ayarlanan Parametresi	Parametre Değeri
 SUMFILTER	1	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı c Giriş Kazanç Katsayısı	0.6 1 9 10
 SUMFILTER	2	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.6 0.95 1
 SUMFILTER	3	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.6 1.75 14.286
 TRANSFER FUNCTION	4		

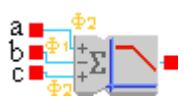
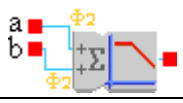




Şekil 4.27. (a) FPAA-Tabanlı 3-Çekerli Kaos Üreticinin $(x-y)$ Düzlemindeki Çeker Gösterimi, (b) FPAA-Tabanlı İki-Çift Çekerli Kaos Üreticinin $(x-y)$ Düzlemindeki Çeker Gösterimi.

olduğu için devre kazançlarında da farklılık söz konusudur. Ayrıca (q) ve (δ) değerlerine göre $h(x)$ doğrusal olmayan fonksiyon yeniden oluşturulmuş ve yapılan tasarıma yüklenmiştir. Bu tasarımda kullanılan bloklara ait parametre değerleri Tablo 4.11' de verilmiş olup, deneysel sonuçlardan elde edilen iki çift çekerli gösterim Şekil 4.27(b)' de yer almaktadır.

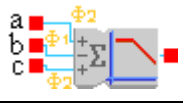
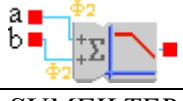


5-çekerli yapıyı oluşturmak için (δ) ve (η) ölçeklendirme parametreleri 5 olarak belirlenirken 7-çekerli yapıyı oluşturmak için sırasıyla 10 ve 12 olarak belirlenmiştir. Bu iki yapı için de FPAA tasarım şeması Şekil 4.26' da gösterilmiştir. Bu iki tasarım için devre kazançları ayrı ayrı belirlenmiştir. Bunun yanı sıra $h(x)$ doğrusal olmayan fonksiyon (q) ve (δ) değerlerine göre yeniden hesaplaması yapılarak (.csv) dosyaları oluşturulmuş ve oluşturulan bu dosyalar ayrı ayrı *TRANSFER FUNCTION* bloğa yüklenmiştir.

Tablo 4.11. FPAA-Tabanlı İki-Çift Çekerli Kaos Üreticinin CAM Modülleri ve Parametre Değerleri.

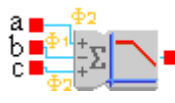
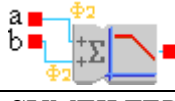
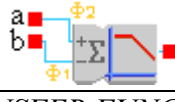

CAM Modülü	CAM No.su	Ayarlanan Parametresi	Parametre Değeri
<i>SUMFILTER</i> 	1	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı c Giriş Kazanç Katsayısı	0.6 1 2.5 1.8
<i>SUMFILTER</i> 	2	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.6 6.2 7
<i>SUMFILTER</i> 	3	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.6 1 2.04
<i>TRANSFER FUNCTION</i> 	4		

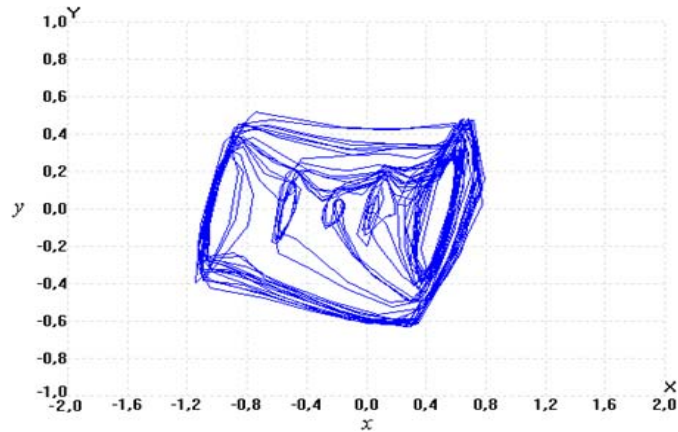
Bu tasarım şemasındaki bloklar parametre değerleri ile birlikte 5-çekerli yapı için Tablo 4.12’ de yer alırken 7-çekerli yapı için Tablo 4.13’ de yer almaktadır. FPAA-tabanlı 5-çekerli ve 7-çekerli yapılardan elde edilen (x-y) düzlemindeki çeker gösterimleri sırasıyla Şekil 4.28(a) ve (b)’ de verilmiştir.

Tablo 4.12. FPAA-Tabanlı 5-Çekerli Kaos Üreticinin CAM Modülleri ve Parametre Değerleri.

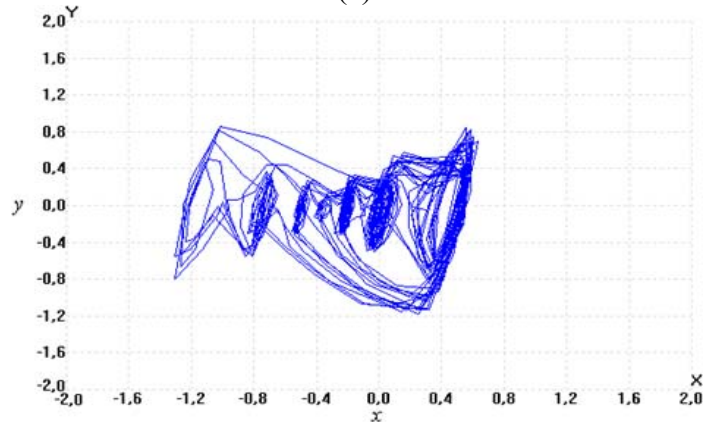
CAM Modülü	CAM No.su	Ayarlanan Parametresi	Parametre Değeri
<i>SUMFILTER</i> 	1	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı c Giriş Kazanç Katsayısı	0.9 3.6 3 1.8
<i>SUMFILTER</i> 	2	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.9 5 5
<i>SUMFILTER</i> 	3	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.9 1 1.89
<i>TRANSFER FUNCTION</i> 	4		

Tablo 4.13. FPAA-Tabanlı 7-Çekerli Kaos Üreticinin CAM Modülleri ve Parametre Değerleri.

CAM Modülü	CAM No.su	Ayarlanan Parametresi	Parametre Değeri
<i>SUMFILTER</i> 	1	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı c Giriş Kazanç Katsayısı	0.8 1 2.6 1
<i>SUMFILTER</i> 	2	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.8 12.8 12.25
<i>SUMFILTER</i> 	3	Köşe Frekansı (kHz) a Giriş Kazanç Katsayısı b Giriş Kazanç Katsayısı	0.8 1 1.1905
<i>TRANSFER FUNCTION</i> 	4		



(a)



(b)

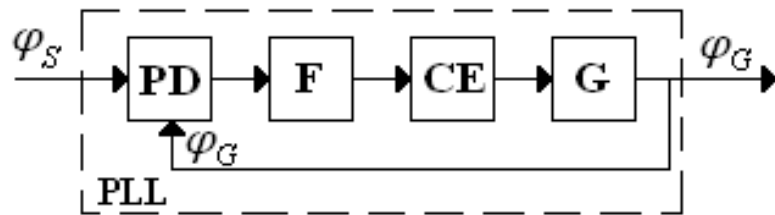
Şekil 4.28. (a) FPAA-Tabanlı 5-Çekerli Kaos Üreticinin (x-y) Düzlemindeki Çeker Gösterimi, (b) FPAA-Tabanlı 7-Çekerli Kaos Üreticinin (x-y) Düzlemindeki Çeker Gösterimi.

Bu kısımda FPAA' nın yeniden yapılandırılabilir özelliği sayesinde n -çekerli kaos üreteçlerinden birisi çalışırken diğer bir üreteç üzerinde işlemler yapıлып anında FPAA' ya aktarılarak deneysel sonuçlar alınabilmektedir. Ama bu n -çekerli üreteçleri ayrık elemanlarla gerçekleştirmek istenseydi ilk olarak hepsi için ayrı ayrı uğraşılacaktı ve bu da çok vakit alacaktır. İkinci olarak çok çekerli devre gerçekleştirimi yapıldığı için çok fazla devre elemanı kullanılacağından karmaşık bir devre ortaya çıkacaktır. Devreyi kurarken bir hata yapılırsa hatayı bulmak çok zorlaşacaktır. Ayrıca devre için gerekli tam eleman değerlerini bulmak ve devrede kullanılacak işlemsel kuvvetlendiricileri temin etmek büyük birer problemdir. Bu sorunların üstesinden gelmek için FPAA' nın iyi bir çözüm olduğu bu kısımda da görülmüştür.

4.5. FPAA-Tabanlı PLL Modeli

Bu kısımda FPAA-tabanlı kaotik osilatörlere ilaveten farklı modellemelerin de FPAA' da gerçekleştirilebileceğini göstermek amacıyla faz kilitlemeli döngü (Phase-Locked Loop-PLL) modeli incelenmiştir. PLL, üreteç frekanslarını kontrol eden bir yapı olup döngü şeması Şekil 4.29' da verilmiştir. Mevcut fazı φ_G olan kontrollü **G** üretecinin çıkışındaki osilasyonlar ve mevcut fazı φ_S olan referans üretecin osilasyonları, faz dedektörünün (Phase Dedector-PD) $\varphi = \varphi_S - \varphi_G$ faz farklı çıkış voltajı üretmesi için faz dedektörü tarafından çarpma işlemine tabii tutulur. Faz dedektörünün çıkışındaki sinyal, $K(p)$ transfer fonksiyonuna sahip **F** döngü filtresi tarafından filtrelendikten sonra **CE** kontrol elemanının girişine gönderilir. **CE** kontrol elemanı, **G** üretecinin çıkış frekans ve fazını değiştirebilmektedir. Kontrollü **G** üretecinin çıkış frekansını ve fazını değiştirmekteki amaç, bu işaretin referans üreteçten alınan sinyalin frekansı ve fazı açısından uyumlu olmasını sağlamaktır [55].

Faz kilitlemeli döngü sisteminin matematiksel modeli aşağıda verilmiştir.



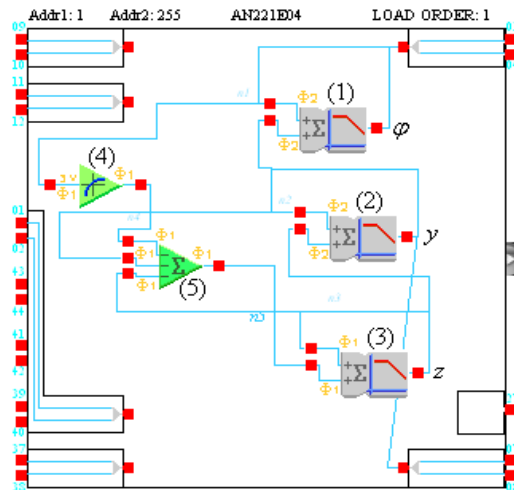
Şekil 4.29. Faz Kilitlemeli Döngü Şeması.

$$\frac{p\varphi}{\Omega} + K(p)\text{Sin}\varphi = \gamma \quad (4.30)$$

PLL dinamikleri, ikinci dereceden filtre kullanıldığında daha karmaşık hale gelir. $K(p)=1/(a_2p^2+a_1p+1)$ transfer fonksiyonu ile basit bir ikinci dereceden filtre için Denklem (4.30) aşağıdaki ifadeye dönüşür [55].

$$\begin{aligned} \frac{d\varphi}{d\tau} &= y \\ \frac{dy}{d\tau} &= z \\ \mu \frac{dz}{d\tau} &= \gamma - \text{Sin}\varphi - y - \varepsilon z \end{aligned} \quad (4.31)$$

Bu ikinci dereceden filtreye sahip PLL modeli, doğrusal olmayan dinamikler ve kaos sergileme potansiyelinde olduğu için bu model kaos üretici olarak da kullanılabilir. Burada sistemin kaotik dinamik sergilemesi için ($\gamma=0$) ve ($\varepsilon=1$) seçilmiştir. (μ) ise ayarlanabilir parametredir. Bu parametre değerlerine sahip Denklem (4.31) ile ifade edilen PLL modelin FPAA' da gerçekleştirimi mümkündür [80] ve gerçekleştirim şeması da Şekil 4.30' da verilmiştir. Gerçekleştirmede kullanılan *SUMFILTER* bloklar, (φ), (y) ve (z) sistem dinamiklerini elde etmek amacıyla kullanılmışlardır. Sistem kazançları ise, *SUMFILTER* ve *SUMDIFF* blokların kazançları ile gerçekleştirilmektedir. Sistem denkleminde bulunan ($\text{sin } \varphi$) ifadesini gerçekleştirmek için *TRANSFER FUNCTION* blok kullanılmıştır. Denklem (4.31)' deki denklem



Şekil 4.30. PLL Modelinin FPAA Gerçekleştirim Şeması.

takımında yer alan (μ) katsayısı, sistem davranışını periyodik davranıştan kaotik davranışa kadar değişimini sağlayan ayarlanabilen bir katsayıdır. $\mu=6.66$ iken sistem davranışı, kaotik davranış olup Şekil 4.31(a) ve (b)' de verilmiştir. Sistem $\mu=5.55$ ' de periyot-2 davranışını sergilerken $\mu=5$ ' de periyodik davranış sergilemektedir ve bu davranışlar, Şekil 4.31(c) ve (d)' de gösterilmektedir.

4.6. FPAA-Tabanlı Kaotik Sistemlerin Kullanıldığı Senkronizasyon Gerçekleştirmeleri

Son 10 yılı aşkın süredir kaos kontrol ve kaotik senkronizasyon, gizlilik ve güvenlik açısından haberleşme sistemlerinde çok önemli bir rol oynamaktadır. Senkronizasyon kavramı, yayılı spektruma sahip bir kaotik işaretin alıcıda tekrar çözülebilmesidir. Kaotik senkronizasyon, güvenilir haberleşme, sinyal işleme gibi uygulamalarda yaygın olarak kullanılmaktadır. Bugüne kadar değişik senkronizasyon teknikleri literatüre sunulmuştur. Pecora & Carroll metodu [133], kuplajlama metodu [134], kaskad bağlantı metodu [134], doğrusal ve doğrusal olmayan geribeslemeli kontrol [135], aktif ve pasif kontrol [136, 137], impulsif kontrol [124, 138-140] bu tekniklerden sadece birkaç tanesidir. Tez çalışmasının bu kısmında FPAA-tabanlı otonom Chua osilatörünün ve otonom olmayan özelliğe sahip MLC osilatörünün kullanıldığı senkronizasyon çalışmaları hakkında detaylı bilgiler verilmiştir.

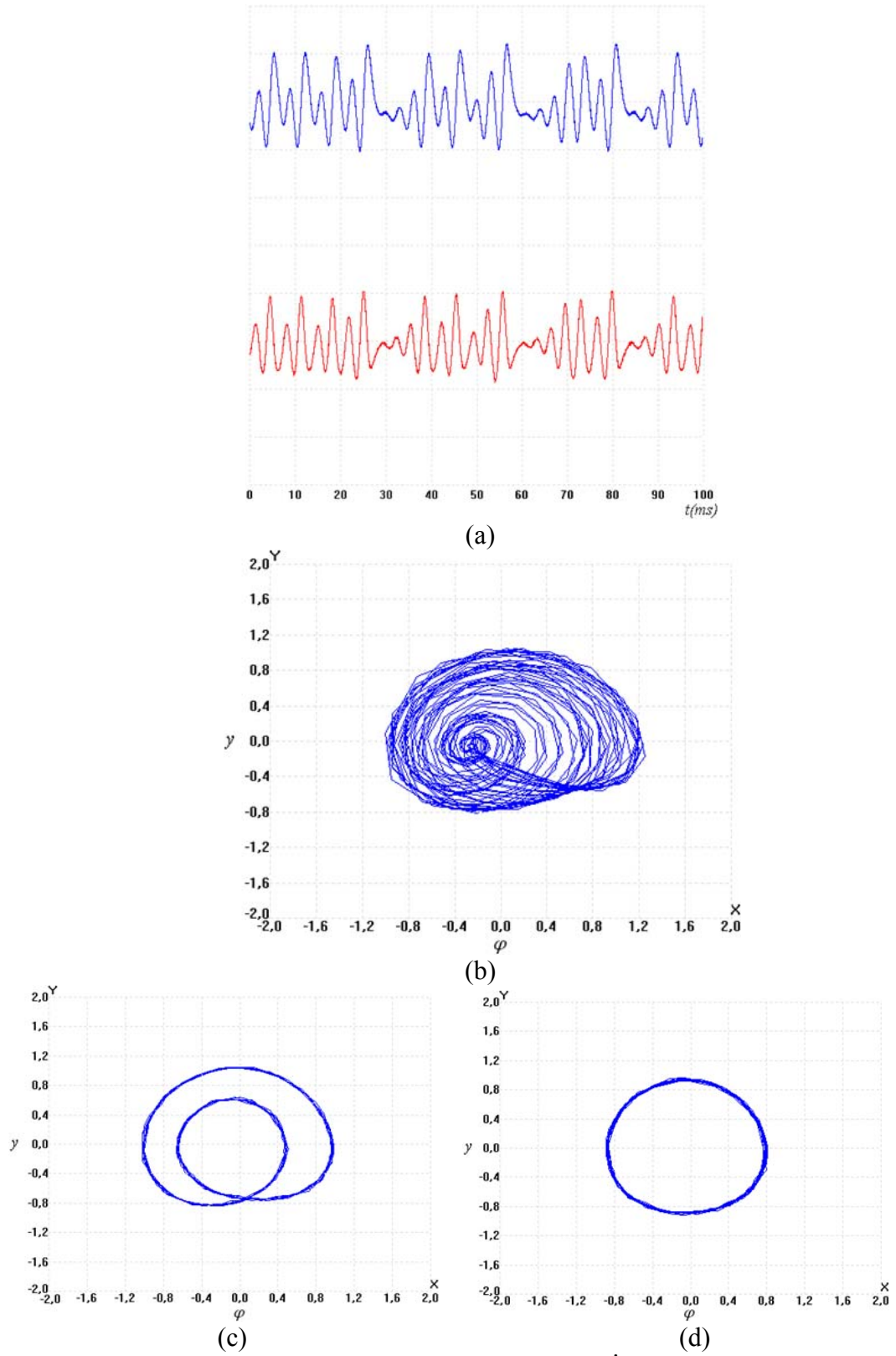
Bu bölümde zaman domeninde ve X-Y modunda senkronizasyonu görüntülemek için Şekil 4.32' deki deney düzeneği kullanılmıştır.

4.6.1. FPAA-Tabanlı Chua Devresinin Kullanıldığı Senkronizasyon Gerçekleştirmeleri

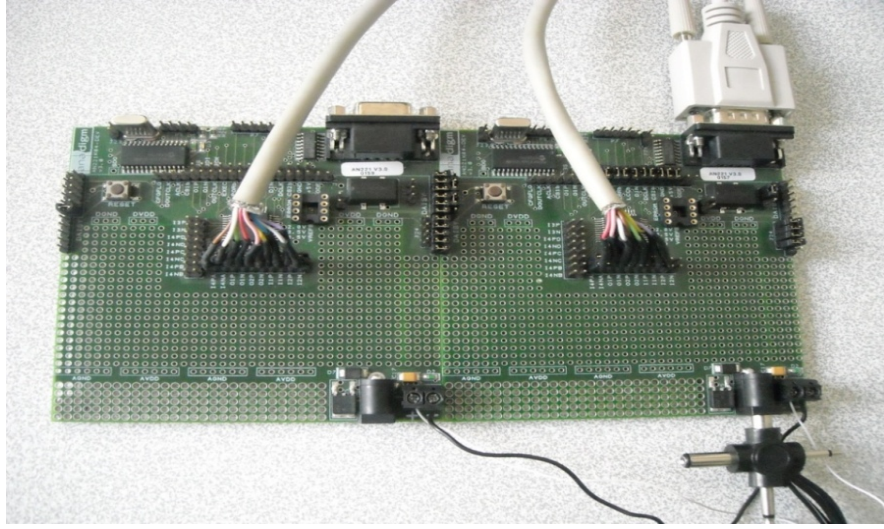
Bu kısımda Chua devresinin kullanıldığı senkronizasyon gerçekleştirmeleri, kaskad bağlantı yöntemi, Pecora&Carroll yöntemi ve kuplajlama yöntemi ile yapılmıştır.

4.6.1.1. FPAA-Tabanlı Chua Devresinin Kullanıldığı Kaskad Bağlantı Yöntemi İle Senkronizasyon Gerçekleştirimi

Chua devresinin kullanıldığı senkronizasyon gerçekleştirimi farklı yöntemlerle yapılabilir. Bunlardan birisi, kaskad bağlantılı senkronizasyon yöntemidir [141]. Bu yöntemde alıcı sistemin davranışı verici sistemin davranışına bağlıdır. $u(y,z)=y+z$ şeklinde birbirlerine kaskad bağlanan denklem takımı aşağıda verilmiştir [142]:



Şekil 4.31. FPAA-Tabanlı PLL Modelden, (a) $\mu=6.66$ İken Elde Edilen Kaotik Devre Dinamikleri: Üstteki İşaret φ (1 V/div), Alttaki İşaret y (1 V/div), (b) $\mu=6.66$ İken Elde Edilen (φ - y) Düzlemindeki Kaotik Çeker Gösterimi, (c) $\mu=5.55$ İken Elde Edilen Periyot 2 Davranışı, (d) $\mu=5$ İken Elde Edilen Periyot 1 Davranışı.



Şekil 4.32. FPAA-Tabanlı Senkronizasyon Gerçekleştirmeleri İçin Deneysel Kurulum. Bu Kurulumlarda Anadigm Firmasının AN221E04 Tipi FPAA [6] Kullanılmıştır.

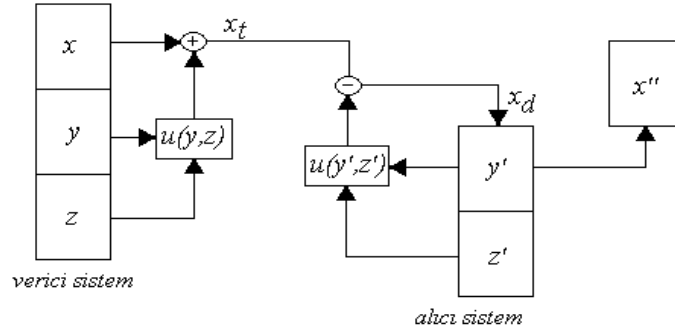
Verici sistem:

$$\begin{aligned}
 \dot{x} &= \alpha(y - x - f(x)) \\
 \dot{y} &= x - y + z \\
 \dot{z} &= -\beta y \\
 x_t &= x + u(y, z) = x + y + z
 \end{aligned} \tag{4.32}$$

Alıcı sistem:

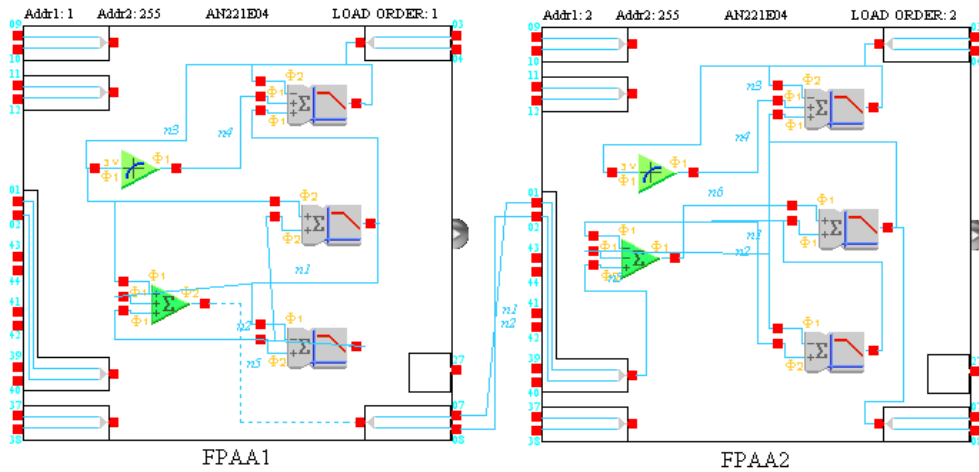
$$\begin{aligned}
 x_d &= x_t - u(y', z') = x_t - (y' + z') \\
 \dot{x}'' &= \alpha(y' - x'' - f(x'')) \\
 \dot{y}' &= x_d - y' + z' \\
 \dot{z}' &= -\beta y'
 \end{aligned} \tag{4.33}$$

Kaskad bağlantı yöntemi kullanılarak yapılan senkronizasyonun blok diyagramı, Şekil 4.33' de gösterilir iken, FPAA-tabanlı Chua devresinin kullanıldığı senkronizasyon gerçekleştirim şeması, Şekil 4.34' de yer almaktadır. Bu gerçekleştirimde FPAA' nın sınırlı kapasitesinden dolayı iki adet FPAA kullanılmıştır. Birinci FPAA uygulama kartı, FPAA1 olarak adlandırılmış olup, verici modül olarak tasarlanmıştır. İkinci FPAA uygulama kartı ise, FPAA2 olarak adlandırılmış olup, alıcı modülü gerçekleştirir.

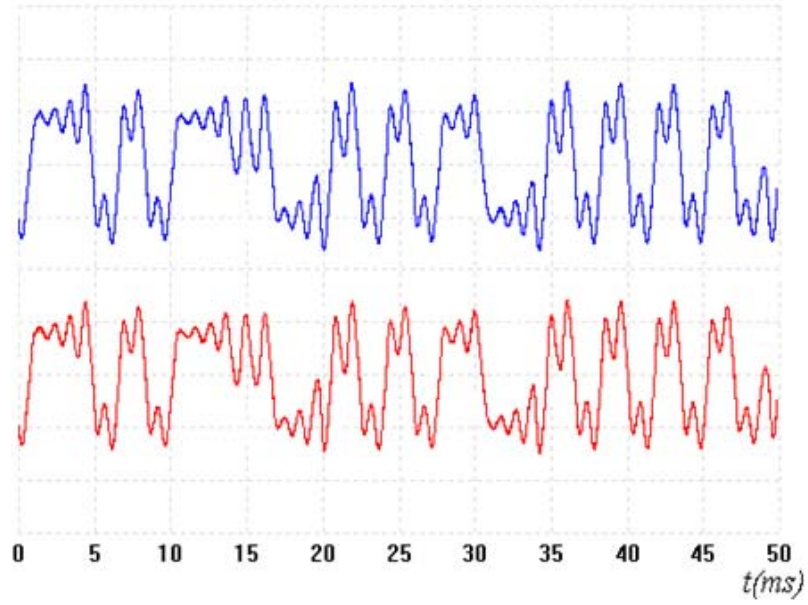


Şekil 4.33. Chua Devresinin Kullanıldığı Kaskad Bağlantı Yöntemi İle Yapılan Senkronizasyonda Kullanılan Blok Diyagram.

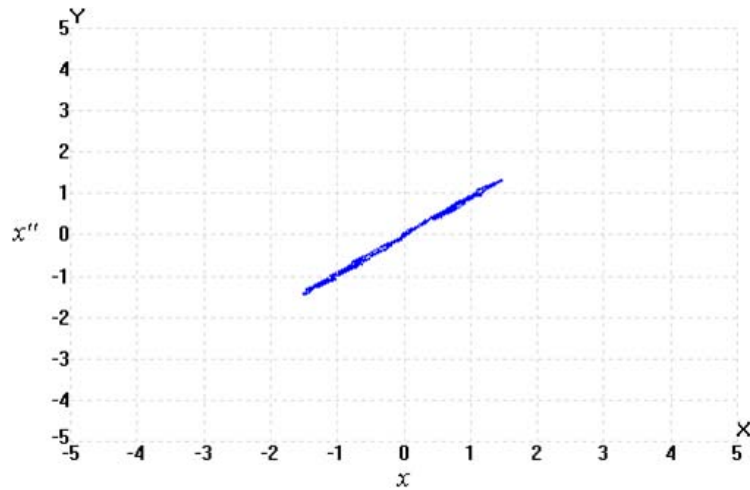
FPAA' da gerçekleştirilen verici kısımda öncelikle Chua devresinin dinamikleri oluşturulmuştur. Daha sonra $x_t = x + u(y, z) = x + y + z$ ifadesindeki (x_t) taşıyıcı işareti oluşturmak için *SUMDIFF* bloğu kullanılmıştır. *SUMDIFF* bloğu, Chua devresinin dinamiklerini toplama işlemi gerçekleştirmiştir. Toplanan bu işaret alıcı sisteme yani FPAA2' ye bağlantıları yapılarak aktarılmıştır. Alıcı sistemde ise ilk önce Chua devresinin (y') ve (z') dinamikleri oluşturulmuştur. Burada $x_d = x_t - u(y', z') = x_t - (y' + z')$ ifadesini gerçekleştirmek için *SUMDIFF* bloğu kullanılmıştır. *SUMDIFF* bloğun çıkışından elde edilen işaret, (y') dinamiğinin oluşturulmasında kullanılmıştır. En son olarak da (y')' ne bağlı olarak (x'') işareti elde edilmiştir. FPAA-tabanlı Chua devresinin kullanıldığı senkronizasyon uygulamasından elde edilen sonuçlar, Şekil 4.35' de verilmiştir. Bu sonuçlardan alıcıdaki (x'') işaretinin verici kısımdaki (x) işareti ile aynı olduğu görülmüştür.



Şekil 4.34. FPAA-Tabanlı Chua Devresinin Kullanıldığı Kaskad Bağlantı Yöntemi İle Yapılan Senkronizasyon Gerçekleştirim Şeması.



(a)



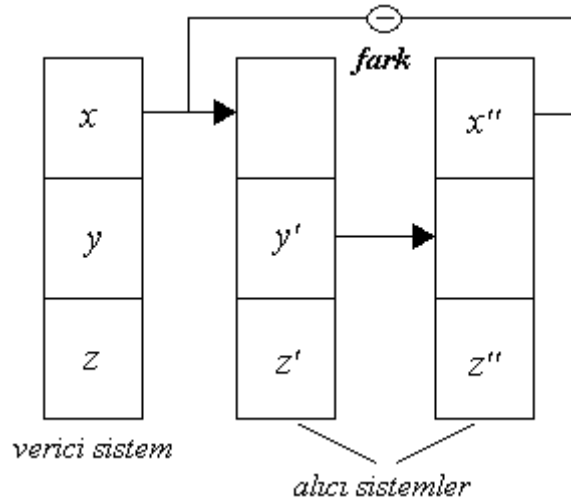
(b)

Şekil 4.35. FPAА-Tabanlı Chua Devresinin Kullanıldığı Kaskad Bağlantı Yöntemi İle Yapılan Senkronizasyon Gerçekleştiriminden Elde Edilen (a) Kaotik Devre Dinamikleri: Üstteki İşaret x (1 V/div), Altteki İşaret x'' (1 V/div), (b) $(x-x'')$ Gösterimleri.

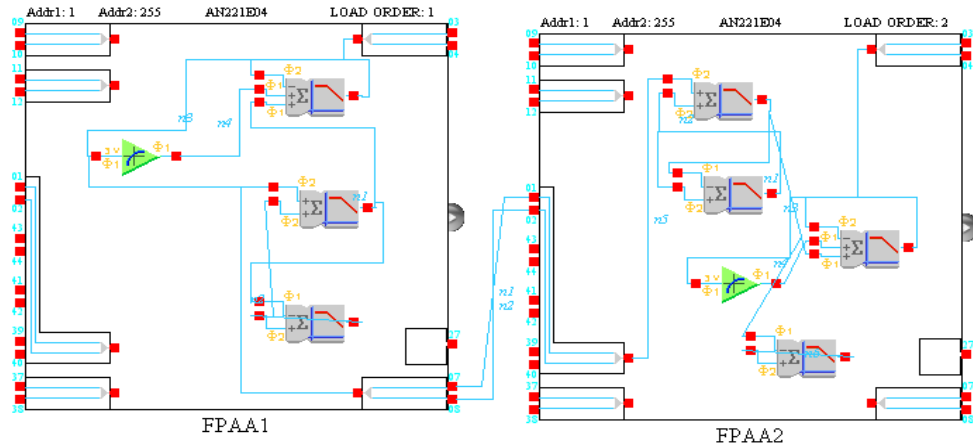
4.6.1.2. FPAА-Tabanlı Chua Devresinin Kullanıldığı Pecora&Carroll Metodu İle Senkronizasyon Gerçekleştirimi

Bir diğer senkronizasyon yöntemi olan Pecora&Carroll yöntemi, literatüre ilk kez Pecora&Carroll tarafından sunulmuştur ve kaotik senkronizasyon uygulamalarında en çok kullanılan metottur [133, 143]. Bu yöntemde, n -boyutlu bir sistem, alıcı ve verici alt sistemler olarak iki parçaya bölünür. Verici sistemin durum değişkenlerinden birisi, alıcı sistemde aynı durum değişkenini üretmek için kullanılır. İki sistemde de aynı

durum deęişkenleri üretildięi zaman, alıcı ve verici sistem, senkronize olmuş olur [133]. Pecora&Carroll yöntemi, literatürde yer alan ve ayrıık elemanlarla tasarlanmış Chua sistemi [141], Lorenz sistemi [144] gibi kaotik sistemlerde başarıyla uygulanmıştır. FPAA-tabanlı Chua devresinin kullanıldığı senkronizasyon uygulaması için bu yöntem denenmiş olup, senkronizasyon blok diyagramı Şekil 4.36' da yer almaktadır. Bu yöntemde verici sistemdeki (x) dinamięi alıcı sisteme gönderilir. Alıcı sistem-1' de bu (x) dinamięine baęlı olarak (y') ve (z') dinamikleri oluşturulur. Oluşturulan (y') dinamięi alıcı sistem-2' ye aktarılır ve burada (x'') ve (z'') işaretlerinin elde edilmesinde kullanılır. Alıcı sistem-2' den elde edilen (x'') işareti verici sistemdeki (x) dinamięi ile uyuşuyorsa senkronizasyon başarıyla tamamlanmış olur. Bu yöntem kullanılarak yapılan FPAA-tabanlı gerçekleştirim ise Şekil 4.37' de gösterilmektedir. Bu gerçekleştirimde verici olarak (x), (y) ve (z) kullanılmış olup, alıcı olarak ($y-z$) ve ($x-z$) sistemler kullanılmıştır. Burada senkronizasyon için verici sisteme ait x dinamięi, alıcıya gönderilmiştir. Verici sistem FPAA1' de, alıcı sistemler FPAA2' de gerçekleşmişlerdir. Bu gerçekleştirim için verici sistemin matematiksel modeli Denklem (4.32) ile ifade edilmiş olup, alıcı sistemin durum denklemleri aşağıda yer almaktadır.



Şekil 4.36. Chua Devresinin Kullanıldığı Pecora&Carroll Yöntemi İle Yapılan Senkronizasyonda Kullanılan Blok Diyagram.



Şekil 4.37. FPAA-Tabanlı Chua Devresinin Kullanıldığı Pecora&Carroll Yöntemi İle Yapılan Senkronizasyon Gerçekleştirim Şeması.

Alıcı sistem-1:

$$y' = x - y' + z' \quad (4.34)$$

$$z' = -\beta y'$$

Alıcı sistem-2:

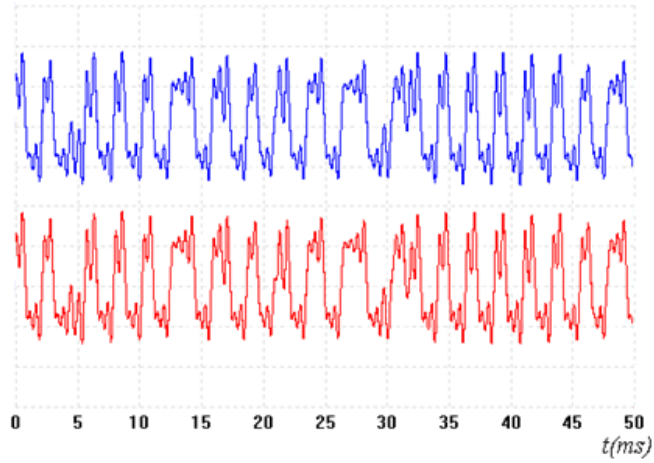
$$x'' = \alpha(y' - x'' - f(x'')) \quad (4.35)$$

$$z'' = -\beta y'$$

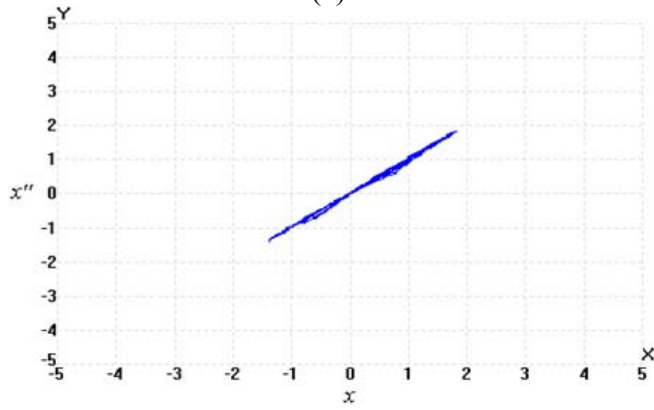
Bu senkronizasyon yönteminden elde edilen sonuçlar hem zaman domeninde hem de $(x - x'')$ düzleminde kaydedilmiş olup, Şekil 4.38' de gösterilmektedir. Şekilden görüldüğü üzere senkronizasyon gerçekleştirimi başarıyla tamamlanmıştır.

4.6.1.3. FPAA-Tabanlı Chua Devresinin Kullanıldığı Kuplajlama Metodu İle Senkronizasyon Gerçekleştirimi

FPAA-tabanlı Chua devresinin kullanıldığı senkronizasyon gerçekleştiriminde kullanılan bir diğer yöntem kuplajlamadır. Kuplajlama metodu da alıcı ve verici sistemden oluşmaktadır. Bu yöntem, Chua devresine iki yönlü olarak uygulanmıştır. İki yönlü kuplajlama metodunda alıcı ve verici sistemlerin davranışı birbirlerine bağlıdır [134]. Chua devresinin senkronizasyon çalışmasında (y) dinamiği kuplajlanmıştır. Bu yöntemle gerçekleştirilen Chua devresinin verici ve alıcı sistemlerinin diferansiyel denklem takımı aşağıda verilmiştir.



(a)



(b)

Şekil 4.38. FPAA-Tabanlı Chua Devresinin Kullanıldığı Pecora&Carroll Yöntemi İle Yapılan Senkronizasyon Gerçekleştiriminden Elde Edilen (a) Kaotik Devre Dinamikleri: Üstteki İşaret x (1 V/div), Alttaki İşaret x'' (1 V/div), (b) $(x-x'')$ Gösterimleri.

Verici sistem:

$$\dot{x} = \alpha(y - x - f(x))$$

$$\dot{y} = x - y + z + \delta_y(y' - y) \quad (4.36)$$

$$\dot{z} = -\beta y$$

Alıcı sistem:

$$\dot{x}' = \alpha(y' - x' - f(x'))$$

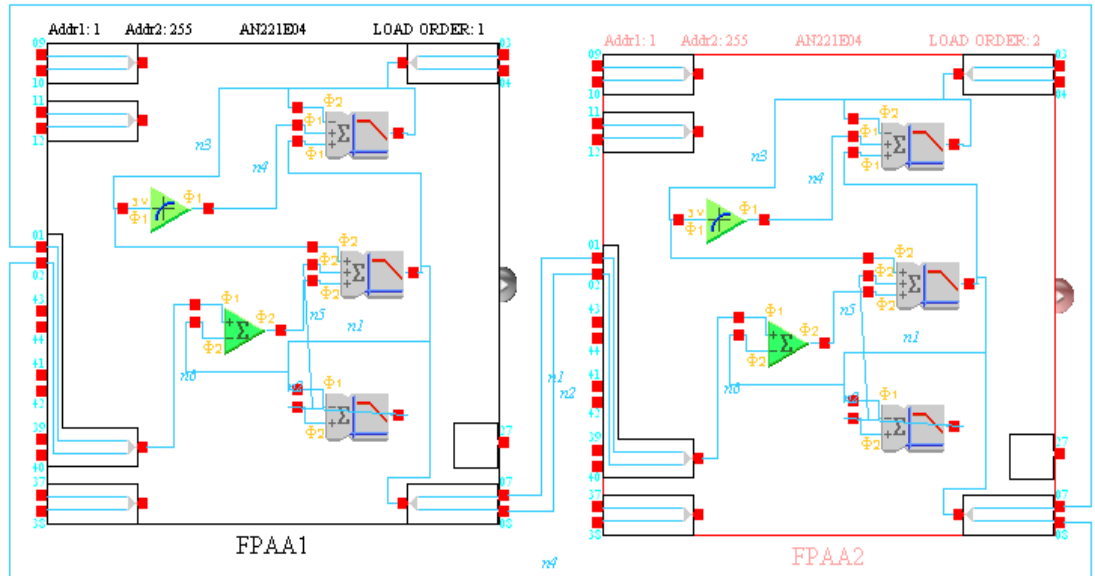
$$\dot{y}' = x' - y' + z' + \delta_y(y - y') \quad (4.37)$$

$$\dot{z}' = -\beta y'$$

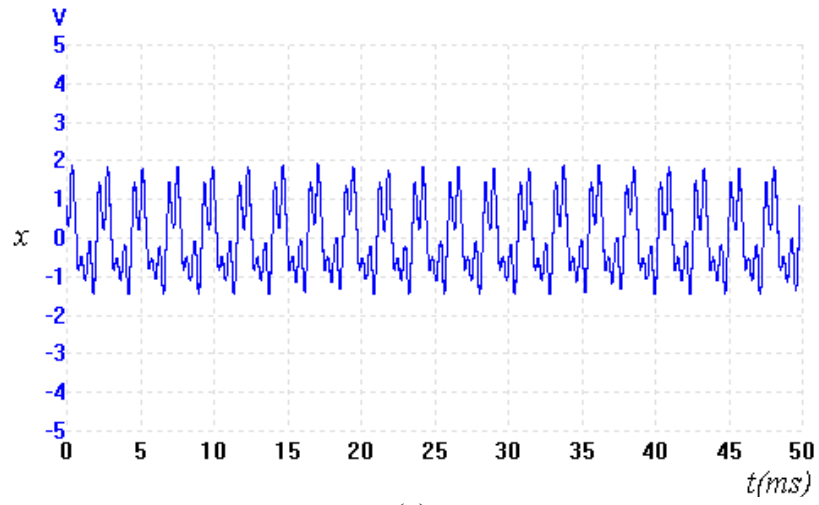
Burada sistemin senkron olabilmesi için $\delta_y > 5.5$ olması gerekmektedir. Deneysel çalışmada $\delta_y = 6$ seçilmiştir. Kuplajlama yöntemi ile yapılan FPAA-tabanlı Chua devresinin kullanıldığı senkronizasyonun gerçekleştirim şeması Şekil 4.39’ da gösterilmektedir. Bu gerçekleştirimde Denklem (4.36)’ ya göre Chua devresinin dinamikleri verici kısımda oluşturulmuştur. Bu dinamikler oluşturulurken (y) dinamiğinde yer alan $\delta_y(y' - y)$ ifadesi *SUMDIFF* bloğunda gerçekleştirilmiştir. Benzer şekilde alıcı kısımda Chua devresinin dinamikleri Denklem (4.37)’ ye göre elde edilmiştir. Bu kısımda da (y') dinamiği elde edilirken $\delta_y(y - y')$ ifadesi *SUMDIFF* bloğunda gerçekleştirilmiştir. Bu yöntemle gerçekleştirilen senkronizasyon uygulamasından elde edilen sonuçlar Şekil 4.40’ da yer almaktadır. Elde edilen bu sonuçlar senkronizasyonun başarı ile gerçekleştiğini göstermektedir.

4.6.2. FPAA-Tabanlı MLC Sisteminin Kullanıldığı Senkronizasyon Gerçekleştirmeleri

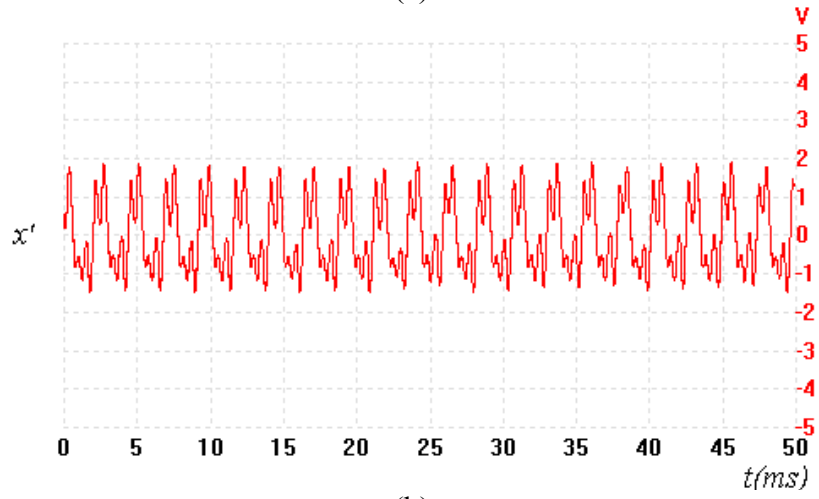
Bu kısımda FPAA-tabanlı MLC sisteminin kullanıldığı senkronizasyon gerçekleştirimleri, hem Pecora&Carroll yöntemi ile hem de tek yönlü kuplajlama yöntemi ile yapılmıştır.



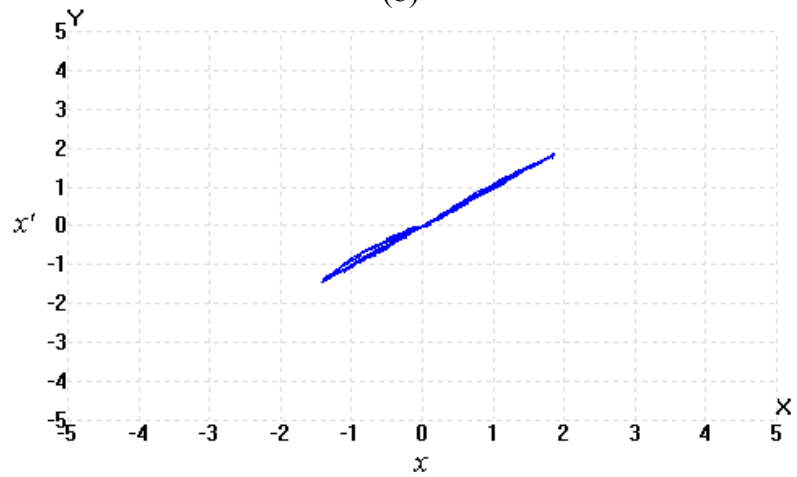
Şekil 4.39. FPAA-Tabanlı Chua Devresinin Kullanıldığı Kuplajlama Yöntemi İle Yapılan Senkronizasyon Gerçekleştirim Şeması.



(a)



(b)



(c)

Şekil 4.40. FPAA-Tabanlı Chua Devresinin Kullanıldığı Kuplajlama Yöntemi İle Yapılan Senkronizasyon Gerçekleştiriminden Elde Edilen, (a) $x(t)$ Kaotik Devre Dinamiği (b) $x'(t)$ Kaotik Devre Dinamiği, (c) $(x-x')$ Gösterimleri.

4.6.2.1. FPAA-Tabanlı MLC Sisteminin Kullanıldığı Pecora&Carroll Metodu İle Senkronizasyon Gerçekleştirimi

Bu kısımda MLC devresinin kullanıldığı Pecora&Carroll verici-alıcı senkronizasyon yöntemi ile senkronizasyon olgusu incelenmiştir. Bu yöntemde verici sisteme ait (x) dinamiği, alıcıya gönderilmiştir. (x) dinamiği ile sürülen MLC sisteminin FPAA' daki senkronizasyon gerçekleştirim şeması, Şekil 4.41' de verilmiştir. Bu gerçekleştirimde kullanılan verici sisteme ait durum denklemleri aşağıda yer almaktadır.

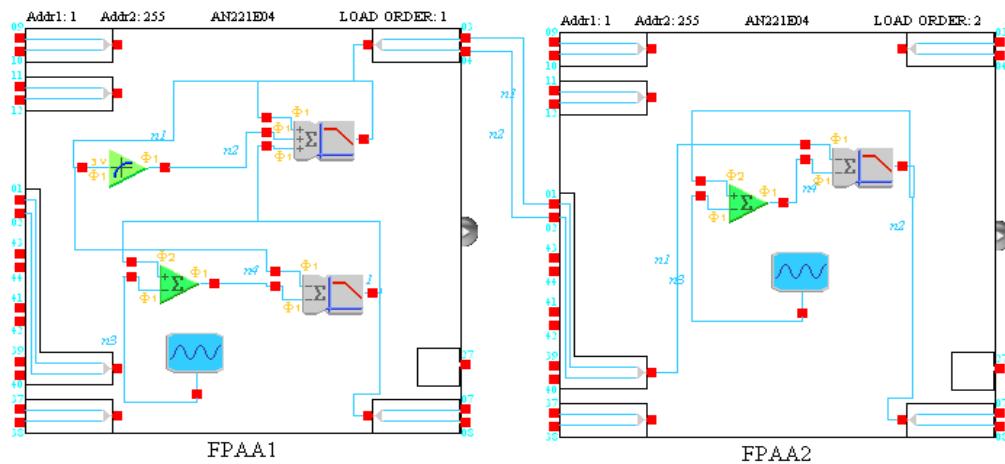
$$\begin{aligned} \dot{x} &= y - f(x) \\ \dot{y} &= -y - x - v y + F \sin(\omega t) \end{aligned} \quad (4.38)$$

Alıcı sisteme ait denklemler ise:

$$\dot{y}' = -y' - x - v y' + F \sin(\omega t) \quad (4.39)$$

Burada (x) dinamiği sistemi süren işarettir.

Bu gerçekleştirimde de Chua devresinin kullanıldığı senkronizasyon çalışmalarında olduğu gibi FPAA' nın sınırlı kapasitesinden dolayı 2 adet FPAA kullanılmıştır. Birinci FPAA uygulama kartı, FPAA1 olarak adlandırılmış olup, verici kısmı gerçekleştirir. İkinci FPAA uygulama kartı ise, FPAA2 olarak adlandırılmış olup alıcı kısmı gerçekleştirir. Verici kısımda MLC sisteminin kaotik dinamikleri oluşturulmuştur. Bu dinamiklerden (x) değişkeni alıcı kısma gönderilmiştir. Alıcı kısımda (y') dinamiği (x)

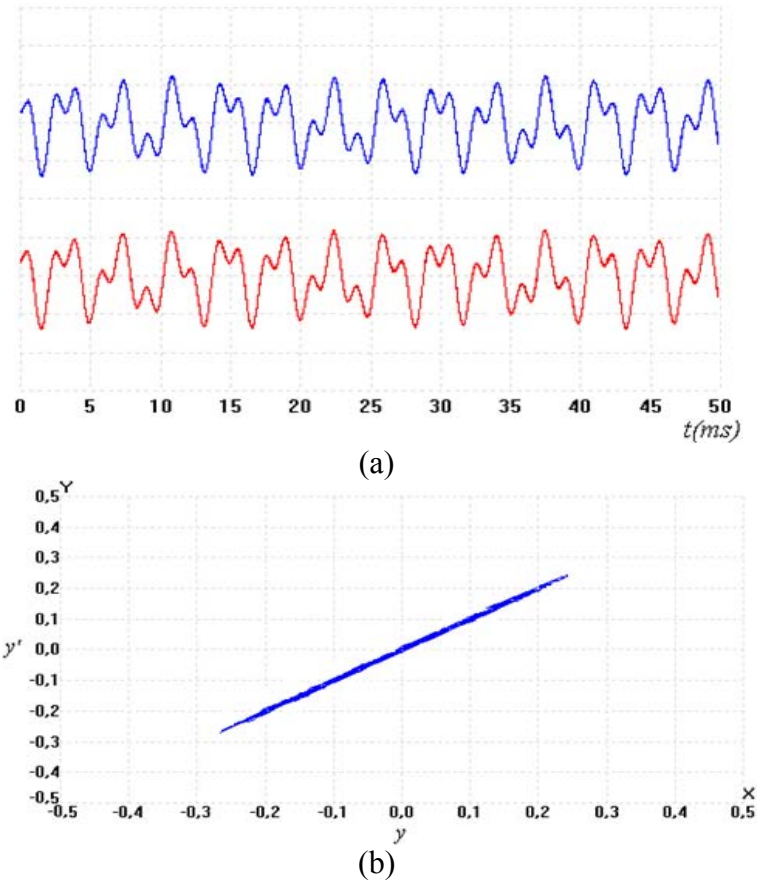


Şekil 4.41. FPAA-Tabanlı MLC Sisteminin Kullanıldığı Pecora&Carroll Yöntemi İle Yapılan Senkronizasyon Gerçekleştirim Şeması.

değişkenine göre elde edilmiştir. Elde edilen (y') dinamiği verici kısımdaki (y) dinamiği ile benzer işaretler üretip üretmediğine bakılmıştır. FPAA kullanılarak gerçekleştirilen verici ve alıcı sistemin kaotik dinamikleri ve senkronizasyon grafiği Şekil 4.42' de yer almaktadır. Grafikten görülüyor ki senkronizasyon başarı ile gerçekleştirilmiştir.

4.6.2.2. FPAA-Tabanlı MLC Sisteminin Kullanıldığı Tek Yönlü Kuplajlama Metodu İle Senkronizasyon Gerçekleştirimi

Tek yönlü kuplajlama metodunda alıcı sistemin davranışı, verici sistemin davranışına bağlı iken verici sistemin davranışı, alıcı sistemin davranışından etkilenmez. Bu kısımda tek yönlü kuplajlama metodu, iki MLC sistemini senkronize etmek için kullanılmıştır [134]. $[u(y)=y]$ tek yönlü kuplajlama parametresi aracılığı ile alıcı sistemi kuplajlayan verici sistemin durum denklemleri aşağıda verilmiştir.



Şekil 4.42. FPAA-Tabanlı MLC Sisteminin Kullanıldığı Pecora&Carroll Yöntemi İle Yapılan Senkronizasyon Gerçekleştiriminden Elde Edilen (a) Kaotik Devre Dinamikleri: Üstteki İşaret y (0.2 V/div), Altteki İşaret y' (0.2V/div), (b) ($y - y'$) Gösterimleri.

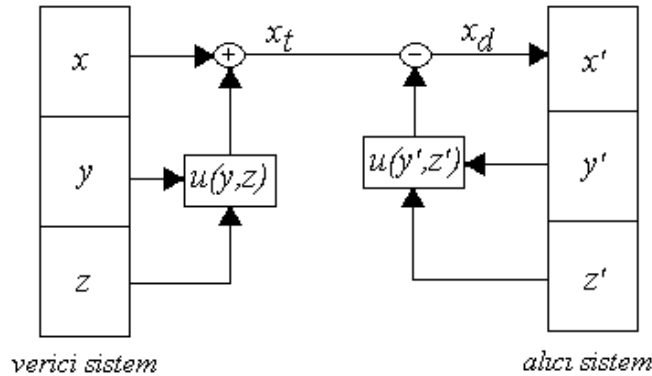
$$\begin{aligned}
\dot{x} &= y - f(x) \\
\dot{y} &= -y - x - v y + F \sin(\omega t) \\
x_t &= x + u(y) = x + y
\end{aligned} \tag{4.40}$$

Alıcı sistem:

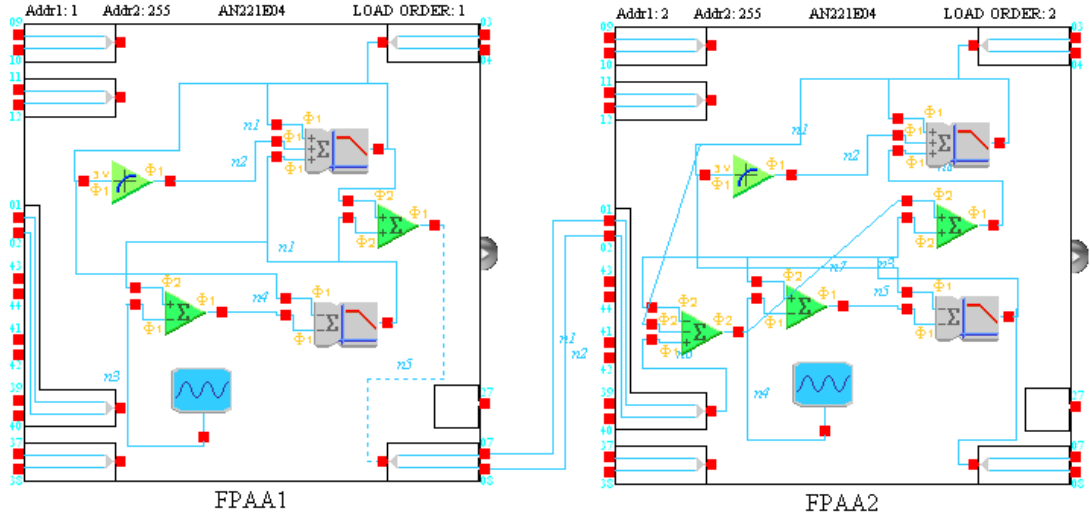
$$\begin{aligned}
x_d &= x_t - u(y') = x_t - y' \\
\dot{x}' &= y' - f(x') + \varepsilon(x_d - x') \\
\dot{y}' &= -y' - x' - v y' + F \sin(\omega t)
\end{aligned} \tag{4.41}$$

Burada (ε), kuplajlama parametresidir.

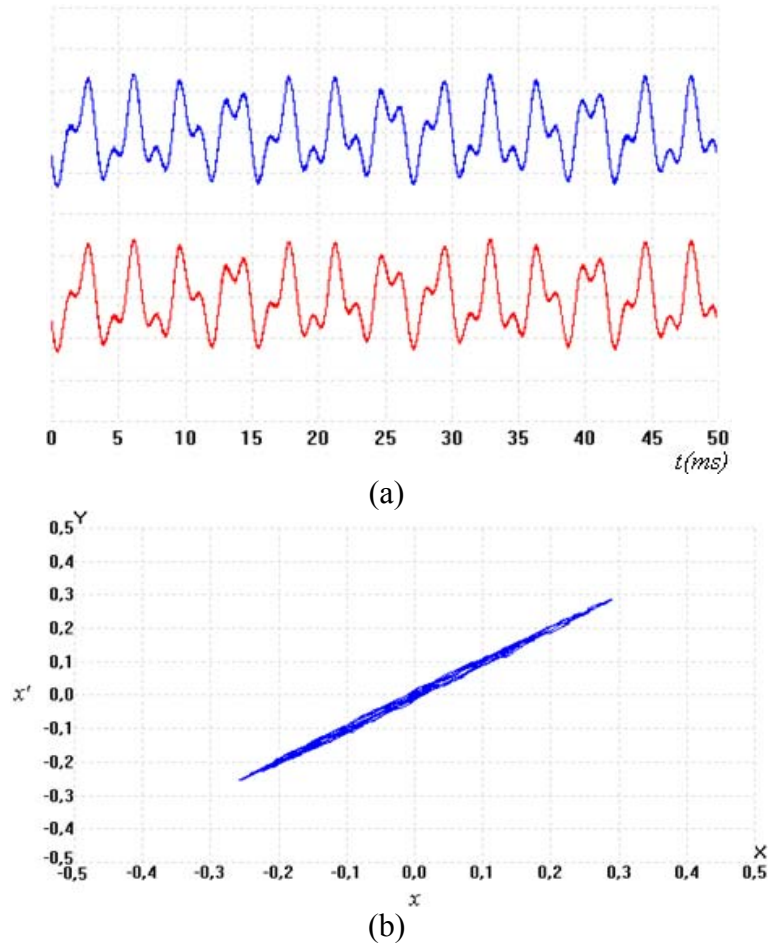
Tek yönlü kuplajlama metodu kullanılarak yapılan senkronizasyonun blok diyagramı Şekil 4.43' de gösterilir iken FPAA-tabanlı MLC sisteminin kullanıldığı senkronizasyon gerçekleştirim şeması, Şekil 4.44' de yer almaktadır. Bu gerçekleştirimde de FPAA1' de verici sistem, FPAA2' de alıcı sistem gerçekleştirilmiştir. Verici sistemde MLC sisteminin dinamikleri elde edilmiştir. Bunun yanı sıra *SUMDIFF* blok kullanılarak (x_t) taşıyıcı işareti oluşturulmuş ve alıcı sisteme gönderilmiştir. Alıcı sistemde Denklem (4.41)' de yer alan (x_d) ve $\varepsilon(x_d - x')$ ifadesi *SUMDIFF* bloklar kullanılarak elde edilmiştir. Bu ifadeler kullanılarak oluşturulan (x') ve (y') sistem dinamikleri ise *SUMFILTER* blokların çıkışlarından elde edilmiştir. Bu senkronizasyon yöntemine göre elde edilen sistem dinamikleri Şekil 4.45' de verilmiştir. Bu sonuçlar, tek yönlü kuplajlama metodu ile kaotik senkronizasyonun başarı ile sağlandığı görülmüştür.



Şekil 4.43. MLC Sisteminin Kullanıldığı Tek Yönlü Kuplajlama Yöntemi İle Yapılan Senkronizasyonda Kullanılan Blok Diyagram.



Şekil 4.44. FPAAs-Tabanlı MLC Sisteminin Kullanıldığı Tek Yönlü Kuplajlama Yöntemi İle Yapılan Senkronizasyon Gerçekleştirme Şeması.



Şekil 4.45. FPAAs-Tabanlı MLC Sisteminin Kullanıldığı Tek Yönlü Kuplajlama Yöntemi İle Yapılan Senkronizasyon Gerçekleştiriminden Elde Edilen (a) Kaotik Devre Dinamikleri: Üstteki İşaret x (0.2 V/div), Altındaki İşaret x' (0.2 V/div), (b) $(x - x')$ Gösterimleri.

5. BÖLÜM

FPAA YAZILIM UYGULAMASI

5.1. Giriş

AnadigmDesigner 2, FPAA tasarımında kullanılan ve Anadigm firması tarafından geliştirilen FPAA arayüz programıdır. Bu program içerisinde tasarımın yapılmasına kolaylık sağlamak ve yardımcı olmak amacıyla *AnadigmFilter* ve *AnadigmPID* gibi yardımcı araçlar (tool) içermektedir. Bu yardımcı araçlardan *AnadigmFilter*, yüksek dereceli filtrelerin tasarımı için kullanılırken *AnadigmPID*, kapalı döngü kontrolör tasarımına yardımcı olmak için kullanılırlar. Bu yardımcı araçlara benzer şekilde kaotik sistem tasarımıyla uğraşan araştırmacılara ve bu konuyla ilgili eğitim gören lisans ve lisansüstü öğrencilerine dönük olarak bu tez kapsamında AnadigmCHAOS ismiyle bir yazılım uygulaması geliştirilmiştir. Bu yazılım uygulaması AnadigmDesigner 2 yazılımına göre uyarlanmış olup, yardımcı araçlar içinde yer alması düşünülmüştür. AnadigmCHAOS yardımcı aracı ile 3. ve 4. Bölümlerde incelenen kaotik yapılardan literatürde üzerinde en çok çalışılanlarını tek bir kullanıcı arayüzüyle hem nümerik analizleri yapılabilmekte hem de FPAA-tabanlı tasarımları FPAA programına gönderilebilmektedir. Ayrıca bu kaotik yapılara kendi parametre değerleri dışında başka parametre değerleri verilebilmekte ve bu parametrelere göre de nümerik analizlerinin yapılabilmesi kolaylığı sağlanmaktadır. Bu arayüzün tasarımı, MATLAB programı altında çalışan grafiksel kullanıcı arayüzü (Graphic User Interface, GUI) kullanılarak gerçekleştirilmiştir. GUI, bilgisayarda gerçekleştirilen işlemlerin veya uygulamaların bir bütün olarak çalışmasını sağlayan ve görsel özellikleri olan bir arayüz programıdır.

AnadigmCHAOS yardımcı aracın arayüzü verilmeden önce FPAA' nın arayüz programı olan AnadigmDesigner 2' den bahsedilecektir.

5.2. AnadigmDesigner 2 Arayüzü ve Yardımcı Araçları

Anadigm firmasının FPAAsı için geliştirilen AnadigmDesigner 2 arayüzü, tasarımcıya dinamik olarak yeniden yapılandırılabilir analog devrelerin tasarlanması ve gerçekleştirilmesi imkânını veren bir yazılım uygulamasıdır. Programda tanımlanmış olan ve farklı fonksiyonları gerçekleştirebilen bloklara yapılandırılabilir analog modüller (**Configurable Analog Module, CAM**) adı verilir ve bu CAM'lerin listesi Şekil 2.14' de verilmiştir. Tasarımcı, CAM bloklarından hangisini kullanacağına karar verdikten ve seçtiği CAM blokların parametre değerlerini ayarladıktan sonra, arayüzde *sürükle-bırak* yöntemiyle devresini tasarlayabilmektedir. AnadigmDesigner 2, tasarımcının tasarladığı devreyi test etmesi için zaman domeni fonksiyonuna sahip bir simülasyon ihtiva etmektedir. Fakat bu simülasyon kaotik osilatör yapılarının testi için kullanılamamaktadır. Bunun sebebi, simülasyonun test yapabildiği için tasarlanan yapıya dışarıdan bir kaynak bağlanması gerekliliğidir. Kaotik osilatör yapılarında ise dışarıdan kaynak kullanılmadığından dolayı simülasyon hata vermekte ve istenilen testi yapamamaktadır. Kaynak olarak, CAM modüllerinde bulunan voltaj kaynaklarından biri verildiğinde de simülasyon hata vermektedir. Tasarımcının analog devre tasarımı konusunda uzman olup olmaması, karmaşık bir analog sistemi hızlı bir şekilde yapmasına, simüle etmesine, test ve doğrulama için FPAAsı kartına göndermesine engel değildir. AnadigmDesigner 2, tasarlanan herhangi bir sistem FPAAsı uygulama kartında çalışırken sistem parametrelerinin değiştirilmesine veya o anda başka bir sistemin yüklenmesine izin veren elektronik tasarım otomasyonu (**Electronic Design Automation, EDA**) ürünüdür [6]. Bu program, gerçekleştirilen bir tasarımı istenildiğinde gömülü bir sistemdeki mikro işlemci tarafından kontrol edilebilen ve ayarlanabilen C-koduna çevirebilir. AnadigmDesigner 2'nin görünümü daha önceden Şekil 2.13' de verilmiştir. AnadigmDesigner 2 programı *AnadigmFilter* ve *AnadigmPID* olmak üzere iki adet yardımcı araç içermektedir [6]. Bu yardımcı araçları ile ilgili açıklayıcı bilgiler aşağıda verilecektir.

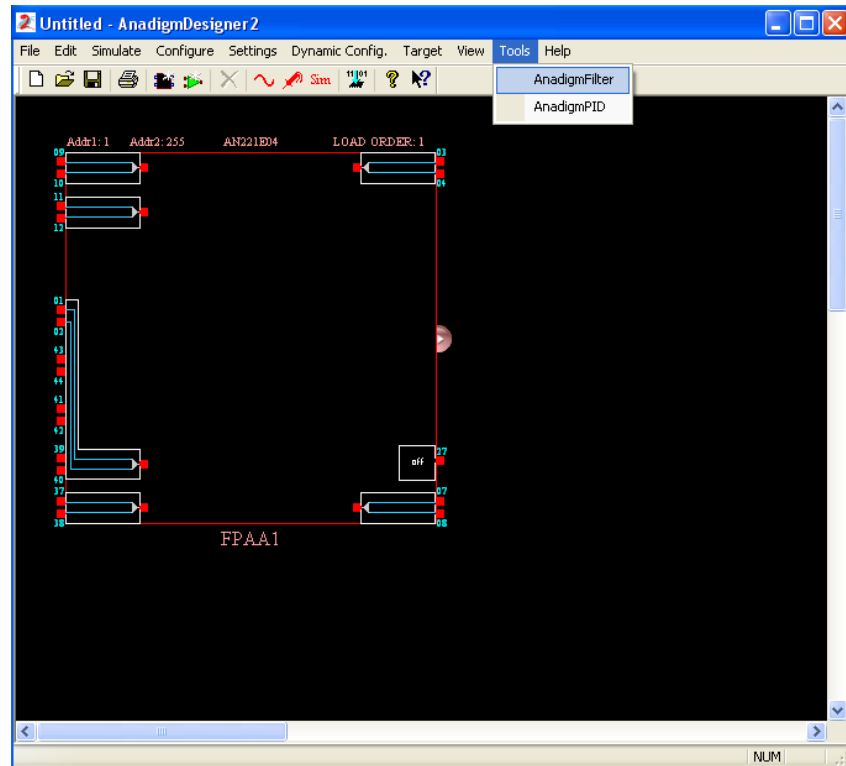
5.2.1. AnadigmDesigner 2 Arayüzünde Filtre Tasarım Aracı: *AnadigmFilter*

AnadigmDesigner 2 içerisinde bulunan *AnadigmFilter* isimli yardımcı araç menüsü, yüksek dereceli filtrelerin tasarımı için güçlü bir tasarım yardımcısıdır. Standart CAM kütüphaneleri, köşe frekansının, kazancının ve Q kalite faktörünün tasarımcı tarafından

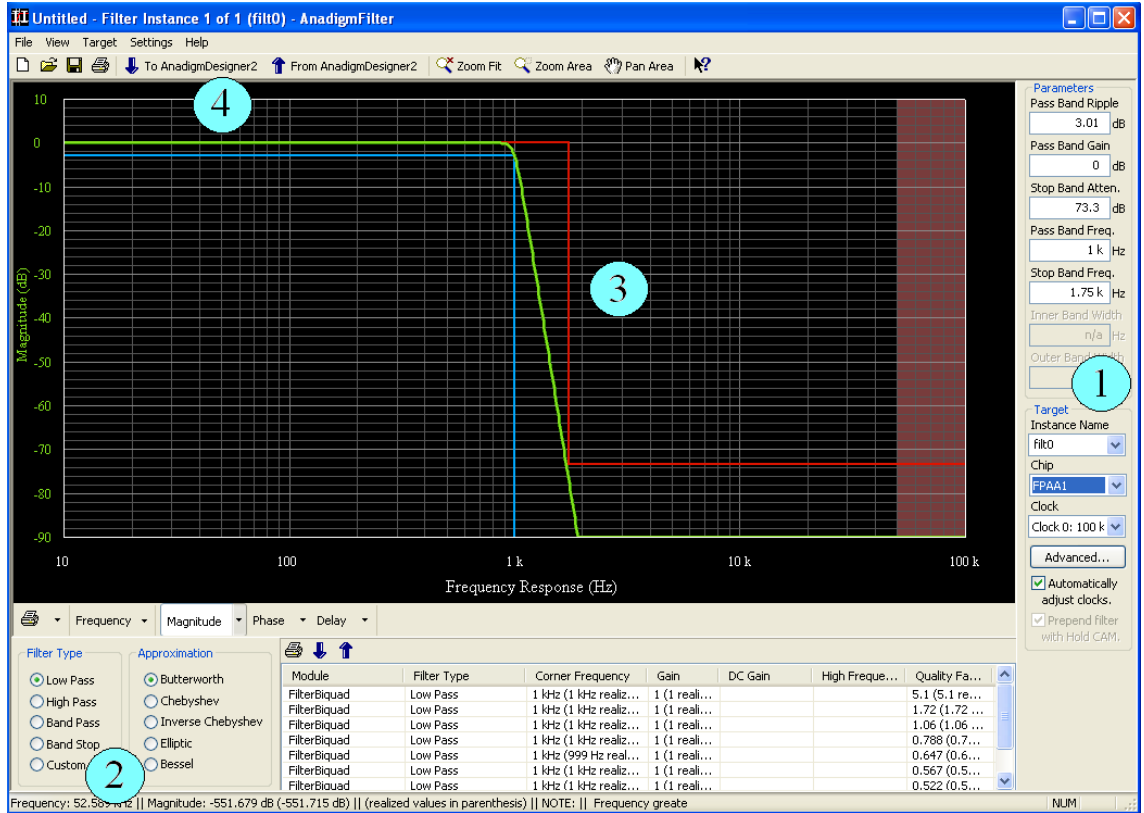
belirlendiği birinci ve ikinci dereceden filtreleri içermektedir. Program kütüphanesinde bulunan bu standart filtre elemanları, yüksek dereceden filtre gerçekleştirmek için kaskad bağlanabilir. Kaskad bağlantı beraberinde, bazı hesaplamaları ve filtre bloklarının kullanımını gerektirmektedir. Buna alternatif olarak *AnadigmFilter* aracı kullanılarak, yüksek dereceden filtrelerin tasarım ve gerçekleştirimi komple bir sistem olarak yapılabilir. Bu araca Şekil 5.1’ de görüldüğü gibi *AnadigmDesigner 2* programında *Tools* menüsünden *AnadigmFilter* seçimi ile ulaşılabilir [6].

AnadigmFilter aracının ana ekranı Şekil 5.2’ de yer almakta olup bu ekran, tasarımda kullanılacak tasarım parametrelerinin hepsini içermektedir. Şekil 5.2’ de verilen 1-4 arasındaki rakamlar izlenecek işlem adımlarını tarif etmektedir. Bu işlem adımları aşağıda açıklanmıştır.

1. Zamanlama işareti, kırkımık gibi FPAA’ nın özellikleri belirlenir.
2. Kullanılacak olan filtrenin tipi seçilir (alçak geçiren, yüksek geçiren, band geçiren, band durduran).



Şekil 5.1. AnadigmDesigner 2 Programında *AnadigmFilter* Yardımcı Aracının Seçimi.

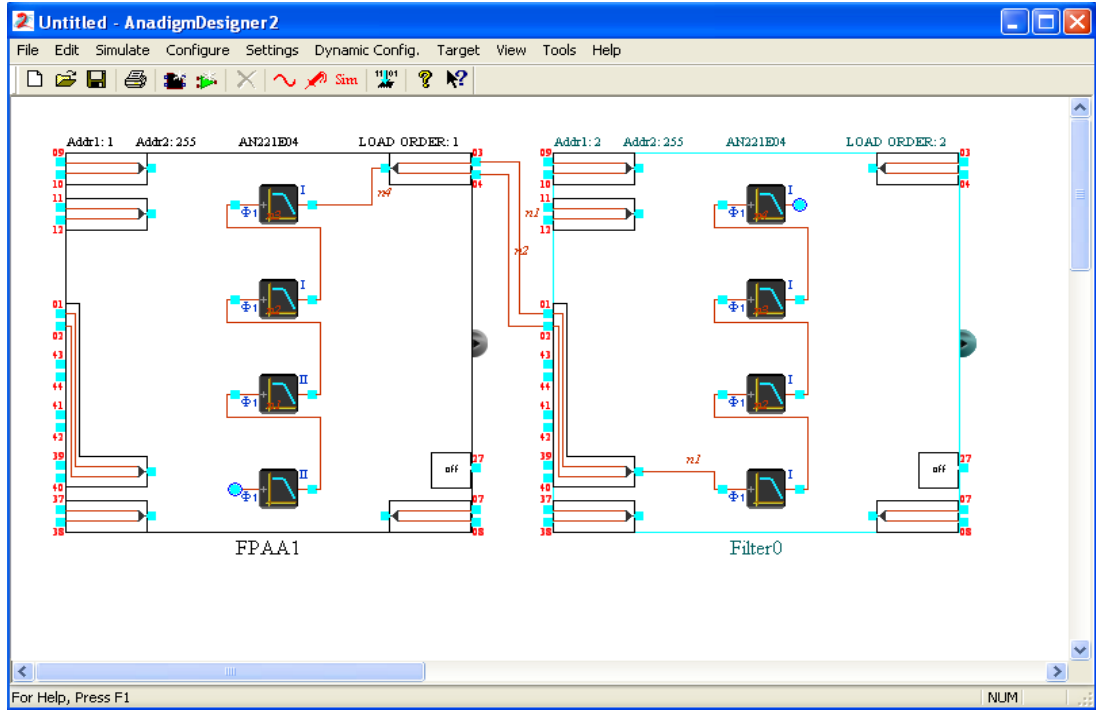


Şekil 5.2. *AnadigmFilter* Aracının Ekran Görünümü.

3. Frekans ve kazanç ayarlamaları için filtre karakteristiği belirlenir.
4. “To AnadigmDesigner 2” sekmesine tıklandığında tasarlanan yapı FPAA uygulama modülüne aktarılır.

Böylelikle *AnadigmFilter* ortamında genel hatlarıyla düşünülen filtre tasarlanmış ve *AnadigmDesigner 2*’ye gönderilmiş olur. Bu yardımcı arayüzde filtrelerin yaklaşım tipinin (Butterworth, Chebyshev, ters Chebyshev, Elliptical ve Bessel) seçimine izin verilir. Ekranın alt tarafında yer alan alanda ise tasarımı yapılan filtrede kullanılan bloklar, bu blokların köşe frekansları, kazançları ve kalite faktörleri gibi parametreler gösterilir [6].

Şekil 5.2’de örnek bir tasarım yapılmış olup bu tasarımda 8. dereceden Butterworth tipi alçak geçiren filtre gerçekleştirilmiştir. Seçilen parametre değerlerine ve filtre tipine uygun olarak *AnadigmDesigner 2* programı tarafından filtrenin blokları otomatik olarak seçilmektedir. Oluşturulan yapı Şekil 5.3’de verilmiştir. Bu özel filtre, 2 adet FPAA’nın kullanımını gerektirmektedir. Farklı filtre ve yaklaşım tipleri kullanılarak tek FPAA’da filtre gerçekleştirilebilir [6].



Şekil 5.3. AnadigmFilter Aracında Tasarlanan Filtrenin AnadigmDesigner 2 Arayüz Ortamına Aktarılması.

5.2.2. AnadigmDesigner 2 Arayüzünde Kontrolör Tasarım Aracı: *AnadigmPID*

AnadigmPID aracı, kapalı döngü kontrolör tasarımına yardımcı bir programdır. Bu programlama aracı, P, PI, PD ve PID formlarını içeren kapalı döngü kontrol devre topolojileri oluşturmak için orantılı (**P**roportional, P), integral alıcılı (**I**ntegral, I) ve türev alıcılı (**D**erivative, D) yapı bloklarını kullanır. *AnadigmPID* aracında gerçekleştirilen kontrolör devre tasarımı, otomatik olarak AnadigmDesigner 2 arayüzüne aktarılır [6].

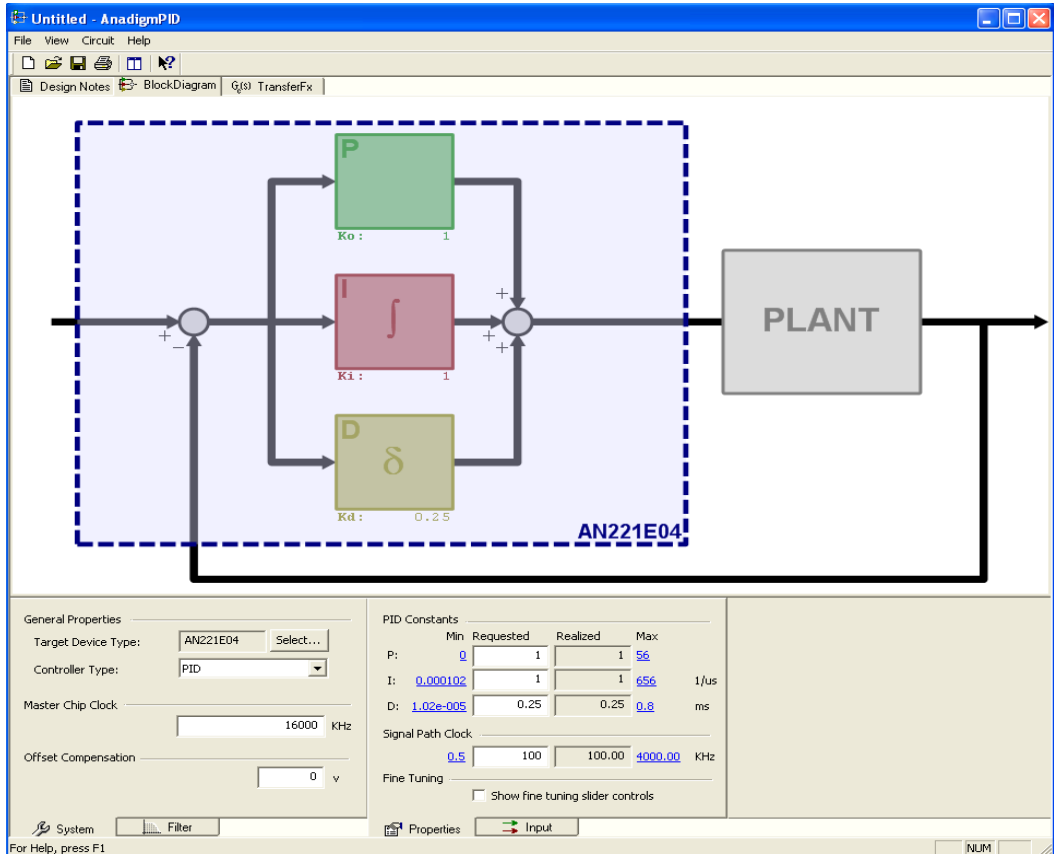
Kapalı döngü kontrol devresinin tasarımı, AnadigmDesigner2’ deki yeni bir devrenin açılmasıyla başlar. *AnadigmPID*, *Tools* kısmından *AnadigmPID* menüsü seçilerek çağırılır. *AnadigmPID* program aracının ekranı açıldığında öncelikle hangi tip FPAA’ nın kullanılacağı seçilir. *AnadigmPID* ekranının üst yarı kısmında tasarım bilgileri (*Design Notes*), blok diyagramı (*Block Diagram*) ve transfer fonksiyonu (*TransferFx*) sekmelerini içeren ekran bulunmaktadır.

Tasarım bilgileri kısmında kısaca FPAA’ daki PID tasarımdan bahsedilmektedir. Blok diyagramı sekmesi seçildiğinde PID kontrol şeması ekrana gelmektedir. Transfer fonksiyonu kısmında ise PID kontrole ait transfer fonksiyonu verilmiştir. Örnek olarak

Şekil 5.4' de blok diyagram seçilmiş ve PID kontrol şeması verilmiştir. *AnadigmPID* ekranının en alt kısmında sekmelerin bulunduğu bir ekran bulunmaktadır. Bu sekmeler: *Sistem (System)*, *Filtre (Filter)*, *Özellikler (Properties)* ve *Giriştir (Input)* [6].

Ekran görünümü Şekil 5.5(a)' da verilen *Sistem* sekmesi (*System*) sayesinde hedef FPAA tipi seçilir. Burada AN221E04 tipi FPAA seçilmiştir. İstenildiğinde *Select* tuşuna basılarak FPAA tipi değiştirilebilir. Bu sekmede ana kırmık zamanlama işareti, FPAA için analog zamanlama işaretinin frekansını ayarlar. Kontrolör tipi kısmından ise P, PI, PD ve PID kontrolörlerinden birinin seçilmesi işlemi yapılır. Dengeleme (*Offset Compensation*) kısmında ise kontrol devresinin çıkışına DC dengeleme değeri eklenir [6].

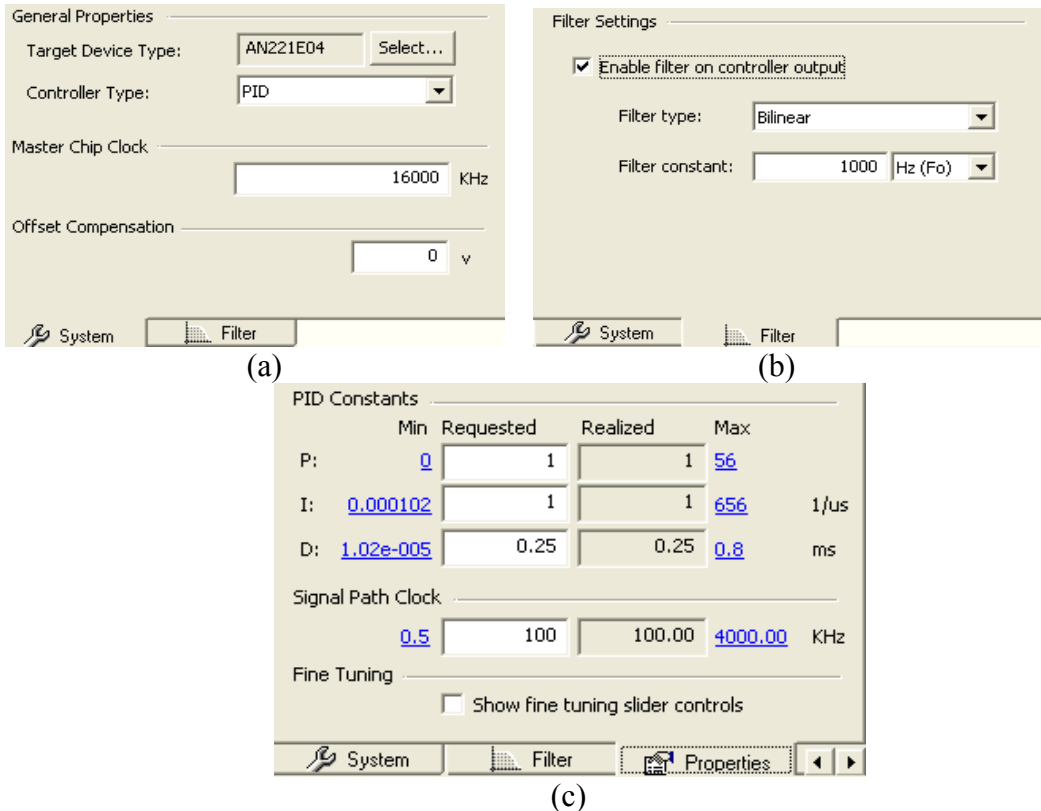
Bir diğer sekme olan *Filtre (Filter)*, kontrolörün çıkışında alçak geçiren filtrenin var olup olmadığını kontrol eder ve filtre parametrelerini ayarlar. Filtre tipi olarak bilinear veya biquadratic seçilir. Filtre sabiti, Hz birimine sahip köşe frekansıdır. *Filtre* sekmesine ait ekran Şekil 5.5(b)' de yer almaktadır [6].



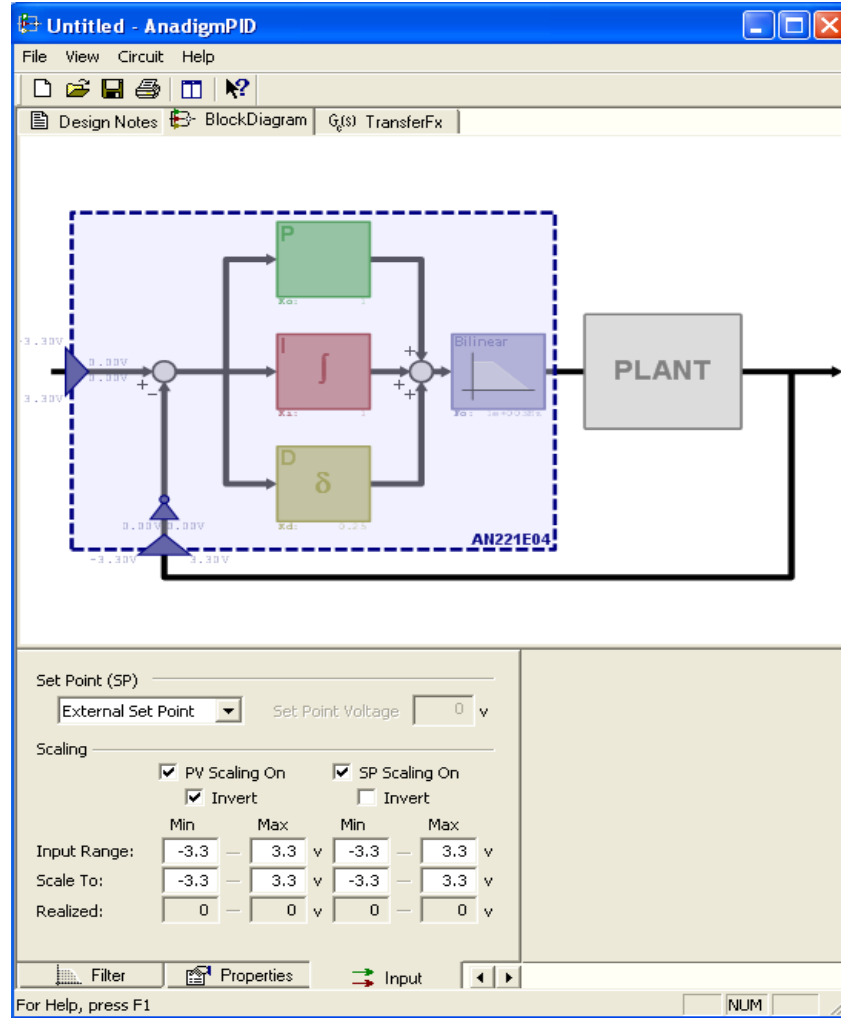
Şekil 5.4. Blok Diyagram Sekmesi Seçildiğindeki AnadigmPID Ekranı.

Özellikler sekmesi (*Properties*) sayesinde kontrolör devresinin her bir kolundaki P, I ve D sabitlerin değerleri ayarlanabilir. Gerçek devrede, K_P , K_I ve K_D sabitlerine karşılık burada kontrolörlerin baş harfleri ile tanımlanan parametrelerin her biri için istenilen değerler, tasarımcı tarafından program ekranına girilirken *AnadigmPID*' nin elde edebileceği değerler ise hemen yan kısımda belirtilmiştir. Sabitlerin her birisinin alabileceği değer aralığı, yine bu ekranda belirtilmektedir. Sinyal yolu zamanlama işareti (*Signal Path Clock*) kısmında tasarımcı tarafından tayin edilen ve buna karşılık kullanılan CAM' lerin her birinden alınan frekans değerleri bulunmaktadır. *Özellikler* kısmına ait ekran çıktısı Şekil 5.5(c)' de verilmiştir [6].

Giriş sekmesi ise kontrolör giriş devresinin birkaç açıdan kontrolüne izin verir. Ölçeklendirme ve/veya tersine çevirme işlemi için PV işareti (kontrol edilecek sistemden alınan geribesleme işareti) kullanılabilir. Benzer şekilde ölçeklendirme ve/veya tersine çevirme işlemi, ayarlama noktası işareti (SP) için de kullanılabilir. *Giriş* sekmesinde ayarlamalar yapılarak oluşturulan PID tasarımın blok diyagramı Şekil 5.6' da yer almaktadır [6].



Şekil 5.5. *AnadigmPID* Aracında Yer Alan, (a) Sistem Ekran Görünümü, (b) Filtre Ekran Görünümü, (c) Özellikler Ekran Görünümü.

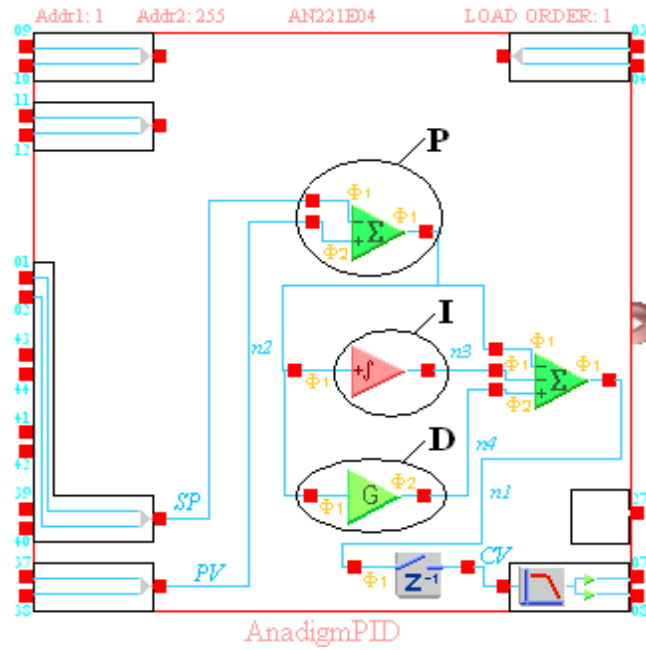


Şekil 5.6. Giriş Sekmesinde Ayarlamalar Yapılarak Oluşturulan PID Tasarımın Blok Diyagram Ekranı.

AnadigmPID' de tasarlanan devre, otomatik olarak *AnadigmDesigner 2*' ye transfer edilir ve transfer edilen tasarımın FPAA devre şeması Şekil 5.7' de verilmiştir. Bu devre gerçekleştiriminde yer alan SP, PV ve CV, sırasıyla ayarlama noktasının, kontrol edilen sistem geribeslemesinin ve kontrolör çıkışının kısaltmasıdır [6].

5.3. AnadigmDesigner 2 Arayüzünde Kaotik Sistem Tasarım Aracı: AnadigmCHAOS

Bu tez çalışmasının yazılım uygulaması kapsamında *AnadigmDesigner 2* arayüzünün araçlar kısmında yer alan *AnadigmFilter* ve *AnadigmPID* gibi yardımcı araçlarına ek olarak *AnadigmCHAOS* isminde üçüncü bir yardımcı araç arayüzünün gerçekleştirimi düşünülmüştür. Bu arayüz sayesinde kaotik osilatörlerle ilgilenen araştırmacılara, bilim



Şekil 5.7. *AnadigmPID*' de Tasarlanan Kontrolörün AnadigmDesigner 2' ye Aktarımı.

adamlarına, lisans ve lisansüstü öğrencilerine FPAA-tabanlı faydalı bir eğitim ve tasarım aracı sunulmuş olacaktır. Bu amaç doğrultusunda, AnadigmCHAOS isimli arayüz programı tasarlanmıştır. Bu arayüz grafiksel kullanıcı arayüzlerinden biri olan MATLAB™ GUI kullanılarak gerçekleştirilmiştir. GUI sayesinde bilgisayarda gerçekleştirilen işlemler veya uygulamalar bir bütün olarak çalışmakta ve elde edilen sonuçlar görsel öğelerle kullanıcıya aktarılabilmektedir. Ayrıca bu arayüz sayesinde gerçekleştirilen uygulama, kullanıcıya net ve anlaşılır bir program olarak sunulmuş olur [145-148]. GUI arayüzüne MATLAB™ penceresinde iken *File* → *New* → *Gui* adımlarını kullanarak, komut penceresine “*guide*” yazılarak veya MATLAB™ ana penceresi üzerinde yer alan GUI ikonuna tıklanarak ulaşılabilir. MATLAB™ GUI' nin özellikleri:

- MATLAB™ GUI içerisinde radyo butonları, liste kutuları, değiştirilebilir veya sabit metin kutuları, çoklu seçim düğmeleri, kaydırıcılar v.b. nesnelere yer almaktadır. Bunların yanı sıra MATLAB™ GUI ile MATLAB™' in sahip olduğu hesaplama potansiyeli, dışarıdan veri transferi ve grafiksel çizimler gibi pek çok özellik tasarımcının hizmetine sunulmaktadır.
- MATLAB™ GUI yüzeyi, görsel panel üzerinde kullanılacak elemanların ve bu elemanların bilgilerinin tutulduğu yerdir.

- MATLAB™ GUI yüzeyinde yer alan nesnelere işlevleri, arka planda çalışan m dosyası veya m fonksiyonu içerisinde tarif edilir. GUI’ de seçilen her bir nesnenin MATLAB™, da komut satırları bulunmaktadır.

MATLAB GUI kullanılarak geliştirilen AnadigmCHAOS uygulaması aşağıda ayrıntılı şekilde tanıtılmıştır.

Üçüncü bölümde ele alınan kaotik osilatör devreleri, hem ayırık elemanlarla deneysel olarak gerçekleştirilmiş hem de diferansiyel denklem takımları ile nümerik analizleri yapılarak incelenmiştir. Sonraki bölümde ise kaotik osilatör devrelerinin FPAA tasarımları üzerinde durulmuştur. AnadigmDesigner 2 programındaki yardımcı araçlara eklenebilecek bir yazılım aracı olarak geliştirilen AnadigmCHAOS, kaotik sistem tasarımı ile ilgilenen araştırmacılara ve öğrencilere faydalı bir eğitim ve tasarım aracıdır. Bu hedef doğrultusunda kaotik osilatör yapılarının hem kod-tabanlı nümerik analiz gerçekleştiriminin yapıldığı hem de FPAA-tabanlı tasarımlarını FPAA programına gönderilebildiği kullanıcı arayüzlü bir uygulama GUI’ de gerçekleştirilmiştir. Ayrıca bu arayüzde incelenen kaotik osilatörlerin önceden tayin edilmiş parametre değerlerinin yanı sıra kullanıcının belirleyebileceği farklı parametre değerleriyle de nümerik analizi yapılabilmektedir. Arayüzün bir diğer özelliği de istenilen kaotik üreteç seçilerek “To AnadigmDesigner 2” sekmesiyle, FPAA arayüzü AnadigmDesigner 2’ de seçilen üreticinin önceden hazırlanmış olan FPAA dosyası açılmaktadır. Bu arayüz sayesinde kaotik osilatörlerle ilgilenen araştırmacılar, bilim adamları, lisans ve lisansüstü öğrenciler tek bir platformda kaotik osilatörlerin nümerik analizini yaparken, kolayca sistem parametrelerini değiştirebilmekte ve bu değişimlerin osilatörün davranışını nasıl etkilediğini görebilmektedirler. Ayrıca araştırmacıların aynı kaotik osilatörlerin FPAA tasarımlarına ulaşmalarını sağlamak için faydalı bir kaynak sunulmaktadır. Yapılan arayüzün ekran görüntüsü Şekil 5.8’ de gösterilmektedir.

Bu tez çalışması kapsamında hazırlanan ve AnadigmCHAOS olarak adlandırılan yardımcı araç arayüzü ile ilgili genel özellik aşağıda özetlenmiştir:

- Programdaki “*Chaotic Oscillator*” kısmında osilatör olarak Chua, Colpitts, Duffing, Kennedy, Lorenz, MLC, Rössler ve Sprott osilatörleri yer almaktadır.

AnadigmCHAOS

Chaotic Oscillator's Equation

$$dx/dt = k \cdot (y - x - f(x))$$

$$dy/dt = x - y + z$$

$$dz/dt = -d \cdot y$$

$$f(x) = a \cdot x - b \cdot x^3$$

Seçilen kaotik osilatörün denklemler takımını

Nümerik analizden elde edilen sonuçların zaman domeninde ve faz-uzay düzleminde grafiksel gösterimini

Function Type

Cubic

Piecewise Linear

Piecewise Quadratic

Trigonometric

Doğrusal olmayan fonksiyon tipleri

Chaotic Oscillator

Chua's Oscillator

Colpitts Oscillator

Duffing Oscillator

Kennedy's Oscillator

Lorenz Oscillator

MLC System

Rossler Oscillator

Sprott's System

Kaotik Osilatörler

Parameters

k: 10

d: 14.87

a: -1.27

b: -0.0157

c: 0

F: n

Seçilen konfigürasyonun varsayılan parametreleri

To AnadigmDesigner2

Seçilen konfigürasyon AnadigmDesigner 2'ye aktarılır

TEST

Seçilen konfigürasyonun nümerik analizini yapar

Modified Oscillator

Seçilen konfigürasyonun değişen parametreleri ile nümerik analizini yapar

PRINT

Arayüzün çıktısı alınır

EXIT

Arayüzden çıkış yapılır

Şekil 5.8. AnadigmCHAOS Yardımcı Aracın Arayüzü.

- Kullanıcı osilatör tipini seçtikten sonra eğer varsa seçilen osilatör tipi için farklı doğrusal olmayan fonksiyon tiplerini “*Function Type*” kısmından seçebilir.
- Kullanıcı seçim işlemini tamamladığı anda seçtiği osilatörün denklem takımı program arayüzünde görülmektedir.
- Seçilen konfigürasyonun varsayılan bir parametre grubu mevcut olup bu değerler, “*Parameters*” kısmında gösterilmektedir.
- Bu işlemlerden sonra kullanıcı, “*To AnadigmDesigner2*” sekmesine tıkladığı takdirde seçtiği konfigürasyon, FPAA’ nın programlanmasında kullanılan AnadigmDesigner 2’ ye aktarılır.
- Kullanıcı, “*TEST*” sekmesine tıkladığında seçtiği osilatör ve doğrusal olmayan fonksiyon tipine göre “*Parameters*” kısmında tanımlı mevcut parametre değerlerini kullanarak nümerik analiz yapabilmektedir. Elde edilen sonuçlar arayüzün ilgili kısmında zaman domeninde ve faz-uzay düzleminde grafiksel olarak kullanıcıya sunulmaktadır.
- Kullanıcı, varsayılan parametre değerlerini değiştirebilmekte, “*Modified Oscillator*” sekmesine tıklayarak değişen parametre değerleri ile nümerik analiz yapabilmekte ve bu değişimlerin osilatörün davranışı üzerindeki etkilerini görebilmekte ve sonuçları, arayüz programının üzerinde grafiksel olarak alabilmektedir.
- Kullanıcı “*PRINT*” sekmesine tıkladığı takdirde arayüz programının o anki görüntüsünü yazıcıdan çıktı olarak alabilmektedir.
- Kullanıcı “*EXIT*” sekmesine tıkladığında arayüz programı kapanmaktadır.

AnadigmCHAOS arayüzüne ait iki örnek aşağıda verilmiştir. Birinci örnekte Şekil 5.9’ da gösterildiği gibi “*Chaotic Oscillator*” kısmından *Chua Osilatörü* ve “*Function Type*” kısmından *Trigonometric* fonksiyonu seçilmiştir. Şekilden görüldüğü gibi seçilen osilatörün varsayılan parametre değerleri ve denklemleri arayüzde görülmektedir. Bu seçimlere ilişkin yazılım kodları Tablo 5.1 ve 5.2’ de verilmiştir.

Kullanıcı, “*TEST*” sekmesine tıkladığında arayüz, trigonometrik fonksiyonla gerçekleştirilen Chua osilatörünün mevcut parametre değerleriyle nümerik analizini yapar. Elde edilen sonuçların hem zamana hem de birbirlerine göre değişimleri, arayüz

AnadigmCHAOS

Chaotic Oscillator's Equation

$$dx/dt = k \cdot (y - x - f(x))$$

$$dy/dt = x - y + z$$

$$dz/dt = -d \cdot y$$

$$f(x) = -a \cdot \tanh(b \cdot x)$$

Chaotic Oscillator

- Chua's Oscillator
- Colpitts Oscillator
- Duffing Oscillator
- Kennedy's Oscillator
- Lorenz Oscillator
- MLC System
- Rossler Oscillator
- Sprott's System

Function Type

- Cubic
- Piecewise Linear
- Piecewise Quadratic
- Trigonometric

To AnadigmDesigner2

Parameters

k	10
d	5.6
a	2
b	0.38
c	0
F	0

TEST

Modified Oscillator

PRINT

EXIT

Şekil 5.9. Örnek Olarak Trigonometrik Fonksiyona Sahip Chua Osilatörün Seçilmesi.

Tablo 5.1. AnadigmCHAOS Arayüzünde “Chaotic Oscillator” Kısmından Chua Osilatörünün Seçimi İçin Hazırlanan Kodlar.

```

function generator_kontrol_SelectionChangeFcn(hObject, eventdata)
% Chaotic Oscillator kismından seçim yapılabilmesi için gerekli fonksiyon dosyası.
handles = guidata(hObject); % Handles yapı değişkenini yeni değerlerle günceller.
switch get(eventdata.NewValue, 'Tag') % Seçili nesnenin Tag bilgisini alma
    case 'chua' % chua osilatörü seçili ise yapılacak işlemler
        sec_generator=1; % Chaotic oscillator kısmında birinci üreteç seçilmiş olmalı
        disp('Üreteç Seçimi: Chua Devresi') % Üreteç Seçimi: Chua Devresi şeklinde
        %göster.
        handles = guidata(hObject); % Handles yapı değişkenini yeni değerlerle günceller.
        set(handles.type, 'Visible', 'on') %Function type kısmını görünür yapar.
        set(handles.cub, 'String', 'Cubic') %Function type kısmında kübik fonksiyonu
        %görünür yapar.
        set(handles.piece_lin, 'String', 'Piecewise Linear') %Function type kısmında parçalı
        % doğrusal fonksiyonu görünür yapar.
        set(handles.piece_qua, 'String', 'Piecewise Quadratic') %%Function type kısmında
        % parçalı ikinci dereceden fonksiyonu görünür yapar.
        set(handles.trigon, 'String', 'Trigonometric') %Function type kısmında trigonometrik
        %fonksiyonu görünür yapar.
        save generator_secimi sec_generator % Yukarıdaki satırlarda yapılanları
        %kaydeder.
        guncelle; % guncelle denklemini çalıştırır.
        denklemler_guncelle; % denklemler_guncelle denklemini çalıştırır.

```

üzerinde grafiksel olarak kullanıcıya sunulmaktadır ve bu sonuçlar, Şekil 5.10’ da verilmektedir. Bu TEST işleminin yapılabilmesi için gerekli kodlar Tablo 5.3’ de yer almaktadır.

Tablo 5.2. AnadigmCHAOS Arayüzünde “Function Type” Kısmından *Trigonometric* Fonksiyonunun Seçimi İçin Hazırlanan Kodlar.

```

function type_kontrol_SelectionChangeFcn(hObject, eventdata)
% Function type kısmından seçim yapılabilmesi için gerekli fonksiyon dosyası.
load generator_secimi % Seçilen osilatör tipini yükler.
load type_secimi % Seçilen fonksiyon tipini yükler.
handles = guidata(hObject);
switch get(eventdata.NewValue,'Tag')
case 'trigon' % trigonometrik fonksiyon seçili ise yapılacak işlemler.
    sec_type=4; % Function type kısmında dördüncü tip seçilmiş olmalı.
    if sec_generator==1 % Chaotic oscillator kısmında birinci üreteç seçili ise
        disp('Fonksiyon Seçimi: Trigonometrik') % Fonksiyon seçimi: Trigonometrik
        % şeklinde gösterir.
        handles = guidata(hObject);
        set(handles.k,'String','10'); % Parameters kısmındaki k değerini 10' a ayarlar.
        set(handles.d,'String','5.6'); % Parameters kısmındaki d değerini 5.6' ya ayarlar.
        set(handles.a,'String','2'); % Parameters kısmındaki a değerini 2' ye ayarlar.
        set(handles.b,'String','0.38'); % Parameters kısmındaki b değerini 0.38' e
        % ayarlar.
        set(handles.c,'String','0'); % Parameters kısmındaki c değerini 0' a ayarlar.
        set(handles.F,'String','0'); % Parameters kısmındaki F değerini 0' a ayarlar.
        axes(handles.foto2) % foto 2 yüzeyinde yani program ekranının sol kısmında
        % çizim görülür.
        imshow('chua_trigonometric1.bmp') % Yukarıdaki komutla tanımlı yüzeyde
        % trigonometrik fonksiyona ait denklem takımı gösterilir.
    save type_secimi sec_type % Yukarıdaki satırlarda yapılanları kaydeder.

```

Tablo 5.3. AnadigmCHAOS Arayüzünde TEST İşlemi İçin Hazırlanan Kodlar.

```

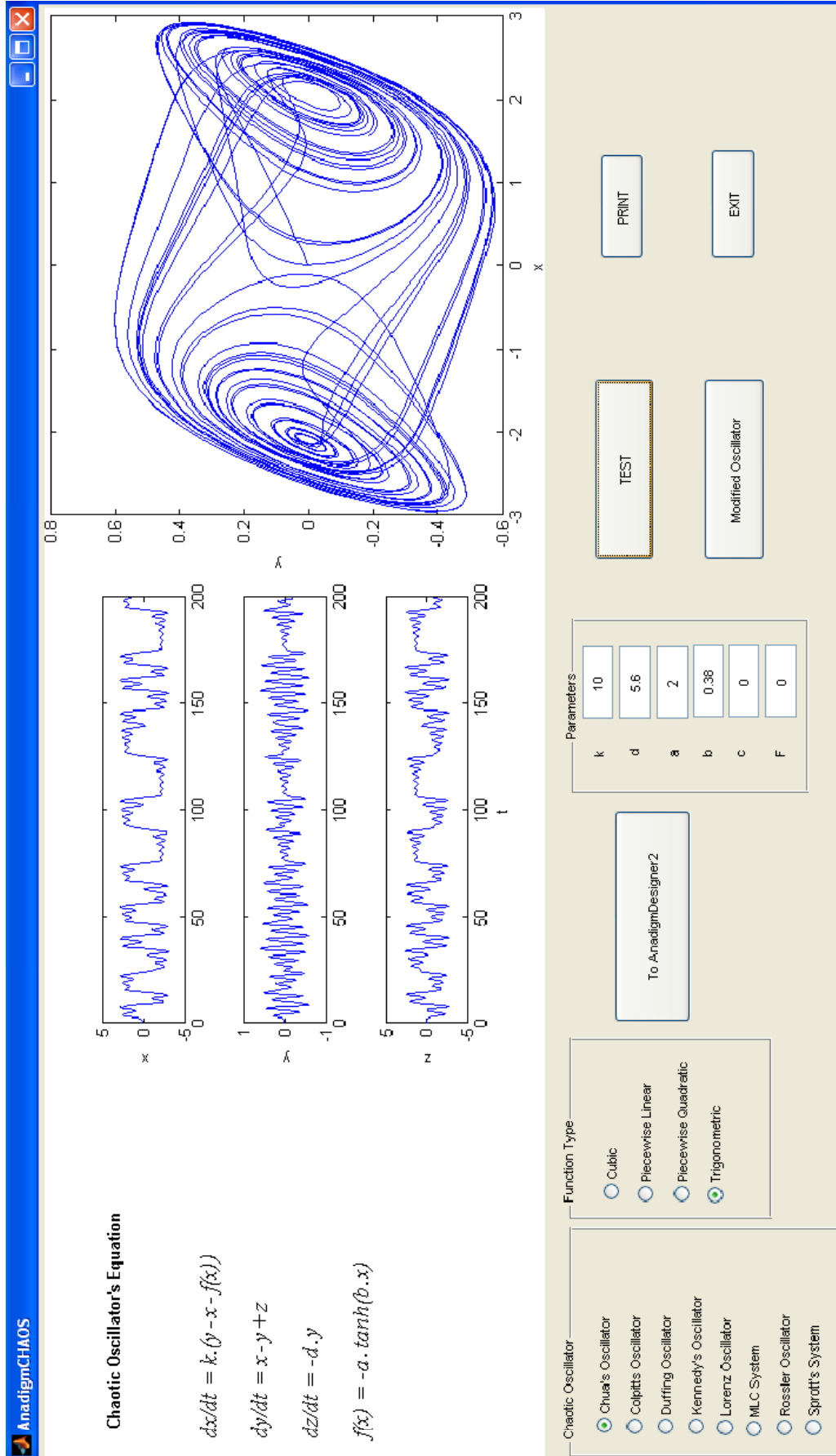
function Test_Callback(hObject, eventdata, handles) % TEST işlemi için gerekli
%fonksiyon dosyası.

load generator_secimi % Seçilen osilatör tipini yükler.
load type_secimi % Seçilen fonksiyon tipini yükler.
if sec_generator==1 & sec_type==4; % Eğer birinci osilatör tipi ve dördüncü
%fonksiyon tipi seçili ise
    load chua_denklem_reset_4 % Trigonometrik fonksiyona sahip Chua osilatörünün
%reset parametreleri yüklenir.
    guncelle % guncelle denklemini çalıştırır.
    plot_secim=1; % plot_seçimi 1 yapılarak nümerik analiz sonuçları hem (x-t), (y-t) ve
% (z-t) olarak verir hem de (x-y) düzleminde verilir.
    save cizim_secimi_plot_secim % seçilen çizim biçimi kaydedilir.
    chua_trigonometric1; % Trigonometrik fonksiyona sahip Chua osilatörünün
%denklem takımı çalıştırılır.

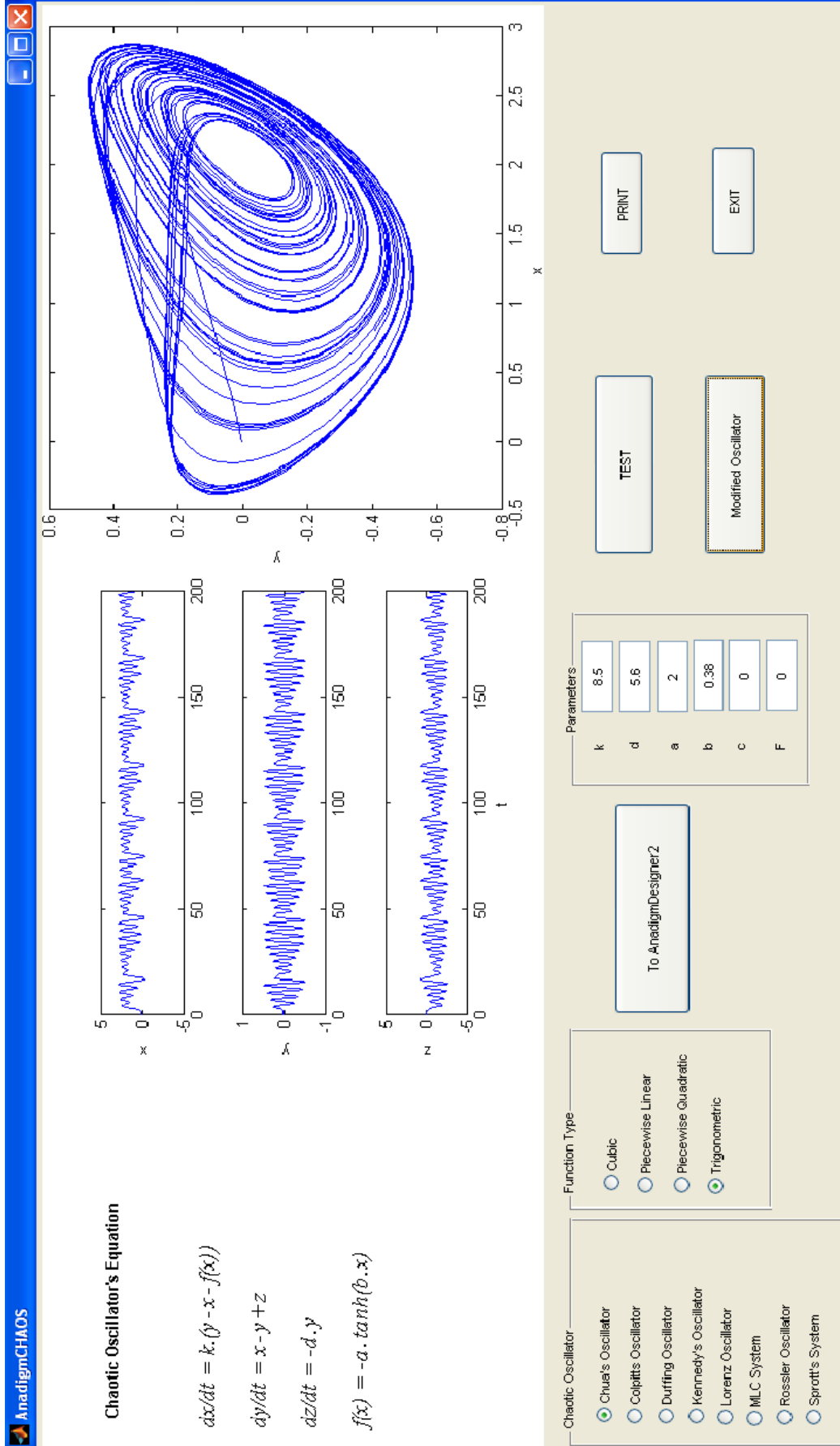
```

Kullanıcı trigonometrik fonksiyona sahip Chua osilatörünün davranışının farklı parametre değerleri için nasıl değiştiğini görmek isterse, yapması gereken şey öncelikle varsayılan parametre değerlerini değiştirmek ve daha sonra “*Modified Oscillator*” sekmesine tıklamaktır. Bu işlem yapıldığında, değişen parametrelere göre nümerik analiz yapılmakta ve elde edilen sonuçlar program üzerinde grafiksel olarak gösterilmektedir. Örnek olarak Şekil 5.11’ de, (k) parametresinin değeri 10’ dan 8.5’ e çekildiğinde osilatörün tek çeker oluşturduğu görülmektedir.

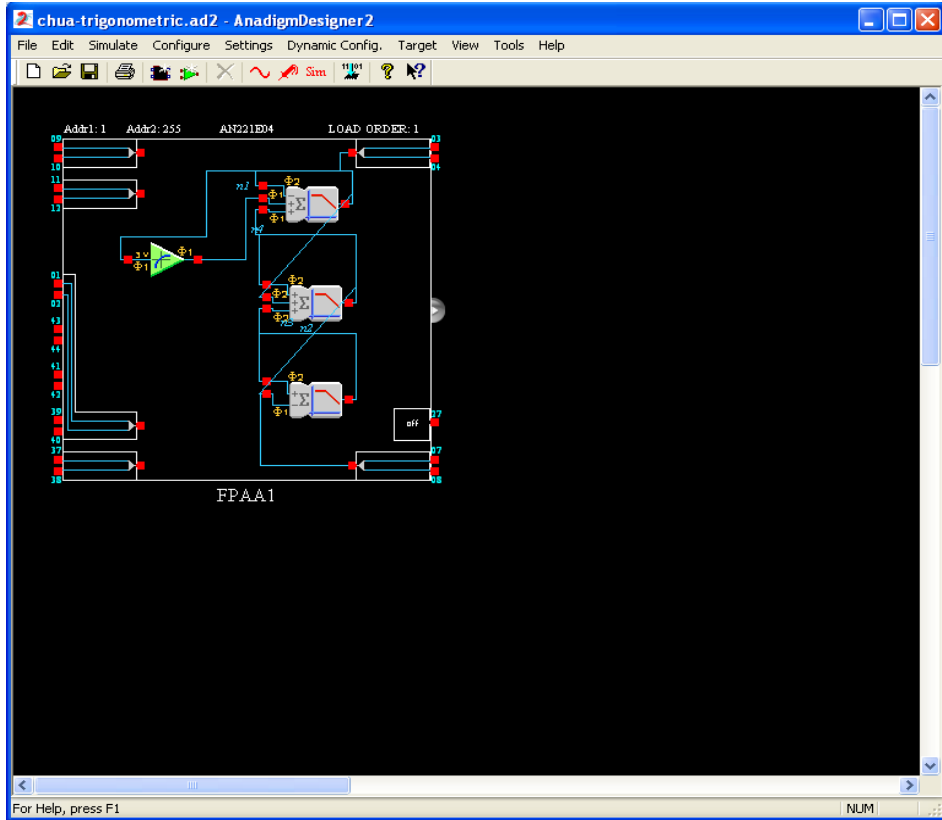
Kullanıcı, seçtiği trigonometrik doğrusal olmayan fonksiyona sahip Chua osilatörünü “*To AnadigmDesigner2*” sekmesine tıkladığı takdirde FPAA-tabanlı tasarımını AnadigmDesigner 2’ ye aktarır. Aktarılan program, Şekil 5.12’ de gösterilmektedir.



Şekil 5.10. Arayüzde Trigonometrik Fonksiyona Sahip Chua Osilatörün Mevcut Parametreleri ile Nümerik Analizinden Elde Edilen Sonuçlar.

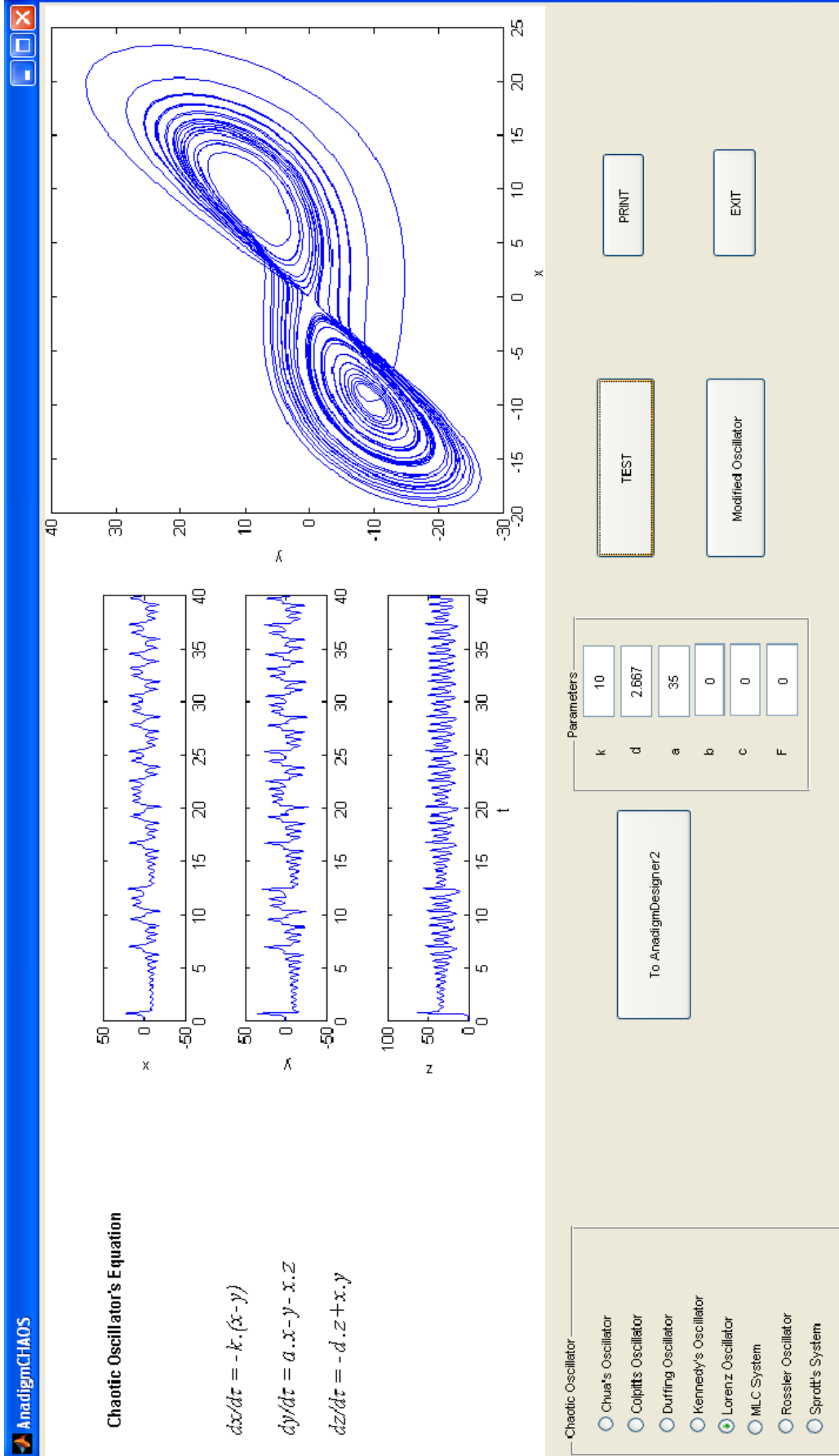


Şekil 5.11. Arayüzde Trigonometrik Fonksiyona Sahip Chua Osilatördeki (k) Parametresi Değiştirilerek Yapılan Nümerik Analizden Elde Edilen Tek Çeker Kaotik Dinamikleri ve Çeker Gösterimi.

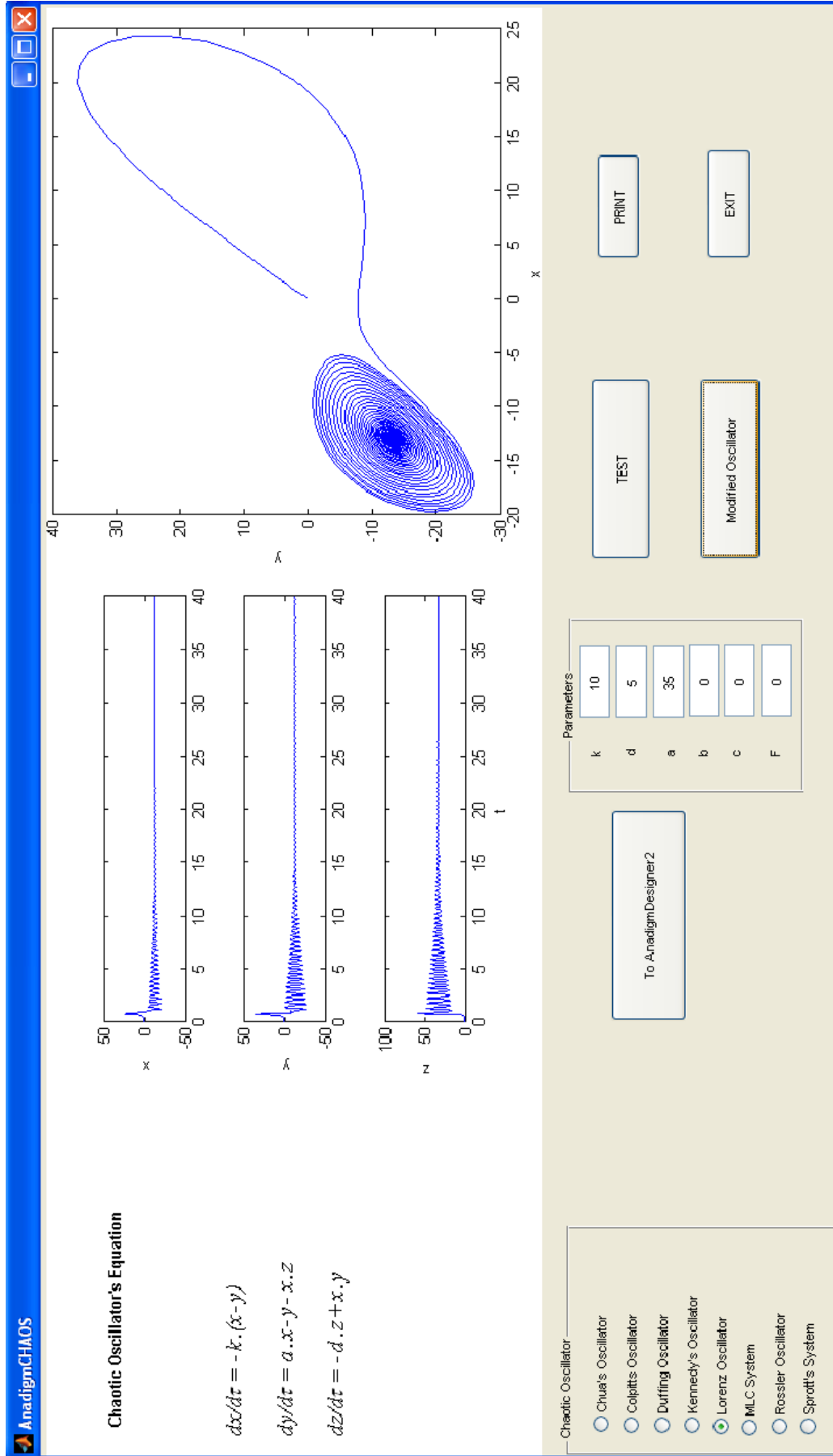


Şekil 5.12. Seçilen Trigonometrik Fonksiyona Sahip Chua Osilatörün AnadigmDesigner 2'ye Aktarılması.

Örneklerden ikincisinde ise “*Chaotic Oscillator*” kısmından *Lorenz Sisteminin* seçildiğini farz edelim. Seçilen Lorenz sisteminin parametre değerleri ve denklemleri, arayüz ekranında görülür. Kullanıcı, Lorenz sisteminin mevcut parametre değerleriyle nümerik analizini yapmak istediğinde, “*TEST*” sekmesine tıklaması yeterlidir. Arayüz üzerinde hem zaman domeninde hem de $(x-y)$ düzleminde elde edilen sonuçlar, Şekil 5.13’ de verilmektedir. Kullanıcı, Lorenz sisteminin farklı parametre değerleri için sistem davranışının nasıl değiştiğini görmek istiyor ise, varsayılan parametre değerlerini isteğine göre değiştirir ve “*Modified Oscillator*” sekmesine tıklar. “*Modified Oscillator*” sekmesine tıkladığında, değişen parametrelere göre nümerik analiz yapılmakta ve elde edilen sonuçlar, program üzerinde grafiksel olarak gösterilmektedir. Örnek olarak Şekil 5.14’ de, (d) parametresinin değeri 2.667’ den 5’ e değiştirildiğinde Lorenz sisteminin nasıl bir davranış sergilediği görülmektedir.

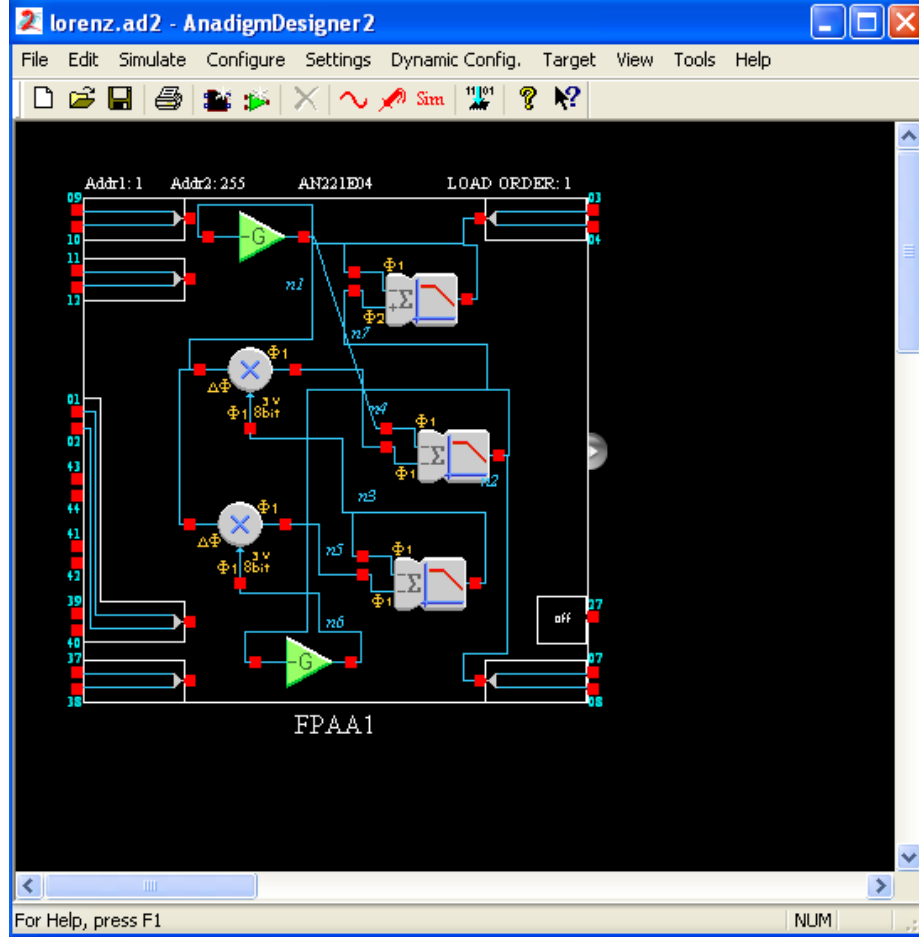


Şekil 5.13. Arayüzde Lorenz Osilatörün Mevcut Parametreleri ile Nümerik Analizinden Elde Edilen Sonuçlar.



Şekil 5.14. Arayüzde Lorenz Sistemindeki d Parametresi Değiştirilerek Yapılan Nümerik Analizden Elde Edilen Sonuçlar.

Kullanıcı, seçtiği bu osilatörü, “*To AnadigmDesigner 2*” sekmesine tıkladığı takdirde FPAA’ nın programlanmasında kullanılan AnadigmDesigner 2’ ye aktarır ve aktarılan bu program, Şekil 5.15’ de görülmektedir.



Şekil 5.15. Seçilen Lorenz Sisteminin AnadigmDesigner 2’ ye Aktarılması.

6. BÖLÜM

SONUÇ VE DEĞERLENDİRME

Bu tez çalışmasında kaotik osilatör devrelerinin programlanabilir analog dizi (FPAA) yapıları ile yeniden yapılandırılabilir ve programlanabilir tasarımı, gerçekleştirimi ve senkronizasyon uygulamaları yapılmıştır. Bunların yanı sıra kaotik osilatörlerle ilgilenen araştırmacıların, lisans ve lisansüstü öğrencilerinin yararlanabileceği bir kaynak olacak olan AnadigmCHAOS arayüzü tasarlanmıştır. Bu tasarımlar, teorik çalışma, nümerik analiz ve laboratuvar deneylerini kapsayacak şekilde yapılan üç aşamalı çalışma programı ile başarılı bir şekilde gerçekleştirilmiştir.

Kaotik işaret üreten osilatör devre modelleri, kaos tabanlı uygulamalarının en önemli tasarım aşamasını oluşturmaktadır. Çünkü uygulamalarda kaos dinamiklerinin özelliklerinden yararlanmak ve bu özellikleri fiziki hayata geçirebilmek için öncelikle bu dinamiklerin pratik olarak üretilmesi gerekmektedir. Bu amaç doğrultusunda kaotik osilatör yapılarının deneysel olarak yeniden yapılandırılabilir ve programlanabilir tasarımlarını gerçekleştirmek için FPAA' ların uygun bir çözüm noktası oluşturacağı düşünülmüştür. Bu düşünceden hareketle FPAA' lar ile kaotik osilatör tasarımları yapılmıştır. Kaotik osilatörlerin, FPAA gerçekleştirimlerinden önce nümerik analizleri yapılmıştır. FPAA elemanı, $\pm 2V$ saturasyon seviyesine sahip olduğu için nümerik analizleri başarıyla tamamlanan kaotik yapıların bazıları ölçeklendirme işlemine tabii tutulmuştur.

Bölüm 3' de literatürde üzerinde en çok çalışılan kaotik osilatör devreleri incelenmiştir. Bu devrelerden Bölüm 3.3.1.1' de incelenen Chua osilatörü, çok sayıda doğrusal olmayan fonksiyon ile farklı gerçekleştirime sahip bir osilatör tipi olup bu tez çalışmasında 4 farklı doğrusal olmayan fonksiyon kullanılarak irdelenmiştir. Bu devre yapısının SIMULINKTM modellemesi yapılarak nümerik analizi tamamlanmıştır. Ayrıca parçalı doğrusal fonksiyona sahip Chua osilatörü ayrı devre elemanları ile de deneysel

olarak gereklenmiřtir. Blm 4.4.1' de verilen ve farklı dođrusal olmayan fonksiyonlara sahip bu osilatr devresinin, leklendirme iřlemlerinden sonra FPAA zerinde yeniden yapılandırılabilir ve programlanabilir tasarımı yapılmıř, FPAA tasarımının deneysel sonuları laboratuvar ortamında kaydedilmiřtir. Bahsedilen FPAA tasarımlarında programlanabilir parametre (μ) olup bu parametrenin deđiřimi ile Chua osilatr devresinin davranıřı kaotik davranıřtan periyodik davranıřa kadar farklı karakteristik zellikler sergilediđi grlmřtir. Bu davranıřlar, Blm 4.4.1' de Chua osilatrnn her bir dođrusal olmayan fonksiyonu iin ayrıca gsterilmiřtir. Bu FPAA tasarım sonularının Blm 3.3.1.1' de verilen hem nmerik analiz sonularıyla hem de ayrıık elemanlarla gerekleřtirilen devrenin deneysel sonularıyla rtřtđ grlmřtir.

Blm 3' de incelen diđer kaotik sistemlerden olan kaotik Colpitts osilatrnn, Kennedy osilatrnn, Lorenz sisteminin ve Rssler sisteminin, SIMULINK™ modellemeleri yapılarak nmerik analizleri incelenmiřtir. Ayrıca bu osilatr yapılarının ayrıık elemanlarla gerekleřtirimi de yapılmıřtır. Bu sistemlerin yeniden yapılandırılabilir ve programlanabilir tasarımlarının yapılabilmesi iin ncelikle devre denklem takımlarına leklendirme iřlemi uygulanmıř sonra leklendirilmiř devre denklemleri FPAA uygulama kartı zerinde gerekleřtirilmiřtir. Bu sistemlere ait FPAA gerekleřtirim řemaları ve bu gerekleřtirimden elde edilen sonular Blm 4' de yer almaktadır. Yukarıda bahsedilen kaotik osilatr yapılarına ait FPAA tasarımlarının deneysel sonuları da nmerik analizlerden ve ayrıık elemanlarla gereklenmiř devrelere ait kaotik dinamik ve ekerler ile uyumlu olduđunu belirtmekte fayda bulunmaktadır.

Blm 3' de incelenen kaotik osilatrler arasında yer alan Duffing osilatr ve MLC sistemi harici bir kaynađa sahip olduklarından dolayı otonom olmayan kaotik osilatrler arasında yer almaktadırlar. Bu yapılar, Blm 3.3.1.3 ve 3.3.1.6' da nmerik analizleri ve donanım gerekleřtirimleri yapılarak incelenmiřlerdir. Bu osilatr yapılarının hem nmerik analizlerinde hem de deneysel gerekleřtirimlerinde harici kaynađın genlik ve frekansının ayarlanması sırasında zorluk yařanmıřtır. Blm 4.4.3 ve 4.4.6' da bu osilatrlerin FPAA-tabanlı gerekleřtirimi zerinde durulmuřtur. FPAA, dahili bir AC sinyal kaynađına sahip olduđu iin Duffing ve MLC osilatrlerin harici kaynak ihtiyacı FPAA ierisinde bulunan bu dahili sinyal reteci ile karřılanmıřtır. Bu osilatrlerin kaosa girmesinde kullanılan (μ) parametresi, programlanabilir formda kolayca deđiřtirilebilmiřtir. Bu osilatrlerin Blm 3.3.1.3 ve 3.3.1.6' da elde edilen analiz ve

deneysel sonuçların FPAA-tabanlı gerçekleştirim sonuçları ile olan benzerliği vurgulanmalıdır.

Bölüm 3.3.1.8' de incelenen diğer bir kaotik sistem olan Sprot sistemi, beş farklı doğrusal olmayan fonksiyon kullanılarak incelenmiştir. Sistemin her bir doğrusal olmayan fonksiyon kullanılarak SIMULINK™ modellemesi yapıldığında farklı modellemeye ve kaotik davranışlara sahip oldukları görülmüştür. Ayrıca beklendiği üzere donanımsal gerçekleştirimleri de birbirinden farklıdır. Donanımsal gerçekleştirimler sırasında örnek uygulama olarak Denklem (3.28)' de yer alan doğrusal olmayan fonksiyona sahip Sprot sisteminin deneysel kurulumu yapılmış ve sonuçları kaydedilmiştir. Bölüm 4.4.8' de ise Sprot sisteminin nümerik analizlerde de kullanılan beş farklı doğrusal olmayan fonksiyon kullanılarak FPAA üzerinde programlanabilir gerçekleştirimleri yapılmıştır. Bu gerçekleştirimler yapılmadan önce sistem denklemleri ölçeklendirme işlemine tabii tutulmuşlardır. Ölçeklendirme işlemleri tamamlandıktan sonra Sprot sistemleri FPAA üzerinde gerçekleştirilmiş olup bu gerçekleştirimlerden elde edilen deneysel sonuçlar yine aynı bölümde sunulmuştur. Bu FPAA gerçekleştirimlerinin sonuçları ile Bölüm 3.3.1.8' de verilen hem nümerik analizlerden elde edilen hem de ayırık elemanlarla kurulan pratik devreye ait sonuçlar örtüşmüştür.

Bölüm 3.3.1.9' da HSA-tabanlı yapılardan Chua osilatörü incelenmiş olup bu osilatörün SIMULINK™ modellemesi yapılmıştır. Nümerik analizin yanı sıra bu osilatör yapısı deneysel olarak da gerçekleştirilmiştir. Bu yapıların da FPAA üzerinde programlanabilir olarak gerçekleştirilebileceğini göstermek için Bölüm 4.4.9' da HSA-tabanlı Chua ve MLC devrelerinin FPAA gerçekleştirimi yapılmıştır. Bu gerçekleştirimlerin yapılması için öncelikle devre denklemleri ölçeklendirilmiş ve daha sonra FPAA tasarımına geçilmiştir. Bu tasarımlardan elde edilen sonuçlar Şekil 4.22 ve 4.24' de verilmiştir. Şekil 4.22' de verilen HSA-tabanlı Chua devresinin FPAA gerçekleştirim sonuçları hem Bölüm 3.3.1.9' da verilen nümerik analiz ve deneysel sonuçlarla hem de Bölüm 4.4.1' de verilen FPAA-tabanlı orijinal Chua osilatör sonuçları ile uyuşmaktadır. Şekil 4.24' de verilen ve FPAA' da gerçekleştirilen HSA-tabanlı MLC sisteminin sonuçları ise orijinal MLC sisteminin Bölüm 3.3.1.6' da verilen nümerik ve deneysel sonuçları ile Bölüm 4.4.6' da verilen FPAA gerçekleştirim sonuçları ile benzer olduğu görülmüştür. Diğer bir deyişle hem orijinal hem de HSA-tabanlı Chua ve MLC devreleri FPAA üzerinde tasarlanmış ve tasarım sonuçları başarıyla elde edilmiştir.

Doğrusal olmayan fonksiyonun yapısı değiştirilerek oluşturulan n -çekerli (3, 4, 5 ve 7 çekerli) Chua osilatörünün de, bu tez çalışması sırasında, FPAA gerçekleştirmeleri yapılmıştır. FPAA tasarımlarından önce yapılan nümerik analizlerden bu yapıların ölçeklendirme işlemine sokulması gerektiği görülmüştür. 3-çekerli, 4-çekerli, 5-çekerli ve 7-çekerli kaotik üreteçler için sistem denklemleri ayrı ayrı ölçeklendirilmiş ve ölçeklendirilen denklem takımları ile FPAA tasarımları ayrı ayrı gerçekleştirilmiştir. Bu gerçekleştirmeler sonucunda elde edilen deneysel sonuçlar, 3-çekerli üreteç için Şekil 4.27(a)' da, 4-çekerli üreteç için Şekil 4.27(b)' de, 5-çekerli üreteç için Şekil 4.28(a)' da ve 7-çekerli üreteç için de Şekil 4.28(b)' de verilmiştir. Bu sonuçlara bakarak FPAA-tabanlı n -çekerli yapıların gerçekleştirilebileceği görülmüştür.

Bu tez çalışmasında Bölüm 4.5' de FPAA-tabanlı PLL modelin tasarımı yapılmış ve bu tasarımdan elde edilen deneysel sonuçlar kaydedilmiştir. Bu sonuçlar sadece osilatör yapılarının değil bununla birlikte içerisinde osilatörlerin de olduğu daha karmaşık yapıların FPAA-tabanlı tasarımının mümkün olabileceğinin ispatı açısından önemlidir.

Yukarıda sonuçları mukayese edilen deneysel çalışmalarda karşılaşılan en önemli zorluk, ayrık elemanlarla gerçekleştirilen devrelerin parametrelerinin ayarlanmasıdır. Bilindiği gibi kaotik sistemler devre parametrelerindeki en ufak değişikliğe karşılık çok farklı davranışlar sergileyebilmektedir. Dolayısıyla ayrık elemanlarla devre tasarımında arzu edilen kaotik davranışı elde edene kadar devre parametreleri hassas bir şekilde ayarlanmak zorundadır. Bununla birlikte deney düzeneğinin farklı zamanda ve/veya mekanda tekrar kurulması gerektiğinde devre parametrelerinin ayarlanması ile ilgili aynı problemler tekrar tasarımcının karşısına çıkmaktadır. Buna ilaveten donanım gerçekleştirmede kullanılacak işlemsel kuvvetlendiricilerin temin edilmesi de başka bir sıkıntıdır. Buna karşın bu tez çalışmasında programlanabilir ve yeniden yapılandırılabilir özelliğe sahip FPAA' ların devre tasarımında ve devre parametrelerinin farklı zamanlarda ve mekanlarda yeniden ayarlanması konusunda çok büyük kolaylık ve avantaj sağladığı görülmüştür.

Kaotik haberleşme sistemleri için kaotik osilatörlerin kullanıldığı senkronizasyon uygulamaları önemlidir. Kaotik davranışların senkronizasyon olmaksızın birebir kopyalanmayacağı düşünüldüğünden güvenilir ve gizli haberleşmedeki önemi daha iyi anlaşılabilir. Vericide kaotik işaretin içerisine en basit yöntem kullanılarak gizlenmiş bir

bilgi işareti verici ile senkron olmayan bir alıcı da kesinlikle yeniden elde edilemez. Bu bilgiler ışığında Chua devresinin ve MLC devresinin kullanıldığı yeniden yapılandırılabilir senkronizasyon uygulamaları yapılmıştır. Bölüm 4.6.1’ de verilen bu senkronizasyon uygulamalarından FPAA-tabanlı Chua devresinin kullanıldığı senkronizasyon gerçekleştirimi üç farklı yöntem kullanılarak incelenmiştir. Bu yöntemler, kaskad bağlantılı senkronizasyon yöntemi, Pecora & Carroll yöntemi ve kuplajlama yöntemidir. Bu yöntemlerin FPAA üzerinde gerçekleştirmeleri, sırasıyla Şekil 4.34, 4.37 ve 4.39’ da verilmiş olup bu gerçekleştirmelerden elde edilen deneysel sonuçlar ise, sırasıyla Şekil 4.35, 4.38 ve 4.40’ da yer almaktadır. Bu sonuçlar incelendiğinde FPAA-tabanlı Chua devresinin kullanıldığı senkronizasyon uygulamalarının başarı ile tamamlandığı görülmektedir. Bölüm 4.6.2’ de MLC sisteminin kullanıldığı senkronizasyon gerçekleştirimi için Pecora & Carroll yöntemi ve tek yönlü kuplajlama yöntemi ayrı ayrı denenmiştir. Bu yöntemlerin kullanıldığı FPAA-tabanlı senkronizasyon için gerçekleştirim şemaları sırasıyla Şekil 4.41 ve 4.44’ de verilmiştir. FPAA-tabanlı senkronizasyon işlemlerinden elde edilen kaotik dinamikler ve senkronizasyon grafikleri sırasıyla Şekil 4.42 ve 4.45’ de yer almaktadır. Chua osilatörünün senkronizasyon uygulamasında olduğu gibi FPAA-tabanlı MLC sisteminin kullanıldığı senkronizasyon uygulamaları da başarıyla sonuçlanmıştır. Bu devre modelleri aynı senkronizasyon yöntemleri kullanılarak ayrık elemanlarla gerçekleştirilmek istenildiği takdirde, en başta MLC devresindeki harici kaynağın genlik ve frekans parametrelerinin ve hem Chua hem de MLC devresinin devre parametrelerinin ayarlanmasından kaynaklanan problemlerin yaşanabileceği ihtimali her zaman var olduğu için bu devrelerinin kullanıldığı senkronizasyon gerçekleştirmeleri zor olacaktır. Buna karşılık programlanabilir ve yeniden yapılandırılabilir özelliğe sahip FPAA’ nın içersinde bulunan AC kaynağın genlik ve frekansı kolaylıkla ayarlanabildiği ve tasarım kolaylığı sağladığı için bahsedilen problemlerin çözümüne yönelik olarak FPAA-tabanlı senkronizasyon uygulamaları iyi bir alternatiftir.

Tez çalışmasının bu bölümünde verici sistem FPAA’ da, alıcı sistem ise ayrık devre elemanları ile gerçekleştirilerek senkronizasyon uygulaması denenmiş fakat elde edilen sonuçlar tatmin edici olmamıştır. İkinci bir çalışmada ise ilk gerçekleştirimin tersine verici sistem ayrık devre elemanları ile alıcı sistem FPAA’ da gerçekleştirilerek de senkronizasyon uygulaması yapılmaya çalışılmış, fakat benzer sonuçlar elde edilmiştir.

Bu kötü sonuçların iki sebebi olduğu düşünülmektedir. Bunlardan birincisi FPAA' da tasarlanan devrenin voltaj aralığı $\pm 2V$ ' tur. Ayrık elemanlarla gerçekleştirilen devre ise $\pm 2V$ aralığının dışında da voltaj üretebilmektedir ve bundan kaynaklanan bir genlik uyumsuzluğu olduğu görülmüş ve çözüm olarak kuvvetlendirici ve tampon devreleri kullanılarak voltaj sahaları uyumlu hale getirilmiştir. Deneysel çalışma sırasında yapılan ölçümler neticesinde FPAA' da tasarlanan devrenin frekansı ile ayrık elemanlarla gerçekleştirilen devrenin frekansının aynı olmadığı görülmüş ve senkronizasyonun sağlanamamasının ikinci sebebinin bu frekans uyumsuzluğu olduğu tespit edilmiştir. Bu ikinci problem çözülemediğinden FPAA-ayrık devre, ayrık devre-FPAA birlikteliği beklenen sonuçları üretecek şekilde tam olarak gerçekleştirilememiştir.

Bu tez çalışmasında kaotik osilatörlerle ilgilenen bilim adamları, araştırmacılar, lisans ve lisansüstü öğrencileri için eğitim amaçlı AnadigmCHAOS isimli arayüz 5. Bölümde sunulmuştur. Bu arayüz, bu tez çalışmasında 3. ve 4. Bölümde yer alan kaotik osilatör yapılarından literatürde üzerinde en çok durulanlarının olduğu hem nümerik analiz tabanlı gerçekleştiriminin yapıldığı hem de FPAA-tabanlı tasarımların FPAA programına gönderilebildiği GUI-tabanlı bir uygulamadır. Bu arayüz, kaotik osilatörlerin hem varsayılan parametre değerleriyle hem de kullanıcının belirleyeceği farklı parametre değerleriyle nümerik analizini yapabilmektedir. Ayrıca istenilen kaos üretici seçildiğinde seçilen üretcin FPAA-tabanlı gerçekleştirimi FPAA' nın arayüz programı olan AnadigmDesigner 2' de açılabilir. MATLABTM-GUI' de hazırlanan bu arayüz Şekil 5.8' de verilmiştir. AnadigmCHAOS arayüzü GUI' ye alternatif olarak C# programlama dili ile de oluşturulmuştur. Bu tasarım uygulamasında FPAA-tabanlı kaotik osilatörler tek tek C# programlama dili ile kodları yazılarak oluşturulmuş ve programa ilişkin bir arayüz hazırlanmıştır. Kullanıcı bu arayüzde çalışacağı osilatör tipini seçebilmekte ve "To AnadigmDesigner2" tuşuna bastığında FPAA' nın arayüz programı olan AnadigmDesigner 2' de seçilen üretici için en başta boş bir FPAA dosyası açılmaktadır. Daha sonra seçilen osilatör yapısında kullanılan bloklar teker teker AnadigmDesigner 2 program ekranına gelmekte ve devamında ara bağlantılar da görsel olarak tek tek yapılmaktadır. Bu yazılımla ilgili çalışmalar devam etmektedir.

Gelecek yıllarda, üzerinde çalışma yapılan kaos ve kaotik sistemlerle ilgili olarak gerek kaotik osilatör tasarımı, gerekse bu osilatörlerle gerçekleştirilecek kaotik sistemlerde

kaotik kontrol ve kaotik haberleşme amaçlı çalışmaların artarak devam etmesi beklenmektedir. Yeniden yapılandırma ve programlama özellikleriyle öne çıkan FPAA'ların kaotik sistemlerin tasarımında tercih edilmesi ayırık elemanlarla yapılan tasarımlara kıyasla parametre değerlerinin kolaylıkla ayarlanmasına ve değiştirilmesine, bu sayede tasarım sürecinin kısaltılmasına, dolayısıyla tasarım maliyetlerinin önemli ölçüde düşürülmesine imkan sağlamaktadır. Bu tez çalışmasında sunulan deneysel sonuçların ve hazırlanan yazılımların bu konu üzerinde çalışan araştırmacılar için yenilikler ve hedefler sunabileceği düşünülmektedir.

KAYNAKLAR

1. Luo, J., Circuit Design and Routing for FPAAs, Ph. D. Thesis, University of Maryland, 2005.
2. Gulak, P.G., D'Mello, D.R., A Review of Field Programmable Analog Arrays, SPIE Conference, 1996.
3. D'Mello, D.R., Gulak, P.G., Design Approaches to Field-Programmable Analog Integrated Circuits, Analog Integrated Circuits and Signal Processing, 17, 7-34, 1998.
4. Xilinx, Inc., 2100 Logic Drive San Jose, CA 95124-3400, USA, 2010, www.xilinx.com
5. Altera Corporation, 101 Innovation Drive, San Jose, California 95134, USA, 2010, www.altera.com
6. Anadigm Inc., 1901 South Bascom Avenue Suite 1550 Campbell CA 95008, USA, 2010, www.anadigm.com
7. Gaudet, V.C., Gulak, P.G., Implementation Issues for High-Bandwidth Field-Programmable Analog Arrays, Journal of Circuits, Systems, and Computers - Special Issue on Analog and Digital Arrays, 8, 541-558, 1998.
8. Znamirovski, L., Palusinski, O.A., Vrudhula, S.B.K., Programmable Analog/Digital Arrays in Control and Simulation, Analog Integrated Circuits and Signal Processing, 39, 55-73, 2004.
9. Anderson, D., Marcjan, C., Bersch, D., Anderson, H., Hu, P., Palusink, O., Gettman, D., Macbeth, I., Bratt, A., A Field Programmable Analog Array and Its Application, 1997 Custom Integrated Circuits Conference Proceedings, 1997.
10. Lee, E.K.F., Hui, W.L., A Novel Switched-Capacitor Based Field-Programmable Analog Array Architecture, Kluwer Analog Integrated Circuits and Signal Processing - Special Issue on Field Programmable Analog Arrays, 17(1-2), 35-50, 1998.
11. Pierzchala, E., Perkowski, M.A., A High-Frequency Field-Programmable Analog Array (FPAA) Part I: Design, Kluwer Analog Integrated Circuits and Signal Processing - Special Issue on Field Programmable Analog Arrays, 17(1-2), 143-156, 1998.

12. Wang H., Field Programmable Analog Array Synthesis, Ph. D. Thesis, University of Arizona, Department of Electrical and Computer Engineering, 2002.
13. Sivilotti, M., A Dynamically Configurable Architecture for Prototyping Analog Circuits, MIT VLSI Conference, 237-258, 1988.
14. Lee, E.K.F., Gulak, P.G., A CMOS Field Programmable Analog Array, IEEE Journal of Solid-State Circuits, 26(12), 1860–67, 1991.
15. Lee, E.K.F., Gulak, P.G., Field Programmable Analogue Array Based on MOSFET Transconductors, Electronics Letters, 28(1), 28-29, 1992.
16. Pierzchala, E., Perkowski, M., Vanhalen, P., Schaumann, R., Current-Mode Amplifier/Integrator for a Field Programmable Analog Array, IEEE International Solid-State Circuits Conference Technical Digest, 196–97, 1995.
17. Pierzchala, E., Perkowski, M.A., A Field Programmable Analog Array for Continuous, Fuzzy, and Multi-Valued Logic Applications, Proceedings of The 24th International Symposium on Multiple-Valued Logic, 148-154, 1994.
18. Edwards, R.T., Strohbehn, K., Jaskulek, S.E., A Field-Programmable Mixed-Signal Array Architecture Using Antifuse Interconnects, IEEE International Symposium on Circuits and Systems, 319-322, 2000.
19. Kutuk, H., Kang, S.M., Filter Design Using A New Field-Programmable Analog Array (FPAA), Analog Integrated Circuits and Signal Processing, 14, 81-90, 1997.
20. Kutuk, H., Kang, S. M., A Field-Programmable Analog Array (FPAA) Using Switched-Capacitor Technique, Proceedings of IEEE International Symposium on Circuits and Systems, 4, 41–43, 1996.
21. Pankiewicz, B., Wojcikowski, M., Szczepanski, S., Sun, Y., A CMOS Field Programmable Analog Array and Its Application in Continuous-Time OTAC Filter Design, IEEE International Symposium on Circuits and Systems, 1, 5-8, 2001.
22. Premont, C., Grisel, R., Abouchi, N., Chante, J. P., Current-Conveyor Based Field Programmable Analog Array, 1996 Midwest Symposium on Circuits and Systems, 155-157, 1996.
23. Motorola Inc., MPAA020 Field Programmable Analog Array Datasheet, 1997.

24. IMP Inc, Preliminary Product Information – IMP50E10 EPAC (Electrically Programmable Analog Circuit) Datasheet, EPAC Design Handbook, 1996.
25. Lattice Semiconductor, Programmable Analog Circuits (ispPAC), <http://www.latticesemi.com>
26. Zetex Semiconductors, TRAC (Totally Reconfigurable Analog Circuit), <http://www.zetex.com>
27. Berenson, D., Estévez, N., Lipson, H., Hardware Evolution of Analog Circuits for In-situ Robotic Fault-Recovery, Proceedings of The 2005 NASA/DoD Conference of Evolution Hardware, 2005.
28. Waters, M., Programmable Monolithic Multi-Pole Filter Design With FPAA's, www.analogzone.com, 2003.
29. Roberts, T., Using Field-Programmable Analog to Build Adaptable RFID Readers, RF and Microwave Technology for Design Engineers – RFDESIGN, 44-48, 2004.
30. Sun Y., Wireless Communication Circuits and Systems, The Institution of Electrical Engineers, London, United Kingdom, 2004.
31. Sanches, P.R.S, Müller, A.F., Carro, L., Susin, A:A., Nohama, P., Analog Reconfigurable Technologies for EMG Signal Processing, Revista Brasileira de Engenharia Biomédica, 23(2), 153-157, 2007.
32. Caponetto, R., Di Mauro, A., Fortuna, L., Frasca, M., Field Programmable Analog Array to Implement A Programmable Chua's Circuit, Int. Journal of Bifurcation and Chaos, 15(5), 1829-1836, 2005.
33. Hulub, M., Frasca, M., Fortuna, L., Arena, P., Implementation and Synchronization of 3x3 Grid Scroll Chaotic Circuits with Analog Programmable Devices, Chaos, 16, 2006.
34. Arena, P., Buscarino, A., Fortuna, L., Frasca, M., Separation and Synchronization of Piecewise Linear Chaotic Systems, Physical Review E, 74(2), 2006.
35. Kılıç, R., Dalkıran, F.Y., FPAA-Based Programmable Implementation of A Chaotic System Characterized With Different Nonlinear Functions, 2008 International Symposium on Nonlinear Theory and its Applications (NOLTA 2008), 132-135, 2008.

36. Kılıç, R., Dalkıran F.Y., Programmable Design and Implementation of A Chaotic System Utilizing Multiple Nonlinear Functions, Turkish Journal of Electrical Engineering and Computer Sciences, 18(4), 647-655, 2010.
37. Kılıç, R., Dalkıran, F.Y., FPAA-Based Implementation of Nonautonomous MLC Circuit, The 17th International Conference on Nonlinear Dynamics of Electronics Systems (NDES), 142-145, 2009.
38. Kılıç, R., Dalkıran, F.Y., Reconfigurable Implementations of Chua's Circuit, International Journal of Bifurcation & Chaos, 19(4), 1339-1350, 2009.
39. Dalkıran F.,Y., Günay, E., Kılıç, R., HSA (Hücreyel Sinir Ağı)-Tabanlı Otonom Olmayan MLC Devresinin Donanımsal Gerçekleştirimleri, Elektrik-Elektronik-Bilgisayar Müh. Sempozyumu-ELECO 2010, 2010.
40. Galliere, J.M., Programmable Analog Array in Control-Systems Laboratory, Proceedings of The 15th Mediterranean Conference on Control&Automation, 2007.
41. Gaudet, V.C., Design of a CMOS Current Conveyor-Based Field-Programmable Analog Array, M.S Thesis, University of Toronto, Department of Electrical and Computer Engineering, 1997.
42. Sanahuja, R., Barcons, V., Balado, L., Figueras, J., Experimental Test Bench for Mixed-Signal Circuits Based on FPAA Devices, Proceedings of The XVIII Conference on Design of Circuits and Integrated Systems - DCIS 2003, 344-349, 2003.
43. Balen, T.R., Andrade, A.Q., Azais, F., Lubaszewski, M., Renovell, M., Applying the Oscillation Test Strategy to FPAA's Configurable Analog Blocks, Journal of Electronic Testing: Theory and Applications, 21, 135-146, 2005.
44. Floyd, T.L., Electronic Devices-Conventional Current Version, Prentice Hall, New Jersey, 2004.
45. Becker, J., Manoli, Y., A Continuous-Time Field Programmable Analog Array (FPAA) Consisting of Digitally Reconfigurable G_M -Cells, International Symposium on Circuits and Systems - ISCAS 2004, 1092-1095, 2004
46. Jankovec, M., Topic, M., Analog Circuit Development System, The IEEE Region 8 EUROCON 2003, 125-129, 2003.

47. Liccardo, A., A Remotely Configurable and Programmable Measurement Laboratory, Ph. D. Thesis, Uni. Degli Studi di Napoli Federico II, 2007.
48. Lakshmanan, S.K., Towards Dynamically Reconfigurable Mixed-Signal Electronics for Embedded and Intelligent Sensor Systems, Ph. D. Thesis, The University of Kaiserslautern, 2008.
49. Fares, O.O., Configurable Analog Building Blocks for Field Programmable Analog Arrays, Ph. D. Thesis, King Fahd University of Petroleum & Minerals, Electrical Engineering, 2004.
50. Jariwala, P., Wu, R., Sensor Signal Conditioning Using Field Programmable Analogue Arrays (FPAA), Analog Signal Processing –ASP 2004, 4-1 - 4-6, 2004.
51. Balen, T.R., Calvano, J.V., Lubaszewski, M.S., Renovell, M., Functional Test of Field Programmable Analog Arrays, Proceedings of The 24th IEEE VLSI Test Symposium – VTS'06, 2006.
52. Schuster, H.G., Just, W., Deterministic Chaos: An Introduction, WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 2005.
53. Ruelle, D., Rastlantı ve Kaos, Tübitak Yayınları, 1995.
54. A Brief History of Chaos Theory,
http://www.schuelers.com/ChaosPsyche/part_1_3.htm
55. Chen, G., Ueta, T., Chaos in Circuits and Systems, World Scientific, 2002.
56. Bilotta, E., Campolo, R., Pantano, P., Stranges, F., Sound Synthesis and Chaotic Behaviour in Chua's Oscillator, Mathematics in Industry, 9, 289-294, 2006.
57. Boukabou, A., Communication Using Chaotic Modulation Demodulation Approach, Arab Research Institute in Sciences & Engineering – ARISER, 3(2), 69-75, 2007.
58. Moon, F.C., Chaotic Vibrations: An Introduction for Applied Scientists and Engineers, John Wiley & Sons, Inc, New Jersey, USA, 2004
59. Strogatz, S.H., Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering, Perseus Books Publishing, 1994.
60. Szemplinska-Stupnicka, W., Chaos, Bifurcations and Fractals Around Us: A Brief Introduction, World Scientific, 2003.

61. Suneel M., Electronic Circuit Realization of The Logistic Map, SADHANA - Academy Proceedings in Engineering Sciences, 31(1), 69-78, 2006.
62. Gürses, S., Akkaş, N., Platin, B. E., Ters Dönmüş Bir Sarkacın Doğrusal Olmayan Konum Denetiminde En Büyük Lyapunov Üstelinin Poincare Kesitinden Elde Edilmesi, Journal of İstanbul Kültür University, 4, 121-137, 2006.
63. Kennedy, M.P., Experimental Chaos from Autonomous Electronic Circuits, Phil. Trans. R. Soc. Lond. A, 353, 13-32, 1995.
64. Kennedy, M.P., Chua L.O., Van der Pol and Chaos, IEEE Trans. Circuits&Systems-I, 33, 974-980, 1986.
65. Kılıç, R., Karışık Modlu Yeni Bir Kaotik Devre Modeli Tasarımı ve Güvenilir Haberleşme Amaçlı Sistem Gerçekleştirmede Kullanılması, Doktora Tezi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, 2000.
66. Ueda, Y., Akamatsu, K., Chaotically Transitional Phenomena in The Forced Negative-Resistance Oscillator, IEEE Trans. Circuits&Systems-I, CAS-28, 217-226, 1980.
67. Testa, J., Perez, J., Jeffries, C., Evidence for Universal Chaotic Behaviour of A Driven Nonlinear Oscillator, Phys. Rev. Lett., 48, 716-717, 1982.
68. Pehlivan, İ., Yeni Kaotik Sistemler: Elektronik Devre Gerçeklemeleri, Senkronizasyon ve Güvenli Haberleşme Uygulamaları, Doktora Tezi, Sakarya Üniversitesi , Fen Bilimleri Enstitüsü, 2007.
69. Lakshmanan, M., Murali, K., Chaos in Nonlinear Oscillators, Controlling and Synchronization, World Scientific, 1996.
70. Chua, L.O., Wu, C.W., Huang, A., Zhong, G.-Q., A Universal Circuit for Studying and Generating Chaos-Part I: Routes to Chaos, IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications, 40, 732-745, 1993.
71. Matsumoto, T., Chua, L. O., Tanama, S., Simplest Chaotic Nonautonomous Circuit, Physical Rev. A, 30, 1155-1157, 1984.
72. Elwakil, A.S., Kennedy, M.P., Chua's Circuit Decomposition: A Systematic Design Approach for Chaotic Oscillators, Journal of the Franklin Institute, 337, 251-265, 2000.

73. Yıldırım, F., Elektronik Devrelerde Yüksek Frekanslı Kaotik Osilatörlerin Tasarımı ve Gerçekleştirimi, Yüksek Lisans Tezi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, 2004.
74. Tamasevicius, A., Namajunas, A., Cenys, A., Simple 4D Chaotic Oscillator, *Electronic Lett.*, 32, 957-958, 1996.
75. Namajunas, A., Tamasevicius, A., Simple RC Chaotic Oscillator, *Electronics Letters*, 32, 945-946, 1996.
76. Endo, T., Chua, L.O., Chaos from Phase-Locked Loops, *IEEE Trans. Circuits&Systems*, 34, 987-1003, 1988.
77. Endo, T., Chua, L.O., Chaos from Phase-Locked Loops- Part II: High Dissipation Case, *IEEE Trans. Circuits&Systems*, 35, 155-263, 1989.
78. Tse, C.K., Fung, S.C., Kwan, M.W., Experimental Confirmation of Chaos in a Current-Programmed Cuk Converter, *IEEE Trans. Circuits&Systems-I*, 43, 605-608, 1996.
79. Kılıç, R., Yıldırım Dalkıran, F., Utilizing SIMULINK in Modeling and Simulation of Generalized Chaotic Systems With Multiple Nonlinear Functions, *Computer Applications in Engineering Education*, 18(4), 684-693, 2010.
80. Kılıç, R., A Practical Guide for Studying Chua' s Circuits, World Scientific, 2010.
81. Kılıç, R., Yıldırım, F., CFOA-Tabanlı Kaotik Chua Devresinin Tasarımı ve Yüksek Frekans Performansının Deneysel Olarak Doğrulanması, 11. Sinyal İşleme ve İletişim Uygulamaları Kurultayı-SIU 2003, 357-360 , 2003.
82. Kennedy, M.P., Robust Op-Amp Realization of Chua's Circuit, *Frequenz*, 46, 66-80, 1992.
83. Zhong, G.Q., Implementation of Chua's Circuit With A Cubic Nonlinearity, *IEEE Trans. on Circuits&Syst.-I*, 41, 934-941, 1994.
84. O'Donoghue K., Kennedy M.P., Forbes, P., A Fast and Simple Implementation of Chua's Oscillator Using A "Cubic-Like" Chua Diode, *Proceedings of The 2005 European Conference on Circuit Theory and Design*, 2 , II-83 – II-86, 2005.
85. Tang, K.S., Man, K.F., Zhong, G.Q., Chen, G., Generating Chaos via $x/x/$, *IEEE Trans. on Circuits&Syst.-I*, 48(1), 635-641, 2001.

86. Sobhy, M.I., Aseeri M.A., Shehata A.E.R., Real Time Implementation of Continuous (Chua and Lorenz) Chaotic Generator Models Using Digital Hardware, Third International Symposium on Communication Systems Networks and Digital Processing, CSNDSP'2002, 38-41, 2002.
87. Kılıç, R., Experimental Study of CFOA-Based Inductorless Chua's Circuit, *Int. J. Bifurcation and Chaos*, 14(4), 1369-1374, 2004.
88. Elwakil, A.S., Kennedy, M.P., Improved Implementation of Chua's Chaotic Oscillator Using Current Feedback Op-Amp, *IEEE Trans. Circuits and Syst-I*, 47(1), 76-79, 2000.
89. Senani, R., Gupta, S.S., Implementation of Chua's Chaotic Circuit Using Current Feedback Op-Amps, *Electronic Lett.*, 34(9), 829-830, 1998.
90. Kennedy, M.P., Chaos in The Colpitts Oscillator, *IEEE Trans. Circuits Systems-I*, 41, 771-774, 1994.
91. Kennedy, M.P., On The Relationship Between The Chaotic Colpitts Oscillator and Chua's Oscillator, *IEEE Trans. Circuits and Systems-I*, 42(6), 376-379, 1995.
92. Elwakil, A.S., Kennedy, M.P., A Family of Colpitts-Like Chaotic Oscillators, *Journal of The Franklin Institute*, 336, 687-700, 1999.
93. Tamasevicius, A., Bumeliene, S., Mykolaitis, G., Evaluation of Bipolar Transistors for Application to RF Chaotic Colpitts Oscillator, *Proc. of The 42nd Int. Conf. of Riga Technical University*, 53- 54, 2001.
94. Baziliauskas, A., Tamaševičius, A., Bumelienė, S., Lindberg, E., Synchronization of Chaotic Colpitts Oscillators, *Proc. of the 42nd Int. Conf. of Riga Technical University*, 55-58, 2001.
95. Volos, C.K., Kyprianidis, I.M., Stouboulos, I.N., Experimental Study of A Nonlinear Circuit Described by Duffing's Equation, *Journal of İstanbul Kültür University*, 4, 45-54, 2006.
96. Lindberg, E., Tamaševičiūtė, E., Mykolaitis, G., Bumelienė, S., Pyragienė, T., Tamaševičius, A., Kirvaitis, R., Autonomous Third-Order Duffing-Holmes Type Chaotic Oscillator, *European Conference on Circuit Theory and Design-ECCTD 2009*, 2009.

97. Tamaševičiūtė, E., Tamaševičius, A., Mykolaitis, G., Bumelienė, S., Lindberg, E., Analogue Electrical Circuit for Simulation of The Duffing-Holmes Equation, *Nonlinear Analysis: Modelling and Control*, 13(2), 241–252, 2008.
98. Volos, C.K., Kyprianidis, I.M., Stouboulos, I.N., Synchronization of Two Mutually Coupled Duffing – Type Circuits, *International Journal of Circuits, Systems and Signal Processing*, 1(3), 274-281, 2007.
99. Elwakil, A.S., Salama, K.N., Kennedy, M.P., A System for Chaos Generation and Its Implementation in Monolithic Form, In *Proceedings of IEEE Int. Conf. of Circuits and Systems- ISCAS 2000*, V217-V220, 2000.
100. Elwakil, A.S., Kennedy, M.P., Construction of Classes of Circuit-Independent Chaotic Oscillators Using Passive-Only Nonlinear Devices, *IEEE Trans. Circuits and Systems-I*, 48(3), 289-307, 2001.
101. Cuomo, K.M., Oppenheim, A.V., Strogatz, S.H., Synchronization of Lorenz-Based Chaotic Circuits With Applications to Communications, *IEEE Trans. Circuits and Systems-II*, 40(10), 626-633, 1993.
102. Corron N.J., A Simple Circuit Implementation of a Chaotic Lorenz System, <http://ccweb.org/documents/physics/chaos/LorenzCircuit3.html>
103. Blakely, J.N., Eskridge, M.B., Corron, N.J., A Simple Lorenz Circuit and Its Radio Frequency Implementation, *Chaos*, 17, 023112-1 - 023112-5, 2007.
104. Murali, K., Lakshmanan, M., Chua, L.O., Bifurcation and Chaos in The Simplest Dissipative Non-Autonomous Circuit, *Int. Journal of Bifurcation and Chaos*, 6, 1511–1524, 1994.
105. Murali, K., Lakshmanan, M., Synchronization Through Compound Chaotic Signal in Chua's Circuit and Murali-Lakshmanan-Chua Circuit, *Int. Journal of Bifurcation and Chaos*, 7, 415–421, 1997.
106. Çam, U., Kılıç, R., Inductorless Realization of MLC Circuit Using Current-Feedback Operational Amplifiers, *Journal of Circuits, Systems and Computers*, 14, 99-107, 2005.
107. Kılıç, R., Yıldırım, F., Wien Köprü Tabanlı Karışık Modlu Kaotik Devre Modeli (W-MMCC), *Elektrik-Elektronik-Bilgisayar Mühendisliği 10. Ulusal Kongresi*, 334-337, 2003.

108. Çam, U., Inductorless Realization of Mixed-Mode Chaotic Circuit Using Current-Feedback Operational Amplifiers, The 10th Workshop on Nonlinear Dynamics of Electronic Systems-NDES 2002, 69-72, 2002.
109. Rössler, O.E., An Equation for Continuous Chaos, *Physics Letters*, 57(5), 397–398, 1976.
110. Carroll, T.L., A Simple Circuit for Demonstrating Regular and Synchronized Chaos, *American Journal of Physics*, 63(4), 377-379, 1995.
111. Yoshinaga, T., Kitajima, H., Kawakami, H., Bifurcation in A Coupled Rössler System, *IEICE Trans. Fundamentals*, 78(10), 1276-1280, 1995.
112. Pisarchik, A.N., Jaimes-Reátegui, R., Homoclinic Orbits in A Piecewise Linear Rössler-Like Circuit, *International Conference on Control and Synchronization of Dynamical Systems*, 122-127, 2005.
113. García-Lopez, J.H., Jaimes-Reátegui, R., Pisarchik, A.N., Murguía-Hernandez, A., Medina-Gutiérrez, C., Valdivia-Hernandez, R., Villafana-Rauda, E., Novel Communication Scheme Based on Chaotic Rössler Circuits, *International Conference on Control and Synchronization of Dynamical Systems*, 276-284, 2005.
114. Sprott, J.C., A New Class of Chaotic Circuit, *Physics Letters A*, 266, 19-23, 2000.
115. Sprott, J.C., Simple Chaotic Systems and Circuits, *Am. J. Phys*, 68, 758-763, 2000.
116. Yıldırım, F., Uzunhisarcıklı, E., Kılıç, R., Alçı, M., Experimental Verification of High Frequency Performance of FTFN-Based Simple Chaotic Circuit, *Int. Journal of Bifurcation & Chaos*, 15(1), 191-205, 2005.
117. Uzunhisarcıklı, E., Yıldırım, F., Alçı, M., Experimental Verification of High Frequency Performance of CFOA-Based RC Chaotic Circuit, *2nd International Symposium on Electrical, Electronics and Computer Engineering (NEU-CEE 2004)*, 2004.
118. Günay, E., Voltaj ve Akım-Modlu Olarak Tasarlanan Hücresel Sinir Ağları ve Uygulamaları, *Doktora Tezi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü*, 2005.
119. Joy, M.P., Tavşanoğlu, V., An Equilibrium Analysis of CNN's, *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 45(1), 1998.

120. Chua, L.O., Roska, T., Cellular Neural Networks and Visual Computing, Cambridge University Press, The Pitt Building, Trumpington, Cambridge, England, 2002.
121. Cimagalli, V., Balsi, M., Cellular Neural Network: A Review, Neural Nets WIRN Vietri-93, World Scientific, Singapore, 55-84, 1994.
122. Chua, L.O., Yang, L., Cellular Neural Networks: Theory, IEEE Trans. Circuits&Syst., 35, 732-745, 1988.
123. Chua, L.O., CNN: A Paradigm for Complexity, World Scientific Series on Nonlinear Science, 31, World Scientific Publishing, 1998.
124. Kılıç, R., Chaos Synchronization in SC-CNN-Based Circuit and An Interesting Investigation: Can A SC-CNN-Based Circuit Behave Synchronously With The Original Chua's Circuit, International Journal of Bifurcation & Chaos, 14, 1071-1083, 2004.
125. Caponetto, R., Lavorgna, M., Occhipinti, L., Cellular Neural Networks in Secure Transmission Applications, Proc. of CNNA'96, 411-416, 1996.
126. Arena, P., Baglio, S., Fortuna, L., Manganaro, G., Chua's Circuit Can Be Generated by CNN Cells, IEEE Trans. Circuits&Syst. -I, 42(2), 123-125, 1995.
127. Günay, E., Alçı, M., Kılıç, R., Durum Kontrollü Hücreli Sinir Ağı (DK-HSA) Devresi Kullanılarak Geribeslemeli Kaotik Maskeleye Sisteminin Tasarlanması, Elektrik-Elektronik Bilgisayar Mühendisliği Sempozyumu - ELECO 2002, 89-93, 2002.
128. Arena P., Baglio S., Fortuna L., Manganaro G., Chua's Circuit Can Be Generated by CNN Cells, IEEE Tran on Circuits and Systems-I, 42(2), 123-125, 1995.
129. Günay, E., Alçı, M., Yıldırım, F., An Experimental Study on Chaotic Dynamics of CFOA-Based SC-CNN Circuit, Int. J. Bifurcation and Chaos, 15(8), 2551-2558, 2005.
130. Pico Technology Limited:www.picotech.com.
131. Suykens, J.A.K., Huang, A., Chua, L.O., A Family of n-Scroll Attractors from A Generalized Chua's Circuit, Archiv für Elektronik und Übertragungstechnik, 51(3), 131-138, 1997.

132. Yalçın, M.E., Suykens, J.A.K., Vandewalle, J., Experimental Confirmation of 3- and 5-scroll Attractors from A Generalized Chua's Circuit, *IEEE Trans. Circuits&Syst.-Part I: Fundamental Theory and Appl.*, 47(3), 425-429, 2000.
133. Pecora, L.M., Carroll, T.L., Synchronization in Chaotic Systems, *Phys. Rev. Letters*, 64, 821-824, 1990.
134. Murali, K., Lakshmanan, M., Synchronization Through Compound Chaotic Signal in Chua's Circuit and Murali-Lakshmanan-Chua Circuit, *Int. Journal of Bifurcation and Chaos*, 7, 415-421, 1997.
135. Hwang, C.C., Hsieh, J.Y., Lin, R.S., A Linear Continuous Feedback Control of Chua's Circuit, *Chaos Solitons Fractals*, 8, 1507-1515, 1997.
136. Agiza, H.N., Yassen, M.T., Synchronization of Rössler and Chen Chaotic Dynamical Systems Using Active Control, *Phys. Lett. A*, 278, 191-197, 2001.
137. Xiang-Jun, W., Jing-Sen, L., Guan-Rong, C., Chaos Synchronization of Rikitake Chaotic Attractor Using The Passive Control Technique, *Nonlinear Dyn.*, 53, 45-53, 2008.
138. Kılıç R., Alçı M., Günay E, Two Impulsive Synchronization Studies Using SC-CNN-Based Circuit and Chua' s Circuit, *International Journal of Bifurcation & Chaos*, 14, 3277-3293, 2004.
139. Itoh, M., Yang, T., Chua, L.O., Experimental Study of Impulsive Synchronization of Chaotic and Hyperchaotic Circuits, *Int. J. Bifurcation and Chaos*, 9, 1393-1424, 1999.
140. Yang, T., Yang, L-B., Yang, C-M., Impulsive Synchronization of Lorenz Systems, *Physics Letters A*, 226, 349-354, 1997.
141. Chua, L.O., Itoh, M., Kocarev, L., Eckert, K., Chaos Synchronization in Chua's Circuit, *J. of Circuits, Systems and Computers*, 3(1), 93-108, 1993.
142. Lakshmanan, M., Murali K., *Chaos in Nonlinear Oscillators: Controlling and Synchronization*, World Scientific Series on Nonlinear Science, 13, World Scientific Publishing, 1996.
143. Pecora, L.M., Carroll, T.L., Driving Systems With Chaotic Signals, *Physical Rev. A*, 44, 2374-2383, 1991.

144. Cuomo, K.M., Oppenheim, A.V., Circuit Implementation of Synchronized Chaos with Applications to Communication, Phys. Rev. Lett., 71(1), 65-68, 1993.
145. Marchand, P., Holland, O.T., Graphics and GUIs With MATLAB, Third Edition, Chapman & Hall/CRC, CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida, 2003.
146. Alpay, E., Hesaplama Programlarının Grafiksel Kullanıcı Arayüzü (GUI) Hataları Perspektifinde İncelenmesi, Yüksek Lisans Tezi, İstanbul Kültür Üniversitesi, Fen Bilimleri Enstitüsü, 2005.
147. Ak, M., Güç Elektroniği Devrelerinin Kullanıcı Arayüzü Tasarlanarak Mesleki Eğitime Aktarılması, Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, 2009.
148. The MathWorks, Inc., MATLAB, The Language of Technical Computing, The MathWorks, Inc., Natick, MA, 2006.

ÖZGEÇMİŞ

Fatma YILDIRIM DALKIRAN 1979 yılında Karabük' te doğdu. İlk, orta ve lise öğrenimini Karabük' te tamamladı. 1996-1997 öğretim yılında Erciyes Üniversitesi Mühendislik Fakültesi Elektronik Mühendisliği Bölümünü kazandı. Bir yılı hazırlık eğitimi olmak üzere toplam 5 yıllık eğitimden sonra 2001 yılında bu bölümden mezun oldu. 2002 yılında Erciyes Üniversitesi Sivil Havacılık Yüksekokulu Uçak Elektrik-Elektronik Bölümünde araştırma görevlisi olarak göreve başladı. 2004 yılında Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektronik Anabilim Dalında yüksek lisans eğitimini bitiren Fatma YILDIRIM DALKIRAN halen Erciyes Üniversitesi Sivil Havacılık Yüksekokulu Uçak Elektrik-Elektronik Bölümünde araştırma görevlisi olarak görev yapmaktadır.

İletişim

Adres: Erciyes Üniversitesi
Sivil Havacılık Yüksekokulu
Uçak Elektrik-Elektronik Bölümü
38039 Melikgazi / KAYSERİ

Tel: (352) 437 49 01 - 41 057

E-mail: fatmay@erciyes.edu.tr