T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

FPGA TABANLI KAOS SENKRONİZASYONU İÇİN DENETLEYİCİ TASARIMI

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Onur SİLAHTAR
DANIŞMAN: Dr. Öğr. Üyesi. Özkan ATAN

VAN-2018

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

FPGA TABANLI KAOS SENKRONİZASYONU İÇİN DENETLEYİCİ TASARIMI

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Onur SİLAHTAR

VAN-2018

KABUL VE ONAY SAYFASI

Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda Dr. Öğr. Üyesi Özkan ATAN danışmanlığında, Onur SİLAHTAR tarafından sunulan "FPGA Tabanlı Kaos Senkronizasyonu için Denetleyici Tasarımı" isimli bu çalışma Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili hükümleri gereğince 18/07/2018 tarihinde aşağıdaki jüri tarafından oy birliği ile başarılı bulunmuş ve yüksek lisans tezi olarak kabul edilmiştir.

Başkan: Doç. Dr. M. Nuri ALMALI İmza:

Üye: Dr. Öğr. Üyesi Ahmet GÜNDOĞDU İmza:

Üye: Dr. Öğr. Üyesi Özkan ATAN İmza:

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 27/07/2018 tarih ve 2018/35-I sayılı kararı ile onaylanmıştır.

İmza

Enstitü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Onur SİLAHTAR

ÖZET

FPGA TABANLI KAOS SENKRONİZASYONU İÇİN DENETLEYİCİ TASARIMI

SİLAHTAR, Onur
Yüksek Lisans Tezi, Elektrik-Elektronik Mühendisliği Anabilim Dalı
Tez Danışmanı: Dr. Öğr. Üyesi Özkan ATAN
Temmuz 2018, 73 sayfa

Elektronik kaotik osilatörlerin her kaotik sistem için ayrıca tasarlanma zorunluluğu ve tasarımların zaman alması yeni bir yönteme olan ihtiyacı ortaya çıkarmıştır. Son yıllarda gömülü sistem teknolojinin gelişmesiyle, kaotik çalışmaların uygulaması kolaylaşmış ve kısa zamanda daha çok çalışma yapılmasının önü açılmıştır.

Bu çalışmada Lorenz ve Chen kaotik sistemlerine aynı şartlar altında ve aynı başlangıç koşullarında, aktif kayan kipli kontrol (active sliding mode control) ve uyarlamalı kontrol (adaptive control) yöntemleri kullanılarak senkronizasyon yapıldı. Ardından Matlab/Simulink ortamında kaos senkronizasyon modeli oluşturulup simülasyon gerçekleştirildi. Son olarak elde edilen kaos senkronizasyon modeli bu defa da Alanda Programlanabilir Kapı Dizileri'ne (Field Programmable Gate Aray - FPGA) yüklendi ve FPGA'nın, Donanım Yardımcı Simülasyonu (Hardware Co-Simulation) özelliği sayesinde gerçek zamanlı sonuçlar elde edildi. Hem ortaya çıkan gerçek zamanlı sonuçlar ile simülasyon sonuçları karşılaştırıldı hem de aktif kayan kipli kontrol ile uyarlamalı kontrol yönteminin senkronizasyon sistemi üzerinde etkileri karşılaştırıldı.

Sonuç olarak FPGA'nın gerçek zamanlı sonuçlarının Matlab/Simulink simülasyon sonuçlarıyla çok büyük oranda örtüştüğü gözlendi. Ayrıca FPGA'nın daha kolay, hızlı ve güvenilir sonuçlar verdiği gözlemlendi. Bunun yanısıra aktif kayan kipli kontrol yöntemiyle yapılan senkronizasyon sonucunda sistemin, uyarlamalı kontrol yöntemine göre aşma miktarı ve oturma süresinin ise daha az olduğu gözlemlenmiştir.

Anahtar kelimeler: Aktif kayan kipli kontrol, FPGA, Gömülü sistem, Kaos senkronizasyonu, Uyarlamalı kontrol.

ABSTRACT

CONTROLLER DESİGN FOR FPGA-BASED CHAOS SYNCHRONİZATİON

SİLAHTAR, Onur
M.Sc. Thesis, Electrical-Electronics Engineering
Supervisor : Asst. Prof. Dr. Özkan ATAN
July 2018, 73 pages

The necessity of designing electronic chaotic oscillators separately for each chaotic system and design takes time, reveals the need for a new method. With the development of embedded system technology in recent years, the application of chaotic studies has become easier and more study can be done in a short time.

In this study, Lorenz and Chen chaotic systems were synchronized under the same conditions and under the same initial conditions, using active sliding mode control and adaptive control methods. Then, in the Matlab / Simulink program, a chaos synchronization model was created and simulated. Finally, the chaos synchronization model was loaded into the Field Programmable Gate Array (FPGA) and real-time results were achieved thanks to the FPGA's Hardware Co-Simulation feature. Simulation results were compared with real time results, and the effects of active sliding mode control and adaptive control method on synchronizing system were compared.

As a result, the real-time results of FPGA have been observed to overlap with the results of Matlab / Simulink simulation. It has also been observed that the FPGA provides easier, faster and more reliable results. In addition, it has been observed that the synchronized system with active sliding mode control method has less overshoot value and less settling time than the adaptive control method.

Keywords: Active sliding mode control, FPGA, Embedded system, Chaos synchronization, Adaptive control.

ÖNSÖZ

Bu tez çalışmasında, özellikle her türlü ilgi ve yardımlarını esirgemeyen danışmanım Sayın Dr. Öğr. Üyesi. Özkan ATAN' a teşekkürlerimi bir borç bilirim. Çalışmalarım sırasında bilgi ve tecrübelerini esirgemeyen tüm Elektrik-Elektronik Mühendisliği Bölümü Hocalarıma, mesai arkadaşlarıma ve manevi desteklerinden dolayı tüm aileme teşekkürlerimi sunarım.

2018

Onur SİLAHTAR

İÇİNDEKİLER

Sayfa

Sayfa

ŞEKİLLER LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklama**
C	C Programlama Dili
J	Ölçeklenmiş Matris
sgn	Signum Fonksiyonu
lim	Limit Alma Fonksiyonu

Kısaltmalar	**Açıklama**
ASIC	Bir Bilgisayar Programlama Dili
FFT	Hızlı Fourier Dönüşümü
FONECNR	Kesir Dereceli Denge Noktası Olmayan Kübik Nonlineer Rezistör Sistemi
FONCCNR	Kesir Dereceli Kübik Nonlineer Rezistör Sistemi
FPGA	Alanda Programlanabilir Kapı Dizileri
HDL	Donanım Tanımlama Dili
HMLs	Henon Haritası Örgüleri
MIT	Massachusetts Teknoloji Enstitüsü
NCCNR	İki Adet Denge Noktasına(equilibrium) Sahip Kaotik Bir Sistem
NECNR	Denge Noktası Olmayan Kaotik Kübik Nonlineer Rezistör Sistemi
RASMC	Gürbüz Bir Uyarlamalı Kayan Kipli Denetleyici
SMC	Kayan Kipli Kontrol
VHDL	Çok Yüksek Hızlı Donanım Tanımlama Dili

1. GİRİŞ

Nonlineer ve öngörülemeyen sürprizlerin bilimi olarak tanımlanan kaos bize aslında öngörülebilirliği ve öngörülemezliği anlatır. Örneğin çoğu geleneksel bilim dalı, yerçekimi, elektrik, kimyasal reaksiyon gibi öngörülebilir, kolay hesaplanabilir durumlarla ilgiliyken kaos; türbülans, hava tahmini, beyin faaliyetleri gibi kontrölü ve öngörülebilirliği neredeyse imkansıza yakın nonlineer yapıya sahip durumlarla ilgilenir. Bu fenomen genellikle doğanın sonsuz kompleksliğini konu alan parçalanmış (fraktal) matematik yardımıyla incelenir (Tufan, 2017). Tabiat, bulutlar, ağaçlar, organlar, nehirler ve bunun gibi içerisinde yaşadığımız sistemlerin çoğu kompleks ve kaotik bir davranış sergiler. Örneğin; yıldırım düşmesi olayını gözönüne aldığımızda yıldırım düşeceği yeri ve şiddeti tam olarak saptamak mümkün değildir. Çünkü yıldırımı meydana getiren faktörlerin tamamı bilinemez formülize edilemez ve sözkonusu faktörlerin birbirlerini ne oranda etkilediği tam olarak hesaplanamaz.

Yukarıda sözü edilen durumların varlığı yeni bir araştırma konusunun ortaya çıkmasını sağlamıştır. Bahsi geçen sistemler "kaotik sistemler" başlığı altında incelenmiş, matematiksel ifadeler kullanılarak tanımlanmaya çalışılmış ve mevcut fiziksel sistemlerde etkileri gözlenmeye başlanmıştır. 1950 lerin sonunda Massachusetts Teknoloji Enstitüsünden (MIT) Edward Lorenz, kendi adını verdiği Lorenz denklemleriyle kaos'u matematiksel bir sistem olarak göstermeyi başarmıştır (Lorenz, 1963). Lorenz'in bu çalışmasından sonra kaos'un gözlemlenebilir, hesaplanabilir, kontrol edilebilir olmasının yolu açılmıştır. Lorenz'den sonra birçok kaotik denklem sistemi üretilmiş olup "kaotik osilatörler" tasarlanmaya başlamıştır. Tasarlanan bu sistemler bilgisayar simülasyonları yardımıyla "kaotik sinyalleri" üretmiş olup bu sinyallerin görüntülenebilmiş ve kontrol edilebilmiştir. Ancak 1980 yılına kadar sadece bilgisayar simülasyonları yardımıyla kaos incelenebilmiş ve fizksel olarak bir kaotik üreteç diğer bir deyişle kaotik osilatör tasarlanamamıştır.

1980 yılında Kaliforniya Üniversitesinden Leon O. Chua, "Chua Devresi" olarak bilinen ilk elektronik kaotik elektrik devresini tasarlamıştır (Chua, 1980). Elektronik devre elemanları kullanılarak tasarlanan bu devre sayesinde kaos sinyallerinin fiziksel olarak üretimi sağlanmış ve başta şifreli haberleşme olmak üzere fizik,

otomotiv, elektronik, vb. mühendisliğin diğer birçok alanında varolan kaosun modellenmesinde kullanılmaya başlanmıştır.

Kaosun en önemli uygulama alanlarından biri olan haberleşme alanında "kaos senkronizasyonu" konusu önemli bir yer kaplamaktadır. Kaos senkronizasyonu temel olarak iki kaotik sistemin birbirleriyle uyumlu hale getirilip örtüştürülme olayına denir. Başka bir deyişle aynı olmayan iki kaotik sistemden biri referans olarak alınır diğer kaotik sistem ise referans alınan kaotik sisteme benzetilir. Kaotik sistemlerde senkronizasyon konsepti ilk olarak Pecora ve Carroll tarafından hazırlanmıştır (Pecora ve Carroll, 1990). Maskelenmiş iki kaotik sinyal bir elektronik devre yardımıyla üretilip aynı kanal üzerinden ve bozulmadan alıcıya iletilmiştir. Bu çalışmadan itibaren senkronizasyonun verimli, bozuculardan mümkün oldukça az etkilenen, gürbüz bir halde gerçekleşmesini sağlamak için birçok kontrol yöntemi kullanılmaya başlanmıştır. Günümüzde de özellikle haberleşme sektöründe senkronizasyon temelli uygulamalar kullanılmaya devam etmektedir.

Son yıllarda kaos ve senkronizasyon temelli uygulamaların artması ve malzeme bilimindeki gelişmeler sayesinde kaotik osilatör kaynağı olarak elektronik devre osilatörlerine alternatif başka bir kaynak ortaya çıkmştır. "Gömülü Sistemler" olarak bilinen bu sistemler içlerinde bulunan dijital elemanlar (dijital kapılar) yardımıyla kaotik sinyaller üretebilmişlerdir. Bunlardan en bilineni FPGA gömülü sistemleridir.

Analog devre osilatörlerine oranla, FPGA'nın hızlı, güvenilir ve kolay tasarlanabilir olması hem kaos alanında çalışmaların daha hızlı tamamlanmasını hemde yeni yöntemler geliştirmenin kolaylaşmasını sağlamıştır. Ayrıca FPGA'nın özellikle radyasyona ve diğer iç ve dış bozuculara karşı dayanıklı olması savunma sanayi, uzay araştırmaları, otomotiv gibi hassas üretim yapılan alanlarda kullanılmasının önünü açmıştır.

Bu çalışmada FPGA kullanılarak kaos senkronizasyonu uygulamalı olarak gerçekleştirildi. Bunun için Lorenz ve Chen kaotik sistemleri seçilip her birine ayrı ayrı aktif kayan kipli ve uyarlamalı kontrol yöntemleri tatbik edilerek kaos senkronizasyonu gerçekleştirildi. Senkronizasyon türü olarak komple (complete) senkronizasyon yöntemi seçilmiş olup Matlab/Simulink yardımıyla tasarlanan sistem, bir gömülü sistem olan FPGA'ya yüklenmiştir. FPGA'nın "hardware co-simulation" özelliği sayesinde FPGA'dan gerçek zamanlı sonuçlar alınıp aktif kayan kipli ve uyarlamalı kontrol

yöntemlerinin, seçilmiş Lorenz ve Chen kaotik sistemleri üzerindeki avantajları ve dezavantajları tartışılmıştır.

2. KAYNAK BİLDİRİŞLERİ

Lorenz'in kaotik sinyalleri formülize edip Chua'nın bu kaotik sinyalleri üretebilen kaotik osilatör devresi tasarlaması ve Pecora ve Carroll'un kaos senkronizasyonunu gerçekleştirmesinden sonra kaos konusuyla ilgili birçok çalışma yapılmıştır. Çeşitli kontrol yöntemleri kullanılarak birçok kaotik sistem senkronize edilmiş ve farklı denetleyiciler tasarlanmıştır. Ayrıca elektronik devre elemanları kullanılarak oluşturulan kaotik osilatör devrelerine alternatif olarak FPGA gibi gömülü sistemler de kullanılarak kaotik sinyaller üretilip senkronizayon çalışmaları yapılmıştır.

Zhao ve ark. (2014), uyarlamalı kontrol temelli bilinmeyen parametrelere sahip bir ağ üzerindeki birleştirilmiş sistemler arasında kaos senkronizasyonu yapılabileceği önerisini sundu. Biçimlendirici (former) N adet sistemin birbirleriyle ve daha sonra diğer N adet diğer sistemle birleştirildiği sistemlerden oluşan 2N adet (hiperkaotik) sistemler arasında ağ senkronizasyonu tanımlandı. Daha sonra senkronizasyonun başarılı olmasını sağlamak için 2N adet birleştirilmiş kaotik sisteme sahip dinamik sistemlerin kararlılık teorisine dayalı olarak uygun şartlar elde edildi. Uyarlamalı kontrol tekniği kullanılarak kontrol kuralları ve uyumlu güncellenmiş parametre kuralları oluşturuldu. Böylece özdeş olmayan veya hiperkaotik sistemlerin ağ senkronizasyonu elde edildi.

Bilinmeyen parametrelere sahip kaotik kompleks nonlineer sistemlerin uyarlamalı anti-senkronizasyon yöntemi geliştirildi. Lyapunov kararlılık teoremi baz alınarak iki kaotik kompleks sistemi anti-senkronize etmek için bir uyarlamalı kontrol şeması ve parametrelerin uyarlamalı kuralları geliştirildi. Sunulan şemaların etkinliğini ve uygulanabilirliğini kanıtlamak için iki özdeş kompleks Lorenz sisteminin ve iki farklı kompleks Chen ve Lü sistemlerinin anti-senkronizasyonu incelendi (Liu ve ark., 2011).

Chen ve Qiu (2013), uyarlamalı kontrol yöntemiyle N adet farklı kaotik sistemin komple (complete) senkronizasyonunu gerçekleştirdi. Hata sistemini, özel bir asimetrik yapıya sahip nonlineer bir sisteme dönüştürmek için bir senkronizasyon denetleyicisi tasarlandı. Gerekli kararlılık şartları sunuldu ve kaotik sistemlerin komple senkronizasyonu gerçekleştirildi.

Thang ve ark. (2016), birleştirilmiş kompleks ağların küme (cluster) senkronizasyonu için bir uyarlamalı problemi önerdi. Zamanla değişen gecikmeli birleşmelere, rastgele (stochastic) bozulmalara ve farklı kümelerdeki özdeş olmayan nodlara (nodes) sahip kompleks ağlar için küme senkronizasyonu keşfedildi. Denetleyicileri tanımlamak için, rastgele oluşan denetleyiciler baz alınarak bazı Bernoulli rastgele değişkenleri oluşturuldu. Daha sonra diğer kümelere direkt bağlantıları olan kümelerin içindeki nodların kısımları kontrol edildi ve tüm dinamik ağların birleşik durumlarının global olarak amaçlanan küme durumlarına dönüşmesi sağlandı. Lyapunov kararlılık teoremi yardımıyla tüm başlangıç koşulları için ortalama kare küme senkronizasyonu modelinin gerçekleşmesini garanti edecek gerekli koşullar üretildi.

Karışık gecikmelere (mixed delays) ve özdeş olmayan düzensizliklere sahip birleştirilmiş süreksiz sinirsel ağların sonlu-zamanlı (finite-time) senkronizasyonunu gerçekleştirildi. Öncelikle uyarlamalı sabitleme (adaptive pinning) kontrol stratejisi altında nonlineer bir şekilde birleştrilmiş Lur ağları (Lur'e networks) küme senkronizasyonunun gerçekleşmesi amaçlanmıştır. Özdeş ve özdeş olmayan Lur sistemlerini içeren zamanla değişen gecikmeli ağlar, uygulama kenar bazlı sabitlenmiş kontrol şemaları yardımıyla birbirleriyle ilişkilendirildi. Her bir kümede, kapsama ağacına (spanning tree) ait kenarlar sabitlendi. Ağların nonlineer bir şekilde birleşmesinden ötürü, nodların dinamik davranışlarının lokal bilgilerine dayalı etkili bir nonlineer uyarlamalı güncellenmiş kural oluşturulmuş oldu (Tang ve ark., 2015).

Kayan kipli kontrol şemaları kullanılarak birleştirilmiş "Henon Haritası Örgüleri (HMLs)" ile mekânsal-zamansal (spatiotemporal) kaotik senkronizasyonu gerçekleştirildi. Uygulanan yaklaşımın, sistemin parametre uyumsuzluğuna karşı gürbüz ve mühendislik uygulamaları için avantajlı olduğu görüldü. Nümerik simülasyonlar, uygulanan kontrol stratejisinin geçerliliğini ve gürbüzlüğünü gösterdi (Yin ve ark., 2002).

Pourmahmood ve ark. (2011), kararsızlıklara, harici bozuculara ve tamamen bilinmeyen parametrelere sahip iki farklı kaotik sistem arasında kaos senkronizasyonu gerçekleştirmek için gürbüz bir uyarlamalı kayan kipli denetleyici (RASMC) tasarladı. Hem master hem de slave kaotik sistemlerinin kararsızlıklar, harici bozucular ve bilinmeyen parametreler tarafından etkilendiği, ayrıca kararsızlıkların ve harici

bozucuların sınırlarının da önceden bilinmediği varsayıldı. Kararsızlıkları, harici bozucuları ve bilinmeyen parametreleri ortadan kaldırmak için uygun güncellenmiş kontrol kuralları tasarlandı. RASMC'yi oluşturmak için öncelikle basit bir kayma yüzeyi tasarlandı. Daha sonra kayan kipli (sliding) hareket oluşumunu sağlayacak RASMC elde edildi. Uygulanan RASMC'nin gürbüzlüğü ve kararlılığı Lyapunov kararlılık teorisi kullanılarak sağlandı. Son olarak kararsızlıklara, harici bozuculara ve bilinmeyen parametrelere sahip Lorenz–Chen, Chen–Lorenz, ve Liu–Lorenz kaotik sistemlerine ikili olarak RASMC uygulanarak kaos senkronizasyonu gerçekleştrildi. Nümerik simülasyonlar yardımıyla RASMC'nin verimliliği ve gürbüzlüğü incelendi.

Nonlineer kararsız kaotik sistemlere uygulanacak kaos kontrolü ve senkronizasyon için güncel bir çatırdamasız (chatter free) kayan kipli kontrol (SMC) yöntemi tasarlandı. Süreksiz signum fonksiyonunun anahtarlamasını kontrol girişimnin ilk türevine doğru yönlendirmek için hem integral hem de diferansiyel operatörler içeren yeni bir kayma yüzeyi oluşturuldu. Böylece kararsız kaotik sistemler için çatırdamasız kontrol girişi elde edildi. Lyapunov kararlılık teorisi ve SMC yöntemi baz alınarak kararlılık analizi gerçekleştirildi ve çatırdamasız kayan kipli kontrol girişi tasarımı olarak sunulan bir teorem ortaya çıktı. Simülasyon kısmında ise kaos kontrolü ve senkronizasyon ile ilgili sonuçlar uygulanan stratejinin, kararsız kaotik sistemlerin istenildiği gibi hızlı bir şekilde kontrol edilebildiğini gösterdi. Uygulanan yöntem ile çatırdamaların nasıl elimine edildiğini göstermek için geleneksel SMC ve geliştirilen sözkonusu yöntem ayrı ayrı kullanılıp sonuçlar karşılaştırıldı. Simülasyon sonuçları sözkonusu yöntemin geleneksel SMC yöntemine göre daha verimli olduğunu gösterdi (Li ve ark., 2011).

Rajagopal ve ark. (2017), kübik nonlineer rezistöre sahip iki adet kesir dereceli kaotik sistemi ve bunların uyarlamalı kayan kipli senkronizsayonunu FPGA yardımıyla gerçekleştirdi. Öncelikle kübik nonlineer rezistöre ve iki adet denge noktasına (equilibrium) sahip kaotik bir sistem (NCCNR) elde edildi ve dinamik özellikleri araştırıldı. Daha sonra kesir dereceli kübik nonlineer rezistör sistemi (FONCCNR), tamsayı cinsi (integer-order) modelinden elde edildikten sonra kararlılığı ve kesir dereceli dallanmaları (bifurcation) incelendi. Ardından NCCNR sisteminden, yeni bir denge noktası olmayan kaotik kübik nonlineer rezistör sistemi (NECNR) elde edilip bu sistemin dinamik özellikleri gözlemlendi. Daha sonra NECNR sisteminden kesir

dereceli denge noktası olmayan kübik nonlineer rezistör sistemi (FONECNR) elde edilip yine bu sistemin kararlılığı ve kesir dereceli dallanmaları (bifurcation) incelendi. Ardından özdeş olmayan FONCCNR ve FONECNR sistemlerinin uyarlamalı kayan kipli senkronizasyonu gerçekleştirildi. Son olarak elde edilen sistem, kontrol kuralları, kayma yüzeyleri ve uyarlamalı denetleyici FPGA içine gömüldü.

Parametrelerin değerlerine göre kendinden uyarmalı (self-excited) ve gizli (hidden) çekici (attractor) olarak özelliği değişen çekicileri bulunduran kaotik sistemler elde edilerek bu sistemlerin dinamik özellikleri araştırıldı. Devre elemanları kullanılarak geliştirilen bu yeni elektronik sistem tasarlandıktan sonra FPGA içine gömülerek FPGA temelli kaotik osilatör elde edildi (Rajagopal ve ark., 2017).

Karthikeyan ve ark. (2015), Lyapunov kararlılık teoremine uygun olarak uyarlamalı güncellenmiş parametre kuralları ve uyarlamalı kontrol şeması önerdi. Burada amaç bilinmeyen parametrelere sahip yeni bir kaotik sistemin senkronizasyonunu yapmaktı. Kaotik sistemin senkronizasyonu için geri besleme kontrol sistemi komple (complete) senkronizasyon yaklaşımı kullanılarak gerçekleştirilmiştir. Uygulanan yöntemde; her bir sistem Matlab/Simulink blokları yardımıyla tasarlanmış ve Matlab/Simulink tasarımları "Sistem Üreteci (System Generator)" tasarımına dönüştürülmüştür. Böylece FPGA'nın kullanacağı "bitstream" program uzantısı oluşturulmuştur. Son olarak "bitstream" program uzantısı yardımıyla tasarım, FPGA'nın içine gömülmüş ve böylece kaotik sistem senkronizasyonunun uygulaması FPGA yardımıyla gerçekleştirilmiştir.

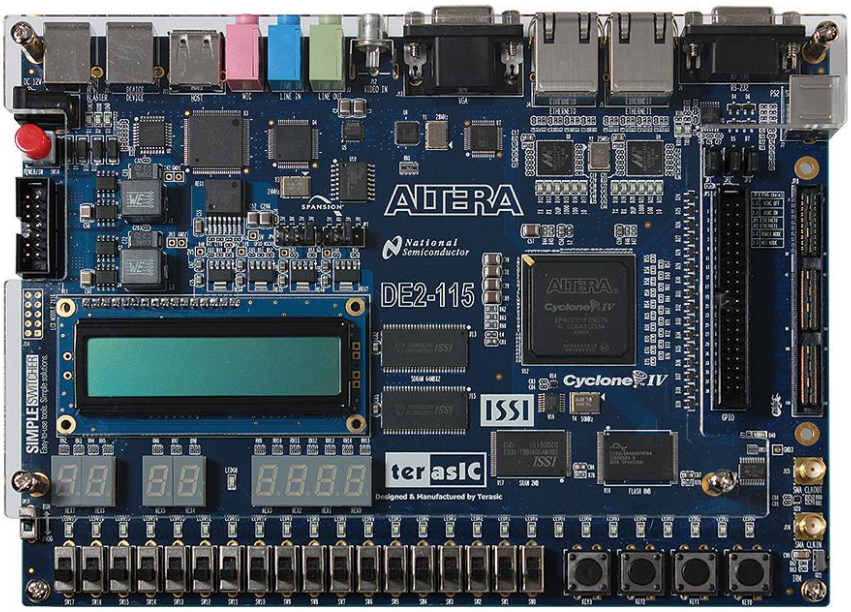
3. MATERYAL VE YÖNTEM

Bu çalışmada master ve slave sistemleri olarak sırasıyla Lorenz ve Chen kaotik denklem sistemleri tercih edildi ve bu sistemlerin matematiksel denklem modelleri çıkarıldı. Ardından uyarlamalı ve aktif kayan kipli kontrol parametreleri ve denklemleri elde edildi. Matlab/Simulink programı kullanılarak bu kontrol yöntemlerinin her biri için ayrı ayrı senkronizasyon Matlab/Simulink modelleri oluşturuldu. Bu modeller oluşturulurken daha öncesinde Matlab/Simulink programının kütüphanesine eklenmiş "xilinx bloksets" yardımıyla matematiksel işlem blokları kullanıldı ve modeller çalıştırıldıktan sonra çıkış grafikleri simülasyon ortamında gözlemlendi. Daha sonra FPGA, bilgisayara tanıtılıp bağlandıktan sonra her bir kontrol yöntemi için ayrı ayrı elde edilmiş modeller, "system generator" yardımıyla dijital işaret haline getirilip FPGA'ya gönderildi. Bunun için "Kintex7 xc7k325t-2ffg900" model FPGA kullanıldı. FPGA'ya gönderilen bu işaretler FPGA tarafından gerçek zamanlı olarak senkronize edilip sinyal üretilmesini ve sinyalin çıkış grafiğinin tekrar FPGA'nın "hardware co-simulation" özelliği sayesinde Matlab/Simulink üzerinde görüntülenmesi sağlandı. Böylelikle farklı kontrol yöntemleri kullanılarak ve gerçek zamanlı olarak elde edilen senkronize edilmiş sistemin çıkış grafikleri arasında aşma miktarı, oturma zamanı ve verimliliklerinin birbirleriyle karşılaştırılıp bir sonuca varılması sağlandı.

3.1 FPGA

Alanda Programlanabilir Kapı Dizileri "Field Programmable Gate Array - FPGA" günümüzde tüketici elektroniğinden özellikle uzay ve savunma sanayisinden, otomotiv sanayisi alanlarına kadar çok farklı alanlarda kullanılmaktadır. FPGA'i, üretimden sonra istenen fonksiyona göre donanım yapısı kullanıcı tarafından değiştirilebilen entegre devreler olarak tanımlayabiliriz. FGPA bir tür entegredir fakat onu diğer entegrelerden ayıran yanı donanım yapısını yani iç konfigürasyonunu istediğimiz gibi değiştirebilmemizdir. FPGA'i, içindeki transistörleri birbirinden bağımsız ve serbest olarak üretilmiş ham bir entegre olarak düşünebiliriz. Bizim belirlediğimiz fonksiyona göre FPGA içindeki transistörler birbirlerine bağlanır ve bu

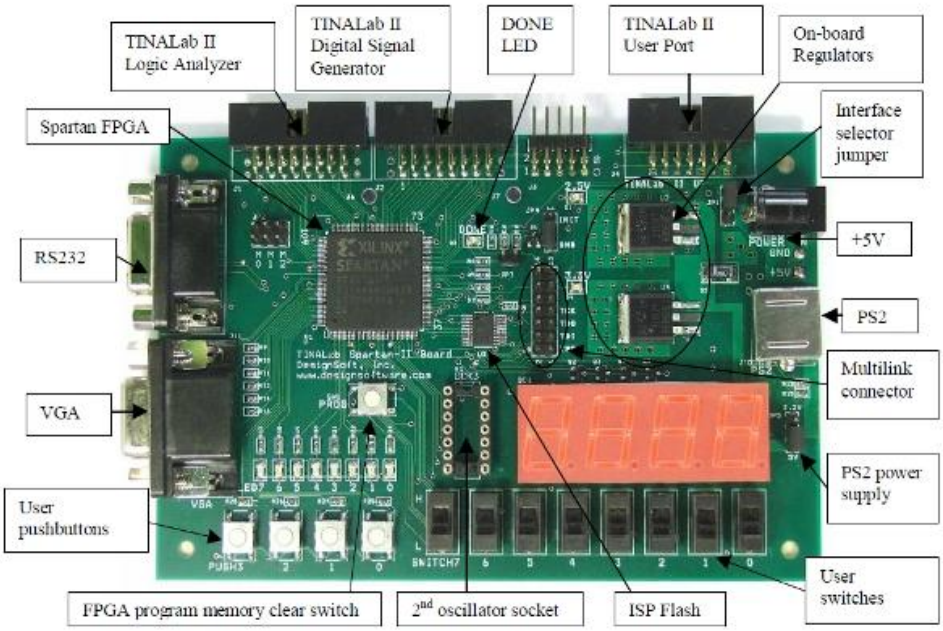
sayede istenilen fonksiyonu gerçekleştirir. Yani teorik olarak transistör kapasitesi dahilinde aklımıza gelen herhangi bir entegrenin yaptığı işi FPGA ile yapabiliriz. FPGA'lerin en önemli özelliklerinden biri de paralel işlem yapabilme yeteneğidir. Paralel işlem yapabilmek aynı anda birden fazla işlemi yapabilmek demektir. Sıradan entegreler ya hiç paralel işlem yapamazlar ya da çok sınırlı yapabilirler. FPGA'de ise uygulamaya ve kapasiteye göre, birbirine paralel onlarca belki binlerce işlemi aynı anda yapabiliriz. Bu da paralel işlem gerektiren uygulamalarda FPGA'leri eşsiz kılmaktadır. Örnek bir FPGA gösterilmiştir (Şekil 3.1).



Şekil 3.1. Örnek bir FPGA modeli (Memeşil, 2015).

3.2 FPGA Mimarisi ve Özellikleri

FPGA'lar birçok firma tarafından üretilir ve birbirinden farklı yapıda bulunabilirler. Bu farklılıklara rağmen cihazlar arasında tasarım mimarisi açısından birçok benzerlik bulunmaktadır. FPGA; programlanabilir mantık blokları, bu blok dizisini çevreleyen giriş-çıkış blokları ve ara bağlantılar olmak üzere düzenlenebilir üç ana bölümden oluşmuştur. FPGA üzerindeki bazı devre elemanları gösterilmiştir (Şekil 3.2).



Şekil 3.2. FPGA ve üzerindeki bazı devre elemanları (Kılıçkır, 2014).

Mantık bloklarında lojik operasyonları gerçekleştirmeye yarayan LUT'lar (Look-up Table) ile bunların sonuçlarını saklayacak hafıza elemanları (Flip-Flop) bulunmaktadır. Ara bağlantılar geliştiricinin ihtiyacına göre mantık blokları arasında bağlantıları sağlayarak istenen fonksiyonun çalışmasını sağlar. Giriş-çıkış portları ise FPGA'in dışarıyla iletişim kurarak veri aktarabilmesini sağlar.

3.3. FPGA Programlama Dilleri

FPGA tasarımını oluşturmak için kullanıcı tarafından ya HDL dili ya da şematik dizayn oluşturulmalıdır. Günümüzde FPGA tasarlamak için kullanılan en yaygın yöntem HDL (donanım tanımlama dili)'dir. Bundan farklı olarak verilog ve VHDL dilleri de kullanılmaktadır. Bu diller arasında çok büyük bir fark yoktur. Ayrıca AHDL, ABEL, CUPL dilleri diğer donanım tanıma dillerine örnek verilebilir.

FPGA'in iç konfigürasyonunu belirlemek için kullanılan dillere HDL denmektedir. Bu dillerinin başlıcaları Çok Yüksek Hızlı Entegre Devre Donanım Tanımlama Dili (Very High Speed Integrated Circuit Hardware Description Language – VHDL) ve Verilog'dur. Verilog, C'ye benzeyen bir yapıya sahipken, VHDL daha alt seviye bir dil gibidir. Ama bu diller geleneksel programlama dillerinden tamamıyla

farklı olup, bir akış ve işleyişten çok donanımın yapısını yani elemanların birbiriyle bağlantısını ve entegrenin iç konfigürasyonunu ifade etmektedirler.

Bir projeye başlamadan önce ihtiyaçlar belirlenmeli ve tasarım aşamasında Mikroişlemci, FPGA yada ASIC teknolojilerinden hangilerinin hangi aşamada kullanılması gerektiği planlanmalıdır. Bir mikroişlemci kullanarak rahatlıkla yapılabilecek bir proje, FPGA üzerinde gerçekleştirildiğinde zaman ve maliyet açısından zarar ettirecektir.

Bununla birlikte entegre devre teknolojileri birbirinden bağımsız teknolojiler değillerdir. Örnek olarak FPGA entegrelerin üretilmesinde ASIC teknolojisi kullanılırken, herhangi bir uygulamaya yönelik ASIC tasarımında ilk önce FPGA kullanılarak istenen fonksiyon tasarlanıp gerçekleştirildikten sonra tasarım ASIC'e taşınmakta, yani FPGA prototip aşamasında kullanılmaktadır.

FPGA üzerinde tasarım geliştirirken bir çeşit kontrolcüye ihtiyaç duyulabilir. Bu gibi durumlarda FPGA içerisine mikroişlemci gömüp sonrasında bu mikroişlemci programlanarak kullanılabilir. Bu işlemciyle birlikte FPGA üzerinde yoğun işlem yapacak bloklar paralel olarak çalışıtırılabilir. Bu sayede mikroişlemcinin getirdiği kolay kontrol, hızlı geliştirme ve test özelliğiyle FPGA'in getirdiği yüksek performans birlikte kullanılmış olur. Bu şekilde FPGA üzerine gömülerek kullanılan işlemcilere "softcore" işlemci denilmektedir. MicroBlaze ve OpenRISC bunlardan en popüler olanlarıdır. Bu tip işlemciler benzer tasarıma sahip olan entegre mikroişlemcilere göre daha düşük performansa sahip olmaktadırlar. Bu sistemlere alternatif olarak FPGA entegrenin üretimi aşamasında bir mikroişlemci çekirdeğinin de "hardcore" olarak aynı çip üzerinde gömülmesidir. İçerisinde ARM mimarili mikroişlemci bulunduran Xilinx firmasının Zynq-7000 All Programmable SoC'leri ve Intel (Altera) Cyclone V SoC'leri bunlara örnek olarak verilebilir. Piyasada birçok FPGA entegre üretici firması olmasına karşın, Xilinx ile Intel'in 2015 yılında satın aldığı Altera firmaları en büyük iki firmadır ve bu firmalar FPGA entegre üretiminin yanı sıra kullanıcıya geliştirme ortamları da sunmaktadırlar.

3.4. FPGA Uygulama Alanları

FPGA; sayısal işaret işleme, radar haberleşme sistemi, uzay, savunma sistemleri, ASIC, medikal resimleme, robotik, ses tanıma, şifreleme ve kod çözme gibi birçok alanda kullanılmaktadır. Ayrıca hızlı Fourier dönüşümü (FFT) ve konvolüsyon gibi yüksek performanslı hesaplamaların yapılmasını gerektiren uygulamalarda FPGA'ların kullanımı giderek artmaktadır. Fakat FPGA'ların yüksek performans hesaplamalarında kabul görmeleri, tasarım zorluğundan dolayı hala belli bir limittedir. Birden fazla FPGA kullanılarak gibi bir sistem oluşturulabilir (Şekil 3.3).

Şekil 3.3. İki adet FPGA kullanılarak oluşturulan sistem (Kılıçkır, 2014).

3.5. Kaos Terimi ve Tarihi

Kaos ilk defa 19.yüzyılın sonlarında astronomi alanıyla ilgili bir problem ile ortaya çıkmıştır. Paris Üniversitesinden astronomi alanında çalışmalar yapan profesör Henri Poincare, Güneş-Dünya ve Ay'ın yörüngelerinin mevcut sistemlerle (liner-nonlineer) açıklanamayacak bir şekilde farklı bir biçimde birbirlerini etkilediğini gözlemledi ve bu gezegenlerin uzay düzlemi içinde özel eğriler (yörüngeler) çizdiğini keşfetti. Poincare'nin bu çığır açan çalışmasından sonra kaos, 1920 lere kadar gündeme gelmedi (Alligood, Sauer ve Yorke 1996).

1912 yılında ise Edwin H.Armstrong yüksek frekanslı salınımlar (oscillations) için yenilemeli (regenerative) bir devre keşfetti. Böylece ilk defa kaosun gözlemlenmesi

mümkün oldu (Chen ve Ueta). 1927 yılında ise Hollandalı fizikçiler Van der Pol ve Van der Mark, ilk nonlineer osilatör devrelerinden birini tasarladı ve belirli frekanslarda düzensiz bir gürültü sinyalinin meydana geldiğini gözlemlediler. Bu, devrelerdeki kaosun keşfedilmiş ilk örneklerinden biri oldu (Van Der Pol ve Van Der Mark, 1927).

1950 lerin sonunda Massachusetts Teknoloji Enstitüsünden (MIT) Edward Lorenz, kaos'u matematiksel bir sistem olarak göstermek istedi. Başlangıçta 20 değişkenli bir diferansiyel denklem sistemi kuran Lorenz, daha sonra modelini bugün Lorenz denklemleri olarak bilinen 3 değişkenli nonlineer diferansiyel denklem sistemine dönüştürdü (Lorenz, 1963). Bu denklem sistemi Eş. 3.1, Eş. 3.2 ve Eş.3.3'de verilmiştir.

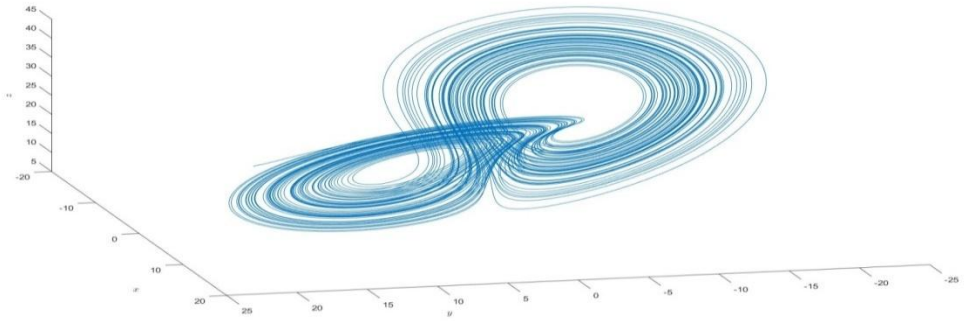
$$\dot{x} = -\alpha x + \alpha y \tag{3.1}$$

$$\dot{y} = -xz + \rho x - y \tag{3.2}$$

$$\dot{z} = xy - \beta z \tag{3.3}$$

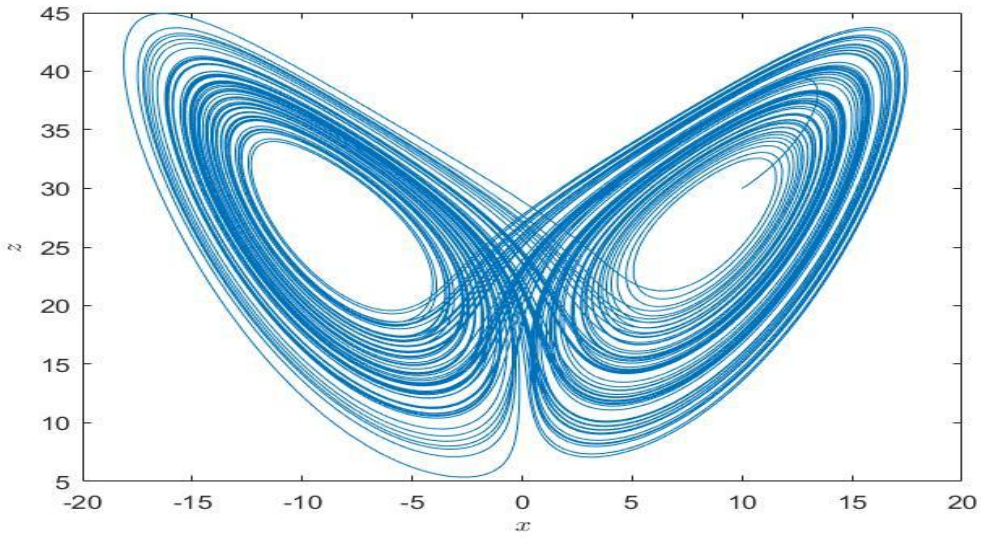
Burada x, y ve z; zamana(t) bağlı durum değişkenleri α, ρ ve β ise Lorenz katsayılarıdır.

Eş. 3.1, Eş. 3.2 ve Eş. 3.3'de verilen diferansiyel denklem sistemi, Matlab/Simulink gibi bir çizim programında çizdirilebilir. Bunun için α, ρ ve β sabitlerini sırasıyla 10, 28 ve 8/3 olarak alalım. Ayrıca fonksiyonların başlangıç koşullarını x(0) = 10, y(0) = 20 ve z(0) = 30 olarak alalım. Denklem sistemini 3 boyutlu düzlemde x(t), y(t) ve z(t) fonksiyonlarına göre çizdirelim (Şekil 3.4).



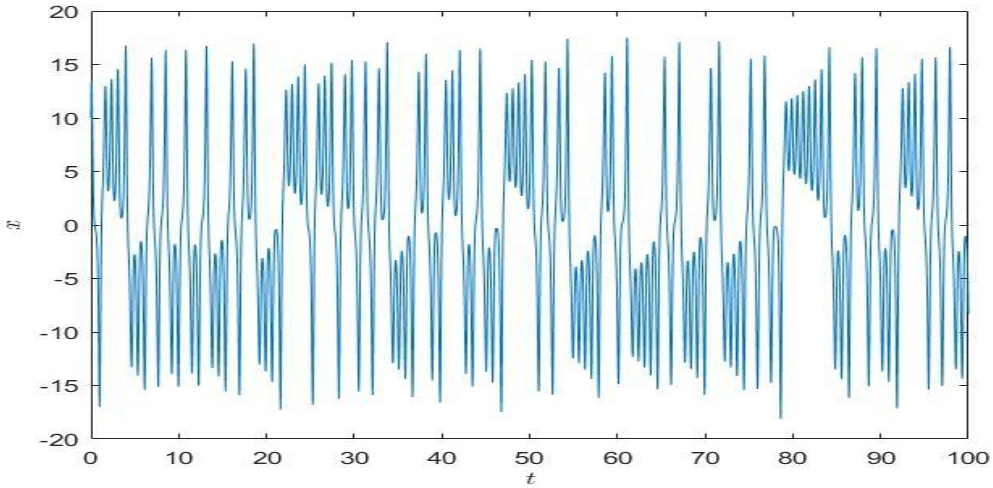
Şekil 3.4. Denklem sisteminin x(t) , y(t) ve z(t) fonksiyonlarına göre grafiği (Muthuswamy ve Banerjee, 2015).

Aynı zamanda herhangi iki giriş fonksiyonunun birbirlerine göre grafiği de çizdirilebilir. Örneğin; x(t) ve z(t) fonksiyonlarının birbirlerine göre değişim grafiği görülebilir (Şekil 3.5). Bu grafik aynı zamanda Lorenz kelebeği olarak da adlandırılır.



Şekil 3.5. x(t) ve z(t) fonksiyonlarının birbirlerine göre değişim grafiği (Muthuswamy ve Banerjee, 2015).

Herhangi bir giriş fonksiyonunun zamana göre değişim grafiği de incelenebilir. Örneğin x(t) fonksiyonunun zamana(t) göre değişim grafiği gösterilmiştir (Şekil 3.6).



Şekil 3.6. x(t) fonksiyonunun zamana göre değişim grafiği (Muthuswamy ve Banerjee, 2015).

Şekil 3.4'de görülen yapı kaos çekicisi (attractor) olarak adlandırılır. Lorenz, bilgisayar simülasyonları kullanarak sistemin hassasiyetinin; başlangıç koşullarına, eğrilerin periyodikliğine ve sınırlandırılmasına, kullanılan kaos denkleminin niteliğine göre değiştiğini gözlemlemiştir.

1970'lerde ve 1980'lerin başında başka kaotik sistemler de bilgisayar simülasyonları yardımıyla tasarlanıp çalıştırıldı. Bu yüzden kaos çekicisinin, bilgisayar simülasyonunun gerçekleştirdiği yapay bir olgu olup olmadığı sorusu cevaplanamadı. Dolayısıyla kaosun gürbüz bir fiziksel olgu olup olmadığının bilinmesine ihtiyaç vardı.

Elektronik devreleri kullanmak Lorenz sistemlerini gerçekleştirmede başvurulabilecek doğal bir yoldu. Çünkü elektronik elemanlar 1970'lerin başında operasyonel yükseltgeçler gibi yaygın olarak kullanılıyorlardı. Ancak Lorenz ve benzer kaotik sistemlerin elektronik devreler yardımıyla gerçekleştrilmesindeki zorluk, çoğaltılmış iki fonksiyon içeren kaotik sistemlerin eşitliklerinin elektroniksel olarak uyarlanmasından ileri gelmekteydi. Çünkü 1970'lerde ve 1980'lerin başında kullanılan analog çoğaltıcılar güvenilebilir elektronik elemanlar değillerdi. Analog çoğaltıcıların bu güvenilirsizliği, 1983'te (Lorenz'in çalışmasından yaklaşık 20 yıl sonra) ilk elektronik kaotik devrenin keşfine sebep oldu (Muthuswamy ve Banerjee, 2015). Kaliforniya Üniversitesinden Leon O. Chua, "Chua Devresi" olarak bilinen ilk elektronik kaotik elektrik devresini tasarladı.(Chua, 1980).

Chua kaotik denklem sistemi Eş. 3.4, Eş. 3.5 ve Eş. 3.6'da gösterilmiştir. Burada α , m_1 , m_0 ve $\beta \in$ sistem parametreleri olarak adlandırılır.

$$\dot{x} = \alpha[y - x - m_1 x - \frac{1}{2}(m_0 - m_1)(|x+1| - |x-1|) \tag{3.4}$$

$$\dot{y} = x - y + z \tag{3.5}$$

$$\dot{z} = -\beta y \tag{3.6}$$

Chua'nın devresi; simülasyon yardımıyla varlığı kabul edilen, elektronik devre yardımıyla da deneysel olarak ve 1984'de Shilnikov teoremi ile kesin olarak varlığı kabul edilen ilk kaotik sistemdir. 1983 sonunda Chua devresinin ortaya çıkması ile 1984'de kaosun kesin kanıtlarının ortaya çıkması arasında geçen süre yaklaşık 1 senedir. Buna karşılık olarak Lorenz'in sistemi ilk olarak 1963'teki çalışmasından sonra yaklaşık 36 yıl sonra 1999'da Warwick Tucker tarafından sistemin kesin olarak kaotik

olduğu kesinleşti. Kaosun fiziksel yorumlamasının ve gerçekleştirilmesinin zorluğu bu konu için FPGA kullanılmasının arkasındaki asıl motivasyon kaynağıdır (Muthuswamy ve Banerjee, 2015).

Chua'nın çalışmasından sonra birçok diğer kaotik devre tasarlandı. Böylesi devreler, Wisconsin Üniversitesinden Julien Clinton Sprott tarafından hazırlanan jerky dinamiklerini de içerisinde barındırır (Sprott, 2010). Eş. 3.7 Sprott'un jerky dinamik sistemleri için genel sistem denklemini gösterir. Eğer $x(t)$ konum olarak düşünülürse $\ddot{x}(t)$ ivmeyi verir. Ancak Eş. 3.7 ivmenin değişimini yani jerk'i içerir. Jerky dinamikleri roket biliminde yoğun olarak kullanılır.

$$\dddot{x} = J(x, \dot{x}, \ddot{x}) \tag{3.7}$$

Sprott'dan sonra farklı birçok kaotik devre (histerizis bazlı kaos generatörleri, senronize edilmiş osilatörlerden elde edilen kaos vb.) geliştirildi.

3.6. Kaos Uygulamaları

Kaosun en çok kullanıldığı uygulama alanlarından biri güvenli haberleşme alanıdır. Bunun için en çok kullanılan argüman kaos senkronizasyonudur. Senkronizasyon temel olarak iki farklı kaotik sinyalin birbirine benzetilip örtüştürülmesi olayına denir. Bir alıcı ve bir verici kaotik sistem birbirlerini senkronize edebilir. Eğer kaotik sinyali büyük bir maskelenmiş kaotik sinyal şeklinde kullanabilirsek, mesajı da yine kaotik maske ile iletebiliriz. Kaotik sistemlerde senkronizasyon konsepti ilk olarak Pecora ve Carroll tarafından hazırlandı (Pecora ve Carroll, 1990). Güvenli haberleşmede kullanılması için de Cuomo ve Oppenheim tarafından öneride bulunuldu (Cuomo ve Oppenheim, 1993).

Senkronizasyonun temel mantığı şöyledir; Eğer bir kaotik sistem (örn. Lorenz sistemi), hareketli (drive) ve sabit cevap (stable response) altsistemleri bölünebiliyorsa, orijinal mesaj, sadece iletilen sinyal kullanılarak alıcı tarafında alınabilir. Lorenz sistemini tekrar göz önüne getirirsek;

$$\dot{x} = -\alpha x + \alpha y \tag{3.8}$$

$$\dot{y} = -xz + \rho x - y \tag{3.9}$$

$$\dot{z} = xy - \beta z \tag{3.10}$$

Pecora ve Carroll Eş. 3.8, Eş. 3.9 ve Eş. 3.10'da Lorenz sisteminin iki sabit cevap sistemine bölünebildiği gösterdi.

$$\dot{x}_1 = -\alpha x_1 + \alpha y \tag{3.11}$$

$$\dot{z}_1 = x_1 y - \beta z_1 \tag{3.12}$$

$$\dot{y}_2 = -xz_2 + \rho x - y_2 \tag{3.13}$$

$$\dot{z}_2 = xy_2 - \beta z_2 \tag{3.14}$$

Eş. 3.8, Eş. 3.9 ve Eş. 3.10, hareketli (drive) sistem olarak adlandırılabilir. Çünkü sistemin dinamikleri cevap (response) alt sistemlerinden bağımsızdır. Buna rağmen iki cevap alt sistem (Eş. 3.11, Eş. 3.12, Eş. 3.13, Eş. 3.14), hareketli sistemde gelişen tam boyutlu dinamikleri yeniden üretmek için kullanılabilirler**.**

Kaos senkronizasyonu gerçekleştirmek için Lorenz ve Chen kaotik sistemlerini seçtik. Master sistem olarak kullanacağımız Lorenz kaotik denklem sistemi ve slave sistem olarak kullanacağımız Chen kaotik denklem sistemi sırasıyla Eş. 3.15 ve Eş. 3.16 eşitliklerinde verilmiştir.

$$\begin{aligned} \dot{x}_1 &= \alpha_1(x_2 - x_1) \\ \dot{x}_2 &= \alpha_2 x_1 - x_1 x_3 - x_2 \\ \dot{x}_3 &= x_1 x_2 - \alpha_3 x_3 \end{aligned} \tag{3.15}$$

$$\begin{aligned} \dot{y}_1 &= \beta_1(y_2 - y_1) + \mathrm{u}_1 \\ \dot{y}_2 &= (\beta_2 - \beta_1) y_1 - y_1 y_3 + \beta_2 y_2 + \mathrm{u}_2 \\ \dot{y}_3 &= y_1 y_2 - \beta_3 y_3 + \mathrm{u}_3 \end{aligned} \tag{3.16}$$

Burada; x_1, x_2, x_3 master sistemin durum değişkenleri, y_1, y_2, y_3 slave sistemin durum değişkenleri, $\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3$ pozitif sabit parametreler ve u_1, u_2, u_3 ise kontrol fonksiyonlarıdır.

$\alpha = [10, 28, 2.67]^T$ ve $\beta = [35, 28, 3]^T$ olduğunda sistem kaotik davranış sergiler.

3.7. Aktif Kayan Kipli Kontrol (Active Sliding Mode Control)

Pecora ve Carroll'un etkileyici çalışmasından sonra fizik, matematik, mühendislik, haberleşme vb. alanlarında kaos senkronizasyonu çalışmalarına karşı artan bir ilgi meydana geldi. Ayrıca kaos senkronizasyonunu gerçekleştirmek için çeşitli verimli teknikler ve yöntemler geliştirildi. Böylece gizli haberleşme tekniğinde kilit rol oynayan kaos senkronizasyonu, çok önemli bir hedef ve devam eden çoğu araştırmanın önemli bir konusu haline geldi.

Son yılarda iki Lorenz sistemini senkronize etmek için aktif kayan kipli kontrol tekniğini kullanılmaya başlandı. Dahası birçok araştırmacı tekniği daha ileri düzeyde inceledi ve bu tekniğin sahip olduğu doğal avantajlardan ötürü diğer sistemlere de uyguladılar. Kaos senkronizasyonuyla ilgili yapılan yayınlarda, kaotik sistemlerin tüm parametrelerin sabit ve belirli, iyi bilinen kaotik modeller olduğu görüldü. Ancak gözlemlenebilir durumlarda veya gerçek fiziksel sistemlerde ise sistem parametrelerinin sürekli dalgalandığı ve kararsız olduğu görüldü. Bunun nedeni ise sistem parametrelerinin, kaçınılmaz bir şekilde çevre sıcaklığı, votaj dalgalanması, fiziksel elemanlar arasında karşılıklı parazitler vb. harici doğal faktörlerden etkilenmesidir.

Kayan kipli kontrol tekniği süreksiz bir kontrol stratejisi olup şu şekilde yapılır; Öncelikle arzu edilen dinamikler için bir anahtarlama (kayma) yüzeyi seçilir. Ardından sistem eğrilerini yüzeye ulaştıracak ve onları sürekli orada tutacak süreksiz bir kontrol kuralı tasarlanır. Böylesi bir kontrol mekanizmasının en çok farkedilen özelliği; aktif kayan kipli bir sistem, harici bozuculara ve parametrik belirsizliklere karşı daha dayanıklıdır. Yakın zamanda kaotik sistemleri kontrol etmek için birçok araştırmacı aktif kayan kipli kontrol yöntemini kullandı.

Zhang ve ark. (2006), bozucularla birleştirilmiş kaotik sistem senkronizasyonunu kayan kipli kontrol kullanarak gerçekleştirmişlerdir. Li ve ark. (2009), bilinmeyen parametreler, bozucular ve nonlineer kontrol girişlerini içeren kaotik sistemler için uyarlamalı kontrol ve kayan kipli kontrol yöntemlerine dayalı bir senkronizasyon yöntemi keşfettiler. Son zamanlarda "aktif kayan kipli kontrol" tekniği olarak adlandırılan bir teknik kullanılarak kaos senkronizasyonu gerçekleştirdiler. Bu teknik ile aktif kontrol ve kayan kipli kontrol yönteminin avantajlarını kullanmayı hedeflemişlerdir.

3.7.1. Aktif kayan kipli denetleyici tasarımı

Eş. 3.17 ve Eş. 3.18'de verilen kaotik sistemi gözönüne alırsak;

$$\dot{x} = A_1 x + f_1(x) \tag{3.17}$$

$$\dot{y} = A_2 y + f_2(\mathrm{y}) \tag{3.18}$$

Burada $x = [x_1, x_2, ...x_n]^T$ ve $y = [\mathrm{y}_1, \mathrm{y}_2, ... \mathrm{y}_n]^T$ durum değişkenlerini, A_1 ve A_2 sabit matrisleri ise sürekli nonlineer fonksiyonları ifade eder. Eğer $\lim_{t\to\infty} \|x - Jy\| = 0$ iken $J = diag\{j_1, j_2, ...j_n\}$ sabit matrisi bulunabiliyorsa Eş. 3.17 ve Eş. 3.18'deki sistemler "iyileştirilmiş izdüşümsel senkronizasyon (modified projective synchronization)" durumuna ulaşmıştır diye tabir ederiz. Ayrıca buradaki J simgesi ölçeklenmiş matris (scalling matrix)'i temsil eder.

Not 1. Eğer $A_1 \neq A_2$, $f_1(.) = f_2(.)$ ise Eş. 3.17 ve Eş. 3.18'deki sistemler özdeş olmayan (non-identical) kaotik sistemlerdir.

Not 2. $j_1 = j_2 =, ... = j_n = 1$, $j_1 = j_2 =, ... = j_n = -1$ ve $j_1 = j_2 =, ... = j_n$ durumlarında komple senkronizasyon (complete synchronization), anti-senkronizasyon ve izdüşümsel senkronizasyon**;** iyileştirilmiş izdüşümsel senkronizasyonun özel durumlarıdır.

Not 3. Genelleştirilmiş izdüşümsel senkronizasyonun tüm dinamik durumları senkronize bir şekilde yükseltilmiş ve düşürülmüştür. Ancak iyileştirilmiş izdüşümsel senkronizasyon bize farklı durumların ölçeklerini bağımsız olarak esnetmemize izin

verir. Çünkü ölçeklenmiş bir J matrisinin içinde birçok ölçekleme faktörü (scalling faktör) bulunur.

Bozuculara sahip iki özdeş olmayan kaotik sistemi Eş. 3.19 ve Eş. 3.20'de modellenmiştir.

$$\dot{x} = A_1 x + f_1(\mathrm{x}) + \mathrm{D}_1(t) \tag{3.19}$$

$$\dot{y} = A_2 y + f_2(\mathrm{y}) + \mathrm{D}_2(t) + u(t) \tag{3.20}$$

Burada $u(t) = [u_1(t), u_2(t), ..., u_n(t)]^T$ kontrol giriş vektörü ve $\mathrm{D}_1(t)$, $\mathrm{D}_2(t)$ ise sistemlerin bozucularıdır. Ayrıca α ve β bilinen parametreler olup $\|D_1(t)\| \le \alpha$, $\|D_2(t)\| \le \beta$ ve $J = diag\{j_1, j_2, ... j_n\}$ şeklindedir.

Eğer hata durumunu, $e = x - Jy$, $J = diag\{j_1, j_2, ... j_n\}$ şeklinde tanımlarsak senkronizasyon hata dinamikleri Eş. 3.21'deki gibi ifade edilir.

$$\begin{aligned} \dot{e} &= \dot{x} - J\dot{y} = A_1 x + f_1(\mathrm{x}) + \mathrm{D}_1(t) - J(A_2 y + f_2(\mathrm{y}) + \mathrm{D}_2(t) + u(t)) \\ &= A_1 e + (A_1 J - J A_2) y + f_1(\mathrm{x}) - J f_2(\mathrm{y}) - J u(t) + \mathrm{D}_1(t) - J\,\mathrm{D}_2(t) \end{aligned} \tag{3.21}$$

Bizim buradaki amacımız Eş. 3.21'deki hata sistemi, asimptotik olarak kararlı halde iken Eş. 3.20'deki slave(response) sistemi için uygun bir denetleyici tasarlamaktır. Bu aynı zamanda Eş. 3.19 ve Eş. 3.20'deki sistemlerin arasında iyileştirilmiş izdüşümsel senkronizasyonunun gerçekleştiğini göstermiş olur. Yani Eş. 3.22 sağlanmış olur.

$$\lim_{t\to\infty} \|e\| = \lim_{t\to\infty} \|x - Jy\| = 0 \tag{3.22}$$

Kaotik sistemlerin iyileştirilmiş izdüşümsel senkronizasyonunu gerçekleştirmek için kayan kipli kontrol uygulanır. Bu teknik iki ana aşamadan oluşur. Birinci aşama ardışık kayan kip denetleyecisinin tasarımını kolaylaştırmak için uygun bir aktif denetleyici seçmektir. İkinci aşama ise iyileştirilmiş izdüşümsel senkronizasyonu gerçekleştirmek için bir kayan kipli denetleyici tasarlamaktır.

Aktif kayan kipli denetleyici tasarım stratejsine uygun olarak, hata (e) formunda gösterilemeyen tüm bilinen öğeleri elimine edebilmek için bir $u(t)$ kontrol giriş vektörü seçilmelidir.

$$u(t) = J^{-1}[(A_1 J - JA_2)y + f_1(\mathrm{x}) - Jf_2(\mathrm{y})] - J^{-1}H(t) = f(x,y) - J^{-1}H(t) \tag{3.23}$$

Burada;

$$f(x,y) = J^{-1}[(A_1 J - JA_2)y + f_1(\mathrm{x}) - Jf_2(\mathrm{y})] \tag{3.24}$$

Burada Eş. 3.21 kullanılarak hata sistemi Eş. 3.25'deki gibi tekrar düzenlenebilir.

$$\dot{e} = A_1 e + H(t) + D_1(t) - JD_2(t) \tag{3.25}$$

Eş. 3.25'deki hata dinamik sisteminin orijinde kararlı olmasını kesin olarak sağlayan birçok $H(t)$ kontrol kuralı seçilebilir. Genellikle Eş. 3.2'daki gibi bir kayan kipli kontrol kuralı seçilir.

$$H(t) = Kw(t) \tag{3.26}$$

Burada $K = [k_1, k_2, ..., k_n]^T$ sabit kazanç vektörü olup $w(t) \in \mathrm{R}$ ise kontrol girişidir.

$$w(t) = \begin{cases} w^+(t) & s \geq 0 \\ w^-(t) & s < 0 \end{cases} \tag{3.27}$$

Burada "s" istenilen dinamikleri sağlayan kayma yüzeyidir. Nihai hata dinamiği ise Eş. 3.28'deki gibidir.

$$\dot{e} = A_1 e + Kw(t) + D_1(t) - JD_2(t) \tag{3.28}$$

Kayan kipli kontrol teorisine göre kayma yüzeyi Eş. 3.29'daki gibi seçilebilir.

$$s(e) = Ce \tag{3.29}$$

Burada "C" sabit bir vektördür. Sistem kayma yüzeyi üzerinde hareket ettiğinde kontrol edilen sistem Eş. 3.30'u ve Eş. 3.31'deki şartları sağlar.

$$s(e) = 0 \tag{3.30}$$

ve

$$\dot{s}(e) = 0 \tag{3.31}$$

Kayan kipli dentleyici tasarlamak için Eş. 3.32'deki eşitlik gözönünde bulundurulur.

$$\dot{s} = -q\,\mathrm{sgn}(s) - rs \tag{3.32}$$

Burada sgn(.) ifadesi, signum fonksiyonunu temsil eder. Burada $q > 0$ ve $r > 0$ kazançları kayma şartını sağlayacak şekilde seçilir. Eş. 3.28, Eş. 3.29 ve Eş. 3.32'ye göre;

$$w(t) = -(CK)^{-1}[CA_1 e + CD_1(t) - CJD_2(t) + q\,\mathrm{sgn}(s) + rs] \tag{3.33}$$

şeklinde yazılabilir. Pratik mühendislik uygulamalarında $D_1(t)$ ve $D_2(t)$ bozucuları bilinmeyenlerdir. Bu yüzden kontrol girişi $w(t)$, Eş. 3.34'deki gibi yazılabilir.

$$w(t) = -(CK)^{-1}[CA_1 e + q\,\mathrm{sgn}(s) + rs] \tag{3.34}$$

Not 4. Eğer sapma(inequality), $\|C\|(\alpha+\beta)<\mathrm{q}$ eşitsizliğini sağlıyorsa Eş. 3.32'deki kontrol kuralı kullanılarak Eş. 3.19 ve Eş. 3.20'deki sistemler arasında iyileştirilmiş izdüşümsel senkronizasyonu gerçekleştirilebilir.

Örneğin Eş. 3.35'deki gibi bir Lyapunov fonksiyonunu ele alalım.

$$V=\frac{1}{2}s^2 \tag{3.35}$$

V fonksiyonunun zamana göre türevi

$$\begin{aligned}\dot{V}=s\dot{s}&=sC\{A_1e-K(CK)^{-1}[CA_1e+q\,\mathrm{sgn}(s)+rs]+D_1(t)-D_2(t)\}\\&=-sq\,\mathrm{sgn}(s)+sCD_1(t)-sCJD_2(t)-rs^2\le-\|s\|q-rs^2+\|sC\|\|D_1(t)\|+\|sC\|\|JD_2(t)\|\\&\le-\|s\|q-rs^2+\|sC\|\alpha+\|sC\|\beta=\|s\|\|C\|(\alpha+\beta)-\mathrm{q}]-\mathrm{rs}^2\end{aligned} \tag{3.36}$$

Sapma ($\|C\|(\alpha+\beta)<\mathrm{q}$ eşitsizliği) sağlandığından $\dot{V}=s\dot{s}<0$ şartı sağlanır. Lyapunov kararlılık teorisine göre hata sistemi asimptotik olarak kararlıdır. Böylece Not 4. kesin olarak kanıtlanmış olur.

Not 5. Eğer $f_1(.)=f_2(.)=f(.)$ ve $A_1=A_2=A.$ ise master sistem ve slave sistem özdeştir. Sapma ($\|C\|(\alpha+\beta)<\mathrm{q}$ eşitsizliği) sağlandığından iyileştirilmiş izdüşümsel senkronizasyon, Eş. 3.37'deki denetleyici altında gerçekleştirilebilir.

$$u=J^{-1}[(AJ-JA)y+f(\mathrm{x})-Jf(\mathrm{y})]+J^{-1}K(CK)^{-1}[CAe+q\,\mathrm{sgn}(s)+rs] \tag{3.37}$$

Not 6. $D_2(t)=0$ olduğunda, master sistem bozucular tarafından etkilenmez. Eğer sapma ($\|C\|(\alpha+\beta)<\mathrm{q}$ eşitsizliği) sağlanırsa bozucuya sahip Eş. 3.19'daki master sistem ve Eş. 3.20'deki slave sistem arasında iyileştirilmiş izdüşümsel senkronizasyon, Eş. 3.38'deki denetleyici tarafından gerçekleştirilir.

$$u=f(x,y)+J^{-1}K(CK)^{-1}[CA_1e+q\,\mathrm{sgn}(s)+rs] \tag{3.38}$$

Not 7. $D_1(t)=0$ olduğunda, slave sistem bozucular tarafından etkilenmez. Eğer sapma ($\|C\|(\alpha+\beta)<\mathrm{q}$ eşitsizliği) sağlanırsa bozucuya sahip Eş. 3.20'deki slave sistem ve Eş. 3.19'daki master sistem arasında iyileştirilmiş izdüşümsel senkronizasyon, Eş. 3.38' deki denetleyici tarafından gerçekleştirilir.

3.7.2 Aktif kayan kipli kontrol yöntemiyle senkronizasyon

3.7.2.1 Matematiksel modelleme

Eğer master sistem Eş. 3.39'daki formda yazılabiliyor ise;

$$\dot{x}=A_1x+f_1(x) \tag{3.39}$$

Burada $f_1(x)$; nx1 ve A_1; n x m boyutlu iki matristir. Ayrıca slave sistem;

$$\dot{y}=A_2y+f_2(\mathrm{y})+u(\mathrm{t}) \tag{3.40}$$

yazılabiliyor olsun. Burada $f_2(y)$; n x 1 ve A_2; n x q boyutlu iki matristir. Sistemin hata matrisi;

$$e=y-x \tag{3.41}$$

ve

$$\dot{e}=A_2y+f_2(y)-A_1x-f_1(x)+u(t)=A_2e+F(x,y)+u(t) \tag{3.42}$$

şeklinde hatanın değişimi bulunabilir. Burada $F(x,y)$, Eş. 3.43'deki gibi yazılır.

$$F(x,y)=f_2(y)-f_1(x)+(A_2-A_1)x \tag{3.43}$$

Ayrıca burada denetleyici tasarlamanın temel amacı $\lim_{t\to\infty}\|e(t)\| = 0$ şartının sağlanmasıdır. Bunun için aktif kontrol tasarım prosedürüne göre hata dinamiklerinin nonlineer kısmını yok etmek için kontrol girişi $u(t)$ kullanılır. Diğer taraftan kontrol girişi Eş. 3.44'deki gibi düşünülebilir.

$$u(t) = H(t) - F(x, y) \tag{3.44}$$

Eş. 3.42'de verilen denklem sistemi yeniden yazılırsa;

$$\dot{e} = A_2 e + H(t) \tag{3.45}$$

Şeklinde yeni tanımlanmış $H(t)$ kontrol girişine sahip hata dinamikleri tanımlanabilir. $H(t)$ kontrol girişi bir çok farklı yolla seçilebilir.

$$\mathrm{H(t)} = \mathrm{Kw(t)} \tag{3.46}$$

Burada $K = [k_1, k_2, k_3]^T$ sabit kazanç vektörü olup $w(t) \in \mathrm{R}$ kontrol girişi, Eş. 3.47'deki şartı sağlar.

$$w(t) = \begin{cases} w^{+}(t) & s(e) \geq 0 \\ w^{-}(t) & s(e) < 0 \end{cases} \tag{3.47}$$

$s = s(e)$ arzu edilen dinamiklerin yerine gelmesini sağlayan kayma yüzeyidir. Sonuçlanan hata dinamikleri ise Eş. 3.48'deki gibidir.

$$\dot{e} = \mathrm{A}_2\, e + Kw(t) \tag{3.48}$$

Böylece kayan kipli kontrol teorisine uygun kayan kipli denetleyici tasarlanabilir. Kayma yüzeyi Eş. 3.49'daki gibi tanımlanabilir.

$$\mathrm{s(e)} = \mathrm{Ce} \tag{3.49}$$

Burada $\mathrm{C} = [\mathrm{c}_1, c_2, c_3]$ sabit bir vektördür. Eşdeğer kontrol yaklaşımı, durum eğrilerinin $\mathrm{s(e)} = 0$ anahtarlama yüzeyinde kalması için gerekli şart olan $\dot{\mathrm{s}}(\mathrm{e}) = 0$ durumunun gerçekleşmesi gözönüne alınarak bulunur. Bu yüzden kayan kip ile kontrol edilen sistem;

$$s(e) = 0 \tag{3.50}$$

ve

$$\dot{\mathrm{s}}(\mathrm{e}) = 0 \tag{3.51}$$

şartlarını sağlar. Şimdi Eş. 3.48, Eş. 3.49 ve Eş. 3.51 denklemleri kullanarak;

$$\dot{\mathrm{s}}(\mathrm{e}) = \frac{\partial s(e)}{\partial e}\dot{e} = C[Ae + Kw(t)] = 0 \tag{3.52}$$

denklemini elde edebiliriz. Eş. 3.52'yi $w(t)$ için çözdüğümüzde;

$$\mathrm{w}_{eq}(t) = -(CK)^{-1}CAe(t) \tag{3.53}$$

İfadesini bulabiliriz. Burada $\mathrm{w}_{eq}(t)$, eşdeğer kontrol girişidir. Ayrıca $-(CK)^{-1}$, gerekli bir şarttır. Eş. 3.48 eşitliğindeki $w(t)$ yerine Eş. 3.53 eşitliğindeki $\mathrm{w}_{eq}(t)$ yazılırsa kayan kip durum eşitliği, Eş. 3.54'deki gibi yazılır.

$$\dot{\mathrm{e}} = [\mathrm{I} - \mathrm{K}(\mathrm{CK})^{-1}C]Ae \tag{3.54}$$

Eş. 3.54'deki sistemin tüm özdeğerleri, negatif gerçel kısımlara sahip olduğu sürece sistem, asimptotik olarak kararlıdır.

Eş. 3.50, Eş. 3.51 ve Eş. 3.52'deki eşitliklerin sağlandığı farzedilirse Eş. 3.55'e ulaşılabilir.

$$\dot{s} = -q\,\mathrm{sgn}(s) - rs \tag{3.55}$$

Burada sgn ifadesi işaret (signum) fonksiyonunu temsil eder. Ayrıca $q > 0$ ve $r > 0$ sabitleri kayma şartı sağlayacak ve kayan kip etkisi yaratacak şekilde seçilir. Eş. 3.48 ve Eş. 3.49 denklemlerinden kontrol girişi, Eş. 3.56'deki gibi seçilir.

$$w(\mathrm{t}) = -(\mathrm{CK})^{-1}[C(rI + A)e(t) + q\,\mathrm{sgn}(s)] \tag{3.56}$$

Kontrol edilen sistemin kararlılığını test edebilmek için Eş. 3.57'deki gibi bir Lyapunov fonksiyonunu gözönüne getirelim.

$$\mathrm{V} = \frac{1}{2}s^2 \tag{3.57}$$

Eş. 3.25'in zamana göre türevi;

$$\dot{V} = \dot{s}s = -qs\,\mathrm{sgn}(s) - rs^2 \tag{3.58}$$

$s\,\mathrm{sgn}(s) > 0$, $r > 0$, $q > 0$ olduğundan $\dot{V} = \dot{s}s < 0$ olmak zorundadır. Bu yüzden $\dot{V}(e)$ negatif tanımlıdır. Bu durum, kayma yüzeyi s'nin sonsuzluğunu gösterir. Eş. 3.56'yı, Eş. 3.48'de yerine yazarsak;

$$\dot{e} = [A_2 - K(CK)^{-1}C(rI + A_2)]\mathrm{e} - \mathrm{K}(\mathrm{CK})^{-1}q\,\mathrm{sgn}(s) \tag{3.59}$$

$s \geq 0$ için; $-K(CK)^{-1}q$ ve $s < 0$ için; $K(CK)^{-1}q$ sınırlandırılmış girişlere sahip bir lineer sistem olan hata sistemi sadece ve sadece $[A_2 - K(CK)^{-1}C(rI + A_2)]$ ifadesi negatif özdeğerlere sahip olduğunda asimptotik olarak kararlıdır. Kaotik sistemlerde

verilen A_2 matrisinin özel yapısından dolayı özdeğerlerden biri $-r$ değerine eşit olur ve bu yüzden kararlıdır. Diğer iki özdeğer r değerinden bağımsızdır ve K ve C gibi diğer kontrol parametreleri tarafından seçilirler. K ve C'ye bağlı olarak daha sonraki iki özdeğer pozitif veya negatif değerler alabilir. r, K ve C'nin uygun seçimiyle, sadece hata sistemini kararlı hale getirmekle kalmaz aynı zamanda hata yakınsama oranını da ayarlayabiliriz. q parametresi, bir kayan kipli denetleyiciden beklenen gürbüzlük özelliğini arttırmak için kullanılabilir.

3.7.2.2 Lorenz ve Chen sistemlerinin aktif kayan kipli senkronizasyon parametrelerini hesaplama

Eş. 3.15 ve Eş. 3.16'da verilen Lorenz ve Chen sistemlerini tekrar yazalım.

$$\dot{x}_1 = \alpha_1(x_2 - x_1)$$
$$\dot{x}_2 = \alpha_2 x_1 - x_1 x_3 - x_2$$
$$\dot{x}_3 = x_1 x_2 - \alpha_3 x_3$$

ve

$$\dot{y}_1 = \beta_1(y_2 - y_1) + \mathrm{u}_1$$
$$\dot{y}_2 = (\beta_2 - \beta_1) y_1 - y_1 y_3 + \beta_2 y_2 + \mathrm{u}_2$$
$$\dot{y}_3 = y_1 y_2 - \beta_3 y_3 + \mathrm{u}_3$$

$\alpha = [10,\ 28,\ 2.67]^{\mathrm{T}}$ ve $\beta = [35,\ 28,\ 3]^{\mathrm{T}}$ olduğunda sistem kaotik davranış sergiler. Ayrıca durum değişkenlerinin başlangıç değerleri sırasıyla $x(0) = [10,\ 10,\ 10]^{\mathrm{T}}$ ve $y(0) = [2,\ 2,\ 2]^{\mathrm{T}}$ olarak seçilsin. Master sistemi, Eş. 3.39'da verilen formda yazarsak;

$$\dot{x} = A_1 x + f_1(x)$$

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix}, \qquad A_1 = \begin{bmatrix} -\alpha_1 & \alpha_1 & 0 \\ \alpha_2 & 0 & 0 \\ 0 & 0 & -\alpha_3 \end{bmatrix}, \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \qquad f_1(x) = \begin{bmatrix} 0 \\ -x_1 x_3 - x_2 \\ x_1 x_2 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} -10 & 10 & 0 \\ 28 & 0 & 0 \\ 0 & 0 & -2.67 \end{bmatrix}$$

Ayrıca slave sistemi, Eş. 3.40'da verilen formda yazarsak;

$$\dot{y} = A_2 y + f_2(\mathrm{y}) + u(\mathrm{t})$$

$$\dot{y} = \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -\beta_1 & \beta_1 & 0 \\ \beta_2 - \beta_1 & \beta_2 & 0 \\ 0 & 0 & -\beta_3 \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}, \quad f_2(\mathrm{y}) = \begin{bmatrix} 0 \\ -y_1 y_3 \\ y_1 y_2 \end{bmatrix}, \quad \mathrm{u(t)} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} -35 & 35 & 0 \\ -7 & 28 & 0 \\ 0 & 0 & -3 \end{bmatrix}$$

Eş. 3.43'de verilen değerleri yerine yazarsak;

$$F(x,y) = f_2(y) - f_1(x) + (A_2 - A_1)x$$

$$= \begin{bmatrix} 0 \\ -y_1 y_3 \\ y_1 y_2 \end{bmatrix} - \begin{bmatrix} 0 \\ -x_1 x_3 - x_2 \\ x_1 x_2 \end{bmatrix} + \begin{bmatrix} -25 & 25 & 0 \\ -35 & 28 & 0 \\ 0 & 0 & -0.33 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$= \begin{bmatrix} 25x_2 - 25x_1 \\ x_1 x_3 - y_1 y_3 - 35x_1 + 29x_2 \\ y_1 y_2 - x_1 x_2 - 0.33x_3 \end{bmatrix}$$

Ayrıca $K = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$, $C = \begin{bmatrix} -1.5 \\ 5 \\ -3 \end{bmatrix}^T$, $r = 1.5$ ve $q = 0.35$ seçelim. Bu durumda kayma yüzeyi denklemi Eş. 3.60'daki gibi olur.

$$s = Ce = -1.5e_1 + 5e_2 - 3e_3 \tag{3.60}$$

Eş. 4.32'de verilen değerleri yerine yazarsak;

$$w(\mathrm{t}) = -(\mathrm{CK})^{-1}[C(rI + A)e(t) + q\,\mathrm{sgn}(s)]$$

$$= -\left(\begin{bmatrix} -1.5 & 5 & -3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} \right)^{-1} \left(\begin{bmatrix} -1.5 & 5 & -3 \end{bmatrix} \left(\begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1.5 \end{bmatrix} + \begin{bmatrix} -35 & 35 & 0 \\ -7 & 28 & 0 \\ 0 & 0 & -3 \end{bmatrix} \right) \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + 0.35\,\mathrm{sgn}(s) \right)$$

$$w(t) = -6.1e_1 - 38e_2 - 1.8e_3 - 0.14\,\mathrm{sgn}(s) \tag{3.61}$$

Eş. 3.46'dan;

$$\mathrm{H} = \mathrm{Kw(t)} = \begin{bmatrix} w(t) \\ 2w(t) \\ 2w(t) \end{bmatrix}$$

Son olarak Eş. 3.44'den

$$u(t) = H(t) - F(x, y)$$

$$= \begin{bmatrix} w(t) \\ 2w(t) \\ 2w(t) \end{bmatrix} - \begin{bmatrix} 25x_2 - 25x_1 \\ x_1x_3 - y_1y_3 - 35x_1 + 29x_2 \\ y_1y_2 - x_1x_2 - 0.33x_3 \end{bmatrix}$$

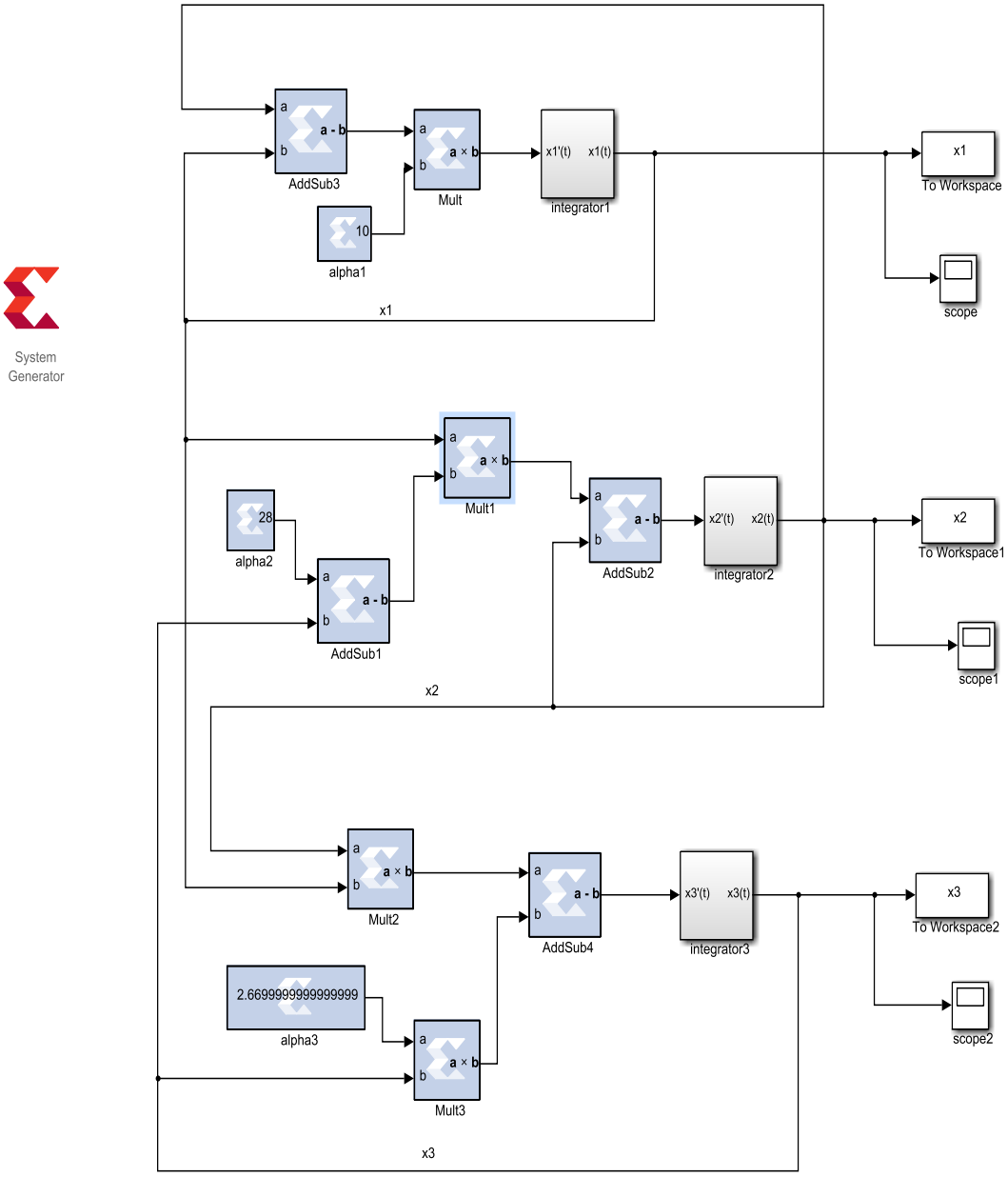
$$u(t) = \begin{bmatrix} w(t) - 25\,\mathrm{x}_2 + 25x_1 \\ 2w(t) - \mathrm{x}_1\,x_3 + y_1y_3 + 35x_1 - 29x_2 \\ 2w(t) - \mathrm{y}_1\,y_2 + x_1x_2 + 0.33x_3 \end{bmatrix} \tag{3.62}$$

3.7.2.3 Sistemin Matlab/Simulink modelinin oluşturulması

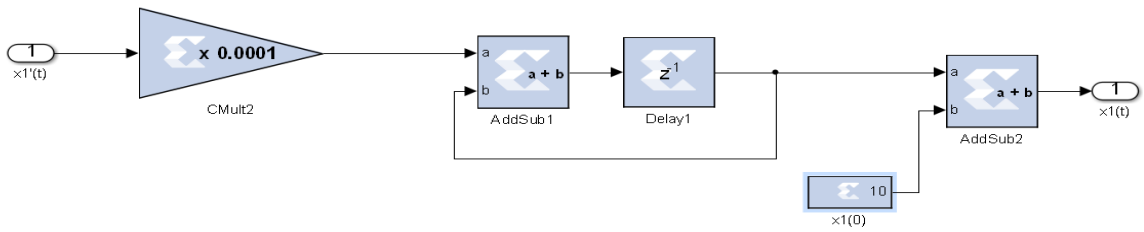
Hardware co-simulation için önceki bölümlerde bulunan matematiksel ifadeleri, Matlab/Simulink bloklarına dönüştürelim. Bunun için her bir alt sistem ayrı ayrı kurulup birleştirilecektir.

3.7.2.3.1 Master sistemin Matlab/Simulink modellemesi

Eş. 3.15'deki master sistemi, xilinx bloksetleri kullanılarak tasarlanmıştır. Burada en büyük problem xilinx blokset içerisinde hazır integrator bloğunun bulunmamasıdır. Bu yüzden integrator bloğu da master sistem içerisinde altsistem olarak tasarlanmıştır. Master sistem ve integratör alt sistemi sırasıyla Şekil 3.7 ve Şekil 3.8'de gösterilmiştir.



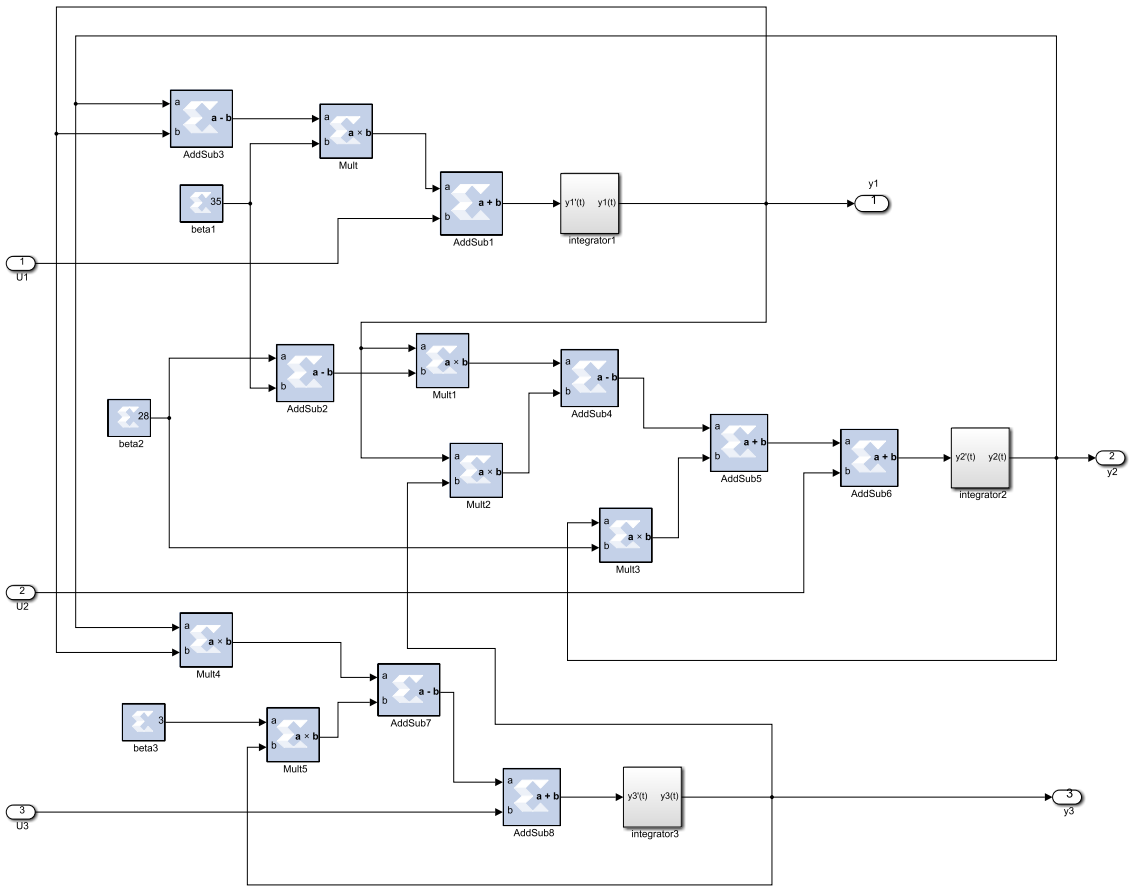
Şekil 3.7. Master sistem.



Şekil 3.8. İntegrator alt sistem.

3.7.2.3.2 Slave sistemin Matlab/Simulink modellemesi

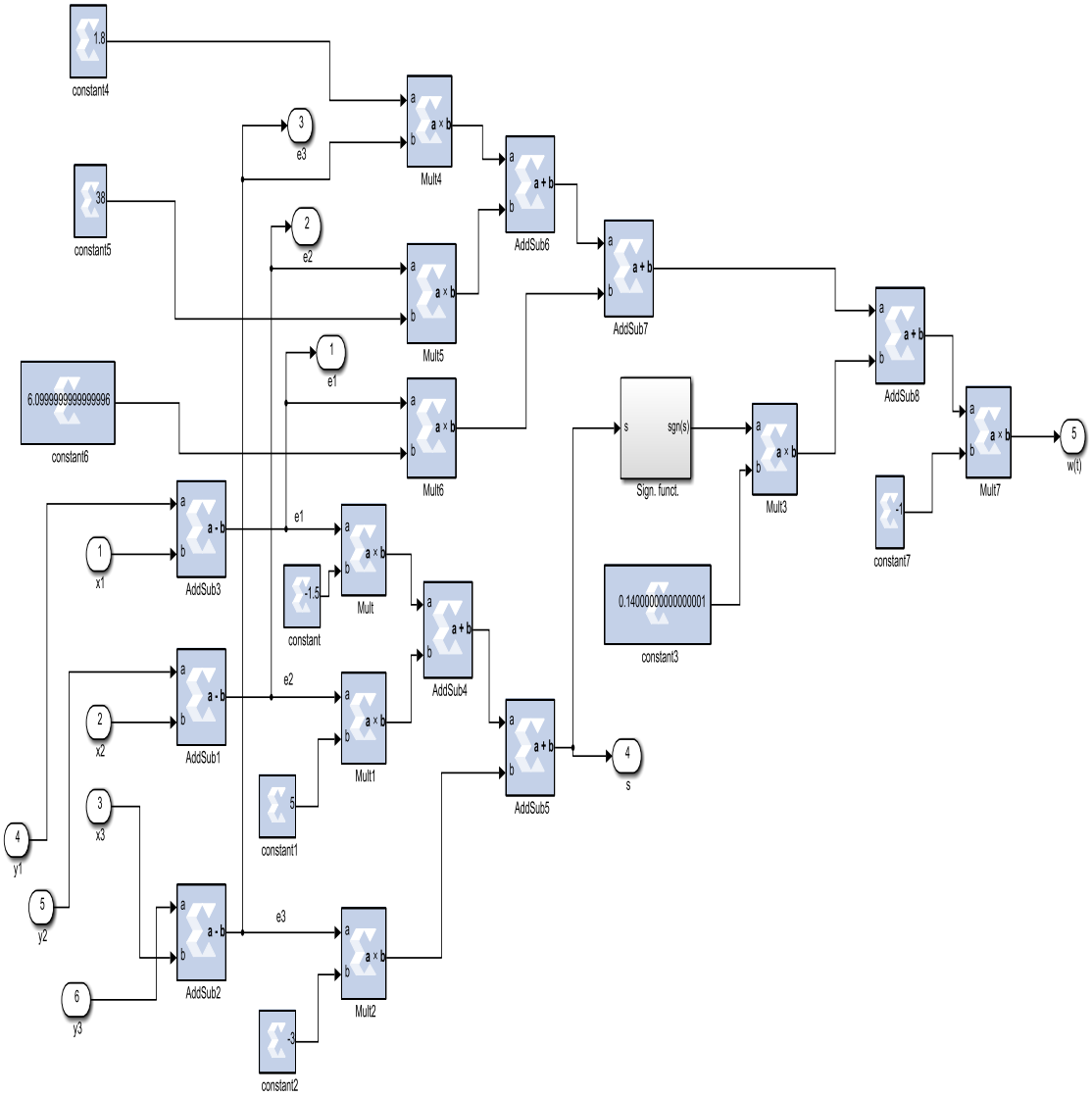
Eş. 3.16'daki slave sistemi xilinx bloksetleri kullanılarak tasarlanmış ve yine master sistem içerisinde kullanılan integrator alt sistemi kullanılmıştır. Slave sistemin Matlab/Simulink modeli gösterilmiştir (Şekil 3.9).



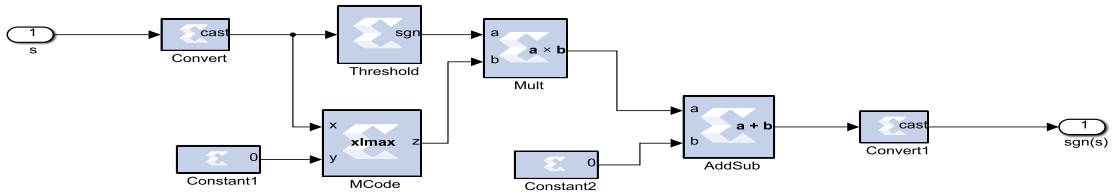
Şekil 3.9. Slave sistem.

3.7.2.3.3. Parametreler sisteminin Matlab/Simulink modellemesi

Eş. 3.41'de senkronizasyon hataları (e_1, e_2, e_3), Eş. 3.60'da kayma yüzeyi (s) ve Eş. 3.61'de kontrol girişi $w(t)$ matematiksel olarak elde edilmişti. Bu denklemler, xilinx bloksetleri kullanılarak blok tasarımı yapılmıştır (Şekil 3.10). Burada en büyük problem xilinx blokset içerisinde hazır "signum" fonksiyonunun bulunmamasıdır. Bu yüzden "signum" bloğu parametreler sistemi içerisinde altsistem olarak tasarlanmıştır (Şekil 3.11).

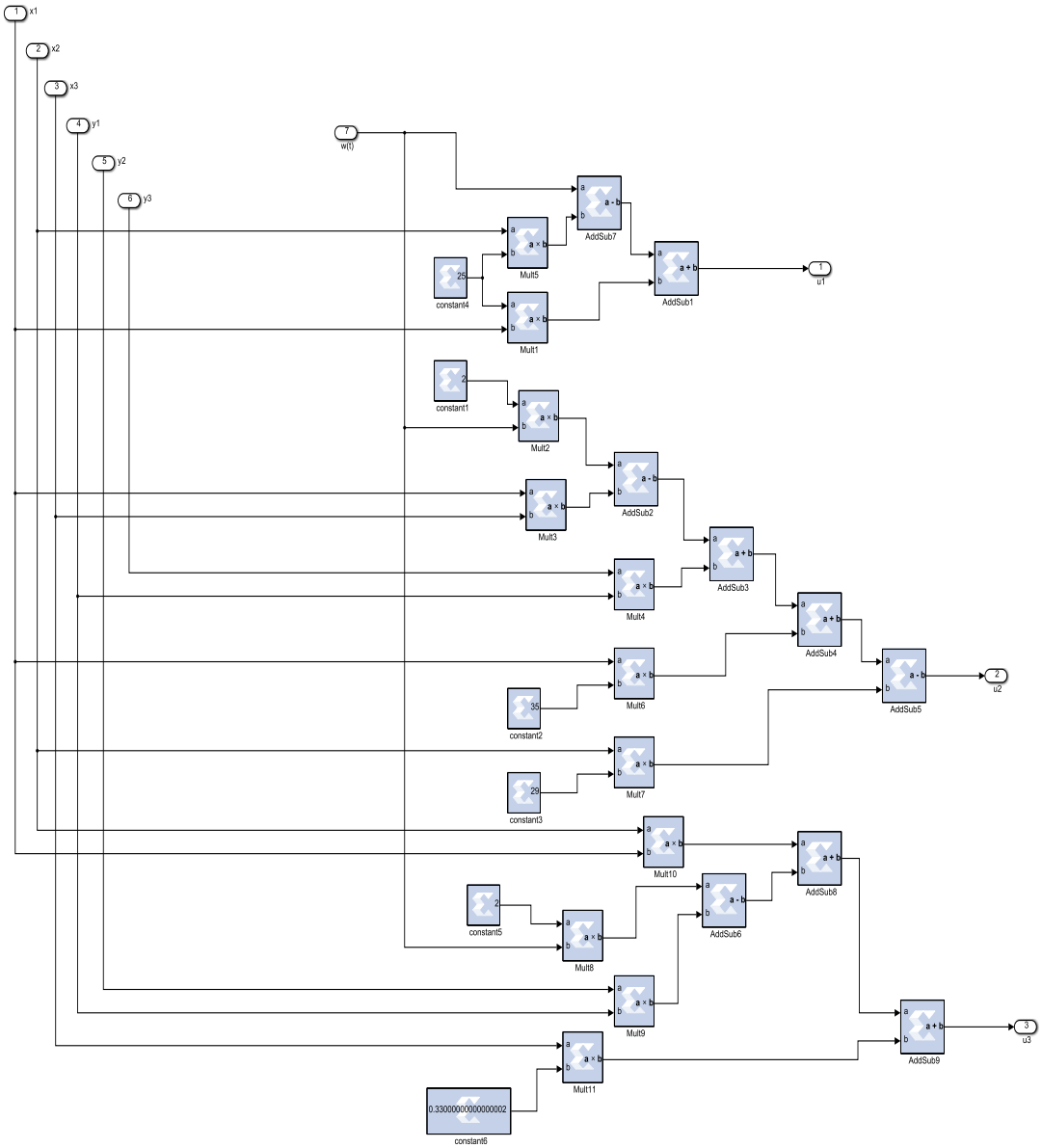


Şekil 3.10. Parametreler sistemi.



Şekil 3.11. Signum alt sistemi.

3.7.2.3.4. Denetleyici sistemin Matlab/Simulink modellemesi

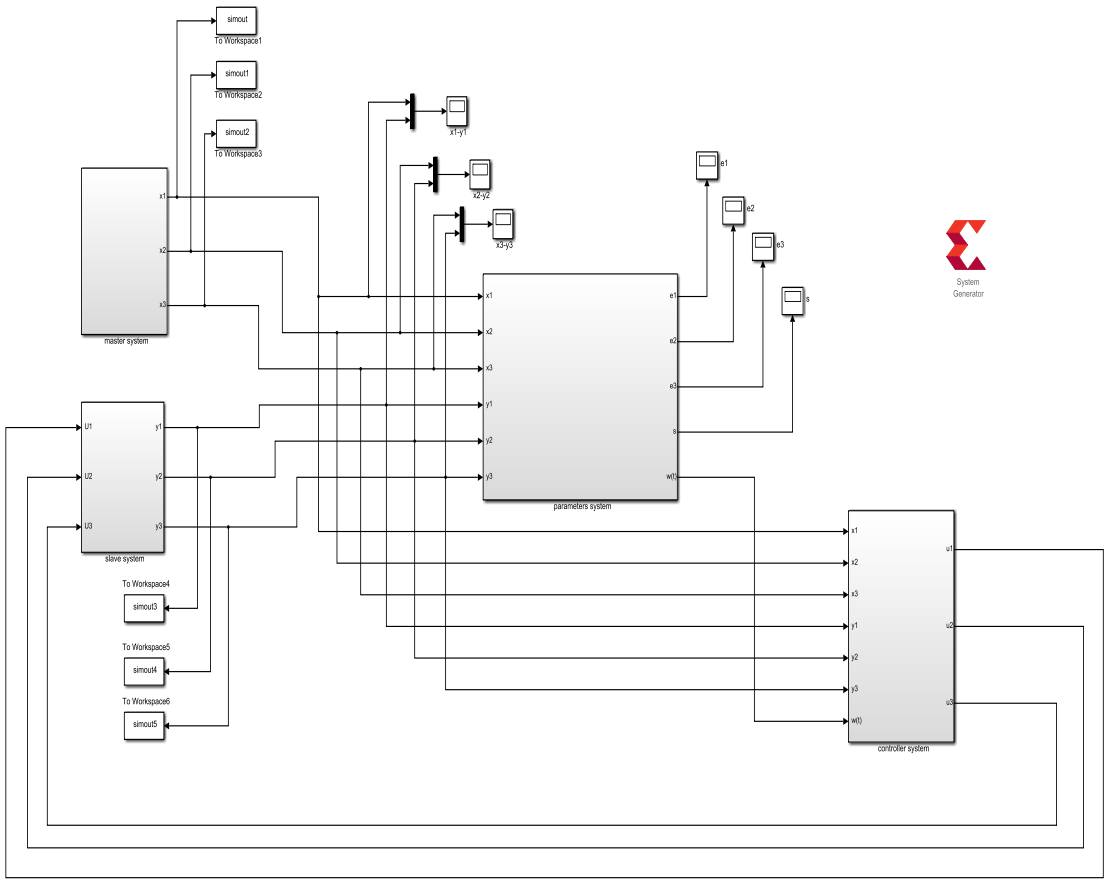


Şekil 3.12. Kontrol sistemi.

Eş. 3.62'de denetleyicilerin (u_1, u_2, u_3), matematiksel ifadeleri elde edilmiş olup u_1, u_2 ve u_3 kontrol fonksiyonlarının Matlab/Simulink modeli oluşturulup çıkışlar slave sisteme giriş olarak gönderilmiştir. Kontrol sisteminin Matlab/Simulink modeli gösterilmiştir (Şekil 3.12).

3.7.2.3.5. Oluşturulan sistemlerin birleştirilmesi

Ayrı ayrı matematiksel ifadeleri bulunan sistemlerin ayrı ayrı modellenmeleri yapılmış ve önceki bölümlerde açıklanmıştır. Ayrı ayrı tasarlanan bu sistemler temel olarak 4 farklı alt sistem haline getirilip ardından "system generator" bloğu da eklenerek sistem nihai halini almıştır. Böylece senkronize edilmiş kaotik sistemin Matlab/Simulink tasarımı tamamlanmış ve modelin blok diyagramı daha sade ve kolay anlaşılır hale getirilmiştir. Senkronize edilmiş olan kaotik sistemin nihai Matlab/Simulink modeli gösterilmiştir (Şekil 3.13).

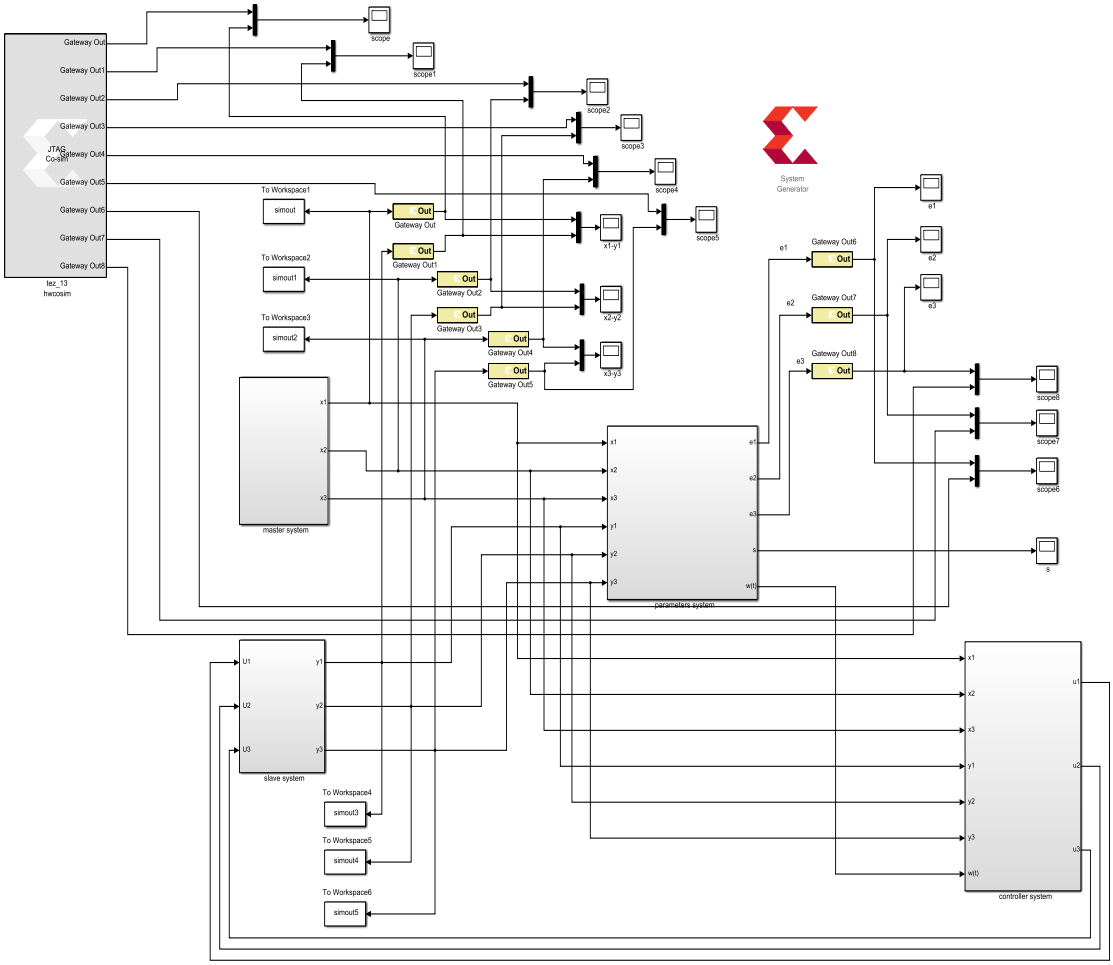


Şekil 3.13. Senkronize edilmiş kaotik sistem modeli.

3.7.2.3.6. Hardware co-simulation yöntemiyle modellenen sistemin gerçek zamanlı uygulaması

Senkronize edilmiş kaotik sistem Matlab/Simulink ortamında tasarlandıktan sonra hardware co-simulation yöntemiyle FPGA'ya gönderilecek ve gerçek zamanlı çıkış bilgileri yine Matlab/Simulink ortamında gözlenecektir.

Sistemin çıkışlarını elde etmek ve bilgisayar ile FPGA'nın bilgi alışverişini sağlamak gerekir. Bunun için "system generator" bloğuna çift tıklanır ve açılan pencerede, kullanılan FPGA modeli ve sistem periyodu uygun şekilde seçilerek generate sekmesine tuşlanır ve JTAG bloğu üretilmiş olur. Şekil 3.14'de tüm sistemin final tasarımı görülmektedir. Burada kapsamlı gözlem yapabilmek için; master ve slave sistemin durum değişkenleri $(x_1, x_2, x_3, y_1, y_2, y_3)$ ve sistem senkronizasyon hataları (e_1, e_2, e_3) çıkış olarak alınmıştır.



Şekil 3.14. Senkronize edilmiş sistemin hardware co-simulation uygulaması.

3.8. Uyarlamalı Kontrol (Adaptive Control)

Pecora ve Carrol'un çalışmasından sonra kaos senkronizasyonu sayısız araştırmanın konusu olmuş ve güvenli haberleşme, biyolojik sistemler, ekolojik sistemler, fiziksel sistemler gibi alanlarda birçok defa uygulanmıştır. Kaotik sistemlerin senkronizasyonu için PC yöntemi, OGY yöntemi, aktif kontrol yöntemi, uyarlamalı kontrol yöntemi, zaman geciktirmeli geribesleme yaklaşımı, ters adımlama yöntemi, kayan kipli kontrol gibi çeşitli çözüm stratejileri geliştirilmiştir. Birçok pratik durumda parametre değerleri iyi bilinememesine rağmen kaotik sistemleri içeren araştırmaların çoğunda bu parametrelerin değerlerine ihtiyaç duyulur. Bu belirsizlik, senkronizasyonu büyük oranda etkiler. Bu yüzden iyi bilinmeyen parametrelere sahip kaotik sistemlerin senkronizasyonu ve kontrolü için bir uyarlamalı denetleyici tasarlamak esastır.

Nonlineer sistemlerin izleme sonuçlarına ve global kararlılığına ışık tutan etkileyici ve hızlı gelişmeler, uyarlamalı kontrol yönteminin bilinmeyen parametrelerinin tahmin edilmesi için verimli bir yöntem olduğunu ortaya çıkarmıştır. Bu yöntem, bilinmeyen parametrelere sahip kaotik sistemleri senkronize etmede başarıyla uygulanmış ve birçok önemli sonuçlar elde edilmiştir. Örneğin; Li ve ark. (2015), bilinmeyen parametrelere sahip kesir dereceli kaotik kaotik sistemler için uyarlamalı impulsif senkronizasyon tekniği üzerinde çalıştı.

Salarieha ve ark. (2008), bilinmeyen değişkenlerin zaman içerisinde çeşitlenmesini göz önünde bulundurarak iki farklı kaotik sistemin uyarlamalı senkronizasyonunu adreslemeyi başardı.

Shi ve ark. (2009), kaosun eklenmiş dereceli anti senkronizasyonunu, Mossa Al-sawalha ve ark. (2010) ise kaosun azaltılmış dereceli anti senkronizasyonunu keşfetti.

He ve ark. (2016), çoklu bilinmeyen parametrelere sahip hiperkaotik sistemlerin senkronizasyonu hakkında ayrıntılı bir çalışma yaptı.

3.8.1 Uyarlamalı denetleyici tasarımı

n_1 - boyutlu kaotik(hiperkaotik) sistem Eş. 3.63'deki gibi bir formda verilir.

$$\dot{x} = F(x, p) \tag{3.63}$$

Burada $x = (\mathrm{x}_1, \mathrm{x}_2 ..., \mathrm{x}_{n1})^T \in R^{n1}$ ve $p \in R^{m1}$ bilinmeyen parametrelerdir. $F(x, p) = (F_1(x, p), F_2(x, p), ..., F_{n1}(x, p))^T$ yapısına sahip kaotik (hiperkaotik) dinamik sistemin bilindiğini düşünürsek tüm değişkenler için zaman serileri Eş. 3.63'ün çıkışındaki gibi uygun formda olur. Eş. 3.63'deki sistemi master sistem, Eş. 3.64'deki sistemi de slave sistem olarak kabul edelim.

$$\dot{y} = G(y, q) + u(x, y, \hat{p}, \hat{q}) \tag{3.64}$$

Burada $y = (y_1, y_2, ..., y_{n2})^T \in R^{n2} (n_2 \le n_1)$, bilinmeyen parametreler $q \in R^{m2}$ ve dinamik değişim eşitlikleri $G(y, q) = (G_1(y, q), G_2(y, q), ..., G_{n2}(y, q))^T$ aynı şekilde bilinen ifadelerdir.

Sistemin denetleyici denklemleri Eş. 3.65'de verilmiştir.

$$u(x, y, \hat{p}, \hat{q}) = (u_1(x, y, \hat{p}, \hat{q}), u_2(x, y, \hat{p}, \hat{q}), ..., u_{n2}(x, y, \hat{p}, \hat{q}))^T \tag{3.65}$$

İki özdeş veya tamamen bilinmeyen parametrelere (p ve q) sahip iki farklı kaotik (hiperkaotik) sistemi senkronize etmek amacıyla Eş. 3.65'deki denetleyici sistemi tercih edilir. Burada $\hat{p}$ ve $\hat{q}$; p ve q 'nun tahmini değerleri olup bir uyarlamalı kontrol döngüsü tarafından uygun bir şekilde üretilirler. İki kaotik (hiperkaotik) sistemin senkronizasyon hataları ise Eş. 3.66'daki gibidir.

$$e = (e_1, e_2, ..., e_{n2})^T = (y_1 - x_1, y_2 - x_2, ..., y_{n2} - x_{n2})^T \in R^{n2} \tag{3.66}$$

Eş. 3.63 ve Eş. 3.64'deki master ve slave sistem arasındaki hata dinamik sistemi Eş. 3.67'deki gibi yazılabilir.

$$\dot{e} = G(y, q) - F^*(x, p) + u(x, y, \hat{p}, \hat{q}) \tag{3.67}$$

Burada $F^*(x, p) = (F_1(x, p), F_2(x, p), ..., F_{n2}(x, p))^T \in R^{n2}$

p ve q parametrelerinin tahmini hataları ise sırasıyla $\tilde{p}$ ve $\tilde{q}$ ile sembolize edilip Eş. 3.68 ve Eş. 3.69'de gösterilmiştir.

$$\tilde{p} = p - \hat{p} \tag{3.68}$$

$$\tilde{q} = q - \hat{q} \tag{3.69}$$

Senkronizasyon hatalarını ve parametrelerin tahmini hatalarını içeren dinamik bir Lyapunov fonksiyonu ise Eş. 3.70'deki gibi yapılandırılmıştır.

$$V(e, \tilde{p}, \tilde{q}) = \frac{1}{2} e^T P e + \frac{1}{2} \tilde{p}^T Q \tilde{p} + \frac{1}{2} \tilde{q}^T R \tilde{q} \tag{3.70}$$

Burada $P \in R^{n2xn2}$, $Q \in R^{m1xm1}$, $R \in R^{m2xm2}$ pozitif tanımlı sabit matrislerdir. Çoğu durumda P, Q ve R'den biri, birim matrislere karşılık gelecek şekilde seçilir.

Eş. 3.67 ve Eş. 3.70 eşitliklerini kullanarak V'nin zamana göre türevi alındığında Eş. 3.71 elde edilir.

$$\dot{e} = e^T P(G(y, q) - F^*(x, p) + u(x, y, \hat{p}, \hat{q})) - (\tilde{p}^T Q \dot{\hat{p}} + \tilde{q}^T R \dot{\hat{q}}) \tag{3.70}$$

Burada denetleyici $u(x, y, \hat{p}, \hat{q})$ ve güncel parametreler ($\dot{\hat{p}}$, $\dot{\hat{q}}$) uygun olarak seçildiğinde $\frac{dV}{dt}$'nin negatif tanımlı olduğu görülür. Böylece Lyapunov kararlılık teoremine göre hiperkaotik sistemin senkronizasyonu kararlı hale gelmiş olur.

3.8.2. Uyarlamalı kontrol yöntemiyle senkronizasyon

3.8.2.1. Matematiksel modelleme

Eş. 3.15 ve Eş. 3.16'da durum değişkenlerinin başlangıç değerleri sırasıyla

$x(0)=[10, 10, 10]^{T}$ ve $y(0)=[2, 2, 2]^{T}$ olarak seçilsin.

Eğer master sistem Eş. 3.72'deki gibi bir formda yazılabiliyor ise;

$$\dot{x} = f(x) + \mathrm{F}(x)\,\alpha \tag{3.72}$$

Burada $f(x)$; nx1 ve $\mathrm{F}(x)$; n x m boyutlu iki matristir. Ayrıca slave sistem;

$$\dot{y} = \mathrm{g(y)} + \mathrm{G(y)}\beta + \mathrm{u} \tag{3.73}$$

yazılabiliyor olsun. Burada $\mathrm{g(y)}$; n x 1 ve $\mathrm{G(y)}$; n x q boyutlu iki matristir. Sistemin hata matrisi;

$$e = y - x \tag{3.74}$$

şeklinde yazılabilir. Buna göre optimize edilmiş uyarlamalı parametreler matrisi;

$$\dot{\tilde{\alpha}} = -\left[F(x)\right]^{T} e \tag{3.75}$$

$$\dot{\tilde{\beta}} = \left[\mathrm{G(y)}\right]^{T} e \tag{3.76}$$

şeklindedir. Sistemin kontrol fonksiyonu ise;

$$\mathrm{u} = f(x) + \mathrm{F}(x)\,\tilde{\alpha} - \mathrm{g(y)} - \mathrm{G(y)}\tilde{\beta} - ke \tag{3.77}$$

gibi yazılır. Burada $k \in Z^{+}$, $\tilde{\alpha}$ ve $\tilde{\beta}$ ise sırasıyla bilinmeyen α ve β parametrelerinin tahmini değeri olarak da adlandırılır. Eş. 3.72 – Eş. 3.77'den hata dinamik sistemi;

$$\dot{e} = \mathrm{F}(x)\,(\tilde{\alpha} - \alpha) - \mathrm{G(y)}\,(\tilde{\beta} - \beta) - ke \tag{3.78}$$

α ve β parametrelerinin hata denklemleri;

$$\hat{\alpha} = \tilde{\alpha} - \alpha \tag{3.79}$$

$$\hat{\beta} = \tilde{\beta} - \beta \tag{3.80}$$

Lyapunov fonksiyonunu;

$$V(\mathrm{e}, \hat{\alpha}, \beta) = \frac{1}{2}[e^T e + (\tilde{\alpha} - \alpha)^T (\tilde{\alpha} - \alpha) + (\tilde{\beta} - \beta)^T (\tilde{\beta} - \beta)]$$

şeklinde seçip V fonksiyonunun zamana göre türevini alalım;

$$\begin{aligned}\dot{V}(\mathrm{e}, \hat{\alpha}, \beta) &= \dot{e}^T e + (\tilde{\alpha} - \alpha)^T \dot{\tilde{\alpha}} + (\tilde{\beta} - \beta)^T \dot{\tilde{\beta}} \\ &= [F(x)(\tilde{\alpha} - \alpha) - \mathrm{G(y)}(\tilde{\beta} - \beta) - ke]^T e - (\tilde{\alpha} - \alpha)^T [F(x)]^T e + (\tilde{\beta} - \beta)^T [G(y)]^T e \\ &= -ke^T e \leq 0\end{aligned}$$

Görüldüğü üzere V fonksiyonu pozitif tanımlıyken $\dot{V}$ ise negatif yarı-tanımlıdır. Bu yüzden sistem kararlıdır ve slave sistem, master sistem ile asimptotik olarak senkronize edilebilir.

3.8.2.2. Lorenz ve Chen sistemlerinin uyarlamalı senkronizasyon parametrelerini hesaplama

Eş. 3.15 ve Eş. 3.16'da verilen Lorenz ve Chen sistemlerini Eş. 3.72 ve Eş. 3.73'de verilen formda yazalım. Master sistem için;

$$\dot{x}_1 = \alpha_1 (x_2 - x_1)$$

$$\dot{x}_2 = \alpha_2 x_1 - x_1 x_3 - x_2$$

$$\dot{x}_3 = x_1 x_2 - \alpha_3 x_3$$

ve

$$\dot{x} = f(x) + \mathrm{F}(x)\, \alpha$$

Dolayısıyla;

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix}, \quad F(x) = \begin{bmatrix} x_2 - x_1 & 0 & 0 \\ 0 & x_1 & 0 \\ 0 & 0 & -x_3 \end{bmatrix}, \quad f(x) = \begin{bmatrix} 0 \\ -x_1 x_3 - x_2 \\ x_1 x_2 \end{bmatrix}, \quad \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

bulunur.

Slave sistem için;

$$\dot{y}_1 = \beta_1 (y_2 - y_1) + \mathrm{u}_1$$
$$\dot{y}_2 = (\beta_2 - \beta_1) y_1 - y_1 y_3 + \beta_2 y_2 + \mathrm{u}_2$$
$$\dot{y}_3 = y_1 y_2 - \beta_3 y_3 + \mathrm{u}_3$$

ve

$$\dot{y} = \mathrm{g(y)} + \mathrm{G(y)}\beta + \mathrm{u}$$

Dolayısıyla;

$$\dot{y} = \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \end{bmatrix}, \ \mathrm{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad \mathrm{G(y)} = \begin{bmatrix} y_2 - y_1 & 0 & 0 \\ -y_1 & y_1 + y_2 & 0 \\ 0 & 0 & -y_3 \end{bmatrix}, \ \mathrm{g(y)} = \begin{bmatrix} 0 \\ -y_1 y_3 \\ y_1 y_2 \end{bmatrix}, \ \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}$$

bulunur. Kontrol matrisini (u) bulmak için Eş. 3.77'yi kullanalım;

$$\mathrm{u} = f(x) + \mathrm{F}(x)\ \tilde{\alpha} - \mathrm{g(y)} - \mathrm{G(y)}\tilde{\beta} - ke$$

$$\mathrm{u} = \begin{bmatrix} \tilde{\alpha}_1 (x_2 - x_1) - \tilde{\beta}_1 (y_2 - y_1) - ke_1 \\ \tilde{\alpha}_2 x_1 - x_1 x_3 - x_2 + y_1 y_3 - (\tilde{\beta}_2 - \tilde{\beta}_1) y_1 - \tilde{\beta}_2 y_2 - ke_2 \\ x_1 x_2 - \tilde{\alpha}_3 x_3 - y_1 y_2 + \tilde{\beta}_3 y_3 - ke_3 \end{bmatrix} \qquad (3.81)$$

Optimize edilmiş uyarlamalı parametreler matrislerini bulmak için Eş. 3.75 ve Eş. 3.76 nolu denklemleri kullanalım.

$$\dot{\hat{\alpha}} = -[F(x)]^T e = \begin{bmatrix} (x_1 - x_2)\mathrm{e}_1 \\ -x_1 e_2 \\ x_3 e_3 \end{bmatrix}, \qquad \dot{\hat{\beta}} = [G(y)]^T e = \begin{bmatrix} (\mathrm{y}_2 - y_1)\mathrm{e}_1 - y_1 e_2 \\ (y_1 + y_2)\mathrm{e}_2 \\ -y_3 e_3 \end{bmatrix} \tag{3.82}$$

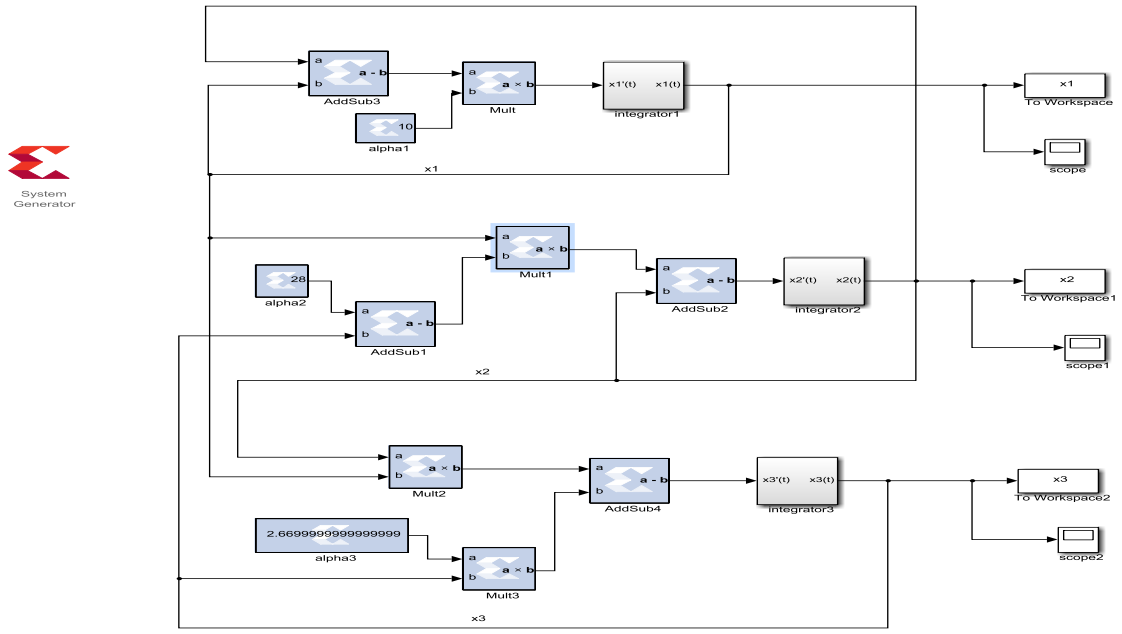
Bu parametrelerin başlangıç değerleri sırasıyla;

$$\tilde{\alpha}(0) = [3,\ 3,\ 3]^{\mathrm{T}} \qquad \text{ve} \qquad \tilde{\beta}(0) = [3,\ 3,\ 3]^{\mathrm{T}} \tag{3.83}$$

3.8.2.3. Sistemin Matlab/Simulink modelinin oluşturulması

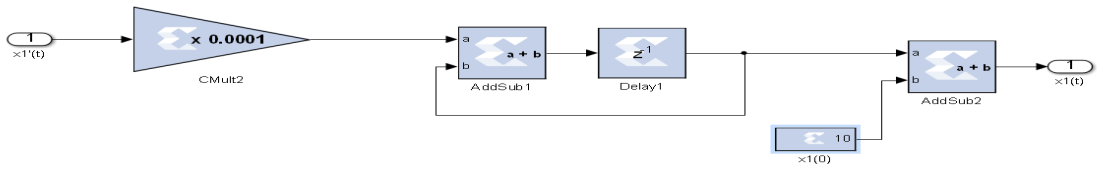
Hardware co-simulation için önceki bölümlerde bulunan matematiksel ifadeleri, Matlab/Simulink bloklarına dönüştürelim. Bunun için her bir alt sistem ayrı ayrı kurulup birleştirilecektir.

3.8.2.3.1. Master sistemin Matlab/Simulink modellemesi



Şekil 3.15. Master sistem.

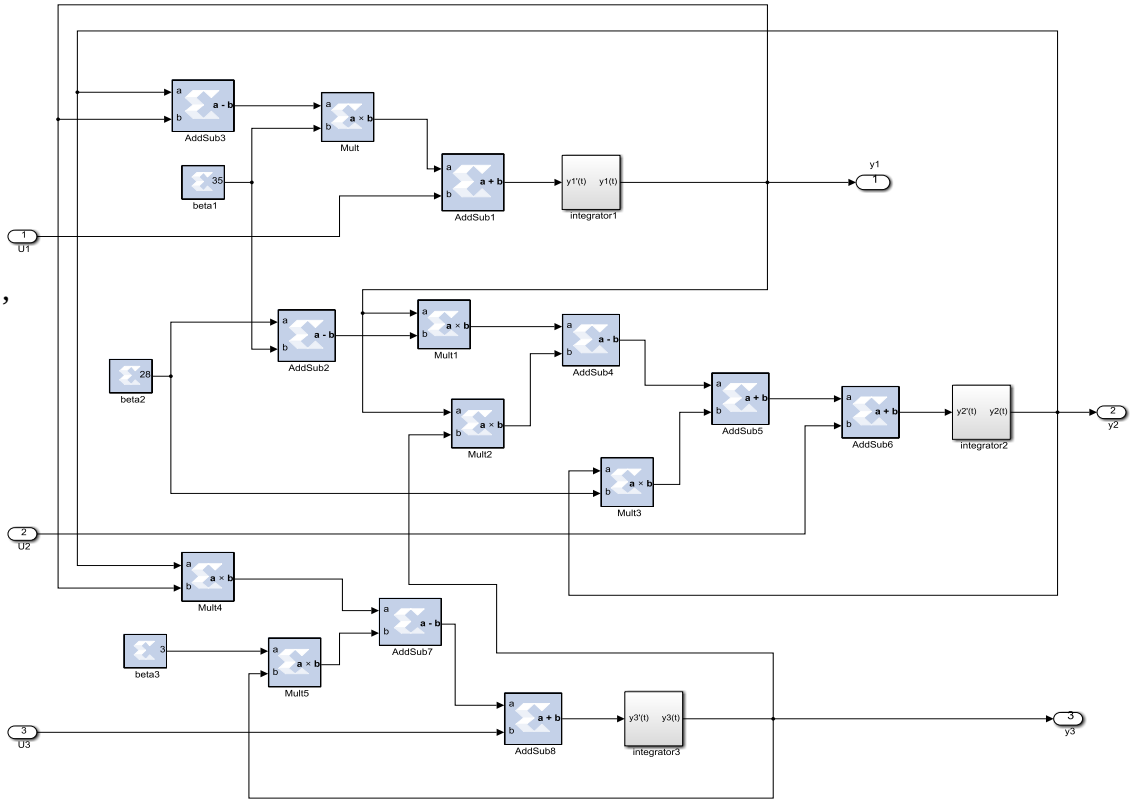
Eş. 3.15'deki master sistem xilinx bloksetleri kullanılarak tasarlanmıştır. Burada en büyük problem xilinx blokset içerisinde hazır integrator bloğunun bulunmamasıdır. Bu yüzden integrator bloğu da master sistem içerisinde altsistem olarak tasarlanmıştır. Master sistem ve integratör alt sistemi sırasıyla Şekil 3.15 ve Şekil 3.16'da gösterilmiştir.



Şekil 3.16. İntegrator alt sistem.

3.8.2.3.2. Slave sistemin Matlab/Simulink modellemesi

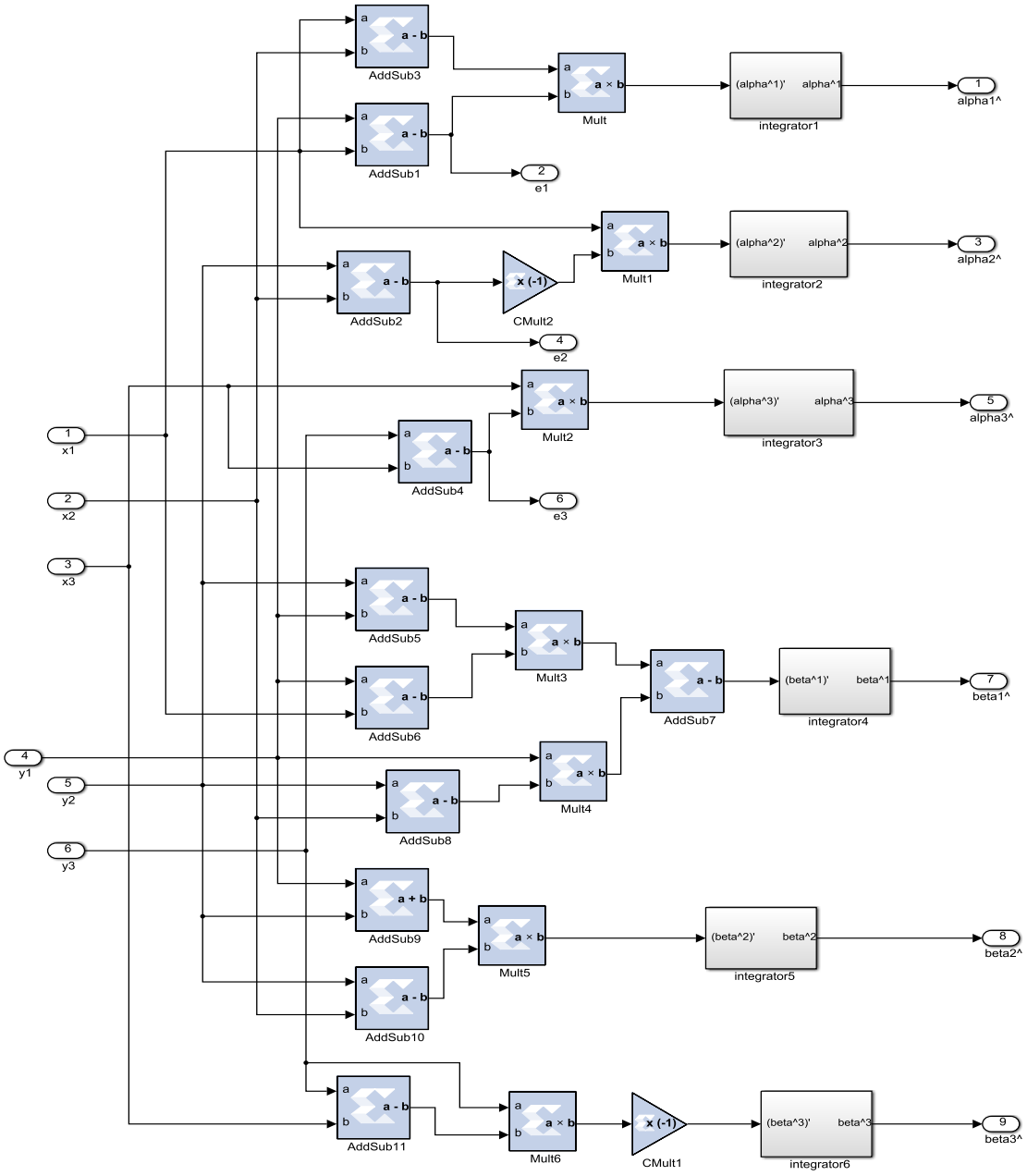
Eş. 3.16'daki slave sistem, xilinx bloksetleri kullanılarak tasarlanmış ve yine master sistem içerisinde kullanılan integrator alt sistemi kullanılmıştır. Slave sistemin Matlab/Simulink modeli gösterilmiştir (Şekil 3.17).



Şekil 3.17. Slave sistem.

3.8.2.3.3. Optimize edilmiş uyarlamalı parametrelerin Matlab/Simulink modellemesi

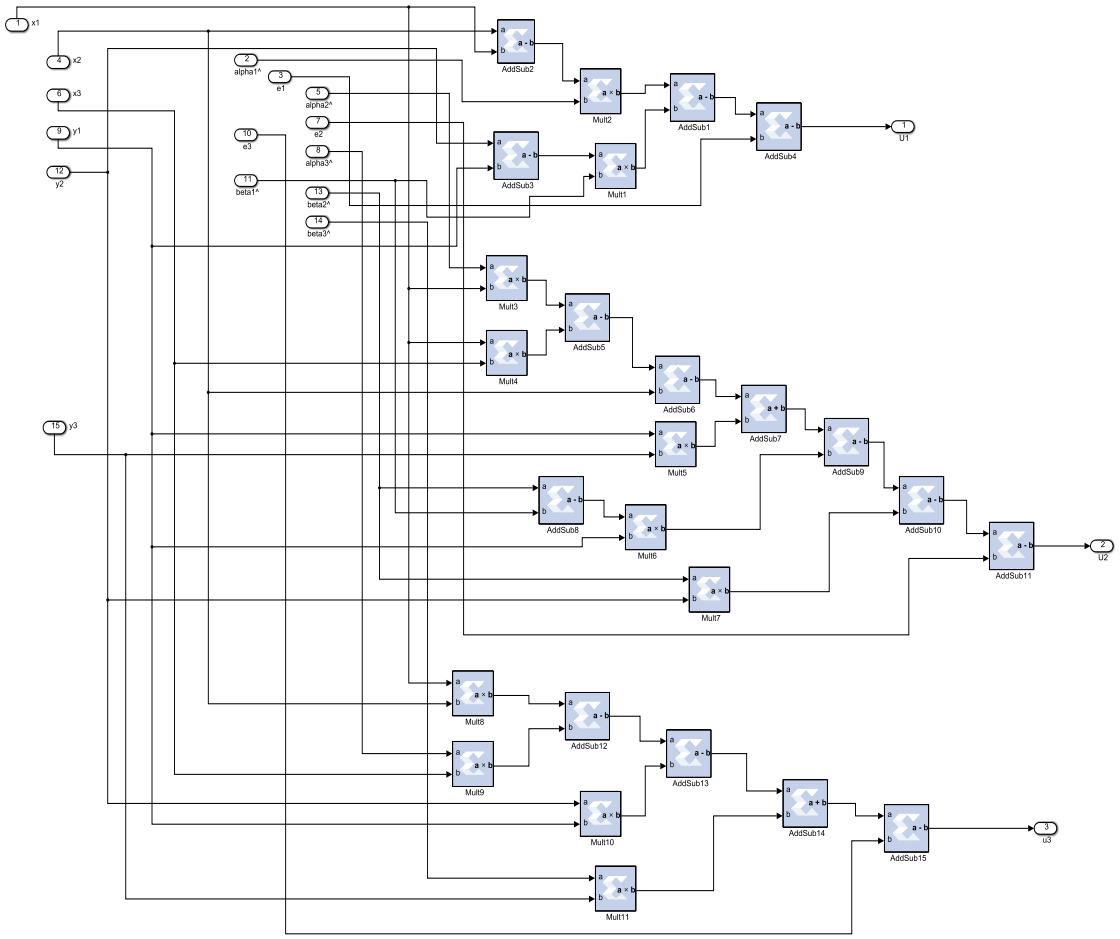
Eş. 3.82 ve Eş. 3.83'de verilen matematiksel ifadelerin Matlab/Simulink modellenmesi yapılıp yine master sistem içerisinde kullanılan integrator alt sistemi kullanılmıştır. Sistemin Matlab/Simulink modeli gösterilmiştir (Şekil 3.18).



Şekil 3.18. Optimize edilmiş uyarlamalı parametreler.

3.8.2.3.4. Kontrol fonksiyonlarının Matlab/Simulink modellemesi

Eş. 3.81'deki matris denklemine göre oluşturulan u_1, u_2 ve u_3 uyarlamalı kontrol fonksiyonlarının Matlab/Simulink modeli oluşturulup çıkışlar slave sisteme giriş olarak gönderilmiştir. Denetleyici sisteminin Matlab/Simulink modeli gösterilmiştir (Şekil 3.19).

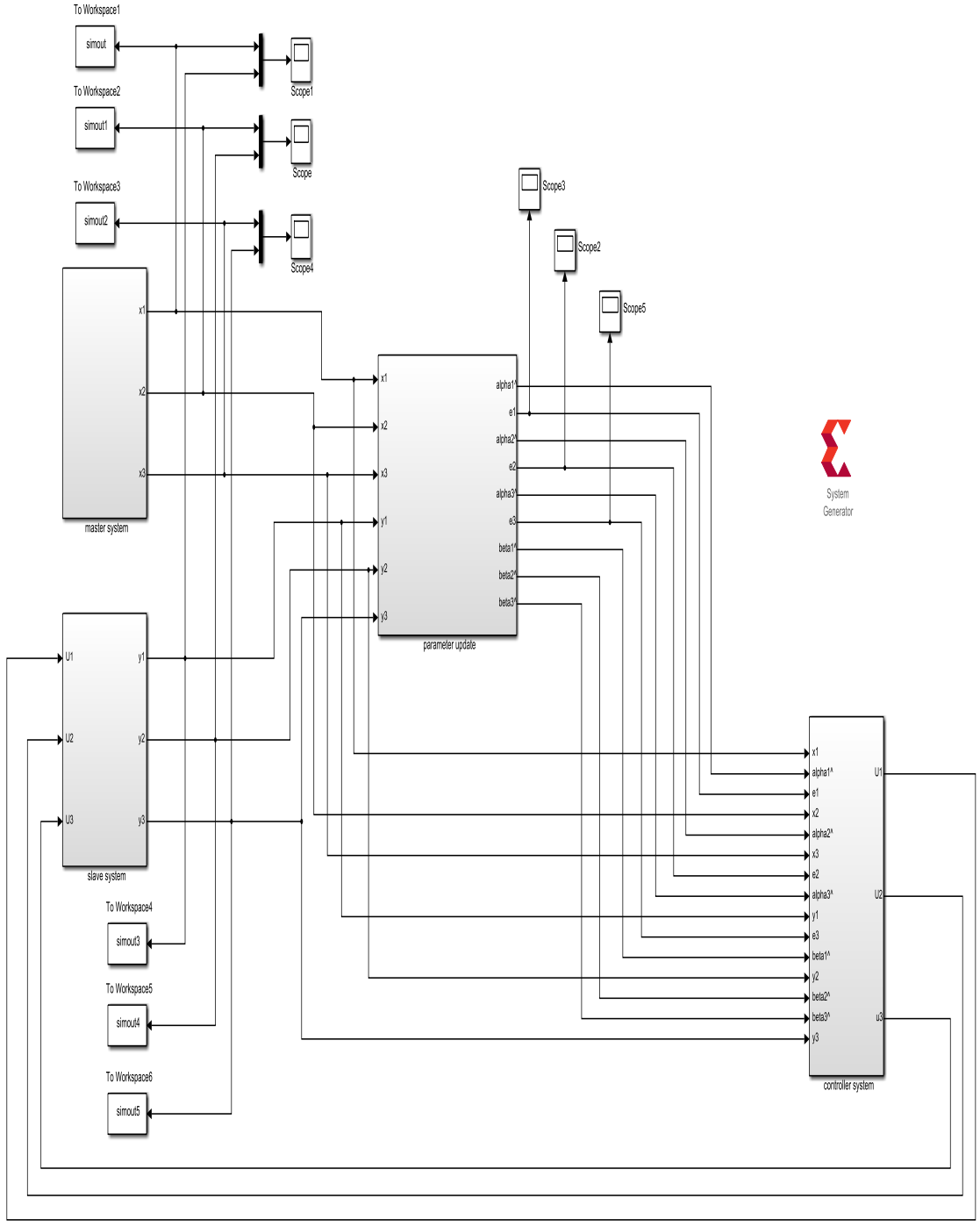


Şekil 3.19. Denetleyici sistem.

3.8.2.3.5. Oluşturulan sistemlerin birleştirilmesi

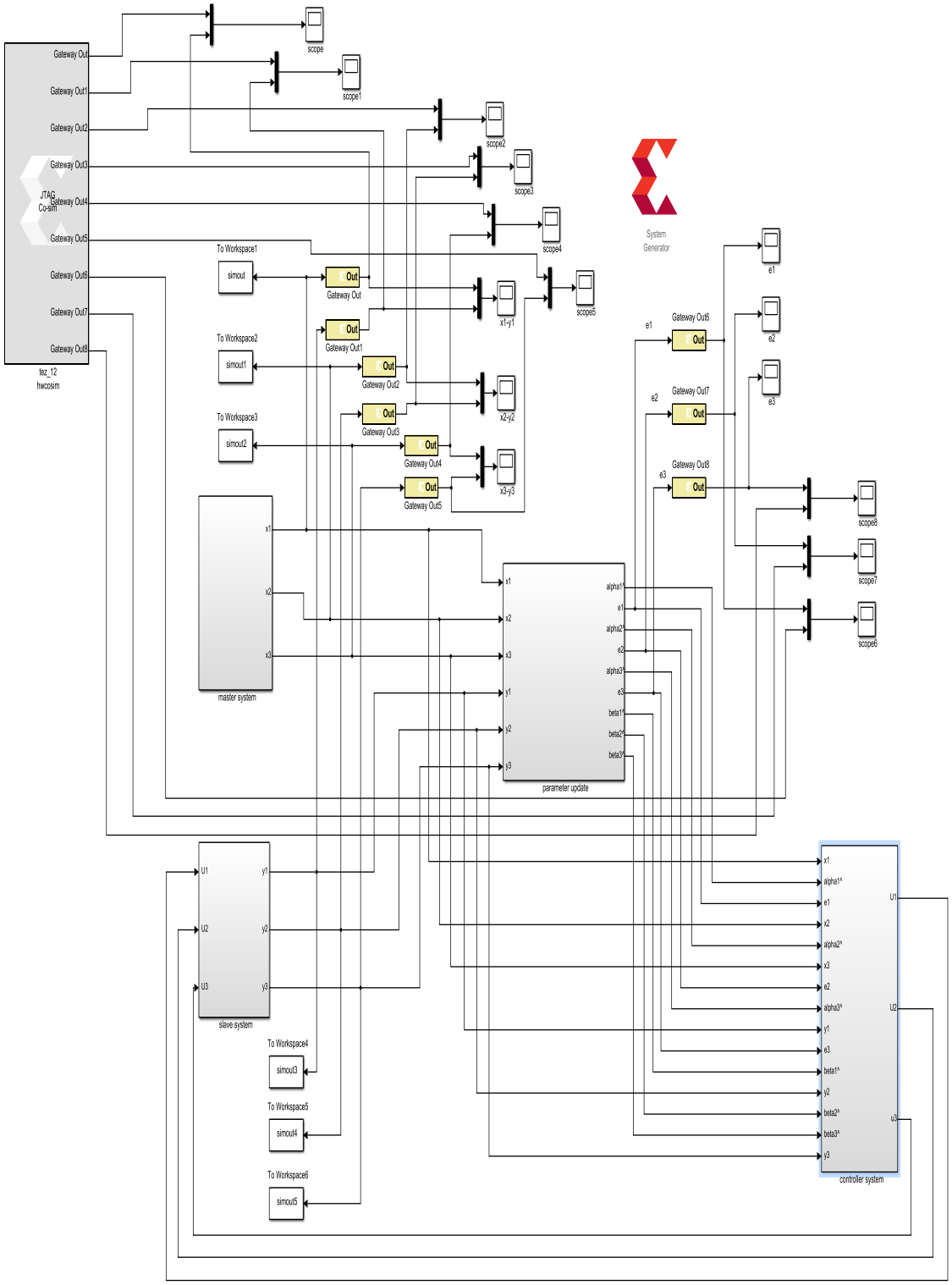
Ayrı ayrı matematiksel ifadeleri bulunan sistemlerin ayrı ayrı modellenmeleri yapılmış ve önceki bölümlerde açıklanmıştır. Ayrı ayrı tasarlanan bu sistemler temel olarak 4 farklı alt sistem haline getirilip ardından "system generator" bloğu da

eklenerek sistem nihai halini almıştır. Böylece senkronize edilmiş kaotik sistemin Matlab/Simulink tasarımı tamamlanmış ve modelin blok diyagramı daha sade ve kolay anlaşılır hale getirilmiştir. Senkronize edilmiş olan kaotik sistemin nihai Matlab/Simulink modeli gösterilmiştir (Şekil 3.20).



Şekil 3.20. Senkronize edilmiş kaotik sistem modeli.

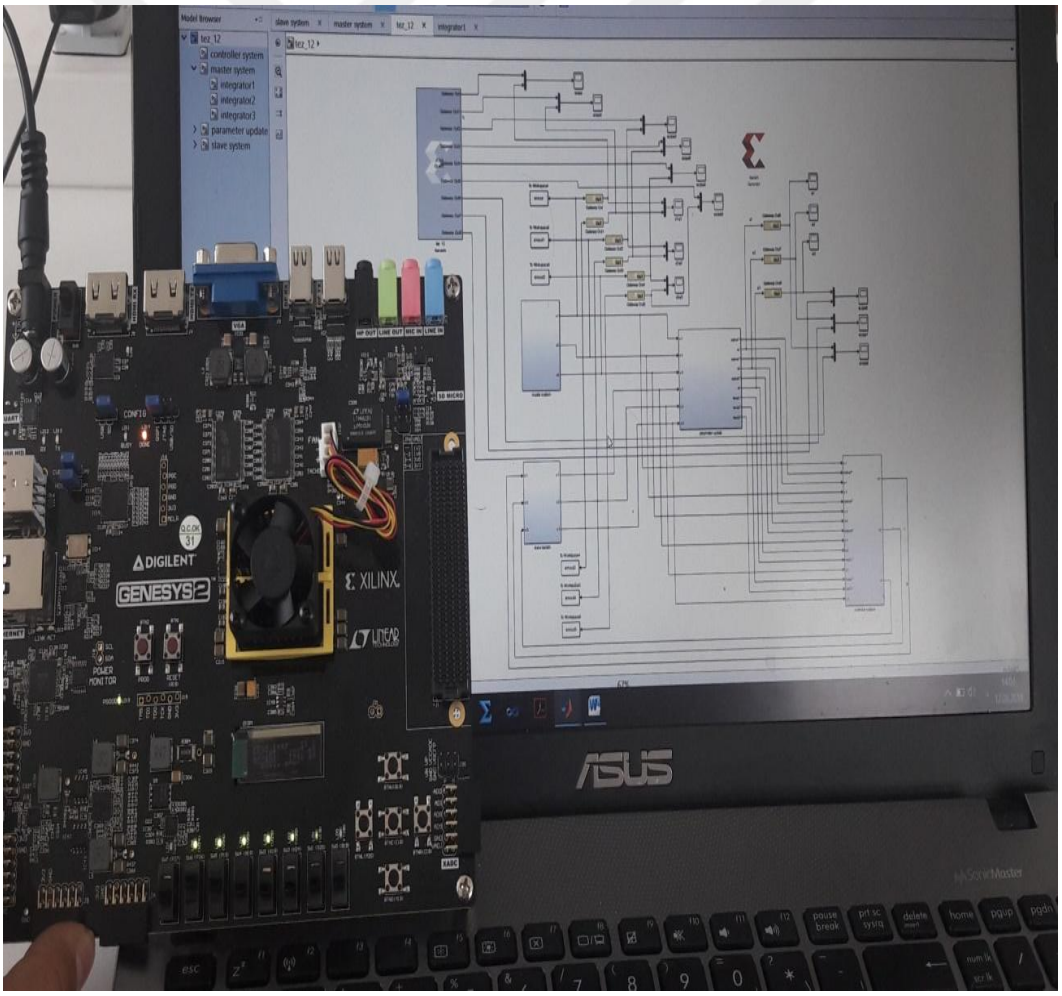
3.8.2.3.6. Hardware co-simulation yöntemiyle modellenen sistemin gerçek zamanlı uygulaması



Şekil 3.21. Senkronize edilmiş sistemin hardware co-simulation tasarımı.

Senkronize edilmiş kaotik sistem Matlab/Simulink ortamında tasarlandıktan sonra hardware co-simulation yöntemiyle FPGA'ya gönderilecek ve gerçek zamanlı çıkış bilgileri yine Matlab/Simulink ortamında gözlenecektir.

Sistemin çıkışlarını elde etmek ve bilgisayar ile FPGA nın bilgi alışverişini sağlamak gerekir. Bunun için "system generator" bloğuna çift tıklanır ve açılan pencerede, kullanılan FPGA modeli ve sistem periyodu uygun şekilde seçilerek "generate" sekmesine tuşlanır ve JTAG bloğu üretilmiş olur. Tüm sistemin final tasarımı görülmektedir (Şekil 3.21). Burada kapsamlı gözlem yapabilmek için; master ve slave sistemin durum değişkenleri ($x_1, x_2, x_3, y_1, y_2, y_3$) ve sistem senkronizasyon hataları (e_1, e_2, e_3) çıkış olarak alınmıştır. Ayrıca final tasarımın FPGA'ya uygulanma anı gösterilmiştir (Şekil 3.22).

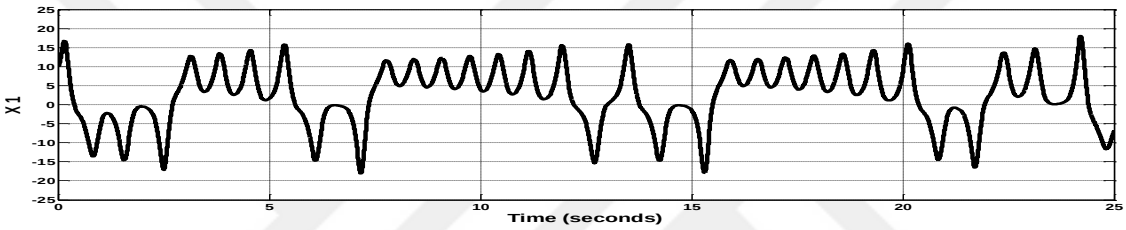


Şekil 3.22. Senkronize edilmiş sistemin hardware co-simulation uygulaması.

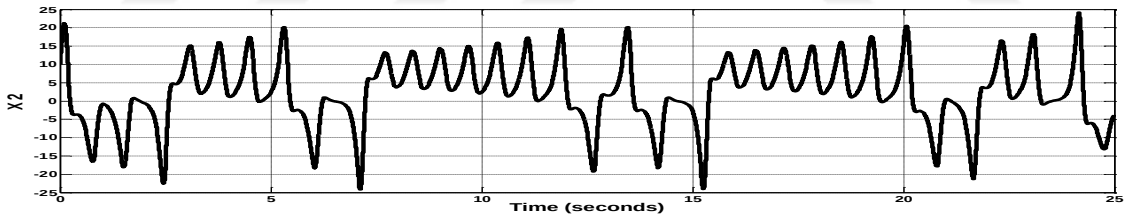
4. BULGULAR

Önceki bölümlerde Matlab/Simulink blokları kullanılarak aktif kayan kipli ve uyarlamalı kontrol yöntemleriyle ayrı ayrı tasarımları yapılan senkronizasyon sistemlerinin yine Matlab/Simulink üzerinde simülasyonu yapılmış ve ardından FPGA'ya gömülerek gerçek zamanlı sonuçlar elde edilmiştir.

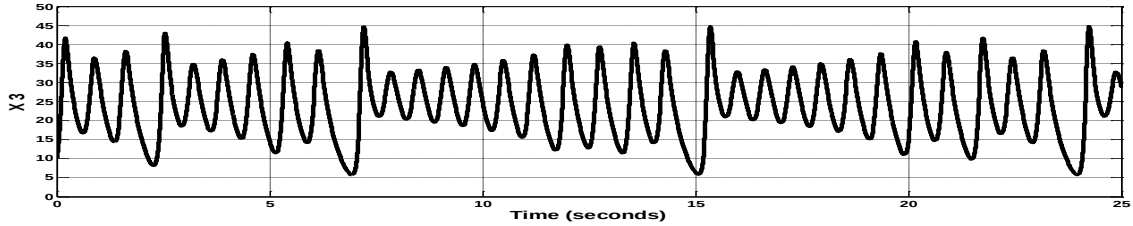
4.1. Uyarlamalı Kontrol Yöntemiyle Gerçekleştirilen Senkronizasyon Sonuçları



a. x_1 durum değişkeninin zamana göre değişimi.



b. x_2 durum değişkeninin zamana göre değişimi.

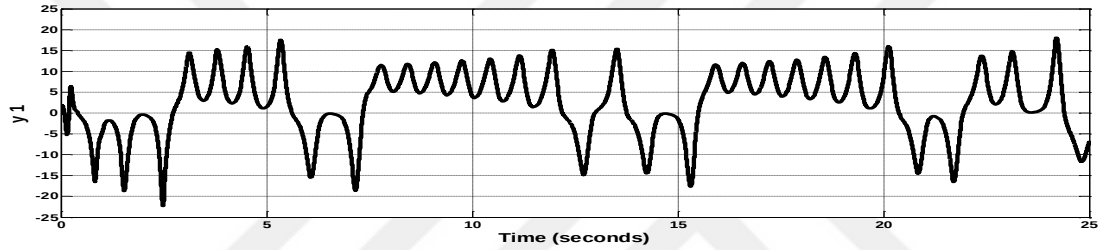


c. x_3 durum değişkeninin zamana göre değişimi.

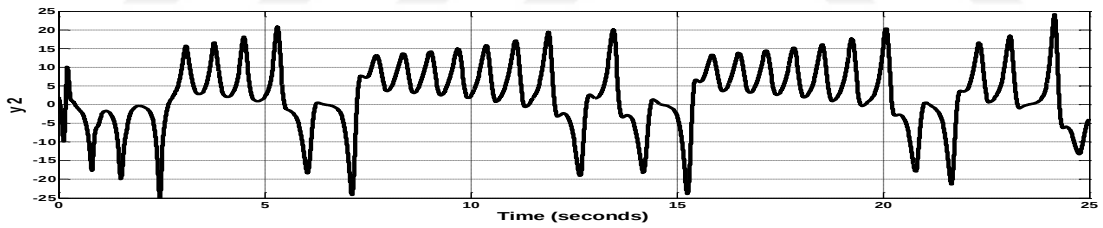
Şekil 4.1. Master sistem durum değişkenlerinin zamana göre değişimi a. x_1 durum değişkeninin zamana göre değişimi, b. x_2 durum değişkeninin zamana göre değişimi, c. x_3 durum değişkeninin zamana göre değişimi.

Senkronize edilmiş sistemin hardware co-simulation tasarımı elde edildikten sonra yine Matlab/Simulink üzerinde program çalıştırılarak FPGA'ya sistem bilgileri gönderilir ve aynı şekilde FPGA'dan gerçek zamanlı bilgi alınır. Scope, Scope1 ve Scope2'de master sistem durum değişkenlerinin sonuçları izlenebilir (Bkz. Şekil 3.21). Master sistem durum değişkenlerinin zamana bağlı grafikleri gösterilmiştir (Şekil 4.1).

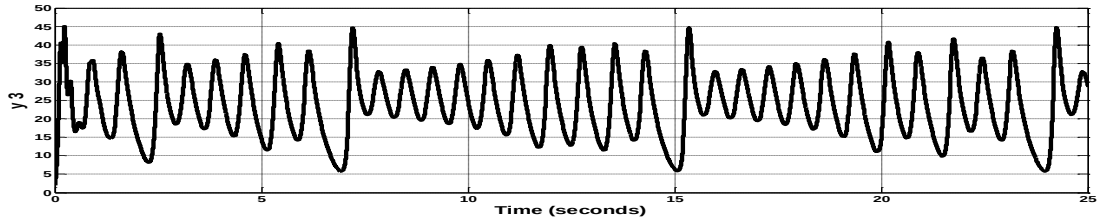
Burada her 3 durum değişkeninin (x_1, x_2, x_3) zamana göre değişimi incelendiğinde beklenen kaotik bir yapı olduğunu söylemek mümkündür. Ayrıca Lorenz sisteminin doğası gereği x_1 ve x_2 durum değişkenlerinin birbirlerine benzer, x_3 durum değişkenine göre ise farklı bir grafik çizdikleri görülmektedir.



a. y_1 durum değişkeninin zamana göre değişimi.



b. y_2 durum değişkeninin zamana göre değişimi.



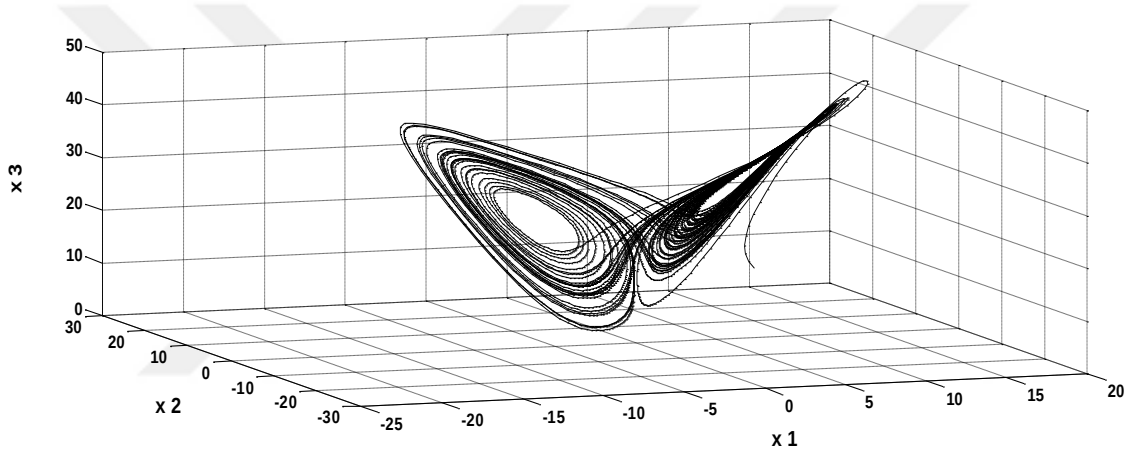
c. y_3 durum değişkeninin zamana göre değişimi.

Şekil 4.2. Slave sistem durum değişkenlerinin zamana göre değişimi a. y_1 durum değişkeninin zamana göre değişimi, b. y_2 durum değişkeninin zamana göre değişimi, c. y_3 durum değişkeninin zamana göre değişimi.

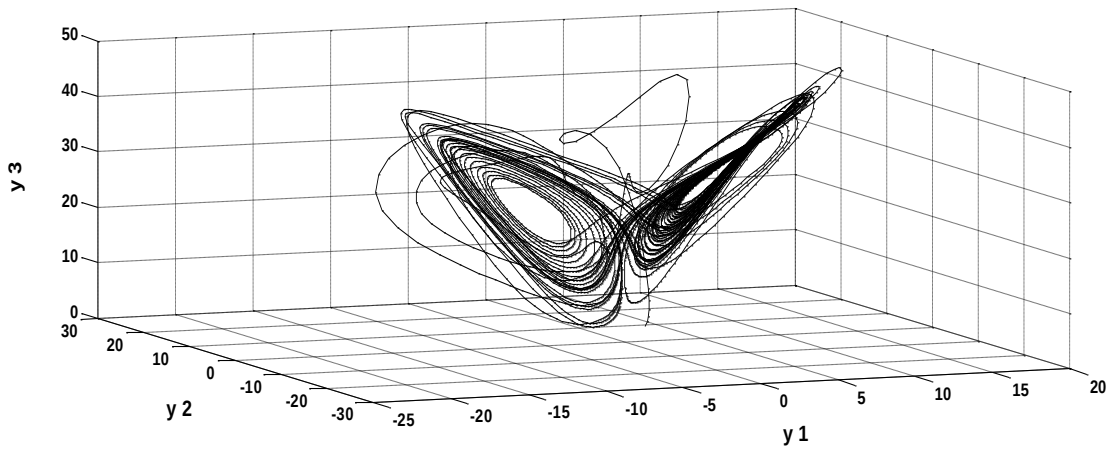
Scope3, Scope4 ve Scope5'de slave sistem durum değişkenlerinin sonuçları izlenir (Bkz. Şekil 3.21). Slave sistem durum değişkenlerinin zamana bağlı grafikleri gösterilmiştir (Şekil 4.2).

Yine burada her 3 durum değişkeninin (y_1, y_2, y_3) zamana göre grafikleri incelenip Şekil 4.1 ile karşılaştırıldığında slave sistemin, master sisteme senkronize olduğu görülür.

Hem master sistem hem de slave sistem durum değişkenlerinin kendi içlerinde birbirlerine göre grafikleri çizilirse senkronizasyonun gerçekleştiği daha net görülür (Şekil 4.3 ve Şekil 4.4).

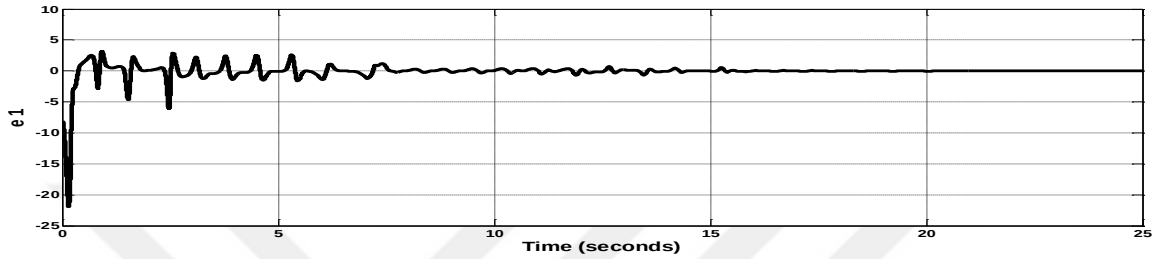


Şekil 4.3. Master sistem durum değişkenlerinin birbirlerine göre değişimi.

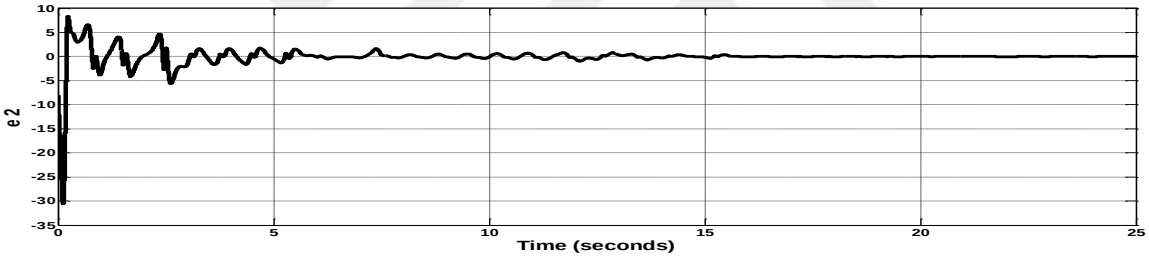


Şekil 4.4. Slave sistem durum değişkenlerinin birbirlerine göre değişimi.

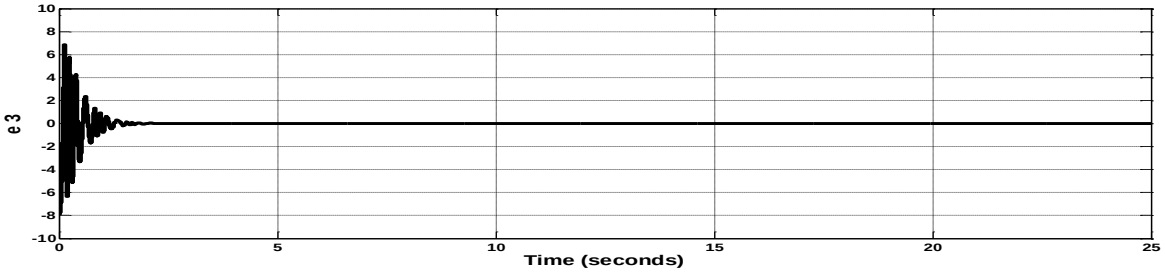
Şekil 4.3 ve Şekil 4.4'ü birlikte ele aldığımızda sistemin senkronize olduğu ama slave durum değişkenlerine uygulanan denetleyici etkisiyle, sistem oturana kadar bazı aşmalar olduğu görülür. Sistem hataları olarak adlandırdığımız bu aşmaları daha iyi anlamak için Scope6, Scope7 ve Scope8'deki sistem senkronizasyon hataları izlenir (Bkz. Şekil 3.21). Senkronizasyon hatalarının zamana göre değişimi gösterilmiştir (Şekil 4.5).



a. Sistem birinci durum değişkeni hatasının (y_1 - x_1) zamana göre değişimi.



b. Sistem ikinci durum değişkeni hatasının (y_2 - x_2) zamana göre değişimi.

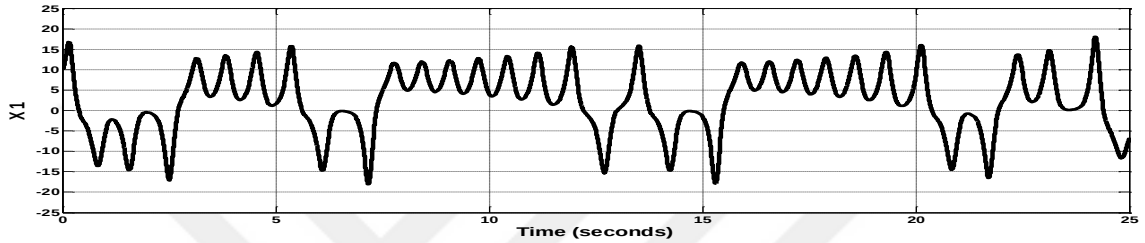


c. Sistem üçüncü durum değişkeni hatasının (y_3 - x_3) zamana göre değişimi.

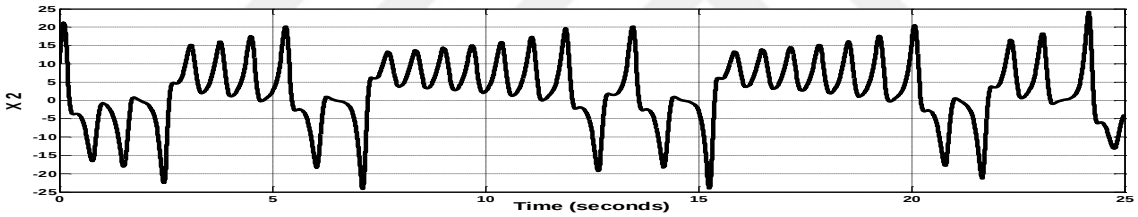
Şekil 4.5. Senkronizasyon hatalarının zamana göre değişimi a. Sistem birinci durum değişkeni hatasının (y_1 - x_1) zamana göre değişimi, b.Sistem ikinci durum değişkeni hatasının (y_2 - x_2) zamana göre değişimi, c. Sistem üçüncü durum değişkeni hatasının (y_3 - x_3) zamana göre değişimi.

Burada e_1 ve e_2 sistem durum değişkeni hatalarının birbirlerine yakın sürelerde 0'a yaklaştığı ve benzer aşma miktarlarına sahip olduğu görülmektedir. Ancak e_3 sistem durum değişkeni hatasının e_1 ve e_2'ye göre daha çabuk 0'a yaklaştığı ancak salınımın çok daha fazla olduğu görülmektedir.

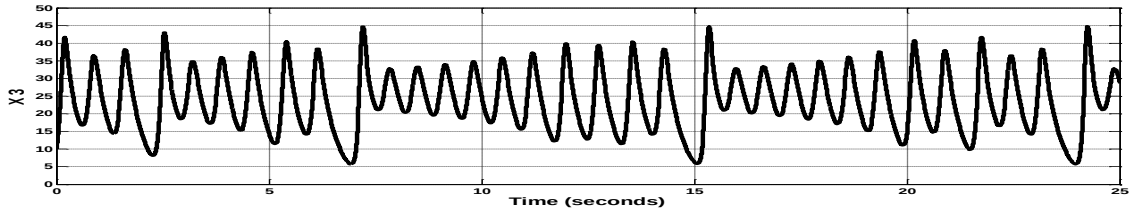
4.2. Aktif Kayan Kipli Kontrol Yöntemiyle Gerçekleştirilen Senkronizasyon Sonuçları



a. x_1 durum değişkeninin zamana göre değişimi.



b. x_2 durum değişkeninin zamana göre değişimi.



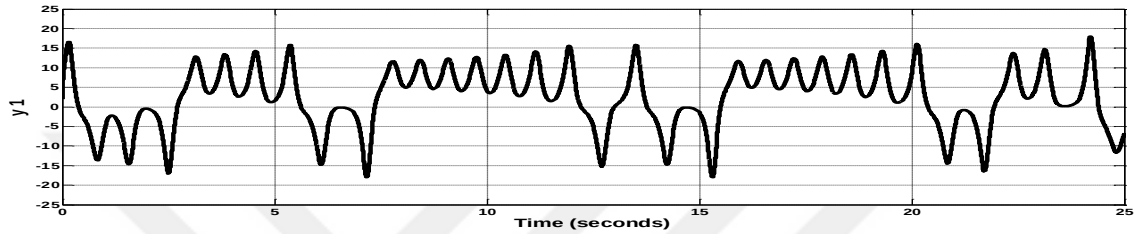
c. x_3 durum değişkeninin zamana göre değişimi.

Şekil 4.6. Master sistem durum değişkenlerinin zamana göre değişimi a. x_1 durum değişkeninin zamana göre değişimi, b. x_2 durum değişkeninin zamana göre değişimi, c. x_3 durum değişkeninin zamana göre değişimi.

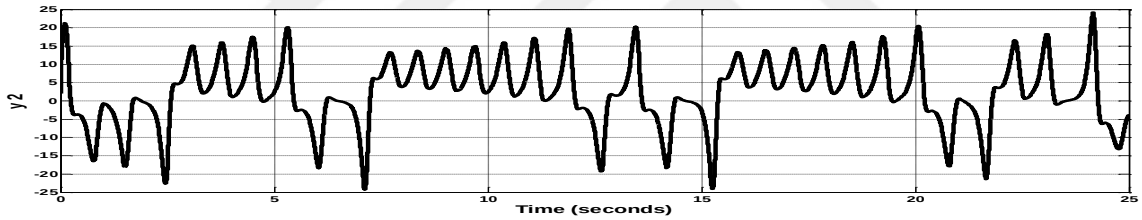
Senkronize edilmiş sistemin hardware co-simulation tasarımı elde edildikten sonra yine Matlab/Simulink üzerinde program çalıştırılarak FPGA'ya sistem bilgileri

gönderilir ve aynı şekilde FPGA'dan gerçek zamanlı bilgi alınır. Scope, Scope1 ve Scope2'de master sistem durum değişkenlerinin sonuçları izlenebilir (Bkz. Şekil 3.14). Master sistem durum değişkenlerinin zamana bağlı grafikleri gösterilmiştir (Şekil 4.6).

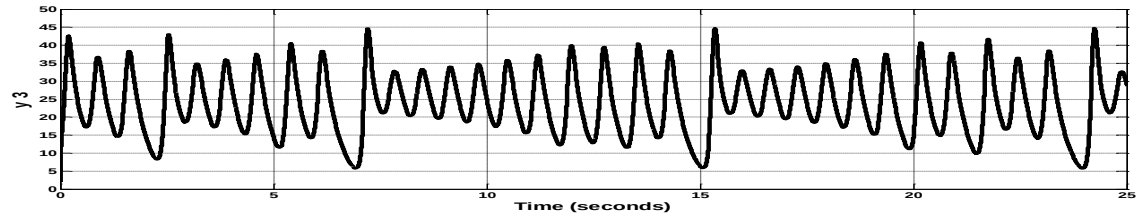
Burada her 3 durum değişkeninin (x_1, x_2, x_3) zamana göre değişimi incelendiğinde beklenen kaotik bir yapı olduğunu söylemek mümkündür. Ayrıca Lorenz sisteminin doğası gereği x_1 ve x_2 durum değişkenlerinin birbirlerine benzer, x_3 durum değişkenine göre farklı bir grafik çizdikleri görülmektedir.



a. y_1 durum değişkeninin zamana göre değişimi.



b. y_2 durum değişkeninin zamana göre değişimi.



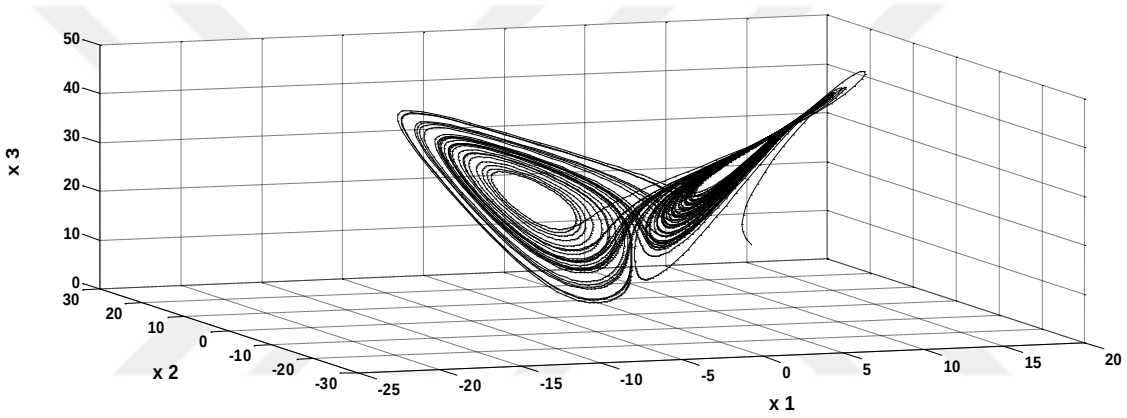
c. y_3 durum değişkeninin zamana göre değişimi.

Şekil 4.7. Slave sistem durum değişkenlerinin zamana göre değişimi a. y_1 durum değişkeninin zamana göre değişimi, b. y_2 durum değişkeninin zamana göre değişimi, c. y_3 durum değişkeninin zamana göre değişimi.

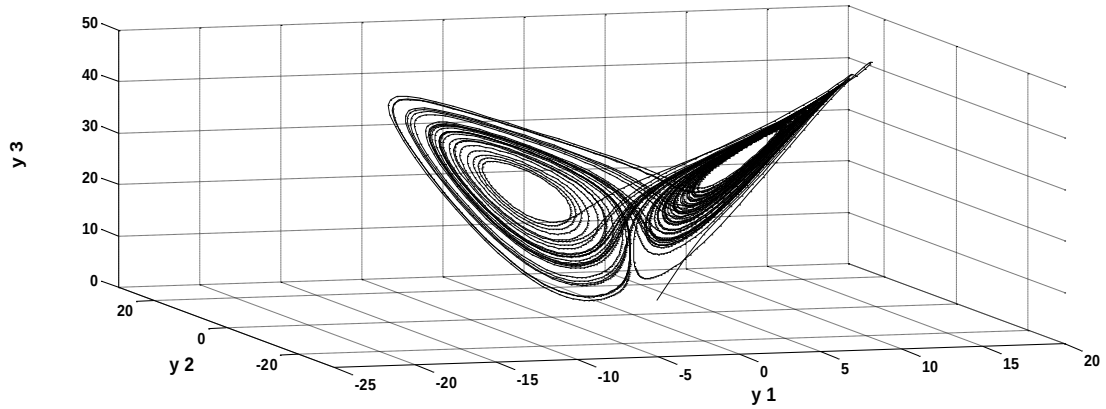
Scope3, Scope4 ve Scope5'de slave sistem durum değişkenlerinin sonuçları izlenir (Bkz. Şekil 3.14). Slave sistem durum değişkenlerinin zamana bağlı grafikleri gösterilmiştir (Şekil 4.7).

Yine burada her 3 durum değişkeninin (y_1, y_2, y_3) zamana göre grafikleri incelenip Şekil 4.6 ile karşılaştırıldığında slave sistemin master sisteme senkronize olduğu görülür.

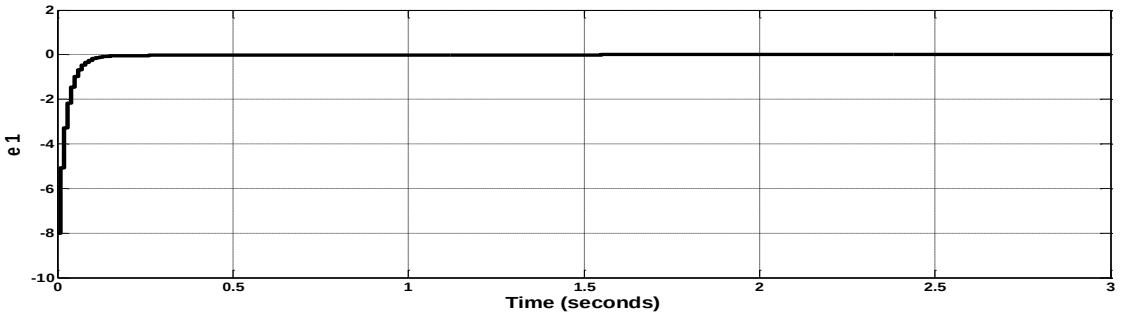
Hem master sistem hem de slave sistem durum değişkenlerinin kendi içlerinde birbirlerine göre grafikleri çizilirse senkronizasyonun gerçekleştiği daha net görülür (Şekil 4.8 ve Şekil 4.9).



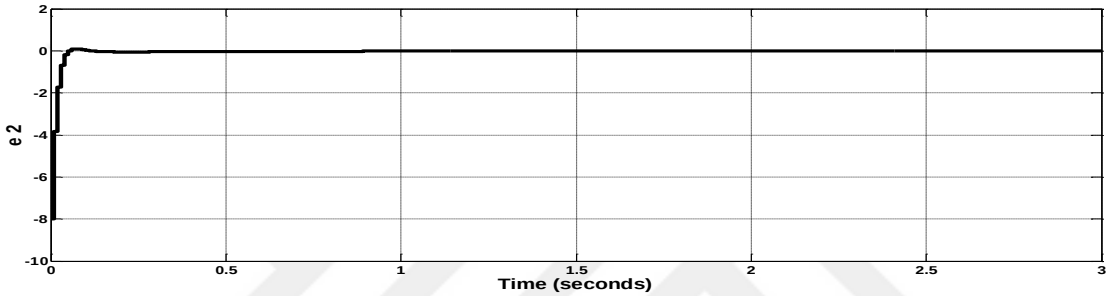
Şekil 4.8. Master sistem durum değişkenlerinin birbirlerine göre değişimi.



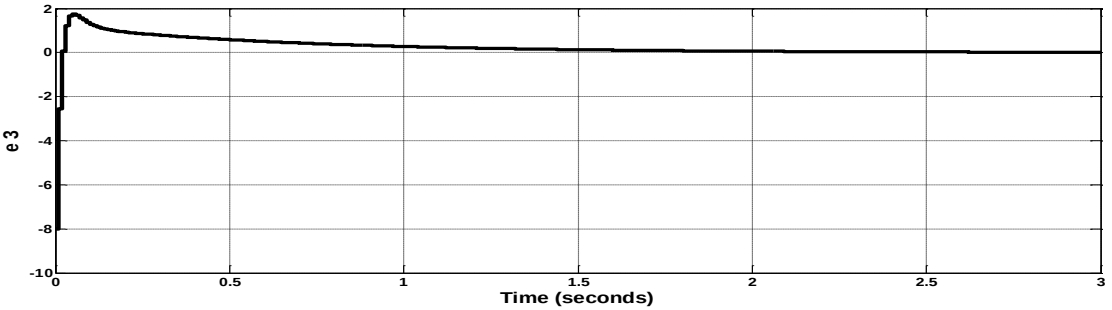
Şekil 4.9. Slave sistem durum değişkenlerinin birbirlerine göre değişimi.



a. Sistem birinci durum değişkeni hatasının ($y_1 - x_1$) zamana göre değişmi.



b. Sistem ikinci durum değişkeni hatasının ($y_2 - x_2$) zamana göre değişimi.



c. Sistem üçüncü durum değişkeni hatasının ($y_3 - x_3$) zamana göre değişimi.

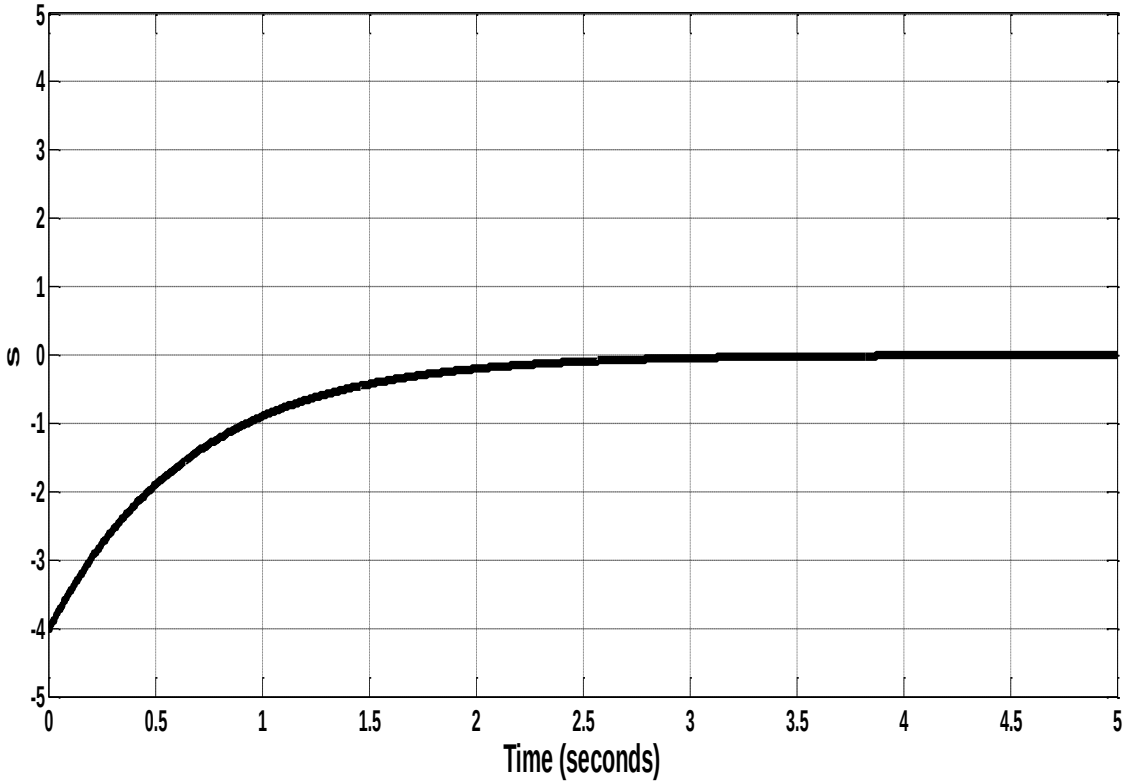
Şekil 4.10. Senkronizasyon hatalarının zamana göre değişimi a. Sistem birinci durum değişkeni hatasının ($y_1 - x_1$) zamana göre değişmi, b. Sistem ikinci durum değişkeni hatasının ($y_2 - x_2$) zamana göre değişimi, c. Sistem üçüncü durum değişkeni hatasının ($y_3 - x_3$) zamana göre değişimi.

Şekil 4.8 ve Şekil 4.9'u birlikte ele aldığımızda sistemin senkronize olduğu ama slave durum değişkenlerine uygulanan denetleyici etkisiyle, sistem oturana kadar bazı aşmalar olduğu görülür. Sistem hataları olarak adlandırıdığımız bu aşmaları daha iyi

anlayabilmek için Scope6, Scope7 ve Scope8'deki sistem senkronizasyon hataları izlenir (Bkz. Şekil 3.14). Senkronizasyon hatalarının zamana göre değişimi gösterilmiştir (Şekil 4.10).

Burada e_1 ve e_2 sistem hatalarının birbirlerine yakın sürelerde 0'a yaklaştığı ve benzer aşma miktarlarına sahip olduğu görülmektedir. Ancak e_3 sistem hatasının e_1 ve e_2'ye göre daha geç 0'a yaklaştığı ve aşma miktarının çok daha fazla olduğu görülmektedir.

Ayrıca "s" kayma yüzeyinin zamana göre değişimi gösterilmiştir (Şekil 4.11).

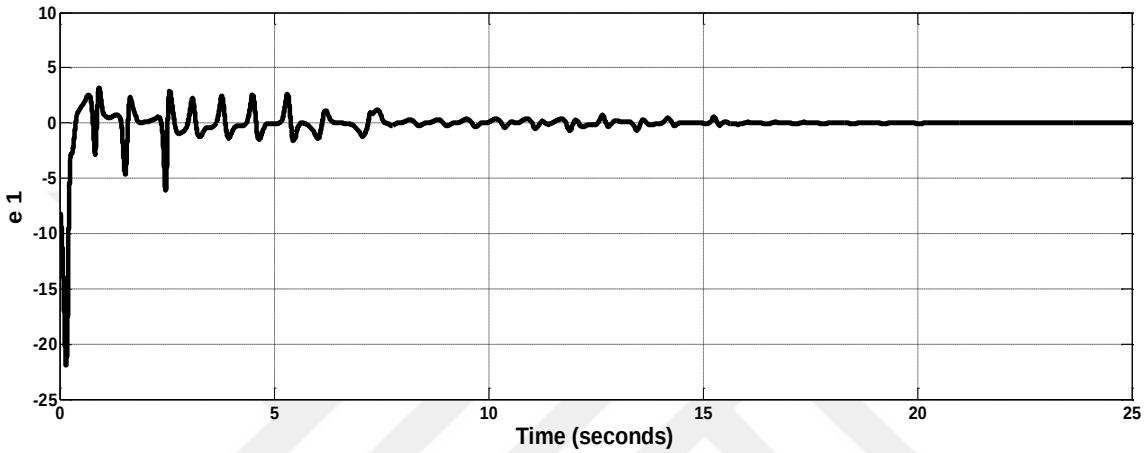


Şekil 4.11. "s" kayma yüzeyinin zamana göre değişimi.

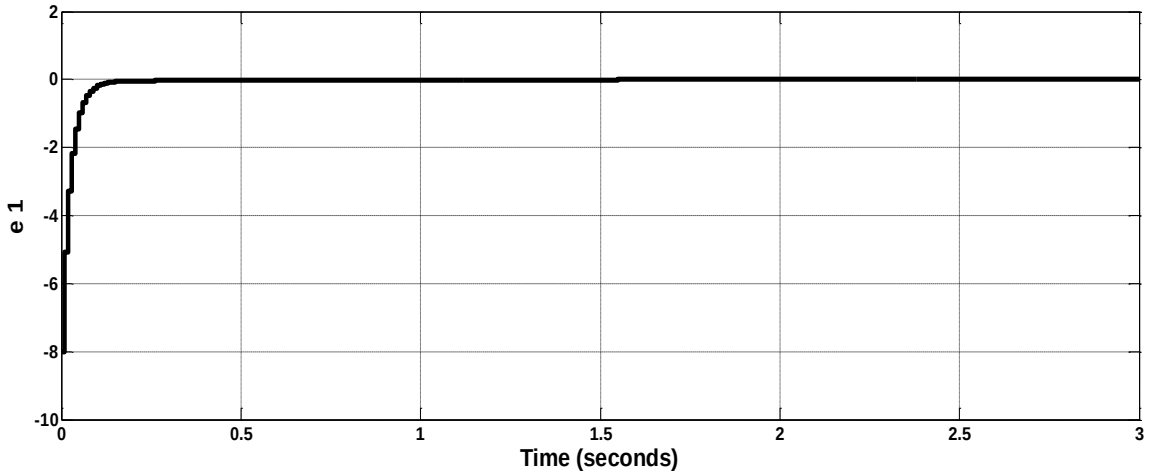
Şekil 4.11 gözönüne aldığında kayma yüzeyi değerinin yaklaşık 4 saniye sonra 0'a yaklaştığı görülmektedir. Bu da sistemin yaklaşık 4 saniyede oturduğunu gösterir.

4.3. Uyarlamalı Kontrol ve Aktif Kayan Kipli Kontrol Yöntemleriyle Gerçekleştirilen Senkronizasyon Sonuçlarının Karşılaştırılması

Senkronizasyon sistemine uyarlamalı kontrol ve aktif kayan kipli kontrol uygulandığında oluşan sistem birinci durum hataları (e_1) sırasıyla verilmiştir (Şekil 4.12).



a. Uyarlamalı kontrol.

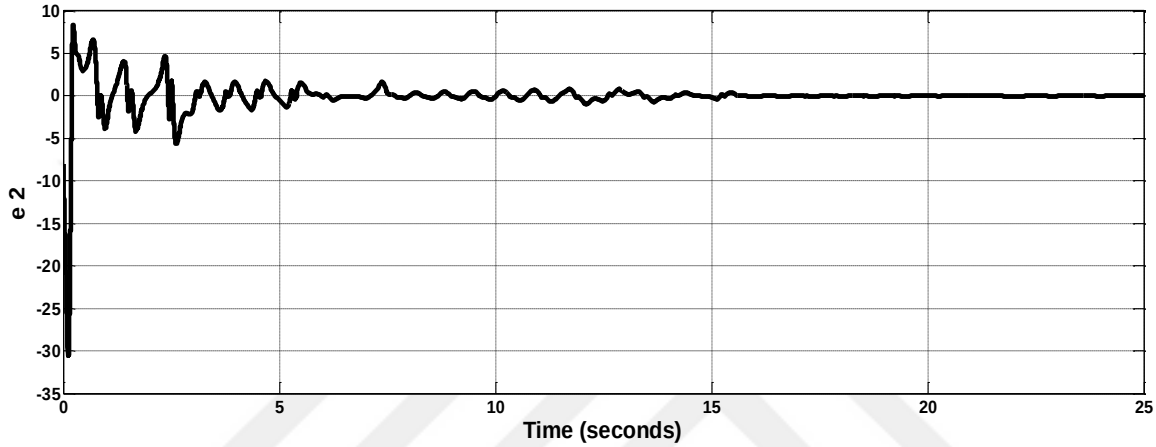


b. Aktif kayan kipli kontrol.

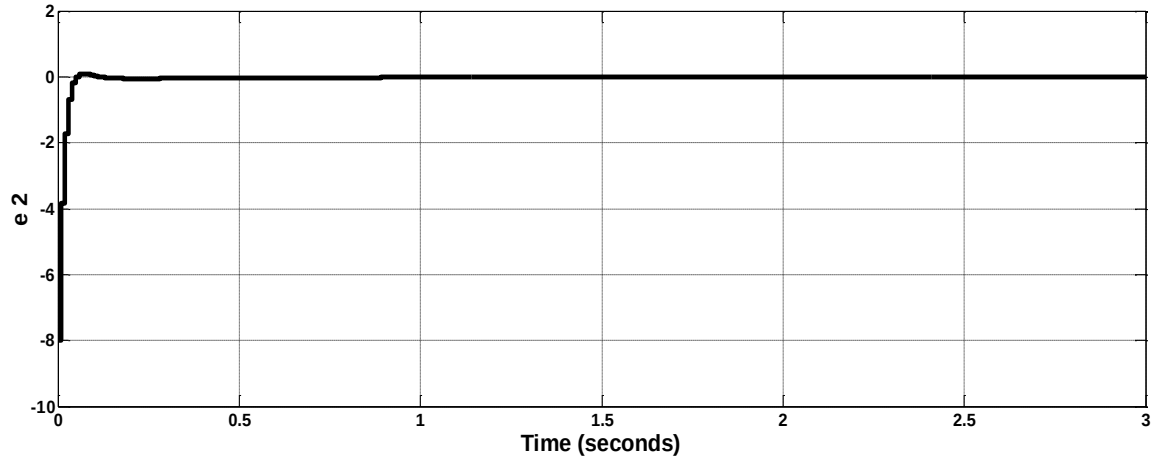
Şekil 4.12. Sistem birinci durum değişkeni hataları a. Uyarlamalı kontrol, b. Aktif kayan kipli kontrol.

Şekil 4.12'de verilen her iki kontrol yöntemi için sistem hataları ikili olarak karşılaştırıldığında sistem birinci durum değişkeni hatasının aktif kayan kipli kontrol altında yaklaşık 0.1 saniyede 0'a yaklaştığı görülmektedir. Buna karşılık uyarlamalı kontrol altında aynı hata, yaklaşık 20 saniyede 0'a yaklaşmıştır.

Sistem ikinci durum hataları (e_2) ise sırasıyla verilmiştir (Şekil 4.13).



a. Uyarlamalı kontrol.

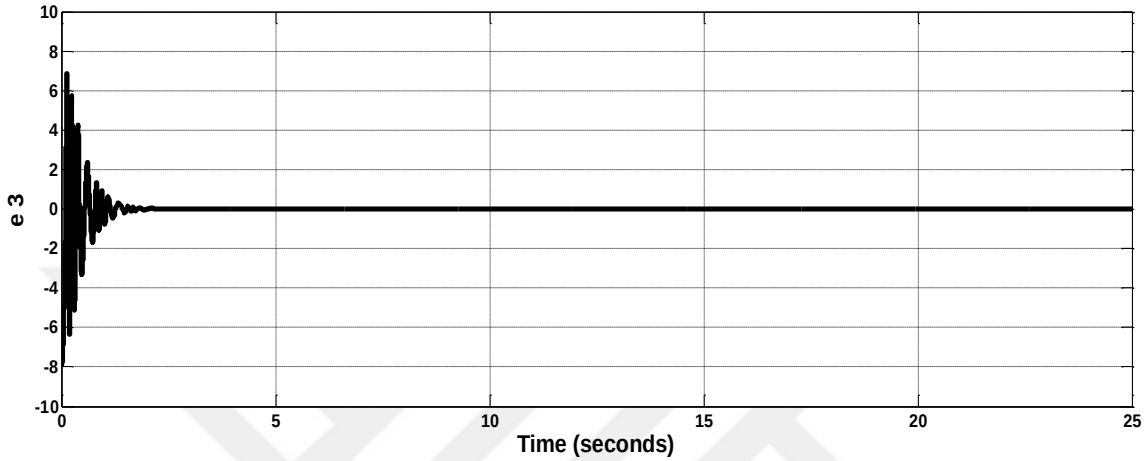


b. Aktif kayan kipli kontrol.

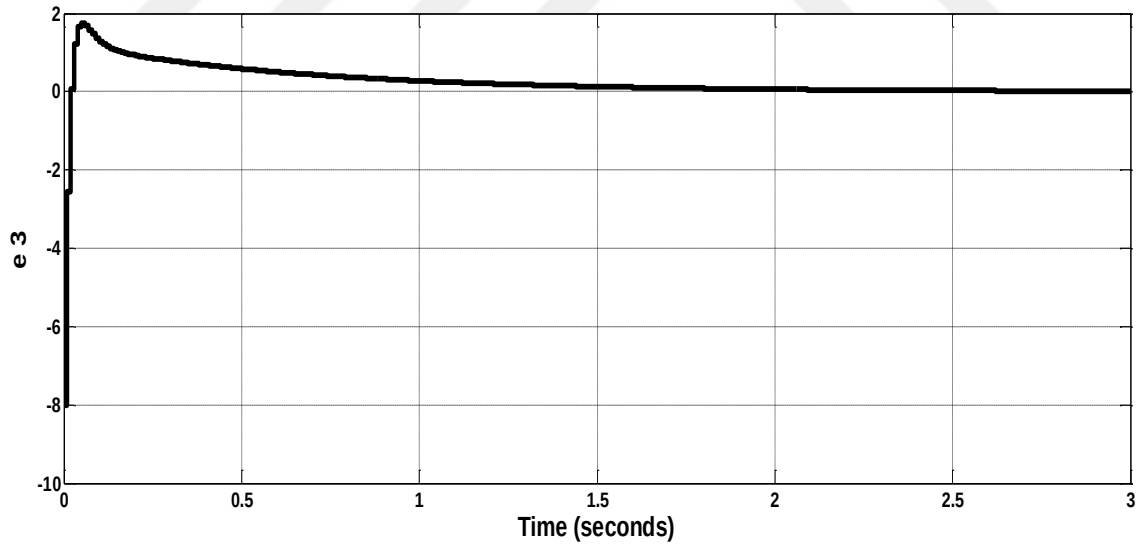
Şekil 4.13. Sistem ikinci durum değişkeni hataları a. Uyarlamalı kontrol, b. Aktif kayan kipli kontrol.

Şekil 4.13'de verilen ikinci durum değişkeni hatasının aktif kayan kipli kontrol altında yaklaşık 0.2 saniyede 0'a yaklaştığı görülmektedir. Buna karşılık uyarlamalı kontrol altında aynı hata yaklaşık 20 saniyede 0'a yaklaşmıştır.

Sistem üçüncü durum hataları (e_3) sırasıyla verilmiştir (Şekil 4.14).



a. Uyarlamalı kontrol.

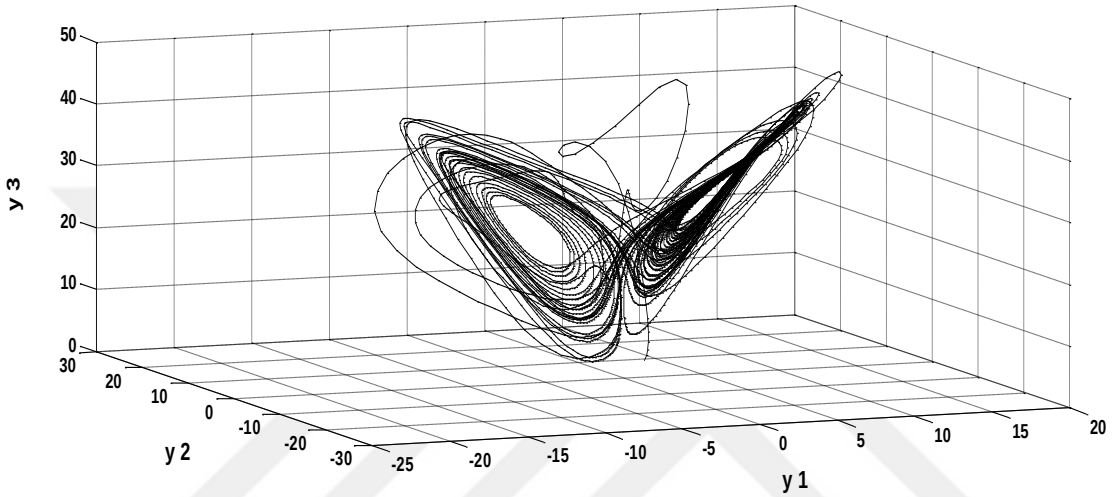


b. Aktif kayan kipli kontrol.

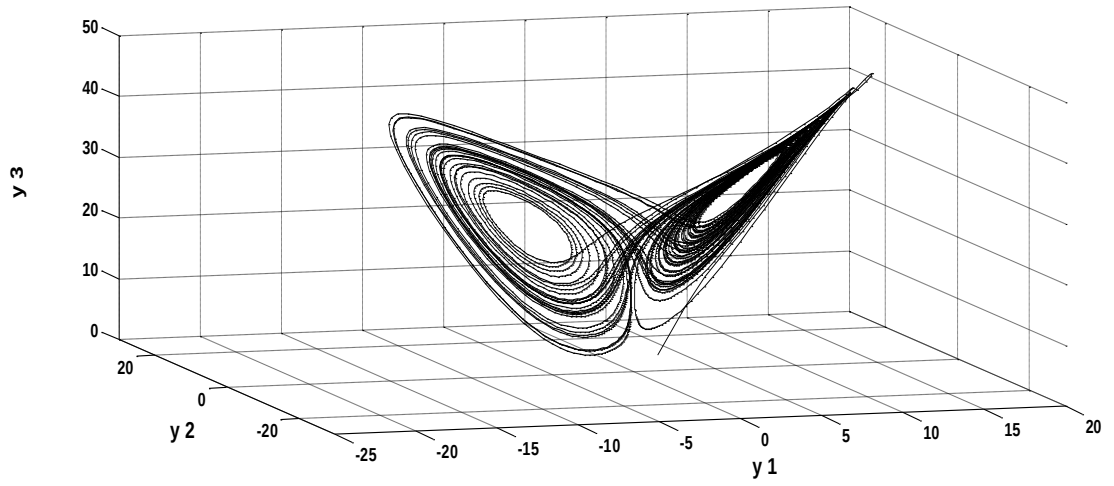
Şekil 4.14. Sistem üçüncü durum değişkeni hataları a. Uyarlamalı kontrol, b. Aktif kayan kipli kontrol.

Şekil 4.14'de verilen sistem üçüncü durum değişkeni hatasının aktif kayan kipli kontrol altında yaklaşık 1.8 saniyede 0'a yaklaştığı görülmektedir. Buna karşılık uyarlamalı kontrol altında aynı hata yaklaşık 2 saniyede 0'a yaklaşmıştır.

Senkronizasyon sistemine uyarlamalı kontrol ve aktif kayan kipli kontrol uygulandığında slave sistem durum değişkenlerinin (y_1, y_2, y_3) birbirlerine göre değişimleri verilmiştir (Şekil 4.15).



a. Uyarlamalı kontrol.



b. Aktif kayan kipli kontrol.

Şekil 4.15. Senkronizasyon sonucu slave sistem durum değişkenlerinin birbirlerine göre grafiği a. Uyarlamalı kontrol, b. Aktif kayan kipli kontrol.

Şekil 4.15’de aktif kayan kipli kontrol uygulandığı durumda, uyarlamalı kontrol yöntemine göre slave sistemin bütün olarak daha hızlı ve daha az aşma miktarıyla master sisteme senkron olduğu görülür. Bu yüzden senkronizasyon işleminin aktif kayan kipli kontrol altında, uyarlamalı kontrole göre daha verimli gerçekleştiği ve senkronizasyon sisteminin oturma süresinin daha kısa olduğu sonucu gözlemlenmiştir.

5. TARTIŞMA VE SONUÇ

Bu çalışmada Lorenz ve Chen kaotik sistemlerine aynı şartlar altında aynı başlangıç koşullarında kayan kipli kontrol ve uyarlamalı kontrol yöntemleri uygulanarak Matlab/Simulink ortamında kaos senkronizasyonu simülasyonu yapıldı. Burada komple(complete) senkronizasyon yöntemi benimsenmiş olup master sistem olarak Lorenz, slave sistem olarak ise Chen kaotik sistemi belirlendi. Ardından gömülü sistemlerden, FPGA kullanılarak senkranizasyon modeli gerçek zamanlı olarak FPGA'ya yüklenip FPGA'nın hardware co-simulation özelliği sayesinde yine Matlab/Simulink ortamında gerçek zamanlı sonuçlar alındı ve hem gerçek zamanlı sonuçlar ile simülasyon sonuçları hem de kayan kipli kontrol ile uyarlamalı kontrol yönteminin sonuçları karşılaştırıldı.

Karthikeyan ve arkadaşları (2015), Lyapunov karalılık teoremine bağlı olarak bir uyarlamalı kontrol şeması ve parametre güncelleme yasası getirerek bilinmeyen parametrelere sahip yeni bir kaotik sistem senkronizasyonu önerdi. Tam (complete) senkronizasyon yaklaşımı kullanılarak kaotik sistem kontrolü için, geri besleme kontrol sistemi tasarlandı. Uygulanan yöntemde, her bir alt sistem Matlab/Simulink bloklarıyla birleştirildi ve daha sonra Matlab/Simulink tasarımları "System Generator" yardımıyla "bitstream" dosyasına dönüştürüldü. Böylece FPGA'nın kullanacağı program uzantısı elde edilmiş oldu. Sonuç olarak elde edilen bitstream dosyası FPGA'ya gönderilerek sistem tasarımı FPGA'ya gömülerek implementation sağlanmış oldu. Böylece kaotik sistemlerin senkronizasyonu, bir FPGA uygulaması ile gerçekleştirildi. Ayrıca sayısal sonuçlar, önerilen yöntemin verimliğini gösterdi. Bu çalışmada ise aynı başlangıç koşulları altında FPGA'nın implementation özelliği yerine hardware co-simulation özelliği kullanıldı ve uyarlamalı kontrol yöntemine ek olarak senkronizasyon sistemine, aktif kayan kipli kontrol da uygulanıp her iki kontrol yönteminin senkronizasyon sistemine olan etkileri incelendi. Uyarlamalı kontrol yöntemi uygulandığında sistemin oturma süresi ve aşma miktarının söz konusu çalışmayla hemen hemen aynı olduğu ve FPGA'nın hardware co-simulation özelliği ile implementation özelliği karşılaştırıldığında sistemde oturma süresi ve aşma miktarı açısından bir değişme

olmadığı görülmüştür. Ancak sisteme, aktif kayan kipli kontrol yöntemi uygulandığında sistemin oturma süresinin ve aşma miktarının daha az olduğu görülmüştür.

Haeri ve Emadzadeh (2005), Lorenz–Chen, Chen–Lu ve Lu–Lorenz gibi 3 farklı kaotik sistemin ikili senkronizasyonlarını gerçekleştirmek için master-slave yapısında bir aktif kayan kipli denetleyici tasarladı. Burada sistem parametreleri biliniyor olup kapalı hata dinamikleri, kontrol parametrelerine ve sistem parametrelerinin kapalı döngü hata dinamiklerine göre değişiyordu. Böylece senkronizasyon durumu bu parametreler sayesinde ayarlanabiliyordu. Önerilen yöntem için kararlılık analizi, Lyapunov kararlılık teoremi baz alınarak yapıldı. Sonuç olarak, önerilen kontrol tekniğinin verimliliği sayısal sonuçlarla sunulmuş oldu. Bu çalışmada ise aktif kayan kipli kontrol yöntemine ek olarak senkronizasyon sistemine, aynı başlangıç koşulları altında uyarlamalı kontrol yöntemi de uygulandı. Ayrıca Matlab/Simulink programı üzerinde FPGA xilinx bloksetleri kullanılarak senkronizasyon sistemi tasarlanarak gerçek zamanlı sonuçlar alınması sağlandı. Uyarlamalı kontrol yöntemi uygulandığında, söz konusu çalışmadaki aktif kayan kipli kontrol yöntemi uygulanan duruma göre sistemin oturma süresi ve aşma miktarının daha fazla görülmüştür.

FPGA'nın dijital yapısı sayesinde gerçek zamanlı sonuçların simülasyon sonuçlarıyla oldukça iyi bir şekilde örtüştüğü gözlemlenmiştir. Ayrıca FPGA'nın tasarım kolaylığı sayesinde kaotik sinyallerin üretilmesi amacıyla FPGA kullanıldığında kolay, hızlı ve güvenilir sonuçlar alındığı gözlemlenmiştir.

Çalışmanın başında amaçlanan bir diğer konu ise kaos senkronizasyonun kontrolü maksadıyla ayrı ayrı kayan kipli kontrol ve uyarlamalı kontrol yöntemlerinin aynı şartlar ve aynı başlangıç koşulları altında seçilen kaotik sistemlere uygulanıp sonuçlarının gözlemlenmesiydi.

Senkronizasyon sistemi birinci durum hataları (e_1) incelendiğinde (Bkz. Şekil 4.12) birinci durum değişkenleri (y_1 ve x_1) arasındaki senkronizasyonun aktif kayan kipli kontrol yöntemiyle, uyarlamalı kontrol yöntemine göre yaklaşık 200 kat daha hızlı bir sürede gerçekleştiğini dolayısıyla oturma süresinin hemen hemen 20 kat daha az olduğunu gösterir. Ayrıca aktif kayan kipli kontrol yöntemi altında, aşma miktarının ve salınımın uyarlamalı kontrol yöntemine göre daha az olduğu görülür.

Aynı şekilde senkronizasyon sistemi ikinci durum hataları (e_2) incelendiğinde (Bkz. Şekil 4.13) ikinci durum değişkenleri (y_2 ve x_2) arasındaki senkronizasyonun aktif kayan kipli kontrol yöntemiyle, uyarlamalı kontrol yöntemine göre yaklaşık 100 kat daha hızlı bir sürede gerçekleştiğini dolayısıyla oturma süresinin hemen hemen 100 kat daha az olduğunu gösterir. Ayrıca aktif kayan kipli kontrol yöntemi altında, aşma miktarının ve salınımın uyarlamalı kontrol yöntemine göre daha az olduğu görülür.

Son olarak senkronizasyon sistemi üçüncü durum hataları (e_3) incelendiğinde (Bkz. Şekil 4.14) birinci durum değişkenleri (y_3 ve x_3) arasındaki senkronizasyonun aktif kayan kipli kontrol yöntemiyle, uyarlamalı kontrol yöntemine göre yaklaşık aynı sürede gerçekleştiğini dolayısıyla oturma süresinin hemen hemen aynı olduğunu gösterir. Ancak aktif kayan kipli kontrol yöntemi altında, aşma miktarının ve salınımın uyarlamalı kontrol yöntemine göre daha az olduğu görülür.

Sonuçlar incelendiğinde, senkronizasyon sistemine kayan kipli kontrol uygulandığında, uyarlamalı kontrol yöntemine göre sistemin çok daha kısa sürede oturduğu ve aşma miktarının daha az olduğu gözlemlenmiştir. Dolayısıyla Lorenz ve Chen sistemlerinin senkronizasyonunda aktif kayan kipli kontrol yönteminin, uyarlamalı kontrol yöntemine göre daha uygun olduğu görülmüştür.

KAYNAKLAR

Agiza, H. N., Yassen, M. T., 2001. Synchronization of Rossler and Chen chaotic dynamical systems using active control. ***Physics Letters A,*** **278**(4): 191-197.

Al-sawalha, M. M., Noorani, M. S. M., 2010. Adaptive reduced-order anti-synchronization of chaotic systems with fully unknown parameters. ***Communications in Nonlinear Science and Numerical Simulation,*** **15**(10): 3022-3034.

Alligood, KT., Sauer, TD., Yorke JA., 1996. ***Chaos: An Introduction to Dynamical Systems.*** Springer, New York.

Arecchi, F. T., Boccaletti, S., Ciofini, M., Meucci, R., Grebogi, C., 1998. The control of chaos: theoretical schemes and experimental realizations. ***International Journal of Bifurcation and Chaos,*** **8**(08): 1643-1655.

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

Chen, G., Ueta, T., 2002. ***Chaos in Circuits and Systems***. World Scientific, Singapore.

Chen, X., Cao, J., Qiu, J., Alsaedi, A., Alsaadi, F. E., 2016. Adaptive control of multiple chaotic systems with unknown parameters in two different synchronization modes. ***Advances in Difference Equations,*** **2016**(1): 231.

Chua, L., 1980. Dynamic nonlinear networks: State-of-the-art. ***IEEE Transactions on Circuits and Systems,*** **27**(11): 1059-1087.

Closson, T. L., Roussel, M. R., 2000. Synchronization by irregular inactivation. ***Physical Review Letters,*** **85**(18): 3974.

Cuomo, K. M., Oppenheim, A. V., 1993. Circuit implementation of synchronized chaos with applications to communications. **Physical Review Letters, 71**(1): 65-68.

Haeri, M., Emadzadeh, A. A., 2007. Synchronizing different chaotic systems using active sliding mode control***. Chaos, Solitons & Fractals,*** **31**(1): 119-129.

Haeri, M., Tavazoei, M. S., Naseh, M. R., 2007. Synchronization of uncertain chaotic systems using active sliding mode control. ***Chaos, Solitons & Fractals,*** **33**(4): 1230-1239.

He, J., Cai, J., & Lin, J., 2016. Synchronization of hyperchaotic systems with multiple unknown parameters and its application in secure communication. ***Optik-International Journal for Light and Electron Optics,*** **127**(5): 2502-2508.

Ho, M. C., Hung, Y. C., 2002. Synchronization of two different systems by using generalized active control. ***Physics Letters A,*** **301**(5-6): 424-428.

Israel, G., 2004. ***Technological Concepts and Mathematical Models in the Evolution of Modern Engineering Systems.*** Birkhäuser, Basel.

Itoh, M., Yang, T., Chua, L. O., 2001. Conditions for impulsive synchronization of chaotic and hyperchaotic systems. ***International Journal of Bifurcation and Chaos,*** **11**(02): 551-560.

Karthikeyan, R., Prasina, A., Babu, R., Raghavendran, S., 2015. FPGA implementation of novel synchronization methodology for a new chaotic system***. Indian Journal of Science and Technology,*** **8**(11).

Kılıçkır, H., 2014. FPGA Nedir. ***http://www.elektrikport.com/teknik-kutuphane/fpga-nedir/14702#ad-image-0.*** Erişim tarihi: 06.03.2018.

Li, D., Zhang, X. P., Hu, Y. T., Yang, Y. Y., 2015. Adaptive impulsive synchronization of fractional order chaotic system with uncertain and unknown parameters**.** ***Neurocomputing,*** **167**: 165-171.

Li, H., Liao, X., Li, C., Li, C., 2011. Chaos control and synchronization via a novel chatter free sliding mode control strategy. ***Neurocomputing,*** **74**(17): 3212-3222.

Li, W. L., Chang, K. M., 2009. Robust synchronization of drive–response chaotic systems via adaptive sliding mode control. ***Chaos, Solitons & Fractals,*** **39**(5): 2086-2092.

Liu, S., Liu, P. 2011. Adaptive anti-synchronization of chaotic complex nonlinear systems with unknown parameters. ***Nonlinear Analysis: Real World Applications,*** **12**(6): 3046-3055.

Lorenz, E. N., 1963. Deterministic nonperiodic flow. ***Journal of the Atmospheric Sciences***, **20**(2): 130-141.

Martens, M., Pécou, E., Tresser, C., Worfolk, P., 2002. On the geometry of master–slave synchronization. ***Chaos: An Interdisciplinary Journal of Nonlinear Science,*** **12**(2): 316-323.

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

Memeşil, A., 2015. FPGA Nedir. ***http://roboturka.com/fpga/fpga-nedir/.*** Erişim tarihi: 11.03.2018.

Murakami, A., Ohtsubo, J., 2001. Chaos synchronization based on a continuous chaos control method in semiconductor lasers with optical feedback. ***Physical Review E,*** **63**(6): 066203.

Muthuswamy, B., Banerjee, S., 2015. ***A Route to Chaos Using FPGAs.*** Springer International Publishing, İsviçre. 219.

Park, J. H., 2005. Adaptive synchronization of a unified chaotic system with an uncertain parameter. ***International Journal of Nonlinear Sciences and Numerical Simulation,*** **6**(2): 201-206.

Park, J. H., 2007. Adaptive modified projective synchronization of a unified chaotic system with an uncertain parameter. ***Chaos, Solitons & Fractals,*** **34**(5): 1552-1559.

Pecora, L. M., Carroll, T. L., 1990. Synchronization in chaotic systems. ***Physical Review Letters,*** **64**: 821- 824.

Pourmahmood, M., Khanmohammadi, S., Alizadeh, G., 2011. Synchronization of two different uncertain chaotic systems with unknown parameters using a robust adaptive sliding mode controller. ***Communications in Nonlinear Science and Numerical Simulation,*** **16**(7): 2853-2868.

Rajagopal, K., Akgul, A., Jafari, S., Karthikeyan, A., Koyuncu, I., 2017. Chaotic chameleon: Dynamic analyses, circuit implementation, FPGA design and fractional-order form with basic analyses. ***Chaos, Solitons & Fractals,*** **103*:*** 476-487.

Rajagopal, K., Karthikeyan, A., Srinivasan, A. K., 2017. FPGA implementation of novel fractional-order chaotic systems with two equilibriums and no equilibrium and its adaptive sliding mode synchronization. ***Nonlinear Dynamics,*** **87**(4): 2281-2304.

Salarieh, H., Shahrokhi, M., 2008. Adaptive synchronization of two different chaotic systems with time varying unknown parameters. ***Chaos, Solitons & Fractals,*** **37**(1): 125-136.

Shi, X. R., Wang, Z. L., 2009. Adaptive added-order anti-synchronization of chaotic systems with fully unknown parameters. ***Applied Mathematics and Computation,*** **215**(5): 1711-1717.

Sprott, J. C., 2010. ***Elegant Chaos: Algebraically Simple Chaotic Flows***. World Scientific, Singapore.

Tang, Z., Park, J. H., Lee, T. H., 2016. Distributed adaptive pinning control for cluster synchronization of nonlinearly coupled Lur'e networks. ***Communications in Nonlinear Science and Numerical Simulation,*** **39**: 7-20.

Tang, Z., Park, J. H., Lee, T. H., Feng, J., 2016. Random adaptive control for cluster synchronization of complex networks with distinct communities**.** ***International Journal of Adaptive Control and Signal Processing,*** **30**(3): 534-549.

Tufan, T., 2017. Dünya Forum: Tanımsızlığın Formülleri/Kaos Teorisi Nedir. ***(https://www.gazeteduvar.com.tr/dunya-forum/2017/09/24/dunya-forum-tanimsizligin-formulleri-kaos-teorisi-nedir/.*** Erişim tarihi: 24.02.2018.

Vaidyanathan, S., Sampath, S., 2011. Global chaos synchronization of hyperchaotic Lorenz systems by sliding mode control. ***Advances in Digital Image Processing and Information Technology***: 156-164.

Van Der Pol, B., 1926. LXXXVIII. On "relaxation-oscillations". ***The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science,*** **2**(11): 978-992.

Van Der Pol, B., Van Der Mark, J., 1927. Frequency demultiplication. ***Nature,*** **120**(3019): 363.

Van Der Pol, B., Van Der Mark, J., 1928. LXXII. The heartbeat considered as a relaxation oscillation, and an electrical model of the heart. ***The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science,*** **6**(38): 763-775.

Varıcı, A., 2017. FPGA nedir. Özellikleri nelerdir. Hangi alanlarda kullanılırlar. ***(http://abdullahvarici.blogspot.com.tr/2017/11/fpga-nedir-ozellikleri-nelerdir hangi.html.*** Erişim tarihi: 06.03.2018.

Yang, T., Chua, L. O., 1997. Impulsive stabilization for control and synchronization of chaotic systems: theory and application to secure communication. ***IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications,*** **44**(10): 976-988.

Yau, H. T., Chen, C. O. K., Chen, C. L., 2000. Sliding mode control of chaotic systems with uncertainties. ***International Journal of Bifurcation and Chaos,*** **10**(05): 1139-1147.

Yin, X., Ren, Y., Shan, X., 2002. Synchronization of discrete spatiotemporal chaos by using variable structure control. ***Chaos, Solitons & Fractals,*** **14**(7): 1077-1082.

Zhang, H., Huang, W., Wang, Z., Chai, T., 2006. Adaptive synchronization between two different chaotic systems with unknown parameters. **Physics Letters A, 350**(5-6): 363-366.

Zhang, H., Ma, X. K. 2006. Active control of uncertain chaotic systems with parametric perturbation. ***International Journal of Modern Physics B,*** **20**(16): 2255-2264.

Zhang, H., Ma, X. K., Liu, W. Z., 2004. Synchronization of chaotic systems with parametric uncertainty using active sliding mode control**.** ***Chaos, Solitons & Fractals,*** **21**(5): 1249-1257.

Zhang, Q., Chen, S., Hu, Y., Wang, C., 2006. Synchronizing the noise-perturbed unified chaotic system by sliding mode control. ***Physica A: Statistical Mechanics and Its Applications,*** **371**(2): 317-324.

Zhao, J., Wu, Y., Liu, Q., 2014. Chaos synchronization between the coupled systems on network with unknown parameters. ***Applied Mathematics and Computation,*** **229**: 254-259.

ÖZ GEÇMİŞ

Onur SİLAHTAR, 1989 yılında Diyarbakır'da doğdu. İlk ve orta öğrenimini Diyarbakır'da tamamladı. Dicle Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü'nden mezun olduktan sonra 2016 yılında Van Yüzüncü Yıl Üniversitesi Mühendislik Fakültesi'nde Elektrik-Elektronik Mühendisliği Bölümü'nde Araştırma Görevlisi olarak göreve başladı. Halen aynı üniversitede Araştırma Görevlisi olarak çalışmaktadır.

T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
LİSANSÜSTÜ TEZ ORİJİNALLİK RAPORU

Tarih: 03/08/2018

Tez Başlığı / Konusu:

FPGA Tabanlı Kaos Senkronizasyonu için Denetleyici Tasarımı

Yukarıda başlığı/konusu belirlenen tez çalışmamın Kapak sayfası, Giriş, Ana bölümler ve Sonuç bölümlerinden oluşan toplam 73 sayfalık kısmına ilişkin, 03/08/2018 tarihinde tez danışmanım tarafından Turnitin intihal tespit programından aşağıda belirtilen filtreleme uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 2 (iki) dir.

Uygulanan filtreler aşağıda verilmiştir:

- Kabul ve onay sayfası hariç,
- Teşekkür hariç,
- İçindekiler hariç,
- Simge ve kısaltmalar hariç,
- Gereç ve yöntemler hariç,
- Kaynakça hariç,
- Alıntılar hariç,
- Tezden çıkan yayınlar hariç,
- 7 kelimeden daha az örtüşme içeren metin kısımları hariç (Limit inatch size to 7 words)

Van Yüzüncü Yıl Üniversitesi Lisansüstü Tez Orijinallik Raporu Alınması ve Kullanılmasına İlişkin Yönergeyi inceledim ve bu yönergede belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini bilgilerinize arz ederim.

03/08/2018

Adı Soyadı: Onur SİLAHTAR

Öğrenci No: 169101019

Anabilim Dalı: Elektrik-Elektronik Mühendisliği Anabilim Dalı

Programı: Elektrik-Elektronik Mühendisliği Bölümü

Statüsü: Y. Lisans ☒ Doktora ☐

DANIŞMAN ONAYI
UYGUNDUR

Dr.Öğr.Üyesi Özkan ATAN

ENSTİTÜ ONAYI
UYGUNDUR

Prof.Dr.Suat ŞENSÖZ
Enstitü Müdürü