T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KAOTİK SİMULASYON LABORATUVARI UYGULAMASI

YÜKSEK LİSANS TEZİ

Elek-Elektronik Müh. Metin VARAN

Enstitü Anabilim Dalı : ELEKTRİK ELEKTRONİK MÜH.

Enstitü Bilim Dalı : ELEKTRİK

Tez Danışmanı : Doç. Dr. Zafer DEMİR

Ocak 2009

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KAOTİK SİMULASYON LABORATUVARI UYGULAMASI

YÜKSEK LİSANS TEZİ

Elek-Elektronik Müh. Metin VARAN

Enstitü Anabilim Dalı : **Elektrik Elektronik Müh**

Enstitü Bilim Dalı : **Elektrik**

Bu tez .. / .. /2006 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Doç.Dr. Zafer DEMİR	Doç.Dr. Cemil Öz	Y.Doç.Dr. Yılmaz UYAROĞLU
Jüri Başkanı	**Üye**	**Üye**

TEŞEKKÜR

Bu tez çalışmamda, çalışma boyunca gerek öngördüğü mühendislik yaklaşım modelleri, gerekse de yakın ilgisi dolayısıyla tez yazmayı bana sevdiren kıymetli hocam Doç. Dr. Zafer DEMİR'e ve Yrd. Doç. Yılmaz UYAROĞLU'na teşekkür ediyorum.

İÇİNDEKİLER

SİMGELER LİSTESİ

α	:	Öteleme değeri
ε	:	Dielektrik sabiti
μ	:	Magnetik geçirgenlik
σ	:	Gibbs salınımı katsayısı
Ω	:	Fourier integrali üst sınırı
ρ	:	Özdirenç
ρ_e	:	Toprak özdirenci
λ	:	Özdeğer
A	:	Kablo hattının alt matrisi
B	:	Kablo hattının alt matrisi
C	:	Kapasitans(F)
c	:	Kablo nüvesi
E_S	:	Generatör uyartım gerilimi
f	:	Frekans (Hz)
f(t)	:	Zaman domeni fonksiyonu
F(w)	:	Frekans domeni fonksiyonu
G	:	Birim uzunluğundaki propagasyon matrisinin karekökü
h	:	Yeraltı kablosunun derinliği
Ic	:	Kablo nüvesinin akımı
Is	:	Kablonun kılıf akımı
Ig	:	Norton eşdeğer akımı
Ix	:	Herhangi bir noktadan geçen akım
ℓ	:	Kablo uzunluğu(m)
n	:	Örnekleme katsayısı
N	:	Tek harmoniklerin maksimum sayısı
P	:	Propagasyon matrisi
Q	:	Elektrik yükü (C)
r	:	Kablo yarıçapı(m)
R	:	Direnç(Ohm)
s	:	Kablo kılıfı
S	:	Kısa devre gücü
s_{ji}	:	i. ve j. kablo merkezleri arasındaki uzaklık(m)
t	:	Zaman(sn)
T_0	:	Gözlem zamanı
Vc	:	Iletken gerilimi
Vi	:	Kablo başından ilerleyen gerilim dalgası
Vr	:	Kablo sonunda yansıyan gerilim dalgası

ŞEKİLLER LİSTESİ

BÖLÜM 1. GİRİŞ

Günümüz bilişim teknolojisinde yaşanan hızlı gelişmeler sonucu, internet üzerinden bankacılık işlemleri, e-devlet uygulamaları, uzaktan eğitim uygulamaları, otomasyon sistemlerinin uzaktan kontrolü, güvenlik sistemleri uygulamaları gibi farklı uygulamaların gerçekleştirilebilmesi mümkün olmaktadır. Bu uygulamaların çoğu toplumsal yaşamda kolaylık, hızlılık, zaman ve işgücünden tasarruf sağlamakta, değerlendirme ve geri besleme almada önemli yeri olan verilerin saklanabilmesi ve işlenebilmesi gibi uygulamalarda esnek ve pratik çözümler sunmaktadır. İnternet teknolojisinin kullanıldığı ve son derece önemli faydaların elde edildiği alanlardan birisi de uzaktan eğitimdir. Önceleri mektupla başlayan uzaktan eğitim uygulamaları zamanla televizyon, radyo, video gibi görsel yöntemleri kullanmaya başlamıştır. Ancak bütün bu yöntemlerin en önemli eksikliği, bilginin kaynaktan alıcıya doğru tek yönlü aktarılmasıdır. Günümüzde ise bahsedilen bu yöntemler güncelliğini yitirmiş, yeni bir öğrenme yöntemi olan e-öğrenme kavramı geliştirilmiştir.

E-öğrenme, internet tabanlı eğitim modelleri için kullanılan genel bir kavramdır. e-öğrenme sistemlerinin, uzaktan eğitimde kullanılan geleneksel yöntemlere göre en önemli üstünlüğü, alıcı ile kaynağın eş zamanlı yada eş zamansız olarak karşılıklı etkileşim içerisinde bulunabilmeleridir. Özellikle son zamanlarda gelişen teknoloji sayesinde, bilgiyi alan ile bilgi kaynağı internet üzerinden gerçek zamanlı olarak esli ve/veya görüntülü konuşma yapabilmekte, algılamada sorun çekilen konular üzerinde tartışabilmektedirler.

Ülkemizde genç nüfusun fazlalığı, bir yandan işsizlik sorunu yaşanırken bir yandan endüstri ve sanayi de istihdam edilebilecek nitelikli işgücüne sahip bireylerin eksikliği karşılaşılan problemlerdir. Bunun en önemli nedenlerinden birisi, özellikle mesleki eğitim kurumlarında güncel teknolojilere sahip olan laboratuvarların azlığı ve mevcutların güncellenememesidir. Bilindiği üzere mesleki ve teknik eğitim,

ülkemizde başlıca Teknik ve Mesleki Eğitim Fakülteleri ile Mühendislik Fakültelerinde verilmektedir. Bu fakülteler; geleceği şekillendirmede, genç beyinleri eğitmede ve yönlendirmede önemli rolü olan ve dolayısıyla ülke geleceğine yüksek katkı sağlayan fakültelerdir. Geleceği şekillendiren bu fakültelerde yapılan eğitimlerin güncel teknolojilerle donatılması, eğitim ve öğretim kalitesinin yükseltilmesi ve en önemlisi laboratuvarlarının yaygın ve güncel teknolojilere sahip olması gerekmektedir. Günümüz teknolojileri incelendiğinde, bu fakültelerde e-öğrenme ortamları ile eğitim öğretime destek sağlanması ve mezunlarının da bu teknolojilerle tanışmaları bir zorunluluk haline gelmiştir[1].

Gerek e-öğrenme ortamlarında gerekse de yüz yüze öğrenmede, öğretilecek konunun kapsamına bağlı olarak kullanılacak ders materyalleri değişiklik gösterir. Özellikle mesleki ve teknik eğitim kapsamında verilecek dersler için materyal seçimi ve uygulama çalışmaları son derece önemlidir. Mesleki ve teknik alanlarda yapılacak olan eğitimde pratik deneyim sağlamanın geleneksel yolu laboratuvar temelli sistemler kullanmaktır. Geleneksel eğitimde uygulamalar, laboratuvarlarda birebir deney yapılması şeklinde gerçekleşmektedir. Bu sistemler deney donanım düzenekleri ve laboratuvar ölçüm setleri gerektirmektedirler. Ancak, fiziki koşulların yetersizliğinden dolayı öğrenciler laboratuvarlardan sınırlı zaman aralığında faydalanmaktadırlar. Ayrıca, emniyet ve güvenlik amaçlı olarak uzman bir kişiye ihtiyaç duyulmaktadır. Günümüz şartları ile bakıldığında, laboratuvarlar çoğunlukla kararlı durumda ölçüm yapabilecek cihazlarla donatılmıştır. Dolayısıyla dinamik değişimlerin izlenebileceği, eş zamanlı örnekleme yapılabilecek ve bu süreçte elde edilen verilerin kaydedilebileceği, her türlü koruma ve kontrol sinyallerinin oluşturulabileceği, bütün bu bilgilerin sunucu bilgisayara transfer edilebileceği bir sisteme ihtiyaç duyulmaktadır[2].

Günümüz bilişim teknolojisi sayesinde, internet üzerinden laboratuvar düzenek ve cihazlarına uzaktan erişim mümkün olmaktadır. Bu teknolojik gelişim, özellikle teknik eğitim ve mühendislik eğitiminde oldukça faydalıdır. Öğrenciler mesleki derslerinin teorik kısımlarını ve laboratuvar deneylerini uzaktan yapabilirler[3].

1.1. Kaotik Simulasyon Laboratuvarı

21. yüzyıl elektrik-elektronik mühendisliği alanında önemli gelişmelerin yaşandığı bir dönem haline gelmiştir. Bu alanda gerçekleşen ilerlemelerden, eğitim-öğretim metotları ve kullanılan materyaller de payını almaktadır. Yaklaşık 20 yıllık bir geçmişi olan internet teknolojisi, günümüz insanının mesleki yaşamından sosyal yaşamına kadar vazgeçemediği bir araç konumuna ulaşmıştır. İnternet teknolojisinin hızlılığı, iş gücü ve zamandan tasarruf sağlaması, değerlendirme ve ölçme konularındaki kolaylaştırıcı etkileri gibi özellikleri eğitim alanında kullanılmasında önemli bir etken olmuştur. Mühendislik eğitiminde pratik deneyim kazanmanın temel yolu, laboratuar sistemlerini kullanmaktır. Ancak üniversitelerimizde bulunan laboratuar alt yapıları ve öğrenci sayısı göz önüne alındığında sıkıntılı durumlarla karşılaşılmaktadır. Bu problemlerin giderilmesi için internet üzerinden erişilebilen laboratuar düzenekleri geliştirilmiştir Bu düzenek ve sanal cihazlara, günün 24 saati, uzaktan erişilerek çalışmalar istenildiği kadar tekrarlanabilmektedir[4].

Kaotik Simulasyon Laboratuvarı tez çalışmasında ,Mühendislik problemlerin çözümünde bugüne kadar kullanılana gelen kaotik yaklaşım modellerini Sayısal ortam aracılığıyla daha etkin ve gözlemlenebilir bir araç olarak kullanılması amaçlanmıştır.

Kaotik Simulasyon Laboratuvarı Uygulamasında,ASP.NET web programlama dili kullanılarak Kaotik Uygulamaların yayımlandığı bir Web sayfası ve ayrıca bu sayfa üzerinden C# .NET nesne yönelimli yazılım dili ve Visual Studio Görsel Araç kutusu kullanılarak geliştirilen , interaktif bir Kaotik Simulasyon Programı da yayımlanarak ziyaretçilerin Kaos Teorisini daha yakından tanıması ve bu konuda akademik dünyada yapılan çalışmaları zengin bir içerik vasıtasıyla pekiştirmesi amaçlanmıştır.

Kaotik Simulasyon Laboratuvarı Uygulamasında, Kaos teorisi ile ilgili tanımlamalar, Kaotik Analiz Metodları, Kaotik Bileşenler Kaos Teorisi Hakkındaki yayınlanan Kitaplar ve Makaleler, Kaotik Denklem modelleri ve Kaotik Uygulama Örnekleri yer almaktadır. Kaotik Simulasyon Laboratuvarı Uygulamasında bulunan

gerek görsel içerikli, gerekse de yazılı zengin içeriklerle Kaos Teorisini ve mühendislik alanlarındaki Uygulamalarının etkin olarak tanıtılması amaçlanmıştır.Belirli başlangıç koşulları altında Chua, Vanderpol, Rossler,Lorenz,Duffing, ve Colpitts gibi Kaos Yöntemleri ile MATLAB® Ortamında yapılan Simulasyonların Eksenel Çıkış Diyagramları alınarak Uygulamanın zengin görsel içeriği hazırlanmıştır.

Bundan başka Simulasyon Uygulamasında ,arka planda MATLAB® Hesaplama Motoru kabiliyetleri kullanılarak, bir mühendislik problemine MATLAB® Hesaplama Motoruna ,Simulasyon Aracı üzerinden ilgili Kaotik Analiz Yöntemi Denklem Takımlarına gerçek zamanlı parameter geçilerek kaotik hesaplama sonuçlarını hem grafiksel hem sayısal büyüklüklerle gösterme ve bu sonuçları analiz etme imkanı vardır[5].

1.2. Kullanılan Araçlar:

a) Web Tabanlı Uygulama Geliştirme Araçları ve ASP.NET Teknolojileri
b) Nesne Yönelimli Programlam Dili ve C# .NET Teknolojileri
c) MATLAB® Hesaplama ve Sayısal Analiz Motoru
d) MATLAB® .NET Builder Araçkutusu

1.2.1. Web tabanlı uygulama geliştirme araçları ve ASP.NET

Günümüzde IP tabanlı programlamanın önemi farkedilmiştir ve bir çok kurum ve kuruluş mevcut uygulamalarını web üzerine geçirme çalışmaları yapmaktadır. Yapılması tasarlanan uygulamaların planlarının temelinde web teknolojileri yatıyor. Bunun başlıca sebeplerinden biri, kullanıcının uygulamayı çalıştırmak için markası, modeli ve üzerinde çalıştırdığı işletim sistemi farketmeden, sadece internete bağlanabilen bir cihaza (bilgisayar,palm,telefon) sahip olması yetiyor[6].

Yazılım geliştirme ve bakım süreçlerinde büyük bir vakit ve iş gücü gerektiren kurulum, yönetim ve güncelleme işlemlerinin merkezileştirilmesini ve kolay

yönetim, bakım ve güncelleştirmeyi sağlamasıda web uygulamalarının tercih edilmesindeki diğer önemli etkenler arasında.

İnternet'e bağlanmak masasütü bilgisayarlarının tekelinden çıkmış durumdadır. Şehir içindeki bir çok restaurant, kafe ve alışveriş merkezinin sağladığı kablosuz Internet erişimi sayesinde dizüstü bilgisayarları ile yüksek hızda bağlantı gerçekleştirmek mümkün. Bu durumda özellikle otomasyon projelerinin web üzerinde hazırlanması ihtiyaç duyulan işlem ve raporların istenilen yerden istenilen cihazla ulaşılabilmesini sağlamaktadır ve "IP tabanlı programla"nın günümüz ve geleceğin teknolojisi olduğu açıkça ortaya çıkmaktadır[7].

Bahsettiğimiz tüm bu olanaklar bir çok donanım ve yazılım firmalarını harekete geçirmiş çok geniş bir ürün yelpazesi oluşturulmuştur. Bu firmalar temelde TCP/IP dediğimiz temel iletişim protokolünü kullansada ürün çeşitliliği, cihazların ve yazılımların kullandıkları özgün teknolojiler ve standartlar farklı ürünlerle iletişimi zorlaştırmaktadır.

Farklı teknolojiler kullanan yazılım ürünlerinin birbiriyle veri alışverişi için XML standartlarını kullanmaktadır.

1.2.1.1. ASP.NET tanımı

ASP.NET, web sayfaları ve web servisleri oluşturmak için Microsoft tarafından .NET vizyonu çerçevesinde geliştirilen teknolojidir. Yeni bir yaklaşım ortaya konarak klasik web uygulamalarının object-oriented paradigmasıyla çalışması sağlandı. Böylelikle web sayfaları ve web sayfalarındaki kontroller birer obje olarak kullanılabilir hale geldi[8].

1.2.1.2. ASP.NET ve .NET framework

Her .NET uygulamasında olduğu gibi Asp.NET sayfaları da çalıştırılırken öncelikle .NET CLR compiler tarafından MSIL'e çevrilmektedir. Daha sonra JIT compiler tarafından bu aradil makine koduna çevrilir ve işlemci tarafından çalıştırılır.

.NET Framework bu safhaları sayfanın her çağrılmasında yapmaya gerek duymaz[9]. Birkere sayfa derlendiğinde daha sonra derlenmiş kod bellekten çalıştırılmaktadır. Bu sayede uygulamalar klasik asp sayfalarına göre daha performanslı çalışır.

1.2.1.3. Geliştirme araçları

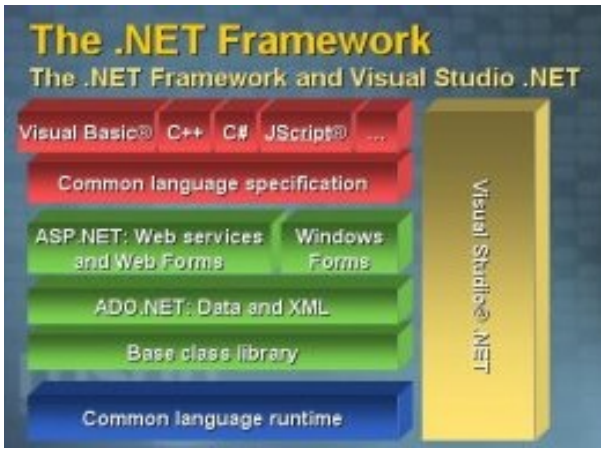
ASP.NET uygulamaları geliştirmek için çeşitli araçlar var. Bunlardan ücretsiz olanlar arasında en yaygın olanı Visual Web Developer 2005 Express Edition ve Visual Web Developer 2008 Express Edition 'dır. Profesyonel anlamda en yaygın kullanılan araç Visual Studio Enterprise 2005 , Visual Studio Enterprise 2008 dir[10].

1.2.2. Nesne yönelimli programlama dili ve C# .NET teknolojileri

.NET platformu, sağlam, ölçeklenebilir ve dağıtılmış uygulamalar oluşturmak için yeni ortam oluşturan ve bu uygulamaları oluştururken bünyesindeki birçok dilden herhangi birini seçme özgürlüğü sunan teknolojilerdir.Bu uygulamaları geliştirmemiz için .NET'in bir parçası olan .NET Framework'ü kullanılır.

.NET Framework .NET tabanlı uygulamaların oluşturulmasını ve çalışmasını sağlayan bir yapıdır. NET Framework bu uygulamaları yaparken birçok dilden faydalanır. Bu diller; temel dil tanımları (CLS- Common Language Spefication) ve ortak tip sistemi (CTS –Common Type System) ile aynı özellikleri taşımak zorundadırlar.

.NET Framework; Common Language Runtime (CLR) ve Framework Sınıf Kütüphanelerinden (Framework Class Library-FCL) oluşur. (FCL bazen Base Class Library-BCL olarak da adlandırılmaktadır)[11].



Şekil 1.1. .NET Framework Yapısı

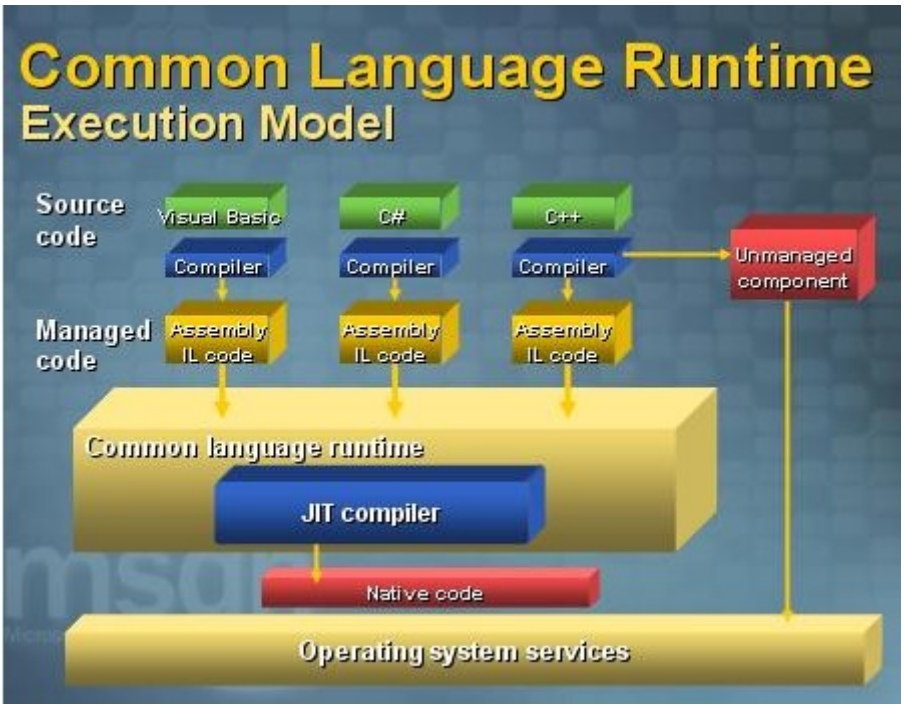
1.2.2.1. Common language runtime

CLS kurallarına uygun farklı dillerde yazılmış uygulamaların birlikte çalışmasını sağlayan bir ortak çalışma zamanı platformudur. C# programlarının taşınabilirliğinin sağlanması için CLR'de çalışmak üzere yaratılırlar. Böylelikle C veya C++ gibi dillerde bir program yaratmaya kalktığınızda bu programı Linux ve Windows makinelerde çalıştırmak isteseydiniz bu programı ayrı ayrı Linux ve Windows makinelerde derlemek zorunda kalmanın önü alınmış olur.

1.2.2.2. .NET framework class library:

FCL, tüm .NET dillerinin ortak sınıf kitaplıklarıdır.Yani dilden bağımsız bir yapıdır. Bu sayede tüm .NET dilleri teorik olarak aynı yeteneklere sahip olmuşlardır. Bu kütüphaneler web form (ASP.NET), ADO.NET,windows form, web servisleri ve consol uygulamaları için gerekli sınıfları içermektedirler.

1.2.2.3. .NET framework çalışma yapısı

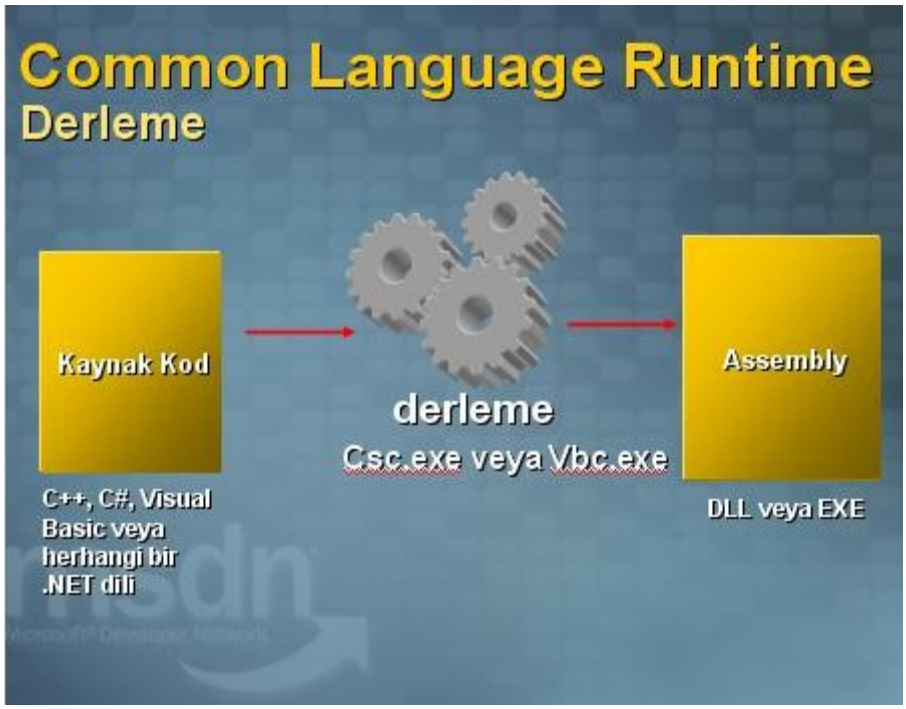


Şekil 1.2. Common Language Runtime Yapısı

C veya C++ gibi dillerde program derlendiği zaman direk makine koduna çevrilmekte böylelikle uygulamanın aynı anda tek bir plaotform ve Mikroişlemci tipi üzerinde çalışması sınırlanmış olmaktaydı. Diğer yandan .NET Platformu üzerinde geliştirilen program kodu derlendiği zaman, kod makine diline değil işlemciden bağımsız bir dil olan IL'ye çevrilir.

1.2.2.4. JIT (just-in-time) derleyicileri

.NET de IL'e çevrilen kod çalıştırılırken JIT derleyicileri devreye girer. Bu derleyiciler IL kodunu bilgisayarın anlayabileceği dile yani makine diline (Native Code) dönüştürürler(Bakınız Şekil 1.2.).



Şekil 1.3. Common Language Runtime Derleyici Yapısı

Şekil 1.3 de görülen Assembly'den kasıt bir programlama dili olan Assembly değildir. Şekil 1.4'den de anlaşılacağı gibi Assembly .NET platformundaki herhangi bir dili kullanarak yazdığımız kodların derlenmesi sonucunda oluşan DLL veya EXE uzantılı dosyalardır.IL dosyalarının temelini bu dosyalar oluşturmaktadır.Bu dosyaların içerisinde programa ait bilgileri barındıran ve Metadata olarak adlandırılan veriler oluşmaktadır.Metadataların amacı CLR'nin çalışma zamanında hangi türlerin ayrılacağını ve hangi metotların çağrılacağını bilmesine olanak sağlamaktır. Böylece CLR daha daha verimli çalışır.Metadata'nın bu şekilde sorgulanmasına yansıma (reflection) adı verilir.Buna Visual Studio .NET ,IntelliSense özelliğini uygulamak için yansıma yöntemlerinin kullanılması örnek verilebilir[12].

1.2.3. MATLAB® hesaplama ve sayısal analiz motoru

MATLAB®®, temel olarak nümerik hesaplama, grafiksel veri gösterimi ve programlamayı içeren teknik ve bilimsel hesaplamalar için yazılmış yüksek performansa sahip bir yazılımdır. MATLAB® programının tipik kullanım alanları:

a) Matematik ve hesaplama işlemleri
b) Algoritma geliştirme
c) Modelleme, simülasyon (benzetim) ve öntipleme
d) Veri analizi ve görsel efektlerle destekli gösterim
e) Bilimsel ve mühendislik grafikleri
f) Uygulama Geliştirme

şeklinde özetlenebilir.

MATLAB® adı, *MAT*rix LABoratory (Matrix Laboratuarı) kelimelerinden gelir. MATLAB®, ilk olarak Fortran Linpack ve Eispack projeleriyle geliştirilen ve bu programlara daha etkin ve kolay erişim sağlamak amacıyla 1970'lerin sonlarında yazılmıştır. İlk başlarda bilim adamlarına problemlerin çözümüne matris temelli teknikleri kullanarak yardımcı olmaktaydı. Bugün ise geliştirilen yerleşik kütüphanesi ve uygulama ve programlama özellikleri ile gerek üniversite ortamlarında (başta matematik ve mühendislik olmak üzere tüm bilim dallarında) gerekse sanayi çevresinde yüksek verimli araştırma, geliştirme ve analiz aracı olarak yaygın bir kullanım alanı bulmuştur. Ayrıca işaret işleme, kontrol, fuzzy, sinir ağları, wavelet analiz gibi bir çok alanda ortaya koyduğu Toolbox adı verilen yardımcı alt programlarla da özelleştirilmiş ve kolaylaştırılmış imkanlar sağlamış ve sağlamaya da devam etmektedir.

MATLAB®, temel olarak 5 ana kısımdan oluşur:

1. Ortam Geliştirme (MATLAB® masaüstü ve Komut penceresi, Komut Geçmişi, Yardım, ÇalışmaAlanı, Dosyalar vb).

2. Matematiksel Fonksiyon Kütüphanesi (Standart sapma, sinüs, cosinüs ve complex işlemlerden matris tersi alma, Bessel fonksiyonlar ve Fast Fourier Dönüşümlere kadar bir çok hesapsal algoritmalar).
3. MATLAB® Dili (Akış şemaları, çevrimler, giriş-çıkış dizgeleri, veri yapıları ve nesne yönelimli programlama vb).
4. Handle Grafik Sistemi (iki ve üç boyutlu grafikler, görüntü işleme, animasyonlar vb).
5. MATLAB® Uygulama Programı Arabirimi-API (C ve Fortran dillerine uyarlanabilen programlar yazmayı ve MATLAB® alanından iş programlarını çağırmayı sağlayan bir kutüphanedir).
6. MATLAB®'ın yardımcı programlarından olan Simulink, en çok lineer olmayan dinamik sistemlerin simülasyonu için interaktif bir ortamdır. Ekran üzerinde blok diyagramlarıyla bir sistemi çizip modelleyebilir ve fare ile kontrol edebilirsiniz. Simulink, lineer, lineer olmayan, sürekli veya ayrık zamanlı sistemler ve çok değişkenli sistemlerle uyumlu çalışır.
7. Blocksets, haberleşme, sinyal işleme ve güç sistemleri gibi özel uygulamalar için ilave blok kütüphaneleri eklemeyi sağlar.
8. Real-time Workshop ise blok diyagramlarınızdan C kodu üretmenizi ve bunun çeşitli gerçek-zaman sistemlerinde çalışmasını sağlar.

MATLAB® diğer programlama dillerine göre kullanımı çok daha kolaydır. Derlemeye (compiler) ihtiyaç duymadan sadece yorumlanan (interpeted) bir dildir. Bu özelliği MATLAB®'e büyük bir esneklik ve bilgisayar çalışma ortamından bağımsızlık getirmiştir. Günümüzde MATLAB®'in Windows, Unix, Linux ve Macintosh gibi tüm bilgisayar ortamları ve işletim sistemleri için ticari versiyonları mevcuttur. Hatta kısıtlı kullanım imkanları çerçevesinde daha ucuza satılan öğrenci versiyonları (student edition) da vardır. Önceden tanımlanmış yüzlerce hazır fonksiyonlar ile alt programlar yazmadan işlem hızını arttırmıştır. Veri gösterim ve grafik özellikleri bakımından eşşiz özellikleri vardır. Ayrıca ayrı bir derleyici ile MATLAB® programlarınızı diğer (özellikle üçüncü parti) kullanıcılara satmak ve dağıtmak için çalıştırılabilir kodlara (*.exe) dönüştürmek de mümkündür.

MATLAB® diğer programlama dillerine göre kullanımı çok daha kolaydır. Derlemeye (compiler) ihtiyaç duymadan sadece yorumlanan (interpeted) bir dildir. Bu özelliği MATLAB®'e büyük bir esneklik ve bilgisayar çalışma ortamından bağımsızlık getirmiştir. Günümüzde MATLAB®'in Windows, Unix, Linux ve Macintosh gibi tüm bilgisayar ortamları ve işletim sistemleri için ticari versiyonları mevcuttur. Hatta kısıtlı kullanım imkanları çerçevesinde daha ucuza satılan öğrenci versiyonları (student edition) da vardır. Önceden tanımlanmış yüzlerce hazır fonksiyonlar ile alt programlar yazmadan işlem hızını arttırmıştır. Veri gösterim ve grafik özellikleri bakımından eşşiz özellikleri vardır. Ayrıca ayrı bir derleyici ile MATLAB® programlarınızı diğer (özellikle üçüncü parti) kullanıcılara satmak ve dağıtmak için çalıştırılabilir kodlara (*.exe) dönüştürmek de mümkündür.

MATLAB®, hem MathCAD, Mathematica ve Maple gibi matematik temelli işlemler de kullanılan paket programlar gibi hem de C, Visual Basic ve Fortran gibi programlama dilleri gibi işlev görür.

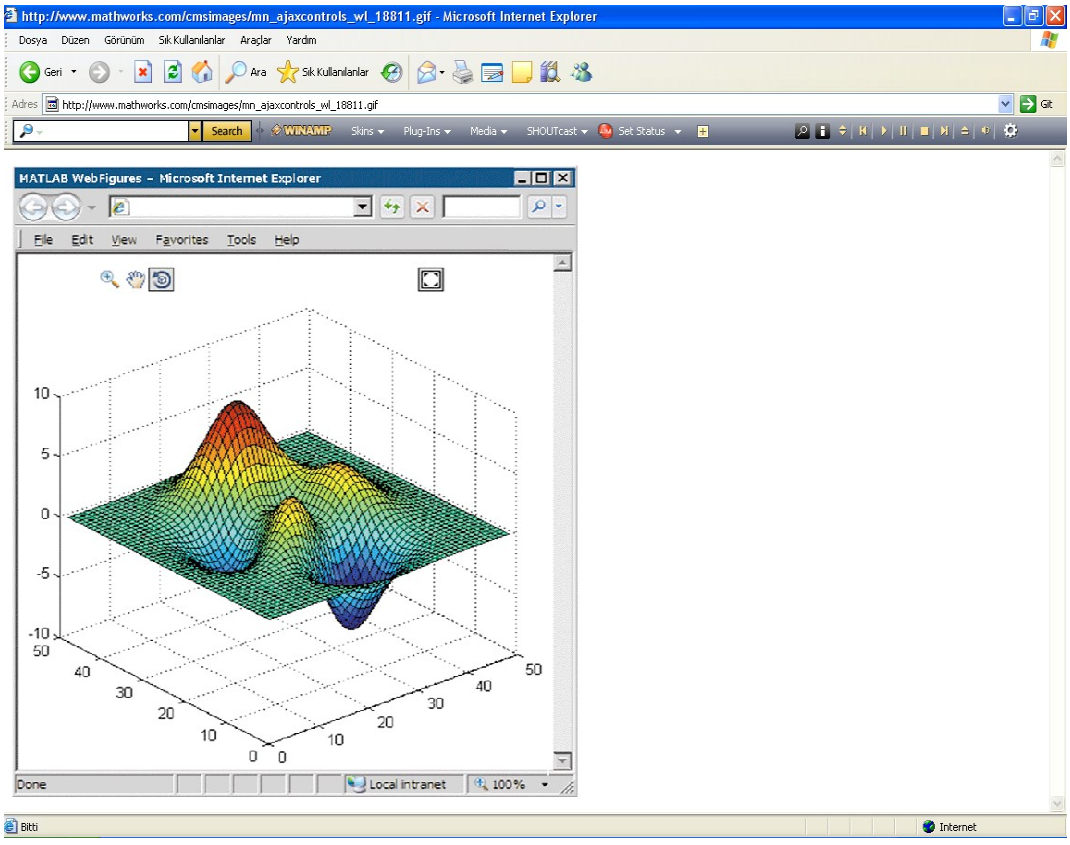
MATLAB®, gerek internet üzerinden (online documentation) gerekse programın kendi içinde bulunan yardım (help) fonksiyonları ile de kullanıcıya çok büyük imkanlar sunmaktadır.

MATLAB® sürümlerine kısaca değinirsek, MATLAB®'in ilk versiyonları Dos tabanlı olup 4.0 sürümü ile Windows ortamına geçmiştir. 5.0 sürümü ile birlikte yerleşik yani tümleşik bir Editör/Debugger (Düzenleyici/Hata Tarayıcı) eklentisi gelmiştir. Bu yenilik ile daha önce dış ortamlarda yazılan m-dosyaları artık MATLAB® içerisinden yazılabilmektedir. Ayrıca yine 5.0 sürümü ile birlikte çekirdek programa (ana MATLAB® programı) ek olarak gelen programlar bir CD içerisinde toplanmış ve örneğin MATLAB® 5.0 sürümü Release 10 olarak adlandırılmıştır. Bugün itibariyle MATLAB® 7.0 versiyonu yani Release 14 mevcuttur. Daha ucuza satılan öğrenci sürümlerinin en önemli kısıtlaması, matris işlem sayısında yapılan sınırlamadır. MATLAB® dili tamamen ingilizce olup henüz Türkçe versiyonu yoktur.

1.2.4. MATLAB® .NET builder

MATLAB® .NET BUILDER, MATLAB® tabanlı .NET ve COM bileşenleri oluşturarak , bu bileşenlerin doğrudan Masaüstü ve Web Ortamlarında servis edilebilmesini sağlamak için geliştirilen bir araçtır.

MATLAB® .NET BUILDER tarafından .NET ve COM nesnesi haline getirilen bileşenleri devam eden yazılım projelerine referans göstererek uygulamalarınızı çalışma zamanında derleyebilirsiniz.Böylelikle hem .NET ortamının geniş kontrol esnekliği , hem MATLAB® in güçlü hesaplama ve analiz yetenekleri tek çatı altına toplanmıştır.



Şekil 1.4. MATLAB® .NET BUILDER Araç Kutusu Kullanılarak Yapılan Bir Simulasyon Çıktısı

MATLAB® .NET BUILDER ile oluşturulan bu bileşenler arkaplanda MATLAB® Çalışma Zamanı Derleyicisi tarafından bütün MATLAB® kütüphaneleri tarafından desteklenen bir yorumlamadan geçerek kullanıcıya MATLAB® hesaplama esnekliği sağlar. Tüm MATLAB® kütüphane desteği sağlamak için Uygulama

yazılımının koşmuş olduğu sunucu üzerine MATLAB® Çalışma Zamanı Derleyicisinin(MCR) kurulu olması gerekmektedir.

Web uygulamaları için , MATLAB® .NET BUILDER , MATLAB® çizimleri için , Ajax- tabanlı büyütme, kesme ve döndürme kontrolleri sağlar. Bundan başka hesaplama sonuçları için elde edilen sonuçları MATLAB® veri tipinden , .NET ve COM veri tipine , veri kaybı olmaksızın dönüşüm sağlar.

MATLAB® ortamında –m file şeklinde yazılan uygulamların .NET ve COM bileşenleri haline getirilmesi , arkaplanda MATLAB® Deployment Tool aracılığıyla gerçekleştirilmektedir. Deployment Tool , MATLAB® .NET BUILDER ın iki katman arasında dönüşüm yapmak için kullanıcıya sunduğu görsel bir ara katmandır.

Bu ara katman kullanılarak MATLAB® ortamında geliştirilen –m file uzantılı çalışmalar, bir Deployment Projesi içine C, C++ ve C# ile geliştrilen projelerde olduğu gibi sınıf ve bileşen ekleme yapılır gibi projeye eklenip COM ve .NET nesnesi haline getirilen bileşenlerin hangi MATLAB® fonksiyonunu kullanacağı şeklinde özelleştirilir.

Deployment Tool katmanı bundan başka, MATLAB® Derleyici ayarlarının manuel kontrolüne de imkan sağlamaktadır, böylelikle derleme yapılırken .NET ortamında kullanılan Proje spesifikasyonlarına uygun ayarlı , opsiyonel derleme imkanı oluşturulmuş olur. Böylelikle .NET de, .NET ve COM bileşeni haline getirilmiş MATLAB® içerikli fonksiyonların çağrılması esnasında , .NET Platformunun Ortak Çalışma Zamanında (CLR) gereksiz işlemci yüklemelerinin önü alınmış olur.

Sonuç olarak MATLAB® .NET BUILDER desteğiyle, MATLAB® uygulamalarının getirmiş olduğu geniş bilimsel hesaplama ve analiz spektrumu kişiye veya şirkete özel uygulamalara uygulanabilmesinin yolu açılmıştır.

BÖLÜM 2. KAOS VE KAOTİK SİSTEMLER

2.1. Kaos'un Tanımı

Kaos kısaca düzensizliğin düzeni şeklinde tanımlanan doğrusal olmayan bir bilim dalıdır. Kaosun ve kaotik işaretlerin en temel özelliği başlangıç şartlarına aşırı duyarlı olmalarıdır. Ayrıca gürültü benzeri geniş güç spektrumuna sahip olmaları yüzünden uzun bir zaman ses ile karıştırılmıştır.

Gerçek hayatta fiziksel sistemlerin çoğu, sistem değişikliklerinin belli bir bölgedeki değişimi için doğrusal davranış gösterir. Ancak bu değişkenlerin doğrusal bölgenin dışındaki değişimi, sistemin doğrusal olmayan davranış göstermesine neden olur.Kaosun ve kaotik işaretlerin başlıca özellikleri; başlangıç şartlarına hassas bağlılık ve gürültü benzeri geniş güç spektrumuna sahip olmalarıdır[13].

Kaos, yıldırımlı fırtınaları, köpüren nehirleri, kasırgaları, sivri dağ zirvelerini, girintili çıkıntılı kıyı boylarını ve nehir deltalarından vücudumuzdaki sinirlerle kan damarlarına kadar her tür karmaşık biçim düzenlerini meydana getiren hareketleri anlamaya yönelik bir bilim dalıdır[14]. Yani Kaos bilimi, gizli biçim düzenleri, ince farklar, nesnelerin "duyarlılığı" ve tahmin edilemeyenin yeniye nasıl yol açtığına dair "kurallar" üzerine odaklanır.

Kaos, düzenli bir hale erişen yada kendini durmadan tekrarlayan bir davranış biçimidir. Faz uzayında dinamik bir sisteme ait bütün bilgilerin zaman içinde belirli bir andaki durumu tek bir noktaya indirgenmektedir. Bu nokta, tam o andaki dinamik sistemin kendisidir. Buna karşılık, bu anı takip eden bir sonraki durumda sistem çok hafifte olsa değişecek ve nokta yerinden oynayacaktır. Tuhaf çekici, modern bilimin en önemli buluşlarından biri olan faz uzayında meydana gelmektedir.

Doğrusal olmayan sistem teorilerindeki ilerleme, yeni deneysel teknikler, pahalı ve işlem gücü yüksek bilgisayarların ucuzlayıp yaygınlaşması, karmaşık ve doğrusal olmayan davranışları daha iyi analiz etmeye ve anlamaya sebep olmuş ve sonuç olarak Kaos Bilimi gelişmiştir. Kaos ve karmaşıklıkla ilgili gözlemlere paralel olarak, bu olayın mekanizmasının anlaşılması, kaotik davranışın nitelendirilmesi, özelliklerinin belirlenmesi, deneysel verilerin ölçülmesi ve analizinin yapılması ile ilgili araştırmalarda çok hızlı gelişmeler kaydedilmiştir.

Kaos ve kaotik davranış, Newton'un 1600'lerde bilime kazandırdığı teoremlere dayanır. Aslında uzay bilimci Kepler'in güneş-dünya-ay yörüngelerinin birbirine göre değişimine yönelik çalışması ve matematikçi Poincare' nin 1800'ün sonlarına doğru solar sistemin kararlılığını sorgulayan çalışması bilimsel olarak isim verilmeden kaosun varlığının sezildiğini göstermektedir. Yirminci yüzyılın sonuna doğru bilgisayarların yüksek performansından dolayı, dinamik sistemlerin genel davranışlarının incelenmesi ile önem kazanmıştır. Teorik alandaki çalışmalar Van de Pol, Andronov, Littlewood, Cartwright, Levinson ve Smale gibi kişiler tarafından yürütülmüş, bu çalışmaları temel alan Brikhoff, Arnold ve Moser, Poincere'nin düşüncelerini daha anlamlı ve derinlikli olarak geliştirmişlerdir.

Yukarıda anlatılan tarihi gelişim detaylı olarak incelendiğinde özellikle ilk zamanlarda araştırmacılar kaosun birçok sistemde varlığı veya var olabileceği olgusunu, bu davranışı anlamak yerine ya görmezlikten gelerek ya da doğa üstü güçlere terk etmek şeklinde yorumlamışlardır. Hatta laboratuar ortamındaki mühendislik sistemlerinde dahi kaosu gördüklerinde bunun, sistemi dışardan etkileyen faktörlerin sonucu olduğunu düşünmüşlerdir. Ancak Lorenz ve May'in çalışmaları sonrasında kaotik dinamik çalışmalarında birçok önemli sıçramalar olmuştur. Lorenz 1963 de 3 tane doğrusal olmayan birinci dereceden adi diferansiyel denklemi bulmuştur. Denklemler oldukça basit olmalarına rağmen elde edilen davranışlar şaşırtacak derecede karmaşıktır.

Kaosu daha iyi anlayabilmek ve fizik kanunlarını kullanarak, kuramsal ve uzun süreli tahminler yapabilmek için konuyu altı alt başlık halinde incelemek gerekir.Bunun için; determinizm felsefesi, başlangıç koşulları, ölçümlerin kesinsizliği, dinamik

kararsızlıklar, kaosun görünümleri, flaktarlar başlıkları altında fizikçilerin kaos ve doğrusal olmayan sistemlerine bakış açısını irdeleyelim[15].

2.1.1. Determinizm felsefesi

Determinizm, her olay veya hareketin, geçmişteki olay veya hareketlerin kaçınılmaz bir sonucu olduğu yönündeki felsefi inanıştır. Deterministlik bilim modeline göre evren, önceden belirlenmiş kurallardan hiç bir sapma ve en küçük bir rasgelelik göstermeden, mükemmel bir makinenin işlemesi gibi zaman içinde kendini gerçekleştirmekteydi.

Determinizmin modern bilimin merkezine yerleştirilmesinde en büyük pay sahibi olan kişi, yaklaşık 300 yıl önce İngiltere'de yaşamış olan Isaac Newton'dur. Newton, sadece bir kaç cümle ile ifade edilebilecek özet ilkeler bularak, bunların şaşırtıcı derecede çeşitli sistemlerin hareketlerini büyük bir kesinlikle öngörebileceklerini gösterdi[16]. Bu üç hareket yasasının mantık süreci ile birleştirildiği takdirde, diğer bir çok şeyin yanı sıra, gezegenlerin güneş etrafındaki yörüngelerinin, fırlatılan nesnelerin dünya üzerideki seyir güzergahlarının ve gel-gitlerin aylık veya yıllık döngülerinin doğru bir biçimde öngörülmesinde kullanılabileceğini ortaya koydu. Newton'un üç adet hareket yasası o denli başarılıydı ki, buluşundan yüzlerce yıl sonra bile fizik bilimi büyük bir oranda, bu yasaların neredeyse tüm tasavvur edilebilir fiziksel sistemlerin hareketlerini açıklamakta nasıl kullanılabileceğini göstermekten ibaret olmuştur. Newton'un yasaları 1900'lerde yerlerini daha geniş bir fizik yasaları dizgesine bırakmış olsa da, determinizm bu gün halen fizik biliminin merkezi felsefesi ve amacı durumundadır.

2.1.2. Başlangıç koşulları

İster güneş sistemi, ister dünya üzerinde düşmekte olan bir nesne veya isterse okyanus akıntıları olsun, herhangi bir sistem için uygun olan ölçümleri ifade ederken bir başlangıç zamanındaki ölçüm değerleri, o sistem için "başlangıç koşulları" olarak adlandırılır. Dinamik yasalar olarak Newton yasaları, herhangi bir sistem için aynı başlangıç koşullarının her zaman aynı sonuçları ortaya çıkaracağını söylediği için, deterministtirler.Evrenin Newton'cu modeli genellikle, sonuçların başlangıç

koşullarından önceden belirlenmiş bir şekilde, adeta zamanda ileri veya geri doğru oynatılabilen bir film gibi, matematiksel olarak zamanla ortaya çıktığı bir bilardo oyunu şeklinde tasarlanır. Bilardo oyunu örneği, mikroskobik düzeyde moleküllerin hareketlerinin bilardo masasındaki topların çarpışmalarına benzetilebileceği ve her iki durumda da aynı dinamik yasalarının geçerli olduğu göz önüne alındığında yararlı bir benzetmedir.

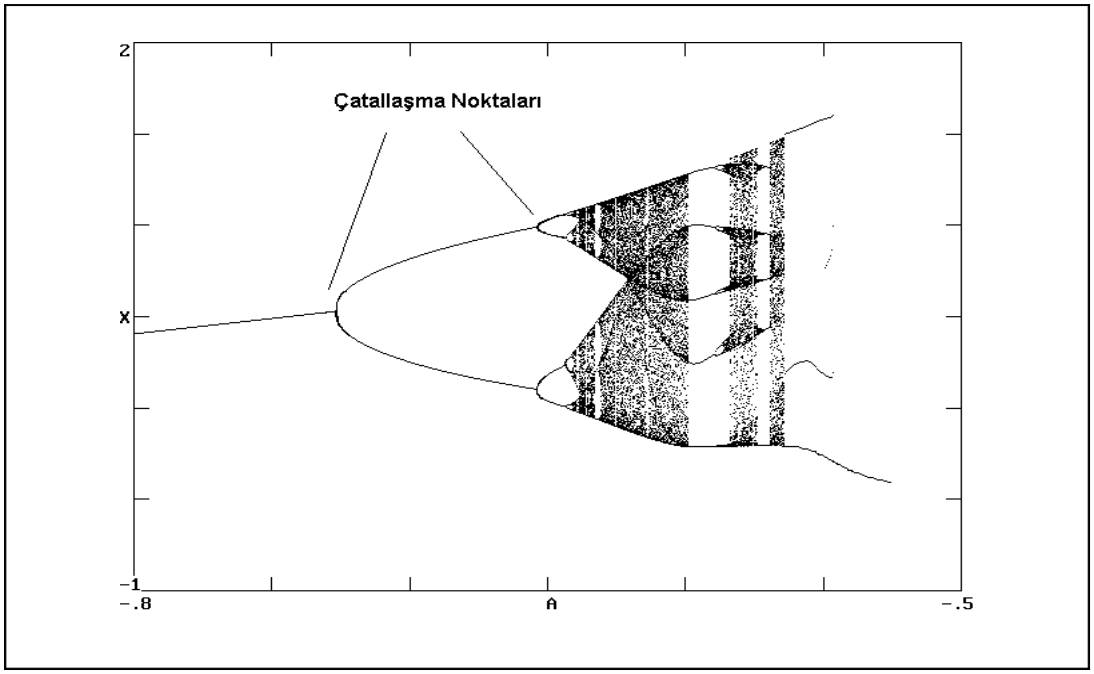
2.1.3. Ölçümlerin kesinsizliği

Deneysel bilimde kaosun incelenmesindeki temel ilkelerinden bir tanesi de, gerçek bir ölçümün hiçbir zaman sonsuz derecede kesin olmayacağı, bir derece kesinlik sizlik içeren bir değer olması gerektiği ilkesidir. Dinamik bilimi açısından, her gerçek ölçümde bir kesinsizlik bulunması, bir sistem üzerinde çalışılırken başlangıç koşullarının sonsuz duyarlılıkta belirlenemeyeceği anlamına gelir. Newton yasaları kullanılarak yapılan hareket çalışmalarında bir sistemin başlangıç koşullarındaki kesinsizlik küçük de olsa daha sonraki veya önceki bir zamanı tahmin etme sürecinde buna karşılık gelen bir kesinsizliğin ortaya çıkmasına neden olur. Fiziğim modern tarihinin büyük bir kısmı boyunca başlangıç koşulların gittikçe daha duyarlı bir biçimde ölçülebilmesi durumunda nihai dinamik tahminlerdeki kesinsizliğin küçültülebileceği kabul edilmiştir.

2.1.4. Dinamik kararsızlıklar ve çatallaşma

Dinamik kararsızlık ve çatallaşma bazı fiziksel sistemlerde gözlenen zamana bağlı özel bir davranış biçimidir. 1900 yılında fizikçi Henri Poincaré tarafından keşfedilmiştir. Poincaré güneşin etrafındaki gezegenlerin hareketleri ile ilgili matematiksel denklemlerle ilgilenen bir fizikçiydi. Poincaré bir sistemin karakteristiğini etkileyen kontrol parametrelerinin değişiminin belirli bir noktadan sonra sistemi kararsızlığa iteceğini saptamıştır[17]. Sistemin durum değişkenleri ile kontrol parametreleri arasındaki ilişki durum-kontrol uzayı olarak adlandırılır. Özellikle güç sistemlerinde çatallaşma yüksek oranda görünür.Bunun sebebi çekilen reaktif güç arttıkça sistemin dinamiği bozulmaya başlar ve sistem osilasyona kayarak çatallaşmaya başlar. Sistemin durum uzayındaki değişkenleri çizdirilirse kritik denge noktasından sonra eğrinin iki'ye ayrıldığı görülür. Bu noktaya "çatallaşma noktası" ,

eğriye de "çatallaşma eğrisi" denir (Şekil 2.1). Çatallaşmayı değişik geri besleme teknikleri ile engellemek mümkündür.



Şekil 2.1 Çatallaşma eğrisi

2.1.5. Kaosun görünümleri

İlk dört bölümde, kaotik bir sistem için fizik kanunları kullanılarak, kuramsal olarak bile olsa, uzun vadeli tahminler yapmanın imkansız olduğunu gördük. Bir derece doğruluğa sahip bir uzun vadeli tahminde bulunabilmek için, başlangıç koşullarının sonsuz bir doğrulukta bilinmesi gerekiyordu. İlk keşfedildiğinde kaotik hareket olayı, matematiksel bir gariplik olarak değerlendirilmişti. O günden bu güne geçen yıllar içinde fizikçiler, kaotik hareketin çok daha yaygın olduğunu, hatta belki de evrendeki temel ilkelerden biri olduğunu keşfetmeye başladılar. En önemli keşiflerden biri, 1963 yılında, havanın basitleştirilmiş bir modelini çalışmak üzere basit bir matematiksel bilgisayar programı yazan meteorolog Edward Lorenz tarafından yapıldı. Yine Lorenz'in atmosfer modelinde kullandığı matematik 1970'lerde geniş bir biçimde araştırıldı. Zamanla, kaotik bir sistemin temel özelliği olarak, iki farklı başlangıç koşulları dizgesindeki düşünülebilecek en küçük farklılığın, daima, sonraki veya önceki zamanlarda büyük farklılıklara yol açacağı, bilinen bir gerçek haline geldi.

Fiziksel olayların çoğu sistemin bileşenlerinin belli bölgelerdeki değişimleri için doğrusal bir davranış gösterir. Bu bölgelerin dışı için sistem doğrusal olmayan davranış sergiler. Genel olarak bir sistemin matematiksel durum denklemini yazarsak;

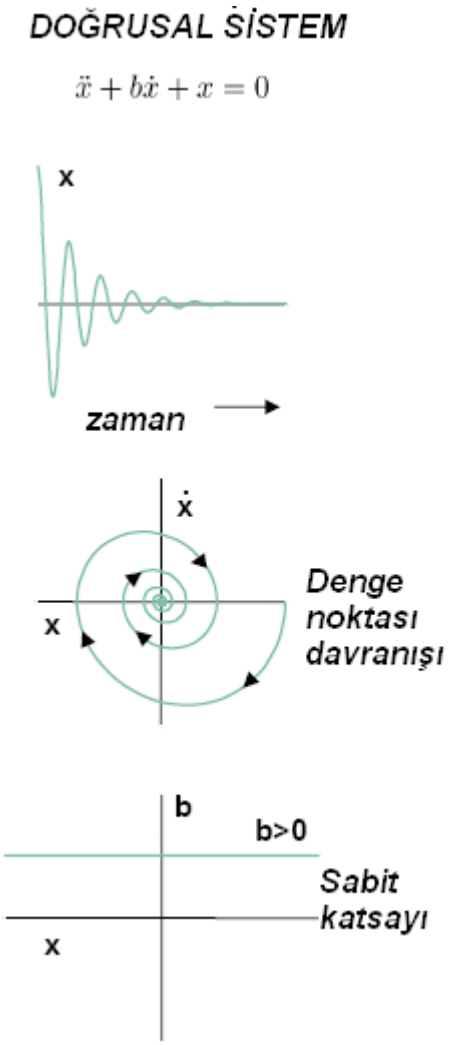
$$X_i = f_i(X_1, X_2, ,,,,,,,,,X_N) \quad X(0) = X_0 \quad i = 1 , 2, ,,,,,,,,,N$$

Şeklinde tanımlanan fonksiyonda eğer fi fonksiyonlarının hepsi xi bileşenlerine göre doğrusal ise; Bu sistem doğrusal olur ve durum denklemleri matris formunda basit bir şekilde yazılır. Bu durumda sistemin cevabı kalıcı ve dengeli bir özellik gösterir. Matematiksel denklemi yine yukarıda ki gibi tanımlanan fonksiyon, Xi bileşenleri doğrusal olmayan karakteristik özelliği gösterirse sistem doğrusal olmayan sistem olur. Sistemin doğrusal olmayan bir yapıda olması durum denklemlerinin matris formunda ifade edilmez.

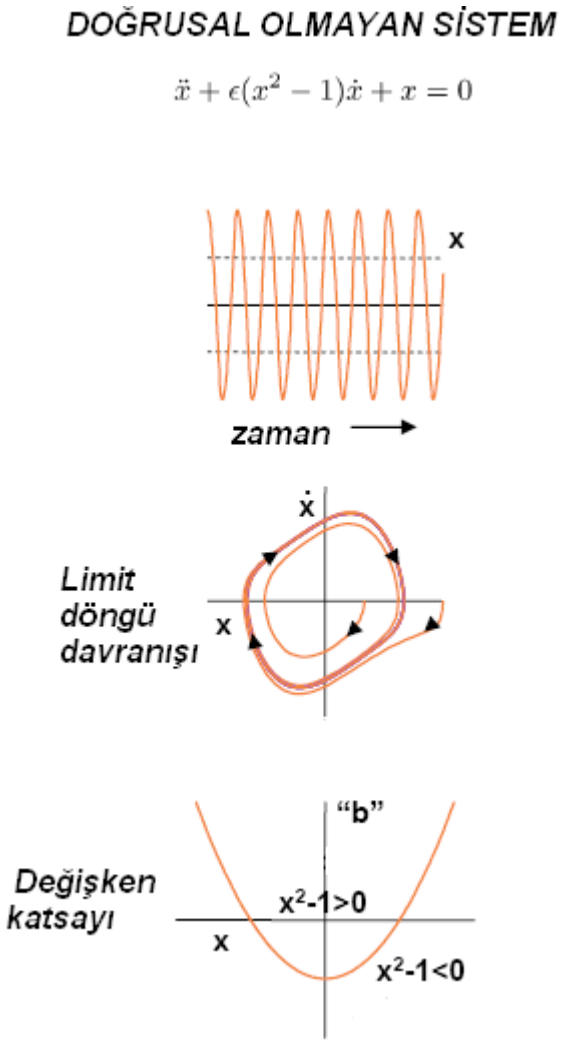
Doğrusal olmayan sistemlerin dinamik davranışlarının incelenmesi için çeşitli metodlar bulunmaktadır. Bunlardan bazıları; doğrusallaştırma tekniği, sinusodial tanımlama fonksiyonu, Lyapunov'un 2. kriteri, Popov metodu' dur. Ancak bu teknikler, sadece lokal davranışları göz önüne aldığı için veya sadece sistemin kararlılığını incelediği için sistemin global davranışlarını elde etmede yetersiz kalmaktadır.

Doğrusal olmayan davranışta olan sistemin durum denklemlerini çeşitli yöntemlerle çıkarılmaktadır. Fakat bu yöntemlerinde açıklayamadığı doğrusal olmayan davranış türleri de vardır. Bunlardan herhangi sisteme bir giriş verilmeden elde edilen türüne kaotik davranış denir. Kaotik davranışın denge noktası şartından farklılıklarını şöyle sırlayabiliriz:

a) Rasgele değil de gerekirci bir yapıya sahip olması,
b) Başlangıç şartına olan hassasiyeti,
c) Sınırsız sayıda değişik periyodik salınımlar içermesi,
d) Genliği ve frekansı tespit edilemeyen, ancak sınırlı bir alanda değişen işaretler içermesi
e) Gürültü benzeri güç spektrumlarına sahip olması gibi sınırlanır.



Şekil 2.2. Doğrusal sistem ve denge davranışı



Şekil 2.3. Doğrusal olmayan sistem ve limit noktası döngü davranışı

Yirminci yüzyılın başlarında kaos hep arka planlarda kalmış, yapılan laboratuar çalışmalarında araştırmacılar kaosun var olduğu ve var olabileceği olgusunu, bu davranışı anlamak yerine görmezlikten gelerek ya da doğaüstü güçlerle bağdaştırarak çıkan sonuçları dikkate almamışlardır. Günümüzde özellikle E. LORENZ ve R.MAY in çalışmaları sonucunda kaos olgusu yüksek performanslı bilgisayarlarında yardımı ile çok büyük gelişmeler olmuştur.

1983 yılına Prof. Dr. L.O. Chua bilim tarihinde kendi anılan bir devre tasarlayarak kaotik osilatör yapmıştır. Bu araştırmalar ile elektronik yeni bir boyut kazanmıştır.

Keza bununla birlikte birçok bilim kaosun getirdiklerini kullanmıştır. Öyle ki kaos yöntemleri ile galaksinin oluşumundan hücre yapılarının tanımlanması bilginin iletiminden, hava ve deprem olaylarına kadar bir çok alanda yararlanabilir popüler bir bilim dalı haline gelmiştir. Bu bağlamda kaos ve kaotik işaretlerinin bulgularını kendi alanlarında yeni çözümler arayışında olan çalışmaları şu şekilde sıralayabiliriz:

a) Kaotik Haberleşme
b) Doğrusal olmayan Sistemlerin Modellenmesi,
c) Doğrusal olmayan Filtreleme,
d) Dinamik Bilgi Sıkıştırma ve Kodlama,
e) Hassas Desen Tanıma,
f) Kaotik Salınımların Yapay üretimi
g) Kaotik Dinamiklerin Elektronik, Optik ve Optoelektronik Gerçekleştirilmesi
h) Kaotik Titreşimlerin Belirlenmesi ve Kontrolü

Kaos ve kaotik sistem dinamiği ile ilgili en geniş çalışma alanı ise; bu derece ilginç özelliklere sahip kaotik işaretler ve sistemlerden olumlu yönde yararlanma fikri doğrultusunda yapılan çalışmalarla oluşmuştur. Bu çalışmalar özellikle kaotik işaretlerin ve sistemlerin senkronizasyonu ile bu senkronize kaotik sistemlerin güvenilir ve gizli haberleşme amaçlı tasarım ve uygulamalarda kullanılabilme olasılığını kapsamaktadır[18]. Fakat, daha önceden de ifade edildiği gibi ilk başlarda kaotik sistemlerin bu tür haberleşme uygulamalarında kullanılabilmeleri için senkronizasyonlarının sağlanması, bu konunun önündeki en büyük engel olarak görülüyordu. Pecora ve Carroll'un yapacakları bir çalışmaya kadar, başlangıç şartları ve sistem parametrelerine hassas bağımlı olmalarından dolayı iki ya da daha fazla kaotik sistemin senkronize olamayacağı düşünülüyordu. Pecora ve Carroll bu düşünceyi ortadan kaldıran çalışmalarında , ele aldıkları orijinal bir kaotik sistemi keyfi olarak iki ayrı kısma ayırıp bunları sürücü ve cevaplayıcı alt-sistemler olarak adlandırmışlardır[19]. Alıcı modülde cevaplayıcı alt-sistemin aynısı oluşturularak bu alt-sistemin orijinal sistemin sürücü kısmıyla sürülmesi durumunda, kaotik senkronizasyonun sağlanabileceğini yani, alıcı modülde üretilen kaotik işaretin orijinal sistemden gelen kaotik işarete yakınsayacağını gerek teorik gerekse deneysel olarak göstermişlerdir[20].

Kaosun bilime getirdiği yeni açılımlar çeşitli amaçlarda kullanılmak üzere kaotik işaretler oluşturan osilatörler gelişmesine ya da var olan osilatörler devreler üzerinde araştırmaların yapılmasına neden olmuştur. Bu osilatörlerin en tanınmışlarını şöyle sırlayabiliriz:

1. Lorenz Kaotik Osilatörü
2. Chua Kaotik Osilatörü
3. Rossler Kaotik Osilatörü
4. Vanderpol Kaotik Osilatörü
5. Duffing Kaotik Osilatörü

2.2. Kaos Teorisinin Kullanım Alanları

Kaotik sistemler, kendi içerisindeki sistematiğinden dolayı incelenebilmektedir. Bilakis Choa, Lorenz, Rossler gibi bilim adamları bu sistematikleri incelemiş ve kendi yöntemlerini ortaya koymuşlardır.

Dünyanın nonlineer sistemler halinde yaratılmış olması itibariyle birçok bilimsel yöntemde kaotik sistemler kullanılmaktadır. Bunların başlıcaları şunlardır[21].

a) Yapay zekâ
b) Sanal ağ çözümlerinde
c) Görüntü şifreleme teknikleri
d) Sağlık sistemleri
e) Savunma sistemlerinde (haberleşme)
f) Güvenlik sistemleri

2.2.1. Yapay zeka

Yapay zekâ için kaos teorisi vazgeçilmezdir. Çünkü yapay zekanın temel yaklaşımlarından olan matematiksel yaklaşımın temelini kaos teorisi oluşturmaktadır.

2.2.1.1. Matematiksel yaklaşım

Kaos konusunda bu uzun girişten sonra konunun beyinle ilişkisine gelelim. Beynin fizik yapısı ve görünüşü fraktaldır. Bu yapı, beynin gerek evrimsel, gerekse canlının yaşamı sürecindeki gelişimin ürünüdür ki, bu gelişimin deterministik (genlerle belirli), ancak çevre ve başlangıç koşullarına son derece duyarlı, yani kaotik olduğu açıktır. Beynin yalnızca oluşumu değil, çalışma biçimi de kaotiktir. Beyni oluşturan inanılmaz boyuttaki nöron ağının içinde bilgi akışı kaotik bir şekilde gerçekleşir. Kaotik davranışın tarama özelliği ve bunun getirdiği uyarlanırlık (adaptivite) sayesinde, beyin çok farklı durumlara uyum sağlar, çok farklı problemlere çözüm getirebilir, çok farklı fonksiyonları gerçekleştirir.

EEG sinyalleri üzerine yapılan araştırmalar göstermiştir ki, sağlıklı bir insanın sinyalleri kaotik bir davranış gösterirken, epilepsi krizine girmiş bir hastanın sinyalleri çok daha düzenli, periyodik bir davranış sergilemektedir. Yani epilepsi krizindeki hastanın beyni, kendini tekrarlayan bir davranışa takılmış ve kaotik (yani sağlıklı) durumda sahip olduğu adaptivite özelliğini yitirmiştir. Bunun sonucu hasta, kriz sırasında en basit fonksiyonlarını bile yerine getiremez olur.

Kaos bilimini ortaya çıkaran, karmaşık olguları basit parçalara ayırmak yerine onları bir bütün olarak görme eğilimi, beyni inceleyen bilim adamlarının da yaklaşımını belirlemiştir.Eskiden beyin farklı fonksiyonlardan sorumlu merkezler şeklinde modellenirken, artık holistik (bütünsel) beyin modeli geçerlilik kazanmıştır. Bu modele göre herhangi bir işlev gerçekleştirilirken, beynin tümü bu olguya katılmaktadır.

Önümüzdeki yıllarda beynin yalnız alt düzey fizyolojik işleyişinin değil, öğrenme, hatırlama, fikir yürütme gibi üst düzey işlevlerinin de modellenmesinde kaosun çok önemli bir rol oynayacağı görülmektedir.

2.2.2. Sanal ağ çözümleri

Sanal Özel Ağ çözümlerimiz sayesinde şirketinizin yurt çapındaki tüm şube, bölge ofisi, bayiler, iş ortakları, üretim ve satış birimleriniz, mağaza ve depolarınız arasında en uygun, ekonomik ve güvenli şekilde iletişim kurma imkânı elde edersiniz. Aynı anda ağınızdaki tüm bilgisayarlar ile veri, ses ve görüntü iletişimi kurarak, ayrıca ödeyeceğiniz iletişim masraflarından kurtulur, şirket işlerini daha hızlı ve güvenli bir şekilde uzaktan yürütme ve kontrol etme imkânına kavuşursunuz. İsterseniz merkeze sağlayacağınız Internet erişim hizmeti ile coğrafi olarak şirket merkezine uzak olan şirket birimlerinin Internet erişimini paylaşarak kullanmasını sağlayabilirsiniz. Böylece şirketinizin Internet erişim maliyetlerinde önemli bir avantaj elde edersiniz.

Görüntü, ses veri transferi hedefe gönderilen kaotik sinyalin üzerine iletilmek istenilen verinin bindirilmesi şeklinde günümüz teknolojisi için kolay bir uygulamadır.

Gönderilen verinin kaotik sinyaller üzerine bindirilmiş olması hasebiyle çözülmesi çok zor olan bir şifreleme metodu kullanılmış olunur.Karşı taraf gönderilen kaotik sinyalin aynısını üretemeden kesinlikle üzerindeki bilgiyi ayıklayıp çözememektedir[22].

2.2.3. Görüntü şifreleme teknikleri

Genel olarak bilgi güvenliği alanında üç temel özellik vardır.

1. Gizlilik: Yetkisiz birisi mesajı okuyamamalıdır.

2. Bütünlük: Yetkisiz birisi mesajı değiştirmemeli ya da mesajı bozmamalıdır.

3. Kullanılabilirlik: Mesajlar yetkili kişilere tam olarak erişilebilir olmalıdır.

Resim kriptosisteminin güvenliğinin değerlenbirilebilmesi için aşağıdaki 5 saldırı tipi önerilmiştir. Bunların herbiri kriptoanalistin kullanılan şifreleme algoritmasını bildiğini varsayar[23].

2.2.3.1. Cipherimage-Only Attack

Bu saldırıda yetkisiz kullanıcının ağdan cipherimage'i aldığı ve K gizli anhtarına sahip olmadığı kabul edilir. Diğer bir deyişle saldırgan, gizli anahtarı adece ele geçirilen cipherimage'i kullanarak elde etmelidir[24].

2.2.3.2. Known-Plainimage attack

Yetkisiz kullanıcının birkaç plainimage, cipherimage çifti ele geçirdiği varsayılır. Kriptoanalist plainimage'leri şifrelemek için kullanılan 3 anahtarı belirlemeli ya da aynı anahtarla yeni şifrelenen cipherimage'leri deşifreleyecek bir algoritma geliştirmelidir[25].

2.2.3.3. Chosen-Plainimage attack

Bu saldırıda saldırgan, plainimage'leri ve onların cipherimage'lerini seçebilmektedir. Bu yöntem known-plainimage saldırısından daha güçlüdür. Çünkü kriptoanalist şifrelemek için bazı özel plainimage'leri seçebilir, bu da gizli anahtar hakkında daha fazla bilgi verir[26].

2.2.3.4. Jigsaw Puzzle attack

Bu saldırı tipinde saldırgan cipherimage'i daha küçük parçalara ayırır. Daha sonra kriptoanalist bu parçaları teker teke kırar. Her bir alan cipherimage'den çok daha küçük olacağına göre herbirini kırmak için gereken hesaplama zamanı ipherimage'i kırmak için gereken zamandan çok daha azdır. Bu nedenle jigsaw puzzle saldırısı diğerlerinden çok daha güçlü bir yöntemdir[27].

2.2.3.5. Neighbor attack

Saldırganın resmin bir parçasını bildiği kabul edilir. Birçok resimde alan sınırları boyunca değişimler düzgündür. Bu nedenle kriptoanalist bu özelliği kullanarak komşu alanların sınırlarını daha hızlı bir şekilde seçebilir. Birçok resim düzgün yapıda olduğu için kriptoanalist resmin bilinen kısmı için komşu pikselleri elde edebilir ve tim cipherimage'i kırabilir.

2.2.4. Sağlık sistemleri

Günümüzde kalp, karaciğer böbrek ve diğer organların ve sistemlerin dinamiğine ilişkin bilgiler toplanmakta, kayıtları kullanarak boyut analizi yapılıp, Fizyolojik sistemi tanımlamak için kullanılması gereken parametre sayısını belirlendikten sonra, çeşitli yöntemlerle hesaplamalar yapılarak sistemin kaotik davranış gösterip göstermeyeceği konusunda bilgi sahibi olunmaktadır.

Bu bilgiler ışığında organ ve sistemlerin davranışları hakkında hesaplamalar yapılarak tedavi şekli ve metodu belirlenmektedir. Hatta ne zaman nasıl bir rahatsızlık meydana geleceği tahmin edilebilmektedir.

2.2.5. Savunma sistemleri

Savunma sistemlerinde de görüntü işleme sistemlerinde olduğu gibi gönderilecek bilgi kaotik bir sinyale bindirilmekte ve bu yolla şifreli olarak gönderilmiş olmaktadır.

Aynı şekilde bilginin karşı taraftan çözülebilmesi için aynı kaotik sinyal kullanılarak ayıklanması gerekmektedir. Kaotik parametrelerinde küçük değişiklikler farklı sonuçlar vereceği için karşı tarafın aynı kaotik sinyali kullanması gerekir ki bu da bilgi saklamak için çok güvenli bir sistemdir.

2.2.6. Güvenlik sistemleri

Güvenlik sistemlerinde de kaotik sistemler iris, parmak izi tanıma gibi konularda kullanılmaktadır.Bilindiği üzere kaotik sistemler ilk noktada matematiksel formda başlamaktadır. Çeşitli bilim adamlarının ortaya koyduğu hesaplama yöntemleriyle

sistemin parametreleri belirlenmektedir. Güvenlik sistemlerinde sensörlerle algılanan iris veya parmak izi örnekleri matrissel formlarda data olarak saklanmaktadır. Sensörlerden geçen diğer örnekler dijital ortamda mikro işlemci tabanlı olarak saklanmış olan datalarla karşılaştırılıp, sistemin açılıp açılmayacağına karar verilmektedir[28].

2.2.7. Meteoroloji ve gök bilimi hesaplamarı

Gökbilimciler, yıldızların, gezegenlerin ve Güne Sistemindeki uydu ve kuyruklu yıldızların hareketini modellemek için kaos teorisini kullanmaktadır. Güneten yayılan ve dünya manyetosferi tarafından yakalanan yüklü parçacıkların atmosferde meydana getirdii orora olayının açıklanmasında kaos teorisi yardımcı olmaktadır. Havayı belirleyen atmosferin hareketlerinin incelenmesi kaos teorisinin en heyecanverici uygulama alanlarından birini oluturmaktadır[29].

Meteorolojistler, atmosferin hareketini açıklayan oldukça karmaık matematiksel modelleri kullanarak gelecekte havanın nasıl olacaını öngörmeye çalımaktadır. Ayrıca, örnein Muson yamurlarının mevsimlik tahmini gibi daha uzun süreli tahminler yapabilmeye yönelik çalımalar gündemdedir. Meteorolojistler bu çalımaların ötesinde, sera etkisi gibi insan aktivitelerinden kaynaklanan iklim deiikliklerinin tahmini yönünde çalımalar yapmaktadır. Bununla birlikte atmosferin kaotik bir sistem olduunu biliyoruz. Atmosfer doası gerei öngörülemez bir karaktere sahiptir. Öyleyse uzun vade hava ve iklim öngörüsü bouna bir çaba mıdır? Televizyonlardaki hava durumu raporları ile yetinip, gerisini ansa mı bırakmalıyız? bazı hava durumları hafta, ay hatta mevsimler boyunca sürebilir. Bu tür hava durumları bireysel hava sistemleri ile deil, yukarı atmosferde esen ve jet akımları olarak adlandırılan kuvvetli rüzgarların konumu ile yakından ilikilidir. Jet akımları önümüzdeki yaz mevsiminin yaılımıyoksa kurakmı olacaını; kı mevsiminin ılımanmı yoksa sertmi geçeceğini belirler[30].

2.2.8 Haberleşme

Haberleşme sistemlerin kaos dinamiğinin kullanılması güvenli bilgi aktarımının araştırılması sonucu ortaya çıkmıştır. Kaos dinamiği girişe olan hassas bağımlılığı ve doğrusal olmayan yapıda olması haberleşme sistemlerini, kaotik işaret üreten

osilatörlerle bilgi işaretinin taşıması ilgi çekici bir konu haline getirmiştir. Bu işlem için kullanılan ve bilen en önemli kaotik işaret üreten osilatörler; Chua, Lorenz ve Rossler dır. Bunların dışında ayrıca geliştirilmiş olan çok sayıda kaotik osilatör devre mevcuttur[31].

1990 lı yıllarda kaos sistemlerin çekiciliğinin artması ve bu sistemler üzerinde daha çokaraştırma yapılmasına neden olmuştur. Araştırmacıların bu yıllarda en çok üzerinde durdukları konu, kaotik işaretlerle maskelenerek gönderilen bilgilerin eşlemesi olmuştur. Kaotik bir işarete bindirilen bilgi işaretinin tekrar elde edilmesi oldukça güç ve zor bir dizi işlemler ile gerçekleşmektedir. Bu alanda yapılan çalışmalar yeni ufuklar açmıştır. Kaotik sistemlerin bu alanda kullanılması ve zengin işaret üretme yeteneğinin olması diğer bilimler için de ayrıca bir çekicilik teşkil etmiştir[32].

Kaotik işaret üretmek basit bir yapıda ki doğrusal olmayan bir devre ile kolaylıkla gerçekleştirilmektedir. Asıl sorun bu işaretin kullanıldığı sistemlerde alıcı taraf için eşlemeyi sağlamaktır[33].

Kaotik sinyallerin karakteristiği olan genişbandlılık ve gürültüye benzer özelliği bilgi sinyalinin kaotik işaretin özelliğinin üzerinden taşınması haberleşme sistemlerinde modülasyon için güvenli bir ortam oluşturur. Haberleşme alanında oldukça popüler olan Code Division Multiple Access (kod bölmeli çoklu erişim) ya benzer bir yöntem ile kaotik dalga üzerinden bilgi sinyalinin modülasyon gerçekleştirmiştir. Güvenli ve gizli haberleşme maçlı olmak üzere analog ve dijital sistemler üzerinde bu işlemler gerçekleştirilmektedir[34].

Analog modlu sistem tasarımı için temel prensip bilgi işaretinin kaotik bir işaret ile maskelenmesinin yapılması ve iletilmesi işlemlerine dayanır. Maskelen bilgi işareti alıcı tarafında aynı formu oluşturulup kaotik ortamda üretilen işaretten çıkartılıp bilgi işareti tekrar elde edilmektedir. Böyle bir işlem için birçok devre geliştirilebilir. Haberleşeme alanı için kaotik işaretlerin kullanılmasında ki çekicilik burada olmaktadır. Çeşitliliğin çok olması bilginin gizlenmesi ve güvenliği için gerekli ve yeter nedenlerdendir.

Diğer bir yöntem ise dijital modlu sistem tararımı ile gerçekleştirilmesidir. Bu işlem için temel prensip kaotik işaret üreten sitemlerin parametreleri için değişik kaotik işaretlerin ikili (binary) sistemde kodlanması ve iletilmesi işlemlerine dayanır.

Bir ya da birden fazla kaotik devrenin eşlemesinde pratikte özel teknikler kullanılmadan kolayca sağlanamaz. Kaotik sistemlerin genel özelliklerinden bir olan başlangıç şatlarına duyarlı olması pratikte bire bir uyumlu devre gerçekleştirilse bile başlangıç şartlarını aynı şekilde vermek mümkün değildir.

2.2.8.1. Kaotik haberleşme sistemlerinin genel yapısı

Bilgi iletimi yukarıda anlatılan haberleşme sistemleri ile olmaktadır. Teknolojinin gelişmesi ile bu sistemler daha da gelişerek insanoğlu yararına sunulmaktadır. Bilginin bu kadar önemli olduğu bir dönemde elbette bu bilginin güvenli bir şekilde ulaştırılması istenmektedir. Bu nedenle gizli ve güvenilir haberleşme sistemleri için araştırmalar yapılarak şifreli yazı/görüntü sistemlerin gelişmesi neden olmuştur. Bu alanda, kaotik işaretlerin özellikleri önemli ilerlemelere neden olmuştur. Yapısal özelliği olan tahmin edilemezlik bu işaretlerin bilgi iletiminde kullanılmasını uygun hale getirmiştir. Ayrıca haberleşme sistemlerinde genişbandlı sinyaller kullanılmaktadır. Bu sayede veri iletimi ve alımı daha kaliteli olmaktadır. Bunun paralelinde kaotik bir işarette genişbanlı bir karakteristiği sahip olması taşıyıcı olarak kullanılmasını cazip hale getirmektedir.

Mevcut haberleşme sistemlerine ek olarak dâhil edilen sistemlerle bilgi iletimi daha güvenli bir ortamda gerçekleşmektedir. Aşağıda ki Şekil 2- 6 da böyle bir sistemin şematik yapısı görülmektedir. Bilgi kaotik bir işaret ile şifrelenerek alıcıya ulaştırılmaktadır. Alıcıda benzer bir yapı ile eşleme yaparak gerçekleşen sistem sayesinde gönderilen bilgiyi tekrar elde etmek mümkündür.

BÖLÜM 3. KAOTİK OSİLATÖRLER

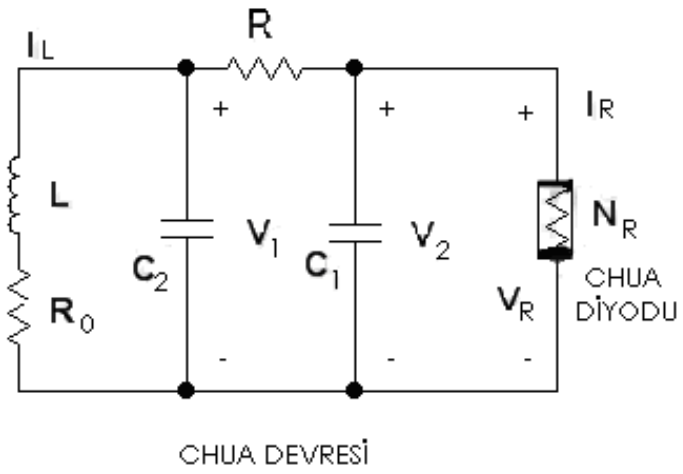
3.1. Chua Kaotik Osilatörü

Kaos işaretlerin ne anlama geldiği konusunda herhangi bir fikirleri olmayan bilim adamları çıkan sonuçları deterministik sonuçlarla bağdaşmayınca ya doğaüstü güçlere ya da sistem parametrelerin yanlış girildiğine kanat getirdiler[35]. 1970 li yıllarda Japonya da Prof. Utea, laboratuarda ilk olarak kaosu gördüğünde bir anlam verememiş, işe yaramayan işaretler olarak yorumlamıştır.

Ancak benzer sonuçları bulan ABD li araştırmacılar bu sonuçlar üzerinde yoğunlaşarak bu işaretleri anlamaya çalıştılar. Nitekim Prof. Dr. L.O. Chua'nın 1983 yılında bilim literatürlerinde kendi adı ile anılan kaotik işaret üreten bir osilatör bulmasıyla bu işaretle anlam kazandı.Böylece elektronik yeni bir boyut kazandı.

Chua devresi en karmaşık kaosun varlığının deneysel olarak kurulabildiği, sayısal olarak doğrulanabildiği ve matematiksel olarak kanıtlanabildiği en basit devrelerden biridir[36].

Şekil 2-2 de görüldüğü gibi Chua devresi bir doğrusal indüktans (L, iç direnç Ro), iki doğrusal kapasitör (C1, C2), bir doğrusal direnç (R), ve Chua diyotu olarak adlandırılan doğrusal olmayan yapıda gerilim kontrollü direnç (NR)'den oluşur. Herhangi bir elemanın değeri değiştirildiğinde kaotik davranış dizileri elde edilir.



Şekil 3.1. Chua Devresi

Chua devresi üç adet durum denklemi ile tanımlanır:

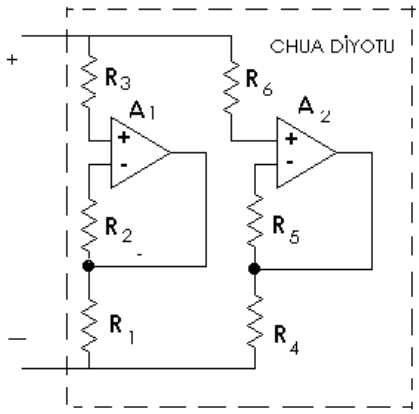
$$\frac{dV_1}{dt} = \frac{G}{C_1}(V_2 - V_1) - \frac{1}{C_1} g(V_1)$$

$$\frac{dV_2}{dt} = -\frac{G}{C_2}(V_2 - V_1) + \frac{1}{C_2} I_L$$

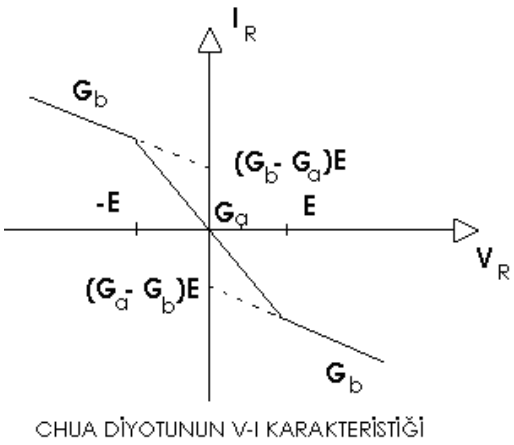
$$\frac{dI_L}{dt} = -\frac{R_c}{L} I_L - \frac{1}{L} V_2$$

Burada g(V1) olarak tanımlanan NR direncinin Şekil 2–3(b) gösterilen parça parça doğrusal karakteristiği temsil etmektedir. [37]

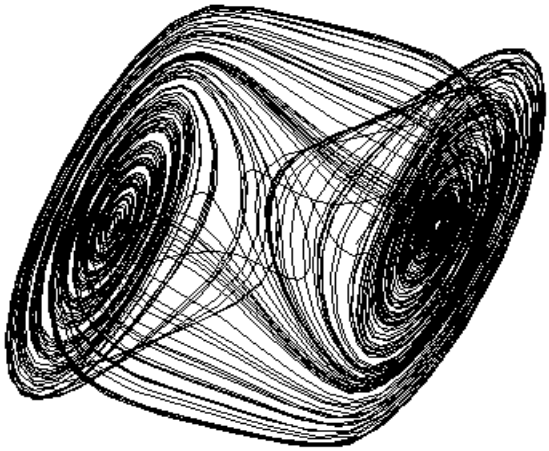
$$g(V_1) = \begin{cases} G_b V_R + (G_b - G_a)E & ; \ V_R < -E \\ G_a V_R & ; -E < V_R < E \\ G_b V_R + (G_a - G_b)E & ; \ V_R > E \end{cases}$$



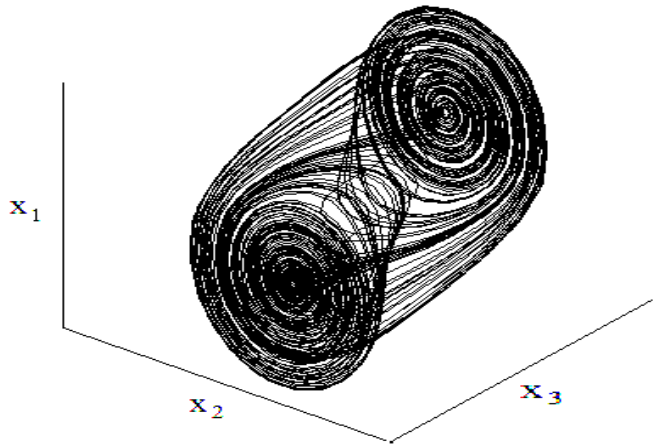
Şekil 3.2. Chua Doğrusal Olmayan diyodu



Şekil 3.3. Chua diyotunun Karakteristiği



Şekil 3.4. Chua Dinamik Denklemleri ile oluşturulan ‘Çift Spiralli Çekici’



Şekil 3.5. Chua Kaotik osilatörünün x_1, x_2 ve x_3 fazlarının karşılaştırılması

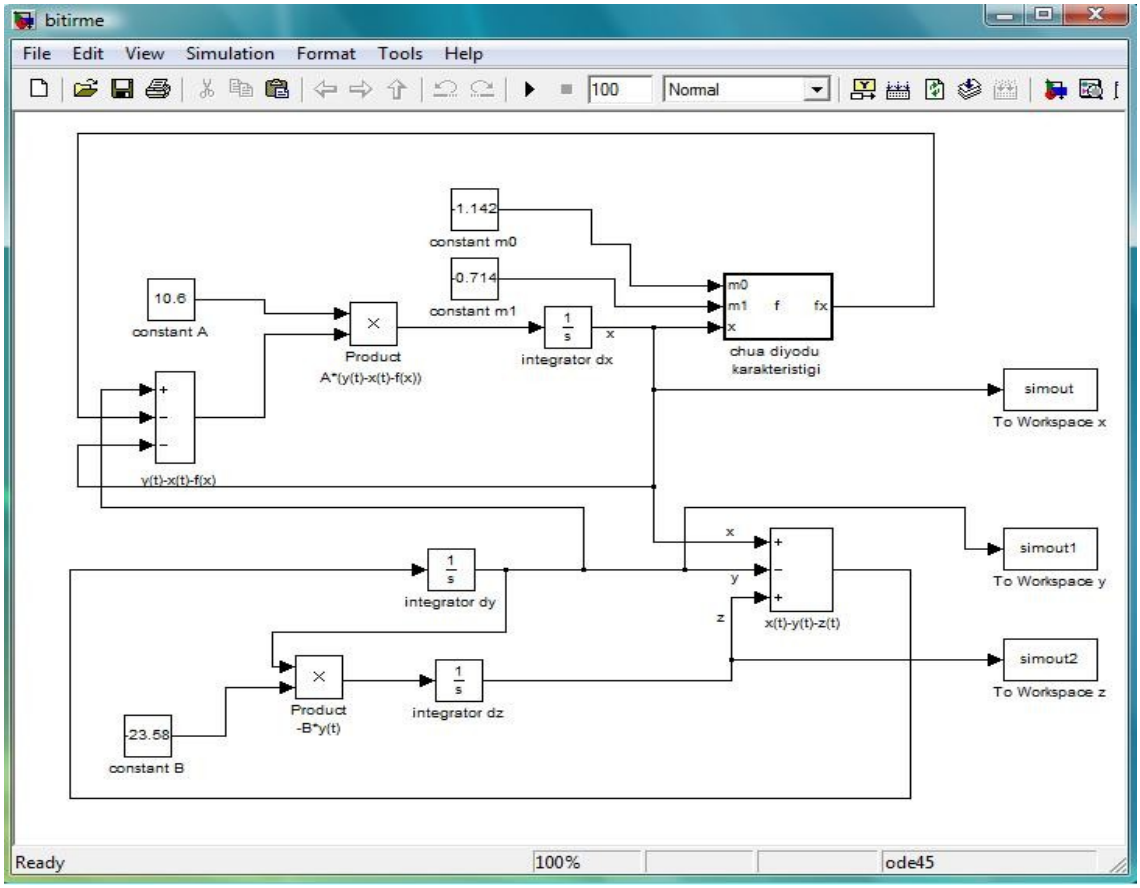
Chua dinamik denklemleri ile elde edilen benzetim çalışmaları Şekil 3.5 de verilmiştir. Farklı giriş değerleri için Değişik sonuçlar ortaya çıkmıştır. Bu dinamik denklemlerle elde edilen sonuçlar güvenirli ve gizli haberleşme için gerekli olan kaotik taşıyıcı özelliğini sağlar[38].

3.1.1. Chua devresinin MATLAB simulink ile tasarımı

Bu bölümde birinci bölümde elde etmiş olduğumuz x,y ve z denklemlerini MATLAB simulink ortamında adım adım tasarlayacağız ve inceleyeceğiz.

3.1.1.1. Chua devresinin MATLAB simulink ile tasarımında kullanılacak simulink bloklarının tanıtılması

Bu bölümü bitirdiğimizde elde edeceğimiz matlab simulink dosyası aşağıdaki resimde görüldüğü gibi olacaktır.



Şekil 3.6. MATLAB Simulink Tasarımının Bitmiş Hali

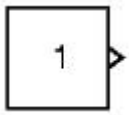
Şimdi bu tasarımda kullandığımız simulink bloklarını tek tek ele alalım.

3.1.1.2. Sabit bloğu (Constant block)

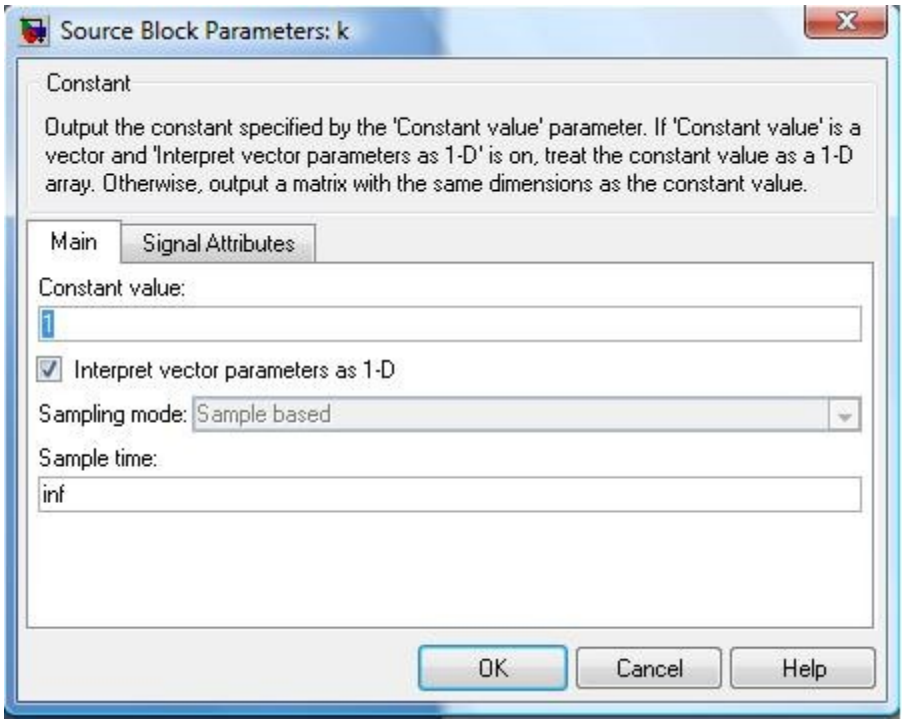
Bu blok sabit bir değer üretir.

Bulunduğu kütüphane: SOURCES

Bloğun şekli:



Sabit bloğu reel veya kompleks bir sayı üretir. Bu blok, Constant Value parametresinin boyutuna ve Interpret vector parameters as 1-D parametresinin ayarına göre, bir sabit, bir vektör (1 boyutlu dizi) veya bir matris (2 boyutlu dizi) üretebilir. Biz bu bloktan sadece sabit bir sayı üretmesini isteyeceğiz, ayrıntılı bilgi için bkz matlab help.



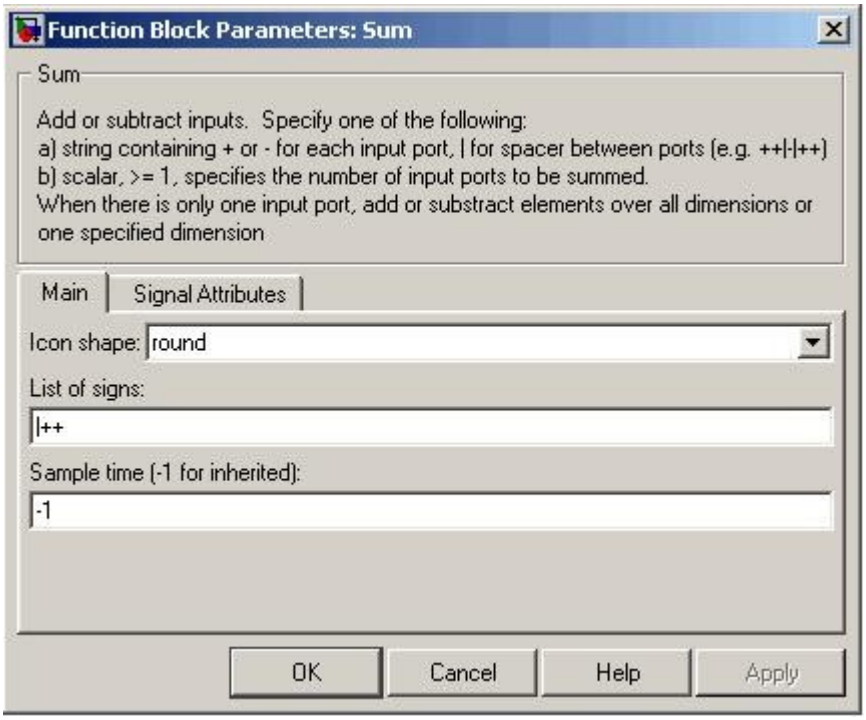
Şekil 3.7. Constant Bloğu

Tasarımın son hali şeklinde görülen k,m0,m1,alpha ve beta constant bloklarıdır.

3.1.1.3. Toplama-Çıkarma bloğu (Sum block)

Kendisine gelen girişlerle toplama veya çıkarma işlemleri yapılabilen blokdur.
Bulunduğu kütüphane: MATH OPERATİONS
Bloğun şekli:

Bu blok vektör veya matrisler içinde toplama çıkarma yapabilir.



Şekil 3.8. Sum Bloğu

Yaptırılacak olan işlem sayısı ve sırası yukarıdaki şekildeki List of signs kısmına yazılacak olan (+), (-) ve (|) işaretleriyle yapılır.

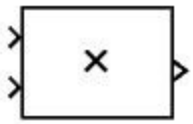
Örneğin "+-+" girilmesi durumunda üç tane veri girişi gerekir, bu işlem birinci veriden ikinci veriyi çıkartır ve üçüncü veri ile toplar.(|) işareti ise blok üzerindeki giriş kapıları arasında boşluk bırakmayı sağlar.

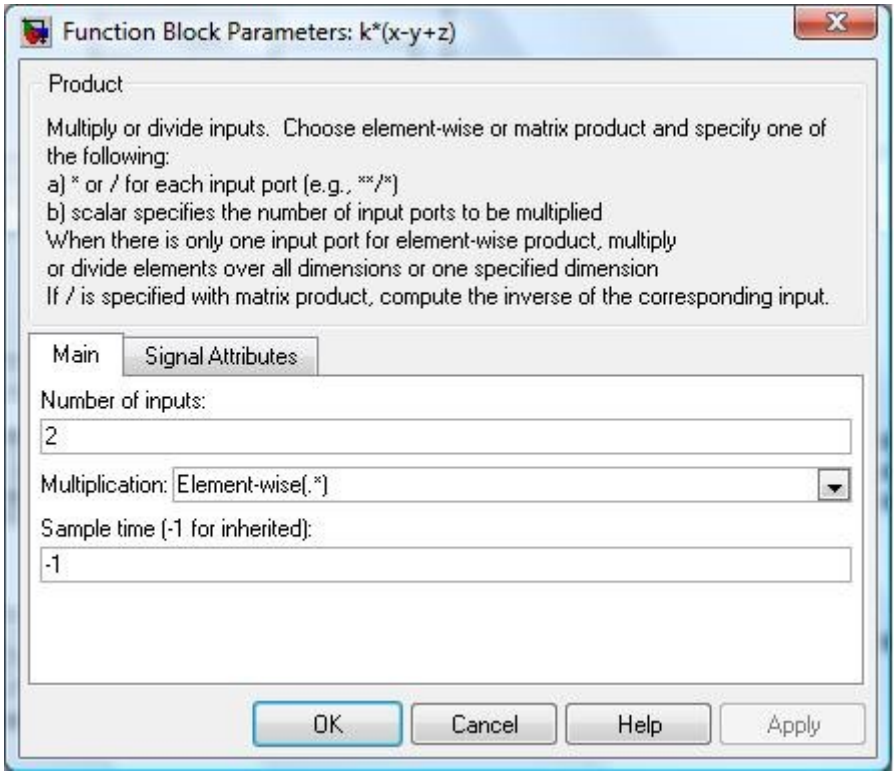
Şekildeki Icon shape kısmı ise toplama-çıkarma bloğunun şeklini değiştirmeye yarar. Eğer round seçeneği seçilmiş ise sum bloğu daire şeklini alır, eğer rectangular seçilmiş ise sum bloğu dikdörtgenimsi bir hal alır.

3.1.1.4. Çarpma-Bölme bloğu (Product block)

Girişine gelen verileri çarpmaya veya bölmeye yarayan bloktur.
Bulunduğu kütüphane: Math Operations

Bloğun şekli:





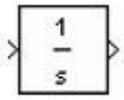
Şekil 3.9. Product Bloğu

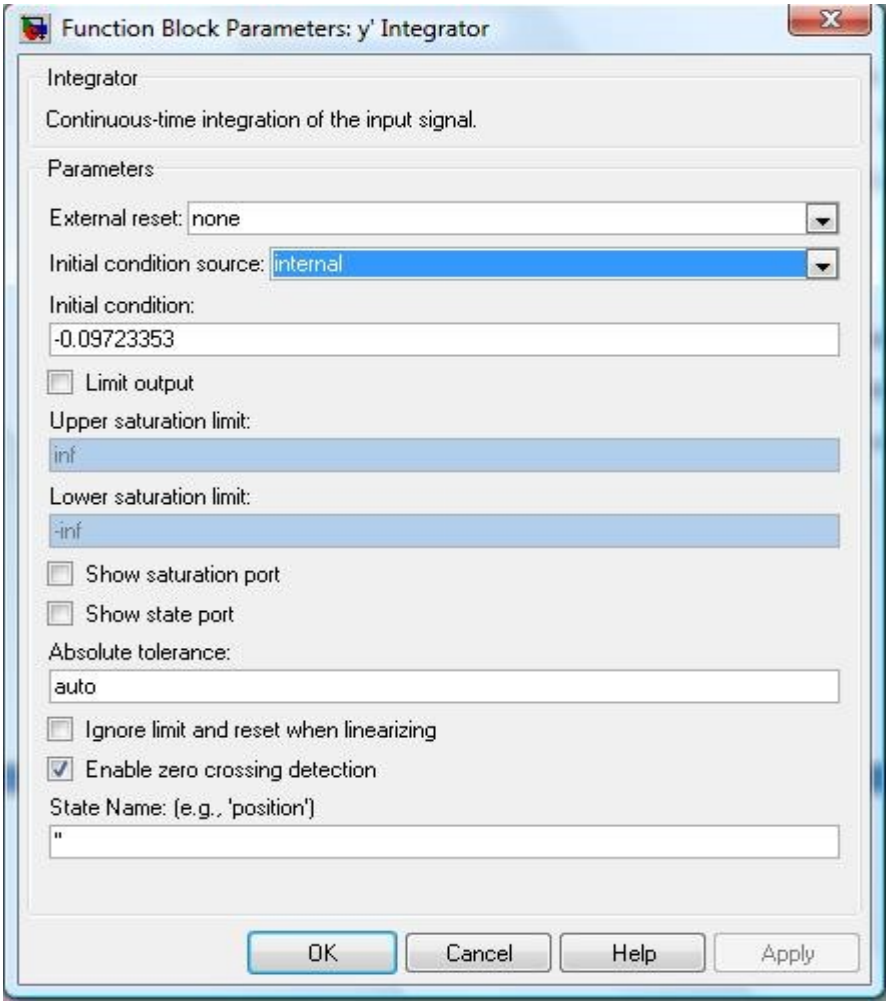
Bu blok ile matris çarpımları da yapılabilir. Fakat bu çalışmada matris olmadığı için blok parametrelerinde bulunan multiplication kısmında Element-wise(.*) kısmı

seçilmiş olmalıdır. Number of inputs parametresinde ise kendisi ile işlem yapılacak giriş sayısı belirlenir, yani çarpılacak veri sayısı belirlenir.

3.1.1.5 İntegral alma bloğu (Integrator block)

Girişine gelen verinin integralini çıkışa veren bloktur.
Bulunduğu kütüphane: Continuous
Bloğun şekli:





Şekil 3.10. İntegrator Bloğu

Girişine u(t) fonksiyonu uyguladığımızda çıkışta alacağımız fonksiyon y(t) olsun ve başlangıç şartımızda y_0 ise,bu blok içinde aşağıdaki işlem uygulanır.

$$y(t) = \int_{t_0}^{t} u(t)dt + y_0$$

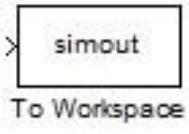
Blok parametreleri:

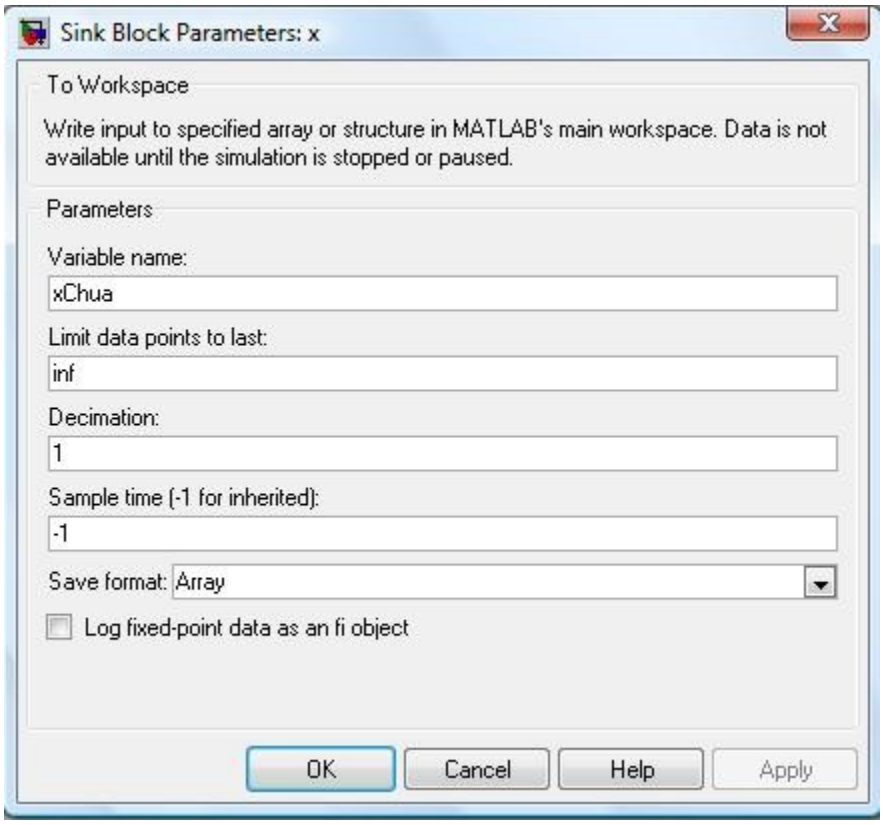
External reset parametresi integrator bloğunun seçilen tetikleme işareti geldiğinde başlangıç şartını almasını sağlar. Bu çalışmamızda dışarıdan gelen bir sinyale göre integratörümüzü resetlemeyeceğimizden dolayı bu parametreyi none olarak seçiyoruz.İnitial condition source parametresi ise başlangıç şartının alınacağı kaynağın belirlenmesinde kullanılır, biz başlangıç şartını kendimiz belirleyeceğimizden dolayı bu parametreyi internal olarak belirleyeceğiz.İnitial condition parametresi ise integralmizin başlangıç şartını yazdığımız parametre olacaktır.Diğer parametreler ise şekilde görüldüğü gibi kalacaktır.

3.1.1.6. Workspace'ye veri aktarma bloğu (To Workspace block)

MATLAB Simulink içinde üç boyurlu grafik çizdirme bloğu mevcut olmadığından dolayı, simulink çalışmamızdan alacağımız xChua, yChua ve zChua değerlerini MATLAB Workspace'ye aktaracağız ve Command window ortamında üç boyutlu grafiğimizi elde edeceğiz.

Bu blok kendisine gelen veriyi workspace'ye aktarır.
Bulunduğu kütüphane: Sinks
Bloğun şekli:





Şekil 3.11. To Workspace Bloğu

Blok parametreleri:

Variable name parametresine bloğun almasını istediğimiz ismi yazarız.
Limit data points tol ast parametresi ise alınabilecek en fazla örnek sayısının girildiği yerdir, inf yazılırsa sonsuz sayıda örnek alınabilir.

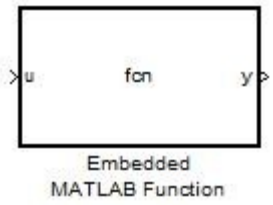
Save format parametresi ise workspace'ye aktarılacak olan verinin formatını belirler,biz verimizi dizi olarak saklayacağımızdan dolayı array seçeneğini seçmiş olduk.

3.1.1.7. Gömülü MATLAB fonksiyon bloğu (Embedded MATLAB function block)

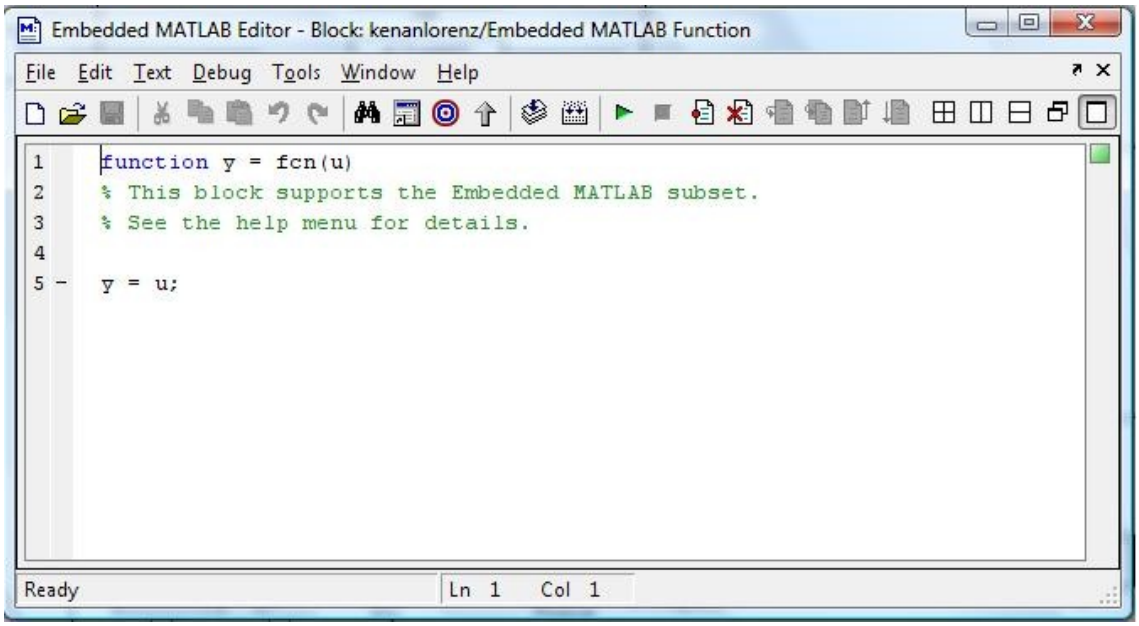
Simulink dosyaları içine matlab istediğimiz matlab programını gömmemizi sağlar. Chua diyodunun karakteristiği x'in 1 den büyük, -1 den küçük ve -1 ile 1 arasındaki her bir değer için farklı sonuç ürettiğinden gömülü matlab fonksiyonu kullanmak zorundayız.

Bulunduğu kütüphane: User-Defined Functions

Bloğun şekli:

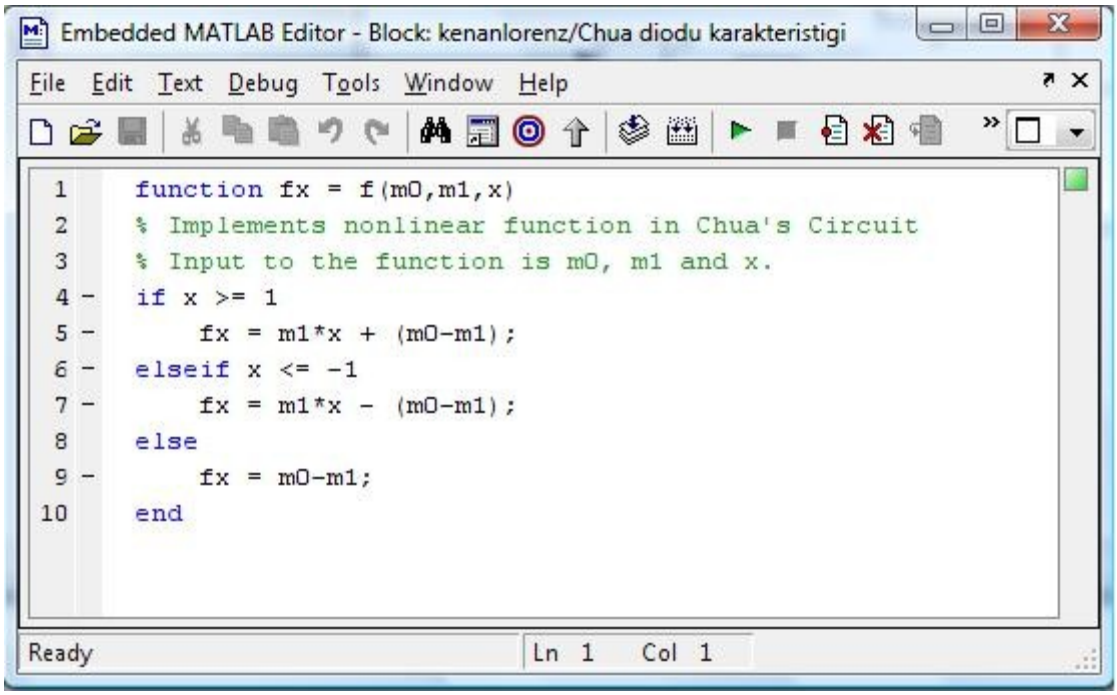


Bu bloğu açtığımızda karşımıza aşağıdaki şekil gelecektir.



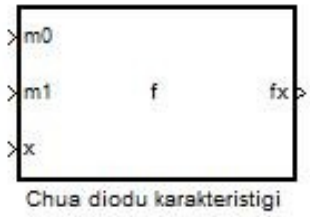
Şekil 3.12.MATLAB Fonksiyon Yazma Editörü

Chua diyodunun karakteristiğini elde edebilmemiz için bu bloğun içeriğini aşağıdaki şekilde görüldüğü gibi tekrar düzenlemeliyiz.



Şekil 3.13. MATLAB Fonksiyon Yazma Editörü

Şeklin içinde görülen fonksiyon chua diyodunun karakteristiğinin matematiksel olarak karşılığıdır. Bu değerler fonksiyona girilip kaydedildiğinde bloğun simulinkteki görüntüsü aşağıdaki gibi olacaktır.



Şekillerden görüldüğü gibi "function fx= f(m0,m1,x)" yazıldığında m0,m1 ve x, fx fonksiyonunda kullanılan giriş değerleri olmaktadır ve x değerinin -1 den küçük, 1 den büyük ve -1 ile 1 arasındaki üç ayrı değer için fx fonksiyonunun değeri değişmektedir.

Şimdi ise şimdiye kadar tanıtmış olduğumuz MATLAB simulink blokları ile chua devresinde çıkartmış olduğumuz diferansiyel denklemleri nasıl oluşturacağımızı ele alalım.

3.1.2. Chua devresinin MATLAB simulink ortamında tasarlanması

3.1.2.1. Dx,dy ve dz diferansiyel denklemlerinin integralleri alınıp X,Y ve Z denklemlerinin eldesi

Tasarladığımız chua devresinden elde ettiğimiz diferansiyel denklemlerin matematiksel ifadeleri olarak (1),(2) ve (3) denklemlerini chua diyotunun karakteristiği için ise (4) denklemini elde etmiştik.

$$\frac{dx}{dt} = A\big(y(t) - x(t) - f(x)\big),$$
$$\frac{dy}{dt} = x(t) - y(t) + z(t),$$
$$\frac{dz}{dt} = -By(t),$$
$$f(x) = m_1 x(t) + 0.5(m_0 - m_1)\, X\, (|x(t) + 1| - |x(t) - 1|)$$

dx,dy ve dz denklemlerinin integrallerini alırsak x,y ve z değerlerini elde etmiş oluruz, böylece x,y ve z aşağıdaki gibi olur.

$$x(t) = \int A\big(y(t) - x(t) - f(x)\big)dt + x_0$$
$$y(t) = \int \big(x(t) - y(t) - z(t)\big)dt + y_0$$
$$z(t) = \int -By(t)dt + z_0$$
$$f(x) = m_1 x(t) + 0.5(m_0 - m_1)\, X\, (|x(t) + 1| - |x(t) - 1|)$$

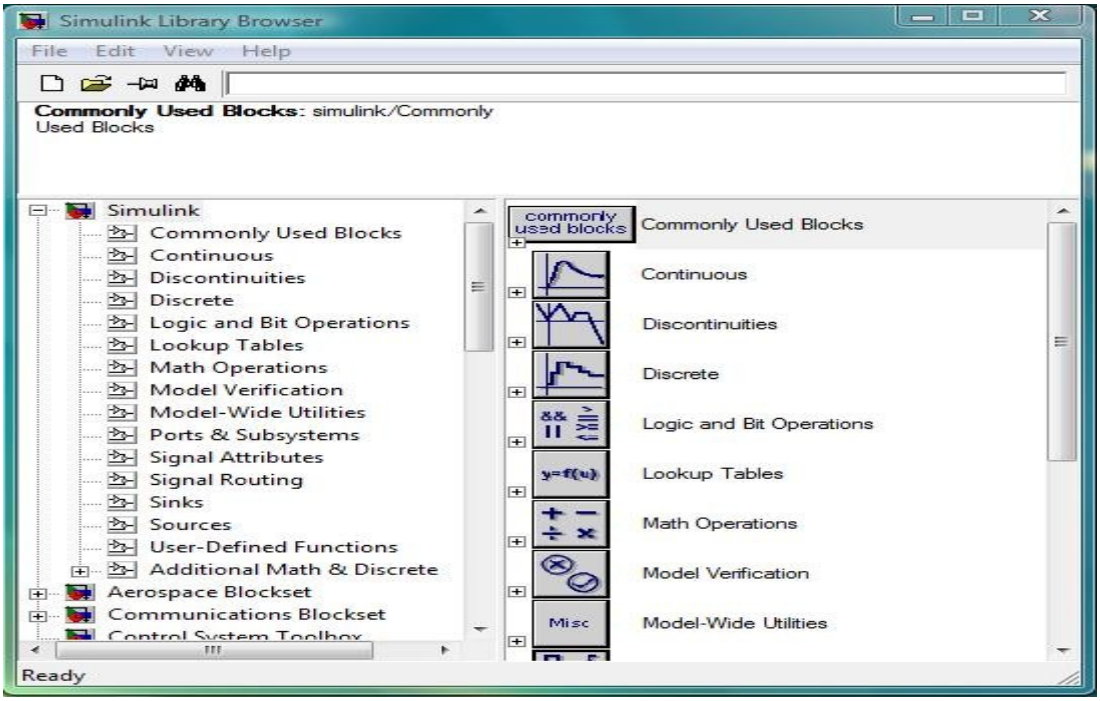
x,y ve z İntegral denklemlerindeki x0,y0 ve z0 değerleri İntegral denklemlerinin başlangıç şartlarıdır ve bu değerleri simulink dosyamızda integrator bloklarının içine ilk değer olarak atayacağız.

Denklemlerin integrallerini de aldıktan sonra simulink dosyamızı oluşturmaya başlayabiliriz.

3.1.2.2. Yeni bir MATLAB simulink dosyası oluşturmak

Yeni bir MATLAB simulink dosyası iki farklı şekilde oluşturulabilir, öncelikle simulink kütüphanesinin açılması gerekir buda ya MATLAB command window ortamında "simulink" yazılarak yahut matlab programının araç çubuğunda bulunan simulink simgesi tıklanarak açılabilir.
Simulink simgesi aşağıdaki şekilde görülen siyah kare içersindeki simgedir.

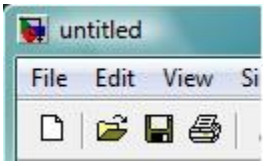
Bu simgeye tıklandığında karşımıza simulink kütüphanesi gelecektir, aşağıdaki şekil simulink kütüphanesi göstermektedir.



Şekil 3.14. Simulink Library Browser Görünümü

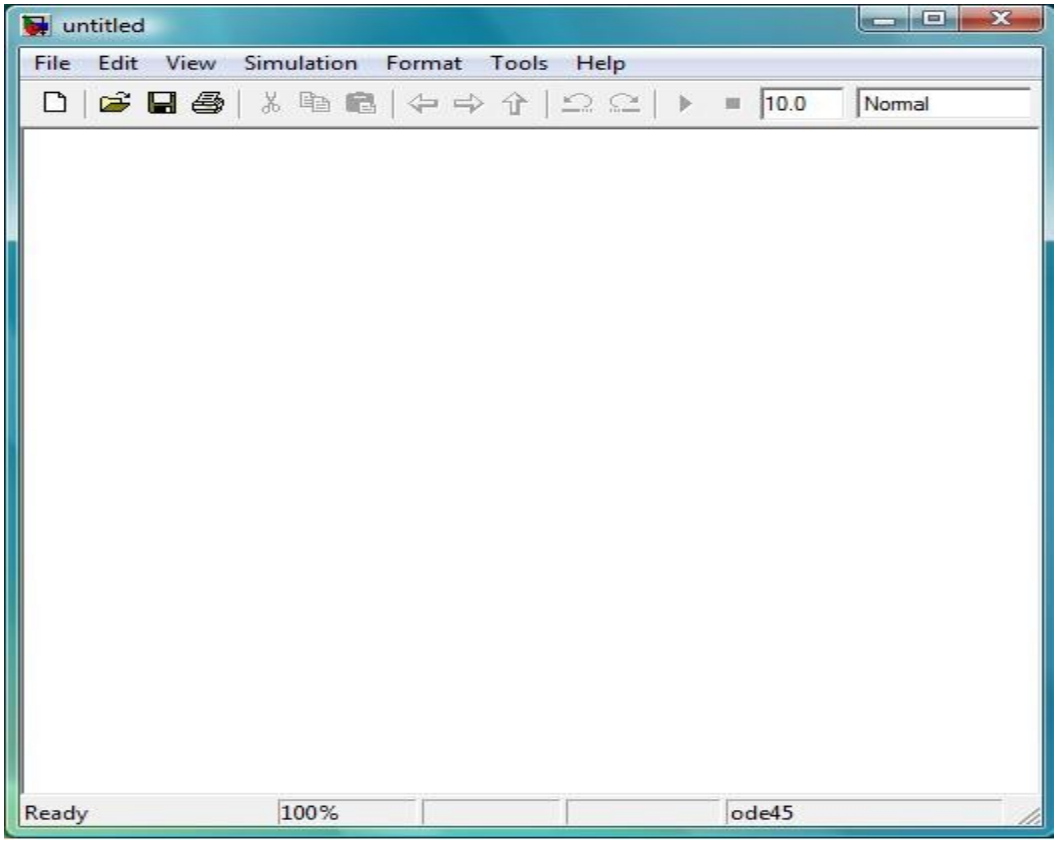
Simulink kütüphanesini elde ettikten sonra bir simulink dosyasının nasıl açıldığını öğrenelim. Bir simulink dosyası da iki farklı şekilde açılabilir, ya simulink kütüphanesindeki file→new→Model seçilerek yahut araç çubuğundan new model simgesi tıklanarak açılır.

New model simgesi aşağıdaki şekilde siyah kutucuk içinde bulunan şekildir.



Şekil 3.15.MDL Dialog Menüsünden Bir Kısım

Bu iki yöntemden herhangi biriyle simulink dosyası oluşturduğumuzda karşımıza aşağıdaki şekilde göründüğü gibi bir pencere gelecektir.

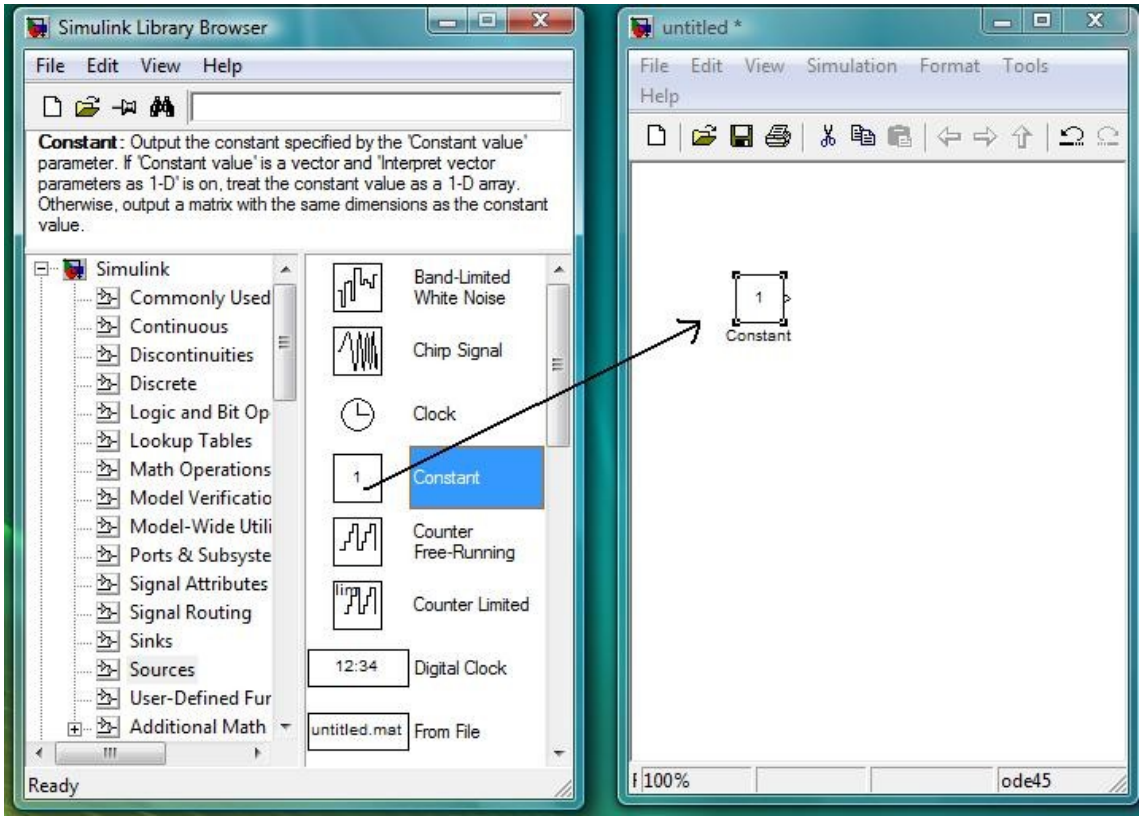


Şekil 3.16. Boş bir MDL dosyası görünümü

Şimdiye kadar hangi blokları kullanacağımızı ve kullanacağımız blokların özelliklerini, simulink kütüphanesinin nasıl açılacağını ve bu simulink kütüphanesinden yeni bir simulink dosyasının nasıl açılacağını öğrendik ve açtık. Şimdi ise özelliklerini öğrendiğimiz bu blokların simulink dosyasının içine nasıl çağrılacağını ve blokların birbirlerine nasıl bağlanacaklarına bir göz atalım.

3.1.2.3. MATLAB Simulink kütüphanesindeki blokların simulink dosyası içinde kullanılması

Simulink kütüphanesinden blokları tıkla ve sürükle (click and drag) yöntemiyle simulink mdl* dosyamızın içine atacağız, tıkla ve sürükle yöntemiyle blokların nasıl simulink dosyasına atıldığı aşağıdaki resimde gösterilmektedir.



Şekil 3.17. Bir Bloğun Simulink Dosyasına Atılması

$$x = \int A(y(t) - x(t) - f(x))dt + x0 \ldots (1)$$

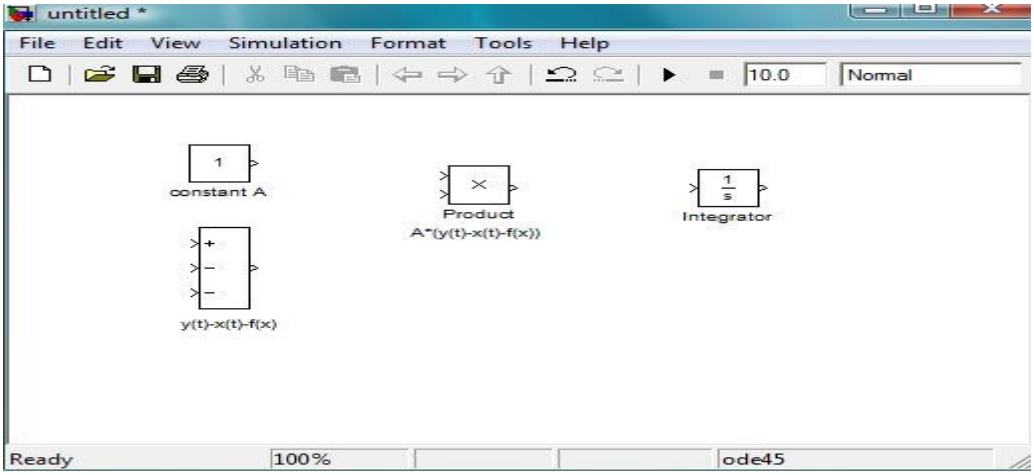
$$y = \int \big(x(t) - t(t) + z(t)\big)dt + y0 \ldots\ldots\ldots (2)$$

$$z = \int -By(t)dt + z0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

$$f(x) = m1x(t) + 0.5(m0 - m1)X(|x(t) + 1| - |x(t) - 1|) \ldots\ldots (4)$$

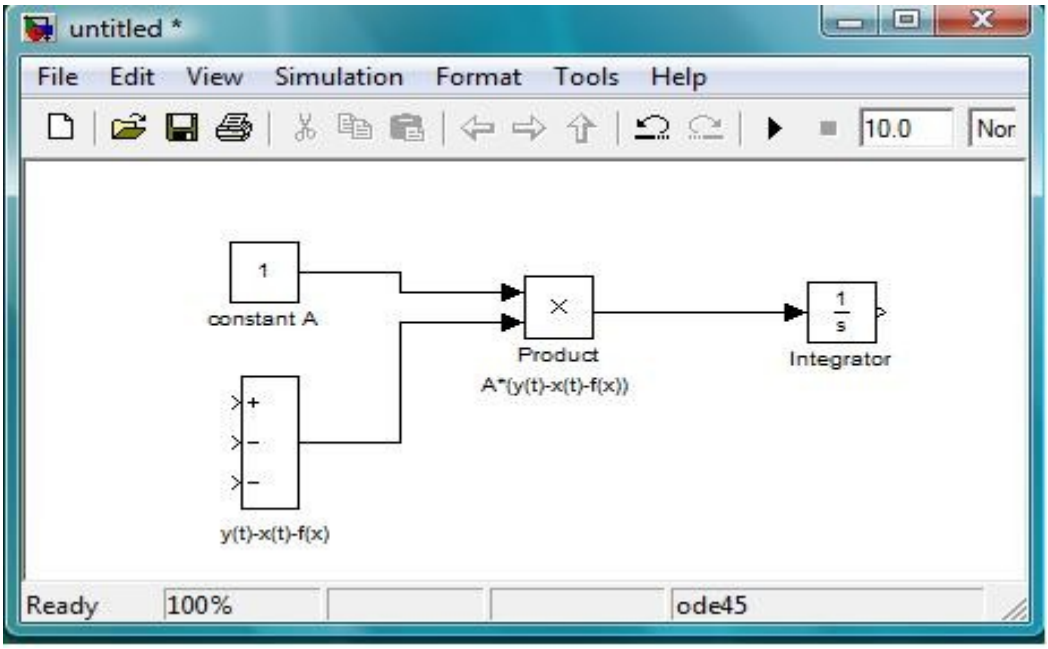
Yukarıdaki daha önce bulduğumuz x,y ve z denklemleri dikkatlice incelenirse bu denklemleri simulink çözebilmemiz için gereken bloklar sırasıyla; A,B,m0 ve m1 için dört tane sabit (constant) bloğu, dx,dy ve dz nin integrallerini alcağımız için üç tane integrator bloğu, bulduğumuz x,y ve z değerlerini matlab'ın workspace bölümüne aktarmak için üç tane To Workspace bloğu,chua diyodunun karakteristiğini hesaplamak için bir tane embedded matlab function bloğu,çarpma işlemlerini yapmak için iki tane Product bloğu ve son olarak toplama işlemlerini yapabilmemiz için iki tane sum bloğudur.

Şimdi ise bu blokları sırasıyla simulink dosyamıza alalım ve denklemleri oluşturmaya başlayalım. İlk olarak x denkleminde başlayacağız; x denklemindeki, y(t)-x(t)-f(x) için bir sum bloğu, A*(y(t)-x(t)-f(x)) durumu için bir çarpma-bölme (product) bloğu ve İntegral almamız için ise bir integrator bloğunu simulink dosyamıza alıp blok bağlantılarının nasıl yapılacağını açıklayalım.



Şekil 3.18. Blokların Ayrık Olarak Simulink Penceresinden Görünümü

Şimdi ise blokların birbirleri ile nasıl bağlanacağına gelelim. Yukarıdaki şekil incelenilecek olursa constant A bloğunun ve sum bloğunun çıkışlarını (>) Product bloğunun iki girişine (<), Product bloğunun çıkışını (>) sum bloğunun girişine (<) bağlayacağız ve bunun için tıkla ve sürükle (click and drag) işlemi kullanacağız. Bu işlemleri gerçekleştirdiğimizde elde edeceğimiz simulink dosyamızın görüntüsü aşağıdaki gibi olacaktır.



Şekil 3.19. Blokların Birbirlerine Bağlanması

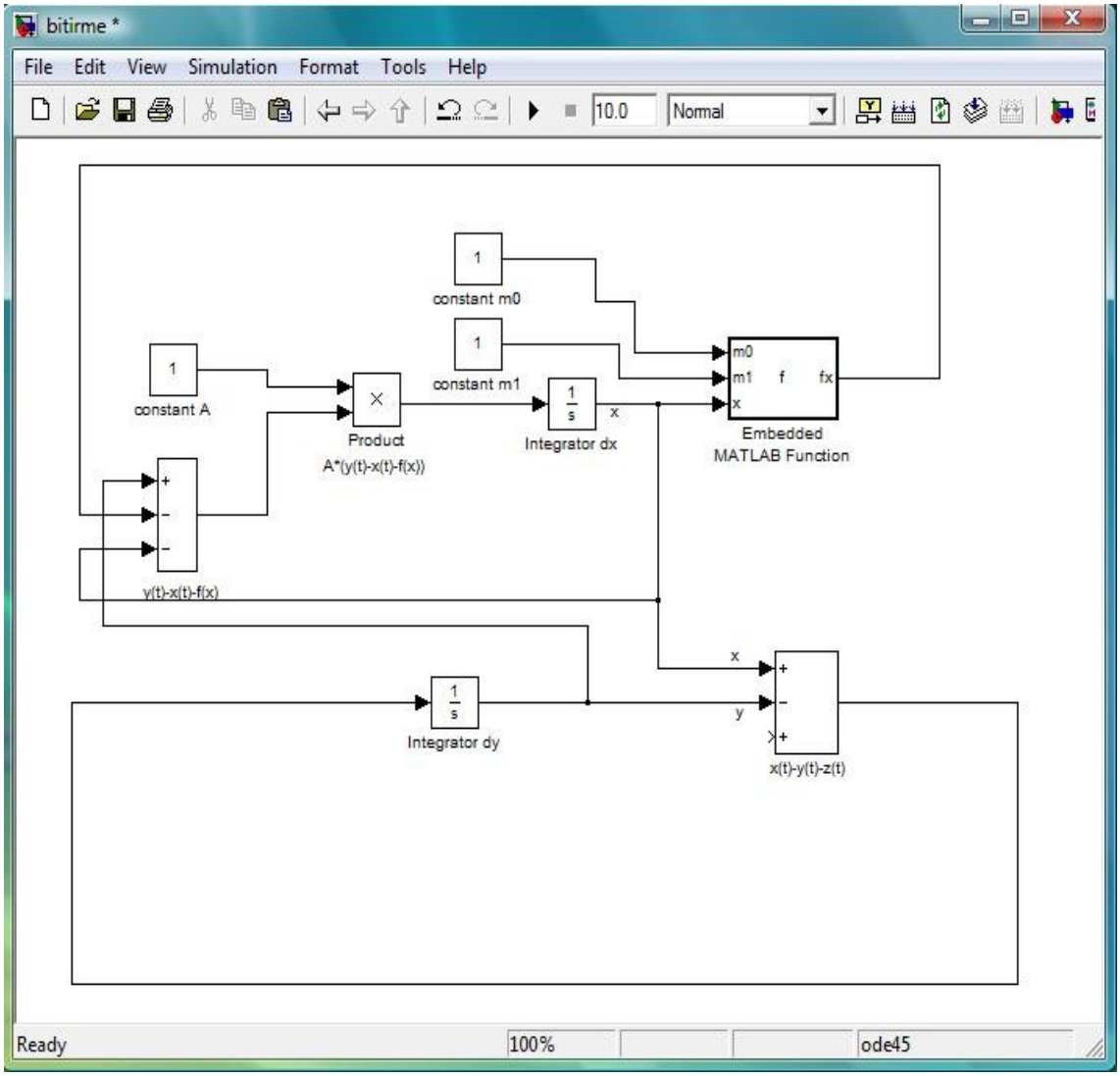
Şekilde gördüğümüz İntegral alıcı bloğun çıkışı x olacaktır, denklemlerden anlaşılacağı üzere bu x diyot karakteristiğini kurmada (embedded matlab function bloğunda) ve y denkleminde kullanılacaktır. Şimdi dosyamıza bir embedded matlab function bloğu, bu bloğun girişine bağlayacağımız iki tane constant bloğu (m0 ve m1 için), y denkleminin içerdiği x(t)-y(t)+z(t) durumu için bir sum bloğu ve dy'in integralini almak için birde integrator bloğu ekleyelim.

Tasarımımızda kullandığımız chua diyodunun karakteristiği,
f(x)=m1.x(t) + 0,5.(m0-m1).(|x(t)+1|-|x(t)-1|) olduğundan f(x) değerinin x=<-1, x>=1 ve 1>x>-1 durumları için üç farklı değeri vardır, bunlarda;
x <= -1 ise f(x) = m1*x - (m0-m1), x >= 1 ise f(x) = m1*x + (m0-m1) ve 1>x>-1 ise f(x)=m0-m1 durumlarıdır, bu durumları karşılamak için embedded matlab function bloğuna girilen matlab kodları aşağıdaki gibidir.

```
function fx = f(m0,m1,x)
if x >= 1
   fx = m1*x + (m0-m1);
elseif x <= -1
   fx = m1*x - (m0-m1);
```

```
else
    fx = m0-m1;
end
```

Bu kodlar embedded matlab function bloğuna girildiğinde ve yukarda bahsettiğimiz blokları simulink dosyamıza eklediğimizde elde edeceğimiz mdl dosyamız aşağıdaki gibi görünecektir.

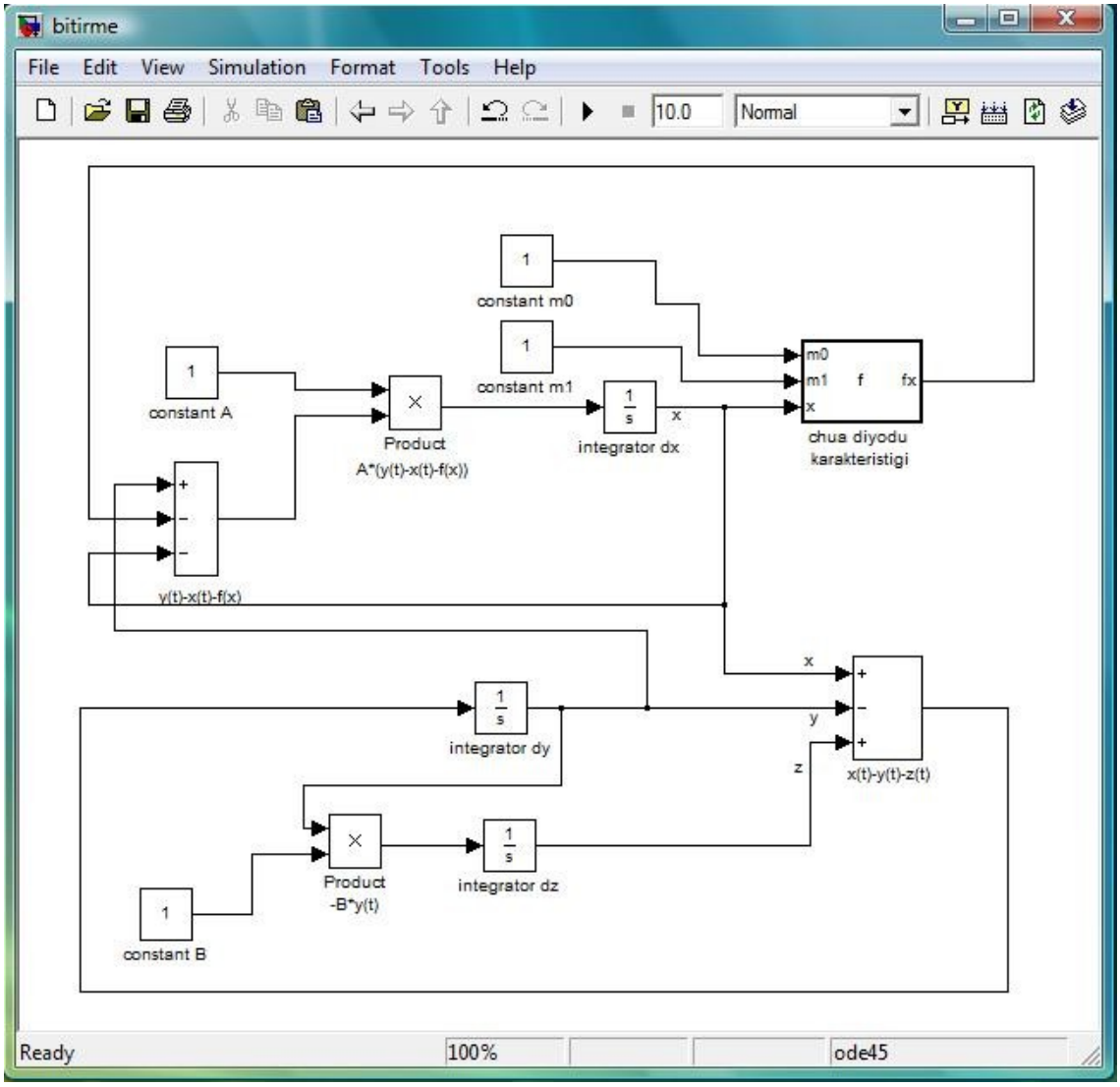


Şekil 3.20. Blokların Birbirlerine Bağlanması

Yukarıdaki şekil dikkatlice incelendiğinde x denkleminin tasarımının bitmiş olduğu, y denklemi için z girişinin eksik olduğu ve z denkleminin yapılmak üzere olduğu görülmektedir.

Şimdi ise simulink dosyamıza, z denklemini tasarlamak için gerekli olan B sabiti için bir constant bloğu, B ile y(t) nin birbiri ile çarpılması için gerekli olan bir çarpma-bölme (product) bloğu ve bu çarpının integralini almak için bir integrator bloğu ekleyelim.

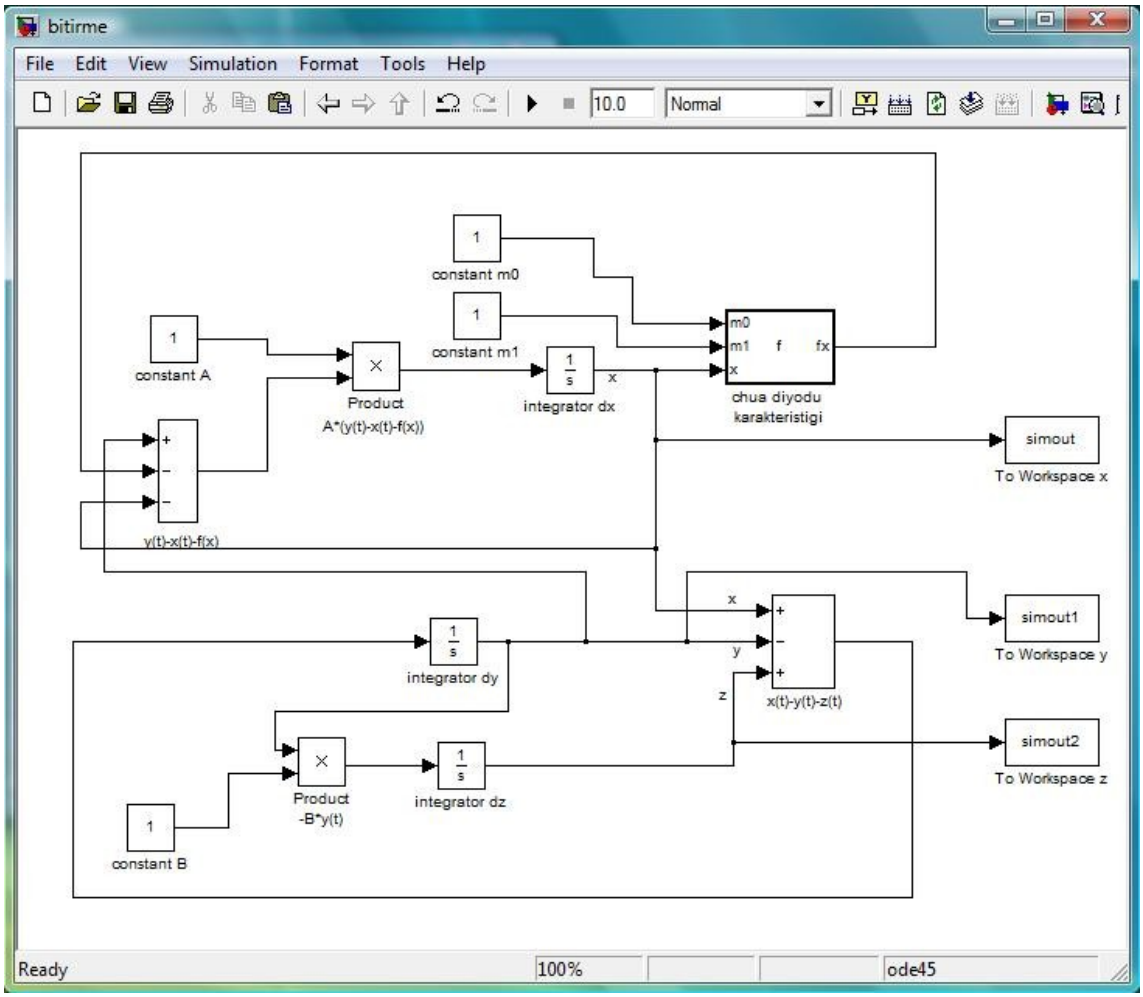
Simulink dosyamızın yeni hali aşağıdaki şekilde görüldüğü gibi olacaktır.



Şekil 3.21. Blokların Birbirlerine Bağlanması

Şimdiye kadar yaptıklarımızla chua devremizi tasarlamış olduk bundan sonra yapacaklarımız ise gerekli değerleri girip To Workspace bloğu aracılığıyla x,y ve z denklemlerinin sonuçlarını MATLAB workspace'ye aktarıp bu değerlerle x,y ve z denklemlerine ait grafikleri elde etmek olacaktır. Şimdi simulink dosyamıza x,y ve z için birer tane To Workspace bloğu ekleyerek tasarımımızın sonuna bir adım daha

yaklaşmış olalım. To Workspace bloklarını simulink dosyamıza ekleyip gerekli bağlantıları yaptıktan sonra elde edeceğimiz simulink mdl dosyamızın görüntüsü aşağıdaki gibi olacaktır.



Şekil 3.22. Blokların Birbirlerine Bağlanması Nihai Hal

Geriye sadece x,y ve z denklemlerimizin, başlangıç koşulları, A, B, m0 ve m1 durum değişkenlerini girip tasarımımızın sonuçlarını incelemek kaldı. Şimdi ilk bölümde ele almış olduğumuz;

x0 = 1 (x denkleminin başlangıç şartı)

y0 = -0.009623353 (y denkleminin başlangıç şartı)

z0 = -1.90999 (z denkleminin başlangıç şartı)

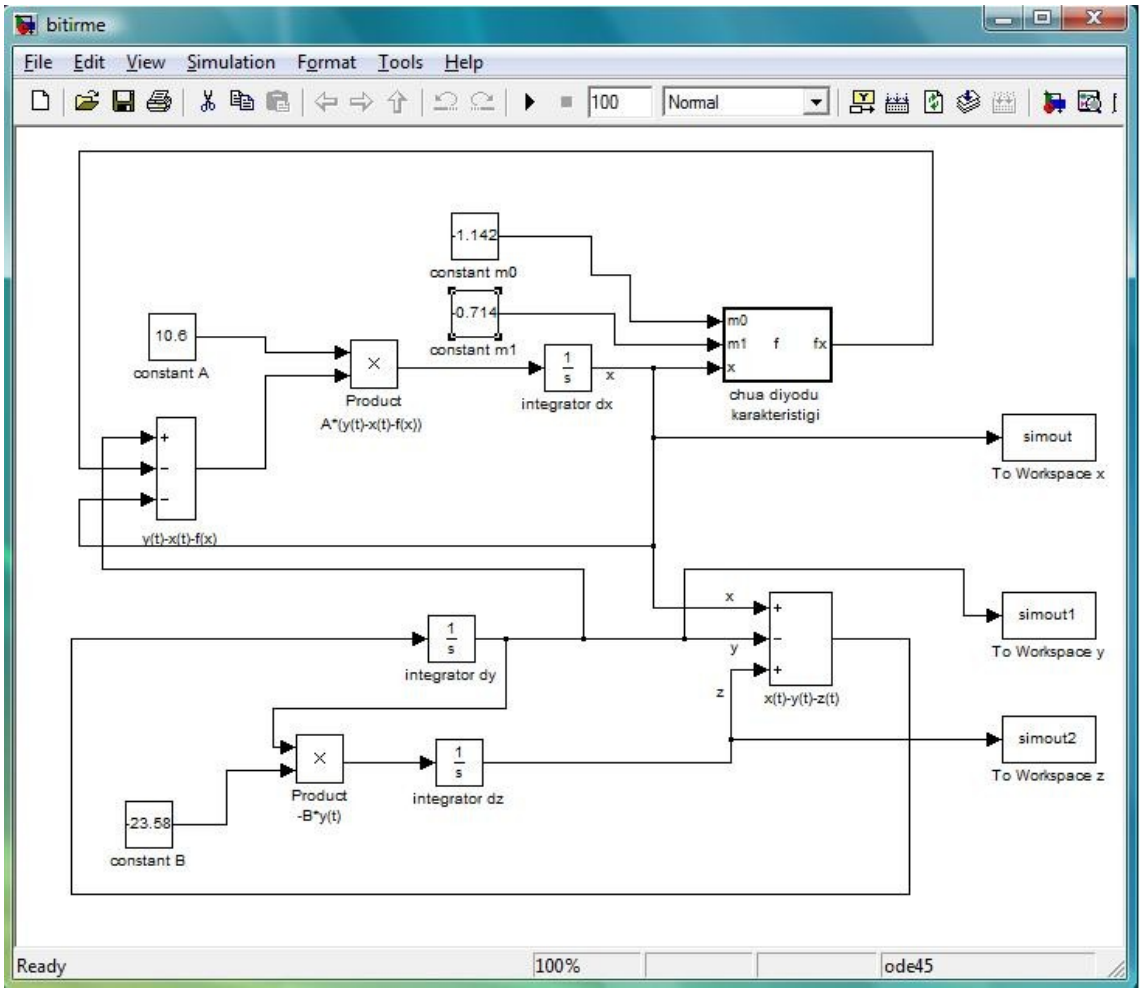
A = 10.6

B = -23.58

m0 = -1.142

m1 = -0.714

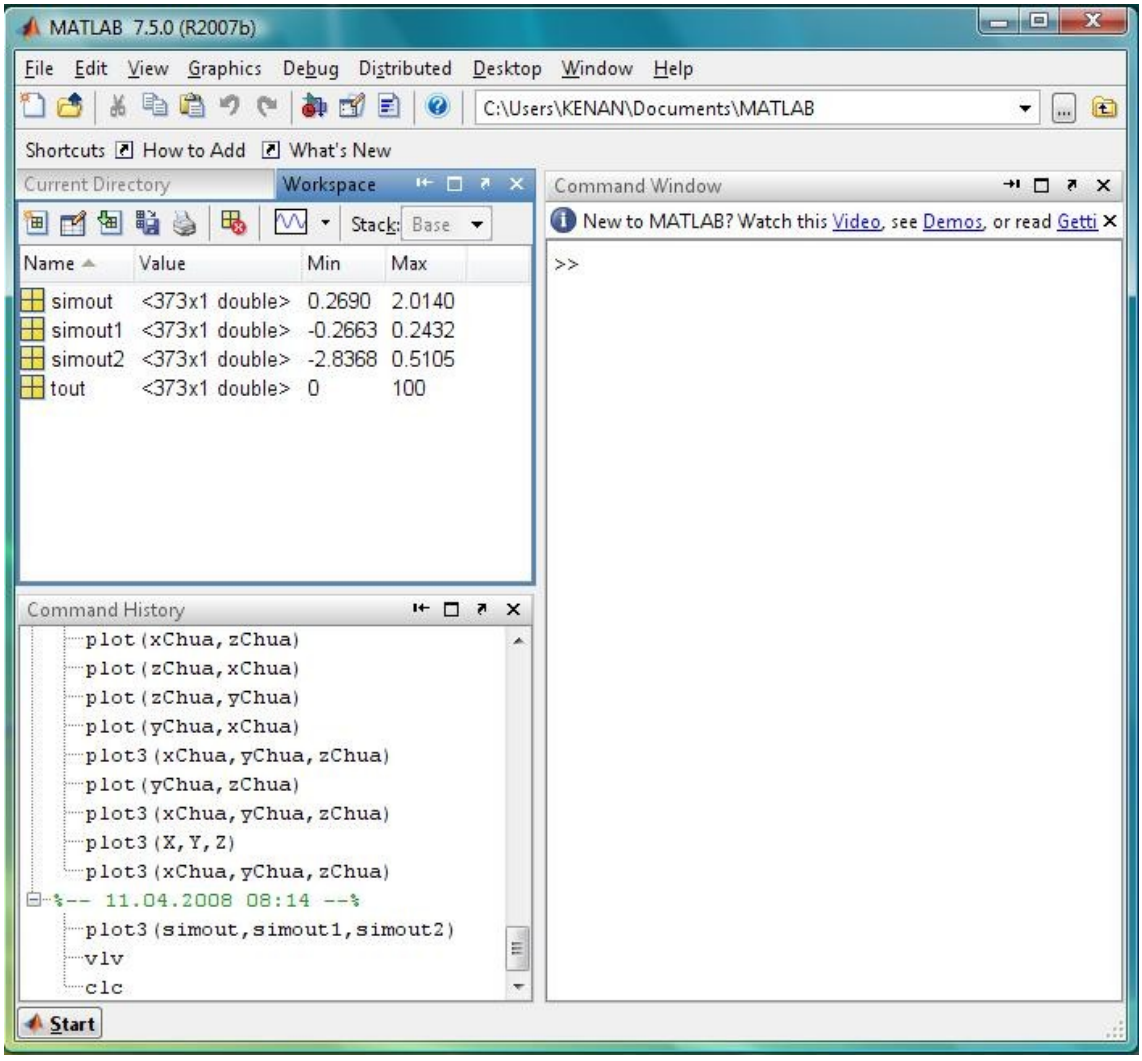
değerlerini yukarıdaki şekildeki uygun bloklara girdiğimizde simulink dosyamızın görünümü aşağıdaki gibi olacaktır.



Şekil 3.23. Blokların Birbirlerine Bağlanması Nihai Hal

To Workspace bloklarındaki save format parametrelerini Array olarak kaydettikten sonra simulink dosyamızı çalıştıralım. Simulink dosyasını çalıştırmak için durum çubuğunda bulunan ok butonuna basmamız gerekmektedir. Simülasyon bittikten sonra simulink, MATLAB'in workspace kısmına simout, simout1, simout2 ve tout olmak üzere 4 tane dizi gönderecektir. Bu diziler yukarıdaki şekle bakıldığı zaman anlaşıldığı gibi simout dizisi x denkleminin, simout1 y denkleminin ve simout2 z denkleminin sonuçlarını içermektedir, tout ise zaman dizisidir ki simout, simout1 ve simout2 dizilerinin içerdiği eleman sayısı kadar eleman içermektedir. Simülasyon

bittikten sonra MATLAB workspace'nin görünümü aşağıdaki şekilde görüldüğü gibi olacaktır.



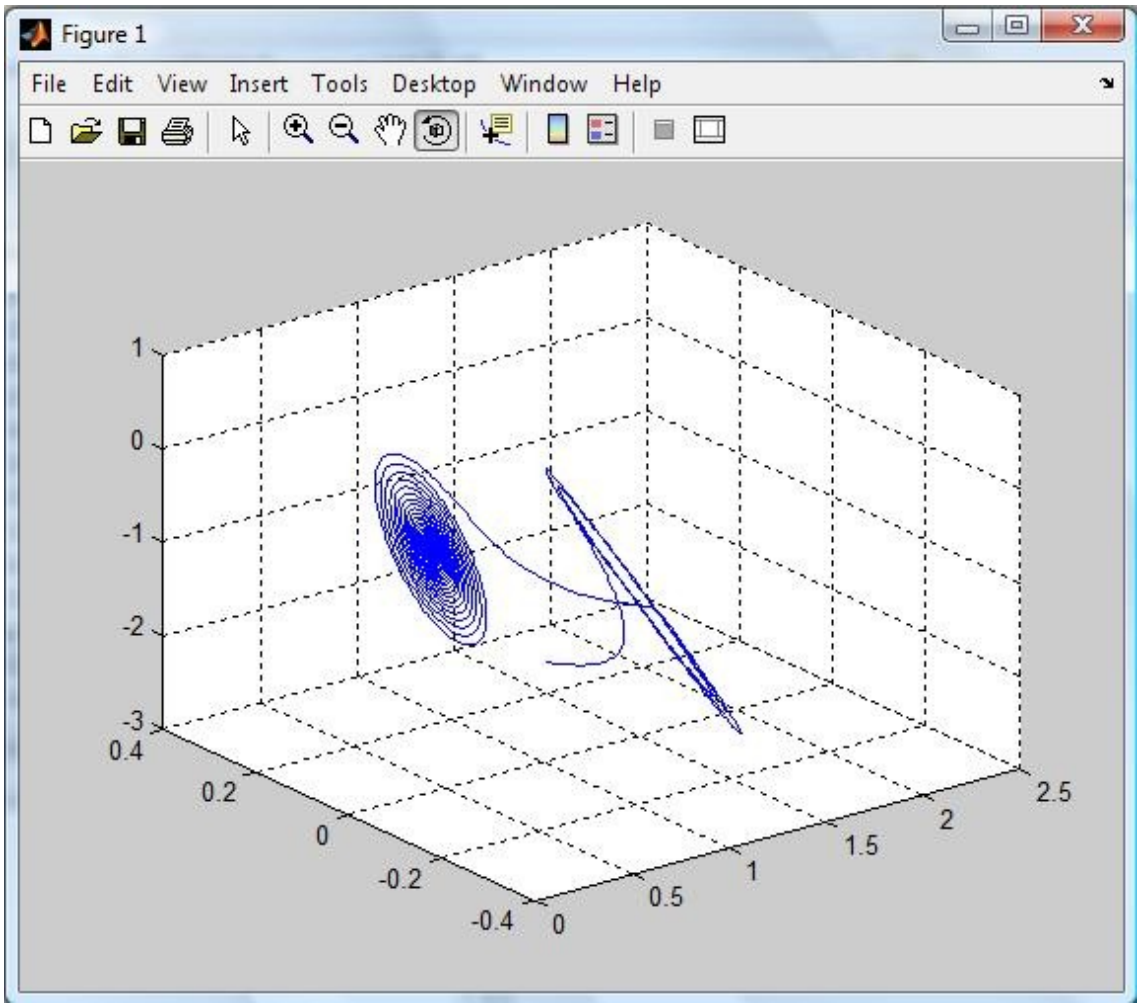
Şekil 3.24. Simülasyon Sonuçlarının Workspace'ye Aktarılması

Tasarımımızda şimdiye kadar, kullanacağımız blokları tanıdık, bu blokların MATLAB simulink ortamında nasıl kullanılması gerektiğinden bahsettik ve son olarak da MATLAB simulink ortamından MATLAB workspace'ye nasıl veri aktaracağımızı gösterdik, şimdi ise MATLAB workspace ortamına aktardığımız bu simout, simout1 ve simout2 dizilerini yani simülasyon sonuçlarımızı nasıl grafikleştireceğimizi anlatalım.

3.1.3. Simülasyon Sonuçlarının Grafiklere Dökülmesi

Yukarıda, şekil 4.18'de, görüldüğü gibi x,y ve z denklem değerlerimizi matlab workspace'ye aktarmış olduk, bu x, y ve z fazlarını grafiklendirmek için iki tane matlab komutu kullanacağız bunlarda plot ve plot3 komutlarıdır.

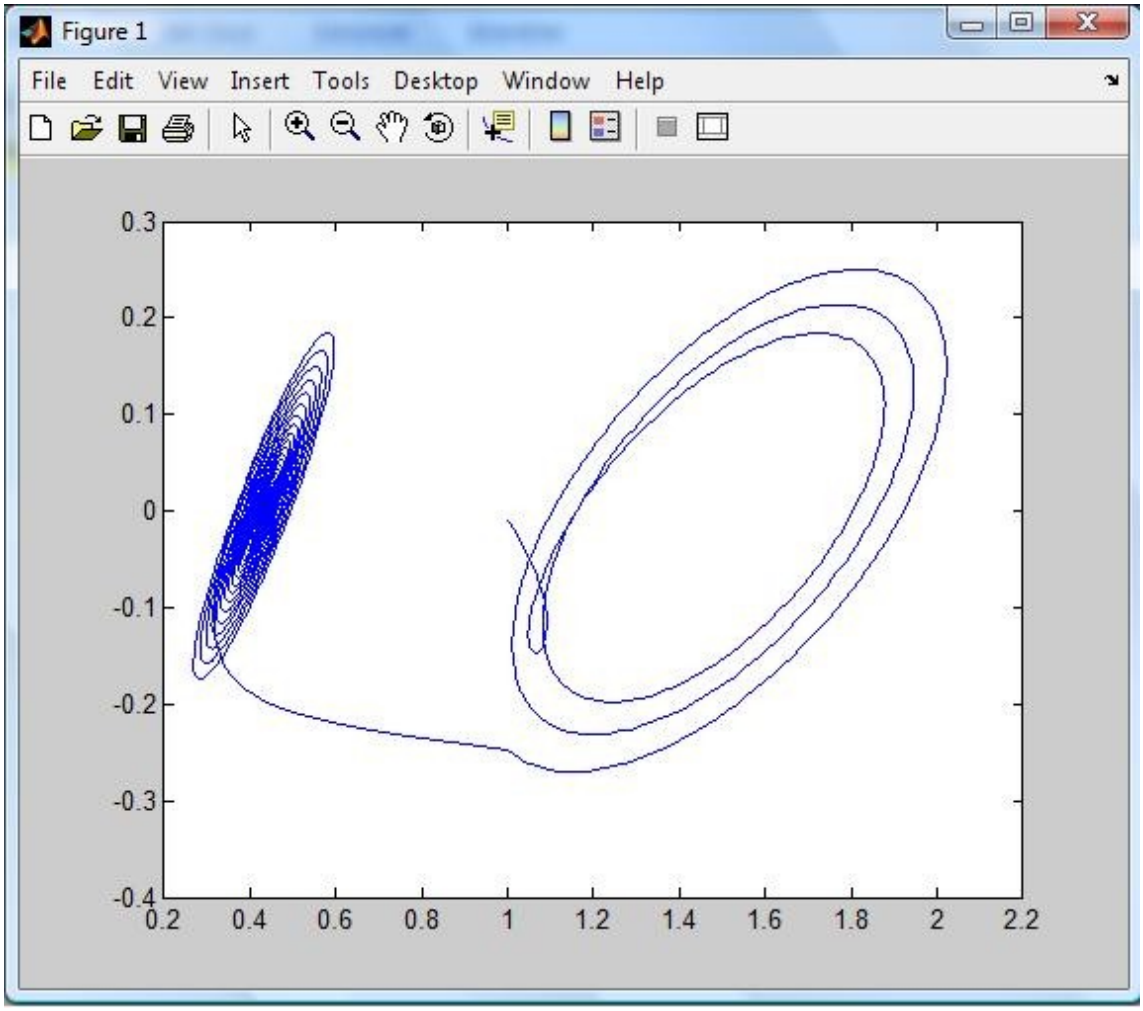
Plot komutu iki boyutlu (2-D), yani x, y ve z fazlarının zamana göre ve kendi aralarında ikişerli grafikler için, plot3 komutunu ise üç boyutlu (3-D) grafikleri çizdirmek için kullanacağız. Simülasyonumuzu çalıştırdıktan sonra Matlab command window ortamına plot3(simout, simout1, simout) komutunu yazdığımızda aşağıdaki grafiği elde ederiz.



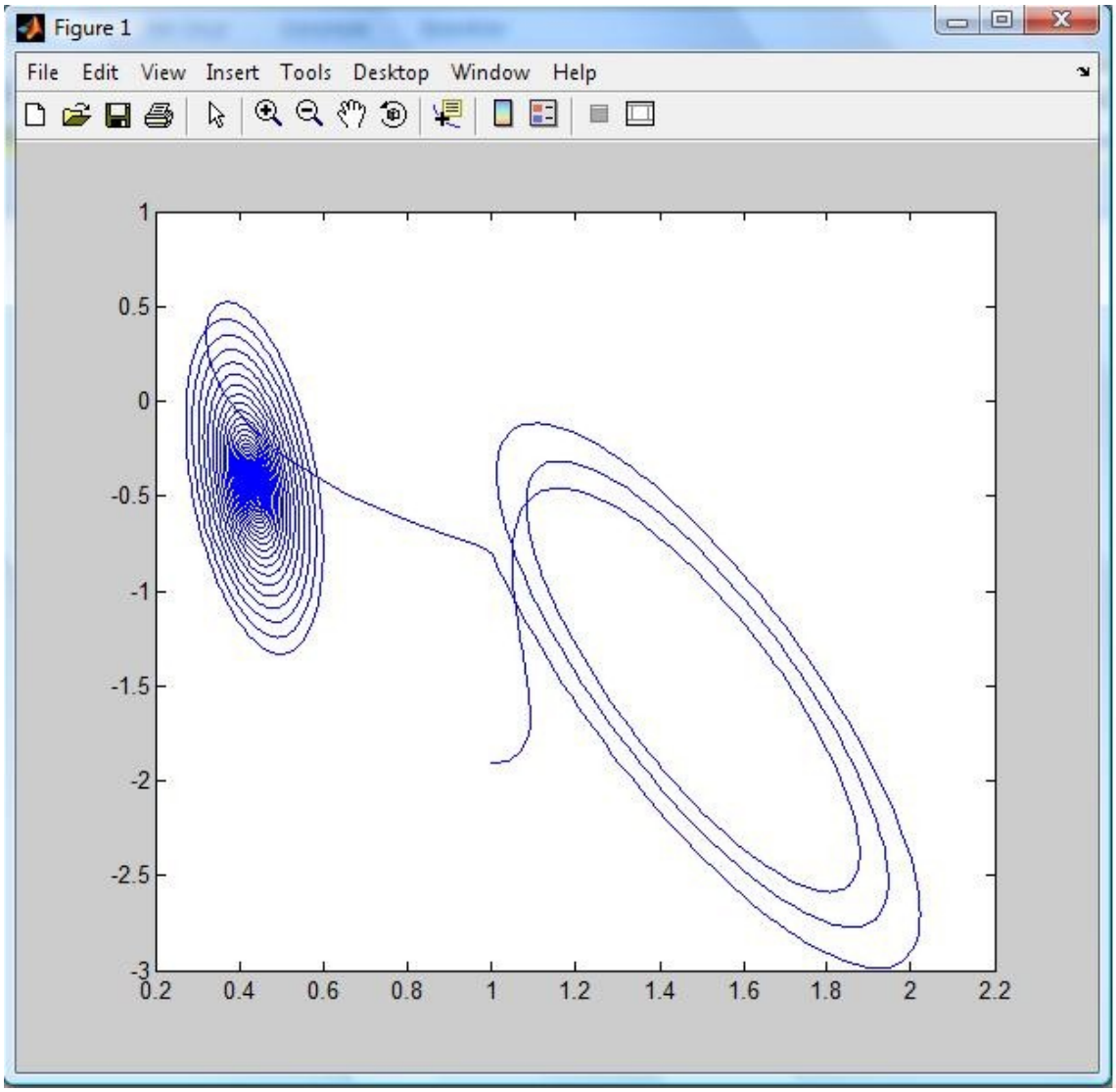
Şekil 3.25. (X,Y,Z) Grafiği

x fazının y fazına göre, x fazının z fazına göre ve y fazının z fazına göre grafikleri için, sırasıyla plot(simout,simout1), plot(simout,simout2) ve plot(simout1,simout2)

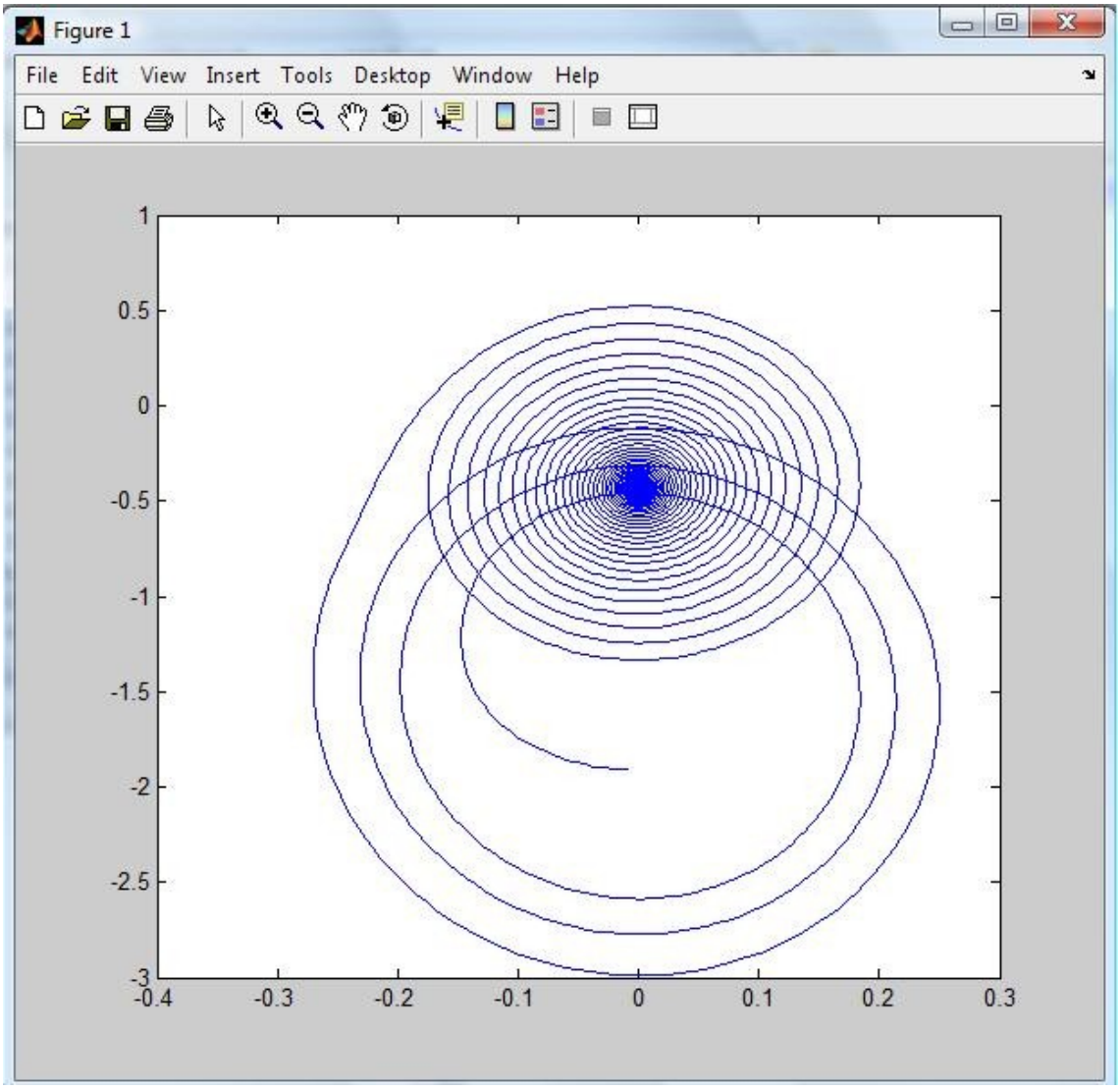
komutlarını matlab command window ortamına yazdığımızda aşağıdaki Şekil 4.20, Şekil 4.21 ve Şekil 4.22 grafiklerini elde ederiz.



Şekil 3.26. Plot(Simout,Simout1), X Fazı İle Y Fazı Arasındaki İlişki

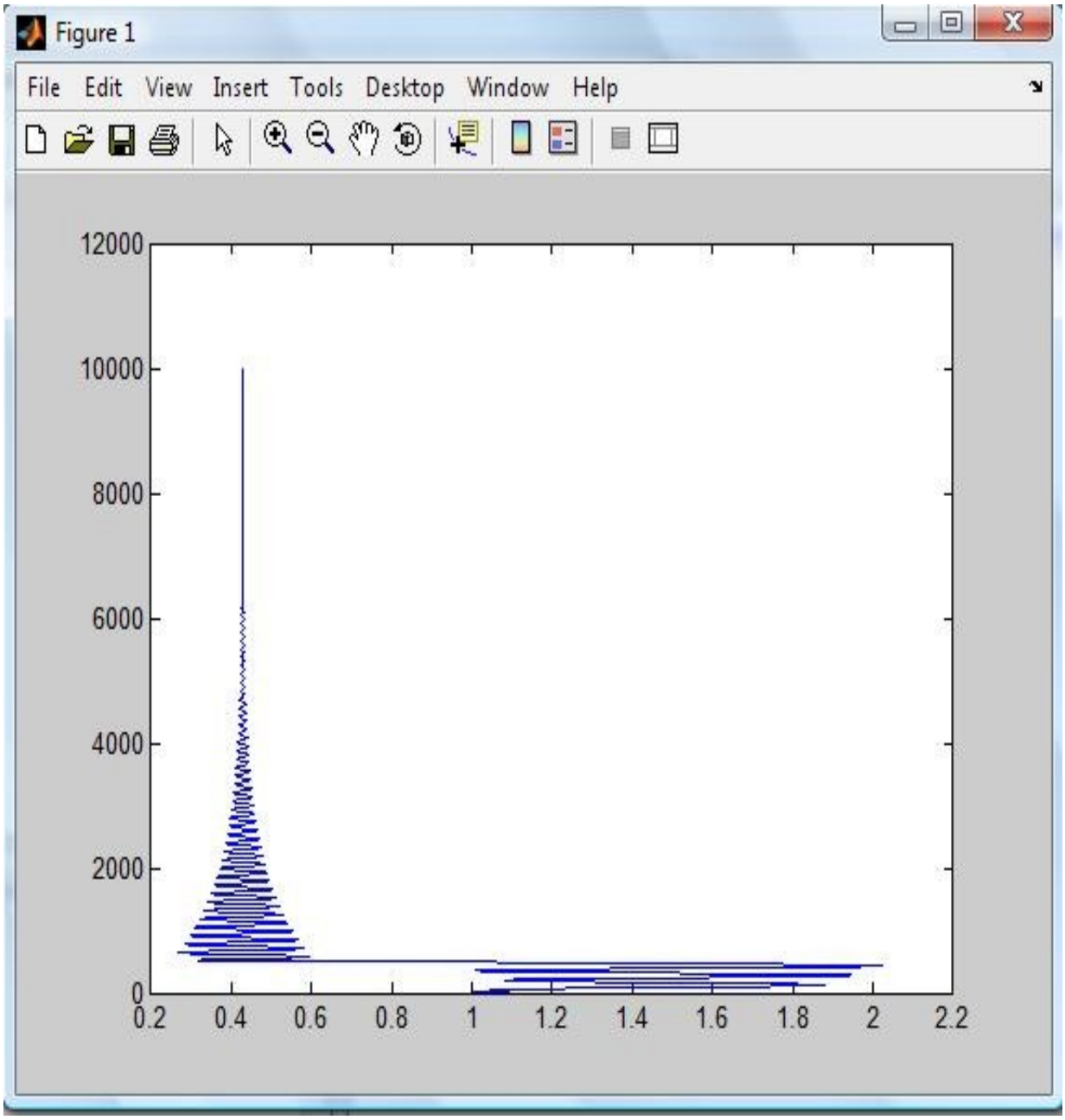


Şekil 3.27. Plot(Simout,Simout2), X Fazı İle Z Fazı Arasındaki İlişki

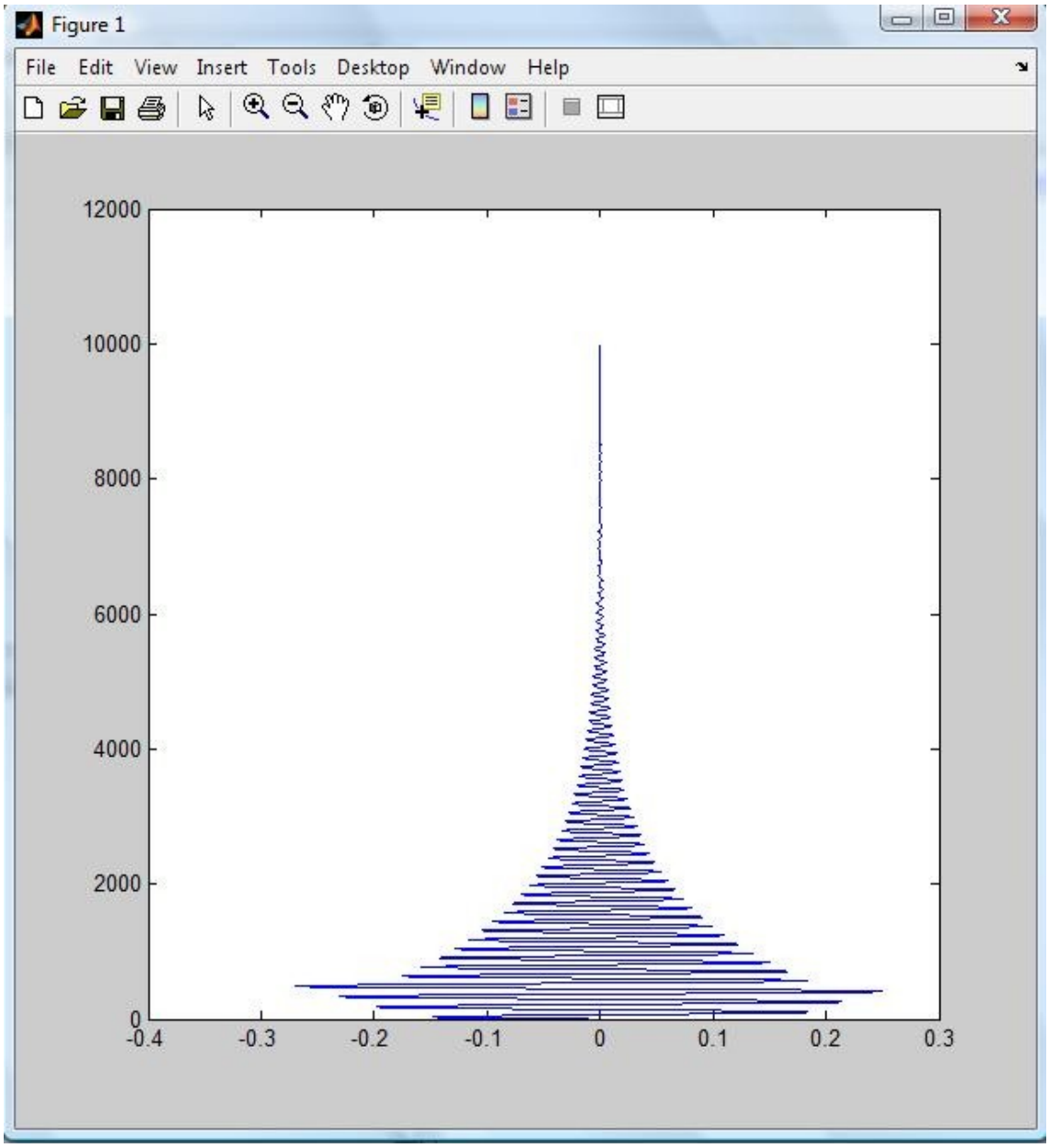


Şekil 3.28. Plot(Simout1,Simout2), Y Fazı İle Z Fazı Arasındaki İlişki

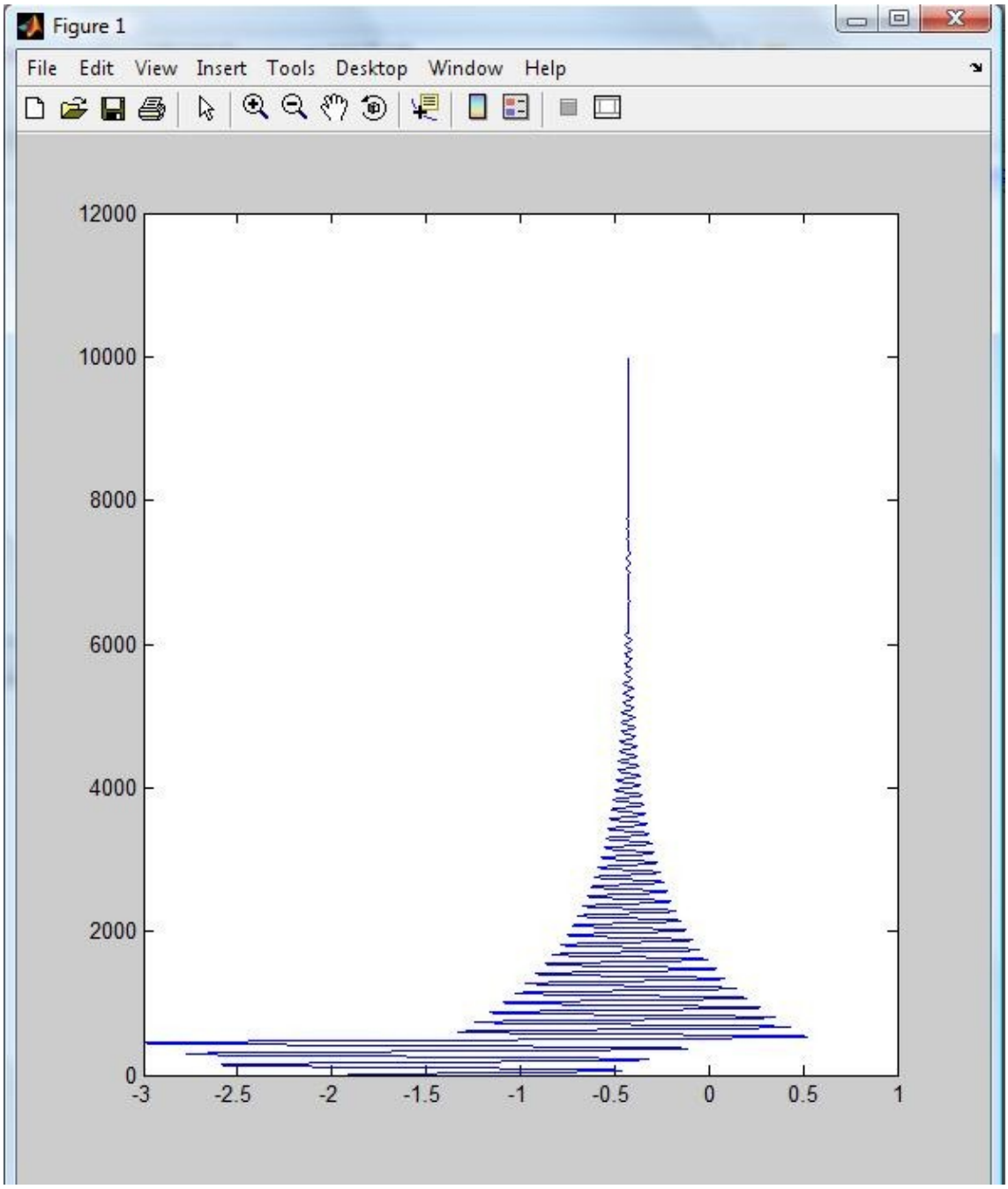
Şimdi ise x fazının zamana göre, y fazının zamana göre ve z fazının zamana göre grafikleri için, sırasıyla plot(simout,tout), plot(simout,tout) ve plot(simout1,tout) komutlarını matlab command window ortamına yazdığımızda aşağıdaki Şekil 4.23, Şekil 4.24 ve Şekil 4.25 grafiklerini elde ederiz.



Şekil 3.29. Plot(X,Tout), X Fazı İle Zaman Arasındaki İlişki



Şekil 3.30. Plot(Y,Tout), Y Fazı İle Zaman Arasındaki İlişki



Şekil 3.31. Plot(Z,Tout), Z Fazı İle Zaman Arasındaki İlişki

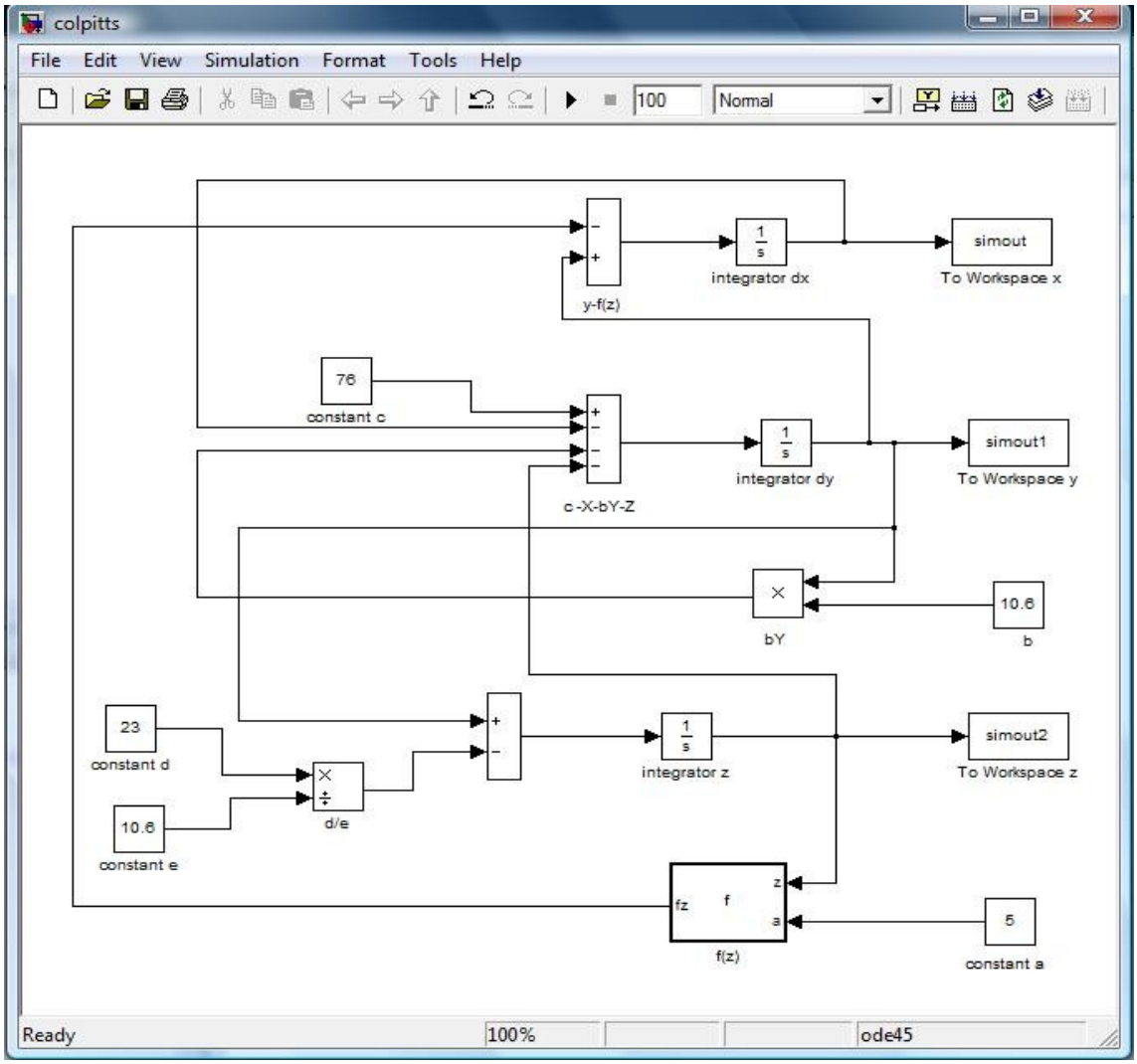
3.2. Colpitts Kaotik Osilatörü

Colpitts dinamik denklemleri c,b,d ve e sistem parametreleri olmak üzere x, y *ve* z durum değişkenleridir.

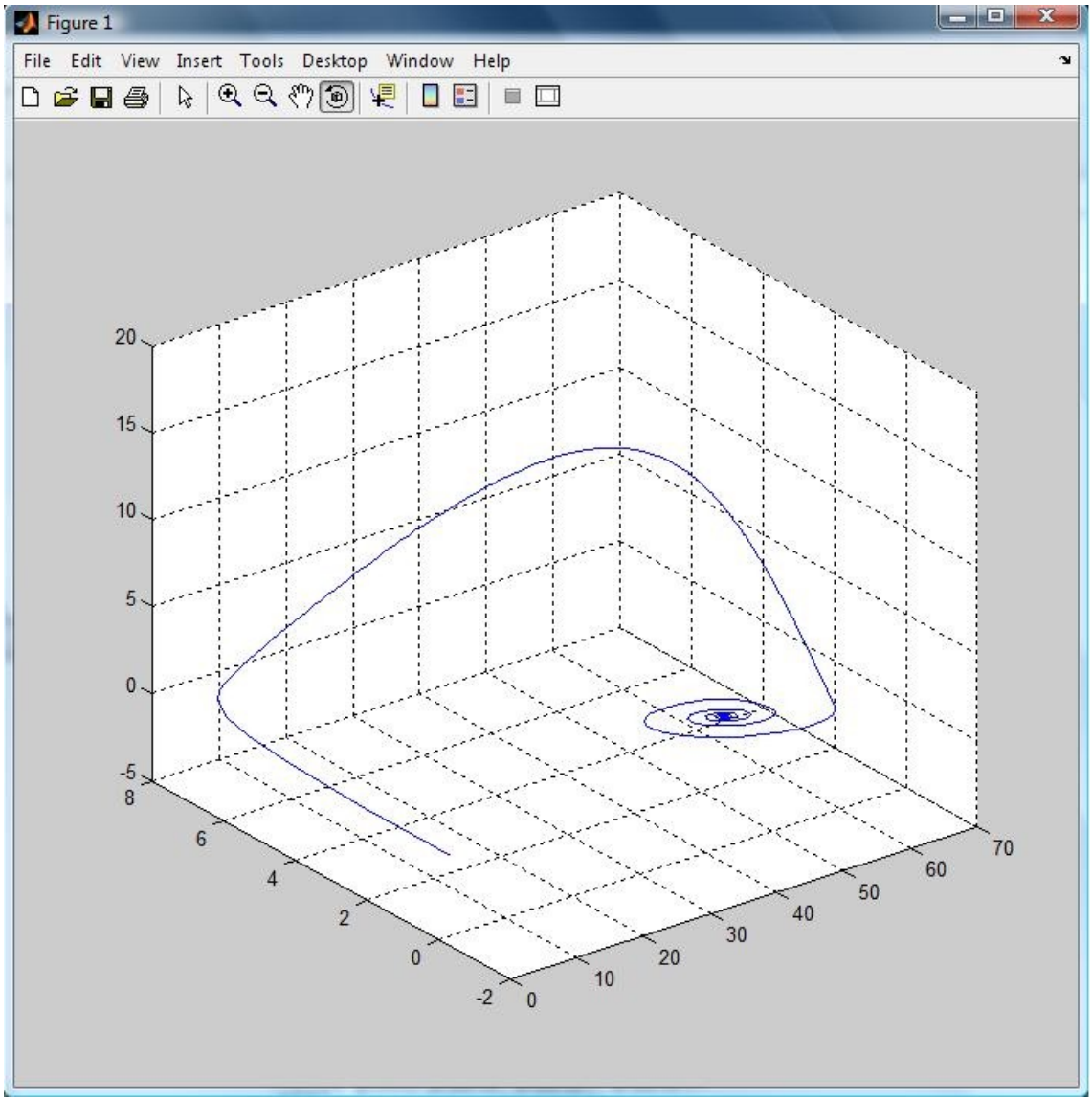
$$\frac{dx}{dt} = y - f(z)$$

$$\frac{dy}{dt} = c - x - by - z$$

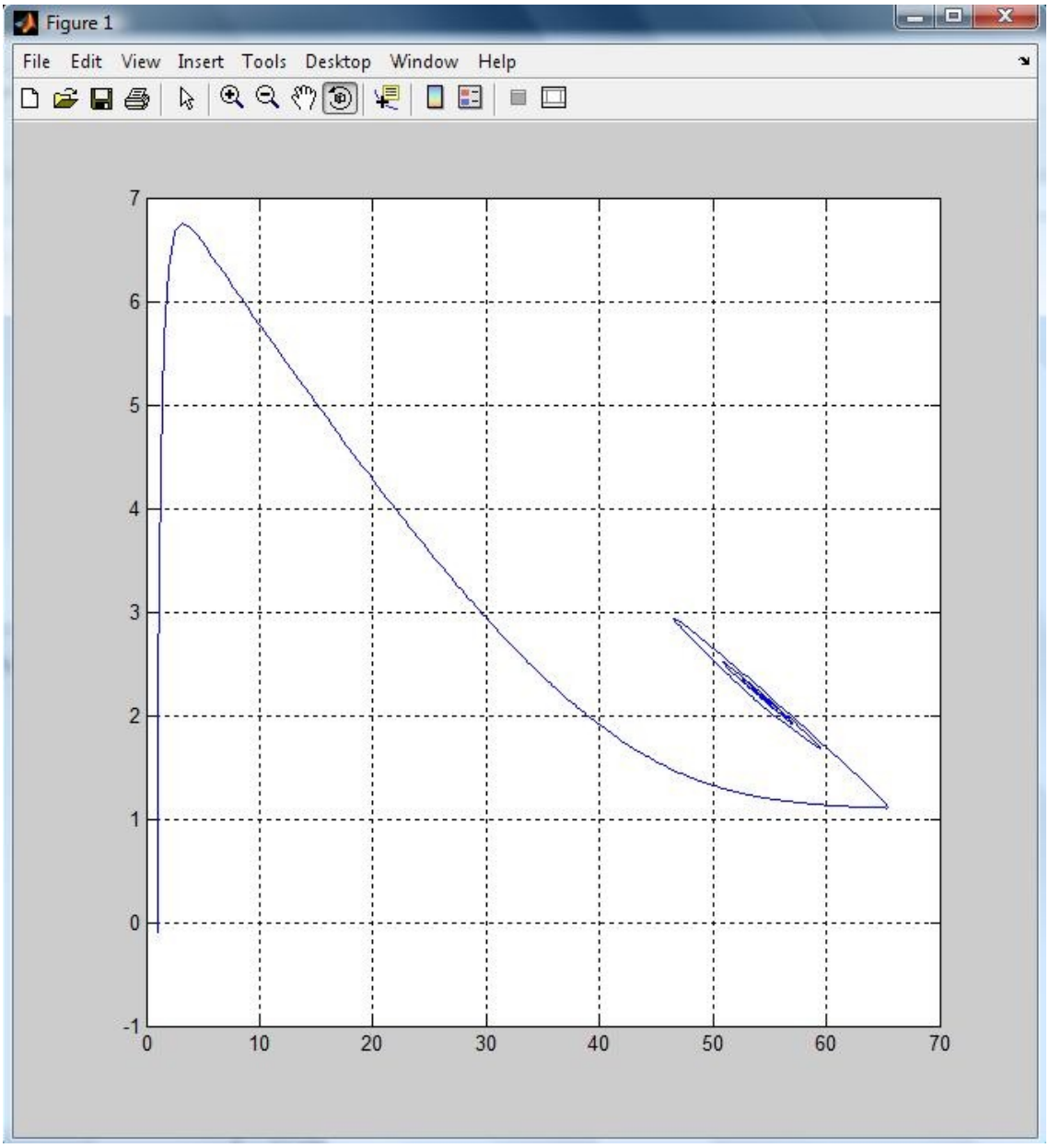
$$\frac{dz}{dt} = y - d/e$$



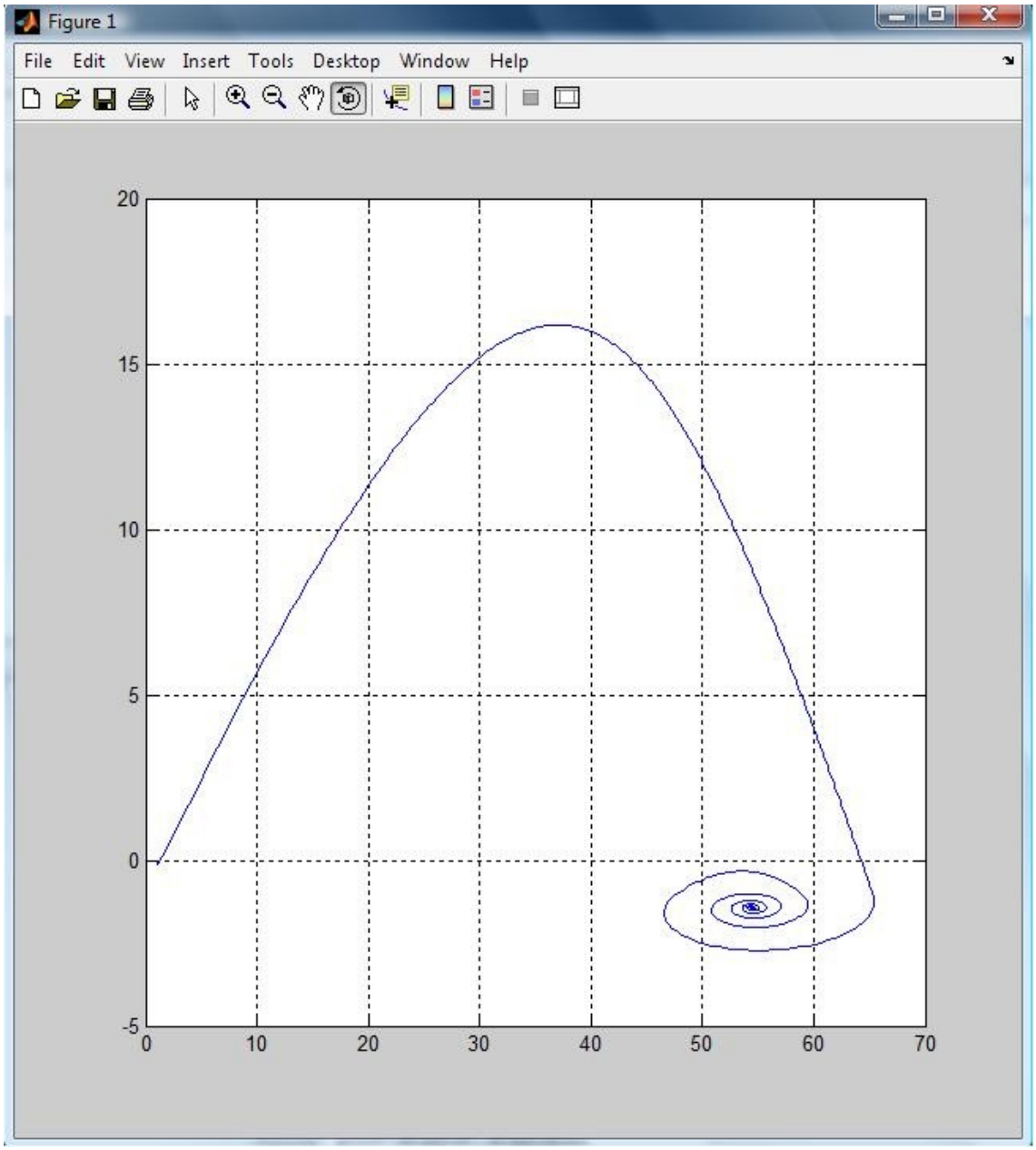
Şekil 3.32. MATLAB Simulink Colpitts Kaotik Osilatörü



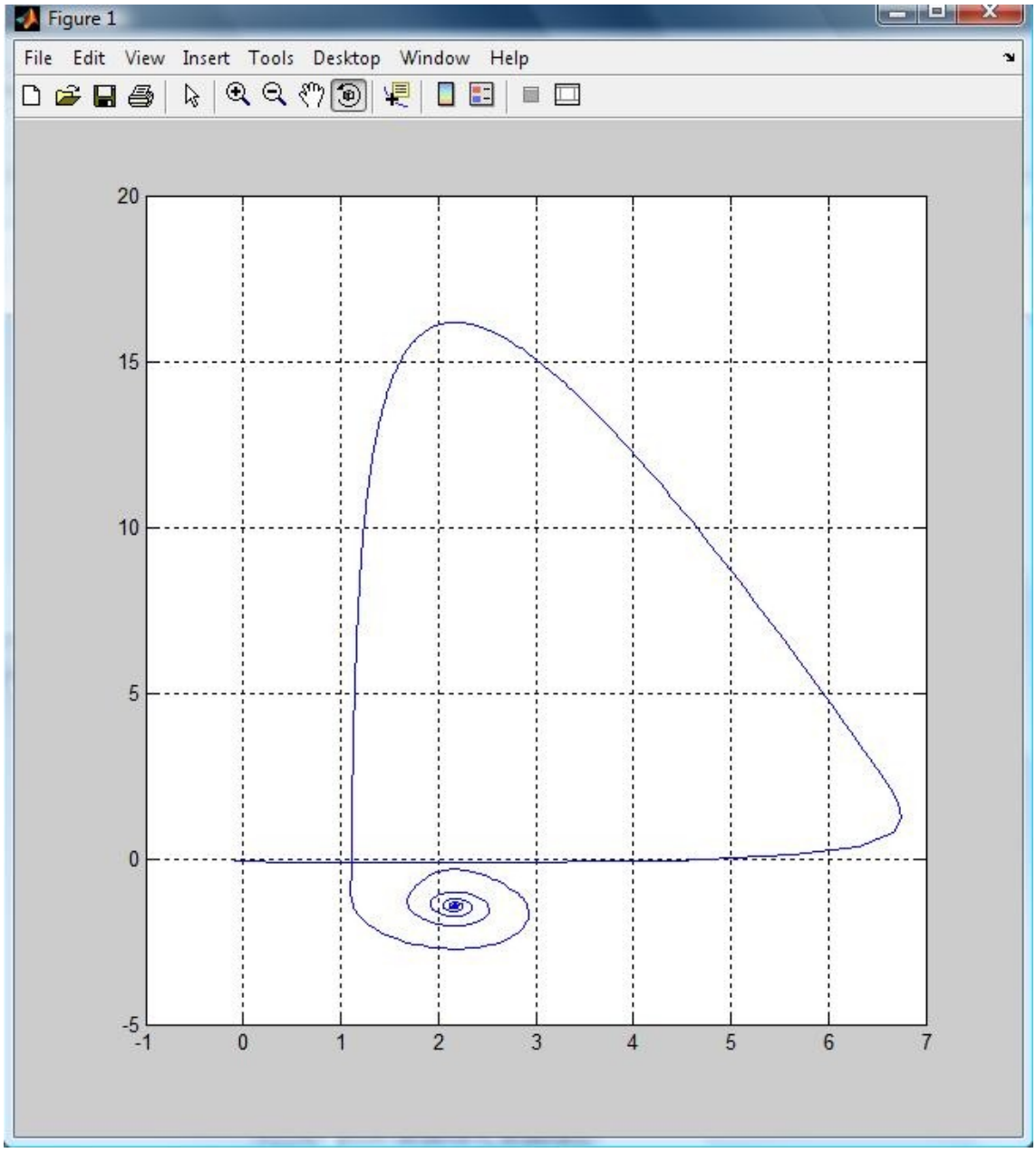
Şekil 3.33. (X, Y, Z) Grafiği, Plot3(Simout, Simout1, Simout2)



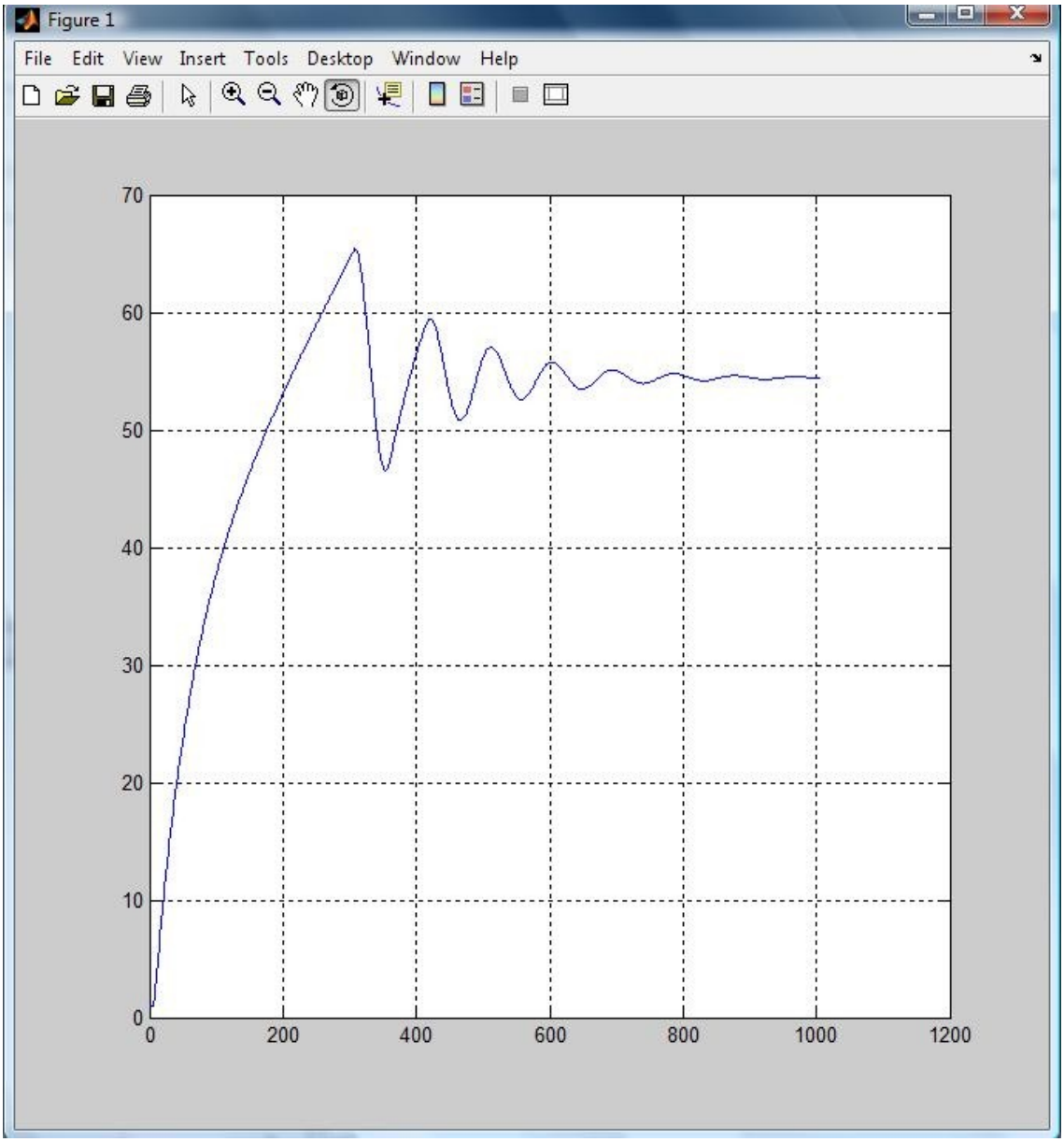
Şekil 3.34. (X, Y) Grafiği, Plot(Simout,Simout1)



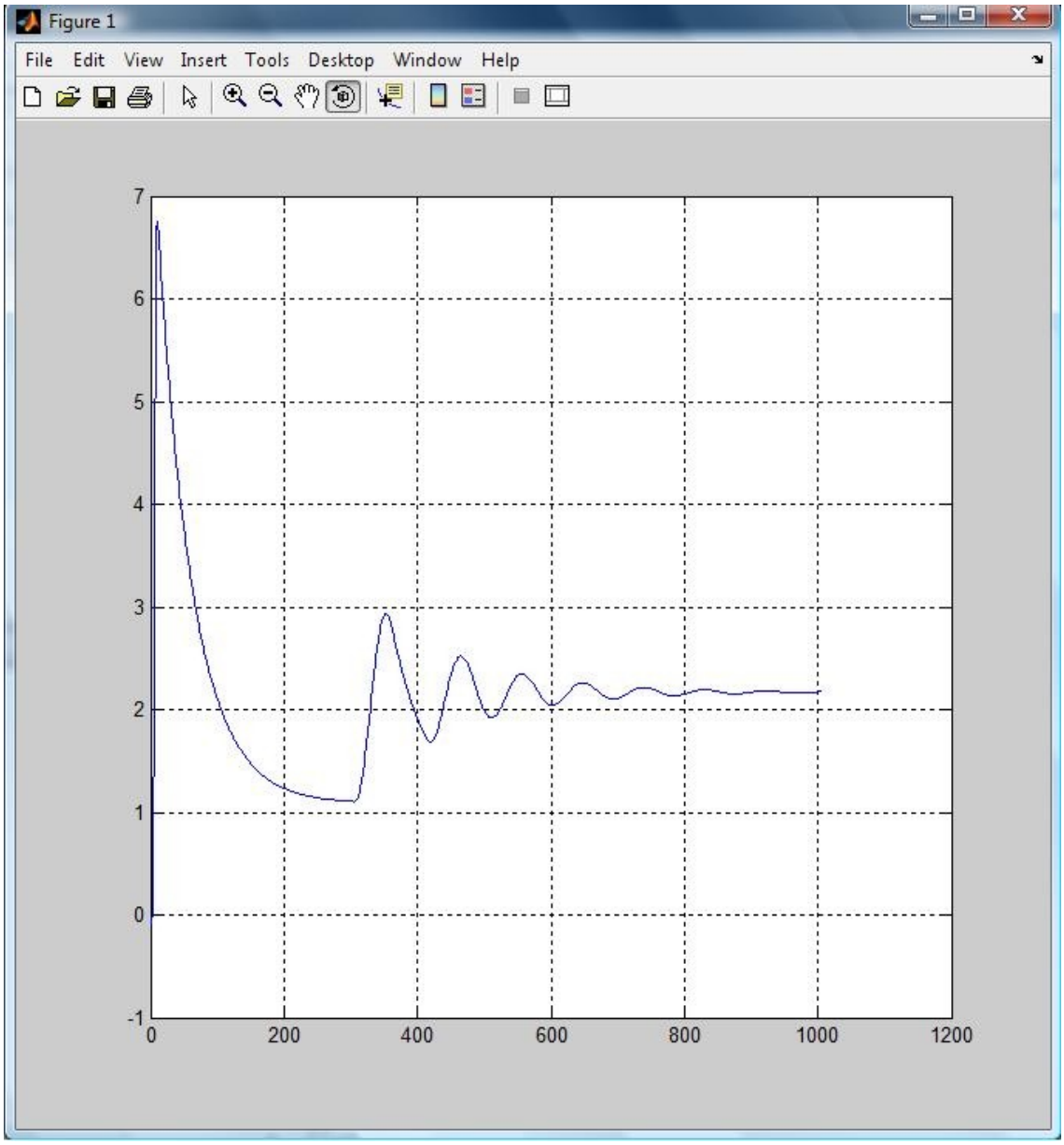
Şekil 3.35. (X, Z) Grafiği, Plot(simout,simout2)



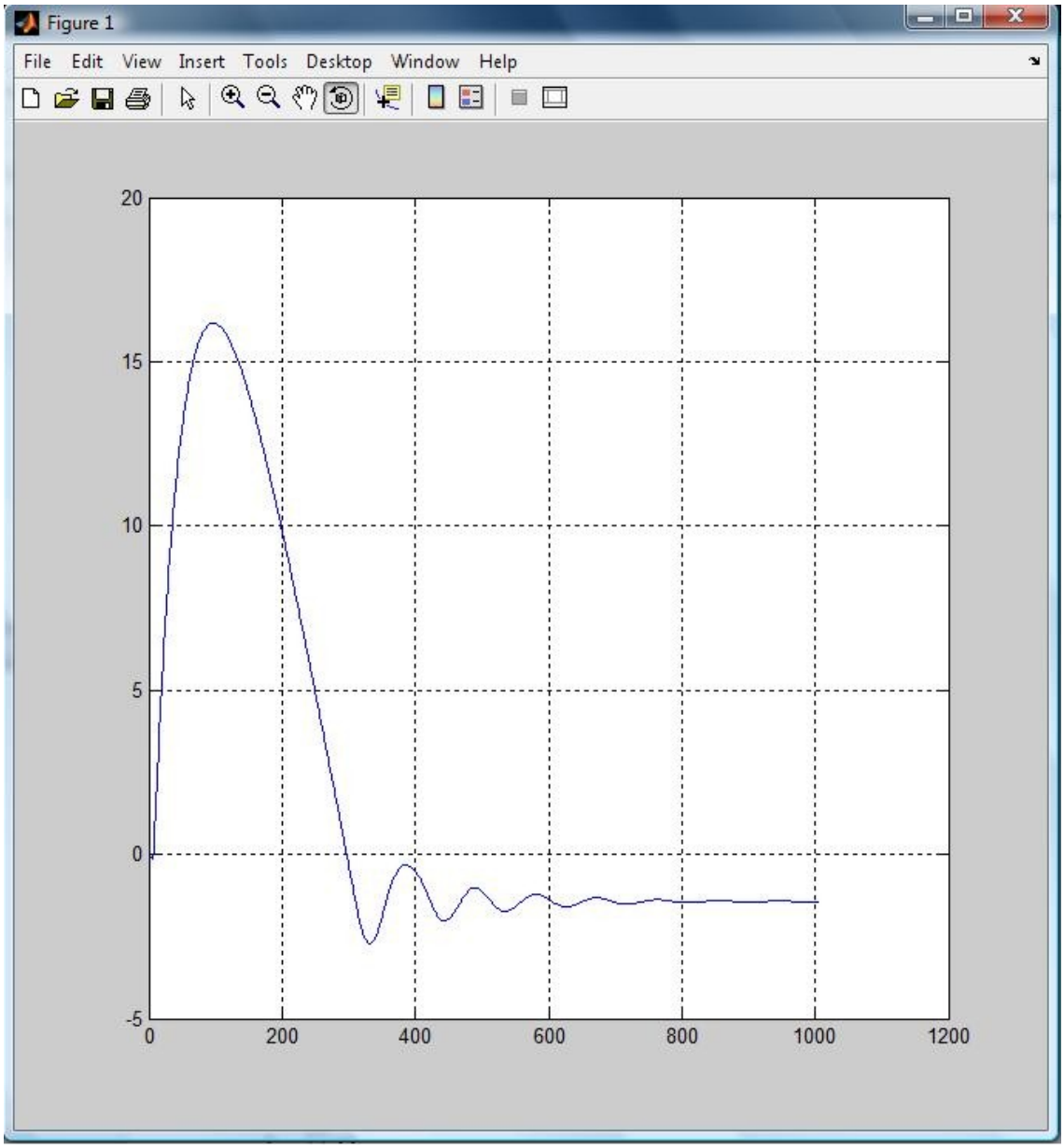
Şekil 3.36. (Y, Z) Grafiği, Plot(Simout1,Simout2)



Şekil 3.37. (X, T) Grafiği, Plot(T,Simout)



Şekil 3.38. (Y, T) Grafiği, Plot(T,Simout1)



Şekil 3.39. (Z, T) Grafiği, Plot(T,Simout1)

3.3. Lorenz Kaotik Osilatörü

1963 yılında, havanın basitleştirilmiş bir modelini çalışmak üzere basit bir matematiksel bilgisayar programı yazan meteorolog Edward Lorenz, bir hava akımının güneş tarafından ısıtıldıkça nasıl azalıp çoğalacağına ilişkin bir model üzerinde çalışıyordu. Lorenz' in yazdığı bilgisayar kodları hava akımlarının akışlarını düzenleyen matematiksel formülleri içermekteydi.[39] Bilgisayar kodu tamamen gerekirci özellikte olduğundan Lorenz, aynı başlangıç koşulları verildiği

takdirde, programın çalıştırılması sonucu hep aynı sonuçları almayı bekliyordu. Fakat aynı zannettiği başlangıç değerlerini girdiği zaman, her seferinde kökten farklı sonuçlar elde ettiğini görmek Lorenz' i şaşkına çevirmişti. Daha dikkatli bir inceleme yaptığında her seferinde tamamen aynı değerleri değil, birbirinden hafifçe farklı değerleri girmiş olduğunu fark etti.[40] Her deneme sırasındaki başlangıç değerlerinin farklı olduğunu anlayamamıştı, çünkü faklılıklar, alışılmış standartlara göre mikroskobik ve önemsiz addedilecek kadar inanılmaz düzeyde küçük farklılıklar vardı[41].

Lorenz' in atmosfer modelinde kullandığı matematik 1970'lerde geniş bir biçimde araştırıldı[42]. Zamanla, kaotik bir sistemin temel özelliği olarak, iki farklı başlangıç koşulları dizgesindeki düşünülebilecek en küçük farklılığın, daima, sonraki veya önceki zamanlarda büyük farklılıklara yol açacağı, bilinen bir gerçek haline geldi.

Günümüzde bilim adamları, havanın, Lorenz' in hava akımlarına ilişkin basit bilgisayar modeli gibi kaotik bir sistem olduğuna inanmaktalar. Yani belli bir doğrulukta uzun vadeli bir hava tahmini yapabilmek için sonsuz sayıda ölçüm yapılması gereklidir. Dünyanın tüm atmosferini kocaman bir ölçüm araçları –bu durumda termometreler, rüzgâr-ölçerler ve basınç-ölçerler- ağı ile doldurmak mümkün olsaydı bile, başlangıç koşullarındaki belirsizlikler bu kez de ağdaki her bir aracın yapacağı ölçüm değerleri arasındaki minik farklılıklardan meydana çıkacaktı. Atmosfer kaotik bir yapıda olduğundan ne kadar küçük olursa olsun bu belirsizlikler gittikçe hesapları geçersizleştirecek ve hava tahminin doğruluğunu ortadan kaldıracaktır. Bu ilke "Kelebek Etkisi" olarak adlandırılır[43].

E.Lorenz "Kelebek Etkisi" olarak tanınan görüşü aslında atmosferik konveksiyon olgusuna ilişkin bazı araştırmalar yaparken tanık olmuştur. Bu olgu: Güneş ışınlarının yeryüzünü ısıtması ve ısınsın havaya yansıması ile atmosferin alt katmanlarındaki hava üst katmanlarından daha sıcak ve daha hafif durma gelir. Yoğun olan üst katmandaki hava aşağıya hareket eder. Bu iki yönlü hareketlenemeye atmosferik konveksiyon denir. Hava da su gibi akışkan olduğu için sonsuz sayıda boyutları bulunan uzayda bir nokta ile tanımlanması gerekir. E.Lorenz yaklaşık bir biçimde sonsuz boyutlu uzaydaki gerçek zamansal değişimi bilgisayarda

inceleyebileceği üç boyutlu bir değişim kullanarak bugünkü Lorenz çekeri olarak bilinen bir nesne ortaya çıkmıştır.

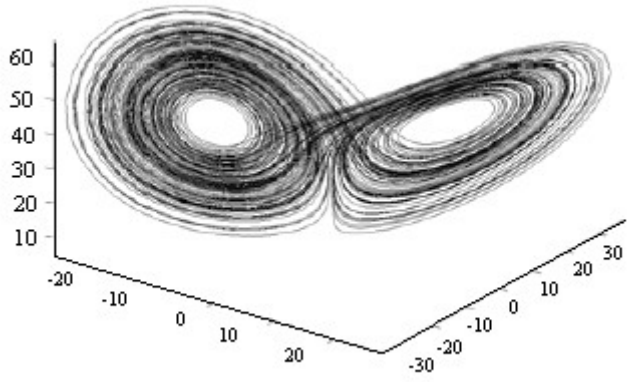
Lorenz' in hava tahminlerini hesaplanmasında kullandığı iki boyutlu akışkan konveksiyonu için bir model olarak tanımladığı ve σ, r ve b sistem parametreleri olmak üzere;

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \sigma(x_2 - x_1) \\ rx_1 - x_2 - x_1x_3 \\ -bx_3 + x_1x_2 \end{bmatrix}$$

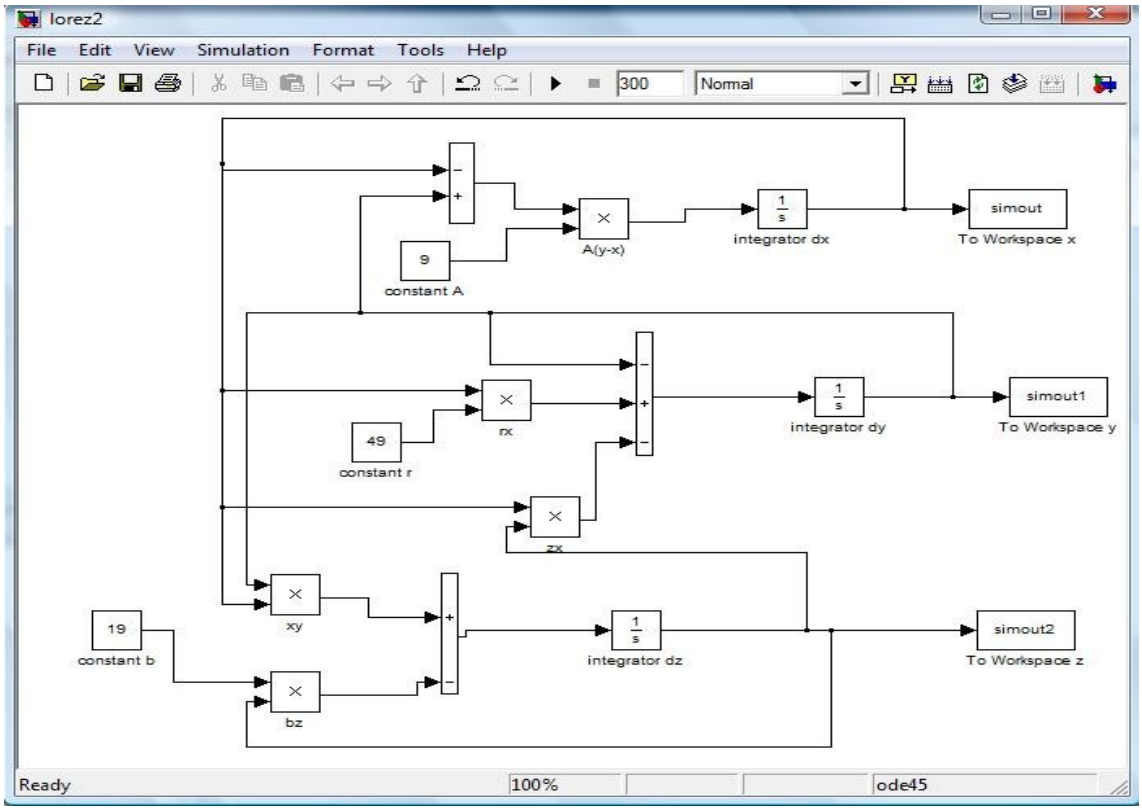
denklemi ile verilir. Burada *x 1* , *x 2* ve *x 3* durum değişkenleridir [44].

Denklemdeki sistem parametrelerinin değişmesi ile sistemin cevabı da değişecektir. Nitekim E.Lorenz de yapmış olduğu ölçümlerde çok küçük değişikliklerde sistemin cevabının ne kadar farklı bir şekil aldığını görmüştür.

Sistem parametrelerin değiştiği zaman değişik cevaplar vermektedir. Bu değişikliği MATLAB ortamında yapılan benzetimlerle görmek mümkündür. x_1 , x_2 ve x_3 durum değişkenlerini σ, r ve b sistem parametrelerini değiştirildiği zaman farklı davranışlar gösterdiğini sistemin o anki cevabı olduğu görülmektedir.Sistem parametrelerin değişik değer için simülasyon sonuçları aşağıda verilmiştir.

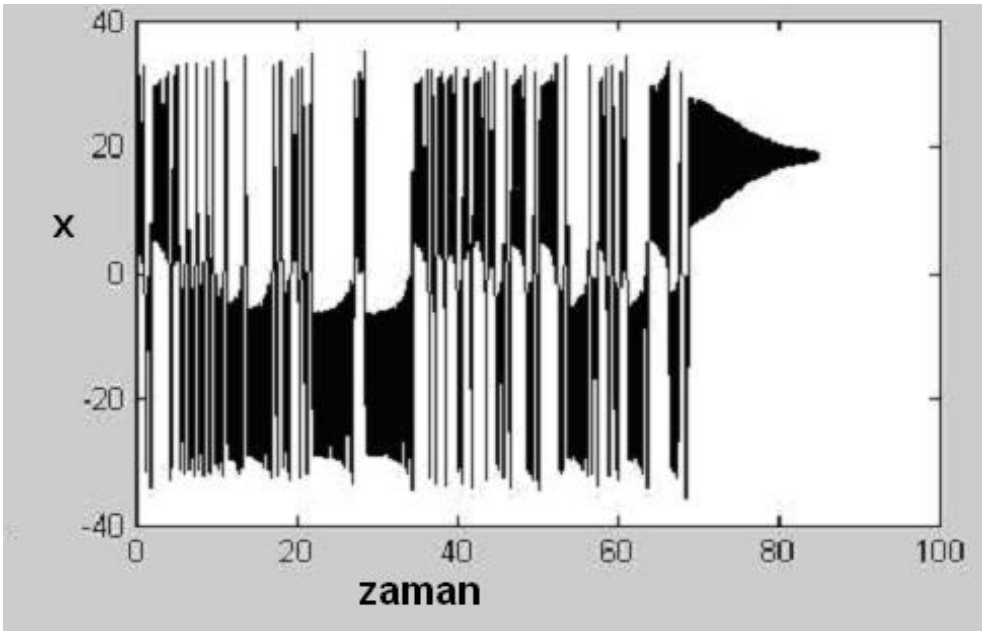


Şekil 3.40. Lorenz Çekici

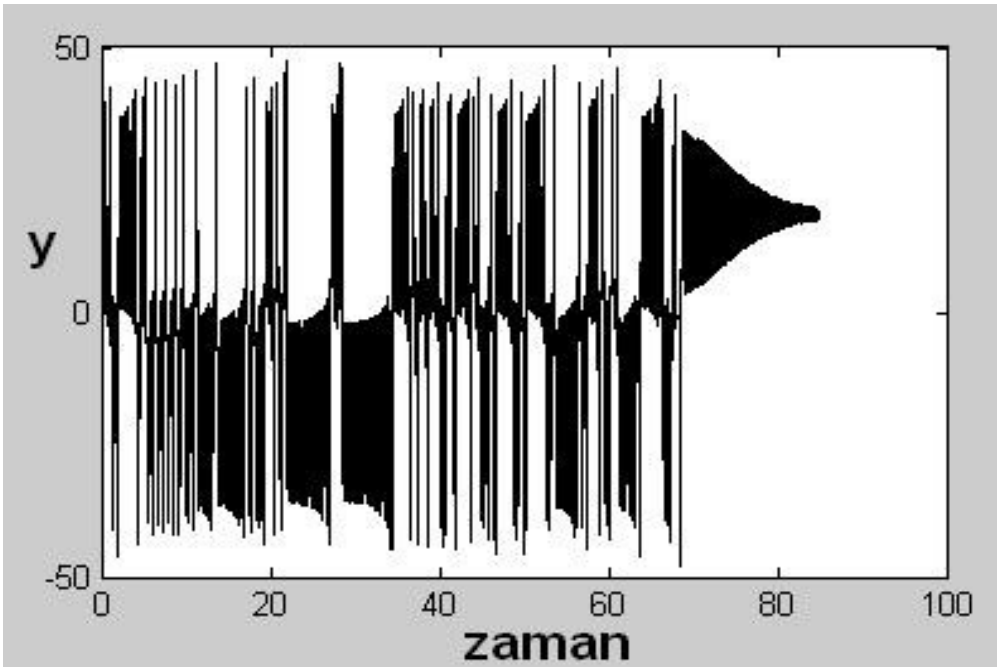


Şekil 3.41. Matlab Simulink Lorenz Kaotik Osilatörü

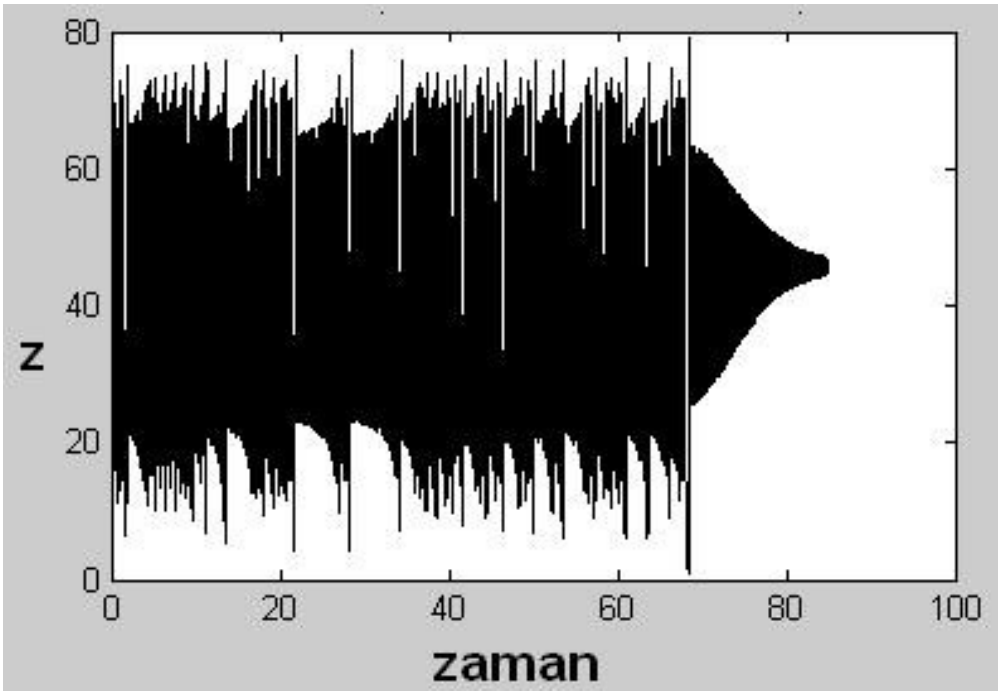
3.3.1. Lorenz dinamik denklemlerinin simülasyon çıktıları



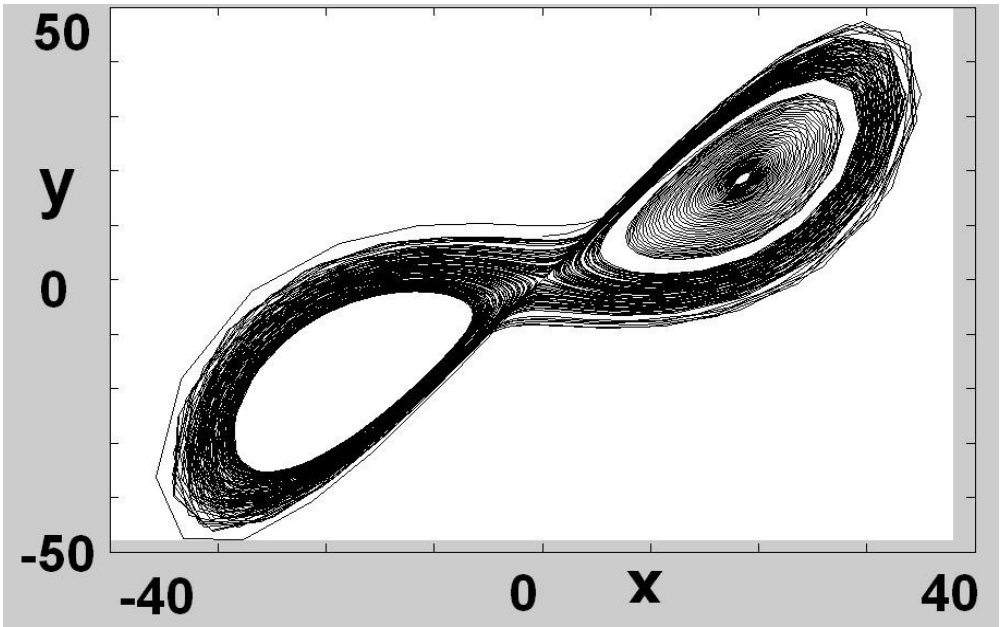
Şekil 3.42. Lorenz Sisteminin X Durum Değişkeninin Kaotik Değişimi



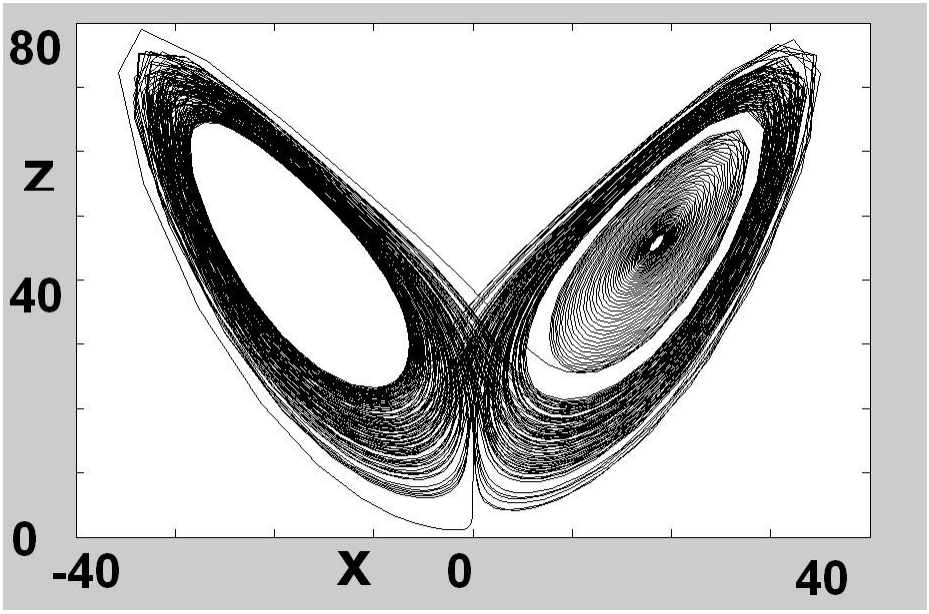
Şekil 3.43. Lorenz Sisteminin Y Durum Değişkeninin Kaotik Değişimi



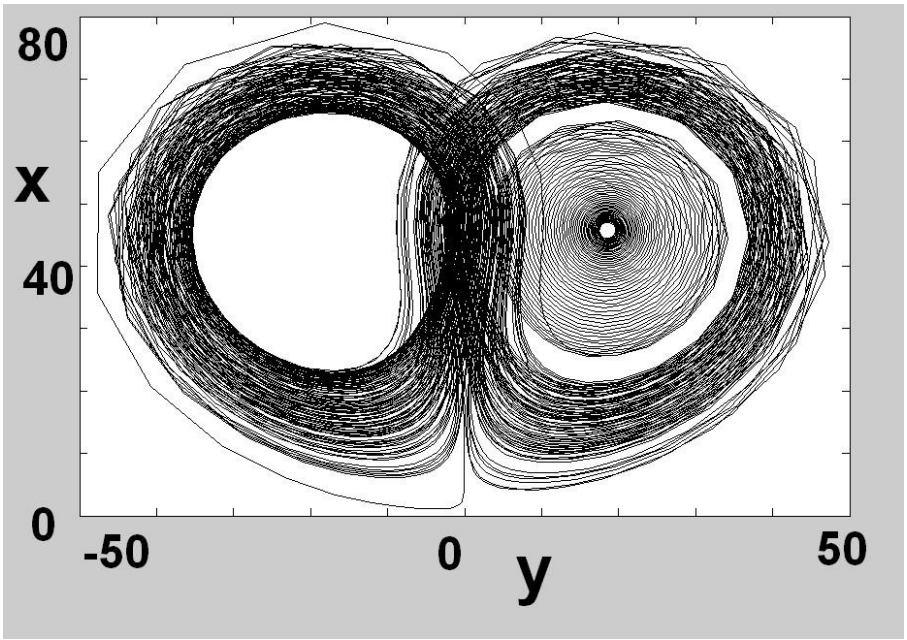
Şekil 3.44. Lorenz Sisteminin Z Durum Değişkeninin Kaotik Değişimi



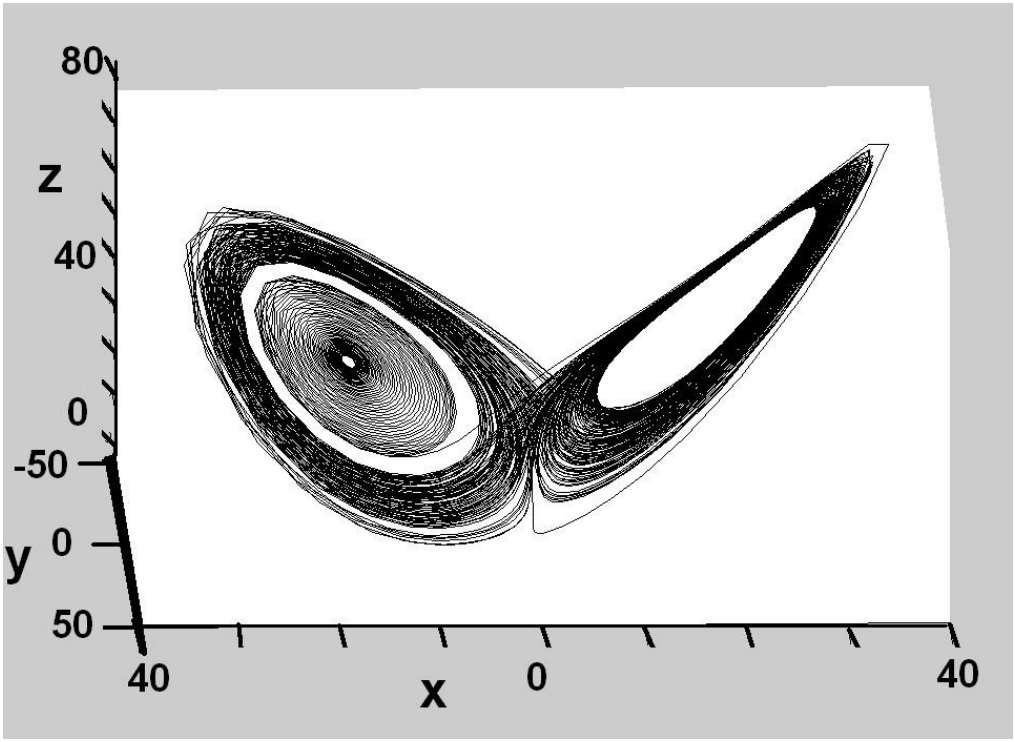
Şekil 3.45. (X-Y) Durum Değişkenleri Arasındaki Kaos Çekicisi



Şekil 3.46. (X-Z) Durum Değişkenleri Arasındaki Kaos Çekicisi



Şekil 3.47. (Y-Z) Durum Değişkenleri Arasındaki Kaos Çekicisi



Şekil 3.48. (X-Y-Z) Değişkenleri Arasındaki 3 Boyutlu Kaos

Yukarıdaki (Şekil 3.1.1'de) Lorenz durum denklemlerinin değişkenlerinin çizdirilmesi için kullanılan sistem parametreleri $\sigma = 7.5$, $r = 46.8$, $b = 18.5$ ' dir.

a)'da Grafikte **x,** değişkenini girilen değişken parametrelerine gösterdiği tepki, geçici durumundan sonra 10. sn de sürekli hal davranışı gözlenmektedir.

b)'de şekilde **y** durum denkleminin değişken parametrelere verdiği tepki

c)'de Benzer şekilde **z** durum denkleminin değişken parametrelere verdiği tepki.

d)'de **x** ve **y** durum denklemlerinin birbirlerinin fazlarına verdikleri tepki, görüldüğü gibi çift sarmalın

e)'de **x** ve **z** durum denklemelerinin karşılaştırılması

f)'de **y** ve **z** durum denklemelerinin karşılaştırılması.

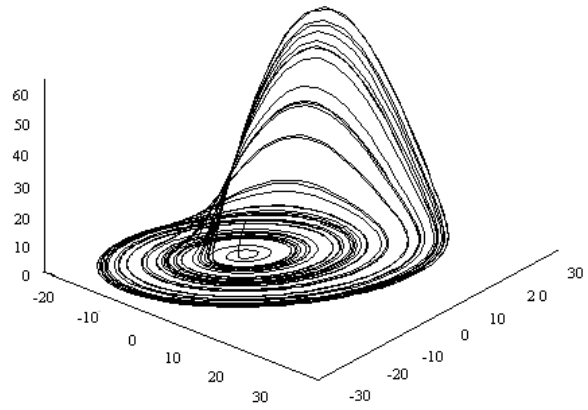
g)'de Bu grafikte **x, y** ve z durum denklemlerinin zamanla ilişkisini

3.4. Rossler Kaotik Osilatörü

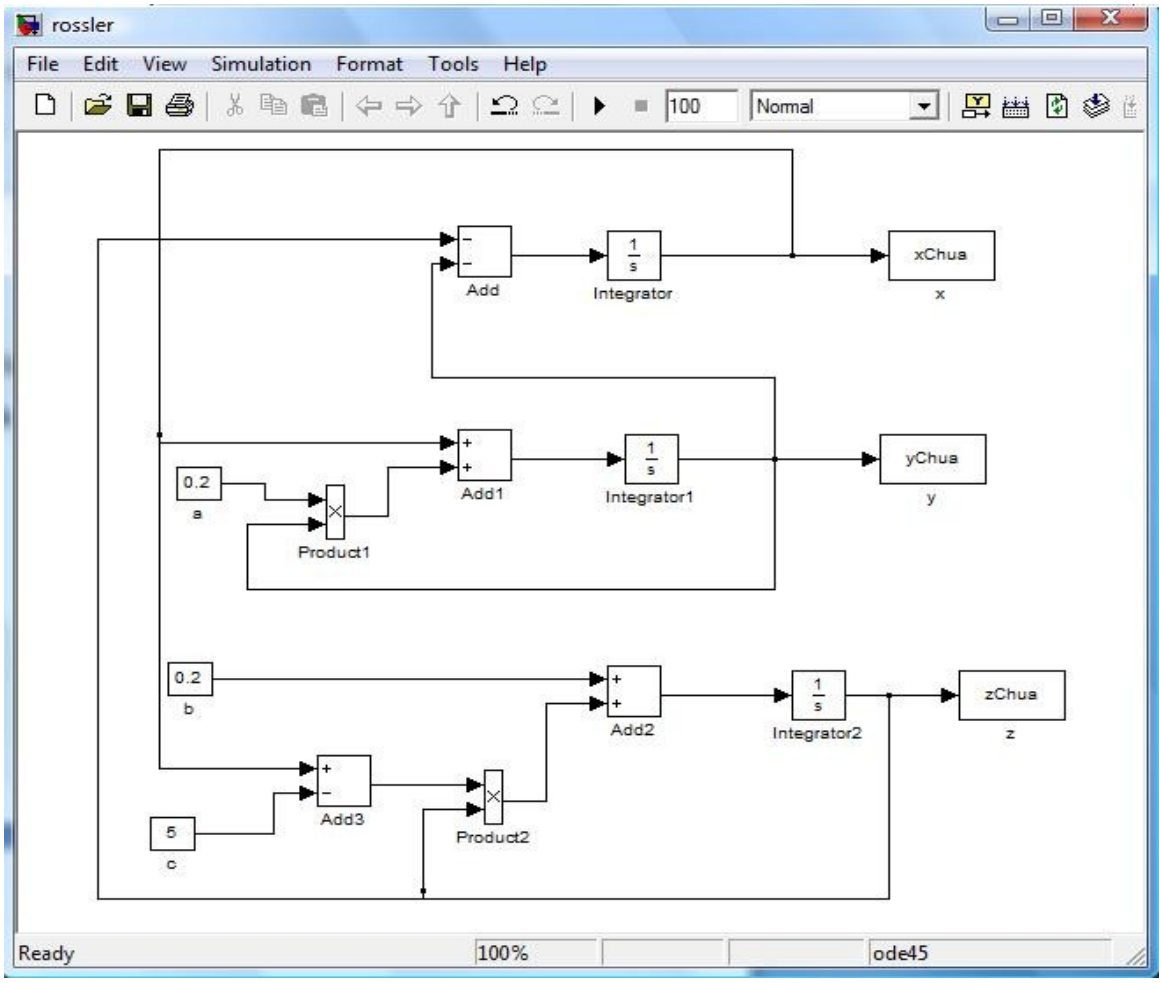
Kaotik işaretler üreten bir başka dinamik denklemde Rossler dinamik denklemleridir. Rossler dinamik denklemleri a, b ve c sistem parametreleri olmak üzere x_1, x_2 ve x_3 durum değişkenleridir.

$$\frac{dx}{dt} = -(y - z)$$
$$\frac{dy}{dt} = x(1 + a)$$
$$\frac{dz}{dt} = b + z(x - c)$$

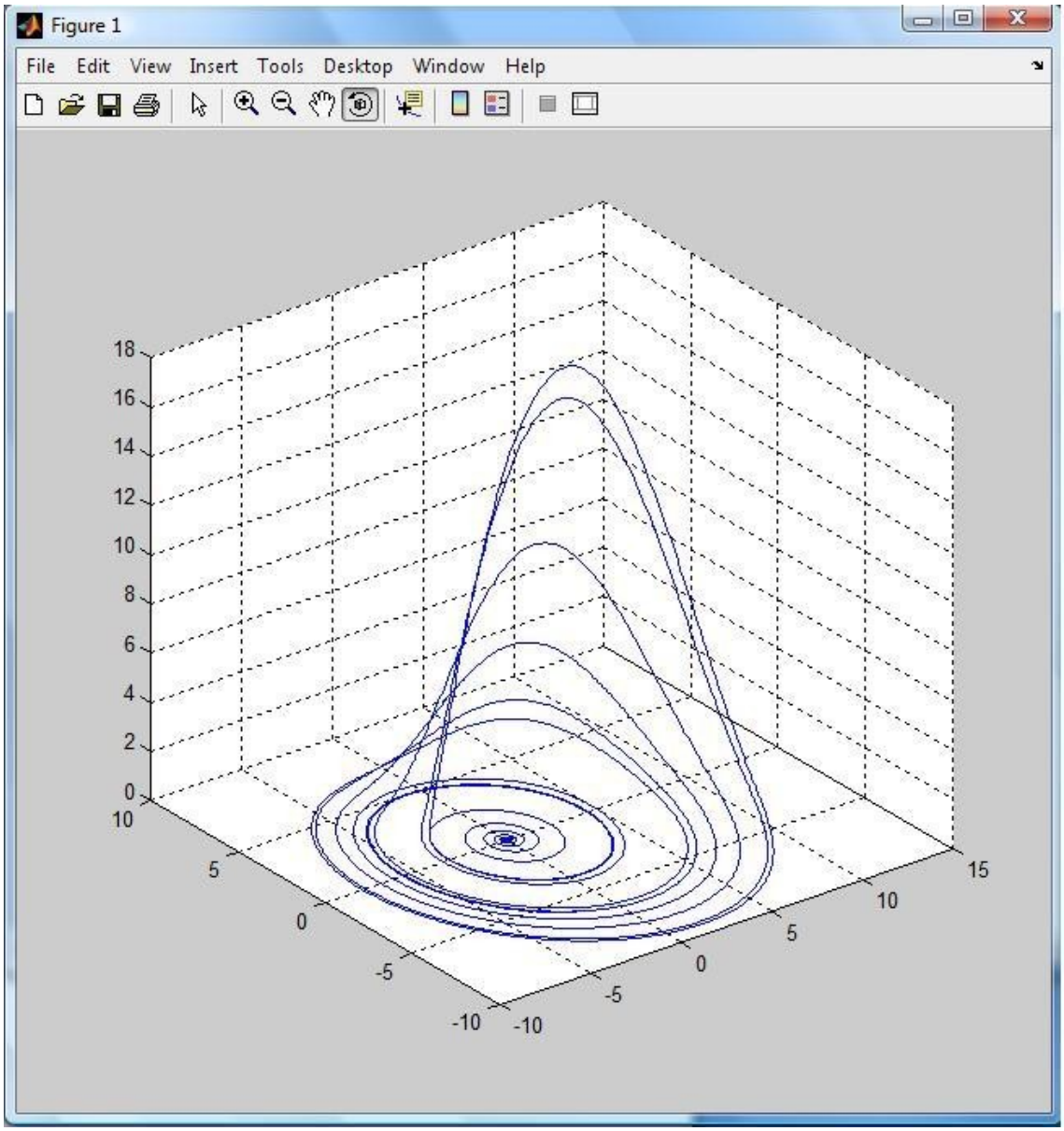
Rossler dinamik denklerini kullanarak kaotik işaretler üretilmektedir.[45] Sistem parametrelerinin değiştirmesi ile MATLAB ortamında yapılan benzetim sonuçları dedir.



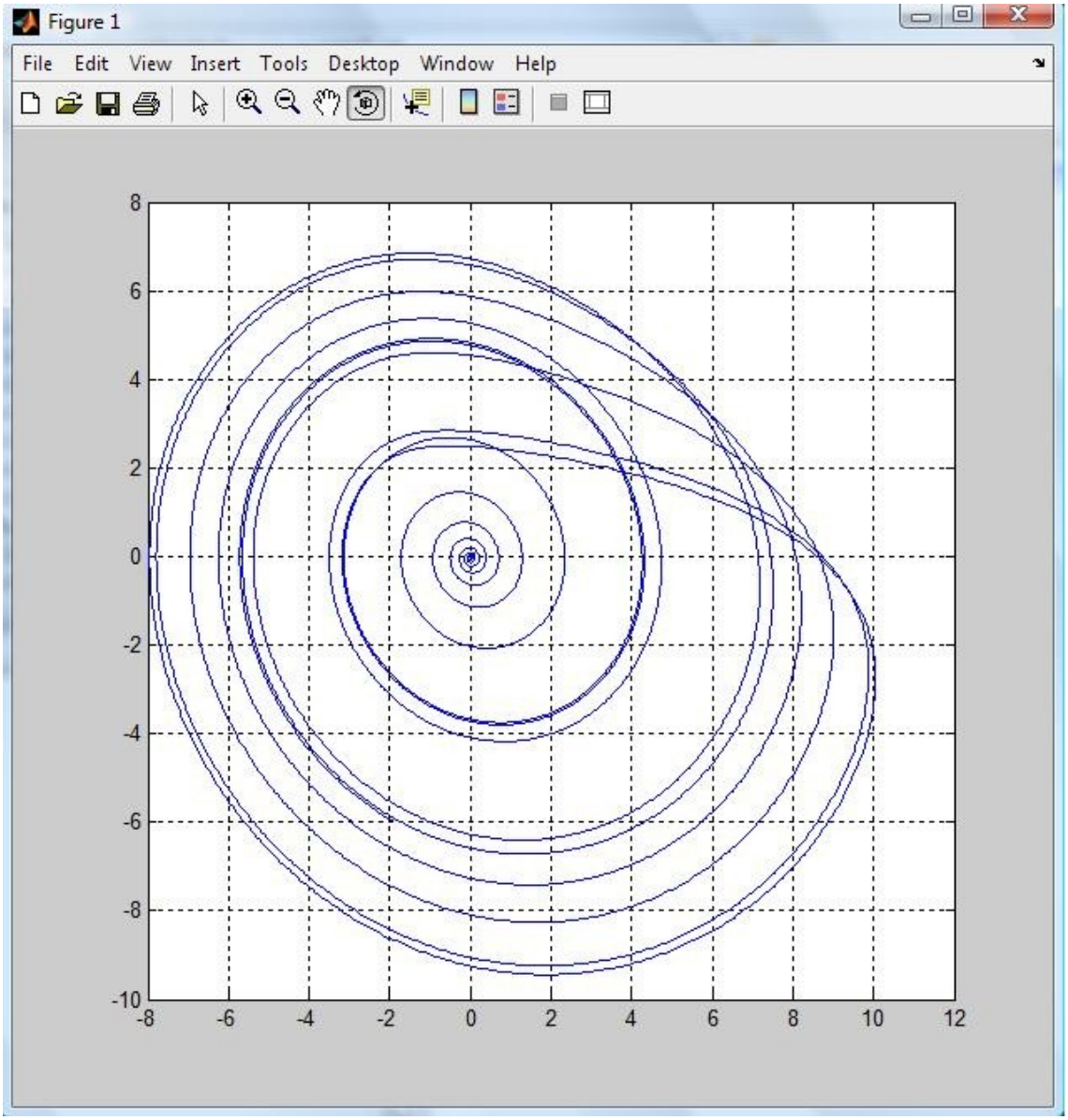
Şekil 3.49. Rossler Dinamiği



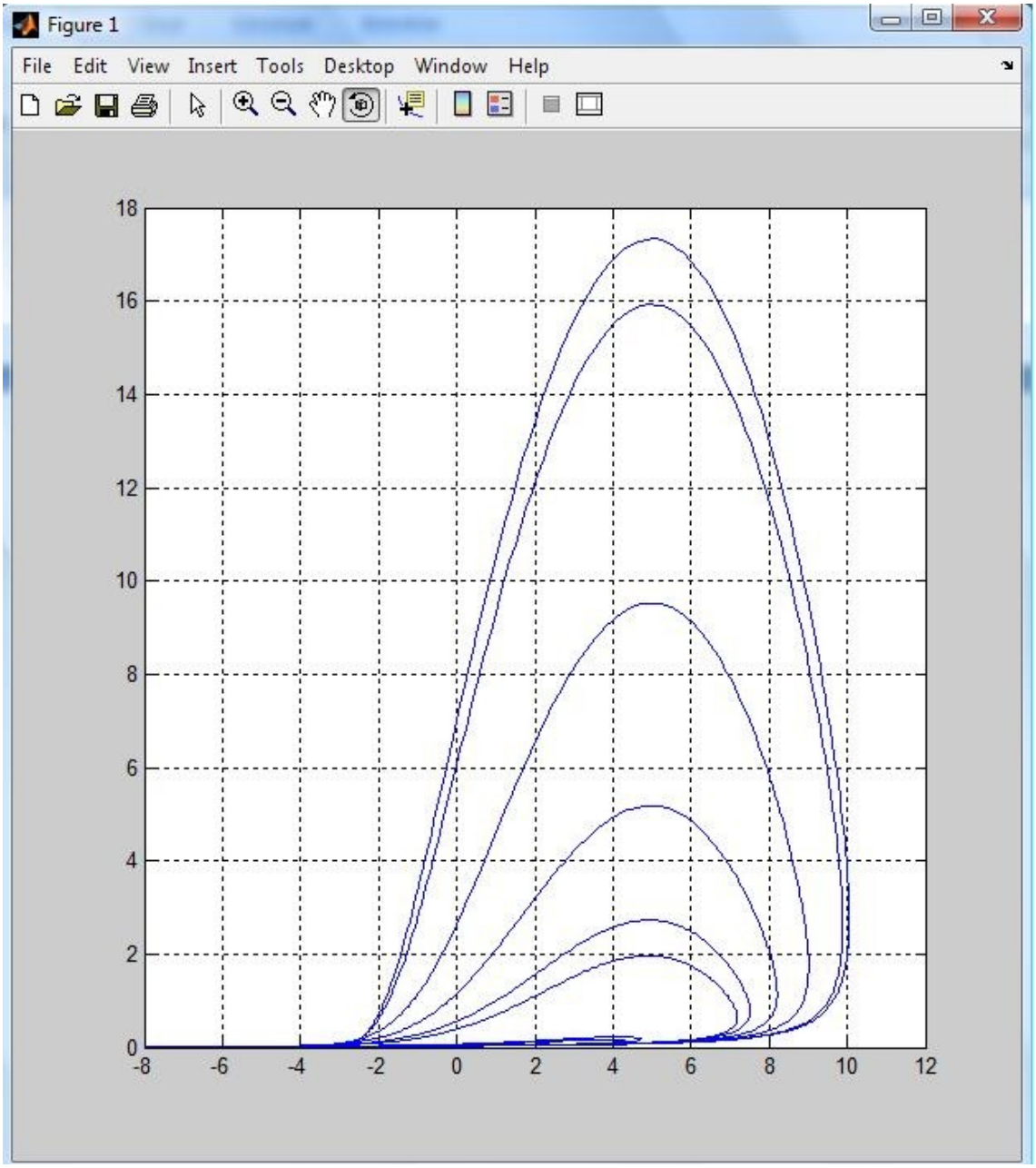
Şekil 3.50. MATLAB Simulink Rossler Kaotik Osilatörü



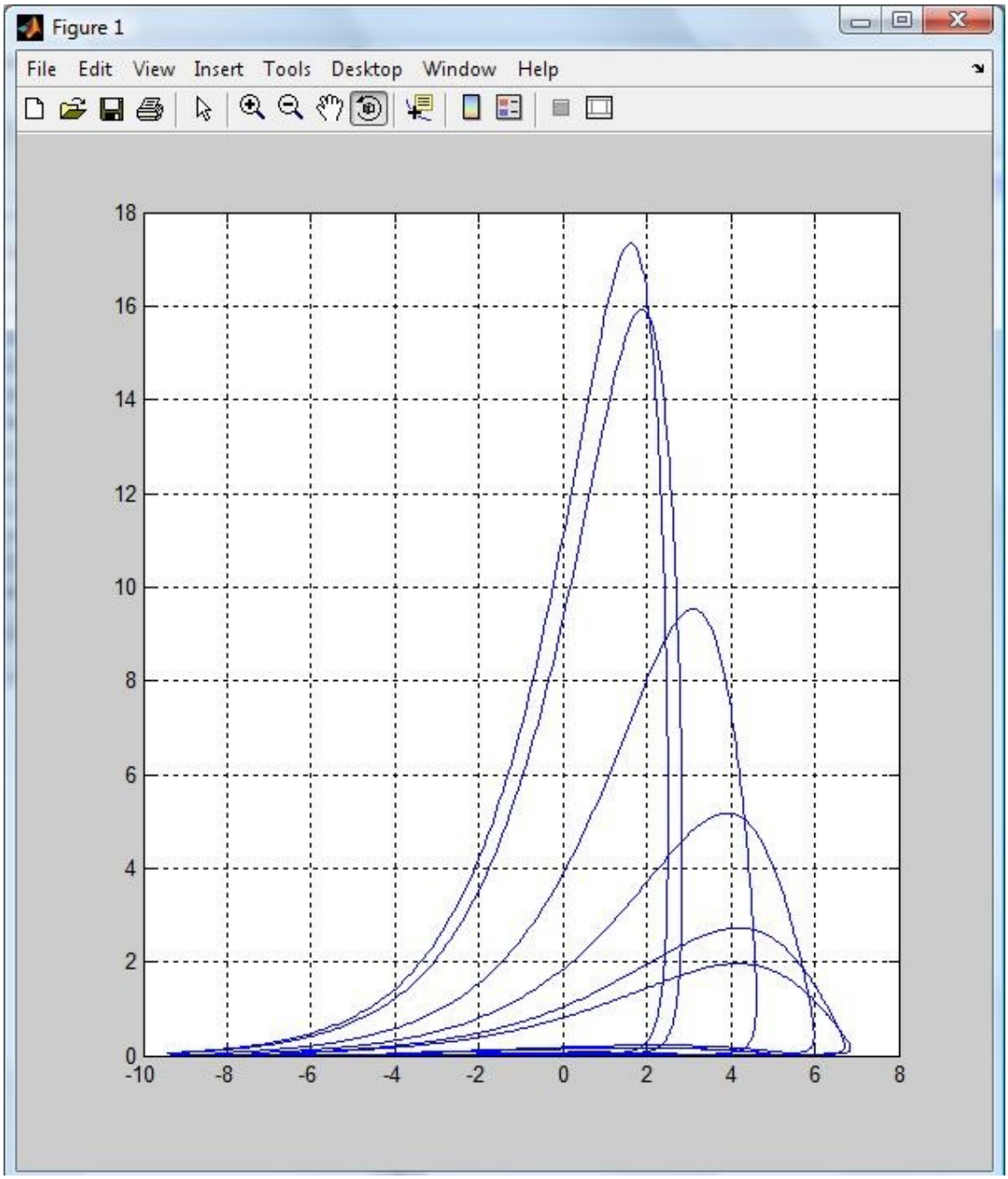
Şekil 3.51. (X, Y, Z) Grafiği, plot3(xChua,yChua,zChua)



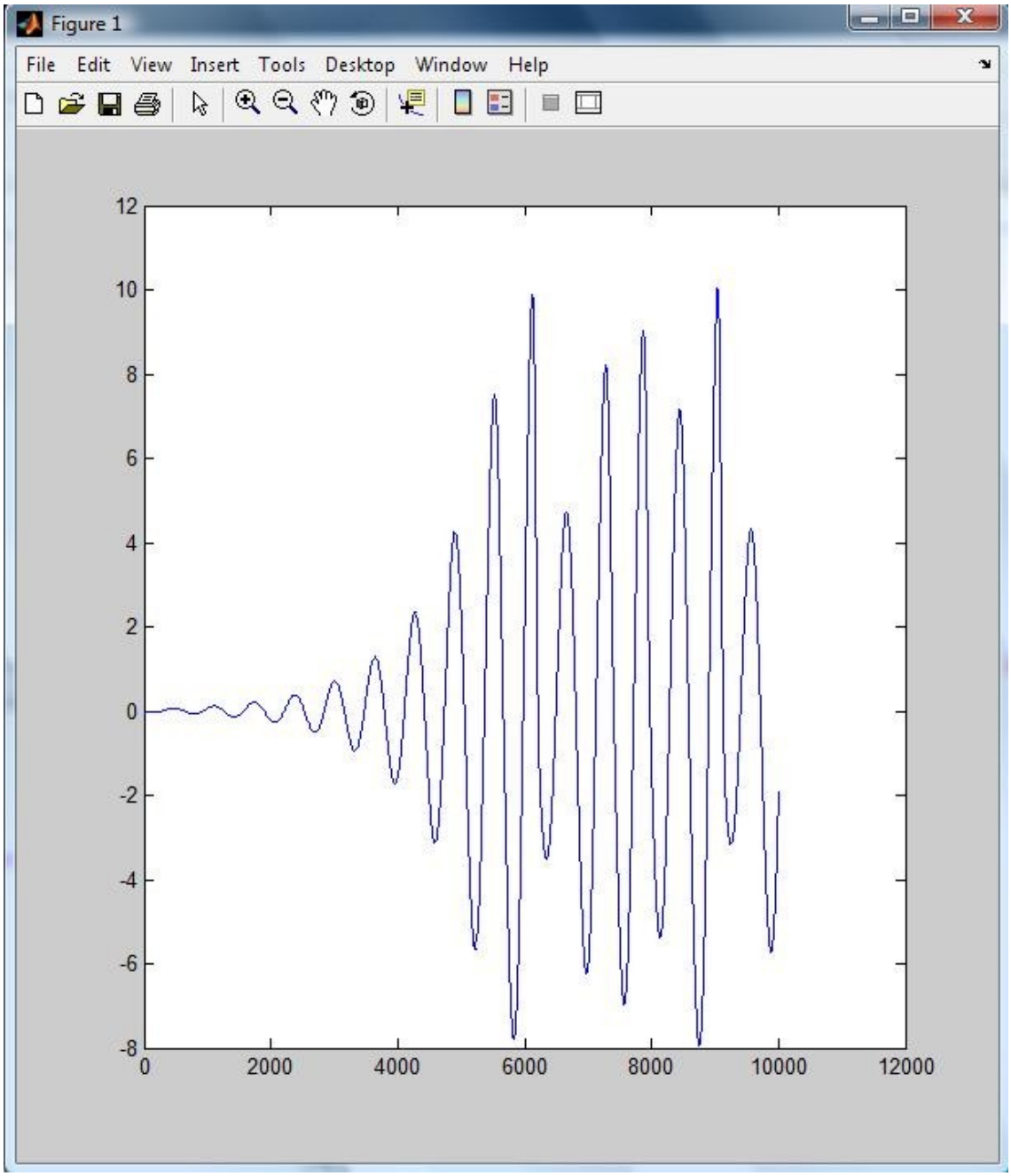
Şekil 3.52. (X, Y) Grafiği, plot(xChua,yChua)



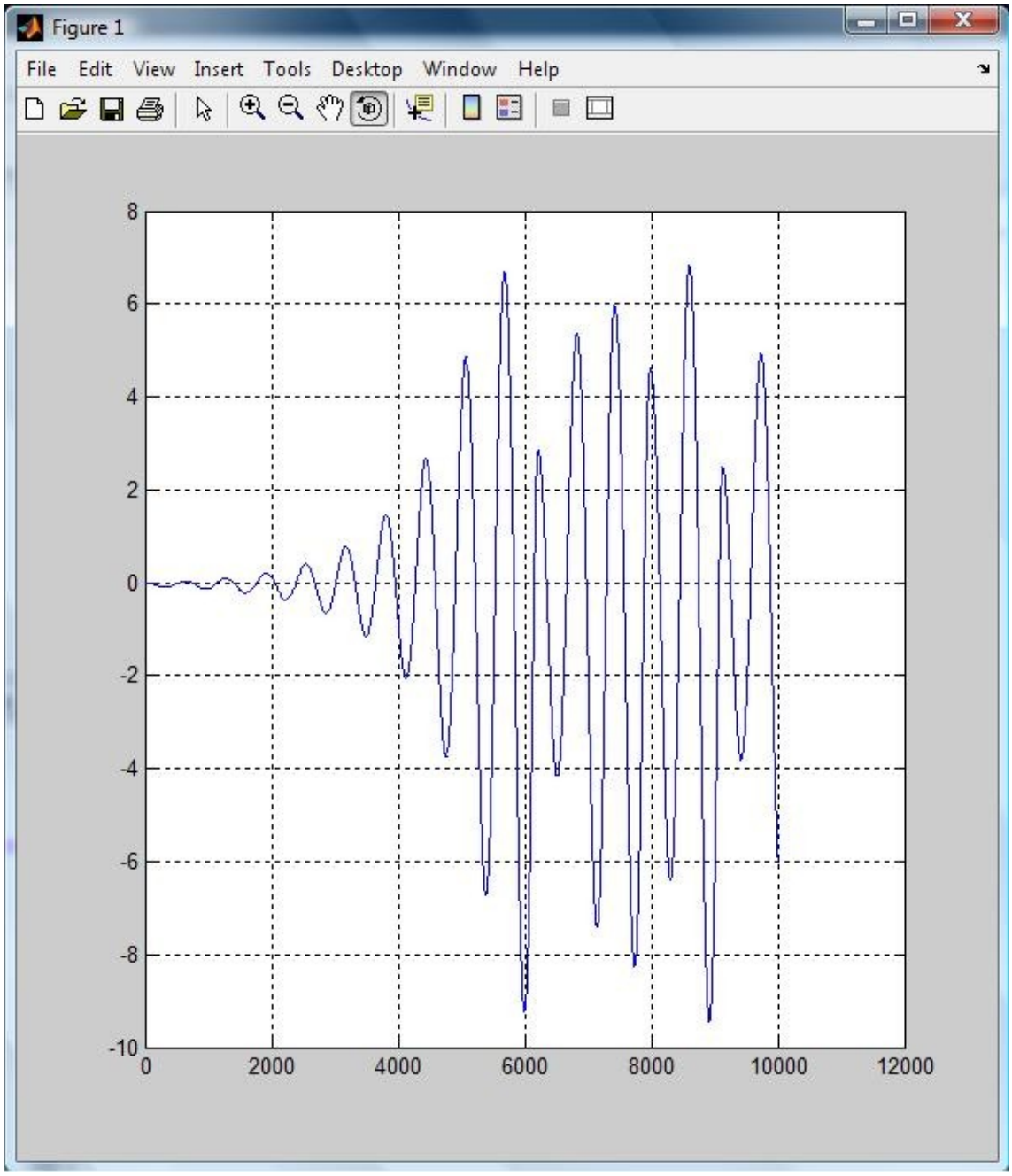
Şekil 3.53. (X, Z) Grafiği, plot(xChua,zChua)



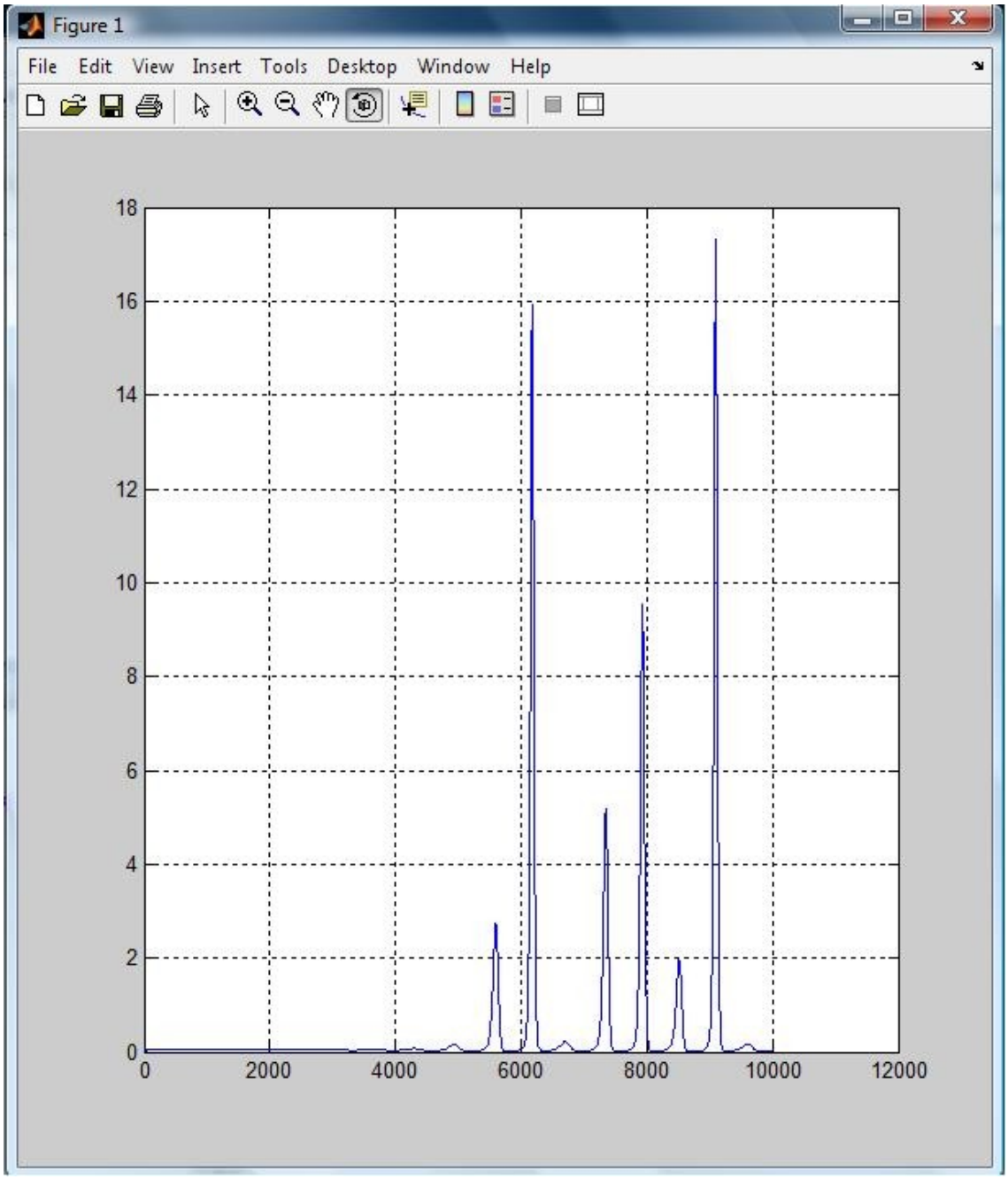
Şekil 3.54. (Y, Z) Grafiği, plot(yChua,zChua)



Şekil 3.55. (X, T) Grafiği, plot(t,xChua)



Şekil 3.56. (Y, T) Grafiği, plot(t,yChua)



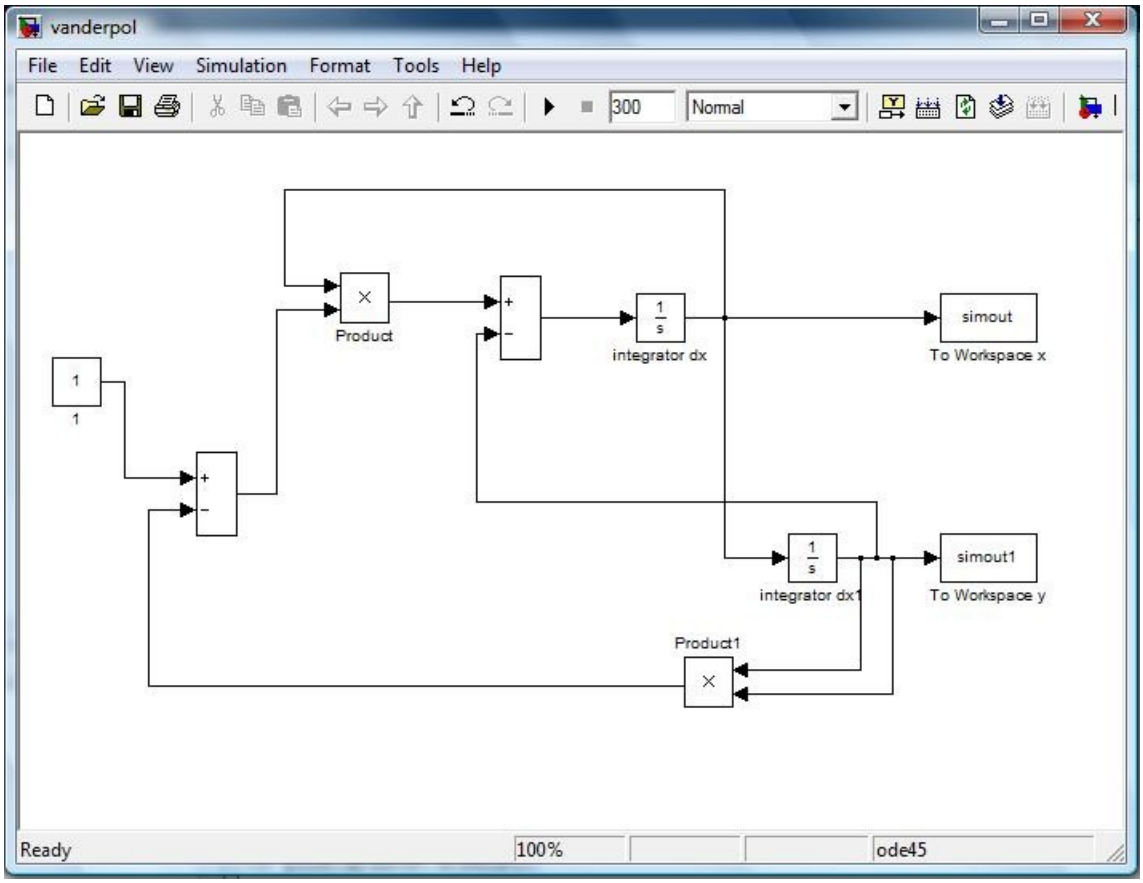
Şekil 3.57. (Z, T) Grafiği, plot(t,zChua)

3.5. Vanderpol Kaotik Osilatörü

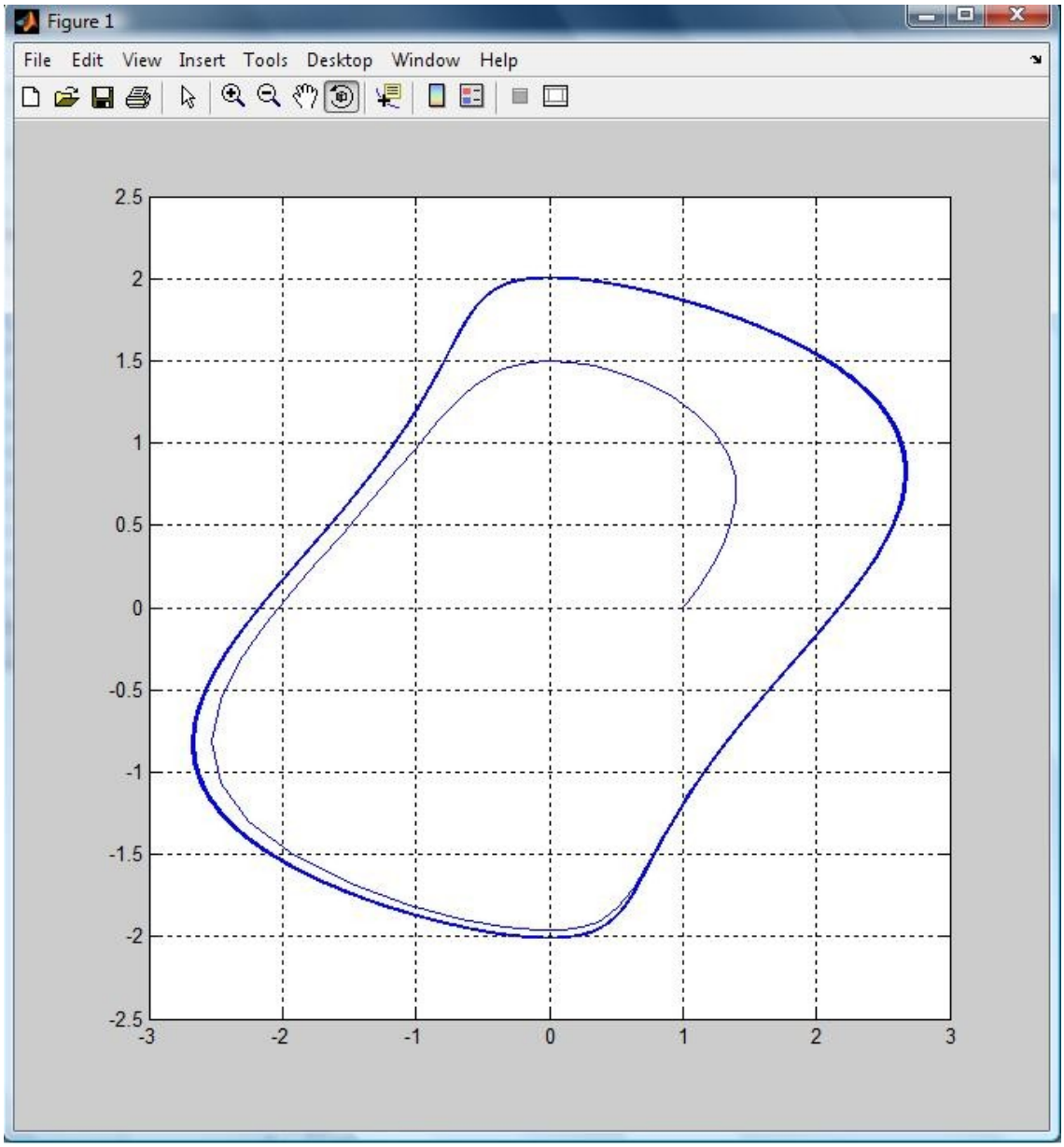
Van der pol dinamik denklemlerinin x, y *ve* z durum değişkenleridir.

$$\frac{dx}{dt} = x(1 - y^2) - y$$

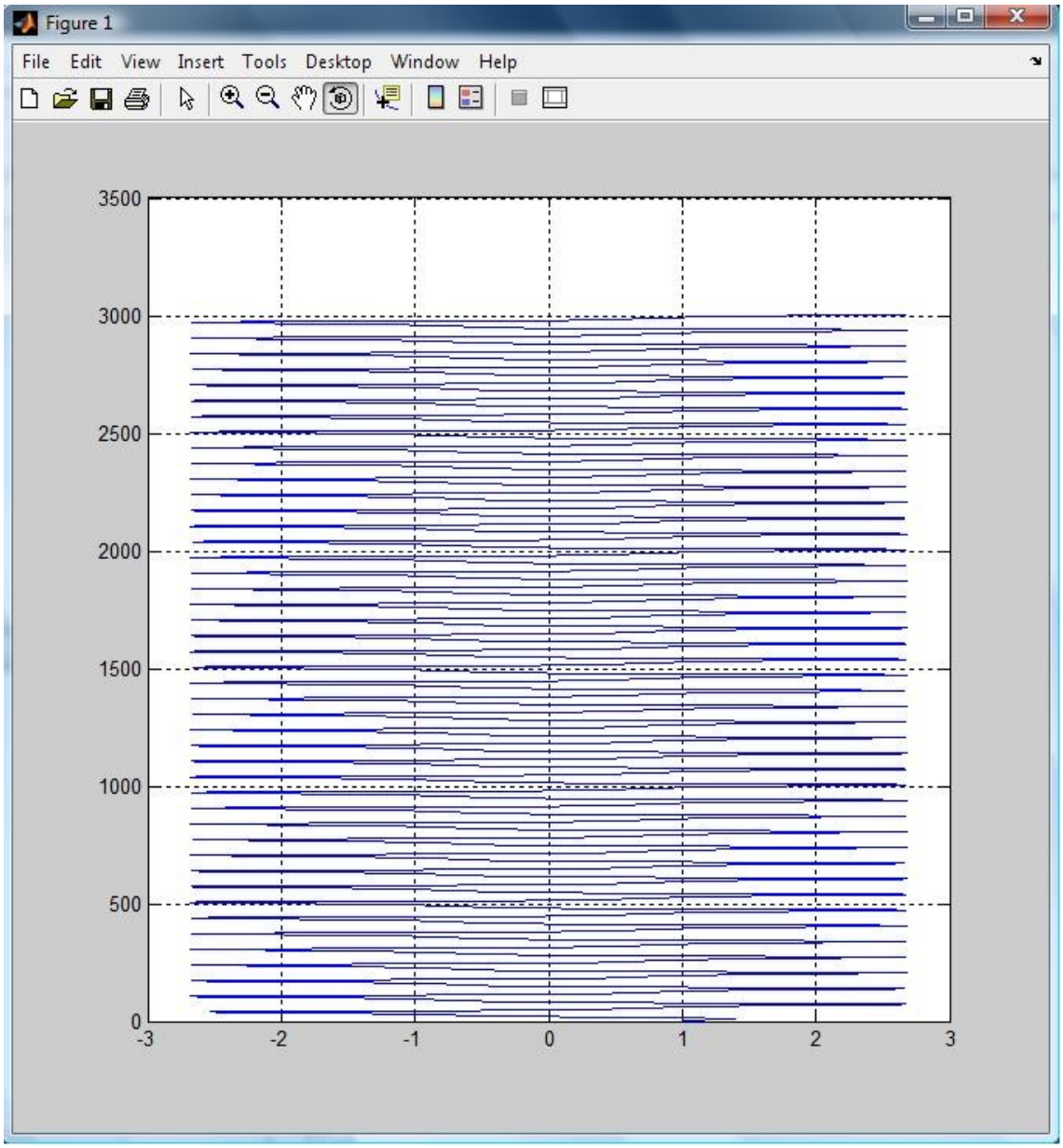
$$\frac{dy}{dt} = x$$



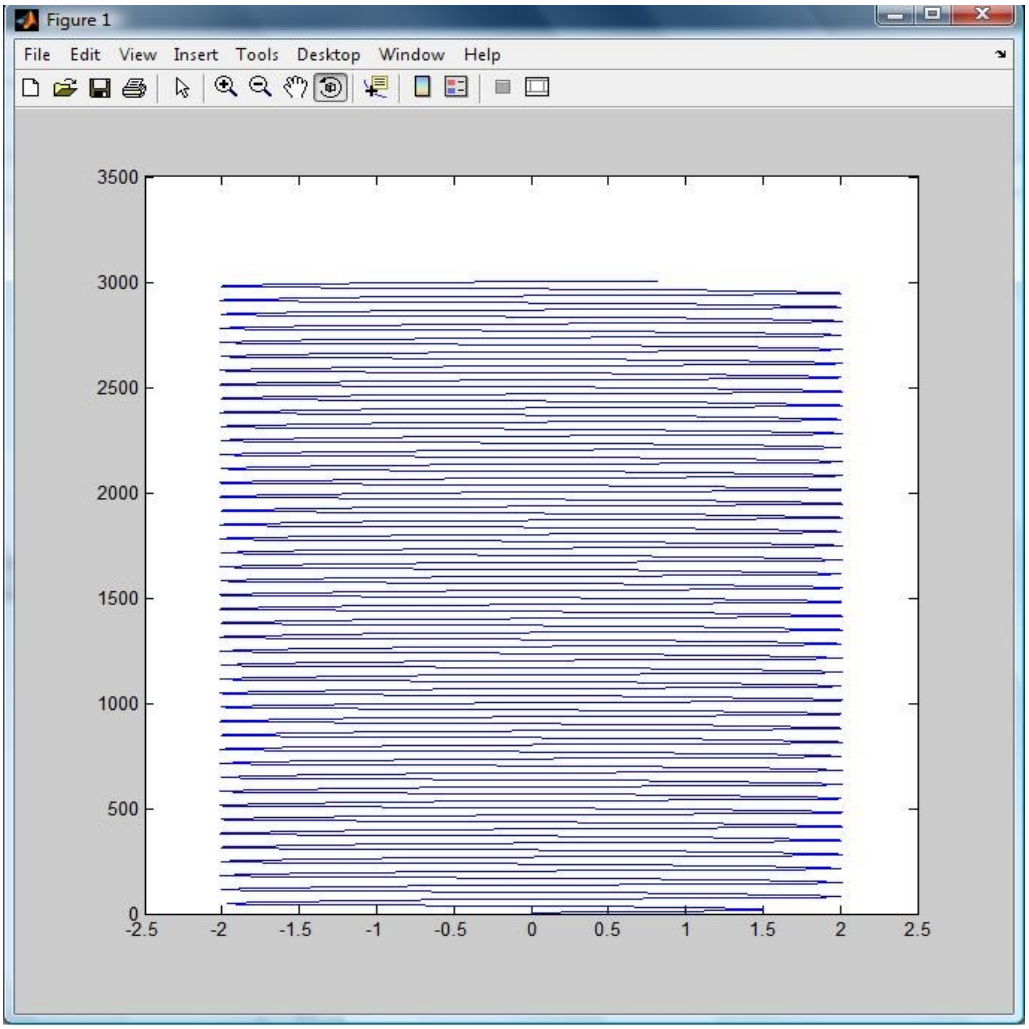
Şekil 3.58. Matlab Simulink Vanderpol Kaotik Osilatörü



Şekil 3.59. (X, Y) Grafiği, plot(Simout,Simout1)



Şekil 3.60. (X, T) Grafiği, plot(simout,t)



Şekil 3.61. (Y, T) Grafiği, plot(simout1,t)

3.6 Duffing Kaotik Osilatörü

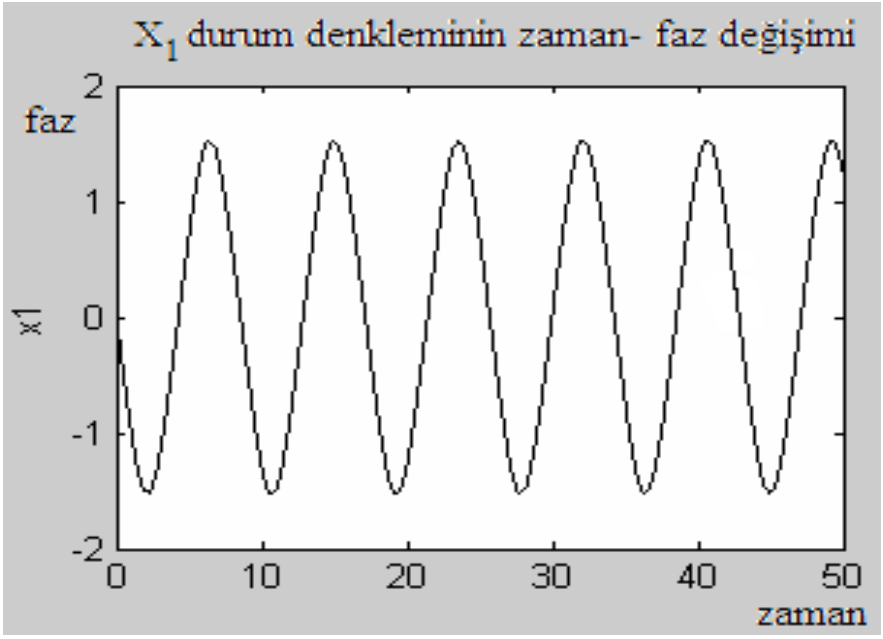
Duffing dinamik denklemlerinin x, y *ve* z durum değişkenleridir

$$\frac{dx}{dt} = y$$

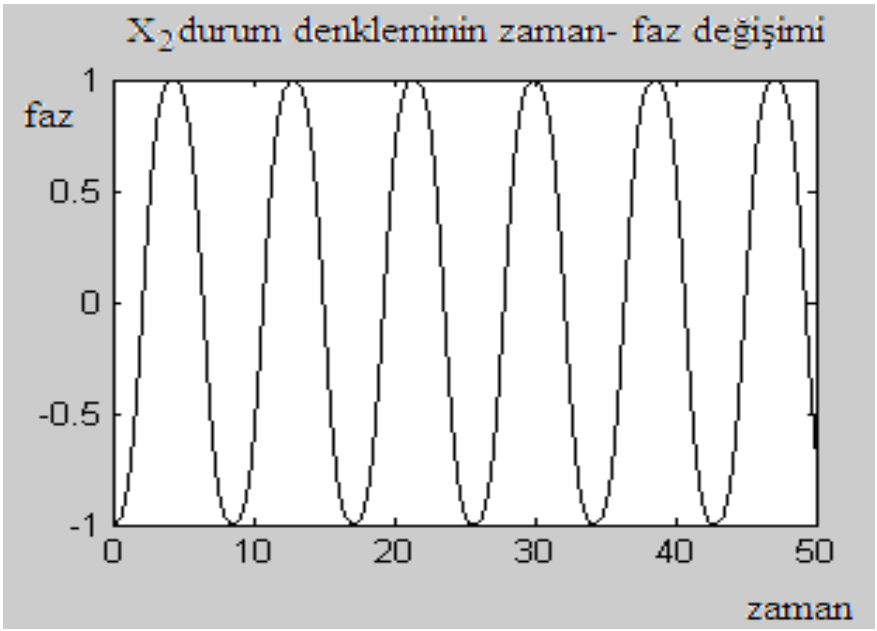
$$\frac{dy}{dt} = -\varepsilon y - ax - bx^3 + B\cos z$$

$$\frac{dz}{dt} = \omega$$

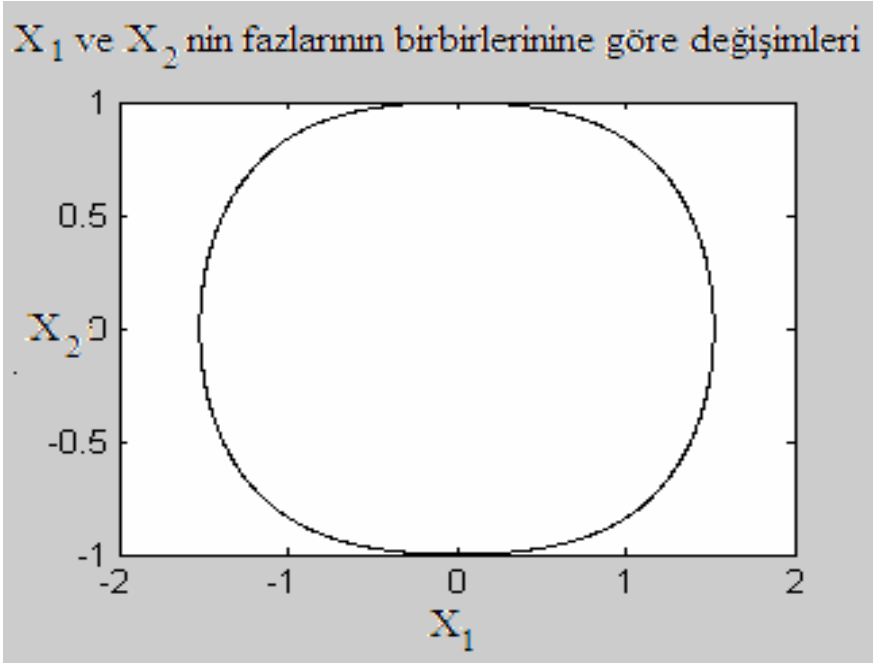
3.6.1. Duffing dinamik denklemlerinin simülasyon çıktıları



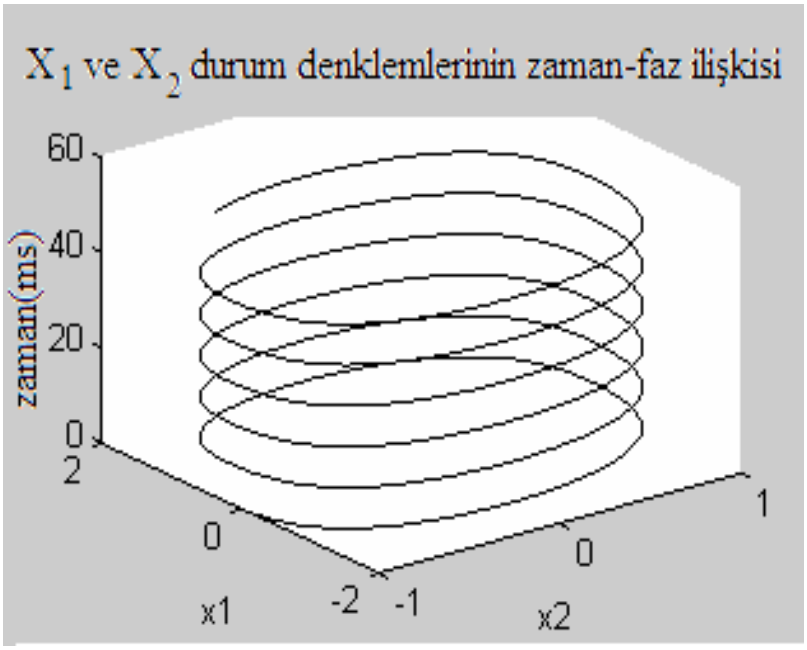
Şekil 3.62. Duffing Sistemin X_1 Değişkeninin Zamanla Kaotik Değişimi



Şekil 3.63. Duffing Sistemin X_2 Değişkeninin Zamanla Kaotik Değişimi



Şekil 3.64. Duffing Sistemin X_1 Ve X_2 Durum Denklemlerinin Değişimi



Şekil 3.65. Duffing Sistemin X_1 Ve X_2 Durum Denklemlerinin Zamanla Değişimi

Yukarıdaki Duffing durum değişkenlerindeki benzetim çalışması için kullanılan sistem parametreleri a = 0,2 ve b = 0,2 değerleridir.

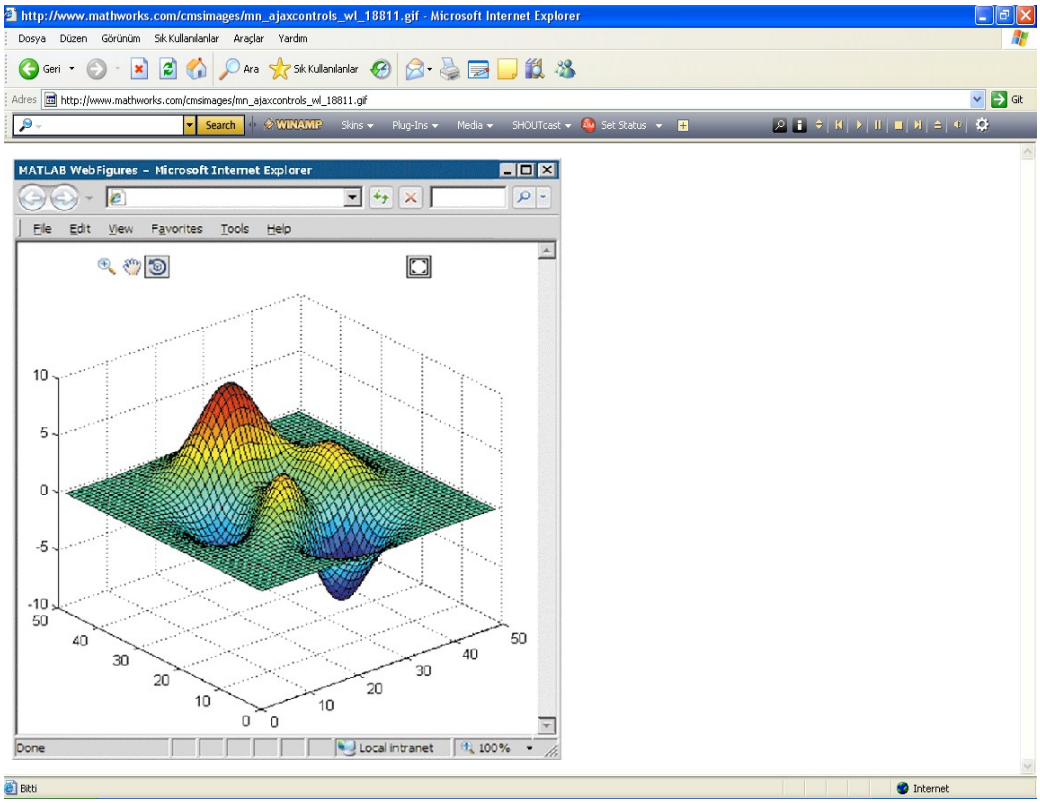
Duffing dinamik denklemlerini kullanarak yapılan benzetim çalışması için durum denklemleri elde edilip mevcut parametrelerin oranı, çarpımı ile değişken parametre sayısı ikiye düşürülmüş ve bunları Matlab programında uygulayarak benzetim çalışmaları yapılmıştır.

Bu benzetimde daha önceki benzetim çalışmalarında olduğu gibi zaman- faz ilişkileri incelenmiş ve birbirlerine göre durumları gösterilmiştir. a) ve b) durumlar bu yönde yapılan çalışmalardır. Diğer çalışmalar benzer şekilde durum denklemlerin birbirleri ile olan ilişkilerini gösterir.

BÖLÜM 4. MATLAB .NET BUILDER ARAÇKUTUSU

MATLAB .NET BUILDER, MATLAB tabanlı .NET ve COM bileşenleri oluşturarak , bu bileşenlerin doğrudan Masaüstü ve Web Ortamlarında servis edilebilmesini sağlamak için geliştirilen bir araçtır.

MATLAB .NET BUILDER tarafından .NET ve COM nesnesi haline getirilen bileşenleri devam eden yazılım projelerine referans göstererek uygulamalarınızı çalışma zamanında derleyebilirsiniz.Böylelikle hem .NET ortamının geniş kontrol esnekliği , hem MATLAB in güçlü hesaplama ve analiz yetenekleri tek çatı altına toplanmıştır.



Şekil 4.1. Matlab .NET BUILDER Araç Kutusu Kullanılarak Yapılan Bir Simulasyon Çıktısı

MATLAB .NET BUILDER ile oluşturulan bu bileşenler arkaplanda MATLAB Çalışma Zamanı Derleyicisi tarafından bütün MATLAB kütüphaneleri tarafından desteklenen bir yorumlamadan geçerek kullanıcıya MATLAB hesaplama esnekliği sağlar. Tüm MATLAB kütüphane desteği sağlamak için Uygulama yazılımının koşmuş olduğu sunucu üzerine MATLAB Çalışma Zamanı Derleyicisinin(MCR) kurulu olması gerekmektedir.

Web uygulamaları için , MATLAB .NET BUILDER , MATLAB çizimleri için , Ajax- tabanlı büyütme, kesme ve döndürme kontrolleri sağlar. Bundan başka hesaplama sonuçları için elde edilen sonuçları MATLAB veri tipinden , .NET ve COM veri tipine , veri kaybı olmaksızın dönüşüm sağlar.

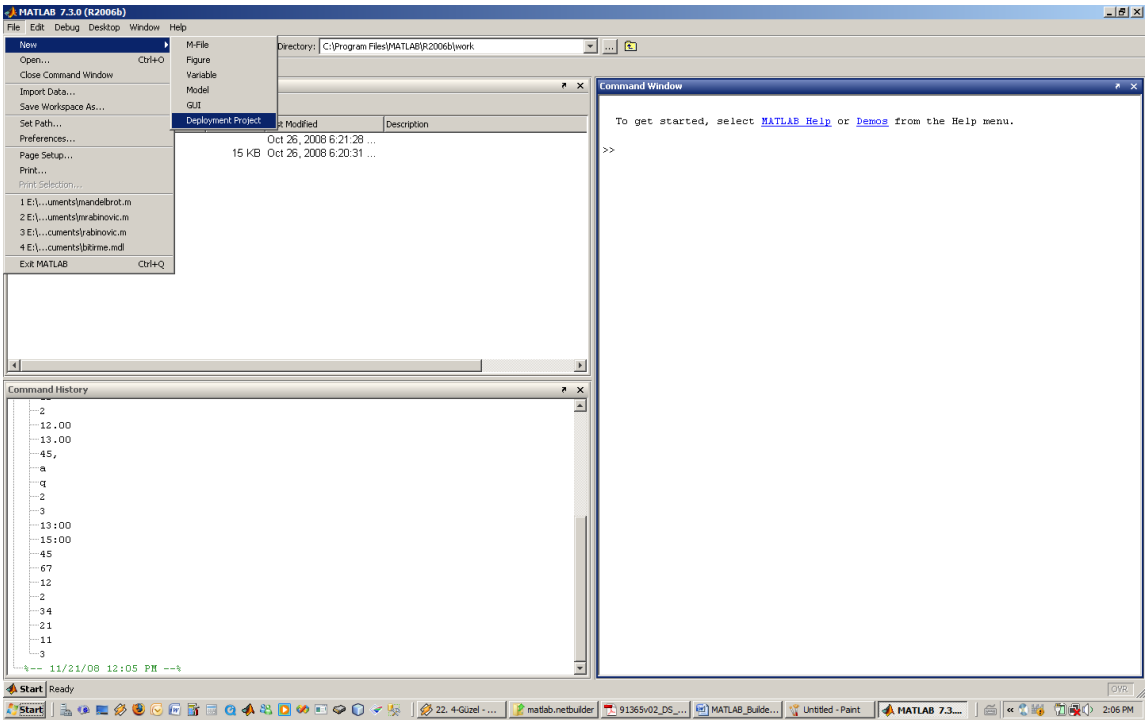
MATLAB ortamında –m file şeklinde yazılan uygulamların .NET ve COM bileşenleri haline getirilmesi , arkaplanda MATLAB Deployment Tool aracılığıyla gerçekleştirilmektedir. Deployment Tool , MATLAB .NET BUILDER ın iki katman arasında dönüşüm yapmak için kullanıcıya sunduğu görsel bir ara katmandır.
Bu ara katman kullanılarak MATLAB ortamında geliştirilen –m file uzantılı çalışmalar, bir Deployment Projesi içine C, C++ ve C# ile geliştrilen projelerde olduğu gibi sınıf ve bileşen ekleme yapılır gibi projeye eklenip COM ve .NET nesnesi haline getirilen bileşenlerin hangi MATLAB fonksiyonunu kullanacağı şeklinde özelleştirilir.

Deployment Tool katmanı bundan başka, MATLAB Derleyici ayarlarının manuel kontrolüne de imkan sağlamaktadır, böylelikle derleme yapılırken .NET ortamında kullanılan Proje spesifikasyonlarına uygun ayarlı , opsiyonel derleme imkanı oluşturulmuş olur. Böylelikle .NET de, .NET ve COM bileşeni haline getirilmiş MATLAB içerikli fonksiyonların çağrılması esnasında , .NET Platformunun Ortak Çalışma Zamanında (CLR) gereksiz işlemci yüklemelerinin önü alınmış olur.

Sonuç olarak MATLAB .NET BUILDER desteğiyle, MATLAB uygulamalarının getirmiş olduğu geniş bilimsel hesaplama ve analiz spektrumu kişiye veya şirkete özel uygulamalara uygulanabilmesinin yolu açılmıştır.

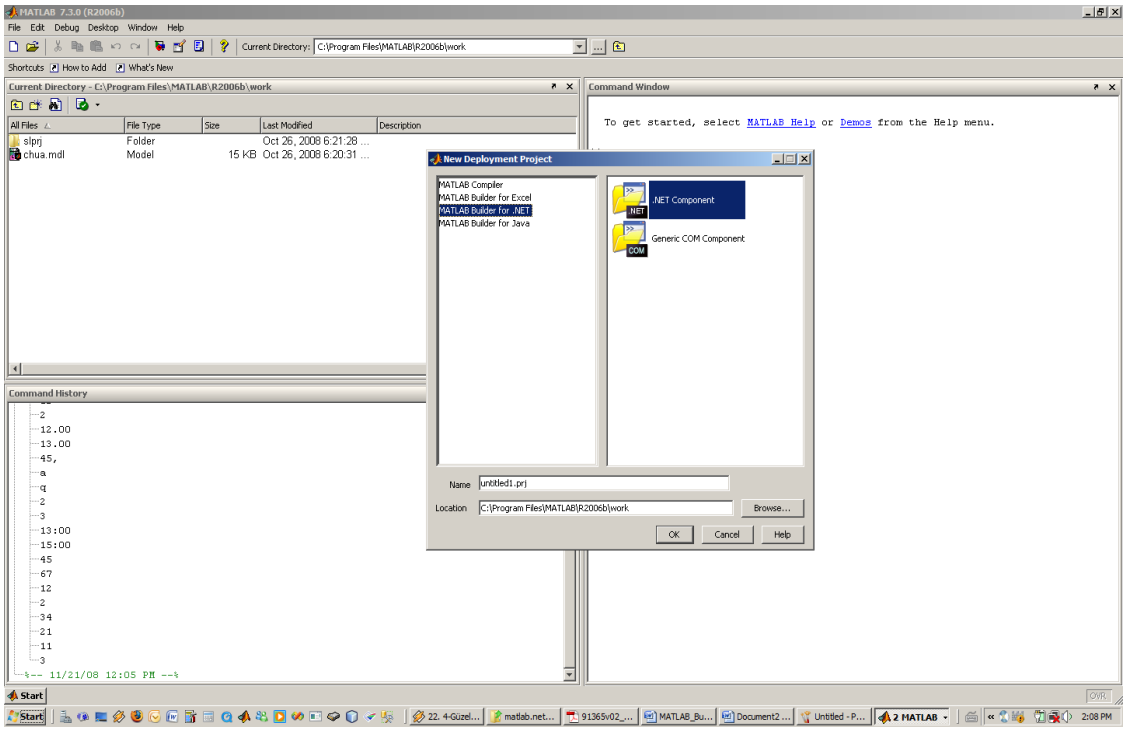
4.1. MATLAB .NET BULDER -Deployment Tool Kullanılarak Matlab .NET ve COM Bileşeni Oluşturmak

1-MATLAB Ana Program Arayüzü açılır. Açılan Ana Arayüzden File sekmesinden New seçeneğinden , Deployment Project seçeneği seçilir.



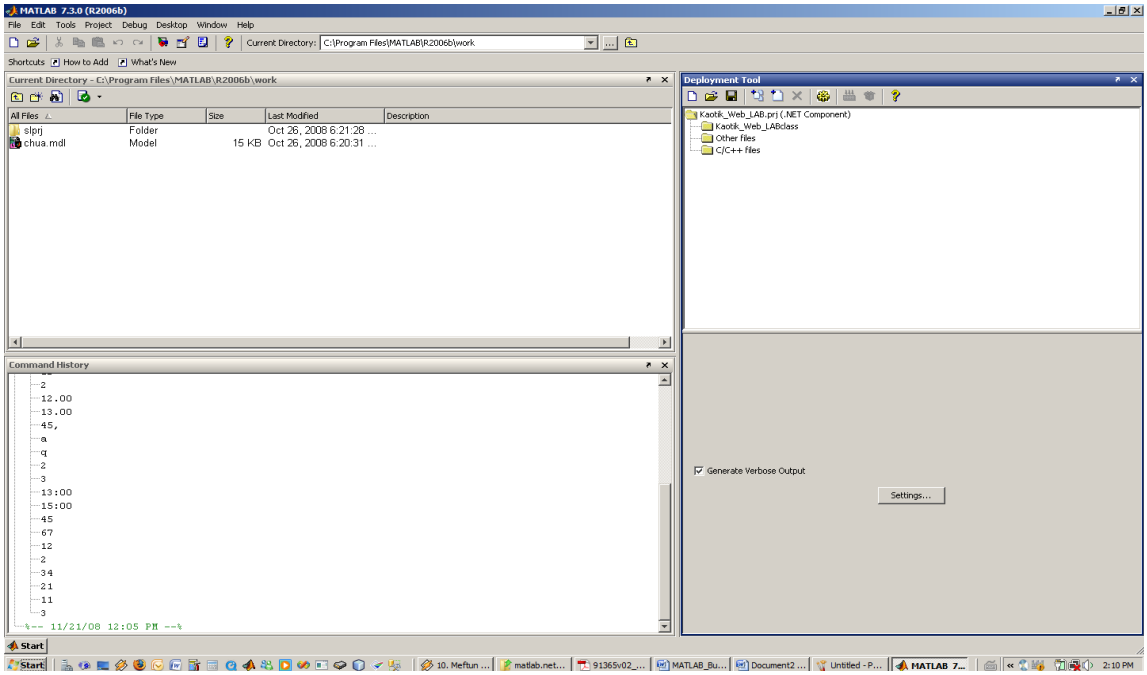
Şekil 4.2. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-1

2-Ekrana açılan New Deployment Project penceresinden ne tür bir bileşen oluşturulacak ,seçilir.(Biz burada .NET Bileşeni oluşturmak istediğimizden , .NET Component seçeneğini seçeceğiz.)



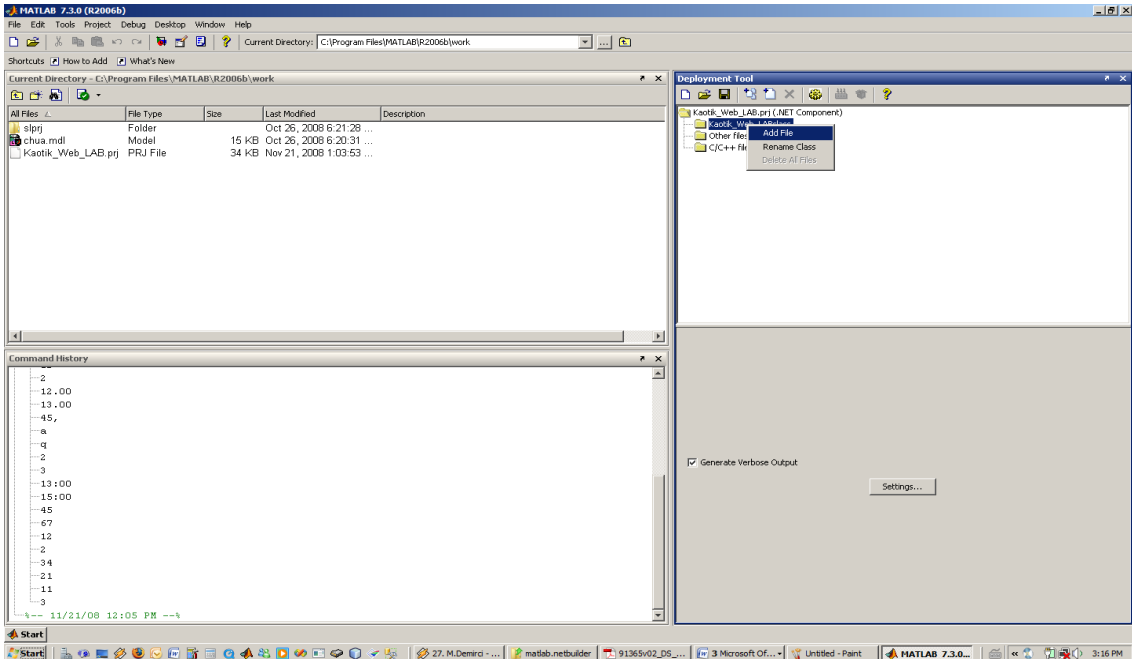
Şekil 4.3. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-2

3- Şekilde de , Görüldüğü üzereAna Arayüzün sağ penceresinde iç bir pencere olarak Deployment Tool penceresi açılır.



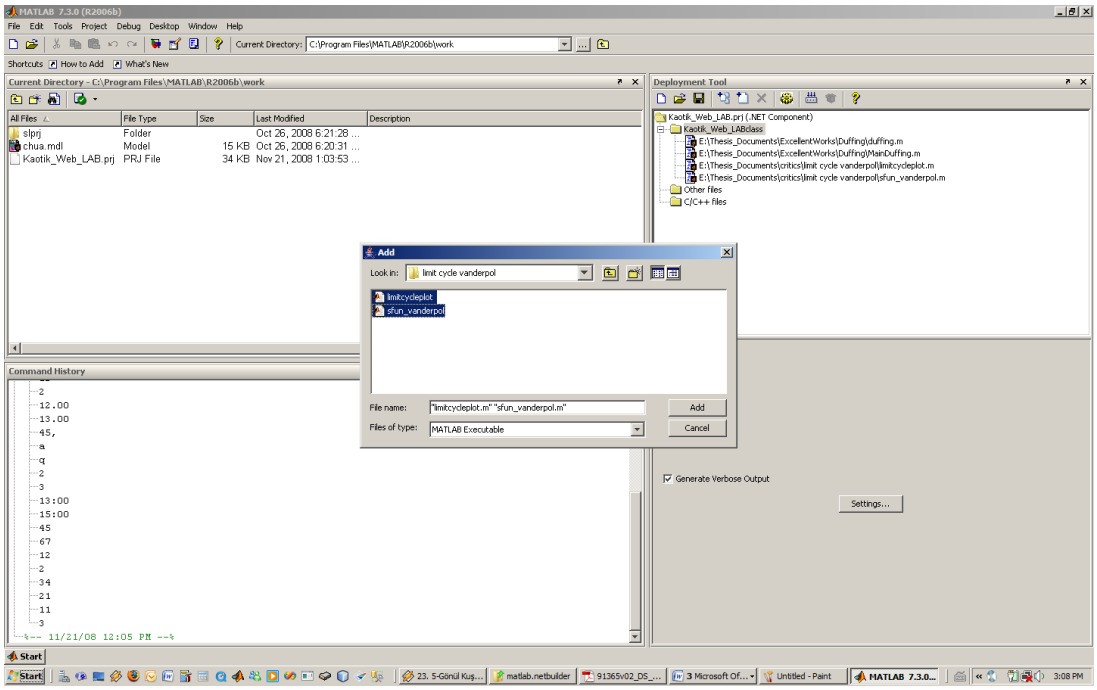
Şekil 4.4. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-3

4- DeploymentTool penceresinde ,Projeyi İsimlendirdiğimiz Class dosyası altında kullanılacak dosyalar gösterilerek , deployment işlemi devam ettirilir.

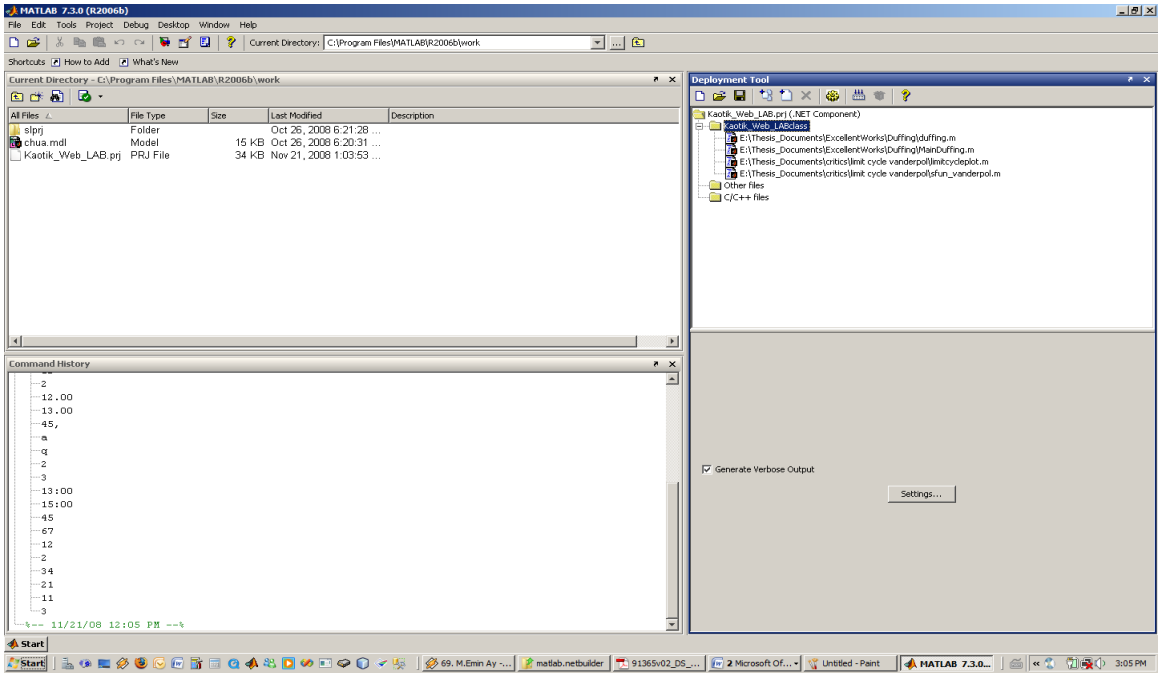


Şekil 4.5. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-4

5-Add File seçeneğine tıklanarak , açılan pencereden MATLAB ortamında –m file veya –mdl uzantılı ve .NET ortamında kullanmak istediğimiz dosyalar gösterilir.Bu işlem projede kullanılacak bütün dosyaları ekleyinceye kadar sürdürülür.

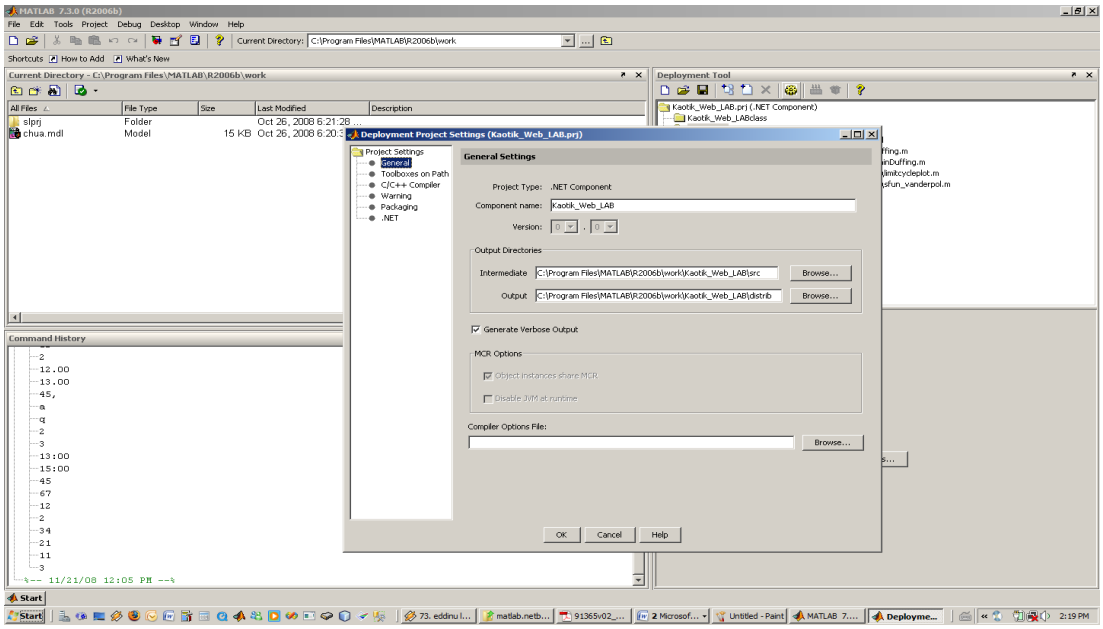


Şekil 4.6. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-5



Şekil 4.7. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-6

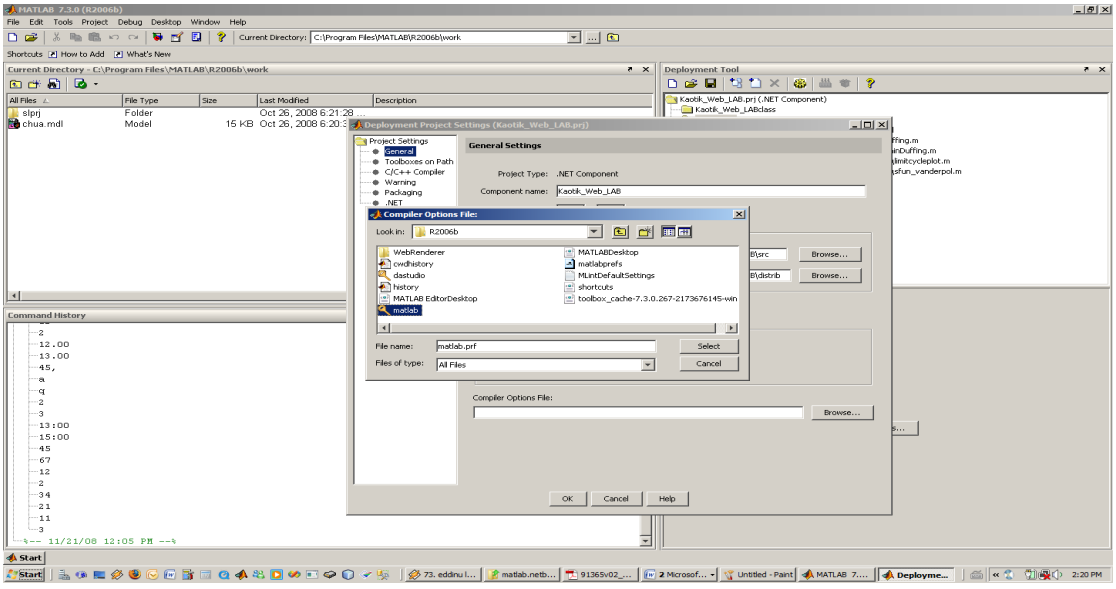
6- Projede kullanılacak Bütün dosyalar eklendikten sonra , Aşağıdaki gibi bir Pencere durumu oluşur.



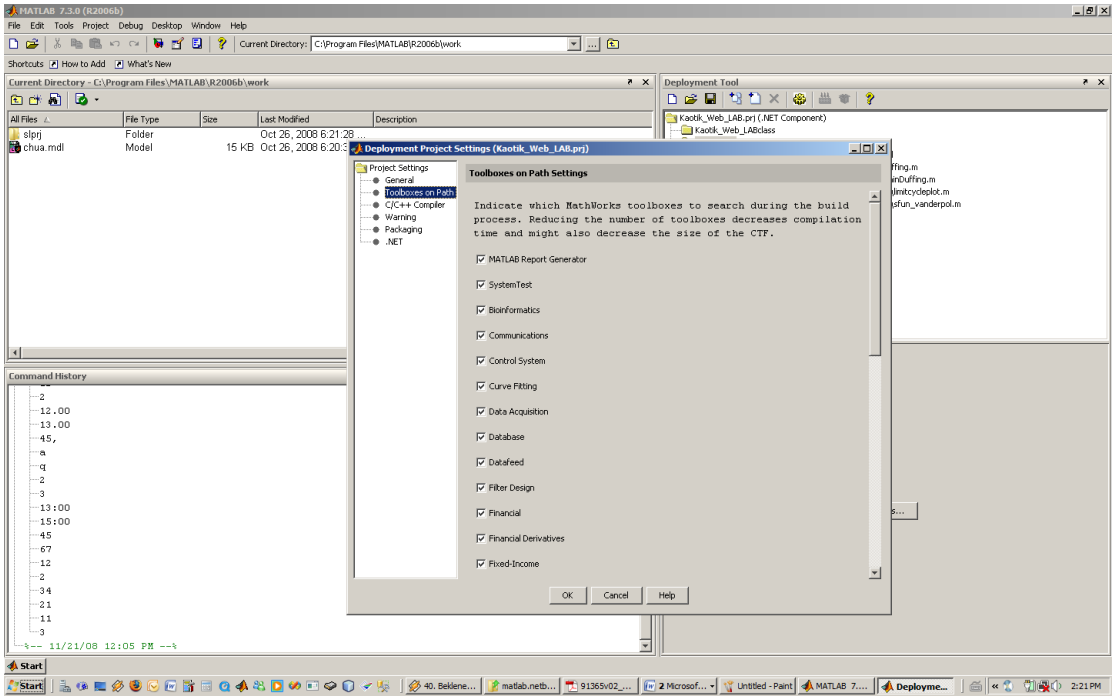
Şekil 4.8. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-7

7-Bir sonraki aşamada , Deployment Tool Penceresinin ortasında yerleşik Settings butonuna tıklanarak ,Deployment Projesi genel ayarlamaları yapılır.Açılan ilk pencerede Generals sekmesi tıklandığında aşağıdaki ekran açılır.Burada Compiler Options kutucuğuna Browse seçeneği tıklanarak , Deployment Projesinde kullanılacak MATLAB Compiler versiyonu gösterilir.

8-Burada biz ana MATLAB derleyicisi olan matlab.prf yi seçeceğiz.



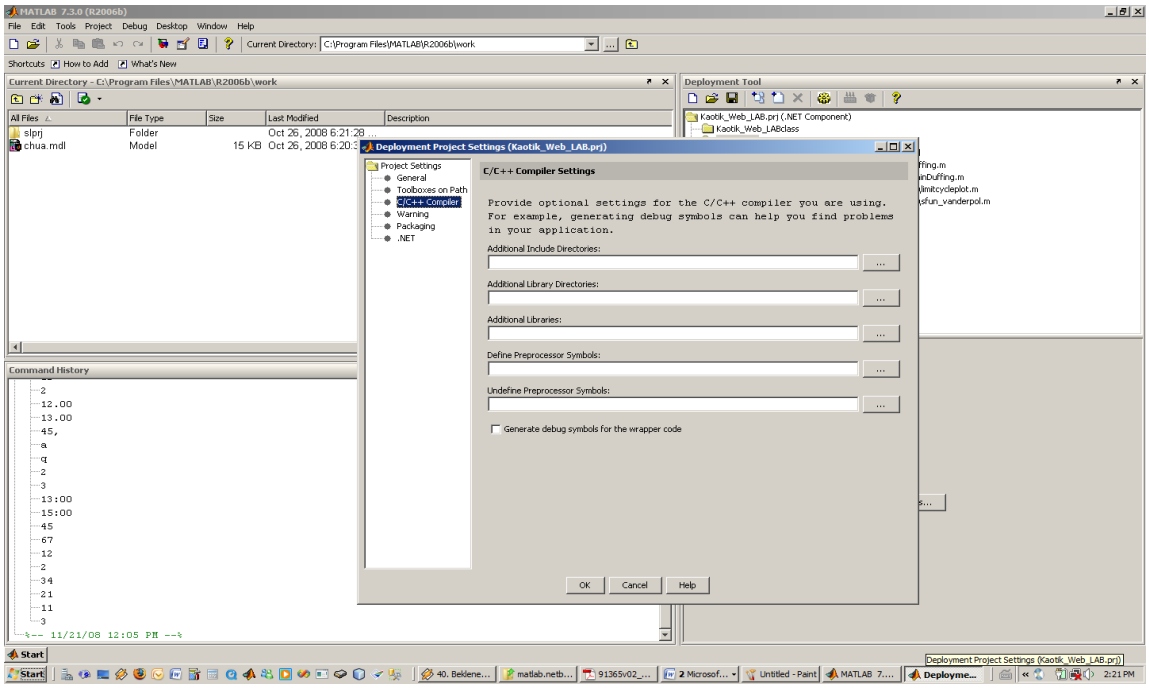
Şekil 4.9. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-8



Şekil 4.10. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-9

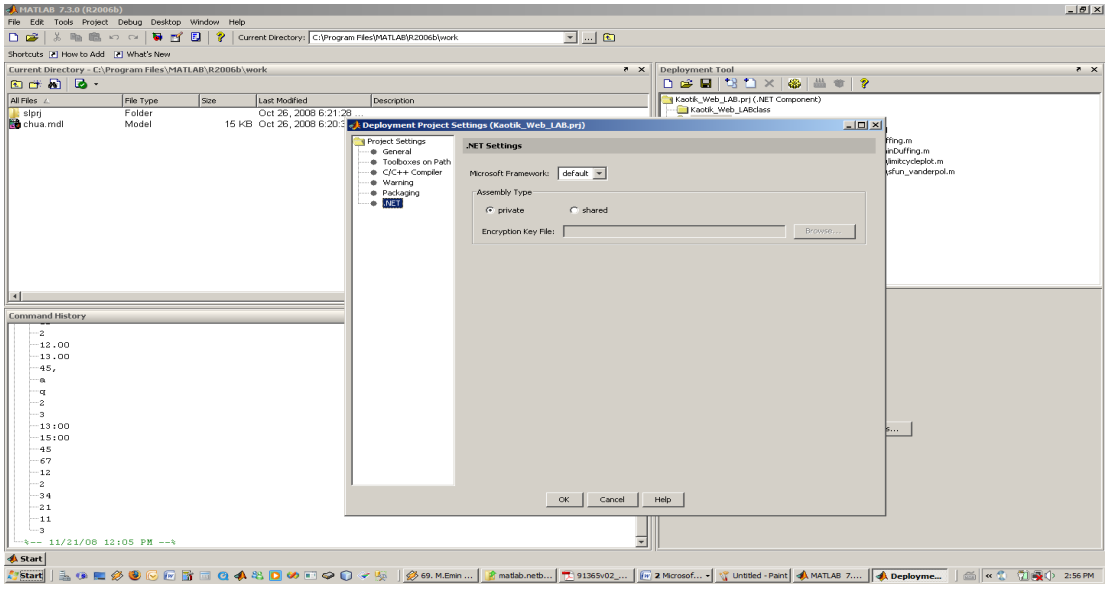
9-Toolbox on Path sekmesine tıklanarak , derleme süresince arkaplanda derleme sürecine dahil edilecek MATLAB Araçları seçilir.Örneğin Kontrol Sistemleri Toolbox ı kullanılarak yapılan bir projede kullanılmayan diğer Toolbox seçeneklerinin işaretli olması derleme süresinin bir kaç kat arttıracağı gibi , oluşan .NET bileşeninin hantallığına sebebiyet verecektir.

10- C/C++ Compiler sekmesine tıklanarak, derleme sürecine C ve C++ derleyicileri I, yardımcı kütüphaneleri , 3. Parti ilave kütüphaneler, ön tanımlama dosyaları, eklenebilir. Bu seçenek özellikle m-file ve .mdl haline getirilmiş MATLAB fonksiyonları içerisinde gömülü C ve C++ kodları yazılması durumunda çarpraz derleme (Cross-Compile) ihtiyacı durumunda kullanılır.



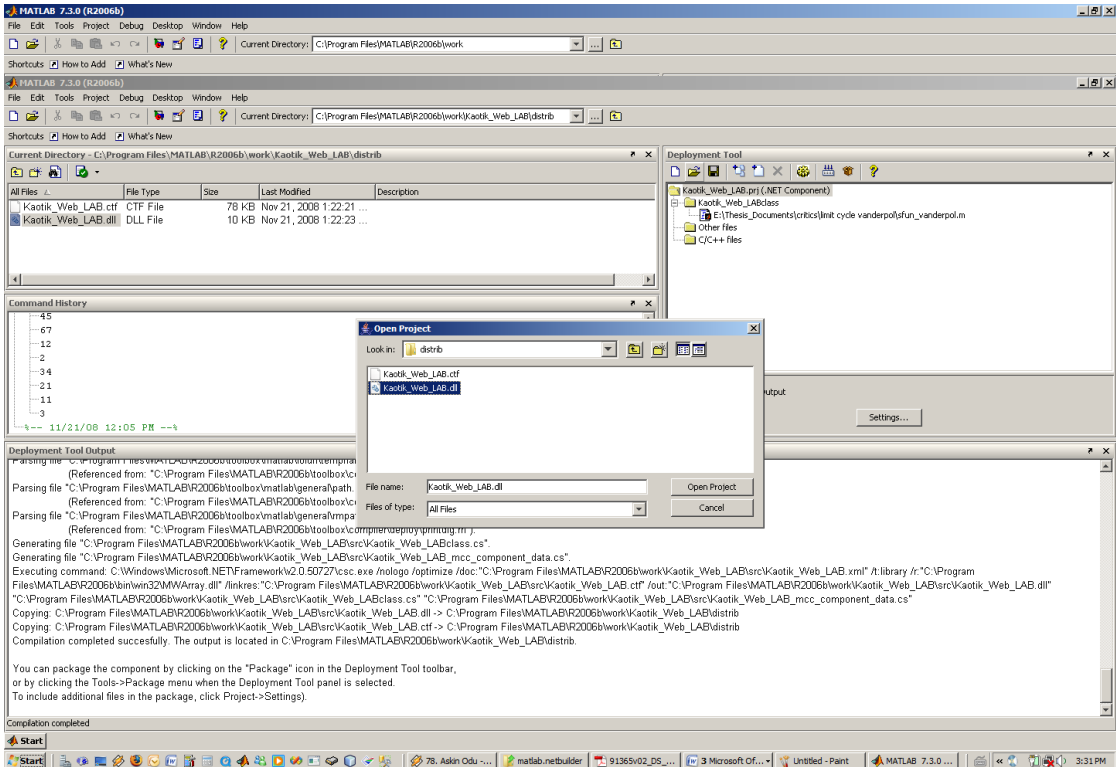
Şekil 4.11. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-10

11-.NET Settings sekmesine tıklanarak, oluşturulacak .NET Bileşeninin hangi .NET Versiyonuna uygun olarak oluşturulması gerektiği , Assembly dosyasının şifre ile dağıtımı sağlanır. Assembly şifresi ile oluşturuşan bu .NET bileşenin ticari amaçla kullanılabilmesi kolaylaştırılmıştır. Böylelikle bileşeni kullanacak olan Uygulama geliştiricilere , ticari anlamda bir Bileşen Lisansı anahtarı sağlanmış olur.



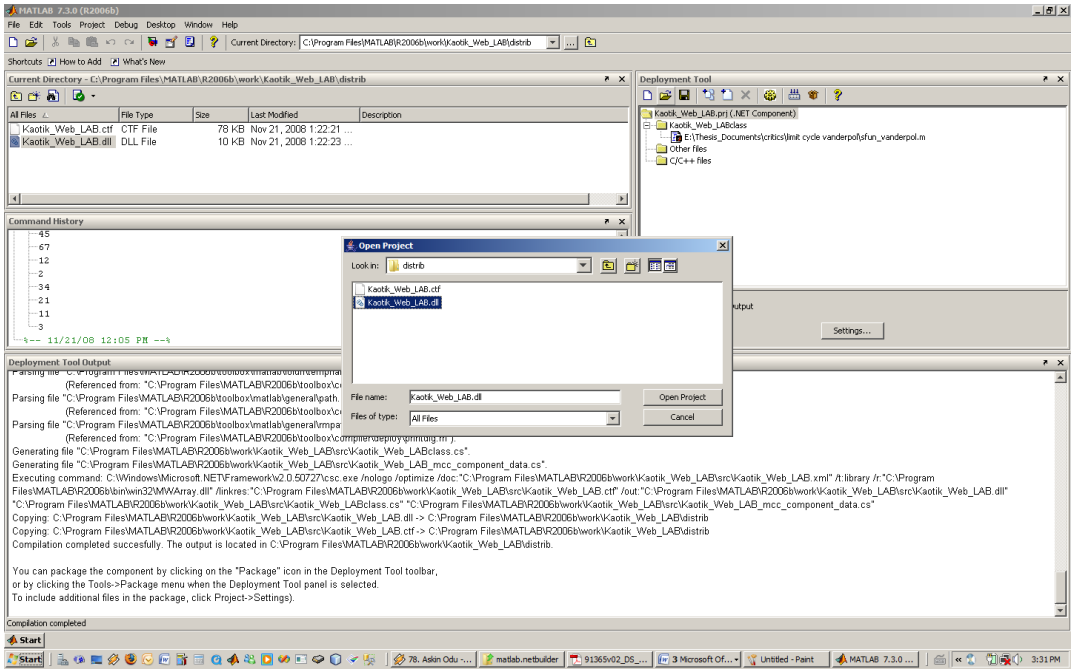
Şekil 4.12. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-11

12- Gerekli ayaralmalar yapıldıktan sonra Deployment Penceresinin sağ üstünde bulunan Compile seçeneği tıklanarak derleme işlemi başlatılır. Bir kaç saniye süren bir işlem sonrasında aşağıdaki gibi bir derleme Çıktı ekranı açılır.Sol alt köşede Derleme işleminin başarılı olup olmadığı da gösterilir.

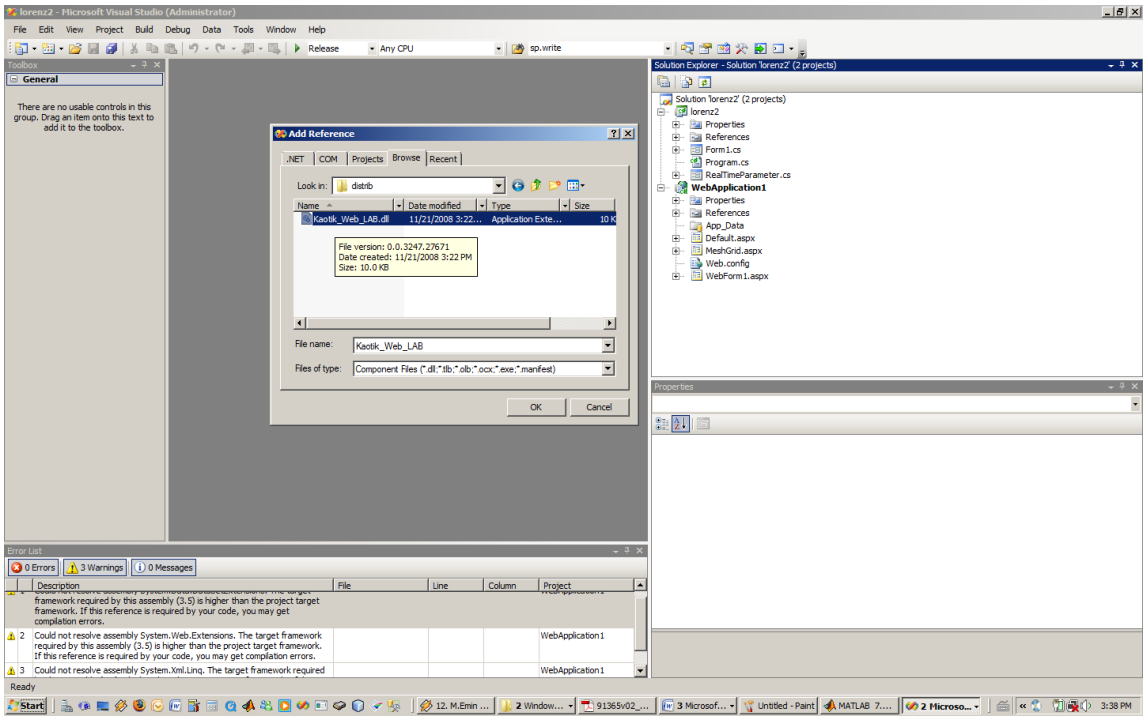


Şekil 4.13. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-12

Böylelikle .NET Bileşeni oluşturma işlemi tamamlanmış olur. Deployment Projesi sonucunda MATLAB Çalıma Zamanı Derleyicisi bize –m file içindeki bütün MATLAB fonksiyonlarımızı içinde barındıran bir ".NET dll" dosyası üretir.Oluşan bu ".NET dll" dosyasını çalıştığımız proje içerisinden referans göstererek MATLAB Fonksiyonlarına Kullanabilir hale getiriririz. İlgili bu ".NET dll " dosyasına Deployment Tool Penceresinin Solunda Sarı Dosya seçeneği tıklanarak açılan pencere altında açılan distrib dosyasının altından ulaşabiliriz. Görüldüğü gibi bu projemizde oluşan dll dosyası "Kaotik_Web_LAB.dll" adında bir .NET dll dosyasıdır.



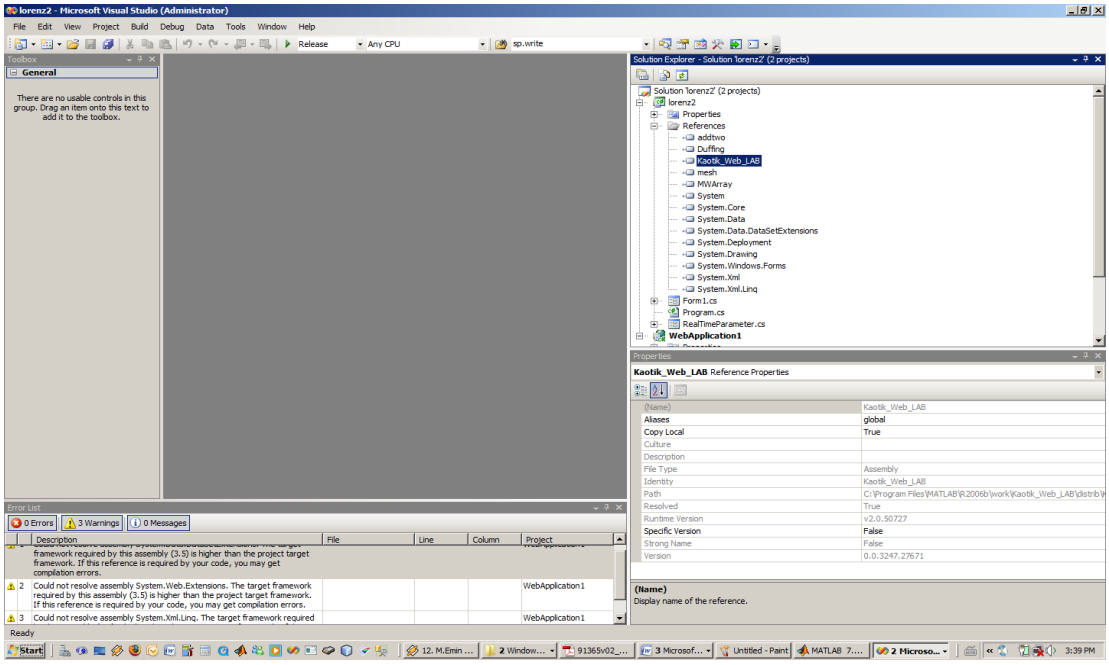
Şekil 4.14. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-13



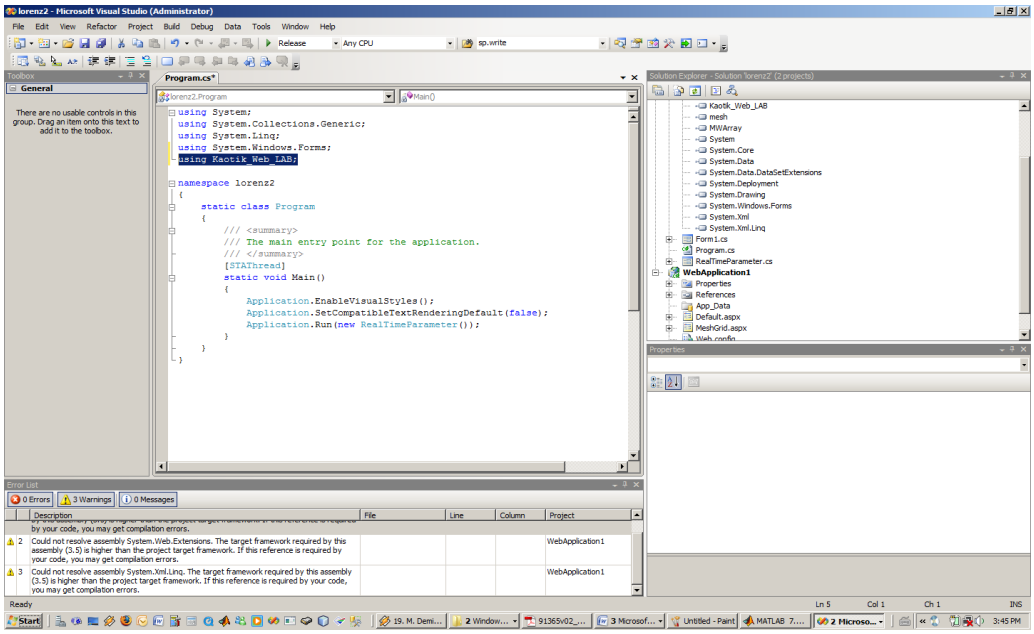
Şekil 4.15. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-14

Eklenen referansın eklendiği References kısmından kontrol edililebilir. Böylelikle MATLAB .NET Builder Toolbox ı ile oluşturduğumuz .NET bileşenimiz , .NET Ortamında kullanılabilir bir forma gelmiştir.

Eklenen referansın eklendiği References kısmından kontrol edililebilir. Böylelikle MATLAB .NET Builder Toolbox ı ile oluşturduğumuz .NET bileşenimiz , .NET Ortamında kullanılabilir bir forma gelmiştir.



Şekil 4.16 . NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-15



Şekil 4.17. NET BULDER Deployment Tool Kullanılarak MATLAB .NET ve COM Bileşeni Oluşturmak Adım-16

Eklemiş olduğumuz referansı .NET te ilgili sınıf içerisinde kullanmak için sınıfın başında bu .NET bileşenini çağırmamız gerekir.

BÖLÜM 5. KAOTİK SİMULASYON LABORATUVARI UYGULAMASI İÇERİK TANITIMI

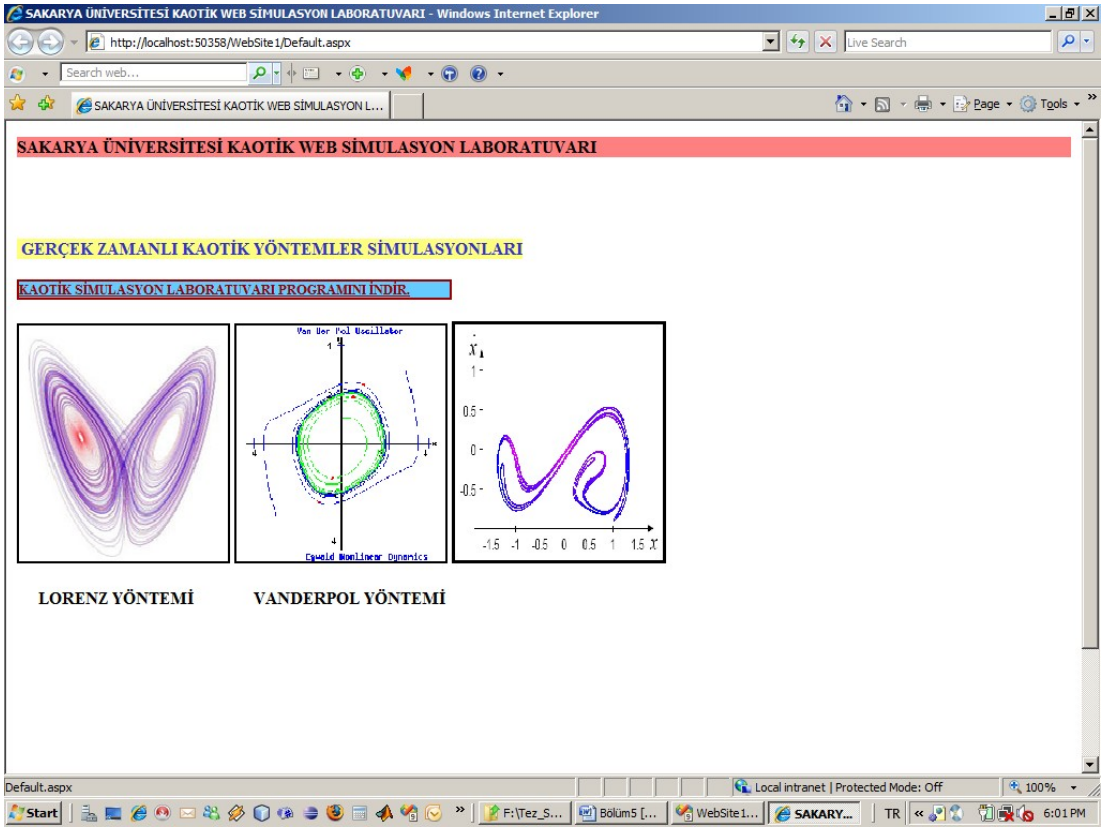
Kaotik Simulasyon Laboratuvarı Uygulamasında,ASP.NET web programlama dili kullanılarak Kaotik Uygulamaların yayımlandığı bir Web sayfası ve ayrıca bu sayfa üzerinden C# .NET nesne yönelimli yazılım dili ve Visual Studio Görsel Araç kutusu kullanılarak geliştirilen , interaktif bir Kaotik Simulasyon Programı da yayımlanarak ziyaretçilerin Kaos Teorisini daha yakından tanıması ve bu konuda akademik dünyada yapılan çalışmaları zengin bir içerik vasıtasıyla pekiştirmesi amaçlanmıştır.

Kaotik Simulasyon Laboratuvarı Uygulamasında, Kaos teorisi ile ilgili tanımlamalar, Kaotik Analiz Metodları, Kaotik Bileşenler Kaos Teorisi Hakkındaki yayınlanan Kitaplar ve Makaleler, Kaotik Denklem modelleri ve Kaotik Uygulama Örnekleri yer almaktadır. Kaotik Simulasyon Laboratuvarı Uygulamasında bulunan gerek görsel içerikli, gerekse de yazılı zengin içeriklerle Kaos Teorisini ve mühendislik alanlarındaki Uygulamalarının etkin olarak tanıtılması amaçlanmıştır.Belirli başlangıç koşulları altında Chua, Vanderpol, Rossler,Lorenz,Duffing, ve Colpitts gibi Kaos Yöntemleri ile MATLAB Ortamında yapılan Simulasyonların Eksenel Çıkış Diyagramları alınarak Uygulamanın zengin görsel içeriği hazırlanmıştır.

Bundan başka Simulasyon Uygulamasında ,arka planda MATLAB Hesaplama Motoru kabiliyetleri kullanılarak, bir mühendislik problemine MATLAB Hesaplama Motoruna ,Simulasyon Aracı üzerinden ilgili Kaotik Analiz Yöntemi Denklem Takımlarına gerçek zamanlı parameter geçilerek kaotik hesaplama sonuçlarını hem grafiksel hem sayısal büyüklüklerle gösterme ve bu sonuçları analiz etme imkanı vardır.

5.1. Kaotik Web Simulasyon Laboratuvarı Sayfası

Kaotik WEB Simulasyon Sayfasına girilerek Kaotik Simulasyon Programı indirilir.



Şekil5.1. Kaotik WEB Simulasyon Sayfası

5.2. Kaos Nedir Sekmesi

Kaos Nedir Sekmesi Kaosun akademik anlamda yaygın olan tanımlamaları gösterilmektedir. Her bir tanımlamaya tanım numarası butonu tıklanarak ulaşılır.

Şekil 5.2. Kaosun 1. Tanımı Sekmesi

Programdan alınan 2. Tanımlama Sekmesi içeriği Şekil 5.3 de görülmektedir.

Şekil 5.3. Kaosun 2. Tanımı Sekmesi

Programdan alınan 3. Tanımlama Sekmesi içeriği Şekil 5.4de görülmektedir.

Şekil 5.4. Kaosun 3. Tanımı Sekmesi

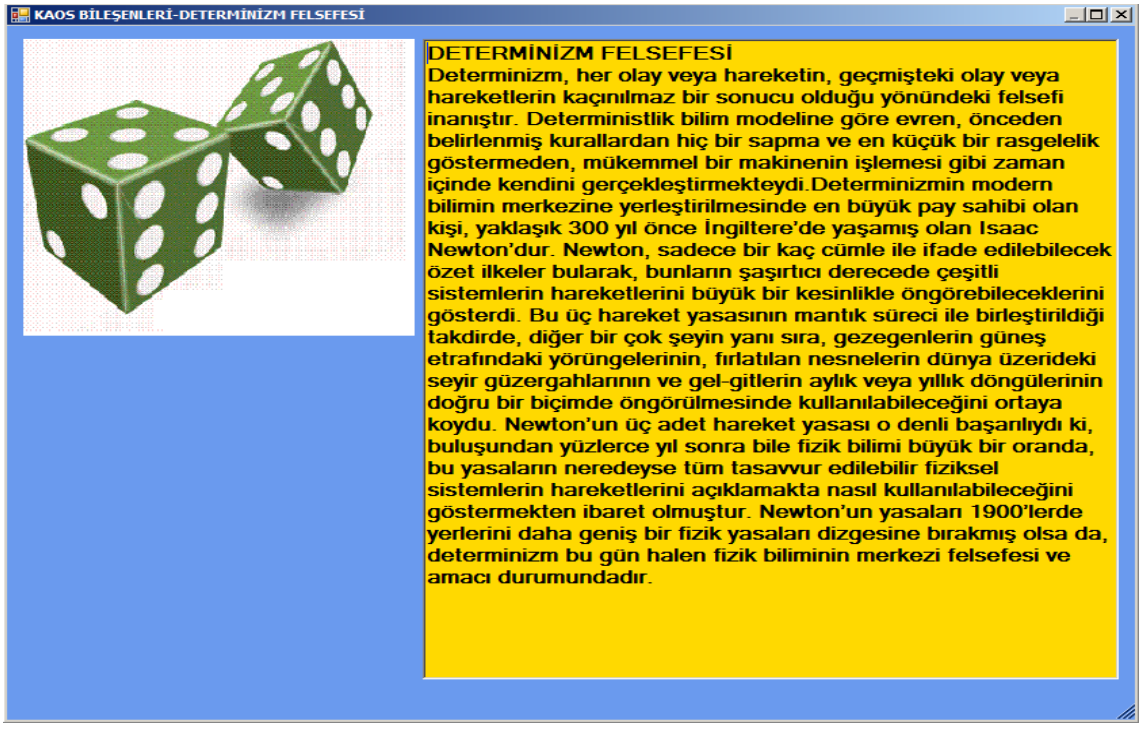
5.3. Kaos Bileşenleri Sekmesi

Kaos Bileşenleri Sekmesinde Kaos Teorisinin üzerine Bina Edildiği 4 Temel Kuram Anlatılmaktadır.

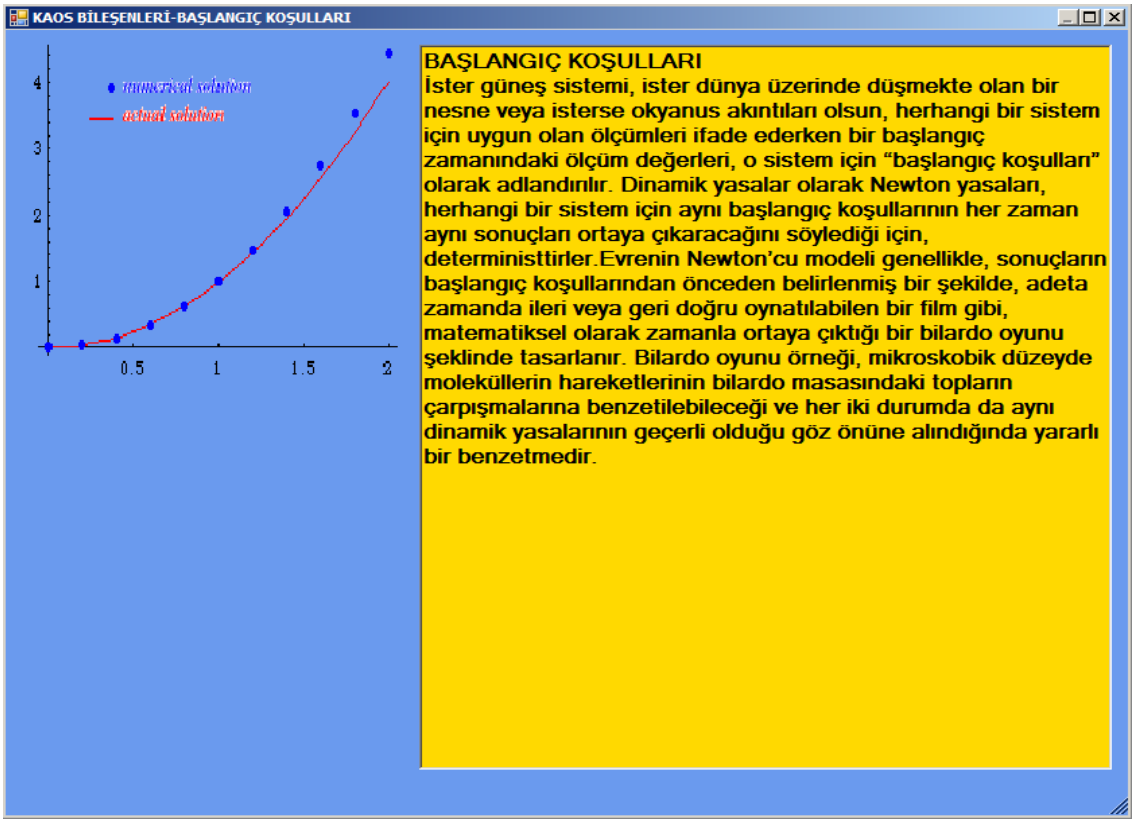
.

Şekil 5.5. Kaos Bileşenleri Sekmesi

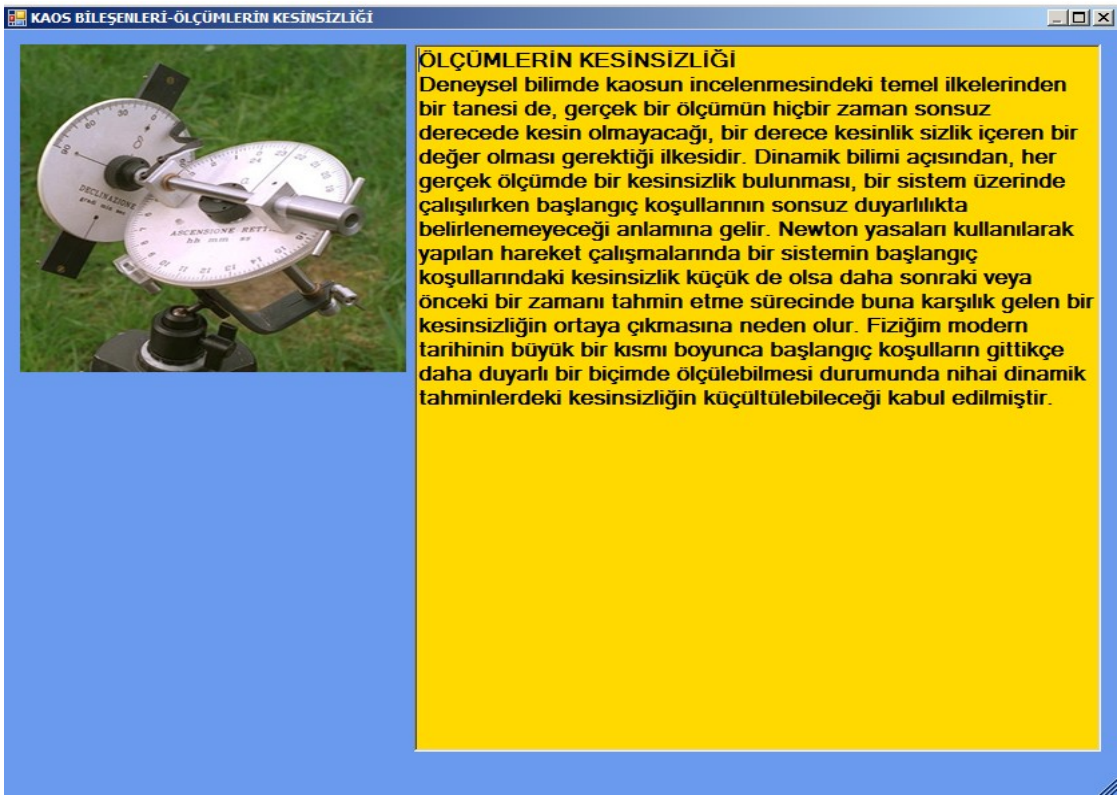
Herbir Kuramın ayrı olarak anlatıldığı sayfalar aşağıda görülmektedir.



Şekil 5.6. Kaos Bileşenlerinden Determinism Kuramı Sekmesi



Şekil 5.7. Kaos BileşenlerindenBaşlangıç Koşulları Kuramı Sekmesi

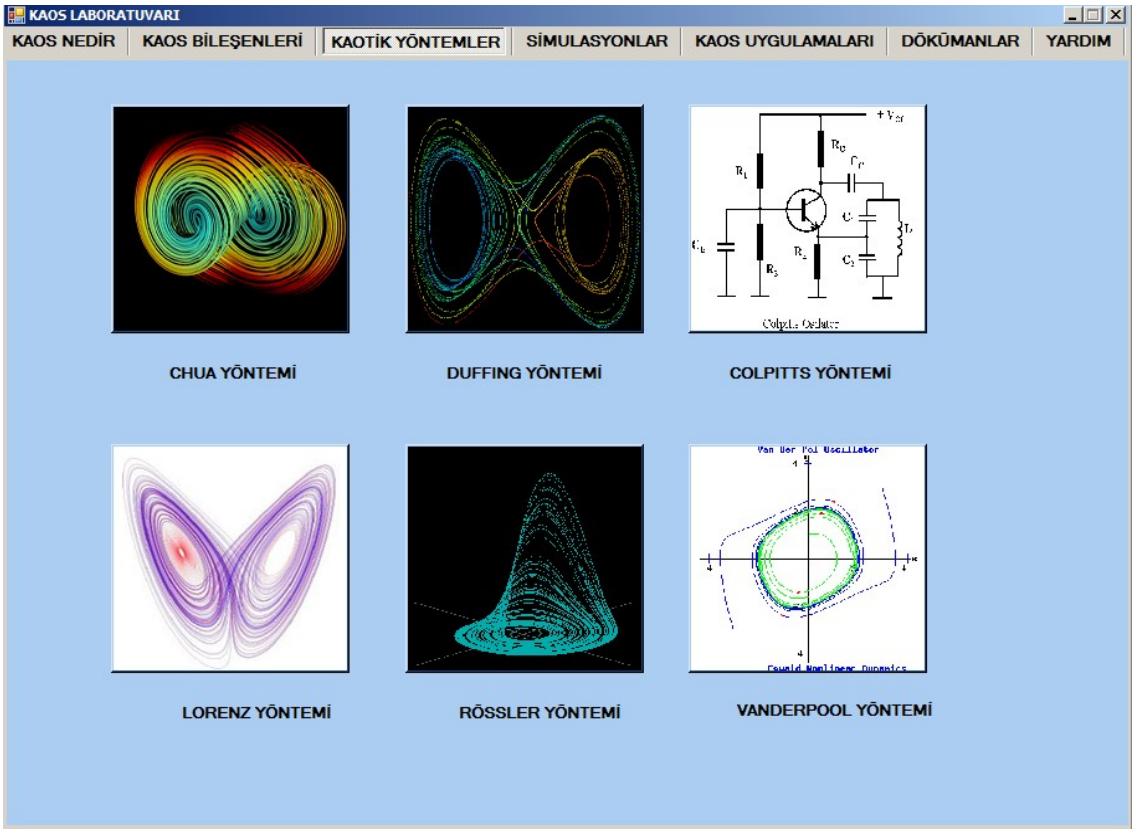


Şekil 5.8. Kaos Bileşenlerinden Ölçümlerin Kesinsizliği Kuramı Sekmesi

Şekil 5.9. Kaos Bileşenlerinden Dinamik Karasızlıklar ve Çatallaşma Kuramı Sekmesi

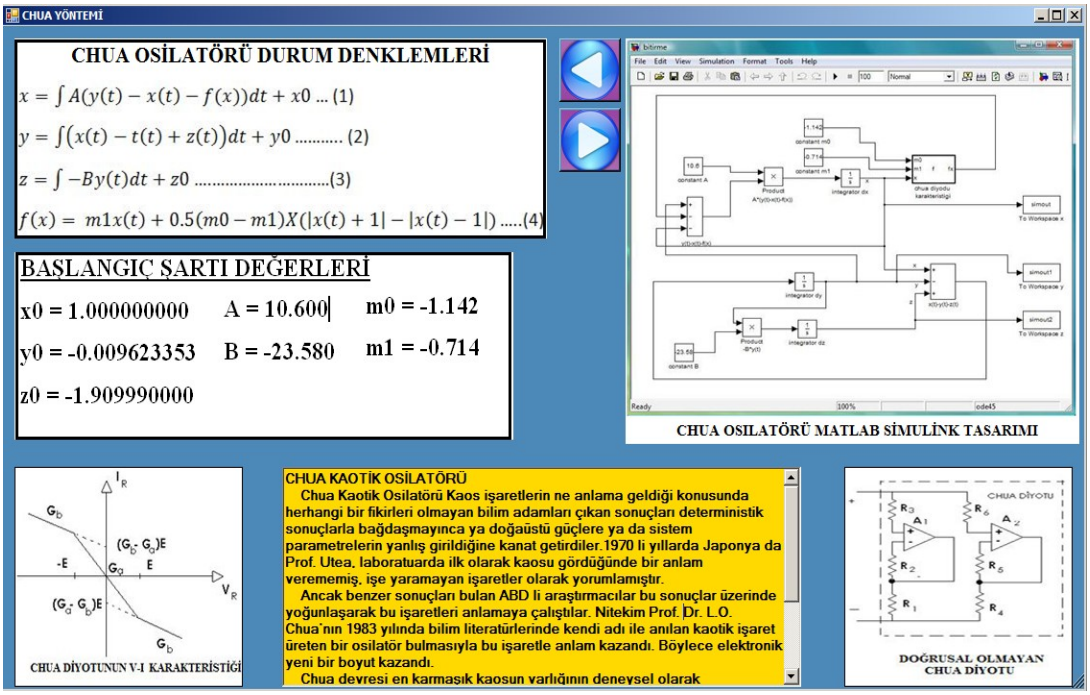
5.4. Kaotik Yöntemler Sekmesi

Kaotik Yöntemler Sekmesinde Akademik dünyada en çok kullanılan 6 Kaotik Analiz Yöntemi ele alınmıştır. Her bir Yönteme ilişkin zengin Görsel içerik hazırlanmış olup kullanıcının Kaos Kuramlarını pekiştirmesi amaçlanmıştır



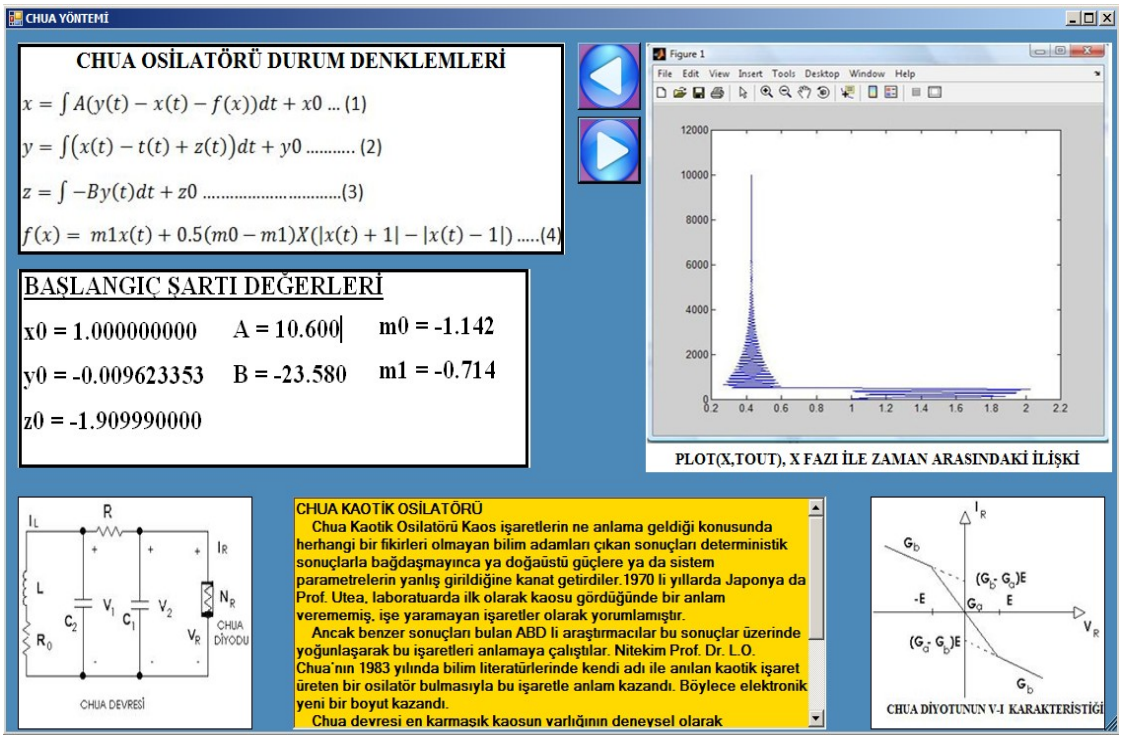
Şekil 5.10. Kaotik Analiz Yöntemleri Sekmesi

Burada Chua yöntemi ayrıntılı görsel içeriği üzerinde durulacaktır.



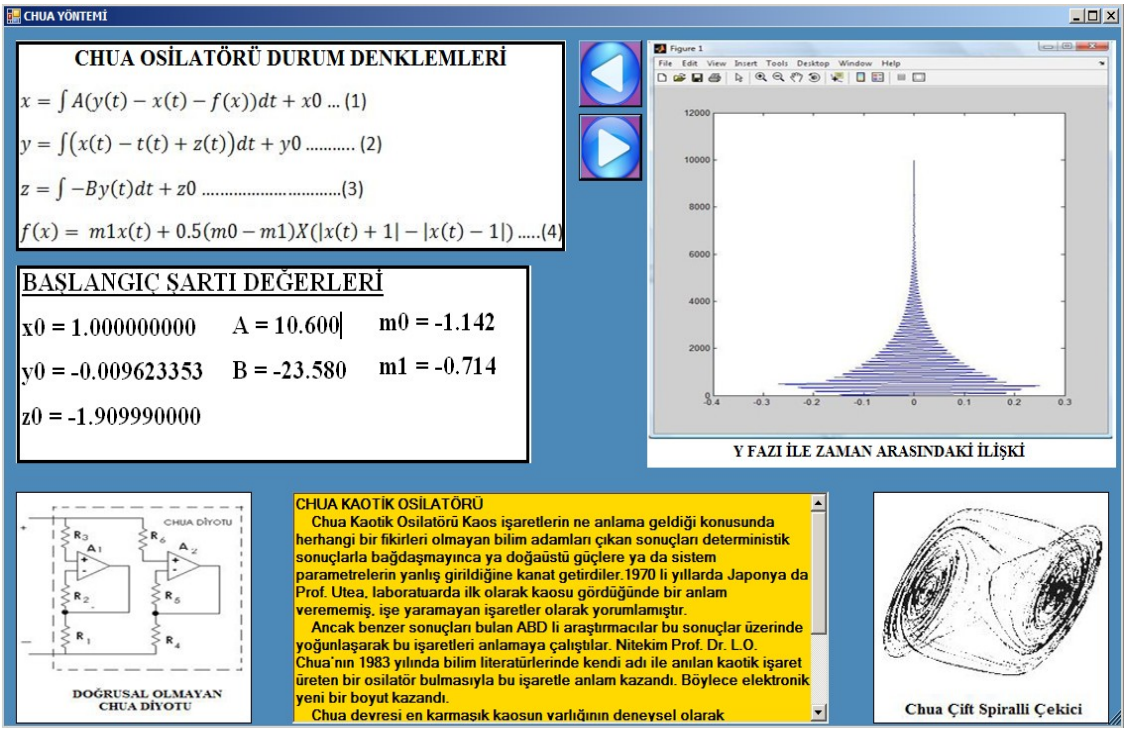
Şekil 5.11. Chua Kaotik Analiz Yöntemi Sekmesi 1.Kısım

İlgili Parametreler için Eksenel MATLAB Simulasyon çıktıları Görülmektedir. Ok işareti ile gösterilen butonlarla Simulasyon Çıktıları arasında geçiş yapılabilmektedir.

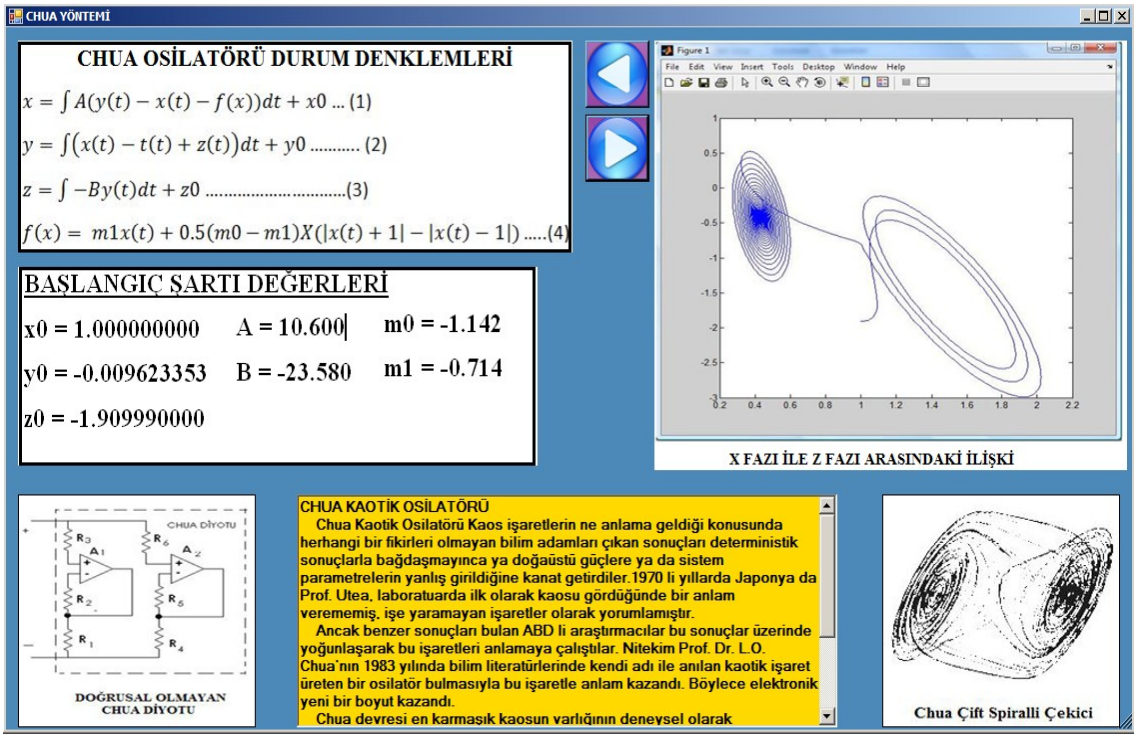


Şekil 5.12. Chua Kaotik Analiz Yöntemi Sekmesi 2.Kısım

Butonlarla görsel gezinti devam ettiriliyor. Bu arada Chua Osilatörü hakkında bilgilerde ScrollBardan izlenebilir. Bundan başka resimler üzerine tıklanarak diğer görsel nesnelerde görülebilir.



Şekil 5.13. Chua Kaotik Analiz Yöntemi Sekmesi 3. Kısım

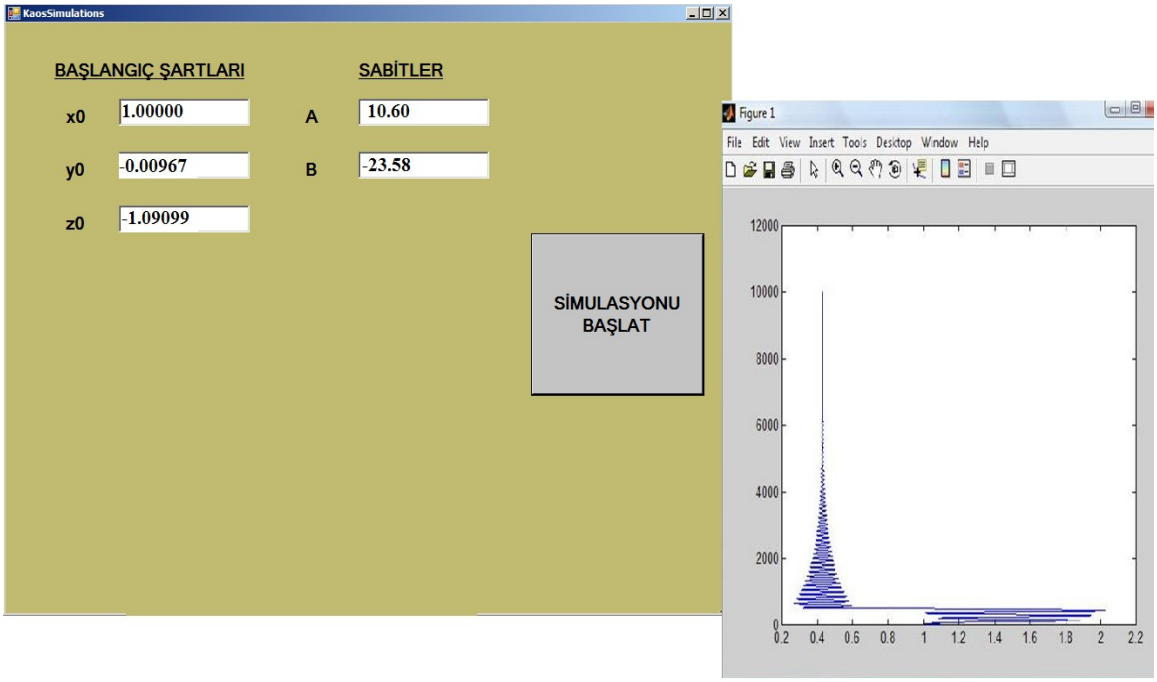


Şekil 5.14. Chua Kaotik Analiz Yöntemi Sekmesi 4. Kısım

5.5. Kaos Simulasyonlar Sekmesi

Kaos Simulasyonlar Sekmesinde Lorenz , Chua ve Rossler Kaotik Analiz Yöntemleri ile değişken Başlangıç Değerleri ve Sabitler girilerek ,MATLAB kabiliyetleri kullanılarak Analizler yapılmaktadır.

Şekil 5.15. Kaos Simulasyonları Sekmesi



Şekil 5.16. Chua Yöntemi Kullanılarak Gerçek Zamanlı parametre Geçilerek Yapılan Simulasyon Örneği

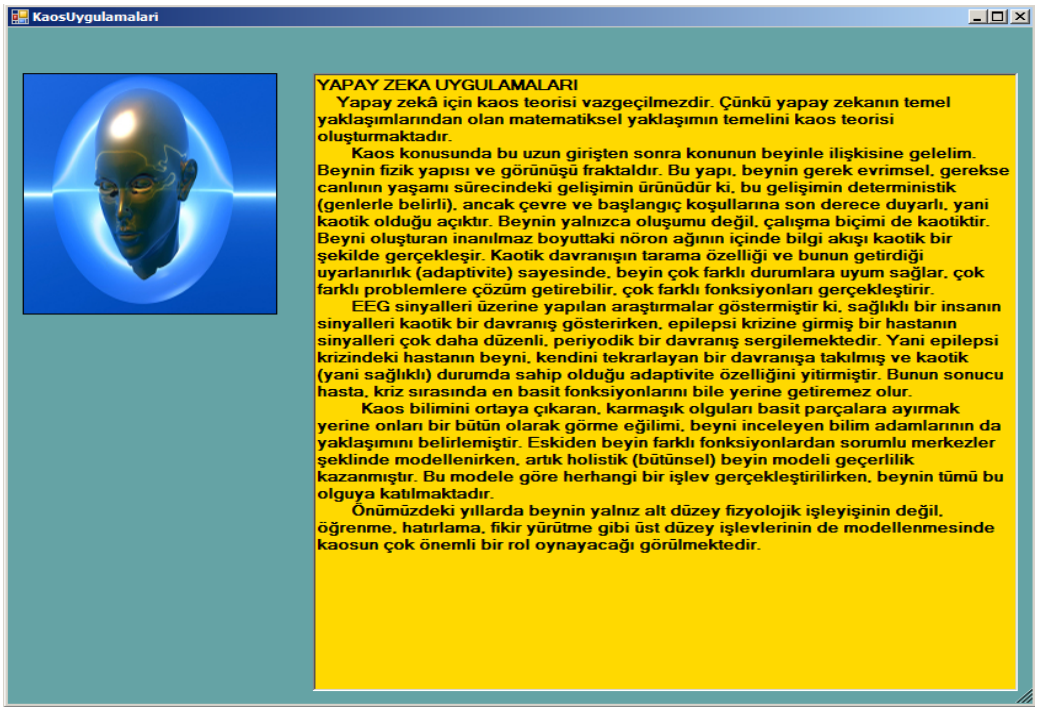
Burada gerekli parametreler ilgili alanlara girildikten sonra Simulasyonu Başlat Butonuna basılarak hesaplama yapılır. Program gerçek zamanlı simulasyon amaçlı olarak ilk çalıştırıldığı zaman MATLAB Hesaplama Motorunun 2 saniyelik bir hazırlanma gecikmesi oluşmaktadır. Program kapatılmadan yapılacak her bir parametre değerleri için tekrar bir gecikme oluşmayacaktır.

5.6. Kaos Uygulama Alanları Sekmesi

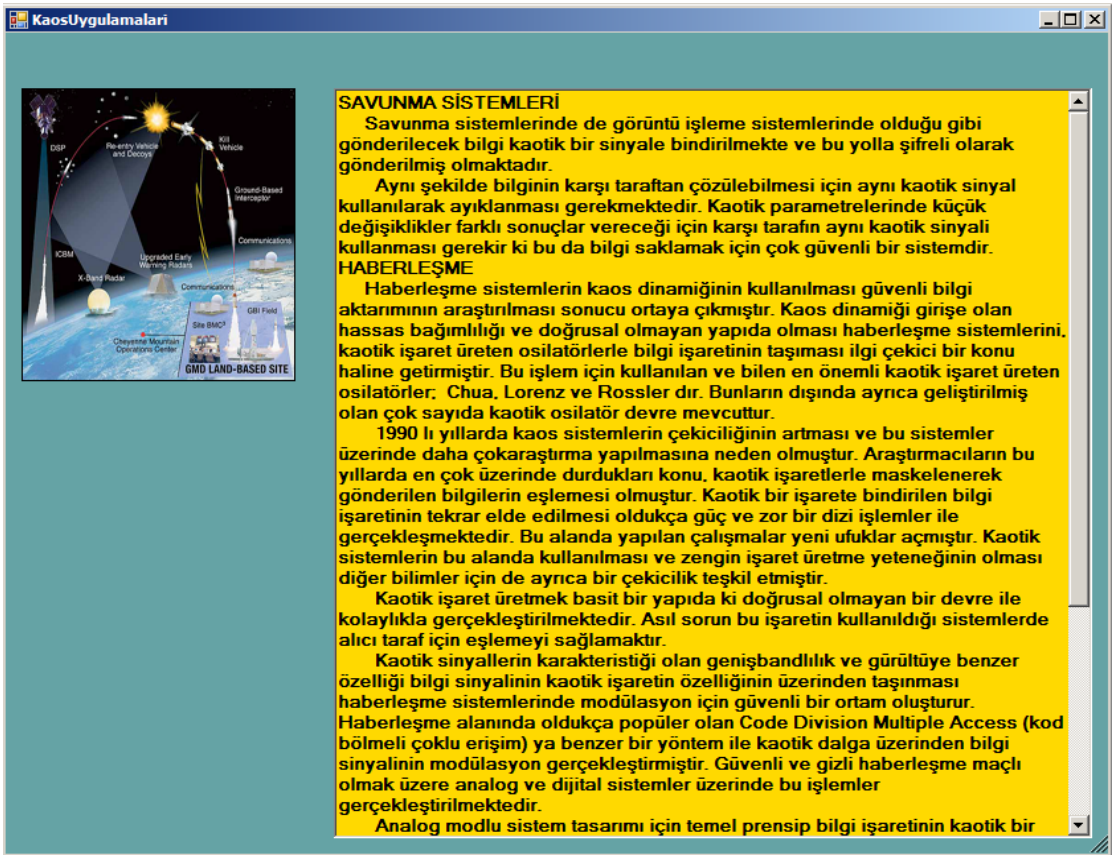
Kaos Uygulama Alanları Sekmesinde günümüzde Kaotik Yöntemlerin kullanıldığı sahalar anlatılmaktadır.

Şekil 5.17. Kaos Uygulama Alanları Sekmesi

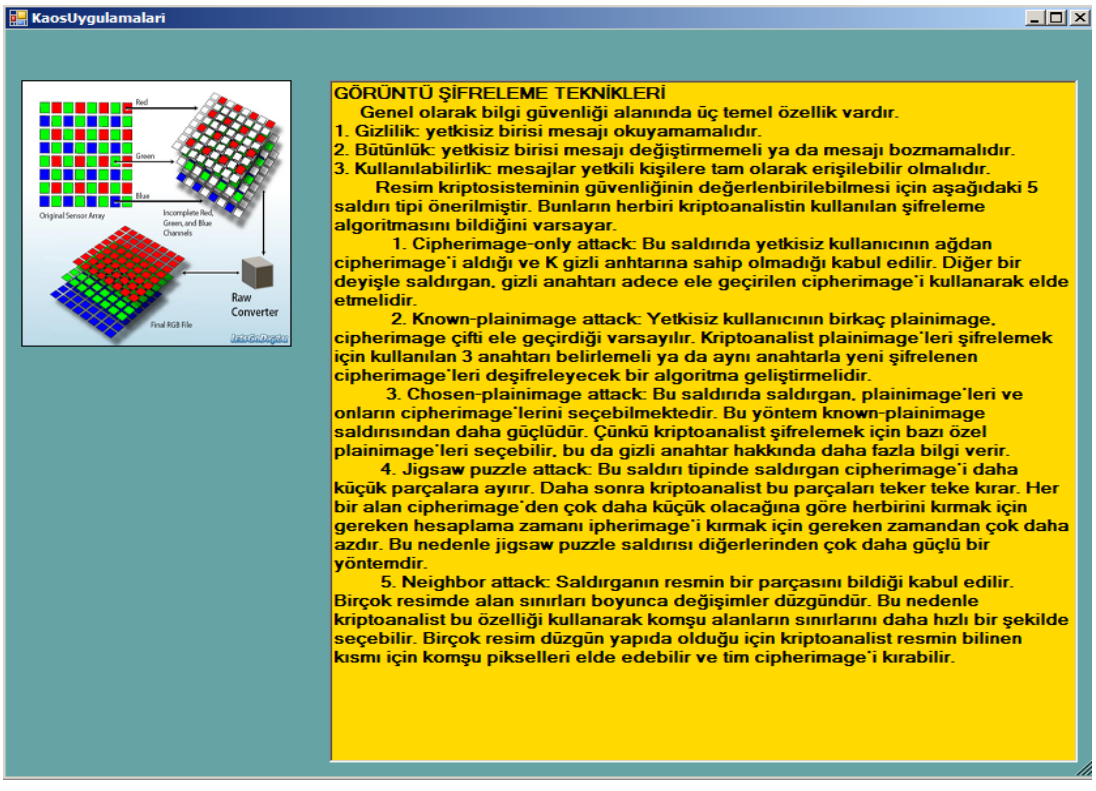
Herbir uygulama alanı Butonunun tıklanmasıyla ilgili uygulama alanı için ayrı birer sekme açılmaktadır.



Şekil 5.18. Kaos Uygulama Alanları Yapay Zeka Sekmesi



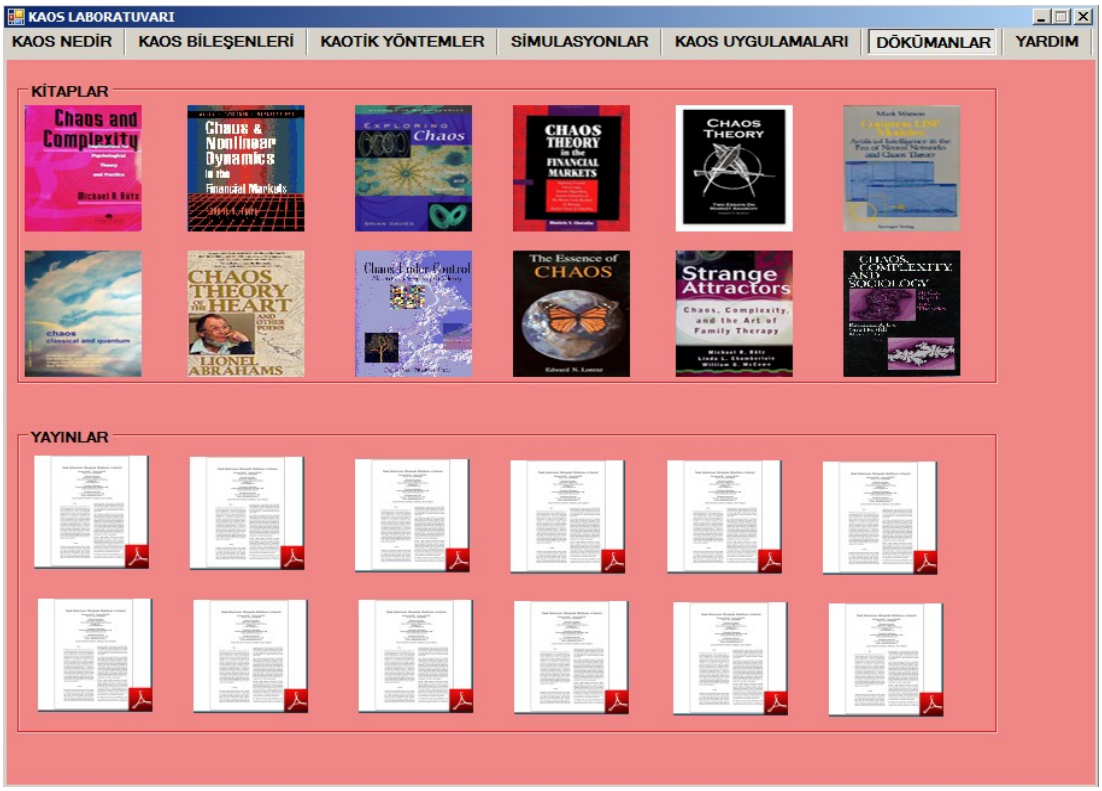
Şekil 5.19. Kaos Uygulama Alanları Savunma Sistemleri Sekmesi



Şekil 5.20. Kaos Uygulama Alanları Görüntü Şifreleme Teknikleri Sekmesi

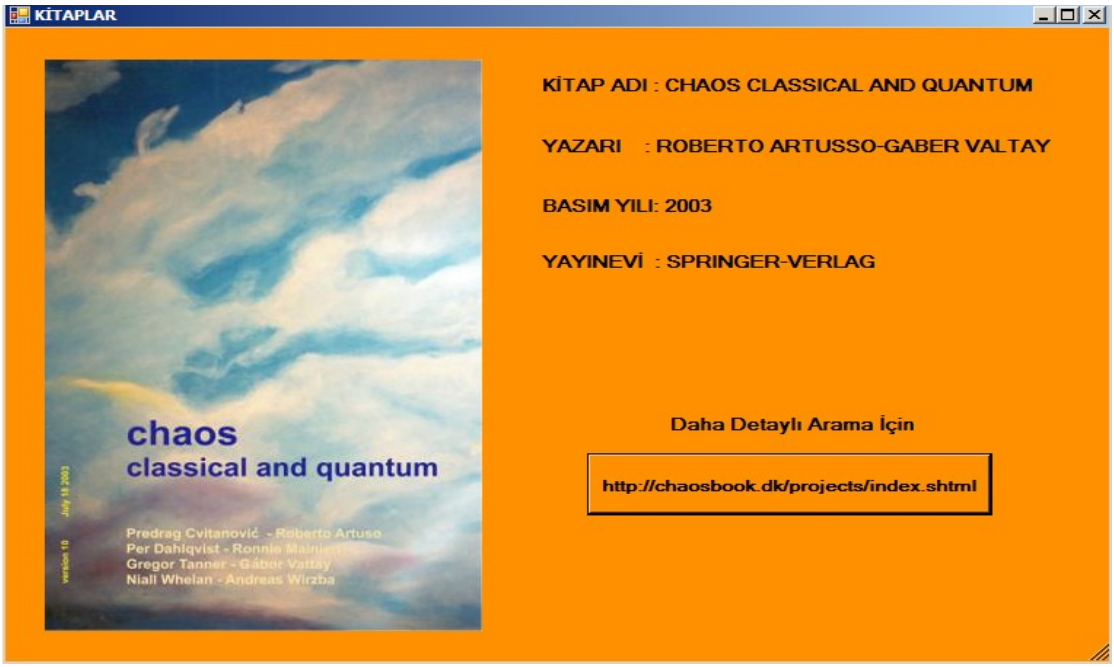
5.7. Kaos Dökümanları Sekmesi

Kaos Dökümanları Sekmesinde Kaos Teorisi ile ilgili yayınlanmış belli başlı kitaplar bu konuda hazırlanan bilimsel makaleler hakkında poster bilgileri yayınlanmıştır. İlgiki dökümanlarla ilgili daha fazla ayrıntılar için gerekli linkler eklenmiştir.



Şekil 5.21. Kaos Teorisi Döküman Sekmesi Kısım-1

İlgili resim üzerine tıklanarak yayınla ilgili ayrıntılara ulaşılır.



Şekil 5.22. Kaos Teorisi Döküman Sekmesi Kısım-2

BÖLÜM 6. SONUÇ VE ÖNERİLER

E-öğrenme, internet tabanlı eğitim modelleri için kullanılan genel bir kavramdır. e-öğrenme sistemlerinin, uzaktan eğitimde kullanılan geleneksel yöntemlere göre en önemli üstünlüğü, alıcı ile kaynağın eş zamanlı yada eş zamansız olarak karşılıklı etkileşim içerisinde bulunabilmeleridir. Özellikle son zamanlarda gelişen teknoloji sayesinde, bilgiyi alan ile bilgi kaynağı internet üzerinden gerçek zamanlı olarak esli ve/veya görüntülü konuşma yapabilmekte, algılamada sorun çekilen konular üzerinde tartışabilmektedirler.

Kaotik Simulasyon Laboratuvarı tez çalışmasında ,Mühendislik problemlerin çözümünde bugüne kadar kullanılana gelen kaotik yaklaşım modellerini sayısal ortam aracılığıyla daha etkin ve gözlemlenebilir bir araç olarak kullanılmasının mümkün olduğu, çalışmalarla ortaya konulmuştur.

Simulasyon Laboratuvarı Uygulamasında Sayısal Ortam araçları olarak ASP.NET ve C# ın daha etkin ve farklı geliştirme ortamlarıyla esnek bir dil olduğu görülmüştür.

Bundan sonraki çalışma önerisi olarak donanımsal bir kaotik uygulamanın Web üzerinden gerçek zamanlı kontrol edilebileceği önerilebilir.

KAYNAKLAR

[1] IRMAK, E., Uzaktan Eğitim Amaçlı İnternet Tabanlı Laboratuvar Uygulaması, Gazi Üniversitesi, Elektrik Eğitimi, Doktora Tezi, sf. 4-6 2007

[2] IRMAK, E., Uzaktan Eğitim Amaçlı İnternet Tabanlı Laboratuvar Uygulaması, Gazi Üniversitesi, Elektrik Eğitimi, Doktora Tezi, sf. 8-9 2007

[3] IRMAK, E., Uzaktan Eğitim Amaçlı İnternet Tabanlı Laboratuvar Uygulaması, Gazi Üniversitesi, Elektrik Eğitimi, Doktora Tezi, sf. 1-2 2007

[4] KOŞALAY, İ.,Yrd. Doç Dr. -TRT Genel Müdürlüğü,Öğrenme ve Teknoloji, II. Uluslararası Bilgisayar ve Öğretim Teknolojlleri Sempozyumu**,** İzmir, sf. 1, 2008.

[5] CHRİSTOS, H. S., Chaotic Modelling And Simulation: Analysis Of Chaotic Models, Attractors And Forms, Technical University Of Crete, Greece

[6] WALTHER, S., Part IX: Sample ASP.NET Applications, ASP.NET Unleashed, Second Edition

[7] WALTHER, S., "Part IV:Working with ASP.NET Applications", ASP.NET Unleashed, Second Edition

[8] GOSNEY, J., CHAPTER 6 Using Forms, Premier ASP Programming , 109

[9] GOSNEY, J., Working with ASP Objects, Premier ASP Programming , 49

[10] NAR, Y.,Elektr.Müh,www.noktavirgul.com Moderatörü, http://www.noktalivirgul.com/ASP_NET_Nedir__p284.aspx, 27 Nisan 2008

[11] NAR, Y.,Elektr.Müh,www.noktavirgul.com Moderatörü, http://www.noktalivirgul.com/ASP_NET_Nedir__p284.aspx , 27 Nisan 2008

[12] THAI LEE, T., LAM , H.,CHAPTER-D Common Utilities,.NET Framework Essentials –Second Edition ,244-246

[13] J.C. Sprott C., Simple Chaotic Systems and Circuits ,Am. J. Physics. 68, 758-763., 2000.

[14] G. CHEN and X. Yu, (2003), Chaos Control: Theory and Applications., Berlin, Germany: Springer- Verlag,.

[15] LÜ, J., Chen, G.et.al., Dynamical analysis of a new chaotic attractor, Int. J. Bif. and Chaos 12, 2002.

[16] LORENZ, E.N., Deterministic Nonperiodic Flow, J.Atmos. Sci., 20:130–141, 1963

[17] HOLMES, PJ., Poincare celestial mechanics, dynamical-systems theory and "chaos", *Phys. Rep.*, 1990;193(3):138-163.

[18] YANG, T., A survey of chaotic secure communication systems, International Journal of Computational Cognition, 2004;2(2):81–130.

[19] CUOMO, KM., OPPENHEİM AV., Circuit Implementation of Synchronized Chaos with applications to Communication, *Phys. Rev. Lett.*, 1993;71:65-68.

[20] PEHLİVAN, İ., UYAROGLU Y., Rikitake Attractor and it's synchronization application for secure communication systems, *Journal of Applied Sciences*, 2007;7(2):232-236

[21] PEREZ, JM. Mechanism for global features of chaos in a driven nonlinear oscillator.,Phys Rev A 1985;32:2513.2516.

[22] KHADRA, A., LİU, X., SHEN ,X., Application of Impulsive Synchronization to Communication Security, IEEE Transactions on Circuits and Systems I, 2003;50(3):341-351.

[23] Lİ, Z., Lİ, K., WEN, C, SOH, YC., A New Chaotic Secure Communication System, *IEEE Transactions on communications*, 2003;51(8);1306-1312

[24] BLAKELY, JN., CORRON, NJ., Multiplexing symbolic dynamics-based chaos communications using synchronization, *Journal of Physics: Conference Series*, 2005; 23:259–266

[25] Lİ, Z., Lİ, K., WEN, C, SOH, YC., A New Chaotic Secure Communication System, *IEEE Transactions on communications*, 2003;51(8);1309-1318

[26] KHADRA, A., LİU, X., SHEN, X., Application of Impulsive Synchronization to Communication Security, IEEE Transactions on Circuits and Systems I, 2003;50(3):347-357.

[27] KHADRA, A., LİU, X., SHEN, X., Application of Impulsive Synchronization to Communication Security, IEEE Transactions on Circuits and Systems I, 2003;50(3):367-377.

[27] DOBSON, I, "Computing a closest bifurcation instability in multidimensional parameter space", Journal of Nonlinear Science, vol. 3, no. 3, pp.307-327, 1993.

[28] CUOMO, KM., OPPENHEİM, AV., Circuit Implementation of Synchronized Chaos with applications to Communication, *Phys. Rev. Lett.*, 1993;71:69-73.

[29] CUOMO, KM., OPPENHEİM, AV., Circuit Implementation of Synchronized Chaos with applications to Communication, *Phys. Rev. Lett.*, 1993;71:78-79.

[30] Lİ, Z., Lİ, K., WEN, C., SOH, YC., A New Chaotic Secure Communication System, *IEEE Transactions on communications*, 2003;51(8);1309-1318

[31] CHEN, G. & DONG, X., "From Chaos to Order: Methodologies"; Perspectives and Applications (World Scientific, Singapore), 1998

[32] CUOMO, K.M., and OPPENHEIM, A.V., "Circuit Implementation of Synchronized Chaos with Applications to Communications", Physical Review Letters, vol. 71, 65, 1993.

[33] NAKAGAWA, S. and SAITO, T., (1996), An RC OTA hysteresis chaos Generator IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 43, no. 12, pp. 1019–1021.

[34] MASANOBU, N., TAKASHİ, M. E., "Pecora-Carroll Chaotic Synchronization and Masking in an R-L-Diode Circuit",8. Electronics and Communications in Japan (Part-III: Fundamental Electronic Science) Volume 83, Issue 2, Date: February 2000, Pages: 44-54

[35] PECORA, LM., CARROLL, TL., Driving systems with chaotic signals, *Physical review A* , 1991;44:2374-2383

[36] INABA MORİS, N., Chaotic Phenomena in a Circuit with a Diode due to Change of the Oscillation Frequency, *IEICE Trans. Fundamentals.*, 1988;E71:842-849.

[37] CUOMO, KM., OPPENHEİM, AV., STROGATZ, S. H., Synchronization of Lorenz-based chaotic circuits with applications to communications, *IEEE Trans. Circuits Syst.*, 1993;40(10):626–633.

[38] LU, J., CHEN, G., "A new chaotic attractor coined", Int. J. Bifurcation and Chaos, 12, 659-661, 2002.

[39] ROSSLER, O. E. "Continuous chaos;four prototype equations"; Ann.(N.Y.) A. Sci., 316, 376-392. 1979.

[40] PEHLİVAN, İ., UYAROGLU, Y., "Simplified Chaotic Diffusionless Lorenz Attractor and its Application to Secure Communication Systems", IET Communications, 1, 5, 1015-1022, 2007

[41] ÖZOĞUZ, S., ELWAKIL, A. & KENNEDY, M., "Experimental verification of the butterfly attractor in a modified Lorenz system," IEEE Trans. Circuits Syst. I, 2001.

[42] LÜ, J., CHEN G., CHENG, D., A new chaotic system and beyond: The generalized Lorenz-Like system , *Int. J. Bifurcation and Chaos* , 2004;14 (5):1507-1537.

[43] ELWAKIL, A. S. And SOLIMAN, A., "Chaos from a family of minimum-Component oscillators", Chaos, Solitons, and Fractals, 8, 335-356., 1997.

[44] LORENZ, E.N., "Deterministic nonperiodic flow", J.Atmos. Sci., 20:130–141, 1963

[45] RÖSSLER, OE., An equation for continuous chaos, *Phys. Lett. A,* 1976;57:397–398.

ÖZGEÇMİŞ

Metin VARAN 20.11.1981 de Bingöl' de doğdu. İlk, orta ve lise eğitimini İstanbul'da tamamladı. 1997 yılında Mehmet Niyazi Altuğ Süper Lisesi, Matematik Bölümünden mezun oldu. 2003 yılında başladığı Sakarya Üniversitesi Elektrik-Elektronik Mühendisliği bölümünü 2006 yılında bitirdi. Temmuz 2006 yılından beri İstanbul-ANADOLU Bilişim AŞ. adlı Yazılım Şirketinde, Yazılım ve Uygulama Geliştirme Mühendisi pozisyonunda çalışmaktadır.