



T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**DİNAMİK SİSTEMLERDE ÇÖZÜMLERİN
HAREKETİNİN PARAMETRİK OLARAK
DEĞERLENDİRİLMESİ VE HASSASİYET
ANALİZİ**

REYYAN NUR GÖKTAŞ

YÜKSEK LİSANS

Matematik Anabilim Dalı

Haziran-2025
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Reyyan Nur Göktaş tarafından hazırlanan “Dinamik Sistemlerde Çözümlerin Hareketinin Parametrik Olarak Değerlendirilmesi Ve Hassasiyet Analizi” adlı tez çalışması 27/06/2025 tarihinde aşağıdaki jüri tarafından oy birliği ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof. Dr. Kemâl AYDIN

Danışman

Prof. Dr. Kemâl AYDIN

Üye

Dr. Öğr. Üyesi Ali Osman ÇIBIKDİKEN

Üye

Dr. Öğr. Üyesi Güner TOPCU

Yukarıdaki sonucu onaylarım.

Prof. Dr. Hasan AYDOĞAN
FBE Müdürü

Bu tez çalışması Selçuk Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü tarafından 24201001 nolu proje ile desteklenmiştir.

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Reyyan Nur GÖKTAŞ

Tarih : 27.06.2025

ÖZET

YÜKSEK LİSANS TEZİ

DİNAMİK SİSTEMLERDE ÇÖZÜMLERİN HAREKETİNİN PARAMETRİK OLARAK DEĞERLENDİRİLMESİ VE HASSASİYET ANALİZİ

Reyyan Nur GÖKTAŞ

Selçuk Üniversitesi Fen Bilimleri Enstitüsü

Matematik Anabilim Dalı

Danışman: Prof. Dr. Kemâl AYDIN

2025, 132 Sayfa

Jüri

Prof. Dr. Kemâl AYDIN

Dr. Öğr. Üyesi Ali Osman ÇIBIKDİKEN

Dr. Öğr. Üyesi Güner TOPCU

Dinamik sistemlerde çözümlerin hareketi mühendislikten biyolojiye, ekonomiden epidemiyolojiye kadar geniş bir disiplin yelpazesinde önemli bir araştırma alanı olarak öne çıkmaktadır. Zamana bağlı olarak değişen durumları matematiksel modeller aracılığıyla tanımlayan bu sistemler, dış etkilere, başlangıç koşullarına veya sistemlerin giriş elemanlarının (giriş parametrelerinin) değişimlerine bağlı olarak karmaşık davranışlar sergileyebilir. Çözümlerin uzun vadedeki hareketi, sistemin kararlılık özellikleri ile doğrudan bağlantılıdır.

Lineer dinamik sistemler için çözümlerin hareketi, kararlılık özellikleri ve kararlılığın kalitesi kavramları iyi bilinmektedir. Bunun için çözümleri hesaplamadan sistemin hareketi ve kararlılığın kalitesini gösteren kararlılık parametreleri (Schur ve Hurwitz kararlılık parametreleri, sırasıyla ω ve κ) olarak adlandırılan parametreler bulunmaktadır. Kararlılık parametrelerine bağlı olarak bu sistemlerin kararlılık özelliklerinin dış etkilere dayanıklılığının tespiti hassasiyet analiziyle belirlenebilmekte, genellikle literatürde süreklilik teoremleri olarak bilinen teoremler yardımıyla yapılabilmektedir.

Bir problemin istenen özelliklerinin korunabilmesi için gereken şartların araştırılması ters problem olarakta bilinmektedir. Bir ters problem olarak sistemlerin davranışının kararlılık özelliğini ele almak, parametrik giriş elemanları için kararlılık aralıkları tespit etmek, bu aralıkları olabildiğince genişletmek, bir diğer ifadeyle kararlılık bölgeleri oluşturmak önemli bir araştırma problemidir.

Literatürde, genellikle, çözümlerin davranışına sistemlere bağlı matrislerinin spektral özellikleri yardımıyla karar verilmektedir. Bu matrisler, sistemler lineer ise katsayı matrisleri, sistemler lineer değilse denge noktalarındaki sistemin Jakobiyen matrisleri olarak karşımıza çıkmaktadır. Ancak simetrik olmayan matrislerin spektral analizinin kötü konulmuş bir kavram olduğu da iyi bilinen bir kavramdır.

Bu tez çalışmasında, simetrik olmayan matrislerin spektral analizinin kötü konulmuş olduğu da dikkate alınarak, lineer dinamik sistemler için tanıtılmış olan ω ve κ kararlılık parametreleri üzerine elde

edilmiş önemli sonuçlar yardımıyla bazı reel dünya problemlerinin kararlılık analizleri incelenmektedir. Bu analizler iki önemli amaca odaklanmaktadır. Bir ters problem olarak matematiksel modeller için,

- (1) denge noktalarının
- (2) parametrik giriş elemanlarının

kararlılık bölgelerinin bulunması ve bu bölgelerin uygun yordamlarla genişletilmesi amaçlanmaktadır. Bu amaçlar doğrultusunda bir tip iki ekonomili makroekonomik model, bir tip Lotka-Volterra modeli, bir tip salgın hastalık modelinin detaylı bir şekilde kararlılık analizleri yapılmıştır.

Anahtar Kelimeler: Dinamik sistemler, matematik modeller, denge noktaları, kararlılık-asimtotik kararlılık, lineerleştirme, Jakobiye matrisi, Schur kararlılık, Hurwitz kararlılık, kararlılık parametreleri, kararlılığın hassasiyeti.



ABSTRACT

MS THESIS

SENSITIVITY ANALYSIS AND PARAMETRIC EVALUATION OF THE BEHAVIOR OF SOLUTIONS IN DYNAMIC SYSTEMS

Reyyan Nur GÖKTAŞ

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
SELÇUK UNIVERSITY**

THE DEGREE OF MASTER OF SCIENCE IN MATHEMATICS

Advisor: Prof. Dr. Kemâl AYDIN

2025, 132 Pages

Jury

Prof. Dr. Kemâl AYDIN

Dr. Öğr. Üyesi Ali Osman ÇIBIKDİKEN

Dr. Öğr. Üyesi Güner TOPCU

The behavior of solutions in dynamical systems emerges as an important area of research in a wide range of disciplines, including engineering, biology, economics, and epidemiology. These systems, which describe time-varying states through mathematical models, may exhibit different behaviors depending on external influences, initial conditions, or changes in the system's input elements (input parameters). The long-term behavior of the solutions is directly related to the stability properties of the system.

The concepts of the motion, stability properties, and stability quality of solutions of linear dynamical systems are well known. Therefore, there are parameters called stability parameters (Schur stability and Hurwitz stability parameters, ω and κ , respectively) that provide information about the motion and stability quality of the system without calculating the solutions. Depending on the stability parameters, the stability properties of these systems and their resistance to external influences are determined by sensitivity analysis; for this, theorems generally known as continuity theorems are used in the literature.

The investigation of the conditions necessary for a problem to maintain its desired properties is also known as an inverse problem. As an inverse problem, addressing the stability properties of system behavior, determining stability intervals for parametric input elements, and expanding these intervals as much as possible—in other words, establishing stability regions—is an important research problem.

In the literature, the behavior of solutions is generally determined by the spectral properties of the matrices associated with the systems. These matrices appear as coefficient matrices if the systems are linear, and as Jacobian matrices of the system at the equilibrium points if the systems are nonlinear. However, it is well known that spectral analysis of non-symmetric matrices is an ill-posed concept.

In this thesis, considering the poorly constructed spectral analysis of non-symmetric matrices, stability analyses of some real-world problems are performed using results obtained on the stability parameters ω and κ . These analyses focus on two important objectives.

As an inverse problem, the aim is to find (1) equilibrium points and (2) stability regions of parametric input elements in mathematical models and to expand these regions using appropriate procedures. For these purposes, the stability analyses of a type of two-economy macroeconomic model, a type of Lotka-Volterra model, and an epidemic disease model have been studied in detail.

Keywords: Dynamical systems, mathematical models, equilibrium points, stability-asymptotic stability, linearization, Jacobian matrix, stability according to Lyapunov, Schur stability, Hurwitz stability, stability parameters, nonlinear dynamical systems, perturbed systems, sensitivity of stability.



ÖNSÖZ

Bu çalışma sürecinde, bilgi ve tecrübeleriyle yolumu aydınlatan, akademik gelişimime büyük katkı sağlayan, akademik rehberliğinin yanı sıra, her aşamada bir baba şefkatiyle yaklaşarak, sadece bir akademisyen olarak değil aynı zamanda büyük bir destek kaynağı olarak yanımda olan çalışmamın her aşamasında sabrıyla, anlayışıyla ve yönlendirmeleriyle yol gösteren saygıdeğer danışman hocam Prof. Dr. Kemâl AYDIN' a en içten teşekkürlerimi sunarım. Yüksek lisans sürecim boyunca bana rehberlik eden, bilgi ve tecrübeleriyle yolumu aydınlatan, bir abla gibi şefkatle ve içtenlikle desteğini hiçbir zaman esirgemeyen çok kıymetli hocam Dr. Öğr. Üyesi Güner Topcu'ya en içten teşekkürlerimi sunarım.

Bu çalışma süresince karşılaştığım her zorlukta yanımda olan, destekleriyle beni motive eden ve en değerli anları benimle paylaşan sevgili aileme sonsuz teşekkürlerimi sunarım. Hayatım boyunca bana her zaman inanan, cesaretlendiren ve sevgisiyle güç veren babam Bayram Göktaş'a, engin şefkati, sabrı ve fedakârlığıyla her zaman yolumu aydınlatan, bu süreçte de varlığıyla bana güven aşılayan annem Meral Göktaş'a ve enerjisi, neşesi ve desteğiyle zor anlarımda bile yüzümü güldüren, bana ilham olan kardeşim Ubeydullah Göktaş'a en içten duygularıyla teşekkür ederim. Bu süreç boyunca başarı ve motivasyonumu paylaşan, her anımda desteklerini esirgemeyen değerli dostlarıma da bu yolculuğu daha güzel bir hale getirdikleri için minnettarım.

Yüksek lisans eğitimim boyunca 2210-A Yurtiçi Yüksek Lisans Burs Programı kapsamında maddi destek veren TÜBİTAK'a teşekkürlerimi sunarım.

Selçuk Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü'ne 24201001 nolu proje kapsamında verilen kırtasiye desteği için teşekkür ederim.

Reyyan Nur GÖKTAŞ
KONYA-2025

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	vi
ÖNSÖZ	viii
İÇİNDEKİLER	ix
ŞEKİL LİSTESİ.....	xii
TABLO LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xii
SİMGELER VE KISALTMALAR	xiiv
1. GİRİŞ	1
1.1. Literatür Özeti.....	4
1.2. Tezin Yapısı.....	7
2. DİNAMİK SİSTEMLERİN ÇÖZÜMLERİNİN HAREKETİ.....	9
2.1. Denge Noktası.....	10
2.1.1. Geometrik yöntem (akış alanı yöntemi)	11
2.1.2. Faz düzlemi analizi	13
2.2. Kararlılık	14
2.2.1. Fark otonom sistemlerde kararlılık	14
2.2.2. Diferensiyel otonom sistemlerde kararlılık.....	16
2.3. Asimtotik Kararlılık	17
2.3.1. Schur kararlılık	17
2.3.2. Hurwitz kararlılık.....	18
2.4. Lineer Olmayan Sistemlerin Lineerleştirilmesi.....	19
2.5. Lineer Dinamik Sistemlerin Kararlılığı	21
2.5.1. Kesikli ve sürekli otonom sistemler için Spektral Kriter	22
2.5.2. Lyapunov matris denklem yardımıyla kararlılık ve kararlılığın kalitesi	22
2.5.2.1. Fark Lyapunov matrisi denklemi ile kararlılık ve Schur kararlılık parametresi (ω)	22
2.5.2.2. Dinamik sistemlerin pratik Schur kararlılığı (ω^*)	24
2.5.2.3. Lyapunov matris denklemi ile kararlılık ve Hurwitz kararlılık parametresi (κ)	25
2.5.2.4. Dinamik sistemlerin pratik Hurwitz kararlılığı (κ^*)	26

3. KARARLILIK ARALIĞININ GENİŞLETİLMESİ ÜZERİNE ALGORİTMALAR VE PYTHON YORDAMLARI 28

3.1. Matris Ailesinin Schur Kararlılığı ve Pratik Schur Kararlılığı	28
3.1.1. Genişletilmiş Schur kararlılık aralığının elde edilmesi için algoritma ve Python kodları.....	29
3.1.1.1. $\mathfrak{T}^{Se}$ aralığı için üst ve alt sınır algoritmaları ile python kodları	29
3.1.1.2. $\mathfrak{T}^{S*e}$ aralığı için üst ve alt sınır algoritmaları ile python kodları.....	33
3.2. Matris Ailesinin Hurwitz Kararlılığı ve Pratik Hurwitz Kararlılığı	36
3.2.1. Genişletilmiş Hurwitz kararlılık aralığının elde edilmesi için algoritma ve Python kodları.....	37
3.2.1.1. $\mathfrak{T}^{He}$ aralığı için üst ve alt sınır algoritmaları ile Python kodları.....	37
3.2.1.2. $\mathfrak{T}^{H*e}$ aralığı için üst ve alt sınır algoritmaları ile Python kodları	41

4. İKİ EKONOMİLİ MAKRO EKONOMİK MODELİN KARARLILIĞININ PARAMETRİK ANALİZİ..... 44

4.1. Makroekonomik Modelin Parametrik İfadelendirilmesi	46
4.2. Makroekonomik Modelin Schur Kararlılığı	48
4.2.1. Tek parametrelili makroekonomik model.....	48
4.2.1.1. Schur kararlı aralığın genişletilmesi	50
4.2.1.2. Genişletilmiş aralık için etkinlik analizi	53
4.2.1.2.1. Üst sınır için etkinlik analizi	54
4.2.1.2.2. Alt sınır için etkinlik analizi	56
4.2.2. İki parametrelili makroekonomik model.....	60
4.2.2.1. Önce α parametresine sonra β parametresine müdahale.....	65
4.2.2.2. Önce β parametresine sonra α parametresine müdahale.....	66
4.2.2.3. α ve β parametrelerine birlikte müdahale	67
4.3. Makroekonomik Modelin ω^* – Schur Kararlılığı.....	68
4.3.1. Tek parametrelili makroekonomik model.....	68
4.3.2. İki parametrelili makroekonomik model.....	71
4.3.2.1. Önce α parametresine sonra β parametresine müdahale.....	76
4.3.2.2. Önce β parametresine sonra α parametresine müdahale.....	77
4.3.2.3. α ve β parametrelerine birlikte müdahale	78
4.4. Değerlendirme	79

5. LOTKA VOLTERRA MATEMATİKSEL MODELİNİN KARARLILIĞININ PARAMETRİK ANALİZİ..... 82

5.1 (5.3) Lotka-Volterra Modelin Hurwitz Kararlılık Analizi.....	84
5.1.2. Genişletilmiş aralık için etkinlik analizi	85
5.1.2.1. Üst sınır için etkinlik analizi.....	86
5.1.2.2. Alt sınır için etkinlik analizi	87
5.2. (5.3) Lotka-Volterra Modelin κ^* – Hurwitz Kararlılık Analizi.....	89
5.3. Değerlendirme	91

6. SIR MATEMATİKSEL MODELİNİN KARARLILIĞININ PARAMETRİK ANALİZİ	92
6.1. SIR Modelin Schur Kararlılığı.....	93
6.1.1. (6.5) sisteminin s parametresine göre Schur kararlılığı.....	94
6.1.1.1. Genişletilmiş aralık için etkinlik analizi	96
6.1.1.1.1. Üst sınır için etkinlik analizi.....	96
6.1.1.1.2. Alt sınır için etkinlik analizi	98
6.1.2. (6.6) sisteminin ι parametresine göre Schur kararlılığı	100
6.1.2.1. Genişletilmiş aralık için etkinlik analizi	101
6.1.2.1.1. Üst sınır için etkinlik analizi.....	101
6.1.2.1.2. Alt sınır için etkinlik analizi	103
6.2. SIR Modelin ω^* – Schur Kararlılığı	105
6.2.1. (6.5) sisteminin s parametresine göre ω^* – Schur kararlılığı	105
6.2.2. (6.6) sisteminin ι parametresine göre ω^* – Schur kararlılığı.....	106
 7. SONUÇ VE DEĞERLENDİRME.....	109
 KAYNAKLAR	111
ÖZGEÇMİŞ	116

ŞEKİL LİSTESİ

Şekil		Sayfa
Şekil 2.1.	(2.4) denkleminin denge noktaları	10
Şekil 2.2.	(2.5) denkleminin geometrik yöntem gösterimi	12
Şekil 2.3.	(2.7) denkleminin denge noktaları	12
Şekil 2.4.	(2.8) denkleminin daz düzlemi analizi yöntemi ile gösterimi	14
Şekil 2.5.	x^* denge noktasının sağdan yarı kararlı olma durumu	15
Şekil 2.6.	x^* denge noktasının soldan yarı kararlı olma durumu	15
Şekil 2.7.	x^* denge noktasının kararlılık tipleri	16
Şekil 2.8.	x^* denge noktasının kararlılık tipleri	18
Şekil 2.9.	$\frac{\ g(y(t))\ }{\ y(t)\ }$ oranının, (2.15) lineer kısmın ve (2.14) lineer olmayan sistemin grafikleridir.	21
Şekil 3.1.	Teorem 3.2 ile elde edilen Schur kararlı aralık ve Yordam 3.1, Yordam 3.2 ile elde edilen genişletilmiş aralık	32
Şekil 3.2.	Schur kararlı aralık ve genişletilmiş Schur kararlı aralık, pratik Schur kararlı aralık ve genişletilmiş Schur kararlı aralık	36
Şekil 4.1.	$\alpha = 0.65$ parametresine göre farklı başlangıç noktaları için (a) x_n yörüngelerinin, (b) y_n yörüngelerinin (c) x_n, y_n yörüngelerinin birlikte hareketidir.	52
Şekil 4.2.	$\alpha = 0.95$ parametresine göre farklı başlangıç noktaları için yörüngelerin hareketi	53
Şekil 4.3.	(a) Tablo 4.3 etkinlik analizinin grafiği. (b), (a) grafiğinin $[1,4] \times [1,9000]$ dikdörtgeninde gösterimidir	54
Şekil 4.4.	Tablo 4.4' deki hesaplamalar sonucu (a) r_i 'nin i ' e göre grafiği, (b) ω_i 'nin i ' e göre grafiği, (c) ω_i 'in u_i ' e göre grafiği, (d) u_i 'in r_i ' e göre grafiği	55
Şekil 4.5.	Tablo 4.5 etkinlik analizinin grafiği.	56
Şekil 4.6.	Tablo 4.5' de hesaplanan değerlerin Schur kararlı Schur kararsız ve belirsiz bölgeleri gösteren grafikleri	58
Şekil 4.7.	Tablo 4.5' de hesaplanan değerlerin Schur kararlı, Schur kararsız ve belirsiz bölgeleri gösteren sayı doğrusu.	59
Şekil 4.8.	Tablo 4.6' deki hesaplamalar sonucu (a) r_i 'nin i ' ye göre grafiği, (b) ω_i 'in i ' ye göre grafiği, (c) ω_i 'in l_i ' e göre grafiği, (d) l_i 'in r_i ' e göre grafiği.	60
Şekil 4.9.	Farklı ω^* değerleri ile bulunan alt ve üst sınırları gösteren grafikler.	71

Şekil 4.10.	(4.6) sisteminin parametreleri Schur kararlılık, pratik Schur kararlılık için bulunan aralıklar ile oluşturulan bölgeler	81
Şekil 5.1.	Tablo 5.4' deki hesaplamalar sonucu (a) r_i ' nin i 'e göre grafiği, (b) κ_i ' nin i 'e göre grafiği, (c) κ_i ' in u_i ' e göre grafiği, (d) u_i 'in r_i ' e göre grafiği,	87
Şekil 5.2.	Tablo 5.6' daki hesaplamalar sonucu (a) r_i ' nin i ' ye göre grafiği, (b) κ_i ' nin i 'e göre grafiği, (c) κ_i ' in l_i ' e göre grafiği, (d) l_i 'in r_i ' e göre grafiği,	89
Şekil 6.1.	Tablo 6.3' deki hesaplamalar sonucu (a) r_i ' nin i ' ye göre grafiği, (b) ω_i ' nin i 'e göre grafiği, (c) ω_i ' in u_i ' e göre grafiği, (d) u_i 'in r_i ' e göre grafiği	97
Şekil 6.2.	Tablo 6.6' deki hesaplamalar sonucu (a) r_i ' nin i ' ye göre grafiği, (b) ω_i ' in i ' ye göre grafiği, (c) ω_i ' in l_i ' e göre grafiği, (d) l_i ' in r_i ' e göre grafiği	99
Şekil 6.3.	Tablo 6.10' daki hesaplamalar sonucu (a) r_i ' nin i 'e göre grafiği, (b) ω_i ' nin i 'e göre grafiği, (c) ω_i ' in u_i ' e göre grafiği, (d) u_i 'in r_i ' e göre grafiği	103
Şekil 6.4.	Tablo 6.12' deki hesaplamalar sonucu (a) r_i ' nin i ' ye göre grafiği, (b) ω_i ' in i ' ye göre grafiği, (c) ω_i ' in l_i ' e göre grafiği, (d) l_i 'in r_i ' e göre grafiği	104

TABLO LİSTESİ

Tablo		Sayfa
Tablo 2.1.	A_k matrislerinin $\omega(A_k)$ değerleri	23
Tablo 2.2.	A_k matrislerinin $\kappa(A_k)$ değerleri	26
Tablo 4.1.	Farklı r^* - pratiklik parametreleri ile durulan adım ve genişletilmiş üst sınır değerleri.	51
Tablo 4.2.	Farklı r^* - pratiklik parametreleri ile durulan adım ve genişletilmiş alt sınır değerleri.	51
Tablo 4.3.	$\omega(A(\alpha_i)) = \omega_i$, genişletilen üst sınır için etkinlik analizi.	54
Tablo 4.4.	Üst sınır genişletilirken r, u, ω parametrelerinin değişimi.	55
Tablo 4.5.	$\omega(A(\alpha_i)) = \omega_i$, genişletilen alt sınır için etkinlik analizi.	56
Tablo 4.6.	Alt sınır genişletilirken r, l, ω parametrelerinin değişimi.	59
Tablo 4.7.	α ve β elemanlarına müdahale edilerek bulunan Schur kararlılık aralıkları	68
Tablo 4.8.	Farklı r^*, ω^* - pratiklik parametreleri ile durulan adım ve genişletilmiş pratik üst sınır değerleri.	69
Tablo 4.9.	Farklı r^*, ω^* - pratiklik parametreleri ile durulan adım ve genişletilmiş pratik alt sınır değerleri.	69
Tablo 4.10.	α ve β elemanlarına müdahale edilerek bulunan Schur kararlılık aralıkları	79
Tablo 5.1.	(5.5) sistemi için farklı r^*, κ^* - pratiklik parametreleri ile durulan adım ve genişletilmiş üst sınır değerleri.	85
Tablo 5.2.	(5.5) sistemi için farklı r^*, κ^* - pratiklik parametreleri ile durulan adım ve genişletilmiş alt sınır değerleri.	85
Tablo 5.3.	$\kappa(J(\alpha_i)) = \kappa_i$, genişletilen üst sınır için etkinlik analizi.	86
Tablo 5.4.	$r^* = 0.001$ hassasiyetinde üst sınır genişletilirken r, u, κ parametrelerinin değişimi.	87
Tablo 5.5.	$\kappa(J(\alpha_i)) = \kappa_i$, genişletilen üst sınır için etkinlik analizi.	88
Tablo 5.6.	$r^* = 0.001$ hassasiyetinde alt sınır genişletilirken r, u, κ parametrelerinin değişimi.	88
Tablo 5.7.	Farklı r^*, κ^* - pratiklik parametreleri ile durulan adım ve genişletilmiş pratik üst sınır değerleri.	90
Tablo 5.8.	Farklı r^*, κ^* - pratiklik parametreleri ile durulan adım ve genişletilmiş pratik alt sınır değerleri.	90
Tablo 6.1.	(6.5) sistemi için farklı r^* - pratiklik parametreleri ile durulan adım ve genişletilmiş üst sınır değerleri.	95
Tablo 6.2.	(6.5) sistemi için Farklı r^* - pratiklik parametreleri ile durulan adım ve genişletilmiş alt sınır değerleri.	95

Tablo 6.3.	$\omega(J(s_i)) = \omega_i$, genişletilen üst sınır için etkinlik analizi.	96
Tablo 6.4.	$r^* = 0.001$ hassasiyetinde üst sınır genişletilirken r, u, ω parametrelerinin değişimi.	97
Tablo 6.5.	$\omega(J(s_i)) = \omega_i$, genişletilen alt sınır için etkinlik analizi.	98
Tablo 6.6.	$r^* = 0.001$ hassasiyetinde alt sınır genişletilirken r, l, ω parametrelerinin değişimi.	99
Tablo 6.7.	(6.6) sistemi için farklı r^* - pratiklik parametreleri ile durulan adım ve genişletilmiş üst sınır değerleri.	101
Tablo 6.8.	(6.6) sistemi için farklı r^* - pratiklik parametreleri ile durulan adım ve genişletilmiş alt sınır değerleri.	101
Tablo 6.9.	$\omega(J(l_i)) = \omega_i$, genişletilen üst sınır için etkinlik analizi.	102
Tablo 6.10.	$r^* = 0.001$ hassasiyetinde üst sınır genişletilirken r, u, ω parametrelerinin değişimi.	102
Tablo 6.11.	$\omega(J(l_i)) = \omega_i$, genişletilen alt sınır için etkinlik analizi.	103
Tablo 6.12.	$r^* = 0.001$ hassasiyetinde alt sınır genişletilirken r, l, ω parametrelerinin değişimi.	104
Tablo 6.13.	Farklı r^*, ω^* - pratiklik parametreleri ile durulan adım ve genişletilmiş pratik üst sınır değerleri.	105
Tablo 6.14.	Farklı r^*, ω^* - pratiklik parametreleri ile durulan adım ve genişletilmiş pratik alt sınır değerleri.	106
Tablo 6.15.	Farklı r^*, ω^* - pratiklik parametreleri ile bulunan ω^* – Schur genişletilmiş üst sınır değerleri.	107
Tablo 6.16.	Farklı r^*, ω^* - pratiklik parametreleri ile bulunan ω^* – Schur genişletilmiş üst sınır değerleri.	107

SİMGELER VE KISALTMALAR

Simgeler

$x^\Delta(t)$	: Diferensiyel denklemler ve fark denklemler için ortak gösterim
$\sigma(A)$	: A matrisinin spektrumu
$\lambda(A)$	: A matrisinin özdeğerleri
C_S	: birim disk
C_H	: sol yarı düzlem
$\omega(A)$	: A matrisinin Schur kararlılık parametresi
$\kappa(A)$	: A matrisinin Hurwitz kararlılık parametresi
ω^*	: pratik Schur kararlılık parametresi
κ^*	: pratik Hurwitz kararlılık parametresi
S_N	: Schur kararlı matris ailesi
H_N	: Hurwitz kararlı matris ailesi
S_N^*	: ω^* – Schur kararlı matris ailesi
H_N^*	: κ^* – Hurwitz kararlı matris ailesi
$\lesssim$	: $a \lesssim b$; a, b den küçük ama çok yakın

Kısaltmalar

$\mathcal{L}(A, B)$	: A ve B matrislerinin lineer toplamından oluşan matris ailesi
$\mathfrak{I}_L^S$	: Schur kararlılık aralığı
$\mathfrak{I}_L^{S^*}$	: ω^* – Schur kararlılık aralığı
$\mathfrak{I}_L^H$	: Hurwitz kararlılık aralığı
$\mathfrak{I}_L^{H^*}$	: κ^* – Hurwitz kararlılık aralığı
$\mathfrak{I}_L^{Se}$	: genişletilmiş Schur kararlılık aralığı
$\mathfrak{I}_L^{S^*e}$	: genişletilmiş ω^* – Schur kararlılık aralığı
$\mathfrak{I}_L^{He}$	: genişletilmiş Hurwitz kararlılık aralığı
$\mathfrak{I}_L^{H^*e}$	: genişletilmiş κ^* – Hurwitz kararlılık aralığı

1. GİRİŞ

Gök cisimlerinin hareketinden bir yapının mimarisine, aşı çalışmalarından nöronlar arasındaki etkileşimlere kadar neredeyse her fiziksel, teknik veya biyolojik sürecin modellenmesinde matematiğin önemli yeri vardır. Doğa olaylarında, mühendislikte, ekonomi ve teknoloji alanında yapılan çalışmalar, matematik modelleri sürekli sistemler şeklinde karşımıza getirirken bazen belirli zaman dilimleri söz konusu olduğundan problemler kesikli sistemler şeklinde de karşımıza gelmektedir (Tu, 1992; Agarwal, 2000; Chen, 2004; Allen, 2007; Bacaer, 2011; Enszer ve ark., 2015; Strogatz, 2015; Chern, 2016; Taghikhani ve ark., 2020; Neusser, 2021; Kumar ve Gunasundari, 2023; Li ve Chen, 2023).

Zamanla değişen durumları matematiksel modeller aracılığıyla tanımlayan bu sistemler, dış etkilere, başlangıç koşullarına veya sistem parametrelerinin değişimlerine bağlı olarak karmaşık davranışlar sergileyebilir. Ancak, bir sistemin uzun vadedeki hareketi ve güvenilirliği, kararlılık özelliğiyle doğrudan ilişkilidir. Kararlılık, sistemin denge noktasında kalabilme veya dış etkilere rağmen öngörülebilir davranışlar gösterme yeteneği olarak tanımlanır ve bu bağlamda sistem tasarımı, kontrol ve optimizasyon süreçlerinde kritik bir rol oynar (Tu, 1992; Agarwal, 2000; Chen, 2004; Allen, 2007; Bacaer, 2011; Enszer ve ark., 2015; Strogatz, 2015; Chern, 2016; Taghikhani ve ark., 2020; Neusser, 2021; Kumar ve Gunasundari, 2023; Li ve Chen, 2023).

Literatürde, lineer sistemlerin kararlılık analizi için geliştirilen yöntemler (ters problem yaklaşımları, bozunum yöntemleri (perturbation methods) ve süreklilik teoremleri), temel sonuçlar olarak süreklilik ve kararlılık bölgelerinin genişletilmesi gibi konuları ele almaktadır. Ancak, bu yöntemlerin ve birikimlerin, lineer olmayan dinamik sistemlerin, özellikle diferensiyel ve fark otonom sistemlerin kararlılık analizlerinde ne ölçüde kullanılabileceği temel bir problem olarak ortaya çıkmaktadır. Bu bağlamda, “Verilen bir sistem kararlılık özelliğine sahipken hangi şartlar altında bu özelliğini korur?” sorusu, sistem parametreleri ve başlangıç koşulları üzerine koşulların araştırılmasını gerektiren bir ters problem olarak karşımıza çıkmaktadır. Bu tez çalışması, belirtilen soruya yanıt ararken iki temel hedefi merkeze almaktadır:

(1) matematiksel modellerin denge noktaları için komşuluk bölgelerinin belirlenmesi,

(2) modellerin parametrik giriş elemanları için kararlılık aralıklarının tespit edilmesi.

Bu hedefler doğrultusunda, lineer olmayan dinamik sistemlerin kararlılık analizine yönelik özgün bir perspektif sunarak, hem teorik hem de pratik katkı sağlanması amaçlanmaktadır.

Yapılan bu çalışmada, öncelikle lineer bir model olan iki ekonomili makroekonomik model ele alınacaktır. Bu tür modeller, iki ekonomi arasındaki bağımlılıkların, denge durumlarının ve şoklara karşı tepki verme biçimlerinin incelenmesi amacıyla kullanılmaktadır. Örneğin, bir ülkenin ihracatının diğer ülkenin ithalatına nasıl etki ettiği veya bir ekonomideki yatırım artışının diğer ekonomiyi nasıl etkilediği analiz edilebilir (Halkos ve ark., 2012; Blanchard ve Johnson, 2013).

Halkos ve arkadaşlarının 2012'de yayımlanan çalışmasında modelin kararlılık analizleri yapılmıştır. Ancak tez çalışmasında modelin kararlılık analizi literatürdekilerin aksine Lyapunov matris denkleminin pozitif tanımlı çözümünü kullanan dinamik sistemlerin hem kararlılığını tespit etmeye hem de kararlılığın kalitesi hakkında değerlendirme yapmaya imkan sağlayan ω kararlılık parametresi kullanılarak yapılacaktır (Aydın, 1995; Akın ve Bulgak, 1998; Elaydi, 1999; Bulgak, 1999).

Bu parametre kullanılarak, sistemlerin kararlılığının dış etkilere karşı dayanıklılığı, parametrelerin hassasiyet analizi aracılığıyla belirlenebilmektedir. Hassasiyet analizi ise, süreklilik teoremleri yardımıyla gerçekleştirilmektedir (Akın ve Bulgak, 1998; Bulgakov ve Godunov, 1988; Bulgak, 1999; Duman ve Aydın, 2011, 2017). Literatürde bu süreklilik teoremlerini kullanarak oluşturulan algoritmalar yardımıyla kararlı olduğu bilinen bir aralık yine kararlılık özelliği korunacak şekilde genişletilebilmektedir (Topcu, 2023b, Topcu ve Aydın, 2023).

Bu birikimler kullanılarak makroekonomik modelin parametrik giriş elemanları için kararlılık aralıkları tespit edilecek ve bu aralıklar kararlı kalacak şekilde genişletilecektir.

Daha sonra lineer olmayan Lotka-Volterra modeli ele alınacaktır. Bu tür av-avcı modellerinin analizi, biyolojik sistemlerin dengelerini, türlerin bir arada var olma

koşullarını ve nüfusun zamanla nasıl değiştiğini anlamak açısından kritik bir öneme sahiptir. Özellikle, türlerin birlikte var olma veya rekabet dışı bırakma gibi ekolojik sonuçları öngörebilme yeteneği, bu modelin teorik ve uygulamalı ekoloji çalışmalarında önemli rol oynar.

Literatürde Lotka-Volterra modelinin kararlılık analizi farklı yöntemler ile çalışılmıştır (Strogatz, 2018).

Ancak tez çalışmasında modelin kararlılık analizi literatürdekilerin aksine Lyapunov matris denkleminin pozitif tanımlı çözümünü kullanan dinamik sistemlerin hem kararlılığını tespit etmeye hem de kararlılığın kalitesi hakkında değerlendirme yapmaya imkan sağlayan κ kararlılık parametresi kullanılarak yapılacaktır (Aydın, 1995; Akın ve Bulgak, 1998; Elaydi, 1999; Bulgak, 1999).

Schur kararlılığa benzer şekilde, Hurwitz kararlılık içinde hassasiyet analizi, süreklilik teoremleri yardımıyla gerçekleştirilmektedir (Akın ve Bulgak, 1998; Bulgakov ve Godunov, 1988; Bulgak, 1999; Duman ve Aydın, 2011, 2017). Yine literatürde bu süreklilik teoremlerini kullanarak oluşturulan algoritmalar yardımıyla kararlı olduğu bilinen bir aralık yine kararlılık özelliği korunacak şekilde genişletilebilmektedir (Topcu, 2023a, Topcu ve Aydın, 2023). Bu birikimler kullanılarak Lotka-Volterra modelinin parametrik giriş elemanları için kararlılık aralıkları tespit edilecek ve bu aralıklar kararlı kalacak şekilde genişletilecektir.

Burada bir noktaya dikkat edilmelidir. Birincisi, makroekonomik model lineer bir modeldi ve literatür bilgileri lineer sistemler için tasarlanmış bilgilerdir. Ancak Lotka-Volterra modeli lineer olmayan bir modeldir. Dolayısıyla öncelikle model lineerleştirilip daha sonra kararlılık analizi yapılacaktır. Yani lineer olmayan modelin kararlılık analizi lineer hale getirilen model üzerinden incelenecektir.

Son olarak, epidemolojik salgın hastalık modeli olarak bilinen sistem ele alınacaktır. Salgın hastalık modelleri hastalık dinamiklerini analiz incelemek ve insan sağlığını etkileyen salgın hastalıkların önlenmesinde ve ölümlerin azaltılmasında büyük rol oynamaktadır. İncelen model salgının büyüklüğüne veya etkisine göre salgının seyri hakkında doğru tedavi yöntemlerine sahip olmamızı sağlar. Enfekte (hastalık bulaşmış

birey) olmuş kişilerin sayısını, iyileşen kişilerin sayısını tahmin etmek bize hastalığın yayılma hızını ve etkisini dolayısıyla insan sağlığı için salgının sonucunu nasıl etkileyebileceğini gösterebilir.

Literatürde salgın hastalık modelinin kararlılık analizi farklı yöntemler ile çalışılmıştır (Wang, 2006; Mata ve Dourado, 2021; Chen ve ark., 2020). Tez çalışmasında modelin kararlılık analizi literatürdekilerin aksine Lyapunov matris denkleminin pozitif tanımlı çözümünü kullanan dinamik sistemlerin hem kararlılığını tespit etmeye hem de kararlılığın kalitesi hakkında değerlendirme yapmaya imkan sağlayan ω kararlılık parametresi kullanılarak yapılacaktır. 4. Bölüm ve 5. Bölümlere benzer şekilde literatürde mevcut olan süreklilik teoremlerini kullanarak oluşturulan algoritmalar yardımıyla kararlı olduğu bilinen bir aralık yine kararlılık özelliği korunacak şekilde genişletilebilmektedir bilgisi kullanılacaktır. Ancak 6. bölümde diğer bölümlerden farklı olarak;

Tezin iki temel hedefinden biri olan matematiksel modellerin denge noktaları için komşuluk bölgelerinin belirlenmesi üzerine çalışılacaktır.

1.1. Literatür Özeti

Dinamik sistemlerde karşılaşılan bazı matematiksel modellerin analitik çözümlerini bulmak bazen kolay olmayabilir, hatta mümkün de olmayabilir. Çoğu zaman bir modelin çözümünü elde etmek yerine çözümün hareketi hakkında bilgi sahibi olmak yeterli olabilmektedir. Örneğin, virüs hücrelerinin ilerleyen bir zamanda nasıl davrandığını önceden görebilmek ve analiz edebilmek hastalığın tedavisi ve erken teşhis için önemli bir rol oynamaktadır (Tu, 1992; Agarwal, 2000; Allen, 2007; Chern, 2016; Strogatz, 2018).

Matematik modellerin hareketi genellikle literatürde kararlılık analizi olarak bilinen analizlerle tespit edilebilmektedir. Bu noktada modellerin denge noktası kavramı öne çıkmaktadır.

$$x^\Delta(t) = \begin{cases} x'(t) = f(x(t)) & , t \in \mathbb{R} \\ x(t+1) = f(x(t)), & t \in \mathbb{Z} \end{cases} \quad (1.1)$$

otonom sistemini alalım. x^* noktası,

$$f(x^*) = \begin{cases} 0 & , t \in \mathbb{R} \\ x^* & , t \in \mathbb{Z} \end{cases} \quad (1.2)$$

ise x^* noktasına (1.1) otonom sisteminin denge noktası adı verilir (Elaydi, 2005; Bereketoğlu, 2012).

(1.1) otonom denklemi en az iki boyutlu olduğunda sistemlerin davranışı, denge noktasında oluşan Jakobiye matrisin spektral özellikleri yardımıyla da incelenebilmektedir. Bunun için öncelikle $f: X \rightarrow \mathbb{R}^k$, $X \subset \mathbb{R}^k$ fonksiyonu x^* – denge noktasında türevlenebilir fonksiyon olmak üzere,

$$J(x^*) = \left(\frac{\partial f(x^*)}{\partial x} \right) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x^*) & \cdots & \frac{\partial f_1}{\partial x_k}(x^*) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_k}{\partial x_1}(x^*) & \cdots & \frac{\partial f_k}{\partial x_k}(x^*) \end{pmatrix} \quad (1.3)$$

matrisine f' in x^* denge noktasındaki **Jakobiye matrisi** denilmektedir (Elaydi, 2005; Allen, 2007; Strogatz, 2015).

(1.1) otonom sistemi genel olarak lineer olmayan sistemlerdir. Ancak bu sistemler, yukarıda açıklanan Jakobiye matrisler yardımıyla lineerleştirilebilmektedir. Dolayısıyla lineer olan yeni sistemlerin kararlılık analizleri ile lineer olmayan otonom sistemlerin hareketi hakkında değerlendirmeler yapılabilmektedir.

$A \in M_N(\mathbb{R})$, $x - N$ boyutlu vektör olmak üzere, sırasıyla

$$x^\Delta(t) = Ax(t) \quad (1.4)$$

sistemi ele alalım. (1.4) sisteminin çözümlerinin hareketini genel olarak incelemek gerekirse; sırasıyla,

- $\sigma(A) = \{\lambda | \lambda = \lambda_i(A)\}$ – A matrisinin spektrumu
- $C_H = \{z | \operatorname{Re}(z) < 0\}$ – sol yarı düzlem
- $C_s = \{z | |z| < 1\}$ – birim disk

şeklinde olmak üzere,

- $\sigma(A) \subset C_H$ ise (1.4) sistemi Hurwitz kararlı (asimtotik kararlı),
- $\sigma(A) \subset C_s$ ise (1.4) sistemi Schur kararlı (fark asimtotik kararlı)

denir. Bu kriter literatürde Spektral kriter olarak bilinmektedir (Lyapunov, 1892; Akın ve Bulgak, 1998; Bulgak, 1999; Elaydi, 1999; Voicu ve Pastravanu, 2006).

Simetrik olmayan matrislerin öz değerlerinin hesaplanması problemi, bir diğer deyişle, spektral özelliklerin incelenmesi literatürde kötü konulmuş bir problem olarak bilinmektedir. Girdilerde yapılan küçük değişikliklerin, çıktılarda büyük değişikliklere yol açtığı problemlere kötü konulmuş problem denilmektedir (Wilkinson, 1965; Aydın, 1995; Akın ve Bulgak, 1998; Bulgak, 1999; Duman ve Aydın, 2017; Topcu ve Aydın, 2023-a; Topcu ve Aydın, 2023-b; Topcu, 2023).

Bu sebeple kararlılık analizi için spektral özellikler yerine Lyapunov matris denkleminin çözümüne ihtiyaç duyulmaktadır. (1.4) sisteminin Hurwitz kararlı olması için gerek ve yeter şart

$$A^*H + HA + I = 0; H = H^* > 0$$

Lyapunov matris denkleminin çözümünün olması iken (1.4) sisteminin Schur kararlı olması için gerek ve yeter şart ise

$$A^*HA - H + I = 0; H = H^* > 0$$

fark Lyapunov matris denkleminin çözümünün olmasıdır (Aydın, 1995; Akın ve Bulgak, 1998; Elaydi, 1999; Bulgak, 1999).

Bu yöntem spektral kritere göre avantajlar içermekle birlikte kararlılığın kalitesi hakkında fikir vermemektedir. Bu noktada literatürde bulunan, dinamik sistemlerin kararlılığını tespit eden ve kararlılığın kalitesini değerlendirmeye imkan sağlayan ω ve κ kararlılık parametrelerine başvurulmaktadır (bak. Bulgakov ve Godunov, 1988; Akın ve Bulgak, 1998; Bulgak, 1999). Ayrıca sistemlerin kararlılığının dış etkilere ne kadar dayanıklı olduğunun tespiti de parametrelerin hassasiyet analizi ile belirlenebilmektedir. Hassasiyet analizi sistemlerin bozunumu yöntemi ve süreklilik teoremleri yardımıyla yapılmaktadır (Bulgakov ve Godunov, 1988; Akın ve Bulgak, 1998; Bulgak, 1999; Duman ve Aydın, 2011, 2017).

$$H_N = \{A \in M_N(\mathbb{C}) | \kappa(A) < \infty\}$$

ve

$$S_N = \{A \in M_N(\mathbb{C}) | \omega(A) < \infty\}$$

olmak üzere bazı süreklilik teoremleri aşağıdaki gibidir;

- $A \in H_N$ ve $\frac{\|B\|}{\|A\|} \leq \frac{1}{15\kappa(A)}$ ise $A + B \in H_N$ (Bulgak, 1999),
- $A \in H_N$ ve $\|B\| < \frac{\|A\|}{\kappa(A)}$ ise $A + B \in H_N$ (Duman ve Aydın, 2017),

- $A \in S_N$ ve $\|B\| \leq \frac{1}{20\omega^2(A)}$ ise $A + B \in S_N$ (Bulgak, 1999),
- $A \in S_N$ ve $\|B\| < \sqrt{\|A\|^2 + \frac{1}{\omega(A)}} - \|A\|$ ise $A + B \in S_N$ (Aydın ve ark., 2002; Duman ve Aydın, 2011).

Ayrıca yukarıda verilen süreklilik teoremleri modifiye edilerek,

$$\mathcal{L}(A_1, A_2) = \{A(r) = A_1 + rA_2 | A_1, A_2 \in M_N(\mathbb{C})\}$$

lineer toplam aileleri Schur ve Hurwitz kararlı olacak şekilde r için aralıklar bulunmakta ve daha sonrasında bu aralıklar, kararlılık özellikleri korunacak şekilde, verilen yöntem ve algoritmalar yardımıyla genişletilmektedir (Topcu, 2023, Topcu ve Aydın, 2023).

Yukarıda, lineer sistemler için kısaca açıklanan, mevcut yöntem ve temel sonuçların (süreklilik teoremleri, kararlılık bölgesinin genişletilmesi vb.), bilgi ve birikimlerin lineer olmayan dinamik sistemlerin çözümlerinin hareketinde (diferensiyel ve fark otonom sistemlerin kararlılık analizlerinde) kullanılıp kullanılmayacağı ya da hangi ölçüde kullanılabileceği temel problem olarak ortaya çıkmaktadır.

Süreklilik teoremleri bir bakıma ters problemler olarak karşımıza çıkmaktadır. “Verilen sistem kararlılık özelliğine sahipken hangi şartlar altında yine de kararlı kalır?” sorusu problemin verileri üzerine şartlar araştırmayı zorunlu kılar ki, bu da bir ters problemidir.

Bu noktada projenin temel iki hedefi; matematiksel modellerin hareketinin incelenmesinde;

- 1) denge noktaları için komşuluk belirlemek,
 - 2) modelin parametrik giriş elemanları için aralıklar tespit etmek,
- olacaktır. Bu hedefler bakımından proje özgün bir çalışma niteliğine sahiptir.

1.2. Tezin Yapısı

Bu tez yedi bölümden oluşmaktadır:

- Birinci bölümde, problem tanıtılmış ve konuyla ilgili literatür taraması gerçekleştirilmiştir. Bu tarama sonucunda, dinamik sistemlerin kararlılığına ilişkin literatür bilgilerine yer verilmiştir.

- İkinci bölümde, kesikli sistemler ve sürekli sistemlerin denge noktası, kararlılık tanımları, lineerleştirme yaparken kullanılan yöntemlerden biri olan Jakobiye matris yöntemi, Schur kararlılık, ω^* – Schur kararlılık, Hurwitz kararlılık, κ^* – Hurwitz kararlılık ile ilgili temel bilgiler verilmiştir.
- Üçüncü bölümde, literatürde var olan lineer toplamlardan oluşan $\mathcal{L}(A, B)$ matris ailelerinin ve bu ailelerin Schur ve Hurwitz kararlılığını koruyan bozunum sınırlarını belirleyen aralıkların süreklilik teoremleri ile bulunması ve bu ailelerin kararlılık aralığının genişletilmesi için var olan algoritmalar verilmiştir. Verilen algoritmalara dayalı Python kodları yazılmıştır.
- Dördüncü bölümde, bir tür iki ekonomili makroekonomik model ele alınmış modelin parametrik olarak ifadelendirilmesi yapılmıştır. Devamında parametrik olan bu model üzerinden sistemin kararlılığı incelenmiş ve bu parametrik katsayılar için kararlılık özelliği korunacak şekilde kararlılık aralıkları tespit edilmiştir.
- Beşinci bölümde, bir tür Lotka-Volterra modeli ele alınmış modelin parametrik olarak ifadelendirilmesi yapılmıştır. Devamında parametrik olan bu model üzerinden sistemin kararlılığı incelenmiş ve bu parametrik katsayılar için kararlılık özelliği korunacak şekilde kararlılık aralıkları tespit edilmiştir.
- Altıncı bölümde, bir tür epidemolojik salgın hastalık modeli ele alınmış modelin denge noktaları yaklaşık olarak hesaplanmıştır. Devamında denge noktaları parametrik olarak ifade edilmiş ve denge noktaları üzerinden sistemin kararlılığı incelenmiş ve parametrik denge noktaları için kararlılık özelliği korunacak şekilde kararlılık aralıkları tespit edilmiştir.
- Yedinci bölümde, tezin genel değerlendirilmesi yapılmıştır.

2. DİNAMİK SİSTEMLERİN ÇÖZÜMLERİNİN HAREKETİ

Dinamik sistemler, zamanla değişen süreçleri matematiksel olarak modellemek ve analiz etmek için kullanılan güçlü bir araçtır. Durumların (örneğin, bir sistemin fiziksel, biyolojik veya ekonomik özelliklerinin) zaman içindeki evrimini tanımlayan denklemlerle çalışır. Fark denklemleri (kesikli zamanlı) ve diferensiyel denklemler (sürekli zamanlı) gibi yöntemlerle modellenen dinamik sistemler, teorik ve pratik birçok alanda kritik roller oynar. Dinamik sistemler, sistemlerin davranışını anlamak, tahmin etmek, tasarlamak ve kontrol etmek için kullanılır.

Dinamik sistemler, zamanla değişen süreçleri modelleme ve analiz etme yetenekleriyle çok çeşitli disiplinlerde kritik uygulamalara sahiptir. Fizikte gezegen hareketlerinden akışkanlar dinamiğine, mühendislikte kontrol sistemlerinden robotik ve elektrik devrelerine, biyolojide nüfus dinamiklerinden sinir ağlarına, ekonomide piyasa dengesi ve finansal modellere, meteorolojide hava durumu tahmininden iklim değişikliği analizine, tıpta salgın modellemesinden ilaç optimizasyonuna ve sosyal bilimlerde sosyal ağ dinamiklerinden kitle davranışlarına kadar geniş bir yelpazede kullanılır (Tu, 1992; Agarwal, 2000; Chen, 2004; Allen, 2007; Bacaer, 2011; Enszer ve ark., 2015; Strogatz, 2015; Chern, 2016; Taghikhani ve ark., 2020; Kumar ve Gunasundari, 2023; Li ve Chen, 2023).

$x^\Delta(t)$ operatörü

$$x^\Delta(t) = \begin{cases} x'(t) & , \quad t \in \mathbb{R} \\ x(t+1), & t \in \mathbb{Z} \end{cases}$$

şeklinde tanımlı olsun. Kesikli zamanlı sistemler fark denklemler, sürekli zamanlı sistemler diferensiyel denklemler ile modellenir.

$f: \mathbb{R}^N \rightarrow \mathbb{R}^N$, f türevlenebilir bir fonksiyon ve t bağımsız değişken olmak üzere, denklemler t' yi içinde açıkça barındırmıyorsa

$$x^\Delta(t) = f(x(t)) \quad (2.1)$$

otonom sistem, t' yi içinde açıkça barındırmıyorsa

$$x^\Delta(t) = f(t, x(t)) \quad (2.2)$$

sisteminde otonom olmayan sistem denir. Bu çalışmada otonom sistemler ele alınmıştır.

2.1. Denge Noktası

Dinamik sistemlerde karşılaşılan bazı matematiksel modellerin analitik çözümlerini bulmak bazen kolay olmayabilir, hatta mümkün de olmayabilir (bak, Karaoğlu, 2013; Anapalı, 2015). Çoğu zaman bir modelin çözümünü elde etmek yerine çözümün hareketi hakkında bilgi sahibi olmak yeterli olabilmektedir. Matematik modellerin hareketi genellikle literatürde kararlılık analizi olarak bilinen analizlerle tespit edilebilmektedir. Bu noktada modellerin denge noktası kavramı öne çıkmaktadır.

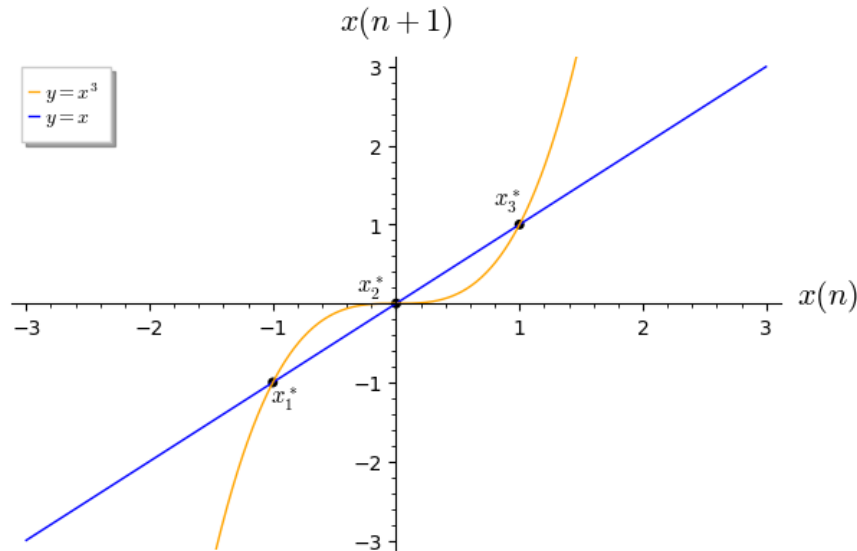
Tanım 2.1. (Otonom fark sistemlerde denge noktası):

$$x(n+1) = f(x(n)), n \in \mathbb{Z} \quad (2.3)$$

otonom fark denklemini ele alalım. f' in tanım kümesinde ki x^* noktası, f' in sabit bir noktasıysa, yani $f(x^*) = x^*$ ise *denge noktasıdır* denir (Elaydi, 2005; Bereketoğlu, 2012). Grafiksel olarak bir denge noktası, f grafiğinin $y=x$ köşegen doğrusuyla kesiştiği noktanın x koordinatıdır. Örneğin,

$$x(n+1) = x^3(n) \quad (2.4)$$

fark denkleminin $x_1^* = -1$, $x_2^* = 0$, $x_3^* = 1$ şeklinde üç denge noktası vardır. Bulunan bu denge noktaları grafiksel olarak Şekil 2.1 de gösterilmiştir.



Şekil 2.1. (2.4) denkleminin denge noktaları.

2.1.1. Geometrik yöntem (akış alanı yöntemi)

Akış alanı yöntemi veya Verhulst diyagramı fonksiyonların davranışını incelemek için dinamik sistemlerde kullanılan bir grafik metodudur (Elaydi, 2005; Bereketoğlu, 2012; Neusser, 2021; Emre, 2022). (2.3) fark denkleminin denge noktasının kararlılık durumunu analiz ederken kullanılır. (2.3) denkleminin bir $x(0) = x_0$ başlangıç değerine karşılık gelen çözümü ardışık hesaplamalar sonucu

$$(x_0, x_1, x_2, \dots, x_n, \dots) = (x_0, f(x_0), f^2(x_0), \dots, f^n(x_0), \dots)$$

şeklinde bulunur. Bu çözümün grafiksel olarak hesaplanması Geometrik yöntem (Akış Alanı Yöntemi) ortaya çıkarır. Bu yöntem aşağıdaki adımlar izlenerek uygulanır.

1. Adım: $\Omega = \{(x(n), x(n+1)) | x(n+1) = f(x(n)), n \in \mathbb{Z}\}$

kümesini ele alalım. Bu küme kullanılarak, f' in grafiği ve $x(n+1) = x(n)$ (yani $y = x$) doğrusu çizilir.

2. Adım: $x(0)$ başlangıç değeri $x(n)$ üzerinde işaretlenir ($x(0)$ noktası, fonksiyonun $x(n+1) = x(n)$ doğrusu ile kesiştiği noktanın civarında alınır) ve bu noktadan $f(x(n))$ deki görüntüsü, yani $(x(0), f(x(0)))$ bulunur.

3. Adım: $(x(0), f(x(0)))$ noktasından $x(n+1) = x(n)$ doğrusuna yatay çizgi çizerek $(f(x(0)), f(x(0)))$ noktası bulunur. Bu noktanın $x(n)$ ekseninde dik izdüşümü alınarak $x_1(0)$ elde edilir.

4. Adım: Aynı şekilde devam ederek $x_i(0)$, $i = 2, 3, \dots, n$ değerleri ile onların $x(n)$ üzerindeki konumları için önceki adımlar tekrarlanır.

Adımlar tekrarlanırken akış denge noktasına gidiyorsa o noktaya asimtotik kararlı; akış o noktadan uzaklaşıyorsa kararsızdır denir.

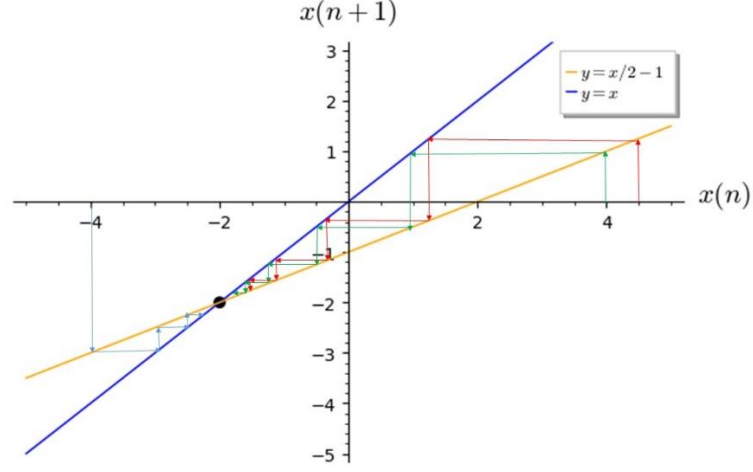
Örnek 2.1.

$$x(n+1) = \frac{1}{2}x(n) - 1, \quad n = 0, 1, \dots \quad (2.5)$$

fark Cauchy Problemini ele alalım. Denge noktası,

$$f(x^*) = x^* \Rightarrow \frac{1}{2}x^* - 1 = x^* \Rightarrow x^* = -2$$

bulunur. Geometrik yöntem ile kararlılığı incelenirse Şekil 2.2' de görüldüğü üzere verilen problem kararlıdır.



Şekil 2.2. (2.5) denkleminin Geometrik yöntem gösterimi.

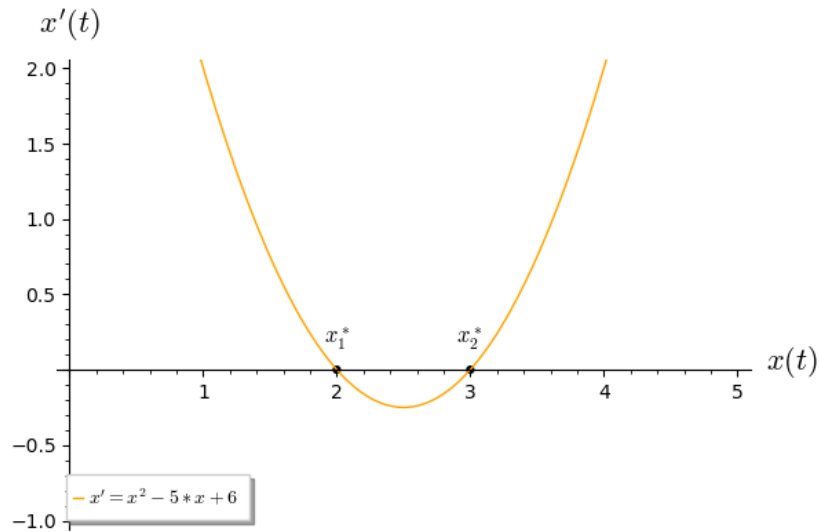
Tanım 2.2. (Otonom diferensiyel sistemlerde denge noktası):

$$x'(t) = f(x(t)), t \in \mathbb{R} \quad (2.6)$$

otonom diferensiyel denklemini ele alalım. f ' in tanım kümesinde ki x^* noktası, f ' in sabit bir noktasıysa, yani $f(x^*) = 0$ ise denge noktasıdır (Tu, 1992; Scheinerman, 2012). Grafiksel olarak bir denge noktası, f grafiğinin x - eksenini kestiği noktalardır. Örneğin,

$$x'(t) = x^2(t) - 5x(t) + 6 \quad (2.7)$$

diferensiyel denkleminin $x_1^* = 2$, $x_2^* = 3$ şeklinde iki denge noktası vardır. Bulunan bu denge noktaları grafiksel olarak Şekil 2.3 de gösterilmiştir.



Şekil 2.3. (2.7) denleminin denge noktaları.

2.1.2. Faz düzlemi analizi

Faz düzlemi analizi, dinamik sistemlerin kararlılık analizinde kullanılan yöntemdir ve lineer olmayan sistemlerin uzun vadeli davranışlarını görselleştirerek anlamayı sağlar. Bu yöntem, sistemin durum uzayındaki yörüngelerini incelemek için faz portreleri veya faz diyagramları kullanır. Faz analizi, sistemin denge noktalarını, yörüngelerin bu noktalara yakınsama veya uzaklaşma eğilimlerini anlamada kullanılır (Tu, 1992; Scheinerman, 2012).

(2.5) otonom diferensiyel denkleminin davranışı (x, x') grafiği çizilerek anlaşılır. (x, x') grafiğini çizerek çözüm eğrilerinin şeklini tahmin etme işlemine faz düzlemi analizi denir. Faz düzlemi analizi için aşağıdaki adımlar izlenerek uygulanır.

1. (x, x') grafiğini çiz (burada x ekseninin kestiği noktalar x^* lardır.)
2. $x' > 0$ da x ekseninde vektörlerin ok yönü sağa doğru çiz.
 $x' < 0$ da x ekseninde vektörlerin ok yönü sola doğru çiz.
3. Bu adımlar sonrasında eğer;
 - ✓ x^* noktasının solunda $x' < 0$, sağında $x' > 0$ ise x^* a kararsız denge noktası,
 - ✓ x^* noktasının solunda $x' > 0$, sağında $x' < 0$ ise x^* a kararlı denge noktası denir.

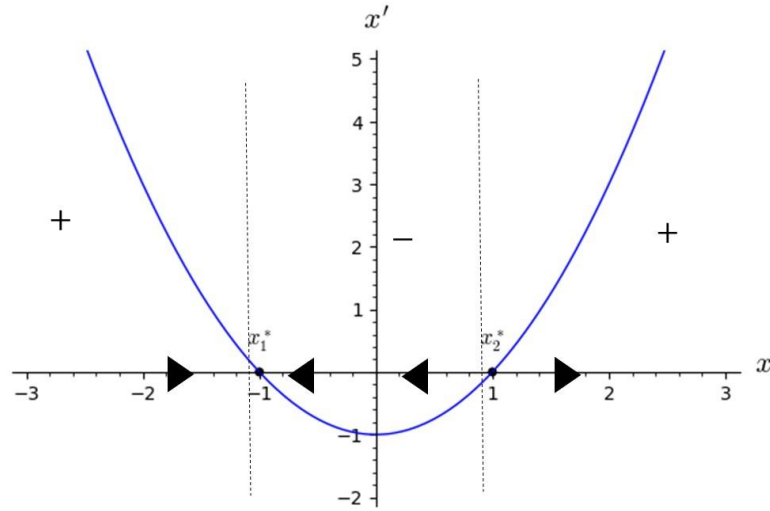
Diğer taraftan (t, x) grafiğini (x, x') grafiğinden yararlanarak farklı $x(t_0)$ başlangıç noktaları için yörüngelerin hareketini tahmin edebiliriz. Örneğin,

$$x' = x^2 - 1, x(t_0) = x_0 \quad (2.8)$$

diferensiyel Cauchy problemini ele alalım. Denge noktası,

$$f(x) = x^2 - 1 = 0 \Rightarrow x_{1,2}^* = \mp 1$$

şeklinde bulunur. Faz düzlemi yöntemi ile kararlılığı incelenirse Şekil 2.4' ten $x_1^* = -1$ denge noktasının kararlı, $x_2^* = 1$ denge noktasının kararsız olduğu görülür.



Şekil 2.4. (2.8) denkleminin faz analizi yöntemi ile gösterimi.

2.2. Kararlılık

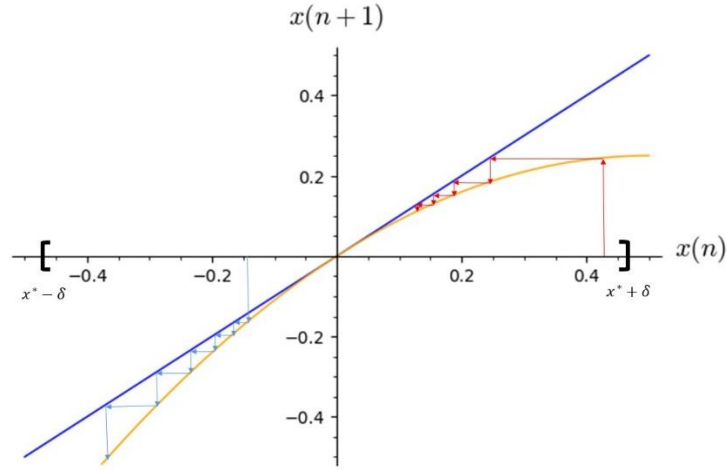
Kararlılık, dinamik sistemlerin (örneğin, fark denklemlerle veya diferensiyel denklemlerle tanımlı sistemlerin) denge noktaları civarındaki davranışlarını incelemek için kullanılan matematiksel bir yöntemdir. Bununla ilgili bazı tanımları aşağıda verelim (Elaydi, 2005; Bertotti ve ark., 2009; Bereketoğlu, 2012; Neusser, 2021).

2.2.1. Fark otonom sistemlerde kararlılık

$$\begin{aligned} x(n+1) &= f(x(n)), n_0 \leq n, n_0, n \in \mathbb{Z} \\ x(n_0) &= x_0 \end{aligned} \quad (2.9)$$

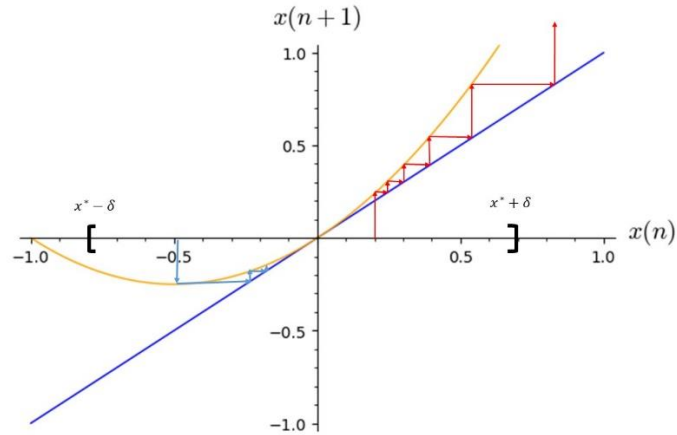
x^* , (2.9) Cauchy probleminin denge noktası olsun.

Tanım 2.3. (Sağdan yarı kararlılık): Her $\varepsilon > 0$ ' a karşılık (2.9) denkleminin $x(n) = x(n, x_0)$ çözümü için başlangıç noktaları denge noktasının sağından alınmak üzere ($x_0 > x^*$) $x_0 - x^* < \delta$ olduğunda her $n_0 < n$ için $|x(n) - x^*| < \varepsilon$ olacak şekilde $\delta = \delta(\varepsilon)$ bulunuyorsa x^* denge noktası sağdan yarı kararlıdır (Şekil 2.5).



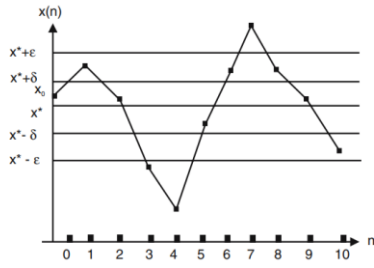
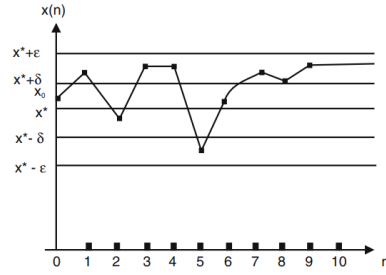
Şekil 2.5 x^* denge noktasının sağdan yarı kararlı olma durumu.

Tanım 2.4. (Soldan yarı kararlılık): Her $\varepsilon > 0$ ' a karşılık (2.9) denkleminin $x(n) = x(n, x_0)$ çözümü için başlangıç noktaları denge noktasının solundan alınmak üzere ($x_0 < x^*$) $x^* - x_0 < \delta$ olduğunda her $n_0 < n$ için $|x(n) - x^*| < \varepsilon$ olacak şekilde $\delta = \delta(\varepsilon)$ bulunuyorsa x^* denge noktası sağdan yarı kararlıdır (Şekil 2.6).



Şekil 2.6 x^* denge noktasının soldan yarı kararlı olma durumu.

Tanım 2.5. (Kararlılık): Her $\varepsilon > 0$ ' a karşılık (2.9) denkleminin $x(n) = x(n, x_0)$ çözümü için başlangıç noktaları denge noktasının komşuluğundan alınmak üzere $x_0 \in U_\delta(x^*)$ olduğunda her $n_0 < n$ için $x(n) \in U_\varepsilon(x^*)$ olacak şekilde $\delta = \delta(\varepsilon)$ bulunuyorsa x^* denge noktası kararlıdır (Şekil 2.7).

(a) x^* noktası kararsız.(b) x^* noktası kararlı.Şekil 2.7. x^* noktasının kararlılık tiplerinin görselleştirilmesi (Elaydi, 2005).

2.2.2. Diferensiyel otonom sistemlerde kararlılık

$$\begin{aligned} x'(t) &= f(x(t)), t_0 \leq t, t_0, t \in \mathbb{R} \\ x(t_0) &= x_0 \end{aligned} \quad (2.10)$$

x^* , (2.10) Cauchy probleminin denge noktası olsun.

Tanım 2.6. (Sağdan yarı kararlılık): Her $\epsilon > 0$ ' a karşılık (2.10) denkleminin $x(t) = x(t, x_0)$ çözümü için başlangıç noktaları denge noktasının sağından alınmak üzere ($x_0 > x^*$) $x_0 - x^* < \delta$ olduğunda her $t_0 < t$ için $|x(t) - x^*| < \epsilon$ olacak şekilde $\delta = \delta(\epsilon)$ bulunuyorsa x^* denge noktası sağdan yarı kararlıdır.

Tanım 2.7. (Soldan yarı kararlılık): Her $\epsilon > 0$ ' a karşılık (2.10) denkleminin $x(t) = x(t, x_0)$ çözümü için başlangıç noktaları denge noktasının solundan alınmak üzere ($x_0 < x^*$) $x^* - x_0 < \delta$ olduğunda her $t_0 < t$ için $|x(t) - x^*| < \epsilon$ olacak şekilde $\delta = \delta(\epsilon)$ bulunuyorsa x^* denge noktası soldan yarı kararlıdır.

Tanım 2.5. (Kararlılık): Her $\epsilon > 0$ ' a karşılık (2.10) denkleminin $x(t) = x(t, x_0)$ çözümü için başlangıç noktaları denge noktasının komşuluğundan alınmak üzere $x_0 \in U_\delta(x^*)$ olduğunda her $t_0 < t$ için $x(t) \in U_\epsilon(x^*)$ olacak şekilde $\delta = \delta(\epsilon)$ bulunuyorsa x^* denge noktası kararlıdır.

2.3. Asimtotik Kararlılık

Asimtotik kararlılık, dinamik sistemlerin denge noktalarının uzun vadeli davranışlarını analiz etmede temel bir kavramdır. Bir denge noktasının asimtotik kararlı olması, sistemin bu noktanın etrafındaki bir başlangıç noktasından zamanla istenilen noktaya yaklaşmasını ifade eder. Aşağıda bununla ilgili bazı tanımlar verilecektir (Elaydi, 2005 ; Bertotti ve ark., 2009; Bereketoğlu, 2012; Morey, 2020; Neusser, 2021).

2.3.1. Schur kararlılık

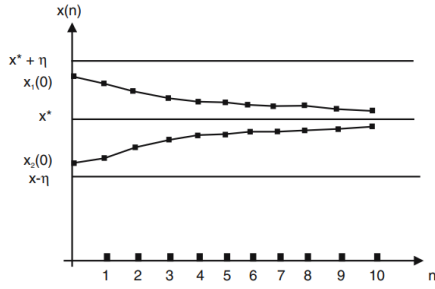
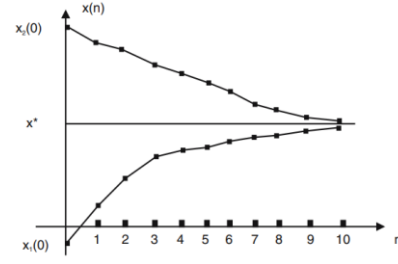
Burada fark dinamik sistemlerin asimtotik kararlılığı ile ilgili tanımlar verilecektir.

Tanım 2.6. (Yarı Çekicilik): (2.9) Cauchy probleminin $x(n)$ çözümü sağdan (veya soldan) kararlı ve $x_0 - x^* < \delta$ ($x^* - x_0 < \delta$) için $\lim_{n \rightarrow \infty} x(n) = x^*$ olacak şekilde $\delta > 0$ varsa x^* sağdan yarı (ya da soldan yarı) çekicidir.

Tanım 2.7. (Çekici ve global çekicilik): (2.9) Cauchy probleminin $x(n)$ çözümü kararlı ve $x_0 \in U_\delta(x^*)$ için $\lim_{n \rightarrow \infty} x(n) = x^*$ olacak şekilde $\delta > 0$ varsa x^* a çekicidir denir. Eğer $\delta = \infty$ için çözümler x^* a yaklaşıyorsa x^* global çekicidir.

Tanım 2.8. (Asimtotik kararlılık): x^* denge noktası sağdan yarı kararlı (ya da soldan yarı kararlı) ve sağdan çekici (ya da soldan çekici) ise yarı asimtotik kararlıdır. x^* denge noktası hem kararlı hem çekici ise asimtotik kararlıdır. Bu denge noktasının Lyapunov' a göre fark asimtotik kararlılığı olarak ifade edilmektedir. (2.9) için Lyapunov' a göre fark asimtotik kararlılık Schur kararlılık olarak adlandırılmaktadır.

Tanım 2.9. (Global asimtotik kararlılık): x^* denge noktası hem kararlı hem global çekici ise global asimtotik kararlıdır.

a) x^* noktası asimtotik kararlı.b) x^* noktası global asimtotik kararlı.Şekil 2.8 x^* noktasının kararlılık tiplerinin görselleştirilmesi (Elaydi, 2005).

2.3.2. Hurwitz kararlılık

Diferensiyel dinamik sistemlerin asimtotik kararlılığı ile ilgili tanımlar aşağıdaki gibidir (Elaydi, 2005 ; Bertotti ve ark., 2009; Bereketoğlu, 2012; Neusser, 2021).

Tanım 2.10. (Yarı Çekicilik): (2.10) Cauchy probleminin $x(t)$ çözümü sağdan (veya soldan) kararlı ve $x_0 - x^* < \delta$ ($x^* - x_0 < \delta$) için $\lim_{t \rightarrow \infty} x(t) = x^*$ olacak şekilde $\delta > 0$ varsa x^* a sağdan yarı (ya da soldan yarı) çekicidir.

Tanım 2.11. (Çekici ve global çekicilik): (2.10) Cauchy probleminin $x(t)$ çözümü kararlı ve $x_0 \in \cup_{\delta} (x^*)$ için $\lim_{t \rightarrow \infty} x(t) = x^*$ olacak şekilde $\delta > 0$ varsa x^* çekicidir. Diğer taraftan $\delta = \infty$ için çözümler x^* a yaklaşıyorsa x^* global çekicidir.

Tanım 2.12. (Asimtotik kararlılık): x^* denge noktası sağdan yarı kararlı (ya da soldan yarı kararlı) ve sağdan çekici (ya da soldan çekici) ise yarı asimtotik kararlıdır. x^* denge noktası hem kararlı hem çekici ise asimtotik kararlıdır. Bu durum denge noktasının Lyapunov' a göre asimtotik kararlılığı olarak ifade edilmektedir. (2.10) denklemini için Lyapunov' a göre asimtotik kararlılık Hurwitz kararlılık olarak adlandırılmaktadır.

Tanım 2.13. (Global asimtotik kararlılık): x^* denge noktası hem kararlı hem global çekici ise global asimtotik kararlıdır.

Uyarı: Bu bölümden itibaren sonra özellikle lineer sistemler için Schur ve Hurwitz kararlılık kavramları kullanılacaktır.

2.4. Lineer Olmayan Sistemlerin Lineerleştirilmesi

Lineer olmayan sistemlerin lineerleştirilmesi, lineer olmayan dinamik sistemlerin denge noktaları civarındaki davranışlarını yaklaşık olarak analiz etmek için kullanılan bir yöntemdir. Lineer olmayan sistemi lineer hale getirmek için farklı yöntemler mevcuttur. Bu çalışmada bu yöntemlerden bir olan Jakobiye matris yardımıyla Lineerleştirme kullanılacaktır. Bu yöntem, lineer olmayan bir sistemin denge noktası etrafında Taylor serisi açılımı kullanılarak lineer bir sistem elde edilmesini sağlar. Öncelikle Jakobiye matris ile lineerleştirme yöntemini verelim (Horowitz, 2002; Elaydi, 2005; Allen, 2007).

$f: K \rightarrow \mathbb{R}^N, K \subset \mathbb{R}^N, x^* = (x_1^*, x_2^*, \dots, x_k^*)^T$ denge noktasında türevlenebilir fonksiyon olmak üzere,

$$x^\Delta(t) = \begin{cases} x'(t) = f(x(t)) & , t \in \mathbb{R} \\ x(t+1) = f(x(t)), & t \in \mathbb{Z} \end{cases} \quad (2.11)$$

lineer olmayan otonom dinamik sistemi alınsın. (2.11) sisteminin açık hali

$$\begin{pmatrix} x_1^\Delta(t) \\ \vdots \\ x_N^\Delta(t) \end{pmatrix} = \begin{pmatrix} f_1(x_1(t), x_2(t), \dots, x_N(t)) \\ \vdots \\ f_N(x_1(t), x_2(t), \dots, x_N(t)) \end{pmatrix}$$

şeklindedir. $g: K \rightarrow \mathbb{R}^N, K \subset \mathbb{R}^N$ ve

$$y^\Delta(t) = \begin{cases} y'(t) = f(y(t)) & , t \in \mathbb{R} \\ y(t+1) = f(y(t)), & t \in \mathbb{Z} \end{cases}$$

olmak üzere (2.11) sistemi Taylor formülü yardımıyla

$$y^\Delta(t) = Jy(t) + g(y(t)) \quad (2.12)$$

şeklinde yazılabilir. Burada $g(y(t)) = o(y(t))$ ve $J = J(x^*) = \frac{\partial f(x^*)}{\partial x}$ dir. Burada, $y^\Delta(t) = Jy(t)$ sistemi (2.12) sisteminin lineer kısmı olup $g(y(t))$ terimi lineerliği bozan kısımdır. Dolayısıyla $\|y(t)\| \rightarrow 0$ iken $g(y(t)) = o(y(t))$ oluyorsa sistemin kararlılığına J matrisi üzerinden karar verilebilir. Burada $o(y(t))$ sonsuz küçük fonksiyondur ve $g(y(t)) = o(y(t))$ ifadesi $\lim_{t \rightarrow \infty} \frac{\|g(y(t))\|}{\|y(t)\|} = 0$ ile aynı anlama gelmektedir.

Dolayısıyla lineer olmayan sistemin kararlılığına lineerleştirilmiş sistem üzerinden karar verilebilir. Yani lineer sistemin kararlı olması lineer olmayan sistemin kararlı olduğu anlamına gelebilir. Verilen J matrisi f fonksiyonunun Jakobiyen matrisi olmak üzere

$$J(x^*) = \frac{\partial f(x)}{\partial x} \Big|_{x=x^*} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x^*) & \dots & \frac{\partial f_1}{\partial x_k}(x^*) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_k}{\partial x_1}(x^*) & \dots & \frac{\partial f_k}{\partial x_k}(x^*) \end{pmatrix} \quad (2.13)$$

şeklinde gösterilir. Bir başka deyişle (2.13) matrisine f' in x^* denge noktasındaki **Jakobiyen matrisi** denir. Örneğin;

$$\begin{cases} x(t+1) = \frac{ay(t)}{1+x^2(t)} \\ x(t+1) = \frac{bx(t)}{1+y^2(t)} \end{cases} \quad (2.14)$$

lineer olmayan denklem sistemi için Jakobiyen matris,

$$J(x, y) = \begin{pmatrix} \frac{\partial f_1}{\partial x}(x, y) & \frac{\partial f_1}{\partial y}(x, y) \\ \frac{\partial f_2}{\partial x}(x, y) & \frac{\partial f_2}{\partial y}(x, y) \end{pmatrix} = \begin{pmatrix} -\frac{2x}{(1+x^2)^2} & \frac{a}{1+x^2} \\ \frac{b}{1+y^2} & -\frac{2y}{(1+y^2)^2} \end{pmatrix}$$

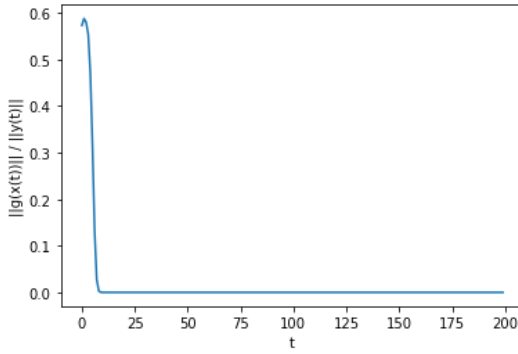
şeklinde dir. Ve $(x^*, y^*) = (0,0)$ denge noktası için,

$$J(0,0) = \begin{pmatrix} 0 & a \\ b & 0 \end{pmatrix}$$

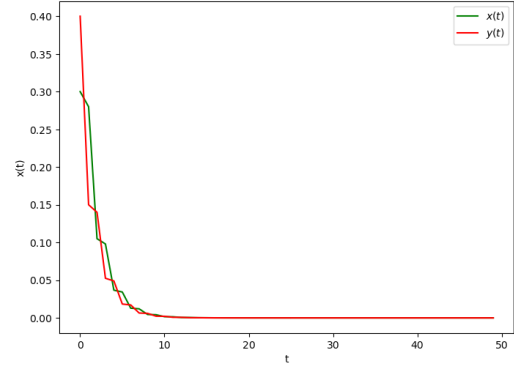
dır. Dolayısıyla sistem (2.12)' de kine benzer şekilde

$$\begin{pmatrix} x(t+1) \\ y(t+1) \end{pmatrix} = \begin{pmatrix} 0 & a \\ b & 0 \end{pmatrix} \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} + \begin{pmatrix} -\frac{ay(t)x^2(t)}{1+x^2(t)} \\ -\frac{bx(t)y^2(t)}{1+y^2(t)} \end{pmatrix} \quad (2.15)$$

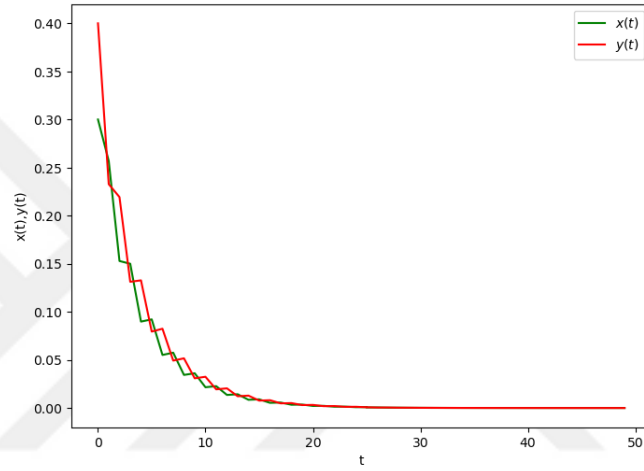
dır. $g(x(t))$ normunun $x(t)$ normuna oranının limiti $t \rightarrow \infty$ iken 0' a gitmektedir (Şekil 2.9 (a)). Bu ise bize Jakobiyen matris üzerinden karar verme imkanı sunmaktadır. Şekil 2.9 (b)' de lineer hale getirilen sistemin çözümlerinin grafiği, Şekil 2.9 (c)' de lineer olmayan sistemin çözümlerinin grafiği verilmiştir. Görüldüğü üzere iki şekilde de denge noktasına çekilme vardır. Lineer sistem asimtotik kararlı olduğundan lineer olmayan sistemde asimtotik kararlıdır.



(a) $\frac{\|g(y(t))\|}{\|y(t)\|}$ oranının grafiği



(b) (2.15) sisteminin çözümlerinin grafiği



(c) (2.14) sisteminin çözümlerinin grafiği

Şekil 2.9 $\frac{\|g(y(t))\|}{\|y(t)\|}$ oranının, (2.15) lineer kısmın ve (2.14) lineer olmayan sisteminin grafikleridir.

2.5. Lineer Dinamik Sistemlerin Kararlılığı

Bu kısımda lineer sistemlerin asimtotik kararlılığı için literatürde bulunan spektral kriter, Lyapunov matris denklemi ve bu denklemin çözümüne bağlı kararlılığın kalitesini gösteren parametreler verilecektir.

$$x^\Delta(t) = \begin{cases} x'(t) = Ax(t) & , t \in \mathbb{R}, A \in M_N(\mathbb{R}) \\ x(t+1) = Ax(t) & , t \in \mathbb{Z}, A \in M_N(\mathbb{R}) \end{cases} \quad (2.16)$$

sistemi problemlerde direk lineer olarak gelebileceği gibi Jakobiye matris ile lineerleştirilmiş sistemlerin katsayı matrisi olarakta gelebilmektedir. Aşağıda lineer fark sistemler ve diferensiyel sistemler için asimtotik kararlılık yöntemleri verilmiştir.

2.5.1. Kesikli ve sürekli otonom sistemler için spektral kriter

Bu kısımda Spektral kriteri tanıtmadan önce konuyla ilgili birkaç kümeyi aşağıdaki gibi verelim (Lyapunov, 1892; Akın ve Bulgak, 1998; Bulgak, 1999; Elaydi, 1999; Voicu ve Pastravanu, 2006).

- $\sigma(A) = \{\lambda | \lambda = \lambda_i(A)\}$ kümesi spektrum olarak adlandırılır ve A matrisinin öz değerlerinin kümesini temsil eder.
- $C_s = \{z \in \mathbb{C} | |z| < 1\}$ kümesi birim disk olarak adlandırılır.
- $C_H = \{z \in \mathbb{C} | \text{Re}(z) < 0\}$ kümesi sol yarım düzlem olarak adlandırılır.

Bu tanımlarla birlikte eğer;

- ✓ $\sigma(A) \subset C_s \Leftrightarrow A$ matrisine (2.16 kesikli sistemine) Schur kararlıdır denir.
- ✓ $\sigma(A) \subset C_H \Leftrightarrow A$ matrisine (2.16 sürekli sistemine) Hurwitz kararlıdır denir.

Bu durum literatürde spektral kriter olarak bilinmektedir.

- $S_N = \{A \in M_N(\mathbb{C}) | |\lambda_i(A)| < 1 (i = 1, 2, \dots, N)\}$
- $H_N = \{A \in M_N(\mathbb{C}) | \text{Re}\lambda_i < 0 (i = 1, 2, \dots, N)\}$

şeklinde tanımlı olmak üzere A matrisi Schur kararlı ise S_N kümesinin bir elemanıdır. A matrisi Hurwitz kararlı ise H_N kümesinin bir elemanıdır. Yani

$$\begin{aligned} A \in S_N &\Leftrightarrow \sigma(A) \subset C_s \\ A \in H_N &\Leftrightarrow \sigma(A) \subset C_H \end{aligned}$$

olması ile aynı anlama gelmektedir.

2.5.2. Lyapunov matris denklemi yardımıyla kararlılık ve kararlılığın kalitesi

2.5.2.1. Fark Lyapunov matris denklemi ile kararlılık ve Schur kararlılık Parametresi (ω)

Bu kısımda Schur kararlılık için Lyapunov Teoremi ve bu teoreme bağlı kararlılığın kalitesini belirleyen ω parametresi verilmiştir.

Teorem 2.1. (Lyapunov Teoremi): A matrisi (2.16) fark sisteminin katsayı matrisi olmak üzere,

$$A^*HA - H + C = 0, C = C^* > 0 \quad (2.17)$$

Lyapunov fark matris denkleminin simetrik pozitif tanımlı çözümü ($H = H^*$) olması için gerek ve yeter şart A matrisinin Lyapunov' a göre fark asimtotik kararlı (Schur kararlı) olmasıdır (Akın ve Bulgak, 1998; Bulgak, 1999; Elaydi, 1999).

Lyapunov fark matris denkleminin çözümü sistemin Schur kararlılığı hakkında bilgi verse de kararlılığın kalitesi hakkında yorum yapma imkanı sağlamaz. Bu noktada dinamik sistemlerin hem kararlılığını tespit etmeye hem de kararlılığın kalitesi hakkında değerlendirme yapmaya imkan sağlayan kararlılık parametresi bulunmaktadır. Bu parametre, A matrisi Schur kararlı ise

$$\omega(A) = \|H\|; A^*HA - H + I = 0, H = H^* > 0$$

şeklinde tanımlanmaktadır. A matrisinin Schur kararlı olması $1 \leq \omega(A) < \infty$ ($\omega(A)$ sonlu) anlamına gelmektedir. $\omega(A) = 1$ olma durumu mükemmel durum olarak ifade edilmektedir (bak. Bulgakov ve Godunov, 1988; Akın ve Bulgak, 1998; Bulgak, 1999). Kararlılığın kalitesini daha iyi kavrayabilmek için

$$A_k(\gamma) = \begin{pmatrix} -\gamma & 10^{k-1} - 1 \\ 0 & \gamma \end{pmatrix}, |\gamma| < 1, k \in \mathbb{N}$$

matrisi ele alınsın. $\gamma = 0.1$ alınırsa $A_k(0.1)$ matrisleri Schur kararlıdır ve $k \in \mathbb{N}$ için $\sigma(A_k(0.1)) = \{-0.1, 0.1\}$ dir. $A_k(0.1)$ matrislerinin öz değerleri aynıdır. Fakat Tablo 2.1' den görüldüğü üzere hepsinin ω değeri farklıdır. Ayrıca k değeri arttıkça ω değeri artmaktadır. Dolayısıyla kararlılığın kalitesi düşmektedir (bak. Topcu, 2023).

k	1	2	3	4
$\omega(A_k)$	1.0101	82.0282	9803.0003	998102.9301

Tablo 2.1 A_k matrislerinin $\omega(A_k)$ değerleri

Bu parametre kullanılarak, sistemlerin kararlılığının dış etkilere karşı dayanıklılığı, parametrelerin hassasiyet analizi aracılığıyla belirlenebilmektedir. Hassasiyet analizi ise, süreklilik teoremleri yardımıyla gerçekleştirilmektedir. Süreklilik teoremlerinden bazılarını aşağıdaki gibi verelim (Bulgak, 1999; Aydın ve ark., 2002; Duman ve Aydın, 2011).

Teorem 2.2: A Schur kararlı bir matris ($A \in S_N, \omega(A) < \infty$) ve

$$\|B\| < \frac{1}{20\omega^{\frac{3}{2}}(A)}$$

şartını sağlayan B matrisleri için $A + B$ matrisleri de Schur kararlıdır ($A + B \in S_N$) (Bulgak, 1999).

Teorem 2.3: A Schur kararlı bir matris ($A \in S_N, \omega(A) < \infty$) ve

$$\|B\| < \sqrt{\|A\|^2 + \frac{1}{\omega(A)}} - \|A\| \quad (2.18)$$

şartını sağlayan B matrisleri için $A + B$ matrisleri de Schur kararlıdır ($A + B \in S_N$) (Aydın ve ark., 2002; Duman ve Aydın, 2011).

Schur kararlılık aralıklarının belirlenmesinde süreklilik teoremleri kullanılmaktadır. Ancak, bu aralıklar her zaman etkili olmayabilir veya kullanılan programlardan kaynaklı hatalar ortaya çıkabilir. Bu tür sorunların önlenmesi amacıyla, kullanıcı tarafından seçilen ω^* parametresi ile daha güvenli bir bölge sağlayan pratik Schur kararlılık kavramı ön plana çıkmaktadır.

2.5.2.2. Dinamik sistemlerin pratik Schur kararlılığı (ω^*)

$1 < \omega^* \in \mathbb{R}$ ve ω^* kullanıcı tarafından belirlenen ve bir parametre olmak üzere, eğer belirlenen bir A matrisi için $\omega(A) \leq \omega^*$ koşulu sağlanıyorsa, A matris ω^* – Schur kararlı olarak adlandırılır. Aksi durumda ise ω^* – Schur kararsız kabul edilir (Akın ve Bulgak, 1998; Aydın ve ark., 2001). ω^* – Schur kararlılık özelliği taşıyan matrisler, $\omega(A) \leq \omega^*$ koşuluyla tanımlı olan

$$S_N^* = \{A \in S_N \mid \omega(A) \leq \omega^*\} \subset S_N$$

kümesine aittir. Pratik Schur kararlılık, Schur kararlık ile karşılaştırıldığında, daha güvenli sonuçlar verdiği ifade edilmektedir. Bunun sebebi, pratik Schur kararlılık bölgesinin, Schur kararlılık bölgesinin içinde yer almasıdır ($S_N^* \subset S_N$). Schur kararlılık ile bulunan aralık teorik olarak güçlüdür, ancak kullanışlı olmayabilir. Schur kararlılık ile bulunan aralıklarda ω parametresi aşırı büyüyebilir ve ele alınan problem için bu durum kullanışlı olmayabilir. Bu noktada pratik Schur kararlılık parametresi (ω^*) devreye girer. ω^* – Schur kararlılık ile ilgili süreklilik teoremi aşağıda verilecektir.

Teorem 2.4: A, ω^* – Schur kararlı bir matris ($A \in S_N^*, \omega(A) \leq \omega^*$) ve

$$\|B\| < \sqrt{\|A\|^2 + \frac{\omega^* - \omega(A)}{\omega^* \omega(A)}} - \|A\| \quad (2.19)$$

şartını sağlayan B matrisleri için $A + B$ matrisleri de pratik Schur kararlıdır ($A + B \in S_N^*$) (Duman, 2008; Duman ve Aydın, 2011).

2.5.2.3. Lyapunov matris denklemi ile kararlılık ve Hurwitz kararlılık parametresi (κ)

Bu kısımda Hurwitz kararlılık için Lyapunov Teoremi ve bu teoreme bağlı kararlılığın kalitesini belirleyen κ parametresi verilmiştir.

Teorem 2.5: (Lyapunov Teoremi) A karesel ve (2.16) diferensiyel sisteminin katsayı matrisi olmak üzere

$$A^*F + FA + C = 0, C = C^* > 0 \quad (2.20)$$

Lyapunov matris denkleminin simetrik pozitif tanımlı çözümü ($F = F^*$) olması için gerek ve yeter şart A matrisinin Lyapunov' a göre asimtotik kararlı (Hurwitz kararlı) olmasıdır (Lyapunov, 1892; Bulgakov, 1995; Bulgakov, 1999) (Aider Yakub Bulgakov (Bulgakov A. Y.) ile Haydar Bulgakov aynı kişidir).

Lyapunov matris denkleminin çözümü sistemin Hurwitz kararlılığı hakkında bilgi verse de kararlılığın kalitesi hakkında yorum yapma imkanı sağlamaz. Bu noktada dinamik sistemlerin hem kararlılığını tespit etmede hem de kararlılığın kalitesini hakkında değerlendirmede imkan sağlayan kararlılık parametresi bulunmaktadır. Bu parametre, A matrisi Hurwitz kararlı ise

$$\kappa(A) = 2\|A\|\|F\|; A^*F + FA + I = 0, F = F^* > 0$$

şeklinde tanımlanmaktadır. A matrisinin Hurwitz kararlı olması $1 \leq \kappa(A) < \infty$ ($\kappa(A)$ sonlu) anlamına gelmektedir. $\kappa(A) = 1$ olma durumu mükemmel durum olarak ifade edilir (Bulgakov, 1980; Bulgakov ve Godunov, 1988; Bulgakov, 1995). Kararlılığın kalitesini daha iyi kavrayabilmek için

$$A_k(\gamma) = \begin{pmatrix} -\gamma & 10^{k-1} - 1 \\ 0 & -\gamma \end{pmatrix}, \gamma > 0, k \in \mathbb{N}$$

matrisi ele alalım $\gamma = 1$ için $A_k(\gamma)$ matrisleri Hurwitz kararlıdır ve $k \in \mathbb{N}$ için $\sigma(A_k(\gamma)) = \{-1, -1\}$ dir. A_k matrislerinin öz değerleri aynıdır fakat Tablo 2.2' den

görüldüğü üzere hepsinin κ değeri farklıdır. Ayrıca k değeri arttıkça κ değeri artmaktadır. Dolayısıyla kararlılığın kalitesi düşmektedir (bak Topcu, 2023).

k	1	2	3	4
$\kappa(A_k)$	1	382.5555	485347.5050	498503497.5005

Tablo 2.2 A_k matrislerinin $\kappa(A_k)$ değerleri

Fark sistemlere benzer şekilde κ parametresi kullanılarak, sistemlerin kararlılığının dış etkilere karşı dayanıklılığı, parametrelerin hassasiyet analizi aracılığıyla belirlenebilmektedir. Hassasiyet analizi ise, süreklilik teoremleri yardımıyla gerçekleştirilmektedir. Aşağıda diferensiyel sistemler için süreklilik teoremlerinden bazıları aşağıda verilmektedir (Bulgak, 1999; Duman ve Aydın, 2017).

Teorem 2.6: A Hurwitz kararlı bir matris ($A \in H_N, \kappa(A) < \infty$) ve

$$\frac{\|B\|}{\|A\|} \leq \frac{1}{15\kappa(A)}$$

şartını sağlayan B matrisleri için $A + B$ matrisleri de Hurwitz kararlıdır ($A + B \in H_N$) (Bulgak, 1999).

Teorem 2.7: A Hurwitz kararlı bir matris ($A \in H_N, \kappa(A) < \infty$) ve

$$\|B\| < \frac{\|A\|}{\kappa(A)} \quad (2.21)$$

şartını sağlayan B matrisleri için $A + B$ matrisleri de Hurwitz kararlıdır ($A + B \in H_N$) (Duman ve Aydın, 2017).

2.5.2.4. Dinamik sistemlerin pratik Hurwitz kararlılığı (κ^*)

$1 < \kappa^* \in \mathbb{R}$ ve κ^* kullanıcı tarafından belirlenen ve bir parametre olmak üzere, eğer belirlenen bir A matrisi için $\kappa(A) \leq \kappa^*$ koşulu sağlanıyorsa, A matrisi κ^* – Hurwitz kararlı olarak adlandırılır. Aksi durumda ise κ^* – Hurwitz kararsız kabul edilir (Akın ve Bulgak, 1998; Aydın ve ark., 2001). κ^* – Hurwitz kararlılık özelliği taşıyan matrisler, $\kappa(A) \leq \kappa^*$ koşuluyla tanımlı olan

$$H_N^* = \{A \in H_N \mid \kappa(A) \leq \kappa^*\} \subset H_N$$

kümesine aittir. Pratik Hurwitz kararlılık, Hurwitz kararlılık ile karşılaştırıldığında, daha güvenli sonuçlar verdiği ifade edilmektedir. Bunun sebebi, pratik Hurwitz kararlılık bölgesinin, Hurwitz kararlılık bölgesinin içinde yer almasıdır ($H_N^* \subset H_N$). Hurwitz kararlılık ile bulunan aralık teorik olarak güçlüdür, ancak Hurwitz kararlılık ile bulunan aralıklarda κ parametresi aşırı büyüyebilir ve ele alınan problem için bu durum kullanışlı olmayabilir. Bu noktada pratik Hurwitz kararlılık parametresi (κ^*) devreye girer. κ^* – Hurwitz kararlılık ile ilgili süreklilik teoremi aşağıda verilecektir.

Teorem 2.8: A pratik Hurwitz kararlı bir matris ($A \in H_N^*$, $\kappa(A) \leq \kappa^*$) ve

$$\|B\| \leq \frac{(\kappa^* - \kappa(A))\|A\|}{(1 + \kappa^*) \kappa(A)} \quad (2.22)$$

şartını sağlayan B matrisleri için $A + B$ matrisleri de pratik Hurwitz kararlıdır ($A + B \in H_N^*$) (Duman ve Aydın, 2017).

3. KARARLILIK ARALIĞININ GENİŞLETİLMESİ ÜZERİNE ALGORİTMALAR VE PYTHON YORDAMLARI

Matris teorisinde, bir matris ailesi, belirli bir kurallar kümesi altında birleşen matrislerden oluşur. Bu bölümde, matrislerin lineer kombinasyonlarından oluşan matris ailelerini ele alınmıştır.

$M_N(\mathbb{C})$, elemanları karmaşık sayılar olan N boyutlu karesel matrislerin vektör uzayı, $A_1, A_2 \in M_N(\mathbb{C}), r \in \mathbb{R}$ için, $A(r) = A_1 + rA_2$ lineer toplam şeklinde tanımlı olsun. Lineer toplamlardan oluşan matris ailesi

$$\mathcal{L}(A_1, A_2) = \{A(r) = A_1 + rA_2 \mid A_1, A_2 \in M_N(\mathbb{C})\}$$

şeklinde gösterilir (Topcu, 2023).

3.1. Matris Ailesinin Schur Kararlılığı ve Pratik Schur Kararlılığı

A_1 Schur kararlı bir matris ve $A_2 \in M_N(\mathbb{C})$ olmak üzere $\mathcal{L}(A_1, A_2)$ matris ailesinin Schur kararlılığını sağlayan $\mathfrak{I}_{\mathcal{L}}^S$ aralığını ve ω^* – Schur kararlılığını sağlayan $\mathfrak{I}_{\mathcal{L}}^{S*}$ aralığını belirleyen süreklilik teoremleri aşağıda verilmiştir.

Teorem 3.1: $A_1 \in S_N, A_2 \in M_N(\mathbb{C})$ için

$$-l = u = -\frac{\|A_1\|}{\|A_2\|} + \frac{1}{\|A_2\|} \sqrt{\|A_1\|^2 + \frac{1}{\omega(A_1)}}, l < \underline{r} < \bar{r} < u$$

olmak üzere $r \in \mathfrak{I}_{\mathcal{L}}^S = [\underline{r}, \bar{r}]$ ise $\mathcal{L}(A_1, A_2) \subset S_N$ dir (Topcu, 2023; Topcu ve Aydın, 2023b).

Teorem 3.2: $A_1 \in S_N^*, A_2 \in M_N(\mathbb{C})$ için

$$-l^* = u^* = -\frac{\|A_1\|}{\|A_2\|} + \frac{1}{\|A_2\|} \sqrt{\|A_1\|^2 - \frac{1}{\omega^*} + \frac{1}{\omega(A_1)}}, l^* \leq \underline{r} \leq \bar{r} < u^*$$

olmak üzere $r \in \mathfrak{I}_{\mathcal{L}}^{S*} = [\underline{r}, \bar{r}]$ ise $\mathcal{L}(A_1, A_2) \subset S_N^*$ dir (Topcu, 2023; Topcu ve Aydın, 2023b).

Bulunan $\mathfrak{I}_{\mathcal{L}}^S$ ve $\mathfrak{I}_{\mathcal{L}}^{S*}$ aralıkları lineer toplamlardan oluşan matris ailesi için Schur kararlılığı ve ω^* – Schur kararlılığı sağlayan aralıklar olmasına karşın Schur kararlılığı ve ω^* – Schur kararlılığı sağlayan daha geniş aralıklar vardır. Genişletilmiş aralıklar için

algoritma ve yöntemler Teorem 2.3, Teorem 2.4, Teorem 3.1 ve Teorem 3.2' ye dayanarak literatürde ayrıntılı olarak verilmiştir (Topcu, 2023; Topcu ve Aydın, 2023b).

Teorem 3.1 ve Teorem 3.2 ile varlığı garanti edilen $\mathfrak{I}_{\mathcal{L}}^S$ ve $\mathfrak{I}_{\mathcal{L}}^{S*}$ aralıkların kararlılık özellikleri korunacak şekilde genişletilmesine yönelik yazılan algoritmalar ve python kodlarını verelim.

3.1.1. Genişletilmiş Schur kararlılık aralığının elde edilmesi için algoritma ve Python kodları

Bu kısımda $\mathfrak{I}_{\mathcal{L}}^S$ ve $\mathfrak{I}_{\mathcal{L}}^{S*}$ aralıklarının üst ve alt sınırını genişletmek için literatürde bulunan bazı algoritmalar verilmiştir. Daha sonra bu algoritmalarla hesap yapmak için yordamlar yazılmış hesaplama fonksiyonları tanıtılmıştır.

3.1.1.1. $\mathfrak{I}_{\mathcal{L}}^{Se}$ aralığı için üst ve alt sınır algoritmaları ile Python kodları

Algoritma 3.1. (Schur Kararlılık için Üst Sınır (u^e)): Teorem 3.1 ile elde edilen aralığın üst sınırını genişletmek için aşağıda verilen algoritma kullanılmaktadır (Topcu ve Aydın, 2023b; Topcu, 2023).

1. Veri girişi;

$$A \in S_N, B \in M_N(\mathbb{C})$$

yeterince küçük r^* ,

$$\gamma \lesssim 1$$

2. Hesaplama; $\omega(A), \|A\|, \|B\|$

$$\beta = -\frac{\|A\|}{\|B\|} + \frac{1}{\|B\|} \sqrt{\|A\|^2 + \frac{1}{\omega(A)}}, r_1 = \gamma\beta$$

3. Veri yenileme; $k = 1, A_1 := A, u_1 := r_1$

4. Hesaplama;

$$A_{k+1} = A_k + r_k B, \omega(A_{k+1}),$$

$$\beta_{k+1} = -\frac{\|A_{k+1}\|}{\|B\|} + \frac{1}{\|B\|} \sqrt{\|A_{k+1}\|^2 + \frac{1}{\omega(A_{k+1})}},$$

$$r_{k+1} = \gamma\beta_{k+1}.$$

5. Eğer $r_{k+1} \geq r^*$ ise

$$u_{k+1} = u_k + r_{k+1} \text{ eşitliğini hesapla,}$$

$k := k + 1$ al e 4. adıma git.

6. $M := k$ al e aralığın üst sınırını $u^e = u_M$ olarak yaz.

Şimdi Algoritma 3.1 ile ilgili Python kodunu verelim.

Yordam 3.1 ($\text{üstsinir}(A, B, ry)$) : Verilen A katsayı matrisi ve B bozunum matrisi olmak üzere yeterince küçük $r^*(= ry)$ pratik değerleri için bize r^* hassasiyetinde üst sınırı veren Python dilinde $\text{üstsinir}(A, B, ry)$ fonksiyonunu aşağıda ki gibi verelim. Burada r^* kullanıcı tarafından seçilen yeterince küçük pratiklik parametresi olarak adlandırılır (Çelik Kızıllan, 2009).

```
def üstsinir(A,B,ry):
    import math ; import numpy as np; from scipy import linalg ; import sympy as sp ; from numpy import linalg as LA
    print("Matrisin Boyutu:");n = int(input())
    print("A Matrisinin Elemanları:")
    A = [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
    A= np.array(elemanlar).reshape(n,n) ; print("\n A=", A)
    print("B Matrisinin Elemanları:")
    B= [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
    B= np.array(elemanlar).reshape(n,n) ; print("\n B=", B)
    b=LA.norm(B, 2) ; print("||B||=",b) ; ry=float(input()) ; print("r0=",ry)
    i=0; u=0
    while True:
        i=i+1; print("\n i=",i)
        Q=np.eye(n) # Birim matris
        H = linalg.solve_discrete_lyapunov(A, Q) ; print("H= ", H)
        k= np.all(np.linalg.eigvals(H) > 0)

        if(k):
            print("H matrisi Pozitif tanımlıdır.")
            h=LA.norm(H, 2) ; a=LA.norm(A, 2) ; print("ω(A)=", h, ", ", "||A||=", a)
            z=(-a/b)+math.sqrt(a**2+(1/h))/b ; print("z=", z)
            r=z*0.90 ; print("r=", r)
            if r<ry :
                break

            A=A+r*B ; print("A{ }={ }".format(i,A))
            u=u+r ; print("u=", u)

        else:
            print("H matrisi Pozitif Tanımlı değildir.")

    print("\nİterasyon sayısı N : ", i-1)
```

Yordam 3.1: Algoritma 3.1. ile Python yazılım dilinde $\text{üstsinir}(A, B, ry)$ fonksiyonunun kodları.

Algoritma 3.2 (Schur Kararlılık için Alt Sınır (l^e)): Teorem 3.1 ile elde edilen aralığın alt sınırını genişletmek için aşağıda verilen algoritma kullanılmaktadır (Topcu, G. ve Aydın, K., 2023b; Topcu, G., 2023).

1. Veri girişi;

$$A \in S_N, B \in M_N(\mathbb{C})$$

yeterince küçük r^* ,

$$\gamma \lesssim 1$$

2. Hesaplama; $\omega(A)$, $\|A\|$, $\|B\|$

$$\beta = -\frac{\|A\|}{\|B\|} + \frac{1}{\|B\|} \sqrt{\|A\|^2 + \frac{1}{\omega(A)}}, r_1 = \gamma\beta$$

3. Veri yenileme; $k = 1, A_1 := A, u_1 := r_1$

4. Hesaplama;

$$A_{k+1} = A_k - r_k B, \omega(A_{k+1}),$$

$$\beta_{k+1} = -\frac{\|A_{k+1}\|}{\|B\|} + \frac{1}{\|B\|} \sqrt{\|A_{k+1}\|^2 + \frac{1}{\omega(A_{k+1})}},$$

$$r_{k+1} = \gamma\beta_{k+1}.$$

5. Eğer $r_{k+1} \geq r^*$ ise

$$l_{k+1} = l_k - r_{k+1} \quad (l_1 := -r_1) \text{ eşitliğini hesapla,}$$

$$k := k + 1 \text{ al e 4. adıma git.}$$

6. $M := k$ al e aralığın üst sınırını $l^e = l_M$ olarak yaz.

Şimdi Algoritma 3.2 ile ilgili Python yordamı verelim.

Yordam 3.2 (altsınır(A, B, ry)): Verilen A katsayı matrisi ve B bozunum matrisi olmak üzere yeterince küçük $r^*(=ry)$ pratik değerleri için, r^* hassasiyetinde alt sınırı veren Python dilinde altsınır(A, B, ry) fonksiyonunu aşağıda ki gibi verelim.

```
def altsınır(A,B,ry):
    import math ; import numpy as np; from scipy import linalg ; import sympy as sp ; from numpy import linalg as LA
    print("Matrisin Boyutu:"); n = int(input())
    print("A Matrisinin Elemanları:")
    A = [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
    A= np.array(elemanlar).reshape(n,n) ; print("\n A=", A)
    print("B Matrisinin Elemanları:")
    B= [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
    B= np.array(elemanlar).reshape(n,n) ; print("\n B=", B)
    b=LA.norm(B, 2) ; print("||B||=", b) ; ry=float(input()) ; print("r0=", ry)
    i=0; l=0
    while True:
        i=i+1; print("\n i=", i)
        Q =np.eye(n) # Birim matris
        H = linalg.solve_discrete_lyapunov(A, Q) ; print("H= ", H)
        k= np.all(np.linalg.eigvals(H) > 0)

        if(k):
            print("H matrisi Pozitif tanımlıdır.")
            h=LA.norm(H, 2) ; a=LA.norm(A, 2) ; print("ω(A)= ", h, ", ", "||A||=", a)
            z=(-a/b)+math.sqrt(a**2+(1/h)/b) ; print("z=", z)
            r=z*0.90 ; print("r=", r)
            if r<ry :
                break

            A=A-r*B ; print("A{}={}".format(i,A))
            l=l-r ; print("l=", l)

        else:
            print("H matrisi Pozitif Tanımlı değildir.")

    print("\n İterasyon sayısı N : ", i-1)
```

Yordam 3.2: Algoritma 3.2. ile Python yazılım dilinde altsınır(A, B, ry) fonksiyonunun kodları.

Elde edilen u^e genişletilmiş üst sınır ve l^e genişletilmiş alt sınır değerleri, $\mathfrak{I}_L^{Se} = [l^e, u^e]$ genişletilmiş Schur kararlılık aralığını oluşturmaktadır.

Uyarı 3.1: γ değeri 1 den küçük ve 1' e yeterince yakın seçilmelidir. Algoritma içerisindeki $\gamma \lesssim 1$ bu anlama gelmektedir. Bazı durumlarda süreklilik teoremi yardımıyla elde edilen sınır keskin olabilir dolayısıyla bu durum elde edilen A_{k+1} matrisini Schur kararsız matris yapabilmektedir. Örneğin;

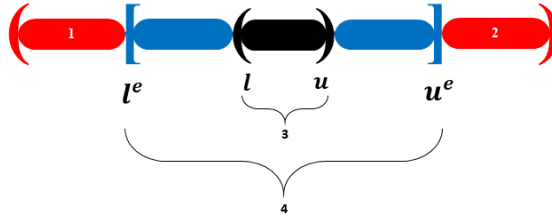
$$A = \begin{pmatrix} 0.8 & 0 \\ 0 & 0.1 \end{pmatrix}$$

matrisini ele alırsak süreklilik teoremi ile belirlenen sınır 0.19999999999999996 olarak bulunur. Bir adım daha ilerletilince

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0.1 \end{pmatrix}$$

matrisi elde edilir. Elde edilen bu matris kararsızdır. Bu nedenle, daha küçük adımlar atılmasını sağlayarak kontrollü bir genişleme imkanı sunan γ parametresi kullanılmaktadır. Yapılan çalışmalarda, $\gamma = 0.9$ değeri seçilerek analizler gerçekleştirilecektir.

Uyarı 3.2:



Şekil 3.1: Teorem 3.2 ile elde edilen Schur kararlı aralık ve Yordam 3.1, Yordam 3.2 ile elde edilen genişletilmiş aralık

Şekil 3.1 de 1 ve 2 bölgelerinden seçilen değerler, kararlı veya kararsız olabilir. 3 aralığı Teorem 3.1 ile r için elde edilen Schur kararlı aralığı temsil etmektedir. 4 aralığı $\text{üst sınır}(A, B, ry)$ ve $\text{alt sınır}(A, B, ry)$ fonksiyonu ile r için elde edilen genişletilmiş Schur kararlı aralığı temsil etmektedir.

3.1.1.2 $\mathfrak{T}_{\mathcal{L}}^{S^*e}$ aralığı için üst ve alt sınır algoritmaları ile Python kodları

Algoritma 3.3 (Pratik Schur Kararlılık için Üst Sınır (u^{*e})): Teorem 3.2 ile elde edilen aralığın üst sınırını genişletmek için aşağıda verilen algoritma kullanılmaktadır (Topcu ve Aydın 2023b; Topcu, 2023).

1. Veri girişi;

$$A \in S_N, B \in M_N(\mathbb{C})$$

yeterince küçük r^* ,

$1 < \omega^*$ problemin türüne göre kullanıcı tarafından belirlenen bir sayı.

2. Hesaplama; $\omega(A)$.

3. Eğer $\omega(A) > \omega^*$ ise “ A matrisi ω^* -Schur kararlı değildir” yaz ve algoritmayı bitir.

4. Hesaplama; $\|A\|, \|B\|$,

$$u^* = -\frac{\|A\|}{\|B\|} + \frac{1}{\|B\|} \sqrt{\|A\|^2 - \frac{1}{\omega^*} + \frac{1}{\omega(A)}}.$$

5. Veri yenileme; $k = 1, A_1 := A, r_1 := u^*, u_1 := r_1$.

6. Hesaplama;

$$A_{k+1} = A_k + r_k B, \omega(A_{k+1}),$$

$$r_{k+1} = -\frac{\|A_{k+1}\|}{\|B\|} + \frac{1}{\|B\|} \sqrt{\|A_{k+1}\|^2 - \frac{1}{\omega^*} + \frac{1}{\omega(A_{k+1})}}.$$

7. Eğer $r_{k+1} \geq r^*$ ise

$$u_{k+1} = u_k + r_{k+1} \text{ eşitliğini hesapla,}$$

$k := k + 1$ al e 6. adıma git.

8. $M := k$ al e aralığın üst sınırını $u^{*e} = u_M$ olarak yaz.

Algoritma 3.3 için Python yordamını verelim.

Yordam 3.3 ($pr_üstsınır(A, B, wy)$): Verilen A katsayı matrisi ve B bozunum matrisi olmak üzere yeterince küçük ω^* pratik değerleri için, ω^* hassasiyetinde üst sınırı veren Python dilinde $pr_üstsınır(A, B, wy)$ fonksiyonunu aşağıda ki gibi verelim.

```

def pr_üstün(A,B,wy):
    import math ; import numpy as np; from scipy import linalg ; import sympy as sp ; from numpy import linalg as LA
    print("Matrisin Boyutu:") ; n = int(input())
    print("A Matrisinin Elemanları:")
    u=0; A = [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
    A= np.array(elemanlar).reshape(n,n) ; print("\n A=", A)
    print("B Matrisinin Elemanları:")
    B= [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
    B= np.array(elemanlar).reshape(n,n) ; print("\n B=", B)
    b=LA.norm(B, 2) ; print("||B||=", b)
    print("r* değeri girin") ; ry=float(input()) ; print("r*=",ry); print("ω* değeri girin") ; wy=float(input()) ; print("ω*=",wy)
    i=0
    while True:
        i=i+1; print("\n i=",i)
        Q =np.eye(n) # Birim matris
        H = linalg.solve_discrete_lyapunov(A, Q) ; print("H= ", H)
        k= np.all(np.linalg.eigvals(H) > 0)

        if(k):
            print("H matrisi Pozitif tanımlıdır.")
            h=LA.norm(H, 2) ; a=LA.norm(A, 2) ; print("ω(A)= ", h, ", ", "||A||=", a)
            z=(-a/b)+math.sqrt(a**2-(1/wy)+(1/h))/b ; print("z=", z)
            r=z*0.90 ; print("r=", r)
            if r<ry :
                break

            A=A+r*B ; print("A{ }={ }".format(i,A))
            u=u+r ; print("u=", u)
        else:
            print("H matrisi Pozitif Tanımlı değildir.")

    print("\nİterasyon sayısı N : ", i-1)

```

Yordam 3.3: Algoritma 3.3' ün Python yazılım dilinde kodlanması

Algoritma 3.4 (Pratik Schur Kararlılık için Alt Sınır (l^*)): Teorem 3.2 ile elde edilen aralığın alt sınırını genişletmek için aşağıda verilen algoritma kullanılmaktadır (Topcu ve Aydın 2023b; Topcu, 2023).

1. Veri girişi;

$$A \in S_N, B \in M_N(\mathbb{C})$$

yeterince küçük r^* ,

$1 < \omega^*$ problemin türüne göre kullanıcı tarafından belirlenen bir sayı.

2. Hesaplama; $\omega(A)$.

3. Eğer $\omega(A) > \omega^*$ ise “A matrisi ω^* - Schur kararlı değildir” yaz ve algoritmayı bitir.

4. Hesaplama; $\|A\|, \|B\|$,

$$l^* = -\frac{\|A\|}{\|B\|} + \frac{1}{\|B\|} \sqrt{\|A\|^2 - \frac{1}{\omega^*} + \frac{1}{\omega(A)}}.$$

5. Veri yenileme; $k = 1, A_1 := A, r_1 := l^*, l_1 := r_1$.

6. Hesaplama;

$$A_{k+1} = A_k - r_k B, \omega(A_{k+1}),$$

$$r_{k+1} = -\frac{\|A_{k+1}\|}{\|B\|} + \frac{1}{\|B\|} \sqrt{\|A_{k+1}\|^2 - \frac{1}{\omega^*} + \frac{1}{\omega(A_{k+1})}}.$$

7. Eğer $r_{k+1} \geq r^*$ ise

$l_{k+1} = l_k - r_{k+1}$ eşitliğini hesapla,

$k := k + 1$ al e 6. adıma git.

8. $M := k$ al e aralığın alt sınırını $l^{*e} = l_M^*$ olarak yaz.

Algoritma 3.4 için Python yordamı verelim.

Yordam 3.4 ($pr_altsınır(A, B, wy)$): Verilen A katsayı matrisi ve B bozunum matrisi olmak üzere yeterince küçük ω^* pratik değerleri için, ω^* hassasiyetinde alt sınırı veren Python dilinde $pr_altsınır(A, B, wy)$ fonksiyonunu aşağıda ki gibi verelim.

```
def pr_altsınır(A,B,wy):
    import math ; import numpy as np; from scipy import linalg ; import sympy as sp ; from numpy import linalg as LA
    print("Matrisin Boyutu:"); n = int(input())
    print("A Matrisinin Elemanları:")
    l=0; A = [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
    A= np.array(elemanlar).reshape(n,n) ; print("\n A=", A)
    print("B Matrisinin Elemanları:")
    B= [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
    B= np.array(elemanlar).reshape(n,n) ; print("\n B=", B)
    b=LA.norm(B, 2) ; print("||B||=", b)
    print("r* değeri girin") ; ry=float(input()) ; print("r*=",ry); print("ω* değeri girin") ; wy=float(input()) ; print("ω*=",wy);
    i=0
    while True:
        i=i+1; print("\n i=",i)
        Q =np.eye(n) # Birim matris
        H = linalg.solve_discrete_lyapunov(A, Q) ; print("H= ", H)
        k= np.all(np.linalg.eigvals(H) > 0)

        if(k):
            print("H matrisi Pozitif tanımlıdır.")
            h=LA.norm(H, 2) ; a=LA.norm(A, 2) ; print("ω(A)=", h, ", ", "||A||=", a)
            z=(-a/b)+math.sqrt(a**2-(1/wy)+(1/h))/b ; print("z=", z)
            r=z*0.90 ; print("r=", r)
            if r<ry :
                break

            A=A-r*B ; print("A{ }={ }".format(i,A))
            l=l-r ; print("l=", l)

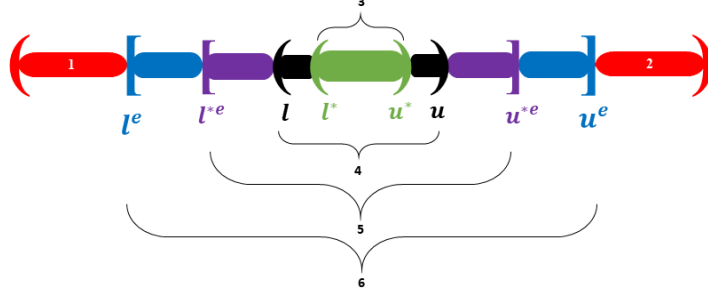
        else:
            print("H matrisi Pozitif Tanımlı değildir.")

    print("\nİterasyon sayısı N : ", i-1)
```

Yordam 3.4: Algoritma 3.4'ün Python yazılım dilinde kodlanması

Son olarak bulunan genişletilmiş üst sınır u^{*e} ve genişletilmiş alt sınır l^{*e} değerleri, genişletilmiş $\omega^* -$ Schur kararlılık aralığını $\mathfrak{T}^{Se} = [l^{*e}, u^{*e}]$ oluşturmaktadır.

Uyarı 3.3:



Şekil 3.2: Schur kararlı aralık, pratik Schur kararlı aralık ve kararsız aralığı temsil eden bir şekil.

Şekil 3.2’ de 4 aralığı Teorem 3.1 ile r için elde edilen Schur kararlı aralığı temsil etmektedir. 6 aralığı $\text{üstsinir}(A, B, ry)$ ve $\text{altsinir}(A, B, ry)$ fonksiyonu ile r için elde edilen genişletilmiş Schur kararlı aralığı temsil etmektedir. 3 aralığı Teorem 3.2 ile r için elde edilen ω^* – Schur kararlı aralığı temsil etmektedir. 5 aralığı $\text{pr_üstsinir}(A, B, \omega^*)$ ve $\text{pr_altsinir}(A, B, \omega^*)$ fonksiyonu ile r için elde edilen genişletilmiş ω^* – Schur kararlı aralığı temsil etmektedir. Benzer mantıkla Hurwitz kararlılık için aynı şekiller düşünülebilir.

3.2. Matris Ailesinin Hurwitz Kararlılığı ve Pratik Hurwitz Kararlılığı

A_1 Hurwitz kararlı bir matris ve $A_2 \in M_N(\mathbb{C})$ için $\mathcal{L}(A_1, A_2)$ matris ailesinin Hurwitz kararlılığını sağlayan $\mathfrak{T}_{\mathcal{L}}^H$ aralığını ve κ^* – Hurwitz kararlılığını sağlayan $\mathfrak{T}_{\mathcal{L}}^{H*}$ aralığını belirleyen süreklilik teoremleri aşağıda verilmiştir.

Teorem 3.3: $A_1 \in H_N, A_2 \in M_N(\mathbb{C})$ için $-l = u = \frac{\|A_1\|}{\|A_2\|\kappa(A_1)}, l < \underline{r} < \bar{r} < u$ olmak üzere $r \in \mathfrak{T}_{\mathcal{L}}^H = [\underline{r}, \bar{r}]$ ise $\mathcal{L}(A_1, A_2) \subset H_N$ dir (Topcu, 2023; Topcu ve Aydın, 2023b).

Teorem 3.4: $A_1 \in H_N^*, A_2 \in M_N(\mathbb{R})$ için

$$-l^* = u^* = \frac{(\kappa^* - \kappa(A))\|A_1\|}{(1 + \kappa^*)\|A_2\|\kappa(A_1)}, l^* \leq \underline{r} \leq \bar{r} < u^*$$

olmak üzere $r \in \mathfrak{T}_{\mathcal{L}}^{H*} = [\underline{r}, \bar{r}]$ ise $\mathcal{L}(A_1, A_2) \subset H_N^*$ dir (Topcu, 2023; Topcu ve Aydın, 2023b).

Elde edilen $\mathfrak{I}_{\mathcal{L}}^H$ ve $\mathfrak{I}_{\mathcal{L}}^{H*}$ aralıkları, $\mathcal{L}$ matris ailesi için Hurwitz kararlılığı ve κ^* – Hurwitz kararlılığı sağlayan aralıklar olmasına karşın Hurwitz kararlılığı ve κ^* – Hurwitz kararlılığı sağlayan daha geniş aralıklar vardır. Genişletilmiş aralıklar için algoritma ve yöntemler Teorem 2.6, Teorem 2.7, Teorem 3.3 ve Teorem 3.4’ ye dayanarak literatürde ayrıntılı olarak verilmiştir (Topcu, 2023; Topcu ve Aydın, 2023a).

Şimdi Teorem 3.3 ve Teorem 3.4 ile varlığı garanti edilen $\mathfrak{I}_{\mathcal{L}}^H$ ve $\mathfrak{I}_{\mathcal{L}}^{H*}$ aralıkların kararlılık özellikleri korunacak şekilde genişletilmesine yönelik algoritmaları ve Python kodlarını verelim.

3.2.1. Genişletilmiş Hurwitz kararlılık aralığının elde edilmesi için algoritma ve Python kodları

Bu kısımda $\mathfrak{I}_{\mathcal{L}}^H$ ve $\mathfrak{I}_{\mathcal{L}}^{H*}$ aralıklarının üst ve alt sınırını genişletmek için literatürde bulunan bazı algoritmalar verilmiştir. Daha sonra bu algoritmalarla hesap yapmak için yordamlar yazılmış hesaplama fonksiyonları tanıtılmıştır.

3.2.1.1. $\mathfrak{I}_{\mathcal{L}}^{He}$ aralığı için üst ve alt sınır algoritmaları ile python kodları

Algoritma 3.5. (Hurwitz Kararlılık için Üst Sınır (u^e)): Teorem 3.3 ile elde edilen aralığın üst sınırını genişletmek için aşağıda verilen algoritma kullanılmaktadır (Topcu ve Aydın, 2023a; Topcu, 2023).

1. Veri girişi;

$$A \in H_N, B \in M_N(\mathbb{C})$$

yeterince küçük r^* ,

yeterince büyük κ^*, R ,

$$\gamma \lesssim 1$$

2. Hesaplama; $\kappa(A)$, $\|A\|$, $\|B\|$

$$\beta = \frac{\|A\|}{\|B\|\kappa(A)}, r_1 = \gamma\beta$$

3. Veri yenileme; $k = 1, A_1 := A, u_1 := r_1$

4. Hesaplama;

$$A_{k+1} = A_k + r_k B, \kappa(A_{k+1}),$$

$$\beta_{k+1} = \frac{\|A_{k+1}\|}{\|B\|\kappa(A_{k+1})},$$

$$r_{k+1} = \gamma\beta_{k+1}$$

$$u_{k+1} = u_k + r_{k+1}$$

5. Eğer $r_{k+1} < r^*$ veya $r_{k+1} > R$ veya $\kappa(A_{k+1}) > \kappa^*$ ise algoritmayı bitir aksi takdirde $k := k + 1$ al ve 4. adıma git.
6. $M := k$ al e aralığın üst sınırını $u^e = u_M$ olarak yaz.

Şimdi de bu algoritma için Python yordamını verelim.

Yordam 3.5 (*üstsınır*(A, B, ry, ky, R)): Verilen A katsayı matrisi ve B bozunum matrisi olmak üzere yeterince küçük $r^*(= ry)$ pratik değerleri, yeterince büyük $\kappa^*(= ky)$ ve yeterince büyük R değerleri için r^*, κ^*, R hassasiyetinde üst sınırı veren Python dilinde *üstsınır*(A, B, ry, ky, R) fonksiyonunu aşağıda verelim.

```
def üstsınır(A,B,ry,ky,R):
    import math ; import numpy as np; from scipy import linalg ; import sympy as sp ; from numpy import linalg as LA
    print("Matrisin Boyutu:") ; n = int(input())
    print("A Matrisinin Elemanları:")
    A = [[0 for i in range(n)] for i in range(n)] ; elemanlar= list(map(float, input().split()));
    A= np.array(elemanlar).reshape(n,n) ; print("\n A=", A)
    print("B Matrisinin Elemanları:")
    B= [[0 for i in range(n)] for i in range(n)] ; elemanlar= list(map(float, input().split()));
    B= np.array(elemanlar).reshape(n,n) ; print("\n B=", B)
    b=LA.norm(B, 2) ; print("||B||=",b)
    print("r* değeri girin") ; ry=float(input()) ; print("r*=",ry)
    print("R değeri girin") ; R=float(input()) ; print("R=",R)
    print("κ* değeri girin") ; ky=float(input()) ; print("κ*=",ky)
    i=0 ; u=0
    while True:
        i=i+1; print("\n i=",i)
        Q =-np.eye(n) # Birim matris
        H= linalg.solve_continuous_lyapunov(A, Q) ; print("H= ", H)
        m= np.all(np.linalg.eigvals(H) > 0)

        if(m):
            print("H matrisi Pozitif tanımlıdır.")
            h=LA.norm(H, 2) ; a=LA.norm(A, 2) ; k=2*a*h ; print("||A||=", a, ", κ(A)=", k)
            z=a/(b*k) ; print("z=", z)
            r=z*0.90 ; print("r=", r)
            if k>ky or r<ry or r>R :
                break

            A=A+r*B ; print("A{}={}".format(i,A))
            u=u+r ; print("u=", u)

        else:
            print("H matrisi Pozitif Tanımlı değildir.")

    print("\nİterasyon sayısı N : " , i-1)
```

Yordam 3.5: Algoritma 3.5. ile Python yazılım dilinde *üstsınır*(A, B, ry, ky, R) fonksiyonunun kodları.

Algoritma 3.6. (Hurwitz Kararlılık için Alt Sınır (l^e)): Teorem 3.3 ile elde edilen aralığın alt sınırını genişletmek için aşağıda verilen algoritma kullanılmaktadır (Topcu ve Aydın, 2023b; Topcu, 2023).

1. Veri girişi;

$$A \in H_N, B \in M_N(\mathbb{C})$$

yeterince küçük r^* ,

yeterince küçük κ^*, R ,

$$\gamma \lesssim 1$$

2. Hesaplama; $\kappa(A), \|A\|, \|B\|$

$$\beta = \frac{\|A\|}{\|B\|\kappa(A)}, r_1 = \gamma\beta$$

3. Veri yenileme; $k = 1, A_1 := A, l_1 := r_1$

4. Hesaplama;

$$A_{k+1} = A_k - r_k B, \kappa(A_{k+1}),$$

$$\beta_{k+1} = \frac{\|A_{k+1}\|}{\|B\|\kappa(A_{k+1})},$$

$$r_{k+1} = \gamma\beta_{k+1},$$

$$l_{k+1} = l_k - r_{k+1}.$$

5. Eğer $r_{k+1} < r^*$ veya $r_{k+1} > R$ veya $\kappa(A_{k+1}) > \kappa^*$ ise algoritmayı bitir aksi takdirde $k := k + 1$ al ve 4. adıma git.

6. $M := k$ al e aralığın alt sınırını $l^e = l_M$ olarak yaz.

Şimdi de bu algoritma için Python yordamını verelim.

Yordam 3.6 ($altsınır(A, B, ry, ky, R)$): Verilen A katsayı matrisi ve B bozunum matrisi olmak üzere yeterince küçük $r^*(= ry)$ pratik değerleri, yeterince büyük $\kappa^*(= ky)$ ve yeterince büyük R değerleri için r^*, κ^*, R hassasiyetinde üst sınırı veren Python dilinde $altsınır(A, B, ry, ky, R)$ fonksiyonunu aşağıda verelim.

```

def altsınır(A,B,ry,ky,R):
import math ; import numpy as np; from scipy import linalg ; import sympy as sp ; from numpy import linalg as LA
print("Matrisin Boyutu:") ; n = int(input())
print("A Matrisinin Elemanları:")
A = [[0 for i in range(n)] for i in range(n)] ; elemanlar= list(map(float, input().split()));
A= np.array(elemanlar).reshape(n,n) ; print("\n A=", A)
print("B Matrisinin Elemanları:")
B= [[0 for i in range(n)] for i in range(n)] ; elemanlar= list(map(float, input().split()));
B= np.array(elemanlar).reshape(n,n) ; print("\n B=", B)
b=LA.norm(B, 2) ; print("||B||=", b)
print("r* değeri girin") ; ry=float(input()) ; print("r*=",ry)
print("R değeri girin") ; R=float(input()) ; print("R=",R)
print("κ* değeri girin") ; ky=float(input()) ; print("κ*=",ky)
i=0 ; l=0
while True:
    i=i+1; print("\n i=",i)
    Q =-np.eye(n) # Birim matris
    H= linalg.solve_continuous_lyapunov(A, Q) ; print("H= ", H)
    m= np.all(np.linalg.eigvals(H) > 0)

    if(m):
        print("H matrisi Pozitif tanımlıdır.")
        h=LA.norm(H, 2) ; a=LA.norm(A, 2) ; k=2*a*h ; print("||A||=", a, ", κ(A)=", k)
        z=a/(b*k) ; print("z=", z)
        r=z*0.90 ; print("r=", r)
        if k>ky or r<ry or r>R :
            break

        A=A-r*B ; print("A{}={}".format(i,A))
        l=l-r ; print("l=", l)

    else:
        print("H matrisi Pozitif Tanımlı değildir.")

print("\nIterasyon sayısı N : ", i-1)

```

Yordam 3.6: Algoritma 3.6. ile Python yazılım dilinde altsınır(A, B, ry, ky, R) fonksiyonunun kodları.

Böylece genişletilmiş üst sınır u^e ve genişletilmiş alt sınır l^e değerleri ile birlikte, genişletilmiş Hurwitz kararlılık aralığı $\mathfrak{T}_L^{He} = [l^e, u^e]$ elde edilir.

Uyarı 3.4: Uyarı 3.1’ de Schur kararlılık için belirtilen durumun benzeri burada da mevcuttur. Gerçekten

$$A = \begin{pmatrix} -2 & 0 \\ 0 & -1 \end{pmatrix}$$

matrisini ele alırsak süreklilik teoremi ile belirlenen sınır 2 olarak bulunur. Bir adım daha ilerletilince

$$A = \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix}$$

matrisi elde edilir. Elde edilen bu matris kararsızdır. Bu nedenle, daha küçük adımlar atılmasını sağlayarak kontrollü bir genişleme imkanı sunan γ parametresi kullanılmaktadır. Yapılan çalışmalarda, $\gamma = 0.9$ değeri seçilerek analizler gerçekleştirilecektir.

3.2.1.2. $\mathfrak{I}_{\mathcal{L}}^{H^*e}$ aralığı için üst ve alt sınır algoritmaları ile python kodları

Algoritma 3.7 (Pratik Hurwitz Kararlılık için Üst Sınır (u^{*e})): Teorem 3.4 ile elde edilen aralığın üst sınırını genişletmek için aşağıda verilen algoritma kullanılmaktadır (Topcu ve Aydın, 2023b; Topcu, 2023).

1. Veri girişi;

$$A \in H_N, B \in M_N(\mathbb{C})$$

yeterince küçük r^* ,

yeterince küçük κ^*, R ,

$1 < \kappa^*$ problemin türüne göre kullanıcı tarafından belirlenen bir sayı.

2. Hesaplama; $\kappa(A)$.

3. Eğer $\kappa(A) > \kappa^*$ ise “ A matrisi κ^* – Hurwitz kararlı değildir” yaz ve algoritmayı bitir.

4. Hesaplama; $\|A\|, \|B\|$,

$$u^* = \frac{(\kappa^* - \kappa(A))\|A\|}{(1 + \kappa^*)\|B\|\kappa(A)}.$$

5. Veri yenileme; $k = 1, A_1 := A, r_1 := u^*, u_1 := r_1$.

6. Hesaplama;

$$A_{k+1} = A_k + r_k B, \|A_{k+1}\|, \kappa(A_{k+1}),$$

$$r_{k+1} = \frac{(\kappa^* - \kappa(A_{k+1}))\|A_{k+1}\|}{(1 + \kappa^*)\|B\|\kappa(A_{k+1})},$$

$$u_{k+1} = u_k + r_{k+1}$$

7. Eğer $r_k < r^*$ veya $r_k > R$ ise o halde algoritmayı bitir aksi takdirde $k := k + 1$ al ve 6. adıma git.

8. $M := k$ al e aralığın üst sınırını $u^{*e} = u_M$ olarak yaz.

Algoritma 3.7 için Python yordamını verelim.

Yordam 3.7 ($pr_üstsınır(A, B, \kappa^*)$): Verilen A katsayı matrisi ve B bozunum matrisi olmak üzere yeterince küçük κ^* pratik değerleri için, κ^* hassasiyetinde üst sınırı veren Python dilinde $pr_üstsınır(A, B, \kappa^*)$ fonksiyonunu aşağıda verelim.

```

def pr_üstsmr(A, B, ky):
import math ; import numpy as np; from scipy import linalg ; import sympy as sp ; from numpy import linalg as LA
print("Matrisin Boyutu:") ; n = int(input())
print("A Matrisinin Elemanları:")
A = [[0 for i in range(n)] for i in range(n)] ; elemanlar= list(map(float, input().split()));
A= np.array(elemanlar).reshape(n,n) ; print("\n A=", A)
print("B Matrisinin Elemanları:")
B= [[0 for i in range(n)] for i in range(n)] ; elemanlar= list(map(float, input().split()));
B= np.array(elemanlar).reshape(n,n) ; print("\n B=", B)
b=LA.norm(B, 2) ; print("||B||=",b)
print("r* değeri girin") ; ry=float(input()) ; print("r*=",ry)
print("R değeri girin") ; R=float(input()) ; print("R=",R)
print("κ* değeri girin") ; ky=float(input()) ; print("κ*=",ky)
i=0 ; u=0
while True:
    i=i+1; print("\n i=",i)
    Q =-np.eye(n) # Birim matris
    H= linalg.solve_continuous_lyapunov(A, Q) ; print("H= ", H)
    k= np.all(np.linalg.eigvals(H) > 0)

    if(k):
        print("H matrisi Pozitif tanımlıdır.")
        h=LA.norm(H, 2) ; a=LA.norm(A, 2) ; k=2*a*h ; print("κ(A)= ", k, ", ", "||A||=", a)
        r=((ky-k)*a)/((1+ky)*b*k) ; print("r=", r)
        if k>ky or r<ry or r>R :
            break

        A=A+r*B ; print("A{ }={ }".format(i,A))
        u=u+r ; print("u=", u)

    else:
        print("H matrisi Pozitif Tanımlı değildir.")

print("\nİterasyon sayısı N : ", i-1)

```

Yordam 3.7: Algoritma 3.7' nin Python yazılım dilinde kodlanması

Algoritma 3.8 (Pratik Hurwitz Kararlılık için Alt Sınır (l^{*e})): Teorem 3.4 ile elde edilen aralığın alt sınırını genişletmek için aşağıda verilen algoritma kullanılmaktadır (Topcu ve Aydın, 2023b; Topcu, 2023).

1. Veri girişi;

$$A \in H_N, B \in M_N(\mathbb{C})$$

yeterince küçük r^* ,

yeterince küçük κ^*, R ,

$1 < \kappa^*$ problemin türüne göre kullanıcı tarafından belirlenen bir sayı.

2. Hesaplama; $\kappa(A)$.

3. Eğer $\kappa(A) > \kappa^*$ ise “A matrisi κ^* – Hurwitz kararlı değildir” yaz ve algoritmayı bitir.

4. Hesaplama; $\|A\|, \|B\|$,

$$u^* = \frac{(\kappa^* - \kappa(A))\|A\|}{(1 + \kappa^*)\|B\|\kappa(A)}.$$

5. Veri yenileme; $k = 1, A_1 := A, r_1 := u^*, u_1 := r_1$.

6. Hesaplama;

$$A_{k+1} = A_k - r_k B, \|A_{k+1}\|, \kappa(A_{k+1}),$$

$$r_{k+1} = \frac{(\kappa^* - \kappa(A_{k+1}))\|A_{k+1}\|}{(1 + \kappa^*)\|B\|\kappa(A_{k+1})},$$

$$l_{k+1} = l_k - r_{k+1}$$

7. Eğer $r_k < r^*$ veya $r_k > R$ ise o halde algoritmayı bitir aksi takdirde $k := k + 1$ al ve 6. adıma git.
8. $M := k$ al ve aralığın alt sınırını $l^{*e} = l_M$ olarak yaz.

Algoritma 3.8 için Python yordamını verelim.

Yordam 3.8 ($pr_altsınır(A, B, \kappa^*)$): Verilen A katsayı matrisi ve B bozunum matrisi olmak üzere yeterince küçük κ^* pratik değerleri için bize κ^* hassasiyetinde alt sınırı veren Python dilinde $pr_altsınır(A, B, \kappa^*)$ fonksiyonunu aşağıda verelim.

```
def pr_üstsinir(A, B, ky):
import math; import numpy as np; from scipy import linalg; import sympy as sp; from numpy import linalg as LA
print("Matrisin Boyutu:") ; n = int(input())
print("A Matrisinin Elemanları:")
A = [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
A= np.array(elemanlar).reshape(n,n); print("\n A=", A)
print("B Matrisinin Elemanları:")
B= [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
B= np.array(elemanlar).reshape(n,n); print("\n B=", B)
b=LA.norm(B, 2); print("\|B\|=", b)
print("r* değeri girin"); ry=float(input()); print("r*=", ry)
print("R değeri girin"); R=float(input()); print("R=", R)
print("κ* değeri girin"); ky=float(input()); print("κ*=", ky)
i=0; l=0
while True:
    i=i+1; print("\n i=", i)
    Q = np.eye(n) # Birim matris
    H= linalg.solve_continuous_lyapunov(A, Q); print("H= ", H)
    k= np.all(np.linalg.eigvals(H) > 0)

    if(k):
        print("H matrisi Pozitif tanımlıdır.")
        h=LA.norm(H, 2); a=LA.norm(A, 2); k=2*a*h; print("κ(A)= ", k, ", ", "\|A\|=", a)
        r=((ky-k)*a)/((1+ky)*b*k); print("r=", r)
        if k>ky or r<ry or r>R:
            break

        A=A-r*B; print("A{ }={}".format(i,A))
        l=l-r; print("l=", l)

    else:
        print("H matrisi Pozitif Tanımlı değildir.")
    print("\n İterasyon sayısı N : ", i-1)
```

Yordam 3.8: Algoritma 3.8.' in Python yazılım dilinde kodlanması

Son olarak bulunan genişletilmiş üst sınır u^{*e} ve genişletilmiş alt sınır l^{*e} değerleri birlikte genişletilmiş κ^* – Hurwitz kararlılık aralığını $\mathfrak{L}_{\mathcal{L}}^{H^{*e}} = [l^{*e}, u^{*e}]$ oluşturur.

4. İKİ EKONOMİLİ MAKRO EKONOMİK MODELİN KARARLILIĞININ PARAMETRİK ANALİZİ

İki ekonomili modeller, genellikle iki ülke arasındaki ekonomik etkileşimlerin ve bu etkileşimlerin, ekonomilerin dinamik dengesi üzerindeki etkilerinin incelenmesi amacıyla kullanılır. Ticaret, sermaye akışları ve döviz kurları gibi uluslararası faktörlerin ekonomik denge üzerindeki etkileri analiz edilmektedir. Ekonomi politikalarının etkileri tahmin edilmekte ve olası senaryolar değerlendirilmektedir.

İki ekonomili makroekonomik modelde ise, iki farklı ekonominin birbirleriyle nasıl etkileşime girdiği ve bu etkileşimlerin ekonomik büyüme, denge ve kararlılık üzerindeki etkileri incelenmektedir. Bu modelde, bir ekonomideki değişkenlerin (tüketim, yatırım, ithalat, ihracat vb.) diğer ekonomi üzerindeki etkileri analiz edilmektedir. Uluslararası ticaretin ve ekonomik politikaların sonuçları anlamaya çalışılmaktadır.

Bu tür modeller, iki ekonomi arasındaki bağımlılıkların, denge durumlarının ve şoklara karşı tepki verme biçimlerinin incelenmesi amacıyla kullanılmaktadır. Örneğin, bir ülkenin ihracatının diğer ülkenin ithalatına nasıl etki ettiği veya bir ekonomideki yatırım artışının diğer ekonomiyi nasıl etkilediği analiz edilebilir (Blanchard ve Johnson, 2013).

Halkos ve arkadaşlarının 2012' de yayımlanan çalışmasında iki ekonomili bir model

$$\begin{aligned}
 C_{1,t} &= 0.55Y_{1,t-1} & C_{2,t} &= 0.7Y_{2,t-1} \\
 I_{1,t} &= 25 + 0.4Y_{1,t-1} & I_{2,t} &= 30 + 0.1Y_{2,t-1} \\
 M_{1,t} &= 12 + 0.3Y_{1,t-1} & M_{2,t} &= 20 + 0.1Y_{2,t-1} \\
 X_{1,t} &= M_{2,t} & X_{2,t} &= M_{1,t} \\
 Y_{1,t} &= C_{1,t} + I_{1,t} + X_{1,t} - M_{1,t} & Y_{2,t} &= C_{2,t} + I_{2,t} + X_{2,t} - M_{2,t}
 \end{aligned} \tag{4.1}$$

şeklinde yer almaktadır. Bu modelde;

- Tüketim (C): Bir ekonominin tüketim fonksiyonu, önceki dönem toplam çıktıya bağlıdır. Bu, tüketim harcamalarının ekonomi üzerindeki etkisini gösterir.
- Yatırım (I): Yatırım fonksiyonu, ekonominin büyümesi için yapılan harcamaları ifade eder ve toplam çıktı ile ilişkilidir.

- İthalat (M): İthalat fonksiyonu, bir ekonominin dışarıdan aldığı malları ve hizmetleri ifade eder ve toplam çıktıya bağlıdır.
- İhracat (X): İhracat fonksiyonu, bir ekonominin diğer ekonomiye sattığı malları ve hizmetleri ifade eder.
- Toplam Gelir (Y): Bir ekonominin toplam geliri, tüketim, yatırım, ihracat ve ithalat değişkenlerinin birleşiminden oluşur. Toplam gelir, ekonominin büyüklüğü ve gücünü yansıtan bir kavramdır. Yüksek bir toplam gelir, daha geniş iş imkanları, daha fazla gelir ve genel refah düzeyinde artış gibi birçok olumlu sonucun habercisidir. Ancak, bu noktada bunun sürdürülebilirliği de önemlidir. Bu nedenle kısa vadeli büyüme, uzun vadede dengesizliklere yol açabilir.

(4.1) modelinden de görüldüğü üzere bir ekonominin ihracatının diğer ekonominin ithalatı olduğu ve bu değişkenlerin nasıl birbirleriyle ilişkili olduğunu gösterilmektedir (Halkos ve Tsilika, 2012; Blanchard ve Johnson, 2013). Modeldeki verilenleri yerlerine koyarak düzenlersek;

$$\begin{aligned}
 Y_{1,t} &= C_{1,t} + I_{1,t} + X_{1,t} - M_{1,t} \\
 &= 0.55Y_{1,t-1} + 25 + 0.4Y_{1,t-1} + M_{2,t} - M_{1,t} \\
 &= Y_{1,t-1}(0.55 + 0.4 - 0.3) + 0.1Y_{2,t-1} + (25 + 20 - 12) \\
 &= 0.65Y_{1,t-1} + 0.1Y_{2,t-1} + 33
 \end{aligned}$$

olur. Buradan

$$Y_{1,t+1} = 0.65Y_{1,t} + 0.1Y_{2,t} + 33$$

elde edilir. Benzer şekilde,

$$\begin{aligned}
 Y_{2,t} &= C_{2,t} + I_{2,t} + X_{2,t} - M_{2,t} \\
 &= 0.7Y_{2,t-1} + 30 + 0.1Y_{2,t-1} + M_{1,t} - M_{2,t} \\
 &= 0.3Y_{1,t-1} + Y_{2,t-1}(0.7 + 0.1 - 0.1) + (30 + 12 - 20) \\
 &= 0.3Y_{1,t-1} + 0.7Y_{2,t-1} + 22
 \end{aligned}$$

buradan

$$Y_{2,t+1} = 0.3Y_{1,t} + 0.7Y_{2,t} + 22$$

bulunur. Böylece

$$\begin{cases} Y_{1,t+1} = 0.65Y_{1,t} + 0.1Y_{2,t} + 33 \\ Y_{2,t+1} = 0.3Y_{1,t} + 0.7Y_{2,t} + 22 \end{cases} \quad (4.2)$$

birinci mertebeden homojen olmayan fark denklem sistemine ulaşılır.

4.1. Makro Ekonomik Modelin Parametrik İfadelenirilmesi

Modeldeki temel deęiřkenlerin her birinin katsayılarını,

$$\begin{aligned}
 C_{1,t} &= 0.55Y_{1,t-1} = c_{10} + \alpha_1 Y_{1,t-1} & C_{2,t} &= 0.7Y_{2,t-1} = c_{20} + \beta_1 Y_{2,t-1} \\
 I_{1,t} &= 25 + 0.4Y_{1,t-1} = I_{10} + \alpha_2 Y_{1,t-1} & I_{2,t} &= 30 + 0.1Y_{2,t-1} = I_{20} + \beta_2 Y_{2,t-1} \\
 M_{1,t} &= 12 + 0.3Y_{1,t-1} = M_{10} + \alpha_3 Y_{1,t-1} & M_{2,t} &= 20 + 0.1Y_{2,t-1} = M_{20} + \beta_3 Y_{2,t-1} \\
 X_{1,t} &= M_{2,t} & X_{2,t} &= M_{1,t} \\
 Y_{1,t} &= C_{1,t} + I_{1,t} + X_{1,t} - M_{1,t} & Y_{2,t} &= C_{2,t} + I_{2,t} + X_{2,t} - M_{2,t}
 \end{aligned}$$

řeklinde parametrik olarak ifade eder ve dzenlersek;

$$\begin{cases}
 Y_{1,t+1} = (\alpha_1 + \alpha_2 - \alpha_3)Y_{1,t} + \beta_3 Y_{2,t} + (c_{10} + I_{10} + M_{20} - M_{10}) \\
 Y_{2,t+1} = \alpha_3 Y_{1,t} + (\beta_1 + \beta_2 - \beta_3)Y_{2,t} + (c_{20} + I_{20} + M_{10} - M_{20})
 \end{cases} \quad (4.3)$$

elde edilir. Burada,

- c_{10}, c_{20} ; birinci ve ikinci ekonominin gelire baęlı olmayan tüketimleri (otonom tüketimler),
- α_1, β_1 ; birinci ve ikinci ekonominin marjinal tüketim eęimidir. $\alpha_1, \beta_1 \in (0,1)$ aralıęındadır ve genellikle 0.5-0.9 deęerleri arasındadır. Daha yüksek bir α_1 veya β_1 deęerleri ekonominin daha tüketim odaklı olduęunu gösterir. *
- I_{10}, I_{20} ; birinci ve ikinci ekonominin gelire baęlı olmayan yatırımları (otonom yatırımlar), *
- α_2, β_2 ; birinci ve ikinci ekonominin marjinal yatırım eęimidir. Yatırımların gelire duyarlılık katsayıları gelirdeki artışın yatırımlara nasıl yansıdığını gösterir. $\alpha_2, \beta_2 \geq 0$ deęerleri 0.1-0.5 arasındadır. Daha yüksek bir α_2 deęeri, birinci ekonomide yatırımların gelir artışına daha duyarlı olduęunu gösterir. *
- M_{10}, M_{20} ; birinci ve ikinci ekonominin gelire baęlı olmayan ithalatları (otonom ithalatlar), *
- α_3, β_3 ; birinci ve ikinci ekonominin marjinal ithalat eęimidir. Gelirdeki artışın ne kadarının ithalata harcandığını gösterir. $\alpha_3, \beta_3 \in [0,1)$ ve genellikle 0.1-0.5 deęerleri

* Bu veriler basılı kaynaklardan teyit edilememiř yapay zekanın Grok 3 platformu kullanılarak bulunmuřtur.

arasındadır. Daha yüksek bir α_3 değeri, birinci ekonominin daha dışa bağımlı olduğunu ve gelir artışının daha büyük bir kısmının ithalata gittiğini gösterir. *

- $\alpha = (\alpha_1 + \alpha_2 - \alpha_3)$ terimi birinci ekonominin kendi gelirine etkisidir. Genellikle gelişmiş ekonomilerde $\alpha \in (0.6, 0.8)$ iken, gelişmekte olanlarda $\alpha \in (0.4, 0.6)$ gibi daha düşük olmalıdır. *

- α_3 terimi birinci ekonominin ikinci ekonomiye etkisi (birinci ekonominin ithalat eğilimi). İthalat eğilimi, ekonominin büyüklüğüne ve açıklığına bağlı olarak değişir. Büyük ekonomilerde α_3 daha küçük, küçük ekonomilerde ise daha büyük olabilir. Benzer şekilde, küçük ekonomilerde $\alpha_3 \in (0.1, 0.3)$, büyük ekonomilerde $\alpha_3 \in (0, 0.1)$ dir. *

- $\beta = (\beta_1 + \beta_2 - \beta_3)$ terimi ikinci ekonominin kendi gelirine etkisidir. Birinci ekonomideki α değerine benzer şekilde, bu katsayı ikinci ekonominin iç dinamiklerini (tüketim, yatırım, ithalat) yansıtır. $\beta < 1$ olmalıdır çünkü sızıntılar (tasarruf, ithalat) vardır. Daha yüksek marjinal tüketim eğilimine sahip ekonomilerde β değeri daha büyük olabilir. Gelişmiş ekonomilerde $\beta \in (0.6, 0.8)$, gelişmekte olanlarda $\beta \in (0.4, 0.6)$ aralığındadır. *

- β_3 terimi ikinci ekonominin birinci ekonomiye etkisi (ikinci ekonominin ithalat eğilimi). İthalat eğilimi genellikle düşüktür, çünkü bir ekonominin geliri büyük ölçüde kendi iç dinamiklerine bağlıdır. Daha büyük değerler (örneğin, 0.3), dış ticarete çok bağımlı ekonomilerde görülebilir (örneğin, küçük açık ekonomiler). Küçük açık ekonomilerde $\beta_3 \in (0.1, 0.3)$, büyük ekonomilerde $\beta_3 \in (0, 0.1)$ dir. *

Şimdi (4.3) sisteminde $Y_{1t} = x_n, Y_{2t} = y_n, c_{10} = a_0, c_{20} = b_0$ ve $I_{10} = u_0, I_{20} = v_0, M_{10} = w_0, M_{20} = z_0$ alınıp değişken değiştirme yapılırsa parametrik katsayılı

$$\begin{cases} x_{n+1} = (\alpha_1 + \alpha_2 - \alpha_3)x_n + \beta_3 y_n + (a_0 + u_0 + z_0 - w_0) \\ y_{n+1} = \alpha_3 x_n + (\beta_1 + \beta_2 - \beta_3)y_n + (b_0 + v_0 + w_0 - z_0) \end{cases} \quad (4.4)$$

sistemine ulaşılır.

* Bu veriler basılı kaynaklardan teyit edilememiş yapay zekanın Grok 3 platformu kullanılarak bulunmuştur.

$C_{1,t} = c_{10} + \alpha_1 Y_{1,t-1};$ $I_{1,t} = I_{10} + \alpha_2 Y_{1,t-1};$ $M_{1,t} = M_{10} + \alpha_3 Y_{1,t-1};$ $X_{1,t} = M_{2,t};$	$C_{2,t} = c_{20} + \beta_1 Y_{2,t-1};$ $I_{2,t} = I_{20} + \beta_2 Y_{2,t-1};$ $M_{2,t} = M_{20} + \beta_3 Y_{2,t-1};$ $X_{2,t} = M_{1,t},$
$Y_{1,t} = C_{1,t} + I_{1,t} + X_{1,t} - M_{1,t}$	$Y_{2,t} = C_{2,t} + I_{2,t} + X_{2,t} - M_{2,t}$
$\begin{cases} Y_{1,t+1} = (\alpha_1 + \alpha_2 - \alpha_3)Y_{1,t} + \beta_3 Y_{2,t} + (I_{10} + M_{20} - M_{10}) \\ Y_{2,t+1} = \alpha_3 Y_{1,t} + (\beta_1 + \beta_2 - \beta_3)Y_{2,t} + (I_{20} + M_{10} - M_{20}) \end{cases}$	
ya da $Y_{1,t} = x_n; Y_{2,t} = y_n; c_{10} = a_0; c_{20} = b_0; I_{10} = u_0; I_{20} = v_0; M_{10} = w_0; M_{20} = z_0$ ile dönüştürülmüş sistem $(\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3, a_0, b_0, u_0, v_0, w_0, z_0 - \text{sabitler})$	
$\begin{cases} x_{n+1} = (\alpha_1 + \alpha_2 - \alpha_3)x_n + \beta_3 y_n + (a_0 + u_0 + z_0 - w_0) \\ y_{n+1} = \alpha_3 x_n + (\beta_1 + \beta_2 - \beta_3)y_n + (b_0 + v_0 + w_0 - z_0) \end{cases}$	

(4.4) sisteminde;

$$\alpha_1 = 0.55; \alpha_2 = 0.4; \alpha_3 = 0.3; \beta_1 = 0.7; \beta_2 = 0.1; \beta_3 = 0.1,$$

$$a_0 = 0; b_0 = 0; u_0 = 25; v_0 = 30; w_0 = 12; z_0 = 20,$$

olarak alınırsa, (4.2) fark denklem sistemi,

$$\begin{cases} x_{n+1} = 0.65x_n + 0.1y_n + 33 \\ y_{n+1} = 0.3x_n + 0.7y_n + 22 \end{cases} \quad (4.5)$$

olup, (4.2) ile (4.5) aynı fark-dinamik sistemi elde edilmiş olur.

4.2. Makro Ekonomik Modelin Schur Kararlılığı

Bu bölümde (4.4) sisteminin Schur kararlılığı incelenecektir.

4.2.1. Tek parametrelili makroekonomik model

(4.4) fark-dinamik sistemde x_n değişkeninin katsayısını $(\alpha_1 + \alpha_2 - \alpha_3) = \alpha$ şeklinde parametrik ve $\alpha_3 = 0.3; \beta_1 = 0.7; \beta_2 = 0.1; \beta_3 = 0.1$ olarak alalım. Bu durumda (4.4) dinamik sistemi,

$$\begin{cases} x_{n+1} = \alpha x_n + 0.1y_n + 33 \\ y_{n+1} = 0.3x_n + 0.7y_n + 22 \end{cases} \quad (4.6)$$

şekline dönüşür. Şimdi (4.6) fark-dinamik sisteminin kararlılığını α belirsizine göre analiz edelim. Modelin denge ve kararlılık analizi, ekonomilerin uzun vadeli davranışlarını ve kararlılık durumlarını incelemek için önemlidir. Ekonomik değişkenlerin değerlerinin zaman içinde nasıl değiştiği ve dengeye nasıl ulaştığı analiz edilir. Bu analizler, ekonomik politikaların ve dış şokların ekonomiler üzerindeki etkilerini anlamak için kritik öneme sahiptir.

(4.6) sisteminin denge noktaları için fonksiyonlar

$$\begin{cases} f_1(x_n, y_n) = \alpha x_n + 0.1y_n + 33 \\ f_2(x_n, y_n) = 0.3x_n + 0.7y_n + 22 \end{cases}$$

olmak üzere $f_1(x^*, y^*) = x^*$ ve $f_2(x^*, y^*) = y^*$ dan denge noktaları

$$x^* = -\frac{1210}{30\alpha - 27}, y^* = \frac{2200\alpha - 3190}{30\alpha - 27}$$

şeklinde bulunur.

Uyarı: Matematikte, özellikle uygulamalı alanlarda (ekonomi, fizik, biyolojide vb.), pozitif denge noktaları genellikle daha büyük bir öneme sahiptir. Bunun temel nedeni, bu alanlardaki gerçek dünya problemlerinin genellikle yalnızca pozitif değerlerle anlam taşımamasıdır. Örneğin, ekonomi modellerinde fiyat, talep veya üretim miktarı negatif olamaz. Benzer şekilde, biyolojide nüfus büyüklüğü negatif bir değer alamaz. Dolayısıyla, bu tür sistemlerde negatif denge noktaları pratik açıdan geçerli değildir. Matematiksel açıdan negatif denge noktaları teorik olarak mevcut olsa da gerçek dünya problemleri açısından bir anlam taşımayabilir.

(4.6) sisteminin katsayı matrisi

$$A(\alpha) = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & 0.7 \end{pmatrix}$$

şeklinde olup Teorem 2.1 ve Teorem 2.3' ten

$$A(\alpha) = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & 0.7 \end{pmatrix} = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0.7 \end{pmatrix} + \alpha \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = A_1 + \alpha E_{11} \quad (4.7)$$

şeklinde düzenlenirse (2.17) de verilen Lyapunov fark matris denkleminde

$$H = \begin{pmatrix} 1.02276 & 0.16428 \\ 0.16428 & 2.27656 \end{pmatrix}$$

matrisi bulunur. $H = H^* > 0$ olduğundan A_1 matrisi Teorem 2.1' e göre Schur kararlıdır.

Not: A_1 matrisinin Schur kararlı olduğunu görmek için Lyapunov matris denkleminin çözümüne ihtiyaç yoktur. Normu 1 den küçük ($\|A_1\| < 1$) olan matrislerin Schur kararlı olduğu bilinmektedir (Akın ve Bulgak, 1998). Dolayısıyla Schur kararlı olduğu rahatça söylenir. H matrisinin hesaplanması kararlılığın kalitesi hakkında yorum yapma imkanı sağlar. Ayrıca Süreklilik teoremi yardımıyla sistemin Schur kararlılığı için güvenli bir aralık bulmak amacıyla H matrisine ($\omega(A_1)'$ e) ihtiyaç vardır.

(4.7) denkleminde

$$\|A_1\| = 0.767, \quad \omega(A_1) = 2.297, \quad \Delta = \alpha E_{11}, \quad \|\Delta\| = |\alpha|$$

değerleri elde edilir. Elde edilen bu değerler Teorem 2.3 de verilen

$$\|B\| < \sqrt{\|A\|^2 + \frac{1}{\omega(A)}} - \|A\|$$

eşitsizliğinde yerlerine konulup hesaplanırsa

$$|\alpha| < 0.2446$$

eşitsizliğine ulaşılır. Yani $\alpha \in (-0.2446, 0.2446)$ elde edilir. Teorem 2.3 Schur kararlılığın sağlandığı güvenli bir aralık vermektedir. Ancak bu aralık dışında Schur kararlı sistemler olabileceği de önceki bölümde ifade edilmiştir. Şimdi bulunan bu aralığı kararlılık özellikleri korunacak şekilde genişletelim.

4.2.1.1. Schur kararlı aralığın genişletilmesi

(4.6) sistemi için Teorem 2.3 ile bulunan ve Schur kararlı olduğu bilinen aralığı $\bar{u}stsinır(A, B, ry)$, $\bar{a}ltsinır(A, B, ry)$ fonksiyonları yardımı ile genişletelim. Yani α parametresi üzerinde değişiklikler yaparak sistemin kararlılığı korunacak şekilde bir kararlılık aralığı bulalım. Ve α parametresi için sistemin kararlılığını bozmayacak şekilde bir kararlılık aralığı bulalım.

Şimdi (4.7) matrisine göre (4.6) sistemini farklı r^* değerleri alarak $\bar{u}stsinır(A, B, ry)$ fonksiyonu (Yordam 3.1) kullanılırsa Tablo 4.1 deki $\bar{a}ltsinır(A, B, ry)$ fonksiyonu (Yordam 3.2) kullanılırsa Tablo 4.2 deki veriler hesaplanır.

A	B	r^*	u	u^e	M
$\begin{pmatrix} 0 & 0.1 \\ 0.3 & 0.7 \end{pmatrix}$	E_{11}	0.1	0.2446	0.733	4
		0.01		0.8906	8
		0.001		0.8989	11
		0.0001		0.8998	14

Tablo 4.1: Farklı r^* - pratiklik parametreleri ile durulan adım ve genişletilmiş üst sınır değerleri.

A	B	r^*	l	l^e	M
$\begin{pmatrix} 0 & 0.1 \\ 0.3 & 0.7 \end{pmatrix}$	E_{11}	0.1	-0.2446	-0.89245	4
		0.01		-0.98004	6
		0.001		-0.98199	7
		0.0001		-0.98229	8

Tablo 4.2: Farklı r^* - pratiklik parametreleri ile durulan adım ve genişletilmiş alt sınır değerleri.

Buna göre genişletilmiş üst sınır $r^* = 0.0001$ pratik adım genişliği kullanılarak 14 adımda $u^e = 0.8998$ (Tablo 4.1), genişletilmiş alt sınır 8 adımda $l^e = -0.98229$ (Tablo 4.2) elde edilir. Sonuç olarak $\text{üstsinir}(A, B, ry)$ fonksiyonu (Yordam 3.1) ve $\text{altsinir}(A, B, ry)$ fonksiyonu (Yordam 3.2) kullanılarak

$$\mathfrak{T}^e = [l^e, u^e] = [-0.98229, 0.8998]$$

aralığı elde edilir. α bu aralıktan seçildiği sürece (4.6) sistemi Schur kararlı bir sistemdir. Şimdi elde ettiğimiz bu sonucu deneyelim.

$\alpha = 0.65$ için;

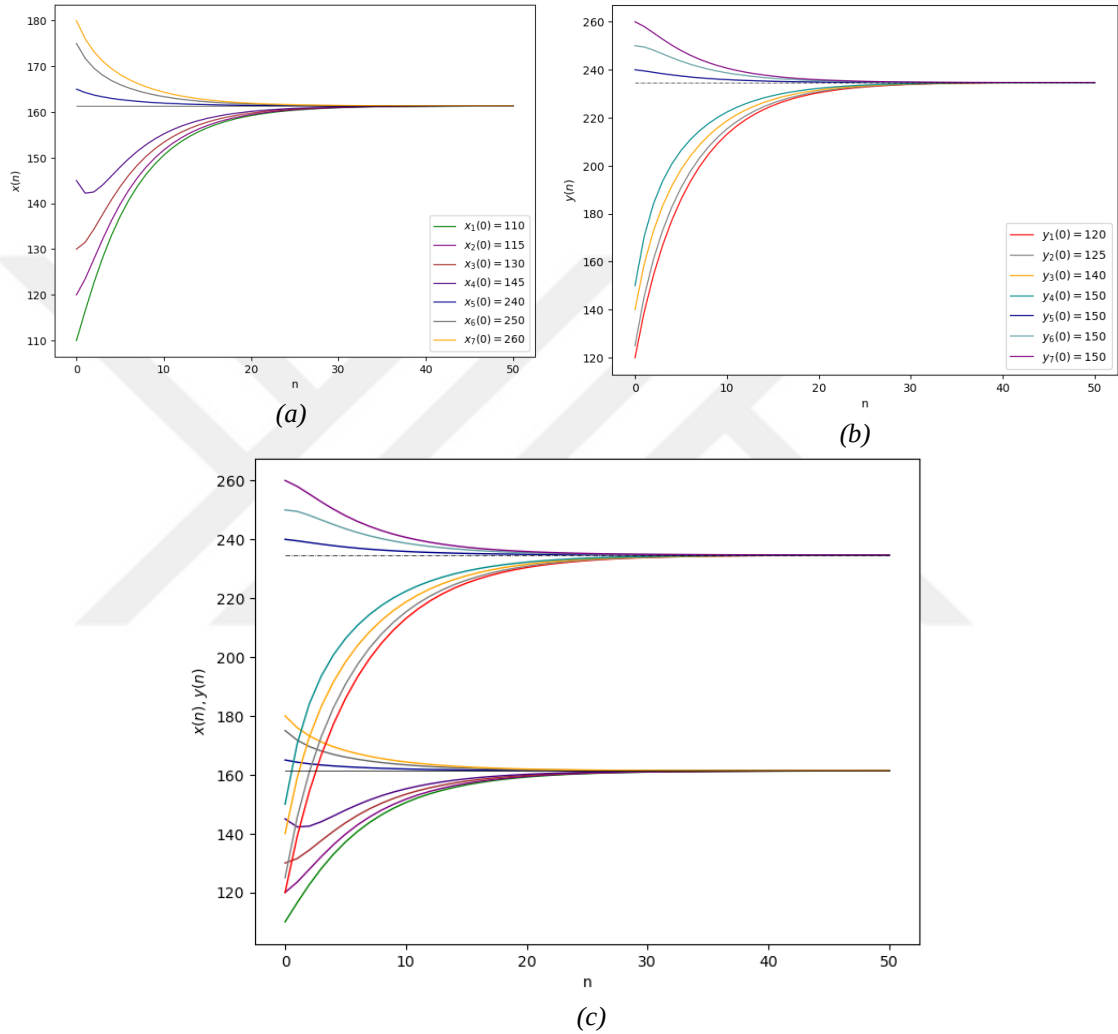
- Halkos ve arkadaşlarının 2012' de yayımlanan 'Stability analysis in economic dynamics: A computational approach' adlı çalışmasında, $\alpha = 0.65$ değeri için farklı yollarla kararlılık incelenmiştir. Bu nedenle, bu değer üzerinden bir değerlendirme yapılması uygun olacaktır. (4.6) sisteminde $\alpha = 0.65$ olarak alırsak sistemin denge noktası $(x^*, y^*) = (161.33, 234.66)$ ve katsayı matrisi

$$A_1 = \begin{pmatrix} 0.65 & 0.1 \\ 0.3 & 0.7 \end{pmatrix}$$

ve $A_1^* H A_1 - H + I = 0$ LMD'den

$$H = \begin{pmatrix} 3.0207 & 1.356 \\ 1.356 & 2.392 \end{pmatrix}$$

pozitif tanımlı bir H matrisi elde edilir dolayısıyla $A_1 \in S_N$ dir. Yani A_1 matrisi, buna bağlı olarakta dinamik sistem Schur kararlıdır. Aşağıda verilen grafiklerde sonuçlarla uyumlu olduğu görülmektedir (Şekil 4.1).



Şekil 4.1 : $\alpha = 0.65$ parametresine göre farklı başlangıç noktaları için (a) x_n yörüngelerinin, (b) y_n yörüngelerinin ve (c) x_n, y_n yörüngelerinin birlikte hareketidir.

$\alpha = 0.95$ için;

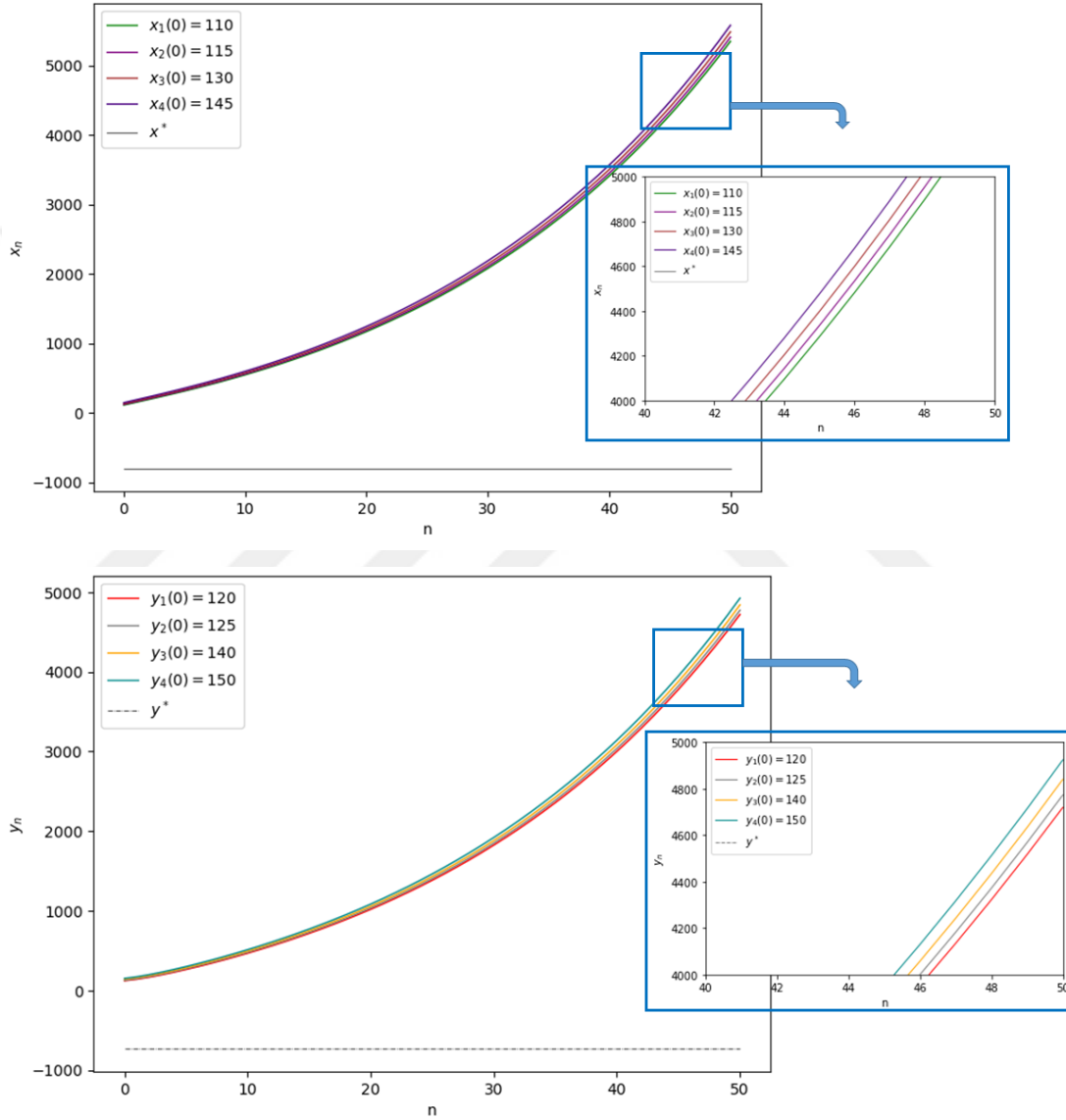
- (4.6) sisteminde $\alpha = 0.95$ olarak alırsak sistemin denge noktası $(x^*, y^*) = (-806.66, -733.3)$ ve katsayı matrisi

$$A_1 = \begin{pmatrix} 0.95 & 0.1 \\ 0.3 & 0.7 \end{pmatrix}$$

şeklindedir. $A_1^*HA_1 - H + I = 0; H = H^* > 0$ LMD'den

$$H = \begin{pmatrix} -15.195 & -4.424 \\ -4.424 & 0.448 \end{pmatrix}$$

H matrisi simetrik bir matristir fakat pozitif tanımlı değildir. Dolayısıyla A_1 matrisi Schur kararlı değildir ($A_1 \notin S_N$). Aşağıda verilen grafiklerde bu sonuçlarla uyumlu olduğu görülmektedir (Şekil 4.2).



Şekil 4.2: $\alpha = 0.95$ parametresine göre farklı başlangıç noktaları için yörüngelerin hareketi.

4.2.1.2. Genişletilmiş aralık için etkinlik analizi (kalite kontrolü)

Bu kısımda bir önceki bölümünde bulunan Schur kararlı aralığın genişletilmesi çalışılacaktır.

4.2.1.2.1. Üst sınır için etkinlik analizi

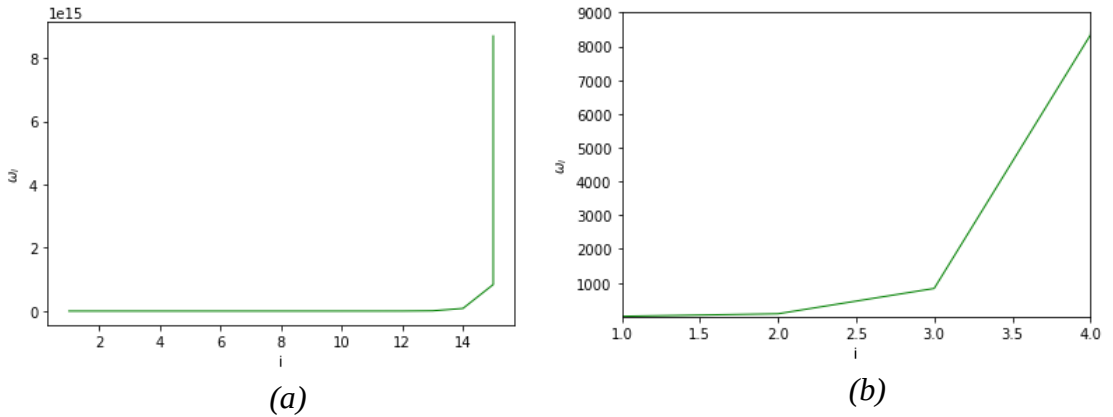
$r^* = 0.001$ değerine karşılık $\text{üstsinir}(A, B, ry)$, $\text{altsinir}(A, B, ry)$ fonksiyonları yardımıyla elde edilen genişletilmiş $[l^e, u^e] = [-0.98229, 0.8998]$ aralığının etkinlik analizi yapalım.

$$A(\alpha_i) = \begin{pmatrix} \alpha_i & 0.1 \\ 0.3 & 0.7 \end{pmatrix}; \quad \alpha_i = \frac{8}{10} + \sum_{j=2}^i \frac{9}{10^j}, \quad (i = 1, 2, \dots)$$

olmak üzere yapılan hesaplamalar ve bulunan ω_i değerleri Tablo 4.3'te verilmiştir. Mesela tabloda 3. adımda $\alpha \in (0.899, 0.9)$ için sınıra yaklaştıkça, ω değerinin arttığı ve dolayısıyla kalitenin düştüğü görülmektedir. Diğer taraftan $\alpha_3 = 0.899$ da sistem Schur kararlı iken $\alpha_\infty = 0.9$ olduğunda $|A(\alpha_3) - A(\alpha_\infty)| = 10^{-3}$ olurken $\omega(A(\alpha_3)) < \infty$ olmasına rağmen $\omega(A(\alpha_\infty)) = \infty$ olmaktadır. α değerindeki çok küçük değişimlere karşılık problemin yapısı tamamen değişebilmektedir. Yani α_3 için (4.6) sistemi Schur kararlı iken $\alpha_\infty = 0.9$ için Schur kararsız olmaktadır. Aralığın uç noktasına yaklaştıkça sistemin Schur kararlılığının kalitesini belirleyen parametre büyümekte, yani kalitesi azalmakta, dolayısıyla sistemin dış etkilere karşı dayanıklılığı azalmaktadır.

i	1	2	3	4	...	$i \rightarrow \infty$
α_i	0.8	0.89	0.899	0.8999	...	0.9
ω_i	8.880	83.740	833.725	8333.724	...	∞
$\lim_{i \rightarrow \infty} \alpha_i = 0.9$						

Tablo 4.3: $\omega(A(\alpha_i)) = \omega_i$, genişletilen üst sınır için etkinlik analizi.



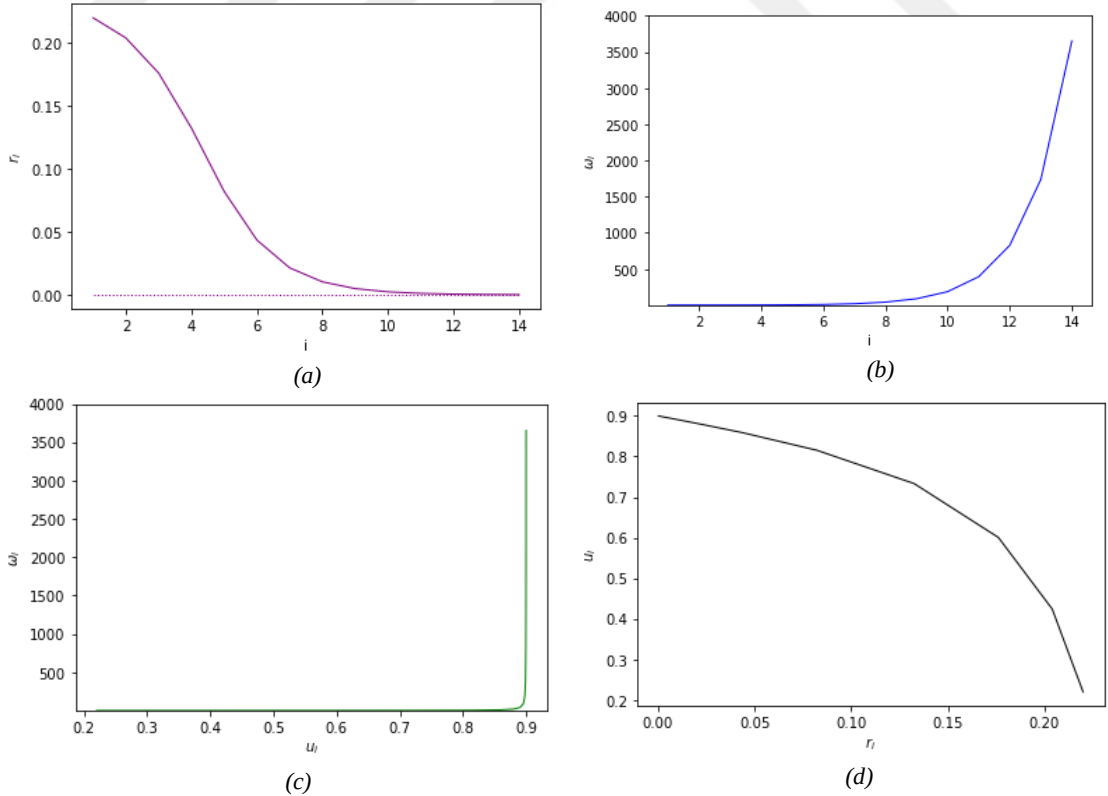
Şekil 4.3: (a) Tablo 4.3 etkinlik analizinin grafiği.

(b), (a) grafiğinin $[1, 4] \times [1, 9000]$ dikdörtgeninde gösterimi

Yordam 3.1 kullanılarak oluşturulan Tablo 4.4' den de görüldüğü üzere her adımda sınır genişletilmiş ve 0.899 değerinden sonra yavaş hareket etmiştir. ω değeri her adımda daha çok büyümüş kararlılığın kalitesi düşmüştür. Yani kararlılık 0.899 civarında bozulmaya daha yakındır. Şekil 4.5 (a)' dan anlaşılabacağı üzere i değerleri arttıkça r değerleri azalmaktadır. Benzer şekilde Şekil 4.5 (b)' de i değerleri arttıkça ω_i değeri artmaktadır. Şekil 4.5 (c)' de u_i değerleri sınıra yaklaştıkça ω_i değeri artmaktadır.

i	r_i	u_i^e	ω_i
1	0.22018762769410913	0.22018762769410913	2.2977383911475346
2	0.20418533126117902	0.42437295895528815	2.4580142971401084
3	0.17629094546376273	0.6006639044190509	2.7994103618139086
4	0.13264696532398415	0.733310869743035	3.6176893618141737
$\vdots$			
12	0.00052871931407036	0.8995203442631599	826.8043577701324
13	0.000251504394536	0.8997718486576969	1737.7485762371373
14	0.000119631318840	0.8998914799765378	3652.9372625032584
15	0.00005690291	--	--

Tablo 4.4: Üst sınır genişletilirken r, u, ω parametrelerinin değişimi.



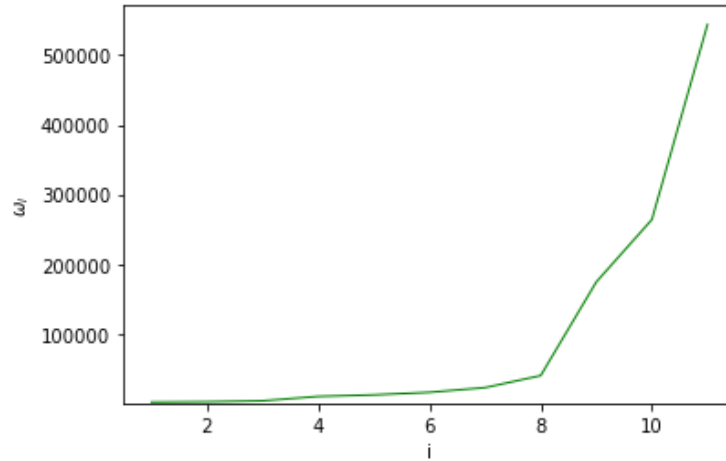
Şekil 4.4: Tablo 4.4' deki hesaplamalar sonucu (a) r_i ' nin i 'e göre grafiği, (b) ω_i ' nin i 'e göre grafiği, (c) ω_i ' in u_i ' e göre grafiği, (d) u_i 'in r_i ' e göre grafiği

4.2.1.2.2. Alt sınır için etkinlik analizi

Alt sınır $l^e = -0.982$ olarak alındığında yapılan hesaplamalar ve bulunan ω_i değerleri Tablo 4.5' te verilmiştir. Benzer şekilde, sınıra yaklaştıkça ω değerinin arttığı ve dolayısıyla kalitenin düştüğü görülmüştür. Burada -0.9824 , -0.98236 , -0.982354 değerleri kararsız değerlerdir dolayısıyla Tablo 4.5' te ki gibi hareket etmektedir.

i	α_i	ω_i
1	-0.982	1451.055446
2	-0.9821	2024.619193
$\vdots$	$\vdots$	$\vdots$
4	-0.9823	9672.136054
5	-0.98231	11924.486920
$\vdots$	$\vdots$	$\vdots$
9	-0.98235	174093.749435
10	-0.982351	263778.265835
11	-0.982352	544042.379562
$\vdots$	$\vdots$	$\vdots$

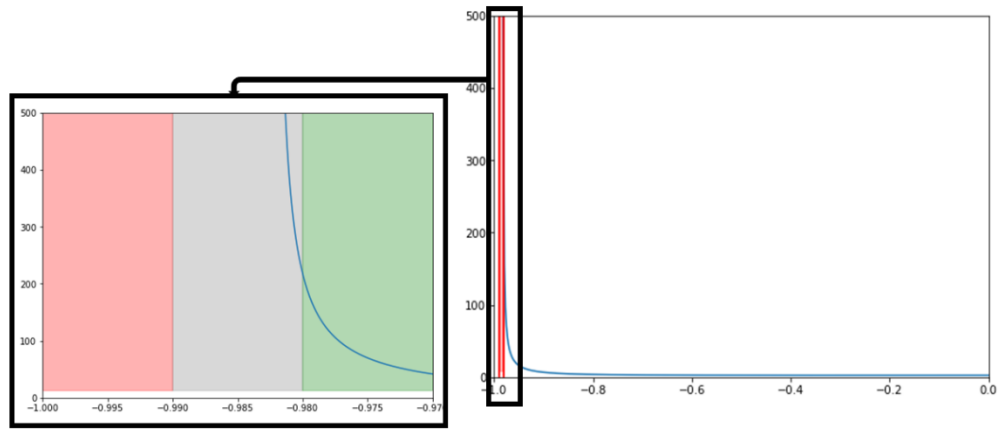
Tablo 4.5: $\omega(A(\alpha_i)) = \omega_i$, genişletilen alt sınır için etkinlik analizi.



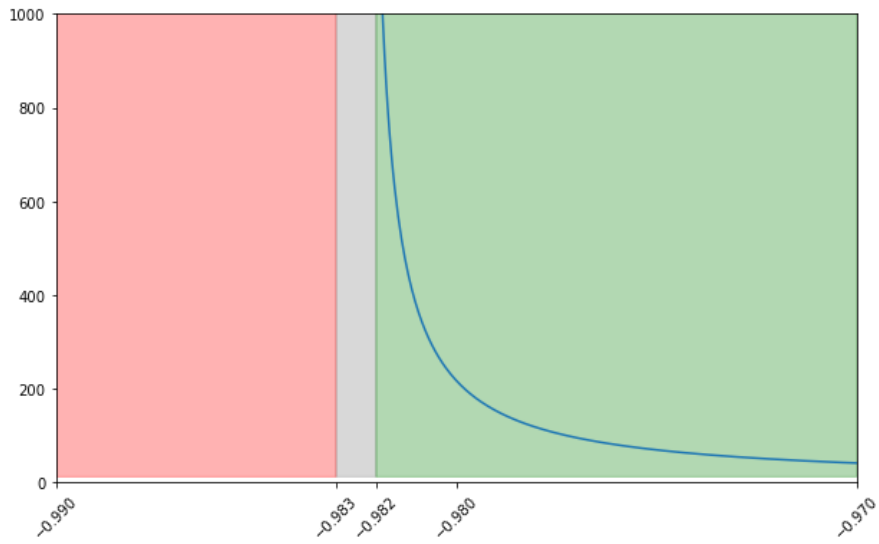
Şekil 4.5: Tablo 4.5 etkinlik analizinin grafiği.

Tablo 4.5' i daha iyi anlamak için aşağıda verilen Şekil 4.6' yı incelersek, elde edilen aralığın uzunluğu çok küçük olmasına rağmen bu aralıkta hem Schur kararlı, hem de Schur kararsız sistemler vardır. Şekil 4.6' da kırmızı bölgeler kararsız, gri bölgeler hem kararlı hem kararsız, yeşil bölgeler ise kararlı sistemleri içeren alanları temsil

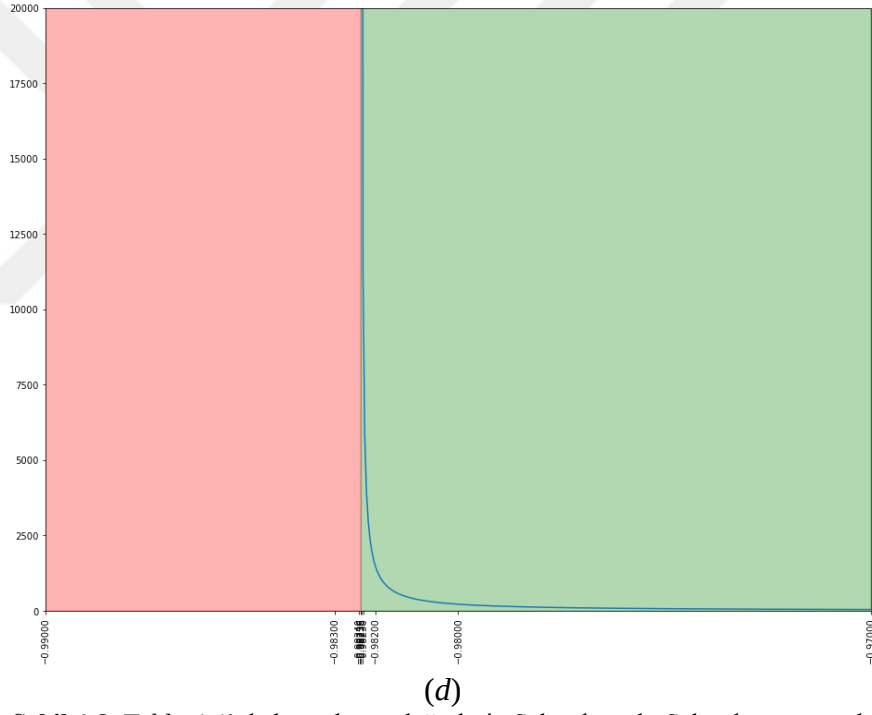
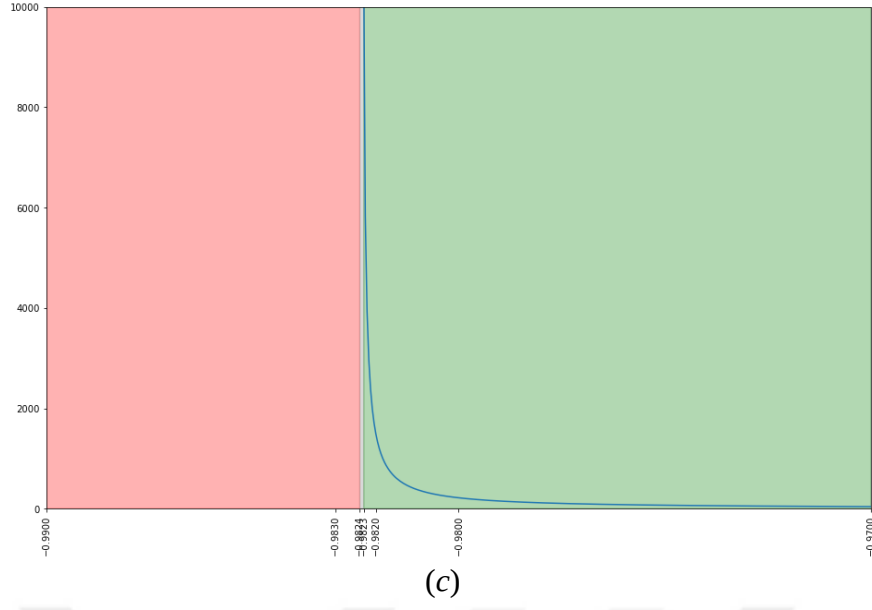
etmektedir. Şekil 4.6 (a)'da $(-0.99, -0.98)$ aralığı çok küçük olmasına rağmen daha hassas incelemeler yapıldığında hem Schur kararlı, hem de Schur kararlı olmayan sistemler vardır Şekil 4.6 (b)' ye bakıldığında $(-0.99, -0.98)$ aralığının bir kısmının kırmızı bölgeye katıldığı bir kısmının da yeşil bölgeye katıldığı görülmektedir. Gri bölgenin $(-0.983, -0.982)$ aralığını gösterdiğini ve benzer mantıkla yine hem Schur kararlı hem Schur kararsız sistemler olduğu görülmektedir. Şekil 4.6 (c) ve Şekil 4.6 (d)' de $(-0.9824, -0.9823)$ ve $(-0.98236, -0.98235)$ aralıkları içinde benzer mantık geçerlidir. Bu fikri pekiştirmek açısından Tablo 4.4' te hesaplanan aralıklar ile oluşturulan Şekil 4.7'ye bakılabilir.



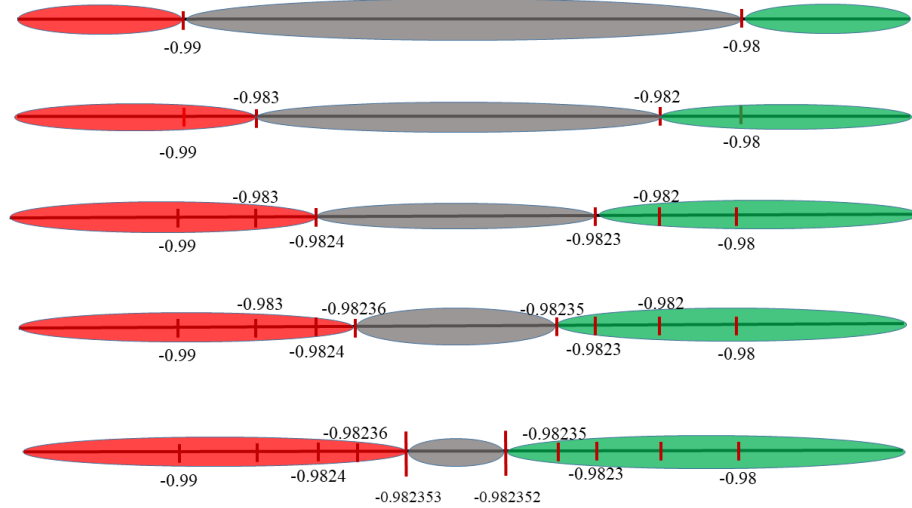
(a)



(b)



Şekil 4.6: Tablo 4.4’ de hesaplanan değerlerin Schur kararlı, Schur kararsız ve belirsiz bölgeleri gösteren grafikleri.

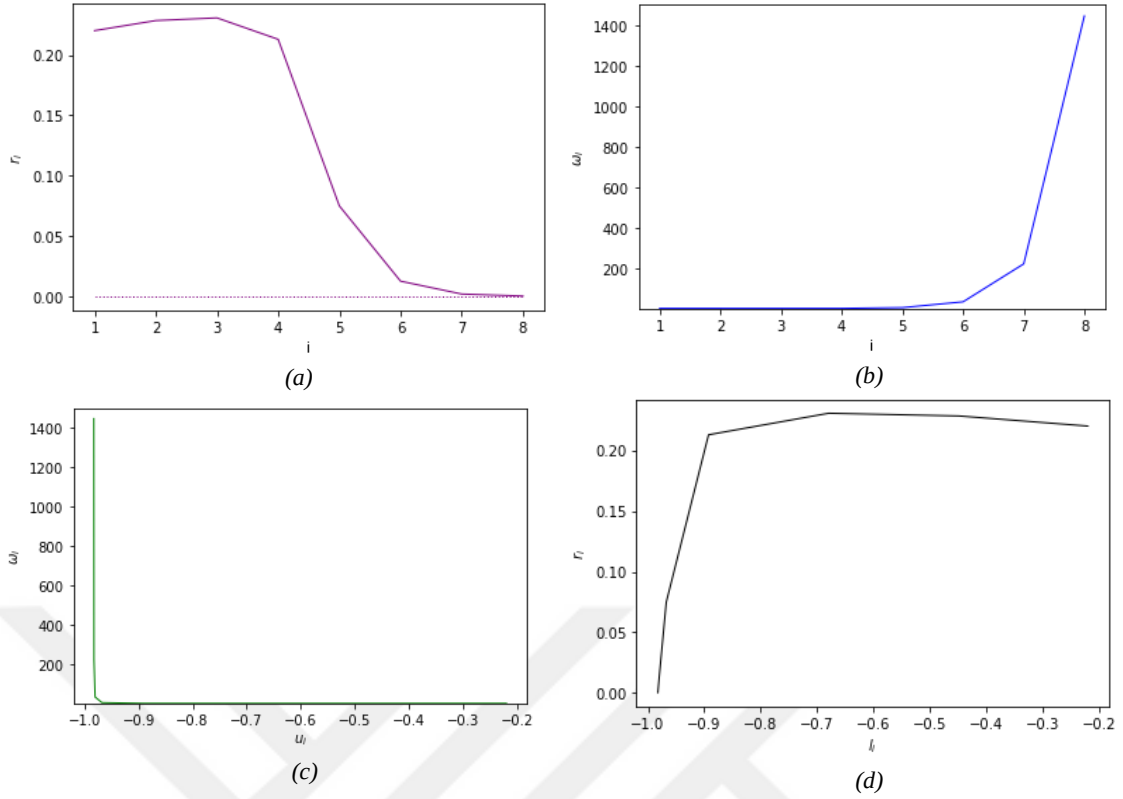


Şekil 4.7: Tablo 4.4’ de hesaplanan değerlerin Schur kararlı, Schur kararsız ve belirsiz bölgeleri gösteren sayı doğrusu.

Algoritma 3.1 ve $\text{üstsinir}(A, B, ry)$ kullanılarak oluşturulan Tablo 4.6’ dan da görüldüğü üzere her adımda sınır genişletilmiş ve -0.98235 değerinden sonra yavaş hareket etmiştir. ω değeri her adımda daha çok büyümüş kararlılığın kalitesi düşmüştür. Yani kararlılık -0.982 değeri civarında bozulmaya daha yakındır. Şekil 4.8 (a)’ dan anlaşılacağı üzere n değerleri arttıkça r_i değerleri azalmakta, Şekil 4.8 (b)’ den i değerleri arttıkça ω_i değeri artmakta ve Şekil 4.8 (c)’ den u_i değerleri sınıra yaklaştıkça ω_i değeri artmaktadır.

i	r_i	l_i^e	ω_i
1	0.22018762769410913	-0.22018762769410913	2.2977383911475346
2	0.22851796474299377	-0.4487055924371029	2.2162035856089872
3	0.2307562050211534	-0.6794617974582563	2.1741316749909867
4	0.21299132419683425	-0.8924531216550906	2.254822836585305
5	0.07496989626904062	-0.9674230179241312	5.99407996191137
6	0.01261934501423092	-0.9800423629383621	34.57575345655018
7	0.00195638386021743	-0.9819987467985796	221.88360883097684
8	0.00029997663669147	-0.982298723435271	1445.9223376991583
9	0.0000510225	--	--

Tablo 4.6: Alt sınır genişletilirken r, l, ω parametrelerinin değişimi.



Şekil 4.8: Tablo 4.6' daki hesaplamalar sonucu (a) r_i 'nin i 'ye göre grafiği, (b) ω_i 'in i 'ye göre grafiği, (c) ω_i 'in l_i 'e göre grafiği, (d) l_i 'in r_i 'e göre grafiği.

Şimdi de tek parametrelili model olan (4.4) sistemi için yaptığımız Schur kararlılık analizini iki parametreye bağlı olarak inceleyeceğiz.

4.2.2. İki parametrelili makroekonomik model

(4.4) fark-dinamik sistemde x_n değişkeninin katsayısını $(\alpha_1 + \alpha_2 - \alpha_3) = \alpha$, y_n değişkeninin katsayısını ise $(\beta_1 + \beta_2 - \beta_3) = \beta$ şeklinde parametrik ve $\alpha_3 = 0.3$; $\beta_3 = 0.1$ olarak alalım. Bu durumda (4.4) dinamik sistemi,

$$\begin{cases} x_{n+1} = \alpha x_n + 0.1 y_n + 33 \\ y_{n+1} = 0.3 x_n + \beta y_n + 22 \end{cases} \quad (4.8)$$

şekline dönüşür. Elde edilen (4.8) fark-dinamik sisteminin kararlılığını α ve β belirsizlerine göre analiz edelim.

$$A = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & \beta \end{pmatrix}$$

matrisi için Algoritma 3.1 ve Algoritma 3.2 aşağıdaki gibi revize edilerek Algoritma 4.1 ve Algoritma 4.2 bulunur. Algoritmalarındaki revizyonlar aşağıda verildiği gibidir. Bu algoritmaları ve algoritmalara dayanan python kodlarını da aşağıda ki gibi verelim.

Algoritma 4.1 (α, β için Schur Kararlılık Üst Sınırı (u^e)):

1. Veri girişi;

$$A \in S_N, B = E_{11}$$

yeterince küçük r^* ,

$\varepsilon \in (0,1)$ aralığında bir sayı,

$$\gamma \lesssim 1.$$

2, 3, 4 ve 6. adımlar Algoritma 3.1' deki gibidir. 5. adım aşağıdaki gibi olmalıdır.

5. Eğer $r_{k+1} \geq r^* \times \varepsilon$ ise

$$u_{k+1} = u_k + r_{k+1} \text{ eşitliğini hesapla,}$$

$k := k + 1$ al ve 4. adıma git.

6. $M := k$ al ve α için üst sınırını $u^e = u_M$ olarak yaz.

Algoritma 3.1' e ek olarak tekrar veri yenileme yapılır ve başa dönülür. Bu noktada 1. adım aşağıdaki gibi alınır.

1. Veri yenileme; $A_{k+1} := A$ ve $B = E_{22}$ al,

2, 3, 4 ve 6. adımlar Algoritma 3.1' deki gibidir. 5. adım aşağıdaki gibi olmalıdır.

5. Eğer $r_{k+1} \geq r^* \times \varepsilon$ ise

$$u_{k+1} = u_k + r_{k+1} \text{ eşitliğini hesapla,}$$

$k := k + 1$ al ve 4. adıma git.

6. $M := k$ al ve β için üst sınırını $u^e = u_M$ olarak yaz.

Önce β sonra α parametresine müdahale ederek bir üst sınır bulmak için Algoritma 4.1 deki değişiklikler aşağıdaki gibidir.

Algoritma 4.2 (β, α için Schur Kararlılık Üst Sınırı (u^e)):

1. Veri girişi;

$$A \in S_N, B = E_{22}$$

yeterince küçük r^* ,

$\varepsilon \in (0,1)$ aralığında bir sayı,

$$\gamma \lesssim 1.$$

6. $M := k$ al ve β için üst sınırını $u^e = u_M$ olarak yaz.

1. Veri yenileme; $A_{k+1} := A$ matrisi ve $B = E_{11}$ olarak al,

6. $M := k$ al ve α için üst sınırını $u^e = u_M$ olarak yaz.

Yordam 4.1 $((\alpha, \beta)\text{üst sınırı}(A, B, ry))$: Verilen A katsayı matrisi ve B bozunum matrisi olmak üzere yeterince küçük $r^*(= ry)$ pratik değerleri için r^* hassasiyetinde katsayı matrisinin elemanlarına müdahale ile üst sınırı veren Python dilinde $((\alpha, \beta)\text{üst sınırı}(A, B, ry))$ fonksiyonunu oluşturalım.

```
def ((α,β)üst sınırı(A,B,ry)):
import math ; import numpy as np; from scipy import linalg ; import sympy as sp ; from numpy import linalg as LA
print("Matrisin Boyutu:"); n = int(input())
print("A Matrisinin Elemanları:")
u=0; A = [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
A= np.array(elemanlar).reshape(n,n); print("\n A=", A); print("r* değeri girin:"); ry=float(input()); print("r*=",ry)
i=0; j=0; s=i; k=j

for j in range(2):
    j=j+1; print("\n j=",j)
    if j % 2 != 0:
        print("B Matrisinin Elemanları:")
        B= [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
        B= np.array(elemanlar).reshape(n,n); print("\n B=", B)

        i=s
        while True:
            i=i+1; print("\n i=",i)
            Q=np.eye(n) # Birim matris
            H = linalg.solve_discrete_lyapunov(A, Q); #print("H{ }={ } ".format(i,H))
            h=LA.norm(H, 2)
            a=LA.norm(A, 2); print("||A||=",a)
            b=LA.norm(B, 2)
            z=(-a/b)+math.sqrt(a**2+(1/h))/b
            r=z*0.90; print("r=", r)
            if r<ry :
                break

            A=A+r*B; print("A{ }={ } ".format(i,A))
            u=u+r
            K=A
            print("\n ||A||{ }={ } , ω{ }={ } , r{ }={ } ".format(i,a,i,h,i,r))
            print("\n A=",A)

    else:
        print("B Matrisinin Elemanları:")
        B= [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
        B= np.array(elemanlar).reshape(n,n); print("\n B=", B)

        i=s
        ry=ry*(1/10)
        while True:
            i=i+1; print("\n i=",i)
            Q =np.eye(n) # Birim matris
            H = linalg.solve_discrete_lyapunov(A, Q); #print("\n H{ }={ } ".format(i,H))
            h=LA.norm(H, 2)
            a=LA.norm(A, 2)
            b=LA.norm(B, 2)
            z=(-a/b)+math.sqrt(a**2+(1/h))/b
            r=z*0.90; print("r=", r)
            if r<ry :
                break

            A=A+r*B; print("\n A{ }={ } ".format(i,A))
            u=u+r
            K=A
            print("\n ||A||{ }={ } , ω{ }={ } , r{ }={ } ".format(i,a,i,h,i,r))

        A=K

    print("\n A=",A)
    print("\n İterasyon sayısı N : " , i-1)
```

Yordam 4.1: Algoritma 4.1 ve Algoritma 4.2' nin Python yazılım dilinde kodlanması

Algoritma 4.3 (α, β için Schur Kararlılık Alt Sınırı (l^e):

1. Veri girişi;

$$A \in S_N, B = E_{11}$$

yeterince küçük r^* ,

$\varepsilon \in (0,1)$ aralığında bir sayı,

$$\gamma \lesssim 1.$$

2., 3., 4. ve 6. adımlar Algoritma 3.2' deki gibidir. 5. adım aşağıdaki gibi olmalıdır.

5. Eğer $r_{k+1} \geq r^* \times \varepsilon$ ise

$$l_{k+1} = l_k - r_{k+1} \text{ eşitliğini hesapla,}$$

$k := k + 1$ al e 4. adıma git.

6. $M := k$ al ve α için alt sınırını $l^e = l_M$ olarak yaz.

Algoritma 3.2' e ek olarak tekrar veri yenileme yapılır başa dönülür. Bu noktada 1. adım aşağıdaki gibi alınır.

1. Veri yenileme; bulunan $A_{k+1} := A$ matrisi ve $B = E_{22}$ olarak al,

2., 3., 4. ve 6. adımlar Algoritma 3.1' deki gibidir. 5. adım aşağıdaki gibi olmalıdır.

5. Eğer $r_{k+1} \geq r^* \times \varepsilon$ ise

$$l_{k+1} = l_k - r_{k+1} \text{ eşitliğini hesapla,}$$

$k := k + 1$ al e 4. adıma git.

6. $M := k$ al ve β için alt sınırını $l^e = l_M$ olarak yaz.

Önce β sonra α parametresine müdahale ederek bir alt sınır bulmak için Algoritma 4.3 teki değişiklikler aşağıdaki gibi alınır.

Algoritma 4.4 (β, α için Schur Kararlılık Alt Sınırı (l^e):

1. Veri girişi;

$$A \in S_N, B = E_{22}$$

yeterince küçük r^* ,

$\varepsilon \in (0,1)$ aralığında bir sayı,

$$\gamma \lesssim 1.$$

7. $M := k$ al ve β için üst sınırını $l^e = l_M$ olarak yaz.

2. Veri yenileme; $A_{k+1} := A$ matrisi ve $B = E_{11}$ olarak al,

7. $M := k$ al ve α için alt sınırını $l^e = l_M$ olarak yaz.

Yordam 4.2 $((\alpha, \beta)altsınır(A, B, ry))$: Verilen A katsayı matrisi ve B bozunum matrisi olmak üzere yeterince küçük $r^*(= ry)$ pratik değerleri için bize r^* hassasiyetinde katsayı matrisinin elemanlarına müdahale ile alt sınırı veren Python dilinde $((\alpha, \beta)altsınır(A, B, ry))$ fonksiyonunu oluşturalım.

```
def ((α,β)üstsınır(A,B,ry)):
import math ; import numpy as np; from scipy import linalg ; import sympy as sp ; from numpy import linalg as LA
print("Matrisin Boyutu:") ; n = int(input())
print("A Matrisinin Elemanları:")
u=0; A = [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
A= np.array(elemanlar).reshape(n,n) ; print("\n A=", A) ; print("r* değeri girin:") ; ry=float(input()) ; print("r*=",ry)
i=0; j=0; s=i; k=j

for j in range(2):
j=j+1; print("\n j=",j)
if j % 2 != 0:
print("B Matrisinin Elemanları:")
B= [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
B= np.array(elemanlar).reshape(n,n) ; print("\n B=", B)

i=s
while True:
i=i+1; print("\n i=",i)
Q=np.eye(n) # Birim matris
H = linalg.solve_discrete_lyapunov(A, Q) ; #print("H={}".format(i,H))
h=LA.norm(H, 2)
a=LA.norm(A, 2) ; print("||A||=",a)
b=LA.norm(B, 2)
z=(-a/b)+math.sqrt(a**2+(1/h))/b
r=z*0.90 ; print("r=", r)
if r<ry :
break

A=A+r*B ; print("A={}".format(i,A))
u=u+r
K=A
print("\n ||A||={}, ω={}, r={}".format(i,a,i,h,i,r))
print("\n A=",A)

else:
print("B Matrisinin Elemanları:")
B= [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
B= np.array(elemanlar).reshape(n,n) ; print("\n B=", B)

i=s
ry=ry*(1/10)
while True:
i=i+1; print("\n i=",i)
Q =np.eye(n) # Birim matris
H = linalg.solve_discrete_lyapunov(A, Q) ; #print("\nH={}".format(i,H))
h=LA.norm(H, 2)
a=LA.norm(A, 2)
b=LA.norm(B, 2)
z=(-a/b)+math.sqrt(a**2+(1/h))/b
r=z*0.90 ; print("r=", r)
if r<ry :
break

A=A+r*B ; print("\nA={}".format(i,A))
u=u+r
K=A
print("\n ||A||={}, ω={}, r={}".format(i,a,i,h,i,r))

A=K

print("\n A=",A)
print("\n İterasyon sayısı N : ", i-1)
```

Yordam 4.2: Algoritma 4.3 ve Algoritma 4.4' ün Python yazılım dilinde kodlanması

4.2.2.1. Önce α parametresine sonra β parametresine müdahale

Bu kısımda önce α değerine, daha sonra β ' ya müdahale edecek şekilde Algoritma 4.1 ve $((\alpha, \beta)üstsınır(A, B, ry))$ fonksiyonu kullanılarak genişletilmiş üst sınır (u^e) ve Algoritma 4.3 ile $((\alpha, \beta)altsınır(A, B, ry))$ fonksiyonu kullanılarak genişletilmiş alt sınır (l^e) bulunmuştur. Dolayısıyla α ve β değerleri için $\mathfrak{T}^e$ genişletilmiş aralıklar bulunmuştur.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{11}$ al $r < r^*$ olana kadar işleme devam et α değeri için bir u değeri bul ve A_1^2 olarak yaz. $A_1^2 := A_1^1$ ve $B = E_{22}$ al $r < r^* \times \varepsilon$ olarak güncelle β değeri için bir u değeri bul ve A_2^1 olarak yaz.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{11}$ için $r^* = 0.01$ hassasiyetinde

$$A_1^2 = \begin{pmatrix} 0.95881346 & 0.1 \\ 0.3 & 0 \end{pmatrix}$$

bulunur. $A_1^2 := A_1^1$, $B = E_{22}$ için $\varepsilon = 10^{-1}$ ve $r^* = 0.01 \times \varepsilon$ hassasiyetinde

$$A_2^1 = \begin{pmatrix} 0.95881346 & 0.1 \\ 0.3 & 0.24875128 \end{pmatrix}$$

bulunur. Buradan

$$A = A(\alpha, \beta) = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & \beta \end{pmatrix} ; \begin{matrix} \alpha \in [0, 0.95881346] \\ \beta \in [0, 0.24875128] \end{matrix}$$

olarak elde edilir.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{11}$ al $r < r^*$ olana kadar işleme devam et α değeri için bir l değeri bul ve A_1^2 olarak yaz. $A_1^2 := A_1^1$ ve $B = E_{22}$ al $r < r^* \times \varepsilon$ olarak güncelle β değeri için bir l değeri bul ve A_2^1 olarak yaz.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{11}$ için $r^* = 0.01$ hassasiyetinde

$$A_1^2 = \begin{pmatrix} -0.95881346 & 0.1 \\ 0.3 & 0 \end{pmatrix}$$

bulunur. $A_1^2 := A_1^1$, $B = E_{22}$ için $\varepsilon = 10^{-1}$ ve $r^* = 0.01 \times \varepsilon$ hassasiyetinde

$$A_2^1 = \begin{pmatrix} -0.95881346 & 0.1 \\ 0.3 & -0.24875128 \end{pmatrix}$$

bulunur. Buradan

$$A = A(\alpha, \beta) = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & \beta \end{pmatrix} ; \quad \alpha \in [-0.95881346, 0] \\ \beta \in [-0.24875128, 0]$$

olarak elde edilir.

Sonuç 4.1: Algoritma 4.1, Algoritma 4.3 kullanılarak $r^* = 0.01$ hassasiyetinde önce α sonra β elemanına müdahale edildiğinde aralıklar $\alpha \in [-0.95881346, 0.95881346]$ ve $\beta \in [-0.24875128, 0.24875128]$ şeklinde elde edilir. α, β değerleri bu aralıklardan seçilirse (4.8) sistemi Schur kararlıdır.

4.2.2.2. Önce β parametresine sonra α parametresine müdahale

Bu kısımda önce β değerine sonra α değerine müdahale edecek şekilde Algoritma 4.2 kullanılarak genişletilmiş üst sınır (u^e) ve Algoritma 4.4 kullanılarak genişletilmiş alt sınır (l^e) bulunmuştur. α ve β değerleri için $\mathfrak{T}^e$ genişletilmiş aralıklar bulunmuştur.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{22}$ al $r < r^*$ den küçük olana kadar işleme devam et β değeri için bir u değeri bul ve A_1^2 olarak yaz. $A_1^2 := A_1^1$ ve $B = E_{11}$ al $r < r^* \times \varepsilon$ olarak güncelle α değeri için bir u değeri bul ve A_2^1 olarak yaz.
- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{22}$ için $r^* = 0.01$ hassasiyetinde

$$A_1^2 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0.96965939 \end{pmatrix}$$

bulunur. $A_1^2 := A_1^1$, $B = E_{11}$ için $\varepsilon = 10^{-1}$ ve $r^* = 0.01 \times \varepsilon$ hassasiyetinde

$$A_2^1 = \begin{pmatrix} 0.24875128 & 0.1 \\ 0.3 & 0.96965939 \end{pmatrix}$$

bulunur. Buradan

$$A = A(\alpha, \beta) = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & \beta \end{pmatrix} ; \quad \alpha \in [0, 0.24875128] \\ \beta \in [0, 0.96965939]$$

olarak elde edilir.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{22}$ al $r < r^*$ den küçük olana kadar işleme devam et β değeri için bir l değeri bul ve A_1^2 olarak yaz. $A_1^2 := A_1^1$ ve $B = E_{11}$ al $r < r^* \times \varepsilon$ olarak güncelle α değeri için bir l değeri bul ve A_2^1 olarak yaz.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}, B = E_{22}$ için $r^* = 0.01$ hassasiyetinde

$$A_1^2 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & -0.96965939 \end{pmatrix}$$

bulunur. $A_1^2 := A_1^1, B = E_{11}$ için $\varepsilon = 10^{-1}$ ve $r^* = 0.01 \times \varepsilon$ hassasiyetinde

$$A_2^1 = \begin{pmatrix} -0.24875128 & 0.1 \\ 0.3 & 0.96965939 \end{pmatrix}$$

bulunur. Buradan

$$A = A(\alpha, \beta) = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & \beta \end{pmatrix} ; \begin{matrix} \alpha \in [-0.24875128, 0] \\ \beta \in [-0.96965939, 0] \end{matrix}$$

olarak elde edilir.

Sonuç 4.2: Algoritma 4.2, Algoritma 4.4 kullanılarak $r^* = 0.01$ hassasiyetinde önce β sonra α elemanına müdahale edildiğinde aralıklar $\alpha \in [-0.24875128, 0.24875128]$ ve $\beta \in [-0.95881346, 0.95881346]$ elde edilir. α, β değerleri bu aralıklardan seçilirse (4.8) sistemi Schur kararlıdır.

4.2.2.3. α ve β parametrelerine birlikte müdahale

Bu kısımda α, β elemanlarına birlikte müdahale edecek şekilde Algoritma 3.1 kullanılarak genişletilmiş üst sınır (u^e) ve Algoritma 3.2 kullanılarak genişletilmiş alt sınır (l^e) bulunmuştur. Yani α ve β elemanları için $\mathfrak{T}^e$ genişletilmiş aralıklar bulunmuştur.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}, B = E_{11} + E_{22}$ al u değeri bul ve A_1^2 olarak yaz. $r < r^* = 0.001$ olana kadar işleme devam et.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}, B = E_{11} + E_{22}$ için $A_1^2 = \begin{pmatrix} 0.632967 & 0.1 \\ 0.3 & 0.632967 \end{pmatrix}$

bulunur. Buradan $r^* = 0.001$ hassasiyetiyle

$$A = A(\alpha, \beta) = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & \beta \end{pmatrix} ; \begin{matrix} \alpha \in [0, 0.826315] \\ \beta \in [0, 0.826315] \end{matrix}$$

olarak elde edilir.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{11} + E_{22}$ al u değeri bul ve A_1^2 olarak yaz.
 $r < r^* = 0.001$ olana kadar işleme devam et.
- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{11} + E_{22}$ için $A_1^2 = \begin{pmatrix} -0.632967 & 0.1 \\ 0.3 & -0.632967 \end{pmatrix}$
 bulunur. Buradan $r^* = 0.001$ hassasiyetiyle

$$A = A(\alpha, \beta) = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & \beta \end{pmatrix} ; \begin{matrix} \alpha \in [-0.826315, 0] \\ \beta \in [-0.826315, 0] \end{matrix}$$

olarak elde edilir.

Sonuç 4.3: Algoritma 3.1, Algoritma 3.2 kullanılarak $r^* = 0.001$ hassasiyetinde α ve β elemanlarına aynı anda müdahale edildiğinde α, β değerleri $[-0.826315, 0.826315]$ aralığından seçilirse (4.8) sistemi Schur kararlıdır.

Tablo 4.7’ de α ve β elemanlarına müdahale edilerek bulunan Schur kararlılık aralıkları özetlenmiştir.

Önce α Sonra β		
$\mathfrak{T}^e = [l^e, u^e]$	$[-0.95881346, 0.95881346]$	$[-0.24875128, 0.24875128]$
Önce β Sonra α		
$\mathfrak{T}^e = [l^e, u^e]$	$[-0.24875128, 0.24875128]$	$[-0.96965939, 0.96965939]$
α, β aynı anda		
$\mathfrak{T}^e = [l^e, u^e]$	$[-0.826315, 0.826315]$	$[-0.826315, 0.826315]$

Tablo 4.7: α ve β elemanlarına müdahale edilerek bulunan Schur kararlılık aralıkları

4.3. Makroekonomik Modelin ω^* – Schur Kararlılığı

Bu bölümde (4.4) sisteminin önce tek parametreye bağlı (4.6) modeli için sonra da iki parametreye bağlı (4.8) modeli için ω^* – Schur kararlılığı incelenmiştir.

4.3.1. Tek parametrelili makroekonomik model

(4.7) denkleminde

$$\|A_1\| = 0.767118, \omega(A_1) = 2.297738, \Delta = \alpha E_{11}, \|\Delta\| = |\alpha|, \omega^* = 100$$

hesaplanır. Teorem 2.4’ den

$$|\alpha| < 0.23969896$$

eşitsizliğine ulaşılır. Dolayısıyla $\alpha \in (-0.23969896, 0.23969896)$ elde edilir. Teorem ω^* – Schur kararlılığın sağlandığı güvenli bir aralık vermektedir. Buradan görüldüğü üzere bulunan ω^* – Schur kararlı aralık daha önce bulduğumuz Schur kararlı aralığın içinde kalmaktadır. Şimdi ω^* – Schur kararlı olduğunu bildiğimiz aralığı $pr_üstsınır(A, B, \omega^*)$, $pr_altsınır(A, B, \omega^*)$ fonksiyonları yardımı ile genişletelim.

(4.7) matrisine göre (4.6) sistemini farklı r^* ve ω^* değerleri alarak $pr_üstsınır(A, B, \omega^*)$ fonksiyonu (Yordam 3.1) ile Tablo 4.8 deki $pr_altsınır(A, B, \omega^*)$ fonksiyonu (Yordam 3.2) ile Tablo 4.9 daki veriler hesaplanır.

A	B	ω^*	r^*	u^*	u^{*e}	M
$\begin{pmatrix} 0 & 0.1 \\ 0.3 & 0.7 \end{pmatrix}$	E_{11}	10	0.1	0.19396494	0.64762667	4
			0.01		0.80314709	8
			0.001		0.81118215	11
		100	0.1	0.23969896	0.76931138	4
			0.01		0.88194446	7
			0.001		0.88992982	9

Tablo 4.8: Farklı r^* , ω^* parametreleri ile durulan adımlar ve genişletilmiş üst sınır değerleri.

A	B	ω^*	r^*	l^*	l^{*e}	M
$\begin{pmatrix} 0 & 0.1 \\ 0.3 & 0.7 \end{pmatrix}$	E_{11}	10	0.1	-0.19396494	-0.91580151	5
			0.01		-0.92859497	6
			0.001		-0.92859497	6
		100	0.1	-0.23969896	-0.94373139	4
			0.01		-0.97505165	5
			0.001		-0.97708769	6

Tablo 4.9: Farklı r^* , ω^* parametreleri ile durulan adımlar ve genişletilmiş alt sınır değerleri.

Buna göre genişletilmiş üst sınır (Tablo 4.8) $r^* = 0.001$, $\omega^* = 10$ pratiklik parametreleri kullanılarak 11 adımda $u^{*e} = 0.81118215$, genişletilmiş alt sınır (Tablo 4.9) $r^* = 0.001$, $\omega^* = 10$ pratiklik parametreleri kullanılarak 6 adımda $l^{*e} = -0.92859497$ elde edilir. $r^* = 0.001$, $\omega^* = 100$ pratiklik parametreleri kullanılarak, genişletilmiş üst sınır 9 adımda $u^{*e} = 0.88992982$ ve genişletilmiş alt sınır 11 adımda $l^{*e} = -0.97708769$ elde edilir.

Sonuç 4.4: $pr_üstsınır(A, B, \omega^*)$ fonksiyonu (Yordam 3.3) ve $pr_üstsınır(A, B, \omega^*)$ fonksiyonu (Yordam 3.4) kullanılarak, $r^* = 0.001$, $\omega^* = 10$ pratiklik parametreleri ile

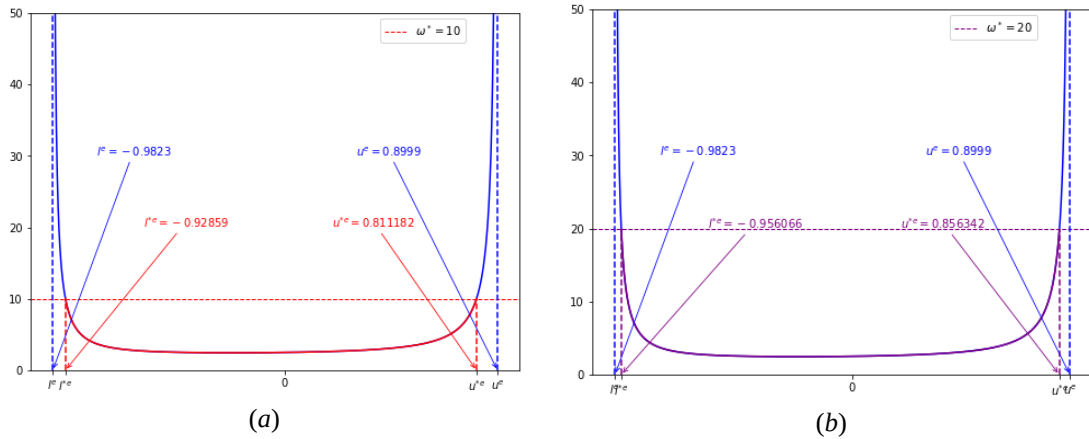
$$\mathfrak{T}^{*e} = [l^{*e}, u^{*e}] = [-0.92859497, 0.81118215]$$

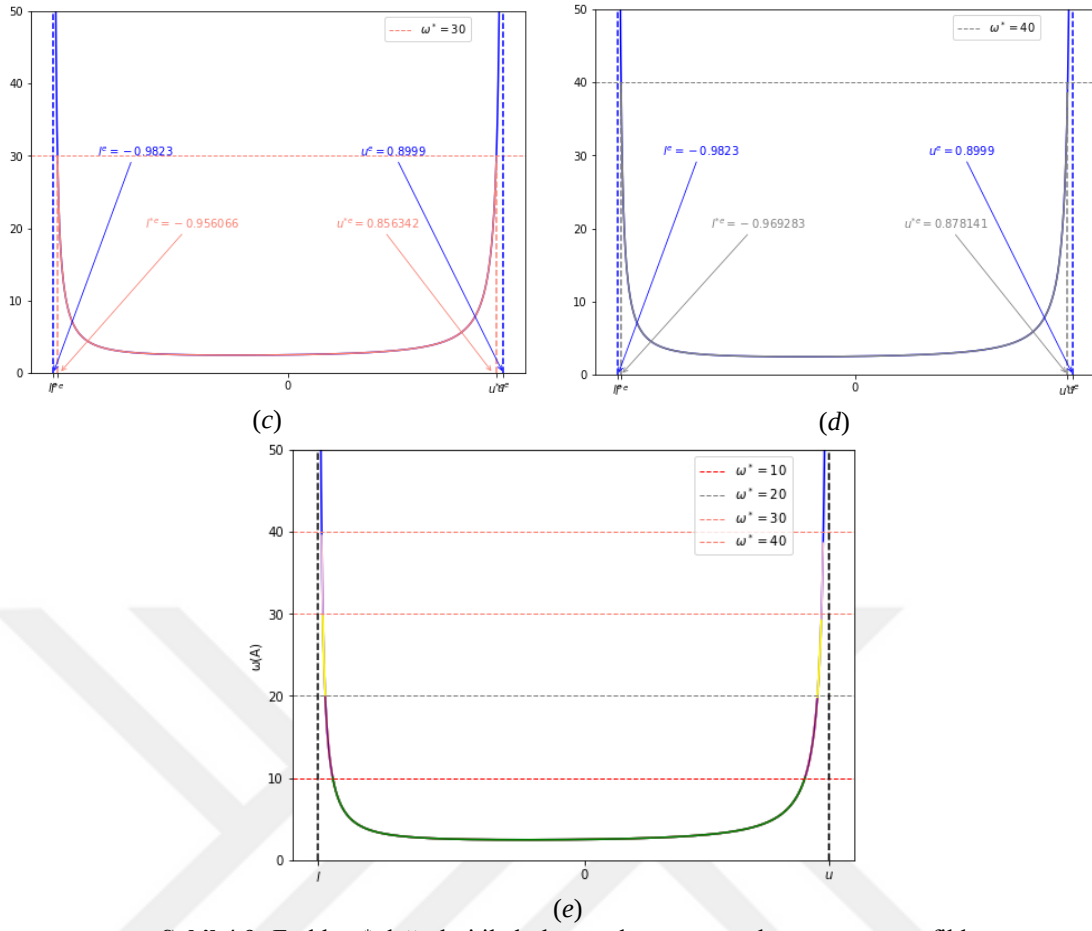
aralığı elde edilir. Buradan α bu aralıktan seçildiği sürece (4.6) sistemi $\omega^* = 10$, ω^* – Schur kararlı bir sistemdir. $r^* = 0.001$, $\omega^* = 100$ pratiklik parametreleri ile

$$\mathfrak{T}^{*e} = [l^{*e}, u^{*e}] = [-0.97708769, 0.88992982]$$

aralığı elde edilir. Buradan α bu aralıktan seçildiği sürece (4.6) sistemi $\omega^* = 100$, ω^* – Schur kararlı bir sistemdir.

Aşağıda farklı ω^* – Schur kararlılık parametrelerine göre hesaplanan u^{*e} ve l^{*e} sınırları grafiklerle gösterilmiştir. Şekil 4.9 (a) da $\omega^* = 10$ olarak seçildiğinde hesaplanan aralığın $üstsınır(A, B, ry)$ ve $üstsınır(A, B, ry)$ fonksiyonları ile bulunan aralığın içinde olduğu ve ω^* değeri ile hesaplanan aralıklardan seçilen parametre için ω değerlerinin küçük olduğu dolayısıyla bulunan aralığın Schur kararlılığının dış etkilere dayanıklı olduğu görülür. Benzer şekilde Şekil 4.9 (b), Şekil 4.9 (c) ve Şekil 4.9 (d) içinde seçilen ω^* değerlerine göre aynı durum geçerlidir.





Şekil 4.9: Farklı ω^* değerleri ile bulunan alt ve üst sınırları gösteren grafikler.

4.3.2. İki parametrelili makroekonomik model

(4.5) fark-dinamik sisteminde x_n değişkeninin katsayısını $(\alpha_1 + \alpha_2 - \alpha_3) = \alpha$, y_n değişkeninin katsayısını ise $(\beta_1 + \beta_2 - \beta_3) = \beta$ şeklinde parametrik olarak alalım. Bu durumda (4.5) dinamik sistemi,

$$\begin{cases} x_{n+1} = \alpha x_n + 0.1 y_n + 33 \\ y_{n+1} = 0.3 x_n + \beta y_n + 22 \end{cases} \quad (4.9)$$

şekline dönüşür. Şimdi (4.9) fark-dinamik sisteminin ω^* – Schur kararlılığını α ve β belirsizlerine göre analiz edelim.

$$A = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & \beta \end{pmatrix}$$

matrisi için Algoritma 3.3 ve Algoritma 3.4 aşağıdaki gibi revize edilerek Algoritma 4.5 ve Algoritma 4.6 bulunur. Elde edilen bu algoritmaları ve algoritmalara dayanan python kodlarını verelim.

Önce α sonra β parametresine müdahale etmek için Algoritma 4.5' i verelim.

Algoritma 4.5 (α ve β değerlerine göre ω^* – Schur Kararlılık için Üst Sınır (u^{*e}):

1. Veri girişi;

$$A \in S_N, B = E_{11}$$

yeterince küçük r^* ,

$1 < \omega^*$ problemin türüne göre kullanıcı tarafından belirlenen bir sayı.

$$\varepsilon \in (0,1).$$

2., 3. ve 4. adımlar Algoritma 3.3' deki gibidir. 5. ve 6. adımlar aşağıdaki gibi olmalıdır.

5. Eğer $r_{k+1} \geq r^* \times \varepsilon$ ise

$$u_{k+1} = u_k + r_{k+1} \text{ eşitliğini hesapla,}$$

$k := k + 1$ al ve 4. adıma git.

6. $M := k$ al ve β değeri için üst sınırı $u^{*e} = u_M^*$ olarak yaz.

Algoritma 3.3' e ek olarak tekrar veri yenileme yapılır ve başa dönülür. Bu noktada 1. adım aşağıdaki gibi alınır.

1. Veri yenileme; $A_{k+1} := A$ ve $B = E_{22}$ al,

2., 3. ve 4. adımlar Algoritma 3.3' deki gibidir. 5. ve 6. adımlar aşağıdaki gibi olmalıdır.

5. Eğer $r_{k+1} \geq r^* \times \varepsilon$ ise

$$u_{k+1} = u_k + r_{k+1} \text{ eşitliğini hesapla,}$$

$k := k + 1$ al ve 4. adıma git.

6. $M := k$ al ve α değeri için üst sınırı $u^{*e} = u_M^*$ olarak yaz.

Önce β sonra α parametresine müdahale ederek bir üst sınır bulmak için Algoritma 4.5 deki değişiklikler aşağıdaki gibidir.

Algoritma 4.6 (β ve α değerleri için ω^* – Schur Kararlılık Üst Sınırı (u^{*e}):

1. Veri girişi;

$$A \in S_N, B = E_{22}$$

yeterince küçük r^* ,

$1 < \omega^*$ problemin türüne göre kullanıcı tarafından belirlenen bir sayı.

$$\varepsilon \in (0,1) \text{ aralığında bir sayı.}$$

Veri yenileme; $A_{k+1} := A$ matrisi ve $B = E_{11}$ olarak al,

6. $M := k$ al ve α için üst sınırını $u^{*e} = u_M^*$ olarak yaz.

Algoritma 3.3' e ek olarak tekrar veri yenileme yapılır ve başa dönlür. Bu noktada 1. adım aşağıdaki gibi alınır.

1. Veri yenileme; $A_{k+1} := A$ ve $B = E_{11}$ al,
- 2., 3. ve 4. adımlar Algoritma 3.3' deki gibidir. 5. ve 6. adımlar aşağıdaki gibi olmalıdır.
5. Eğer $r_{k+1} \geq r^* \times \varepsilon$ ise

$$u_{k+1} = u_k + r_{k+1}$$
 eşitliğini hesapla,
 $k := k + 1$ al ve 4. adıma git.
6. $M := k$ al ve β değeri için üst sınırını $u^{*e} = u_M^*$ olarak yaz.

Yordam 4.3 $((\alpha, \beta)pr_üstsnır(A, B, wy))$: Verilen A katsayı matrisi ve B bozunum matrisi olmak üzere yeterince küçük $r^* (= ry)$ ve kullanıcı tarafından seçilen $\omega^* (= wy)$ pratik değerleri için r^* ve ω^* hassasiyetinde katsayı matrisinin elemanlarına müdahale ile üst sınırı veren Python dilinde $((\alpha, \beta)pr_üstsnır(A, B, wy))$ fonksiyonunu oluşturalım.

```
def ((α,β)pr_üstsnır(A,B,wy)):
import math ; import numpy as np; from scipy import linalg ; import sympy as sp ; from numpy import linalg as LA
print("Matrisin Boyutu:"); n = int(input())
print("A Matrisinin Elemanları:")
u=0; A = [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
A= np.array(elemanlar).reshape(n,n); print("n A=", A); print("r* değeri girin:"); ry=float(input()); print("r*=",ry)
print("ω* değeri girin"); wy=float(input()); print("ω*=",wy)
i=0; j=0; s=i; k=j
for j in range(2):
    j=j+1; print("\n j=",j)
    if j % 2 != 0:
        B=np.array([[1, 0],
                     [0, 0]]); print("\nB=",B)
        i=s
        while True:
            i=i+1; print("\n i=",i)
            Q=np.eye(n) # Birim matris
            H = linalg.solve_discrete_lyapunov(A, Q); #print("H{}={}".format(i,H))
            h=L.A.norm(H, 2)
            a=L.A.norm(A, 2); print("||A||=",a)
            b=L.A.norm(B, 2)
            r=(-a/b)+math.sqrt(a**2-(1/wy)+(1/h))/b; print("r=", r)
            if r<ry:
                break
            A=A+r*B; print("A{}={}".format(i,A))
            u=u+r
            K=A
            print("\n ||A||{}={}, ω{}={}, r{}={}".format(i,a,i,h,i,r))
            print("\n A=",A)
        else:
            B=np.array([[0, 0],
                         [0, 1]]); print("\nB=",B)
            i=s
            ry=ry*(1/10)
            while True:
                i=i+1; print("\n i=",i)
                Q=np.eye(n) # Birim matris
                H = linalg.solve_discrete_lyapunov(A, Q); #print("\nH{}={}".format(i,H))
                h=L.A.norm(H, 2)
                a=L.A.norm(A, 2)
                b=L.A.norm(B, 2)
                r=(-a/b)+math.sqrt(a**2-(1/wy)+(1/h))/b; print("r=", r)
                if r<ry:
                    break
                A=A+r*B; print("\nA{}={}".format(i,A))
                u=u+r
                K=A
                print("\n ||A||{}={}, ω{}={}, r{}={}".format(i,a,i,h,i,r))
            A=K
print("\n A=",A)
print("\n İterasyon sayısı N : ", i-1)
```

Yordam 4.3: Algoritma 4.5 ve Algoritma 4.6 Python yazılım dilinde kodlanması

Benzer şekilde, önce α sonra β parametrelerine müdahale ederek alt sınırı bulan Algoritma 4.7' yi, ardından ise önce β sonra α parametrelerine müdahale ederek alt sınırı bulan Algoritma 4.8' i verelim.

Algoritma 4.7 (α, β için ω^* – Schur Kararlılık Alt Sınırı (l^{*e}):

1. Veri girişi;

$$A \in S_N, B = E_{11}$$

yeterince küçük r^* ,

$1 < \omega^*$ problemin türüne göre kullanıcı tarafından belirlenen bir sayı.

$$\varepsilon \in (0,1).$$

2., 3. ve 4. adımlar Algoritma 3.4' deki gibidir. 5. ve 6. adımlar aşağıdaki gibi olmalıdır.

5. Eğer $r_{k+1} \geq r^* \times \varepsilon$ ise

$$l_{k+1} = l_k - r_{k+1} \text{ eşitliğini hesapla,}$$

$k := k + 1$ al ve 4. adıma git.

6. $M := k$ al ve β değeri için alt sınırı $l^{*e} = l_M^*$ olarak yaz.

Algoritma 3.4' e ek olarak tekrar veri yenileme yapılır ve başa dönlür. Bu noktada 1. adım aşağıdaki gibi alınır.

1. Veri yenileme; bulunan $A_{k+1} := A$ matrisi ve $B = E_{22}$ olarak al,

2., 3. ve 4. adımlar Algoritma 3.4' deki gibidir. 5. ve 6. adımlar aşağıdaki gibi olmalıdır.

5. Eğer $r_{k+1} \geq r^* \times \varepsilon$ ise

$$l_{k+1} = l_k - r_{k+1} \text{ eşitliğini hesapla,}$$

$k := k + 1$ al ve 4. adıma git.

6. $M := k$ al ve α değeri için alt sınırı $l^{*e} = l_M^*$ olarak yaz.

Algoritma 4.8 (β ve α değerleri için ω^* – Schur Kararlılık Alt Sınırı (l^{*e}):

1. Veri girişi;

$$A \in S_N, B = E_{22}$$

yeterince küçük r^* ,

$1 < \omega^*$ problemin türüne göre kullanıcı tarafından belirlenen bir sayı.

$$\varepsilon \in (0,1).$$

2. Veri yenileme; $A_{k+1} := A$ ve $B = E_{11}$ al,

6. $M := k$ al ve α için üst sınırı $l^{*e} = l_M^*$ olarak yaz.

Algoritma 3.3' e ek olarak tekrar veri yenileme yapılır ve başa dönülür. Bu noktada 1. adım aşağıdaki gibi alınır.

1. Veri yenileme; $A_{k+1} := A$ ve $B = E_{11}$ al,
- 2., 3. ve 4. adımlar Algoritma 3.3' deki gibidir. 5. ve 6. adımlar aşağıdaki gibi olmalıdır.
5. Eğer $r_{k+1} \geq r^* \times \varepsilon$ ise

$$l_{k+1} = l_k - r_{k+1} \text{ eşitliğini hesapla,}$$

$$k := k + 1 \text{ al ve 4. adıma git.}$$
6. $M := k$ al ve β değeri için üst sınırı $l^* = l_M^*$ olarak yaz.

Yordam 4.4 $((\alpha, \beta)pr_altsınır(A, B, wy))$: Verilen A katsayı matrisi ve B bozunum matrisi olmak üzere yeterince küçük $r^* (= ry)$ ve kullanıcı tarafından seçilen $\omega^* (= wy)$ pratik değerleri için, r^* ve ω^* hassasiyetinde katsayı matrisinin elemanlarına müdahale ile alt sınırı veren Python dilinde $((\alpha, \beta)pr_altsınır(A, B, wy))$ fonksiyonunu oluşturalım.

```
def ((α, β)pr_altsınır(A, B, wy)):
import math ; import numpy as np; from scipy import linalg ; import sympy as sp ; from numpy import linalg as LA
print("Matrisin Boyutu:"); n = int(input())
print("A Matrisinin Elemanları:")
l=0; A = [[0 for i in range(n)] for i in range(n)]; elemanlar= list(map(float, input().split()));
A= np.array(elemanlar).reshape(n,n); print("\n A=", A) ; print("r* değeri girin:"); ry=float(input()); print("r*=",ry)
print("ω* değeri girin"); wy=float(input()); print("ω*=",wy)
i=0; j=0; s=i; k=j
for j in range(2):
    j=j+1; print("\n j=",j)
    if j % 2 != 0:
        B=np.array([[1, 0],
                     [0, 0]]); print("\nB=",B)
        i=s
        while True:
            i=i+1; print("\n i=",i)
            Q=np.eye(n) # Birim matris
            H = linalg.solve_discrete_lyapunov(A, Q) ; #print("\nH{ }={ } ".format(i,H))
            h=LA.norm(H, 2)
            a=LA.norm(A, 2) ; print("||A||=",a)
            b=LA.norm(B, 2)
            r=(-a/b)+math.sqrt(a**2-(1/wy)+(1/h))/b ; print("r=", r)
            if r<ry :
                break
            A=A-r*B ; print("A{ }={ } ".format(i,A))
            l=l-r
            K=A
            print("\n ||A|| { } = { } , ω { } = { } , r { } = { } ".format(i,a,i,h,i,r))
            print("\n A=",A)
        else:
            B=np.array([[0, 0],
                     [0, 1]]); print("\nB=",B)
            i=s
            ry=ry*(1/10)
            while True:
                i=i+1; print("\n i=",i)
                Q =np.eye(n) # Birim matris
                H = linalg.solve_discrete_lyapunov(A, Q) ; #print("\nH{ }={ } ".format(i,H))
                h=LA.norm(H, 2)
                a=LA.norm(A, 2)
                b=LA.norm(B, 2)
                r=(-a/b)+math.sqrt(a**2-(1/wy)+(1/h))/b ; print("r=", r)
                if r<ry :
                    break
                A=A-r*B ; print("\nA{ }={ } ".format(i,A))
                l=l-r
                K=A
                print("\n ||A|| { } = { } , ω { } = { } , r { } = { } ".format(i,a,i,h,i,r))
            A=K
            print("\n A=",A)
            print("\n İterasyon sayısı N : ", i-1)
```

Yordam 4.4: Algoritma 4.7 ve Algoritma 4.8' in Python yazılım dilinde kodlanması

4.3.2.1. Önce α parametresine sonra β parametresine müdahale

Bu kısımda önce α sonra β ' değerlerine müdahale edecek şekilde Algoritma 4.5 ve $((\alpha, \beta)pr_üstsınır(A, B, ry))$ fonksiyonu kullanılarak genişletilmiş pratik üst sınır (u^{*e}) ve Algoritma 4.7 ile $((\alpha, \beta)pr_altsınır(A, B, ry))$ fonksiyonu kullanılarak genişletilmiş pratik alt sınır (l^{*e}) bulunmuştur. Yani α ve β değerleri için $\mathfrak{T}^{*e}$ genişletilmiş aralıklar bulunmuştur.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{11}$, $\omega^* = 10$ al $r < r^*$ den küçük olana kadar işleme devam et ve α değeri için bir u^* değeri bul ve A_1^2 olarak yaz. $A_1^2 := A_1^1$ ve $B = E_{22}$ al $r < r^* \times \varepsilon$ olarak güncelle β için bir u^* değeri bul ve A_2^1 olarak yaz.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{11}$ için $r^* = 0.01$ hassasiyetinde

$$A_1^2 = \begin{pmatrix} 0.91264831 & 0.1 \\ 0.3 & 0 \end{pmatrix}$$

bulunur. $A_1^2 := A_1^1$, $B = E_{22}$ için $\varepsilon = 10^{-1}$ ve $r^* = 0.01 \times \varepsilon$ hassasiyetinde

$$A_2^1 = \begin{pmatrix} 0.91264831 & 0.1 \\ 0.3 & 0.03337832 \end{pmatrix}$$

bulunur. Buradan

$$A = A(\alpha, \beta) = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & \beta \end{pmatrix} ; \begin{matrix} \alpha \in [0, 0.91264831] \\ \beta \in [0, 0.03337832] \end{matrix}$$

olarak elde edilir.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{11}$, $\omega^* = 10$ al $r < r^*$ olana kadar işleme devam et α değeri için bir u^* değeri bul ve A_1^2 olarak yaz. $A_1^2 := A_1^1$ ve $B = E_{22}$ al $r < r^* \times \varepsilon$ olarak güncelle β değeri için bir u^* değeri bul ve A_2^1 olarak yaz.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{11}$ için $r^* = 0.01$ hassasiyetinde

$$A_1^2 = \begin{pmatrix} -0.91264831 & 0.1 \\ 0.3 & 0 \end{pmatrix}$$

bulunur. $A_1^2 := A_1^1$, $B = E_{22}$ için $\varepsilon = 10^{-1}$ ve $r^* = 0.01 \times \varepsilon$ hassasiyetinde

$$A_2^1 = \begin{pmatrix} -0.91264831 & 0.1 \\ 0.3 & -0.03337832 \end{pmatrix}$$

bulunur. Buradan

$$A = A(\alpha, \beta) = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & \beta \end{pmatrix} ; \quad \alpha \in [-0.91264831, 0] \\ \beta \in [-0.03337832, 0]$$

olarak elde edilir.

Sonuç 4.4: Algoritma 4.5, Algoritma 4.7 kullanılarak $r^* = 0.01$ hassasiyetinde $\omega^* = 10$ pratiklik parametresi ile önce α sonra β elemanına müdahale edildiğinde aralıkları $\alpha \in [-0.91264831, 0.91264831]$ ve $\beta \in [-0.03337832, 0.03337832]$ şeklinde elde edilir. α ve β değerleri bu aralıklardan seçilirse (4.8) sistemi ω^* – Schur kararlıdır.

4.3.2.2. Önce β parametresine sonra α parametresine müdahale

Bu kısımda önce β sonra α değerine müdahale edecek şekilde Algoritma 4.6 ve $((\alpha, \beta)pr_üstsınır(A, B, ry))$ fonksiyonu kullanılarak genişletilmiş pratik üst sınır (u^{*e}) ve Algoritma 4.8 ile $((\alpha, \beta)pr_altsınır(A, B, ry))$ fonksiyonu kullanılarak genişletilmiş pratik alt sınır (l^{*e}) bulunmuştur. Yani α ve β değerleri için $\mathfrak{T}^{*e}$ genişletilmiş aralıklar bulunmuştur.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{22}$, $\omega^* = 10$ al $r < r^*$ olana kadar işleme devam et β değeri için bir u^* değeri bul ve A_1^2 olarak yaz. $A_1^2 := A_1^1$ ve $B = E_{11}$ al $r < r^* \times \varepsilon$ olarak güncelle α değeri için bir u^* değeri bul ve A_2^1 olarak yaz.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{22}$ için $r^* = 0.01$ hassasiyetinde

$$A_1^2 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0.91264831 \end{pmatrix}$$

bulunur. $A_1^2 := A_1^1$, $B = E_{11}$ için $\varepsilon = 10^{-1}$ ve $r^* = 0.01 \times \varepsilon$ hassasiyetinde

$$A_2^1 = \begin{pmatrix} 0.03337832 & 0.1 \\ 0.3 & 0.91264831 \end{pmatrix}$$

bulunur. Buradan

$$A = A(\alpha, \beta) = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & \beta \end{pmatrix} ; \quad \alpha \in [0, 0.03337832] \\ \beta \in [0, 0.91264831]$$

olarak elde edilir.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{22}$, $\omega^* = 10$ al $r < r^*$ olana kadar işleme devam et β değeri için bir u^* değeri bul ve A_1^2 olarak yaz. $A_1^2 := A_1^1$ ve $B = E_{11}$ al $r < r^* \times \varepsilon$ olarak güncelle α değeri için bir u^* değeri bul ve A_2^1 olarak yaz.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}, B = E_{22}$ için $r^* = 0.001$ hassasiyetinde

$$A_1^2 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & -0.91264831 \end{pmatrix}$$

bulunur. $A_1^2 := A_1^1, B = E_{11}$ için $r^* = 0.0001$ hassasiyetinde

$$A_2^1 = \begin{pmatrix} -0.03337832 & 0.1 \\ 0.3 & -0.91264831 \end{pmatrix}$$

bulunur. Buradan

$$A = A(\alpha, \beta) = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & \beta \end{pmatrix} ; \begin{matrix} \alpha \in [-0.03337832, 0] \\ \beta \in [-0.91264831, 0] \end{matrix}$$

olarak elde edilir.

Sonuç 4.5: Algoritma 4.6, Algoritma 4.8 kullanılarak $r^* = 0.01$ hassasiyetinde $\omega^* = 10$ pratiklik parametresi ile önce β sonra α elemanına müdahale edildiğinde aralıklar $\alpha \in [-0.03337832, 0.03337832]$ ve $\beta \in [-0.91264831, 0.91264831]$ şeklinde elde edilir. α ve β değerleri elde edilen bu aralıklardan seçilirse (4.8) sistemi ω^* – Schur kararlıdır.

4.3.2.3. α ve β parametrelerine birlikte müdahale

Bu kısımda α ve β elemanlarına birlikte müdahale edecek şekilde Algoritma 3.3 kullanılarak genişletilmiş üst sınır (u^{*e}) ve Algoritma 3.4 kullanılarak genişletilmiş alt sınır (l^{*e}) bulunmuştur. Yani α ve β değerleri için $\mathfrak{T}^{*e}$ genişletilmiş aralıklar elde edilmiştir.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}, B = E_{11} + E_{22}$ al, u değeri bul ve A_1^2 olarak yaz. $r < r^* = 0.001$ olana kadar işleme devam et.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}, B = E_{11} + E_{22}$ için $A_1^2 = \begin{pmatrix} 0.65215834 & 0.1 \\ 0.3 & 0.65215834 \end{pmatrix}$

bulunur. Buradan $r^* = 0.001$ hassasiyetiyle

$$A = A(\alpha, \beta) = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & \beta \end{pmatrix} ; \begin{matrix} \alpha \in [0, 0.76376453] \\ \beta \in [0, 0.76376453] \end{matrix}$$

olarak elde edilir.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{11} + E_{22}$, $\omega^* = 10$ al, u^* değeri bul ve A_1^2 olarak yaz.
 $r < r^* = 0.001$ olana kadar işleme devam et.

- $A_1^1 = \begin{pmatrix} 0 & 0.1 \\ 0.3 & 0 \end{pmatrix}$, $B = E_{11} + E_{22}$ için

$$A_1^2 = \begin{pmatrix} -0.65215834 & 0.1 \\ 0.3 & -0.65215834 \end{pmatrix}$$

bulunur. Buradan $r^* = 0.001$ hassasiyetiyle

$$A = A(\alpha, \beta) = \begin{pmatrix} \alpha & 0.1 \\ 0.3 & \beta \end{pmatrix} ; \begin{matrix} \alpha \in [-0.76376453, 0] \\ \beta \in [-0.76376453, 0] \end{matrix}$$

olarak elde edilir.

Sonuç 4.6: Algoritma 3.3 ve Algoritma 3.4 kullanılarak $r^* = 0.01$ hassasiyetinde $\omega^* = 10$ pratiklik parametresi ile α ve β elemanlarına aynı anda müdahale edildiğinde $\alpha, \beta \in [-0.76376453, 0.76376453]$ elde edilir. α ve β değerleri bu aralıklardan seçilirse (4.8) sistemi ω^* – Schur kararlıdır.

Tablo 4.10’ de α ve β elemanlarına müdahale edilerek bulunan Schur kararlılık aralıkları özetlenmiştir.

	Önce α Sonra β	
$\mathfrak{T}^{*e} = [l^{*e}, u^{*e}]$	$[-0.91264831, 0.91264831]$	$[-0.03337832, 0.03337832]$
	Önce β Sonra α	
$\mathfrak{T}^{*e} = [l^{*e}, u^{*e}]$	$[-0.03337832, 0.03337832]$	$[-0.91264831, 0.91264831]$
	α, β aynı anda	
$\mathfrak{T}^{*e} = [l^{*e}, u^{*e}]$	$[-0.76376453, 0.76376453]$	$[-0.76376453, 0.76376453]$

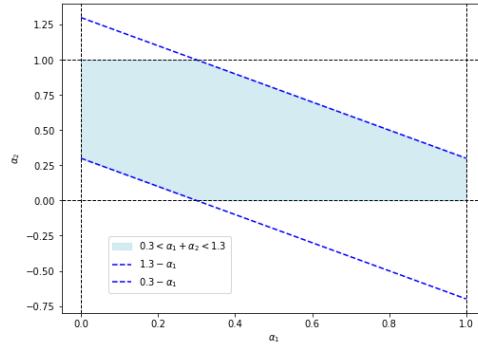
Tablo 4.10: α ve β elemanlarına müdahale edilerek bulunan $\omega^* = 10$ için Schur kararlılık aralıkları

4.4. Değerlendirme

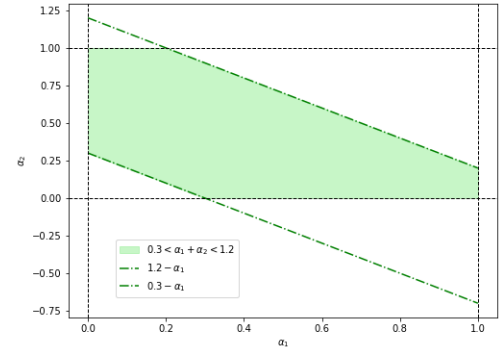
Matematiksel yöntemlerle analizler yaptığımızda bu analizlerin bir kısmı ekonomik olarak anlamsız olabilmektedir. α parametresine bağlı (4.6) dinamik sisteminin kararlılığı (istikrarı) $\alpha \in [-0.982, 0.899]$ aralığından değerler aldığı sürece

bozulmamaktadır. Ancak bu aralık problemin yapısına uygun olarak ekonomik verilerle örtüşmeyen durumlar oluşabilir. Şöyle ki;

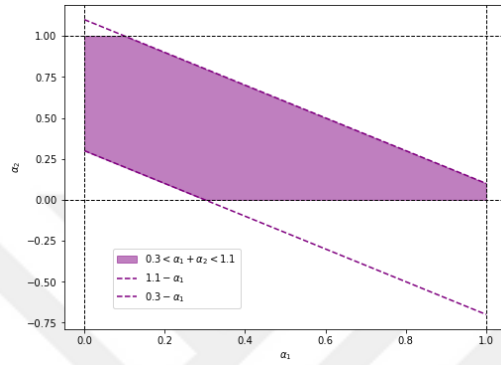
- $\alpha = 0$ için (4.6) sistemi kararlı iken ekonomik olarak modelin yorumu pratikte “1. ekonominin gelirlerinin tamamen 2. ekonominin ithalatıyla oluştuğu” anlamına gelmektedir.
- $\alpha \in [-0.982, 0)$ için (4.6) sistemi kararlı olmasına rağmen ekonomik olarak modelin yorumu pratikte “1. ekonominin gelirlerinin, sadece 2. ekonominin ithalatına bağlı kalmayıp aynı zamanda 1. ekonomin gelirlerinin bir kısmının da kaybolmasıyla – yitirilmesiyle (gerekli olmayan sebepler, non-essential reason), yani yatırımın azalmasına karşın ithalatın daha da artmasıyla oluştuğu” anlamına gelir. Yani kısaca bu durum, tüketim ve yatırım oranlarının toplamının ithalat oranından az olması anlamına gelir ki sürdürülebilir bir durum olmadığı anlaşılmaktadır.
- $\alpha \in (0, 0.899]$ olması durumunda ise, (4.6) sistemi yine kararlı, ancak α değerinin aralığın uç noktalarına göreceli yakın değerler alması da benzer yaklaşımlarla sürdürülebilir bir ekonomik yapı olmadığı anlamına gelmektedir.
- Aşağıda Şekil 4.10 (a) da parametreler için modelde verilen özel kısıtlamalar ile α_1 ve α_2 parametrelerinin olması gereken bölge gösterilmiştir. $\alpha = \alpha_1 + \alpha_2 - \alpha_3$ ve $0 < \alpha < 1$ olması gerektiğinden $0 < \alpha_1 + \alpha_2 - \alpha_3 < 1$ dır. İncelenen problemde $\alpha_3 = 0.3$ alındığından $0.3 < \alpha_1 + \alpha_2 < 1.3$ olmalıdır. Şekil 4.10 (b) de $\text{üstsınır}(A, B, ry)$ fonksiyonu (Yordam 3.1) kullanılarak bulunan $u^e = 0.899$ kullanılırsa, α_1 ve α_2 parametrelerinin olması gereken bölge gösterilmiştir. Yani $0 < \alpha_1 + \alpha_2 - 0.3 < 0.9$ eşitsizliğinden $0.3 < \alpha_1 + \alpha_2 < 1.2$ aralığı bulunur. Şekil 4.10 (c) de pratik Schur kararlılık parametresi kullanılarak oluşturulan $\text{pr_üstsınır}(A, B, \omega^*)$ fonksiyonu (Yordam 3.3) ile bulunan pratik üst sınır $u^{*e} = 0.8$ kullanılırsa, α_1 ve α_2 parametrelerinin olması gereken bölge gösterilmiştir. Yani $0 < \alpha_1 + \alpha_2 - 0.3 < 0.8$ eşitsizliğinden $0.3 < \alpha_1 + \alpha_2 < 1.1$ aralığı bulunur. Şekil 4.10 (d) de bu aralıklar ile oluşturulan bölgelerin üst üste gösterimi verilmiştir. Buradan $0 < \alpha < 1$ olması problemin kararlı olmasını garantilemez ancak $0 < \alpha < 0.9$ olması sistemin Schur kararlı olmasını garanti eder. Benzer şekilde $0 < \alpha < 0.8$ aralığında olması da pratik Schur kararlı olmasını garanti eder.



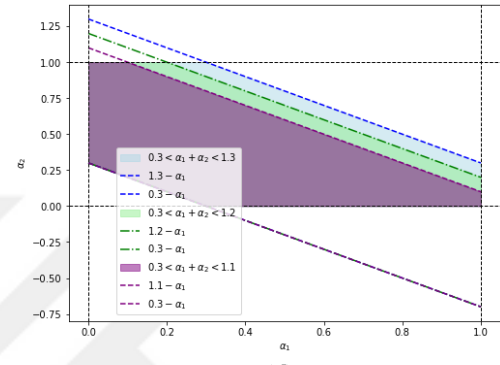
(a)



(b)



(c)



(d)

Şekil 4.10: (4.6) sisteminin parametreleri Schur kararlılık, pratik Schur kararlılık için bulunan aralıklar ile oluşturulan bölgeler

Başlangıç kısmında modelin parametreleri için verilen aralıklar göz önüne alındığında, verilen yöntemlerle bulunan Schur kararlılık ve pratik Schur kararlılık aralıkları uyumludur.

5. LOTKA-VOLTERRA MATEMATİKSEL MODELİNİN KARARLILIĞININ PARAMETRİK ANALİZİ

Lotka-Volterra modeli, ekolojide türler arası etkileşimleri anlamak ve nüfus dinamiklerini matematiksel olarak incelemek için geliştirilmiş temel bir çerçeve sunar. 1920'lerde Alfred Lotka ve Vito Volterra tarafından ortaya konan bu model, avcı-av ilişkileri ve rekabet dinamiklerini diferensiyel denklemlerle ifade ederek ekosistemlerin karmaşık davranışlarını sadeleştirilmiş bir şekilde analiz etme olanağı sağlar. Modelin analizi, biyolojik sistemlerin dengelerini, türlerin bir arada var olma koşullarını ve nüfusun zamanla nasıl değiştiğini anlamak açısından kritik bir öneme sahiptir. Özellikle, türlerin birlikte var olma veya rekabet dışı bırakma gibi ekolojik sonuçları öngörebilme yeteneği, bu modelin teorik ve uygulamalı ekoloji çalışmalarında önemli rol oynar (Hassel, 2000; Strogatz, 2015; Chern 2016).

Analizin önemi, yalnızca teorik bir çalışma ile sınırlı kalmaz. Avcı ve av nüfuslarının osilasyonları, türlerin birbirine bağımlı dinamiklerini ortaya koyarken, rekabet modelleri sınırlı kaynakların paylaşıldığı ekosistemlerde türlerin hayatta kalma stratejilerini aydınlatır. Ayrıca, modelin denge noktaları ve kararlılık analizi, ekosistemlerin dışsal şoklara (iklim değişikliği, habitat kaybı) karşı direncini değerlendirme imkanı sunar. Bu bağlamda, Lotka-Volterra modelinin kapsamlı analizi, ekolojik sistemlerin temel dinamiklerini anlamak ve gelecekteki nüfusu tahmin etmek için bir araçtır (Lotka, 1920; Volterra, 1939). Bu bölümde, Lotka-Volterra modelinin Hurwitz kararlılığı incelenmiştir. Parametrelerin biyolojik anlamlarını ve ekosistem dinamiklerine etkilerini, Hurwitz kararlılık yöntemleri ve κ kararlılık parametresi kullanılarak detaylı bir şekilde incelemeyi amaçlamaktadır.

$$\begin{cases} \frac{dx}{dt} = r_1 x \left(1 - \frac{x}{K_1} - \alpha_1 \frac{y}{K_1} \right) \\ \frac{dy}{dt} = r_2 y \left(1 - \frac{y}{K_2} - \alpha_2 \frac{x}{K_2} \right) \end{cases} \quad (5.1)$$

- $x(t)$, birinci türün nüfusunu (örneğin tavşanlar); $y(t)$, ikinci türün nüfusunu (örneğin koyunlar); $\frac{dx}{dt}, \frac{dy}{dt}$ nüfusların zamanla değişimlerini temsil etmektedir.
- r_1, r_2 katsayıları türlerin üreme hızını gösteren parametrelerdir. Örneğin, r_1 tavşanların, r_2 koyunların üreme hızını yansıtır. $r_1 = 3$ ise tavşanlar uygun

koşullarda birim zamanda nüfuslarını 3 kat artırabilir. Bu, türün biyolojik üreme potansiyelini gösterir. Örneğin, $r_2 = 2$ ise koyunlar birim zamanda nüfuslarını 2 kat artırabilir. Koyunların daha yavaş ürediği bir senaryoyu yansıtır. Tavşanlar gibi hızlı üreyen türler için r_1 yüksek, koyunlar gibi daha yavaş üreyen türler için r_2 daha düşük olur.*

- K_1, K_2 türlerin taşıma kapasitelerini gösterir. Yani türlerin çevresel kaynaklarla destekleyebileceği maksimum nüfus büyüklüğüdür. Örneğin, tavşanlar yalnız olsaydı ve koyunlarla rekabet etmeseydi, nüfusları K_1 ' e kadar büyürdü. Koyunlar yalnız olsaydı, nüfusları K_2 ' ye kadar büyürdü. *
- a_1 , ikinci türün birinci tür üzerindeki rekabet katsayısıdır. Benzer mantıkla a_2 , birinci türün ikinci tür üzerindeki rekabet katsayısıdır.
- $r_1, r_2, K_1, K_2 > 0$ olmalıdır, çünkü negatif veya sıfır değerler biyolojik olarak anlamsızdır (popülasyonun doğal olarak azalması veya sıfır büyüme oranı dışsal faktörlerle modellenir). *
- r_1, r_2 genellikle $[0.1, 3]$ arasında olur, çünkü memeliler gibi türlerin üreme hızları bu aralıkta makuldür. *
- a_1, a_2 türlerin rekabet dinamiklerine bağlı olarak $[0.5, 3]$ arasında olmalıdır. Ancak aşırı durumlarda 5' e kadar çıkabilir. *

Bu bölümde (5.1) sisteminin Strogatz' ın “Nonlinear Dynamics And Chaos” adlı kitabındaki $\alpha_1, \alpha_2, r_1, r_2, K_1, K_2$ özel değerleri için ele alınan sistemin Hurwitz kararlılığı incelenecektir. Şimdi bu özel sistemi verelim,

$$\alpha_1 = 2, \alpha_2 = 1, r_1 = 3, r_2 = 2, K_1 = 3, K_2 = 2$$

değerleri için

$$\begin{cases} \frac{dx}{dt} = 3x - x^2 - 2xy \\ \frac{dy}{dt} = 2y - y^2 - xy \end{cases} \quad (5.2)$$

* Bu veriler basılı kaynaklardan teyit edilememiş yapay zekanın Grok 3 platformu kullanılarak bulunmuştur.

sistemi elde edilir.

(5.2) sisteminde x değişkeninin katsayısını α şeklinde parametrik olarak alalım. Bu durumda (5.2) dinamik sistemi,

$$\begin{cases} \frac{dx}{dt} = \alpha x - x^2 - 2xy \\ \frac{dy}{dt} = 2y - y^2 - xy \end{cases} \quad (5.3)$$

şekline dönüşür. Elde edilen bu modelin kararlılık analizini yapalım.

5.1. (5.3) Lotka-Volterra Modelinin Hurwitz Kararlılık Analizi

Şimdi (5.3) sürekli dinamik sisteminin kararlılığını α belirsizine göre analiz edelim. (5.3) sisteminin denge noktaları için fonksiyonlar

$$\begin{cases} f_1(x_n, y_n) = \alpha x_n + x_n^2 - 2x_n y_n \\ f_2(x_n, y_n) = 2y_n - y_n^2 - x_n y_n \end{cases}$$

olmak üzere $f_1(x^*, y^*) = x^*$ ve $f_2(x^*, y^*) = y^*$ dan denge noktaları $(0,0), (0, \alpha), (3,0), (2\alpha - 3, 3 - \alpha)$ şeklinde bulunur. (5.3) sisteminin 2.4 bölümündeki lineerleştirme yöntemiyle elde edilen $(3,0)$ denge noktasında ki Jakobiyen matrisi

$$J(x^*, y^*) = \begin{pmatrix} -3 & -6 \\ 0 & \alpha - 3 \end{pmatrix} \quad (5.4)$$

şeklindedir. Teorem 2.5 ve Teorem 2.7' den

$$J(\alpha) = \begin{pmatrix} -3 & -6 \\ 0 & \alpha - 3 \end{pmatrix} = \begin{pmatrix} -3 & -6 \\ 0 & -3 \end{pmatrix} + \alpha \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = A + \alpha E_{22} \quad (5.5)$$

şeklinde düzenlenirse (2.20) ile verilen Lyapunov matris denkleminde

$$H = \begin{pmatrix} 0.5 & -0.16666667 \\ -0.16666667 & 0.16666667 \end{pmatrix}$$

matrisi bulunur. $H = H^* > 0$ olduğundan A matrisi (Teorem 2.5' e göre) Hurwitz kararlıdır. (5.5) eşitliğinden

$$\|A\| = 7.24264, \quad \kappa(A) = 8.24264, \quad \Delta = \alpha E_{22}, \quad \|\Delta\| = |\alpha|$$

hesaplanır. Ve

$$\|B\| < \frac{\|A\|}{\kappa(A)}$$

eşitliğinden (Teorem 2.7' den)

$$|\alpha| < 0.87867$$

eşitsizliğine ulaşılır. Yani $\alpha \in (-0.87867, 0.87867)$ elde edilir. Bu aralık Hurwitz kararlılığın sağlandığı güvenli bir aralık vermektedir. Ancak bu aralık dışında Hurwitz kararlı sistemler olması da mümkündür. Şimdi bulunan bu aralığı kararlılık özellikleri korunacak şekilde genişletelim.

A	B	r^*	κ^*	R	u	u^e	M	S.P.
$\begin{pmatrix} -3 & -6 \\ 0 & -3 \end{pmatrix}$	E_{22}	0.1	50	50	0.7908117	2.5291997	7	κ^*
			100			2.5291997	7	r^*
		0.01	500	100	0.7908117	2.9426830	17	κ^*
			1000			2.9531566	18	r^*
		0.001	50	200	0.7908117	2.5291997	7	κ^*
			120			2.7565879	10	κ^*
			10000			2.9947907	29	r^*

Tablo 5.1: (5.5) sistemi için farklı r^*, κ^* - pratiklik parametreleri ile durulan adım ve genişletilmiş üst sınır değerleri.

A	B	r^*	κ^*	R	l	l^e	M	S.P.
$\begin{pmatrix} -3 & -6 \\ 0 & -3 \end{pmatrix}$	E_{22}	0.1	50	50	-0.7908117	-148.6912637	59	κ^*
			100			-297.0740574	114	
		0.01	500	100	-0.7908117	-1498.4781169	559	κ^*
			1000			-2999.6661316	1115	
		0.001	50	200	-0.7908117	-148.6912637	59	κ^*
			120			-359.1535094	137	
			450			-1347.2808011	503	

Tablo 5.2: (5.5) sistemi için farklı r^*, κ^* - pratiklik parametreleri ile durulan adım ve genişletilmiş alt sınır değerleri.

Buna göre $r^* = 0.001$ pratik adım genişliği, $\kappa^* = 450, \kappa^* = 1000$ pratik Hurwitz parametresi ve $R = 200$ parametreleri kullanılarak, genişletilmiş üst sınır (Tablo 5.1) 29 adımda r^* değerine göre $u^e = 2.9947907$, genişletilmiş alt sınır (Tablo 5.2) 503 adımda κ^* a göre $l^e = -1347.2808011$ elde edilir. Sonuç olarak $\text{üstsınır}(A, B, ry)$ fonksiyonu (Yordam 3.1) ve $\text{altsınır}(A, B, ry)$ fonksiyonu (Yordam 3.2) kullanılarak

$$\mathfrak{T}^e = [l^e, u^e] = [-1347.2808011, 2.9947907]$$

aralığı elde edilir. α bu aralıktan seçildiği sürece (5.5) sistemi Hurwitz kararlı bir sistemdir. Şimdi elde ettiğimiz bu sonucu deneyelim.

5.1.2. Genişletilmiş aralık için etkinlik analizi (kalite kontrolü)

Bu kısımda 5.1 bölümünde bulunan genişletilmiş Hurwitz kararlı aralığın etkinlik analizi çalışılacaktır.

5.1.2.1. Üst sınır için etkinlik analizi

$r^* = 0.001$ pratik adım genişliği, $\kappa^* = 450, \kappa^* = 1000$ pratik Hurwitz parametreleri ve $R = 200$ parametresine karşılık $\text{üstsınır}(A, B, ry)$, $\text{altsınır}(A, B, ry)$ fonksiyonları yardımıyla elde edilen $[l^e, u^e] = [-1347.2808011, 2.9947907]$ aralığının etkinlik analizini yapalım. α parametresine bağlı Jakobiye matris

$$J(\alpha) = \begin{pmatrix} -3 & -6 \\ 0 & a - 3 \end{pmatrix}$$

olmak üzere $u^e = 2.9947907$ üst sınırı için bu değer civarında κ_i değerleri Tablo 5.3'te verilmiştir. Tablodan da açıkça görüldüğü üzere aralığın uç noktasına yaklaştıkça sistemin Hurwitz kararlılığının kalitesini belirleyen parametre büyümektedir, yani kalitesi ve sistemin dış etkilere karşı dayanıklılığı azalmaktadır.

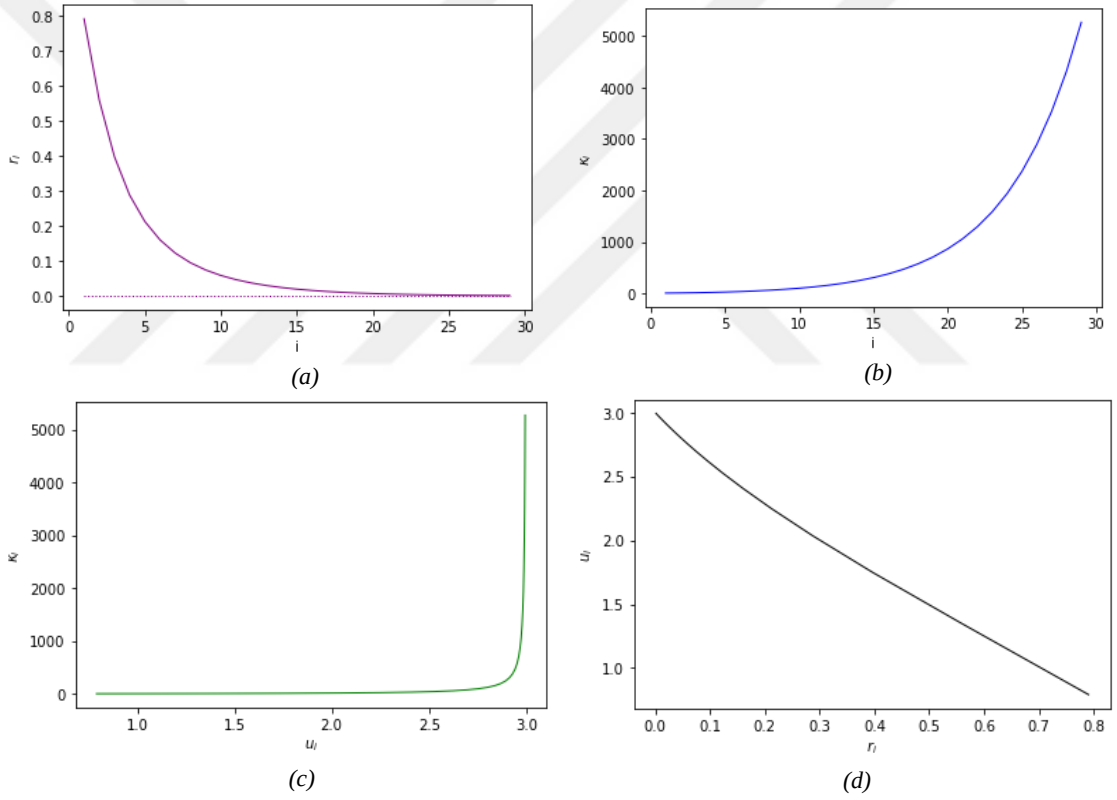
i	α_i	κ_i
1	2.99	3345.1963260744837
2	2.991	3717.870470526098
3	2.992	4183.7141115192235
4	2.993	4782.657034363783
⋮		
9	2.998	16761.57330595775
10	2.999	33532.07926517006
11	2.9991	37258.85884031739

Tablo 5.3: $\kappa(J(\alpha_i)) = \kappa_i$, genişletilen üst sınır için etkinlik analizi.

Yordam 3.1 kullanılarak oluşturulan Tablo 5.4' den de görüldüğü üzere her adımda sınır genişletilmiş ve 2.99479065 değerinden sonra yavaş hareket etmiştir. κ değeri her adımda daha çok büyümüş kararlılığın kalitesi düşmüştür. Yani kararlılık 2.99479065 civarında bozulmaya daha yakındır. Şekil 5.1 (a)' dan anlaşılabacağı üzere i değerleri arttıkça r değerleri azalmaktadır. Benzer şekilde Şekil 5.1 (b)' den i değerleri arttıkça κ_i değeri artmakta ve Şekil 5.1 (c)' den u_i değerleri sınıra yaklaştıkça κ_i değeri artmaktadır.

i	r_i	u_i^e	κ_i
1	0.7908116907963215	0.7908116907963215	8.242640687119286
2	0.5587313459200185	1.34954303671634	11.273679656810186
3	0.39733711945426303	1.7468801561706029	15.562371058144981
$\vdots$			
27	0.0017072824930580409	2.992246219414317	3536.2562244389337
28	0.0013985633338642028	2.993644782748181	4316.849026068605
29	0.0011458761236270677	2.9947906588718083	5268.794403036237
30	0.0009389830852907522	--	--

Tablo 5.4: $r^* = 0.001$ hassasiyetinde üst sınır genişletilirken r, u ve κ parametrelerinin değişimi.



Şekil 5.1: Tablo 5.4' deki hesaplamalar sonucu (a) r_i ' nin i 'e göre grafiği, (b) κ_i ' nin i 'e göre grafiği, (c) κ_i ' in u_i ' e göre grafiği, (d) u_i 'in r_i ' e göre grafiği,

5.1.2.2. Alt sınır için etkinlik analizi

$r^* = 0.001$ pratik adım genişliği, $\kappa^* = 450, \kappa^* = 1000$ pratik Hurwitz parametreleri ve $R = 200$ parametresine karşılık $\text{üstsinir}(A, B, ry)$, $\text{altsinir}(A, B, ry)$ fonksiyonları yardımıyla elde edilen $[l^e, u^e] = [-1347.2808011, 2.9947907]$ aralığının etkinlik analizini yapalım. a parametresine bağlı Jakobiye matris

$$J(\alpha) = \begin{pmatrix} -3 & -6 \\ 0 & a-3 \end{pmatrix}$$

olmak üzere $l^e = -1347.2808011$ alt sınırı için bu değer civarında κ_i değerleri Tablo 5.5' te verilmiştir. Tablodan da açıkça görüldüğü üzere aralığın uç noktasına yaklaştıkça sistemin Hurwitz kararlılığının kalitesini belirleyen parametre büyümektedir, yani kalitesi ve sistemin dış etkilere karşı dayanıklılığı azalmaktadır.

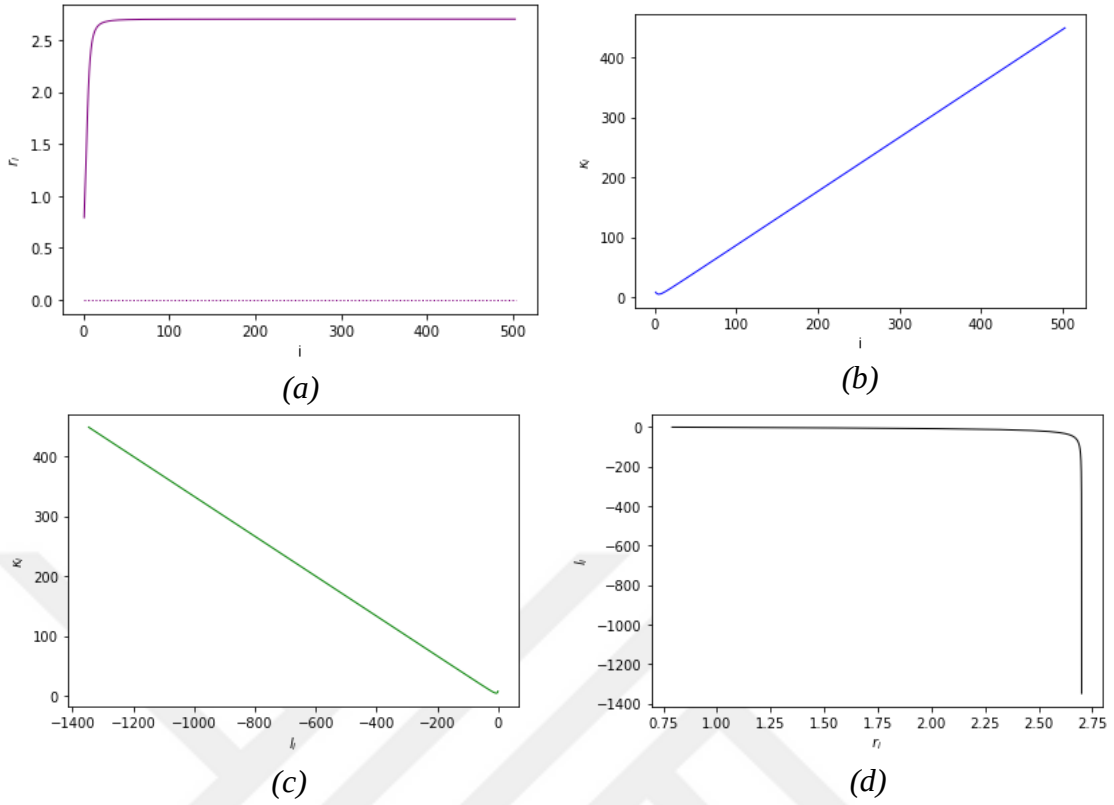
i	α_i	κ_i
1	-1347	450.0133138
2	-1348	450.3466372
3	-1349	450.6799608
4	-1350	451.0132843
⋮		
9	-1355	452.6799021
10	-1356	453.0132257
11	-1357	453.3465493

Tablo 5.5: $\kappa(J(\alpha)) = \kappa_i$, genişletilen alt sınır için etkinlik analizi.

Yordam 3.2 kullanılarak oluşturulan Tablo 5.6' den de görüldüğü üzere her adımda sınır genişletilmiş ve -1347.2808011 değerinden sonra yavaş hareket etmiştir. κ değeri her adımda daha çok büyümüş kararlılığın kalitesi düşmüştür. Yani kararlılık -1347.2808011 civarında bozulmaya daha yakındır. Şekil 5.2 (a)' dan anlaşılaçağı üzere i değerleri arttıkça r değerleri artmaktadır. Benzer şekilde Şekil 5.2 (b)' den i değerleri arttıkça κ_i değeri artmaktadır. Şekil 5.2 (c)' den l_i değerleri sınıra yaklaştıkça κ_i değeri artmaktadır.

i	r_i	l_i^e	κ_i
1	0.7908116907963215	-0.7908116907963215	8.242640687119286
2	1.0150029070531146	-1.805814597849436	6.700605127843519
3	1.2770867328208355	-3.0829013306702713	5.677013979670831
⋮			
501	2.6999461645944995	-1341.880908084638	447.40704517599835
502	2.6999463802912236	-1344.5808544649292	448.30700038637633
503	2.6999465946942487	-1347.2808010596234	449.2069557760639
504	--	--	450.1069113439863

Tablo 5.6: $r^* = 0.001$ hassasiyetinde alt sınır genişletilirken r, u, κ parametrelerinin değişimi.



Şekil 5.2: Tablo 5.6’ daki hesaplamalar sonucu (a) r_i ’ nin i ’e göre grafiği, (b) κ_i ’ nin i ’e göre grafiği, (c) κ_i ’ in l_i ’ e göre grafiği, (d) l_i ’ in r_i ’ e göre grafiği,

5.2. (5.3) Lotka-Volterra Modelinin κ^* – Hurwitz Kararlılık Analizi

Bu bölümde α parametresine bağlı (5.3) sisteminin κ^* – Hurwitz kararlılığı incelenmiştir. (5.5) eşitliğinden

$$\|A\| = 7.24264, \quad \kappa(A) = 8.24264, \quad \Delta = \alpha E_{22}, \quad \|\Delta\| = |\alpha|$$

hesaplanır ve

$$\|B\| \leq \frac{(\kappa^* - \kappa(A))\|A\|}{(1 + \kappa^*) \kappa(A)}$$

eşitliğinden (Teorem 2.8’ den) $\kappa^* = 10$ pratik değeri için

$$|\alpha| < 0.14037$$

$\kappa^* = 100$ pratik değeri için

$$|\alpha| < 0.79827$$

eşitsizliklerine ulaşılır. Teorem pratik Hurwitz kararlılığın sağlandığı güvenli bir aralık vermektedir. Buradan görüldüğü üzere bulunan pratik Hurwitz kararlı aralık daha önce

bulduğumuz Hurwitz kararlık aralığının içinde kalmaktadır. Şimdi κ^* – Hurwitz kararlı olduğunu bildiğimiz aralığı $pr_üstsınır(A, B, \kappa^*)$, $pr_altsınır(A, B, \kappa^*)$ fonksiyonları yardımı ile r^* ve κ^* değerleri olarak genişletelim.

A	B	κ^*	r^*	u^*	u^{*e}	M
$\begin{pmatrix} -3 & -6 \\ 0 & -3 \end{pmatrix}$	E_{22}	10	0.1	0.1403778	0.2437967	2
			0.01		0.4949486	9
			0.001		0.5227437	16
		100	0.1	0.7982705	2.2897305	6
			0.01		2.6535063	15
			0.001		2.6855905	24

Tablo 5.7: Farklı r^* , κ^* parametreleri ile durulan adım ve genişletilmiş üst sınır değerleri.

A	B	κ^*	r^*	l^*	l^{*e}	M
$\begin{pmatrix} -3 & -6 \\ 0 & -3 \end{pmatrix}$	E_{22}	10	0.1	-0.1403778	-23.9491564	47
			0.01		-25.0543648	75
			0.001		-25.1556989	102
		100	0.1	-0.7982705	-286.7878871	345
			0.01		-295.8113018	576
			0.001		-296.72043400	808

Tablo 5.8: Farklı r^* , κ^* parametreleri ile durulan adım ve genişletilmiş alt sınır değerleri.

Buna göre Tablo 5.7 ve Tablo 5.8' den, $r^* = 0.001$ ve $\kappa^* = 10$ pratiklik parametreleri kullanılarak genişletilmiş üst sınır 16 adımda $u^{*e} = 0.5227437$, genişletilmiş alt sınır 102 adımda $l^{*e} = -25.1556989$ şeklinde elde edilir. Diğer taraftan $r^* = 0.001$, $\kappa^* = 100$ pratiklik parametreleri kullanılarak, genişletilmiş üst sınır 24 adımda $u^{*e} = 2.6855905$ ve genişletilmiş alt sınır 808 adımda $l^{*e} = -296.72043400$ şeklinde elde edilir. Sonuç olarak $pr_üstsınır(A, B, ky)$ fonksiyonu (Yordam 3.3) ve $pr_altsınır(A, B, ky)$ fonksiyonu (Yordam 3.4) kullanılarak $r^* = 0.001, \kappa^* = 10$ için

$$\mathfrak{T}^{*e} = [l^e, u^e] = [-25.1556989, 0.5227437]$$

aralığı elde edilir ve α değerleri bu aralıktan seçildiği sürece (5.3) sistemi $\kappa^* = 10$ – Hurwitz kararlı bir sistemdir. Benzer şekilde $r^* = 0.001, \kappa^* = 100$ için

$$\mathfrak{T}^{*e} = [l^e, u^e] = [-296.72043400, 2.6855905]$$

aralığı elde edilir ve α değerleri bu aralıktan seçildiği sürece (5.3) sistemi $\kappa^* = 100$ – Hurwitz kararlı bir sistemdir.

5.3. Değerlendirme

Literatürde var olan Lotka – Volterra problemi için kararlılığı belli olan (3,0) denge noktası esas alınarak (5.3) sistemindeki α parametresi ne kadar değiştirilirse, Lotka–Volterra problemi yinede Hurwitz kararlı kalır sorusu üzerine değerlendirmeler yapılmıştır. Yapılan analizler sonucu

$$\mathfrak{I}^e = [l^e, u^e] = [-1347.2808011, 2.9947907]$$

aralığı elde edilmiştir.

- $\alpha = 0$ için x nüfusun, kendi kendini sınırlar ve y ile rekabet nedeniyle azalır. Başlangıçta $x > 0$ olduğu düşünülürse zamanla $0'$ a yaklaşır, çünkü büyüme kaynağı yoktur. Dolayısıyla $\alpha = 0$ olduğunda x nüfusu uzun vadede yok olma eğilimindedir.
- $\alpha \in [-1347.2808011, 0)$ değerleri x 'in doğal olarak azalmasına neden olur. Bu durum x türünün kendi başına hayatta kalamayacağını gösterir. Rekabet ve kendi kendini sınırlama ile birlikte x nüfusu hızla sıfıra iner, bu da ekolojik olarak x ' in bu ortamda sürdürülemez olduğunu ima eder.
- $\alpha < 0$ teorik olarak mümkündür ancak tipik rekabet modellerinde $\alpha > 0$ varsayılır.

Başlangıç kısmında modelin parametreleri için verilen aralıklar göz önüne alındığında verilen yöntemlerle bulunan Hurwitz kararlılık ve κ^* – Hurwitz kararlılık aralıkları uyumludur.

6. SIR MATEMATİKSEL MODELİNİN KARARLILIĞININ PARAMETRİK ANALİZİ

Salgın hastalık modelleri, hastalık dinamiklerini analiz etmede, insan sağlığını etkileyen salgın hastalıkların önlenmesinde ve ölümlerin azaltılmasında büyük rol oynamaktadır. İncelenen model, salgının büyüklüğüne veya etkisine göre salgının seyri hakkında doğru tedavi yöntemlerine sahip olmamızı sağlar. Enfekte (hastalık bulaşmış birey) olmuş kişilerin sayısını, iyileşen kişilerin sayısını tahmin etmek bize hastalığın yayılma hızını ve etkisini dolayısıyla insan sağlığı için salgının sonucunu nasıl etkileyebileceğini gösterebilir. Nüfusu etkileyen salgın hastalıkları tanımlayan matematiksel modeller genellikle diferensiyel denklemler veya fark denklemler ile ifade edilir. Diferensiyel denklem modelleri, değişimin sürekli olduğu durumları tanımlamak için kullanılır. Değişim kesikli ise, modelleme için fark denklemlerinin kullanılması daha uygun olacaktır (Wang, 2006; Brauer F. Ve ark., 2008 Chen ve ark., 2020; Mata ve Dourado, 2021).

$$\begin{cases} \frac{dS}{dt} = \Lambda - \beta SI - dS \\ \frac{dI}{dt} = \beta SI - (d + \gamma)I - h(I) \\ \frac{dR}{dt} = \gamma I + h(I) - dR \end{cases} \quad (6.1)$$

(6.1) ile verilen modelde,

- S - hastalığa duyarlı kişiler, I - hastalığa yakalanmış ve başkalarına bulaştıran enfekte kişiler, R - hastalığa yakalanmış ancak iyileşmiş kişiler,
- $h(I) = \begin{cases} m(> 0) & I > 0 \\ 0 & I = 0 \end{cases}$ - enfekte kişilerin tedavi sonucu iyileşme oranı,
- $\Lambda, d, \gamma, \lambda$ pozitif parametreler,
- Λ - doğum ile nüfusa eklenen kişiler, d - nüfustaki doğal ölüm oranı, γ - enfekte kişilerin doğal iyileşme oranı, β - hastalığın bulaşma oranıdır (enfeksiyon katsayısı) .

Wendi Wang, “Backward bifurcation of an epidemic model with treatment” adlı çalışmasında (6.1) sistemi için kararlılık analizi çalışmıştır.

Hui Cao ve ark. “Bifurcation analysis of a discrete SIR epidemic model with constant recovery” adlı çalışmalarında ise (6.1) sistemini İleri Euler yöntemi ile elde edilen

$$\begin{cases} S_{n+1} = S_n + \Lambda - \beta S_n I_n - d S_n \\ I_{n+1} = I_n + \beta S_n I_n - (d + \gamma) I_n - h(I_n) \\ R_{n+1} = R_n + \gamma I_n + h(I_n) - d R_n \end{cases} \quad (6.2)$$

kesikli zaman modelinin kararlılık analizi üzerine çalışmışlardır. Bu bölümde (6.2) modeli ele alınmıştır. Denge noktasının kararlı olduğu yerlerde yine hangi şartlarda kararlı kalır sorusuna cevap aranmaktadır.

(6.2) ile verilen modelde,

- $n \in \mathbb{N}$ ve S - hastalığa duyarlı kişiler, I - hastalığa yakalanmış ve başkalarına bulaştıran enfekte kişiler, R - hastalığa yakalanmış ancak iyileşmiş kişiler, $h(I_n) = m \geq 0$ - enfekte kişilerin tedavi sonucu iyileşme oranıdır.
- $\Lambda, d, \gamma, \lambda$ pozitif parametrelerdir.
- Λ - doğum ile nüfusa eklenen kişiler, d - nüfustaki doğal ölüm oranı, γ - enfekte kişilerin doğal iyileşme oranı, β - hastalığın bulaşma oranıdır (enfeksiyon katsayısı).
- $0 < d < 1$ ve $0 < d + \gamma < 1$ olması gerekir.

(6.2) ile verilen sistem, ilk iki denklemi üçüncü denklemden bağımsız olduğundan ve ilk iki denklem yeterli olacağından,

$$\begin{cases} S_{n+1} = S_n + \Lambda - \beta S_n I_n - d S_n \\ I_{n+1} = I_n + \beta S_n I_n - (d + \gamma) I_n - m \end{cases} \quad (6.3)$$

sistemine indirgenir.

- $I_n = 0$ ise yani enfekte kişi yoksa $m = 0$,
- $S_n = 0$ ise yani hastalığa duyarlı kişi yoksa $\Lambda = 0$

olur.

6.1. SIR Modelin Schur Kararlılığı

(6.3) sisteminin denge noktaları için fonksiyonlar

$$\begin{aligned} f_1(S_n, I_n) &= S_n + \Lambda - \beta S_n I_n - d S_n \\ f_2(S_n, I_n) &= I_n + \beta S_n I_n - (d + \gamma) I_n - m \end{aligned}$$

olmak üzere $f_1(S^*, I^*) = S^*$ ve $f_2(S^*, I^*) = I^*$ dan

$$S^* = \frac{\Lambda}{\beta I^* + d}, \quad I^* = \frac{m}{\beta S^* - d - \gamma}$$

şeklinde bulunur.

(6.3) sisteminin 2.4 bölümünde verilen lineerleştirme yöntemiyle elde edilen Jakobiyen matrisi

$$J(S, I) = \begin{pmatrix} 1 - \beta I - d & -\beta S \\ \beta I & 1 + \beta S - (d + \gamma) \end{pmatrix} \quad (6.4)$$

şeklindedir.

Cao ve arkadaşlarının 2020' de yayımlanan çalışmasında ele alınan SIR modelinde kullanılan

$$\Lambda = 200, \beta = 0.0011308665, d = 0.07, \gamma = 0.1, m = 0.1$$

parametrelerine göre denge noktası $(S^*, I^*) \cong (150.4154, 1113.9465)$ şeklinde bulunur. İlk olarak $S^* = s, I^* = 1113.9465$, ikinci olarak $S^* = 150.4154, I^* = \iota$ olarak ve

$$\Lambda = 200, \beta = 0.0011308665, d = 0.07, \gamma = 0.1, m = 0.1$$

parametreleri için (6.3) sisteminin parametrelendirilmiş denge noktalarında kararlılık analizi yapalım. Yani, sırasıyla

$$\varphi(n+1) = J(s, I^*)\varphi(n) \quad (6.5)$$

$$\varphi(n+1) = J(S^*, \iota)\varphi(n) \quad (6.6)$$

sistemlerinin kararlılık analizlerini inceleyelim.

6.1.1 (6.5) sisteminin s parametresine göre Schur kararlılığı

(6.5) sisteminin s parametresine bağlı Jakobiyen matrisi

$$J(s, I^*)|_{\substack{I=1113.9465 \\ S=s}} = \begin{pmatrix} 0.93 - (0.0011308)(1113.9465) & -0.0011308s \\ (0.0011308)(1113.9465) & 0.0011308s + 0.83 \end{pmatrix}$$

şeklinde elde edilir. $J(s, I^*)$ matrisi Teorem 2.1 ve Teorem 2.3' den

$$\begin{aligned} J(s, I^*) &= \begin{pmatrix} 0.93 - (0.0011308)(1113.9465) & -0.0011308s \\ (0.0011308)(1113.9465) & 0.0011308s + 0.83 \end{pmatrix} \\ &= \begin{pmatrix} -0.32965 & 0 \\ 1.25965 & 0.83 \end{pmatrix} + s \begin{pmatrix} 0 & -0.0011308 \\ 0 & 0.0011308 \end{pmatrix} = A + sB \end{aligned} \quad (6.7)$$

şeklinde düzenlenirse (2.17) ile verilen Lyapunov fark matris denkleminin çözümü

$$H = \begin{pmatrix} 1.12191 & -0.36578 \\ -0.36578 & 6.47798 \end{pmatrix}$$

şeklinde bulunur. $H = H^* > 0$ olduğundan A matrisi (Teorem 2.1' e göre) Schur kararlıdır. (6.7) eşitliğinden

$$\|A\| = 1.533774, \omega(A) = 6.502848, \Delta = sB, \|\Delta\| = 0.00159919|s|$$

değerleri hesaplanır ve

$$\|B\| < \sqrt{\|A\|^2 + \frac{1}{\omega(A)}} - \|A\|$$

eşitsizliğinden (Teorem 2.3' den)

$$|s| < 30.8513$$

eşitsizliğine ulaşılır. Yani $s \in (-30.8513, 30.8513)$ elde edilir. Bu aralık Schur kararlılığın sağlandığı güvenli bir aralık olmasına rağmen. Ancak bu aralık dışında Schur kararlı sistemlerin olduğuda bilinmektedir. Elde edilen bu aralığın kararlılık özellikleri korunacak şekilde genişletelim.

A	B	r^*	u	u^e	M
$\begin{pmatrix} -0.32965 & 0 \\ 1.25965 & 0.83 \end{pmatrix}$	$\begin{pmatrix} 0 & -0.0011308 \\ 0 & 0.0011308 \end{pmatrix}$	0.1		1186.17214	660
		0.01	30.8513	1208.46736	1240
		0.001		1210.80346	1839

Tablo 6.1: (6.5) sistemi için farklı r^* - pratiklik parametreleri ile durulan adım ve genişletilmiş üst sınır değerleri.

A	B	r^*	l	l^e	M
$\begin{pmatrix} -0.32965 & 0 \\ 1.25965 & 0.83 \end{pmatrix}$	$\begin{pmatrix} 0 & -0.0011308 \\ 0 & 0.0011308 \end{pmatrix}$	0.1		-561.92303	25
		0.01	-30.8513	-562.07964	29
		0.001		-562.09349	33

Tablo 6.2: (6.5) sistemi için Farklı r^* - pratiklik parametreleri ile durulan adım ve genişletilmiş alt sınır değerleri.

Buna göre Tablo 6.1 ve Tablo 6.2 den $r^* = 0.001$ pratik adım genişliği kullanılarak genişletilmiş üst sınır 1839 adımda $u^e = 1210.80346$, genişletilmiş alt sınır 33 adımda $l^e = -562.09349$ şeklinde elde edilir. Sonuç olarak $\text{üst sınır}(A, B, ry)$ fonksiyonu (Yordam 3.1) ve $\text{alt sınır}(A, B, ry)$ fonksiyonu (Yordam 3.2) kullanılarak $\mathfrak{T}^e = [l^e, u^e] = [-562.09349, 1210.80346]$ aralığı elde edilir. s değeri bu aralıktan seçildiği sürece (6.5) sistemi Schur kararlı bir sistemdir.

6.1.1.1 Genişletilmiş aralık için etkinlik analizi

Bu kısımda (6.5) sistemi için bulunan $r^* = 0.001$ hassasiyetinde, $\text{üstsnır}(A, B, ry)$ ve $\text{altsnır}(A, B, ry)$ fonksiyonları yardımıyla elde edilen

$$[l^e, u^e] = [-562.09349, 1210.80346]$$

aralığının etkinlik analizini yapalım.

6.1.1.1.1 Üst sınır için etkinlik analizi

$$J(s) = \begin{pmatrix} -0.32965 & -0.0011308s \\ 1.25965 & 0.83 + 0.0011308s \end{pmatrix}$$

olmak üzere $u^e = 1210.80346$ üst sınırı için bu nokta civarında ω_i değerleri Tablo 6.3' de verilmiştir. Tablodan da açıkça görüldüğü üzere aralığın uç noktasına yaklaştıkça sistemin Schur kararlılığının kalitesini belirleyen parametre büyümekte, Schur kararlılığın kalitesi ve sistemin dış etkilere karşı dayanıklılığı azalmaktadır.

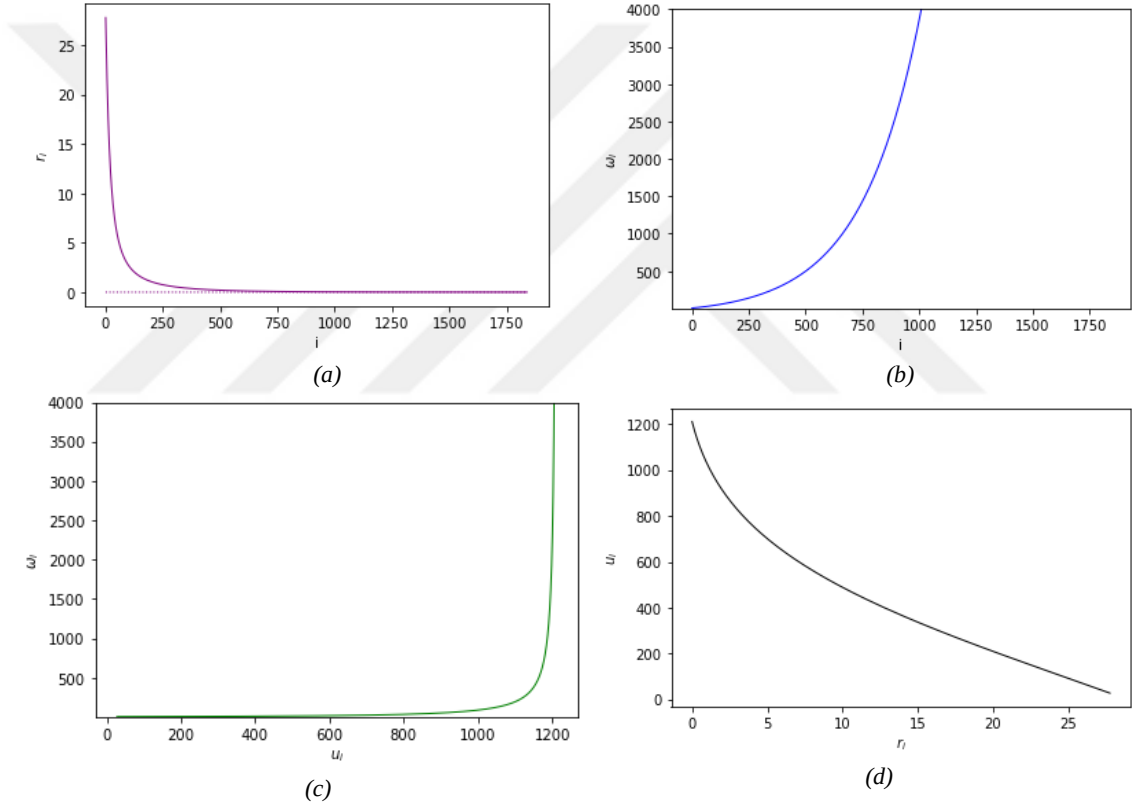
i	s_i	ω_i
1	1210	24005.31062
2	1210.1	26496.05492
3	1210.2	29562.53057
4	1210.3	33430.44186
⋮		
9	1210.8	96517.02992
10	1210.9	154947.09621
11	1211	392483.00340

Tablo 6.3: $\omega(J(s_i)) = \omega_i$, genişletilen üst sınır için etkinlik analizi.

Yordam 3.1 kullanılarak oluşturulan Tablo 6.4' den de görüldüğü üzere her adımda sınır genişletilmiş ve $u^e = 1210$ dan sonra yavaş hareket etmiştir. ω değeri her adımda daha çok büyümüş kararlılığın kalitesi düşmüştür. Yani kararlılık $u^e = 1210$ civarında bozulmaya daha yakındır. Şekil 6.1 (a)' dan anlaşılabacağı üzere i değerleri arttıkça r değerleri azalmaktadır. Benzer şekilde Şekil 6.1 (b)' den i değerleri arttıkça ω_i değeri artmakta ve Şekil 6.1 (c)' den u_i değerleri sınıra yaklaştıkça ω_i değeri artmaktadır.

i	r_i	u_i^e	ω_i
1	27.766238677530897	27.766238677530897	6.50284923245268
2	26.629522217277316	54.39576089480821	6.699604695332942
3	25.531535191982098	79.9272960867903	6.905999143656107
$\vdots$			
1837	0.0010111312501521	1210.801458778976	96680.25573956736
1838	0.0010072683154248	1210.8024660472913	97050.98347113004
1839	0.0010034201540065	1210.8034694674452	97423.13152740692
1840	0.0009995867085990	--	--

Tablo 6.4: $r^* = 0.001$ hassasiyetinde üst sınır genişletilirken r, u ve ω parametrelerinin değişimi.



Şekil 6.1: Tablo 6.4' deki hesaplamalar sonucu (a) r_i ' nin i 'ye göre grafiği, (b) ω_i ' nin i 'ye göre grafiği, (c) ω_i ' in u_i ' ye göre grafiği, (d) u_i 'in r_i ' ye göre grafiği,

6.1.1.1.2 Alt sınır için etkinlik analizi

$$J(s) = \begin{pmatrix} -0.32965 & -0.0011308s \\ 1.25965 & 0.83 + 0.0011308s \end{pmatrix}$$

olmak üzere $l^e = -562.09349$ alt sınırı için bulunan sınır civarında ω_i değerleri Tablo 6.5' de verilmiştir. Tablodan da açıkça görüldüğü üzere aralığın uç noktasına yaklaştıkça sistemin Schur kararlılığının kalitesini belirleyen parametre büyümekte, yani kalitesi azalmakta ve sistemin dış etkilere karşı dayanıklılığı azalmaktadır.

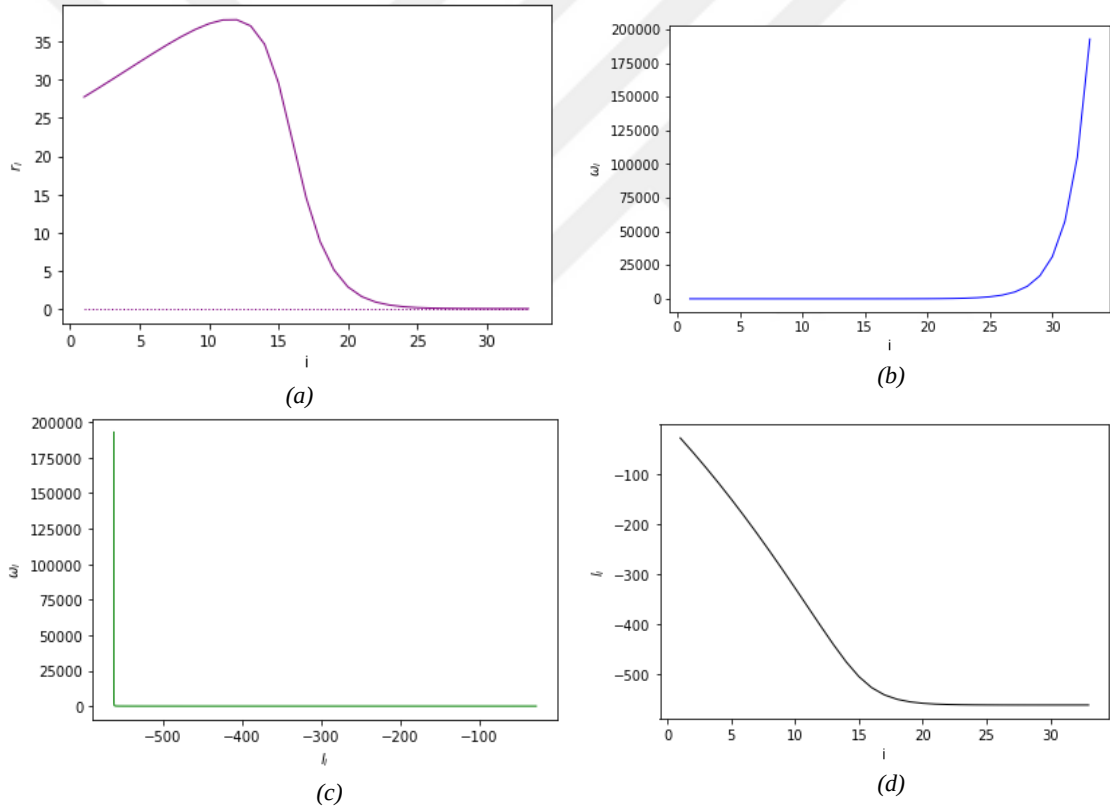
i	s_i	ω_i
1	-562	5010.118675863649
2	-562.01	5600.593432275027
3	-562.02	6348.8753866666375
4	-562.03	7327.984027067787
⋮		
9	-562.08	32023.547659689142
10	-562.09	98255.09507655681
11	-562.091	123875.73879215652
12	-562.092	167571.13488390832

Tablo 6.5: $\omega(J(s_i)) = \omega_i$, genişletilen alt sınır için etkinlik analizi.

Algoritma 3.1 ve $altsınır(A, B, ry)$ kullanılarak oluşturulan Tablo 6.6' dan da görüldüğü üzere her adımda sınır genişletilmiş ve $l^e = -562.09349$ değerinden sonra yavaş hareket etmiştir. ω değeri her adımda daha çok büyümüş kararlılığın kalitesi düşmüştür. Yani kararlılık $l^e = -562.09349$ civarında bozulmaya daha yakındır. Şekil 6.2 (a)' dan anlaşılacağı üzere i değerleri arttıkça r_i değerleri azalmaktadır. Benzer şekilde Şekil 6.2 (b)' den i değerleri arttıkça ω_i değeri artmakta ve Şekil 6.2 (c)' den l_i değerleri sınıra yaklaştıkça ω_i değeri artmaktadır.

i	r_i	l_i^e	ω_i
1	27.766238677530897	-27.766238677530897	6.50284923245268
2	28.889062644664193	-56.655301322195086	6.323650798981803
3	30.036069511871563	-86.69137083406665	6.154633567348888
4	31.197949044172038	-117.88931987823868	5.996618985083551
5	32.36222297810697	-150.25154285634565	5.850593104605477
$\vdots$			
31	0.0037663714956693187	-562.0903195223772	57362.16534649109
32	0.002053531651540652	-562.0923730540287	105207.77745022994
33	0.0011196318901511404	-562.0934926859188	192963.11013490733
34	0.000610445260394954	--	--

Tablo 6.6: $r^* = 0.001$ hassasiyetinde alt sınır genişletilirken r, l ve ω parametrelerinin değışimi.



Şekil 6.2: Tablo 6.6' deki hesaplamalar sonucu (a) r_i 'nin i 'ye göre grafiğı, (b) ω_i 'nin i 'ye göre grafiğı, (c) ω_i 'nin l_i 'ye göre grafiğı, (d) l_i 'nin r_i 'ye göre grafiğı.

6.1.2 (6.6) sisteminin ι parametresine göre Schur kararlılığı

(6.6) sisteminin ι parametresine bağlı Jakobiyeen matrisi

$$J(S^*, \iota)|_{\substack{I=\iota \\ S^*=150.4154}} = \begin{pmatrix} 0.93 - (0.0011308)\iota & (-0.0011308)(150.4154) \\ (0.0011308)\iota & (0.0011308)(150.4154) + 0.107 \end{pmatrix}$$

şeklinde elde edilir. $J(s, I^*)$ matrisi Teorem 2.1 ve Teorem 2.3' den

$$\begin{aligned} J(S^*, \iota) &= \begin{pmatrix} 0.93 - (0.0011308)\iota & (-0.0011308)(150.4154) \\ (0.0011308)\iota & (0.0011308)(150.4154) + 0.107 \end{pmatrix} \\ &= \begin{pmatrix} 0.93 & -0.170089 \\ 0 & 0.277089 \end{pmatrix} + \iota \begin{pmatrix} -0.0011308 & 0 \\ 0.0011308 & 0 \end{pmatrix} \\ &= A + \iota B \end{aligned} \tag{6.8}$$

şeklinde düzenlenirse (2.17) ile verilen Lyapunov fark matris denkleminin çözümü

$$H = \begin{pmatrix} 7.4019245 & -1.577303 \\ -1.577303 & 1.476138 \end{pmatrix}$$

şeklinde elde edilir. $H = H^* > 0$ olduğundan A matrisi (Teorem 2.1' e göre) Schur kararlıdır. (6.8) eşitliğinden

$$\|A\| = 0.9468577, \omega(A) = 7.795610, \Delta = \iota B, \|\Delta\| = 0.0015991926|\iota|$$

değerleri hesaplanır ve

$$\|B\| < \sqrt{\|A\|^2 + \frac{1}{\omega(A)}} - \|A\|$$

eşitsizliğinden (Teorem 2.3' den)

$$|\iota| < 40.9423237$$

eşitsizliğine ulaşılır. Yani $\iota \in (-40.9423237, 40.9423237)$ elde edilir. Bu aralık Schur kararlılığın sağlandığı güvenli bir aralık vermektedir. Ancak bu aralık dışında Schur kararlı sistemler olmasıda mümkündür. Şimdi bulunan bu aralığı kararlılık özellikleri korunacak şekilde genişletelim.

A	B	r^*	u	u^e	M
$\begin{pmatrix} 0.93 & -0.170089 \\ 0 & 0.277008 \end{pmatrix}$	$\begin{pmatrix} -0.0011308 & 0 \\ 0.0011308 & 0 \end{pmatrix}$	0.1	36.8484577	1966.5188597	139
		0.01		1968.7679099	196
		0.001		1968.9914634	253

Tablo 6.7: (6.6) sistemi için farklı r^* - pratiklik parametreleri ile durulan adım ve genişletilmiş üst sınır değerleri.

A	B	r^*	l	l^e	M
$\begin{pmatrix} 0.93 & -0.170089 \\ 0 & 0.277008 \end{pmatrix}$	$\begin{pmatrix} -0.0011308 & 0 \\ 0.0011308 & 0 \end{pmatrix}$	0.1	-36.8484577	-50.0599308	5
		0.01		-50.1101363	7
		0.001		-50.1126678	8

Tablo 6.8: (6.6) sistemi için farklı r^* – pratiklik parametreleri ile durulan adımlar ve genişletilmiş alt sınır değerleri.

Buna göre Tablo 6.7 ve Tablo 6.8 den $r^* = 0.001$ pratik adım genişliği kullanılarak genişletilmiş üst sınır 253 adımda $u^e = 1968.9914634$, genişletilmiş alt sınır (tablo 6.8) 8 adımda $l^e = -50.1126678$ şeklinde elde edilir. Sonuç olarak $\text{üst sınır}(A, B, ry)$ fonksiyonu (Yordam 3.1) ve $\text{alt sınır}(A, B, ry)$ fonksiyonu (Yordam 3.2) kullanılarak $\mathfrak{T}^e = [l^e, u^e] = [-50.1126678, 1968.9914634]$ aralığı elde edilir. ι bu aralıktan seçildiği sürece (6.6) sistemi Schur kararlı bir sistemdir.

6.1.2.1 Genişletilmiş aralık için etkinlik analizi

Bu kısımda (6.6) sistemi için bulunan $r^* = 0.001$ hassasiyetinde, $\text{üst sınır}(A, B, ry)$ ve $\text{alt sınır}(A, B, ry)$ fonksiyonları yardımıyla elde edilen

$$[l^e, u^e] = [-50.1126678, 1968.9914634]$$

aralığının etkinlik analizini yapalım.

6.1.2.1.1 Üst sınır için etkinlik analizi

$$J(\iota) = \begin{pmatrix} 0.93 - 0.0011308\iota & -0.170089 \\ 0.0011308\iota & 0.277089 \end{pmatrix}$$

olmak üzere $u^e = 1968.9914634$ üst sınırı için bu nokta civarında ω_i değerleri Tablo 6.9' da verilmiştir. Tablodan da açıkça görüldüğü üzere aralığın uç noktasına yaklaştıkça sistemin Schur kararlılığının kalitesini belirleyen parametre büyümektedir, yani kalitesi azalmakta dolayısıyla sistemin dış etkilere karşı dayanıklılığı azalmaktadır.

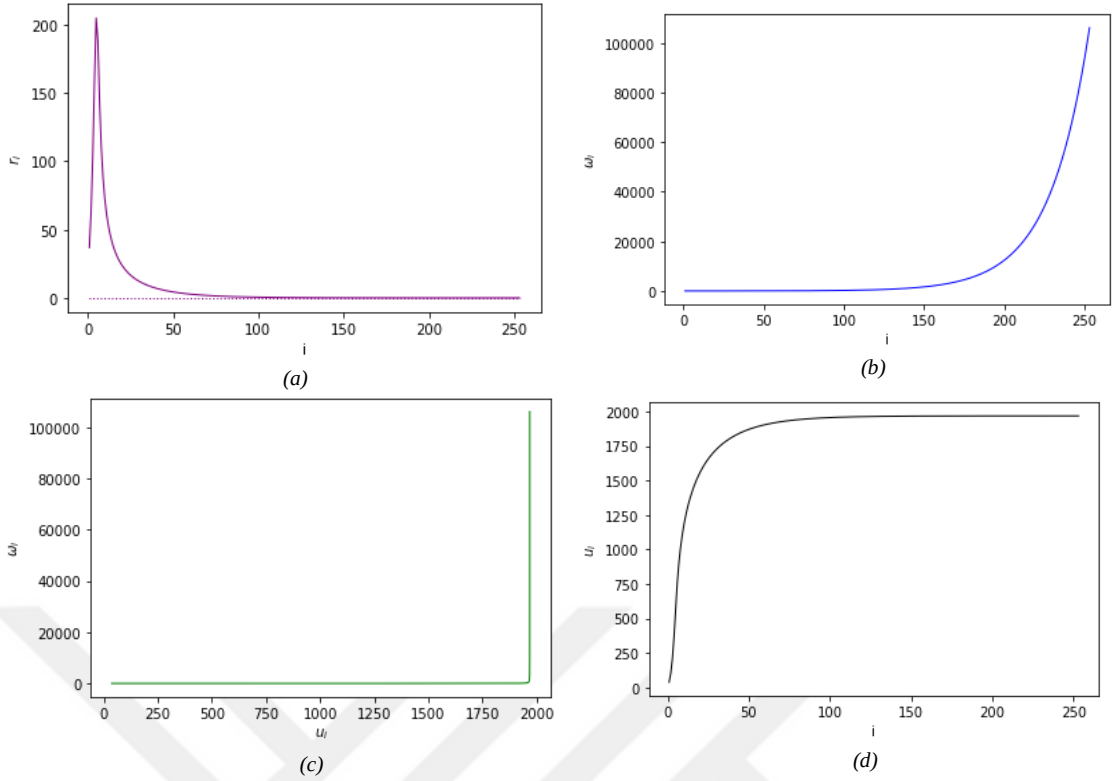
i	l_i	ω_i
1	1968	2688.06566
2	1968.1	2981.49648
3	1968.2	3346.83068
4	1968.3	3814.18776
⋮		
9	1968.8	12636.61017
10	1968.9	23513.52380
11	1969	168831.40577

Tablo 6.9: $\omega(J(l)) = \omega_i$, genişletilen üst sınır için etkinlik analizi.

Yordam 3.1 kullanılarak oluşturulan Tablo 6.10' dan da görüldüğü üzere her adımda sınır genişletilmiş ve $u^e = 1968$ değerinde sonra yavaş hareket etmiştir. ω değeri her adımda daha çok büyümüş kararlılığın kalitesi düşmüştür. Yani kararlılık $u^e = 1968$ civarında bozulmaya daha yakındır. *Şekil 6.3 (a)*' dan anlaşılacağı üzere i değerleri arttıkça r değerleri azalmaktadır. Benzer şekilde *Şekil 6.3 (b)*' den n değerleri arttıkça ω_i değerinin arttığı ve *Şekil 6.3 (c)*' den u_i değerleri sınıra yaklaştıkça ω_i değeri arttığı görülmektedir.

i	r_i	u_i^e	ω_i
1	36.84845779791031	36.84845779791031	7.79553638386821
2	62.72356904585515	99.57202684376546	4.6718308214518975
3	103.4466776146471	203.01870445841257	2.9287040603120498
4	158.837959179482	361.8566636378946	2.0065318076197456
5	204.86392038083906	566.7205840187337	1.6241261394424713
6	188.29825909870598	755.0188431174397	1.6655128027606079
⋮			
251	0.0011069514079508735	1968.989379561458	97886.4810273172
252	0.0010630446818368	1968.9904426061398	101929.40202622229
253	0.0010208795588596332	1968.9914634856987	106139.30049108247
254	0.0009803869515508268	--	--

Tablo 6.10: $r^* = 0.001$ hassasiyetinde üst sınır genişletilirken r, u, ω parametrelerinin değişimi.



Şekil 6.3: Tablo 6.10' daki hesaplamalar sonucu (a) r_i ' nin i 'ye göre grafiği, (b) ω_i ' nin i 'ye göre grafiği, (c) ω_i ' nin u_i 'ye göre grafiği, (d) u_i ' nin r_i 'ye göre grafiği

6.1.2.1.2 Alt sınır için etkinlik analizi

$$J(l) = \begin{pmatrix} 0.93 - 0.0011308l & -0.170089 \\ 0.0011308l & 0.277089 \end{pmatrix}$$

olmak üzere $l^e = -50.1126678$ alt sınırı için bulunan sınır civarında ω_i değerleri Tablo 6.11'de verilmiştir. Tablodan da açıkça görüldüğü üzere aralığın uç noktasına yaklaştıkça sistemin Schur kararlılığının kalitesini belirleyen parametre büyümektedir, yani kalitesi azalmakta dolayısıyla sistemin dış etkilere karşı dayanıklılığı azalmaktadır.

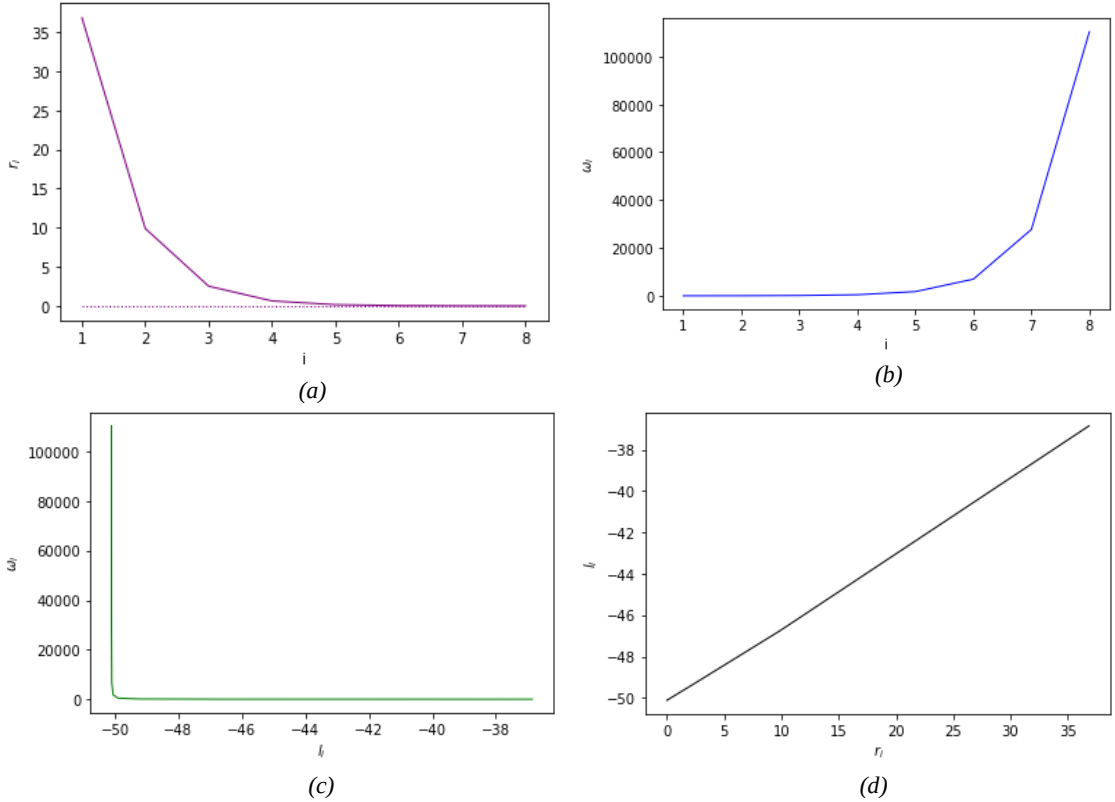
i	l_i	ω_i
1	-50	3287.9573348566155
2	-50.1	27610.49091025011
3	-50.11	106117.71655990981
4	-50.111	148279.7102666514
5	-50.1111	154414.82408943458
6	-50.11111	155056.374166057
7	-50.111111	155120.82249186677
		$\vdots$

Tablo 6.11: $\omega(J(l)) = \omega_i$, genişletilen alt sınır için etkinlik analizi.

Algoritma 3.1 ve $\text{üst sınır}(A, B, ry)$ fonksiyonu kullanılarak oluşturulan Tablo 6.12' den de görüldüğü üzere her adımda sınır genişletilmiş ve $l^e = -50.1126678$ değerinde sonra yavaş hareket etmiştir. ω değeri her adımda daha çok büyümüş ve kararlılığın kalitesi düşmüştür. Yani kararlılık $l^e = -50.1126678$ civarında bozulmaya daha yakındır. Şekil 6.4 (a)' dan anlaşılabacağı üzere i değerleri arttıkça r_i değerleri azalmaktadır. Benzer şekilde Şekil 6.4 (b)' den i değerleri arttıkça ω_i değeri artmakta ve Şekil 6.4 (c)' den l_i değerleri sınıra yaklaştıkça ω_i değeri artmaktadır.

i	r_i	l_i^e	ω_i
1	36.84845779791031	-36.84845779791031	7.79553638386821
2	9.895019010032867	-46.743476807943175	28.451057631390
3	2.521303727040447	-49.26478053498362	111.049932869977
4	0.6354211124380982	-49.90020164742172	440.02333099423737
5	0.15972915611341706	-50.05993080353513	1749.8477684771879
6	0.040126635176488894	-50.10005743871162	6964.874312106966
7	0.01007889309461234	-50.11013633180624	27728.31834485877
8	0.0025314873705497122	-50.112667819176785	110397.23075650985
9	0.0006358202867886575	--	--

Tablo 6.12: $r^* = 0.001$ hassasiyetinde alt sınır genişletilirken r, l, ω parametrelerinin değişimi.



Şekil 6.4: Tablo 6.12' deki hesaplamalar sonucu (a) r_i 'nin i 'ye göre grafiği, (b) ω_i 'in i 'ye göre grafiği, (c) ω_i 'in l_i 'e göre grafiği, (d) l_i 'in r_i 'e göre grafiği.

6.2 SIR Modelinin ω^* – Schur Kararlılığı

Bu bölümde önce s parametresine bağlı (6.5) sisteminin ω^* – Schur kararlılığı, sonra t parametresine bağlı (6.6) sisteminin ω^* – Schur kararlılığı incelenmiştir.

6.2.1 (6.5) sisteminin s parametresine göre ω^* – Schur kararlılığı

(6.7) eşitliğinden

$$\|A\| = 1.533774, \quad \omega(A) = 6.502848, \quad \Delta = sB, \quad \|\Delta\| = 0.00159919|s|$$

değerleri hesaplanır ve

$$\|B\| < \sqrt{\|A\|^2 + \frac{\omega^* - \omega(A)}{\omega^* \omega(A)}} - \|A\|$$

eşitliğinden (Teorem 2.4' den) $\omega^* = 10$ pratik değeri için

$$|s| < 10.9007732$$

$\omega^* = 100$ pratik değeri için

$$|s| < 28.8744455$$

eşitsizlikleri elde edilir. Verilen teoremler ω^* – Schur kararlılığın sağlandığı güvenli bir aralık vermektedir. Buradan görüldüğü üzere bulunan Pratik Schur kararlı aralık daha önce bulduğumuz Schur kararlı aralığının içinde kalmaktadır. Şimdi ω^* – Schur kararlı olduğunu bildiğimiz aralığı $\text{pr_üstsnır}(A, B, \omega^*)$ ve $\text{pr_altsnır}(A, B, \omega^*)$ fonksiyonları yardımı ile r^* ve ω^* değerleri belirleyerek genişletelim.

A	B	ω^*	r^*	u^*	u^{*e}	M
$\begin{pmatrix} -0.32965 & 0 \\ 1.25965 & 0.83 \end{pmatrix}$	$\begin{pmatrix} 0 & -0.0011308 \\ 0 & 0.0011308 \end{pmatrix}$	10	0.1	10.9007732	301.1070122	138
			0.01		303.8995963	210
			0.001		304.1774178	282
		100	0.1	28.8744455	1003.0921829	421
			0.01		1014.4423098	713
			0.001		1015.6004774	1009

Tablo 6.13: Farklı r^* , ω^* - pratiklik parametreleri ile durulan adım ve genişletilmiş üst sınır değerleri.

A	B	ω^*	r^*	l^*	l^{*e}	M
$\begin{pmatrix} -0.32965 & 0 \\ 1.25965 & 0.83 \end{pmatrix}$	$\begin{pmatrix} 0 & -0.0011308 \\ 0 & 0.0011308 \end{pmatrix}$	10	0.1	-10.9007732	-507.7361155	41
			0.01		-507.9652771	46
			0.001		-507.9901887	52
		100	0.1	-28.8744455	-557.1256697	23
			0.01		-557.2920419	27
			0.001		-557.3020149	30

Tablo 6.14: Farklı r^* , ω^* - pratiklik parametreleri ile durulan adım ve genişletilmiş alt sınır değerleri.

Buna göre Tablo 6.13 ve Tablo 6.14 den $r^* = 0.001$, $\omega^* = 10$ pratiklik parametreleri kullanılarak genişletilmiş üst sınır 282 adımda $u^{*e} = 304.1774178$, genişletilmiş alt sınır 52 adımda $l^{*e} = -507.9901887$ elde edilir. $r^* = 0.001$, $\omega^* = 100$ pratiklik parametreleri kullanılarak, genişletilmiş üst sınır 1009 adımda $u^{*e} = 1015.6004774$ ve genişletilmiş alt sınır 30 adımda $l^{*e} = -557.3020149$ elde edilir.

Sonuç 6.1: $pr_üstsınır(A, B, \omega^*)$ fonksiyonu (Yordam 3.3) ve $pr_üstsınır(A, B, \omega^*)$ fonksiyonu (Yordam 3.4) kullanılarak, $r^* = 0.001$ ve $\omega^* = 10$ pratiklik parametreleri ile

$$\mathfrak{T}^e = [l^e, u^e] = [-507.9901887, 304.1774178]$$

aralığı elde edilir. Buradan s değerleri bu aralıktan seçildiği sürece (6.5) sistemi $\omega^* = 10$ – Pratik Schur kararlı bir sistemdir. $r^* = 0.001$, $\omega^* = 100$ pratiklik parametreleri ile

$$\mathfrak{T}^e = [l^e, u^e] = [-557.3020149, 1015.6004774]$$

aralığı elde edilir. Buradan s değerleri bu aralıktan seçildiği sürece (6.5) sistemi $\omega^* = 100$ – Pratik Schur kararlı bir sistemdir.

6.2.2 (6.6) sisteminin ι parametresine göre ω^* – Schur kararlılığı

(6.8) eşitliğinden

$$\|A\| = 0.9468577, \quad \omega(A) = 7.795610, \quad \Delta = \iota B, \quad \|\Delta\| = 0.0015991926|\iota|$$

hesaplanır ve

$$\|B\| < \sqrt{\|A\|^2 + \frac{\omega^* - \omega(A)}{\omega^* \omega(A)}} - \|A\|$$

eşitliğinden (Teorem 2.4' den) $\omega^* = 10$ pratik değeri için

$$|l| < 9.2652458$$

$\omega^* = 100$ pratik değeri için

$$|l| < 37.8466702$$

eşitsizliklerine ulaşılır. Teorem Pratik Schur kararlılığın sağlandığı güvenli bir aralık vermektedir. Buradan görüldüğü üzere bulunan Pratik Schur kararlı aralık daha önce bulduğumuz Schur kararlı karalığın içinde kalmaktadır. Şimdi ω^* – Schur kararlı olduğunu bildiğimiz aralığı $\text{pr_üstsnır}(A, B, \omega^*)$ ve $\text{pr_altsnır}(A, B, \omega^*)$ fonksiyonları yardımı ile r^* ve ω^* değerleri belirleyerek genişletelim.

A	B	ω^*	r^*	u^*	u^{*e}	M
$\begin{pmatrix} 0.93 & -0.170089 \\ 0 & 0.277008 \end{pmatrix}$	$\begin{pmatrix} -0.0011308 & 0 \\ 0.0011308 & 0 \end{pmatrix}$	10	0.1	9.2652458	1702.4460389	105
			0.01		1704.0394079	145
			0.001		1704.2014459	186
		100	0.1	37.8466702	1939.6140220	125
			0.01		1941.5622094	175
			0.001		1941.7562362	225

Tablo 6.15: Farklı r^* , ω^* - pratiklik parametreleri ile bulunan ω^* – Schur kararlılık içingenişletilmiş üst sınır değerleri.

A	B	ω^*	r^*	l^*	l^{*e}	M
$\begin{pmatrix} 0.93 & -0.170089 \\ 0 & 0.277008 \end{pmatrix}$	$\begin{pmatrix} -0.0011308 & 0 \\ 0.0011308 & 0 \end{pmatrix}$	10	0.1	-9.2652458	-11.4063035	3
			0.01		-11.4828459	5
			0.001		-11.4851352	6
		100	0.1	-37.8466702	-46.3279087	4
			0.01		-46.3628106	5
			0.001		-46.3697223	7

Tablo 6.16: Farklı r^* , ω^* - pratiklik parametreleri ile bulunan ω^* – Schur kararlılık için genişletilmiş üst sınır değerleri.

Buna göre Tablo 6.15 ve Tablo 6.16 dan $r^* = 0.001$, $\omega^* = 10$ pratiklik parametreleri kullanılarak genişletilmiş üst sınır 186 adımda $u^{*e} = 1704.2014459$, genişletilmiş alt sınır 6 adımda $l^{*e} = -11.4851352$ elde edilir. $r^* = 0.001$, $\omega^* = 100$

pratiklik parametreleri kullanılarak, genişletilmiş üst sınır 225 adımda $u^{*e} = 1941.7562362$ ve genişletilmiş alt sınır 7 adımda $l^{*e} = -46.3697223$ elde edilir.

Sonuç 6.2: $pr_üstsınır(A, B, \omega^*)$ fonksiyonu (Yordam 3.3) ve $pr_üstsınır(A, B, \omega^*)$ fonksiyonu (Yordam 3.4) kullanarak, $r^* = 0.001$, $\omega^* = 10$ pratiklik parametreleri ile

$$\mathfrak{T}^e = [l^e, u^e] = [-11.4851352, 1704.2014459]$$

aralığı elde edilir. Buradan ι bu aralıktan seçildiği sürece (6.8) sistemi $\omega^* = 10$ – Pratik Schur kararlı bir sistemdir. $r^* = 0.001$, $\omega^* = 100$ pratiklik parametreleri ile

$$\mathfrak{T}^{*e} = [l^{*e}, u^{*e}] = [-46.3697223, 1941.7562362]$$

aralığı elde edilir. Buradan ι bu aralıktan seçildiği sürece (6.8) sistemi $\omega^* = 100$ – Pratik Schur kararlı bir sistemdir.



7. SONUÇLAR VE DEĞERLENDİRMELER

Dinamik sistemlerin kararlılığı ve kararlılık analizleri ile ilgili yapılan çalışmalar incelenmiş ve bazı gerçek dünya problemlerinin kararlılık analizleri ve parametrik değerlendirmeleri yapılmıştır. Tezde kullanılan üç temel matematik model kullanılmıştır. Bunlar İki Ekonomili Makroekonomik model, Lotka-Volterra modeli ve Salgın Hastalık (SIR) modelidir. Bu modellerin parametrik katsayıları ve denge noktaları üzerinden kararlılık analizleri yapılmıştır.

Kesikli zamanlı sistemler ve sürekli zamanlı sistemler için denge noktası, Schur kararlılık, ω^* – Schur kararlılık, Hurwitz kararlılık ve κ^* – Hurwitz kararlılık hakkında temel literatür bilgileri verilmiştir.

Lineer olmayan sistemlerin kararlılıkları, lineer sistemler için mevcut olan ω, κ parametreleri ile incelenmiştir. Bunun için lineer olmayan sistemlerin lineerleştirilmesi ile ilgili yöntem verilmiştir.

Schur ve Hurwitz kararlılık kavramları ile ilgili süreklilik teoremleri tanıtılmıştır. Literatürde verilen süreklilik teoremlerine dayalı olan ve matris aileleri için kararlılık aralığı veren ve bu aralığı genişleten algoritmalar verilmiştir. Bu algoritmalar Python yazılım dilinde kodlanmıştır.

Yapılan bu çalışmada, öncelikle lineer bir model olan iki ekonomili makroekonomik model ele alınmıştır. Modelin kararlılık analizi literatürdekilerin aksine Lyapunov matris denkleminin pozitif tanımlı çözümünü kullanan dinamik sistemlerin hem kararlılığını tespit etmeye hem de kararlılığın kalitesi hakkında değerlendirme yapmaya imkan sağlayan ω kararlılık parametresi kullanılarak yapılmıştır (Aydın, 1995; Akın ve Bulgak, 1998; Elaydi, 1999; Bulgak, 1999).

Literatürde var olan kararlılık aralıklarının genişletilmesi ile ilgili birikimler kullanılarak makroekonomik modelin parametrik giriş elemanları için kararlılık aralıkları tespit edilmiş ve bu aralıklar kararlı kalacak şekilde genişletilmiştir.

Daha sonra lineer olmayan Lotka-Volterra modeli ele alınmış modelin kararlılık analizi literatürdekilerin aksine Lyapunov matris denkleminin pozitif tanımlı çözümünü kullanan dinamik sistemlerin hem kararlılığını tespit etmeye hem de kararlılığın kalitesi hakkında değerlendirme yapmaya imkan sağlayan κ kararlılık parametresi kullanılarak yapılmıştır (Aydın, 1995; Akın ve Bulgak, 1998; Elaydi, 1999; Bulgak, 1999).

Lotka-Volterra modelinin parametrik giriş elemanları için kararlılık aralıkları tespit edilecek ve bu aralıklar kararlı kalacak şekilde genişletilmiştir. Burada bir noktaya dikkat edilmelidir. Birincisi, makroekonomik model lineer bir modeldi ve literatür bilgileri lineer sistemler için tasarlanmış bilgilerdir. Ancak Lotka-Volterra modeli lineer olmayan bir modeldir. Dolayısıyla öncelikle model lineerleştirilip daha sonra kararlılık analizi yapılmıştır. Yani lineer olmayan modelin kararlılık analizi lineer hale getirilen model üzerinden incelenmiştir.

Son olarak, epidemolojik salgın hastalık modeli olarak bilinen sistem ele alınmış ve modelin kararlılık analizi literatürdekilerin aksine Lyapunov matris denkleminin pozitif tanımlı çözümünü kullanan dinamik sistemlerin hem kararlılığını tespit etmeye hem de kararlılığın kalitesi hakkında değerlendirme yapmaya imkan sağlayan ω kararlılık parametresi kullanılarak yapılmıştır. Python yardımıyla yapılan bu çalışmalar görselleştirilmiştir.

Tezde verilen orjinal sonuçlar ve literatürde yapılmış olan çalışmaların farkı da şu şekildedir;

Literatürdeki diğer çalışmalardan farklı olarak, Makroekonomik model, Lotka-Volterra modeli ve Salgın hastalık modelinin kararlılık analizleri fark Lyapunov matris denkleminin pozitif tanımlı çözümü olan H matrisine bağlı kararlılığın kalitesini belirleyen ω ve Lyapunov matris denkleminin pozitif tanımlı çözümü olan F matrisine bağlı kararlılığın kalitesini belirleyen κ parametreleri yardımıyla değerlendirilmiştir. Bu yön itibarıyla yapılan çalışma literatürde ilktir. Bunun yanı sıra literatürde var olan kararlılık aralığının genişletilmesi ile ilgili yöntem ve algoritmalar yardımıyla sistemlerin parametrik katsayıları ve denge noktaları için modellerin kararlılık özellikleri korunacak şekilde aralıklar bulunmuştur. Bu sebeple yapılan çalışma orijinal bir içeriğe sahiptir.

KAYNAKLAR

- Agarwal, R. P., 2000, Difference equations and inequalities: Theory, Methods, and Applications (2nd ed.). *New York: Marcel Dekker.*
- Akın, O. ve Bulgak, H., 1998, Lineer fark denklemler ve kararlılık teorisi, Selçuk Üniversitesi basımevi, Konya.
- Allen, L. J. S., 2007, An introduction to mathematical biology, *Pearson Prentice Hall*, Upper Saddle River.
- Aydın, K., 1995, Periyodik adi diferensiyel denklem sistemlerinin asimtotik kararlılığı için şart sayısı (*Danışman: Prof. Dr. Hasan Şenay, Yard. Danışman: Prof. Dr. Y. Aider Bulgakov*), Doktora tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü.
- Aydın, K., Bulgak, H., and Demidenko, G. V., 2000, Numeric characteristics for asymptotic stability of solutions to linear difference equations with periodic coefficients, *Siberian Mathematical Journal*, 41(6), 1227-1237.
- Aydın, K., Bulgak, H., and Demidenko, G. V., 2001, Continuity of numeric characteristics for asymptotic stability of solutions to linear difference equations with periodic coefficients, *Selçuk J. Appl. Math.*, 2: 5-10.
- Aydın, K., Bulgak, H., and Demidenko, G. V., 2002, Asymptotic stability of solutions to perturbed linear difference equations with periodic coefficients, *Siberian Mathematical Journal*, 43(3), 389-401.
- Bacaer, N., 2011, A short history of mathematical population dynamics. *Springer-Verlag*, London
- Bereketoğlu, H. ve Kutay, V. 2012, Fark Denklemleri, Gazi Kitabevi, Ankara.
- Bertotti, G., Mayergoyz, I.D. and Serpico, C. 2009, Nonlinear Magnetization Dynamics in Nanosystems. *Elsevier Science*.
- Blanchard, O. and Johnson, D. R., 2013, Macroeconomics (6th ed.), Boston: Pearson Education.
- Bulgak, H., 1999, Pseudoeigenvalues, spectral portrait of a matrix and their connections with different criteria of stability, *Error Control and Adaptivity in Scientific Computing*, 95–124.
- Bulgak, H., 2020, Pseudospectrum and different quality of stability parameters, *Continuum Mechanics, Applied Mathematics and Scientific Computing: Godunov's Legacy, Springer, Cham.*, 61–67.
- Brauer F., Driessche P., and Wu J., 2008, Mathematical Epidemiology, Department of Mathematics University of British Columbia Vancouver, Canada, Springer.

- Cao, H., Wu, Z., and Wang, X., 2020, Bifurcation analysis of a discrete SIR epidemic model with constant recovery. *Advances in Difference Equations*, 2020(1), 49.
- Chen, G. 2004, Stability of Nonlinear Systems. In *Encyclopedia of RF and Microwave Engineering* (pp. 4881–4896). New York: Wiley.
- Chern, I-L, 2016, Mathematical Modeling and Ordinary Differential Equations. Taipei: National Taiwan University. <http://www.math.ntu.edu.tw/~chern/notes/ode2015.pdf>
- Çelik Kızıllkan, G., 2009, Diferensiyel denklem sistemlerinin nümerik integrasyonunda adım genişliği stratejileri (*Danışman: Prof. Dr. Kemal Aydın*), Doktora tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü.
- Duman, A., 2008, Periyodik lineer fark denklem sistemlerin Schur kararlılığının hassasiyeti (*Danışman: Prof. Dr. Kemal Aydın*), Doktora tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü.
- Duman, A. ve Aydın, K., 2011, Sensitivity of Schur stability of systems of linear difference equations with constant coefficients, *Scientific Research and Essays*, 6(28), 5846-5854.
- Duman, A. ve Aydın, K., 2017, Sensitivity of Hurwitz stability of linear differential equation systems with constant coefficients, *Int. J. Geom. Methods Mod. Phys.*, 14(06):1750084.
- Elaydi, S., 2005, An Introduction to difference equations, *Springer- Verlag*, Newyork
- Emre, B. 2022, Bağımsız Sınırlayıcılar ve Vektörel Çekim Alanı Etkisindeki Parçacığın Kaotik Dinamiği. Yüksek Lisans Tezi, Akdeniz Üniversitesi Fen Bilimleri Enstitüsü, Antalya.
- Enszer, J. A., Maces, D. A., and Stadtherr, M. A., 2015, Probability bounds analysis for nonlinear population ecology models. *Mathematical Biosciences*, 267, 97–108.
- Halkos, G. E., and Tsilika, K. D., 2012, Stability Analysis in Economic Dynamics: A Computational Approach, MPRA Paper No. 41371. Munich Personal RePEc Archive.
- Hassell, M. P., 2000, Host-parasitoid population dynamics. *Journal of Animal Ecology*, 69 (4): 543-566.
- Kumar, G. S., and Gunasundari, C., 2023, Dynamical Analysis of Two-Preys and One Predator Interaction Model with An Allee Effect on Predator. *Malaysian Journal of Mathematical Sciences*, 17(3), 263–281.
- Li, X. and Chen, K., 2023, Equilibria and their stability in an asymmetric duopoly model of Kopel, ArXiv:2301.12628v1 [econ.TH].

- A. M. Lyapunov, 1992, The general problem of the stability of motion. Translated from Edouard Davaux's French translation (1907) of the 1892 Russian original and edited by A. T. Fuller., London: Taylor & Francis, Ltd.
- Neusser, K. 2021, Difference equations for economists, Bern. <http://www.neusser.ch/downloads/DifferenceEquations.pdf>
- Strogatz, S. H., 2018, Nonlinear Dynamics and Chaos. *CRC Press is an imprint of the Taylor & Francis Group, an informa business*, Boca Raton.
- Taghikhani, R., Sharomi, O., and Gumel, A. B., 2020, Dynamics of a two-sex model for the population ecology of dengue mosquitoes in the presence of Wolbachia. *Mathematical Biosciences*, 328, 108426.
- Taşdemir, E., and Soykan, Y., 2021, On the dynamics of a system of difference equations. *Konuralp Journal of Mathematics*, 9(1), 1–9.
- Topcu, G. and Aydın, K., 2023a, Construction of Hurwitz stability intervals for matrix families, *Dolomites Research Notes on Apporximation*, Special Issue ICOMSS'22,16(2), Pages 57–63.
- Topcu, G. and Aydın, K., 2023b, A new approach to construct and extend the Schur stable matrix families, *Commun.Fac.Sci.Univ.Ank.Ser. A1 Math. Stat.*, 73(2), 538–553.
- Topcu, G., 2023, Polinom ailesinin kararlılığı ve kararlılığın hassasiyeti, (*Danışman: Prof. Dr. Kemâl Aydın*), Doktora tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü.
- Tu, P.N.V., 1992, Dynamical systems, *Springer-Verlag Berlin Heidelberg GmbH*, Department of Economics The University of Calgary 2500 University Drive N.W., Calgary, Alberta T2N IN4 Canada
- Voicu, M. and Pastravanu, O., 2006, Generalized matrix diagonal stability and linear dynamical systems, *Linear Algebra Appl.*, 419(2-3),299–310.
- Scheinerman, E. R., 2012, Invitation to dynamical systems, *Dover Publications*.

Yararlanılan Diğer Kaynaklar

- Akkocaoğlu, H. 2011, Genel Gecikmeli bir Diferensiyel Denklemin Horf Çatallanma Analizi. Yüksek Lisans Tezi, Tobb Ekonomi ve Teknoloji Üniversitesi Fen Bilimleri Enstitüsü, Ankara
- Aksoy, Y. ve Özkan, M., 2017, Diferansiyel Denklemler (Cilt 2)., Yıldız Teknik Üniversitesi Basım-Yayın Merkezi, İstanbul.
- Allen, L. J. S., Strauss, M. J., Tnorvilson, H. G., Lipe, W. N., 1991. A preliminary mathematical model of the apple twig borer (Coleoptera: Bostricidae) and grapes on the Texas High Planes. *Ecology Model*, 58: 369-382.

- Barbarossa, M.V. 2012, On a class of neutral equations with state-dependent delay in population dynamics. Technische Universität München Zentrum Mathematik, Deutschland.
- Cortes, J. C. and ark., 2019, Solving the Random Pielou Logistic Equation with the 114 Random Variable Transformation Technique: Theory and Applications, arXiv:1901.10907v1 [math.PR] Chavez, C.C. ve Huang, W., 1994, The Logistic Equation Revisited: The Two-Sex Case, Elsevier.
- Clinger, R.A., 2007, Stability analysis of systems of difference equations, Virginia Commonwealth University.
- Çakıroğlu, R., 2019, Yüksek mertebeden fark denklem sistemlerinin çözümlerinin hareketi, Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.
- Çimencan, T., 2017, Nicholson-Bailey modeli ve bazı özellikleri, Yüksek Lisans Tezi, Erciyes Üniversitesi Fen Bilimleri Üniversitesi, Kayseri.
- Dağ, İ., 1983, Bayağı diferensiyel denklemler, Atatürk Üniversitesi basımevi, Erzurum.
- Dawkins, P., 2018, Differential equations, <http://tutorial.math.lamar.edu>.
- Din, Q., Khan, A. Q., Qureshi, M. N., 2013, Qualitative behavior of a hostpathogen model, *Advances in Difference Equations* 2013, 2013:263. DOI: 10.1186/1687-1847-2013-263.
- Emre, B. 2022, Bağımsız Sınırlayıcılar ve Vektörel Çekim Alanı Etkisindeki Parçacığın Kaotik Dinamiği. Yüksek Lisans Tezi, Akdeniz Üniversitesi Fen Bilimleri Enstitüsü, Antalya.
- Engin ,T. ve Çengel, Y.A., 2008, Mühendisler için Diferansiyel Denklemler, Sakarya.
- Kawamura, T. and Shima, M., 1996, Robust stability analysis of characteristic polynomials whose coecients are polynomials of interval parameters, *Journal of Mathematical Systems, Estimation, and Control* Vol. 6, No. 4, 1996, pp. 1-12.
- Khan, A. Q., Qureshi, M., (2015), Bailey host-parasitoid model. *Advances in Difference Equations*, 23. DOI: 10.1186/s13662-015-0357-2
- Köker, N., 2021, Kararlılık analizi ve bir uygulaması, Yüksek Lisans Tezi, Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Sakarya.
- Kuznetsov, Y.A., 2000, Elements of applied bifurcation theory, *Springer*, New York.
- Li, X., Mou, C., Niu, W., and Wang, D., 2011. Stability analysis for discrete biological models using algebraic methods. *Mathematics in Computer Sciences*, 5 (3): 247-262

- Li, X. ve Chen, K., 2023, Equilibria and their stability in an asymmetric duopoly model of Kopel, ArXiv:2301.12628v1 [econ.TH].
- Linda, J. S., Allen, 2007, An Introduction to Mathematical Biology, *Pearson Prentice Hall*, 348 pp.
- Matsumoto, A. and Szidarovszky, F., 2011. Delay differential neoclassical growth model, *Journal of Economic Behavior and Organization*, 78 (2011) 272–289
- Matsumoto, A. and Szidarovszky, F., 2013. Asymptotic behavior of a delay differential neoclassical growth model. 5, 440-455; doi:10.3390/su5020440
- May, R.M., 1976. Simple mathematical models with very complicated dynamics. *Nature*, 261(5560):459–467.
- May, R. M., Hassell, M. P., Anderson, R. M., and Tonkyn, D. R., 1981. Density dependence in host-parasitoid models. *Journal of Animal Ecology*, 50 (3): 855-865.
- Miao, O. and Zhang, Q., 2022. Dynamical behavior of Pielou's difference system with exponential term. arXiv:2209.10793
- Mnyazikwiye, B.B., Karimi, H. R. and Robbersmyr, K. G., 2013. Mathematical modeling of vehicle frontal crash by a double spring-mass-damper model
- Nelder, J.A., 1961, The Fitting Of A Generalization Of The Logistic Curve, *International Biometric Society*.
- Sedaghat, H., 2003. Nonlinear Difference Equations Theory with Applications to Social Science Models. *Kluwer Academic Dordrecht*, 45 pp.
- Schtickzelle, M., 1981, Pierre-françois verhulst 1804-1849, la première découverte de la fonction logistique. *Population (french edition)*, 541–556
- Ufuktepe, U. , 2013. Stability analysis of a host parasite model. *Advances in Difference Equations*, 79. DOI: 10.1186/1687-1847-2013-79.
- Qureshi, M. N., Khan, A. Q., Din Q., 2014. Asymptotic behavior of a Nicholson-Bailey model. *Advances in Difference Equations*, 62. DOI: 10.1186/1687-1847-2014-62.
- Wade, M.J. and ark., 2015, Perspectives in mathematical modelling for microbial ecology, *Ecological Modelling* 321 (2016) 64–74
- Wang, W. and Chen, W., 2019, Stochastic delay differential neoclassical growth model, *Advances in Difference Equations*.
- xAI. (2025). Grok 3 (Yapay Zeka Destekli Dil Modeli). xAI. Erişim adresi: <https://x.ai/grok>