

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ROBOTİKTE HÜCRESEL SİNİR AĞLARI
UYGULAMALARI**

Elektronik ve Haberleşme Mühendisi Nerhun YILDIZ

**FBE Elektronik ve Haberleşme Mühendisliği Anabilim Dalı Elektronik Programında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof. Dr. Vedat TAVŞANOĞLU

İSTANBUL, 2007

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	iv
KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vii
ÇİZELGE LİSTESİ	viii
ÖNSÖZ	ix
ÖZET	x
ABSTRACT	xi
1 GİRİŞ	1
1.1 Robot ve Robotik Kavramları	1
1.2 Lineer HSA Süzgeçleri	3
2 HÜCRESEL SİNİR AĞLARI (HSA)	4
2.1 HSA'nın Tanımlanması	4
2.2 HSA'nın Vektörel Olarak İfade Edilmesi	6
2.3 HSA'nın Sınır Koşulları	7
2.4 Konumdan Bağımsız HSA	8
3 TEPKİME YAYINIMLI HSA (TY-HSA)	11
3.1 Kendini Yenileyen Dalgalar ve Turing Desenleri	12
3.2 HSA devresi ve matematiksel modeli	12
3.3 HSA ile Kendini Yenileyen Dalga ve Turing Deseni Oluşturma	13
3.3.1 İki Katmanlı TY-HSA'nın Tanımlanması	13
3.3.2 Sınır Koşulları	16
4 TY-HSA İLE ROBOTLARIN HAREKET KONTROLÜ	17
4.1 Hayvanların Hareket Sisteminin Modellemesi	17
4.2 Kendini Yenileyen Dalga Oluşturma Simülasyonu	18
4.2.1 Kendini Yenileyen Dalga Oluşturmak İçin Gereken Sınır Koşulları ve İlk Değerler	18
4.2.2 Sürekli-Zaman TY-HSA Denkleminin Ayrıklaştırılması	19
4.2.3 Kendini Yenileyen Dalga Oluşturma Simülasyonu Sonuçları	20
4.3 Tırtıl Robotların Hareket Kontrolü	23
4.4 Altı Bacaklı Robotların Hareket Kontrolü	25

4.4.1	Altı Bacaklı Robotların Hareket Stratejileri	25
4.4.2	TY-HSA ile Altı Bacaklı Robotların Hareket Stratejilerinin Gerçeklenmesi	27
4.5	TY-HSA ile Turing Deseni Oluşturma	29
5	LİNEER HSA SÜZGEÇLERİ İLE GÖRÜNTÜ ÖN İŞLEME	31
5.1	Gauss Benzeri Alçak Geçiren Süzgeçlerin HSA ile Gerçeklenmesi	31
5.2	Gabor Benzeri Bant Geçiren Süzgeçlerin HSA ile Gerçeklenmesi	36
5.3	Lineer HSA'ların Sayısal Çözüm Yöntemleri	38
5.3.1	Euler İleri Yaklaşıklığı	38
5.3.2	Jacobi Yinelemeli Çözüm Yöntemi	39
5.3.3	Gauss-Seidel Yinelemeli Çözüm Yöntemi	40
5.3.4	Lineer Denklem Takımına Jacobi Yaklaşıklığından Yararlanarak Geçiş	40
5.3.5	Lineer Denklem Takımlarına Doğrudan Geçiş	42
5.4	Sonlu Sayıda İşlemle $\mathbf{Ax}=\mathbf{b}$ Ayırık Lineer HSA Denkleminin Çözümü . . .	43
5.4.1	Matrisin Tersini Alarak Çözüm Yöntemi	44
5.4.2	Gauss-Jordan Eleme Yöntemi	44
5.4.3	Gauss Eleme Yöntemi	45
5.4.4	HSA'nın Durum Matrisinin Özel Yapısı	48
5.4.5	Matrisin Tersini Almanın HSA'ya Uyarlanması	51
5.4.6	Öz Değerler ve Öz Vektörler Yardımıyla Çözüm	51
5.4.7	Gauss-Jordan Eleme Yönteminin HSA'ya Uyarlanması	52
5.4.8	Gauss Eleme Yönteminin HSA'ya Uyarlanması	54
5.5	Lineer HSA Simülasyonu Sonuçları	56
6	SONUÇLAR	59
	KAYNAKLAR	61
	ÖZGEÇMİŞ	63

SİMGE LİSTESİ

$A_{i,j;k,l}$	HSA'nın geri besleme sinaptik ağırlıkları
A_c, I_c	Aktivayör ve inhibitör yoğunlukları
$\mathbf{A}$	Lineer olmayan HSA'nın geri besleme şablonu
$\tilde{\mathbf{A}}$	Lineer HSA'nın geri besleme şablonu
$\mathbf{A}$	Doğrudan ayrıklaştırılmış HSA'nın durum matrisi
$\mathcal{A}$	Jacobi yöntemi ile ayrıklaştırılmış HSA'nın durum matrisi
$\mathbf{A}$	Lineer olmayan HSA'nın geri besleme matrisi
$\mathbf{A}_s$	Sıkıştırılmış $\mathbf{A}$ matrisi
$B_{i,j;k,l}$	HSA'nın giriş sinaptik ağırlıkları
$\mathbf{B}$	HSA'nın giriş şablonu
$\mathbf{B}, \mathbf{B}$	HSA'nın giriş matrisi
$C(i, j)$	i . satırda ve j . sütunda yer alan hücre
D_1, D_2	RD-HSA'nın aktivatör ve inhibitör difüzyon katsayıları
$\mathbf{I}$	Birim matris
i, j	İki boyutlu HSA dizisinin satır ve sütun indisleri
k, l	$S_r(i, j)$ etki alanının içinde kalan hücrelerin indisleri
M, N	İki boyutlu HSA dizisinin satır ve sütun sayısı
r	Etki alanının yarıçapı
$S_r(i, j)$	$C(i, j)$ hücresinin etki alanı
$\mathbf{T}$	$\mathbf{A}$ matrisinin öz vektörlerinin oluşturduğu öz vektörler matrisi
$\mathbf{U}_{ij}$	$S_r(i, j)$ etki alanının içinde kalan giriş değişkenlerinin matrissel ifadesi
u_{ij}	$C(i, j)$ hücresinin giriş değişkeni
$\mathbf{u}$	HSA'nın giriş değişkeni vektörü
x^k	Ayrık $x(n)$ sistemin k . adımındaki değeri.
x_{ij}	$C(i, j)$ hücresinin durum değişkeni
$\mathbf{x}$	HSA'nın durum değişkeni vektörü
$\mathbf{Y}_{ij}$	$S_r(i, j)$ etki alanının içinde kalan çıkış değişkenlerinin matrisel ifadesi
y_{ij}	$C(i, j)$ hücresinin çıkış değişkeni
$\mathbf{y}$	HSA'nın çıkış değişkeni vektörü
z_{ij}	$C(i, j)$ hücresinin eşik değeri
$\mathbf{z}$	HSA'nın eşik vektörü
λ	Gabor filtresinin bant genişliği
λ_q	Bir matrisin q . öz değerleri
$\boldsymbol{\lambda}$	Öz değerler vektörü
$\boldsymbol{\Lambda}$	Öz değerler vektörünü diyagonalinde içeren öz değerler matrisi

KISALTMA LİSTESİ

CNN	Cellular Neural Networks
CNS	Central Neural System
CNs	Command Neurons
CPG	Central Pattern Generator
DSP	Digital Signal Processor
FPGA	Field Programmable Gate Array
HSA	Hücrel Sinir Ağları
IIR	Infinite Impulse Response
LMGNs	Local Motion Generating Neurons
RD-CNN	Reaction Diffusion Cellular Neural Network
TY-HSA	Tepkime Yayınımlı Hücrel Sinir Ağı

ŞEKİL LİSTESİ

	Sayfa
1.1 Bir robot sisteminin basitleştirilmiş blok diyagramı.	2
2.1 $r = 1$ ve $r = 2$ komşuluk miktarları için elde edilen etki alanı.	4
2.2 $r = 1$ komşuluklu HSA'nın sınır hücreleri.	5
2.3 Çıkış ile durum arasındaki lineer olmayan ilişki.	6
2.4 İki boyutlu HSA hücre dizisinin vektör olarak paketlenmesi.	6
2.5 HSA'da en çok kullanılan sınır koşulları.	8
3.1 Tek katmanlı bir HSA hücresinin elektriksel modeli Arena ve Fortuna (2002).	13
4.1 Merkezi hareket üreticisinin (CPG) yapısı (Arena ve Fortuna, 2002).	17
4.2 Yerel hareket üretici nöronlarının (LMGNs) bağlantı şekli.	19
4.3 Ayrıklaştırılmış TY-HSA denklemlerinin simülasyonu ile elde edilmiş olan kendini yenileyen dalgalar.	21
4.4 Dairesel bağlı TY-HSA hücreleri üzerinde oluşan kendini yenileyen dalgaların zaman eksenine göre oluşturduğu spiral.	22
4.5 Ayrıklaştırılmış TY-HSA ile elde edilmiş kendini yenileyen dalgaların yakından görünüşü.	22
4.6 Tırtılın hareket sistemi ile LMGNs biriminin birbiri ile olan bağlantısı.	23
4.7 Gerçek bir tırtılın video görüntüsünden alınmış iki kare.	24
4.8 Sanal tırtılın yürüme fazları.	24
4.9 TY-HSA ile altı bacaklı robotların hareket stratejilerinin gerçekleşmesi.	26
4.10 Altı bacaklı robotun üç ayrı hareket stratejisi için elde edilen TY-HSA simülasyonu sonuçları.	28
4.11 TY-HSA ile elde edilmiş Turing desenleri.	29
5.1 Gerçek Gauss fonksiyonu ile Gauss benzeri bir fonksiyonun karşılaştırılması.	32
5.2 Gauss benzeri süzgecin fark denkleminde elde edilmiş frekans yanıtı.	33
5.3 Farklı λ değerleri için Gauss benzeri HSA süzgecinin birim örnek ve frekans yanıtları.	35
5.4 Farklı λ değerleri için Gabor benzeri HSA süzgecinin birim örnek ve frekans yanıtları.	37
5.5 3×4 boylarındaki bir resmin Gauss-Jordan eleme yöntemi ile işlenmesi.	46
5.6 3×4 boylarındaki bir resmin Gauss eleme yöntemi ile işlenmesi.	48
5.7 3×4 boylarındaki bir resmin lineer HSA için uyarlanmış Gauss-Jordan eleme yöntemi ile işlenmesi.	53

5.8	3×4 boylarındaki bir resmin lineer HSA için uyarlanmış Gauss eleme yöntemi ile işlenmesi.	54
5.9	32×32 boylarındaki bir kare dalganın $\lambda = 0.5$ olarak alınmış bir Gauss benzeri alçak geçiren HSA süzgeç ile süzülmesi ile elde edilen sonuçlar. . .	57
5.10	32×32 boylarındaki bir kare dalganın $\lambda = 0.001$ olarak alınmış bir Gabor benzeri bant geçiren HSA süzgeç ile süzülmesi ile elde edilen sonuçlar. . . .	58

ÇİZELGE LİSTESİ

	Sayfa
5.1 MATLAB simülasyonları sonucunda $t_{\text{Euler}}/t_{\text{Gauss}}$ hız oranı için elde edilen sonuçlar.	56
6.1 FPGA veya DSP tabanlı bir sistemde gözlenmesi beklenen $t_{\text{Euler}}/t_{\text{Gauss}}$ hız oranları.	59

ÖNSÖZ

Öncelikle zorlu tez çalışmam süresince desteğinin ve sabrının yanı sıra sevgisini de benden esirgemeyen çok sevdiğim eşim Endam'a, bilgi ve deneyimlerini benimle paylaşmakla kalmayıp zor dönemlerimde bana destek olan Vedat hocama, başta Evren ve Murathan olmak üzere tez çalışmam boyunca bana destek olan tüm arkadaşlarıma sonsuz teşekkürler.

ÖZET

Kendi kendine karar verebilen ve son dönemde *akıllı* olarak tabir edilmeye başlanan sistemler her geçen gün bilim kurgu olmaktan çıkıp biraz daha günlük hayatımızdaki yerlerini almaktadır. Buradaki *akıllı* kelimesini en az yadırgayarak kabullenebileceğimiz sistemler şüphesiz robotik sistemlerdir.

Robot bilimi ve teknolojisi olarak tanımlanabilecek robotik, aynı zamanda doğa taklit etme işlemidir. Milyarlarca yıl süren ve evrimin bir parçası olan doğal ayıklanma sürecinde yalnızca en iyi olanların hayatta kalıp günümüze geldiği göz önüne alındığında doğayı taklit etmenin mantıklı olduğu görülür. Taklit işleminin inceleme alanı çok geniştir. Örneğin bir böcek türünün bacak kontrolünün veya retina yapısının incelenmesi, birbirinden çok farklı olan taklit etme işlemleridir.

Bu tez robotik ve Hücrel Sinir Ağlarının (HSA) konularının kesişim noktası olan iki parçadan oluşmaktadır. Tezin ilk yarısının başında, çok bacaklı robotların bacak hareketinin senkron bir şekilde kontrolü konusunda literatürde yer alan bazı makaleler incelenmiştir. Daha sonra HSA'nın analog ifadesinin sayısale çevrilerek Tepkime Yayımlı HSA (Reaction Diffusion CNN — RD-CNN) simülasyonunun sayısal sistemler üzerinde yapılabilceği gösterilmiştir. Son olarak elde edilen simülasyon sonuçları verilerek TY-HSA'nın sayısal devrelerle gerçekleştirilebileceği kanıtlanmıştır.

Tezin ikinci kısmında, retinadaki ön işlemleri taklit ederek özellik çıkarılması amacıyla kullanılabilcek Gauss ve Gabor benzeri lineer HSA süzgeçleri ele alınmıştır. Sonra bu süzgeçlerin ayrıklaştırılmasıyla elde edilen durum denklemlerinin çözülmesi amacıyla lineer çözüm yöntemleri ele alınmıştır. Daha sonra bellek gereksinimini azaltacak ve simülasyon hızını arttıracak bazı optimizasyonlar yapılmıştır. Son olarak simülasyon sonuçları verilerek karşılaştırmalar yapılmıştır.

Anahtar kelimeler: Hücrel sinir ağları, robotik, tepkime yayını, seyrek matrisler.

ABSTRACT

Smart systems are one of the most interesting improvements in technology that plot an evaluation trajectory from the science-fiction to the real life. Robotics is one of the few technologies that people easily accept the “intelligence” in it.

In fact robotics — besides of being the science and technology of robots — is the mimicking process of the nature. Considering many billion years of ongoing evolutions that naturally select only the survivors, it is not surprising that mimicking is one of the most common design techniques in robotics. The spectrum of the mimicking process is very wide. For example both the control problem of an insects leg and analysing the same insects retina structure could be individual mimicking processes.

This thesis consist of two parts that have both robotics and Cellular Nonlinear/Neural Networks (CNN) in common. The beginning of the first part introduces some techniques that are already introduced in the literature to control the legs of multi-pod robots synchronously. Then it is shown that it is possible to convert analog representation of the CNN to digital in order to emulate Reaction Diffusion CNN (RD-CNN) structure on a digital environment. Finally simulation results are given to show that it is possible to implement RD-CNN on a digital system.

Second part of this thesis starts with a brief introduction of the Gauss-like and the Gabor-like linear CNN filters to mimic the preprocessing procedures of the retina. Secondly some linear algebra techniques are introduced in order to solve state equations of the discrete linear CNN structure. Thirdly some optimisations have been made on these techniques that increase the calculation speed and decrease the memory requirements. Finally some simulation results are given with comparisons.

Keywords: Cellular neural networks, robotics, reaction diffusion, sparse matrices.

1 GİRİŞ

Bu tezde yapılmış olan çalışmalar her bir bölüm için aşağıdaki şekilde özetlenebilir:

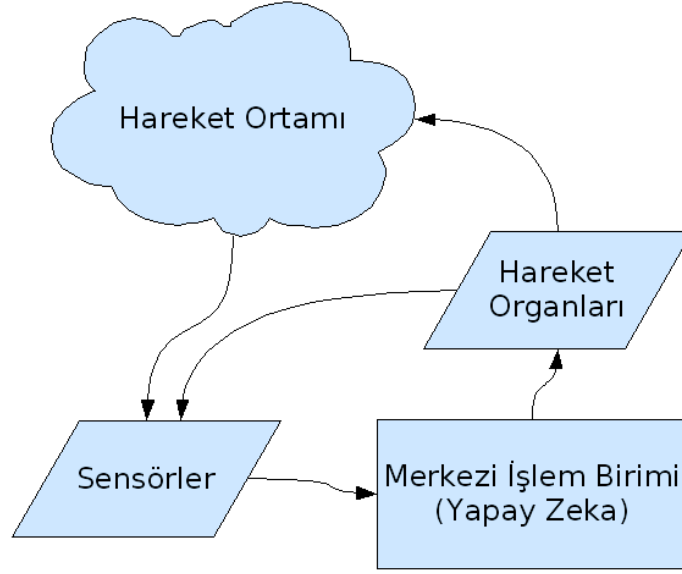
- **BÖLÜM-1** Robot ve robotik kavramlarının tanımlanması.
- **BÖLÜM-2** Hücresel Sinir Ağlarının (HSA) tanımlanması, HSA'nın matrisler ve vektörlerle ifade edilmesi, konumdan bağımsız HSA'nın tanımlanması.
- **BÖLÜM-3** İki katmanlı “tepkime yayılım (reaction diffusion)” eşitliklerinin tanımlanması, HSA'nın iki katmanlı hale getirilerek Tepkime-Yayınımlı HSA'nın (TY-HSA) tanımlanması, kendini yenileyen dalgaların ve Turing desenlerinin TY-HSA ile üretilmesi.
- **BÖLÜM-4** Çok bacaklı robotların hareket stratejilerinin belirlenmesinde ve hareket organlarının senkronizasyonunda kendini yenileyen dalgaların ve Turing desenlerinin kullanılması ile ilgili teorik bilgilerin ortaya konması, sürekli zaman TY-HSA eşitliklerinin zaman boyutunda ayrıklaştırılması ve ayrık TY-HSA simülasyonlarının sonuçları.
- **BÖLÜM-5** Robotun kameradan aldığı görüntüleri ön işlemek amacıyla; Gauss benzeri ve Gabor benzeri lineer HSA süzgeçlerinin tanımlanması, HSA'nın zaman boyutunda ayrıklaştırılması, sayısal devrelerle lineer HSA simülasyonu için kullanılacak lineer çözüm yöntemlerinin gözden geçirilmesi, lineer çözüm yöntemlerinin lineer HSA'nın durum denklemlerine göre optimizasyonu ve simülasyon sonuçları.
- **BÖLÜM-6** Elde edilen sonuçların değerlendirilmesi ve gelecekte bu tezin devamı olarak yapılabilecek çalışmaların ortaya konması.

1.1 Robot ve Robotik Kavramları

Robot kelimesi Slav dillerinde yer alan ve *iş* veya *çalışma* anlamına gelen *robota* kelimesinden türemiştir. İlk olarak Çek yazar Karel Čapek'in 1921 yılında yazdığı *Rossum'un Çok Amaçlı Robotları* adlı tiyatro oyununda karşımıza çıkan bu kelime, ünlü bilim kurgu yazarı Isaac Asimov tarafından dünyaya tanıtılmıştır. Yaklaşık bir yüzyıldan beri tartışılan ve ortak bir görüşe varılamayan robotların tanımı, bu tezin amacı da göz önüne alınarak şu şekilde yapılabilir:

Robot, insan kontrolüne dolaylı olarak bağımlı veya insan kontrolünden tamamen bağımsız olarak; içinde bulunduğu ortamı algılayabilen, algıladığı ortam değişkenlerini belli bir amaca göre işleyebilen, işlenen verilerin sonucunu en uygun fiziksel harekete dönüştürebilen her türlü yapay sisteme verilen isimdir.

Robotik, akademik ve endüstriyel anlamda robot geliştirme ve üretimi ile ilgilenen disiplinler arası bir bilim dalıdır.



Şekil 1.1: Bir robot sisteminin basitleştirilmiş blok diyagramı.

Robotik konusunda çalışan bilim adamlarının meslekleri göz önüne alındığında robotiğin; mekanikten elektroniğe, fizyolojiden metalurjiye birçok disiplinin kesişim noktası olduğu görülür. Örnek vermek gerekirse: Doğayı taklit edebilmek için fizyolojiye, sensörleri geliştirebilmek için malzeme bilimine, hareket organlarını tasarlamak için mekaniğe, sistemin alt birimleri arasındaki koordinasyonunu sağlamak ve strateji geliştirebilmek için elektroniğe ihtiyaç vardır.

Robotik sistemlerin basitleştirilmiş blok diyagramı Şekil 1.1’de verilmiştir. Robotikte elektronik konusunda yapılabilecek çalışmalar temel olarak üç ana başlık altında toplanabilir:

1. Sensörlerden alınan verilerin ön işlemlerinin yapılarak ortamla ilgili bilgilerin elde edilmesi.
2. Yapay zeka ile sensörlerden gelen bilgilerin robotun amacına uygun olarak işlenmesi ve yapılacak harekete karar verilmesi.
3. Yapılması istenen hareketin elektro–mekanik sistemler üzerinde mekanik enerjiye dönüştürülmesi.

Bu tezde robotikle ilgili yapılmış olan iki farklı çalışma üzerinde durulmuştur. Bölüm 3 ve Bölüm 4’te yer alan tezin ilk bölümünde öncelikle çok bacaklı robotların bacak hareketinin senkronizasyonunda TY–HSA kullanımı ile ilgili literatürde yer alan bazı çalışmalar ortaya konmuştur. Daha sonra zamanda sürekli ve uzayda ayırık olan TY–HSA ifadelerinin sayısal devreler üzerinde simülasyonunu yapmak amacıyla bu ifadeler zaman ekseninde ayırıklaştırılmıştır. Ayırık TY–HSA denklemlerinin MATLAB ortamında sayısal simülasyonları

yapılarak bacak senkronizasyonunu sağlayacak dalga benzeri deęişimler elde edilmiştir. Tezin ilk yarısının son aşamasında elde edilen simülasyon sonuçlarının tırtıl benzeri robotlar ile altı bacaklı robotların kontrolünde kullanılabileceęi ortaya konmuş, bazı örnekler üzerinde durulmuştur.

1.2 Lineer HSA Süzgeçleri

Bölüm 5'te yer alan ikinci çalışmada öncelikle kameradan alınan görüntülerin özelliklerini çıkartmak amacıyla kullanılabilecek Gauss ve Gabor benzeri lineer HSA süzgeçleri ele alınmıştır. Ortaya konan çalışmalar, bir önceki bölümde olduęu gibi, sayısal devreler üzerinde gerçekleştirme yapmak amacıyla ayrıklaştırılmıştır. Ayrıca Gauss ve Gabor benzeri lineer HSA süzgeçlerinin çıkışlarının, asimptotik kararlıktan yararlanarak lineer çözüm yöntemleri ile doğrudan hesaplanabileceęi gösterilmiştir. Lineer denklemlerin durum matrisleri yardımıyla çözümü ile ilgili dört farklı yöntem incelenmiş ve bu yöntemlerin ayrık lineer HSA denkleminin durum matrisi için optimize edilmesi sonucunda elde edilen hız ve bellek kazançları karşılaştırılmıştır.

2 HÜCRESEL SİNİR AĞLARI (HSA)

Bu bölümde Hücresel Sinir Ağları ile ilgili temel tanımlar yapılarak matematiksel gösterim şekilleri ortaya konmuştur (Chua ve Roska, 2002).

2.1 HSA'nın Tanımlanması

Bir HSA yapısı aşağıdaki şekilde tanımlanabilir:

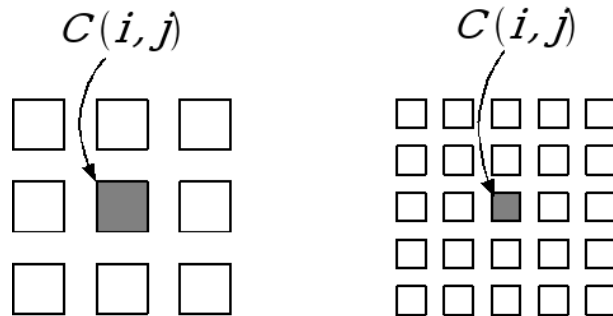
Tanım 1. *Standart HSA yapısı*, $C(i, j)$ hücrelerinin oluşturduğu $M \times N$ boylarında iki boyutlu bir dizi olarak tanımlanabilir. Buradaki (i, j) kartezyen koordinat indisleri $i = 1, 2, \dots, M$ ve $j = 1, 2, \dots, N$ şeklindedir. Aşağıda standart HSA'nın hücre yapısı verilmiştir:

	1	2	3	...	j	...	N
1	□	□	□	...	□	...	□
2	□	□	□	...	□	...	□
3	□	□	□	...	□	...	□
...	...	...	...	...	...	...	...
i	□	□	□	...	$C(i, j)$	...	□
...	...	...	...	...	...	...	...
M	□	□	□	...	□	...	□

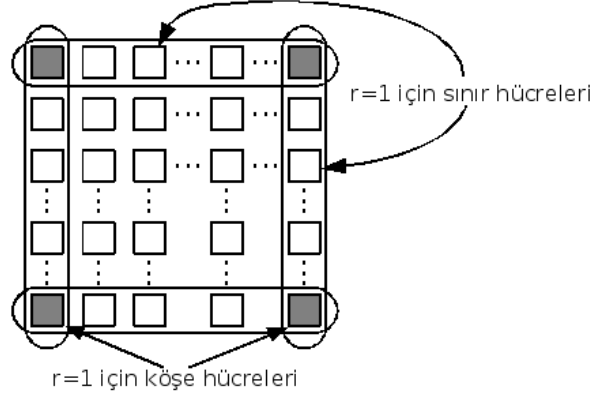
Tanım 2. *Etki alanı* $S_r(i, j)$, $C(i, j)$ hücresinin kendisinin r komşuluğuna kadar olan hücrelerinin oluşturduğu bir kümedir. r pozitif bir tamsayı olmak üzere $S_r(i, j)$ aşağıdaki özellikleri sağlar:

$$S_r(i, j) = \{C(k, l) | 1 \leq k \leq M, 1 \leq l \leq N, \max\{|k - i|, |l - j|\} \leq r\}$$

Şekil 2.1'de sırasıyla 1 ve 2 komşuluk için $S_r(i, j)$ etki alanı gösterilmiştir. Her bir hücrenin yalnızca kendi etki alanının elemanı olan hücrelerle sinaptik bağlantıları vardır.



Şekil 2.1: $r = 1$ ve $r = 2$ komşuluk miktarları için elde edilen etki alanı.



Şekil 2.2: $r = 1$ komşuluklu HSA'nın sınır hücreleri.

$r > \max\{M, N\}/2$ koşulu altında tüm hücreler birbiri ile bağlı hale gelir ve HSA sistemi Hopfield sistemine dönüşmüş olur.

Tanım 3. *Düzenli hücreler ve sınır hücreleri*, HSA sistemini oluşturan hücrelerin sınıflandırabileceği hücre türleridir. Eğer bir $C(i, j)$ hücresinin $S_r(i, j)$ etki alanı içerisindeki bütün hücreler mevcut ise bu hücreye düzenli hücre denir. Ancak etki alanı içerisindeki bazı hücreler eksik ise $C(i, j)$ hücresinin kenardaki veya kenara yakın bir hücre olduğu anlaşılır ve bu hücreye sınır hücresi adı verilir. Şekil 2.2'de görüldüğü üzere $r = 1$ için yalnızca kenarların üzerindeki hücreler sınır hücreleridir.

Tanım 4. *Standart HSA* bir durum denklemi ve çıkış ile durum arasındaki ilişkiyi belirten bir eşitlik ile tanımlanabilir. Bunların yanı sıra sınır koşullarının ve ilk koşulların da tanımlanması gerekir.

1. *Durum denklemi*

$$\dot{x}_{ij} = -x_{ij} + \sum_{C(k,l) \in S_r(i,j)} A_{i,j;k,l} y_{kl} + \sum_{C(k,l) \in S_r(i,j)} B_{i,j;k,l} u_{kl} + z_{i,j}$$

olarak tanımlanır. Burada $x_{ij} \in \mathbb{R}$, $y_{ij} \in \mathbb{R}$, $u_{ij} \in \mathbb{R}$ ve $z_{ij} \in \mathbb{R}$ sırasıyla $C(i, j)$ hücresinin *durum*, *çıkış*, *giriş* ve *eşik* değişkenleridir. $A_{i,j;k,l}$ ve $B_{i,j;k,l}$ ise sırasıyla *geri besleme* ve *giriş* sinaptik operatörleridir.

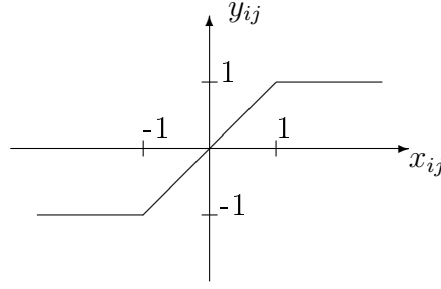
2. *Çıkış ile durum arasındaki ilişki*

$$y_{ij} = f(x_{ij}) = \frac{1}{2}|x_{ij} + 1| - \frac{1}{2}|x_{ij} - 1|$$

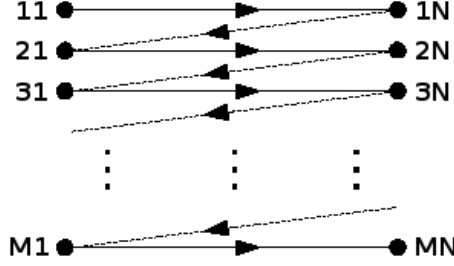
şeklinde tanımlanır (Şekil 2.3).

3. *Sınır koşulları*, $S_r(i, j)$ etki alanının içinde yer almasına rağmen $M \times N$ 'lik iki boyutlu dizinin dışında kalan eksik $C(i, j)$ hücrelerinin bilinmeyen y_{kl} ve u_{kl} değişkenlerine verilen değerlerdir.

4. *İlk koşullar*, $t = 0$ anındaki $x_{ij}(0)$ değerleridir.



Şekil 2.3: Çıkış ile durum arasındaki lineer olmayan ilişki.



Şekil 2.4: İki boyutlu HSA hücre dizisinin vektör olarak paketlenmesi.

2.2 HSA'nın Vektörel Olarak İfade Edilmesi

En genel olarak $p = M N$ tane adi diferansiyel denklem ile HSA tanımlanabilir. Her bir diferansiyel denklem

$$\dot{x}_{ij} = h_{ij}(\tilde{x}_{ij}, t); \quad i = 1, 2, \dots, M; \quad j = 1, 2, \dots, N$$

formundadır ve

$$h_{ij}(\tilde{x}_{ij}, t) = -x_{ij}(t) + \sum_{C(k,l) \in S_r(i,j)} A_{i,j;k,l} y_{kl}(t) + s_{ij}(t)$$

şeklinde tanımlıdır. $y_{kl}(t)$ ve $s_{ij}(t)$ ise

$$y_{kl}(t) = f(x_{kl}(t))$$

ve

$$s_{ij}(t) = \sum_{C(k,l) \in S_r(i,j)} B_{i,j;k,l} u_{kl}(t) + z_{i,j}(t)$$

olarak tanımlanabilir.

Vektörel gösterimin kullanılabilmesi için öncelikle iki boyutlu dizinin paketlenerek tek boyutlu bir vektör haline getirilmesi gerekir. Bu işlem için kullanılabilecek birçok yöntem olmasına karşın bu tezde Şekil 2.4'te verilmiş olan paketleme şeklinin kullanılması uygun

görülmüştür. Paketlenmiş HSA dizisinin vektörel gösterimi

$$\dot{\mathbf{x}} = -\mathbf{A} + \mathbf{A}\mathbf{y} + \mathbf{B}\mathbf{u} + \mathbf{z}$$

şeklindedir. Buradaki $\mathbf{x}$, $\mathbf{y}$, $\mathbf{u}$ ve $\mathbf{z}$ sırasıyla durum, çıkış, giriş ve eşik vektörleridir ve her biri Şekil 2.4'teki şekilde paketlenmiştir. $\mathbf{A}$ ve $\mathbf{B}$ ise durum ve giriş matrisleridir. Bu matrisler r komşuluk oranının küçük olması durumunda çok sayıda sıfır elemanına sahip olan seyrek matrislerdir.

2.3 HSA'nın Sınır Koşulları

r komşuluk miktarına sahip bir HSA'da sınır koşulları göz önüne alınırken öncelikle $M \times N$ boylarındaki hücre dizisinin $(M+2r) \times (N+2r)$ boylarında bir dizinin ortasına oturtulmuş gibi düşünülmesi gerekir. Bu şekilde $M \times N$ boylarındaki dizinin kenardaki elemanlarının eksik olan sinaptik bağlantıları tamamlanmış olur. $M \times N$ 'lik alanın dışında kalan $C(i, j)$ hücrelerine verilecek değerler sınır koşullarıdır. $r = 1$ komşuluğa ve dolayısıyla 3×3 boylarındaki etki alanına sahip bir HSA için en çok kullanılan sınır koşulları aşağıdaki şekilde sıralanabilir.

1. *Sabit (Dirichlet) sınır koşulları:* Eksik hücrelerin x ve u değişkenlerine sabit değerler vermek şeklinde uygulanır. Genelde sıfır olarak alınmakla beraber bu konuyla ilgili bir sınırlama yoktur. Sabit gerilim kaynağı veya toprak olarak düşünülebilecek sinaptik bağlantılardır (Şekil 2.5(a)). Daha çok işlemlerin basitleştirilmesi amacıyla kullanılır.

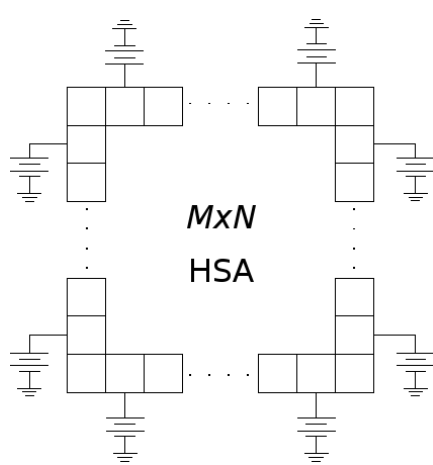
$$\begin{array}{lll} \text{Soldaki eksik hücreler:} & y_{i,0} = \alpha_{i,0}; & u_{i,0} = \beta_{i,0}; & i = 1, 2, \dots, M \\ \text{Sağdaki eksik hücreler:} & y_{i,N+1} = \alpha_{i,N+1}; & u_{i,N+1} = \beta_{i,N+1}; & i = 1, 2, \dots, M \\ \text{Üstteki eksik hücreler:} & y_{0,j} = \alpha_{0,j}; & u_{0,j} = \beta_{0,j}; & j = 1, 2, \dots, N \\ \text{Altındaki eksik hücreler:} & y_{M+1,j} = \alpha_{M+1,j}; & u_{M+1,j} = \beta_{M+1,j}; & j = 1, 2, \dots, N \end{array}$$

2. *Zero-flux (Neumann) sınır koşulları:* Gereken her yerde eksik hücrelerin yerine eksik hücrenin en yakın gerçek komşusu kullanılması şeklinde uygulanır. Örneğin sol kenarın dışındaki eksik bir hücrenin değerlerinin yerine doğrudan sol kenardaki hücrenin değerleri alınır (Şekil 2.5(b)). Özellikle termodinamik yasalarına göre enerji sızıntısından sönmümlenme riski taşıyan sistemlerde kullanılır.

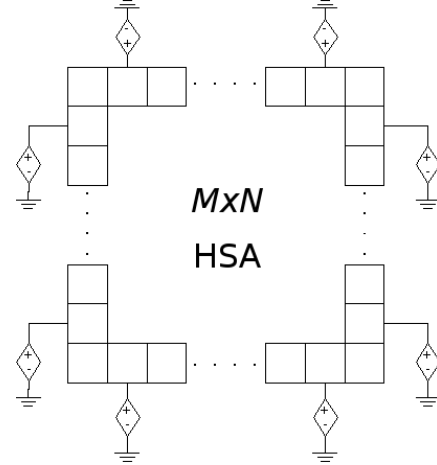
$$\begin{array}{lll} \text{Soldaki eksik hücreler:} & y_{i,0} = y_{i,1}; & u_{i,0} = u_{i,1}; & i = 1, 2, \dots, M \\ \text{Sağdaki eksik hücreler:} & y_{i,N+1} = y_{i,N}; & u_{i,N+1} = u_{i,N}; & i = 1, 2, \dots, M \\ \text{Üstteki eksik hücreler:} & y_{0,j} = y_{1,j}; & u_{0,j} = u_{1,j}; & j = 1, 2, \dots, N \\ \text{Altındaki eksik hücreler:} & y_{M+1,j} = y_{M,j}; & u_{M+1,j} = u_{M,j}; & j = 1, 2, \dots, N \end{array}$$

3. *Periyodik (toroid) sınır koşulları:* İki boyutlu HSA dizisi bir simidin yüzeyi olarak düşünülür. İki boyutlu dizi sol kenarı sağ kenarına komşu olacak şekilde silindir şekline getirilir, silindirin üst ve alt yüzeyleri de silindir bükülerek birleştirilirse iki boyutlu bir toroid, yani bir simit şekli elde edilir (Şekil 2.5(c)).

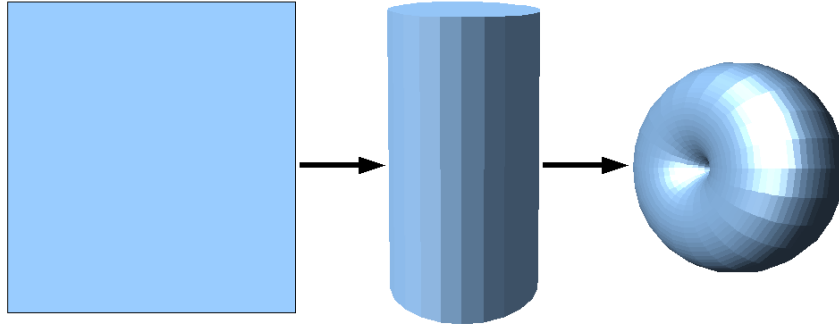
$$\begin{aligned}
 \text{Soldaki eksik hücreler: } & y_{i,0} = y_{i,N}; & u_{i,0} = u_{i,N}; & i = 1, 2, \dots, M \\
 \text{Sağdaki eksik hücreler: } & y_{i,N+1} = y_{i,1}; & u_{i,N+1} = u_{i,1}; & i = 1, 2, \dots, M \\
 \text{Üstteki eksik hücreler: } & y_{0,j} = y_{N,j}; & u_{0,j} = u_{N,j}; & j = 1, 2, \dots, N \\
 \text{Altındaki eksik hücreler: } & y_{M+1,j} = y_{1,j}; & u_{M+1,j} = u_{1,j}; & j = 1, 2, \dots, N
 \end{aligned}$$



(a) Sabit (Dirichlet) Sınır Koşulları.



(b) Zero-flux (Neumann) sınır koşulları.



(c) Periyodik (Toroidal) sınır koşulları.

Şekil 2.5: HSA'da en çok kullanılan sınır koşulları.

2.4 Konumdan Bağımsız HSA

Herhangi bir hücrenin sinaptik bağlantılarındaki ağırlıklar diğer tüm hücreler için de aynı ise HSA konumdan bağımsızdır. Bu koşul altında $r = 1$ komşuluk için A ve B sinaptik bağlantı ağırlıkları 3×3 boylarında bir şablon olarak düşünülebilir. Bu durumda sinaptik ağırlıklar ile çarpma ve toplama işlemi konvolüsyona benzer bir yapıda ifade edilebilir.

Öncelikle $S_1(i, j)$ etki alanının

$$S_1(i, j) = \begin{array}{|c|c|c|} \hline C(i-1, j-1) & C(i-1, j) & C(i-1, j+1) \\ \hline C(i, j-1) & C(i, j) & C(i, j+1) \\ \hline C(i+1, j-1) & C(i+1, j) & C(i+1, j+1) \\ \hline \end{array}$$

şeklinde olacağı söylenebilir. Bir $C(i, j)$ hücresinin $S_1(i, j)$ etki alanındaki hücrelerle üç farklı sinaptik bağlantısı vardır.

1. *Çıkıştan gelen* $A_{i,j;k,l}$: Sinaptik ağırlıkların konumdan bağımsız olduğu durumda $a_{mn} = A(m, n)$ olarak tanımlanacak olursa HSA'nın durum denklemindeki toplam ifadesi

$$\begin{aligned} \sum_{C(k,l) \in S_r(i,j)} A_{i,j;k,l} y_{kl} &= \sum_{|k-i| \leq 1} \sum_{|l-j| \leq 1} A_{k-i, l-j} y_{kl} \\ &= \begin{array}{ccc} a_{-1,-1}y_{i-1,j-1} & +a_{-1,0}y_{i-1,j} & +a_{-1,1}y_{i-1,j+1} \\ +a_{0,-1}y_{i,j-1} & +a_{0,0}y_{i,j} & +a_{0,1}y_{i,j+1} \\ +a_{1,-1}y_{i+1,j-1} & +a_{1,0}y_{i+1,j} & +a_{1,1}y_{i+1,j+1} \end{array} \\ &= \sum_{k=-1}^1 \sum_{l=-1}^1 a_{kl} y_{i+k, j+l} \end{aligned}$$

şeklinde yazılabilir. Daha sonra *geri besleme şablonu* $\mathbf{A}$; y_{ij} çıkış değişkeninin etki alanı matrisi $\mathbf{Y}_{ij}$ ve *şablon nokta çarpımı* $\circledast$ ile

$$\sum_{C(k,l) \in S_r(i,j)} A_{i,j;k,l} y_{kl} \triangleq \mathbf{A} \circledast \mathbf{Y}_{ij}$$

ve

$$\mathbf{A} \circledast \mathbf{Y}_{ij} \triangleq \begin{array}{|c|c|c|} \hline a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ \hline a_{0,-1} & a_{0,0} & a_{0,1} \\ \hline a_{1,-1} & a_{1,0} & a_{1,1} \\ \hline \end{array} \circledast \begin{array}{|c|c|c|} \hline y_{i-1,j-1} & y_{i-1,j} & y_{i-1,j+1} \\ \hline y_{i,j-1} & y_{i,j} & y_{i,j+1} \\ \hline y_{i+1,j-1} & y_{i+1,j} & y_{i+1,j+1} \\ \hline \end{array}$$

tanımları yapılabilir.

2. *Girişten gelen* $B_{i,j;k,l}$: Bir önceki adımda geri besleme şablonunu elde edilirken yapılan işlemler $B_{i,j;k,l}$ giriş sinaptik ağırlıkları için tekrarlanırsa

$$\begin{aligned} \sum_{C(k,l) \in S_r(i,j)} B_{i,j;k,l} u_{kl} &= \sum_{|k-i| \leq 1} \sum_{|l-j| \leq 1} B_{k-i, l-j} u_{kl} \\ &= \begin{array}{ccc} b_{-1,-1}u_{i-1,j-1} & +b_{-1,0}u_{i-1,j} & +b_{-1,1}u_{i-1,j+1} \\ +b_{0,-1}u_{i,j-1} & +b_{0,0}u_{i,j} & +b_{0,1}u_{i,j+1} \\ +b_{1,-1}u_{i+1,j-1} & +b_{1,0}u_{i+1,j} & +b_{1,1}u_{i+1,j+1} \end{array} \\ &= \sum_{k=-1}^1 \sum_{l=-1}^1 b_{kl} u_{i+k, j+l} \end{aligned}$$

şeklinde yazılabilir. Daha sonra *giriş şablonu* $\mathbf{B}$; u_{ij} giriş değişkeninin etki alanı matrisi $\mathbf{U}_{ij}$ ve *şablon nokta çarpımı* $\circledast$ ile

$$\sum_{C(k,l) \in S_r(i,j)} B_{i,j;k,l} u_{kl} \triangleq \mathbf{B} \circledast \mathbf{U}_{ij}$$

ve

$$\mathbf{B} \circledast \mathbf{U}_{ij} \triangleq \begin{array}{|c|c|c|} \hline b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ \hline b_{0,-1} & b_{0,0} & b_{0,1} \\ \hline b_{1,-1} & b_{1,0} & b_{1,1} \\ \hline \end{array} \circledast \begin{array}{|c|c|c|} \hline u_{i-1,j-1} & u_{i-1,j} & u_{i-1,j+1} \\ \hline u_{i,j-1} & u_{i,j} & u_{i,j+1} \\ \hline u_{i+1,j-1} & u_{i+1,j} & u_{i+1,j+1} \\ \hline \end{array}$$

tanımları yapılabilir.

3. *Eşik terimi* z_{ij} : Eşik terimi de konumdan bağımsız olduğuna göre $z_{ij} \triangleq z$ olarak yazılabilir.

Sonuçta konumdan bağımsız sinaptik ağırlıklara sahip bir HSA, bu bölümde gösterilen notasyonlar ve şablon nokta çarpımı operatörü kullanılarak

$$\dot{x}_{ij} = -x_{ij} + \mathbf{A} \circledast \mathbf{Y}_{ij} + \mathbf{B} \circledast \mathbf{U}_{ij} + z$$

şeklinde tanımlanabilir.

3 TEPKİME YAYINIMLI HSA (TY-HSA)

Biyolojik yapıların matematiksel olarak modellenmesinde çoğunlukla zamanda sürekli ve uzaysal dağılımı olan denklemler karşımıza çıkar. Bu tip biyolojik modellemelerden en iyi bilinen ve en çok kullanılanlardan biri *tepkime yayılım (reaction diffusion) eşitlikleridir*. İki katmanlı bir tepkime yayılım eşitliğinin genel ifadesi (3.1)'de verilmiştir.

$$\begin{aligned}\frac{\partial A_c}{\partial t} &= F_1(A_c, I_c) + D_A \nabla^2 A_c \\ \frac{\partial I_c}{\partial t} &= F_2(A_c, I_c) + D_I \nabla^2 I_c\end{aligned}\tag{3.1}$$

Burada sözü geçen A_c ve I_c , aktivatör ve inhibitör kimyasallarının, nöron bağlantı noktalarındaki yoğunluklarını göstermektedir (Murray, 2002). $F_1(A_c, I_c)$ ve $F_2(A_c, I_c)$ lineer olmayan birer fonksiyondur. D_A ve D_I sırasıyla aktivatör ve inhibitörlerin difüzyon katsayıları, ∇^2 ise iki boyutlu Laplacian operatörüdür.

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\tag{3.2}$$

Tepkime yayılım eşitlikleri yalnızca biyolojik yapılarda değil, doğadaki birçok dağılımda da karşımıza çıkar. Aşağıda bu tür kısmi türevli diferansiyel denklemlerin doğada karşımıza çıktığı bazı yerler ve olaylar sıralanmıştır.

- Kendinden geri beslemeli, girişe ihtiyaç duymadan salınım yapan sistemler;
- patlama sonrası ortama saçılan madde konsantrasyonları;
- morfo–genetik desen oluşumları;
- kapalı bir odadaki kablosuz haberleşme taşıyıcı işaretinin dağılımı
- çöl rüzgarlarının oluşturduğu kum tepelerinin şekli;
- meteorolojik hava olaylarının bir bölgedeki dağılımı vb.

Bu tezde yukarıda verilen kısmi türevli diferansiyel denklemler ile ilgili üzerinde durulacak olan konular, HSA ile kendini yenileyen dalga ve Turing deseni oluşturma ile sınırlı kalacaktır. Literatürde yer alan birçok yayında, HSA ile kendini yenileyen dalga ve Turing deseni oluşturma üzerine çalışmalar yapılmıştır (Chua, 1995; Goras vd., 1995). Ayrıca bu yayınları temel alarak fiziksel uygulama yapan yayınlar da literatürde mevcuttur. Tezin bu bölümünde üzerinde durulacak olan konulara Arena ve Fortuna (2002) tarafından yapılan çalışmalar temel oluşturmıştır.

3.1 Kendini Yenileyen Dalgalar ve Turing Desenleri

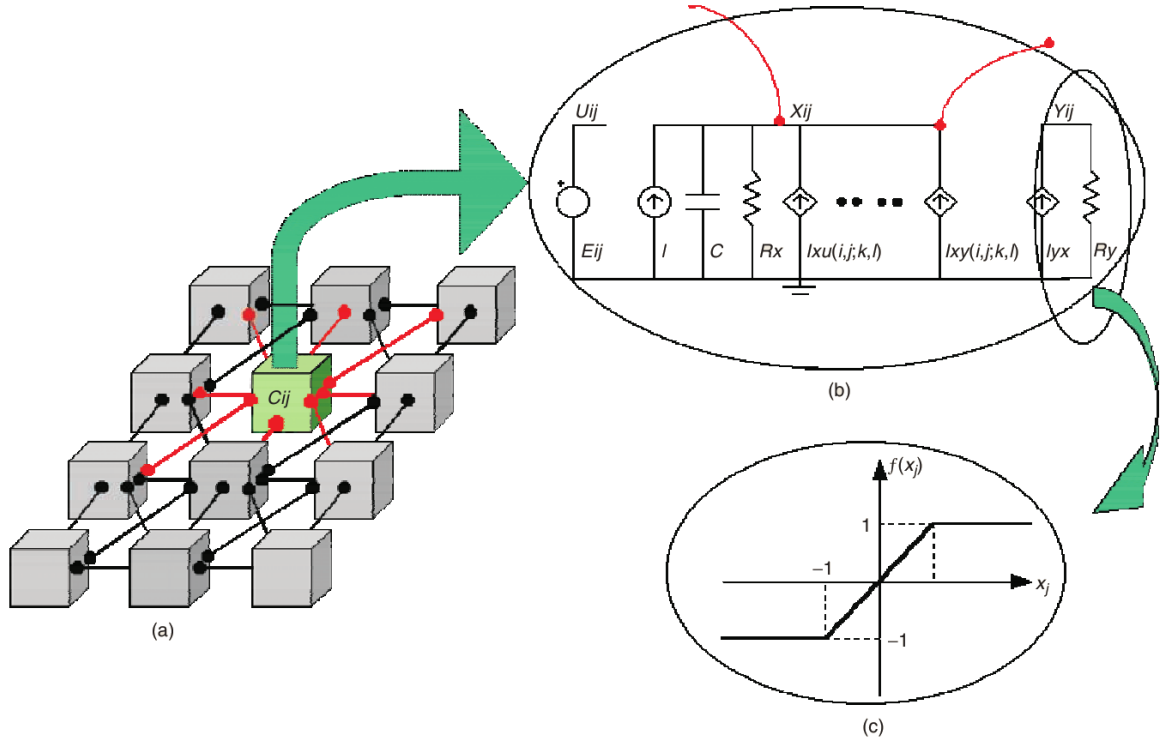
Kendini yenileyen dalgalar (autowave); dalga benzeri, salınım yapan, kendini şeklini dışardan bir giriş olmadan koruyabilen oluşumlardır (Krinsky, 1984). Ayrıca (3.1)'in çözümü ile durağan dalgalara benzer yapılar da elde etmek mümkündür. Kendini yenileyen dalga kavramı ilk olarak Khorhlov tarafından ortaya atılmıştır. Dalga benzeri bir yapıda olmalarına karşın kendini yenileyen dalgalar propagasyon sırasında şekil değiştirmez, yansıma veya girişim özelliği göstermezler. Dalgalar ve kendini yenileyen dalgalar arasındaki ortak nokta, her ikisinin kırınım özelliği göstermesidir. (3.1) eşitliklerini kullanarak kendini yenileyen dalga oluşturmanın birincil koşulu $D_A \cong D_I$ olmasıdır.

Kendini yenileyen dalgaların doğada gözleendiği yerlere bakıldığında; mürekkep balığı, solucan ve ahtapot benzeri, basit hareket organlarına sahip, yumuşak dokuları yardımıyla hareket eden birçok hayvanın, yumuşak dokuları üzerine indüklenen kendini yenileyen dalgalar aracılığıyla hareket ettikleri görülür. Daha gelişmiş canlılara, örneğin örümceklere ve kırkayaklara bakıldığında, sinir sistemi ile çoklu bacak sistemi arasında yüksek karmaşıklık düzeyinde bir bağlantı olduğu görülür. Burada hareket kavramı, bacak üzerinde gezinen dalga benzeri bir yapı olmaktan çıkar ve *merkezi hareket üretici* (CPG) tarafından oluşturulmuş birbiri ile faz farkı olan kendini yenileyen dalgaların senkron bir şekilde bacaklara uygulanması şekline dönüşür. CPG biriminin görevi yapılacak hareketi yürüme, koşma ve yüzmeye gibi farklı görevlere göre organize etmektir.

Turing desenleri ise ilk kez Turing (1952) tarafından ortaya atılmıştır. Homojen kimyasal madde konsantrasyonuna sahip karışımların kararlılığının aktivatörler tarafından kararsız hale getirilmesi, sonrasında ise inhibitörler tarafından kararlı hale getirilerek sürekli halde heterojen bir desen oluşturulması sonucunda gözlenir (Murray, 2002). Turing desenleri de (3.1) eşitliğinin çözülmesi ile elde edilir ve kararlı bir desen oluşturabilmek için $D_A \ll D_I$ koşulunun sağlanması gerektiği söylenebilir.

3.2 HSA devresi ve matematiksel modeli

Şekil 3.1'de tek katmanlı klasik bir HSA hücresinin elektriksel modeli verilmiştir. Buradaki temel problem, HSA'yı kullanarak tepkime yayılım eşitliklerinin nasıl gerçekleşeceğinin belirlenmesidir. Daha önce değinildiği üzere tepkime yayılım eşitlikleri birer kısmi türevli diferansiyel denklem olarak tanımlıdır. HSA ile yapılacak olan, bu kısmi türevli diferansiyel denklemleri adi diferansiyel denklem takımlarına dönüştürerek çözmektir. Öncelikle



Şekil 3.1: Tek katmanlı bir HSA hücresinin elektriksel modeli Arena ve Fortuna (2002).

uzayda sürekli olan kısmi türevli diferansiyel denklem takımının uzayda ayrık olan HSA üzerine iz düşürülmesi, bir başka deyişle ayrıklaştırılması gerekir. Bu yaklaşım kısaca şu şekilde tanımlanabilir:

Kısmi türevli diferansiyel denklemler olan tepkime yayılım eşitliklerinin emülasyonu, adi türevli bir diferansiyel denklem takımı olan HSA ile gerçekleştirilebilir. Bu tür HSA yapılarına TY-HSA adı verilir (Chua, 1995).

3.3 HSA ile Kendini Yenileyen Dalga ve Turing Deseni Oluşturma

Genel TY-HSA'lar literatürde Chua (1995) ve Goras vd. (1995) gibi birçok kişi tarafından ele alınmıştır. Bu bölümde TY-HSA, kendini yenileyen dalga ve Turing deseni oluşturabilecek bir yapıda tanımlanacaktır (Arena ve Fortuna, 2002).

3.3.1 İki Katmanlı TY-HSA'nın Tanımlanması

Önerme 1. Hem kendini yenileyen dalga, hem de Turing deseni oluşturabilecek bir TY-HSA hücresinin dinamik modeli ikinci mertebeden bir sistem olarak (3.3)'teki şekilde tanımlanabilir.

$$\begin{aligned}
\dot{x}_{1;i,j} &= -x_{1;i,j} + (1 + \mu + \varepsilon)y_{1;i,j} - s_1 y_{2;i,j} + i_1 \\
&\quad + D_1(y_{1;i-1,j} + y_{1;i+1,j} + y_{1;i,j-1} + y_{1;i,j+1} - 4y_{1;i,j}) \\
\dot{x}_{2;i,j} &= -x_{2;i,j} + (1 + \mu - \varepsilon)y_{2;i,j} - s_2 y_{1;i,j} + i_2 \\
&\quad + D_2(y_{2;i-1,j} + y_{2;i+1,j} + y_{2;i,j-1} + y_{2;i,j+1} - 4y_{2;i,j}) \\
1 \leq i \leq M; \quad 1 \leq j \leq N
\end{aligned} \tag{3.3}$$

Burada y çıkışı ile x durumu arasındaki ilişki

$$\begin{aligned}
y_{q;i,j} &= 0.5 (|x_{q;i,j} + 1| - |x_{q;i,j} - 1|) \\
q = 1, 2 \quad 1 \leq i \leq M; \quad 1 \leq j \leq N
\end{aligned} \tag{3.4}$$

şeklinde tanımlanmıştır. (3.1) denklem takımındaki ∇^2 Laplacian operatörlerinin uzayda ayrıklaştırılması sonucu (3.3) numaralı TY-HSA denklemlerinin sonlarında bulunan parantez içindeki ifadeler

$$\nabla^2 y \xrightarrow{\text{Uzayda Ayrıklaştırma}} (y_{q;i-1,j} + y_{q;i+1,j} + y_{q;i,j-1} + y_{q;i,j+1} - 4y_{q;i,j}); \quad q = 1, 2$$

olarak elde edilir. Bu parantezlerin başlarında bulunan D_1 ve D_2 değişkenleri difüzyon katsayılarıdır.

Önerme 2. Eğer *Önerme 1*'deki tanımlara uygun $M \times N$ boyutlarındaki bir HSA dizisinin her hücresi için

$$\mu = 0.7, \quad \varepsilon = 0, \quad s_1 = s_2 = s = 1, \quad i_1 = -0.3, \quad i_2 = 0.3$$

olarak alınırsa, buna karşı düşen TY-HSA sistemi, kendini yenileyen dalga oluşturabilecek bir yapıya dönüşmüş olur. Aynı katsayıların

$$\mu = -0.6, \quad \varepsilon = 1.82, \quad s_1 = 2 \quad s_2 = 2.5, \quad i_1 = i_2 = 0$$

olarak alınması durumunda hücre yapısı, Arena vd. (1997, 1998) tarafından verilen Turing desenlerini oluşturabilecek bir yapıya dönüşür.

TY-HSA bu aşamadan sonra klasik HSA denklemi biçiminde tanımlanabilir (Chua ve Yang, 1988).

Tanım 5. Konumdan bağımsız şablonlara sahip iki katmanlı HSA'nın durum modeli

$$\dot{x}_{ij} = -x_{ij} + \mathbf{A} \circledast \mathbf{Y}_{ij} + \mathbf{B} \circledast \mathbf{U}_{ij} + z \quad (3.5)$$

olarak tanımlanır. Burada A , B ve z sırasıyla geri besleme, kontrol ve eşik şablonlarıdır. x , y ve u değişkenleri ise HSA sisteminin, sırasıyla durumu, çıkışı ve girişine karşılık gelmektedir.

$$x_{ij} = [x_{1;i,j} \quad x_{2;i,j}]', \quad y_{ij} = [y_{1;i,j} \quad y_{2;i,j}]', \quad u_{ij} = [u_{1;i,j} \quad u_{2;i,j}]'$$

Tanım 6. Herhangi bir T şablonu için konvolüsyon operatörü olan “ $\circledast$ ” aşağıdaki şekilde tanımlanır.

$$\mathbf{T} \circledast \mathbf{V}_{ij} = \sum_{C(k,l) \in S_r(i,j)} T_{k-i,l-j} v_{kl}$$

Önerme 3. Yukarıdaki önermeler ve tanımlar göz önüne alındığında TY-HSA şablonları

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}; \quad \mathbf{B} = 0; \quad z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

şeklinde tanımlanabilir (Arena vd., 1997). Buradaki A 'lar

$$\begin{aligned} \mathbf{A}_{11} &= \begin{bmatrix} 0 & D_1 & 0 \\ D_1 & -4D_1 + \mu + \varepsilon + 1 & D_1 \\ 0 & D_1 & 0 \end{bmatrix}; \\ \mathbf{A}_{22} &= \begin{bmatrix} 0 & D_2 & 0 \\ D_2 & -4D_2 + \mu + \varepsilon + 1 & D_2 \\ 0 & D_2 & 0 \end{bmatrix}; \\ \mathbf{A}_{12} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & -s_1 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \\ \mathbf{A}_{21} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & -s_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

şeklinde tanımlıdır. Kendini yenileyen dalgalar için

$$D_1 = D_2 = 0.1,$$

Turing desenleri için ise

$$D_1 = 0.01; \quad D_2 = 1$$

değerleri kullanılır.

3.3.2 Sınır Koşulları

Hem kendini yenileyen dalga, hem de Turing deseni oluşturmak için zero-flux (Neumann) sınır koşulları kullanılır (Chua, 1995; Chua ve Roska, 2002). Bunun nedeni sistemdeki **B** şablonunun sıfır olması, bir başka deyişle sistemin girişinin bulunmamasıdır. Girişi olmayan bir sistem termodinamik yasalarına göre enerji kaybetme eğilimindedir. Bu nedenle sınırdaki eksik hücrelerin yerine sınırdaki son hücrenin değerleri olduğu kabul edilir ve böylece **A** şablonu kenarlarda ve köşelerde kırpılmış olur.

Örnek verilecek olursa, Neumann sınır koşullarına göre sağ kenardaki bir $C(i, j)$ hücresinin sağında eksik olan $C(i, j+1)$ hücresinin etkisi, $C(i, j)$ ile eşleşecek biçimde olacaktır. Yani (3.3) eşitliğinin difüzyon kısmı mevcut halinden

$$D_k(y_{k;i+1,j} + y_{k;i-1,j} + y_{k;i,j-1} - 3y_{k;i,j}); \quad k = 1, 2$$

haline dönüşecektir. Benzer şekilde sağ-üst köşedeki hücre için de eksik olan sağdaki ve üstteki terimlerin etkisi, difüzyon ifadesini

$$D_k(y_{k;i+1,j} + y_{k;i,j-1} - 3y_{k;i,j}); \quad k = 1, 2$$

şekline dönüştürmek olacaktır.

4 TY-HSA İLE ROBOTLARIN HAREKET KONTROLÜ

Bölüm 3.1’de tepkime yayılım eşitliklerinin, biyolojik yapıların ve sinir iletim sistemlerinin modellenmesinde kullanıldığına değinilmişti. Bu bölümde ise üzerinde durulacak olan konu, kendini yenileyen dalgaların robot kontrolünde kullanılabileceği yerlerin ortaya konması olacaktır (Arena ve Fortuna, 2002).

4.1 Hayvanların Hareket Sisteminin Modellemesi

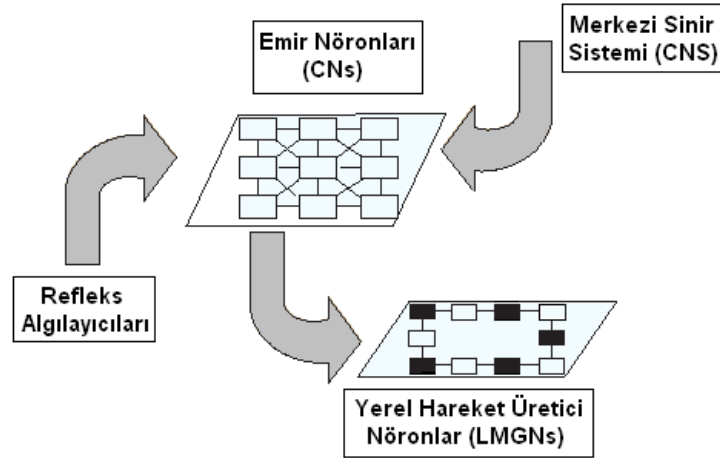
Burada öncelikle yapılması gereken, *merkezi hareket üreticisi* (Central Pattern Generator — CPG) adı verilen ve hayvanların hareketini düzenleyen birimin yapısını incelemektir. Şekil 4.1’de merkezi hareket üreticisinin genel yapısı verilmiştir. CPG biriminin alt birimleri aşağıdaki şekilde tanımlanabilir:

CNS birimi (merkezi sinir sistemi — Central Neural System), beynin bütünsel olarak çalışması sonucunda karar verilen hareket stratejisini CNs birimine iletir;

Refleks algılayıcıları, merkezi sinir sisteminden bağımsız olarak CNs birimine refleks sinyalleri gönderir;

CNs birimi (emir nöronları — Command Neurons), merkezi sinir sisteminden ve refleks algılayıcılarından gelen bilgileri birlikte değerlendirerek, yapılacak hareketi (yürü, koş, yüz, dur, vs.) kesin olarak belirlemekle yükümlüdür;

LMGNs birimi (yerel hareket üretici nöronlar — Local Motion Generating Neurons), CNs tarafından gönderilen hareket bilgisini işleyerek motor sinirlere ve dolayısıyla kaslara iletilecek sinyalleri üretmekle yükümlüdür.



Şekil 4.1: Merkezi hareket üreticisinin (CPG) yapısı (Arena ve Fortuna, 2002).

CPG biriminin yapısında görünen CNS ve refleks algılayıcılarının ürettikleri işaretleri üretme şekli, ürettikleri durumların değerlendirilmesi vb. konular bizim inceleme alanımızın dışında kalmaktadır. Burada üzerinde durulacak olan konu, LMGNs birimini gerçekleştirerek kendini yenileyen dalga üretilmesi olacaktır.

4.2 Kendini Yenileyen Dalga Oluşturma Simülasyonu

Şimdiye kadar kendini yenileyen dalgaların kısmi türevli diferansiyel denklemleri ve HSA ile bu denklemlerin emülasyonu üzerinde duruldu. Bu bölümde de HSA denklemlerinin sayısal simülasyonu ele alınacaktır.

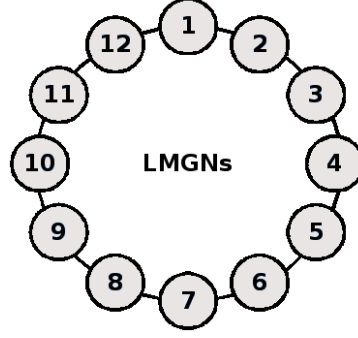
4.2.1 Kendini Yenileyen Dalga Oluşturmak İçin Gereken Sınır Koşulları ve İlk Değerler

Öncelikle iki boyutlu olan HSA sisteminin enini ve boyunu, yani (3.3) numaralı denklemdaki M ve N değerlerini belirlemek gerekmektedir. Literatürde bulunan yayınlarda tek boyuta indirgenmiş bir HSA yapısının robot kontrolünde kullanılabilecek bir kendini yenileyen dalga üretebileceği belirtilmiştir (Arena ve Fortuna, 2002). Buna göre $M = 1$ ve $N = 12$ alındığı takdirde bir kendini yenileyen dalga üretilebilir. Sınır koşullarının Bölüm 3.3.2’de belirtilen şekilde zero-flux olarak alması sonucunda HSA denklemi

$$\begin{aligned}\dot{x}_{1;j} &= -x_{1;j} + (1 + \mu + \varepsilon)y_{1;j} - s_1 y_{2;j} + i_1 + D_1(y_{1;j-1} + y_{1;j+1} - 2y_{1;j}) \\ \dot{x}_{2;j} &= -x_{2;j} + (1 + \mu - \varepsilon)y_{2;j} - s_2 y_{1;j} + i_2 + D_2(y_{2;j-1} + y_{2;j+1} - 2y_{2;j}) \\ 1 &\leq j \leq N\end{aligned}\tag{4.1}$$

haline dönüşür. LMGNs biriminin $N = 12$ için hücre yapısı Şekil 4.2’de verilmiştir. Burada sağ ve sol sınır koşullarının dairesel olarak alınmasının nedeni, üretilen autowaeelerin tüm hücreler üzerinde propagasyon yapmasını sağlamaktır.

Belirlenmesi gereken bir konu da sisteme hangi ilk değerlerin verileceği ile ilgilidir. TY-HSA girişsiz olarak kendi üzerine geri besleme mantığıyla çalıştığından dolayı sistemi kararlı bir salınıma sokmak için x durumlarına belli ilk değerler verilmesi gerekir. Kendini yenileyen dalga oluşturmak için öncelikle 1 ve 12 numaralı nöronların arasındaki bağlantı kopararak halka yapısı geçici olarak bozulur. Daha sonra tüm nöronların $x_{q;j}$; $q = 1, 2$; $j = 2, 3, \dots, 12$ durumlarına sıfıra yakın değerler verilirken en baştaki nöronun durumlarına $x_{1;1} = 1V$ ve $x_{2;1} = -1V$ ilk değerleri verilir. Baştaki nöronun durum gerilim seviyeleri



Şekil 4.2: Yerel hareket üretici nöronlarının (LMGNs) bağlantı şekli.

bir süre sabit kalmaya zorlanarak diğer hücrelerin uygun ilk değerler alması sağlanır. Son olarak baştaki nöronun da durumları serbest bırakılır ve baştaki nöron ile sondaki nöron birbirine bağlanarak halka yapısı tekrar oluşturulur.

Yapılan bu işlemler sonucunda tüm hücreler üzerinde aynı kendini yenileyen dalga sabit bir faz farkı ile oluşmuş olur. Faz farkı

$$\phi_A = \frac{2\pi}{N}$$

olarak ifade edilebilir. Bu durumda $N = 12$ için birbirine komşu olan hücreler üzerinde $\phi = 30^\circ$ 'lik faz farkı ile aynı salınımı yapan kendini yenileyen dalgalar elde edilir.

4.2.2 Sürekli–Zaman TY–HSA Denkleminin Ayırıklaştırılması

Simülasyonunu yapılacak olan TY–HSA denklemi uzayda ayırık, ancak zamanda sürekli bir adi diferansiyel denklem takımıdır. Bu durumda sürekli olan zaman boyutunun belli bir yaklaşıklıkla ayırıklaştırılması gerekmektedir. Zaman değişkeni ayırıklaştırılır ve $\dot{x}_{ij}$ türev ifadesine *Euler ileri yaklaşıklığı* uygulanırsa

$$\begin{aligned} \dot{x}_{ij} &= \frac{dx_{ij}}{dt} = \lim_{T_s \rightarrow 0} \frac{x_{ij}([n+1]T_s) - x_{ij}(nT_s)}{T_s} \\ &\cong \frac{x_{ij}([n+1]T_s) - x_{ij}(nT_s)}{T_s} \\ \frac{dx_{ij}}{dt} &\triangleq \frac{x_{ij}(n+1) - x_{ij}(n)}{T_s} \triangleq \frac{x_{ij}^{n+1} - x_{ij}^n}{T_s} \end{aligned} \quad (4.2)$$

elde edilir. (3.5) ve (4.2)'den yararlanarak

$$\dot{x}_{ij} = -x_{ij} + \mathbf{A} \circledast \mathbf{Y}_{ij} + \mathbf{B} \circledast \mathbf{U}_{ij} + I; \quad \mathbf{B} = 0$$

$$\begin{aligned}\frac{x_{ij}^{n+1} - x_{ij}^n}{T_s} &= -x_{ij}^n + \mathbf{A} \circledast \mathbf{Y}_{ij}^n + I \\ x_{ij}^{n+1} &= x_{ij}^n + T_s(-x_{ij}^n + \mathbf{A} \circledast \mathbf{Y}_{ij}^n + I)\end{aligned}\tag{4.3}$$

ayrık TY–HSA ifadesi elde edilir.

4.2.3 Kendini Yenileyen Dalga Oluşturma Simülasyonu Sonuçları

Simülasyonun gerçekleştirilebilmesi için iki parametrenin sayısal değerinin önceden belirlenmesi gerekmektedir:

- Örnekleme frekansı T_s ;
- Baştaki nöronun birinci ve ikinci katmanındaki durum değişkenlerine verilen $x_{1;1} = 1V$ ve $x_{2;1} = -1V$ ilk koşullarının sabit tutulması gereken N_i adım sayısı.

Bu aşamada bilinmeyenleri hesaplamanın bir yolu bulunamamıştır. Bu nedenle bilinmeyenler simülasyon sırasında deneme ve yanılma yoluyla elde edilmiştir. Yapılan denemeler sonucunda

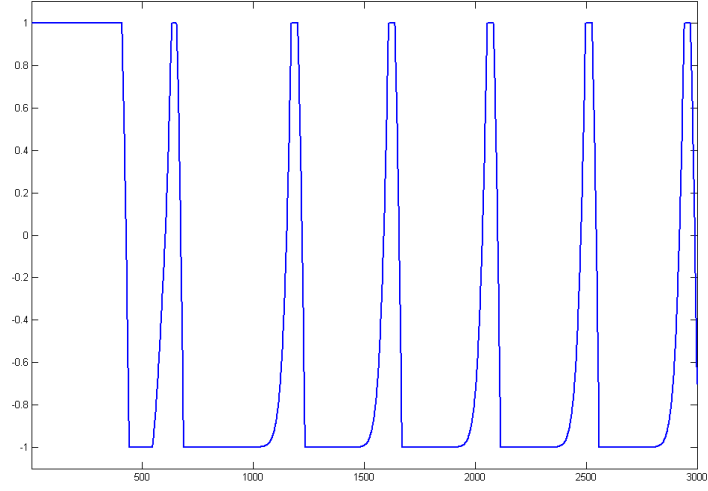
$$T_s = 0.05; \quad N_i = 400$$

olduğunda optimum sonucun elde edildiği gözlenmiştir.

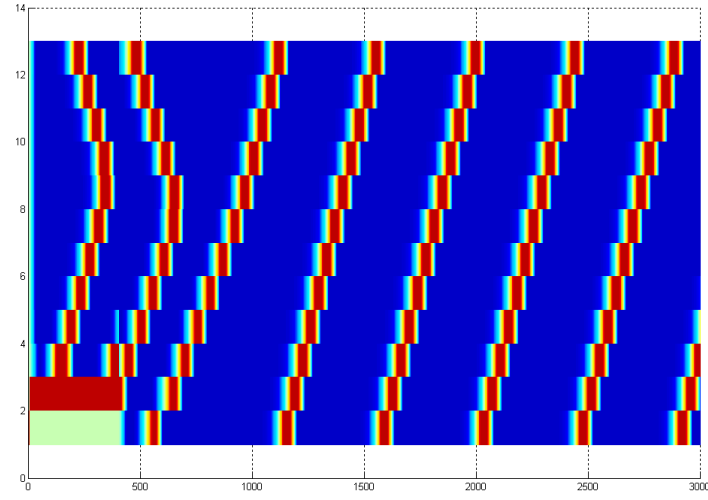
Simülasyon sonucunda elde edilen kendini yenileyen dalgalar Şekil 4.3(a) ve Şekil 4.3(b)'de, ilk koşulların kısa tutulması veya T_s 'in büyük seçilmesi sonucu elde edilecek hatalı çıkış ise Şekil 4.3(c)'de verilmiştir. Her üç şekil için yatay eksen zaman eksenidir; dikey eksen Şekil 4.3(a) için çıkış gerilimi, Şekil 4.3(b) ve Şekil 4.3(c) için ise hücre indisi eksenleridir. Şekil 4.4'te ise dairesel olarak dizilmiş TY–HSA hücreleri üzerinde oluşan kendini yenileyen dalgaların zaman eksenine göre oluşturduğu üç boyutlu spiral görünmektedir.

Bu simülasyonlarda T_s ve N_i parametrelerinin değiştirilmesi sonucunda elde edilen gözlem sonuçları aşağıdaki gibidir:

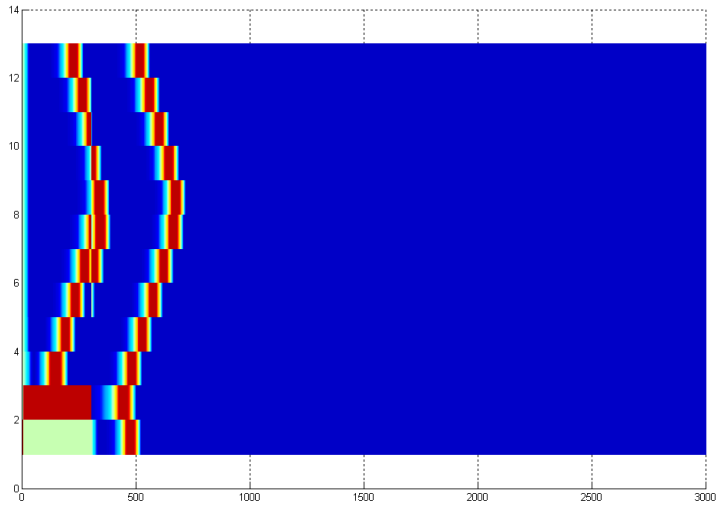
- Örnekleme frekansı T_s büyüdükçe ayrık TY–HSA sisteminin yakınsaklığı bozulmakta ve salınım oluşamamaktadır.
- Örnekleme frekansı T_s çok küçüldüğünde sonuca ulaşmak için gereken simülasyon adım sayısı çok artmaktadır.
- İlk koşulların sabit tutulması gereken N_i adım sayısı küçüldüğünde sistem kararlı bir salınıma başlayamamaktadır.



(a) İlk değer verilen hücre için beklenen çıkış.

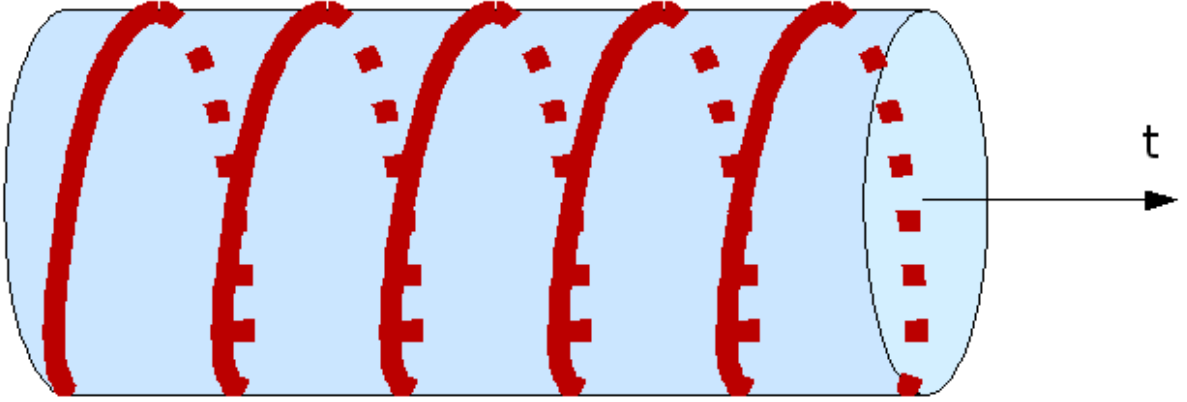


(b) Tüm hücreler için beklenen çıkış.

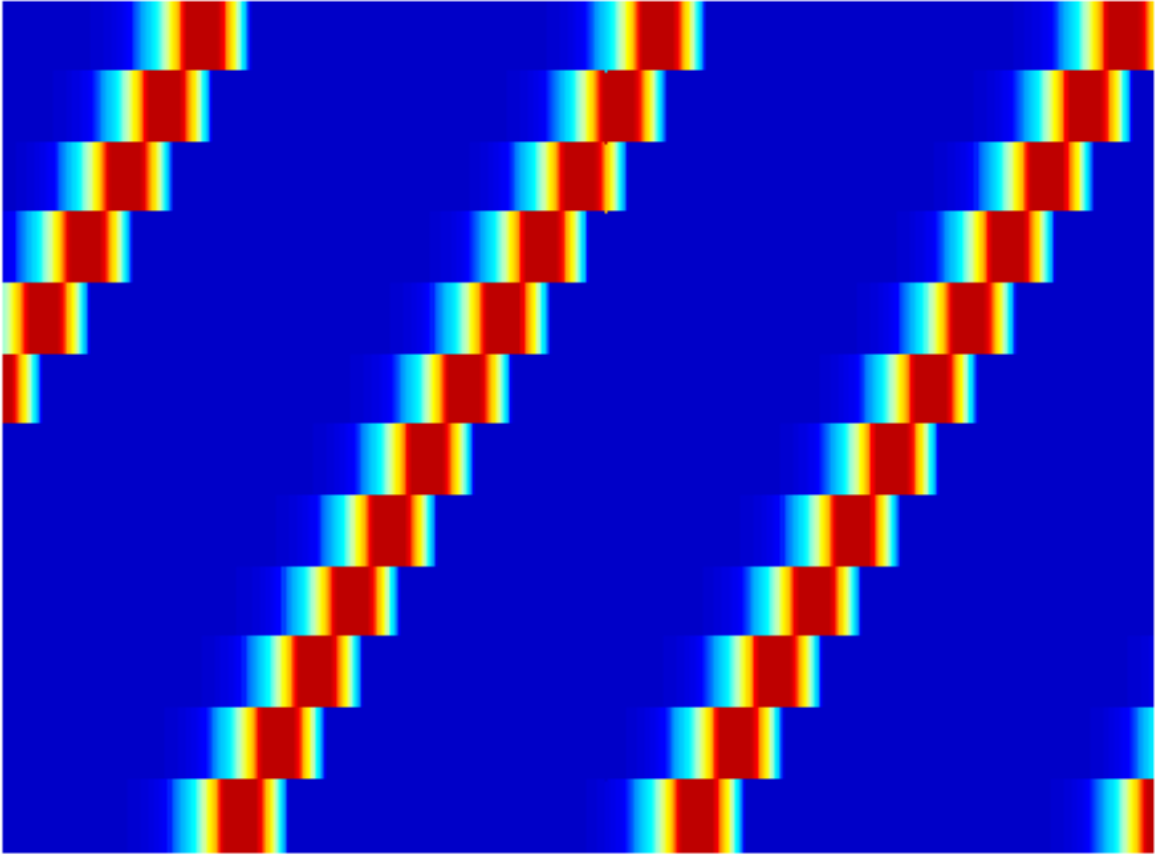


(c) Hatalı durum.

Şekil 4.3: Ayrıklaştırılmış TY-HSA denklemlerinin simülasyonu ile elde edilmiş olan kendini yenileyen dalgalar.



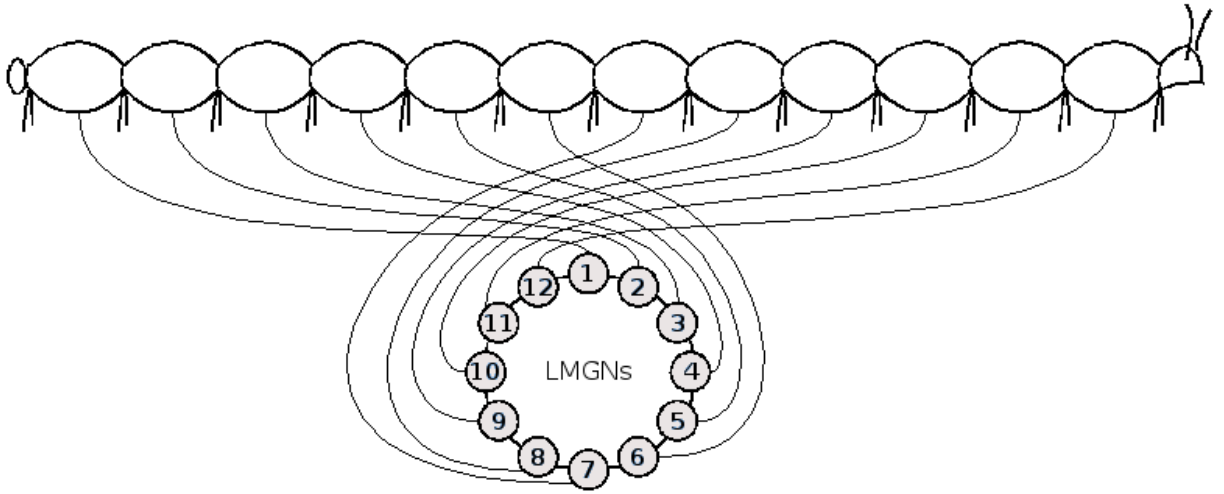
Şekil 4.4: Dairesel bağlı TY-HSA hücreleri üzerinde oluşan kendini yenileyen dalgaların zaman eksenine göre oluşturduğu spiral.



Şekil 4.5: Ayrıklaştırılmış TY-HSA ile elde edilmiş kendini yenileyen dalgaların yakından görünüşü.

4.3 Tırtıl Robotların Hareket Kontrolü

Daha önce Bölüm 3.1’de değinildiği üzere, kendini yenileyen dalgalar yalnızca hareket organları basit olan canlıların taklit edilmesinde doğrudan kullanılabilir. *Tırtıl*, bu tür canlılara güzel bir örnek oluşturur. Tırtıllar çok sayıda kısa ve düşük hareket kabiliyetine sahip bacakları olan canlılardır. Yürüyebilmek için adım atabilecek kadar uzun bacaklara sahip olmadıklarından dolayı bacakları yalnızca zemine sıkıca tutunmaya yarar. O halde tırtıllar nasıl yürür? Bunu yanıtı boğumlu olan vücutlarında saklıdır. Şekil 4.6’da örnek bir tırtıl şekli verilmiştir.

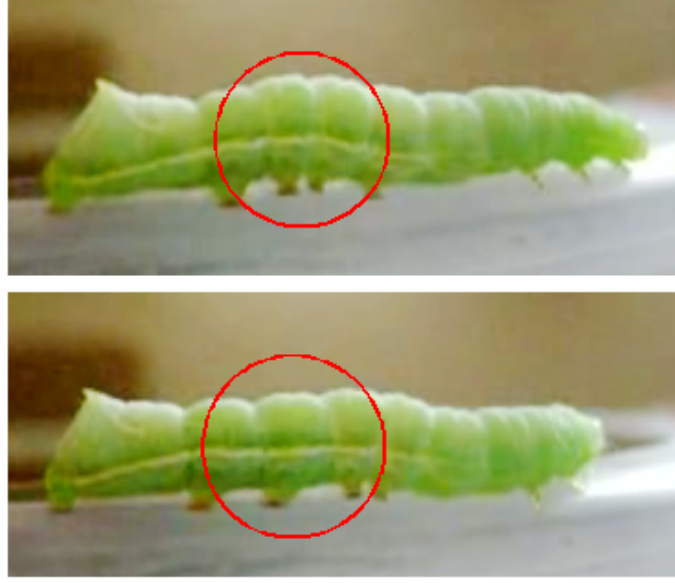


Şekil 4.6: Tırtılın hareket sistemi ile LMGNs biriminin birbiri ile olan bağlantısı.

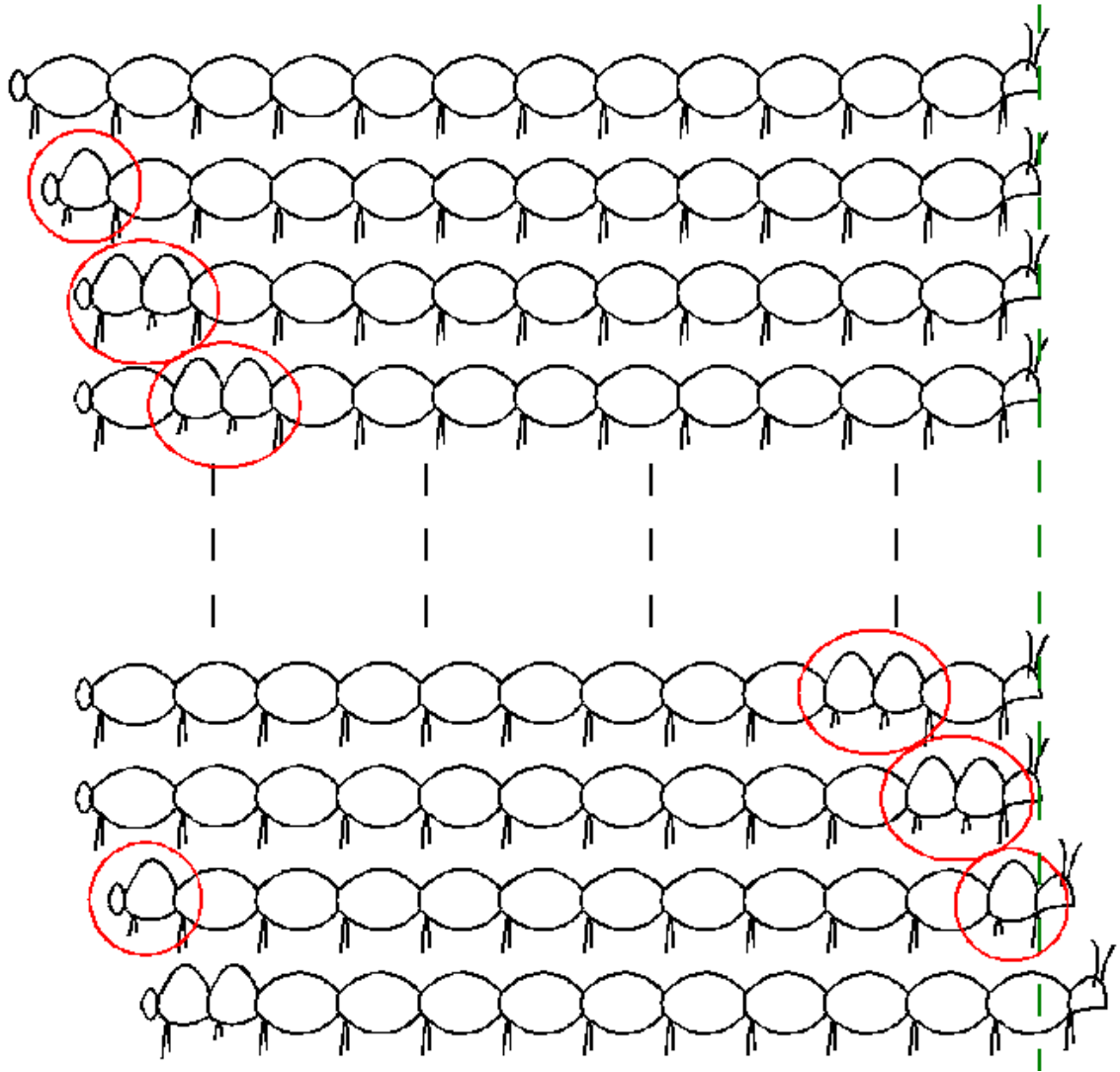
Her iki boğumun ortası bir karın olarak tanımlandığında şu varsayım yapılabilir: Kendini yenileyen dalga üreten TY-HSA hücrelerinin her birinin çıkışı tırtılın bir karnına bağlıdır, her bir karın kası kendisine bağlı olan kendini yenileyen dalganın gerilim seviyesi $-1V$ iken gevşektir ve $+1V$ iken kasılmıştır. Şekil 4.5’de daha yakından görüldüğü üzere aynı anda $+1V$ olan hücre sayısı her t anında bir veya iki tanedir. O halde kasılmış olan karınların tırtılın vücudu üzerinde dalgalanarak hareket edeceği açıktır. Aynı zamanda kasılma sonucunda ilgili karna bağlı olan bacağın da yerden kesildiği düşünülebilir.

Küçük bir animasyon ile bu hipotezin sağlaması yapılabilir. Şekil 4.6’da görünen sanal tırtıl ele alındığında, Şekil 4.8’deki animasyon kareleri elde edilir ve bu animasyonun sonucunda hipotez sağlanmış olur.

Yukarıda yapılan tanımlar gerçek bir tırtılın video kaydı izlenerek elde edilmiş gözlem sonuçlarıdır. Şekil 4.7’de de görüldüğü üzere gerçek bir tırtılın hareketinin iki ayrı fazı, yukarıda tanımlanan kasılma hareketine karşılık düşmektedir.



Şekil 4.7: Gerçek bir tırtılın video görüntüsünden alınmış iki kare.



Şekil 4.8: Sanal tırtılın yürüme fazları.

4.4 Altı Bacaklı Robotların Hareket Kontrolü

Altı bacaklı bir robotun (heksapod) kontrolü ele alınacak olursa tırtıl örneğinden çok daha karmaşık bir yapı ile karşılaşılır. Bunun nedeni hareketin basit bir kasılma olgusundan öte, üç eklemli altı adet bacağın senkron hareketi şeklinde olmasıdır. O halde öncelikle altı bacaklı robotların hareket stratejilerini incelemek gerekir.

4.4.1 Altı Bacaklı Robotların Hareket Stratejileri

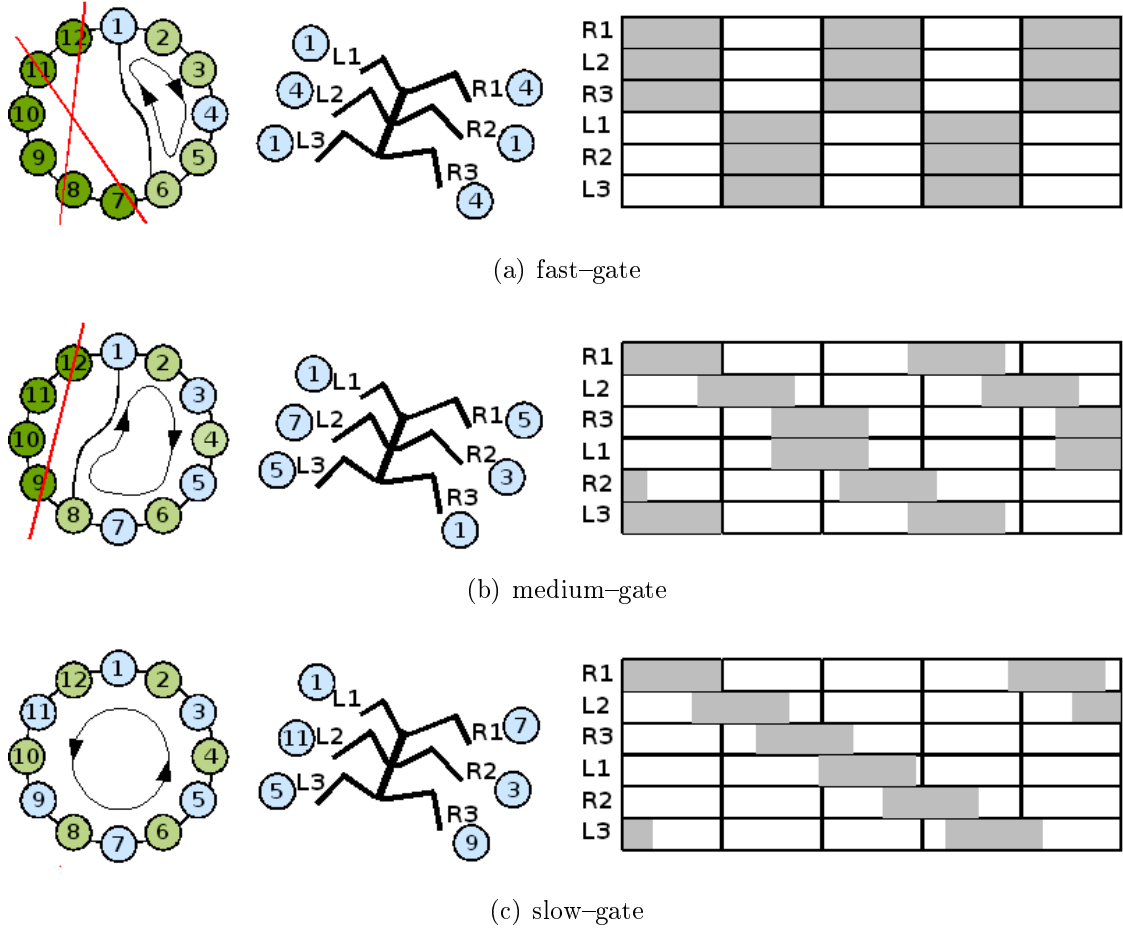
Bacakların eklem kontrolünün nasıl yapıldığını bir kenara bırakarak ve her bacağın tek bir kontrol girişi olduğunu varsayarak; bir bacağın kontrol girişine $+1V$ giriş gerilimi uygulandığında bacağın yukarı kalkarak ileriye adım attığı, $-1V$ giriş gerilimi uygulandığında ise bacağın aşağıya doğru itilerek zemini kavradığı ve gövdeyi ileriye doğru ittiği kabul edilsin. Tek bir bacağın mekanik yapısı ve kontrol devresi bu tezde yapılan çalışmanın konusu dışında kalmaktadır. Burada çözülmesi gereken, birer kontrol girişine sahip altı adet bacağın aynı gövdeye bağlanması durumunda bu bacakların senkron bir şekilde çalıştırılması ve robotun gövdesinin ileriye doğru hareket ettirilmesi problemidir. Bu bölümde Şekil 4.9’da verilmiş olan üç farklı hareket stratejisi ele alınacaktır (Arena ve Fortuna, 2002).

Herhangi bir cisim bir yüzey üzerinde dengede kalabilmek için aşağıdaki gerek koşulları sağlamak zorundadır:

- Cisim ile yüzey birbiri ile en az üç noktada kesişmelidir.
- Kesişim noktaları doğrusal olmamalıdır.
- Cismin ağırlık merkezinin yer çekimi yönünde yüzey üzerine izdüşümü, kesişim noktalarının birleştirilmesi sonucunda elde edilen kapalı alan içerisinde kalmalıdır.

Bu bilgilerin ışığında kurulabilecek en basit mantıkla: Sağ ön, sağ arka ve sol orta ayaklar kendi aralarında; sol ön, sol arka ve sağ orta bacaklar kendi aralarında birer grup oluştururlar. O halde her t anı için herhangi bir bacak havada ise o bacağın ait olmadığı gruba dahil olan bacakların tümü yüzey ile temasta olmalıdır. Tanımlanan koşullar ile oluşturulabilecek hareket stratejisi şu şekilde olacaktır: Bir bacak grubu zeminde sabit durup gövdeyi ileri doğru iterken aynı anda diğer bacak grubu ileriye doğru adım atar. Bu tür bir harekete *fast-gate* adı verilir ve böceğin koşmasına karşılık gelir (Şekil 4.9(a)).

Eğer amaç böceğin daha dengeli bir biçimde ortalama bir hızda hareket etmesi ise hareket stratejisi şu şekilde düzenlenebilir: Birbiri ile çapraz konumda bulunan bacaklar beraber



Şekil 4.9: TY-HSA ile altı bacaklı robotların hareket stratejilerinin gerçekleştirilmesi.

adım atarken orta bacaklardan biri ilk çaprazdan ikinci çapraza, diğeri ise ikinci çaprazdan birinci çapraza hareket geçişlerinde adım atarak dengeyi korumaya yardımcı olur. Çapraz bacakların ve orta bacakların yaptığı adım atma hareketi birbiri ile bir miktar kesişir. Bu hareket stratejisine *medium-gate* adı verilir (Şekil 4.9(b)).

İstenirse hareketin daha da yavaş ve dengeli olması sağlanabilir. Bu amaçla fast-gate stratejisinden esinlenerek *slow-gate* stratejisini tanımlanabilir (Şekil 4.9(c)). Üç bacağın hareketinin aynı anda olması yerine bacaklar şu sıraya göre hareket eder: Sol ön, sağ orta, sol arka, sağ ön, sol orta ve sağ arka. Ayrıca herhangi bir hareketin sonu ile bir sonraki hareketin başının bir miktar kesiştirilir. Sonuçta elde edilen hareket stratejisi robotun yavaş yürümesine karşılık gelecektir.

Sağa veya sola dönme problemi de basit bir şekilde çözülebilir. Dönmek istenilen yöndeki orta bacak durdurulur, veya diğer yöndeki ön bacağın hareketi daha erken bir fazdan başlatılırsa robot dönme hareketi de yapabilir, dolayısıyla robot iki boyutlu bir yüzeyin üzerinde dolaşabilir hale getirilmiş olur.

4.4.2 TY–HSA ile Altı Bacaklı Robotların Hareket Stratejilerinin Gerçeklenmesi

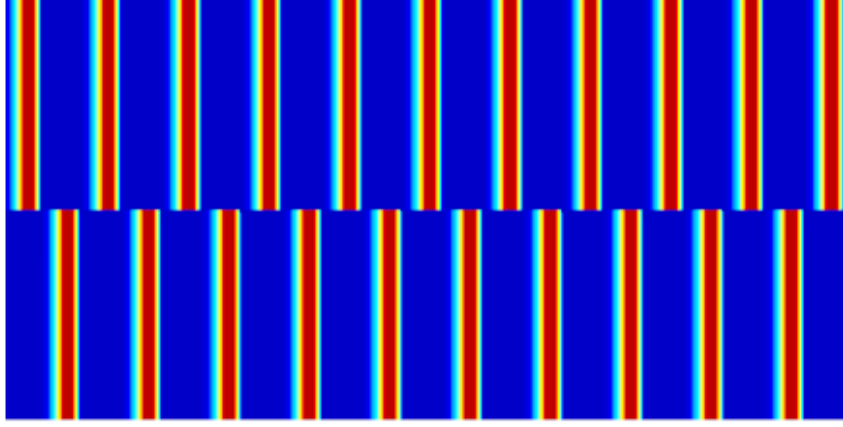
Altı bacaklı robotların kontrolünde Bölüm 4.2.3'te elde edilen kendini yenileyen dalgalar kullanılabilir (Arena ve Fortuna, 2002). Altı bacaklı robotun her bacağının tek bir kontrol girişine ihtiyaç duyduğu göz önüne alındığında, her bacağın girişinde TY–HSA devresinin 12 hücresinden birinin çıkışındaki kendini yenileyen dalga kullanılması gerektiği görünür. Şekil 4.9'da üç farklı kontrol stratejisi için kullanılabilecek üç ayrı bağlantı tipi verilmiştir. Sol sütunda bulunan şekiller TY–HSA hücrelerinin arasındaki bağlantıları; ortada bulunan şekiller altı bacaklı robotun bacak kontrol girişlerinde hangi hücrenin ürettiği kendini yenileyen dalgaların kullanılacağını; sağdaki şekiller ise gri $+1V$, beyaz $-1V$ olacak şekilde bacakların kontrol girişlerinde indüklenen gerilim seviyelerini göstermektedir. Şekil 4.10'da da yapılan simülasyonların sonucunda elde edilen kendini yenileyen dalgaların zaman diyagramları verilmiştir. Her şekil için dikey eksen yukarıdan aşağıya doğru; sol ön, sağ orta, sol arka, sağ ön, sol orta ve sağ arka bacakların kontrol girişlerinde indüklenen gerilim seviyeleridir.

Şekil 4.9(a) göz önüne alındığında, 12 uzunluklu olan TY–HSA dizisinin yarısının kullanılmadığı ve 1 numaralı hücrenin 6 numaralı hücre ile komşu yapıldığı görülmektedir. Bunun nedeni hızlı hareketi elde edebilmektir. Kendini yenileyen dalga katarının propagasyon hızının sabit olduğu düşünüldüğünde hücre sayısı azaldıkça halkanın küçüleceği, dolayısıyla kendini yenileyen dalga'nın $+1V$ olduğu noktanın daha sık tekrarlanacağı kolayca görülebilir. Bir başka deyişle TY–HSA halkası küçüldüğü zaman herhangi bir hücre üzerinde görünen kendini yenileyen dalga'nın frekansı artar, ancak çıkışın $+1V$ kalma süresi aynı kalır. Ancak kararlı bir kendini yenileyen dalga oluşturabilmek için hücre sayısının belli bir sayıdan az olmaması gerektiği unutulmamalıdır. Burada amaç koşma hareketini gerçeklemek olduğu için hücre sayısı az tutulmuştur. Bölüm 4.4.1'de belirtildiği üzere altı bacak üçerli gruplar halinde 180° 'lik faz farkı ile hareket edecektir. Bu nedenle 1 numaralı hücre ile bu hücreye en uzak hücre olan 3 numaralı hücrenin çıkışları kullanılmıştır. Şekil 4.10(a)'da koşma hareketi için elde edilen simülasyon sonucu verilmiştir.

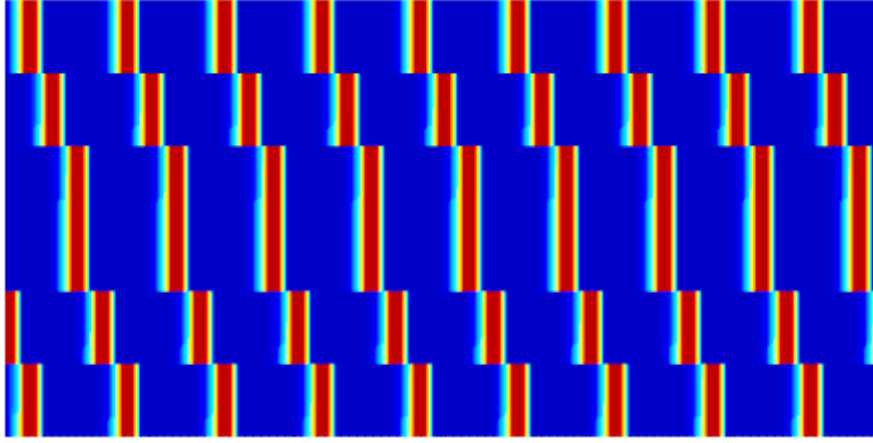
Şekil 4.9(b) göz önüne alındığında, öncelikle hücre sayısının 6'dan 8'e yükseldiği, dolayısıyla hızın azaldığı söylenebilir. Bacak bağlantıları da iki yerine dört farklı noktadan alınmıştır. Yapılan simülasyon sonuçları Şekil 4.10(b)'de verilmiştir.

Son olarak Şekil 4.9(c) incelenirse 12 hücrenin tamamının kullanıldığı görülür. Altı farklı

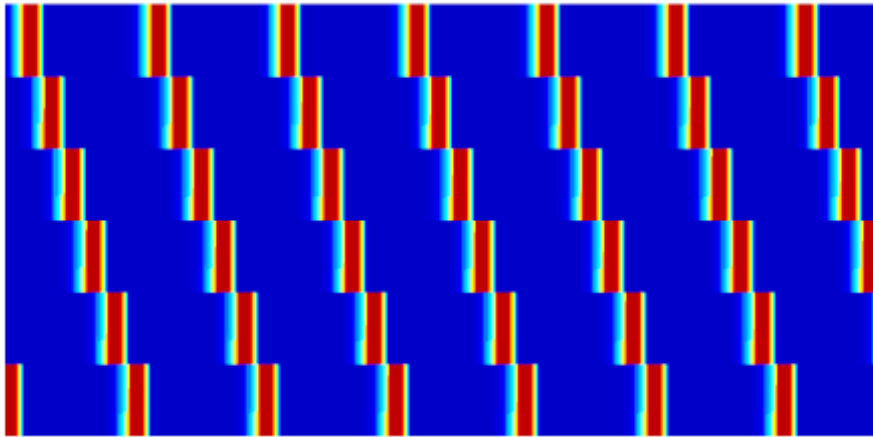
bacak için altı farklı çıkış kullanılmıştır. Şekil 4.10(c)'e bakılarak yapılan simülasyon sonuçlarının beklentileri karşıladığını söylenebilir.



(a) fast-gate



(b) medium-gate



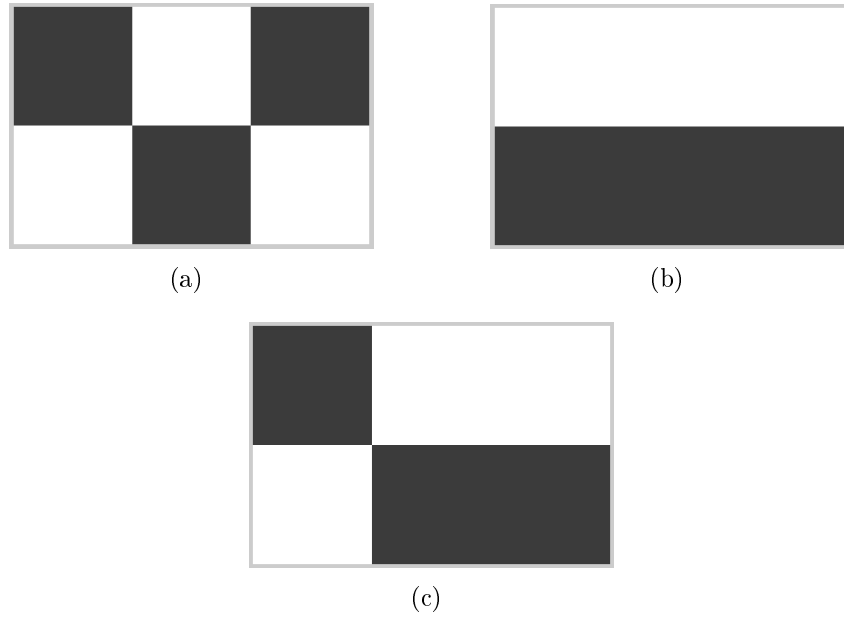
(c) slow-gate

Şekil 4.10: Altı bacaklı robotun üç ayrı hareket stratejisi için elde edilen TY-HSA simülasyonu sonuçları.

4.5 TY–HSA ile Turing Deseni Oluřturma

Bu bölümde son olarak Turing deseni oluřturma üstünde durulmuřtur. Turing deseni oluřturma, CNs biriminin yapılacak harekete karar vermesi olarak düşünölebilir. CNs biriminin çalışması bununla sınırlı kalmasa da, bu tezde üzerinde durulacak konu yalnızca Turing deseni üretimi olacaktır.

Örnek olarak 2×3 boylarında bir TY–HSA sistemi ele alımp, yüksek inhibitör ve düşük aktivatör yoğunluğu kullanıldığında TY–HSA’nın çıkışında sürekli halde Turing deseni denen kararlı bir çıkış elde edilir. Desenler arasında geçiş yapmak istendiğı zaman sınır koşullarını değıřtirmek veya TY–HSA hücrelerinin bir veya birkaç durumunu kısa bir süre için belirli ilk koşullara çekmek yoluyla yeni bir şekillenme süreci başlatılmalıdır. Yapılan simölasyonlar sonucunda Şekil 4.11’deki kararlı çıkışlar elde edilmiştir.



Şekil 4.11: TY–HSA ile elde edilmiş Turing desenleri.

Turing desenlerinin CNs olarak kullanımına örnek vermek gerekirse; Şekil 4.11(a)’nın kořma, Şekil 4.11(b)’nin ise yüzme hareketini kontrol edebildiğı söylenebilir. Yüzme hareketi de kořma ile aynı LMGNs birimi çıkışına ihtiyaç duyar. Yüzmenin kořmadan tek farkı sol bacakların beraber, sağ bacakların beraber çalışmasıdır. Bu durumda altı bacaklı robot kontrolünde kořma hareketi için söylenenler yüzme için de geçerlidir. Sensör girişlerinin CNs biriminin içindeki TY–HSA hücrelerinin ilk koşullarına etkidiğı düşünölürse suya girildiğini algılayan CNs birimi yalnızca LMGNs biriminin çıkışını modüle ederek kořma hareketini, yüzmeye çevirebilir. Aynı örneğın tersi de sudan karaya çıkan bir böce-

ğin CNs biriminin sensör girişlerinden karada olduğunu algılayarak yeni bir Turing deseni üretmesi ve bu desen ile hareketi koşturmaya çevirmesidir.

2×3 boylarında olan bir Turing deseni ile orta veya yavaş hızda ilerleme için LMGNs'i kontrol etmek mümkün değildir. Bunun nedeni her bacak için bir adet *var* veya *yok* bilgisi ile yalnızca 180° faz farkı olan kendini yenileyen dalgaları kontrol etmenin mümkün olmasıdır. O halde en genel olarak tam işlevli bir CNs elde etmek için $M \times N$ boylarında bir TY-HSA yapısına ihtiyaç duyulduğu söylenebilir. Ancak bu yaklaşımın pratik olarak bir fayda sağlamadığı görülmüş, Turing desenleri ile CNs birimi tasarlamak yerine LMGNs birimini geliştirilerek CNs'in işlevini de LMGNs'e yaptırma yolun gidilmiştir (Arena vd., 2004). Bu yöntemle göre LMGNs'in halka yapısı bozularak TY-HSA 2×3 boylarına getirilmiş, hücreler arası bağlantı simetrisi de bozularak yeni TY-HSA eşitlikleri tanımlanmıştır. Şablonları uzaya bağımlı hale gelen TY-HSA yapısı bu tezin inceleme alanının dışında kalmaktadır.

5 LİNEER HSA SÜZGEÇLERİ İLE GÖRÜNTÜ ÖN İŞLEME

Kendi kendine hareket edebilen bir robotun hareket stratejisini belirlerken çevresini algılaması gerekir. Bir robota algı girişi olarak eklenebilecek birimler aşağıdaki şekilde sıralanabilir:

- Kamera,
- ultrasonik verici-alıcı dizileri,
- kızılötesi algılayıcılar,
- yakınlık algılayıcıları,
- ışık algılayıcıları,
- sıcaklık algılayıcıları,
- hız algılayıcıları,
- ivme algılayıcıları.

Bu tezde algılayıcılarla ilgili üzerinde durulacak olan konu, kameradan alınan verilerin işlenerek kullanılabilir hale getirilmesi olacaktır. Konu başlığını daha da daraltmak gerekirse, bir robotta kullanılabilecek görüntü işleme biriminin *ön işleme* kısmının HSA ile gerçekleştirilmesi üzerinde durulacaktır.

5.1 Gauss Benzeri Alçak Geçiren Süzgeçlerin HSA ile Gerçeklenmesi

Gauss fonksiyonu birçok yerde kullanılan ve iyi bilinen bir fonksiyondur. Genel ifadesi

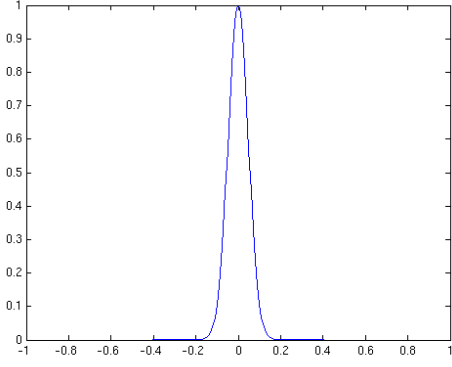
$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

şeklinde olan Gauss dağılım fonksiyonunda μ ortalama değer, σ ise standart sapma faktörü olarak tanımlıdır. Fonksiyonu basitleştirmek için $\mu = 0$ ve $2\sigma^2 = a^{-2}$ olarak alınabilir. Ayrıca ifadenin katsayısı 1 olacak şekilde genlik normalizasyonu da yapılabilir. Bu koşullar altında yukarıdaki ifade

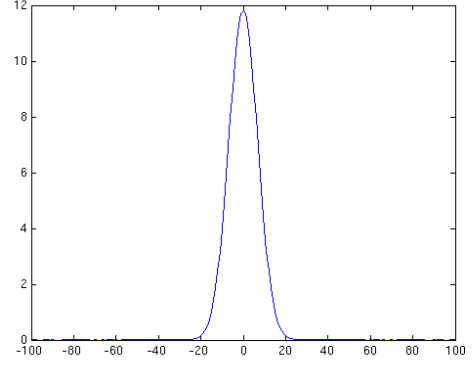
$$f(x) = e^{-(ax)^2} \tag{5.1}$$

haline indirgenmiş olur.

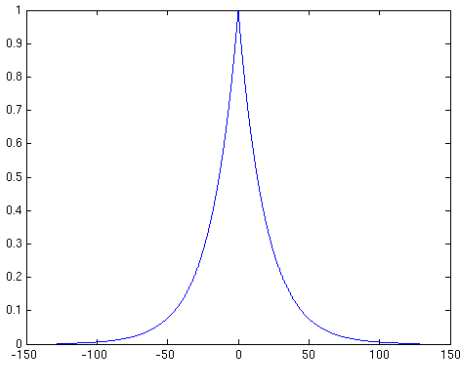
Genel şekli itibariyle Gauss fonksiyonuna benzeyen fonksiyonlara *Gauss benzeri* fonksiyonlar denir. Gauss yerine Gauss benzeri frekans yanıtına sahip süzgeçlerin tercih edilmesinin nedeni, işlevsel olarak çok farkları olmamasına rağmen, Gauss benzeri frekans yanıtına sahip süzgeçlerin HSA'nın 3×3 boylarındaki şablonlarına uygun olarak fark denkleminde elde edilebilmesidir.



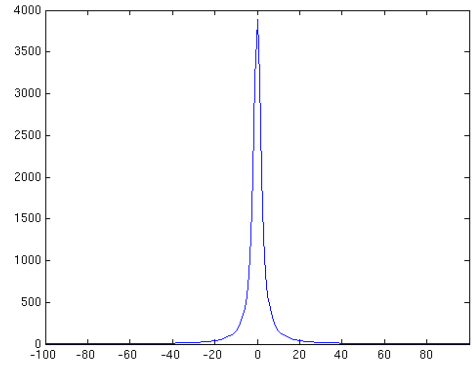
(a) Tek boyutlu Gauss fonksiyonu.



(b) Tek boyutlu Gauss fonksiyonunun Fourier dönüşümü.



(c) Tek boyutlu Gauss benzeri bir fonksiyon.



(d) Tek boyutlu Gauss benzeri fonksiyonun Fourier dönüşümü.

Şekil 5.1: Gerçek Gauss fonksiyonu ile Gauss benzeri bir fonksiyonun karşılaştırılması.

Tek boyutlu, IIR (Infinite Impulse Response) yapısında ve Gauss benzeri bir fonksiyon elde etmek amacıyla nedensel ve nedensel olmayan

$$h_1(n) = \alpha^n u(n) \quad \text{ve} \quad h_2(n) = \alpha^{-n} u(-n - 1)$$

birim örnek yanıtları kullanılabilir. Bu iki birim örnek yanıtı birleştirildiği zaman

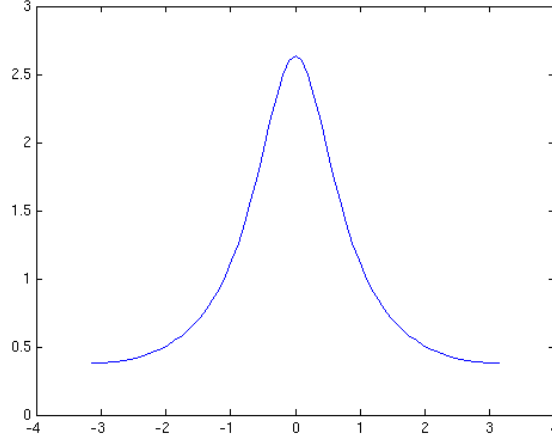
$$h(n) = h_1(n) + h_2(n) = \alpha^n u(n) + \alpha^{-n} u(-n - 1) \quad (5.2)$$

ifadesi elde edilir. (5.1) ve (5.2) ifadelerinin kendileri Şekil 5.1(a) ile Şekil 5.1(c)'de, Fourier dönüşümleri ise Şekil 5.1(b) ile Şekil 5.1(d)'de verilmiştir.

Yukarıda değinildiği üzere, HSA şablonlarına uygulanabilirlik açısından (5.2) ifadesini kullanmak daha uygun olacaktır. (5.2)'nin z -dönüşümü alındığında

$$H(z) = H_1(z) + H_2(z) = \frac{1}{1 - \alpha z^{-1}} + \frac{\alpha z}{1 - \alpha z}$$

$$H(z) = \frac{1 - \alpha^2}{-\alpha z^{-1} + 1 + \alpha^2 - \alpha z}$$



Şekil 5.2: Gauss benzeri süzgecin fark denkleminde elde edilmiş frekans yanıtı.

ve bu ifade kullanılarak da

$$H(e^{j\omega}) = \frac{1 - \alpha^2}{1 + \alpha^2 - 2\alpha \cos \omega}$$

frekans yanıtı ifadesi elde edilir (Şekil 5.2).

Bir sonraki adım olarak Gauss benzeri süzgecin fark denklemi elde edilebilir. Yukarıda elde edilen $H(z)$ ifadesi

$$H(z) = \frac{1 - \alpha^2}{-\alpha z^{-1} + 1 + \alpha^2 - \alpha z} = \frac{V(z)}{U(z)}$$

şeklinde yazılacak olursa, buradan

$$\alpha v(n-1) - (1 + \alpha^2)v(n) + \alpha v(n+1) + (1 - \alpha^2)u(n) = 0$$

fark denklemi elde edilir. Burada toplanan terimler ve bu terimlerin çarpanları tablo şeklinde yazılacak olursa

α	$-(1 + \alpha^2)$	α	0	$1 - \alpha^2$	0
$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
$y(n+1)$	$y(n)$	$y(n+1)$	$x(n-1)$	$x(n)$	$x(n+1)$
+					
$\alpha v(n-1)$	$-(1 + \alpha^2)v(n)$	$+\alpha v(n+1)$	$+0$	$+(1 - \alpha^2)u(n)$	$+0$

buradan da

$$\tilde{\mathbf{A}} = [\alpha \quad -(1 + \alpha^2) \quad \alpha]$$

$$\mathbf{B} = [0 \quad 1 - \alpha^2 \quad 0]$$

şablonları elde edilir. Fark denkleminde değişken dönüşümleri uygulanıp bazı yaklaşıklıklar yapıldığında

$$v(n-1) - (2 + \lambda^2)v(n) + v(n+1) + \lambda^2 u(n) = 0$$

denklemini elde edilir. $\tilde{\mathbf{A}}$ ve $\mathbf{B}$ şablonları da

$$\begin{aligned}\tilde{\mathbf{A}} &= \begin{bmatrix} 1 & -(2 + \lambda^2) & 1 \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} 0 & \lambda^2 & 0 \end{bmatrix}\end{aligned}$$

haline dönüşmüş olur (Shi, 1998).

Elde edilen tek boyutlu fark denklemini ve dolayısıyla şablonlar iki boyutlu hale getirildiğinde

$$\begin{aligned}v(m, n-1) + v(m-1, n) + v(m, n+1) + v(m+1, n) - (4 + \lambda^2)v(m, n) \\ + \lambda^2 u(n) = 0\end{aligned}\tag{5.3}$$

ve

$$\tilde{\mathbf{A}} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -(4 + \lambda^2) & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \lambda^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}\tag{5.4}$$

elde edilir. Buradaki λ süzgecin bant genişliği ölçütüdür. λ ile bant genişliği birbiri ile doğru orantılıdır.

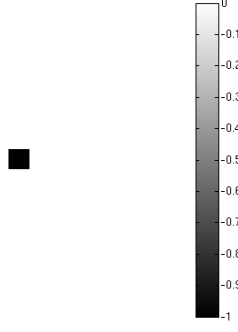
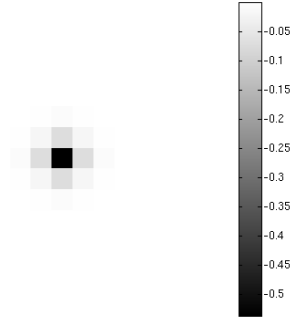
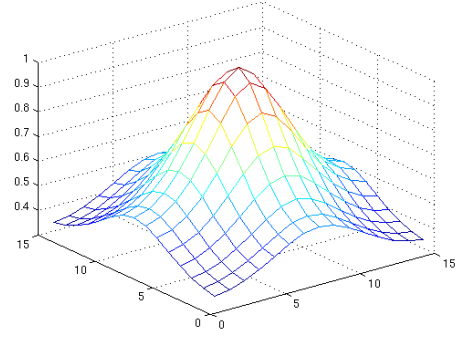
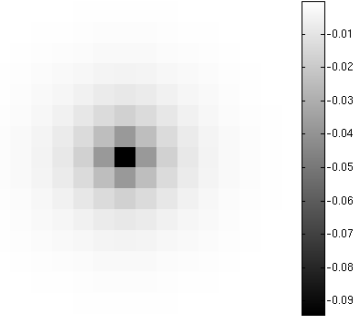
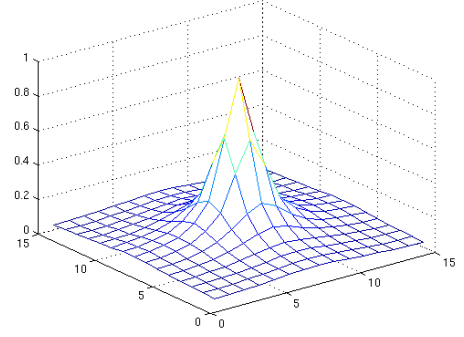
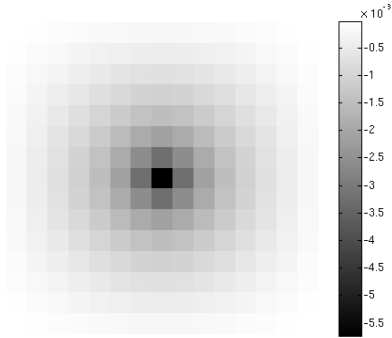
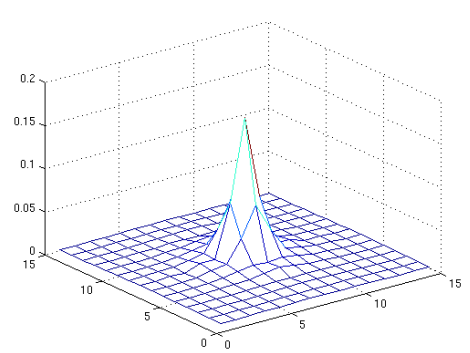
Gauss süzgecini ayırık HSA ile gerçeklemek amacıyla HSA denklemini Bölüm 4.2.2'dekine benzer bir biçimde ayrıklaştırılırsa

$$x_{ij}^{k+1} = x_{ij}^k + T_s(-x_{ij}^k + \mathbf{A} \circledast \mathbf{Y}_{ij}^k + \mathbf{B} \circledast \mathbf{U}_{ij} + I)$$

eşitliği elde edilir. Bunun yanı sıra Gauss benzeri süzgecin çıkışının, $[-1, 1]$ aralığına normalize edilmiş bir giriş verilmesi koşuluyla aynı aralıkta kalacağı bilinmektedir. HSA denklemini de bu aralıkta lineer olduğuna göre başta lineer olmayan bir denklem takımı olarak tanımlanan HSA ifadesi lineer olarak

$$x_{ij}^{k+1} = x_{ij}^k + T_s(\tilde{\mathbf{A}} \circledast \mathbf{X}_{ij}^k + \mathbf{B} \circledast \mathbf{U}_{ij})\tag{5.5}$$

şeklinde tanımlanabilir. $\mathbf{A}$ ile $\tilde{\mathbf{A}}$ şablonlarının arasında $\tilde{a}_{0,0} = -1 + a_{0,0}$ şeklinde bir ilişki vardır. Bunun nedeni (5.5) ifadesinde $y_{ij} = x_{ij}$ ilişkisinin tanımlanarak HSA'nın lineer olmasının sağlanmış olmasıdır. Yani x_{ij} durumu ile y_{ij} çıkışından gelen sinaptik ağırlıklar

(a) İki boyutlu (15×15) birim örnek resmi.(b) $\lambda = 2$ için birim örnek yanıtı.(c) $\lambda = 2$ için frekans yanıtı.(d) $\lambda = 0.5$ için birim örnek yanıtı.(e) $\lambda = 0.5$ için frekans yanıtı.(f) $\lambda = 0.1$ için birim örnek yanıtı.(g) $\lambda = 0.1$ için frekans yanıtı.Şekil 5.3: Farklı λ değerleri için Gauss benzeri HSA süzgecinin birim örnek ve frekans yanıtları.

toplanarak yeni $\tilde{\mathbf{A}}$ şablonunu oluşturur. Gauss benzeri süzgeç için elde edilen $\tilde{\mathbf{A}}$ şablonu (5.4) eşitliğinde elde edilmiştir.

Euler ileri yaklaşıklığı yardımıyla ayrıklaştırılmış HSA ifadesinin farklı λ değerleri için yapılan simülasyonları sonucunda Gauss benzeri süzgeç için elde edilen birim örnek yanıtları Şekil 5.3'ün ilk sütununda, frekans yanıtları ise aynı şeklin ikinci sütununda verilmiştir. Buradan da görülebileceği üzere Gauss benzeri süzgeçlerin frekans karakteristikleri alçak geçiren türdendir.

5.2 Gabor Benzeri Bant Geçiren Süzgeçlerin HSA ile Gerçeklenmesi

Frekans yanıtları alçak geçiren süzgeç karakteristiği gösteren Gauss benzeri süzgeçleri kullanarak Gabor benzeri bant geçiren süzgeçler elde etmek mümkündür. Alçak geçiren bir süzgecin frekans yanıtı ω -ekseninde bir yöne doğru ötelendiğinde bant geçiren bir süzgeç elde edilir. Frekansta ötelemenin ayırık uzayda $e^{-j(k\omega_{xo}+l\omega_{yo})}$ ile çarpmaya karşılık geldiği göz önüne alındığında $\tilde{\mathbf{A}}$ şablonuna

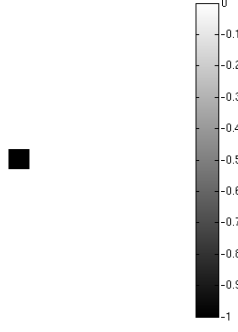
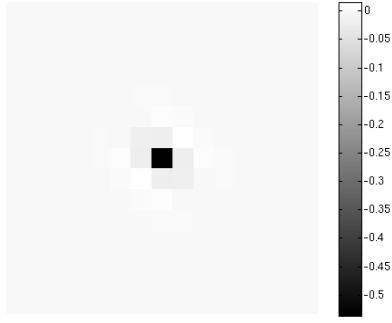
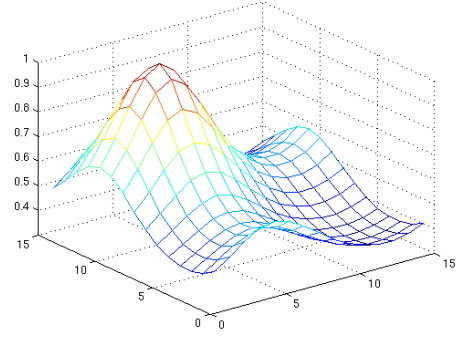
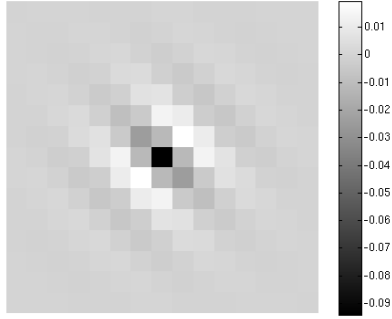
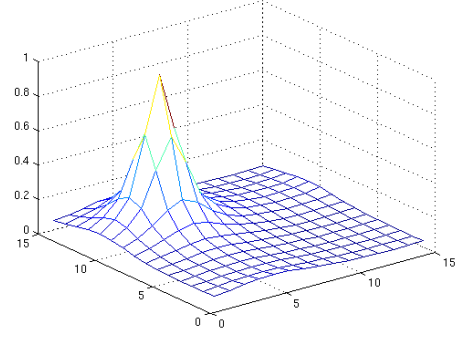
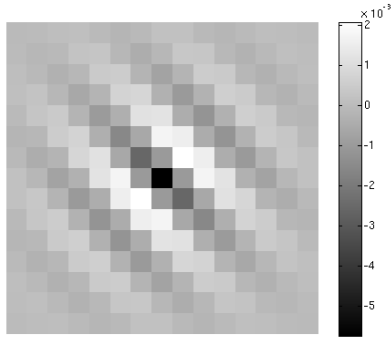
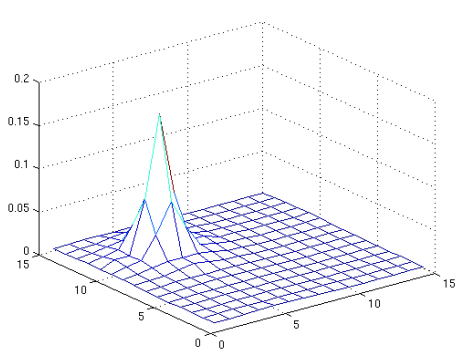
$$\tilde{\mathbf{A}} = [a_k e^{-j(k\omega_{xo}+l\omega_{yo})}]_{k,l=-r}^r; \quad r = 1$$

dönüşümü uygulanarak frekansta öteleme işlemi gerçekleştirilmiş olur (Shi, 1994). Burada frekans boyutundaki ω_{xo} ve ω_{yo} sabitleri sırasıyla ω_x ile ω_y doğrultusundaki öteleme miktarlarını, r ise şablonun komşuluk miktarını göstermektedir. Bu durumda Gabor benzeri bant geçiren süzgecin $\tilde{\mathbf{A}}$ şablonu

$$\tilde{\mathbf{A}} = \begin{bmatrix} 0 & e^{jw_{xo}} & 0 \\ e^{jw_{yo}} & -(4 + \lambda^2) & e^{-jw_{yo}} \\ 0 & e^{-jw_{xo}} & 0 \end{bmatrix}$$

halini alır. Diğer tüm değişkenler ve parametreler Gauss benzeri alçak geçiren süzgeçler için verildiği şekildedir.

Euler ileri yaklaşıklığı yardımıyla ayrıklaştırılmış HSA ifadesinin farklı λ değerleri için yapılan simülasyonları sonucunda Gabor benzeri süzgeç için elde edilen birim örnek yanıtları Şekil 5.4'ün ilk sütununda, frekans yanıtları ise aynı şeklin ikinci sütununda verilmiştir. Buradan da görülebileceği üzere Gabor benzeri süzgeçlerin frekans karakteristikleri bant geçiren türdedir.

(a) İki boyutlu (15×15) birim örnek resmi.(b) $\lambda = 2$ için birim örnek yanıtı.(c) $\lambda = 2$ için frekans yanıtı.(d) $\lambda = 0.5$ için birim örnek yanıtı.(e) $\lambda = 0.5$ için frekans yanıtı.(f) $\lambda = 0.1$ için birim örnek yanıtı.(g) $\lambda = 0.1$ için frekans yanıtı.Şekil 5.4: Farklı λ değerleri için Gabor benzeri HSA süzgecinin birim örnek ve frekans yanıtları.

5.3 Lineer HSA'ların Sayısal Çözüm Yöntemleri

Lineer olmayan kararlı bir denklem takımını sayısal olarak çözmek için öncelikle bu denklem takımını her boyutta ayrıklaştırmak, daha sonra bulmak istenen bilinmeyenleri sırasıyla sol tarafa çekerek bilinenler cinsinden ifade etmek ve yinelemeli olarak sonuca ulaşmak gerekir. Ancak denklem takımı Gauss ve Gabor benzeri süzgeçlerde olduğu gibi lineer ve asimptotik kararlı ise, literatürde yer alan lineer çözüm yöntemlerinden herhangi birini kullanarak çözüme ulaşmak da mümkündür.

Bu durumda lineer denklem takımlarının çözüm yöntemleri temel olarak iki grup altında toplanabilir:

- Tam sonuca ulaşmak için sonsuz sayıda yinelemeli işlem gerektiren yaklaşık çözüm yöntemleri,
- gerçek çözümü bulmaya yönelik, sonuca sonlu sayıda işlem ile varabilen lineer çözüm yöntemleri.

Her iki tür çözüm yöntemin kendilerine özgü olumlu ve olumsuz yanları olduğu söylenebilir. Sonsuz sayıda yinelemeli işlem gerektiren ve yakınsaklığa dayanan çözüm yöntemlerine örnek olarak, Euler ileri yaklaşıklığı ile Jacobi ve Gauss–Seidel yinelemeli çözüm yöntemleri verilebilir. Sonlu sayıda işlemle çözüme ulaşabilen çözüm yöntemlerine ise örnek olarak, genel lineer çözüm yöntemleri ile Jacobi yinelemeli çözüm yönteminin lineer çözüm yöntemlerine uyarlanması alınabilir.

Bu bölümde öncelikle sonsuz sayıda işlemle sonuca varabilen üç farklı yaklaşık çözüm yöntemine değinilecek, sonrasında ise lineer HSA denkleminin lineer bir denklem takımı şeklinde matrisler ve vektörlerle ifade edilmesi üzerinde durulacaktır. Bölüm içerisinde gerektiği yerlerde Gabor benzeri süzgeçler ele alınarak çözüm yöntemleri incelenmiştir. Bunun nedeni Gauss benzeri süzgeçlerin ötelenmemiş Gabor benzeri süzgeçler olarak tanımlanabilmeleri, dolayısıyla Gabor benzeri süzgeçlerin bir alt kümesi olmalarıdır. Bu durumda Gabor benzeri süzgeçler için söylenenlerin Gauss benzeri süzgeçler için de geçerli olduğu söylenebilir.

5.3.1 Euler İleri Yaklaşıklığı

Bölüm 4.2.2 ve Bölüm 5.1'de bu konu üzerinde durulduğu için burada kanıtlara ve ayrıntılara girilmeden ayrık lineer HSA denklemi ortaya konmuş ve hatırlatma amacıyla şablonlar

tekrar verilmiştir. Euler ileri yaklaşıklığı kullanılarak ayrıklaştırılmış Gabor benzeri HSA süzgecin ifadesi

$$x_{ij}^{k+1} = x_{ij}^k + T_s(\tilde{\mathbf{A}} \circledast \mathbf{X}_{ij}^k + \mathbf{B} \circledast \mathbf{U}_{ij}),$$

$\tilde{\mathbf{A}}$ ile $\mathbf{B}$ şablonları ise

$$\tilde{\mathbf{A}} = \begin{bmatrix} 0 & e^{jw_{xo}} & 0 \\ e^{jw_{yo}} & -(4 + \lambda^2) & e^{-jw_{yo}} \\ 0 & e^{-jw_{xo}} & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \lambda^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

şeklindedir.

5.3.2 Jacobi Yinelemeli Çözüm Yöntemi

Bu yöntemin uygulanabilir olması için sistemin sabit giriş altında asimptotik kararlı olması, yani sürekli zaman HSA denkleminde bulunan türev ifadesinin sonsuzda sıfıra düşmesi gereklidir. Bu tezde ele alınan Gabor benzeri süzgeç bu koşulu sağladığına göre sonsuzda HSA eşitliği

$$0 = \tilde{\mathbf{A}} \circledast \mathbf{X}_{ij} + \mathbf{B} \circledast \mathbf{U}_{ij}$$

haline dönüşür. Buradaki $\tilde{\mathbf{A}}$ şablonunun ortasında bulunan $x_{i,j}$ durum değişkeninin kat-sayısı şablondan dışarı çekilirse

$$0 = -(4 + \lambda^2)x_{ij} + \hat{\mathbf{A}} \circledast \mathbf{X}_{ij} + \mathbf{B} \circledast \mathbf{U}_{ij}$$

$$x_{ij}^1 = \frac{1}{4 + \lambda^2} \left(\hat{\mathbf{A}} \circledast \mathbf{X}_{ij}^0 + \mathbf{B} \circledast \mathbf{U}_{ij} \right)$$

denklemleri ile

$$\hat{\mathbf{A}} = \begin{bmatrix} 0 & e^{jw_{xo}} & 0 \\ e^{jw_{yo}} & 0 & e^{-jw_{yo}} \\ 0 & e^{-jw_{xo}} & 0 \end{bmatrix}$$

şablonu elde edilir. Bu eşitlik x_{ij}^0 'lara rastgele ilk koşullar verilerek bir kez çözülürse x_{ij}^1 'ler elde edilir. Elde edilen x_{ij}^1 'ler x_{ij}^0 'ların yerine konarak aynı eşitlik ikinci bir kez çözülürse x_{ij}^2 'ler elde edilir. Bu işlem J kez tekrarlandığında sonuca ulaşılmış olur. Döngüyü durdurma koşulu olarak x_{ij}^J ile x_{ij}^{J-1} arasındaki bağıl hata göz önüne alınabilir. Eğer bağıl hata sıfıra yaklaşmış ise asimptota yaklaşmış ve dolayısıyla çözüm belli bir hata payı ile elde edilmiştir.

5.3.3 Gauss–Seidel Yinelemeli Çözüm Yöntemi

Jacobi yinelemeli çözüm yöntemine çok benzeyen bu yöntemin farkı, yeni elde edilen x_{ij}^J 'lerin resmin taranmasının beklenmeden x_{ij}^{J-1} 'ler üzerine yazılmasıdır. Yeni değerlerin elde edildikleri anda eski değerin üzerine yazılması nedeniyle Gauss-Seidel yaklaşıklığı ile çözüme ulaşma hızı, piksellerin taranma sırasına bağlıdır. Buradaki HSA eşitliği

$$x_{ij} = \frac{1}{4 + \lambda^2} \left(\hat{\mathbf{A}} \circledast \mathbf{X}_{ij} + \mathbf{B} \circledast \mathbf{U}_{ij} \right)$$

şeklinde dir.

5.3.4 Lineer Denklem Takımına Jacobi Yaklaşıklığından Yararlanarak Geçiş

Bölüm 5.3.2'deki işlemlere devam edilerek HSA'nın lineer bir denklem takımı olarak ifade edilmesi sağlanabilir ve bu sayede sonsuz Jacobi döngüsü sonucunda yakınsanması beklenen son değer sonlu sayıda işlem ile hesaplanabilir (Tavsanoglu, 2006). $-1/(4 + \lambda^2) = a_{00}^{-1}$ olduğu da göz önüne alınarak Jacobi eşitliği $\mathcal{A}$ durum matrisi ile

$$\mathbf{x}^1 = \mathcal{A}\mathbf{x}^0 + \mathbf{b} ; \quad \mathcal{A} = a_{00}^{-1} \hat{\mathbf{A}} ; \quad \mathbf{b} = a_{00}^{-1} \mathbf{B} \mathbf{u}$$

şeklinde ifade edilebilir. $\mathbf{u}$ vektörü, işlenecek resmin $M \times N$ boylarında olması durumunda, 1'den M 'e kadar olan N adet satırın arka arkaya eklenmesi sonucunda oluşmuş bir vektördür. Dolayısıyla $\mathbf{b}$ de aynı boyda bir vektör olacaktır. Ayrıca toplama işleminin geçerli olabilmesi açısından $\mathbf{x}$ vektörünün de benzer şekilde paketlenmiş olma koşulu vardır. Örnek olarak 3×4 boylarında bir resim alındığında Gabor için

$$\mathcal{A} = \begin{bmatrix} 0 & \frac{-e^{-j\omega_{yo}}}{4+\lambda^2} & 0 & 0 & \frac{-e^{-j\omega_{xo}}}{4+\lambda^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-e^{j\omega_{yo}}}{4+\lambda^2} & 0 & \frac{-e^{-j\omega_{yo}}}{4+\lambda^2} & 0 & 0 & \frac{-e^{-j\omega_{xo}}}{4+\lambda^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-e^{j\omega_{yo}}}{4+\lambda^2} & 0 & \frac{-e^{-j\omega_{yo}}}{4+\lambda^2} & 0 & 0 & \frac{-e^{-j\omega_{xo}}}{4+\lambda^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-e^{j\omega_{yo}}}{4+\lambda^2} & 0 & 0 & 0 & 0 & \frac{-e^{-j\omega_{xo}}}{4+\lambda^2} & 0 & 0 & 0 & 0 \\ \frac{-e^{j\omega_{xo}}}{4+\lambda^2} & 0 & 0 & 0 & 0 & \frac{-e^{-j\omega_{yo}}}{4+\lambda^2} & 0 & 0 & \frac{-e^{-j\omega_{xo}}}{4+\lambda^2} & 0 & 0 & 0 \\ 0 & \frac{-e^{j\omega_{xo}}}{4+\lambda^2} & 0 & 0 & \frac{-e^{j\omega_{yo}}}{4+\lambda^2} & 0 & \frac{-e^{-j\omega_{yo}}}{4+\lambda^2} & 0 & \frac{-e^{-j\omega_{xo}}}{4+\lambda^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{-e^{j\omega_{xo}}}{4+\lambda^2} & 0 & 0 & \frac{-e^{j\omega_{yo}}}{4+\lambda^2} & 0 & \frac{-e^{-j\omega_{yo}}}{4+\lambda^2} & 0 & \frac{-e^{-j\omega_{xo}}}{4+\lambda^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{-e^{j\omega_{xo}}}{4+\lambda^2} & 0 & 0 & \frac{-e^{j\omega_{yo}}}{4+\lambda^2} & 0 & 0 & 0 & \frac{-e^{-j\omega_{xo}}}{4+\lambda^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{-e^{j\omega_{xo}}}{4+\lambda^2} & 0 & 0 & 0 & \frac{-e^{-j\omega_{yo}}}{4+\lambda^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-e^{j\omega_{xo}}}{4+\lambda^2} & 0 & 0 & \frac{-e^{j\omega_{yo}}}{4+\lambda^2} & 0 & \frac{-e^{-j\omega_{xo}}}{4+\lambda^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-e^{j\omega_{xo}}}{4+\lambda^2} & 0 & \frac{-e^{j\omega_{yo}}}{4+\lambda^2} & 0 & \frac{-e^{-j\omega_{yo}}}{4+\lambda^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-e^{j\omega_{xo}}}{4+\lambda^2} & 0 & 0 & \frac{-e^{j\omega_{yo}}}{4+\lambda^2} & 0 \end{bmatrix}$$

durum matrisi elde edilir. $\mathcal{A}$ durum matrisi; diyagonal elemanları sıfır, diğer elemanları ise ilk tanımlanan $\tilde{\mathbf{A}}$ şablonunun orta elemanına bölünmüş olan seyrek bir matristir. Ayrıca $\mathcal{A}$

durum matrisi $(M \cdot N) \times (M \cdot N)$ boylarında, $\mathbf{u}$ ve $\mathbf{x}$ vektörleri ise $(M \cdot N) \times 1$ boyundadır. Durum matrisinin dört yanal diyagonalı haricindeki bütün yanal diyagonalleri ve ana diyagonalı sıfırdır.

Frekans boyutundaki öteleme miktarı sıfır olduğunda $e^{\pm j\omega_{xo}}$ ve $e^{\pm j\omega_{yo}}$ üstel ifadelerinin yerine 1 gelir ve böylece Gauss benzeri ayrık HSA süzgecinin durum matrisi

$$\mathcal{A} = \begin{bmatrix} 0 & \frac{1}{4+\lambda^2} & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4+\lambda^2} & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4+\lambda^2} & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & 0 & 0 \\ \frac{1}{4+\lambda^2} & 0 & 0 & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & 0 \\ 0 & \frac{1}{4+\lambda^2} & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 \\ 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & \frac{1}{4+\lambda^2} & 0 \\ 0 & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & 0 & 0 & \frac{1}{4+\lambda^2} \\ 0 & 0 & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & \frac{1}{4+\lambda^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & \frac{1}{4+\lambda^2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4+\lambda^2} & 0 & 0 & \frac{1}{4+\lambda^2} & 0 \end{bmatrix}$$

olarak elde edilir.

Jacobi yinelemeli çözüm yöntemindeki her bir döngü sonucunda elde edilecek olan durum vektörleri sırasıyla

$$\begin{aligned} \mathbf{x}^1 &= \mathcal{A}\mathbf{x}^0 + \mathbf{b} \\ \mathbf{x}^2 &= \mathcal{A}\mathbf{x}^1 + \mathbf{b} = \mathcal{A}(\mathcal{A}\mathbf{x}^0 + \mathbf{b}) + \mathbf{b} = \mathcal{A}^2\mathbf{x}^0 + (\mathcal{A} + \mathbf{I})\mathbf{b} \\ \mathbf{x}^3 &= \mathcal{A}^3\mathbf{x}^0 + (\mathcal{A}^2 + \mathcal{A} + \mathbf{I})\mathbf{b} \\ &\vdots \\ \mathbf{x}^J &= \mathcal{A}^J\mathbf{x}^0 + (\mathcal{A}^{J-1} + \mathcal{A}^{J-2} + \dots + \mathbf{I})\mathbf{b} \end{aligned}$$

olacaktır. Bu denklemlerden yararlanarak, sistemin kararlı olabilmesi için ilk koşullardan gelen $\mathcal{A}^J\mathbf{x}^0$ öz çözüm teriminin yeterince büyük bir J değeri için sıfıra gitmesi gerektiği sonucuna varılır. Bu durumda kararlılık kriteri olarak ayrık sistemin $\mathcal{A}$ durum matrisinin z -düzleminde birim çemberin içinde kalması koşulu verilebilir. İfadenin kalanının sonsuzda yakınsayacağı değer seri açılımından faydalananak bulunabilir.

$$\mathbf{S} = \mathcal{A}^{J-1} + \mathcal{A}^{J-2} + \dots + \mathbf{I} \quad (5.6)$$

eşitliğinin her iki tarafı $\mathcal{A}$ ile çarpıldığında

$$\mathcal{A}\mathbf{S} = \mathcal{A}^J + \mathcal{A}^{J-1} + \dots + \mathcal{A} \quad (5.7)$$

elde edilir. (5.6)'dan (5.7) çıkartıldığında

$$\mathbf{S} - \mathcal{A}\mathbf{S} = \mathbf{I} - \mathcal{A}^J$$

ve buradan da

$$\mathbf{S} = (\mathbf{I} - \mathcal{A})^{-1}(\mathbf{I} - \mathcal{A}^J)$$

elde edilir. Yeterince büyük bir J değeri için

$$\mathcal{A}^J \rightarrow 0$$

olduğuna göre yukarıdaki ifade

$$\mathbf{S} = (\mathbf{I} - \mathcal{A})^{-1}(\mathbf{I} - \mathcal{A}^J) \cong (\mathbf{I} - \mathcal{A})^{-1}$$

haline indirgenebilir. O halde $\mathbf{x}$ 'in son değerinin

$$\mathbf{x}^J = (\mathbf{I} - \mathcal{A})^{-1}\mathbf{b} \tag{5.8}$$

olacağı söylenebilir. Bu eşitliğin de

$$(\mathbf{I} - \mathcal{A})\mathbf{x} = \mathbf{b} \tag{5.9}$$

lineer denklem takımının çözümü olduğu göz önüne alındığında; HSA'nın sabit giriş altında ve sürekli halde, türev ifadesinin sonsuzda sifıra yakınsama koşulundan yararlanılarak, basit bir lineer denklem takımına dönüştürülebileceği kanıtlanmış olur (Tavsanoglu, 2006).

5.3.5 Lineer Denklem Takımlarına Doğrudan Geçiş

Sürekli halde türev ifadesinin asimptotik kararlılıktan dolayı sifıra yakınsadığı göz önüne alındığında şablon formunda ifade edilmiş olan

$$\frac{dx_{ij}}{dt} = \tilde{\mathbf{A}} \circledast \mathbf{X}_{ij} + \mathbf{B} \circledast \mathbf{U}_{ij}$$

lineer HSA eşitliği doğrudan

$$0 = \tilde{\mathbf{A}} \circledast \mathbf{X}_{ij} + \mathbf{B} \circledast \mathbf{U}_{ij}$$

şeklinde yazılabilir. Aynı eşitlik matris formunda ifade edildiğinde

$$0 = \mathbf{Ax} + \mathbf{Bu}$$

eşitliği elde edilir. Bu eşitliğin işlem süresince sabit olan giriş ifadesi $-\mathbf{B}\mathbf{u} = \mathbf{b}$ olarak tanımlanırsa

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

lineer denklem takımı eşitliğine ulaşılır. Buradaki $\mathbf{x}$ ve $\mathbf{b}$ vektörleri Bölüm 5.3.4'de bahsi geçtiği şekilde paketlenmiştir. 3×4 boylarındaki bir resim için elde edilecek olan $\mathbf{A}$ matrisi ise

$$\mathbf{A} = \begin{bmatrix} -(4+\lambda^2) & e^{-j\omega_{yo}} & 0 & 0 & e^{-j\omega_{xo}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ e^{j\omega_{yo}} & -(4+\lambda^2) & e^{-j\omega_{yo}} & 0 & 0 & e^{-j\omega_{xo}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{j\omega_{yo}} & -(4+\lambda^2) & e^{-j\omega_{yo}} & 0 & 0 & e^{-j\omega_{xo}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{j\omega_{yo}} & -(4+\lambda^2) & 0 & 0 & 0 & e^{-j\omega_{xo}} & 0 & 0 & 0 & 0 \\ e^{j\omega_{xo}} & 0 & 0 & 0 & -(4+\lambda^2) & e^{-j\omega_{yo}} & 0 & 0 & e^{-j\omega_{xo}} & 0 & 0 & 0 \\ 0 & e^{j\omega_{xo}} & 0 & 0 & e^{j\omega_{yo}} & -(4+\lambda^2) & e^{-j\omega_{yo}} & 0 & 0 & e^{-j\omega_{xo}} & 0 & 0 \\ 0 & 0 & e^{j\omega_{xo}} & 0 & 0 & e^{j\omega_{yo}} & -(4+\lambda^2) & e^{-j\omega_{yo}} & 0 & 0 & e^{-j\omega_{xo}} & 0 \\ 0 & 0 & 0 & e^{j\omega_{xo}} & 0 & 0 & e^{j\omega_{yo}} & -(4+\lambda^2) & 0 & 0 & 0 & e^{-j\omega_{xo}} \\ 0 & 0 & 0 & 0 & e^{j\omega_{xo}} & 0 & 0 & 0 & -(4+\lambda^2) & e^{-j\omega_{yo}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{j\omega_{xo}} & 0 & 0 & e^{j\omega_{yo}} & -(4+\lambda^2) & e^{-j\omega_{yo}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{j\omega_{xo}} & 0 & 0 & e^{j\omega_{yo}} & -(4+\lambda^2) & e^{-j\omega_{yo}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{j\omega_{xo}} & 0 & 0 & e^{j\omega_{yo}} & -(4+\lambda^2) \end{bmatrix}$$

şeklinde olacaktır. Yine seyrek olan bu durum matrisinin ana diyagonalı ile dört yanıl diyagonalı hariç tüm diyagonalleri sıfırdır. Frekans boyutundaki öteleme miktarı sıfır olduğunda $e^{\pm j\omega_{xo}}$ ve $e^{\pm j\omega_{yo}}$ üstel ifadelerinin yerine 1 gelir ve böylece Gauss benzeri HSA süzgecinin durum matrisi

$$\mathbf{A} = \begin{bmatrix} -(4+\lambda^2) & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -(4+\lambda^2) & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -(4+\lambda^2) & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -(4+\lambda^2) & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -(4+\lambda^2) & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & -(4+\lambda^2) & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & -(4+\lambda^2) & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & -(4+\lambda^2) & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -(4+\lambda^2) & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -(4+\lambda^2) & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -(4+\lambda^2) & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -(4+\lambda^2) \end{bmatrix}$$

olarak elde edilir.

5.4 Sonlu Sayıda İşlemle $\mathbf{Ax}=\mathbf{b}$ Ayrık Lineer HSA Denkleminin Çözümü

$\mathbf{A}$ matrisi ve $\mathbf{b}$ vektörünün bilinmesi koşuluyla lineer bir denklem takımı olan $\mathbf{Ax} = \mathbf{b}$ eşitliğindeki $\mathbf{x}$ vektörünü çözmek için birçok genel çözüm yöntemi geliştirilmiştir. Ancak

HSA denklemindeki $\mathbf{A}$ matrisi çok özel bir yapıya sahiptir ve bu yapıya özel bazı yöntemler geliştirmek mümkündür. Bu bölümde öncelikle genel çözüm yöntemlerine değinilecek, daha sonra $\mathbf{A}$ 'nın özel yapıda olması koşulu ile yapılabilecek indirgemeler üzerinde durulacaktır.

5.4.1 Matrisin Tersini Alarak Çözüm Yöntemi

Bu yöntem matematiksel olarak çok basit bir şekilde ifade edilebilen bir çözüm yöntemidir. Bölüm 5.3.5 ve Bölüm 5.3.4'de elde edilen lineer denklem takımlarının çözümü için kullanılabilir. Çözmek istenen lineer HSA denklemi

$$\mathbf{Ax} = \mathbf{b}$$

olduğuna göre basit birkaç düzenleme sonucu

$$\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{A}^{-1}\mathbf{b} ; \quad \mathbf{A}^{-1}\mathbf{A} \triangleq \mathbf{I}$$

$$\mathbf{Ix} = \mathbf{A}^{-1}\mathbf{b}$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{5.10}$$

kolaylıkla elde edilir.

Teorik matematikçiler açısından çok masum görünen bu yöntem bilgisayar programcıları için büyük sorun teşkil eder. Bunun nedeni matrisin boyunun büyüdükçe tersinin alınması için yapılması gereken işlem sayısının çok artmasıdır. Durum matrisi $p \times p$ boyunda ise (5.10) için yapılması gereken çarpım sayısı $p^3 + p^2$, toplam sayısı ise $p^3 - p^2$ kadardır. Örnek vermek gerekirse 32×32 boylarındaki bir resim için yaklaşık 10^9 'ar tane çarpım ve toplam işlemi yapmak gerekir.

5.4.2 Gauss–Jordan Eleme Yöntemi

Eleme yöntemleri bilgisayar programcılarının en çok başvurduğu lineer denklem çözüm yöntemlerinden biridir. Eleme yöntemlerinden en iyi bilinenlerinden ikisi Gauss–Jordan ve Gauss eleme yöntemleridir. Her iki yöntem de ayrık lineer HSA'nın lineer denklem takımlarına doğrudan uygulanabilir. Eleme yöntemleri temel olarak sistemin

$$\mathbf{x} = [\mathbf{A}|\mathbf{b}]$$

şeklde çözümü olarak tanımlanabilir. $[\mathbf{A}|\mathbf{b}]$ matrisini düzenlemek için bu matrise sol tarafta $\mathbf{I}$ birim matrisi kalana kadar satır işlemleri uygulanması durumunda sağ tarafta $\mathbf{x}$ elde edilir.

$$[\mathbf{A}|\mathbf{b}] \xrightarrow{\text{Gauss-Jordan Eleme Yöntemi}} [\mathbf{I}|\mathbf{x}]$$

Gauss-Jordan eleme yönteminin adımları, tekil olmayan ve indirgenemeyen $p \times p$ boyundaki bir $\mathbf{A}$ matrisi için aşağıdaki gibi tanımlanabilir:

1. $i = 1, 2, \dots, p$ olmak üzere $[\mathbf{A}|\mathbf{b}]$ matrisinin içindeki $\mathbf{A}$ alt matrisinin her bir a_{ii} diyagonal elemanına *eksen elemanı* adı verilir.
2. i 'inci satır o satırın $\mathbf{A}_{ii}$ eksen elemanına bölünür.
3. $k = 1, 2, \dots, p$; $k \neq i$ olmak üzere i . satır k . satırın a_{ki} elemanı ile çarpılarak k . satırdan çıkartılır.
4. İşleme sol üst köşedeki a_{11} eksen elemanından başlanır ve bütün i 'ler için yukarıdaki işlemler tekrarlanır.

Örnek olarak 3×4 boylarındaki bir resim Gauss-Jordan eleme yöntemi ile işlenmek istendiğinde, bu boylardaki bir resmi işlemek için 12×12 boylarında bir $\mathbf{A}$ durum matrisi oluşacağı söylenebilir. Şekil 5.5'te Gauss-Jordan eleme yöntemi ile $[\mathbf{A}|\mathbf{b}]$ sisteminin $[\mathbf{I}|\mathbf{x}]$ haline getirilerek çözülmesi gösterilmiştir. Buradaki a ve b 'lerin üst indisleri, ilgili matris elemanının kaç kere değiştiği ile ilgili bilgi vermek amacıyla yazılmıştır. Şekilden de görülebileceği üzere sol üst köşeden başlanarak diyagonal elemanının değerini 1 yapacak satır işlemi uygulandıktan sonra diyagonalin bulunduğu sütun elemanlarını sıfırlayacak satır işlemleriyle çözüme devam edilir.

Çözüme ulaşmak için yapılması gereken çarpım sayısı $\frac{p^3}{2} + \frac{p^2}{2}$, toplam sayısı ise $\frac{p^3}{2} - \frac{p}{2}$ olacaktır. Resim olabilecek kadar büyük olan $M \times N$ boylarındaki bir $\mathbf{u}$ girişi için $p = M \cdot N$ değeri çok büyük olduğundan dolayı p^3 'lü ifadenin yanında p^2 'li ifadenin önemi kalmaz, ihmal edilebilir. Bu durumda çarpım ve toplam işlemlerinin yaklaşık olarak eşit sayıda olacağı söylenebilir. Gauss-Jordan eleme yönteminin matrisin tersini alarak girişle çarpmaya göre yaklaşık iki kat daha hızlı olduğu, ayrıca ters alma işleminin hem durum matrisinin kendisi, hem de tersi için bellek gereksinimine ihtiyaç duyduğu göz önüne alındığında iki kat daha az bellek alanına ihtiyaç duyduğu da söylenebilir. 32×32 boylarındaki bir resim için yaklaşık 5×10^8 'er tane çarpım ve toplam işlemi yapmak gerekir.

5.4.3 Gauss Eleme Yöntemi

Gauss-Jordan eleme yöntemi için yapılan birçok tanım Gauss eleme yöntemi için de geçerlidir. Bu yöntem de Gauss-Jordan eleme yöntemi gibi durum matrisinin diyagonal

$$\begin{array}{ccc}
\left[\begin{array}{c} a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \end{array} \right] & \rightarrow & \left[\begin{array}{c} 1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \end{array} \right] & \rightarrow & \left[\begin{array}{c} 1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \end{array} \right] \dots
\end{array}$$

$$\begin{array}{ccc}
\left[\begin{array}{c} 1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \end{array} \right] & \rightarrow & \left[\begin{array}{c} 1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 1 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \end{array} \right] & \rightarrow & \left[\begin{array}{c} 1 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 1 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \end{array} \right] \dots
\end{array}$$

$$\begin{array}{ccc}
\left[\begin{array}{c} 1 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 1 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \end{array} \right] & \rightarrow & \left[\begin{array}{c} 1 0 0 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 b^3 \\ 0 1 0 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 b^3 \\ 0 0 1 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 b^3 \\ 0 0 0 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 b^3 \\ 0 0 0 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 b^3 \\ 0 0 0 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 b^3 \\ 0 0 0 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 b^3 \\ 0 0 0 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 b^3 \\ 0 0 0 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 b^3 \\ 0 0 0 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 b^3 \\ 0 0 0 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 b^3 \\ 0 0 0 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 b^3 \end{array} \right] & \rightarrow & \left[\begin{array}{c} 1 0 0 0 0 0 0 0 0 0 0 0 x \\ 0 1 0 0 0 0 0 0 0 0 0 0 x \\ 0 0 1 0 0 0 0 0 0 0 0 0 x \\ 0 0 0 1 0 0 0 0 0 0 0 0 x \\ 0 0 0 0 1 0 0 0 0 0 0 0 x \\ 0 0 0 0 0 1 0 0 0 0 0 0 x \\ 0 0 0 0 0 0 1 0 0 0 0 0 x \\ 0 0 0 0 0 0 0 1 0 0 0 0 x \\ 0 0 0 0 0 0 0 0 1 0 0 0 x \\ 0 0 0 0 0 0 0 0 0 1 0 0 x \\ 0 0 0 0 0 0 0 0 0 0 1 0 x \\ 0 0 0 0 0 0 0 0 0 0 0 1 x \end{array} \right]
\end{array}$$

Şekil 5.5: 3×4 boylarındaki bir resmin Gauss–Jordan eleme yöntemi ile işlenmesi.

elemanları sıfır olmayan lineer denklem takımları için uygun bir çözüm yöntemidir, dolayısıyla Bölüm 5.3.5'te elde edilen lineer HSA denklemine uygulanabilir. Çözülmüş sistem Gauss-Jordan yöntemindeki gibi $[\mathbf{A}|\mathbf{b}]$ şeklinde tanımlıdır. Çözüm yöntemi sol tarafta $\mathbf{I}$ birim matrisi kalana kadar gerekli satır işlemlerinin uygulanması şeklinde aşağıdaki gibi tanımlanabilir:

$$[\mathbf{A}|\mathbf{b}] \xrightarrow{\text{Gauss Eleme Yöntemi}} [\mathbf{I}|\mathbf{x}]$$

Gauss eleme yöntemi Gauss–Jordan eleme yönteminden farklı olarak iki fazdan oluşur. Birinci fazda Gauss–Jordan için tanımlanan adımlar yalnızca diyagonalin altında kalan üçgeni sıfırlamak için kullanılır. İkinci fazın başında son satırdaki en sağ matris elemanının $\mathbf{x}_p$ durum değişkeninin değerini gösterdiği görülür. Bilinen durum değişkenlerinin değerleri yukarıya doğru bilinmeyenlerin yerine konduğunda sırasıyla $\mathbf{x}_p - 1, \mathbf{x}_p - 2, \dots, \mathbf{x}_1$ elde edilir.

Tekil olmayan ve indirgenemeyen $p \times p$ boylarındaki bir $\mathbf{A}$ matrisi için Gauss eleme yön-

teminin adımları aşağıdaki şekilde tanımlanabilir:

- $i = 1, 2, \dots, p$ olmak üzere $[\mathbf{A}|\mathbf{b}]$ matrisinin içindeki $\mathbf{A}$ alt matrisinin her bir a_{ii} diyagonal elemanına *eksen elemanı* adı verilir.
- Faz -1-
 1. i 'inci satır o satırın a_{ii} eksen elemanına bölünür.
 2. $k = i + 1, i + 2, \dots, p$ olmak üzere i 'inci satır, kendisinden daha aşağıda bulunması koşulu olan k . satırın a_{ki} elemanı ile çarpılarak k . satırdan çıkartılır.
 3. İşleme sol üst köşedeki $\mathbf{A}_{11}$ eksen elemanından başlanır ve bütün i 'ler için işlemler tekrarlanır.
- Faz -2-
 1. Birinci fazın sonunda matrisin p . satırındaki son elemanın değeri çözülmüştür, $[\mathbf{A}|\mathbf{b}]_{p,p+1} = x_p$ 'dir.
 2. $i = p - 1, p - 2, \dots, p$ ve $k = i - 1, i - 2, \dots, 1$ olmak üzere i . satır k . satırın a_{ki} elemanı ile çarpılarak k . satırdan çıkartılır.
 3. İşleme sağ alt köşedeki $\mathbf{A}_{pp}$ eksen elemanından başlanır ve bütün i 'ler için işlemler tekrarlanır.

Örnek olarak yine 3×4 boylarındaki bir resim alınarak Gauss eleme yöntemi ile işlenmesi problemi ele alınabilir. Şekil 5.6'da Gauss eleme yöntemi ile $[\mathbf{A}|\mathbf{b}]$ sisteminin $[\mathbf{I}|\mathbf{x}]$ haline getirilerek çözülmesi gösterilmiştir. Buradaki a ve b 'lerin üst indisleri, ilgili matris elemanının kaç kere değiştiği ile ilgili bilgi vermek amacıyla gösterilmiştir. Buradan da görülebileceği üzere sol üst köşeden başlanarak diyagonal elemanın değerini 1 yapacak satır işlemi uygulandıktan sonra diyagonalin altında kalan sütun elemanlarını sıfırlayacak satır işlemleriyle çözüme devam edilir. Diyagonalin altında kalan üçgenin sıfırlanmasından sonra en alt satırda elde edilen durum değişkeni üst satırlarda yerine konarak diyagonalin üstünde kalan üçgen sıfırlanana kadar işleme devam edilir.

Çözüme ulaşmak için yapılması gereken çarpım sayısı $\frac{p^3}{3} + \frac{p^2}{1} - \frac{p}{3}$, toplam sayısı ise $\frac{p^3}{3} + \frac{p^2}{2} - \frac{5p}{6}$ olacaktır. Yine p^3 'lü ifadenin yanında p^2 veya p 'li ifadelerin önemi kalmaz. Bu durumda Gauss eleme yönteminin matris tersini alarak girişle çarpmaya göre 3.3 kat, Gauss-Jordan eleme yöntemine göre de 1.5 kat daha hızlı olduğu söylenebilir. Bellek tasarrufu açısından da Gauss-Jordan ile aynı mertebededir. 32×32 boylarındaki bir resim için yaklaşık 3.3×10^8 'er tane çarpım ve toplam işlemi yapmak gerekir.

$$\begin{array}{ccc}
\left[\begin{array}{c} a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \end{array} \right] & \rightarrow & \left[\begin{array}{c} 1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \end{array} \right] & \rightarrow & \left[\begin{array}{c} 1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ a^0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \\ a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 a^0 b^0 \end{array} \right] \dots
\end{array}$$

$$\begin{array}{ccc}
\left[\begin{array}{c} 1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \end{array} \right] & \rightarrow & \left[\begin{array}{c} 1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 1 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \end{array} \right] & \rightarrow & \left[\begin{array}{c} 1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 1 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \end{array} \right] \dots
\end{array}$$

$$\begin{array}{ccc}
\left[\begin{array}{c} 1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 1 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 1 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 b^3 \\ 0 0 0 1 a^4 a^4 a^4 a^4 a^4 a^4 a^4 a^4 b^4 \\ 0 0 0 0 1 a^5 a^5 a^5 a^5 a^5 a^5 a^5 a^5 b^5 \\ 0 0 0 0 0 1 a^6 a^6 a^6 a^6 a^6 a^6 a^6 a^6 b^6 \\ 0 0 0 0 0 0 1 a^7 a^7 a^7 a^7 a^7 a^7 a^7 a^7 b^7 \\ 0 0 0 0 0 0 0 1 a^8 a^8 a^8 a^8 a^8 a^8 a^8 a^8 b^8 \\ 0 0 0 0 0 0 0 0 1 a^9 a^9 a^9 a^9 a^9 a^9 a^9 a^9 b^9 \\ 0 0 0 0 0 0 0 0 0 1 a^{10} a^{10} a^{10} a^{10} a^{10} a^{10} a^{10} a^{10} b^{10} \\ 0 0 0 0 0 0 0 0 0 0 1 a^{11} b^{11} \\ 0 0 0 0 0 0 0 0 0 0 0 1 x \end{array} \right] & \rightarrow & \left[\begin{array}{c} 1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 a^1 b^1 \\ 0 1 a^2 a^2 a^2 a^2 a^2 a^2 a^2 a^2 b^2 \\ 0 0 1 a^3 a^3 a^3 a^3 a^3 a^3 a^3 a^3 b^3 \\ 0 0 0 1 a^4 a^4 a^4 a^4 a^4 a^4 a^4 a^4 b^4 \\ 0 0 0 0 1 a^5 a^5 a^5 a^5 a^5 a^5 a^5 a^5 b^5 \\ 0 0 0 0 0 1 a^6 a^6 a^6 a^6 a^6 a^6 a^6 a^6 b^6 \\ 0 0 0 0 0 0 1 a^7 a^7 a^7 a^7 a^7 a^7 a^7 a^7 b^7 \\ 0 0 0 0 0 0 0 1 a^8 a^8 a^8 a^8 a^8 a^8 a^8 a^8 b^8 \\ 0 0 0 0 0 0 0 0 1 a^9 a^9 a^9 a^9 a^9 a^9 a^9 a^9 b^9 \\ 0 0 0 0 0 0 0 0 0 1 a^{10} a^{10} a^{10} a^{10} a^{10} a^{10} a^{10} a^{10} b^{10} \\ 0 0 0 0 0 0 0 0 0 0 1 a^{11} b^{11} \\ 0 0 0 0 0 0 0 0 0 0 0 1 x \end{array} \right] & \rightarrow & \left[\begin{array}{c} 1 0 0 0 0 0 0 0 0 0 0 0 x \\ 0 1 0 0 0 0 0 0 0 0 0 0 x \\ 0 0 1 0 0 0 0 0 0 0 0 0 x \\ 0 0 0 1 0 0 0 0 0 0 0 0 x \\ 0 0 0 0 1 0 0 0 0 0 0 0 x \\ 0 0 0 0 0 1 0 0 0 0 0 0 x \\ 0 0 0 0 0 0 1 0 0 0 0 0 x \\ 0 0 0 0 0 0 0 1 0 0 0 0 x \\ 0 0 0 0 0 0 0 0 1 0 0 0 x \\ 0 0 0 0 0 0 0 0 0 1 0 0 x \\ 0 0 0 0 0 0 0 0 0 0 1 0 x \\ 0 0 0 0 0 0 0 0 0 0 0 1 x \end{array} \right]
\end{array}$$

Şekil 5.6: 3×4 boylarındaki bir resmin Gauss eleme yöntemi ile işlenmesi.

5.4.4 HSA'nın Durum Matrisinin Özel Yapısı

3×4 'lük bir resmi lineer bir yöntem ile işleme problemi örnek olarak alındığında, **A** durum matrisi en genel olarak aşağıdaki yapıdadır.

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} & a_{1,6} & a_{1,7} & a_{1,8} & a_{1,9} & a_{1,10} & a_{1,11} & a_{1,12} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} & a_{2,6} & a_{2,7} & a_{2,8} & a_{2,9} & a_{2,10} & a_{2,11} & a_{2,12} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} & a_{3,6} & a_{3,7} & a_{3,8} & a_{3,9} & a_{3,10} & a_{3,11} & a_{3,12} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} & a_{4,6} & a_{4,7} & a_{4,8} & a_{4,9} & a_{4,10} & a_{4,11} & a_{4,12} \\ a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} & a_{5,5} & a_{5,6} & a_{5,7} & a_{5,8} & a_{5,9} & a_{5,10} & a_{5,11} & a_{5,12} \\ a_{6,1} & a_{6,2} & a_{6,3} & a_{6,4} & a_{6,5} & a_{6,6} & a_{6,7} & a_{6,8} & a_{6,9} & a_{6,10} & a_{6,11} & a_{6,12} \\ a_{7,1} & a_{7,2} & a_{7,3} & a_{7,4} & a_{7,5} & a_{7,6} & a_{7,7} & a_{7,8} & a_{7,9} & a_{7,10} & a_{7,11} & a_{7,12} \\ a_{8,1} & a_{8,2} & a_{8,3} & a_{8,4} & a_{8,5} & a_{8,6} & a_{8,7} & a_{8,8} & a_{8,9} & a_{8,10} & a_{8,11} & a_{8,12} \\ a_{9,1} & a_{9,2} & a_{9,3} & a_{9,4} & a_{9,5} & a_{9,6} & a_{9,7} & a_{9,8} & a_{9,9} & a_{9,10} & a_{9,11} & a_{9,12} \\ a_{10,1} & a_{10,2} & a_{10,3} & a_{10,4} & a_{10,5} & a_{10,6} & a_{10,7} & a_{10,8} & a_{10,9} & a_{10,10} & a_{10,11} & a_{10,12} \\ a_{11,1} & a_{11,2} & a_{11,3} & a_{11,4} & a_{11,5} & a_{11,6} & a_{11,7} & a_{11,8} & a_{11,9} & a_{11,10} & a_{11,11} & a_{11,12} \\ a_{12,1} & a_{12,2} & a_{12,3} & a_{12,4} & a_{12,5} & a_{12,6} & a_{12,7} & a_{12,8} & a_{12,9} & a_{12,10} & a_{12,11} & a_{12,12} \end{bmatrix}$$

Ancak HSA matrisinin çok özel bir yapıda olması nedeniyle lineer HSA için Bölüm 5.3.5'de

oluşturulan $\mathbf{A}$ matrisi

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdot & \cdot & a_{1,5} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdot & \cdot & a_{2,6} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & a_{3,2} & a_{3,3} & a_{3,4} & \cdot & \cdot & a_{3,7} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & a_{4,3} & a_{4,4} & \cdot & \cdot & \cdot & a_{4,8} & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{5,1} & \cdot & \cdot & \cdot & a_{5,5} & a_{5,6} & \cdot & \cdot & a_{5,9} & \cdot & \cdot & \cdot & \cdot \\ \cdot & a_{6,2} & \cdot & \cdot & a_{6,5} & a_{6,6} & a_{6,7} & \cdot & \cdot & a_{6,10} & \cdot & \cdot & \cdot \\ \cdot & \cdot & a_{7,3} & \cdot & \cdot & a_{7,6} & a_{7,7} & a_{7,8} & \cdot & \cdot & a_{7,11} & \cdot & \cdot \\ \cdot & \cdot & \cdot & a_{8,4} & \cdot & \cdot & a_{8,7} & a_{8,8} & \cdot & \cdot & \cdot & a_{8,12} & \cdot \\ \cdot & \cdot & \cdot & \cdot & a_{9,5} & \cdot & \cdot & \cdot & a_{9,9} & a_{9,10} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & a_{10,6} & \cdot & \cdot & a_{10,9} & a_{10,10} & a_{10,11} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a_{11,7} & \cdot & \cdot & a_{11,10} & a_{11,11} & a_{11,12} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a_{12,8} & \cdot & \cdot & a_{12,11} & a_{12,12} & \cdot \end{bmatrix}$$

yapısındadır. Matrisin özelliklerini daha rahat inceleyebilmek için 0'ların yerine “.” sembolü kullanılmıştır. Matrisin sıfır olmayan elemanları

$$\tilde{\mathbf{A}} = \begin{bmatrix} 0 & e^{jw_{xo}} & 0 \\ e^{jw_{yo}} & -(4 + \lambda^2) & e^{-jw_{yo}} \\ 0 & e^{-jw_{xo}} & 0 \end{bmatrix} = \begin{bmatrix} 0 & A_{-1,0} & 0 \\ A_{0,-1} & A_{0,0} & A_{0,1} \\ 0 & A_{1,0} & 0 \end{bmatrix}$$

şablon indisleri cinsinden yazıldığında $\mathbf{A}$ matrisinin özellikleri daha belirgin hale getirilebilir.

$$\mathbf{A} = \begin{bmatrix} A_{0,0} & A_{0,1} & \cdot & \cdot & A_{1,0} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ A_{0,-1} & A_{0,0} & A_{0,1} & \cdot & \cdot & A_{1,0} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & A_{0,-1} & A_{0,0} & A_{0,1} & \cdot & \cdot & A_{1,0} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & A_{0,-1} & A_{0,0} & \cdot & \cdot & \cdot & A_{1,0} & \cdot & \cdot & \cdot & \cdot & \cdot \\ A_{-1,0} & \cdot & \cdot & \cdot & A_{0,0} & A_{0,1} & \cdot & \cdot & A_{1,0} & \cdot & \cdot & \cdot & \cdot \\ \cdot & A_{-1,0} & \cdot & \cdot & A_{0,-1} & A_{0,0} & A_{0,1} & \cdot & \cdot & A_{1,0} & \cdot & \cdot & \cdot \\ \cdot & \cdot & A_{-1,0} & \cdot & \cdot & A_{0,-1} & A_{0,0} & A_{0,1} & \cdot & \cdot & A_{1,0} & \cdot & \cdot \\ \cdot & \cdot & \cdot & A_{-1,0} & \cdot & \cdot & A_{0,-1} & A_{0,0} & \cdot & \cdot & \cdot & A_{1,0} & \cdot \\ \cdot & \cdot & \cdot & \cdot & A_{-1,0} & \cdot & \cdot & \cdot & A_{0,0} & A_{0,1} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & A_{-1,0} & \cdot & \cdot & A_{0,-1} & A_{0,0} & A_{0,1} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & A_{-1,0} & \cdot & \cdot & A_{0,-1} & A_{0,0} & A_{0,1} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & A_{-1,0} & \cdot & \cdot & A_{0,-1} & A_{0,0} & \cdot \end{bmatrix}$$

Bu matrise yakından bakıldığında içerdiği birçok alt matrisin birbirinin aynı olduğu görülür. Yani

$$\mathbf{R} = \begin{bmatrix} A_{-1,0} & \cdot & \cdot & \cdot \\ \cdot & A_{-1,0} & \cdot & \cdot \\ \cdot & \cdot & A_{-1,0} & \cdot \\ \cdot & \cdot & \cdot & A_{-1,0} \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} A_{0,0} & A_{0,1} & \cdot & \cdot \\ A_{0,-1} & A_{0,0} & A_{0,1} & \cdot \\ \cdot & A_{0,-1} & A_{0,0} & A_{0,1} \\ \cdot & \cdot & A_{0,-1} & A_{0,0} \end{bmatrix}$$

$$\mathbf{T} = \begin{bmatrix} A_{1,0} & \cdot & \cdot & \cdot \\ \cdot & A_{1,0} & \cdot & \cdot \\ \cdot & \cdot & A_{1,0} & \cdot \\ \cdot & \cdot & \cdot & A_{1,0} \end{bmatrix}$$

$$\circ = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

alt matrisleri tanımlanacak olursa $\mathbf{A}$ matrisi

$$\mathbf{A} = \begin{bmatrix} S & T & \circ \\ R & S & T \\ \circ & R & S \end{bmatrix}$$

şeklinde yazılabilir. Bu formda yazılabilen matrislere *blok Toeplitz* adı verilir. Ancak bu matrisin özel yapıda olması bununla sınırlı kalmamaktadır, hem alt matrislerden biri sıfır matrisidir, hem de $\mathbf{R}$ ve $\mathbf{T}$ alt matrisleri

$$\mathbf{R} = \begin{bmatrix} A_{-1,0} & \cdot & \cdot & \cdot \\ \cdot & A_{-1,0} & \cdot & \cdot \\ \cdot & \cdot & A_{-1,0} & \cdot \\ \cdot & \cdot & \cdot & A_{-1,0} \end{bmatrix} = A_{-1,0} \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 \end{bmatrix} = A_{-1,0} \mathbf{I}$$

$$\mathbf{T} = \begin{bmatrix} A_{1,0} & \cdot & \cdot & \cdot \\ \cdot & A_{1,0} & \cdot & \cdot \\ \cdot & \cdot & A_{1,0} & \cdot \\ \cdot & \cdot & \cdot & A_{1,0} \end{bmatrix} = A_{1,0} \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 \end{bmatrix} = A_{1,0} \mathbf{I}$$

şeklinde birim matrislerle bir sabitin çarpımına indirgenebilir. $\mathbf{A}$ matrisinin bu özel yapısı giriş resminin boyundan tamamen bağımsızdır. Yani $\mathbf{A}$ matrisi en genel olarak

$$\mathbf{A} = \begin{bmatrix} S & T & \circ & \circ & \cdots & \circ & \circ \\ R & S & T & \circ & \cdots & \circ & \circ \\ \circ & R & S & T & \cdots & \circ & \circ \\ \circ & \circ & R & S & \cdots & \circ & \circ \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \circ & \circ & \circ & \circ & \cdots & S & T \\ \circ & \circ & \circ & \circ & \cdots & R & S \end{bmatrix}$$

şeklinde yazılabilir. Alt matrislerin genel tanımları da aşağıdaki şekildedir:

$$\mathbf{S} = \begin{bmatrix} A_{0,0} & A_{0,1} & \cdot & \cdot & \cdots & \cdot & \cdot \\ A_{0,-1} & A_{0,0} & A_{0,1} & \cdot & \cdots & \cdot & \cdot \\ \cdot & A_{0,-1} & A_{0,0} & A_{0,1} & \cdots & \cdot & \cdot \\ \cdot & \cdot & A_{0,-1} & A_{0,0} & \cdots & \cdot & \cdot \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \cdot & \cdot & \cdot & \cdot & \cdots & A_{0,0} & A_{0,1} \\ \cdot & \cdot & \cdot & \cdot & \cdots & A_{0,-1} & A_{0,0} \end{bmatrix}$$

$$\mathbf{R} = A_{-1,0}\mathbf{I}$$

$$\mathbf{T} = A_{1,0} \mathbf{I}$$

5.4.5 Matrisin Tersini Almanın HSA'ya Uyarlanması

Bölüm 5.4.1'de $\mathbf{Ax} = \mathbf{b}$ denkleminin çözümü için

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

eşitliği önerilmişti. Ayrıca hesaplama güçlüğü çekilmesinin nedeni $\mathbf{A}$ matrisinin tersini hesaplamadaki zorluk olarak ortaya konmuştu. $\mathbf{A}$ matrisinin özel yapısı göz önüne alındığında ters alma işleminin alt matrisler üzerinde yapılacak işlemlerle gerçekleştirilebileceği söylenebilir. Bir matrisin tersini hesaplamak için kullanılan yöntemlerden en iyi bilinenleri *Gauss eleme* ile *blok ters alma* yöntemleridir. Gauss eleme yöntemini kullanarak matrisin tersini hesaplamak, aynı yöntem ile sonucu doğrudan bulmaya kıyasla çok daha fazla işlem gerektirir. Bu nedenle blok ters alma yöntemi üzerinde durulmuştur. Blok ters alma ise şu şekilde tanımlanabilir: Öncelikle $\mathbf{A}$ matrisi, $\mathbf{A}_A$ ve $\mathbf{A}_D$ alt matrislerinin kare olması koşulu ile

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_A & \mathbf{A}_B \\ \mathbf{A}_C & \mathbf{A}_D \end{bmatrix}$$

şeklinde parçalara ayrılır. Bu aşamadan sonra $\mathbf{A}^{-1}$ 'in $\mathbf{S}_A = \mathbf{A}_D - \mathbf{A}_C\mathbf{A}_A^{-1}\mathbf{A}_B$ olmak üzere

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{A}_A^{-1} + \mathbf{A}_A^{-1}\mathbf{A}_B\mathbf{S}_A^{-1}\mathbf{A}_C\mathbf{A}_A^{-1} & -\mathbf{A}_A^{-1}\mathbf{A}_B\mathbf{S}_A^{-1} \\ -\mathbf{S}_A^{-1}\mathbf{A}_C\mathbf{A}_A^{-1} & \mathbf{S}_A^{-1} \end{bmatrix}$$

şeklinde olacağı görülür. Yapılması gereken blok işlem sayısı; 2 ters alma, 6 çarpma ve 3 toplama şeklindedir. Ayrıca matrisin bölündüğü noktaya göre, $2 < w < 3$ ve k bir sabit olmak üzere $k \cdot p^w$ ile orantılı sayıda toplama ve çarpma işlemi gerektiği söylenebilir (Strassen, 1969).

Bu tezin hazırlanma aşamasında blok ters alma işleminin lineer HSA denklemi için optimizasyonu amacıyla birçok simülasyon yapılmış, ancak yeterli verim elde edilememiştir.

5.4.6 Öz Değerler ve Öz Vektörler Yardımıyla Çözüm

Bölüm 5.3.4'de elde edilen $\check{\mathbf{A}}$ şablonu

$$\check{\mathbf{A}} = \begin{bmatrix} 0 & \frac{-e^{jw_{xo}}}{4+\lambda^2} & 0 \\ \frac{-e^{jw_{yo}}}{4+\lambda^2} & 0 & \frac{-e^{-jw_{yo}}}{4+\lambda^2} \\ 0 & \frac{-e^{-jw_{xo}}}{4+\lambda^2} & 0 \end{bmatrix} = \begin{bmatrix} 0 & A_{-1,0} & 0 \\ A_{0,-1} & 0 & A_{0,1} \\ 0 & A_{1,0} & 0 \end{bmatrix}$$

ve oluşturulan $\mathcal{A}$ matrisi

$$\mathcal{A} = \left[\begin{array}{cccc|cccc|cccc} \cdot & a_{1,2} & \cdot & \cdot & a_{1,5} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{2,1} & \cdot & a_{2,3} & \cdot & \cdot & a_{2,6} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & a_{3,2} & \cdot & a_{3,4} & \cdot & \cdot & a_{3,7} & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & a_{4,3} & \cdot & \cdot & \cdot & \cdot & a_{4,8} & \cdot & \cdot & \cdot & \cdot \\ \hline a_{5,1} & \cdot & \cdot & \cdot & \cdot & a_{5,6} & \cdot & \cdot & a_{5,9} & \cdot & \cdot & \cdot \\ \cdot & a_{6,2} & \cdot & \cdot & a_{6,5} & \cdot & a_{6,7} & \cdot & \cdot & a_{6,10} & \cdot & \cdot \\ \cdot & \cdot & a_{7,3} & \cdot & \cdot & a_{7,6} & \cdot & a_{7,8} & \cdot & \cdot & a_{7,11} & \cdot \\ \cdot & \cdot & \cdot & a_{8,4} & \cdot & \cdot & a_{8,7} & \cdot & \cdot & \cdot & \cdot & a_{8,12} \\ \hline \cdot & \cdot & \cdot & \cdot & a_{9,5} & \cdot & \cdot & \cdot & \cdot & a_{9,10} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & a_{10,6} & \cdot & \cdot & a_{10,9} & \cdot & a_{10,11} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a_{11,7} & \cdot & \cdot & a_{11,10} & \cdot & a_{11,12} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a_{12,8} & \cdot & \cdot & a_{12,11} & \cdot \end{array} \right]$$

şeklinde. Her reel simetrik $\mathcal{A}$ matrisi için; sütunları matrisin öz vektörleri olan $\mathbf{T}$ ve matrisin öz değerlerini diyagonalı üzerinde içeren $\mathbf{\Lambda}$ kullanılarak

$$\mathcal{A} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^t$$

eşitliği yazılabilir. Benzer şekilde her Hermitian matris için de

$$\mathcal{A} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1} = \mathbf{T}\mathbf{\Lambda}\bar{\mathbf{T}}^t$$

eşitliği geçerlidir (Tavsanoglu, 2006). Buradaki $\bar{\mathbf{T}}$ matrisi $\mathbf{T}$ 'nin kompleks eşleniği olarak tanımlıdır. Bu durumda (5.8) eşitliği simetrik ve Hermityen matrisler için sırasıyla

$$\mathbf{x}^J = \mathbf{T}(\mathbf{I} - \mathbf{\Lambda})^{-1}\mathbf{T}^t\mathbf{b}$$

ve

$$\mathbf{x}^J = \mathbf{T}(\mathbf{I} - \mathbf{\Lambda})^{-1}\bar{\mathbf{T}}^t\mathbf{b}$$

şeklinde yazılabilir.

Burada varılmak istenen nokta şu şekilde özetlenebilir: Eğer $\mathcal{A}$ 'nın özel yapısını kullanarak $\mathbf{T}$ ve $\mathbf{\Lambda}$ 'yı hesaplamanın verimli bir yolu bulunabilirse yukarıdaki eşitlikler lineer HSA'nın simülasyonunda verimli bir şekilde kullanılabilir. Bu tezin hazırlanma aşamasında konuyla ilgili birçok simülasyon yapılmış, öz değerler ve öz vektörler arasında birçok ilişki tespit edilmiş, ancak $\mathbf{T}$ ve $\mathbf{\Lambda}$ matrislerini hesaplamanın verimli bir yolu bulunamamıştır.

5.4.7 Gauss–Jordan Eleme Yönteminin HSA'ya Uyarlanması

Bu bölümde Bölüm 5.4.2'de tanımlanan Gauss–Jordan eleme yönteminin lineer HSA'nın seyrek ve özel yapıdaki $\mathbf{A}$ matrisine göre uyarlanması konusu ele alınmıştır. Şekil 5.5'teki eleme işlemi aynı matrisi seyrek yapıda yazılarak yapıldığında Şekil 5.7 elde edilir.

$$\begin{bmatrix} a^0 a^0 0 0 a^0 0 0 0 0 0 0 0 & b^0 \\ a^0 a^0 a^0 0 0 a^0 0 0 0 0 0 0 & b^0 \\ 0 a^0 a^0 a^0 0 0 a^0 0 0 0 0 0 & b^0 \\ 0 0 a^0 a^0 0 0 0 a^0 0 0 0 0 & b^0 \\ a^0 0 0 0 a^0 a^0 0 0 a^0 0 0 0 & b^0 \\ 0 a^0 0 0 a^0 a^0 a^0 0 0 a^0 0 0 & b^0 \\ 0 0 a^0 0 0 a^0 a^0 a^0 0 0 a^0 0 & b^0 \\ 0 0 0 a^0 0 0 a^0 a^0 0 0 a^0 a^0 & b^0 \\ 0 0 0 0 a^0 0 0 0 a^0 a^0 0 0 & b^0 \\ 0 0 0 0 0 a^0 0 0 a^0 a^0 a^0 0 & b^0 \\ 0 0 0 0 0 0 a^0 0 0 a^0 a^0 a^0 & b^0 \\ 0 0 0 0 0 0 0 a^0 0 0 a^0 a^0 a^0 & b^0 \end{bmatrix} \cdots \begin{bmatrix} 1 a^1 a^1 a^1 a^1 0 0 0 0 0 0 0 & b^1 \\ 0 a^1 a^1 a^1 a^1 a^0 0 0 0 0 0 0 & b^1 \\ 0 a^1 a^1 a^1 a^1 0 a^0 0 0 0 0 0 & b^1 \\ 0 a^1 a^1 a^1 a^1 0 0 a^0 0 0 0 0 & b^1 \\ 0 a^1 a^1 a^1 a^1 a^0 0 0 a^0 0 0 0 & b^1 \\ 0 a^0 0 0 a^0 a^0 a^0 0 0 a^0 0 0 & b^0 \\ 0 0 a^0 0 0 a^0 a^0 a^0 0 0 a^0 0 & b^0 \\ 0 0 0 a^0 0 0 a^0 a^0 0 0 a^0 a^0 & b^0 \\ 0 0 0 0 a^0 0 0 0 a^0 a^0 0 0 & b^0 \\ 0 0 0 0 0 a^0 0 0 a^0 a^0 a^0 0 & b^0 \\ 0 0 0 0 0 0 a^0 0 0 a^0 a^0 a^0 & b^0 \\ 0 0 0 0 0 0 0 a^0 0 0 a^0 a^0 a^0 & b^0 \end{bmatrix} \cdots \begin{bmatrix} 1 0 a^2 a^2 a^2 a^2 0 0 0 0 0 0 & b^2 \\ 0 1 a^2 a^2 a^2 a^2 0 0 0 0 0 0 & b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^0 0 0 0 0 0 & b^2 \\ 0 0 a^2 a^2 a^2 a^2 0 a^0 0 0 0 0 & b^2 \\ 0 0 a^2 a^2 a^2 a^2 0 0 a^0 0 0 0 & b^2 \\ 0 0 a^2 a^2 a^2 a^2 a^0 0 0 a^0 0 0 & b^0 \\ 0 0 a^0 0 0 a^0 a^0 a^0 0 0 a^0 0 & b^0 \\ 0 0 0 a^0 0 0 a^0 a^0 0 0 a^0 a^0 & b^0 \\ 0 0 0 0 a^0 0 0 0 a^0 a^0 0 0 & b^0 \\ 0 0 0 0 0 a^0 0 0 a^0 a^0 a^0 0 & b^0 \\ 0 0 0 0 0 0 a^0 0 0 a^0 a^0 a^0 & b^0 \\ 0 0 0 0 0 0 0 a^0 0 0 a^0 a^0 a^0 & b^0 \end{bmatrix} \cdots \\
\begin{bmatrix} 1 0 0 a^3 a^3 a^3 a^3 0 0 0 0 0 & b^3 \\ 0 1 0 a^3 a^3 a^3 a^3 0 0 0 0 0 & b^3 \\ 0 0 1 a^3 a^3 a^3 a^3 0 0 0 0 0 & b^3 \\ 0 0 0 a^3 a^3 a^3 a^3 a^0 0 0 0 0 & b^3 \\ 0 0 0 0 a^3 a^3 a^3 a^3 0 a^0 0 0 0 & b^3 \\ 0 0 0 0 a^3 a^3 a^3 a^3 0 0 a^0 0 0 & b^3 \\ \cdots & \cdots \\ 0 0 0 0 a^3 a^3 a^3 a^3 a^0 0 0 a^0 0 & b^3 \\ 0 0 0 0 a^0 0 0 a^0 a^0 0 0 0 a^0 & b^0 \\ 0 0 0 0 a^0 0 0 0 a^0 a^0 0 0 & b^0 \\ 0 0 0 0 0 a^0 0 0 a^0 a^0 a^0 0 & b^0 \\ 0 0 0 0 0 0 a^0 0 0 a^0 a^0 a^0 & b^0 \\ 0 0 0 0 0 0 0 a^0 0 0 a^0 a^0 a^0 & b^0 \end{bmatrix} \cdots \begin{bmatrix} 1 0 0 0 0 0 0 0 0 a^4 a^4 a^4 a^4 & b^4 \\ 0 1 0 0 0 0 0 0 0 a^4 a^4 a^4 a^4 & b^4 \\ 0 0 1 0 0 0 0 0 0 a^4 a^4 a^4 a^4 & b^4 \\ 0 0 0 1 0 0 0 0 0 a^4 a^4 a^4 a^4 & b^4 \\ 0 0 0 0 1 0 0 0 0 a^4 a^4 a^4 a^4 & b^4 \\ 0 0 0 0 0 1 0 0 0 a^4 a^4 a^4 a^4 & b^4 \\ 0 0 0 0 0 0 1 0 0 a^4 a^4 a^4 a^4 & b^4 \\ 0 0 0 0 0 0 0 1 0 a^4 a^4 a^4 a^4 & b^4 \\ 0 0 0 0 0 0 0 0 1 a^4 a^4 a^4 a^4 & b^4 \\ 0 0 0 0 0 0 0 0 0 a^4 a^4 a^4 a^4 & b^4 \\ 0 0 0 0 0 0 0 0 0 a^4 a^4 a^4 a^4 & b^4 \end{bmatrix} \cdots \begin{bmatrix} 1 0 0 0 0 0 0 0 0 a^5 a^5 a^5 b^5 & b^5 \\ 0 1 0 0 0 0 0 0 0 a^5 a^5 a^5 b^5 & b^5 \\ 0 0 1 0 0 0 0 0 0 a^5 a^5 a^5 b^5 & b^5 \\ 0 0 0 1 0 0 0 0 0 a^5 a^5 a^5 b^5 & b^5 \\ 0 0 0 0 1 0 0 0 0 a^5 a^5 a^5 b^5 & b^5 \\ 0 0 0 0 0 1 0 0 0 a^5 a^5 a^5 b^5 & b^5 \\ 0 0 0 0 0 0 1 0 0 a^5 a^5 a^5 b^5 & b^5 \\ 0 0 0 0 0 0 0 1 0 a^5 a^5 a^5 b^5 & b^5 \\ 0 0 0 0 0 0 0 0 1 a^5 a^5 a^5 b^5 & b^5 \\ 0 0 0 0 0 0 0 0 0 a^5 a^5 a^5 b^5 & b^5 \\ 0 0 0 0 0 0 0 0 0 a^5 a^5 a^5 b^5 & b^5 \end{bmatrix} \cdots \\
\begin{bmatrix} 1 0 0 0 0 0 0 0 0 0 a^6 a^6 b^6 & b^6 \\ 0 1 0 0 0 0 0 0 0 0 a^6 a^6 b^6 & b^6 \\ 0 0 1 0 0 0 0 0 0 0 0 a^6 a^6 b^6 & b^6 \\ 0 0 0 1 0 0 0 0 0 0 0 a^6 a^6 b^6 & b^6 \\ 0 0 0 0 1 0 0 0 0 0 0 a^6 a^6 b^6 & b^6 \\ 0 0 0 0 0 1 0 0 0 0 0 a^6 a^6 b^6 & b^6 \\ \cdots & \cdots \\ 0 0 0 0 0 0 1 0 0 0 0 a^6 a^6 b^6 & b^6 \\ 0 0 0 0 0 0 0 1 0 0 0 a^6 a^6 b^6 & b^6 \\ 0 0 0 0 0 0 0 0 1 0 a^6 a^6 b^6 & b^6 \\ 0 0 0 0 0 0 0 0 0 1 a^6 a^6 b^6 & b^6 \\ 0 0 0 0 0 0 0 0 0 0 a^6 a^6 b^6 & b^6 \\ 0 0 0 0 0 0 0 0 0 0 a^6 a^6 b^6 & b^6 \end{bmatrix} \cdots \begin{bmatrix} 1 0 0 0 0 0 0 0 0 0 0 a^7 b^7 & b^7 \\ 0 1 0 0 0 0 0 0 0 0 0 a^7 b^7 & b^7 \\ 0 0 1 0 0 0 0 0 0 0 0 a^7 b^7 & b^7 \\ 0 0 0 1 0 0 0 0 0 0 0 a^7 b^7 & b^7 \\ 0 0 0 0 1 0 0 0 0 0 0 a^7 b^7 & b^7 \\ 0 0 0 0 0 1 0 0 0 0 0 a^7 b^7 & b^7 \\ 0 0 0 0 0 0 1 0 0 0 0 a^7 b^7 & b^7 \\ 0 0 0 0 0 0 0 1 0 0 0 a^7 b^7 & b^7 \\ 0 0 0 0 0 0 0 0 1 0 a^7 b^7 & b^7 \\ 0 0 0 0 0 0 0 0 0 1 a^7 b^7 & b^7 \\ 0 0 0 0 0 0 0 0 0 0 a^7 b^7 & b^7 \end{bmatrix} \cdots \begin{bmatrix} 1 0 0 0 0 0 0 0 0 0 0 0 & x \\ 0 1 0 0 0 0 0 0 0 0 0 0 & x \\ 0 0 1 0 0 0 0 0 0 0 0 0 & x \\ 0 0 0 1 0 0 0 0 0 0 0 0 & x \\ 0 0 0 0 1 0 0 0 0 0 0 0 & x \\ 0 0 0 0 0 1 0 0 0 0 0 0 & x \\ 0 0 0 0 0 0 1 0 0 0 0 0 & x \\ 0 0 0 0 0 0 0 1 0 0 0 0 & x \\ 0 0 0 0 0 0 0 0 1 0 0 0 & x \\ 0 0 0 0 0 0 0 0 0 1 0 0 & x \\ 0 0 0 0 0 0 0 0 0 0 1 & x \end{bmatrix}$$

Şekil 5.7: 3×4 boylarındaki bir resmin lineer HSA için uyarlanmış Gauss–Jordan eleme yöntemi ile işlenmesi.

Şekil 5.7'nin üzerinde kalın harfler ile gösterilmiş olan ve eleme işlemi ilerledikçe matris üzerinde dolaşan dikdörtgen şeklindeki bölge, her bir sütunun sıfırlanması işlemi sırasında satır işlemlerinden etkilenen matris elemanlarını göstermektedir. O halde her bir sütunu sıfırlamak için tüm matris üzerinde işlem yapmak yerine yalnızca bu dikdörtgen bölge üzerinde işlem yapmak yeterli olacaktır. Gauss–Jordan eleme yönteminin lineer HSA'nın matrislerine uyarlanması sayesinde, eleme işlemi için yapılması gereken toplam ve çarpım sayıları, $M \times N$ boylarındaki bir resim ve $p = M \cdot N$ için, klasik Gauss–Jordan eleme yöntemindeki $\frac{p^3}{2}$ seviyesinden $\frac{(N+1)p^2}{2}$ 'ye düşmüştür. Sonuç olarak lineer HSA simülasyonunda uyarlanmış Gauss–Jordan eleme yönteminin kullanılması durumunda klasik Gauss–Jordan eleme yöntemine göre yaklaşık M kat, klasik Gauss eleme yöntemine göre ise $\frac{2M}{3}$ katlık bir hız kazancı elde edilir. 32×32 boylarındaki bir resim için yaklaşık 1.7×10^7 'şer tane çarpım ve toplam işlemi yapmak gerekir.

Kaplanan alan olarak düşünüldüğünde, Gauss–Jordan eleme yönteminin lineer HSA'nın matrislerine uyarlanması sonucunda matrisin yaklaşık $\frac{(p-N)^2}{2} = \frac{((M-1)^2 N^2)}{2}$ sayıda elema-

gerçekleştirilmesi için gereken çarpım ve toplam sayısı yaklaşık olarak $p(N+1)^2$ 'dir. Bu değerin de p^2 ile orantılı olduğu göz önüne alındığında klasik Gauss eleme yöntemine kıyasla $\frac{p}{3}$ kat hızlı işlem yapıldığı söylenebilir. Bu yöntemin lineer HSA'a uyarlanmış Gauss-Jordan eleme yönteminden de yaklaşık $\frac{N+1}{2}$ kat hızlı olduğu göz önüne alındığında oldukça yüksek bir hız elde edildiği söylenebilir. 32×32 boylarındaki bir resim için yaklaşık 1.1×10^6 'şar tane çarpım ve toplam işlemi yapmak gerekir.

Bellek gereksinimi açısından lineer HSA'a uyarlanmış Gauss eleme yöntemi göz önüne alındığında; durum matrisinin diyagonalinin hem altında, hem de üstünde kalan üçgenlerin elemanlarının yaklaşık $\frac{(p-N)^2}{2} = \frac{(M-1)^2 N^2}{2}$ 'şer tanesinin işlem süresince sıfır kaldığı görülür. Bu durumda p^2 tane matris elemanından işlemin başından sonuna kadar değişime uğramış olanların sayısı yalnızca $(2N-1)p$ kadardır. Geriye kalan bütün matris elemanlarının eleme işleminin başından sonuna kadar sıfır kaldığı düşünüldüğünde, bu matris elemanları için bellek ayırmaya gerek olmadığı görülür. Bunu için **A** matrisine uygun indis dönüşümü yapıldığında ve fazla kısımlar kesildiğinde basit bir algoritma ile matrisi saklamak için gereken RAM bölgesi yaklaşık $\frac{M}{2}$ kat azaltılmış olur. O halde Şekil 5.8'de ele alınan örnekteki

$$[\mathbf{A}|\mathbf{b}] = \left[\begin{array}{cccccccccccc|c} a^0 & a^0 & 0 & 0 & a^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b^0 \\ a^0 & a^0 & a^0 & 0 & 0 & a^0 & 0 & 0 & 0 & 0 & 0 & 0 & b^0 \\ 0 & a^0 & a^0 & a^0 & 0 & 0 & a^0 & 0 & 0 & 0 & 0 & 0 & b^0 \\ 0 & 0 & a^0 & a^0 & 0 & 0 & 0 & a^0 & 0 & 0 & 0 & 0 & b^0 \\ a^0 & 0 & 0 & 0 & a^0 & a^0 & 0 & 0 & a^0 & 0 & 0 & 0 & b^0 \\ 0 & a^0 & 0 & 0 & a^0 & a^0 & a^0 & 0 & 0 & a^0 & 0 & 0 & b^0 \\ 0 & 0 & a^0 & 0 & 0 & a^0 & a^0 & a^0 & 0 & 0 & a^0 & 0 & b^0 \\ 0 & 0 & 0 & a^0 & 0 & 0 & a^0 & a^0 & 0 & 0 & 0 & a^0 & b^0 \\ 0 & 0 & 0 & 0 & a^0 & 0 & 0 & 0 & a^0 & a^0 & 0 & 0 & b^0 \\ 0 & 0 & 0 & 0 & 0 & a^0 & 0 & 0 & a^0 & a^0 & a^0 & 0 & b^0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a^0 & 0 & 0 & a^0 & a^0 & a^0 & b^0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a^0 & 0 & 0 & a^0 & a^0 & b^0 \end{array} \right]$$

sisteminin **A** matrisi

$$\mathbf{A}_s = \mathbf{A}_{i,j}|_{j \rightarrow (j-i+N+1)}; \quad 1 < (j-i+N+1) < (2N+1)$$

şeklinde bir dönüşüme tabi tutulduğunda yeni elde edilecek olan indirgenmiş $[\mathbf{A}_s|\mathbf{b}]$ sistemi aşağıdakişekle dönüşür ve buradan da görülebileceği üzere bellek gereksinimi p^2 mertebesinden $2Np$ mertebesine düşürülmüştür.

$$[\mathbf{A}_s|\mathbf{b}] = \underbrace{\left[\begin{array}{cccccc|c} \cdot & \cdot & \cdot & \cdot & a^0 & a^0 & \cdot & \cdot & a^0 & b^0 \\ \cdot & \cdot & \cdot & a^0 & a^0 & a^0 & \cdot & \cdot & a^0 & b^0 \\ \cdot & \cdot & \cdot & a^0 & a^0 & a^0 & \cdot & \cdot & a^0 & b^0 \\ \cdot & \cdot & \cdot & a^0 & a^0 & \cdot & \cdot & \cdot & a^0 & b^0 \\ a^0 & \cdot & \cdot & \cdot & a^0 & a^0 & \cdot & \cdot & a^0 & b^0 \\ a^0 & \cdot & \cdot & a^0 & a^0 & a^0 & \cdot & \cdot & a^0 & b^0 \\ a^0 & \cdot & \cdot & a^0 & a^0 & a^0 & \cdot & \cdot & a^0 & b^0 \\ a^0 & \cdot & \cdot & a^0 & a^0 & \cdot & \cdot & \cdot & a^0 & b^0 \\ a^0 & \cdot & \cdot & \cdot & a^0 & a^0 & \cdot & \cdot & \cdot & b^0 \\ a^0 & \cdot & \cdot & a^0 & a^0 & a^0 & \cdot & \cdot & \cdot & b^0 \\ a^0 & \cdot & \cdot & a^0 & a^0 & a^0 & \cdot & \cdot & \cdot & b^0 \\ a^0 & \cdot & \cdot & a^0 & a^0 & \cdot & \cdot & \cdot & \cdot & b^0 \end{array} \right]}_{2(N+1)} \Bigg\}^{MN}$$

5.5 Lineer HSA Simülasyonu Sonuçları

Bu tezin içinde tanımlanan tüm lineer HSA çözüm yöntemleri ile Gauss ve Gabor benzeri lineer HSA'ların simülasyonları yapılmıştır. Simülasyonlarda 32×32 boylarında 45° doğrultusunda salınım yapan iki boyutlu bir kare dalga resmi kullanılmıştır. Elde edilen simülasyon sonuçları Şekil 5.9 ve 5.10'da verilmiştir.

Şekil 5.9'da $\lambda = 0.5$ için Gauss benzeri alçak geçiren HSA süzgecinin girişi, çıkışı ve girişi ile çıkışının Fourier dönüşümleri verilmiştir. Alçak geçiren özelliğinden dolayı kare dalganın 1. harmoniği süzgeçten geçmiş, diğer harmonikler zayıflamıştır. Şekil 5.10'da ise bant geçiren Gabor benzeri HSA süzgecinin girişi, çıkışı ve bunların Fourier dönüşümleri verilmiştir. $\lambda = 0.001$ olacak şekilde çok dar bir bant alınmış ve yalnızca üçüncü harmoniğin süzgeçten geçmesine izin verilmiştir.

Bu bölüm içerisinde ele alınan tüm lineer HSA sayısal çözüm yöntemleri için, çift çekirdekli 64-bitlik bir PC üzerinde ve Linux işletim sistemi altında MATLAB simülasyonları yapılmış ve sonuçta aynı çıkışlar elde edilmiştir. Simülasyonların süreleri Çizelge 5.1'de verilmiştir. Yapılan simülasyonların arasında gözlenen farklar aşağıdaki şekilde sıralanabilir:

Çizelge 5.1: MATLAB simülasyonları sonucunda $t_{\text{Euler}}/t_{\text{Gauss}}$ hız oranı için elde edilen sonuçlar.

$t_{\text{Euler}}/t_{\text{Gauss}}$	Hata		
	%5	%0.1	%0.001
$\lambda = 0.001$	2.7	4.5	8.5
$\lambda = 0.5$	0.19	0.31	0.55

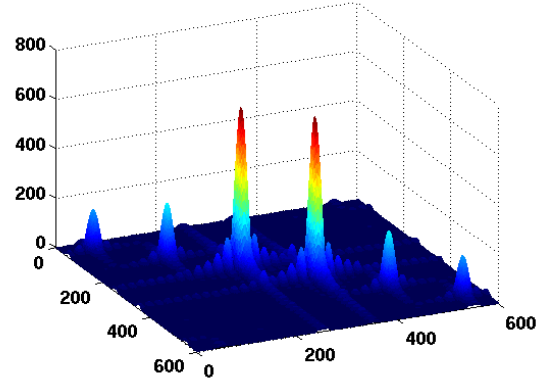
- Simülasyon süresi sonlu sayıda işlem ile sonuca varan sayısal çözüm yöntemleri için

sabit, sonsuz sayıda işlem ile sonuca yakınsayan yaklaşık çözüm yöntemleri için değişkendir.

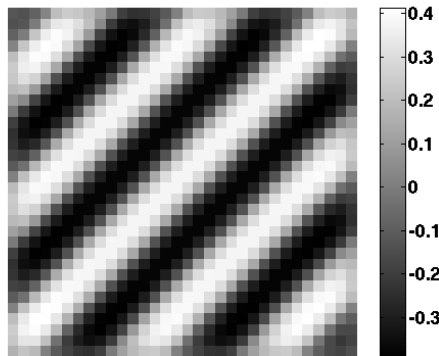
- Yaklaşık çözüm yöntemlerinin belli bir yüzdelik hata ile sonuca ulaşma süresinin bant genişliği λ ile ters orantılı olduğu gözlenmiştir. Ayrıca giriş resminin dokusu da sonuca ulaşma süresini etkilemektedir.
- Yaklaşık çözüm yöntemlerinden en hızlısı Euler ile Infinite Impulse Responseri yaklaşıklığıdır.
- Tam çözüm yöntemlerinden en hızlısı lineer HSA durum matrisi için hız ve yer optimizasyonu yapılmış Gauss eleme yöntemidir.
- Bellek gereksinimi açısından Euler ileri yaklaşıklığı Gauss eleme yöntemine göre kıyaslanamayacak kadar avantajlıdır.
- $\lambda = 0.001$ için Euler ileri yaklaşıklığının sonuca %5, %0.1 ve %0.0001 hata ile yaklaşması ile optimize edilmiş Gauss eleme yönteminin tam sonuca ulaşması arasındaki $t_{\text{Euler}}/t_{\text{Gauss}}$ oranı sırasıyla 2.7, 4.5 ve 8.5'dir (Çizelge 5.1).



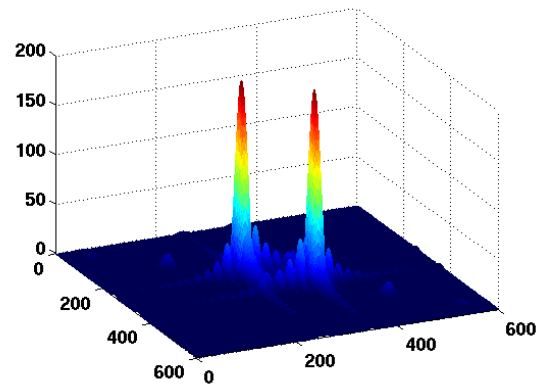
(a) Giriş resmi



(b) Giriş resminin Fourier dönüşümü



(c) Çıkış resmi

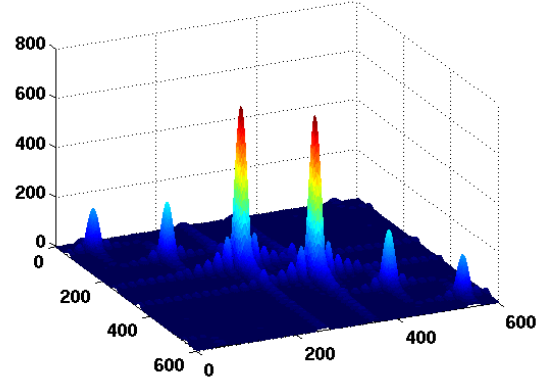


(d) Çıkış resminin Fourier dönüşümü

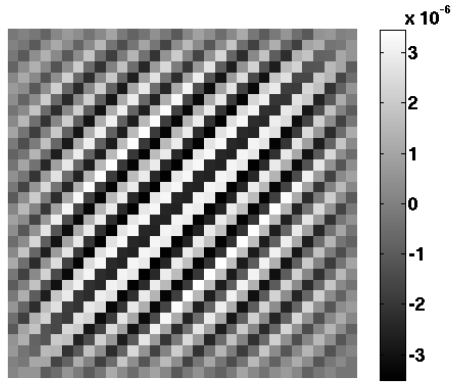
Şekil 5.9: 32×32 boylarındaki bir kare dalganın $\lambda = 0.5$ olarak alınmış bir Gauss benzeri alçak geçiren HSA süzgeç ile süzülmesi ile elde edilen sonuçlar.



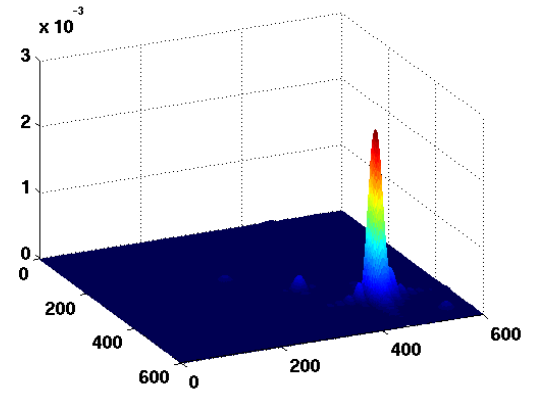
(a) Giriş resmi



(b) Giriş resminin Fourier dönüşümü



(c) Çıkış resmi



(d) Çıkış resminin Fourier dönüşümü

Şekil 5.10: 32×32 boylarındaki bir kare dalganın $\lambda = 0.001$ olarak alınmış bir Gabor benzeri bant geçiren HSA süzgeç ile süzülmesi ile elde edilen sonuçlar.

- $\lambda = 0.5$ için Euler ileri yaklaşıklığının sonuca %5, %0.1 ve %0.0001 hata ile yaklaşması ile optimize edilmiş Gauss eleme yönteminin tam sonuca ulaşması arasındaki $t_{\text{Euler}}/t_{\text{Gauss}}$ oranı sırasıyla 0.19, 0.31 ve 0.55'dir (Çizelge 5.1).
- Yukarıdaki iki sonuca bakılarak büyük λ değerleri için *Euler* yaklaşıklığının, küçük λ değerleri için ise optimize edilmiş Gauss eleme yönteminin daha hızlı olduğu söylenebilir.

6 SONUÇLAR

Temel olarak iki parçadan oluşan bu tezin ilk yarısında TY-HSA eşitlikleri tanımlanarak sayısal devreler üzerinde çalıştırılmak amacıyla ayrıklaştırılmıştır. Ayrıklaştırılmış olan TY-HSA eşitliklerinin MATLAB simülasyonları yapılarak sonuçlar incelenen makalelerde analog olarak elde edilen çıkışlar ile karşılaştırılmış, başarılı sonuçlar elde edildiği gözlenmiştir. Elde edilen simülasyon sonuçlarının çok bacaklı robotların bacak senkronizasyonunda kullanımı için tırtıl ve altı bacaklı robot örnekleri verilmiş, bacak kontrol bağlantıları gösterilerek yürüme mantığı ayrıntılı olarak anlatılmıştır.

Özgün bir çalışma olan tezin ikinci yarısında robot üzerindeki kameradan alınan görüntülerin işlenmesi konusu ele alınmıştır. Öncelikle HSA ile lineer olarak gerçekleştirilebilen Gauss ve Gabor benzeri süzgeçler tanımlanmıştır. Sürekli halde asimptotik kararlı olan HSA sistemlerinin sonsuzdaki çözümünün, işlem yükü nedeniyle durum matrisleri yardımıyla elde edilememesi problemi üzerinde durulmuştur. Öncelikle bilinen lineer çözüm yöntemleri ve bunların işlem ve bellek gereksinimleri ortaya konmuş, daha sonra bu yöntemler üzerinde indirgemeler yapılarak işlem süreleri Euler ileri yaklaşıklığı gibi sonsuz işlem ile sonuca ulaşan yöntemler ile aynı mertebeye indirilmiştir.

Lineer HSA'ların durum matrisleri yardımıyla çözümü için yapılan indirgemeler başarı miktarı açısından değerlendirilecek olursa elde edilen sonuçların biraz yanıltıcı olduğu görülür. MATLAB bazı işlemleri indirgenmiş, bazılarını da indirgenmemiş bir şekilde yaptığından dolayı beklenen sonuçlar ile elde edilenler arasında bir fark gözükmemektedir. FPGA veya DSP tabanlı bir sistemde işletim sistemi optimizasyonu olmadan elde edilmesi beklenen sonuçlar Çizelge 6.1'de verilmiştir.

Çizelge 6.1: FPGA veya DSP tabanlı bir sistemde gözlenmesi beklenen $t_{\text{Euler}}/t_{\text{Gauss}}$ hız oranları.

$t_{\text{Euler}}/t_{\text{Gauss}}$	Hata		
	%5	%0.1	%0.001
$\lambda = 0.001$	5.22	8.72	14.03
$\lambda = 0.5$	0.36	0.61	1.06

Bu tezin devamında elde edilen sonuçların FPGA üzerinde gerçek zamanlı olarak gerçekleştirilmesi konusu ele alınabilir. Tezin ilk kısmında ele alınan robotun hareketi için Euler ileri yaklaşıklığı ile ayrıklaştırılmış TY-HSA kullanılarak bu tezde elde edilen kendini yenileyen dalgaların yürüme veya koşma hızında elde edilmesi planlanmaktadır. Kameradan alınan

görüntülerin işlenmesinde ise 32×32 boylarındaki bir resim için saniyede minimum 24 karelik bir hız, daha büyük resimler için ise resmin parçalı olarak işlenmesi ve durum matrisinin daha çok indirgenmesi için ek yöntemler geliştirilmesi hedeflenmektedir. Örneğin LU faktörizasyonu yapılması durumunda, optimize edilmiş Gauss eleme yöntemine göre N kat daha hızlı işlem yapmak mümkün olacaktır.

KAYNAKLAR

- P. Arena ve L. Fortuna. Analog cellular locomotion control of hexapod robots. *Control Systems Magazine, IEEE*, 22, 2002.
- P. Arena, R. Caponetto, L. Fortuna, ve G. Manganaro. Cellular neural networks to explore complexity. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 1, 1997.
- P. Arena, Baglio S., L. Fortuna, ve G. Manganaro. Self-organization in a two-layer cnn. *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, 45, 1998.
- P. Arena, L. Fortuna, M. Frasca, ve G. Sicurella. An adaptive, self-organizing dynamical system for hierarchical control of bio-inspired locomotion. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 34(4):1823–1837, 2004.
- L. O. Chua. Special issue on nonlinear waves, patterns and spatio-temporal chaos. *IEEE Transactions on Circuits And Systems*, 42, 1995.
- L.O. Chua ve T Roska. *Cellular Neural networks and Visual Computing*. Cambridge University Press, UK, 2002.
- L.O. Chua ve L. Yang. Cellular neural networks: Theory. *IEEE Trans. Circuits Systems*, 35, 1988.
- L. Goras, L. O. Chua, ve G. Manganaro. Turing patterns in cnns—part 1: Once over lightly. *IEEE Transactions on Circuits And Systems*, 42:602–611, 1995.
- V. I. Krinsky. *Self-organisation, Auto-Waves and Structures Far From Equilibrium*. Springer, Berlin, 1984.
- J. D. Murray. *Mathematical Biology I*. Springer, Berlin, 2002.
- B.E. Shi. *Spatio-temporal filtering with cellular neural networks*. PhD thesis, Dept. Elect. Eng. Computer Sci., Univ. California Berkeley, 1994.
- B.E. Shi. Gabor-type filtering in space and time with cellular neural networks. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 45(2): 121–132, 1998.

- V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969.
- V. Tavsanoğlu. Jacobi’s iterative method for solving linear equations and the simulation of linear cnn. *IEEE Workshop on Cellular Neural Networks and their Applications*, pages 159–163, 2006.
- A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society (B)*, 237:37–72, 1952.

ÖZGEÇMİŞ

Doğum tarihi 21.09.1982

Doğum yeri İstanbul

Lise 1996–1999

Özel Getronagan Ermeni Lisesi

Lisans 1999–2004

Yıldız Teknik Üniversitesi
Elektrik–Elektronik Fakültesi
Elektronik ve Haberleşme Mühendisliği Bölümü

Yüksek Lisans 2004–2007

Yıldız Teknik Üniversitesi
Elektronik ve Haberleşme Müh. Anabilim Dalı
Elektronik Programı

Çalıştığı kurumlar

2004–2005

ENTES Elektronik Cihazlar İmalat ve
Ticaret A.Ş

2005–Devam ediyor

YTÜ Elektronik ve Haberleşme Mühendisliği
Bölümü Araştırma Görevlisi