

39618

İKİ BOYUTLU KAFES PARAMETRELERİNİN
SINIRLI VERİ ALANLARINDAN HESAPLANMASI

YÜKSEK LİSANS TEZİ

Müh. Nurşen YILDIZ

Tezin Enstitüye Verildiği Tarih : 13 Haziran 1994

Tezin Savunulduğu Tarih : 30 Haziran 1994

Tez Danışmanı : Prof.Dr.Ahmet Hamdi KAYRAN

Diğer Jüri Üyeleri : Prof.Dr.Mithat İDEMEN

Doç.Dr.Ümit AYGÖLÜ

F.C. E. KURULU
DOĞUMANTAYIN İZLENİ

HAZİRAN 1994

ÖNSÖZ

Bir boyutlu kafes süzgeçleri son yıllarda yapılan çalışmalar sonucunda, yaygın kullanım alanları bulmuştur. Elde edilen başarılı sonuçlar, kullanılan yöntemlerin iki boyutlu durumlar için de genelleştirilmesi yolundaki çalışmalara hız kazandırmıştır.

Tezin konusu; iki boyutlu rastgele alanların Burg Yöntemi esas alınarak, Kayran tarafından geliştirilen model ile incelenmesidir. Bu yöntemle amaçlanan sınırlı veri kayıtları için, gerçek katsayıların bulunmasında diğer yöntemlerden daha başarılı sonuçlar elde edilebileceğini göstermektir.

Çalışmam süresince bana yardımcı olan ve yön veren değerli hocam Sayın Prof. Dr. Ahmet Hamdi KAYRAN' a teşekkür ederim.

İstanbul, Haziran 1994

Nurşen YILDIZ

İÇİNDEKİLER

ÖZET	IV
SUMMARY.....	V
BÖLÜM 1. GİRİŞ.....	1
BÖLÜM 2. BİR BOYUTLU KAFES SÜZGEÇLERİ.....	4
2.1 Sonlu Geçmişe Dayalı Doğrusal Kestirim: Levinson-Durbin Özyinelemesi.....	4
2.2 Kafes Kestirimciler.....	7
2.3 Burg Yöntemi.....	11
BÖLÜM 3. İKİ BOYUTLU KAFES SÜZGEÇLERİ.....	14
BÖLÜM 4. KAFES PARAMETRELERİNİN 2-B 'LU AKTARIM İŞLEVİ İLE İLİŞKİSİ.....	35
BÖLÜM 5. SENTEZ MODELİ VE KARARLILIK.....	38
5.1 2-B 'lu Kararlılık Teoremleri.....	38
5.2 Sentez Modeli: Kafes Yapısı Kararlılık Koşulları.....	41
SONUÇLAR.....	46
KAYNAKLAR.....	47
EKLER.....	49
EK. A. Dik Kafes Süzgeci ile Burg yöntemi Esas Alınarak AR Katsayıların Kestirilmesi ve Sonuçların Yule-Walker Yöntemi ile karşılaştırılması.....	49
EK. B. Program Listeleri.....	68
ÖZGEÇMİŞ.....	99

ÖZET

Bir boyutta geliştirilen kafes yapısının bir uzantısı olarak iki boyutlu (2-B)dik kafes süzgeçleri geliştirilmiştir [7]. Önerilen yöntem çeyrek düzlem veya simetrik olmayan yarı düzleme uygulanabilmektedir.

Bu metod ile matris tersi alınmadan iki boyutlu genişletilmiş normal denklemden, kafes parametreleri kullanılarak kestirim yanılğı süzgeç katsayılarını hesaplamak için matris formunda Levinson algoritmasının tam çözümü sunulmaktadır.

Bu çalışmada, sınırlı veri uzunluğuna sahip rastgele alanların özbağlanımlı (Autoregressive = AR) modellenmesi için 2-B 'lu dik kafes süzgeçleri sunulmuş, yöntemin doğruluğu simülasyon yolu ile desteklenmiştir.

Öncelikle çeyrek veya simetrik olmayan yarı düzlem için seçilen maskeye göre özyinelemeli AR veri alanı yaratılmıştır. Yaratılan 2-B 'lu AR veri alanı uygun indeksleme ile 1-B 'lu gibi düşünülebilir. Yine seçilen indekslemeye bağlı olarak birinci çeyrek düzlem geri yönde kestirim yanılğı süzgeci, diğer çeyrek düzlemlerde ileri yönde kestirim yanılğı süzgeçlerine karşılık gelir.

Elde edilen sonuçlar incelendiğinde, sınırlı veri uzunlukları için, örnek özilişkilerin kullanıldığı yönleme - Bu yöntem Yule-Walker çözümünü vermektedir. - göre daha başarılı sonuçlar elde edildiği gözlenmiştir [18]. Veri boyutu artırıldığında, her iki yöntemle de oldukça iyi kestirimler yapılabildiği görülmüştür.

İleri ve geri yönlerde kestirim alanlarının dik olması, 2-B 'lu sistem parametrelerinin çözümünde LMS ve RLS gibi adaptif algoritmaların kullanılmasına olanak sağlayacaktır.

THE CALCULATION OF THE 2-D LATTICE PARAMETERS FROM SHORT DATA RECORDS

SUMMARY

The multi-dimensional digital signal processing has been developing quite rapidly due to the fact that it has many applications possibilities in various fields. In fields such as image processing, seismic prospecting, visual data communications and target tracking, the data to be processed are inherently multi-dimensional in character and the theoretical developments in 2-D signal processing methods can be applied to these fields.

The one dimensional lattice structure occurs in the analysis and synthesis of speech for simulating the vocal tract and also more generally in systems for linear prediction. It allows the realization of FIR and IIR filters. Lattice form algorithms are modular in structure; hence, if the requirement calls for increasing the order of the predictor, one can simply add one or more stages (as desired) without affecting earlier computations. This modular structure makes lattice filters attractive candidates for VLSI implementation.

The most important facility of a lattice structure arises from its property of orthogonality. The property allows the filter to be updated in order, without recalculation of the previous lower order filter coefficients. Lattice algorithms owe their robust numerical behaviour to this property of orthogonality [4].

In recent years there has been quite a lot of research studies directed to the development of 2-D equivalent lattice structures. However all these formulations are capable of implementing only a restricted class of transfer functions. In the literature, a fundamental approach to modeling 2-D fields by the reflection coefficients was realized by Marzetta [13] who has generalized some results to the 2-D case. He proposed a half-plane support which is infinite in one of the two dimensions. This approach while keeping many of the nice characteristics of the 1-D lattice filters, leads to very long delay filters. Marzetta's algorithm has been successfully applied to 2-D recursive filter design and to linear predictive coding of images.

A different approach proposed by Parker and Kayran [14], simultaneously introduces many points in the support when the model order is increased. The filter uses a quarter-plane support and introduces three parameters at each order update. Therefore, it lacks sufficient parameters to represent all classes of 2-D AR quarter-plane filters. More importantly, it lacks the property of orthogonality so that the cascading of stages may not lead to an optimum filter [1].

Recently, Ertüzün [3] presented a new and improved lattice structure developed from the three parameter lattice filter. It was shown that this new structure approximates the maximum entropy more closely compared to the three-parameter structure. The increase in entropy naturally leads to a more reliable and better modeling of AR fields.

A relation between multi-channel AR model and single channel 2-D models with quadrant support is proposed by Therrien [17] where simultaneous computation of parameters of all four quarter-plane filters is possible. This method is generalized to multichannel 2-D models and applied to the problem of estimation of the 2-D autospectral and cross spectral components.

Lenk and Parker [9], [10] have extended the well-known Levinson and Schur algorithms to the 2-D case for modeling stationary random fields. In this contribution, tensor concepts were used to derive lattice filters presented by considering orthogonalizing coordinate transformations. Kwan and Lui [8] showed that this structure inherits most of the nicer characteristics of the one dimensional lattice filter such as high modularity, low coefficient sensitiveness, low round off noise and elimination of internal overflow.

Türe [18], recently investigated the theory developed by Kayran about the orthogonal 2-D lattice structure for AR modeling of random fields. He showed the validity of the mentioned theory for the given covariance case and by using sample autocorrelation values (Yule-Walker Method) of the original data fields. The result of his simulations showed that the computing of sample autocorrelations would not give sufficient results for short data records. As the data record gets longer, it was observed that the resulting estimation spectrums closely converged the original ones.

In this thesis, the theory developed by Kayran is investigated by using Burg's estimation methods. Simulation programs for various ordered filters and have been developed to verify the theory. All the results obtained are compared by the results obtained in Türe's study. The comparisons have been made made in spectrum and contour plot bases. It was shown that the estimation by Burg's method has priority over the sample autocorrelation method -as expected- for shorter data records. As the data record gets longer, both of the methods prove to give sufficient results.

Two dimensional orthogonal lattice filters are a natural extension of the one dimensional lattice parameter theory. The method offers a complete solution for the Levinson-type algorithm in matrix form to compute the prediction error filter coefficients using lattice parameters from the given two dimensional augmented normal equation. The proposed theory can be used for the quarter plane and half plane models.

Using two dimensional orthogonal lattice filters in two dimensional spectrum estimation can be classified high resolution and parametric method. For the short data records, this method gives better results than the classical periodogram or FFT-based methods. If the data size extends, then both methods get better but in this situation data may not remain stationary.

In the 2-D AR modeling, a stationary random field $y(k_1, k_2)$ is predicted by a linear combination of its neighboring samples. The procedure starts with creating the AR data field according to the prediction region mask which may be quarter-plane or asymmetric half-plane. The 2-D AR data field can be considered as a one dimensional array if the appropriate indexing is chosen. Depending on the indexing specified, for instance the first quadrant backward prediction error filter corresponds to the forward prediction error filters in the second or fourth quadrants.

Let us show the 2-D AR data in the indexed form as;

$$y_{0,N}(k_1, k_2) = [y((k_1, k_2) - 0) \ y((k_1, k_2) - 1) \dots \dots \dots y((k_1, k_2) - N)]$$

where the subscripts 0 and N denote the first and the last elements in the array respectively. The 2-D non-Toeplitz, symmetric correlation matrix;

$$R_{0,N} = E[y_{0,N}(k_1, k_2) \ y_{0,N}^T(k_1, k_2)]$$

without taking the inverse of the correlation matrix, and using the partitioned form, can be written as shown below:

$$R_{0,m} = \begin{bmatrix} R_{0,m-1} & r_{m,0} \\ r_{m,0}^T & r_{m,m} \end{bmatrix} \begin{bmatrix} r_{0,0} & r_{0,m}^T \\ r_{0,m} & R_{1,m} \end{bmatrix}$$

An efficient method for factoring the symmetric 2-D correlation matrix is called Choleskey factorization and is commonly used in lattice form algorithms.

The forward prediction error associated with the prediction of the 0-th sample from the previous m samples within the prediction region can be defined in the compact form shown below:

$$f_0^{(m)}(k_1, k_2) = a_0^{(m)T} y_{0,m}(k_1, k_2)$$

where

$$a_0^{(m)} = [1 \ a_0^{(m)}(1) \dots \dots \dots a_0^{(m)}(m)]^T$$

and

$$y_{0,m}(k_1, k_2) = [y((k_1, k_2) - 0) \ y((k_1, k_2) - 1) \dots y((k_1, k_2) - m)]^T$$

where the notation $y((k_1, k_2) - i)$ denotes the i -th element behind $y(k_1, k_2)$ and the subscripts 0 and m denote the first and the last elements in the array respectively.

The backward prediction error associated with the prediction of the m -th sample (last element), $y((k_1, k_2) - m)$, from the m samples prior to it in the prediction region is defined by:

$$b_m^{(m)}(k_1, k_2) = g_m^{(m)T} y_{0,m}(k_1, k_2)$$

where

$$g_m^{(m)} = [g_m^{(m)}(m) \ g_m^{(m)}(m-1) \dots g_m^{(m)}(1) \ 1]$$

In the Levinson order-update recursions, one can find general expressions for lattice parameters, forward and backward prediction error fields and error powers in more compact form. For $p=1, 2, \dots, m$ and $n=1, 2, \dots, p$, lattice parameters can be written as;

$$\Gamma_{f_{p-n}}^{(n)} = -\frac{\Delta_{b_p}^{(n-1)}}{E_{f_{p-n}}^{(n-1)}} ; \quad \Gamma_{b_p}^{(n)} = -\frac{\Delta_{f_{p-n}}^{(n-1)}}{E_{b_p}^{(n-1)}}$$

and the minimum mean square errors are given by

$$E_{f_0}^{(m)} = E[f_0^{(m)2}(k_1, k_2)] = E_{f_0}^{(m-1)}(1 - \Gamma_{f_0}^{(m)} \Gamma_{b_m}^{(m)})$$

and

$$E_{b_m}^{(m)} = E[b_m^{(m)2}(k_1, k_2)] = E_{b_m}^{(m-1)}(1 - \Gamma_{f_0}^{(m)} \Gamma_{b_m}^{(m)})$$

It is known that, $\Delta_{b_p}^{(n-1)} = \Delta_{f_{p-n}}^{(n-1)}$ and it can be interpreted as a cross-correlation between the forward and backward prediction errors,

$$\Delta_{f_0 b_m}^{(m-1)} = E[f_0^{(m-1)}(k_1, k_2) b_m^{(m-1)}(k_1, k_2)]$$

The error propagation equations or general form of the orthogonal 2-D lattice filters is given by;

$$\begin{bmatrix} f_{p-n}^{(n)}(k_1, k_2) \\ b_p^{(n)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_p}^{(n)} \\ \Gamma_{f_{p-n}}^{(n)} & 1 \end{bmatrix} \begin{bmatrix} f_{p-n}^{(n-1)}(k_1, k_2) \\ b_p^{(n-1)}(k_1, k_2) \end{bmatrix}$$

$p=1, 2, \dots, m$; $n=1, 2, \dots, p$, and starting with

$$f_p^{(0)}(k_1, k_2) = b_p^{(0)}(k_1, k_2) = y((k_1, k_2) - p)$$

for $p=0, 1, \dots, m$, algorithm starts from the 0-th order and continues up to the m -th order. The prediction error powers can be written as;

$$\begin{bmatrix} E_{f_{p-n}}^{(n)} \\ E_{b_p}^{(n)} \end{bmatrix} = \begin{bmatrix} 1 & -\Gamma_{b_p}^{(n)^2} \\ -\Gamma_{f_{p-n}}^{(n)^2} & 1 \end{bmatrix} \begin{bmatrix} E_{f_{p-n}}^{(n-1)} \\ E_{b_p}^{(n-1)} \end{bmatrix}$$

In this thesis, an example has been given in order to explain the outline of the theory by means of a first quadrant support second order quarter - plane model. In addition to this example, simulation programs that verify the theory has been presented in appendix-A.

Synthesis model and the stability conditions has also been investigated. Since the lattice parameter stages are in tandem in the synthesis model, a sufficient condition for overall stability of the synthesis lattice model is that each stage should be stable. The realization of stability related to each stage has been shown by the help of Marzetta's [13] theorem.

The proposed 2-D lattice structures are amenable to systolic implementations. This is quite significant as the processing of the 2-D data fields such as images in real time require high data rates.

The simplicity of the algorithm is the main attractive feature and the only requirement is to select an ordering scheme with two types of shifts (vertical or horizontal) in the prediction support region. As a result of this, the first stages are 1-D lattice filters.

As the lattice structures form orthogonal bases, linear adaptive algorithms such as least mean-square (LMS) and recursive least - squares (RLS) can be applied to solve for 2-D system parameters. It is anticipated that the orthogonality property of the structure can be utilized to derive 2-D lattice autoregressive-moving average (ARMA) models, and to solve the 2-D joint-process estimation problem.

BÖLÜM 1.

GİRİŞ

Çok boyutlu işaret işleme günümüzde bir çok konuda uygulama alanı bulduğundan hızlı bir gelişme içindedir [1]. Görüntü işleme [10], jeofizik araştırmaları, hedef izleme ve görsel veri haberleşmesi gibi konularda uygulama alanı bulduğu için, bu alandaki teorik gelişmeler hızlanmıştır.

İlk olarak Marzetta [11] iki boyutlu alanları yansıma katsayıları ile modellemiş ve minimum fazlı sayısal süzgeçleri geliştirmiştir. Daha sonra PARKER ve KAYRAN tarafından iki boyutlu çeyrek düzlem kafes süzgeçleri sunulmuştur [13]. Sunulan bu çalışmada her bir çeyrek düzlem için dört hata alanı kavramı bulunmaktadır. Fakat önerilen modelin diklik özelliği taşımaması süzgecin yapısal kararlılığı ve optimum süzgeç olması açısından bir eksikliktir [1], [9]. Bununla birlikte bu yaklaşımla başarılı sonuçlar elde edilmiştir.

Ertüzün [3] tarafından sunulan kafes süzgecinde, ilk kademede dört hata alanı daha sonraki kademede ek iki hata alanı bulunmaktadır. Bu modelin, bilişim açısından maksimum entropiye yakınsadığı gösterilmiştir. Entropideki artış doğal olarak özbağlanımlı veri alanlarının daha güvenilir bir şekilde modellenmesini sağlayacaktır.

Bir boyutlu çok kanallı öz bağlanımlı model ile tek kanal çeyrek düzlem destekli iki boyutlu model arasındaki ilişki Therrien [17] tarafından sunulmuştur. Buna göre dört çeyrek düzleme ilişkin parametrelerin aynı anda hesaplanması mümkün olmaktadır. Bu yöntem çok kanallı iki boyutlu modeller için genelleştirilmiş ve çeşitli frekans görüngesi kestirim problemlerine uygulanmıştır.

LENK ve PARKER [9], [10] bilinen Levinson ve Schur algoritmasını genişleterek durağan rastgele alanların modellenmesi için kullanmışlardır.

KAYRAN [7], rastgele alanların özbağımlı modellenmesi için iki boyutlu dik kafes yapılarını sunmuştur. Elde edilen sonuçlar herhangi iki boyutlu bir gözleme uygulanabilecek niteliktedir. Yöntem, verilen iki boyutlu genişletilmiş normal denklemden, kafes parametrelerini kullanarak kestirim yarılgı süzgeç katsayılarını hesaplamak için Levinson tipi algoritmanın tam çözümünü ortaya koymaktadır. Levinson türü özyinelemenin elde edilebilmesi için Burg matris çözümü kullanılmıştır. Bu model çeyrek düzlem ve asimetrik yarı düzleme uygulanabilmektedir. KAYRAN [7] tarafından sunulan bu makalede rastgele alana ait kovaryans matrisinin verildiği kabul edilerek örnek çözüm de yapılmıştır. Ayrıca modelin dikliği ve kararlılığı gösterilmiştir.

Türe [18], Kayran [7] tarafından geliştirilen teoriyi, kısmi özilişki yöntemini kullanarak incelemiş ve simülasyonları sonucunda başarılı sonuçlar elde etmiştir. Ancak veri boyutuna getirilen sınırlama ile, yöntemin verimliliğinin azaldığı gözlenmiştir.

Bu çalışmada KAYRAN [7] tarafından geliştirilen modelin birinci, ikinci ve üçüncü derecelerden çeyrek düzlem ve simetrik olmayan yarı düzlem için doğruluğu simülasyonla ispatlanmıştır. Kestirilen model parametrelerine ilişkin frekans spektrumları, gerçek spektrumlar ve Yule-Walker yöntemi ile kestirilen spektrumlar ile karşılaştırılmıştır. Elde edilen tüm sonuçlar Ek -A ve Ek-B 'de sunulmuştur. Çalışma süresince izlenen çözüm aşamaları şöyledir.

- 1- İki boyutlu özbağımlı veri alanı üretilir.
- 2- Çapraz ilişkiler hesaplanır.
- 3- İleri yönde yansıma katsayıları bulunur.
- 4- Geri yönde yansıma katsayıları bulunur.

- 5- İleri yönde hata alanları hesaplanır.
- 6- Geri yönde hata alanları hesaplanır.
- 7- İleri yönde katsayı ağırlıkları hesaplanır.
- 8- Geri yönde katsayı ağırlıklarının hesaplanır.
- 9- Bir sonraki kafes bölmesine ilişkin hesaplamalar için 2. adıma dönülür.



BÖLÜM 2 .

BİR BOYUTLU KAFES (LATTICE) SÜZGEÇLERİ

2.1 Sonlu Geçmişe Dayalı Doğrusal Kestirim: Levinson-Durbin Özyinelemesi

Levinson (1947) ve bağımsız olarak da Durbin (1960) tarafından gerçekleştirilen özyinelemeli yapıya sahip algoritma, özilişki matrisi $R_{(0,N)}$ 'in Toeplitz olma özelliğinden yararlanır. Temel olarak, bu süreçte $(n-1)$. dereceden genişletilmiş normal denklemin çözümleri kullanılarak, n . dereceden kestirimcinin katsayıları hesaplanır. Süzgeç derecesi $n=1,2,...,m$ olmak üzere, m ulaşılmak istenen en son dereceyi belirtir. m büyüdükçe, yararlanılacak geçmişe a_i bilişim miktarı arttığından, ortalama karesel hatanın küçülerek, $y(n)$ kestiriminin daha iyi olması beklenir.

Özilişki işlevi $R(k) = E[y(n+k) y(n)]$ olan durağan bir $y(n)$ zaman serisi göz önüne alınsın. Verilen herhangi bir m değeri için, aşağıda verilen biçimde en iyi çözümü verecek m .dereceden doğrusal bir kestirimci araştırılmalıdır.

$$\hat{y}(k) = -[a_1 y(k-1) + a_2 y(k-2) + + a_m y(k-m)] \quad (2.1.1)$$

$a_1, a_2, ..., a_m$ kestirim katsayıları, ortalama karesel kestirim hatasını enaza indirecek şekilde seçilir.

$$\varepsilon = E[e(k)^2] = \min \quad (2.1.2)$$

Burada $f(k)$, kestirim hatasıdır.

$$f(k) = y(k) - \hat{y}(k) = y(k) + a_1 y(k-1) + a_2 y(k-2) + + a_m y(k-m) \quad (2.1.3)$$

(2.1.2) denkleminde her bir $a_i, i=1,2,...,m$ katsayısına göre türev alınır, diklik denklemleri elde edilir.

$$E[e(n) y(k-1)] = 0, i=1,2,m \quad (2.1.4)$$

(2.1.3) ifadesindeki $e(n)$, (2.3.4)'de yerine yerleştirilirse, hesaplanacak olan kestirimci katsayıları için m adet doğrusal denklem elde edilir.

$$\sum_{j=0}^m a_j E[y(k-j) y(k-i)] = \sum_{j=0}^m R(i-j) a_j = 0, \quad i=1,2,.....m \quad (2.1.5)$$

(2.1.4) 'deki koşullar kullanılarak,

$$\sigma_e^2 = \varepsilon = E[e(k)^2] = E[e(k) y(k)] = \sum_{i=0}^m R(i) a_i \quad (2.1.6)$$

minimize değeri de hesaplanır.

(2.1.5) ve (2.1.6) denklemleri $(m+1) \times (m+1)$ boyutlu bir matris denklemi şeklinde ifade edilebilir.

$$\begin{bmatrix} R(0) & R(1) & R(2) & \dots & R(m) \\ R(1) & R(0) & R(1) & \dots & R(m-1) \\ R(2) & R(1) & R(0) & \dots & R(m-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R(m) & R(m-1) & R(m-2) & \dots & R(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sigma_e^2 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (2.1.7)$$

(2.1.7) 'deki normal denklemleri doğrudan çözmek yerine, $m=1, m=2, m=3$. derecelerden en iyi doğrusal kestirimcileri belirlemek yoluna gidilir. İşte bu yaklaşımla, Levinson Algoritması ya da başka bir isimle, doğrusal kestirim süzgeçlerinin kafes yapısı ile gerçekleştirilmesi elde edilir. Yöntem aşama aşama incelendiğinde,

$$e^{(1)}(k) = y(k) + a_{11}y(k-1)$$

$$e^{(2)}(k)=y(k)+a_{21}y(k-1)+a_{22}y(k-2)$$

$$e^{(3)}(k)=y(k)+a_{31}y(k-1)+a_{32}y(k-2)+a_{33}y(k-3)$$

•
•
•

$$e^{(m)}(k)=y(k)+a_{m1}y(k-1)+a_{m2}y(k-2)+.....+a_{mm}(k-m) \quad (2.1.8)$$

olduğu görülür.

Burada $(1, a_{11})$, $(1, a_{21}, a_{22})$, $(1, a_{31}, a_{32}, a_{33})$, katsayıları sırasıyla $m=1, 2, 3, \dots$ dereceleri için en iyi kestirimcileri göstermektedir. Levinson Algoritması, bir sonraki kestirimciyi, bir önceki kestirimciden yararlanarak oluşturan iteratif bir algoritmadır. Bu algoritmada, istenilen dereceye ulaşıncaya kadar, daha düşük dereceli bütün optimal kestirimciler hesaplanır.

$$\Delta_m = \sum_{i=0}^m a_{mi} R(m+1-i) = R(m+1) + a_{m1}R(m) + a_{m2}R(m-1) + \dots + a_{mm}R(1) \quad (2.1.9)$$

İlk kez Burg tarafından kullanılan Levinson algoritmasının matris formu ile, ileri ve geri kestirimciler daha biçimsel bir şekilde gösterilmiştir. Levinson - Durbin özyinelemesi aşağıda gösterilen iki biçimden biri şeklinde ifade edilebilir.

1. İleri yönde bir kestirim hata süzgecinin katsayı ağırlık vektörüne ilişkin derece güncellemesi aşağıdaki yapılabilir:

$$a_m = \begin{bmatrix} a_{m-1} \\ 0 \end{bmatrix} + \Gamma_m \begin{bmatrix} 0 \\ a_{m-1}^* \end{bmatrix} \quad (2.1.10)$$

Burada Γ_m bir sabittir. Bu derece güncelleme işleminin skaler gösterimi ise şöyledir:

$$a_{m,k} = a_{m-1,k} + \Gamma_m a_{m-1,m-k}^* \quad k = 0, 1, \dots, m \quad (2.1.11)$$

$a_{m,k}$, m . dereceden ileri yönde bir kestirim hata süzgecinin k . katsayı ağırlığıdır. $a_{m-1,m-k}^*$ ise $(m-1)$. dereceden geri yönde bir kestirim hata süzgecinin k . katsayı ağırlığıdır. (2.1.11) bağıntısında,

$a_{m-1,0} = 1$ ve $a_{m-1,m} = 0$ dır.

2.Geri yönde bir kestirim hata süzgecinin katsayı ağırlık vektörüne ilişkin derece güncellemesi aşağıdaki gibi yapılabilir.

$$a_m^{B*} = \begin{bmatrix} 0 \\ a_{m-1}^{B*} \end{bmatrix} + \Gamma_m^* \begin{bmatrix} a_{m-1} \\ 0 \end{bmatrix} \quad (2.1.12)$$

Bu derece güncelleme işleminin skaler gösterimi ise şöyledir:

$$a_{m,m-k}^* = a_{m-1,m-k}^* + \Gamma_m^* a_{m-1,k} \quad k = 0, 1, \dots, m \quad (2.1.13)$$

$a_{m,m-k}^*$, m . dereceden geri yönde bir kestirim hata süzgecinin katsayı ağırlığıdır.

Levinson - Durbin özyinelemesi genellikle ileri yönde kestirim bağlamında ifade edilir. Geri yönde kestirime ilişkin bağıntılar, doğrudan ileri yöne ilişkin bağıntılardan elde edilmişlerdir.

2.2 Kafes Kestirimciler

Bir boyutlu işaretlerin kafes parametre modeli, doğrusal kestirim probleminin çözümüne bir alternatif oluşturur. $y(k)$ skaler, ayrık zamanlı, sıfır ortalamalı durağan bir süreç olsun. $\{y(k-i); i=1, \dots, m\}$ bu sürecin geçmiş değerlerini oluşturan uzaydır ve $y(k)$ değerini doğrusal olarak kestirmek için kullanılır. $y(k)$ 'nin kestirimi bu uzaya olan izdüşüm veya $y(k-i)$ 'lerin doğrusal kombinasyonudur.

$$\hat{y}(k) = \sum_{i=1}^m a^{(m)}(i) y(k-i) \quad (2.2.1)$$

Burada $\hat{y}(k)$, $y(k)$ 'nın kestirilen değeridir. $\{a^{(m)}(i), i=1\dots m\}$ ler kestirimci katsayılarıdır. Gerçek değer ile kestirilen değer arasındaki fark, ileri kestirim hatasıdır.

$$f^{(m)}(k) = y(k) - \hat{y}(k) = y(k) + \sum_{i=1}^m a^{(m)}(i) y(k-i) \quad (2.2.2)$$

Üst indis m , süzgeç derecesini belirtmektedir. Bir kafes kestirimciyi karakterize eden giriş çıkış bağıntıları çeşitli şekillerde elde edilebilir.

$$a_m = \begin{bmatrix} a_{m-1} \\ 0 \end{bmatrix} + \Gamma_m \begin{bmatrix} 0 \\ a_{m-1}^{B*} \end{bmatrix} \quad (2.2.3)$$

$$a_m^{B*} = \begin{bmatrix} 0 \\ a_{m-1}^{B*} \end{bmatrix} + \Gamma_m^* \begin{bmatrix} a_{m-1} \\ 0 \end{bmatrix} \quad (2.2.4)$$

$(m+1) \times 1$ boyutlu a_m ve $m \times 1$ boyutlu a_{m-1} vektörleri sırasıyla m . ve $m-1$. derecelerden ileri kestirim yanılğı süzgeçlerine karşılık düşerler. $(m+1) \times 1$ boyutlu a_m^{B*} ve $m \times 1$ boyutlu a_{m-1}^{B*} vektörleri ise sırasıyla m . ve $(m-1)$. derecelerden geriye kestirim yanılğı süzgeçlerine karşılık düşerler. Γ_m skaleri ise yansıma katsayısıdır.

İlk olarak katsayı girişleri $u(k), u(k-1), \dots, u(k-m)$ şeklinde gösterilen m . dereceden bir ileri yönde kestirim yanılğı süzgecini göz önüne alalım. Bu süzgecin $(m+1) \times 1$ boyutlu katsayı giriş vektörü olan $u_{m+1}(k)$ vektörünü aşağıdaki gibi ikiye ayırabiliriz [4].

$$u_{m+1}(k) = \begin{bmatrix} u_m(k) \\ u(k-m) \end{bmatrix} \quad (2.2.5)$$

yada,

$$u_{m+1}(k) = \begin{bmatrix} u(k) \\ u_m(k-1) \end{bmatrix} \quad (2.2.6)$$

a) (2.2.3) eşitliğinin sol tarafı için, $u_{m+1}(k)$ 'in a_m^H ile soldan çarpılması aşağıdaki sonucu verir.

$$f^{(m)}(k) = a_m^H u_{m+1}(k) \quad (2.2.7)$$

Burada $f_m(k)$, m. dereceden ileri yönde kestirim yanılğı süzgecinin çıkışındaki ileri kestirim hatasıdır.

b) (2.2.3) eşitliğinin sağ tarafındaki ilk terim için, $u_{m+1}(k)$ 'in (2.2.5) eşitliğiyle verilen şekli kullanılırsa aşağıdaki bağıntı yazılabilir.

$$\begin{aligned} \begin{bmatrix} a_{m-1}^H & 0 \end{bmatrix} u_{m+1}(k) &= \begin{bmatrix} a_{m-1}^H & 0 \end{bmatrix} \begin{bmatrix} u_m(k) \\ u(k-m) \end{bmatrix} \\ &= a_{m-1}^H u_m(k) \\ &= f_{m-1}(k) \end{aligned} \quad (2.2.8)$$

Burada $f_{m-1}(k)$, (m-1). dereceden ileri yönde kestirim yanılğı süzgecinin çıkışındaki ileri kestirim hatasıdır.

c) (2.2.3) eşitliğinin sağ tarafındaki ikinci matris terimi için, $u_{m+1}(k)$ 'in (2.2.6) eşitliği ile verilen şekli kullanılırsa aşağıdaki bağıntı yazılabilir.

$$\begin{aligned} \begin{bmatrix} 0 & a_{m-1}^{BT} \end{bmatrix} u_{m+1}(k) &= \begin{bmatrix} u(k) \\ u_m(k-1) \end{bmatrix} \\ &= a_{m-1}^{BT} u_m(k-1) \\ &= b_m(k-1) \end{aligned} \quad (2.2.9)$$

Burada $b_{m-1}(n-1)$, (m-1). dereceden geri yönde kestirim yanılğı süzgecinin çıkışındaki geciktirilmiş geri yönde kestirim hatasıdır. (2.2.7), (2.2.8) ve (2.2.9) eşitliklerinden elde edilen sonuçları birleştirerek aşağıdaki bağıntıyı yazabiliriz.

$$f^{(m)}(k) = f^{(m-1)}(k) + \Gamma^{(m)*} b^{(m-1)}(k-1) \quad (2.2.10)$$

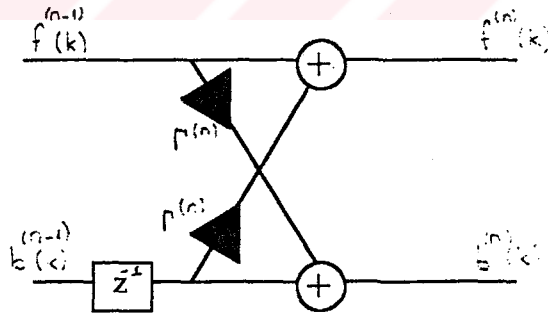
Aynı işlemler geri yönde kestirim yanılığ süzgeci için tekrarlandığında şu sonuç elde edilir.

$$b^{(m)}(k) = b^{(m-1)}(k-1) + \Gamma^{(m)} f^{(m-1)}(k) \quad (2.2.11)$$

(2.2.10) ve (2.2.11) eşitlikleri matris formunda yazılırsa aşağıdaki bağıntı elde edilir.

$$\begin{bmatrix} f^{(m)}(k) \\ b^{(m)}(k) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma^{(m)*} \\ \Gamma^{(m)} & 1 \end{bmatrix} \begin{bmatrix} f^{(m-1)}(k) \\ b^{(m-1)}(k-1) \end{bmatrix} \quad (2.2.12)$$

(2.2.12) eşitliği ile matematiksel tanımı verilen bir boyutlu kafes süzgeci şekil 2.1 'de gösterilmiştir.



Şekil 2.1 Bir boyutta tek adım, (n-1). kademe kafes süzgeci

2.3 Burg Yöntemi

Ölçülmüş bir sinyal bloğuna ait model parametrelerini belirlemek amacıyla en yaygın kullanılan yöntemler şunlardır:

1. Özilişki yada Yule - Walker Yöntemi
2. Kovaryans Yöntemi
3. Burg Yöntemi

Optimal ileri ve geri yönde kestirimciler için, teorik olarak en aza indirmeye ölçütleri aşağıdaki gibi verilir:

$$E[f^{(m)}(n)^2] = \min \quad E[b^{(m)}(n)^2] = \min \quad (2.3.1)$$

$f^{(m)}(n)$ ve $b^{(m)}(n)$ dizileri, $y(n)$ dizisinin $a = [1, a_{m1}, \dots, a_{mm}]^T$ ve tersi olan a^R süzgeçlerinden geçirilmesi ile elde edilmişlerdir. Daha açık bir biçimde ifade edilecek olursa:

$$\begin{aligned} f^{(m)}(n) &= y(n) + a_{m1} y(n-1) + a_{m2} y(n-2) + \dots + a_{mm} y(n-m) \\ b^{(m)}(n) &= y(n-m) + a_{m1} y(n-m+1) + a_{m2} y(n-m+2) + \dots + a_{mm} y(n) \end{aligned} \quad (2.3.2)$$

Yule - Walker Yönteminde, (2.3.1) bağıntısı ile verilen ensemble ortalamalarının yerini enaz kareler zaman ortalaması ölçütü alır. Enaza indirmeye çalışılan hata aşağıdaki gibidir:

$$\epsilon = \sum_{n=0}^{N+m-1} f^{(m)}(n)^2 = \min \quad (2.3.3)$$

$f^{(m)}(n)$ dizisi, $(m+1)$ uzunluklu kestirim hata dizisi $a = [1, a_{m1}, a_{m2}, \dots, a_{mm}]^T$ ile N uzunluklu veri dizisinin katlanmasıyla elde edilir. Bu nedenle $f^{(m)}(n)$ dizisinin

uzunluğu $N+(m+1)-1 = N+m$ dir. Böylece (2.3.3) bağıntısı ile verilen üst sınır doğrulanmış olur. Katlama işlemi ile yapılan, $y(n)$ veri bloğunun sıfırlar eklenmek yolu ile sağa ve sola genişletildiğini ve süzgecin bu genişletilmiş dizi üzerinden çalıştığını kabul etmektir.

Kovaryans Metodunda ise (2.3.1) bağıntısı ile verilen hata aşağıdaki bağıntı ile yer değiştirir:

$$\varepsilon = \sum_{n=p}^{N-1} f^{(m)}(n)^2 = \min \quad (2.3.4)$$

Özilişki yöntemi verimli kullanılabilen bir yöntem olup, kestirim hata süzgecinin minimum fazlı olması garantilenir. Bu yöntemin aksayan yönü, $y(n)$ veri dizisinin sağına ve soluna sıfırlar eklenmek yolu ile önpencereleme etkisinde kalmış olmasıdır. Bu durum, özellikle veri kayıtları N kısa olduğunda yöntemin duyarlılığını düşürür. $y(n)$ dizisini genişletmenin en uygun yolu -eğer genişletilmesi gerekiyorsa-, bu diziyi üreten sinyal modeli ile uyumlu bir çözüm bulmaktır. İşte Enbüyük Entropi Yöntemi [=Maximum Entropy Method (MEM)] adı ile de bilinen Burg yöntemi bu ihtiyaçtan kaynaklanmıştır. Burg Yönteminin enaza indirgeme ölçütü, hem ileri ve hem de geri yöndeki kestirim hatalarının toplamalarının karesini enaza indirmektir.

$$\varepsilon = \sum_{n=p}^{N-1} [f^{(m)}(n)^2 + b^{(m)}(n)^2] = \min \quad (2.3.5)$$

Eğer enaza indirgeme a_{m1} katsayılarına göre gerçekleştirilirse, kestirim hata süzgecinin minimum fazlı olmama olasılığı ortaya çıkar. Bunun için Burg iteratif bir yöntem teklif etmiştir: $(m-1)$. dereceden $[1, a_{m-1,1}, a_{m-1,2}, \dots, a_{m-1,m-1}]$ kestirim hata süzgecinin önceden belirlendiği kabul edilsin. m . dereceden kestirim hata süzgeci belirlenirken, Levinson özyinelemesinin kullanılabilmesi için $\Gamma^{(m)}$ yansıma katsayısının bilinmesi gerekmektedir. Minimum fazlı olma özelliğinin sağlanabilmesi için, $\Gamma^{(m)}$ yansıma katsayısının genliğinin birden küçük olması gerekmektedir. Bu amaçla (2.3.5) ile verilen bağıntının $\Gamma^{(m)}$ 'e göre türevi alınıp, sıfıra eşitlenirse:

$$\frac{\partial \varepsilon}{\partial \Gamma^{(m)}} = 2 \cdot \sum_{n=p}^{N-1} \left[f^{(m)}(n) \frac{\partial f^{(m)}(n)}{\partial \Gamma^{(m)}} + b^{(m)}(n) \frac{\partial b^{(m)}(n)}{\partial \Gamma^{(m)}} \right] = 0 \quad (2.3.6)$$

elde edilir.

Kafes yapısına ilişkin bağıntılar kullanılarak,

$$f^{(m)}(n) = f^{(m-1)}(n) - \Gamma^{(m)} b^{(m-1)}(n-1) \quad (2.3.7)$$

$$b^{(m)}(n) = b^{(m-1)}(n-1) - \Gamma^{(m)} f^{(m-1)}(n)$$

yazılabilir . Her iki bağıntı için de, $m \leq n \leq N-1$ dir.

Yukarıdaki bağıntılar, (2.3.6)'da yerine konulursa:

$$\sum_{n=p}^{N-1} [f^{(m)}(n) b^{(m-1)}(n-1) + b^{(m)}(n) f^{(m-1)}(n)] = 0 \quad (2.3.8)$$

$$\sum_{n=p}^{N-1} ((f^{(m-1)}(n) - \Gamma^{(m)} b^{(m-1)}(n-1)) b^{(m-1)}(n-1) +$$

$$(b^{(m-1)}(n-1) - \Gamma^{(m)} f^{(m-1)}(n)) f^{(m-1)}(n)) = 0 \quad (2.3.9)$$

Yukarıdaki bağıntılardan $\Gamma^{(m)}$ çözülürse:

$$\Gamma^{(m)} = \frac{2 \cdot \sum_{n=p}^{N-1} f^{(m-1)}(n) b^{(m-1)}(n-1)}{\sum_{n=p}^{N-1} [f^{(m-1)}(n)^2 + b^{(m-1)}(n-1)^2]} \quad (2.3.10)$$

elde edilir.

BÖLÜM 3

İKİ BOYUTLU KAFES SÜZGEÇLERİ

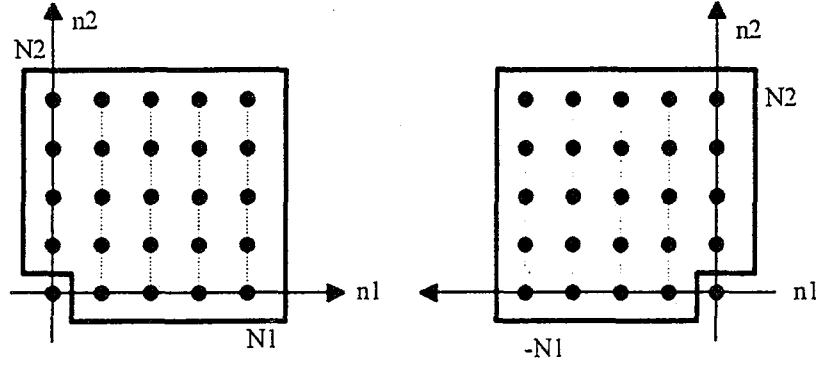
İki boyutlu doğrusal modellemede, durağan rastgele veri alanı $y(k_1, k_2)$ 'nin kestirimi komşu örnek noktaların doğrusal kombinasyonu ile olur. Kestirim bölgesine bağlı olarak çeşitli nedensel modeller bulunmaktadır. Çeyrek düzlem ve simetrik olmayan yarı düzlem modelleri en çok kullanılan modellerdir. İki boyutlu verilerde nedenselliğin engelleyici bir durum oluşturmaması, yarı nedensel ve nedensel olmayan modellerin de uygulama alanı bulmasına yol açmıştır. Geçmişe ait değerlerin uygun tanımlanmasıyla, nedensel modeller özyinelemeli bir yapı oluştururlar. İki boyutlu doğrusal kestirim aşağıdaki gibi ifade edilebilir.

$$\hat{y}(k_1, k_2) = - \sum_{(n_1, n_2) \in S} a(n_1, n_2) y(k_1 - n_1, k_2 - n_2) \quad (3.1)$$

Burada $a(n_1, n_2)$, katsayı ağırlıklarını ve S , $(0,0)$ noktasının içerilmediği kestirim destek bölgesini göstermektedir. Simetrik olmayan yarı düzlem ve çeyrek düzleme ait farklı modeller için kestirim destek bölgeleri Şekil 3.1 ve Şekil 3.2 de verilmiştir.

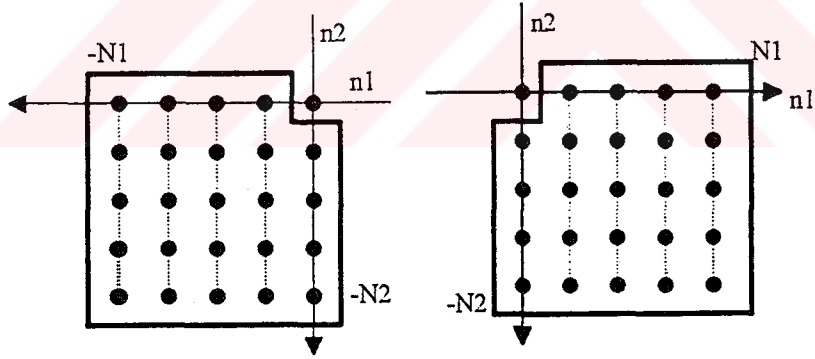
Kestirim hatası, bir boyuttaki duruma benzer şekilde aşağıdaki gibi tanımlanır:

$$e^{(N)}(k_1, k_2) = y(k_1, k_2) - \hat{y}(k_1, k_2) \quad (3.2)$$



a) Birinci çeyrek düzlem

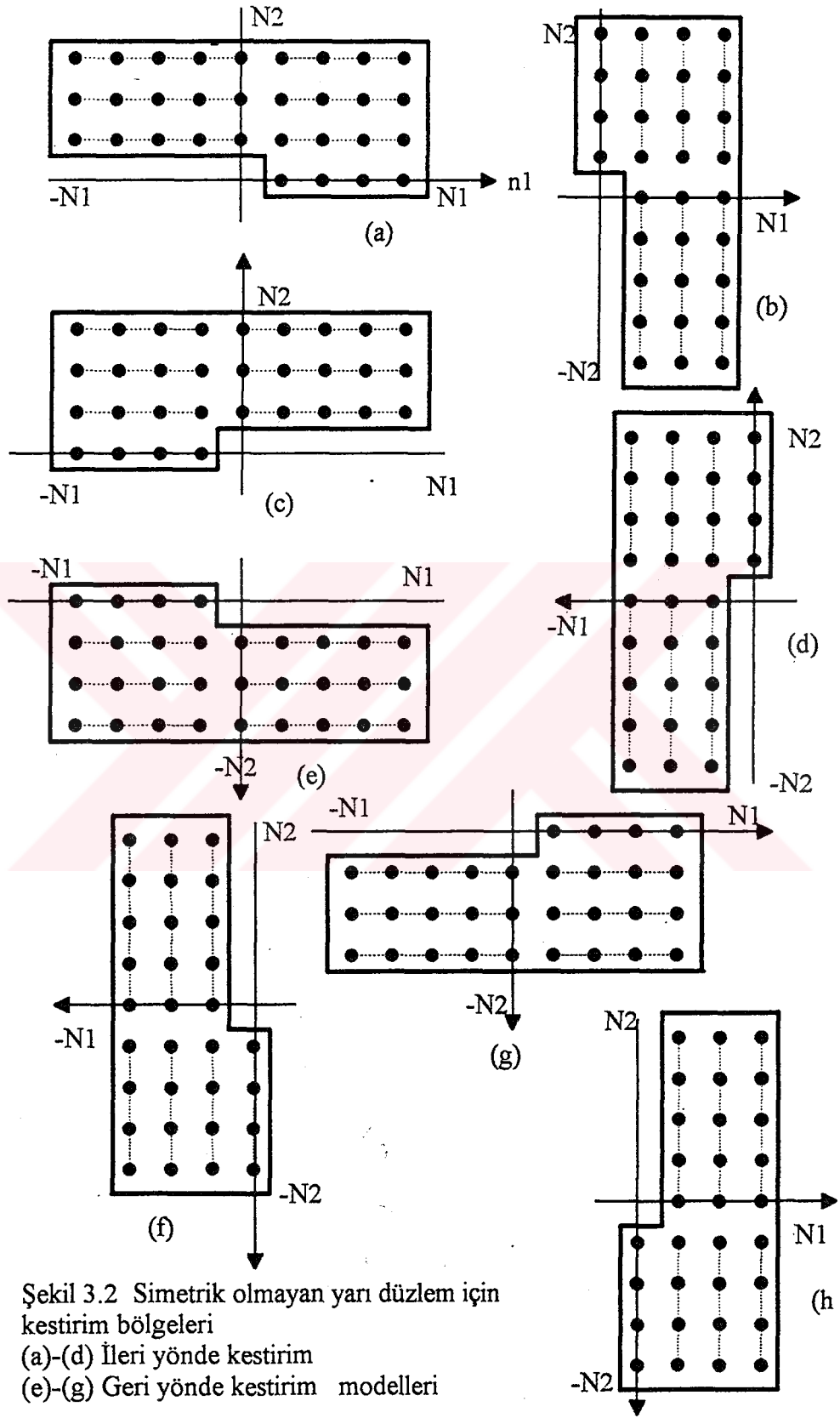
b) İkinci çeyrek düzlem

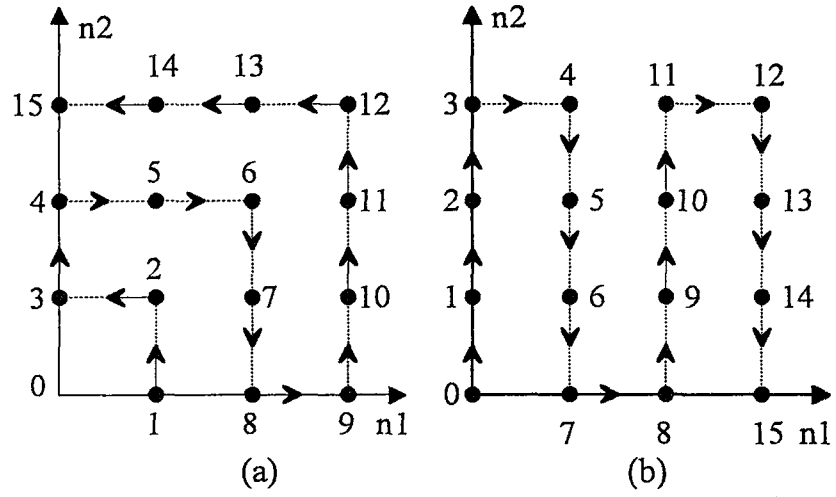


c) Üçüncü çeyrek düzlem

d) Dördüncü çeyrek düzlem

Şekil 3.1 Tek çeyrek düzleme ilişkin kestirim modelleri





Şekil 3.3 Kestirim bölgesinde dizilim düzenlemeleri
a, b) Tek çeyrek düzlem c) Simetrik olmayan yarı düzlem

İşlem kolaylığı açısından, kestirim bölgesi içindeki elemanları bir boyutlu diziler halinde sıralayabiliriz. Şekil 3.3 'de çeyrek düzlem ve simetrik olmayan yarı düzlem modelleri için olası dizilimler görülmektedir. Dizilimin elemanları 0 dan m 'e kadar numaralandırılmıştır.

$(m+1) \times 1$ boyutlu $a_0^{(m)}$ ve $g_m^{(m)}$ vektörleri, kendilerine karşılık gelen ileri ve geri yönde kestirimcilerin katsayı ağırlık vektörlerini gösterirler. İleri yönde kestirimci birinci çeyrek düzlem modeli olarak düşünülürken, kullanılan dizilime bağlı olarak geri yönde kestirimci ikinci yada dördüncü çeyrek düzlem

modeli olarak düşünülecektir. 0. örneğin m geçmiş değere sahip kestirim destek bölgesiyle kestirilmesiyle oluşacak ileri yönde kestirim hatası:

$$\begin{aligned} f_0^{(m)}(k_1, k_2) &= y(k_1, k_2) + \sum_{(n_1, n_2) = ((n_1, n_2) - 1)}^{((n_1, n_2) - m)} a_0^{(m)}(n_1, n_2) y(k_1 - n_1, k_2 - n_2) \\ &= \mathbf{a}_0^{(m)T} \mathbf{y}_{(0,m)}(k_1, k_2) \end{aligned} \quad (3.1.a)$$

Burada,

$$\mathbf{a}_0^{(m)} = [1 \ a_0^{(m)}(1) \dots \dots a_0^{(m)}(m)] \quad (3.1. b)$$

ve

$$\mathbf{y}_{(0,m)}(k_1, k_2) = [y((k_1, k_2) - 0) \ y((k_1, k_2) - 1) \dots \dots y((k_1, k_2) - m)] \quad (3.1. c)$$

olarak tanımlıdır.

$y((k_1, k_2) - i)$ gösterimi $y(k_1, k_2)$ 'den önceki i. elemanı göstermektedir. $y((k_1, k_2) - 0) = y(k_1, k_2)$ birinci elemandır. $f_0^{(m)}(k_1, k_2)$ 'nin z dönüşümünü alırsak:

$$F_0^{(m)}(z_1, z_2) = \mathbf{a}_0^{(m)T} \mathbf{O}_{0,m}(z_1, z_2) Y(z_1, z_2) \quad (3.2. a)$$

elde edilir. Dizilim vektörü,

$$\mathbf{O}_{0,m}(z_1, z_2) = [1 \ \delta_1(z_1, z_2) \dots \dots \delta_m(z_1, z_2)] \quad (3.2. b)$$

$$\delta_p(z_1, z_2) = Z [y((k_1, k_2) - p)] / Y(z_1, z_2) \quad (3.2. c)$$

bağıntıları ile verilir. (3.2. b) bağıntısı ile verilen $\delta_i(z_1, z_2)$ 'ler kestirim bölgesinde kullanılan örnek noktaların yerlerini göstermektedir. Bir boyutlu durumda, bu noktalar doğal bir sıralanmaya sahiptirler, yani: $\delta_i(z) = z^{-i}$, $i = 1, 2, \dots, N$

olarak belirlidir. Bununla birlikte, iki boyutlu kestirimde sıralanma değişik şekillerde yapılabilir. Eğer kestirim bölgesindeki elemanları sıralamak için yalnızca dikey ve yatay yönleri kullanırsak, aşağıdaki bağıntı elde edilir.

$$\frac{\delta_{i+1}(z_1, z_2)}{\delta_i(z_1, z_2)} = \begin{cases} z_2^{\pm 1} & , \text{düşey yönde} \\ z_1^{\pm 1} & , \text{yatay yönde} \end{cases} \quad (3.3)$$

İleri yöndeki kestirim hatasına benzer şekilde, $y((k_1, k_2) - m)$ veya m . örneğin m geçmiş değere sahip kestirim bölgesine göre oluşacak geri yönde kestirim hatası aşağıdaki şekilde tanımlanır.

$$\begin{aligned} b_m^{(m)}(k_1, k_2) &= y((k_1, k_2) - m) + \sum_{(n_1, n_2) = ((n_1, n_2) - 0)}^{((n_1, n_2) - m + 1)} g_m^{(m)}(n_1, n_2) y(k_1 - n_1, k_2 - n_2) \\ &= \mathbf{g}_m^{(m)T} \mathbf{y}_{0,m}(k_1, k_2) \end{aligned} \quad (3.4.a)$$

Burada,

$$\mathbf{g}^{(m)} = [g_m^{(m)}(m) \ g_m^{(m)}(m-1) \dots g_m^{(m)}(1) \ 1] \quad (3.4. b)$$

olarak verilmiştir. Dizilim vektörü de kullanılarak yukarıda verilen bağıntının z dönüşümü yazılırsa:

$$\mathbf{B}_m^{(m)}(z_1, z_2) = \mathbf{g}_m^{(m)T} \mathbf{O}_{o,m}(z_1, z_2) \mathbf{Y}(z_1, z_2) \quad (3.5)$$

bağıntısı elde edilir.

$(m-1)$. derece ileri ve geri yönde kestirim süzgeçlerine ilişkin katsayı ağırlık vektörleri sırasıyla $\mathbf{a}_0^{(m-1)}$ ve $\mathbf{g}_m^{(m-1)}$ olarak verilmiş olsun. Derece aşaması yöntemiyle m . derece katsayıları aşağıdaki gibi elde edilir.

$$\begin{aligned} \mathbf{a}_0^{(m)} &= \begin{bmatrix} \mathbf{a}_0^{(m-1)} \\ 0 \end{bmatrix} + \Gamma_{b_m}^{(m)} \begin{bmatrix} 0 \\ \mathbf{g}_m^{(m-1)} \end{bmatrix} \\ \mathbf{g}_m^{(m)} &= \begin{bmatrix} 0 \\ \mathbf{g}_m^{(m-1)} \end{bmatrix} + \Gamma_{f_0}^{(m)} \begin{bmatrix} \mathbf{a}_0^{(m-1)} \\ 0 \end{bmatrix} \end{aligned} \quad (3.6)$$

Burada $\Gamma_{f_0}^{(m)}$ ve $\Gamma_{b_m}^{(m)}$ sırasıyla m. dereceden ileri ve geri yönde kestirimcilerin yansıma katsayıları veya kısmi ilişki katsayılarıdır. (3.6) eşitliğinin her iki tarafını $(m+1) \times (m+1)$ boyutundaki özilişki matrisi ile çarpalım. Eşitliğin sol tarafı m. derece genişletilmiş normal denklemleri verir.

$$\mathbf{R}_{0,m} \mathbf{a}_0^{(m)} = \begin{bmatrix} \mathbf{E}_{f_0}^{(m)} \\ \mathbf{O}_m \end{bmatrix} \quad \text{ve} \quad \mathbf{R}_{0,m} \mathbf{g}_m^{(m)} = \begin{bmatrix} \mathbf{O}_m \\ \mathbf{E}_{b_m}^{(m)} \end{bmatrix} \quad (3.7)$$

$\mathbf{E}_{f_0}^{(m)}$ ve $\mathbf{E}_{b_m}^{(m)}$ sırasıyla ileri ve geri yönde kestirim hata güçleridir. (3.7) ifadesindeki simetrik ilişki matrisi aşağıdaki gibi bölümlendirilebilir.

$$\begin{aligned} \mathbf{R}_{0,m} &= E[\mathbf{y}_{0,m}(k_1, k_2) \mathbf{y}_{0,m}(k_1, k_2)^T] \\ &= \begin{bmatrix} \mathbf{R}_{0,m-1} & \mathbf{r}_{m,0} \\ \dots & \dots \\ \mathbf{r}_{m,0}^T & \mathbf{r}_{m,m} \end{bmatrix} = \begin{bmatrix} r_{0,0} & \mathbf{r}_{0,m}^T \\ \dots & \dots \\ \mathbf{r}_{0,m} & \mathbf{R}_{1,m} \end{bmatrix} \end{aligned} \quad (3.8.a)$$

Burada,

$$\mathbf{R}_{p,q} = E[\mathbf{y}_{p,q}(k_1, k_2) \mathbf{y}_{p,q}(k_1, k_2)^T] \quad (3.8. b)$$

$$\mathbf{r}_{0,q} = E[\mathbf{y}(k_1, k_2) \mathbf{y}_{1,q}(k_1, k_2)],$$

$$\mathbf{r}_{p,0} = E[\mathbf{y}(k_1, k_2) - \mathbf{p} \mathbf{y}_{0,p-1}(k_1, k_2)],$$

$$\mathbf{r}_{p,p} = E[\mathbf{y}^2((k_1, k_2) - \mathbf{p})] = r_{\infty} \quad (3.8.c)$$

olarak tanımlıdır ve $p < q$ olmak üzere giriş vektörü en genel olarak;

$$\mathbf{y}_{p,q}(k_1, k_2) = [y((k_1, k_2)-p) y((k_1, k_2)-p-1) \dots y((k_1, k_2)-q)]^T \quad (3.8. d)$$

biçiminde yazılır. (3.6.a) bağıntısının sağ tarafındaki ilk terim için,

$$\begin{aligned} \mathbf{R}_{0,m} \begin{bmatrix} \mathbf{a}_0^{(m-1)} \\ 0 \end{bmatrix} &= \begin{bmatrix} \mathbf{R}_{0,m-1} & \mathbf{r}_{m,0} \\ \dots & \dots \\ \mathbf{r}_{m,0}^T & r_{m,m} \end{bmatrix} \begin{bmatrix} \mathbf{a}_0^{(m-1)} \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_{0,m-1} \mathbf{a}_0^{(m-1)} \\ \mathbf{r}_{m,0}^T \mathbf{a}_0^{(m-1)} \end{bmatrix} \end{aligned} \quad (3.9)$$

elde edilir. $(m-1)$. dereceden ileri yönde kestirim hata süzgeci için genişletilmiş normal denklem:

$$\mathbf{R}_{0,m-1} \mathbf{a}_0^{(m-1)} = \begin{bmatrix} E_{f_0}^{(m-1)} \\ \mathbf{0}_{m-1} \end{bmatrix} \quad (3.10)$$

şeklinde yazılır. Burada $E_{f_0}^{(m-1)}$ ileri kestirim hata gücünü verirken, $\mathbf{0}_{m-1}$ ise sıfır (null) vektördür. (3.9) bağıntısı ile elde edilen ifade için $\Delta_{f_0}^{(m-1)}$ biçiminde bir skaler tanımlanırsa,

$$\Delta_{f_0}^{(m-1)} = \mathbf{r}_{m,0}^T \mathbf{a}_0^{(m-1)} = \sum_{i=0}^{m-1} \mathbf{a}_0^{(m-1)}(i) r_{m,0}(i) \quad (3.11)$$

(3.8.b) bağıntısı kullanılarak, $\mathbf{r}_{m,0}$ ilişki vektörünün p . elemanı aşağıdaki gibi yazılabilir.

$$r_{m,0}(p) = E[y((k_1, k_2) - m) y((k_1, k_2) - p)] \quad (3.12)$$

(3.10) ve (3.11) bağıntıları, (3.9) 'da yerine konularak,

$$\mathbf{R}_{0,m} \begin{bmatrix} \mathbf{a}_0^{(m-1)} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{E}_{f_0}^{(m-1)} \\ \mathbf{O}_{m-1} \\ \Delta_{f_0}^{(m-1)} \end{bmatrix} \quad (3.13)$$

(3.6.b) bağıntısının sağ tarafındaki ilk terim için,

$$\begin{aligned} \mathbf{R}_{0,m} \begin{bmatrix} 0 \\ \mathbf{g}_m^{(m-1)} \end{bmatrix} &= \begin{bmatrix} r_{00} & \mathbf{r}_{0,m}^T \\ \dots & \dots \\ \mathbf{r}_{0,m} & \mathbf{R}_{1,m} \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{g}_m^{(m-1)} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{r}_{0,m}^T \mathbf{g}_m^{(m-1)} \\ \mathbf{R}_{1,m} \mathbf{g}_m^{(m-1)} \end{bmatrix} \end{aligned} \quad (3.14)$$

Benzer şekilde, (m-1).dereceden geri yönde kestirim hata süzgeci için;

$$\mathbf{R}_{1,m} \mathbf{g}_m^{(m-1)} = \begin{bmatrix} \mathbf{O}_{m-1} \\ \mathbf{E}_{b_m}^{(m-1)} \end{bmatrix} \quad (3.15)$$

Ayrıca, $\mathbf{r}_{0,m}^T \mathbf{g}_m^{(m-1)}$ skaleri :

$$\begin{aligned} \Delta_{b_m}^{(m-1)} &= \mathbf{r}_{0,m}^T \mathbf{g}_m^{(m-1)} \\ &= \sum_{i=1}^m g_m^{(m-1)}(i) r_{0,m}(i) = \sum_{i=1}^m g_m^{(m-1)}(i) E[y(k_1, k_2) y((k_1, k_2) - i)] \end{aligned} \quad (3.16)$$

(3.15) ve (3.16) bağıntıları (3.14)'de yerine konularak, aşağıdaki eşitlik yazılabilir.

$$\mathbf{R}_{0,m} \begin{bmatrix} 0 \\ \mathbf{g}_m^{(m-1)} \end{bmatrix} = \begin{bmatrix} \Delta_{b_m}^{(m-1)} \\ \mathbf{O}_{m-1} \\ \mathbf{E}_{b_m}^{(m-1)} \end{bmatrix} \quad (3.17)$$

(3.13) ve (3.17) bağıntıları (3.6) 'nın sağ tarafında kullanılırsa,

$$\begin{aligned} \mathbf{R}_{0,m} \left(\begin{bmatrix} \mathbf{a}_0^{(m-1)} \\ 0 \end{bmatrix} + \Gamma_{b_m}^{(m)} \begin{bmatrix} 0 \\ \mathbf{g}_m^{(m-1)} \end{bmatrix} \right) &= \begin{bmatrix} E_{f_0}^{(m-1)} \\ \mathbf{O}_{m-1} \\ \Delta_{f_0}^{(m-1)} \end{bmatrix} + \Gamma_{b_m}^{(m)} \begin{bmatrix} \Delta_{b_m}^{(m-1)} \\ \mathbf{O}_{m-1} \\ E_{b_m}^{(m-1)} \end{bmatrix} \\ &= \begin{bmatrix} E_{f_0}^{(m)} \\ \mathbf{O}_m \end{bmatrix} \end{aligned} \quad (3.18.a)$$

ve

$$\begin{aligned} \mathbf{R}_{0,m} \left(\begin{bmatrix} 0 \\ \mathbf{g}_m^{(m-1)} \end{bmatrix} + \Gamma_{f_0}^{(m)} \begin{bmatrix} \mathbf{a}_0^{(m-1)} \\ 0 \end{bmatrix} \right) &= \begin{bmatrix} \Delta_{b_m}^{(m-1)} \\ \mathbf{O}_{m-1} \\ E_{b_m}^{(m-1)} \end{bmatrix} + \Gamma_{f_0}^{(m)} \begin{bmatrix} E_{f_0}^{(m-1)} \\ \mathbf{O}_{m-1} \\ \Delta_{f_0}^{(m-1)} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{O}_m \\ E_{b_m}^{(m)} \end{bmatrix} \end{aligned} \quad (3.18.b)$$

olarak yazılır. (3.18) bağıntıları kullanılarak, ileri ve geri yönde yansıma katsayıları aşağıdaki gibi elde edilir:

$$\Gamma_{f_0}^{(m)} = -\frac{\Delta_{b_m}^{(m-1)}}{E_{f_0}^{(m-1)}} \quad , \quad \Gamma_{b_m}^{(m)} = -\frac{\Delta_{f_0}^{(m-1)}}{E_{b_m}^{(m-1)}} \quad (3.19)$$

Minimum ortalama karesel hatalar ise şu şekildedir.

$$E_{f_0}^{(m)} = E \left[f_0^{(m)2}(k_1, k_2) \right] = E_{f_0}^{(m-1)} (1 - \Gamma_{f_0}^{(m)} \Gamma_{b_m}^{(m)}) \quad (3.20.a)$$

ve

$$E_{b_m}^{(m)} = E \left[b_m^{(m)2}(k_1, k_2) \right] = E_{b_m}^{(m-1)} (1 - \Gamma_{f_0}^{(m)} \Gamma_{b_m}^{(m)}) \quad (3.20.b)$$

Yukarıda tanımlanan Δ skalerleri, ileri ve geri kestirim hataları arasındaki çapraz ilişki olarak yorumlanabilir. İleri ve geri kestirimciler için çapraz ilişkiler birbirine eşit olup, aşağıdaki bağıntı yazılabilir.

$$\Delta_{f_0}^{(m-1)} = \Delta_{b_m}^{(m-1)} = \Delta_{f_0 b_m}^{(m-1)} \quad (3.21)$$

$$\Delta_{f_0 b_m}^{(m-1)} = E[f_0^{(m-1)}(k_1, k_2) b_m^{(m-1)}(k_1, k_2)] \quad (3.22)$$

(3.19), (3.20) ve (3.21) bağıntıları kullanılarak, yansıma katsayılarına ilişkin aşağıdaki bağıntılar elde edilir.

$$\Gamma_{f_0}^{(m)} = -\frac{\Delta_{f_0 b_m}^{(m-1)}}{E_{f_0}^{(m-1)}}, \quad \Gamma_{b_m}^{(m)} = -\frac{\Delta_{f_0 b_m}^{(m-1)}}{E_{b_m}^{(m-1)}} \quad (3.23)$$

İleri ve geri yönde kestirim hata güçleri ise aşağıdaki matris biçiminde ifade edilebilir.

$$\begin{bmatrix} E_{f_0}^{(m)} \\ E_{b_m}^{(m)} \end{bmatrix} = \begin{bmatrix} 1 & -\Gamma_{b_m}^{(m)^2} \\ -\Gamma_{f_0}^{(m)^2} & 1 \end{bmatrix} \begin{bmatrix} E_{f_0}^{(m-1)} \\ E_{b_m}^{(m-1)} \end{bmatrix} \quad (3.24)$$

İleri ve geri yönde kestirim hata süzgecine ilişkin işlemler aşağıdaki gibi tek bir yapıda birleştirilebilir.

$$\begin{bmatrix} f_0^{(m)}(k_1, k_2) \\ b_m^{(m)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_m}^{(m)} \\ \Gamma_{f_0}^{(m)} & 1 \end{bmatrix} \begin{bmatrix} f_0^{(m-1)}(k_1, k_2) \\ b_m^{(m-1)}(k_1, k_2) \end{bmatrix} \quad (3.25)$$

(2.9) ve (2.14) bağıntıları ile verilen $(m-1)$.dereceden ileri ve geri yönde kestirim hata süzgeçlerine ilişkin katsayı ağırlıkları kullanılarak, $a_0^{(m-1)}$ ve $g_m^{(m-1)}$ katsayı ağırlık vektörlerine ilişkin derece güncellemesi yapılabilir. $R_{0,m-1}$ ve $R_{1,m}$ matrisleri aşağıdaki gibi bölümlendirilebilir:

$$R_{0,m-1} = \begin{bmatrix} R_{0,m-2} & r_{m-1,0} \\ \dots & \dots \\ r_{m-1,0}^T & r_{m-1,m-1} \end{bmatrix} = \begin{bmatrix} r_{0,0} & r_{0,m-1}^T \\ \dots & \dots \\ r_{0,m-1} & R_{1,m-1} \end{bmatrix} \quad (3.26)$$

$$\mathbf{R}_{1,m} = \begin{bmatrix} \mathbf{R}_{1,m-1} & \mathbf{r}_{m,m-1} \\ \dots & \dots \\ \mathbf{r}_{m,m-1}^T & \mathbf{r}_{m,m} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{1,1} & \mathbf{r}_{1,m}^T \\ \dots & \dots \\ \mathbf{r}_{1,m} & \mathbf{R}_{2,m} \end{bmatrix} \quad (3.27)$$

$\mathbf{R}_{p,q}$, $\mathbf{r}_{0,q}$, $\mathbf{r}_{p,0}$ ve $\mathbf{r}_{pp}$ değerlerine ilişkin tanımlar, (3.8) bağıntıları ile verilmiştir. $\mathbf{r}_{m,m-j}$ ve $\mathbf{r}_{i,m}$ değerleri ise aşağıdaki şekilde tanımlıdır.

$$\mathbf{r}_{m,m-j} = E[y((k_1, k_2) - m) y_{j,m-j}(k_1, k_2)] \quad , j = 1, 2, \dots, m-1 \quad (3.28.a)$$

$$\mathbf{r}_{i,m} = E[y((k_1, k_2) - i) y_{i+1,m}(k_1, k_2)] \quad , i = 1, 2, \dots, m-1 \quad (3.28.b)$$

(3.26) ve (3.27) matris eşitlikleri kullanılarak, $\mathbf{a}_0^{(m-1)}$ ve $\mathbf{g}_m^{(m-1)}$ katsayı ağırlıklarının derece güncellemesi yapılabilir.

$$\mathbf{a}_0^{(m-1)} = \begin{bmatrix} \mathbf{a}_0^{(m-2)} \\ 0 \end{bmatrix} + \Gamma_{b_{m-1}}^{(m-1)} \begin{bmatrix} 0 \\ \mathbf{g}_{m-1}^{(m-2)} \end{bmatrix} \quad (3.29.a)$$

$$\mathbf{g}_m^{(m-1)} = \begin{bmatrix} 0 \\ \mathbf{g}_m^{(m-2)} \end{bmatrix} + \Gamma_{f_1}^{(m-1)} \begin{bmatrix} \mathbf{a}_1^{(m-2)} \\ 0 \end{bmatrix} \quad (3.29.b)$$

Bu aşamada yeni karşılaşılan $\mathbf{f}_1^{(m-1)}(k_1, k_2)$ ileri yönde kestirim hatası ve $\mathbf{b}_{m-1}^{(m-1)}(k_1, k_2)$ geri yönde kestirim hatası, aşağıdaki bağıntılarla tanımlıdır.

$$\begin{aligned} \mathbf{f}_1^{(m-1)}(k_1, k_2) &= y((k_1, k_2) - 1) + \sum_{(n_1, n_2)=-}^{((k_1, k_2)-m)} \sum_{((k_1, k_2)-2)} \mathbf{a}_1^{(m-1)}(n_1, n_2) y(k_1 - n_1, k_2 - n_2) \\ &= \mathbf{a}_1^{(m-1)T} \mathbf{y}_{1,m}(k_1, k_2) \end{aligned} \quad (3.30.a)$$

$$\begin{aligned} \mathbf{b}_{m-1}^{(m-1)}(k_1, k_2) &= y((k_1, k_2) - m + 1) + \sum_{(n_1, n_2)=-}^{((k_1, k_2) - m + 2)} \sum_{((k_1, k_2)-0)} \mathbf{g}_{m-1}^{(m-1)}(n_1, n_2) y(k_1 - n_1, k_2 - n_2) \\ &= \mathbf{g}_{m-1}^{(m-1)T} \mathbf{y}_{0,m-1}(k_1, k_2) \end{aligned} \quad (3.30.b)$$

(3.30) bağıntıları ile tanımlanan, ileri ve geri yönde kestirim yanılğı süzgeçleri için katsayı ağırlık vektörleri şöyle güncellenebilir:

$$\mathbf{a}_1^{(m-1)} = \begin{bmatrix} \mathbf{a}_1^{(m-2)} \\ 0 \end{bmatrix} + \Gamma_{b_m}^{(m-1)} \begin{bmatrix} 0 \\ \mathbf{g}_m^{(m-2)} \end{bmatrix} \quad (3.31.a)$$

$$\mathbf{g}_{m-1}^{(m-1)} = \begin{bmatrix} 0 \\ \mathbf{g}_{m-1}^{(m-2)} \end{bmatrix} + \Gamma_{f_0}^{(m-1)} \begin{bmatrix} \mathbf{a}_1^{(m-2)} \\ 0 \end{bmatrix} \quad (3.31.b)$$

Bir önceki adımda izlenen yollar kullanılarak, çapraz ilişkiler için aşağıdaki bağıntı elde edilir.

$$\Delta_{f_0 b_{m-1}}^{(m-2)} = \Delta_{b_{m-1}}^{(m-2)} = \Delta_{f_0}^{(m-2)} = E[f_0^{(m-2)}(k_1, k_2) b_{m-1}^{(m-2)}(k_1, k_2)] \quad (3.32)$$

Yansıma katsayıları için,

$$\Gamma_{f_0}^{(m-1)} = -\frac{\Delta_{f_0 b_{m-1}}^{(m-2)}}{E_{f_0}^{(m-2)}}, \quad \Gamma_{b_{m-1}}^{(m-1)} = -\frac{\Delta_{f_0 b_{m-1}}^{(m-2)}}{E_{b_{m-1}}^{(m-2)}} \quad (3.33)$$

ve kestirim hata güçleri için,

$$\begin{bmatrix} E_{f_0}^{(m-1)} \\ E_{b_{m-1}}^{(m-1)} \end{bmatrix} = \begin{bmatrix} 1 & -\Gamma_{b_{m-1}}^{(m-1)^2} \\ -\Gamma_{f_0}^{(m-1)^2} & 1 \end{bmatrix} \begin{bmatrix} E_{f_0}^{(m-2)} \\ E_{b_{m-1}}^{(m-2)} \end{bmatrix} \quad (3.34)$$

yazılır.

Bu sürece devam edilirse, (m-i) .dereceden ileri ve geri yönde kestirim hata süzgeçlerinin çözümü için, (i+1) adet kafes yapısına ihtiyaç duyulacağı ortaya çıkar.

$$\begin{aligned}
 \begin{bmatrix} f_0^{(m-i)}(k_1, k_2) \\ b_{m-i}^{(m-i)}(k_1, k_2) \end{bmatrix} &= \begin{bmatrix} 1 & \Gamma_{b_{m-i}}^{(m-i)} \\ \Gamma_{f_0}^{(m-i)} & 1 \end{bmatrix} \begin{bmatrix} f_0^{(m-i-1)}(k_1, k_2) \\ b_{m-i}^{(m-i-1)}(k_1, k_2) \end{bmatrix} \\
 \begin{bmatrix} f_1^{(m-i)}(k_1, k_2) \\ b_{m-i+1}^{(m-i)}(k_1, k_2) \end{bmatrix} &= \begin{bmatrix} 1 & \Gamma_{b_{m-i+1}}^{(m-i)} \\ \Gamma_{f_1}^{(m-i)} & 1 \end{bmatrix} \begin{bmatrix} f_1^{(m-i-1)}(k_1, k_2) \\ b_{m-i+1}^{(m-i-1)}(k_1, k_2) \end{bmatrix} \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 \begin{bmatrix} f_i^{(m-i)}(k_1, k_2) \\ b_m^{(m-i)}(k_1, k_2) \end{bmatrix} &= \begin{bmatrix} 1 & \Gamma_{b_m}^{(m-i)} \\ \Gamma_{f_i}^{(m-i)} & 1 \end{bmatrix} \begin{bmatrix} f_i^{(m-i-1)}(k_1, k_2) \\ b_m^{(m-i-1)}(k_1, k_2) \end{bmatrix}
 \end{aligned} \tag{3.35}$$

(3.35) eşitlikleri kullanılarak, hata yayılım denklemleri en kapalı şekli ile aşağıdaki gibi yazılır.

$$\begin{bmatrix} f_{p-n}^{(n)}(k_1, k_2) \\ b_p^{(n)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_p}^{(n)} \\ \Gamma_{f_{p-n}}^{(n)} & 1 \end{bmatrix} \begin{bmatrix} f_{p-n}^{(n-1)}(k_1, k_2) \\ b_p^{(n-1)}(k_1, k_2) \end{bmatrix} \tag{3.36.a}$$

$p=1,2,\dots,M$ ve $n=1,2,\dots,p$ dir.

İlk koşullar,

$$f_0^{(p)}(k_1, k_2) = b_p^{(0)}(k_1, k_2) = y((k_1, k_2) - p) \quad , p = 0, 1, 2, \dots, M \tag{3.36.b}$$

eşitliği ile belirlidir.

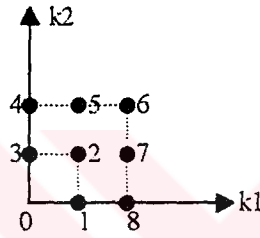
Algoritma 0. dereceden başlar ve m. dereceye kadar devam eder. Kestirim hata güçleri yeniden yazılırsa:

$$\begin{bmatrix} E_{f_{p-n}}^{(n)} \\ E_{b_p}^{(n)} \end{bmatrix} = \begin{bmatrix} 1 & -\Gamma_{b_p}^{(n)^2} \\ -\Gamma_{f_{p-n}}^{(n)^2} & 1 \end{bmatrix} \begin{bmatrix} E_{f_{p-n}}^{(n-1)} \\ E_{b_p}^{(n-1)} \end{bmatrix} \tag{3.37}$$

elde edilir.

ÖRNEK :

İkinci dereceden, çeyrek düzlemde sekiz destek noktalı, iki boyutlu bir kafes süzgecinin Burg Yöntemi ile çözümü için aşağıdaki aşamalar izlenir. Dördüncü çeyrek düzlem modelini, geriye kestirimci hatası olarak elde etmek için, kestirim bölgesi içindeki dizilim Şekil 3.4 'de gösterildiği gibi seçilmelidir:



Şekil 3.4 Örnek problem için kestirim destek bölgesi ve dizilim

0. aşama:

$$\begin{aligned} y((k_1, k_2)-0) &= y(k_1, k_2) \\ y((k_1, k_2)-1) &= y(k_1 -1, k_2) \\ y((k_1, k_2)-2) &= y(k_1 -1, k_2 -1) \\ y((k_1, k_2)-3) &= y(k_1, k_2 -1) \\ y((k_1, k_2)-4) &= y(k_1, k_2 -2) \\ y((k_1, k_2) -5) &= y(k_1 -1, k_2 -2) \\ y((k_1, k_2) -6) &= y(k_1 -2, k_2 -2) \\ y((k_1, k_2) -7) &= y(k_1 -2, k_2 -1) \\ y((k_1, k_2)-8) &= y(k_1 -2, k_2) \end{aligned} \quad (3.38)$$

Birinci çeyrek düzlem modelinin süzgeç katsayıları:

$$f_0^{(8)}(k_1, k_2) = y((k_1, k_2)-0) + a_0^{(8)}(1) y((k_1, k_2)-1) + \dots + a_0^{(8)}(m) y((k_1, k_2)-m) \quad (3.39)$$

Dördüncü çeyrek düzlem modelinin süzgeç katsayıları:

$$b_8^{(8)}(k_1, k_2) = y((k_1, k_2)-8) + g_8^{(8)}(7) y((k_1, k_2)-7) + \dots + g_8^{(8)}(0) y((k_1, k_2)-0) \quad (3.40)$$

$$E_{f_0}^{(8)}(k_1, k_2) = E[f_0^{(8)^2}(k_1, k_2)] \quad ; \quad E_{b_8}^{(8)}(k_1, k_2) = E[b_8^{(8)^2}(k_1, k_2)] \quad (3.41)$$

$$r_{ij} = E[y(k_1, k_2) y(k_1+i, k_2+j)] \quad (3.42)$$

(3.38) nolu bağıntıları kullanarak birinci aşamaya geçebilir ve aşağıdaki eşitlikleri yazabiliriz:

1. aşama:

$$\begin{bmatrix} f_0^{(1)}(k_1, k_2) \\ b_1^{(1)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_1}^{(1)} \\ \Gamma_{f_0}^{(1)} & 1 \end{bmatrix} \begin{bmatrix} f_0^{(0)}(k_1, k_2) \\ b_1^{(0)}(k_1, k_2) \end{bmatrix} \quad (1. basamak)$$

$$\begin{bmatrix} f_1^{(1)}(k_1, k_2) \\ b_2^{(1)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_2}^{(1)} \\ \Gamma_{f_1}^{(1)} & 1 \end{bmatrix} \begin{bmatrix} f_1^{(0)}(k_1, k_2) \\ b_2^{(0)}(k_1, k_2) \end{bmatrix} \quad (2. basamak)$$

$$\begin{bmatrix} f_2^{(1)}(k_1, k_2) \\ b_3^{(1)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_3}^{(1)} \\ \Gamma_{f_2}^{(1)} & 1 \end{bmatrix} \begin{bmatrix} f_2^{(0)}(k_1, k_2) \\ b_3^{(0)}(k_1, k_2) \end{bmatrix} \quad (3. basamak)$$

$$\begin{matrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{matrix}$$

$$\begin{bmatrix} f_7^{(1)}(k_1, k_2) \\ b_8^{(1)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_8}^{(1)} \\ \Gamma_{f_7}^{(1)} & 1 \end{bmatrix} \begin{bmatrix} f_7^{(0)}(k_1, k_2) \\ b_8^{(0)}(k_1, k_2) \end{bmatrix} \quad (8. basamak)$$

$$(3.43)$$

(3.33) bağıntısı ile verilen $\Gamma_{f_{p-n}}^{(n)}, \Gamma_{b_p}^{(n)}$ yansımaya katsayılarının bulunabilmesi için (3.32) bağıntısı ile verilen $\Delta_{f_{p-n} b_p}$ çapraz ilişkilerinin bulunması gerekir.

$$\Gamma_{f_0}^{(1)} = -\frac{\Delta_{f_0 b_1}^{(0)}}{E_{f_0}^{(0)}}, \quad \Gamma_{b_1}^{(1)} = -\frac{\Delta_{f_0 b_1}^{(0)}}{E_{b_1}^{(0)}} \quad (3.44)$$

olduğundan,

$$\Delta_{f_0 b_1}^{(0)} = E[f_0^{(0)}(k_1, k_2) b_1^{(0)}(k_1, k_2)] = E[y(k_1, k_2) y(k_1 - 1, k_2)] = r_{10} \quad (3.45)$$

$$E_{f_0}^{(0)} = E[f_0^{(0)^2}(k_1, k_2)] = E[y^2(k_1, k_2)] = r_{00} \quad (3.46)$$

$$E_{b_1}^{(0)} = E[b_1^{(0)^2}(k_1, k_2)] = E[y^2(k_1 - 1, k_2)] = r_{00} \quad (3.47)$$

Birinci aşama, birinci basamağa ilişkin yansımaya katsayıları için en son ifade yazılacak olursa:

$$\Gamma_{f_0}^{(1)} = -\frac{r_{10}}{r_{00}}, \quad \Gamma_{b_1}^{(1)} = -\frac{r_{10}}{r_{00}} \quad (3.48)$$

olduğu görülür. İkinci basamağa ilişkin yansımaya katsayıları hesaplanırsa:

$$\Delta_{f_1 b_2}^{(0)} = E[f_1^{(0)}(k_1, k_2) b_2^{(0)}(k_1, k_2)] = E[y(k_1 - 1, k_2) y(k_1 - 1, k_2 - 1)] = r_{01} \quad (3.49)$$

$$E_{f_1}^{(0)} = r_{00}, \quad E_{b_2}^{(0)} = r_{00} \quad (3.50)$$

olduğundan,

$$\Gamma_{f_1}^{(1)} = -\frac{r_{01}}{r_{00}}, \quad \Gamma_{b_2}^{(1)} = -\frac{r_{01}}{r_{00}} \quad (3.51)$$

olarak bulunur.

Diğer basamaklar için de aynı işlemlere devam edilirse, birinci adıma ilişkin ileri ve geri yansıma katsayılarının eşit çıktığı gözlenecektir.

Kestirim bölgesindeki elemanları sıralamak için, yalnızca yatay ve düşey yönler kullanıldığından, birinci aşama kafes basamaklarına ilişkin yansıma katsayılarını aşağıdaki gibi tanımlayabiliriz.

$$\Gamma_h^{(1)} = -\frac{r_{10}}{r_{00}} \quad , \quad \Gamma_v^{(1)} = -\frac{r_{01}}{r_{00}} \quad (3.52)$$

Birinci adıma ilişkin ifadeler yeniden düzenlenirse:

$$\begin{bmatrix} f_0^{(1)}(k_1, k_2) \\ b_1^{(1)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_h^{(1)} \\ \Gamma_h^{(1)} & 1 \end{bmatrix} \begin{bmatrix} y(k_1, k_2) \\ y(k_1 - 1, k_2) \end{bmatrix} \quad (1.basamak)$$

$$\begin{bmatrix} f_1^{(1)}(k_1, k_2) \\ b_2^{(1)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_v^{(1)} \\ \Gamma_v^{(1)} & 1 \end{bmatrix} \begin{bmatrix} y(k_1 - 1, k_2) \\ y(k_1 - 1, k_2 - 1) \end{bmatrix} \quad (2.basamak)$$

$$\begin{bmatrix} f_2^{(1)}(k_1, k_2) \\ b_3^{(1)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_h^{(1)} \\ \Gamma_h^{(1)} & 1 \end{bmatrix} \begin{bmatrix} y(k_1 - 1, k_2 - 1) \\ y(k_1, k_2 - 1) \end{bmatrix} \quad (3.basamak)$$

$$\begin{bmatrix} f_7^{(1)}(k_1, k_2) \\ b_8^{(1)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_v^{(1)} \\ \Gamma_v^{(1)} & 1 \end{bmatrix} \begin{bmatrix} y(k_1 - 2, k_2 - 1) \\ y(k_1 - 2, k_2) \end{bmatrix} \quad (8.basamak)$$

(3.53)

2.aşama:

$$\begin{bmatrix} f_0^{(2)}(k_1, k_2) \\ b_2^{(2)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_2}^{(2)} \\ \Gamma_{f_0}^{(2)} & 1 \end{bmatrix} \begin{bmatrix} f_0^{(1)}(k_1, k_2) \\ b_2^{(1)}(k_1, k_2) \end{bmatrix} \quad (1.basamak)$$

$$\begin{bmatrix} f_1^{(2)}(k_1, k_2) \\ b_3^{(2)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_3}^{(2)} \\ \Gamma_{f_1}^{(2)} & 1 \end{bmatrix} \begin{bmatrix} f_1^{(1)}(k_1, k_2) \\ b_3^{(1)}(k_1, k_2) \end{bmatrix} \quad (2.basamak)$$

$$\begin{bmatrix} f_6^{(2)}(k_1, k_2) \\ b_8^{(2)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_8}^{(2)} \\ \Gamma_{f_6}^{(2)} & 1 \end{bmatrix} \begin{bmatrix} f_6^{(1)}(k_1, k_2) \\ b_8^{(1)}(k_1, k_2) \end{bmatrix} \quad (7.\text{basamak})$$

(3.54)

İkinci adım, birinci basamağa ilişkin yansıma katsayıları için:

$$\Gamma_{f_0}^{(2)} = -\frac{\Delta_{f_0 b_2}^{(1)}}{E_{f_0}^{(1)}}, \quad \Gamma_{b_2}^{(2)} = -\frac{\Delta_{f_0 b_2}^{(1)}}{E_{b_2}^{(1)}} \quad (3.55)$$

$$\Delta_{f_0 b_2}^{(1)} = E[f_0^{(1)}(k_1, k_2) b_2^{(1)}(k_1, k_2)] \quad (3.56)$$

$$E_{f_0}^{(1)} = E[f_0^{(1)2}(k_1, k_2)] \quad , \quad E_{b_2}^{(1)} = E[b_2^{(1)2}(k_1, k_2)] \quad (3.57)$$

bağıntılarından yararlanılır. Diğer basamaklara ilişkin yansıma katsayılarını bulmak için de aynı yol izlenir.

3.aşama:

$$\begin{bmatrix} f_0^{(3)}(k_1, k_2) \\ b_3^{(3)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_3}^{(3)} \\ \Gamma_{f_0}^{(3)} & 1 \end{bmatrix} \begin{bmatrix} f_0^{(2)}(k_1, k_2) \\ b_3^{(2)}(k_1, k_2) \end{bmatrix} \quad (1.\text{basamak})$$

$$\begin{bmatrix} f_1^{(3)}(k_1, k_2) \\ b_4^{(3)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_4}^{(3)} \\ \Gamma_{f_1}^{(3)} & 1 \end{bmatrix} \begin{bmatrix} f_1^{(2)}(k_1, k_2) \\ b_4^{(2)}(k_1, k_2) \end{bmatrix} \quad (2.\text{basamak})$$

$$\begin{matrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{matrix}$$

$$\begin{bmatrix} f_5^{(3)}(k_1, k_2) \\ b_7^{(3)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_7}^{(3)} \\ \Gamma_{f_5}^{(3)} & 1 \end{bmatrix} \begin{bmatrix} f_5^{(2)}(k_1, k_2) \\ b_7^{(2)}(k_1, k_2) \end{bmatrix} \quad (6.\text{basamak})$$

(3.58)

4.aşama:

$$\begin{bmatrix} f_0^{(4)}(k_1, k_2) \\ b_4^{(4)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_4}^{(4)} \\ \Gamma_{f_0}^{(4)} & 1 \end{bmatrix} \begin{bmatrix} f_0^{(3)}(k_1, k_2) \\ b_4^{(3)}(k_1, k_2) \end{bmatrix} \quad (1.basamak)$$

$$\begin{bmatrix} f_1^{(4)}(k_1, k_2) \\ b_5^{(4)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_5}^{(4)} \\ \Gamma_{f_1}^{(4)} & 1 \end{bmatrix} \begin{bmatrix} f_1^{(3)}(k_1, k_2) \\ b_5^{(3)}(k_1, k_2) \end{bmatrix} \quad (2.basamak)$$

$$\begin{matrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{matrix}$$

$$\begin{bmatrix} f_4^{(4)}(k_1, k_2) \\ b_8^{(4)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_8}^{(4)} \\ \Gamma_{f_4}^{(4)} & 1 \end{bmatrix} \begin{bmatrix} f_4^{(3)}(k_1, k_2) \\ b_8^{(3)}(k_1, k_2) \end{bmatrix} \quad (5.basamak)$$

(3.59)

5.aşama:

$$\begin{bmatrix} f_0^{(5)}(k_1, k_2) \\ b_5^{(5)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_5}^{(5)} \\ \Gamma_{f_0}^{(5)} & 1 \end{bmatrix} \begin{bmatrix} f_0^{(4)}(k_1, k_2) \\ b_5^{(4)}(k_1, k_2) \end{bmatrix} \quad (1.basamak)$$

$$\begin{bmatrix} f_1^{(5)}(k_1, k_2) \\ b_6^{(5)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_6}^{(5)} \\ \Gamma_{f_1}^{(5)} & 1 \end{bmatrix} \begin{bmatrix} f_1^{(4)}(k_1, k_2) \\ b_6^{(4)}(k_1, k_2) \end{bmatrix} \quad (2.basamak)$$

$$\begin{matrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{matrix}$$

$$\begin{bmatrix} f_3^{(5)}(k_1, k_2) \\ b_8^{(5)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_8}^{(5)} \\ \Gamma_{f_3}^{(5)} & 1 \end{bmatrix} \begin{bmatrix} f_3^{(4)}(k_1, k_2) \\ b_8^{(4)}(k_1, k_2) \end{bmatrix} \quad (4.basamak)$$

(3.60)

6.aşama:

$$\begin{bmatrix} f_0^{(6)}(k_1, k_2) \\ b_6^{(6)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_6}^{(6)} \\ \Gamma_{f_0}^{(6)} & 1 \end{bmatrix} \begin{bmatrix} f_0^{(5)}(k_1, k_2) \\ b_6^{(5)}(k_1, k_2) \end{bmatrix} \quad (1.basamak)$$

$$\begin{bmatrix} f_1^{(6)}(k_1, k_2) \\ b_7^{(6)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_7}^{(6)} \\ \Gamma_{f_1}^{(6)} & 1 \end{bmatrix} \begin{bmatrix} f_1^{(5)}(k_1, k_2) \\ b_7^{(5)}(k_1, k_2) \end{bmatrix} \quad (2.basamak)$$

$$\begin{bmatrix} f_2^{(6)}(k_1, k_2) \\ b_8^{(6)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_8}^{(6)} \\ \Gamma_{f_2}^{(6)} & 1 \end{bmatrix} \begin{bmatrix} f_2^{(5)}(k_1, k_2) \\ b_8^{(5)}(k_1, k_2) \end{bmatrix} \quad \begin{matrix} (3.\text{basamak}) \\ (3.61) \end{matrix}$$

7.aşama:

$$\begin{bmatrix} f_0^{(7)}(k_1, k_2) \\ b_7^{(7)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_7}^{(7)} \\ \Gamma_{f_0}^{(7)} & 1 \end{bmatrix} \begin{bmatrix} f_0^{(6)}(k_1, k_2) \\ b_7^{(6)}(k_1, k_2) \end{bmatrix} \quad (1.\text{basamak})$$

$$\begin{bmatrix} f_1^{(7)}(k_1, k_2) \\ b_8^{(7)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_8}^{(7)} \\ \Gamma_{f_1}^{(7)} & 1 \end{bmatrix} \begin{bmatrix} f_1^{(6)}(k_1, k_2) \\ b_8^{(6)}(k_1, k_2) \end{bmatrix} \quad \begin{matrix} (2.\text{basamak}) \\ (3.62) \end{matrix}$$

8.aşama:

$$\begin{bmatrix} f_0^{(8)}(k_1, k_2) \\ b_8^{(8)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_8}^{(8)} \\ \Gamma_{f_0}^{(8)} & 1 \end{bmatrix} \begin{bmatrix} f_0^{(7)}(k_1, k_2) \\ b_8^{(7)}(k_1, k_2) \end{bmatrix} \quad \begin{matrix} (1.\text{basamak}) \\ (3.63) \end{matrix}$$

BÖLÜM 4.

KAFES PARAMETRELERİNİN İKİ BOYUTLU AKTARIM İŞLEVLERİ İLE İLİŞKİSİ

İki boyutlu z-dönüşümü cinsinden, sıfırıncı dereceden kestirim alanları aşağıdaki gibi verilir:

$$F_p^{(0)}(z_1, z_2) = B_p^{(0)}(z_1, z_2) = \delta_p(z_1, z_2) Y(z_1, z_2) \quad p = 0, 1, \dots, M \quad (4.1)$$

Burada $\delta_p(z_1, z_2)$, (3.2.c) bağıntısı ile tanımlanmıştır. Bu aşamada, daha yüksek derecelerden ileri ve geri yönde kestirim hatalarına ilişkin hata aktarım işlevleri aşağıdaki gibi tanımlanabilir:

$$A_p^{(n)}(z_1, z_2) = \frac{F_p^{(n)}(z_1, z_2)}{Y(z_1, z_2)} = \delta_p(z_1, z_2) \frac{F_p^{(n)}(z_1, z_2)}{F_p^{(0)}(z_1, z_2)} \quad (4.2.a)$$

ve

$$G_p^{(n)}(z_1, z_2) = \frac{B_p^{(n)}(z_1, z_2)}{Y(z_1, z_2)} = \delta_p(z_1, z_2) \frac{B_p^{(n)}(z_1, z_2)}{B_p^{(0)}(z_1, z_2)} \quad p = 0, 1, \dots, M \quad (4.2.b)$$

Bu nedenle (3.36.a) bağıntısı ile verilen iki boyutlu kafes süzgeçlerinin matris formuna ait z-dönüşümü aşağıdaki şekilde yazılabilir:

$$\begin{bmatrix} A_{p-n}^{(n)}(z_1, z_2) \\ G_p^{(n)}(z_1, z_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_p}^{(n)} \\ \Gamma_{f_{p-n}}^{(n)} & 1 \end{bmatrix} \begin{bmatrix} A_{p-n}^{(n-1)}(z_1, z_2) \\ G_p^{(n-1)}(z_1, z_2) \end{bmatrix} \quad (4.3.a)$$

Burada,

$$A_{p-n}^{(n)}(z_1, z_2) = \sum_{i=p-n}^p a_{p-n}^{(n)}(n-p+i) \delta_i(z_1, z_2) \quad (4.3.b)$$

ve

$$G_p^{(n)}(z_1, z_2) = \sum_{i=p-n}^p g_p^{(n)}(p-i) \delta_i(z_1, z_2) \quad (4.3.c)$$

olarak verilmektedir. $a_{p-n}^{(n)}(i)$ ve $g_p^{(n)}(i)$ katsayıları ise, sırasıyla ileri ve geri yönde polinom katsayılarıdır. (3.30) bağıntısından da görüleceği gibi:

$$a_{p-n}^{(n)}(0) = 1 \quad \text{ve} \quad g_p^{(n)}(n) = 1 \quad \text{dir.} \quad (4.4)$$

(4.3) bağıntıları ile verilen iki boyutlu aktarım işlevleri aşağıdaki gibi de yazılabilir.

$$A_{p-n}(z_1, z_2) = \mathbf{a}_{p-n}^{(n)T} \mathbf{O}_{p-n,n}(z_1, z_2) \quad , \quad G_p(z_1, z_2) = \mathbf{g}_p^{(n)T} \mathbf{O}_{p-n,n}(z_1, z_2) \quad (4.5)$$

Burada,

$$\mathbf{a}_{p-n}^{(n)} = \begin{bmatrix} 1 & a_{p-n}^{(n)}(1) & \dots & a_{p-n}^{(n)}(p) \end{bmatrix}^T \quad , \quad \mathbf{g}_p^{(n)} = \begin{bmatrix} g_p^{(n)}(p) & g_p^{(n)}(p-1) & \dots & g_p^{(n)}(1) & 1 \end{bmatrix}^T \quad (4.6.a)$$

ve

$$\mathbf{O}_{p-n,n}^{(n)} = \begin{bmatrix} \delta_{p-n}(z_1, z_2) & \delta_{p-n+1}(z_1, z_2) & \dots & \delta_n(z_1, z_2) \end{bmatrix}^T \quad (4.6.b)$$

olarak tanımlıdır.

İki boyutlu kafes katsayıları biliniyorsa, aşağıdaki yöntem izlenerek, ileri ve geri yönde aktarım işlevleri hesaplanabilir:

(1) Başlangıç koşulu olarak, $n = 0$ alınır. Böylece 0. aşama katsayı vektörleri aşağıdaki gibi tanımlanır.

$$\mathbf{a}_p^{(0)} = \mathbf{b}_p^{(0)} = [1] \quad , \quad p = 0, 1, 2, \dots, M \quad (4.7)$$

(2) n değeri bir birim artırılır, genişletilmiş $\mathbf{a}_{p-n}^{(n-1)}$ ve $\mathbf{g}_p^{(n-1)}$ vektörleri kullanılarak, $\hat{\mathbf{a}}_{p-n}^{(n)}$ ve $\hat{\mathbf{g}}_p^{(n)}$ vektörleri oluşturulur.

$$\hat{\mathbf{a}}_{p-n}^{(n)} = \begin{bmatrix} \mathbf{a}_{p-n}^{(n-1)} \\ 0 \end{bmatrix} ; \quad \hat{\mathbf{g}}_p^{(n)} = \begin{bmatrix} 0 \\ \mathbf{g}_p^{(n-1)} \end{bmatrix} , p = n, n+1, \dots, M \quad (4.8)$$

(3) n. aşama katsayı vektörleri için, (4.3.a) bağıntısında verilen şekli ile ileri ve geri yönde aktarım işlevleri şöyledir:

$$\begin{bmatrix} \mathbf{a}_{p-n}^{(n)} \\ \mathbf{g}_p^{(n)} \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_p}^{(n)} \\ \Gamma_{f_{p-n}}^{(n)} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{a}}_{p-n}^{(n)} \\ \hat{\mathbf{g}}_p^{(n)} \end{bmatrix} \quad (4.9)$$

$p = n, n+1, \dots, M$ değerleri için aşama aşama ilerlenerek, kafes parametrelerine bağlı aktarım işlevleri için tüm katsayılar elde edilir.

BÖLÜM 5.

SENTEZ MODELİ VE KARARLILIK

5.1 İki Boyutlu Kararlılık Teoremleri

Bir boyutlu durum için, aktarım işlevi $H(z) = 1 / A(z)$ şeklinde verilen nedensel bir sistemin kararlılığını test etmek oldukça kolaydır. Bir boyutlu polinom $A(z)$, her zaman birinci derece polinomların çarpımı şeklinde ifade edilebileceğinden, $H(z)$ 'in kutupları kolaylıkla belirlenebilir. Bir boyutlu nedensel bir sistemin kararlılığı, bütün kutuplarının birim daire içinde olmasına eşdeğerdir.

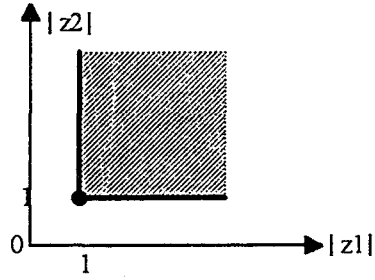
Yukarıdaki yaklaşım, iki boyutlu, birinci çeyrek düzlem destekli bir sistemin kararlılığını test etmek için kullanılamaz. Bu yaklaşımda, bütün kutupların yerlerinin bilinmesi gerekir, oysa iki boyutlu bir $A(z_1, z_2)$ polinomu genellikle daha düşük dereceli polinomların çarpımları cinsinden ifade edilemez. İki boyutta geliştirilen kararlılık ölçütlerinde bütün kutup yüzeylerinin belirgin bir sınırlamasının yapılmasına gerek duyulmamıştır. İki boyutlu kararlılık teoremleri şöyledir:

Teorem1: Bu teorem Shanks teoremi olarak da bilinir. Bir boyutlu duruma ait kararlılık koşulu aşağıdaki şekilde de ifade edilebilir.

$$\text{Kararlılık} \Leftrightarrow A(z) \neq 0, \quad |z| \geq 1 \quad (5.1.1)$$

Bunun iki boyutlu durum için ifadesi aşağıdaki gibidir.

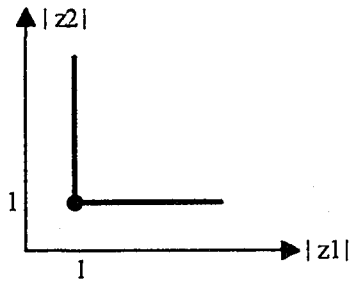
$$\text{Kararlılık} \Leftrightarrow A(z_1, z_2) \neq 0, \quad |z_1| \geq 1, |z_2| \geq 1 \quad (5.1.2)$$



Şekil 5.1.1 Teorem 1' e ilişkin kararlılık bölgesi

Teorem 2: Bu teorem de Shanks tarafından geliştirilmiştir ve aşağıdaki şekilde ifade edilebilir.

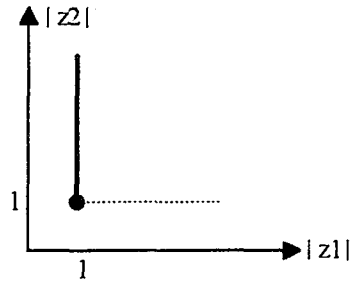
$$\begin{aligned} \text{Kararlılık} \Leftrightarrow & \text{(a) } A(z_1, z_2) \neq 0, \quad |z_1| = 1, |z_2| \geq 1 \\ & \text{(b) } A(z_1, z_2) \neq 0, \quad |z_1| \geq 1, |z_2| = 1 \end{aligned} \quad (5.1.3)$$



Şekil 5.1.2 Teorem 2' ye ilişkin kararlılık bölgesi

Teorem 3: Teorem 2 ile verilen koşulları daha basitleştiren bu teorem Huang teoremi adı ile de bilinir. Aşağıda gösterilen şekilde ifade edilir.

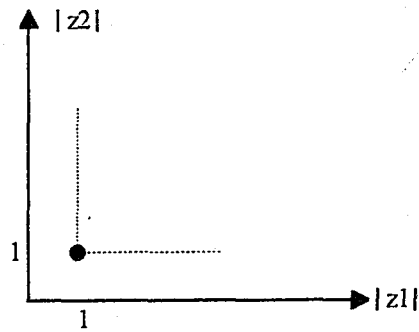
$$\begin{aligned} \text{Kararlılık} \Leftrightarrow & \text{(a) } A(z_1, z_2) \neq 0, \quad |z_1| = 1, |z_2| \geq 1 \\ & \text{(b) } A(z_1, z_2) \neq 0, \quad |z_1| \geq 1, |z_2| = 1. \end{aligned} \quad (5.1.4)$$



Şekil 5.1.3 Teorem 3' e ilişkin kararlılık bölgesi

Teorem 4: Bu teorem De Carlo - Strintzis teoremi olarak bilinir ve aşağıdaki şekilde ifade edilir:

$$\begin{aligned} \text{Kararlılık} \Leftrightarrow & \text{(a) } A(z_1, z_2) \neq 0, \quad |z_1| = |z_2| = 1 \\ & \text{(b) } A(z_1, z_2) \neq 0, \quad |z_1| \geq 1 \\ & \text{(c) } A(z_1, z_2) \neq 0, \quad |z_2| \geq 1. \end{aligned} \quad (5.1.5)$$



Şekil 5.1.4 Teorem 4' e ilişkin kararlılık bölgesi

5.2 Sentez Modeli - Kafes Yapısı Kararlılık Koşulları

Bir boyutlu yapıda olduğu gibi, iki boyutta da, kafes parametreleri bilindiğinde, kafes analiz yapısını, kafes sentez yapısına dönüştürmek mümkündür. Böylelikle, $u(k_1, k_2) = f_0^{(M)}(k_1, k_2)$ veya $u(k_1, k_2) = b_m^{(M)}(k_1, k_2)$ giriş verilerinden, sırasıyla $y(k_1, k_2) = f_0^{(0)}(k_1, k_2)$ veya $y(k_1, k_2) - M = b_m^{(0)}(k_1, k_2)$ orijinal verileri elde edilecektir. Kafes yapısında kullanılan aşama sayısı yeterli ise, son aşama olan M için ileri ve geri kestirimler beyaz gürültüye yakınsayacaktır, yani $u(k_1, k_2)$ beyaz gürültüdür. (3.36) ile verilen bağıntılar kullanılarak, sentez modelini ait yapı aşağıdaki şekilde oluşturulabilir. Birinci çeyrek düzlem kestirim modeli için (3.36.a) bağıntısı yeniden düzenlenirse,

$$f_0^{(n-1)}(k_1, k_2) = f_0^{(n)}(k_1, k_2) - \Gamma_{b_n}^{(n)} b_n^{(n-1)}(k_1, k_2) \quad (5.2.1.a)$$

$$b_n^{(n)}(k_1, k_2) = \Gamma_{f_0}^{(n)} f_0^{(n-1)}(k_1, k_2) + b_n^{(n-1)}(k_1, k_2) \quad (5.2.1.b)$$

yazılabilir. Burada $n = M, M-1, \dots, 2, 1$ dir.

İleri ve geri hata alanları için ise,

$$\begin{bmatrix} f_{p-n}^{(n)}(k_1, k_2) \\ b_p^{(n)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_p}^{(n)} \\ \Gamma_{f_{p-n}}^{(n)} & 1 \end{bmatrix} \begin{bmatrix} f_{p-n}^{(n-1)}(k_1, k_2) \\ b_p^{(n-1)}(k_1, k_2) \end{bmatrix} \quad (5.2.1.c)$$

yazılabilir.

Çeyrek düzlem geri yönde kestirim hataları kullanılarak, ikinci (veya dördüncü) çeyrek düzlem sentez modeli, (4.2.1) bağıntıları ile verilen ileri ve geri yönde kestirim hatalarının kendi aralarında yer değiştirmesi ile elde edilir.

$$b_n^{(n-1)}(k_1, k_2) = b_n^{(n)}(k_1, k_2) - \Gamma_{f_0}^{(n)} f_0^{(n-1)}(k_1, k_2) \quad (5.2.2.a)$$

$$f_0^{(n)}(k_1, k_2) = f_0^{(n-1)}(k_1, k_2) + \Gamma_{b_n}^{(n)} b_n^{(n-1)}(k_1, k_2) \quad (5.2.2.b)$$

(4.1) ve (4.2) eşitlikleri ile verilen bağıntılar, Şekil'de gösterilen birinci ve ikinci (veya dördüncü) çeyrek düzlem için sentez modellerini tanımlamaktadırlar.

Birinci çeyrek düzlem modeline ilişkin, sentez algoritması $u(k_1, k_2) = f_0^{(M)}(k_1, k_2)$ beyaz gürültü alanı ile başlar ve bütün geri yönde hata süzgeçlerinin başlangıç koşulları sıfır kabul edilir. Benzer şekilde, ikinci (veya dördüncü) çeyrek düzlem modeli için, giriş $u(k_1, k_2) = b_M^{(M)}(k_1, k_2)$ veri alanıdır ve bütün ileri yönde hata süzgeçlerinin başlangıç koşulları sıfır kabul edilir.

(4.2.1) ile verilen bağıntıların z-dönüşümü alınırsa,

$$F_0^{(n-1)}(z_1, z_2) = F_0^{(n)}(z_1, z_2) - \Gamma_{b_n}^{(n)} B_n^{(n-1)}(z_1, z_2) \quad (5.2.3.a)$$

$$B_n^{(n)}(z_1, z_2) = \Gamma_{f_0}^{(n)} F_0^{(n-1)}(z_1, z_2) + B_n^{(n-1)}(z_1, z_2) \quad (5.2.3.b)$$

ve

$$\begin{bmatrix} F_{p-n}^{(n)}(z_1, z_2) \\ B_p^{(n)}(z_1, z_2) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{b_p}^{(n)} \\ \Gamma_{f_{p-n}}^{(n)} & 1 \end{bmatrix} \begin{bmatrix} F_{p-n}^{(n-1)}(z_1, z_2) \\ B_p^{(n-1)}(z_1, z_2) \end{bmatrix} \quad (5.2.3.c)$$

elde edilir.

Sentez modelindeki n. aşama için aktarım işlevi aşağıdaki bağıntı ile verilir.

$$\frac{F_0^{(n-1)}(z_1, z_2)}{F_0^{(n)}(z_1, z_2)} = \frac{1}{1 + D^{(n)}(z_1, z_2)} \quad (5.2.4.a)$$

$D^{(n)}(z_1, z_2)$ şöyle tanımlıdır.

$$D^{(n)}(z_1, z_2) = \frac{F_0^{(n)}(z_1, z_2) - F_0^{(n-1)}(z_1, z_2)}{F_0^{(n-1)}(z_1, z_2)} \quad (5.2.4.b)$$

(4.2.3) ve (4.2.4.b) bağıntıları kullanılarak,

$$D^{(n)}(z_1, z_2) = \Gamma_{b_n}^{(n)} \frac{B_n^{(n-1)}(z_1, z_2)}{F_0^{(n-1)}(z_1, z_2)} \quad (5.2.5)$$

yazılabilir.

(5.2.3.a) ile verilen bağıntı aşağıdaki şekilde yeniden düzenlenirse,

$$\frac{F_0^{(n-1)}(z_1, z_2)}{F_0^{(n)}(z_1, z_2)} = \frac{1 + D^{(n-1)}(z_1, z_2)}{1 + D^{(n-1)}(z_1, z_2) + \tilde{D}^{(n)}(z_1, z_2)} \quad (5.2.6.a)$$

elde edilir. Burada,

$$\tilde{D}^{(n)}(z_1, z_2) = \Gamma_{b_n}^{(n)} \frac{B_n^{(n-1)}(z_1, z_2)}{F_0^{(n-2)}(z_1, z_2)} \quad n = 2, 3, \dots, M \quad (5.2.6.b)$$

$$\tilde{D}^{(0)}(z_1, z_2) = D^{(1)}(z_1, z_2) = 0 \quad (5.2.6.c)$$

olarak verilir.

$H(z_1, z_2) = 1 / B(z_1, z_2)$ aktarım işlevinin kararlı olması için,

$$B(z_1, z_2) \neq 0, \quad |z_1| \geq 1, \quad |z_2| \geq 1 \quad (5.2.7)$$

koşullarının sağlanması gerektiği bilinmektedir. Marzetta 'nın teoremi kullanılarak, eğer $F_0^{(n-1)}(z_1, z_2) / F_0^{(n)}(z_1, z_2)$ aktarım işlevi kararlı ise,

$$|1 + D^{(n-1)}(z_1, z_2)| > |\tilde{D}^{(n)}(z_1, z_2)|, \quad |z_1| = |z_2| = 1 \quad (5.2.8)$$

koşulu sağlandığında, $F_0^{(n)}(z_1, z_2) / F_0^{(n+1)}(z_1, z_2)$ 'nin de kararlı olacağı gösterilebilir.

Sentez modelinin kararlı olabilmesi için, her kafes aşamasının kararlı olması gerekecektir. Sentez modelinin aktarım işlevi aşağıdaki gibi verilir.

$$\begin{aligned}
 \frac{Y(z_1, z_2)}{U(z_1, z_2)} &= \frac{F_0^{(0)}(z_1, z_2)}{F_0^{(M)}(z_1, z_2)} \\
 &= \frac{F_0^{(0)}(z_1, z_2)}{F_0^{(1)}(z_1, z_2)} \frac{F_0^{(1)}(z_1, z_2)}{F_0^{(2)}(z_1, z_2)} \cdots \frac{F_0^{(M-1)}(z_1, z_2)}{F_0^{(M)}(z_1, z_2)} \\
 &= \prod_{n=1}^M \frac{F_0^{(n-1)}(z_1, z_2)}{F_0^{(n)}(z_1, z_2)} \\
 &= \frac{1}{\prod_{n=1}^M (1 + D^{(n)}(z_1, z_2))} \tag{5.2.9}
 \end{aligned}$$

(5.2.6.a) ve (5.2.7) kullanılarak, birinci aşamanın kararlılığı için, aşağıdaki bağıntı elde edilir.

$$\begin{aligned}
 B(z_1, z_2) &= 1 + D^{(1)}(z_1, z_2) = 1 + \Gamma_{b_1}^{(1)} \frac{B_1^{(0)}(z_1, z_2)}{F_0^{(0)}(z_1, z_2)} \\
 &= 1 + \Gamma_{b_1}^{(1)} (\delta_1(z_1, z_2) / \delta_0(z_1, z_2)) \tag{5.2.10}
 \end{aligned}$$

Dizilim vektörünün elemanları olan, $\delta_0(z_1, z_2) = 1$ ve $\delta_1(z_1, z_2) = z_1^{-1}$ (veya z_2^{-1}) yukarıdaki bağıntıda yerine konularak ve (5.2.7) bağıntısı ile verilen gerek ve yeter koşul kullanılarak,

$$\left| \Gamma_{b_1}^{(1)} \right| < 1 \tag{5.2.11}$$

olması gerektiği sonucuna varılır.

$\Gamma_{b_1}^{(1)}$ yansıma katsayısı, Burg [] tarafından gösterildiği gibi, daima (4.2.11) ile verilen koşulu sağlayacaktır. Bu koşulun birinci aşama için sağlandığı kabul edilirse, (4.2.8) eşitliği uygulanarak ikinci aşamanın kararlılığı belirlenir.

$$|1 + D^{(1)}(z_1, z_2)| > |\tilde{D}^{(2)}(z_1, z_2)|, \quad |z_1| = |z_2| = 1 \tag{5.2.12}$$

(5.2.4) ve (5.2.10) bağıntıları kullanılarak, (5.2.12) aşağıdaki gibi yeniden düzenlenebilir.

$$\left| 1 + \Gamma_{b_1}^{(1)} \delta_1(z_1, z_2) \right| > \left| \Gamma_{b_2}^{(2)} B_2^{(1)}(z_1, z_2) / Y(z_1, z_2) \right|, |z_1| = |z_2| = 1 \quad (5.2.13.a)$$

Burada,

$$\begin{aligned} B_2^{(1)}(z_1, z_2) &= B_2^{(0)}(z_1, z_2) + \Gamma_{f_1}^{(1)} F_1^{(0)}(z_1, z_2) \\ &= (\delta_2(z_1, z_2) + \Gamma_{f_1}^{(1)} \delta_1(z_1, z_2)) Y(z_1, z_2) \end{aligned} \quad (5.2.13.b)$$

olarak tanımlıdır.

(5.2.13.a) eşitliği, $|z_1| \geq 1$ ve $|z_2| \geq 1$ için $B(z_1, z_2) = 1 + D^{(n)}(z_1, z_2)$ 'ye ait hiçbir sıfır olmamasını gerektirir.

(5.2.12) ile verilen kararlılık koşulu, her kademedede uygulanıp kontrol yapılabilir. İlk aşama için, $\tilde{D}^{(1)}(z_1, z_2) = 0$ koşulunun sağlanması ile, diğer aşamaların kararlılığı da sağlanmış olur.

SONUÇLAR

2-B ' lu dik kafes süzgeçlerinin bir çok üstünlükleri bulunmaktadır. Kafes yapısının dik olması, gereksiz bilgiyi ortadan kaldırmakta ve optimum süzgeç olma avantajını doğurmaktadır. Bu nedenle adaptif uygulamalar için yakınsama hızı açısından tercih edilirler.

Önerilen yöntemle matris tersi alınmadan, 2-B 'lu genişletilmiş normal denklemin matris formunda Levinson algoritması ile tam çözümü sunulmuştur. Yöntem çeyrek düzlem ve simetrik olmayan yarı düzleme uygulanabilmektedir.

Bu çalışmanın amacı, Kayran tarafından geliştirilen yöntemin, Burg yöntemi çözümleri esas alınarak incelenmesidir. Böylece kısa veri kayıtları için dahi verimli kestirimler elde edilebileceği gösterilmiştir. Yöntemin başarısı, doğrudan veri alanları üzerinden çalışılıp, örnek özilişkiler yerine çapraz ilişkiler ile gerçek veriler üzerinden hesaplama yapılmasına dayanır. Veri boyutu arttıkça, örnek özilişki yöntemi ile Burg tabanlı yöntemin sonuçlarının birbirine yakınsadığı gözlenmiştir. Elde edilen tüm sonuçlar ve bunların örnek özilişki yöntemi ile karşılaştırılması Ek-A ' da sunulmuştur.

Yöntemin özbağımlı-kayan ortalama (ARMA = Autoregressive Moving Average) modellere uygulanması ise araştırmaya açık bir konudur.

KAYNAKLAR

- [1] BOSE, N.K., "Multidimensional Digital Signal Processing: Problems, Progress and Future Scopes", Proc. IEEE, Vol .78, April 1990
- [2] DUDGEON, D.E. ve MERSEREAU, R.M., "Multidimensional Digital Signal Processing ", Englewood Cliffs, New Jersey, Prentice Hall, 1984 .
- [3] ERTÜZÜN, A., KAYRAN, A.H. ve PANAYIRCI, E., "Entropy Relations of 2-D Lattice Filters", Proc. Bilcon Conference on New Trends in Comm., Cont. and Signal Processing, Ankara, Türkiye, Temmuz 1990
- [4] HAYKIN, S., "Adaptive Filter Theory ", Englewood Cliffs, New Jersey, Prentice Hall 1986 .
- [5] KAYRAN, A.H., "Stability, Stabilization and Design of Multidimensional Recursive Digital Filters", Ph. D. Dissertation, Department of Elect. Eng., Imperial College of Science and Tech., London , 1981
- [6] KAYRAN, A. H., "Design of 2-D Recursive Filters with Asymmetric Half-plane Modeling", IEEE. Proc. Volume 137, No:6, Dec. 1990.
- [7] KAYRAN, A.H., "Two Dimensional Orthogonal Lattice Structures for Autoregressive Modeling of Random Fields", submitted for publication in IEEE Proc. Vol.137 . Dec. 1990 .
- [8] KWAN, H.K., ve LUI, Y.C., "Lattice Implementation of Two Dimensional Recursive Digital Filters", IEEE . Trans. on ASSP., Vol. 36 , March 1989
- [9] LENK, P.J. ve PARKER, S.R., "Schur and Levinson recursions for two dimensional lattice filters", Proc. Int. Symp. on Circuits and Systems, pp.92-95, San Jose, CA, May 1986
- [10] LENK, P.J. ve PARKER, S.R., "Orthogonal two dimensional lattice filters", 19-th Asilomar Conference on Circuits, Systems and Computers, pp.278-282, Pasific Grove, CA, Nov. 1985
- [11] LEV-ARI, H. ve PARKER, S.R., "Stable and Efficient 2-D Lattice Filters ", Proc. IEEE . Symp. on Circuits and Systems, San Jose, CA, May 1986
- [12] LIM, J.S., "Two Dimensional Signal and Image Processing", Englewood Cliffs, New Jersey Prentice Hall , 1990 .

- [13] MARZETTA, T.L., "Two Dimensional Linear Prediction Error Filters and Reflection Coefficient Arrays", IEEE Transactions on ASSP., Vol. 28, Dec. 1980 .
- [13] ORFANIDIS, S.J., "Optimum Signal Processing", Macmillan Publishing Company, 866 Third Avenue, New York, 1988 .
- [14] PARKER, S.R. ve KAYRAN, A.H., "Lattice Parameter Autoregressive Modeling of 2-D Fields, Part I: The Quarter Plane Case", IEEE. Trans. on ASSP. Vol.32 Aug. 1984 .
- [14] PROAKIS, J.G. ve MANOLAKIS, D.G., "Introduction to Digital Signal Processing", McMillan, New York, 1989 .
- [15] RANGANATH, S. ve JAIN, A.K., "Two Dimensional Linear Prediction Models, Part I: Spectral Factorization and Realization", IEEE. Trans. on. ASSP., Vol. 33, Feb. 1985 .
- [16] SHENTOV, O.V., MILTRA, S.K ve ANDERSON, B.D. , " Stability Testing of 2-D Recursive Digital Filters Based on a Circuit-Theoretic Approach ", IEEE. Trans. on Circuits and Systems, Vol. 36, March 1989.
- [17] THERRIEN, C. W., "Relation Between 2-D and Multichannel Linear Prediction", IEEE. Trans. on ASSP., Vol. 37, Nov. 1989 .
- [18] TÜRE, S., "Rastgele alanların AR Modellenmesi için Dik Kafes Süzgeci ve Spektrum Kestirimine Uygulanması", Y. Lisans Tezi, İTÜ, 1993 .

EK A

Bu bölümde, kararlılık koşullarını sağlayacak şekilde seçilen AR katsayılarına bağlı olarak üretilen veri alanına ait katsayıların Burg Yöntemi tabanlı 2-B ' lu kafes süzgeçleri yöntemi ile kestirilmesine ilişkin sonuçlar sunulmuştur. Sağlıklı bir karşılaştırma yapılabilmesi açısından, Türe [18] tarafından kullanılan katsayılar seçilmiştir. Bütün kestirimler karşılaştırmalı olarak sunulmuştur.

Kararlılık koşulu için pratik ölçüt, katsayı mutlak değerlerinin toplamının "1" 'den küçük olmasıdır. Teoride kullanılan diğer ölçütler Bölüm 5.1 'de verilmiştir.

ÖRNEK A.1. Birinci derece, birinci çeyrek düzlem

Üretilen AR veri alanına ait fark denklemi aşağıda verilen şekildedir.

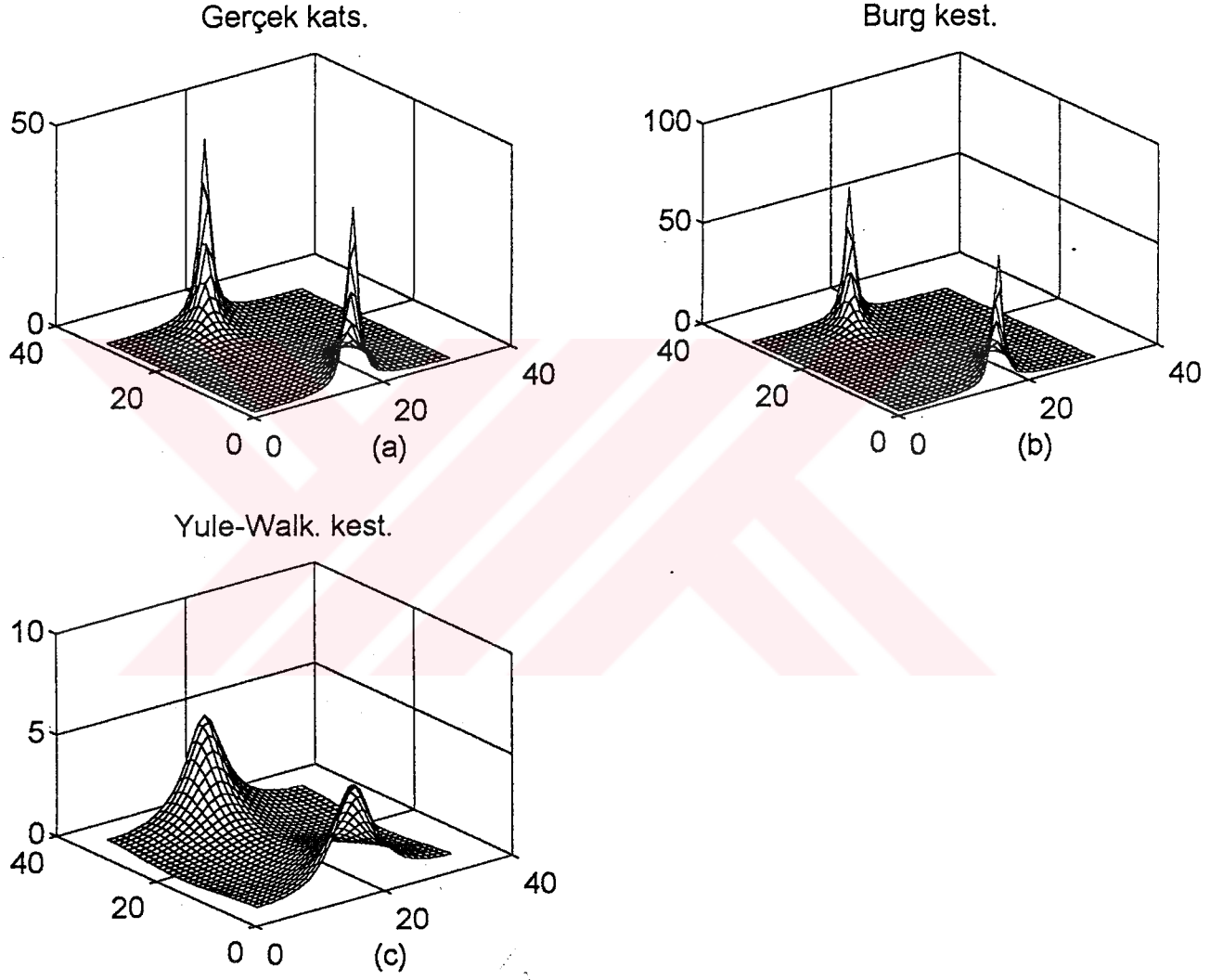
$$y(k_1, k_2) = w(k_1, k_2) + 0.4*y(k_1 - 1, k_2) - 0.3*y(k_1 - 1, k_2 - 1) - 0.15*y(k_1, k_2 - 1)$$

Burada $w(k_1, k_2)$, iki boyutlu, ortalaması "sıfır" ve varyansı "bir" olan beyaz gürültüdür. Değişik veri boyutları için elde edilen karşılaştırmalı sonuçlar tablo A.1 'de sunulmuştur. Türe [13], örnek özdeşliği yöntemini kullandığından, elde ettiği sonuçlar Yule-Walker çözümünü vermiştir. Tablolarda karşılaştırma yapılırken bu sonuçlar Y-W olarak kısaltılmıştır.

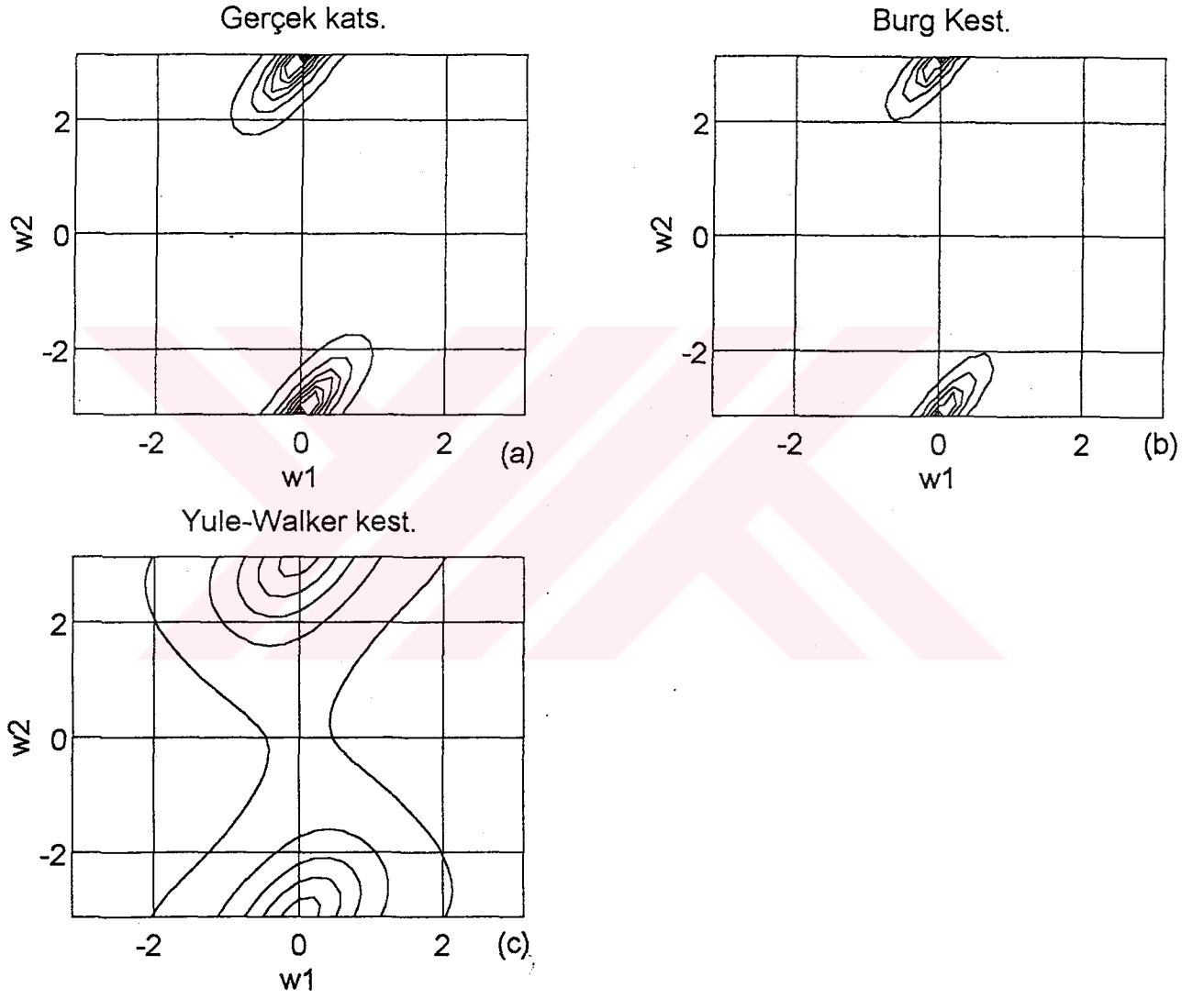
Tablo A.1

Gerçek katsayılar	Veri boyutu ve kestirilen katsayılar		
	10 x 10	20 x 20	35 x 35
-0.4	-0.4087(BURG)	-0.4173(BURG)	-0.4133(BURG)
	-0.3034(Y-W)	-0.3158(Y-W)	-0.3426(Y-W)
0.3	0.3271(BURG)	0.2854(BURG)	0.2882(BURG)
	0.0574(Y-W)	0.2646(Y-W)	0.3167(Y-W)
0.15	0.1393(BURG)	0.0783(BURG)	0.1487(BURG)
	0.2143(Y-W)	0.1654(Y-W)	0.1432(Y-W)

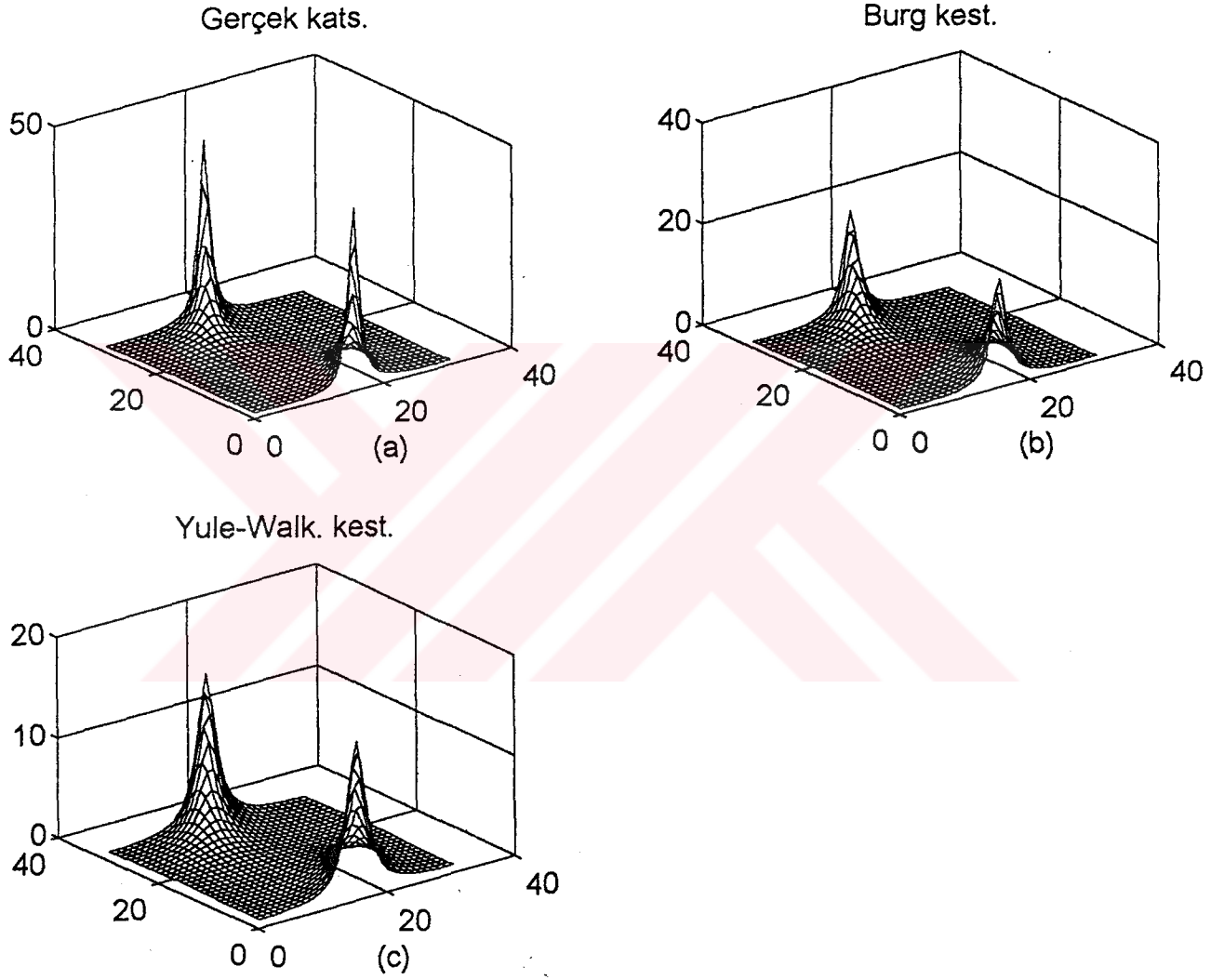
Gerçek katsayılar, Burg ve Yule-Walker yöntemleri ile kestirilen katsayılarla ilişkin AR-spektrumlar ve contour çizimleri Şekil A.1, Şekil A.2 ve Şekil A.3 'de verilmiştir.



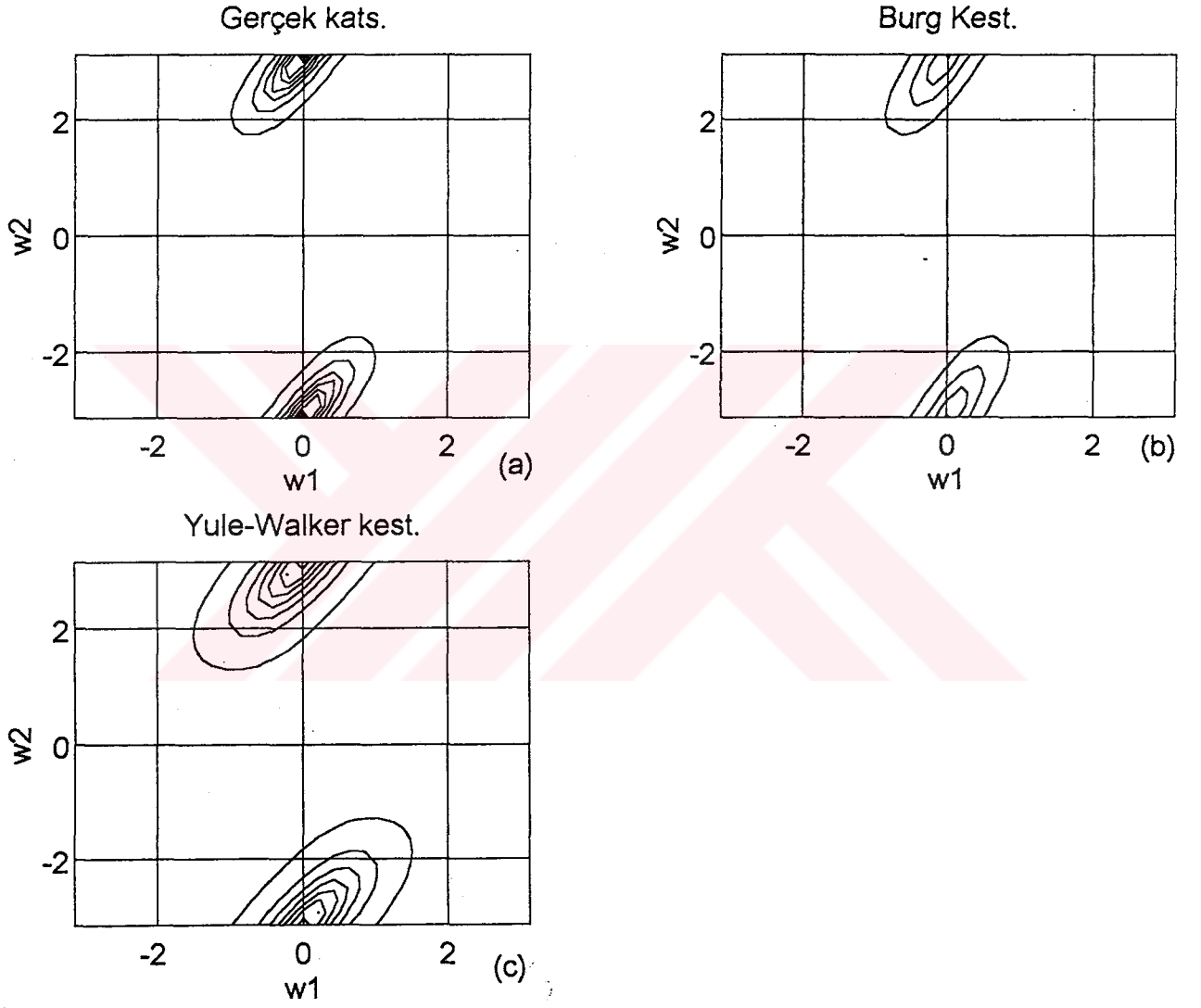
Şekil A.1.a Birinci dereceden üç destek noktalı birinci çeyrek düzlem için tespit edilen katsayılarla göre AR-spektrum örnekleri . Veri boyutu: 10 x 10



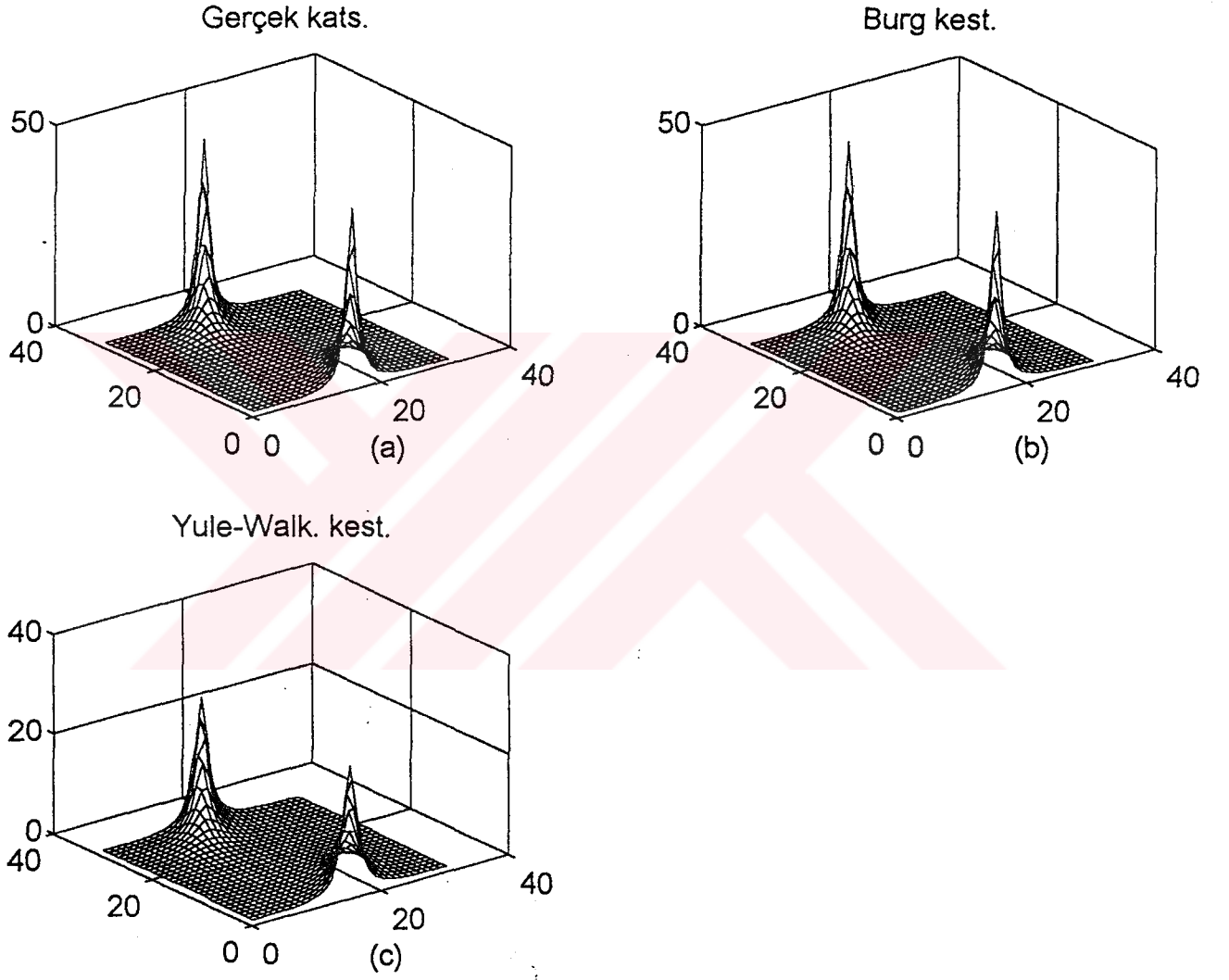
Şekil A.1.b Birinci dereceden üç destek noktalı birinci çeyrek düzlem için tespit edilen katsayılarla göre contour örnekleri . Veri boyutu: 10 x 10



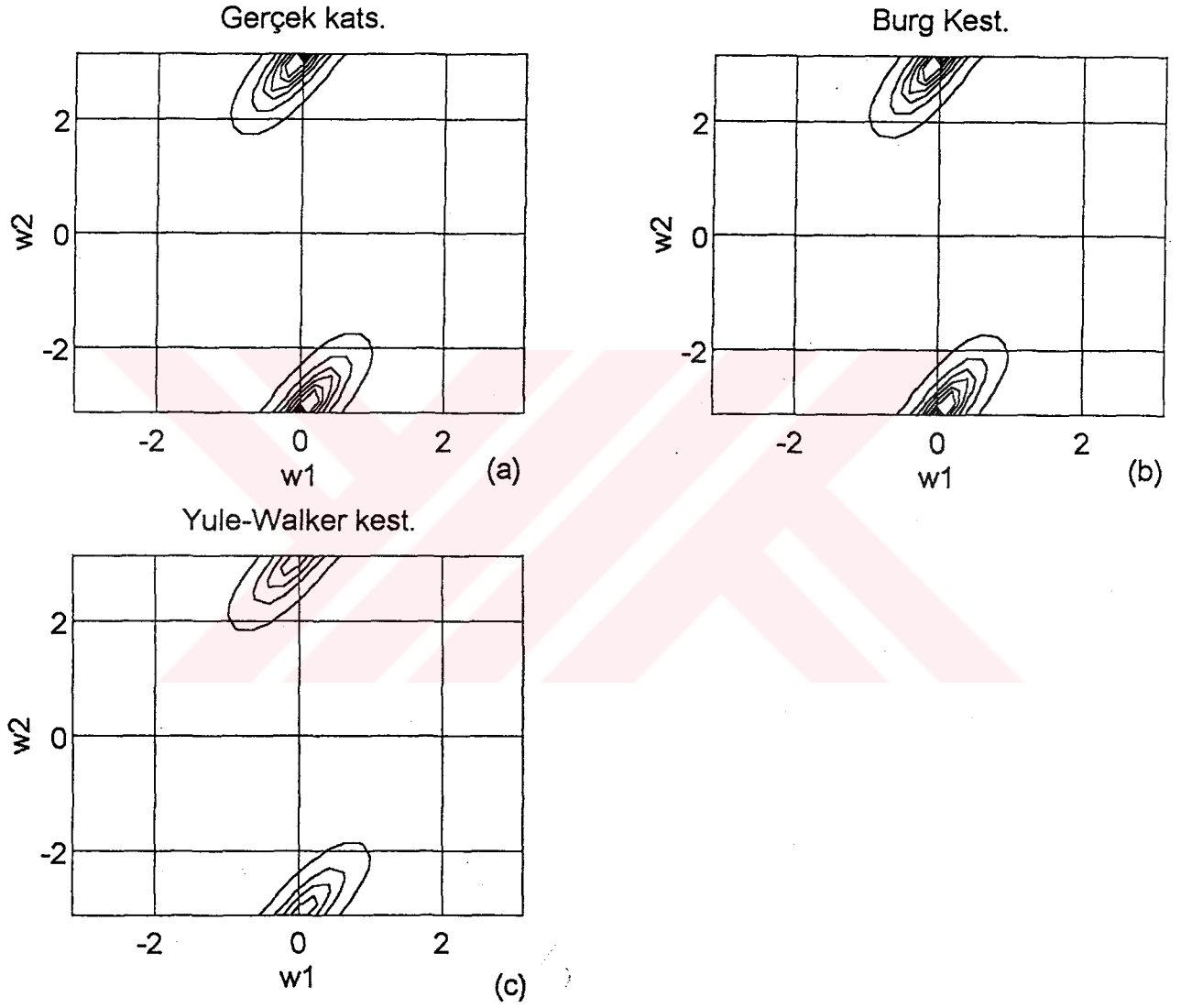
Şekil A.2.a Birinci dereceden üç destek noktalı birinci çeyrek düzlem için tespit edilen katsayılarla göre AR-spektrum örnekleri . Veri boyutu: 20 x 20



Şekil A.2.b Birinci dereceden üç destek noktalı birinci çeyrek düzlem için tespit edilen katsayılarla göre contour örnekleri . Veri boyutu: 20 x 20



Şekil A.3.a Birinci dereceden üç destek noktalı birinci çeyrek düzlem için tespit edilen katsayılarla göre AR-spektrum örnekleri . Veri boyutu: 35 x 35



Şekil A.3.b Birinci dereceden üç destek noktalı birinci çeyrek düzlem için tespit edilen katsayılarla göre contour örnekleri . Veri boyutu: 35 x 35

ÖRNEK A.2: İkinci derece, birinci çeyrek düzlem

Üretilen AR veri alanına ait fark denklemi aşağıda verilen şekildedir.

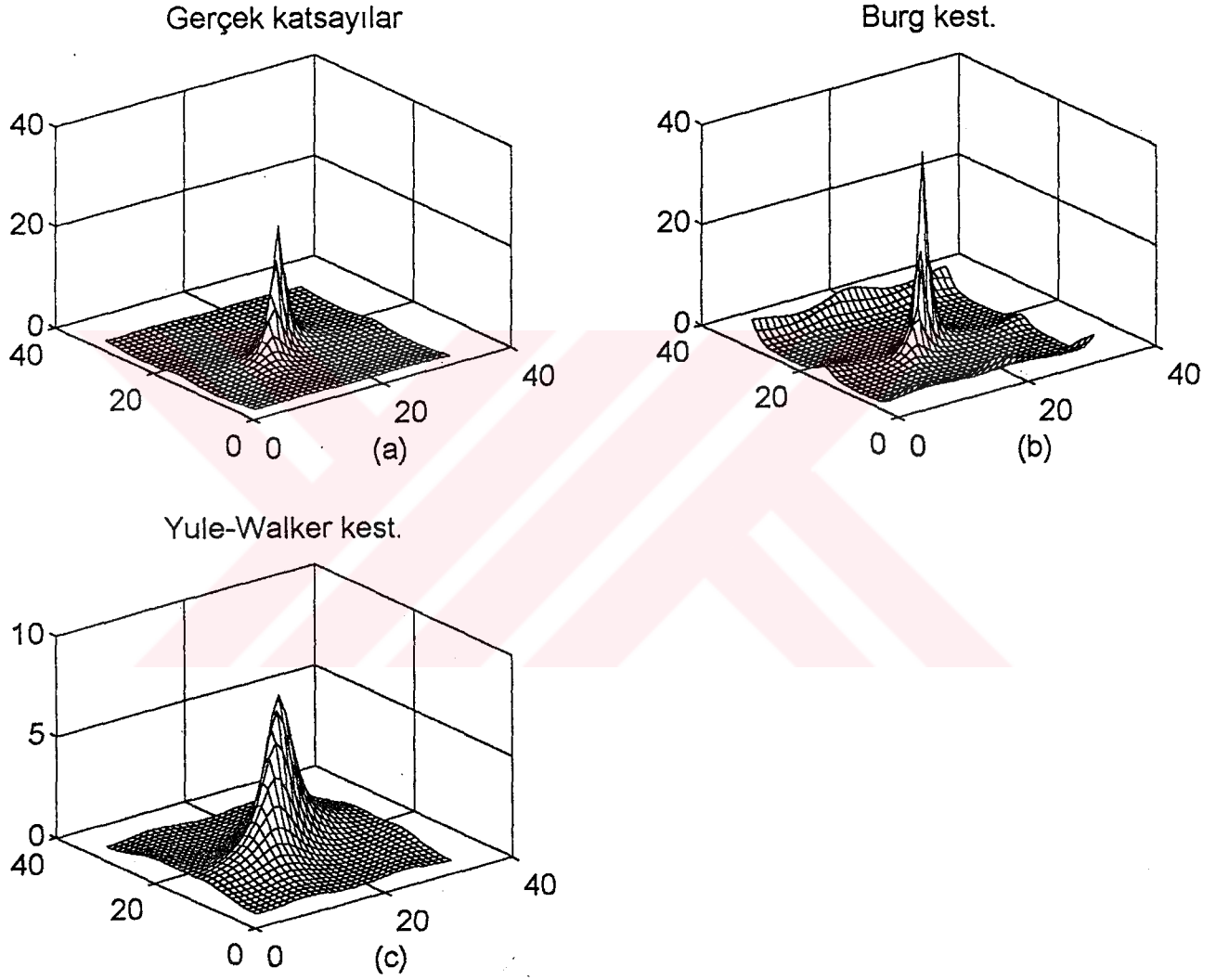
$$y(k_1, k_2) = w(k_1, k_2) + 0.13 * y(k_1 - 1, k_2) + 0.18 * y(k_1 - 1, k_2 - 1) + 0.08 * y(k_1, k_2 - 1) + 0.15 * y(k_1, k_2 - 2) + 0.045 * y(k_1 - 1, k_2 - 2) + 0.075 * y(k_1 - 2, k_2 - 2) + 0.108 * y(k_1 - 2, k_2 - 1) + 0.033 * y(k_1 - 2, k_2)$$

Değişik veri boyutları için elde edilen karşılaştırmalı sonuçlar tablo A.2 'de sunulmuştur.

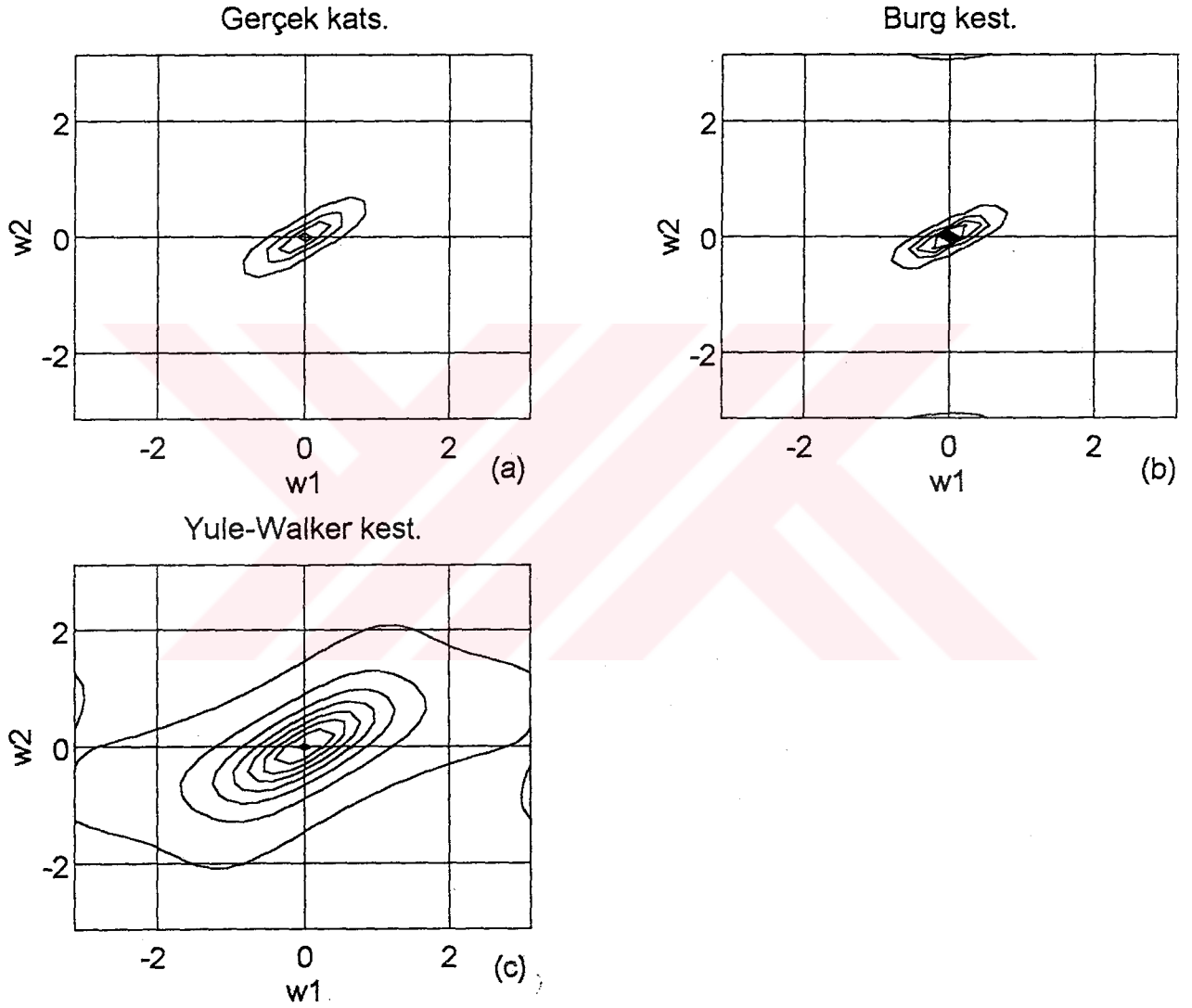
Tablo A.2

Gerçek katsayılar	Veri boyutu ve kestirilen katsayılar		
	10 x 10	20 x 20	35 x 35
-0.13	-0.13(BURG) -0.1105(Y-W)	-0.1879(BURG) -0.0881(Y-W)	-0.1593(BURG) -0.111(Y-W)
-0.18	-0.1170(BURG) -0.1709(Y-W)	-0.2196(BURG) -0.2599(Y-W)	-0.1979(BURG) -0.1705(Y-W)
-0.08	0.1023(BURG) -0.286(Y-W)	-0.0621(BURG) -0.1404(Y-W)	-0.0882(BURG) -0.0708(Y-W)
-0.15	-0.2319(BURG) -0.0567(Y-W)	-0.2161(BURG) -0.1789(Y-W)	-0.1352(BURG) -0.1702(Y-W)
-0.045	-0.0058(BURG) -0.0488(Y-W)	-0.0488(BURG) -0.0269(Y-W)	-0.0563(BURG) -0.0358(Y-W)
-0.075	-0.1619(BURG) 0.0295(Y-W)	-0.0687(BURG) -0.0178(Y-W)	-0.0442(BURG) -0.0946(Y-W)
-0.108	-0.1147(BURG) -0.0672(Y-W)	-0.1275(BURG) -0.1059(Y-W)	-0.0987(BURG) -0.1289(Y-W)
-0.033	-0.1826(BURG) 0.0567(Y-W)	-0.0473(BURG) 0.0123(Y-W)	-0.0413(BURG) -0.0249(Y-W)

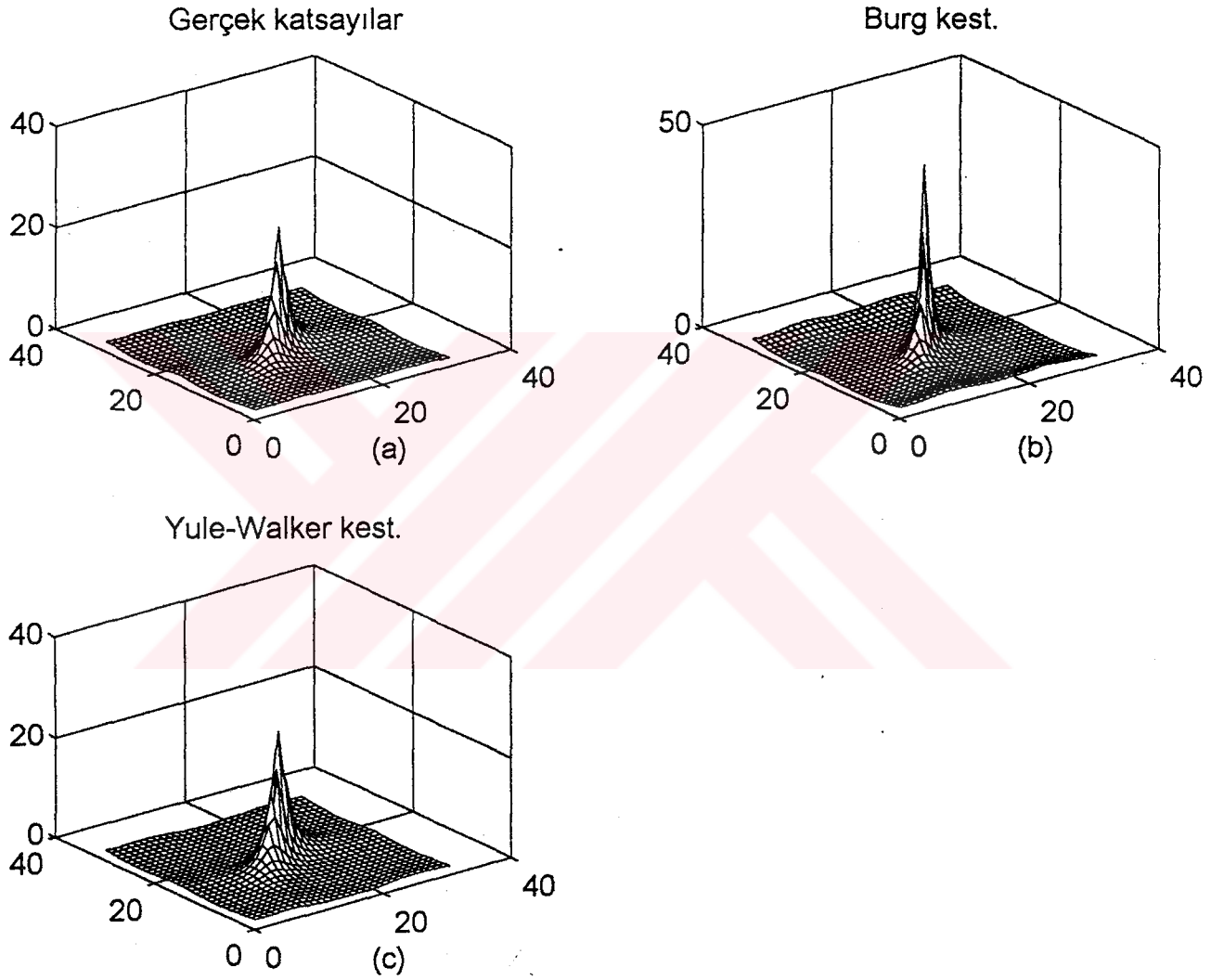
Gerçek katsayılar, Burg ve Yule-Walker yöntemleri ile kestirilen katsayılarla ilişkin AR-spektrumlar ve contour çizimleri Şekil A.4, Şekil A.5 ve Şekil A.6 'da verilmiştir.



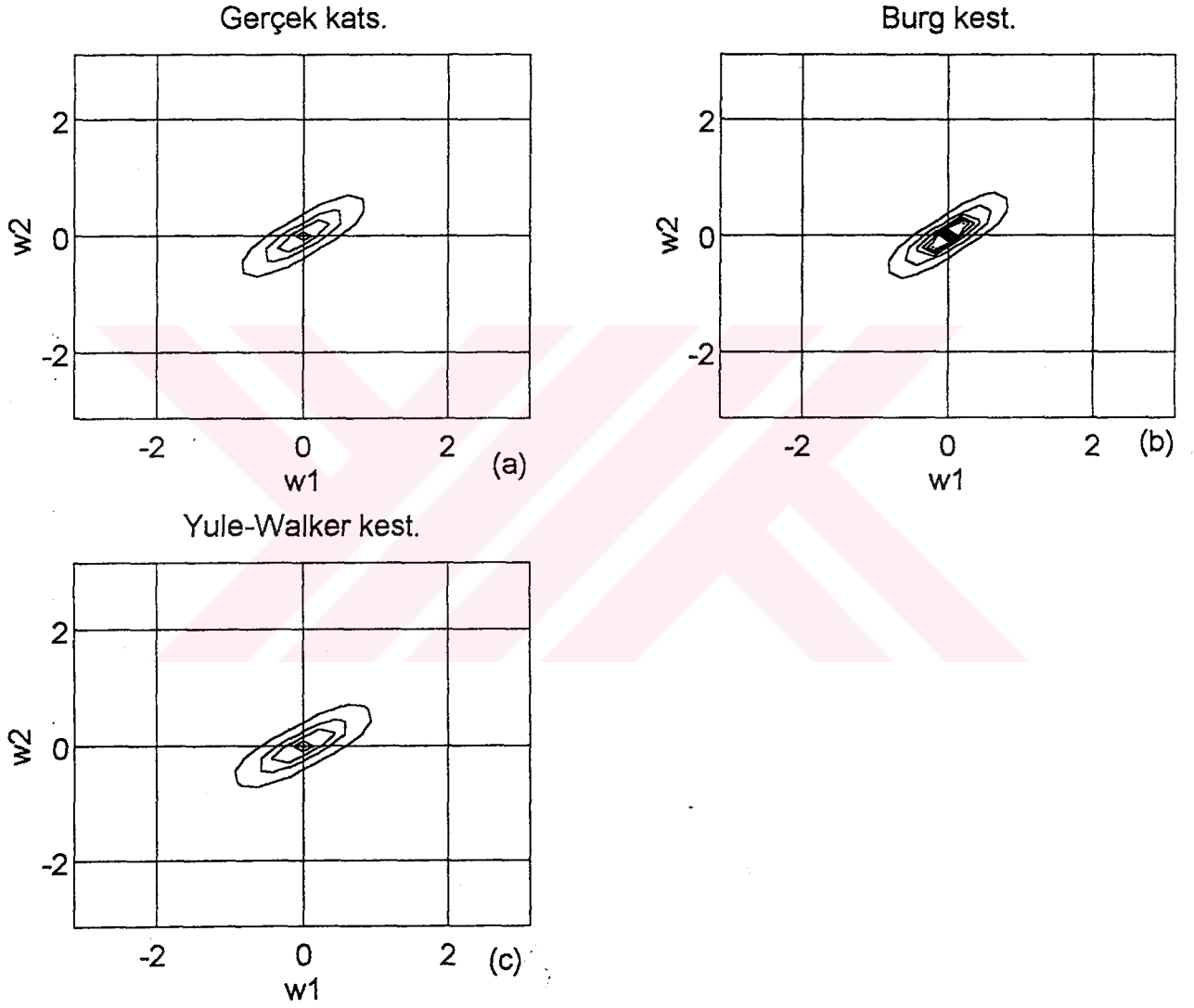
Şekil A.4.a İkinci dereceden sekiz destek noktalı birinci çeyrek düzlem için tespit edilen katsayılara göre AR-spektrum örnekleri . Veri boyutu: 10 x10



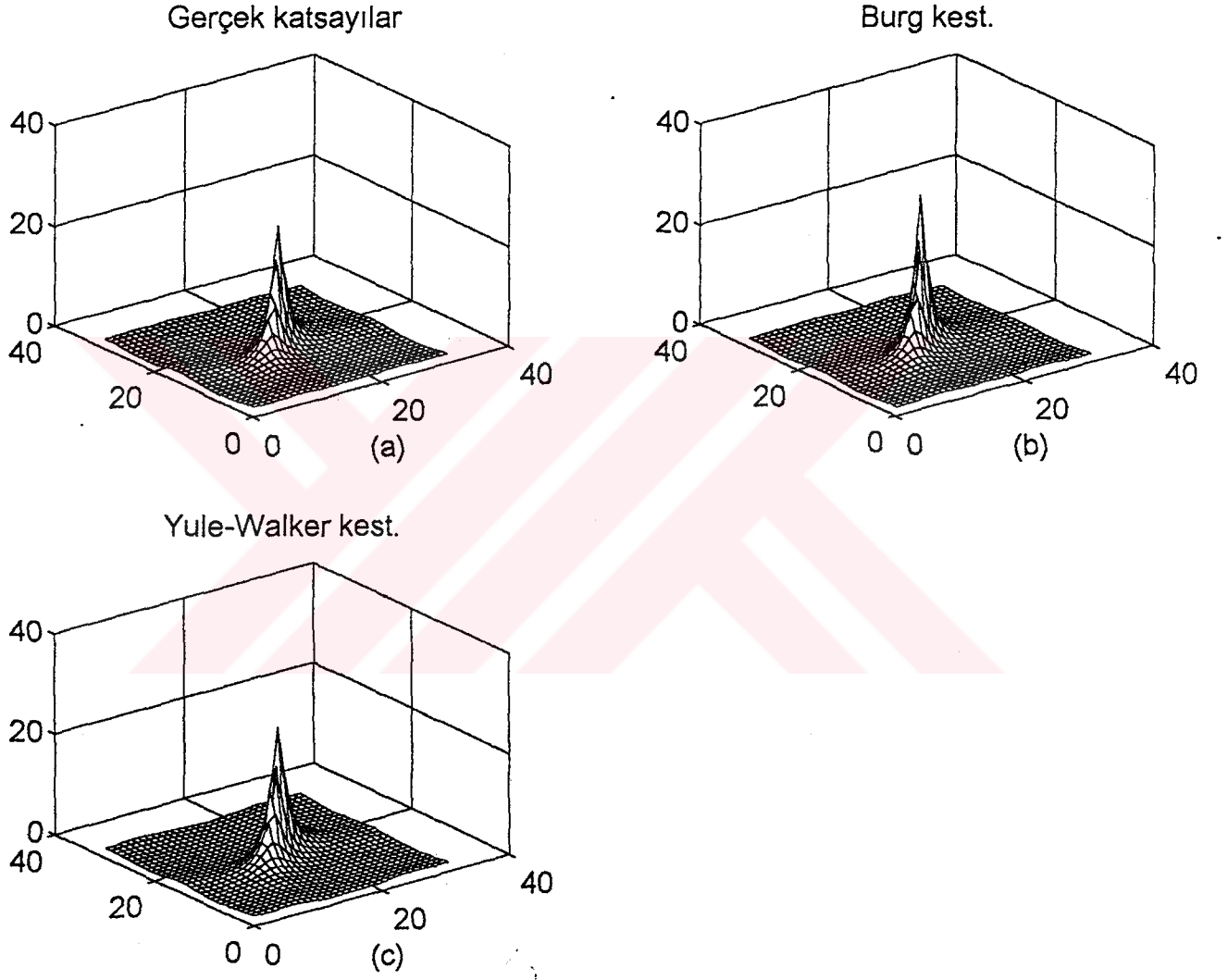
Şekil A.4.b İkinci dereceden sekiz destek noktalı birinci çeyrek düzlem için tespit edilen katsayılarla göre contour örnekleri . Veri boyutu: 10 x10



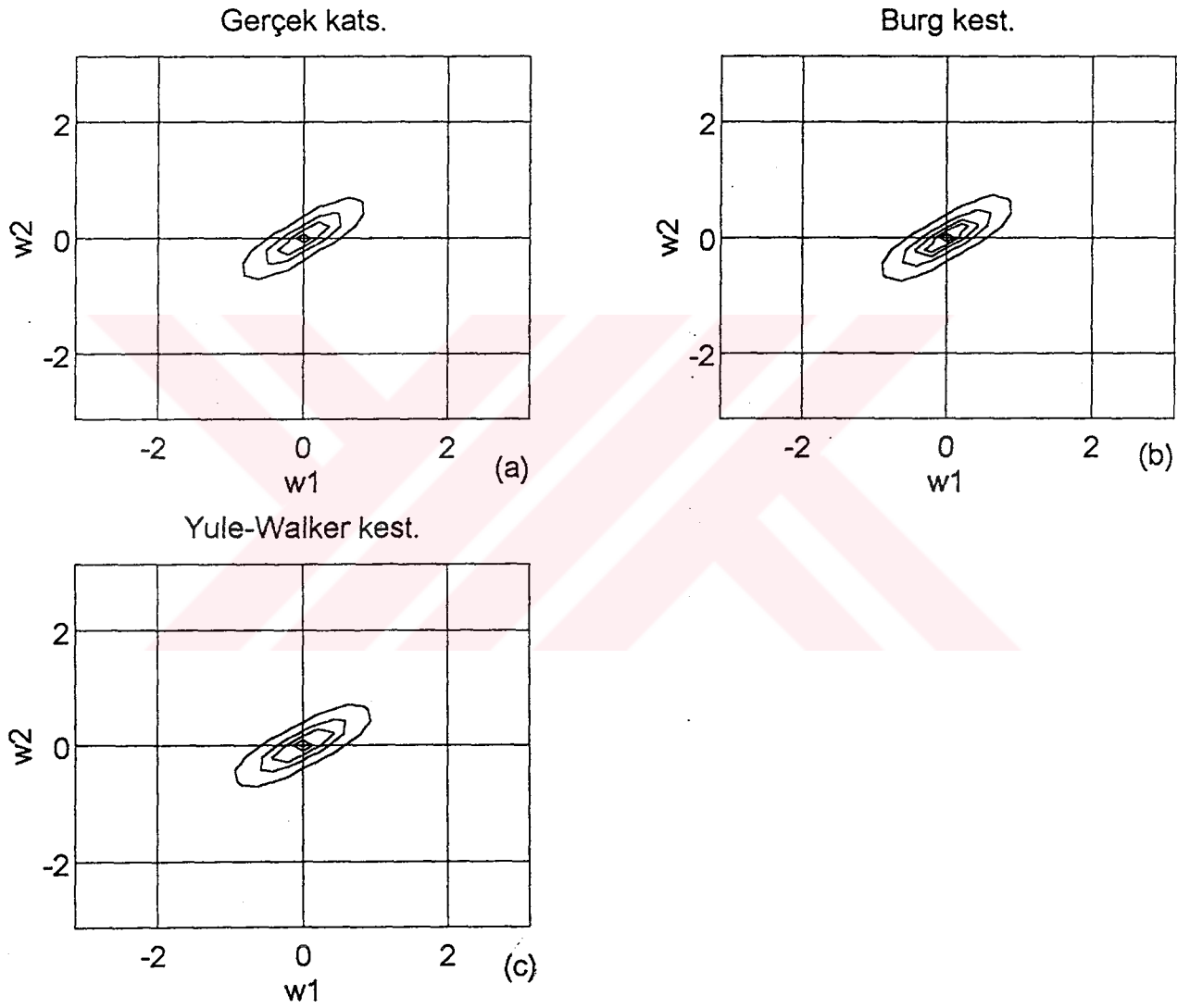
Şekil A.5.a İkinci dereceden sekiz destek noktalı birinci çeyrek düzlem için tespit edilen katsayılarla göre AR-spektrum örnekleri . Veri boyutu: 20 x20



Şekil A.5.b İkinci dereceden sekiz destek noktalı birinci çeyrek düzlem için tespit edilen katsayılarla göre contour örnekleri . Veri boyutu: 20 x20



Şekil A.6.a İkinci dereceden sekiz destek noktalı birinci çeyrek düzlem için tespit edilen katsayılar göre AR-spektrum örnekleri . Veri boyutu: 35 x 35



Şekil A.6.b İkinci dereceden sekiz destek noktalı birinci çeyrek düzlem için tespit edilen katsayılarla göre contour örnekleri . Veri boyutu: 35 x 35

ÖRNEK A.3. Üçüncü derece, birinci çeyrek düzlem

Üretilen AR veri alanına ait fark denklemi aşağıda verilen şekildedir.

$$y(k_1, k_2) = w(k_1, k_2) + a_1 y(k_1-1, k_2) + a_2 y(k_1-1, k_2-1) + a_3 y(k_1, k_2-1) + a_4 y(k_1, k_2-2) + a_5 y(k_1-1, k_2-2) + a_6 y(k_1-2, k_2-2) + a_7 y(k_1-2, k_2-1) + a_8 y(k_1-2, k_2) + a_9 y(k_1-3, k_2) + a_{10} y(k_1-3, k_2-1) + a_{11} y(k_1-3, k_2-2) + a_{12} y(k_1-3, k_2-3) + a_{13} y(k_1-2, k_2-3) + a_{14} y(k_1-1, k_2-3) + a_{15} y(k_1, k_2-3)$$

Gerçek katsayı vektörü ise aşağıda verilen şekilde seçilmiştir.

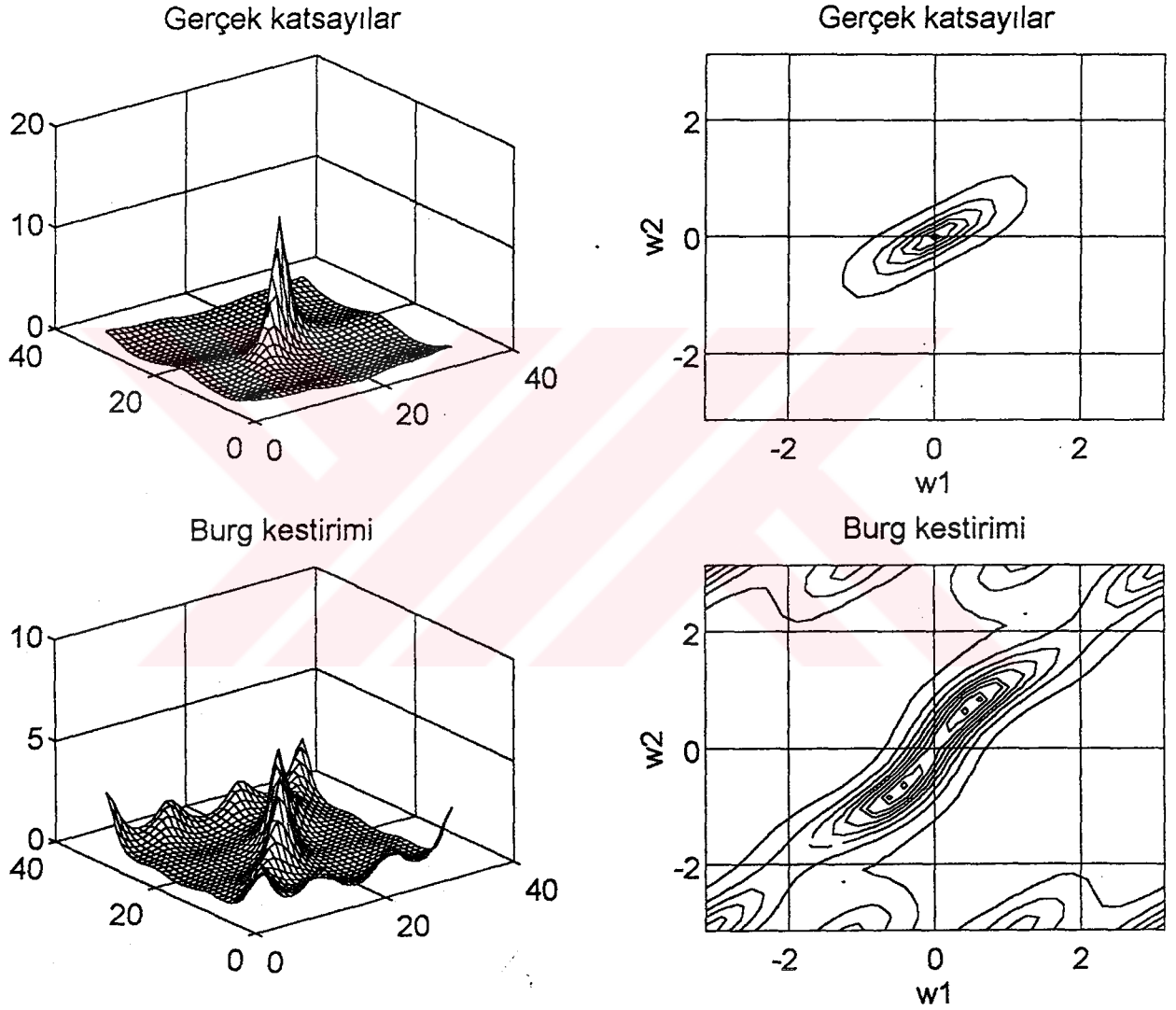
$$A = [-.013 \ -.18 \ -.08 \ -.15 \ -.045 \ -.075 \ -.108 \ -.033 \ -.0034 \ -.0031 \ -.0002 \ -.032 \ -.0034 \ -.0032 \ -.0005];$$

Filtrenin derecesi arttıkça düşük veri boyutlarına ilişkin kestirim verimliliğinin azaldığı gözlenmiştir.

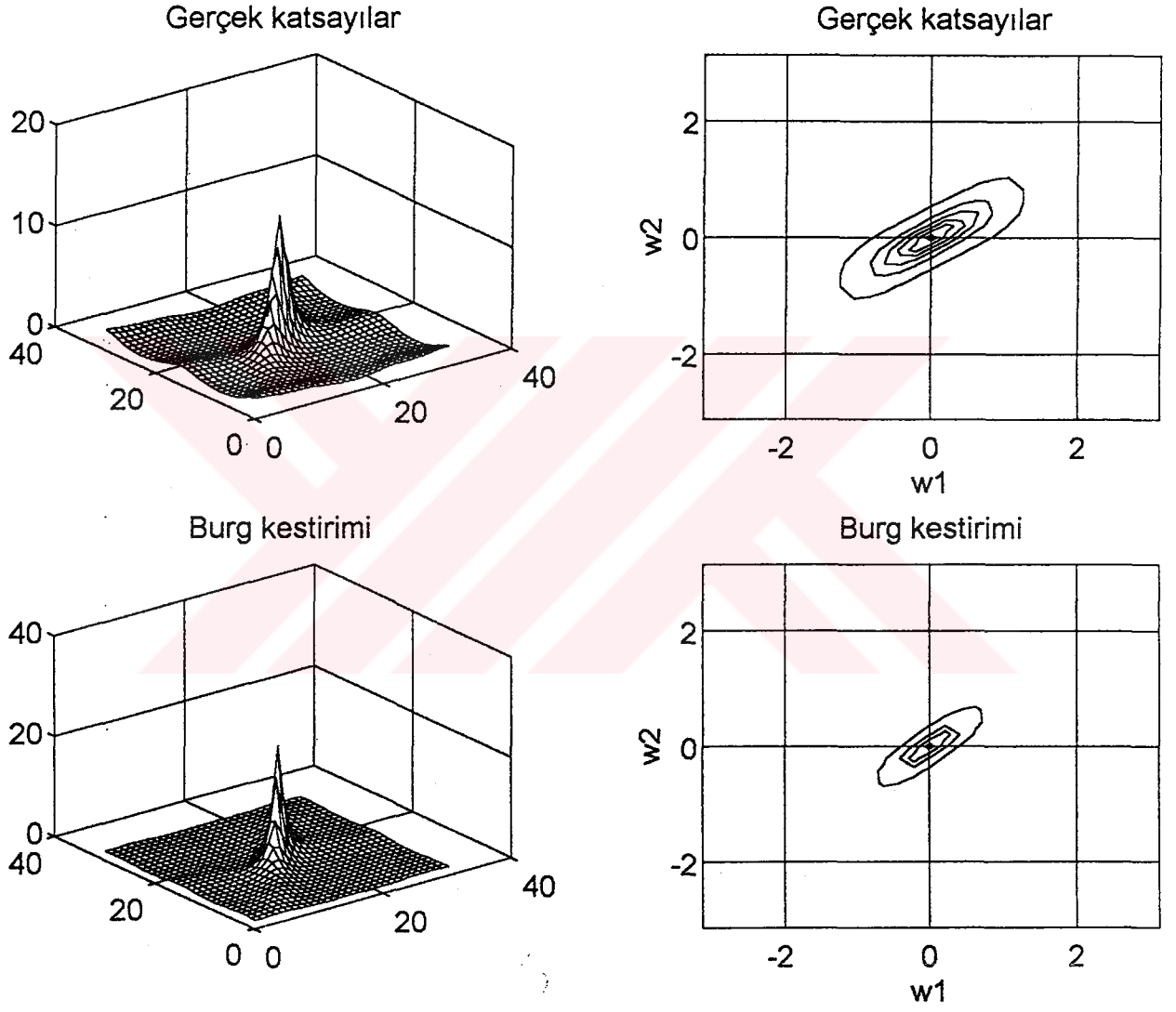
Şekil A.7 'de 20 x20 veri boyutu ve Şekil A.8 ' de ise 30 x 30 veri boyutu için yapılan kestirimler gösterilmiştir. Şekil A.9 'da ise 15 aşamadan oluşan 15 destek noktalı bir kestirimin her üç aşamasında bir frekans spektrumları çizdirilerek, gerçek spektruma yakınsama adımları gösterilmiştir. Örneğin 3. aşama için kestirim yanılğı süzgecinin aktarım işlevi:

$$A_0^{(3)}(z_1, z_2) = \frac{F_0^{(3)}(z_1, z_2)}{Y(z_1, z_2)} = \delta_0(z_1, z_2) \frac{F_0^{(3)}(z_1, z_2)}{F_0^{(0)}(z_1, z_2)}$$

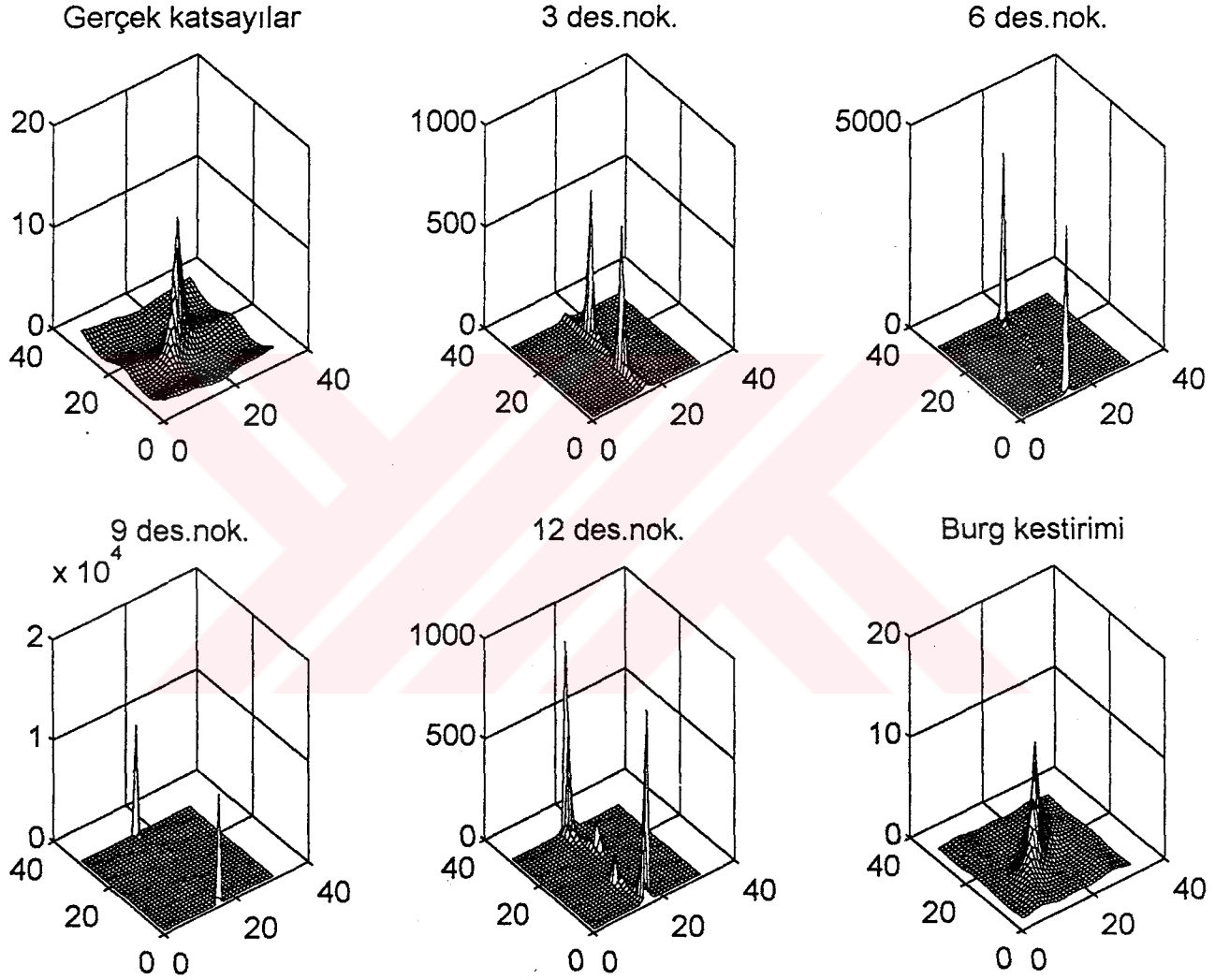
şeklinde verilmektedir. Bu şekil için veri boyutu 40 x 40 dır. Kestirimin 30 x 30 'luk veri boyutuna ilişkin duruma göre daha da iyileştiği gözlenmektedir.



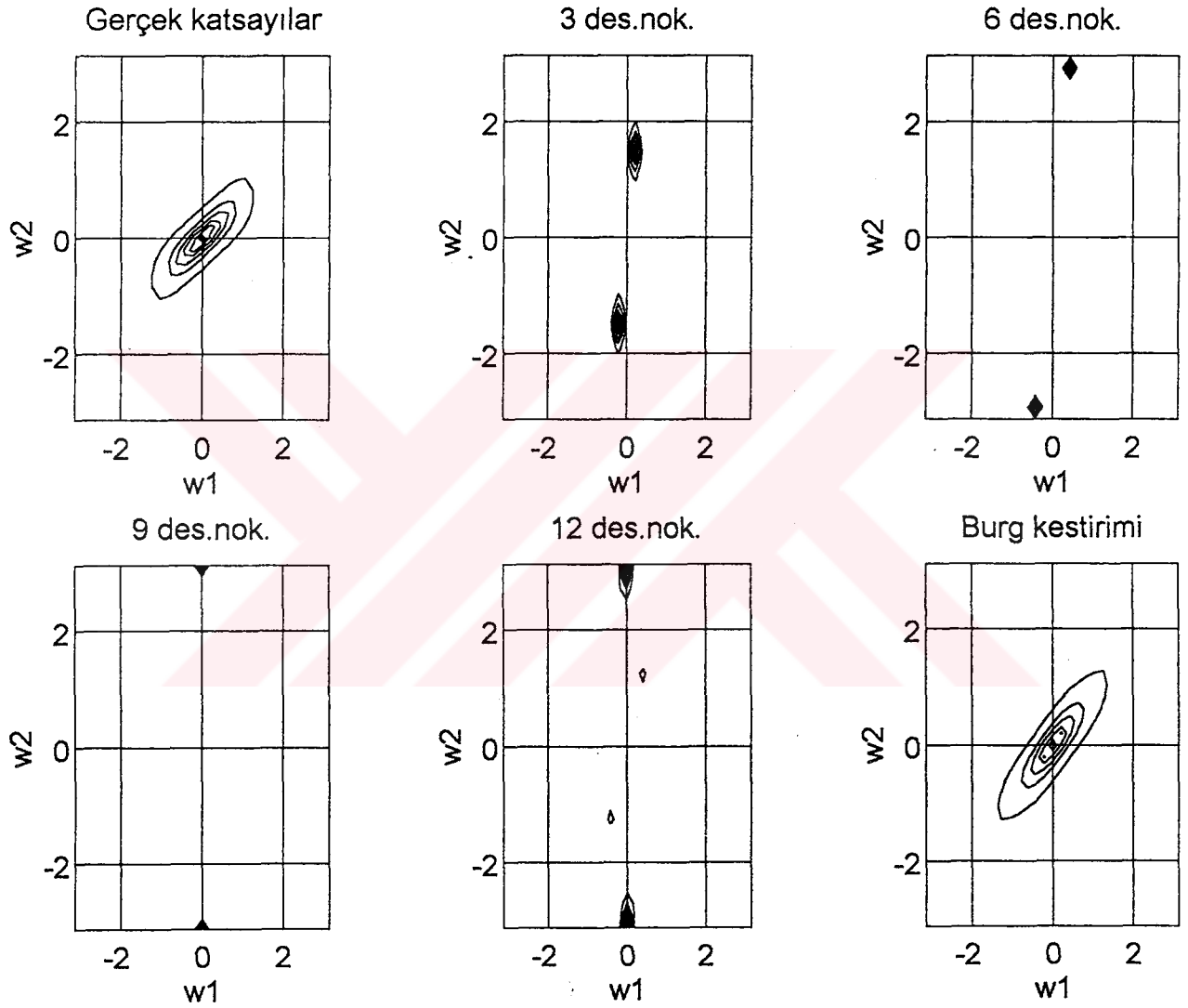
Şekil A.7 Üçüncü dereceden onbeş destek noktalı birinci çeyrek düzlem için tespit edilen katsayılara göre AR-spektrum örnekleri . Veri boyutu: 20 x 20



Şekil A.8 Üçüncü dereceden onbeş destek noktalı birinci çeyrek düzlem için tespit edilen katsayılarla göre AR-spektrum örnekleri . Veri boyutu: 30 x 30



Şekil A.8 Üçüncü dereceden onbeş destek noktalı birinci çeyrek düzlem için tespit edilen katsayılarla göre üç aşamalı AR-spektrum örnekleri . Veri boyutu: 40 x 40



Şekil A.8 Üçüncü dereceden onbeş destek noktalı birinci çeyrek düzlem için tespit edilen katsayılarla göre üç aşamalı contour örnekleri . Veri boyutu: 40 x 40

EK B. PROGRAM LİSTELERİ

Kafes süzgeçleri çözümü için geliştirilen ana programlar lat23, lat28, lat215 şeklinde isimlendirilmişlerdir. Lat2, iki boyutlu kafes süzgecini temsil etmektedir. Son iki rakam ise kestirim destek bölgesinde kullanılan nokta sayısını göstermektedir. Yapılan programlar, katsayıların klavyeden girilmesine de olanak sağlamaktadır. Diğer listelenen programlar yardımcı veya alt programlardır. Bütün alt programların işlevleri listelerde belirtilmiştir.



```
%*****KOKAT.m*****
```

```
%Hazırlayan: Nurşen YILDIZ
```

```
function A = kokat(y);
```

```
%Bu fonksiyon verilen bir "y" matrisinin r00,r01 ve r10 ilişki katsayılarını bulur.
```

```
[m,n]=size(y);  
B=y.*y;  
r00=sum(sum(B));  
    for i=1:m  
        for j=1:(n-1)  
            C(i,j)=y(i,j)*y(i,j+1);  
            r01=sum(sum(C));  
        end  
    end  
    for i=1:(m-1)  
        for j=1:n  
            D(i,j)=y(i,j)*y(i+1,j);  
            r10=sum(sum(D));  
        end  
    end  
    r00=r00/(m*n);r01=r01/(m*n);r10=r10/(m*n);  
    A=[r00 r01 r10];  
end
```

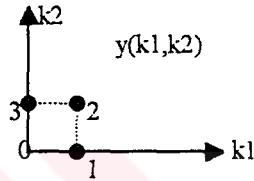
*****AR23.m*****

Hazırlayan : Nurşen YILDIZ

function y = ar23(VA,A);

%Bu program çeyrek düzlemde birinci derece, 3 destek noktalı kafes süzgeci
%için 2-B' lu Özbağılanımlı (Autoregressive) veri alanı üretir. Üretilecek AR
%veri alanına ait dizilim şekilde gösterilmiştir.

%
%
%
%
%
%
%
%
%



```
m=VA(1);n=VA(2);  
m = m+1;n = n+1;  
y = zeros(m,n);  
randn('seed',0);  
w = randn(m,n);  
y(1,1) = w(1,1);  
for k1=2:m  
    for k2 = 2:n  
        y(1,k2) = w(1,k2)+a3*y(1,k2-1);  
        y(k1,1) = w(k1,1)+a1*y(k1-1,1);  
    end  
end  
for k1=2:m  
    for k2=2:n  
        y(k1,k2) = w(k1,k2)+a1*y(k1-1,k2)+a2*y(k1-1,k2-1)+a3*y(k1,k2-1);  
    end  
end  
for p=1:m-1  
    for q=1:n-1  
        u(p,q) = y(p+1,q+1);  
    end  
end  
end
```

%Sınır değerlerin atılması
%Normal dağılımın üretilmesi
%İlk koşul verilmesi

%*****INI23.m*****

%Hazırlayan: Nurşen YILDIZ

function [F,B] = ini23(y);

%Bu alt program, 1. dereceden birinci çeyrek düzlemde 3 destek noktalı bir kafes
%süzgecinin birinci çözüm aşamasına ilişkin ileri ve geri yönde kestirim hata
%alanlarını hesaplar. Fonksiyonun giriş verisi üretilmiş olan AR veri alanıdır.
%F ileri yönde ve B ise geri yönde kestirim hata alanlarını göstermektedir.

```
[m,n] = size(y);
A = kokat(y); %İlişki katsayılarının hesaplanması
r00 = A(1);
r01 = A(2);
r10 = A(3);
yh = -r10/r00;
yv = -r01/r00;
for k2 = 1:n
    f01(1,k2) = y(1,k2);
    b11(1,k2) = yh*y(1,k2);
end
for k1=2:m
    for k2=1:n
        f01(k1,k2) = y(k1,k2)+yh*y(k1-1,k2);
        b11(k1,k2) = yh*y(k1,k2)+y(k1-1,k2);
    end
end
for k1 = 2:m
    f11(k1,1) = y(k1-1,1);
    b21(k1,1) = yv*y(k1-1,1);
end
for k1=2:m
    for k2=2:n
        f11(k1,k2) = y(k1-1,k2)+yv*y(k1-1,k2-1);
        f21(k1,k2) = y(k1-1,k2-1)+yh*y(k1,k2-1);
        b21(k1,k2) = yv*y(k1-1,k2)+y(k1-1,k2-1);
        b31(k1,k2) = yh*y(k1-1,k2-1)+y(k1,k2-1);
    end
end
for k2 = 2:n
    f21(1,k2) = yh*y(k1,k2-1);
    b31(1,k2) = y(k1,k2-1);
end

F=[f01 f11 f21];
B=[b11 b21 b31];
```



```
%*****IEP23.m*****
```

```
%Hazırlayan Nurşen YILDIZ
```

```
function [Ef,Eb] = iep23(F,B)
```

```
%Bu alt program, birinci dereceden üç destek noktalı kafes süzgecinin birinci  
%basamağına ilişkin ileri ve geri yönde kestirim hata alanları güçlerini hesaplar.
```

```
[x,y]=size(F);  
m = x;n = y/3;  
d = 0;  
a = 1-n;  
t = (3*n) - (n-1);  
    while a < t  
        a = a+n;  
        b = a+(n-1);  
        d = d+1;  
        f=F(:,a:b);  
        b = B(:,a:b);  
        Fi = f.*f;  
        Bi=b.*b;  
        FT=sum(sum(Fi));  
        BT=sum(sum(Bi));  
        Ef(d)=FT/(m*n);  
        Eb(d)=BT/(m*n);  
    end  
end
```

```
%*****SPEC23.m*****
```

```
%Hazırlayan: Nurşen YILDIZ
```

```
%Bu alt program, üç katsayı ağırlığına ilişkin frekans spektrumunu hesaplar.
```

```
function Pz = spec23(A);  
[w1,w2] = meshdom(-pi:pi/15:pi,-pi:pi/15:pi);  
ag1=1-A(1).*exp(-j.*w1)-A(2).*exp(-j.*w1).*exp(-j.*w2)-A(3).*exp(-j.*w2);  
A=abs(ag1);  
Py=A.*A;  
clear A;  
[m,n]=size(Py);  
Pz=ones(m,n)./Py;
```



|%*****LAT23.m*****|

%Hazırlayan: Nurşen YILDIZ

%%%%% MAIN %%%%%

%Bu program, birinci dereceden çeyrek düzlemde üç destek noktalı kafes süzgeci
%yöntemini kullanarak AR veri alanına ilişkin parametreleri kestirip,gerçek
%ve Yule-Walker frekans spektrumları ile karşılaştırır.

```
VA=input('Veri alanının boyutunu giriniz...[m,n] = ? ');
disp('Üretilcek olan AR veri alanına ait fark denklemi aşağıdadır :');
disp('y(k1,k2)=w(k1,k2)+a1*y(k1-1,k2)+a2*y(k1-1,k2-1)+a3*y(k1,k2-1)');
disp('Burada w(k1,k2) ortalaması "sıfır",varyansı "1" olan beyaz gürültüdür.')
disp('Katsayı matrisi A=[a1 a2 a3 ] biçimindedir.')
k=menu('Katsayı Seçimi','Default Değerler','Kats.klavyeden girilmesi');
    if k == 1
        A=[-0.4 .3 .15];
    else
        A=input('A= ');
    end
K=A;
[m,n] = size(y);
D = kokat(y);
yh = -D(3)/D(1);
yv = -D(2)/D(1);
f = zeros(m,n);
b = zeros(m,n);
asama = 0;
    while asama < 3
        asama = asama+1
        inc = - (n-1);
        bastop = 4 - asama;
        F=zeros(m,n*bastop);
        B = F;
        Eft = zeros(1,bastop);
        Ebt = Eft;
        YF=zeros(1,bastop);
        YB=YF;
        basamak=0;
        A=ones((asama+1),bastop);
        G=A;
```

```
while basamak < bastop
    basamak = basamak+1;
    if asama > 1
        inc = inc+(m-1);
        t = FS(:,(basamak+inc):(basamak+inc+m-1));
        u = BS(:,(basamak+inc+m):(basamak+inc+(2*m)-1));
        i = t.*u;
        D = um(sum(Di));D=D/(m*n);    %Çapraz ilişkilerin hesaplanması
        yf = -D/EFT(basamak);          %İleri yansıma katsayıları
        yb = -D/EBT(basamak+1);        %Geri yansıma katsayıları
        YF(basamak) = yf;
        YB(basamak) = yb;
    for k1=1:m
        for k2=1:n
            f(k1,k2)=t(k1,k2)+yb*u(k1,k2); %İleri yönde hata alanları
            b(k1,k2)=yf*t(k1,k2)+u(k1,k2); %Geri yönde hata alanları
        end
    end
    F(:,(basamak+inc):(basamak+inc+m-1)) = f;
    B(:,(basamak+inc):(basamak+inc+m-1)) = b;
    Ef = f.*f;Ef = sum(sum(Ef));Ef=Ef/(m*n);
    Bf=b.*b;Bf=sum(sum(Bf));Bf=Bf/(m*n);
    Eft(basamak)=Ef;Ebt(basamak)=Bf;
    X=Ae(:,basamak);
    Xi=yf.*X;
    Z=Ge(:,basamak+1);
    Zi=yb.*Z ;
    Ai=ones(asama+1,1);
    Gi=Ai;
    for q =2:asama
        Ai(q)=X(q)+Zi(q-1);            %İleri yönde katsayı ağırlıkları
        Gi(q)=Xi(q)+Z(q-1);            % Geri yönde katsayı ağırlıkları
    end
    Ai(asama+1)=Zi(asama);
    Gi(1)=Xi(1);
    A(:,basamak)=Ai;
    G(:,basamak)=Gi;

    if basamak == bastop
        Ae=A;
        Ge=G;
    end
else
    basamak=bastop;
    [F,B]=ini23(y);                    %İlk koşul hata alanları
    [Eft,Ebt]=iep23(F,B);              %İlk koşul hata güçleri
    Ae=[ones(1,3);yh yv yh]
    Ge=[yh yv yh ;ones(1,3)]
end
```

```
disp('Ae=[a01 a11 a21] şeklindedir.')
disp('Ge=[g11 g21 g31] şeklindedir.')
disp('yh= ')
disp(yh)
disp('yv= ')
disp(yv)
end
end
FS = F;
BS = B;
EFT = Eft
EBT = Ebt
disp('YF= '); disp(YF);
disp('YB= ');disp(YB) ;
disp('A= ');disp(A);
disp('G= ');disp(G);
end
end
A=A(2:4);
Pz1 = spec23(A);
figure(1);
subplot(222);
mesh(Pz1)
title('Hesaplanan -Burg -kats.')
grid
subplot(221);
A= -K;
Pz2=spec23(A);
mesh(Pz2)
title('Gerçek kats.')
grid
subplot(223);
if VA==[10 10]
    A=[0.3034 -0.0574 -0.2143];
elseif VA==[20 20]
    A=[0.3158 -0.2646 -0.1654];
elseif VA==[35 35]
    A=[0.3426 -0.3167 -0.1432];
else
    break
end
Pz3 =spec23(A);
subplot(213);
mesh(Pz3);
title('Yule-Walk. kest.')
grid
figure(2);
```

```
subplot(212);  
X = [-pi:pi/15:pi]; Y = X;  
contour(X,Y,Pz1);  
xlabel('w1');ylabel('w2');  
title('Hesaplanan -Burg-kats.').;  
grid  
subplot(211);  
contour(X,Y,Pz2);  
title('Gerçek kats.').;  
xlabel('w1');ylabel('w2');  
grid  
subplot(213);  
contour(X,Y,Pz3);  
title('Yule-Walker kestirimi ');  
xlabel('w1');ylabel('w2');  
grid  
end
```

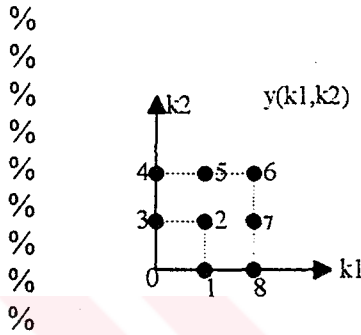


%***** AR28 .m*****

%Hazırlayan : Nurşen YILDIZ

function u = ar28(VA,A)

%Bu program çeyrek düzlemde ikinci derece sekiz destek noktası için iki boyutlu
%Özbağılanımlı (=Autoregressive) veri alanı üretir. Üretilen AR veri alanına ilişkin
%dizilim şeklindeki gibidir.



m = VA(1); n = VA(2);
m = m+3;n = n+3;
y = zeros(m,n);
disp('İki boyutlu beyaz gürültü üretiliyor ...');
randn('seed',0);
w =randn(m,n);
disp('Veri üretiliyor , lütfen bekleyiniz...')
y(1,1) = w(1,1);
y(1,2) = w(1,2)+A(3)*y(1,1);
y(2,1) = w(2,1)+A(1)*y(1,1);
y(2,2) = w(2,2)+A(1)*y(1,2)+A(2)*y(1,1)+A(3)*y(2,1);
for k2=3:n
 y(1,k2) = w(1,k2)+A(3)*y(1,k2-1)+A(4)*y(1,k2-2);
 y(2,k2) = w(2,k2) +A(1)*y(1,k2)+A(2)*y(1,k2-1)+A(3)*y(2,k2-1);
 y(2,k2) = y(2,k2)+A(4)*y(2,k2-2)+A(5)*y(1,k2-2);
end
for k1=3:m
 y(k1,1) = w(k1,1)+A(1)*y(k1-1,1)+A(8)*y(k1-2,1);
 y(k1,2) = w(k1,2) + A(1)* y(k1-1,2)+A(2)*y(k1-1,1) +A(3)*y(k1,1);
 y(k1,2) = y(k1,2)+ A(7)*y(k1-2,1)+A(8)*y(k1-2,2);
end
for k1=3:m
 for k2=3:n
 y(k1,k2) =
w(k1,k2)+A(1)*y(k1-1,k2)+A(2)*y(k1-1,k2-1)+A(3)*y(k1,k2-1);
 y(k1,k2) = y(k1,k2)+A(4)*y(k1,k2-2) +A(5)*y(k1-1,k2-2);

%Sınır değerlerin atılması
%İşlem hızlandırma
%İlk koşul verilmesi

```
        y(k1,k2) = y(k1,k2)+A(6)*y(k1-2,k2-2)+A(7)*y(k1-2,k2-1);  
        y(k1,k2) = y(k1,k2)+ A(8)*y(k1-2,k2) ;  
    end  
end  
  
for i=1:m-3  
    for j=1:n-3  
        u(i,j) = y(i+3,j+3);  
    end  
end
```



%*****INI28.m*****

%Hazırlayan: Nurşen YILDIZ

function [F,B] = ini28(y);

%Bu alt program, 2. dereceden birinci çeyrek düzlemde 8 destek noktalı bir kafes
%süzgecinin birinci çözüm aşamasına ilişkin ileri ve geri yönde kestirim hata
%alanlarını hesaplar. Fonksiyonun giriş verisi üretilmiş olan AR veri alanıdır.
%F ileri yönde ve B ise geri yönde kestirim hata alanlarını göstermektedir.

```
[m,n] = size(y);
A = kokat(y); %r00,r01,r10 ilişkilerinin hesaplanması
r00 = A(1);
r01 = A(2);
r10 = A(3);
yh = -r10/r00;
yv = -r01/r00;
for k2 = 1:n
    f01(1,k2) = y(1,k2);
    b11(1,k2) = yh*y(1,k2);
end
for k1 = 2:m
    for k2 = 1:n
        f01(k1,k2) = y(k1,k2)+yh*y(k1-1,k2);
        b11(k1,k2) = yh*y(k1,k2)+y(k1-1,k2);
    end
end
for k1 = 2:m
    f11(k1,1) = y(k1-1,1);
    b21(k1,1) = yv*y(k1-1,1);
end
for k1 = 2:m
    for k2 = 2:n
        f11(k1,k2) = y(k1-1,k2)+yv*y(k1-1,k2-1);
        f21(k1,k2) = y(k1-1,k2-1)+yh*y(k1,k2-1);
        b21(k1,k2) = yv*y(k1-1,k2)+y(k1-1,k2-1);
        b31(k1,k2) = yh*y(k1-1,k2-1)+y(k1,k2-1);
    end
end
for k2 = 2:n
    f21(1,k2) = yh*y(1,k2-1);
    b31(1,k2) = y(1,k2-1);
end
for k1 = 1:m
    f31(k1,2) = y(k1,1);
    b41(k1,2) = yv*y(k1,1);
```

```
end
for k1 = 1:m
    for k2 = 3:n
        f31(k1,k2) = y(k1,k2-1)+yv*y(k1,k2-2);
        b41(k1,k2) = yv*y(k1,k2-1)+y(k1,k2-2);
    end
end
for k1 = 2:m
    for k2 = 3:n
        f41(k1,k2) = y(k1,k2-2)+yh*y(k1-1,k2-2);
        b51(k1,k2) = yh*y(k1,k2-2)+y(k1-1,k2-2);
    end
end
for k2 = 3:n
    f41(1,k2) = y(1,k2-2);
    b51(1,k2) = yh*y(1,k2-2);
    f51(2,k2) = y(1,k2-2);
    b61(2,k2) = yh*y(1,k2-2);
end
for k1 = 3:m
    f61(k1,2) = yv*y(k1-2,1);
    b71(k1,2) = y(k1-2,1);
    f71(k1,1) = yv*y(k1-2,1);
    b81(k1,1) = y(k1-2,1);
end
for k1 = 3:m
    for k2 = 3:n
        f51(k1,k2) = y(k1-1,k2-2)+yh*y(k1-2,k2-2);
        b61(k1,k2) = yh*y(k1-1,k2-2)+y(k1-2,k2-2);
        f61(k1,k2) = y(k1-2,k2-2)+yv*y(k1-2,k2-1);
        b71(k1,k2) = yv*y(k1-2,k2-2)+y(k1-2,k2-1);
    end
end
for k1 = 3:m
    for k2 = 2:n
        f71(k1,k2) = y(k1-2,k2-1)+yv*y(k1-2,k2);
        b81(k1,k2) = yv*y(k1-2,k2-1)+y(k1-2,k2);
    end
end
F = [f01 f11 f21 f31 f41 f51 f61 f71];
B = [b11 b21 b31 b41 b51 b61 b71 b81];
```

```
%*****IEP28.m*****
```

```
%Hazırlayan: Nurşen YILDIZ
```

```
function [Ef,Eb] = iep28(F,B)
```

```
%Bu alt program " lat28 " programının birinci aşama, birinci basamağına ilişkin  
%ileri ve geri yönde kestirim hata alanları güçlerini hesaplar.
```

```
[x,y] = size(F);  
m = x; n = y/8;  
d = 0;  
a = 1-n;  
t = (8*n) - (n-1);  
while a < t  
    a = a+n;  
    b = a+(n-1);  
    d = d+1;  
    f = F(:,a:b);  
    b = B(:,a:b);  
    Fi = f.*f;  
    Bi = b.*b;  
    FT = sum(sum(Fi));  
    BT = sum(sum(Bi));  
    Ef(d) = FT/(m*n);  
    Eb(d) = BT/(m*n);  
end  
end
```

```
%*****SPEC28.m*****
```

```
%Hazırlayan. Nurşen YILDIZ
```

```
%Bu alt program sekiz katsayı ağırlığına ilişkin frekans spektrumunu hesaplar.
```

```
function Pz = spec28(A)
```

```
[w1,w2] = meshdom(-pi:pi/15:pi,-pi:pi/15:pi);  
ag1=1-A(1).*exp(-j.*w1)-A(2).*exp(-j.*w1).*exp(-j.*w2)-A(3).*exp(-j.*w2);  
ag2= -A(4).*exp(-j*2.*w2)-a5.*exp(-j.*w1).*exp(-j*2.*w2);  
ag3= -a6.*exp(-j*2.*w1) .*exp(-j*2.*w2) -a7.*exp(-j*2.*w1).*exp(-j.*w2);  
ag4= -a8.*exp(-j*2.*w2);  
A=abs(ag1+ag2+ag3+ag4);  
Py=A.*A;  
clear A;  
[m,n]=size(Py);  
Pz=ones(m,n)./Py;
```

%*****LAT28.m*****

%Hazırlayan: Nurşen YILDIZ

%%%%%%%% MAIN %%%%%%%%%

%Bu program, ikinci dereceden çeyrek düzlemde sekiz destek noktalı kafes süzgeci
%yöntemini kullanarak AR veri alanına ilişkin parametreleri kestirip, gerçek
%ve Yule-Walker frekans spektrumları ile karşılaştırır.

```
VA=input('Veri alanının boyutunu giriniz...[m,n] = ? ');
disp('Üretilcek olan AR veri alanına ait fark denklemi:');
disp('y(k1,k2) = w(k1,k2)+a1*y(k1-1,k2)+a2*y(k1-1,k2-1)+a3*y(k1,k2-1)');
disp(' +a4*y(k1,k2-2)+a5*y(k1-1,k2-2)+a6*y(k1-2,k2-2)+a7*y(k1-2,k2-1)');
disp(' +a8*y(k1-2,k2)');
disp('Burada w(k1,k2) ortalaması "sıfır", varyansı "bir" olan beyaz gürültüdür.')
disp('Katsayı matrisi A=[a1 a2 a3 a4 a5 a6 a7 a8] biçimindedir.')
k=menu('Katsayı matrisinin seçimi','Default değerler','Kats.klavyeden girilmesi');
if k==1
    A=[-.13 -.18 -.08 -.15 -.045 -.075 -.108 -.033];
else
    A=input('A=')
end
K=A;
y=ar28(VA,A);
[m,n] = size(y);
D = kokat(y);
r00=D(1);
r01=D(2);
r10=D(3);
yh = -r10/r00;
yv = -r01/r00;
f = zeros(m,n);
b = zeros(m,n);
asama = 0;
while asama < 8
    asama=asama+1
    inc=-(n-1);
    bastop=9-asama;
    F=zeros(m,n*bastop);
    B = F;
    Eft = zeros(1,bastop);
    Ebt = Eft;
    YF=zeros(1,bastop);
    YB=YF;
```

```
basamak=0;
A=ones((asama+1),bastop);
G=A;
while basamak < bastop
    basamak=basamak+1;
    if asama > 1
        inc=inc+(m-1);
        t=FS(:,(basamak+inc):(basamak+inc+m-1));
        u=BS(:,(basamak+inc+m):(basamak+inc+(2*m)-1));
        Di=t.*u; %Çapraz ilişkilerin hesaplanması
        D=sum(sum(Di));D=D/(m*n);
        yf = -D/EFT(basamak); %İleri yansıma katsayıları
        yb = -D/EBT(basamak+1); %Geri yansıma katsayıları
        YF(basamak) = yf;
        YB(basamak) = yb;
        for k1=1:m
            for k2=1:n
                f(k1,k2)=t(k1,k2)+yb*u(k1,k2); %İleri yönde hata alanları
                b(k1,k2)=yf*t(k1,k2)+u(k1,k2); %Geri yönde hata alanları
            end
        end
        F(:,(basamak+inc):(basamak+inc+m-1))=f;
        B(:,(basamak+inc):(basamak+inc+m-1))=b;
        Ef=f.*f;Ef=sum(sum(Ef));Ef=Ef/(m*n);
        Bf=b.*b;Bf=sum(sum(Bf));Bf=Bf/(m*n);
        Eft(basamak)=Ef;Ebt(basamak)=Bf;
        X=Ae(:,basamak);
        Xi=yf.*X;
        Z=Ge(:,basamak+1);
        Zi=yb.*Z;
        Ai=ones(asama+1,1);
        Gi=Ai;
        for q=2:asama
            Ai(q)=X(q)+Zi(q-1); %İleri yönde katsayı ağırlıkları
            Gi(q)=Xi(q)+Z(q-1); %Geri yönde katsayı ağırlıkları
        end
        Ai(asama+1)=Zi(asama);
        Gi(1)=Xi(1);
        A(:,basamak)=Ai;
        G(:,basamak)=Gi;

        if basamak == bastop
            Ae=A;
            Ge=G;
        end
else
    basamak=bastop;
    [F,B]=ini28(y); %İlk koşul hata alanları
```

```
[Eft,Ebt]=iep28(F,B);           %İlk koşul hata güçleri
Ae=[ones(1,8);yh yv yh yv yh yh yv yv]
disp('Ae=[a01 a11 a21 a31 a41 a51 a61 a71] şeklindedir.')
Ge=[yh yv yh yv yh yh yv yv;ones(1,8)]
disp('Ge=[g11 g21 g31 g41 a51 g61 g71 g81] şeklindedir.')
disp('yh= ');disp(yh);
disp('yv= ');disp(yv);
end
end
    FS = F;
    BS = B;
    EFT = Eft
    EBT = Ebt
    disp('YF= ');disp(YF);
    disp('YB= ');disp(YB);
    disp('A= ');disp(A);
    disp('G= ');disp(G);
end
end
end
A=A(2:9);
Pz1=spec28(A);
figure(1);
subplot(222);
mesh(Pz1)
title('Hesaplanan-Burg- kats.')
grid
A= -K;
Pz2=spec28(A);
subplot(221)
mesh(Pz2);
title('Gerçek kats.')
grid
    if VA==[10 10]
        A=[.1105 .1709 .286 .0567 .0488 -.0295 .0672 -.0567];
    elseif VA==[20 20]
        A=[.0881 .2599 .1404 .1789 .0269 .0178 .1059 -.0123];
    elseif VA==[35 35]
        A=[.111 .1705 .0708 .1702 .0358 .0946 .1289 .0249];
    else
        break
    end
Pz3=spec28(A);
subplot(223);
mesh(Pz3)
title('Yule-Walker kest.')
grid
figure(2);
```

```
X=[-pi:pi/15:pi]; Y=X;
subplot(222);
contour(X,Y,Pz1);
xlabel('w1');ylabel('w2');
title('Hesaplanan-Burg- kats.')
grid
subplot(221);
contour(X,Y,Pz2);
xlabel('w1');ylabel('w2');
title('Gerçek kats.')
grid
subplot(223);
contour(X,Y,Pz3);
xlabel('w1');ylabel('w2');
title('Yule-Walker kest.')
grid
```



%*****AR215.m*****

%Hazırlayan: Nurşen YILDIZ

function u=ar215(VA,A)

% Bu program birinci çeyrek düzlemde, 3. derece 15 destek noktası için 2-B 'lu
%Özbağımlı (Autoregressive = AR) veri alanı üretir. Üretilcek olan AR veri
%alanına ait dizilim şekilde gösterilmiştir.

%

%

%

%

%

%

%

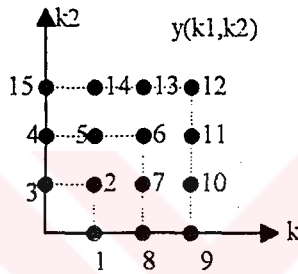
%

%

%

%

%



m = VA(1);n = VA(2);

m = m+3;

n = n+3;

y = zeros(m,n);

disp('İki boyutlu beyaz gürültü üretiliyor ...');

randn('seed',0);

w = randn(m,n)

disp('Veri üretiliyor , lütfen bekleyiniz...')

y(1,1) = w(1,1);

y(1,2) = w(1,2)+A(3)*y(1,1);

y(1,3) = w(1,3)+A(3)*y(1,2)+A(4)*y(1,1);

y(2,1) = w(2,1)+A(1)*y(1,1);

y(2,2) = w(2,2)+A(1)*y(1,2)+A(2)*y(1,1)+A(3)*y(2,1);

y(2,3) = w(2,3)+A(1)*y(1,3)+A(2)*y(1,2)+A(3)*y(2,2)+A(4)*y(2,1)+A(5)*y(1,1);

y(3,1) = w(3,1)+A(1)*y(2,1)+A(8)*y(1,1);

y(3,2) = w(3,2)+A(1)*y(2,2)+A(2)*y(2,1)+A(3)*y(3,1)+A(7)*y(1,1)+A(8)*y(1,2);

y(3,3) = w(3,3)+A(1)*y(2,3)+A(2)*y(2,2)+A(3)*y(3,2)+A(4)*y(3,1)+A(5)*y(2,1);

y(3,3) = y(3,3)+A(6)*y(1,1)+A(7)*y(1,2)+A(8)*y(1,3);

%Sınır değerlerin atılması

%İşlem hızlandırma

```
for k1=4:m
    y(k1,1) = w(k1,1)+A(1)*y(k1-1,1)+A(8)*y(k1-2,1)+A(9)*y(k1-3,1);
    y(k1,2) = w(k1,2)+A(1)*y(k1-1,2)+A(2)*y(k1-1,1) +A(3)*y(k1,1);
    y(k1,2)= y(k1,2)+A(7)*y(k1-2,1)+A(8)*y(k1-2,2)+A(9)*y(k1-3,2);
    y(k1,2) = y(k1,2)+A(10)*y(k1-3,1);
    y(k1,3) =w(k1,3)+A(1)*y(k1-1,3)+A(2)*y(k1-1,2)+A(3)*y(k1,2)+A(4)*y(k1,1);
    y(k1,3) = y(k1,3)+A(5)*y(k1-1,1)+A(6)*y(k1-2,1)+A(7)*y(k1-2,2);
    y(k1,3) = y(k1,3)+A(8) *y(k1-2,3) +A(9)*y(k1-3,3) +A(10)*y(k1-3,2);
    y(k1,3) = y(k1,3)+A(11)*y(k1-3,1);
end

for k2=4:n
    y(1,k2) = w(1,k2)+A(3)*y(1,k2-1)+A(4)*y(1,k2-2)+A(15)*y(1,k2-3);
    y(2,k2) = w(2,k2)+A(1)*y(1,k2)+A(2)*y(1,k2-1)+A(3)*y(2,k2-1);
    y(2,k2) = y(2,k2)+A(4)*y(2,k2-2)+A(5)*y(1,k2-2) +A(14)*y(1,k2-3);
    y(2,k2) = y(2,k2)+A(15)*y(2,k2-3);
    y(3,k2) = w(3,k2)+A(1)*y(2,k2) +A(2)*y(2,k2-1)+A(3)*y(3,k2-1);
    y(3,k2) = y(3,k2)+(4)*y(3,k2-2) +A(5)*y(2,k2-2) +A(6)*y(1,k2-2);
    y(3,k2) = y(3,k2)+A(7)*y(1,k2-1)+A(8)*y(1,k2)+A(13)*y(1,k2-3);
    y(3,k2) = y(3,k2) +A(14)*y(2,k2-3)+A(15)*y(3,k2-3);
end

for k1=4:m
    for k2=4:n
        y(k1,k2) = w(k1,k2)+A(1)*y(k1-1,k2)+A(2)*y(k1-1,k2-1)+A(3)*y(k1,k2-1);
        y(k1,k2) = y(k1,k2)+A(4)*y(k1,k2-2)+A(5)*y(k1-1,k2-2)+A(6)*y(k1-2,k2-2)
        y(k1,k2) = y(k1,k2)+A(7)*y(k1-2,k2-1)+A(8)*y(k1-2,k2)+A(9)*y(k1-3,k2);
        y(k1,k2) = y(k1,k2)+A(10)*y(k1-3,k2-1)+A(11)*y(k1-3,k2-2);
        y(k1,k2) = y(k1,k2)+A(12)*y(k1-3,k2-3)+A(13)*y(k1-2,k2-3);
        y(k1,k2) = y(k1,k2)+A(14)*y(k1-1,k2-3)+A(15)*y(k1,k2-3);
    end
end

for i=1:m-3
    for j=1:n-3
        u(i,j)=y(i+3,j+3);
    end
end
```

%*****INI215.m*****

%Hazırlayan: Nurşen YILDIZ

%Bu alt program 3.dereceden birinci çeyrek düzlemde 15 destek noktalı bir kafes
%süzgecinin birinci çözüm aşamasına ilişkin ileri ve geri yönde kestirim hata
%alanlarını hesaplar. Fonksiyonun giriş verisi üretilmiş olan AR veri alanıdır. F ileri
%yönde, B ise geri yönde kestirim hata alanlarını göstermektedir.

function [F,B] = ini215(y);

```
[m,n] = size(y);
A = kokat(y);
r00 = A(1);
r01 = A(2);
r10 = A(3);
yh = -r10/r00;
yv = -r01/r00;
for k2 = 1:n
    f01(1,k2) = y(1,k2);
    b11(1,k2) = yh*y(1,k2);
    f81(3,k2) = y(1,k2);
    b91(3,k2) = yh*y(1,k2);
end
for k1=2:m
    for k2=1:n
        f01(k1,k2) = y(k1,k2)+yh*y(k1-1,k2);
        b11(k1,k2) = yh*y(k1,k2)+y(k1-1,k2);
    end
end
for k1=2:m
    f11(k1,1) = y(k1-1,1);
    b21(k1,1)=yv*y(k1-1,1);
end
for k1=2:m
    for k2=2:n
        f11(k1,k2) = y(k1-1,k2)+yv*y(k1-1,k2-1);
        f21(k1,k2) = y(k1-1,k2-1)+yh*y(k1,k2-1);
        b21(k1,k2)=yv*y(k1-1,k2)+y(k1-1,k2-1);
        b31(k1,k2)=yh*y(k1-1,k2-1)+y(k1,k2-1);
    end
end
for k2 = 2:n
    f21(1,k2) = yh*y(k1,k2-1);
    b31(1,k2) = y(k1,k2-1);
end
```

```
for k1=1:m
    f31(k1,2) = y(k1,k2-1);
    b41(k1,2) = yv*y(k1,k2-1);
end
for k1=1:m
    for k2 = 3:n
        f31(k1,k2) = y(k1,k2-1)+yv*y(k1,k2-2);
        b41(k1,k2) = yv*y(k1,k2-2)+y(k1,k2-1);
    end
end

for k1 = 2:m
    for k2 = 3:n
        f41(k1,k2) = y(k1,k2-2)+yh*y(k1-1,k2-2);
        b51(k1,k2) = yh*y(k1,k2-2)+y(k1-1,k2-2);
    end
end
for k2 = 3:n
    f41(1,k2) = y(k1,k2-2);
    b51(1,k2) = yh*y(k1,k2-2);
    f51(2,k2) = y(k1-1,k2-2);
    b61(2,k2) = yh*y(k1-1,k2-2);
end
for k1 = 3:m
    f61(k1,2) = yv*y(k1-2,k2-1);
    b71(k1,2) = y(k1-2,k2-1);
    f71(k1,1) = yv*y(k1-2,k2);
    b81(k1,1) = y(k1-2,k2);
end
end
for k1=3:m
    for k2 = 3:n
        f51(k1,k2) = y(k1-1,k2-2)+yh*y(k1-2,k2-2);
        b61(k1,k2) = yh*y(k1-1,k2-2)+y(k1-2,k2-2);
        f61(k1,k2) = y(k1-2,k2-2)+yv*y(k1-2,k2-1);
        b71(k1,k2) = yv*y(k1-2,k2-2)+y(k1-2,k2-1);
    end
end
end
for k1=3:m
    for k2 = 2:n
        f71(k1,k2) = y(k1-2,k2-1)+yv*y(k1-2,k2);
        b81(k1,k2) = yv*y(k1-2,k2-1)+y(k1-2,k2);
    end
end
end
for k1 = 4:m
    f91(k1,1) = y(k1-3,1);
    b101(k1,1) = yv*y(k1-3,1);
    f101(k1,2) = y(k1-3,k2-1);
    b111(k1,2) = yv*y(k1-3,k2-1);
```

```
f111(k1,3) = y(k1-3,k2-2);
b121(k1,3) = yv*y(k1-3,k2-2);

end
for k1 = 4:m
    for k2 = 2:n
        f91(k1,k2) = y(k1-3,k2)+yv*y(k1-3,k2-1);
        b101(k1,k2) = yv*y(k1-3,k2)+y(k1-3,k2-1);
    end
end
for k1 = 4:m
    for k2 = 3:n
        f101(k1,k2) = y(k1-3,k2-1)+yv*y(k1-3,k2-2);
        b111(k1,k2) = yv*y(k1-3,k2-1)+y(k1-3,k2-2);
    end
end
for k1 = 4:m
    for k2 = 4:n
        f111(k1,k2) = y(k1-3,k2-2)+yv*y(k1-3,k2-3);
        b121(k1,k2) = yv*y(k1-3,k2-2)+y(k1-3,k2-3);
        f121(k1,k2) = y(k1-3,k2-3)+yh*y(k1-2,k2-3);
        b131(k1,k2) = yh*y(k1-3,k2-3)+y(k1-2,k2-3);
    end
end
for k2 = 4:n
    f121(3,k2) = yh*y(k1-2,k2-3);
    b131(3,k2) = y(k1-2,k2-3);
    f131(2,k2) = yh*y(k1-1,k2-3);
    b141(2,k2) = y(k1-1,k2-3);
    f141(1,k2) = yh*y(k1,k2-3);
    b151(1,k2) = y(k1,k2-3);
end

for k1=3:m
    for k2 = 4:n
        f131(k1,k2) = y(k1-2,k2-3)+yh*y(k1-1,k2-3);
        b141(k1,k2) = yh*y(k1-2,k2-3)+y(k1-1,k2-3);
    end
end

for k1=2:m
    for k2 = 4:n
        f141(k1,k2) = y(k1-1,k2-3)+yh*y(k1,k2-3);
        b151(k1,k2) = yh*y(k1-1,k2-3)+y(k1,k2-3);
    end
end
```

```
for k1 = 4:m
    for k2 = 1:n
        f81(k1,k2) = y(k1-2,k2)+yh*y(k1-3,k2);
        b91(k1,k2) = yh*y(k1-2,k2)+y(k1-3,k2);
    end
end
F=[f01 f11 f21 f31 f41 f51 f61 f71 f81 f91 f101 f111 f121 f131 f141];
B=[b11 b21 b31 b41 b51 b61 b71 b81 b91 b101 b111 b121 b131 b141 b151];
```

```
%*****IEP215.m*****
```

```
%Hazurlayan: Nurşen YILDIZ
```

```
function [Ef,Eb] = iep215(F,B)
```

```
%Bu alt program, üçüncü dereceden 15 destek noktalı kafes süzgecinin birinci  
%aşama, birinci basamağına ilişkin ileri ve geri yönde kestirim hata alanları  
%güçlerini hesaplar.
```

```
[x,y]=size(F);  
m=x;n=y/15;  
d=0;  
a=1-n;  
t=(15*n)-(n-1);  
while a < t  
    a=a+n;  
    b=a+(n-1);  
    d=d+1;  
    f=F(:,a:b);  
    b=B(:,a:b);  
    Fi=f.*f;  
    Bi=b.*b;  
    FT=sum(sum(Fi));  
    BT=sum(sum(Bi));  
    Ef(d)=FT/(m*n);  
    Eb(d)=BT/(m*n);  
end  
end
```

```
%*****SPEC215.M*****
```

```
%Hazırlayan: Nurşen YILDIZ
```

```
function Pz=spec215(A)
```

```
%Bu alt program 15 katsayı ağırlığına ilişkin frekans spektrumunu hesaplar.
```

```
[w1,w2]=meshdom(-pi:pi/15:pi,-pi:pi/15:pi);  
ag1=1-A(1).*exp(-j.*w1)-A(2).*exp(-j.*w1).*exp(-j.*w2)-A(3).*exp(-j.*w2);  
ag2 = -A(4).*exp(-j*2.*w2)-a5.*exp(-j.*w1).*exp(-j*2.*w2);  
ag3 = -a6.*exp(-j*2.*w1).*exp(-j*2.*w2)-a7.*exp(-j*2.*w1).*exp(-j.*w2);  
ag4 = -a8.*exp(-j*2.*w2)-a9.*exp(-j*3.*w1) -a10.*exp(-j*3.*w1).*exp(-j.*w2);  
ag5 = -a11.*exp(-j*3.*w1).*exp(-j*2.*w2)-a12.*exp(-j*3.*w1).*exp(-j*3.*w2);  
ag6= -a13.*exp(-j*2.*w1).*exp(-j*3.*w2)-a14.*exp(-j.*w1).*exp(-j*3.*w2);  
ag7 = -a15.*exp(-j*3.*w2);  
A=abs(ag1+ag2+ag3+ag4+ag5+ag6+ag7);  
Py=A.*A;  
clear A;  
[m,n]=size(Py);  
Pz=ones(m,n)./Py;  
end
```



```
%*****LAT215.m*****
```

```
%Hazırlayan: Nurşen YILDIZ
```

```
%%%%%%%% MAIN %%%%%%%%%
```

```
%Bu program üçüncü dereceden, çeyrek düzlemde 15 destek noktalı kafes süzgeci  
%yöntemini kullanarak AR veri alanına ilişkin parametreleri kestirip, gerçek ve  
%Yule-Walker frekans spektrumları ile karşılaştırır.
```

```
VA=input('Veri alanının boyutunu giriniz...[m,n] = ? ');  
%disp('AR veri alanı üretmek için katsayı vektörünü giriniz..')  
%disp('A=[a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15] biçimindedir.')  
%A=input('A=');  
A=[-.013 -.18 -.08 -.15 -.045 -.075 -.108 -.033 -.0034 -.0031 -.0002 -.032 -.0034  
-.0032 -.0005];  
K=A;  
y=ar215(VA,A);  
[m,n] = size(y);  
D = kokat(y);  
yh = -D(3)/D(1);  
yv = -D(2)/D(1);  
f = zeros(m,n);  
b = zeros(m,n);  
asama = 0;  
while asama < 15  
    asama=asama+1  
    inc=-(n-1);  
    bastop=16-asama;  
    F=zeros(m,n*bastop);  
    B = F;  
    Eft = zeros(1,bastop);  
    Ebt = Eft;  
    YF=zeros(1,bastop);  
    YB=YF ;  
    basamak=0;  
    A=ones((asama+1),bastop);  
    G=A;  
    while basamak < bastop  
        basamak=basamak+1;  
        if asama > 1  
            inc=inc+(m-1);  
            t=FS(:,(basamak+inc):(basamak+inc+m-1));  
            u=BS(:,(basamak+inc+m):(basamak+inc+(2*m)-1));  
            Di=t.*u; %Çapraz ilişkilerin hesaplanması
```

```
D=sum(sum(Di));D=D/(m*n);
yf = -D/EFT(basamak);           %İleri yönde yansıma katsayıları
yb = -D/EBT(basamak+1);         %Geri yönde yansıma katsayıları
YF(basamak) = yf
YB(basamak) = yb
for k1=1:m
    for k2=1:n
        f(k1,k2)=t(k1,k2)+yb*u(k1,k2);    %İleri yönde hata alanları
        b(k1,k2)=yf*t(k1,k2)+u(k1,k2);    %Geri yönde hata alanları
    end
end
F(:,(basamak+inc):(basamak+inc+m-1))=f;
B(:,(basamak+inc):(basamak+inc+m-1))=b;
Ef=f.*f;Ef=sum(sum(Ef));Ef=Ef/(m*n);
Bf=b.*b;Bf=sum(sum(Bf));Bf=Bf/(m*n);
Eft(basamak)=Ef;Ebt(basamak)=Bf;
X=Ae(:,basamak);
Xi=yf.*X;
Z=Ge(:,basamak+1);
Zi=yb.*Z;
Ai=ones(asama+1,1);
Gi=Ai;
for q=2:asama
    Ai(q)=X(q)+Zi(q-1);
    Gi(q)=Xi(q)+Z(q-1);
end
Ai(asama+1)=Zi(asama);
Gi(1)=Xi(1);
A(:,basamak)=Ai
G(:,basamak)=Gi
if basamak==bastop
    Ae=A
    Ge=G
end
else
    basamak=bastop;
    [F,B] = ini215(y)
    [Eft,Ebt]=iep215(F,B)
    Ae=[ones(1,15);yh yv yh yv yh yv yv yh yv yv yv yh yh yh]
    Ge=[yh yv yh yv yh yv yv yh yv yv yv yh yh yh ;ones(1,15)]
end
end
FS = F;
BS = B;
EFT = Eft
EBT = Ebt
```

```
end
end
A=A(2:16);
Pz=spec215(A);
clg;
subplot(221);
mesh(Pz)
title('Hesaplanan kats. ')
grid
X=[-pi:pi/15:pi];Y=X;
subplot(222);
contour(Pz,X,Y);
xlabel('w1');ylabel('w2');
grid
A=-K
Pz=spec215(A);
subplot(223);
mesh(Pz)
title('Gerçek kats. ')
grid
subplot(224);
contour(Pz,X,Y)
xlabel('w1');ylabel('w2');
grid
end
```

ÖZGEÇMİŞ

Nurşen Yıldız 1970 yılında Erzincan'da doğdu. İlk ve ortaokulu İstanbul - Pendik'de bitirdi. 1983 yılında Haydarpaşa Anadolu Teknik Lisesi Elektronik Bölümüne girdi. 1987 yılında Haydarpaşa Endüstri Meslek Lisesi Elektronik Bölümü mezunu olarak, Yıldız Üniversitesi Kocaeli Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümüne girdi. 1991 yılında lisans öğrenimini tamamladı ve aynı yıl İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Bölümüne kayıt oldu. Halen Kocaeli Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümünde Araştırma Görevlisidir.