

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

MİKRODALGA TRANSİSTÖRLERİNİN  
YAPAY SİNİR AĞI EŞDEĞERLİKLERİ

Elektronik ve Haberleşme Yük.Müh. Hamid TORPİ

F.B.E. Elektronik ve Haberleşme Mühendisliği  
Anabilim dalında hazırlanan  
Doktora Tezi

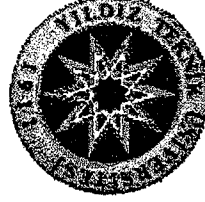
67708

Tez danışmanı: Prof. Dr. Filiz GÜNEŞ  
Tez jürisi üyeleri: Prof. Dr. Uğur ÇİLİNGİROĞLU  
Doç. Dr. Ahmet AKSEN

İSTANBUL, 1997

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ



MİKRODALGA TRANSİSTÖRLERİNİN YAPAY  
SİNİR AĞI EŞDEĞERLİKLERİ

Elektronik ve Hab. Yük.Müh. Hamid TORPİ

F.B.E Elektronik ve Haberleşme Mühendisliği  
Anabilim dalında Hazırlanan

DOKTORA TEZİ

Tez Danışmanı: Prof. Dr. Filiz GÜNEŞ

Prof. Dr. Uğur Çilingiroğlu

Doc. Dr. Ahmet Aksın

Prof. Dr. Filiz GÜNEŞ

İSTANBUL, 1997



## TEŐEKKÖR

Bu tezin hazırlanmasında ve yaptığımız bilimsel alıřmalarda bize önyak olan,gerek yönlendirme gerekse kaynak temininde her türlü desteęi ile bana yardımcı olan saygıdeęer hocam sayın Prof. Dr. Filiz GÖNEŐ'e teőekkürlerimi sunarım.Doktora teziminin paralelinde yapmış olduęumuz alıřmalarda bize kaynak temin eden YTÖ Arařtırma fonuna da ayrıca teőekkür ederim.Yetiřmemizde emeęi geen tüm hocalarımızı, özellikle Do. Dr. Fikret GÖRGEN'i burada anmayı bir bor bilirim.

Ayrıca tezimin yazılması sırasında bana yardımcı olan sevgili eřim Ayře'ye ;bana en güzel alıřma ortamı babama,aęabeylerim Mustafa ve Harun TORPİ'ye ,sevgiyle büyüten anneme ve tüm aileme en içten řükranlarımı sunarım.

Mayıs 1997

Hamid TORPİ



<b>İÇİNDEKİLER</b>	<b>i</b>
Özet	v
Summary	vi
<b>1 MİKRODALGA MÜHENDİSLİĞİNDE MODELLEME</b>	<b>1</b>
1.1 Klasik Modellemeye Giriş	1
1.1.1 Küçük İşaret Modeli	2
1.1.1.1 Eşdeğer Devre Eleman Değerlerinin Fiziki Önemi	2
1.1.1.2 Fizik Temelli Modelleme	5
Doğrudan Akım Analizi	6
Geçiş İletkenliği ve Çıkış İletkenliği	8
$C_{gs}$ , $C_{gd}$ ve $C_{ds}$	9
1.1.1.3 Frekans Bağımlı İletkenliklerin Modellenmesi	10
1.1.2 Büyük İşaret Modeli	11
1.1.2.1 Büyük İşaret Modellemeye Ampirik Yaklaşım	11
Curtice Modeli	13
Diğer Belli Başlı Büyük İşaret Modellerine Toplu Bir Bakış	16
HEMT'ler İçin Büyük İşaret Modelini Geliştirme Prosedürü	19
HEMT Curtice Modeli	20
1.1.3 Modelleme Parametrelerinin Kestirimi	21
1.1.4 Gürültü Modeli	25
1.1.4.1 Gürültünün Tabiatı	25
Isıl Gürültü	27
1.1.4.2 Gürültü Faktörü	29
1.1.4.3 Lineer İki-Kapılıda Gürültünün Temsil Edilmesi	32
Gürültü Matrisinden Gürültü Faktörünü Tayin Etme	35
1.1.4.4 İki-Kapılının Gürültü Parametreleri ve Gürültü Faktörü	37
1.1.4.5 Ard Arda Bağlanmış İki-Kapılıların Gürültü Faktörü	40
<b>2 YAPAY SİNİR AĞLARI HAKKINDA GENEL BİLGİLER</b>	<b>42</b>
2.1 Biyolojik Sinir Ağları	42
2.1.1 Biyolojik Sinir Hücresinin Anatomisi	43
2.1.2 Biyolojik Sinir Hücresinin Fizyolojisi	43

2.2 Yapay Sinir Ağının Yapısı	44
2.2.1 Giriş İşareti Sınıfları	45
2.2.2 Bağlantı Geometrilere	46
2.2.3 Ağ Tipleri	47
2.2.4 Aktivasyon Fonksiyonları	47
2.2.5 Yapay Sinir Ağının Kullanım Nedenleri	49
2.3 Eğitim Algoritmaları	51
2.4 Öğrenme Kuralları	52
2.4.1 YSA Konfigürasyonu	52
2.5 Çok Katmanlı Algılayıcı (Multi Layer Perceptron:MLP)	54
2.6 Geriye Yayılma Algoritması GYA (Back Propagation Algorithm) ve Genelleştirilmiş Delta Kuralı	54
2.6.1 Öğrenme ve Momentum Katsayıları	64
2.7 Yapay Sinir Ağlarının Elektronik Devre Olarak Gerçeklenmesi	64
2.7.1 Giriş	64
2.7.2 Belli Başlı Tasarım Öğeleri	65
2.7.3 VLSI Uygulamalarının Kategorileri	67
2.7.4 YSA Donanımlarına Kısa Bir Bakış	73
<b>3 AKTİF MİKRODALGA ELEMANLARININ YSA İLE MODELLENMESİ</b>	<b>74</b>
3.1 Giriş	74
Amaç	74
3.2 Aktif Mikrodalga Elemanın Küçük-İşaret Gürültü Davranışının Belirlenmesi	76
3.3 Eşdeğer YSA Modelleri	79
Eşdeğer YSA Gürültü Modeli	80
Eşdeğer YSA İşaret Parametreleri Modeli	81
Eşdeğer YSA İşaret&Gürültü Parametreleri Modeli	81
Çok Boyutlu Eşdeğer YSA İşaret&Gürültü Parametreleri Modeli	82
3.4 Hedef Uzayının Belirlenmesi ve Genelleştirme	84
3.5 Çok Boyutlu Eşdeğer YSA İşaret&Gürültü Parametreleri Modelinin Matematiksel olarak tanımlanması	85
3.5.1 Devrenin Yapısı	85
3.5.2 Devre Matrisi P'nin Hesaplanması	86
3.6 Başarım Ölçütü	89

<b>4 YSA MODELİNİN AKTİF MİKRODALGA ELEMANLARINA UYGULANMASI</b>	<b>90</b>
4.1 N24200a Üzerinde Geliştirilen Sabit Kutuplamalı Eşdeğer Sinyal-Gürültü YSA Modeli İçin Elde Edilen Simülasyon Neticeleri	91
4.1.1 N24200A İçin Üretici Firma verileri ve Simülasyon Neticeleri	91
4.2 NE21900 İçin Üretici Firma verileri ve Elde Edilen Simülasyon Neticeleri	96
4.3 N72000A ve NE32684A Transistörü Üzerinde Geliştirilen Çeşitli Tipteki Eşdeğer YSA Modellerinin İncelenmesi	101
4.4 Ortak Emetör Konfigürasyonundaki BFQ74 Transistörü Üzerinde Geliştirilen Tek Kutuplama Gerilimli,Farklı Kutuplama Akımları İçin Eşdeğer Sinyal-Gürültü YSA Modeli	107
4.4.1 BFQ74 İçin Elde Edilen Simülasyon Neticeleri	107
4.4.2 BFQ74 İçin Üretici Firma verileri	112
4.5 Ortak Emetörlü Konfigürasyonundaki NE02135 Transistörü Üzerinde Geliştirilen Tek Kutuplama Gerilimli, Farklı Kutuplama Akımları İçin Eşdeğer Sinyal-Gürültü YSA Modeli	113
4.5.1 NE02135 İçin Üretici Firma verileri	114
4.5.2 NE02135 İçin Elde Edilen Simülasyon Neticeleri	115
4.6 Ortak Emetör Konfigürasyonundaki BFR28 Transistörü Üzerinde Geliştirilen Farklı Kutuplama Noktaları İçin Eşdeğer Sinyal YSA Modeli	118
4.6.1 BFR28 İçin Elde Edilen Simülasyon Neticeleri	118
4.6.2 BFR28 İçin Üretici Firma verileri	119
4.7 Eşdeğer YSA Yardımıyla Her Hangi bir Frekansta ( $f=0.1-5\text{GHz}$ ) NE02135 Transistörünün S-Parametrelerinin kutuplama ile ( $V_{CE}=10\text{V}, I_C=2-25\text{mA}$ ) Değişiminin Hesaplanması Uygulamasına Ait Sonuçların Yansıma Katsayısı Düzleminde Çizdirilmesi	124
4.8 Muhtelif Transistörler İçin Eğitim İçin Gerekli Bilgisayar Zamanının İncelenmesi	132
<b>5 TARTIŞMA VE SONUÇ</b>	<b>133</b>
<b>EK 1 AKTİF MİKRODALGA ELEMANLARI İÇİN EŞDEĞER YSA SİMÜLATÖRÜ</b>	<b>134</b>
EK 1.1 Çeşitli Kutuplama ve Çeşitli Konfigürasyon Tipi İçin Gürültü ve Sinyal Parametrelerinin Simülatörü	134
EK 1.1.1 Çeşitli Kutuplama İçin Sinyal Parametreleri Simülatörü	155
EK 1.1.2 Sabit Kutuplama İçin Sinyal ve Gürültü Parametreleri Simülatörü	173
EK 1.2 Simülasyon Sonuçlarının Smith Abağına Bastırmak İçin Geliştirilen Program	193
EK 1.2.1 Gürültü ve Sinyal Parametreleri İçin	193
EK 1.2.2 Sinyal Parametreleri İçin	200
EK 1.2.3 Renkli Baskılı Sinyal Parametreleri İçin	205

EK 1.3 Optimum Alpha Deęerini Bulan Program	211
EK 1.3.1 Aktif Mikrodalga Elemanı Deęiřtikçe Programde Deęiřmesi Gereken Parametrelerin Birkaç rnek zerinde Gsterilmesi	219
-NE02135 İin	219
-NE219 İin	220
-BFR28 İin	221
-BFQ74 İin	223
<b>EK 2 YAZARIN DOKTORA TEZİ KONUSU İLE İLGİLİ YAYINLARI</b>	<b>225</b>
EK 2.1 Uluslararası Makaleler	225
EK 2.2 Uluslararası Bildiriler	225
EK 2.3 Ulusal Bildiriler	225
IEE Proc.-Circuit Devices Syst.	227
IEE Proc.-Circuit Devices Syst.	235
ANN'96 ENGLAND	247
PIERS 95 USA	257
ECCTD'95 TRKİYE	258
ICT'96 TRKİYE	262
PIERS'96 AUSTRIA	266
SMCA'96 TRKİYE	267
ECCTD'97 HUNGARY	273
SİU'95 Nevřehir TRKİYE	279
EMO 6. Ulusal Kongre Bursa TRKİYE	285
<b>EK 3 İKİ KAPILI PARAMETRELERİ ARASI DNÜřM FORMLLERİ</b>	<b>289</b>
<b>KAYNAKLAR</b>	<b>291</b>
<b>ZGEMİř</b>	<b>296</b>

## ÖZET

Bu tezde bir mikrodalga transistörünün sinyal ve gürültü davranışları ,frekans,kutuplama ve konfigürasyon tipi cinsinden bütün bir çalışma bölgesinde ,bir yapay sinir ağı ile modellenebilmiştir.Böyle bir yaklaşımda,transistör ,sinyal ve gürültü performans parametreleri, bir yapay sinir ağıyla hesaplanabilen bir “kara kutu” ile modellenmiştir.Öyle ki bu hesaplama çoklu kutup ve konfigürasyon için bütün çalışma frekans bandında hedef uzayına uydurulmaya dayandırılmıştır.Önerilen yapay sinir ağı yaklaşımı ,optimizasyon prosedürü esnasında elemanın fizik denklemlerinin tekrar tekrar çözülmesini gerektirmez ve ayrıca bir çok klasik optimizasyon teknikleri ile parametre kestiriminde büyük bir sorun olan başlangıç değer problemi yoktur.Yapay sinir ağının nispeten optimum sürede eğitilmesinden sonra istenilen konfigürasyon tipi (CT),kutuplama koşulu ( $V_{DS}, I_{DS}$ ) ve çalışma frekansı ( $f$ )’in ağa girilmesiyle, sekiz adet işaret parametresi ki mikrodalga transistörü için saçılma (S) parametreleri seçilmiştir,dört adet gürültü parametresi  $F_{min}$ ,  $\Gamma_{opt} = \Gamma_{opt_r} + j\Gamma_{opt_i}$ ,  $R_N$  ağ çıkışından hedef uzayıyla oldukça iyi bir uyum içerisinde hemen elde edilir.Çalışmada eğitim süresi ,hedef uzayı ve yapay sinir ağı mimarisi, bir karesel ortalama objektif fonksiyonu kullanılarak optimize edilmiştir.Sonuçta dört düğümden oluşan bir girişi ,çıkış düğümü kadar ki bizim devremiz için oniki düğüme sahip bir gizli katmandan ve çıkış katmanından oluşan yapay sinir ağı transistörün bütün bir çalışma bölgesinde eşdeğer devresi olarak kullanılabilir bulunmuştur.

Yapay sinir ağı uygulamaları açısından ,bu çalışmada ağ ile sekizi sinyal, dördü gürültü parametresi olmak üzere toplam oniki parametrenin konfigürasyon tipine ,kutup koşuluna ve frekansa bağlılığına eş zamanlı yaklaşıklık yapılmıştır.Bu yaklaşıklıkta baz olarak ,hedef uzayını gerçekleştirme ölçüsü kriteri olan objektif fonksiyonunun minimize edilmesi alınmıştır.

Eşdeğer sinyal ve gürültü yapay sinir ağı iki tip genelleştirme yapabilmektedir:(i) Tek frekans genelleştirme: Oniki performans parametresinin ,seçilen konfigürasyon tipinin, bir kutuplama koşulunun bir çalışma frekansında kestirimidir.(ii) Çalışma Frekans Bandı Genelleştirme :Oniki adet performans parametresinin, seçilen konfigürasyon tipinin, bir kutuplama koşulunun bütün çalışma bandı boyunca kestirimidir.Ayrıca işaret ve gürültü yapay sinir ağı modeli başarımlı ölçütleri tanımlanmıştır.Model çok sayıda transistöre uygulanmış,her bir örnek için başarımlı ölçütü hesaplanarak genellikle makul değerler elde edilmiştir.Sonuç olarak ,böyle bir modelin mikrodalga devre ,özellikle tümleşik mikrodalga devre tasarımında kullanılabilceği sonucuna varılmıştır.



## SUMMARY

Signal and noise behaviours of microwave transistors are modeled through the neural network approach for the whole operating ranges including frequency, bias and configuration types. Here, the device is modeled by a black box whose small-signal and noise parameters are evaluated through a neural network, based upon the fitting of both of these parameters for multiple bias and configuration with the target space. The concurrent modeling procedure does not require to solve any equation of the physics of the device during the optimization procedure. After having trained the network in an optimum time interval compared with the other modelling techniques, inputting the required configuration type (CT), bias condition ( $V_{DS}, I_{DS}$ ) and operating frequency, one can obtain the corresponding eight signal parameters which are the scattering parameters in our study, and four noise parameters which are  $F_{min}$ ,  $\Gamma_{opt} = \Gamma_{opt_r} + j\Gamma_{opt_i}$  and  $R_N$ , from the output of the network in good agreement with the target space. In the work, training time, elements of the target space and the network architecture are optimised by means of the objective function which is chosen as a mean square error function and finally a simple neural network is resulted so that it has four processing nodes at the input layer and twelve nodes separately at the hidden and output layers.

From this point of view of the applications of the neural networks, dependences of the twelve performance parameter functions upon the configuration type, bias condition and operation frequency are approximated simultaneously through a simple neural networks is a good agreement with the target space. This approximation is based upon minimisation of the mean square error function which is formed by differences of the predicted and target space.

Signal-Noise Neural Network can achieve two types of generalisations: (i) Single Frequency Generalisation: This can be defined as prediction of the performance parameters at a single operation frequency of the trained bias condition. (ii) Whole Frequency Band Generalisation: This can be defined as prediction of the unknown multidimensional performance function for the whole operation frequency band, which has eight signal dimensions and four noise dimensions, at an untrained bias condition.

Besides, performance measures of the signal-noise neural network are defined and applied to many transistors; evaluating performance measures of the for each example, one concludes that this type of modelling can be used successfully in the design of microwave integrated circuits.

## 1 MİKRODALGA MÜHENDİSLİĞİNDE MODELLEME

### 1.1 Klasik Modellemeye Giriş

Bir çok farklı yarıiletken devre modelleri mevcuttur. Her model bir kaç farklı kategoride sınıflandırılabilir. Modeller genellikle çıkarılış şekline bağlı olarak sınıflandırılırlar. Fiziksel temelli modeller elemanın fiziki yapısına fizik kanunlarının uygulanması fikrinden hareketle çıkarılırken ampirik modeller elde edilen karakteristiklerin rastgele fonksiyonlar ile ölçüm ile elde edilen karakteristiklere uydurulmasına dayanır. Bir çok modelin hem ampirik hemde fiziksel temele dayanan tarafları vardır. Bunlara ilaveten kestirim (tahmin) yapabilmek için gerekli olan hesaplamaların tipine (nümerik hesaplama tipine) göre de modelleme sınıflandırılabilir. Esas olarak öngörülmek istenen performans tipide modellemeyi sınıflandırmak için kullanılır. Küçük-ışaret, büyük-ışaret ve gürültü modelleri bunların herbirine ait devre karakteristiklerini çıkarmak için faydalıdır.

Tipik uygulamada, performans bilgisinin mevcut olmadığı veya ölçmeyle kolayca elde edilemeyecekse modellemeye başvurulur. Modellemenin diğer önemli bir kullanım alanıda mevcut ölçümlerin içinde olmayan frekanslardaki ölçülmüş datayı interpolate veya extrapole etmektir. Gürültü modelleride devrenin gürültü performansını kestirmek için kullanılır.

Hem devre tasarımcıları hem de devre elemanı tasarımcıları devre elemanı modellemede uygulama alanı bulurlar. Bu iki grubun modeller üzerindeki gereksinimler listesi tıpatıp aynı olması gerekmediğinden farklı modeller kullanırlar. Bununla birlikte devre tasarımcıları gelişen teknoloji ile tümleşik devre tasarımına (üretim aşamasında) dolaylı olarak müdahale edebilmektedirler. Bu nedenle önceleri sadece devre elemanı tasarımcılarının kullandığı fiziksel temelli modellerle devre tasarımcılarında tanışmasına neden olmuştur.

Buda fiziksel temelli modellerin ampirik modellere göre değerinin tartışılmasına neden olmuştur. Her iki modelleme tekniğinde kendine göre avantaj ve dezavantajları vardır. Fiziksel temelli modeller devre tasarımcılarına çok cazip gelmektedir. Böyle modeller fabrikasyon prosesleri üzerinde bazı etkileri (Kontrolleri) olan devre tasarımcıları için kullanışlıdır. Çünkü bu devre elemanının ve devrenin her ikisinin aynı anda (simültane) olarak optimizasyonuna olanak verir. Buna ilaveten, devrenin elektriksel davranışının üretim proseslerine etkisi öngörülmesi için fiziksel temelli modeller kullanışlıdır. Böyle bir yaklaşım kullanılarak, sadece elemanı tanımlayan devre elemanının geometrisi, yarıiletken materyalin özellikleri vs gibi fiziki bilgilerden hareketle devre elemanına ait performans verileri elde edilebilir. Hiçbir elektriksel karakterizasyona ihtiyaç duyulmaz. Bu yaklaşımın değeri devre tasarımcıları için aşıkardır.

Bununla birlikte, bir çok devre tasarım uygulamaları için istenilen hassasiyete sahip modeller değildir. Devre elemanının analizini yapabilmek için yapılan kabuller ve yaklaşıklıklar hatalara neden olmaktadır. Örneğin, yüzey durum etkileri, tuzaklar, homojen olmayan ara yüzeyler v.s. gibi şeyler tipik olarak ihmal edilir. Fakat bu olayların modern mikrodalga elemanlarının performansı üzerinde

önemli etkileri vardır.Fizik temelli modellerin ikinci problemi devre elemanının fiziksel tasarımını ilgilendiren bilgilerin genellikle elde edilmesinin zor veya imkansız olmasıdır.Bu problem ampirik parametre değerleri hesaplanırken kullanılan yolun aynısı olan ölçülmüş bilgilerden hareketle fiziksel parametrelerin kestirilmesi(tahmini) yoluyla çözülebilir.

Bunun aksine, ampirik modeller ölçüm kapasitesinde kestirim hassasiyetine ulaşabilmektedir.Bu modelleme yaklaşımında ise birinci derece zorluk iyi bir hassasiyet için büyük miktardaki karakterizasyon bilgisine ihtiyaç göstermesidir.Buna ilaveten devre elemanı geometrisindeki en küçük değişiklikler veya yeni materyal gereksinimi durumunda,elemanın performansı yeni baştan karakterize edilmesi gerekir.

Tipik olarak fiziksel temelli modeller ampirik modellere göre daha az hassasiyete sahip olduklarından,bir ilgi çekici geliştirme yolu devre elemanının nominal performansı ampirik modelle simüle edilir ve bu nominal davranış hakkındaki sapmalar fizik temelli modelleme ile kestirilir.

Bundan sonraki bölümde literatürde kullanılan bir kaç popüler modele kısaca göz alacağız.Özellikle küçük-ışaret,büyük işaret ve gürültü modelleri üzerinde duracağız.Her alt bölümün içinde fizik temelli ve , veya ampirik yaklaşımlarda incelenecektir.Bu incelemeler yapılırken geniş çalışma bandı,düşük gürültü figürü ,daha üstün çalışma performansı nedeniyle (bipolar transitörler veya JFET'ler üzerinde değilde ) genellikle HEMT ve MESFET'ler üzerinde klasik modelleme tekniklerini tanıtmaya çalışacağız.

### 1.1.1 Küçük İşaret Modeli

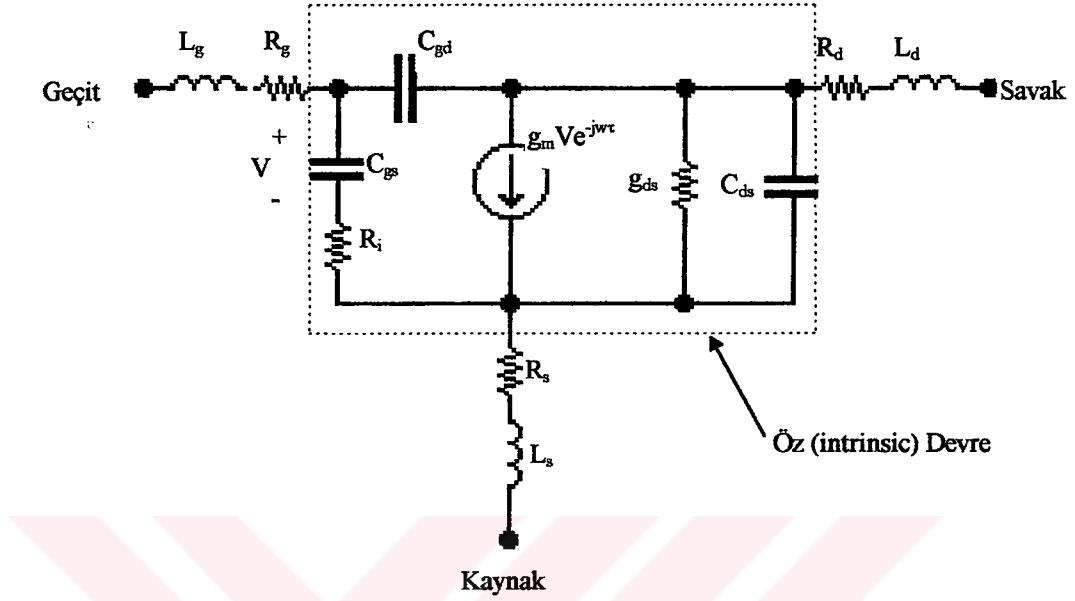
MESFET-HEMT küçük-ışaret modeli aktif mikrodalga devrelerinde oldukça önemlidir.Bu modeller devre elemanının içinde oluşan elektriksel işlemlerle ölçülmüş s-parametreleri arasında bağlantı kurmamızı sağlar.Eşdeğer devredeki elemanların herbiri devre elemanının fiziki yapısının bazı kısımlarının toplu parametrelere devre elemanı yaklaşımıdır.Uygun seçilmiş bir topoloji ,ilaveten fiziksel olarak anlamlıysa ,çok geniş bir frekans kademesinde ölçülmüş s-parametrelerine mükemmel bir uyum sağlarlar.Eğer eleman değerleri yerli yerince kestirilmişse,model ölçüm kademesinin dışındada geçerlidir.Böylelikle ölçme enstrumanlarının ölçme kapasitesinin üstündeki bölgede de devre elemanı performansının kestirimi imkanını sağlar.Buna ilaveten,eşdeğer devre elemanları geçit genişliği ile ilişkilendirilebilir..Böylelikle tasarımcıya farklı ölçülerdeki devre elemanlarının s-parametrelerini kestirmesine olanak tanır.MMIC tasarım uygulamalarında devre elemanının geçit genişliği parametresinin devre tasarım işleminin bir parçası olarak içermesi kabiliyeti önemlidir.

#### 1.1.1.1 Eşdeğer Devre Eleman Değerlerinin Fiziki Önemi

Şekil 1.1'de oldukça standart olan eşdeğer devre topolojisi verilmiştir.Bu topoloji her bir elemanın tek başına ekstrapole edilmesi gibi bir avantajı vardır. Şekil 1.2'de bir devre elemanının kesiti



üzerinde ,eşdeğer devre her bir elemanın fiziksel orijini de göstermek suretiyle oturtulmuştur.Devre elemanının fiziğini modellemede eşdeğer devre elemanlarının herbirinin rolünü kısaca açıklamaya çalışalım.



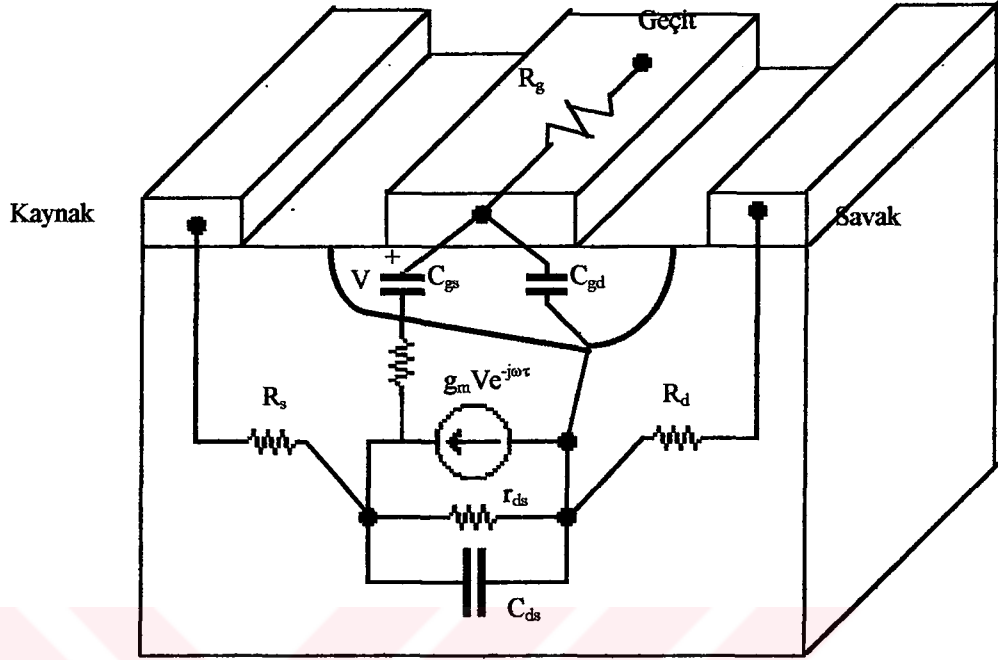
Şekil 1.1 Parazitik elemanlarda içeren MESFET/HEMT Küçük-ışaret modeli

**Parazitik empedanslar  $L_s, L_d$  ve  $L_g$ :** Devre elemanının yüzeyindeki metal kontaklar nedeniyle parazitik endüktanslar ortaya çıkarlar. Çünkü bu değerler devre elemanının yüzey özelliklerine bağlıdır. Temel olarak MESFET ve HEMT'lerde de aynıdır. Modern kısa geçit uzunluklu devre elemanları için ,geçit endüktansı sık sık diğer ikisinden daha fazla olur. Tipik olarak  $L_g$  ve  $L_d$  5 ila 10 pH arası değer alır. Kaynak endüktansı  $L_s$  genellikle küçük ve 1pH civarındadır. Not etmek gerekir ki bağlantı tellerinin endüktans etkilerini de modelin içermesi gerekir. Bir çok durumda bağlantıdan kaynaklanan endüktanslar 0.1 ila 0.3 nH arasında değer alır.

**Parazitik dirençler  $R_s, R_d$  ve  $R_g, R_s$  ve  $R_d$ :** dirençleri ohmik kontakların kontakt direnci ile aktif kanalın sebep olduğu direnci de içerir. Geçit direnci  $R_g$  geçidin Schottky kontağının metalizasyon rezistansından kaynaklanır. Modern mikrodalga elemanları için üçüde  $1\Omega$  civarındadır.  $R_s$  ve  $R_d$  HEMT'de MESFET'lere göre daha az olmaya meyillidir. Ölçümler bu dirençlerin çok az bir kutuplamaya bağımlılığını göstermesine rağmen bugünlerde elimizin altında olan devre simülatörlerindeki büyük-ışaret modellerinde genellikle sabit alınır. Bu dirençler ileri yöndeki kutuplamada ölçümlerden veya doğrudan S-parametrelerinde optimizasyon teknikleri kullanılarak hesaplanabilir.

**Kapasiteler  $C_{gs}, C_{gd}$  ve  $C_{ds}$ :** Tipik kuvvetlendirici ve osilatör için uygun kutuplama koşullarında ,geçit-kaynak kapasitesi daha büyük bir değerdedir. Çünkü geçit-kaynak gerilimindeki dalgalanmalardan kaynaklanan fakirleşmiş bölge yükündeki değişikliği modeller. Bu normal kutuplama koşulları altında





Şekil 1.2 MESFET/HEMT Küçük-ışaret eşdeğer devre modelindeki elemanların fiziki çıkış noktalarını gösteren temsili şema

geçit-savak kapasitesi  $C_{gd}, C_{gs}$ 'den oldukça daha küçüktür. Fakat bununla birlikte hassas S-parametresi kestiriminde kritik bir öneme sahiptir. Kaynak savak elektrodları arası geometrik kapasite etkisini hesaba katmak için savak-kaynak kapasitesi  $C_{ds}$  eşdeğer devreye dahil edilmiştir. Genellikle bu parametrelerin kutuplamaya bağlılığı eleman modellemede hesaba katılmaz. Normal kuvvetlendirici kutuplama koşullarında  $C_{gs}$  değerleri 1pF/mm geçit kalınlığı civarındadır.  $C_{gd}$  ve  $C_{ds}$  değerleri  $C_{gs}$ 'nin onda biri mertebesindedir.

**Geçiş iletkenliği  $g_m$**  : FET'in öz (intrinsic) kuvvetlendirme kazanç mekanizması bu parametre ile sağlanır. Geçiş iletkenliği giriş gerilimi  $V_{gs}$ 'ye verilen değişimin çıkış akımındaki  $I_{ds}$  artımsal değişiminin ölçüsüdür. Matematiksek olarak aşağıdaki gibi tanımlanır.

$$g_m = \frac{\partial I_{ds}}{\partial V_{gs}} \quad (1.1)$$

Geçiş iletkenliği mikrodalga uygulamalarında devre elemanının kalitesinin en önemli göstergesidir. Eğer diğer bütün parametreler eşitse yüksek geçiş iletkenliği daha büyük kazançlar ve yüksek frekans performansı sağlar. 1 MHz'e kadar olan frekanslarda geçiş iletkenliği frekansla



değişir. Geçiş iletkenliği geçit genişliği ile doğru, geçit uzunluğu ile ters orantılıdır. Bu hem MESFET hem de HEMT’de böyledir. Pratikte eşit geçiş geometrileri için HEMT’in geçiş iletkenliği, biraz daha MESFET’ten fazladır.

**Çıkış iletkenliği  $g_{ds}$ :** Çıkış iletkenliği çıkış akımındaki  $I_{ds}$  artımsal değişim ile çıkış gerilimi  $V_{ds}$ ’deki değişimin ölçüsüdür. Çıkış iletkenliği matematiksel olarak aşağıdaki gibi tanımlanır.

$$g_{ds} = \frac{1}{r_{ds}} = \frac{\partial I_{ds}}{\partial V_{ds}} \quad (1.2)$$

Tipik kuvvetlendirici kutuplamalarında  $g_{ds}$  1ms/mm geçiş genişliği mertebesindedir. Ve dahi, geçit uzunluğu azaldıkça ,hem MESFET hem de HEMT’de çıkış iletkenliği artma meyili gösterir. RF için çıkış iletkenliği dc için çıkış iletkenliğinden %100 daha fazladır. Analog uygulamalarda çıkış iletkenliği önemli bir eleman karakteristiğidir. Devre elemanından elde edilebilecek en büyük gerilim kazancının hesaplanmasında, buda optimum çıkış uydurma özelliklerinin hesaplanması için oldukça önemlidir. Genelde bir devre elemanından düşük değerli çıkış iletkenliğine sahip olması istenir eşdeğer olarak yüksek çıkış direnci istenmiş olur.

**Geçiş iletkenlik gecikmesi  $\tau$ :** Geçiş iletkenliği geçit gerilimindeki değişimlere anında cevap veremez. Bu işlevin tabiatındaki bu gecikme geçiş iletkenliği gecikmesi  $\tau$  olarak tanımlanır. Fiziksel olarak, geçiş iletkenliği gecikmesi geçit gerilimindeki dalgalanmalardan sonra yükün dağılımının yeniden düzenlenmesi için gereken zamanı gösterir.  $\tau$  ‘nın tipik değeri mikrodalga MESFET ve HEMT’lerde 1ps’dir. Geçiş iletkenliği gecikmesi HEMT’lerde MESFET’lere göre daha kısadır. Eşdeğer geometrilere azalan geçit uzunluğunda azalma meyili gösterir.

**$R_i$  direnci :**  $R_i$  direncinin öncelikle  $S_{11}$  ölçülen değerle uyumunu geliştirmek için kullanılır. Bundan başka , $R_g$ ’nin varlığı  $S_{11}$ ’in reel kısmının uydurulması için yeterlidir. Diğer taraftan , $R_i$ ’nin kestirimi güçtür.  $R_i$ ’nin dahil edilmesi büyük-ışaret analizini karmaşıktırır.

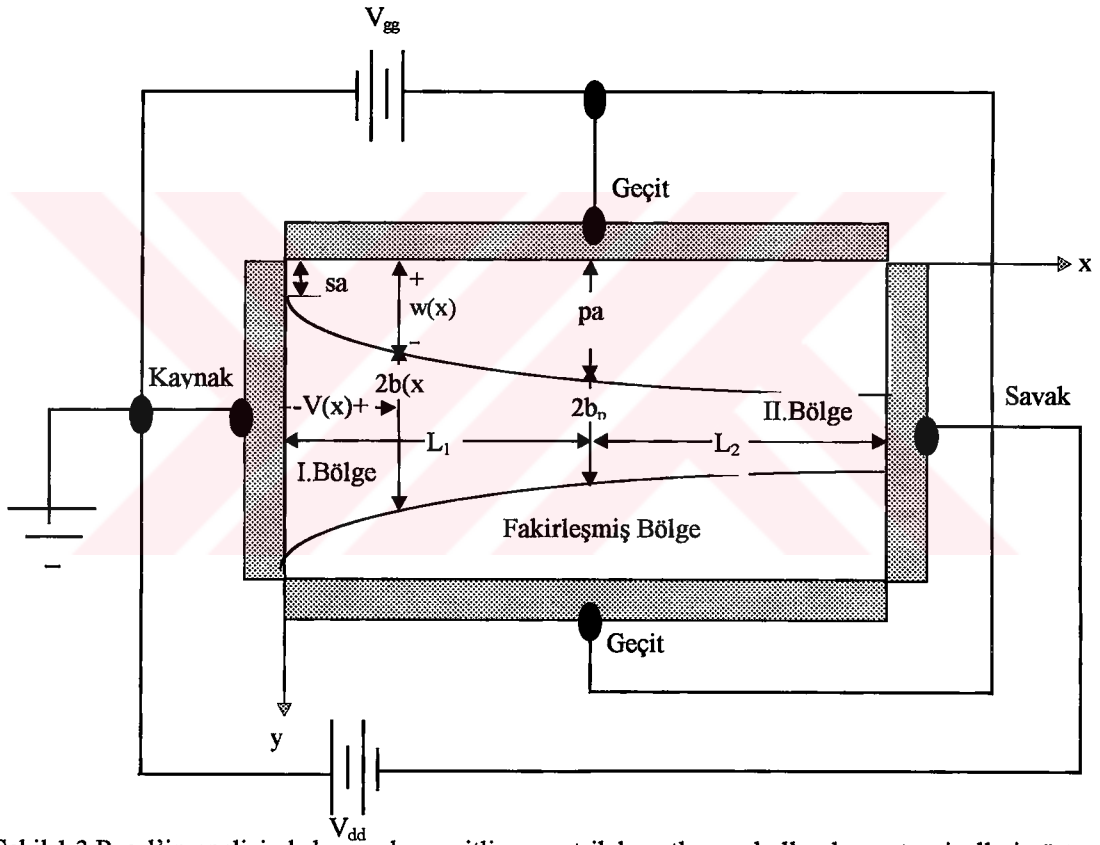
### 1.1.1.2 Fizik Temelli Modelleme

Bu bölümde Pucel’e ait fizik temelli model üzerinde inceleme yapılacaktır. Temel fiziksel parametrelere dayandırıldığı için ,geçit uzunluğu değişimlerinin, katkılama değişimlerinin ,tabaka kalınlığı değişimlerinin, geçiş iletkenliği, geçit-kaynak kapasitesi v.s. gibi küçük-ışaret parametreleri üzerindeki etkilerinin analizi için kullanışlıdır. Bu model aynı zamanda içinde mobilite, doyum hızı ,dielektrik sabiti v.s. gibi temel fiziksel parametreleride ihtive ettiğinden GaAs’den başka materyal kullanıldığında yeni materyalli devre elemanının analizi içinde kullanışlıdır.

**Doğrudan Akım Analizi:** Bu analiz şekil 1.3’de verilen idealize edilmiş simetrik FET yapısına dayandırılacaktır. Kanal taşıyıcı hızın doyuma ulaştığı bölge  $L_1 \leq x \leq L$  ve elektrik alanının mobilite  $\mu_0$  ile orantılı olduğu bölge  $x < L_1$  olmak üzere iki kesime boylamasına ayrılmıştır. Çünkü model ohmik davranıştan doyma durumuna ani bir geçiş yapmaktadır, bu bazı kereler iki-bölgeli model olarak referans edilir.

Bu analizin bir amacı olarak geçit elektroduna bağlı olarak  $W(x)$  kanal potansiyelini tanımlayacağız.

$$W(x) = V_{gs} + V_{bi} - V(x) \quad (1.3)$$



Şekil 1.3 Pucel’in analizinde kullanılan çeşitli geometrik boyutları ve kullanılan potansiyelleri gösteren FET’in bir kesitinin diagramı.

Burada  $V(x)$  kaynak ucuna göre kanal potansiyelidir.  $V_{gs}$  geçit-kaynak kutuplama potansiyeli ve  $V_{bi}$  geçit jonksiyon içi potansiyelidir. Kanal potansiyelleri sırasıyla kaynak, savak ve kırılma (pinch-off) noktaları için aşağıdaki gibi ifade edilebilir.

$$W_s = V_{gs} + V_{bi} \quad (1.4)$$

$$W_d = V_{gs} + V_{bi} - V_{ds} \quad (1.5)$$

$$W_p = V_{gs} + V_{br} - V_p \quad (1.6)$$

Normalize potansiyellerin tanımlanması ile kullanılacak notasyon basitleştirilebilir.

$$s = \sqrt{(W_s / W_{00})} \quad (1.7)$$

$$p = \sqrt{(W_p / W_{00})} \quad (1.8)$$

$$d = \sqrt{(W_d / W_{00})} \quad (1.9)$$

$$w(x) = \sqrt{(W(x) / W_{00})} \quad (1.10)$$

ki burda

$$W_{00} = \frac{qN_d a^2}{2\epsilon_r \epsilon_0} \quad (1.11)$$

dır.  $W_{00}$  kanaldaki taşıyıcıları fakirleştirmek için gerekli geçit-kanal potansiyeli,  $a$  elemanın epitaksiyel tabaka kalınlığıdır.  $N_d$  kanaldaki katkılama yoğunluğudur,  $\epsilon_r = 12.5$  GaAs'in bağlı dielektrik sabiti,  $\epsilon_0 = 8.854 \times 10^{-14}$  F/cm serbest uzay için dielektrik sabiti ve  $q = 1.602 \times 10^{-19}$  C elektron yüküdür. Kanal geçit potansiyeli kanalın enine doğrultusunda Poisson denkleminin tek boyutlu çözümünden çıkarılmıştır. Bu yaklaşım kanal boyunca oluşacak alanın fakirleşmiş bölgedeki enine doğrultudaki alana göre ihmal edilebilir olmasına dayanmaktadır. (Bu yaklaşım alışılmışın dışındaki bölgede bulunan kısa kanallı elemanlarda geçerli değildir) Çözüm için kullanılan sınır koşulu fakirleşmiş bölge sınırında elektrik alanının kaybolmasıdır. Bu koşullar altında kanal potansiyeli  $W(x)$  aşağıdaki gibi verilebilir.

$$W(x) = W_{00} [1 - b(x)/a]^2 \quad (1.12)$$

Burada  $2b(x)$ ,  $x$  pozisyonundaki fakirleşmemiş aralığın boyutudur.

Savak akımı aşağıdaki gibi yazılabilir.

$$I_{ds} = 2b(x)Z\sigma E_x(x) \quad (1.13)$$

Burada  $E_x(x) = dW/dx$  kanal boyu alan,  $Z$  geçit genişliği,  $\sigma = q\mu_0 N_d$  epitaksiyel tabakanın iletkenliği ve  $\mu_0$  düşük alan mobilitesidir.  $x=0$ 'dan  $x=L$ 'ye (1.13)'ün integrali alınır ve (1.7) - (1.10)'da tanımlanan normalize potansiyeller kullanılırsa simetrik FET için savak akımı ifadesi elde edilir.

$$I_{ds} = \frac{g_0 Z W_{00}}{L_1} f_1(s, p) \quad (1.14)$$



burada

$$f_1(s,p)=p^2-s^2-2/3(p^3-s^3) \quad (1.15)$$

ve

$$g_0=2a\sigma \quad (1.16)$$

dır. Buda asimetrik FET'e ait epitaksiyel tabakanın tabaka iletkenliğinin iki katıdır. (1.16)'da 2 katsayısı  $g_0$  ifadesini simetrik FET içinde burada anılan ifadeler geçerli olacaktır.

I. bölge (Ohmik) ile II. Bölge (doymuş) arasındaki akım sürekli kabul ederek  $L_1$  değeri hesaplanabilir. Taşıyıcı hızı doymuş bölgedeki akım

$$I_{ds}(II.Bölge)=I_{dsat}=g_0ZE_s(1-p) \quad (1.17)$$

(1.14) ile (1.17) süreksizlik noktasında hız-alan eğrisi için iki parçalı lineer yaklaşım  $L_1$  için aşağıdaki ifadeyi verir.

$$L_1=L[f_1(s,p)/\xi(1-p)] \quad (1.18)$$

burada  $\xi = E_sL/W_{00}$  'dır.

Kaynaktan savağa potansiyel düşümü, kanal boyunca elektrik alan  $x=0$ 'dan  $x=L_1$  ve  $x=L_1$ 'den  $x=L$ 'ye kadar integrali alınır, elde edilir.  $V_{ds}$  için sonuç ifade Pucel tarafından verilmiştir.

$$V_{ds} = -W_{00} \left[ \left( p^2 - s^2 \right) + \frac{2}{\pi} \left( \frac{a}{L} \right) \xi \sinh \frac{\pi L_2}{2a} \right] \quad (1.19)$$

Burada  $L_2=L-L_1$ 'dir. (1.19) ve (1.18)  $L_1$  elimine edilerek birleştirilirse ve  $p$  için çözümlerse, uygulama  $V_{gs}$  ve  $V_{ds}$  potansiyellerini verir. Sanal akım her iki bölge için (1.14) ve (1.17) 'den kolayca çıkarılabilir.

**Geçiş İletkenliği ve Çıkış İletkenliği:** (1.17)  $V_{gs}$ 'ye göre integrali alınarak geçiş iletkenliği aşağıdaki gibi bulunabilir.

$$g_m = -\frac{dI_{ds}}{dV_{gs}} = \left( I_s / W_{00} \right) f_g(s,p,\xi) \quad (1.20)$$

$f_g$  boyutsuz fonksiyonu

$$f_g(s,p,\xi) = \frac{(1-p)\cosh(\pi L_2/2a) - (1-p)}{[2p(1-p) + \xi(L_1/L)]\cosh(\pi L_2/2a) - 2p(1-p)} \quad (1.21)$$

ve  $I_s/W_{00}=4e_s e_0 v_{sat} Z/a$

Kısa kanallı devre elemanları için ,s=p ve  $L_1=0$  yaklaşımları geçerli olduğunda, $g_m$  ifadesi basitleşir.

$$g_m \approx I_s(2pW_{00}) \quad (1.22)$$

Çıkış direnci aşağıdaki gibi verilebilir.

$$R_{ds} = -dV_{ds}/dI_{ds} = (W_{00}/I_s)f_r(s,p,\xi) \quad (1.23)$$

Burada

$$f_r(s,p,\xi) = \frac{1}{1-p} \left\{ \left[ 2p(1-p) + \xi \frac{L_1}{L} \right] \cosh \pi \frac{L_2}{2a} - 2p(1-p) \right\} \quad (1.24)$$

dir.Kısa geçit uzunlukları ve orta halli savak akımları için , $r_{ds}$  ifadesi aşağıdaki gibi basitleşir.

$$R_{ds} = \frac{\pi L |V_{ds}|}{\xi a I_s} p \quad (1.25)$$

burada  $I_s = g_0 Z E_s$

$C_{gs}$ ,  $C_{gd}$  ve  $C_{ds}$  :[27]'de geçit-kaynak kapasitesi simetrik FET için aşağıdaki gibi verilmiştir.

$$C_{gs} = 2\epsilon_r \epsilon_0 2f_c(s,p,\xi) \quad (1.26)$$

burada  $f_c = f_{c1} + f_{c2} + 1.56$  ve

$$f_{c1}(s,p,\xi) = \frac{2}{f_1} \frac{L_1}{a} \left\{ f_g \left[ \frac{2p_2(1-p)^2 + f_2}{1-p} \right] - s(1-s) \right\} \quad (1.27)$$

$$f_{c2}(s,p,\xi) = 2 \frac{L_2}{a} + (1-2pf_g) \left[ 2 \frac{L}{a} \frac{p}{\xi \cosh(\pi L_2 / 2a)} + \tanh \frac{\pi L_2}{2a} \right] \quad (1.28)$$

$f_{c1}$  ve  $f_{c2}$  ifadeleri 1.bölge ve 2.bölgeden gelen katkılara gösterir çünkü geçit elektrodunun her iki ucundaki kapasitelerin etkisini nümerik terim olarak ifade eder.Geçit-kaynak kapasitesi için ifadeler çok kısa geçit uzunlukları ve kırılma (pinch-off) gerilimine çok yakın olmayan geçit kutuplamaları için oldukça basitleşir.Bu yaklaşım altında,s=p,k1=0 ve şu sonuç elde edilir.

$$C_{gs} = 2\varepsilon_r\varepsilon_0Z(L/ap+1.56) \quad (1.29)$$

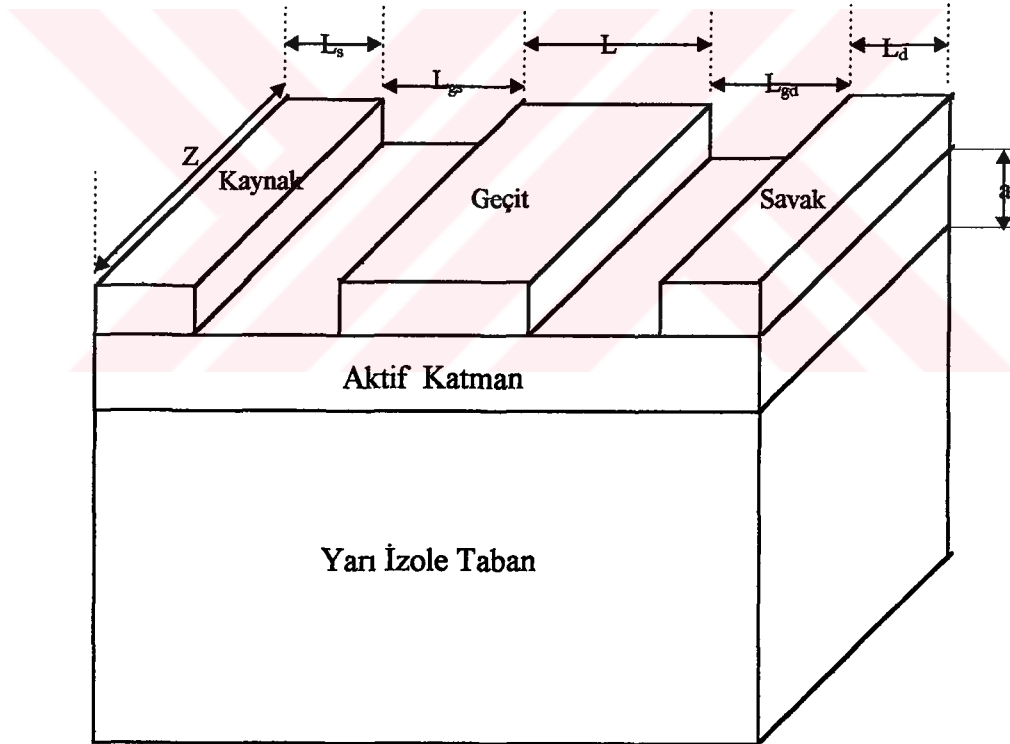
$C_{gd}$  ve  $C_{ds}$  kutuplamadan bağımsızdır. Asimetrik FET için her ikisi de aşağıdaki formda ifade edilebilir.

$$C_{gs}, C_{ds} = (\varepsilon_r + 1)\varepsilon_0Z \frac{K(1-k^2)^{1/2}}{K(k)} \quad (1.30)$$

Burada  $K(k)$  birinci tipin tam eliptik integralidir.  $k$  argümanı kapasiteler için sırasıyla aşağıdaki gibi verilebilir.

$$k_{gd} = \left[ \frac{L_{gd}}{L_{gd} + L} \right]^{1/2} \quad k_{ds} = \left[ \frac{(2L_s + L_{ds})L_{ds}}{(L_s + L_{ds})^2} \right]^{1/2} \quad (1.31)$$

burada  $L_{gd}$  ve  $L_{ds}$  Şekil 1.4'de gösterildiği gibi elektrodlar arası mesafelerdir.



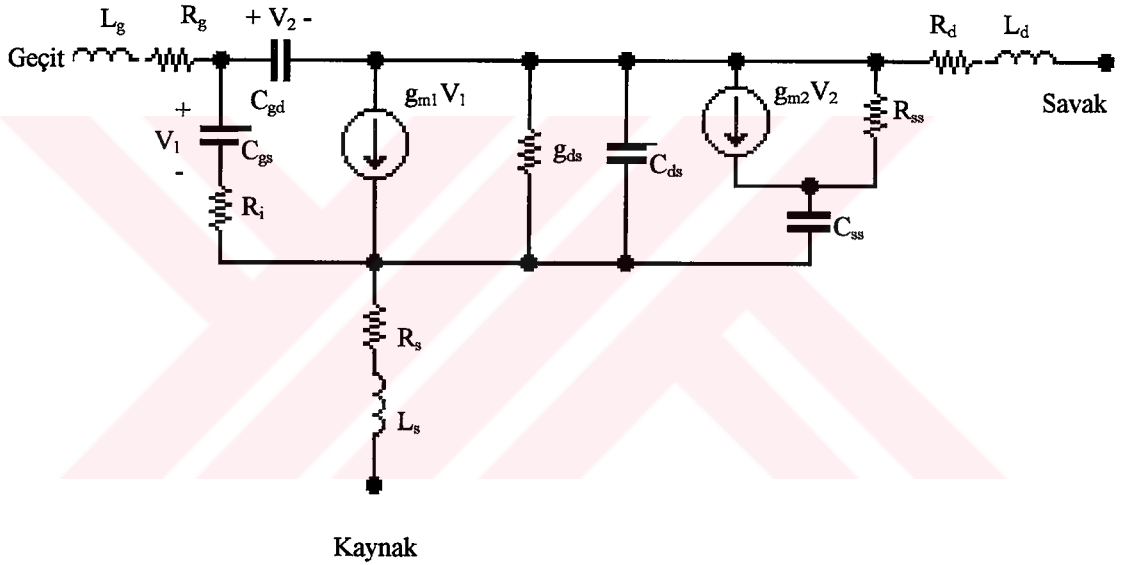
Şekil 1.4 FET yapısının önemli boyutlarını gösteren perspektifi

### 1.1.1.3 Frekans Bağımlı İletkenliklerin Modellenmesi

Mikrodalga tasarımında Şekil 1.1'de gösterilen eşdeğer devre genellikle yeterlidir. Çünkü çıkış

iletkenliği dispersiyonu tasarım frekanslarının çok aşağısında oluşur. Bununla birlikte ,büyük işaret modellemeye ve düşük frekanslı uygulamalarda ,hem dc hem de yüksek frekans cevabı tanımlanmalıdır. Bu durumlarda Şekil 1.1'deki eşdeğer devre yetersizdir.

MESFET çalışmasının bu dispersif tabiatı şekil 1.5'de verilen lineer eşdeğer devre ile modellenebilir. Şekildeki ilave akım kaynağı tuzak bölgesine sokulan akımı modeller. Tuzak bölgesi öncelikle yüzeylerde yer alır, fakat aynı zamanda devre elemanının kanal-taban ara yüzeyinde dahi tuzak bölgesi yer alır. Bu akım kanala  $C_{ss}$  kapasitesi üzerinden kuple edilir. DC işaretler için  $C_{ss}$  üzerindeki empedans sonsuz elemanın geçiş iletkenliği ve çıkış iletkenliği dc savak akımının kısmi türevleri olarak verilir. Mikrodalga çalışmalarında ,bununla birlikte, geçiş iletkenliği ve çıkış iletkenliği, dc değerlerin olduğu kadar ilave elemanlar  $g_{m2}$  ve  $R_{ss}$ 'nin de fonksiyonudur. Eşdeğer devre elemanlarının uygun seçilmesi ile ,bu eşdeğer devre, hem düşük hemde yüksek frekans iletkenliklerini aynı zamanda düşük frekanstan yüksek frekansa geçme cevabını modelleme yeteneğine (kapasitesine) sahiptir.



Şekil 1.5 Dispersiyon elemanları  $g_{m2}$ ,  $R_{ss}$  ve  $C_{ss}$  ihtiva eden FET'in küçük işaret eşdeğer devresi. Burada çıkış direnci  $R_{ds}$ , bu model için dc direnci temsil ettiği not edilmelidir.

### 1.1.2 Büyük İşaret Modeli

MESFET, HEMT ve benzeri transistörlerin elektriksel performanslarını modellemek için hem ampirik hemde fizik temelli yaklaşımlarla sonuca ulaşmak mümkündür. Bu bölümde sadece ampirik temelli Büyük-İşaret modelleme üzerinde duracağız.

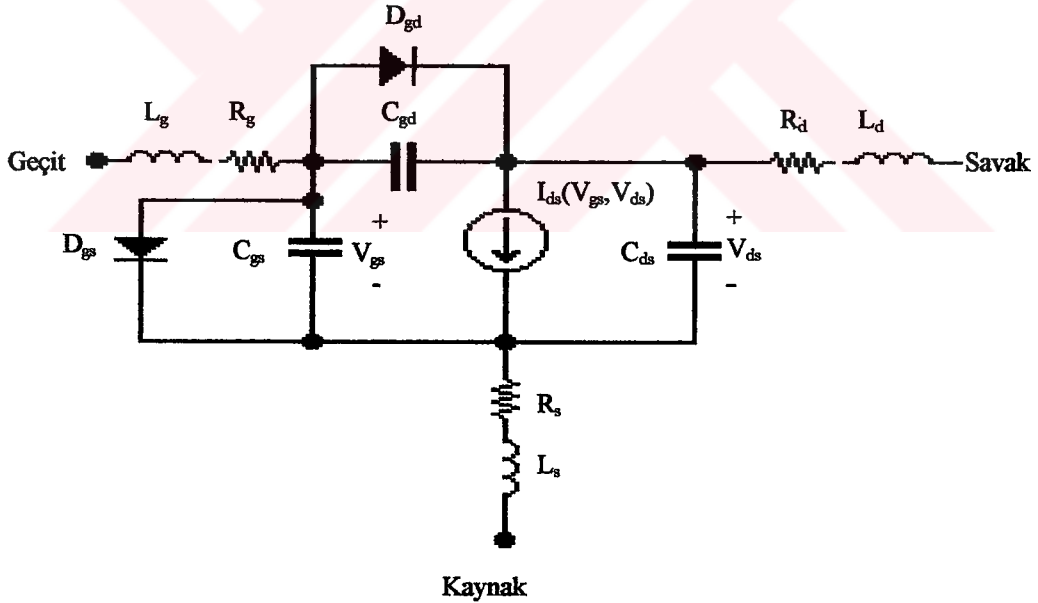
#### 1.1.2.1 Büyük İşaret Modellemeye Ampirik Yaklaşım

Birçok araştırmacı GaAs MESFET'in operasyonel karakteristiğini tasvir edebilmek için

deneyisel (ampirik) modeller geliřtirmişlerdir. Her bir modelin (mikrodalga) MESFET'in büyük işaret özelliklerini tasvir etmede bir diğeri göre bir takım üstünlükleri vardır.

Genel olarak ampirik modelleme yaklaşımında öncelikle elemanın karakteristiği ölçülür ve sonra bu şekilde (deneydeki gibi) davranacak matematiksel fonksiyon aranır. Matematiksel fonksiyon ayarlanabilir parametreler içerir ki bunlara bir kısım değerler verildiğinde fonksiyon yaklaşık olarak ölçülen bilgilere yakınsar. Analitik büyük işaret modelleri elemanın lineer olmayan özelliklerine analitik eşitliklerin kümesi kullanılarak yakınsar. Bu nonlineer özellikler eşdeğer devredeki elemanlarla ilişkilendirilebilir. Şekildeki devrenin ana nonlineer elemanları aşağıdaki gibidir.

- (1) Savak kaynak akımı  $I_{ds}$ ,  $V_{gs}$  ve  $V_{ds}$  gerilimleriyle kontrol edilen,  $\mu$  geçiş iletkenliği ve çıkış iletkenliği bundan çıkarılır.
- (2) Geçit kaynak kapasitesi  $C_{gs}$
- (3) Geçit savak kapasitesi  $C_{gd}$
- (4)  $D_{gs}$  diyodu,  $\mu$  ki bu da ileri yönde kutuplamadaki geçit akımına (veya savak-kaynak kutuplanması durumunda lineer tıkama akımı) ve
- (5)  $D_{gd}$  diyodu da savak geçit çıkış akımı (veya ters savak-kaynak kutuplanması durumunda ileri yönde iletim akımı) modellemede önemlidir.



Şekil-1.6 MESFET'in büyük işaret modelinin eşdeğer devresi

Çeşitli tipteki büyük işaret modellerinin aralarındaki fark, akım gerilim veya kapasite gerilim bağıntısına ne şekilde yaklaştığına bağlıdır. Deneysel kapasite gerilim ilişkisinin geliştirilmesi, akım gerilim ilişkisinden bağımsızdır.



İleri yönde kutuplanmış geçit diyodları  $D_{gs}$  ve  $D_{gd}$  tipik olarak Schockley eşitliği kullanılarak modellenir.

$$I_{df} = I_s (e^{\frac{V_d}{nV_T}} - 1) \quad (1.32)$$

ki burda  $V_d$  fonksiyon uygulanan gerilimdir.  $V_T$  ısı gerimidir.  $(kT/q)$ ,  $I_s$  'de ters doyma akımıdır.  $n$  ideallik faktörüdür. Diyod parametreleri  $I_s$  ve  $n$  ileri yönde kutuplamada ölçülerek çıkartılabilir.

Zener karakteristiğide diyot elemanı kullanılarak çıkarılabilir. Bu da aşağıdaki gibi verilebilir.

$$I_{dr} = I_s (e^{\frac{V_d + V_{br}}{nV_T}}) \quad V_d < -V_{br} \quad \text{iç in} \quad (1.33)$$

$V_{br}$  zener kırılma gerilimidir.

Böyle bir girişten sonra Curtice tarafından geliştirilen modelle Büyük-İşaret modellerini anlamaya çalışalım.

#### **Curtice Modeli:**

Büyük-İşaret devre simülatörlerinde ilk olarak kullanılan modellerden biri olan Büyük-İşaret MESFET modeli Van Tuyl ve Liecht (1974) tarafından önerilmiştir ve sonraları bu model Curtice (1980) tarafından basitleştirilmiştir. Curtice modeli öncelikli olarak gerilim kontrollü akım kaynağı  $I_{ds}$ , geçit-kaynak kapasitesi  $C_{gs}$ , geçit-savak kapasitesi  $C_{gd}$ , savak-kaynak kapasitesi  $C_{ds}$  ve tutma diyodu  $D_{gs}$ 'yi içeriyordu. Orijinal modelde önerilen nonlinear elemanlardan sadece  $I_{ds}$  ve  $C_{gs}$  yeni modele yer bulmuştu.  $I_{ds}$  terimi dahili (dahili parazitik dirençler ve endüktanslar) savak-kaynak ve geçit-kaynak gerilimleri ve de geçitten elektronların geçme zamanı sabiti  $\tau$ 'nin fonksiyonudur.  $C_{gs}$  terimi sadece dahili geçit-kaynak geriliminin fonksiyonu olarak düşünülmüştür. Sadece  $I_{ds}$  ve  $C_{gs}$ 'yi nonlinear elemanlar olarak kullanmak, mahmafih,  $C_{gd}$ 'nin büyük ölçüde nonlinear olan karakteristiğini ihmal eder. Bu nedenle, bir çok Büyük-İşaret devre simülatörleri,  $C_{gd}$ 'yi sadece dahili geçit-savak geriliminin fonksiyonu olarak değişen nonlinear devre elemanı olarak ihtiva eder.

Curtice modelinde savak akımı, savak-kaynak ve geçit-kaynak gerilimlerinin bir fonksiyonudur.

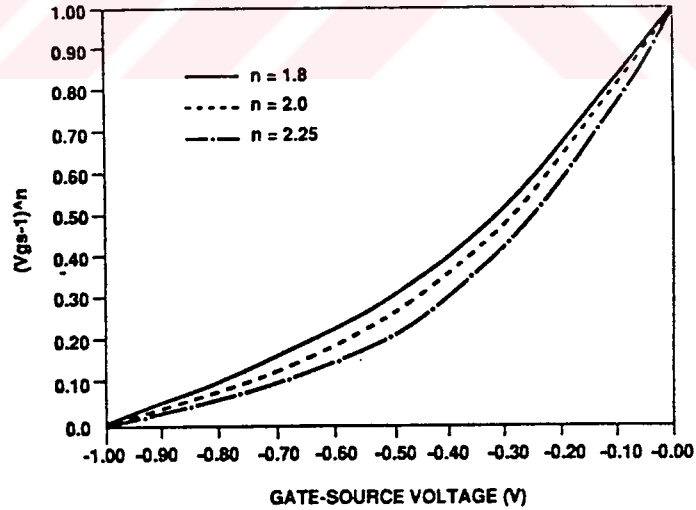
$$I_{ds}(V_{gs}, V_{ds}) = \beta (V_{gs} - V_{T0})^2 (1 + \lambda V_{ds}) \tanh(\alpha V_{ds}) \quad (1.34)$$

Burada  $V_{gs}$  ve  $V_{ds}$  dahili uç gerilimleri ve  $\beta, V_{T0}, \lambda$  ve  $\alpha$  model parametreleridir. Küçük-ışaret geçiş iletkenliği ve çıkış iletkenliği  $I_{ds}$ 'inin sırasıyla geçit-kaynak ve savak-kaynak gerilimlerine göre kısmi türevi alınarak elde edilir.

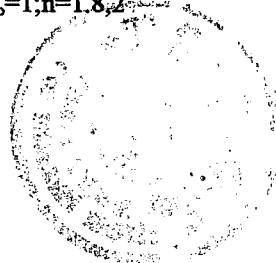
$$g_m = \frac{\partial I_{ds}}{\partial V_{gs}} = I_{ds} \left[ \frac{2}{V_{gs} - V_{To}} \right] \quad (1.35)$$

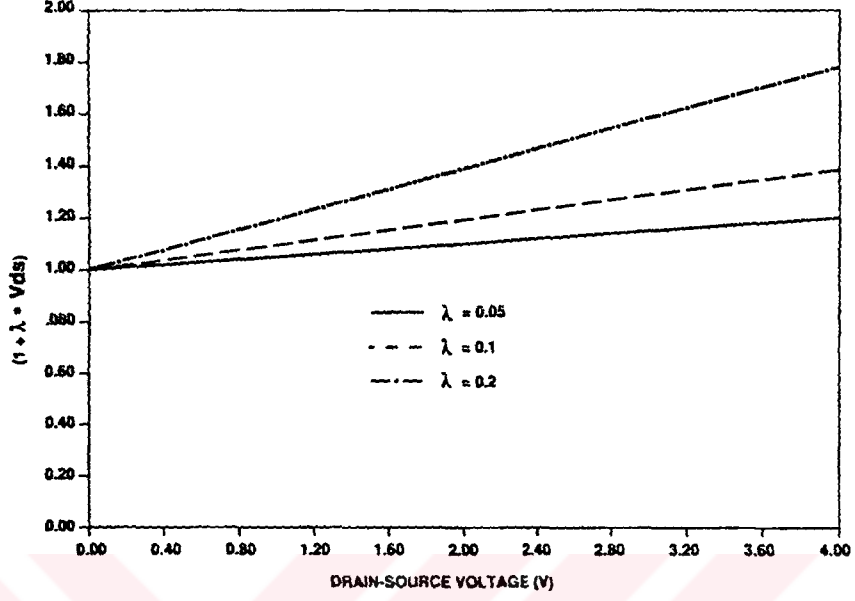
$$g_{ds} = \frac{\partial I_{ds}}{\partial V_{ds}} = \beta (V_{gs} - V_{To})^2 (1 + \lambda V_{ds}) \left( \frac{\alpha}{\cosh^2(\alpha V_{ds})} \right) + \beta (V_{gs} - V_{To})^2 \lambda \tanh(\alpha V_{ds}) \quad (1.36)$$

(1.34) eşitliği üç parçaya ayrılırsa ,savak akımının savak-kaynak gerilimi ve geçit-kaynak gerilimiyle nasıl modellendiği kolaylıkla görebiliriz.Tek tek yukarıda tanımlanan modeldeki parametreleri ve bu parametrelerin karakteristikleri nasıl etkilediklerini inceleyelim.Curtice modelindeki  $\beta(V_{gs} - V_{To})^2$  çarpanı yaklaşık olarak savak akımının geçit-kaynak gerilimine karesel bağlılığını ifade etmek için kullanılmıştır.Şekil 1.7’de geçit-kaynak geriliminin fonksiyonu olarak  $\beta(V_{gs} - V_{To})^n$  ‘in grafiği verilmiştir. $\beta$  [ $A/V^2$ ] boyutunda bir sabittir. n ise karakteristiğin eğriliğini değiştirdiği görülür.Bu modeldeki diğer çarpan ise  $(1 + \lambda V_{ds})$ dir.Buradaki  $\lambda$  parametreside elemanın çıkış iletkenliğini modeller.Bu da savak kaynak gerilimine bağlı olarak savak-kaynak akımının eğimidir.Bu grafik artan  $\lambda$  değerlerinin eğimi eğimi artırdığını göstermektedir.Bu da çıkış iletkenliğinin artmasını modellemektedir.



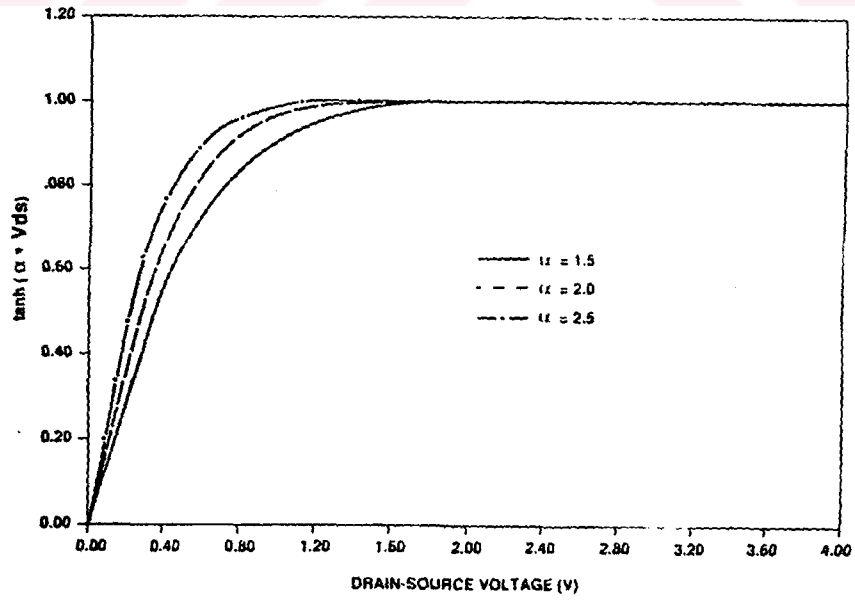
Şekil 1.7 Geçit-kaynak geriliminin fonksiyonu olarak  $\beta(V_{gs} - V_{To})^n$  ‘in grafiği ( $\beta=1, V_{To}=1; n=1.8, 2.0$  ve  $2.25$ )



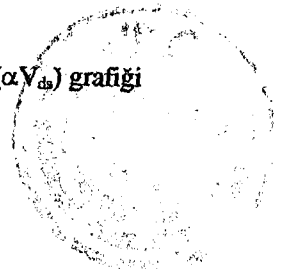


Şekil 1.8  $\lambda=0.05, 0.1$  ve  $0.2$  için  $(1+\lambda V_{ds})$  'in savak-kaynak gerilimini deęiřimi grafięidir.

En son arpan olan  $\tanh(\alpha V_{ds})$  grafięini gstermektedir. Bu grafik bize  $\alpha$  'nın artan deęerleri ile daha iyi bir řekilde akımın doyma etkisinin modellenebileceęini gstermektedir.



Şekil 1.9  $\alpha=1.5, 2$  ve  $2.5$  deęerleri iin savak-kaynak geriliminin fonksiyonu olarak  $\tanh(\alpha V_{ds})$  grafięi





(1.34) ifadesi Büyük-İşaret modelinin tamamını içermez çünkü kapasite-gerilim ilişkisinde ayrıca tanımlanmalıdır. Curtice'nin kullandığı kapasite ifadesi, yarıiletken jonksiyonu teorisinin iki uçlu Schottky yapısına uygulanmasından çıkarılmıştır. Bu ifade hem geçit-kaynak hem de geçit-savak kapasitesi için aşağıdaki gibidir.

$$C_{gs,gd} = C_{gso,gdo} \left( 1 - \frac{V_{uygulanan}}{V_{bi}} \right)^{-1/2} \quad (1.37)$$

Burada  $V_{bi}$  Schottky geçitin (built-in) dahili gerilim düşümü,  $V_{uygulanan}$  ise dahili geçit kaynak ve geçit-savak gerilimidir.  $C_{gso,gdo}$  ise sıfır kutuplama koşulunda geçit-kaynak ve geçit-savak kapasitesidir. (1.37) ifadesi bir çok uygulama için önemli olabilecek savak-kaynak gerilim bağıllığını içermemektedir. Bazı modellerde kapasite gerilim ilişkisini daha iyi modellemek için eksponansiyel [(1.37)'da -1/2'ye denk olan ] ilişkiyi kullanır.

Buna akım-gerilim ve kapasite-gerilim bağlantılarına ilaveten geçit gerilimlerindeki dalgalanmalarda savak akımındaki değişimlerdeki zaman gecikmeside hesaba katılmıştır. Zaman gecikmesi akım kaynağı aşağıdaki formda olduğu düşünülerek modellenebilir.

$$I(V) - \tau \frac{dI(V)}{dt} = \left[ \frac{\partial I(V)}{\partial V_{gs}} \right] \frac{dV_{gs}}{dt} \quad (1.38)$$

Bu teknik matematiksel olarak çok kolay ifade edilmesine rağmen, simülasyon zamanında artışa neden olmasından dolayı bu son anlatılan modifikasyonu içermez.

**Diğer Belli Başlı Büyük İşaret Modellerine Toplu Bir Bakış:** Böyle bir girişten sonra yukarıda anlatılan modellerle diğer belli başlı modellerin akım gerilim bağıntıları bir tablo halinde sunulmuş. Daha sonrada bunlar arasında bir mukayeseli bir değerlendirme yapabiliriz.

Larson modeline geçmeden önce Curtice modelinde öğrendiklerimizin ışığı altında tek tek yukarıda verilen modellerdeki parametreleri ve bu parametrelerin karakteristikleri nasıl etkilediklerini inceleyelim. Curtice modelindeki  $\beta(V_{gs} - V_{To})^2$  çarpanı yaklaşık olarak savak akımının geçit-kaynak gerilimine karesel bağıllığını ifade etmek için kullanılmıştır. Şimdide Statz'ın (1987) modeline gelelim. Bu modelin bize karesel bağıntı yaklaşımının savak-kaynak akımının, geçit-kaynak geriliminin fonksiyonu olarak küçük  $V_{gs} - V_{To}$  değerlerinde geçerli olduğunu söylemektedir.  $V_{To}$  kısılma gerilimidir. Yeni ifademiz ve karesel bağıntı

$$I_{ds} \approx \frac{\beta(V_{gs} - V_{To})^2}{1 + b(V_{gs} - V_{To})} \quad (1.39); \quad I_{ds} \approx \beta(V_{gs} - V_{To})^2 \quad (1.40)$$

şeklindedir. Bu ifadeden küçük ( $V_{gs}-V_{To}$ ) için yeni ifadenin karesel bağıntıyla özdeşleştiği görülür. Budurum Şekil 1.10'da izlenebilir.

Tablo 1.1 Belli Başlı Büyük-İşaret Modellerine Toplu Bir Bakış

MODELLER	$I_{ds}$
Curtice Modeli	$\beta(V_{gs} - V_{To})^2 (1 + \lambda V_{ds}) \tanh(\alpha V_{ds})$
Larson Modeli	$\beta(1 + \lambda k V_{ds}) \alpha k V_{ds} (1/q)^q [(1+q)(V_{gs} - V_T - k V_{ds})]^q$ ; $V_{gs} - V_T \geq k V_{ds}$ $\beta(1 + \lambda k V_{ds})(V_{gs} - V_T)^{1+q}$ ; $V_{gs} - V_T < k V_{ds}$
Statz Modeli	$\frac{\beta(V_{gs} - V_{To})^2}{1 + b(V_{gs} - V_{To})} \left[ 1 - \left( 1 - \frac{\alpha V_{ds}}{3} \right)^3 \right] (1 + \lambda V_{ds})$
Materka Kacprzak Modeli	$I_{dss} \left( 1 - V_{gs} / V_T \right)^2 \tanh\left( \frac{\alpha V_{ds}}{V_{gs} - V_T} \right)$ ; $V_T = V_{To} + \tau V_{ds}$
Geliştirilmiş Curtice Modeli	$\beta_{eff} V_{gst}^{VGEXP} (1 + \lambda V_{ds}) \tanh(\alpha V_{ds})$ ; $\beta_{eff} = \frac{\beta}{1 + \mu_{critt} V_{gst}}$ ; $V_{gst} = V_{gs} - V_T$
Schichmann Hodges Modeli	$\beta V_{ds} [2(V_{gs} - V_T) - V_{ds}] (1 + \lambda k V_{ds})$ ; $V_{gs} - V_T \geq V_{ds}$ $\beta(V_{gs} - V_T)^2 (1 + \lambda k V_{ds})$ ; $V_{gs} - V_T < V_{ds}$

Son olarakta, dikkate değer miktarda bilgisayarın zamanını alan hiperbolik tanjant fonksiyonun yerine, polinom formu kullanılarak geliştirilmiştir.

Materka-kacprzak (1983) modelindeki  $I_{dss}$  sıfır geçit-kaynak gerilimindeki doyma akımıdır. Geliştirilmiş kısılma gerilimi  $V_T$  ile gösterilmiştir. Buna ek olarak  $\alpha$  ve  $\tau$  iki deneysel parametredir.  $\alpha$  parametresi Curtice modelindeki kullanıldığı gibidir.  $\tau$ 'da  $V_{ds}$  ile birleştirildiğinde efektif kısılma gerilimini tasvir etmektedir. Bu dört parametre  $I_{dss}$ ,  $\alpha$ ,  $\tau$  ve  $V_{To}$  ölçüm sonuçları ile uygunluk sağlamak için kullanılmıştır.

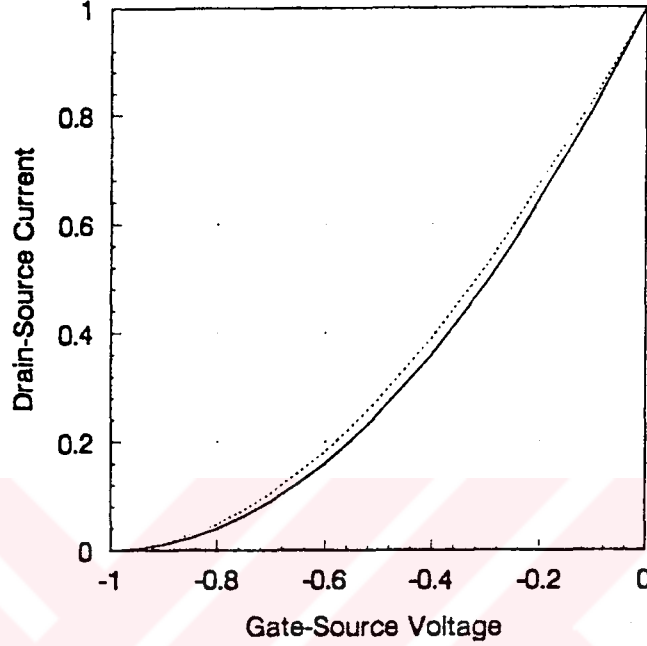
Şimdiye kadar gösterilen modellere ait  $I_{ds}$  ifadelerinin avantajı elemanın bütün bölgelerinde çalışma için akımın sürekli olması ve  $I_{ds}$  doyma gerilimi  $V_{gs}$ 'den bağımsızdır. Maalesef büyük negatif eşik gerilimine sahip elemanlarda mükemmel bir uyum göstermesine rağmen küçük negatif ve pozitif eşik gerilimlerine sahip elemanlara uygun değildir.

“Erken” doyma olayını  $I_{ds}$ 'nin  $V_{gs}$  üzerindeki bağımlılığını mükemmel bir şekilde artırılmış ve arttırılmış mod elemanlar için Larson bize teklif ettiği [33] geliştirilmiş ifade aşağıdaki gibi verilebilir.

$$\beta(1 + \lambda k V_{ds}) \alpha k V_{ds} (1/q)^q [(1+q)(V_{gs} - V_T - k V_{ds})]^q ; V_{gs} - V_T \geq k V_{ds}$$

$$\beta(1 + \lambda k V_{ds})(V_{gs} - V_T)^{1+q} ; V_{gs} - V_T < k V_{ds}$$

İki yeni faktör  $q$  ve  $k$ 'dir.  $Q$  parametresi  $I_{ds}$ 'nin  $V_g$ 'ye üstel bağlılığını modeller ve  $q=1$  için karesel bağıntıyı verir.  $k$  parametresi ise erken doyma hadisesini modeller  $k$ 'nın bir e eşit olması durumunda savak akımı doymaya  $V_{gs}-V_T$  geriliminde ulaşılacaktır.  $k$ 'nın daha büyük değerler için  $I_{ds}$  doyma gerilimi azalacaktır. Schichmann Hodges modeli Larson'un modelinin daha basit bir versiyonudur.



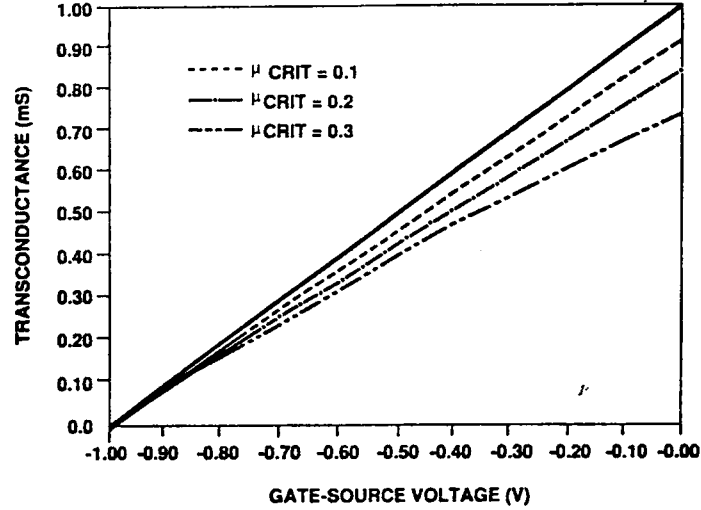
Şekil 1.10 Geçit-Kaynak geriliminin fonksiyonu olarak (1.39) ve (1.40) ifadelerinin grafik olarak karşılaştırılması. (1.39) için  $\beta=1.25, V_{To}=1$  ve  $b=0.25$ . (1.40) için  $\beta=1, V_{To}=1$ 'dir.

Geliştirilmiş Curtice (1984) modeli Curtice modelinin geliştirilmiş bir şeklidir. Bir önceki modeldeki gibi kısılma geriliminin savak-kaynak gerilimine bağlılığı hesaba katılarak geliştirme yapılmıştır. Geçiş iletkenliği parametresi  $\beta$  'da elemanın gerçek davranışına bağlı olarak geçiş iletkenliğini  $g_m$  uydurmak için geliştirilmiştir.

$$\beta_{eff} = \frac{\beta}{1 + \mu_{crit} V_{gst}} \quad (1.41)$$

ki burada  $V_{gst}=V_{gs}-V_T$ 'dir. Burda  $\mu_{crit}$  [V/cm] boyutunda kritik alan için hareket yeteneği azalmasını belirtmektedir. Doyma bölgesinde geçiş iletkenliği  $2\beta(V_{gs}-V_{To})$ 'a yakınsar. Şekil 1.11'de geçiş iletkenliğinin geçit kaynak gerilimine bağlı olarak belirlenmesindeki  $\beta_{eff}$ 'in etkisi gösterilmektedir.

Buraya incelediğimiz örnek modeller Büyük-İşaret modellemede ampirik yaklaşımın gelişiminin düşünme tarzını MESFET'ler için yeteri kadar aydınlatmaktadır.  $I_{ds}$  ifadesi bilindiği müddetçe geçiş iletkenliği ve çıkış iletkenliği (1.35) ve (1.36)'da da gösterildiği gibi  $I_{ds}$ 'ının sırasıyla geçit-kaynak ve savak-kaynak gerilimlerine göre kısmi türevi alınarak elde edilir.



Şekil 1.11 (1.41) bağıntısını kullanarak modellenen geçiş iletkenliğinde  $\beta_{off}$ 'in etkisini gösteren grafik

**HEMT'ler İçin Büyük İşaret Modelini Geliştirme Prosedürü:** Daha önceki kısımlarda anlatıldığı gibi ,elemanın elektriksel karakteristiğinin gerilime bağlılığı tanımlanmak suretiyle GaAs MESFET'lerin büyük-İşaret performansının başarılı bir şekilde modellebildiği görülmüştür.Elemanın davranışını tanımlamak için modele katılan nonlineer elemanlardan en kritik parametre geçiş iletkenliği parametresidir.Ki bu parametre elemanın Büyük-İşaret davranışını tanımlayan özelliklerin çoğunun hassas bir şekilde kestirimini temin eder.HEMT ve MESFET'lerin elektriksel davranışları arasındaki en önemli farklılık geçiş iletkenliği parametresinin karakteristiği arasındaki farklılıktan ileri gelmektedir.

O halde şimdiye kadar geliştirilen MESFET'e ait Büyük-İşaret Modeline ait geçiş iletkenliği parametresinden hareketle HEMT'in Büyük-İşaret modeline ait geçiş iletkenliği parametresi aşağıdaki analitik ifade ile yeni elemana adapte edilir.

$$g_m = g_{mFET} - f(V_{ds})\xi(V_{gs} - V_{pf})^\psi \quad (1.42)$$

Burada  $g_{mFET}$  MESFET'in Büyük-İşaret modeli parametresi, $f(V_{ds})$   $V_{ds}$ 'nin nonlineer ayırma fonksiyonu, $V_{pf}$  geçiş iletkenliğinin azalmaya başladığı gerilim değeri, $\xi$  ve  $\psi$  azalmayı tanımlayan ampirik parametrelerdir.Bu modifikasyonun hareker noktası geçiş iletkenliğinin, geçit-kaynak geriliminin belirli bir  $V_{pf}$  değerinden büyük olması durumunda ,azalmaya başladığı gerçeğidir.Bu gerçek (1.42)'nin ikinci terimi ile hesaba katılmıştır.Azalmanın derecesi  $\psi$  üssü ile hesaba katılmıştır.

Geçiş iletkenliği (1.42) ile tanımlandıktan sonra,(1.42)'nin  $V_{gs}$ 'ye göre integrali alınarak Akım-gerilim karakteristiği yeni elemana ait modelde matematiksel olarak aşağıdaki gibi tayin edilir.

$$I_{ds} = \int g_m dV_{gs} \quad (1.43)$$



Ki bu sonuç MESFET'in standart akım terimine, geçiş iletkenliğindeki azalma terimine bağlı olarak bir modifikasyon terimi ilave edilmek suretiyle ifade edilebilirdi. Çıkış iletkenliğinin ifadesinde akım-gerilim ifadesinin  $V_{ds}$ 'ye göre türevi alınmak suretiyle çıkarılabilir.

$$G_{ds} = g_{dsFET} - g_{dsHPF} \quad (1.44)$$

$g_{dsFET}$  MESFET modeli çıkış iletkenliği ve  $g_{dsHPF}$  HEMT geçişiletkenliği azalma teriminin etkisinin bir sonucudur. Burada anlatılanlar ışığında HEMT için geçerli Curtice Modelini belirleyelim.

**HEMT Curtice Modeli:** Anlatılan prosedürü Curtice MESFET modeline uygulamak için Curtice modeli I-V ifadesini yazalım.

$$I_{dsFET} = \begin{cases} 0 & \text{eğer } V_{To} \geq V_{gs} \\ \beta (V_{gs} - V_{To})^2 (1 + \lambda V_{ds}) \tanh(\alpha V_{ds}) & \text{eğer } V_{To} < V_{gs} \end{cases} \quad (1.45)$$

Geçiş iletkenliği (1.45)'in geçit-kaynak gerilimine göre türevi alınır aşağıdaki gibi bulunur.

$$g_{mFET} = \begin{cases} 0 & \text{eğer } V_{To} \geq V_{gs} \\ 2\beta (V_{gs} - V_{To}) f(V_{ds}) & \text{eğer } V_{To} < V_{gs} \end{cases} \quad (1.46)$$

Burda  $f(V_{ds}) = \tanh(\alpha V_{ds})(1 + \lambda V_{ds})$   $V_{ds}$ 'nin nonlineer ayırma fonksiyonudur. HEMT'e ait doğrultma terimleri eklenirse elemanın çalışması tanımlanmış olur. Modifiye edilmiş geçişiletkenliği aşağıdaki gibi bulunabilir.

$$g_m = \begin{cases} g_{mFET} & \text{eğer } V_{pf} \geq V_{gs} \\ g_{mFET} - f(V_{ds}) \xi (V_{gs} - V_{pf})^\zeta & \text{eğer } V_{pf} < V_{gs} \end{cases} \quad (1.47)$$

(1.43) 'de gösterildiği gibi (1.47) işleme sokulursa akım-gerilim ifadesi aşağıdaki gibi bulunur.

$$I_{ds} = \begin{cases} I_{dsFET} & \text{eğer } V_{pf} \geq V_{gs} \\ I_{dsFET} - \frac{\xi}{\psi + 1} f(V_{ds}) (V_{gs} - V_{pf})^{\zeta+1} & \text{eğer } V_{pf} < V_{gs} \end{cases} \quad (1.48)$$

Son olarakta (1.2), (1.48)'e uygulanırsa HEMT Curtice modeline ait çıkış iletkenliği parametresinde bulunur.

$$g_{ds} = \begin{cases} g_{dsFET} & \text{eğer } V_{pf} \geq V_{gs} \\ g_{dsFET} - \frac{\xi}{\psi + 1} \frac{df(V_{ds})}{dV_{ds}} f(V_{ds}) (V_{gs} - V_{pf})^{\zeta+1} & \text{eğer } V_{pf} < V_{gs} \end{cases} \quad (1.49)$$



### 1.1.3 Modelleme Parametrelerinin Kestirimi

Şekil 1.1'de verilen eşdeğer devreyi kullanacak olursak temelde bu eşdeğer devre iki kısma ayrılır.

(i) Öz parametreleri:  $g_m, g_{ds}, C_{gs}, C_{gd}, C_{ds}, R_i$  ve  $\tau$ . Ki bunlar kutuplama koşullarına bağlıdır

(ii) Dış(parazitik) parametreler :  $R_s, L_s, R_d, L_d, R_g$  ve  $L_g$ . Bunlar ise kutuplama koşullarından bağımsızdır.

Öz devre (Şekil 1.1'de kesikli çizgi ile sınırlandırılan kısım) PI topolojisinde olduğundan dolayı elektriksel özellikleri karakterize etmek için  $Y$  admitans parametreleri kullanmak uygundur. Bu parametreler aşağıdaki gibidir. Bandler (1987)'in makalesinde bu sonuçlar görülebilir.

$$y_{11} = \frac{R_i C_{gs}^2 \omega^2}{D} + j\omega \left( \frac{C_{gs}}{D} + C_{gd} \right) \quad (1.50)$$

$$y_{12} = -j\omega C_{gd} \quad (1.51)$$

$$y_{21} = \frac{g_m e^{-j\omega\tau}}{1 + jR_i C_{gs} \omega} - j\omega C_{gd} \quad (1.52)$$

$$y_{22} = g_d + j\omega (C_{ds} + C_{gd}) \quad (1.53)$$

Burada  $D = 1 + \omega^2 C_{gs}^2 R_i^2$  dir. Matris formunda

$$\mathbf{Y}_{öz} = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \quad (1.54)$$

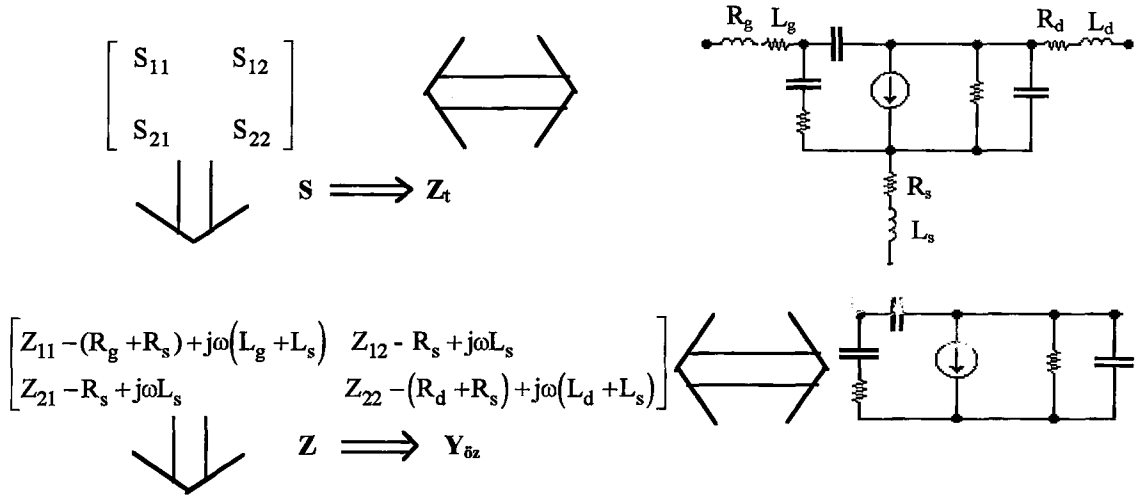
şeklinde verilebilir. Devre elemanlarının tamamını karakterize etmek için dış elemanlarında hesaba katılması gerekir. (Shirakawa v.d.(1995))

$$\mathbf{Z}_t = \mathbf{Z}_{dış} + \mathbf{Y}_{öz}^{-1} \quad (1.55)$$

Burada  $\mathbf{Z}_{dış}$  dış (parazitik) elemanların toplam devreye etkisini gösterir ve aşağıdaki gibi verilir. (Shirakawa v.d. (1995))

$$\mathbf{Z}_{dış} = \begin{bmatrix} (R_g + R_s) + j\omega(L_g + L_s) & R_s + j\omega L_s \\ R_s + j\omega L_s & (R_d + R_s) + j\omega(L_d + L_s) \end{bmatrix} \quad (1.56)$$

Ölçülen S-Parametrelerinden  $\mathbf{Z}_{öz}$ 'e geçmek için izlenen yol Şekil 1.12'deki gibi özetlenebilir. [10]



Şekil 1.12 Ölçülmüş S-parametrelerinden hareketle ( $Y_{\text{öz}}$ ) öz devre admitans parametreleri matrisinin bulunması için izlenen yol.

Küçük-ışaret eşdeğer devresine ait öz parametreler, (1.50-1.53)'de tanımlı  $Y_{\text{öz}}$  parametrelerinden hareketle, özet olarak  $y_{12}$ 'den  $C_{gd}$ ,  $y_{11}$ 'den  $C_{gs}$  ve  $R_i$ ,  $y_{21}$ 'den  $g_m$  ve  $\tau$  son olarak  $y_{22}$ 'den  $g_{ds}$  ve  $C_{ds}$  olmak üzere, aşağıda tarif edildiği gibi bulunur. (Berroth ve Bosh (1990))

$$C_{gd} = -\frac{\text{Im}(y_{12})}{\omega} \quad (1.57)$$

$$C_{gs} = \frac{\text{Im}(y_{11}) - \omega C_{gd}}{\omega} \left( 1 + \frac{(\text{Re}(y_{11}))^2}{(\text{Im}(y_{11}) - \omega C_{gd})^2} \right) \quad (1.58)$$

$$R_i = \left( \frac{\text{Re}(y_{11})}{(\text{Im}(y_{11}) - \omega C_{gd})^2 + (\text{Re}(y_{11}))^2} \right) \quad (1.59)$$

$$g_m = \sqrt{\left( (\text{Re}(y_{21}))^2 + (\text{Im}(y_{21}) + \omega C_{gd})^2 \right) \left( 1 + \omega^2 C_{gs}^2 R_i^2 \right)} \quad (1.60)$$

$$\tau = \frac{1}{\omega} \arcsin \left( \frac{-\omega C_{gd} - \text{Im}(y_{21}) - \omega C_{gs} R_i \text{Re}(y_{21})}{g_m} \right) \quad (1.61)$$

$$C_{ds} = \frac{\text{Im}(y_{22}) - \omega C_{gd}}{\omega} \quad (1.62)$$

$$g_{ds} = \text{Re}(y_{22}) \quad (1.63)$$

Bu eşitlikler bütün frekans kademesinde 0 V'dan büyük savaç gerilimleri için geçerlidir. Yukarıdaki formüllerle eşdeğer devreye ait öz parametreler tesbit edilebilir. Bundan sonraki aşamada dış (parazitik) parametreler belirlenmelidir. Bunun için şekil 1.12'den yararlanılarak aşağıdaki gibi transistörün tamamını temsil eden  $Z$  parametreleri ( $Z_t$ ), öz devreye ait  $Z$  parametreleri ( $Z_{öz}$ ) cinsinden aşağıdaki gibi yazılabilir.

$$Z_{11} = z_{11} + (R_g + R_s) + j\omega(L_g + L_s) \quad (1.64)$$

$$Z_{12} = z_{12} + R_s + j\omega L_s \quad (1.65)$$

$$Z_{21} = z_{21} + R_s + j\omega L_s \quad (1.66)$$

$$Z_{22} = z_{22} + (R_d + R_s) + j\omega(L_d + L_s) \quad (1.67)$$

Büyük  $Z_{ij}$  toplam devreye ait  $Z$  parametrelerini, küçük harf  $z_{ij}$  ise öz devreye ait  $z$  parametrelerini göstermektedir. Yukarıda verilen eşitlikler yardımıyla (yüksek frekanslarda) ölçülen ve modellenen öz devrenin yardımıyla dış kondüktansların aşağıdaki gibi belirlenir.

$$\text{Im}[Z_{11\text{ölç.}}] - \text{Im}[Z_{11\text{mod.}}] = \Delta Z_{11} = \omega(L_g + L_s) \quad (1.68)$$

$$\text{Im}[Z_{12\text{ölç.}}] - \text{Im}[Z_{12\text{mod.}}] = \Delta Z_{12} = \omega L_s \quad (1.69)$$

$$\text{Im}[Z_{21\text{ölç.}}] - \text{Im}[Z_{21\text{mod.}}] = \Delta Z_{21} = \omega L_s \quad (1.70)$$

$$\text{Im}[Z_{22\text{ölç.}}] - \text{Im}[Z_{22\text{mod.}}] = \Delta Z_{22} = \omega(L_d + L_s) \quad (1.71)$$

Dış dirençler ise aşağıdaki ifadelerle belirlenebilir.

$$\text{Re}[Z_{11\text{ölç.}}] - \text{Re}[Z_{11\text{mod.}}] = \Delta Z_{11} = (R_g + R_s) \quad (1.72)$$

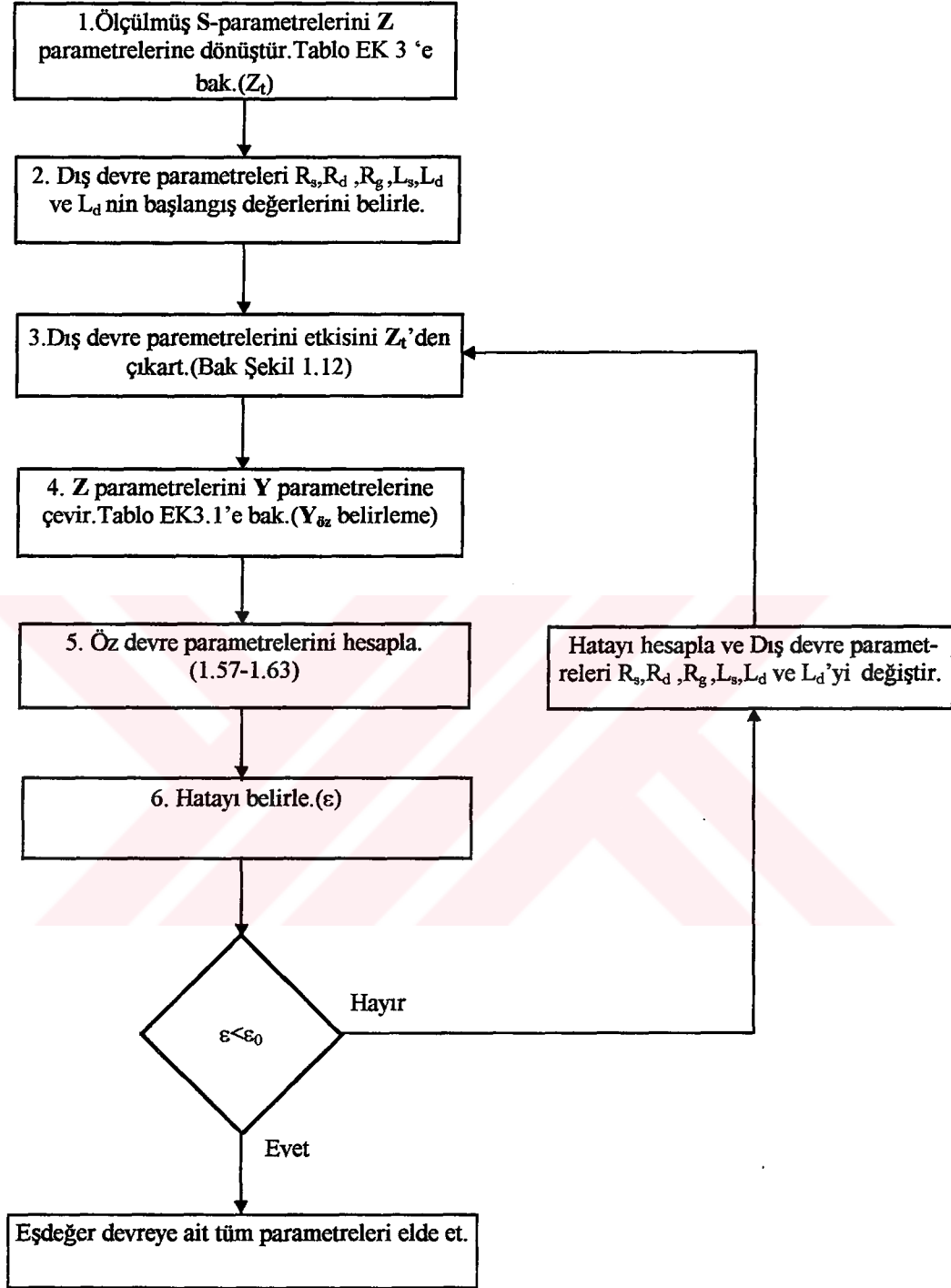
$$\text{Re}[Z_{12\text{ölç.}}] - \text{Re}[Z_{12\text{mod.}}] = \Delta Z_{12} = R_s \quad (1.73)$$

$$\text{Re}[Z_{21\text{ölç.}}] - \text{Re}[Z_{21\text{mod.}}] = \Delta Z_{21} = R_s \quad (1.74)$$

$$\text{Re}[Z_{22\text{ölç.}}] - \text{Re}[Z_{22\text{mod.}}] = \Delta Z_{22} = R_d + R_s \quad (1.75)$$

Bu kestirme tekniğinin püf noktası yeteri kadar düşük frekanslarda kondüktansların küçük-ışaret devresinin modellenmesine çok az bir etkisi olduğu gerçeğine dayanır. Buraya kadar anlatılanlardan yararlanılarak eşdeğer devreye ait tüm parametrelerin kestirimi aşağıdaki şekilde tasvir edilen algoritma ile gerçekleştirilir.





Şekil 1.13 Eşdeğer devreye ait tüm parametrelerin kestirimi algoritması.



### 1.1.4 Gürültü Modeli

#### 1.1.4.1 Gürültünün Tabiatı

Elektriksel gürültünün etkileri elektrik ve elektronik sistemlerin çıkışında gözlenir. Telefon hatlarındaki gürültü, TV ekranındaki karlanma gibi. Bunların sebebidir sistemin ilk katında işaretin kuvvetlendirilmesi sırasında dahili olarak üretilen gürültüdür. Diğer katların sistem çıkışına etkisi yok denecek kadar azdır.

**Gürültünün Tanımlanması:** Gürültü en genel olarak arzulanan işareti belirsizleştiren , etkileyip bozan , istenmeyen herhangi bir işarettir. Bu istenmeyen işaret çalışılan sisteme dışardan gelebilir. 50 Hz'lik güç hatlarında , elektrostatik ve EM kuplaj sırasında, flüoresan lambalarda olduğu gibi. Çok yakın devreler arasında ki karşılıklı haberleşmede de güç kaynaklarından veya mikrofonlardaki parçaların mekanik olarak titreşmelerinden dolayı da istenmeyen işaretler oluşabilir. Galaktik radyasyonlar ve güneş fırtınaları hariç , bunlar hep insanların sebep olduğu istenmeyen işaretlerdir. Bunlar ekranlama , filtreleme , devre elemanlarının gürültü performansı açısından geliştirilmesi ile en düşük seviyeye çekilebilir.

Görülüyor ki haberleşme sistemlerde iki ana gürültü sınıfı vardır. Birincisi iletim hattından kaynaklanır ve harici gürültü olarak adlandırılır. Galaktik, solar, dünya-yer (Earthbound) gürültüsü gibi. İkinci tip gürültü ise alıcının kendisinden kaynaklanır. Buda dahili gürültü olarak ifade edilir. Apaçık bir şekilde görüleceği gibi harici gürültünün varlığı verilen frekans bandında antenin belirli bir miktar gürültü gücü ile ( $P_{EXT}$ ) anılmasına sebep olur. Bu da kazancın antenin ışınım diagramının , antenin yere göre konumunun, antenin kuzey kutbu (gerçek kuzey) ile yaptığı açının ve alınan işaret frekansının fonksiyonudur. İkinci kategorideki gürültü kaynakları maddenin doğasından ortaya çıkmaktadır. Bu tip gürültü devrenin çalışma koşullarına da bağlıdır. Bu tip gürültü devre yapım teknolojisinin gelişmesi veya devre koşullarının değişmesi ile değişir. Fabrikasyon metodunun geliştirilmesi ile transistörlerde çok büyük miktarda gürültü azalmasına erişilebilmiştir.

Biz gürültü kelimesini elektronik sistemi meydana getiren parçaların fiziğinin bir sonucu olarak kendi kendine meydana gelen salınımlar , dalgalanmalar için veya temel rasgele gürültü kaynakları için kullanacağız . Buna mutlak sıfırın ( $-273^{\circ}K$ ) üzerindeki sıcaklıklarda iletkenlerde oluşan, görünen ısı gürültüyü örnek olarak gösterebiliriz. Bu nedenle gürültü tam kesin olarak önceden belirlenemez , hepten ihmal edilemez. Fakat hesaplanabilir.

Gürültü önemli bir büyüklüktür. Zira sistemlerin çalışma kademelerini (Dynamic range) gürültü belirler. Bir devrenin çalışabileceği en yüksek işaret seviyesini devre karakteristiği belirler . Fakat dedekte edilebilecek en düşük seviyeyi ise yine gürültü belirler. Ve yine duyargaların (sensörlerin) duyarlılığının sınırı da gürültüdür.

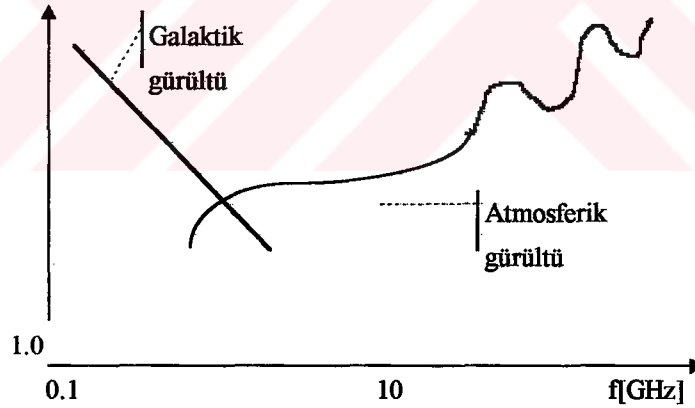
Gürültünü pek çok kaynağı vardır. Geniş olarak tabii ve yapma diye sınıflandırılabilirler. İnsan tarafından yapılanlar (man-made) esas olarak elektrik cihazlarından çıkarlar. Örneğin komütatör

motorları, ateşleme sistemlerinde bujilerden, hatalı anahtarlamalardan , elektrikli traş makinelerinden çıkan gürültü...Bu işaretler çok kere regüler özelliklere sahiptirler. Bu nedenle girişimin bir şekli olarak kabul edilebilirler.

Kıvılcımlanan bujilerin gürültü tesiri TV ekranında bütün resmi kaplayan parlak noktalar seti olarak görülebilir. İnsan tarafından yapılan gürültü üzerindeki etüd , bu gürültünün bastırıcı veya geliştirilmiş elemanlar kullanılarak kaynağında en düşük seviyeye çekilebileceği veya ortadan kaldırılabileceğini göstermiştir.

Gürültünün tabii formları kozmik radyasyondan ,atmosferden gelen gürültüyle elektronik devre elemanlarında bulunan şeklindedir. Kozmik gürültü ve atmosfer parazitleri genellikle sisteme alıcı antenlerden geçerek girerler. Kozmik gürültü, sosolar ve galaktik kaynaklardan gelen EM radyasyondan ibarettir. Solar radyasyon etüdü çok geniş bir konudur. Radyo işaretlerinin alınmasına etkileri iyi bilinmektedir. Solar radyasyonun alıcı anten üzerindeki dolaysız (direk) etkisi anten ışınma paterninin yönsel özelliklerinde değişiklik yaparak azaltılabilir.

Galaktik kaynaklarının gürültü etkisi çok daha yaygındır ve galaksi yıldızlarının radyasyonu nedeni ile oluşurlar. Özellikle belli radyo yıldızları (Cassiopeia gibi) , muntazam olarak belli bir miktar radyasyon saçarlar, ki bu, anten kaynağını başka bir yöne uygun olarak yönelterek indirgenebilir. Gök gürültüsünün toplam etkisi genellikle gök gürültü sıcaklığı ile ifade edilir. Gök gürültüsü devre gürültüsüne kıyasla genellikle küçüktür. Çok geniş frekans bandı boyunca değişir.



Şekil 1.14 Gök gürültü sıcaklığının frekansla değişimi

Şekil 1.14 'den gök gürültüsünün C bandında en az olduğu görülmektedir. Bu bölge alçak gürültü penceresi (A low noise window) diye adlandırılır. Bilhassa radyo astronomi, uzay haberleşmesi ve uydu haberleşmesi alanlarında kullanılır. Mademki radyo yıldızlarından ve uzay uydularından alınan işaretler genellikle küçüktür. O halde göğün çeşitli bölgelerinde alınan arkazemin (background) gürültüsünün mümkün mertebe alçak tutulması zorunludur. Bu arkazemin gürültüsüne 22 GHz'de su buharından, kardan saçılan radyasyondan ve 60 GHz'de atmosferde O<sub>2</sub> absorpsiyonundan dolayı ilave vardır. Ve bu etkiler Şekil 1.14' de rezonans tepeleri olarak görülmektedir. En çok problem olan tabii



gürültü devre gürültüsüdür ve genellikle ısı ve tanecik gürültüsü olarak bilinir. Isıl gürültü bir iletken içinde serbest elektronların rastgele hareketleri neticesi oluşur. Bir iletken içerisindeki titreşen atomlarla çarpışmaları nedeniyle elektron hareketi sıcaklığa bağlıdır. İletkende düzensiz değişen dalgalanmalar akım oluşturur. İletken içindeki ortalama akım sıfırdır. Fakat rastgele düzensizlikteki değişimler iletken uçlarında açık devre gürültü gerilimi oluşturur. Aynı olay alan etkili transistörün iletken kanalında da oluşur. Diğer taraftan ,tanecik gürültüsü nedeniyle, bir havası alınmış tüpte katod yüzeyinde oluşan elektron saçılmasındaki rastgele düzensiz değişimler oluşur. Aynı olay bir transistörde p-n veya metal-yarıiletken diodunda da rastlanır. Çünkü yük taşıyıcılarının potansiyel engelini geçişi rastgele bağımsız olaylar serisinden ibarettir. Tanecik gürültüsü olarak mütalaa edilir.

Devre gürültüsü, anlaşılabilirliği gibi, elektronik bileşenlerin tabiatlarının sonucudur ve ortadan kaldırılamazlar. Bu nedenle devre gürültüsü etüdü haberleşme sistem etüdünün önemli bir bölümünü teşkil eder. Bununla beraber, devre gürültüsü , elektronik bileşenleri uygun şekilde seçerek ve düşük sıcaklık kullanımı (Parametrik kuvvetlendirici gibi) ile indirgenebilir.

Bir alıcıda gürültü genellikle çok geniş bir frekans bandını kapladığından belirli bir frekans bandı seçerek onu önlemek mümkün değildir. Bu sistemde gürültü gücünün hesaplanması sistem için tanımlanmış bir band genişliğinde olur ki bu genellikle kullanılan alıcının bandı, yada etüd edilen transimiyon sisteminin band genişliğidir.

**Isıl Gürültü:** Sıcaklıkla rastgele uyarılan iletken içindeki yük taşıyıcılarının titreşimleri ısı gürültüye neden olur. Isıl gürültü ilk olarak 1927 yılında Bell telefon laboratuvarlarında çalışan J.B. Johnson tarafından gözlemlenmiştir. Teorik olarak analizi ise 1928'de H. Nyquist tarafından gerçekleştirilmiştir. Bu kişilerin çalışmalarından dolayı ısı gürültü Johnson gürültüsü ve Nyquist gürültüsü olarakta anılır.

Her iletkende mutlak sıfır sıcaklığının üzerinde elektronlar rastgele bir hareket içindedirler. Ve bu titreşim sıcaklığa bağlıdır. Her elektron  $1.59 \times 10^{-19}$  C'luk yük taşıdığından dolayı materyalin içinde elektronlar gibi rastgele hareket eden bir çok küçük akım dalgaları mevcuttur. Bu hareketler dolayısı ile oluşan akımın ortalama değeri sıfırdır. Yani net bir akım akışı yoktur. Yalnız kısa süreli olarak akımda dalgalanmalar meydana getirir. Buda iletken uçları arasındaki gerilime artış verir. Bir iletken içindeki en büyük gürültü gücü ( $N_f$ )'nin kullanılan sistemin band genişliği ve sıcaklıkla doğru orantılı olduğu bulunmuştur. Bu aşağıdaki gibi yazılabilir.

$$N_f = kT\Delta f \quad (1.76)$$

Burada

$$k = \text{Boltmann sabiti} = 1.38 \times 10^{-23} \text{ [W-sec/}^\circ\text{K]}$$

$$T = \text{İletkenin kelvin derecesi cinsinden sıcaklığı [}^\circ\text{K]}$$

$$\Delta f = \text{Kullanılan sistemin gürültü band genişliği [Hz]}$$

Oda sıcaklığında ( $290^\circ\text{K}$ ) , 1-Hz'lik bir band genişliği için (1.76) ifadesinden hesapla  $N_f = 4 \times 10^{-21}$  W bulunur. Bu 1 W referans alınırsa -204 dB'dir.



(1.76) ifadesi ile hesaplanan güç taşıyıcıların ısıl uyarılması neticesinde oluşmuştur. Başka sebeplerden dolayı da iletkende gürültü oluşabilir. Ancak burada bunları göz önüne almayacağız.

(1.76)'deki gürültü gücü band genişliği ile doğru orantılıdır. Formüle bakılınca gürültünün hesabında band genişliği önemlidir. Örneğin 1000 Hz ile 1001 Hz arasında çalışan bir sistemle 1 Hz ile 2 Hz arasında çalışan bir sistemde oluşan gürültü gücü (1.76) gereği aynı olacaktır. Isıl gürültüdeki bu sonuç beyaz gürültü olarak anılır. Gürültünün fourier analizi neticesi frekansla değişimi çizilecek olursa bunun bütün frekans bölgesinde sabit genlikli düz bir çizgiyle ifade edilebileceğini söyleyebiliriz. Aslında ısıl gürültünün beyaz ışıkla özdeşleştirilmesi tamamen doğru değildir. Zira beyaz ışık Herz başına sabit enerji değil, dalga boyu başına sabit enerji taşır.

Isıl gürültü ölçme sistemlerinin hassasiyetini sınırlar. Bir kuvvetlendirici gürültüsüz olarak gerçekleştirilmiş olsa bile işaret kaynağının iç direnci gürültü ihtiva edecektir.

(1.76) ifadesi daha kullanışlı bir şekilde sokulabilir. Biz biliyoruz ki işaret kaynağının iç direnci beslediği devrenin iç direncine eşitse o kaynaktan en büyük güç çekilebilir. Elde edilebilecek en büyük güç  $E_t^2/4R$  (Etkin değerler cinsinden) olarak ifade edilebileceğine göre : Gürültü kaynağı  $E_t$  için

$$N_t = kT\Delta f = E_t^2/4R \quad (1.77)$$

Bu nedenle R direncine sahip  $E_t$  (Etkin değere sahip) gürültü gerilimi

$$E_t = \sqrt{4kTR\Delta f} \quad (1.78)$$

Burada

R = Direnç yada iletkenin empedansının gerçel kısmıdır.

$4kT = 1.61 \times 10^{-20}$  oda sıcaklığında (290<sup>0</sup>K)

Gürültü gerilimleri ile çalışırken sık sık büyüklüklerin karesel ortalama değerlerinden söz etmek gereğini duyacağız. Isıl gürültünün karesel beklendik (Ortalama değerini)  $E[e_t^2]$  olarak göstereceğiz. Etkin değer ise karesel ortalama değerlerin kareköküdür. Bu nedenle  $E_t = \sqrt{E[e_t^2]}$  ile  $E_t^2$  eşdeğer oldukları her ikisinde karesel ortalamayı göstermek için kullanılabilirler. Isıl gürültü gerilimini aşağıdaki alternatif formda da yazabiliriz.

$$E_t^2 \text{ veya } E[e_t^2] = 4kTR\Delta f \quad (1.79)$$

(1.78,1.79) eşitliği gürültülü çalışmada oldukça önemlidir. Bunlar akılda tutulması gereken alt sınırı belirler.

(1.78,1.79)'de çok önemli birkaç sonuç çıkarılabilir. Bunlar gürültü gerilimini band genişliğinin karekökü ile doğru orantılıdır ve çalışma frekansının nereye merkezlendiğinin önemi

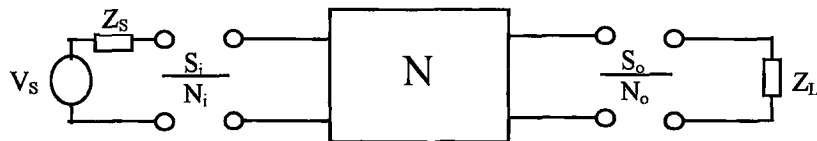
yoktur.Reaktif bileşenler ısı gürültü oluşturmaz.Eşitlik içinde kullanılan direnç basit olarak sadece devrenin ve parçanın direnci değildir.Fakat daha doğru olarak karmaşık empedansın gerçek kısmıdır . Endüktans durumunda bunu Eddy akımları içerir.Kapasite olma durumunda ise dielektrik kayıpları buna sebep olmuştur.

Bir örnek olarak 1 Hz band genişliğine sahip bir 1000  $\Omega$  'luk direncin rms gürültü gerilimi 4 nV olarak bulunabilir.Isıl gürültü haliyle en düşük seviyeye çekilmesi gerekir.(1.78,1.79) nolu eşitlikten bunun bir kaç pratik yolu olduğu görülebilir.Sistemin direnci mümkün olduğu kadar küçük tutulmalı , buna ilaveten sistemde seri direnç elemanlarından kaçınılmalıdır. Band genişliğide mümkün olduğu kadar , yani işaretin geçebileceği kadar dar tutulmalıdır.Bir sistem tasarımında frekansını sınırlamak bir sonraki çare olarak düşünülmelidir.Labaratuvar uygulamalarında frekansın sınırlandırılması dalga analizörleri , ayarlı voltmetreler ve filtrelerle sağlanabilir. Genelde frekans sınırlandırılması giriş kuplaj devreleri ve sensörler için istenmez.Bu işaret ve sensörün gürültüsünü düşürür. Ama bu kuplaj yada kaynak devresini takip eden kuvvetlendiricinin gürültüsünü düşürmez. Görüldüğü gibi her iletken zamanla değişen bir akım ve buna bağlı olarak belirli bir güç vardır. Bir iletken başka bir iletkenle bağlandığında birinden diğerine güç akışı olacaktır.Eğer 300<sup>0</sup>K'daki bir dirence paralel olarak 0<sup>0</sup>K'deki bir direnç bağlanırsa , yüksek sıcaklık dereceli dirençten düşük sıcaklıklı dirence doğru bir güç akışı olacaktır.Ta ki ısı denge kurulana kadar sıcak direnç serinlemeye soğuk direnç ise ısınmaya çalışacaktır . Bu noktadan sonra daha başka bir güç çıkışı sözkonusu olmayacaktır.Yani geçici bir güç akışı sözkonusudur.

#### 1.1.4.2 Gürültü Faktörü

Haberleşme sistemlerinde,alınan sinyaller genellikle düşük güçlü ve gürültüyle birlikte dir.Bu nedenle ilave bir kuvvetlendirme gerekmektedir ve bu hem sinyal hemde gürültü güç seviyelerini artırır. Gürültü elimine edilmediği için,sistem içinde çeşitli noktalarda sinyal gücünün gürültü gücüne oranı,genellikle performansın bir kriteri olarak kullanılır.Sistem performansının bu kriteri, belli bazı aktif sistemler (Amplifikatörler ve alıcılar) için yeterli değildir.Neticesi olarak, sistem performansını ölçen diğer kriterler kullanılması gerekli bulunmuştur.

En çok kullanılan kriter,gürültü faktörü (noise figure)' dir. Genellikle bir sistemin (ki bu bir amplifikatör yada alıcı olabilir) gürültülülüğünü biri giriş kapısında diğeri çıkış kapısında olmak üzere sinyal gürültü oranlarını ele alarak ölçer.



Şekil 1.15 Gürültülü iki kapılı

Eğer  $S_i/N_i$  ve  $S_o/N_o$ , sırasıyla iki kapının giriş ve çıkış kapıları için sinyal gücünün gürültü gücüne oranı ise, iki kapının içinden geçen sinyale etkisi için bir ölçü, bu iki oranı karşılaştırarak elde edilebilir. Buna göre F gürültü faktörü

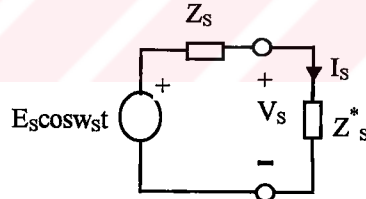
$$F = \frac{(S_i/N_i)}{(S_o/N_o)} \quad (1.80)$$

olarak tanımlanır. Burada

- $S_i$  = Elde edilebilecek en büyük giriş sinyal gücü ;
- $N_i$  = Elde edilebilecek en büyük giriş gürültü gücü ;
- $S_o$  = Elde edilebilecek en büyük çıkış sinyal gücü ;
- $N_o$  = Elde edilebilecek en büyük çıkış gürültü gücü ;

Bu tanım, devrenin giriş empedansının sinyal kaynağının uydurulmuş olduğuna işaret eder, bu esnada yük empedansını devrenin çıkış empedansına uydurulmuş olabilir yada olmayabilir.

Mahmafih, haberleşme sistemlerinin çoğunda, empedans uydurma her iki kapağa birden uygulanır, çünkü uydurma koşulları genellikle çeşitli nedenlerle kullanılır. Buna rağmen bazı özel hallerde, özel uygulamada, optimum koşulları elde etmek için, uydurulmuş empedanslar kullanılmamak daha elverişli olabilir.



Şekil 1.16 Bir sinyal kaynağından elde edilebilecek en büyük güç

Girişte elde edilecek maximum sinyal gücü  $S_i$  ,

$$S_i = \frac{1}{2} R_e [V_s I_s^*] = \frac{1}{2} |I_s|^2 R_s \quad (1.81)$$

burada  $I_s$  yerine  $E_s/2R_s$  koyarsak,  $S_i$  için  $V_s$  ve  $R_s$  cinsinden

$$S_i = \frac{|E_s|^2}{8R_s} \quad (1.82)$$

Girişteki maximum gürültü gücü ( $Z_s$  in  $T_o = 290_oK$  de tutulduğu gürültü faktörü tanımında

Girişteki maximum gürültü gücü ( $Z_g$  in  $T_0 = 290_K$  de tutulduğu gürültü faktörü tanımında kabul edilmiştir.)

$$N_i = kT_0\Delta f \quad (1.83)$$

olarak bulunmuştur. O halde  $S_i/N_i$  oranı için,

$$\frac{S_i}{N_i} = \frac{|E_s|^2}{8kT_0R_s\Delta f} \quad (1.84)$$

elde ederiz, ve çıkış ( $S_o/N_o$ ) oranı için, şayet  $F$ 'yi biliyorsak,

$$\frac{S_o}{N_o} = \frac{|E_s|^2}{8kT_0R_s\Delta f} \frac{1}{F} \quad (1.85)$$

buluruz. Burada,  $F$ 'in belli bir frekansta  $[F(f)]$  iki kapılı gürültü faktörü olduğunu ve  $F > 1$  olduğuna işaret edilir.

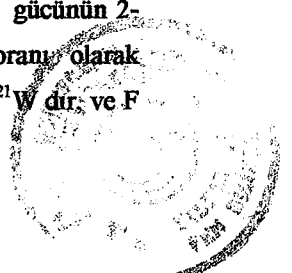
Mademki  $F$  (1.80) tanımı gereğince sistemce iki farklı noktada sinyal gücü/gürültü gücü oranını kullanır, bazen  $F$  için aşağıdaki alternatif ifadeyi kullanmak daha elverişli olur:

$$F = \frac{S_i/N_i}{S_o/N_o} = \frac{N_o}{S_o/S_i \times N_i} \quad (1.86)$$

Şayet  $G_{av}$ , iki kapılı elde edilebilecek en büyük gücü ise ise, o halde  $G_{av} = S_o/S_i$  koyarak (giriş ve çıkış empedansları uydurulmuş) ve  $N_i = kT_0\Delta f$  kullanarak,  $F$  için (1.86)'den

$$F = \frac{N_o/\Delta f}{G_{av}kT_0} = \frac{N_o}{N_{o,s}} \quad N_{o,s} = G_{av}kT_0\Delta f \quad (1.87)$$

elde ederiz. (1.87)'yi sözle ifade edelim : Bir iki-kapılı verilen bir frekansta gürültü faktörü 1-verilen frekansta birim band genişliği için çıkış kapısından elde edilebilecek en büyük toplam gücünün 2- Standart  $T_0$  ( $290_K$ ) sıcaklığındaki giriş sonlandırmasının 1-içindeki katkısına oranı olarak tanımlanabilir. İdeal iki kapılı (gürültüsüz) için  $N_o = N_{o,s}$  olduğundan,  $F=1$  dir.  $kT_0 = 4.10^{-21} W$  dir; ve  $F$  (dB) olarak ifade edilebilir, şöyleki  $F_{dB} = 10 \log F$  dir.





### 1.1.4.3 Linceer İki-Kapılıda Gürültünün Temsil Edilmesi

İçinde güç kaynakları bulundurmayan lineer bir iki-kapılı, şekil 1.17 'de verilen kapı akım ve gerilim polaritelerini referans alarak, 6 matrisden biriyle tanımlanabilir.

Örneğin  $[Z]$  matrisi bir çift denklemlerle tanımlanır:

$$V_1 = z_{11}I_1 + z_{12}I_2$$

(1.88)

$$V_2 = z_{21}I_1 + z_{22}I_2$$

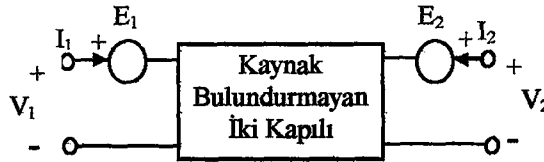
Eğer iki kapılı iç (gürültü) kaynak ihtiva ediyorsa, (1.88) aşağıdaki şekilde modifiye edilmelidir:

$$V_1 = z_{11}I_1 + z_{12}I_2 + E_1$$

(1.89)

$$V_2 = z_{21}I_1 + z_{22}I_2 + E_2$$

(1.88) ve (1.89) 'deki gerilim ve akım fonksiyonlarının frekans domenine transfer edildiğini varsayıyoruz, dolayısıyla  $V, I$  fonksiyonlar periyodik olmadığı zaman Fourier transformlarını; periyodik olduğu zaman kompleks genliklerini göstermektedir. Benzer biçimde  $E_1, E_2$  de, sırasıyla  $-e_1(t)$  ve  $e_2(t)$  stasyoner gürültü akım fonksiyonlarının yerine göre Fourier transformu yada genliğinin istatistiksel ortalamasını temsil etmektedir.  $Z$  katsayıları da frekansın fonksiyonudur. Böylece, Thevenin teoremi yardımıyla iki-kapılı içinde kaynak bulundurmayan (dolayısıyla gürültüsüz) devre ve biri giriş diğeri çıkışa seri bağlı iki kaynağa ayrılmış durumdadır. (Şekil 1.17),

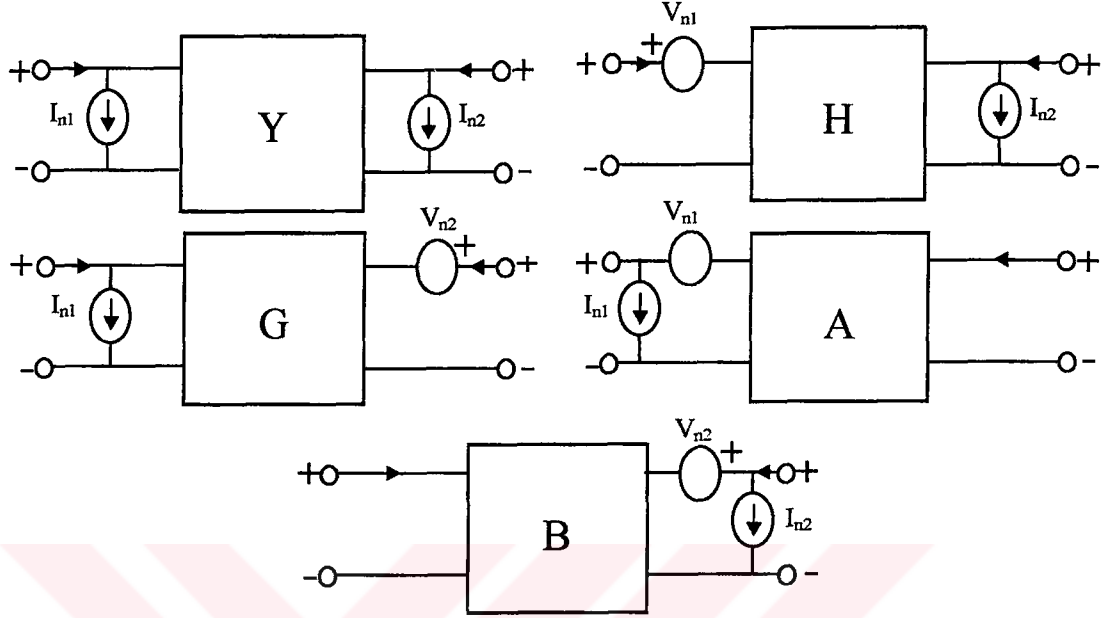


Şekil 1.17 İç kaynak bulunduran iki kapılı bir devrenin kaynak bulundurmamış iki kapılı devreye ve dış gerilim kaynaklarına ayrılması

Netice olarak iki kapılı bir devrenin gürültüsü, biri giriş diğeri çıkışa seri bağlı iki gerilim kaynağıyla tamamen temsil edilebilir. Burada şu notu kaydedelim:  $E_1$  ve  $E_2$  kaynaklarını bağımsız varsaymak için hiç bir neden yoktur, dolayısıyla herhangi bir  $\Delta f$  bandında sadece  $E_1$  ve  $E_2$ 'yi değil, kompleks korelasyonuda (ilişki)  $(E_1 E_2^*)$  vermek gereklidir.



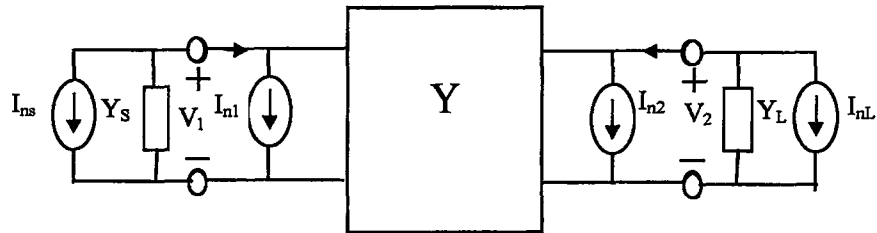
Benzer yöntemle, iki-kapılıyı karakterize eden diğer matrisler ile de devrenin iç kaynaklarını kapılarda dış kaynaklarla temsil edebiliriz. (Şekil 1.18)



Şekil 1.18 İki kapılı iç kaynaklarının kapılarda temsil edilmesi

Kullanılacak olan formun seçimi genellikle hesaplama için uygunluk mütalaasından sonra elde edilir.

Gürültülü bir iki-kapılıın bu gösterimlerini kullanarak iki-kapılı herhangi iki-uçlu arasına bağlandığında çıkış büyüklükleri tayin edebiliriz. Şekil 1.19'daki düzeni ele alalım, ki burada  $I_{n1}$  ve  $I_{n2}$  gürültü akım kaynakları ve  $Y$  matrisiyle belirtilmiş bir ikikapılı, sırasıyla  $I_{ns}$  ve  $I_{nL}$  gürültü kaynaklarıyla  $Y_S$  ve  $Y_L$ 'ye bağlansın. (Şekil 1.19)



Şekil 1.19 Gürültülü iki uçlu arasına bağlanmış gürültülü iki kapılı  
İki kapılı için

$$I_1 = y_{11} V_1 + y_{12} V_2 + I_{n1}$$

$$I_2 = y_{21} V_1 + y_{22} V_2 + I_{n2}$$

ve giriş çıkış sonlandırmaları için

$$\begin{aligned} -I_1 &= Y_s V_1 + I_{ns} \\ -I_2 &= Y_L V_2 + I_{nL} \end{aligned} \quad (1.91)$$

yazılır.(1.91) (1.90)'da yerine koyarak matris biçiminde düzenleyelim:

$$-\begin{bmatrix} I_{n1} + I_{ns} \\ I_{n2} + I_{nL} \end{bmatrix} = \begin{bmatrix} y_{11} + Y_s & y_{12} \\ y_{21} & y_{22} + Y_L \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (1.92)$$

(1.92) lineer denklem takımından  $V_1, V_2$  ve (1.91)'de yerine konularak  $I_1$  ve  $I_2$  çözülebilir. Özellikle  $I_2$  akımı için,

$$I_2 = \frac{1}{\Delta} [Y_s(Y_s + y_{11})I_{n2} - Y_L y_{21}(I_{ns} + I_{n1}) + (y_{12}y_{21} - y_{22}(Y_s + y_{11}))I_{nL}] \quad (1.93)$$

buluruz, burada  $\Delta$  devre determinantı olup, (1.92) den

$$\Delta = (y_{11} + Y_s)(y_{22} + Y_L) - y_{12}y_{21} \quad (1.94)$$

dir.

(1.93) ve (1.94) 'den  $i_2(t)$  akımının spektral güç yoğunluğu fonksiyonunu teşkil edebiliriz. Önceden ,kompleks genliğiyle verilmiş bir stasyonere gürültü akımının negatif frekanslarında içeren spektral yoğunluğu çarpı  $\Delta f$

$$S_i(f_n)df = E[|I(f_n)|^2] = \frac{1}{2}E[I(f_n)I^*(f_n)] \quad (1.95)$$

olduğunu biliyoruz. Buna göre, herhangi  $df$  frekans bandında  $i_2(t)$ 'nin taşıdığı net güç

$$S_{i2}(f) = E[|I_2|^2] = \frac{1}{2\Delta^2} \left[ |Y_L(Y_s + y_{11})|^2 + E[|I_{n2}|^2] + |Y_L y_{21}|^2 (E[|I_{n1}|^2] + E[|I_{ns}|^2]) + y_{21}y_{12} - (Y_s + y_{11})y_{22} E[|I_{nL}|^2] + 2|Y_L|^2 \text{Re}\{Y_{21}^*(Y_s + y_{11})E[I_{n1}I_{n2}^*]\} \right] \quad (1.96)$$

Burada  $E[I_{nL}I_{ns}^*] = E[I_{nL}I_{n1}^*] = E[I_{nL}I_{n2}^*] = E[I_{ns}I_{n1}^*] = E[I_{ns}I_{n2}^*] = 0$  dır, çünkü bu kaynak bağımsız varsayıldı. (1.96)'i güç spektral yoğunlukları cinsinden yazarak, yük cinsinden akan gürültü akımının



spektral yoğunluğunu devrenin admitans matrisi yük ve kaynak terimleri cinsinden elde ederiz. Buna [Y] matrisiyle verilmiş iki kapılının gürültü matrisi aşağıdaki şekilde yazılabilir:

$$\begin{bmatrix} s_{11}(f) & s_{12}(f) \\ s_{21}(f) & s_{22}(f) \end{bmatrix} \quad \text{Y matrisiyle verilmiş bir iki kapılının gürültü matrisi}$$

Burada  $s_{11}(f)$  ve  $s_{22}(f)$   $I_{n1}$  gürültü kaynaklarını karakterize eden spektral yoğunluk fonksiyonu ve  $s_{12}(f)$ ,  $s_{21}(f)$  bu ikisinin karşıt yoğunluk fonksiyonudurlar, ki  $s_{12}(f) = s_{21}^*(f)$  dir.

**Gürültü Matrisinden Gürültü Faktörünü Tayin Etme:** Bir kuvvetlendiricinin gürültü faktörü ifadesi (1.96)da ki çıkış gürültü akımı ifadesinden bulunabilir. Buna göre  $\Delta f$  de iki kapılı çıkışından  $Y_L$  ye aktarılan güç, ( $Y_L$  gürültüsünü hesaba katmayacağız.)

$$N_o = \frac{1}{2} \text{Re} \{ E [ I_2 I_2^* ] Z_L \}$$

ğl.

$$= \frac{1}{2|\Delta|^2} \left[ |Y_S + y_{11}|^2 E [ |I_{n2}|^2 ] + |y_{21}|^2 (E [ |I_{n1}|^2 ] + E [ |I_{ns}|^2 ]) + 2 \text{Re} \{ Y_{21}^* (Y_S + y_{11}) E [ I_{n1} I_{n2}^* ] \} \right] \quad (1.97)$$

tır, ve bu toplam çıkış gürültü gücünde, kaynak gürültüsünün payı (1.96) dan,

$$\frac{\text{ğl.}}{2|\Delta|^2} [ |y_{21}|^2 E [ |I_{ns}|^2 ] ] = \frac{\text{ğl.}}{|\Delta|^2} [ |y_{21}|^2 k T_o g_s \Delta f ] \quad (1.98)$$

$N_o$

olarak bulunur. Bu nedenle gürültü faktörü  $F = \frac{N_o}{N_{o,s}}$

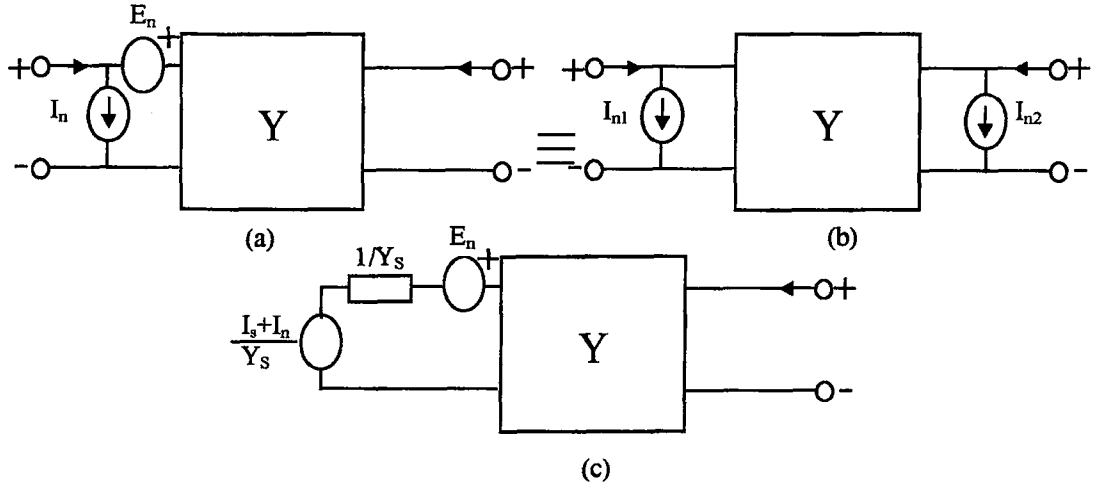
$N_{o,s}$

1

$$= 1 + \frac{1}{(g_{21}^2 + b_{21}^2) 4k T_o g_s \Delta f} \left[ \{ (g_s + g_{11})^2 + (b_s + b_{11})^2 \} E [ |I_{n2}|^2 ] + (g_{21}^2 + b_{21}^2) E [ |I_{n1}|^2 ] + \{ g_{21} (g_s + g_{11}) + b_{21} (b_s + b_{11}) \} E [ I_{n1} I_{n2}^* + I_{n1} I_{n2} ] - j \{ b_{21} (g_s + g_{11}) + g_{21} (b_s + b_{11}) \} (E [ I_{n1} I_{n2} ] - E [ I_{n1} I_{n2}^* ]) \right] \quad (1.99)$$

Şayet devre Şekil 1.18'deki bütün gürültü kaynakları giriş kapısında toplanacak şekilde düzenlenmiş ise  $E_n = I_{n2} / y_{21}$ ,  $I_n = I_{n1} - (y_{11} / y_{21}) I_{n2}$  olarak tesbit edilir. Thevenin teoremi yardımıyla devre Şekil 1.20c'deki gibi tekrar düzenlenebilir.





Şekil 1.20 Devre ve kaynak gürültü kaynaklarının giriş kapasında toplanması

Bu durumda gürültü faktörü ,gürültüsüz kuvvetlendiricinin giriş uçlarından elde edilebilecek en büyük toplam gürültü gücü nün standart sıcaklıktaki kaynak direncinden elde edilebilecek gürültü gücüne oranından bulunabilir:

$$F = \frac{E\left[\left(\frac{I_s + I_n}{Y_s} E_n\right) - \left(\frac{I_s + I_n}{Y_s} E_n\right)^*\right]}{E\left[\left(\frac{I_s}{Y_s}\right) \left(\frac{I_s}{Y_s}\right)^*\right]}$$

$$= 1 + \frac{E[|I_n|^2]}{E[|I_s|^2]} + \frac{E[|E_n|^2]}{E[|I_s|^2]} |Y_s|^2 - \frac{2\text{Re}E\{[E_n I_n^*] Y_s\}}{E[|I_s|^2]} \quad (1.100)$$

(1.100) kaynak empedansına göre optimize edilebilirerek yeniden düzenlenebilir:

$$= 1 + \frac{E[|I_n|^2]}{E[|I_s|^2]} + \frac{E[|E_n|^2]}{E[|I_s|^2]} (G_s^2 - B_s^2) - \frac{2\text{Re}E\{[E_n I_n^*]\}}{E[|I_s|^2]} G_s - j \frac{2\text{Re}E\{[E_n I_n^*]\}}{E[|I_s|^2]} B_s \quad (1.101)$$

Burada  $Y_s = G_s + jB_s$ 'dir. Optimize edebilmek için ,F'in  $G_s$ 'e ve  $B_s$ 'e göre kısmi türevlerini sıfıra eşitlemek gerekir.

$$\begin{aligned}
\frac{\partial F}{\partial G_s} &= \frac{\partial F}{\partial B_s} = 0 \\
\frac{\partial F}{\partial B_s} &= \frac{E[|E_n|^2]}{2B_s - j} \frac{E[E_n I_n^* - E_n^* I_n]}{E[|I_s|^2]} = 0 \text{ 'dan} \\
B_{s,opt} &= j \frac{E[E_n I_n^* - E_n^* I_n]}{2E[|E_n|^2]} = 0
\end{aligned} \tag{1.102}$$

Aynı zamanda

$$\begin{aligned}
\frac{\partial F}{\partial G_s} &= \frac{E[|E_n|^2]}{2G_s} - \frac{E[E_n I_n^* + E_n^* I_n]}{E[|I_s|^2]} + \frac{E[|I_n|^2] + E[|E_n|^2](G_s^2 + B_s^2)}{E[|I_s|^2]} \\
&\quad - \frac{E[E_n I_n^* + E_n^* I_n]G_s - jE[E_n I_n^* - E_n^* I_n]B_s}{(E[|I_s|^2])^2} + \frac{\partial E[|I_s|^2]}{\partial G_s} \\
\text{ve mademki } E[|I_s|^2] &= 4kT_0 G_s \Delta f \text{ dir. O halde } \frac{\partial E[|I_s|^2]}{\partial G_s} = \frac{E[|I_s|^2]}{G_s}
\end{aligned}$$

dir. Ve

$$G_{s,opt} = \left[ \frac{E[|I_n|^2]}{E[|E_n|^2]} + \frac{E[E_n I_n^* - E_n^* I_n]}{4(E[|E_n|^2])^2} \right]^{1/2} \tag{1.103}$$

elde ederiz, (1.102) ve (1.103)'ü (1.101)'de yerine koyarsak

$$F_{min} = 1 + \frac{E[|I_n|^2]}{E[|I_s|^2]} + \frac{E[E_n I_n^* + E_n^* I_n]}{E[|I_s|^2]} \left[ \frac{E[|I_n|^2]}{E[|E_n|^2]} + \frac{E[E_n I_n^* - E_n^* I_n]}{4(E[|E_n|^2])^2} \right]^{1/2} + \frac{(E[E_n I_n^* - E_n^* I_n])^2}{2E[|E_n|^2]E[|I_s|^2]} \tag{1.104}$$

#### 1.1.4.4 İki Kapılının Gürültü Parametreleri Ve Gürültü Faktörü

Daha önce, gürültülü lineer bir iki kapılının iç gürültü kaynaklarının, kapılara çıkarılabileceğini ve giriş kapısında devre gürültüsünün bir gerilim ve akım kaynağıyla toplandığı eşdeğer devrenin, işleme en elverişli olduğunu gördük. Dolayısıyla bu kaynakların tam

spesifikasyonu, iç gürültü kaynaklarının kapı gerilim ve akımlarına katkısı probleminde, iç kaynakların spesifikasyonuna eşdeğerdir. Mademki gürültü faktörü hesabında sadece çıkış akım veya gerilim dalgalanmalarının karesel ortalamaları gerekiyor, o halde kaynakları self ve karşıt spektral yoğunluk fonksiyonları belirlemek yeterlidir. (1.100'e bak) Buna göre, gürültü gerilim kaynağı dalgalanması (fluctuation) bir eşdeğer gürültü direnciyle ifade edilebilir, şöyleki:

$$E[|E_n|^2]=4kT_0R_n\Delta f \quad (1.105)$$

Burada  $R_n, E_n$  kaynağının eşdeğer gürültü direncidir.

Şekil 1.20' deki  $I_n$  gürültü kaynağı akımı bir gürültü gerilimiyle korelasyonu olmayan diğeri tam bir korelasyona sahip olmak üzere iki bileşene ayrılırsa, (1.100) gürültü faktörü ifadesi dahada basitleşebilir.  $E_n$ 'le korelasyonu olmayan bileşene  $I_u$  dersek, bir  $f$  frekansında

$$E[E_n I_u^*]=0 \quad (1.106)$$

$$E[(I_n - I_u) I_u^*]=0 \quad (1.107)$$

$E_n$  le tam korelasyona sahip gürültü akım bileşeni,  $I_n - I_u$ .  $Y_\gamma E_n$  biçiminde yazılabilir,  $Y_\gamma = G_\gamma + jB_\gamma$  korelasyon admitansı adını alır, şu halde  $E[E_n I_n^*]$  aşağıdaki şekilde yazılabilir:

$$E[E_n I_n^*]=E[E_n (I_n - I_u)^*]=Y_\gamma^* E[|E_n|^2] \quad (1.108)$$

$I_n$  gürültü kaynağının  $I_u$  bileşeni, bir  $G_u$  gürültü iletkenliği terimi cinsinden

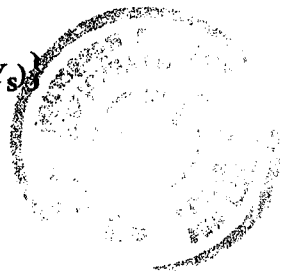
$$E[|I_u|^2]=4kT_0G_u\Delta f \quad (1.109)$$

olarak yazılabilir. Buna göre  $E[|I_n|^2]$ ,

$$E[|I_n|^2]=E[|I_n - I_u|^2]+E[|I_u|^2]=4kT_0[|Y_\gamma|^2R_n+G_u]\Delta f \quad (1.110)$$

dir. (1.105), (1.108), (1.109), (1.110) 'u (1.100) de yerlerine koyalım:

$$F=1+\frac{1}{4kT_0G_s\Delta f}\left\{4kT_0\Delta f[|Y_\gamma|^2R_n+G_u]+4kT_0\Delta fR_n|Y_s|^2-4kT_0\Delta fR_n2R_e(Y_\gamma^*Y_s)\right\}$$



$$F = 1 + \frac{1}{G_s} \left\{ G_u + R_n [G_s^2 + B_s^2 + G_\gamma^2 + B_\gamma^2 - 2(G_\gamma G_s + B_\gamma B_s)] \right\}$$

$$F = 1 + \frac{G_u}{G_s} + \frac{R_n}{G_s} [(G_s - G_\gamma)^2 + (B_s - B_\gamma)^2] \quad (1.111)$$

Böylece gürültü faktörü dört gürültü parametresinin ( $G_u$ ,  $R_n$ ,  $G_\gamma$  ve  $B_\gamma$ ) fonksiyonudur. Bu parametreler genellikle çalışma noktasının ve çalışma frekansının fonksiyonudurlar ve dış devreye (sonlandırmalara) bağlı değildir.

(1.111) de gürültü faktörü kaynak empedansına göre optimize edilir ve optimum kaynak empedansı değeri  $G_{s,opt} + jB_{s,opt}$  iki kapılı gürültü parametreleri cinsinden bulunur.  $G_{s,opt}$ ,  $B_{s,opt}$ 'nin  $G_u, R_n, G_\gamma, B_\gamma$  cinsinden ifadesi, (1.105), (1.108), (1.110)'u (1.102) ve (1.103) de kullanarak da bulunabilir:

$$B_s = j \frac{j2\text{Im}E[(E_n I_n^*)]}{2E[|E_n|^2]} = +B_\gamma \quad B_s = +B_\gamma \quad (1.112)$$

$$G_{s,opt} = \left[ \frac{4kT_0 \Delta f (|Y_\gamma|^2 R_n + G_u)}{4kT_0 \Delta f R_n} - B_\gamma^2 \right]^{1/2} \text{ den}$$

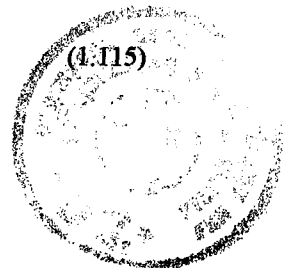
$$G_{s,opt} = \left[ \frac{G_u + R_n G_\gamma^2}{R_n} \right]^{1/2} \quad (1.113)$$

bulunur, ve gürültü faktörünün minimum değeri

$$F_{opt} = 1 + 2R_n(G_\gamma + G_{s,opt}) \quad (1.114)$$

olarak elde edilir. Gürültü faktörü  $F_{opt}, G_{s,opt}, B_{s,opt}$  ve  $R_n$  cinsinden,

$$F = F_{opt} + \frac{R_n}{G_s} [(G_s - G_{s,opt})^2 + (B_s - B_{s,opt})^2] \quad (1.115)$$





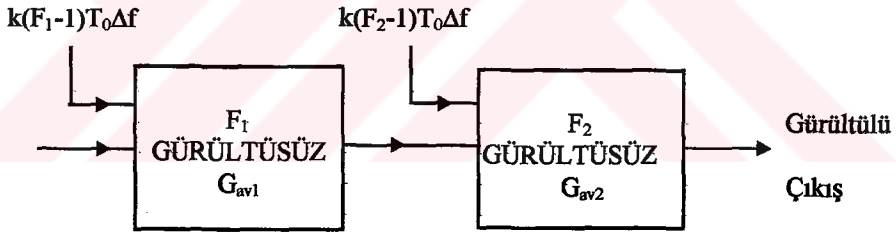
olarak yazılabilir. (1.115), bir lineer iki kapılının her giriş sonlandırması için gürültü faktörünü verir.  $F_{opt}$ ,  $G_{s,opt}$ ,  $B_{s,opt}$  ve  $R_n$  ölçme yoluyla elde edilir.

$F_{opt}$  (yada  $F_{min}$ ),  $G_{s,opt}$ ,  $B_{s,opt}$  (yada  $\Gamma_{opt}$ ) ve  $R_n$  nin verilmiş değerlerinden,  $E[|E_n|^2]$ ,  $E[|I_n|^2]$  ve  $E[E_n I_n^*]$  bulunabilir. Bunun için, önce (1.112), (1.113) ve (1.114) dan  $Y_\gamma$  ve  $G_u$  bulunur. Netice olarak, belli bir kaynak ve yükte bir lineer iki kapılının giriş yada çıkış uçlarındaki akım yada gerilimde istenen dalgalanmalar (fluctuations), iki-kapılı parametreleri ve  $E[|E_n|^2]$  ve  $E[|I_n|^2]$ ,  $E[E_n I_n^*]$  cinsinden bulunabilir. Böylece iki-kapılı kapılarda dalgalanmalar bakımından tamamen  $E[|E_n|^2]$ ,  $E[|I_n|^2]$ ,  $E[E_n I_n^*]$  yada alternatif olarak  $F_{opt}$ ,  $G_{s,opt}$ ,  $B_{s,opt}$  ve  $R_n$  cinsinden belirlenebilir.

#### 1.1.4.5 Ard Arda Bağlanmış İki-Kapılıların Gürültü Faktörü

İki veya daha fazla iki-kapılıları ( kuvvetlendiriciler gibi) daha fazla kazanç vermeleri için kaskad bağladığımızda, sistemin toplam gürültü faktörü önemli olur.

Elde edilebilecek en büyük güç kazancı  $G_{av}$  ve gürültü faktörü  $F$  olan; iki katın kaskad bağlanmasından oluşmuş bir kuvvetlendirici ele alalım, öyleki birbirinden bağımsız katların elde edilebilir en büyük kazançları ve gürültü faktörleri sırasıyla  $G_{av1}$ ,  $G_{av2}$ ;  $F_1$  ve  $F_2$  olsun.



Şekil 1.21 Kaskad bağlı iki lineer iki-kapılı

Geçen bölümde , bir lineer iki-kapılının iç gürültüsünün girişine  $S_i(f)=4k[F(f)-1]T_0\Delta f$  olan eş-değer gürültü kaynağıyla temsil edilebileceğini gördük. Buna göre ,iki bloğun iç gürültülerini girişe transfer ederek ,gürültüsüz hale getirebiliriz ve ikinci bloğun çıkışındaki elde edilebilir toplam gürültü gücünü

$$N_o = kT_0\Delta f G_{av1} G_{av2} + k(F_1-1)T_0\Delta f G_{av1} G_{av2} + k(F_2-1)T_0\Delta f G_{av2}$$

$$= kF_1 T_0\Delta f G_{av1} G_{av2} + k(F_2-1)T_0\Delta f G_{av2}$$

(1.116)

olarak yazabiliriz. Toplam iki kapılının gürültü faktörü



$$\begin{aligned}
F &= \frac{\text{Verilen iki katın toplam elde edilebilecek gürültü gücü}}{\text{Katları ideal olduğu haldeki toplam elde edilebilecek gürültü gücü}} \\
&= \frac{F_1 k T_0 \Delta f G_{av1} G_{av2} + (F_2 - 1) k T_0 \Delta f G_{av2}}{k T_0 \Delta f G_{av1} G_{av2}} \\
F &= F_1 + \frac{F_2 - 1}{G_{av1}} \tag{1.117}
\end{aligned}$$

olarak elde ederiz. Bu ifade N kaskad bağlı iki kapılılar için genişletilebilir:

$$F = F_1 + \sum_{n=2}^N \frac{F_n - 1}{G_{av1} G_{av2} \dots G_{av(n-1)}} \tag{1.118}$$

(1.118)'den, şayet birinci katın kazancı yeteri kadar büyükse, sistemin gürültü faktörüne hakim terim  $F_1$  olduğunu, ve ikinci ve daha sonraki katların gürültü faktörlerinin katkılarının nisbeten küçük olduğu sonucu çıkarılabilir. Mahmafih, eğer ilk kat düşük seviyede kazanca sahipse, ikinci katın gürültüsünün sistem gürültüsündeki payı ağırlık kazanır.



## 2. YAPAY SİNİR AĞI HAKKINDA GENEL BİLGİLER

Yapay sinir ağları, teorik hale getirilmiş zeka ve beyin faaliyetlerinin matematiksel modelleridir. Ancak biyolojik sistemler o kadar karmaşıktır ki yapay sinir ağı için kullandığımız modeller biyolojik modellerin fazlaca basite indirgenmiş biçimleri şeklindedir.

Beyin çok kompleks, nonlineer ve paralel bir (bilgi işleme sistemi) bilgisayardır. Beynin nöronları organize ederek şekil tanıma, algılama ve motor kontrolü gibi belirli hesaplamaları bugün mevcut olan en hızlı bilgisayarlardan daha hızlı yapabilme kapasitesine sahiptir. Örneğin bilgi işleme yolu olan görmeyi düşünelim. (Churcland and Sejnowski(1992), Levine(1985), Marr(1982)) Görme sisteminin fonksiyonu çevremizi tanımlamayı sağlama, daha da önemlisi, çevreyle ilişkiyi sağlayacak bilgiyi sağlamaktır. Daha belirli olarak, beyin rutin bir şekilde 100-200 ms mertebesinde somut hatırlama, algılama işlevi görür (Örneğin tanıdık olmayan bir resim içinde bilinen (tanınan) bir yüzü algılayıp tanıyabilir), daha basit karmaşıklıkta problemi en ileri bilgisayarlarla çözmek için günler gerekir. (Churcland(1986))

Yapay Sinir Ağları yada kısaca YSA insan beyninin çalışma sisteminin yapay olarak benzetimi çalışmalarının bir sonucu olarak ortaya çıkmıştır. En genel anlamda bir YSA insan beynindeki birçok nöronun yada yapay olarak basit işlemcilerin birbirlerine değişik etki seviyeleri ile bağlanması ile oluşan karmaşık bir sistem olarak düşünülebilir. Önceleri temel tıp bilimlerinde insan beynindeki nöronların matematiksel modelleme çabaları ile başlayan çalışmalar geçtiğimiz on sene içinde disipline bir şekil almıştır. YSA bugün fizik, matematik, elektrik, elektronik ve bilgisayar mühendisliği gibi çok farklı bilim dallarında araştırma konusu haline gelmiştir. YSA'nın pratik kullanımı genelde çok farklı yapıda ve formlarda bulunabilen bilgi verilerini hızla bir şekilde tanımlama ve algılama üzerinedir. Aslında mühendislik uygulamalarında YSA'nın geniş çaplı kullanımının en önemli nedeni klasik tekniklerle çözümü zor problemler için etkin bir alternatif oluşturmasıdır.

### 2.1 Biyolojik Sinir Ağları

Yapay sinir ağlarını açıklamadan önce doğal sinir ağı olan biyolojik sinir ağları hakkında bazı temel bilgileri vermekte fayda vardır. Tipik olarak, nöronlar lojik kapılara göre çok daha yavaşlardır. Silikon bir kırmağın (çip) içinde olaylar nano saniyeler ( $10^{-9}$ ) mertebesinde, nöronlar içinde olaylar mili saniye ( $10^{-3}$ ) mertebesinde oluşur. Bununla birlikte beyin oldukça yavaş bir hızda çalışmasına rağmen muazzam sayıda nöronlara (sinir hücresi) ve birbirleri arası çok geniş bir bağlantı ağına sahiptir. Gerçekte en karmaşık sinir ağı "Cerebral Cortex" denen beyindir. İnsan korteksinde yaklaşık  $10^{10}$  sinir hücresi olduğu ve 60 trilyon bağlantı veya synapsis olduğu tahmin edilmektedir. Net sonuç şudur ki insan beyninin çok etkin bir yapısı vardır. Yine hücre başına bağlantı sayısı ise  $10^4$  mertebindedir. Beyin için çalışma frekansı 100 Hz'dir. Fiziksel boyutları ise 1.3 Kg ve 0.15 m<sup>2</sup> kesitlidir.

Beynin temel yapısı sinir hücrelerinden oluşur. İç sınır hücresi ise kendi öz giriş ve çıkış bağlantılarına sahiptir. Çıkış hücreleri ise beynin bir bölgesini diğerine beyni kaslara ve duyu organlarını beyine bağlar.

### 2.1.1 Biyolojik Sinir Hücresinin Anatomisi

Bir biyolojik sinir ağını oluşturan temel öğeler şunlardır.

**Soma:** Sinir hücresinin gövdesi

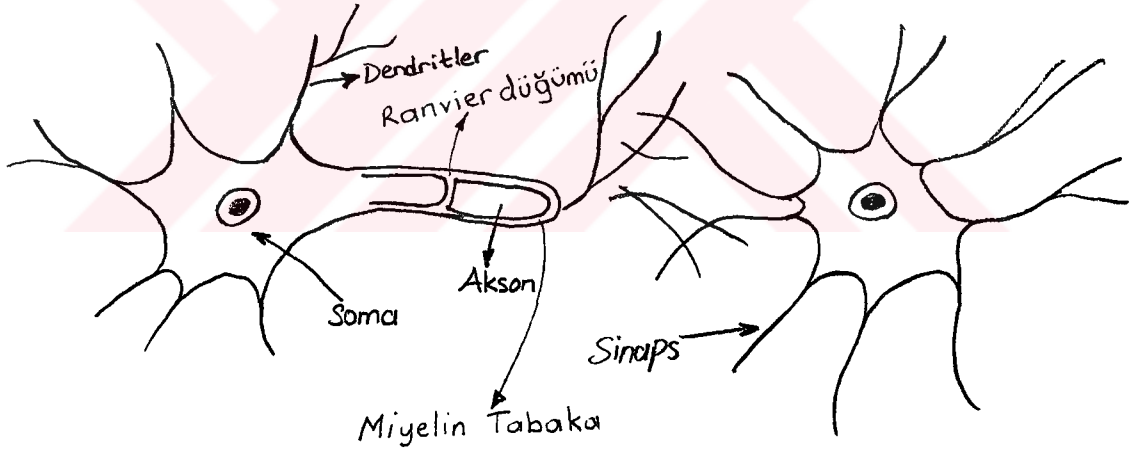
**Akson:** Çıkış darbelerinin üretildiği elektriksel olarak aktif gövde uzantısı

**Dendirit:** Diğer hücrelerden gelen işaretleri toplayan elektriksel olarak pasif hücre kolları

**Sinaps:** İki farklı hücrenin dendiritlerinin bağlantı noktaları.

**Miyelin tabaka:** Stoplazma ile hücreler arası sıvı arasındaki kapasiteyi düşürerek işaretlerin yayılma hızını arttırmaya yarayan yalıtım maddesi.

**Ranvier Düğümü:** Miyelin tabaka ile kaplı akson üzerinde her birkaç milimetrede yer alan ve işaretleri periyodik olarak yeniden üretmeye yarayan boğum. (Bakınız Şekil 2.1)



Şekil 2.1 Biyolojik Sinir Hücresinin (nöron) Basitleştirilmiş Şeması

### 2.1.2 Biyolojik Sinir Hücresinin Fizyolojisi

Hücre, stoplazma dengede iken hücreler arası sıvıya göre  $-85 \text{ mV}$ 'luk bir gerilim sınırına kadar polarizedir. Bu potansiyel farkı  $50 \text{ \AA}$  kalınlığındaki fosfolipid moleküllerini iki katmanlı olarak düzenlenmesinden oluşmuş hücre zarı üzerindedir.

Akson tarafından bir elektriksel darbenin üretilmesi için  $-40 \text{ mV}$ 'luk eşik gerilimini aşacak ve hücre zarını depolarize edecek yönde stoplazmaya akım enjekte edilmelidir. Bu darbenin biçimi darbeyi üreten sistemden bağımsızdır.

Nöron, soma adı verilen hücre gövdesi, dendirit denilen kıvrımlı uzantılar ve somanın dalları sayesinde nöronu dallarına bağlayan tek sinir fiberli aksondan oluşur. Dendiritler hücreye gelen girişleri toplarlar. Dendirit tarafından alınan işaretler hücrede birleştirilerek bir çıkış darbesi üretilip üretilmeyeceğine karar verilir. Eğer bir iş yapılacaksa üretilen çıkış darbesi aksonlar tarafından taşınarak diğer nöronlarla olan bağlantılara veya terminal organlara iletilir. Beyindeki korteksde her nöronun bir karşılığı vardır. Bir nöronun çıkışı ona bağlı olan bütün nöronlara iletilir. Fakat korteks için yapılabilmesi için hangi nöron harekete geçirilecekse sadece ona komut gönderilir.

Somanın içinde ve çevresinde  $Na^{+}$ ,  $K^{+}$ ,  $Ca^{+2}$  ve  $Cl^{-}$  iyonları vardır. Potasyum yoğunluğu nöronun içinde, Sodyum yoğunluğu dışındadır. Somanın zarı elektriksel olarak uyarılınca (Söz konusu uyarı genellikle bir gerilim düşmesidir) zar,  $Na^{+}$  ve  $Ca^{+2}$  gibi diğer iyonların içeri girmesine izin verir ve somanın iç durumunu değiştirir. Nöronlar arası bağlantılar hücre gövdesinde veya sinaps adı verilen dendiritlerdeki geçişlerde olur. Yardımcı bir benzetme aksonlarla dendiritlerin elektrik sinyallerini nörona ileten değişik empedanstaki yalıtılmış iletken olmasıdır. Sinir sistemi milyarlarca nöron ile tek bir nöronun çıkan aksomu 10000 kadar diğer nöronları bağlayan bir ağdır. Sinapslarla düzeltilen işaretleri taşıyan aksonlar ve dendiritlerle iç içe geçmiş nöronlar bir sinir ağı oluştururlar.

Eşik birimi çıkışları toplayan ve sadece girişin toplamı iç eşik değerini aştağında bir çıkış üreten işlem elemanıdır. Bir eşik birimi olarak nöron sinapslarındaki işaretleri alır ve hepsini toplar. Eğer toplanan işaret gücü eşiğı geçecek kadar güçlü ise diğer nöronları ve dendiritleri uyaran akson boyunca bir işaret gönderilir. Kesilen dendiritlerden gelen sinapslarla kapılanan bütün işaretleri soma toplar. Toplam işaret daha sonra nöronun iç eşik değeri ile karşılaştırılır ve eşik değeri aşmışsa aksona bir işaret yayar. YSA bu basit nöronların (düğümünün yada işlem elemanlarının) bağlanarak bir ağa dönüştürülmesi ile meydana getirilir.

## 2.2 Yapay Sinir Ağının Yapısı

Burada biyolojik sinir ağları hakkındaki kısa bilgilerden sonra, bir yapay sinir ağının ne olduğı ve çalışma prensipleri hakkında, konuyu anlamak bakımından bazı bilgileri fazlaca teorisine girmeden vermeye çalışacağız.

Bir YSA'nın temel ögesi hücreler yani işlem birimleridir (Bakınız Şekil 2.2). Bir YSA işlem birimleri (elemanları) ile bağlantı geometrisinden oluşan eylem birimi ile bunlara ait bir öğrenme kuralı açıklanarak tanımlanır.

YSA'nın anatomisi yönlü bir graf biçimindedir. her düğüm hücre denilen n. dereceden doğrusal olmayan (Nonlinear) dinamik bir devredir.

YSA'ların elektrik devreleri bakımından tanımını özet olarak aşağıdaki gibi sıralayabiliriz:

- 1-YSA'nın düğümleri birer işlem elemanlarıdır.
- 2-Düğüm arasındaki bağlantılar tek yönlü işaret iletim yollarıdır.
- 3- Her işlem elemanına istenildiğı kadar girişe bağlanabilir.



4-Her işlem elemanında tek bir çıkış elemanı olabilir.(Fakat bu bağlantı kopyalanabilir)

5-İşlem elemanları yerel bellek taşıyabilir.

6-Her işlem elemanının bir transfer (Aktivasyon) fonksiyonu vardır.Kabul edilebilir ki giriş işaretleri yerel bellekte bulunan işaretlerden biri olabilir.Transfer fonksiyonu sürekli yada kısmen sürekli olabilir.Kısmen sürekli çalışma konumunda aktif halde eleman bir çıkış işareti üretir.

7-Giriş işaretleri YSA'na bilgi taşır sonuç ise çıkış işaretlerinden alınır.

8-Genel işlem elemanı ancak işlem elemanı "aktif" konumunda ise giriş işaretleri ve yerel bellekteki işarete göre çıkış işareti üretir.

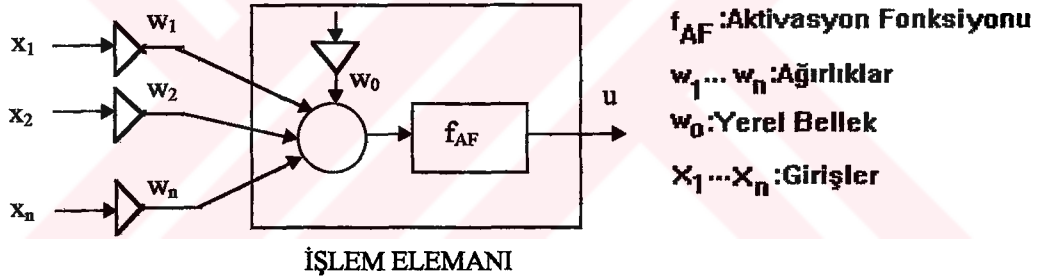
9-YSA bir takım aynı transfer fonksiyonuna sahip alt gruplara ayrılabilir.Bunlara katman denir.

10-Pekçok YSA giriş katmanı içerir,ancak bu katmanın bir transfer fonksiyonu yoktur.

11-YSA'nın transfer fonksiyonu ,yerel bellek elemanları ve bağlantı ağırlıkları,öğrenme kuralı ile giriş çıkış işareti arasındaki bağıntıya göre ayarlanır.

12-Aktif yapma girişi için bir zamanlama girişi gerekir.

13-Bağlantılarda ,taşınan işaretin cinsi tanımlanmalıdır.Bağlantıların geometrisi YSA için önemlidir.Bağlantı işareti her cinsten olabilir.



İŞLEM ELEMANI

Şekil 2.2 Bir İşlem Elemanının Yapısı

Bağlantı ağırlık katsayıları ağ geometrisi ve hücrelerin iç yapısı ve parametreleri belirli bir öğrenme kuralı ile her bir giriş olasılığına göre değiştirilmesi "on-line process" yada bütün giriş olasılıkları değerlendirildikten sonra değiştirilmesi "off-line process" olarak adlandırılır.

Bir YSA'dan beklenen görev gerçek dünyadaki nesnelere biyolojik sinir ağının yaptığı işlemi benzer bir yolla yerine getirmesidir.YSA'nın giriş veri tipleri ikili (Binary) 0-1 veya sürekli değerlerdir.Bu giriş tiplerinden başka işlem elemanlarına ait girişleri matematiksel olarak sınıflamak gerekmektedir.Çünkü bir işlem elemanına gelen girişlerin bir kısmı azaltıcı uyarma girişleri olmaktadır. Bu artırıcı ve azaltıcı girişler giriş sınıflarını oluşturur.

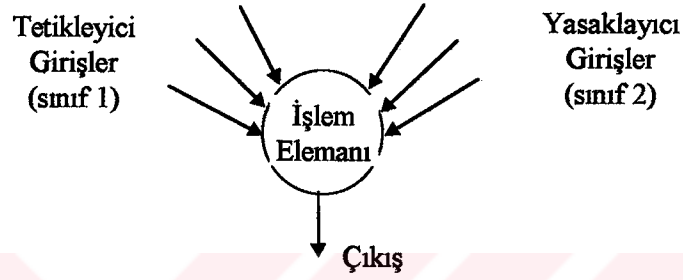
### 2.2.1 Giriş İşareti Sınıfları

İşlem elemanının(nöron) transfer fonksiyonu gelebilecek bütün giriş işaretleri için



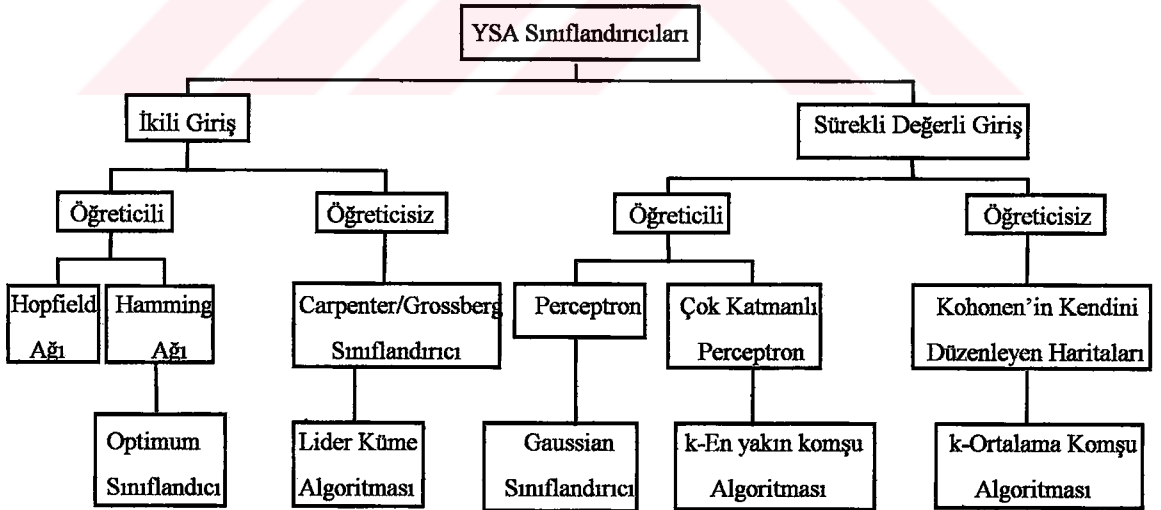
tanımlanır. Bazen değişik katmanlardan gelen işaret tiplerinin ve katman davranışlarının farklı olması doğaldır. İşaretlerin hangi bölgelerden geldiğinin bilinmesi gerekir. Değişik bölgelere göre işaretlerin sınıflarını tanımlayabiliriz.

Sıklıkla karşılaşılan bir yapı merkezde evet/çevrede hayır (on-center/off-surround) yapısıdır. Meksika şapkası bağlantı tipindedir. İşlem elemanı tetikleyici girişlerini kendine yakın komşu girişlerden, yasaklanan girişlerini ise daha uzaktan alır. Böylece işlem elemanına gelen girişler sınıflarına göre değerlendirilmiş olur. Tetikleyici bölgeden gelen girişler yasaklanan sınıfı oluşturur. Şekil 2.3 böyle bir işlem elemanını gösterir.



Şekil 2.3 Tetikleyici ve yasaklanan girişlere sahip bir işlem elemanı

Bir işlem elemanına gelen girişler matematiksel tiplerine göre sınıflandırılırlar. YSA giriş veri tiplerine göre ikili giriş (0,1) ve sürekli değerli giriş olmak üzere aşağıdaki gibi sınıflandırılır.



### 2.2.2 Bağlantı Geometrileri

Bağlantılarda taşınan işaret verisini cinsi tanımlanmalıdır. Bağlantı geometrisi YSA için çok önemlidir. Bağlantının nereden başlayıp nerede bittiğinin bilinmesi işlem elemanı kümesinin bağlantıları aşağıda tanımlandığı gibi  $n \times n$  boyutlu matris biçiminde gösterilebilir.

$$[w_{ij}] = \begin{bmatrix} w_{11} & w_{12} & w_{1n} \\ w_{21} & w_{22} & w_{2n} \\ \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & w_{nm} \end{bmatrix} \begin{array}{l} w_{ij}=w_{ji}=1 \Rightarrow i. \text{ İşlem elemanı } j. \text{ İşlem elemanına bağlı} \\ w_{ij}=w_{ji}=0 \Rightarrow \text{bağlı değil} \end{array}$$

En fazla  $N^2$  bağlantı olur. Bağlantılar çeşitli geometrik bölgeler arasında demetler halinde düşünülebilir. Bu bağlantı demetlerinin uyması gereken kurallar şunlardır.

- 1- Bağlantı demetini oluşturan işlem elemanları aynı bölgeden çıkmalıdır.,
- 2- Bağlantı demetinin işaretleri aynı matematiksel tipten olmalıdır.
- 3- Bağlantı demetinin işaretleri aynı sınıftan olmalıdır.
- 4- Bağlantı demetinin bir seçim fonksiyonu ( $\sigma$ ) olmalıdır.

$\sigma: T \rightarrow 2^S$  T:Hedef Bölgesi S:Kaynak Bölgesi

Hedef bölgesindeki her işlem elemanı kaynak bölgesindeki her elemana giderse "full" tam bağlıdır. (Örnek: Çok katmanlı Algılayıcı) Eğer her hedef bölgesi elemanı N kaynak bölgesi elemanına bağlı ise "Düzgün dağılmış" (Uniform) olasıdır. Ayrıca her bir elemana yine bir kaynak elemanı bağlı ise buna "Bire-Bir" bağlı denir.

### 2.2.3 Ağ Tipleri

Üç çeşit ağ tipi vardır.

1-İleri beslemeli ağ: Her bir katmandaki hücreler sadece bir önceki katmanın hücrelerince beslenir.

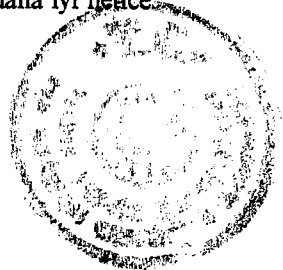
2-Kaskat bağlantılı ağ: Hücreler sadece önceki katmanlardaki hücrelerce beslenir.

3-Geri beslemeli ağ: En az bir hücre sonraki katmanlardaki hücrelercede beslenir.

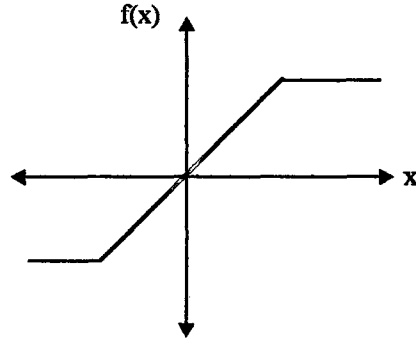
Çalışmamızda kullanılan YSA yapısının ileri beslemeli ve kaskat bağlantılı bir ağ olduğunu şimdiden hatırlatalım.

### 2.2.4 Aktivasyon Fonksiyonları

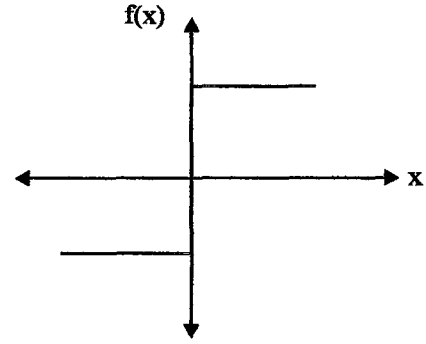
Transfer, eşik fonksiyonu olarak da adlandırılan aktivasyon fonksiyonları muhtemel sonsuz domen girişli işlem elemanlarını önceden belirlenmiş sınırdaki çıkış olarak düzenler. Dört tane yaygın aktivasyon fonksiyonu vardır. Bunlar rampa, basamak, sigmoid ve tanjant-hiperbolik tip aktivasyon fonksiyonlarıdır. Şekil 2.4'de bu fonksiyonlar gösterilmiştir. Çalışmalarımızda sigmoid ve tanjant-hiperbolik tip aktivasyon fonksiyonları üzerinde durulmuştur. Denemeler neticesinde çok daha iyi netice veren sigmoid tip aktivasyon fonksiyonunun kullanılması uygun bulunmuştur.



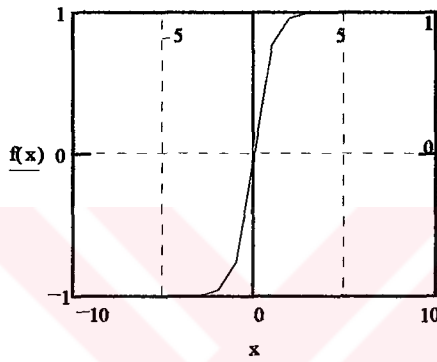




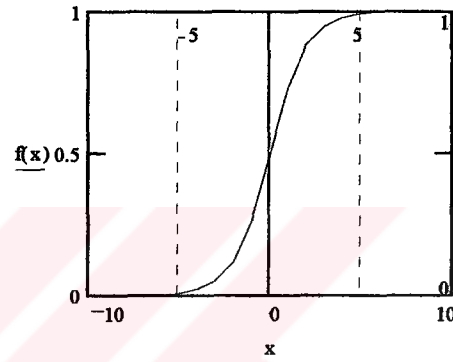
(a) Rampa



(b) Basamak



(c) Sigmoid



(d) Tanjant-hiperbolik

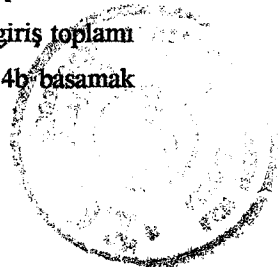
Şekil 2.4 Kullanılan Transfer (Aktivasyon) Fonksiyonları

Şekil 2.4a'da gösterilen lineer fonksiyonun denklemi  $f(x)=ax$  gibidir.  $a$ : İşlem elemanın  $x$  aktivitesini ayarlayan reel değerli bir sabittir. Lineer fonksiyon  $[-\tau, +\tau]$  sınırları arasında kısıtlandığında (a)'deki rampa aktivasyon fonksiyonu olur ve denklemi;

$$f(x) = \begin{cases} +\tau & \text{eğer } x \geq \tau \text{ ise} \\ \tau & |x| < \tau \text{ ise} \\ -\tau & \text{eğer } x \leq -\tau \text{ ise} \end{cases}$$

şeklini alır.

$+\tau$  ( $-\tau$ ) işlem elemanının maksimumu (minimumu) çoğu zaman doyma seviyesi olarak adlandırılan çıkış değeridir. Eğer eşik fonksiyonu bir giriş işaretiyle bağlı ise ve yaydığı  $+\tau$  giriş toplamı pozitif, bağlı değilse basamak tipli aktivasyon fonksiyonu  $[-\delta]$  olarak adlandırılır. Şekil 2.4b basamak tipli aktivasyon fonksiyonunu gösterir ve denklemi;



$$f(x) = \begin{cases} +\tau & \text{eğer } x \geq 0 \text{ ise} \\ -\tau & \text{diğer durumlar} \end{cases}$$

şeklinde dir. Son ve en önemli aktivasyon fonksiyonları (bu çalışmada kullanılan ) bilhassa sigmoid ve tanjant hiperbolik tip aktivasyon fonksiyonlarıdır. Sırasıyla Şekil 2.4 c ve d’de gösterilen S biçimindeki sigmoid ve tanh fonksiyonu seviyeli, lineer olmayan çıkış veren ,sınırlı,monoton artan fonksiyondur. Denklemleri sırasıyla aşağıdaki gibidir.

$$f(x) = \frac{1}{1 + e^{-x}}$$

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Her işlem elemanı kendisine verilen yerel veriye göre kendisini ayarlayacak bütün YSA’nın enformasyon bölgesinin öğrenmesini sağlar. (Enformasyon bölgesi olasılık-yoğunluk fonksiyonu ile de tanımlanabilir.) Enformasyon bölgesi bir çok uygulamada sigmoid için gerçek değerler “0” ile “1” arasında tanh için “1” ile “-1” arasında normalize edilmesi gerekir. Normalize etmek: Örneğin normalizasyon katsayısı yüz ise gerçek değeri 85 olan bir girişi 0.85 şeklinde ağa uygulamaktır. Normalizasyon aynı anda bütün girişlere uygulanabilir.

### 2.2.5 Yapay Sinir Ağlarının Kullanım Nedenleri

Öncelikle resim tanıma (Pattern Recognition) tekniğinin gerekliliği , gerçek dünya ile bilgisayar ilişkisinin başlamasıyla ortaya çıkmıştır. Bu önem YSA’nın çok güçlü örnek tanıma tekniği olarak ortaya çıkmasına ve gelişmesine neden olmuştur. Uygulamada YSA’larının kıymetini anlayabilmek için öncelikle bu konudaki algoritmaları ve geliştirme tekniklerini iyi kavramak ve özümsemek gerekir.

YSA’larının bazı uygulamalardaki kullanımı için yapılan geliştirmeler bunların bu uygulamalar için ne kadar güçlü yada yetersiz olduklarının bilinmesine bağlıdır.

YSA’ları gerçek dünyaya ait ilişkileri tanıyabilir. Sınıflandırma, kestirim (Tahmin) ve fonksiyon uydurma gibi görevleri yerine getirebilirler. Bunların genel ilişkilerle, ayrık örnekler arası boşluğu birbirlerine bir köprü gibi bağladığını da söylemek yanlış olmaz. YSA’larının ani değişen değerlerden örnek olarak, doğrudan doğruya örneklerin birbirleri arasındaki benzer ilişkileri öğrenme yetenekleri vardır. YSA’ları bir konuda veya olaydaki verilerden hareketle önceki bilgileri bilmese bile sonuç çıkarma yetenekleri de vardır. Çünkü YSA’ları klasik programlama ile çalışmazlar. Bir YSA’nın geliştirilmesi bildiğimiz yazılım geliştirmeye benzemez. Zaten aralarındaki en büyük fark ağların bir işi yapmaları için eğitime gereksinim duymalarıdır. Bildiğimiz gibi klasik programlama böyle değildir.

Bu eğitime işlemleri değişkenlerin tanımlanması çevrimlerin oluşturulması şartların tesbiti ve bunların testi ile bir derleyici üzerinde koşturma veyahutta yazılan kodun “debug” edilmesi gibi işlemleri gerektirmezler.İşte bütün bunların yerine eğitime prosedürleri seçim, analiz ve verilerin bir araya getirilmesi ile başlar.Bunlardan başka istatistik ve işaret işlemeden (Signal Processing) de bazı teknikleri kullanır.Bir YSA'nın geliştirilmesindeki ilk ve en kritik adım budur.

Bir ağ gerekli ilişkileri oluştururken ağı geliştiren kişi bunun farkına varmaz.Ağ bu ilişkileri bulup yapar ve eğer bir örnek sonuç çıkarılacaksa bunuda bulup ortaya çıkarır.Oysa klasik programlamada kodun derlenmesi sırasında kaynak programın adım adım izlenmesi mümkündür,hatta yerine göre gereklidir.Bir YSA verilerle uğraşırken yineleme çevrimi olarak bunların nasıl öğrenileceğinin kontrolünü yapmak içinde birtakım parametrelere sahiptir.İşte bu parametreler seçilen değerlerle iyi eğitilmiş bir ağın ne kadar etkili olacağını belirler.Deneyler yaparak uygun parametre değerlerinin bulunması özellikle bazı algoritmaların çok fazla hesaplama zamanı gerektirmesi nedeni ile bu ağların geliştirilmesine engel olmuştur.Hatta öyleki bir tek öğrenme işlemi çok güçlü bir iş istasyonunda bile saatler,günler veya haftalarca sürebilir.Bu yüzden paralel işlemcili mimarilerden yararlanmak yoluna gidilmiştir.Bilgisayar teknolojisindeki baş döndürücü gelişme bu problemide problem olmaktan çıkarma yolundadır.

YSA'ların yeni ortaya çıkmış bir disiplin olmasına rağmen uygulama alanları bir çok konuda hızla artmaktadır.Bazı problemlerin çözümünde oturup bir program yapmaktan daha etkin ve hızlı olabilmektedirler.Ancak bütün bunların yanında YSA'ların uygun olmadığı durumlarda sözkonusudur.Bunun en büyük sebebi eğitime işleminin hedef uzayının çok geniş olması durumunda, zor olması ve uzun zaman almasıdır.

#### *YSA kullanmak istememizin sebepleri:*

1-YSA'lar verilerden hareketle, bilinmeyen ilişkileri akıllıca hemen ortaya çıkarabiliyorlar.bu özellikleri uygulama açısından son derece önemlidir.Ayrıca veri toplam için bir ön sorgulama yada açıklama gerekmiyor.

2-Ağlar genelleştirilebilir.Bir örnekten hareketle diğer örneklerdeki benzerleri doğru olarak anlayabilirler.Genelleştirme yapılabilmesi çok önemli bir özelliğidir.Çünkü gerçek dünya verilerinde sürekli olarak gürültü ve bozucu etkiler mevcuttur.

3-YSA'ları lineer değildir.Bu özellikleri nedeni ile daha karmaşık problemleri lineer tekniklerden daha doğru çözerler.Nonlinear davranışlar hissedilir,algılanır,bilinebilir ancak bu davranışları yada problemleri matematiksel olarak çözmek zordur.

4-YSA'ları son derece paralellığe sahiptir.Bağımsız işlemleri aynı anda çok hızlı yürütebilirler.Paralel donanımlar yapıları gereği YSA'ları uygun olduğundan alternatiflerinden daha elverişlidir.Piyasada elektronik olarak gerçekleştirilmiş YSA işlemcileri mevcuttur.YSA'nın belli başlı uygulama alanları olarak görüntü birleştirme,hedef sınırlandırma,görüntü sınıflandırma,hedef tanıma,robot uygulamaları,risk analizi,sonar işaret sınıflandırma,metinden sese dönüştürme,modelleme,işlem izleme,kelime veya ses tanıma,uyarlamalı kanal denkleme sayılabilir.

İlk geliştirilen bazı YSA işlemcileri aşağıda sıralanmıştır.2.7.4 'de son yıllarda geliştirilen YSA işlencileri üzerinde de durulacaktır.

İŞLEMCI	YILI	İŞLEM ELE.SAY.	AĞ SAY.	KONUMU
PERCEPTRON	1957	8	1	DENEYSSEL
ADALINE	1962	1/8	1	TİCARİ
MARK III	1985	$8 \cdot 10^3$	1	TİCARİ
O DYSSEY	1986	$8 \cdot 10^3$	1	TİCARİ
CROSSBAR	1986	256	1	DENEYSSEL

### 2.3 Eğitim Algoritmaları

Eğitim algoritmaları YSA'nın ayrılmaz bir parçasıdır.Eğitim algoritması eldeki problemi özelliğine göre öğrenme kuralını YSA'na nasıl uyarlayacağımızı belirtir.Üç çeşit eğitim algoritması yaygın olarak kullanılmaktadır.

1-Öğreticili eğitim (Supervised training)

2-Skor ile eğitim ( Graded training)

3-Kendini düzenleme ile eğitim (Self-organization training)

Öğreticili eğitimde,elimizde doğru örnekler vardır.Yani  $(x_1, x_2, \dots, x_n)$  şeklindeki giriş vektörünün,  $(y_1, y_2, \dots, y_n)$  şeklindeki çıkış vektörü, tam ve doğru olarak bilinmektedir.Her bir  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  çifti için ağ doğru sonuçları verecek şekilde bir öğrenme kuralıyla beraber eğitilir.

Skor ile eğitimde giriş işaretlerine karşılık gelen çıkış işaretleri tam olarak bilinmemektedir.Çıkış işareti yerine skor verilir ve ağın değerlendirilmesi yapılır.Özellikle kontrol uygulamaları için idealdir.Çeşitli maliyet (cost) fonksiyonları kullanılır.

Kendini düzenleyen ağ giriş işaretine göre kendini düzenleyerek organize eder.Olasılık yoğunluk fonksiyonlarının sınıflandırma ve şekil tanıma problemlerine uygulanabilir.Ne tür eğitim yöntemi kullanılırsa kullanılsın her hangi bir ağ için gerekli karakteristik özellik ağırlıkların verilen eğitim örneğine nasıl ayarlanacağını belirtilecek şekilde öğrenme kuralının oluşturulmasıdır.Öğrenme kuralının oluşturulması için bir örneğin ağa defalarca tanıtılması gerekebilir.Öğrenme kuralı ile ilişkili parametreler ağın zaman içinde gelişme kaydetmesi ile değişebilir.Hangi YSA algoritmasında ne tür bir eğitim kullanıldığı bu bölümün giriş işaretlerinin sınıflandırılması kısmında gösterilmiştir.



## 2.4 Öğrenme Kuralları

Eşitliklerle gösterilen bu kurallar transfer fonksiyonları ve yerel bellek için kullanılır.Çeşitli YSA yapılarına göre bu kurallar seçilerek uygulanabilir.

### 2.4.1 YSA Konfigürasyonu

Bir yapay sinir ağı konfigürasyonunu belirleyen temel unsurlar şunlardır:

1. Düğüm davranışlarının tanımlanması
2. Eğitim prosedürü
3. Ağ topolojisi
4. Eğitim parametreleri (öğrenme kuralı)

Sonuçta öğrenme kuralları kendi başlarına birşey ifade etmezler.Ancak enformasyon bölgesi ve eğitime yöntemi ile anlam kazanırlar.Eşitliklerle gösterilen öğrenme kuralları transfer fonksiyonu ve yerel bellek için kullanılırlar.Düğüm davranışlarını açıklayabilmek için önce enformasyon bölgesi yani YSA'nın veri kümesinin ve ağırlık uzayının açıklamasını yapmamız gerekir.

*Enformasyon bölgesi(Hedef Uzayı):* (YSA veri kümesinin oluşturulması) Her öğrenme kuralının bir enformasyon bölgesi vardır.Bir YSA geliştirmede ilk ve en önemli adım ağ için gerekli olan veri kümesinin oluşturulmasıdır.Buradaki amaç ham verilerden toplanarak elde edilen verileri analiz edip değişkenleri seçerek veriyi ön işlemlerden geçirip ağına etkili bir şekilde öğrenmesini sağlamaktır.bir çok uygulamada işin belkide 10'da 9'u veri toplamakla ilgilidir.Sonuçta her öğrenme kuralının bir enformasyon bölgesi vardır.

Yapay sinir ağı paralel dağılmış parametrelili bir sistem olduğundan her bir işlem elemanı izole edilmiş bir ada gibi düşünülebilir.İşlem elemanı girişine uygulanan işaretleri transfer fonksiyonuna göre değerlendirip çıkış işareti üretir.

Her işlem elemanı kendisine verilen yerel veriye göre kendisini ayarlayarak, bütün yapay sinir ağı'nın tüm enformasyon bölgesini öğrenmesini sağlar.Enformasyon bölgesi olasılık yoğunluk fonksiyonu olarak tanımlanabilir.YSA paralel dağılmış parametrelili bir sistem olduğundan her bir işlem elemanı izole edilir.İşlem elemanı girişine uygulanan işaretleri transfer fonksiyonuna göre değerlendirip çıkış işareti üretir.

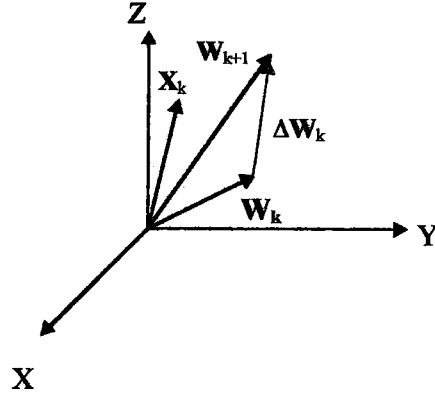
*Ağırlık Uzayı:*Bir çok YSA öğrenme işlemi işlem elemanlarının ağırlıkları değiştirilerek sağlanır.Böylece tanımlanan ağırlıklar tanımlanacak bir hedef kritere göre değiştirilerek,ağ son formunu alır.Basit bir matematiksel model olarak, her bir işlem elemanının "n" adet gerçek ağırlığı olduğu düşünülerek N adet işlem elemanı gözönüne alınırsa

$$w_1^T = (w_{11}, w_{12}, \dots, w_{1n}) \dots w_N^T = (w_{N1}, w_{N2}, \dots, w_{Nn}) \text{ olmak üzere}$$

$$w = (w_1^T, w_2^T, w_3^T, \dots, w_N^T) \quad w_1, w_2, w_3, \dots, w_N : \text{İşlem elemanlarının Ağırlık vektörleridir.}$$



YSA ağırlık vektörü  $N,n$  boyutlu orkid uzayında yayılır.YSA'nın enformasyon işleme performansı ağırlık vektörünün belirli bir değeri ile bulunacaktır.Aşağıda Şekil 2.5'de ağırlıkların düzeltiminin vektörel çizimi verilmiştir.



$X_k$  :Giriş İşaret Vektörü

$W_{k+1}$ :Bir sonraki Ağırlık Vektörü

$\Delta W_k$ :Ağırlık Vektörünün Değişmesi

$W_k$  :Mevcut Ağırlık Vektörü

Şekil 2.5 Üç boyutlu uzayda ağırlık vektörünün değişimi

Şekil 2.5'de görüldüğü gibi  $\Delta W_k, X_k$  ile aynı doğrultuda olduğunda istenen hata düzeltimini en küçük ağırlık değişimi ile elde etmek mümkündür.Böylece yeni bir giriş örüntüsü uygulandığında önceki eğitim örüntülerinin cevabı en az bozulmuş olur.Hata değişimini inceleyen iki çeşit kural vardır.

1-Hata düzeltme kuralları

2- Gradyant kuralları

Hata düzeltme kuralları her bir giriş örüntüsünde ağırlıkları yeniden ayarlayarak çıktı hatasını en aza indirmeye çalışırlar.Gradyant kurallarında ise ağırlık yeniden ayarlanarak ortalama karesel hatayı (MSE) en aza indirilmeye çalışılır.

Ağırlık vektörü ile çalışan YSA'da önemli noktalardan birisi bir öğrenme kuralı geliştirip enformasyon bölgesi kullanarak (Aktivasyon fonksiyonu ile) ağırlık vektörü "w"yi istenilen YSA performansı verecek noktaya yöneltmektir.Genellikle öğrenme kuralları için bir performans yada maliyet fonksiyonu tanımlanır.Optimizasyon ile "w" vektörü bulunur.

Günümüzde kullanılan bir çok öğrenme kuralı vardır.En çok kullanılan öğrenme kuralları şunlardır.

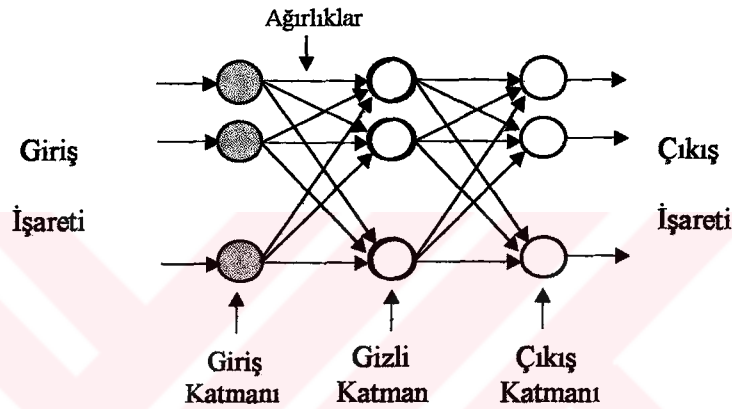
- Raslantısal (HEBB) öğrenme kuralı
- Performans (Widrow ve Adaline) öğrenme kuralı
- Kompetitif (Kohenen) öğrenme
- Filtreleme (Grossberg)
- Spatiotemporal öğrenme
- Genelleştirilmiş Delta kuralı öğrenme



Burada bütün öğrenme kuralları incelenmeyecektir.Sadece tezde kullanılan “Genelleştirilmiş delta kuralı” öğrenmesi ayrıntılı olarak incelenecektir.

## 2.5 Çok Katmanlı Algılayıcı (Multi Layer Perceptron:MLP)

Çok katmanlı algılayıcı giriş ve çıkış katmanları arasında birden fazla katmanın kullanıldığı YSA sistemleridir.Gizli katman (hidden layer) olarak isimlendirilen bu katmanlarda ,düğümleri aracısız giriş olmayan ve aracısız çıkış veremeyen üniteler vardır.Şekil 2.6 'da çok katmanlı algılayıcı genel yapısı verilmiştir.



Şekil 2.6 Çok katmanlı Algılayıcı (ÇKA) Yapısı

## 2.6 Geriye Yayılma Algoritması GYA (Back Propagation Algorithm) ve Genelleştirilmiş Delta Kuralı

Bu çalışmada kullanıldığı için üç katmanlı bir YSA'nın geriye yayılma (GYA) algoritması ile eğitilmesini gözönüne alalım.Ağdaki her bir düğüm tek bir işlem elemanıdır.Genelde daha öncede belirttiğimiz gibi düğümler katmanlarda düzenlenir.Veriler girişleri giriş düğümlerinden yapılır.Giriş düğümleri pasiftir ve hesaplanabilir değildir.Giriş ile çıkış katmanları arasında kalan gizli düğümlere ağırlıklı veri değerleri bu katmandan iletilir.

Her bir düğüm için ayrıca bir eşik (Threshold) değer girişi daha mevcuttur.Bu girişler düğümün kutuplaması yada yerel belleği olarak adlandırılır.Bütün gizli katman düğümleri tüm giriş verilerini alır.Fakat herbiri farklı bir ağırlık kümesine sahip olduğu için değerler kümeside farklıdır.Her bir gizli katman düğümü kendisine gelen girişleri işleyerek sonucu çıkış katmanı düğümlerine aktarırlar.Çıkışlarda farklı ağırlık kümesine sahiptir.Geri yayılma algoritması için gizli ve çıkış katmanı düğümleri kendisine giren girişleri iki aşamada işlerler.Her giriş kendisinin ağırlığı ile çarpılır,çarpımlar toplanır.Sonra bu toplam bir fonksiyon üzerinde (Aktivasyon fonksiyonu) çıkış elde edebilmek üzere,bir sonraki katmana geçirilir.Bu transfer fonksiyonuna aktivasyon fonksiyonu

denir. Aktivasyon fonksiyonları çeşitlidir. Bunlar şekil 2.4'de gösterilmiştir. Aktivasyon fonksiyonu bir nonlineerlik oluşturur. Bu nonlineerlik ağına yeteneğine daha fazla güvenilmesini sağlar.

GYA ile eğitime sırasında ağı her giriş örneğini çıkış düğümlerinde sonuç üretmek üzere gizli katmanlardaki hataları bulmak için amaçlanan hedef sonuçtan gerçek sonucu elde eder. Bundan sonra ağı çıkış hatalarının türevini çıkış katmanından geriye doğru saklı katmanlara geçirirler. Bunu yaparken de orijinal ağırlık bağıntılarını kullanırlar. GYA'ına ismini verende işte bu hataların geriye doğru yayılması özelliği olmuştur.

Her çıkış ve gizli düğümlerin hata değerleri bulunduğundan sonra düğümler kendi hatalarını azaltmak için kendi ağırlıklarını ayarlar. Ağırlık değiştirme denklemleri ağıdaki hataların karelerinin toplamını minimize edecek şekilde düzenler. Genelleştirilmiş delta kuralını eğitime için kullanılır. Şekil 2.6'da mimarisi gösterilen algoritma, ana hatları ile bundan sonraki bölümde açıklanmaya çalışılacaktır.

j. Nöron bir çıkış düğümü (nöronu) olmak üzere j. Çıkış düğümünün hata sinyali aşağıdaki gibidir.

$$E_j(n) = d_j(n) - y_j(n) \quad (2.1)$$

Burada n n. Eğitim örneğini temsil eder. j. düğümden oluşan karesel hatanın ani değerini  $\frac{1}{2} e_j^2(n)$  ile tanımlarız. Çıkış katmanındaki bütün düğümler üzerinden karesel hataların ani değerleri toplanırsa E(n) toplam karesel hata ani değeri bulunur.,

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (2.2)$$

şeklinde ifade edilebilir. Burada C çıkış katmanındaki tüm düğümleri içerir. N hedef uzayındaki örnek patern sayısı olsun. Ortalama karesel hata bütün örnekler üzerinden E(n) 'ler toplanarak bulunur.

$$E_{av} = \frac{1}{N} \sum_{n=1}^N E(n) \quad (2.3)$$

Karesel hatanın ani değeri E(n) ve ortalama karesel hata  $E_{av}$ , YSA'na ait tüm serbest parametrelerin (ağırlıklar ve yerel bellekler) fonksiyonudur. Verilen bir hedef uzayı için  $E_{av}$  hedef uzayının öğrenme performansının ölçüsüdür. Öğrenme prosesinin hedefi serbest parametreleri ayarlayarak  $E_{av}$ 'ı minimize etmektir.

Şekil 2.7'da j. Düğüme ait işaret akış diagramı verilmektedir. Şekil 2.7'dan da görüleceği gibi



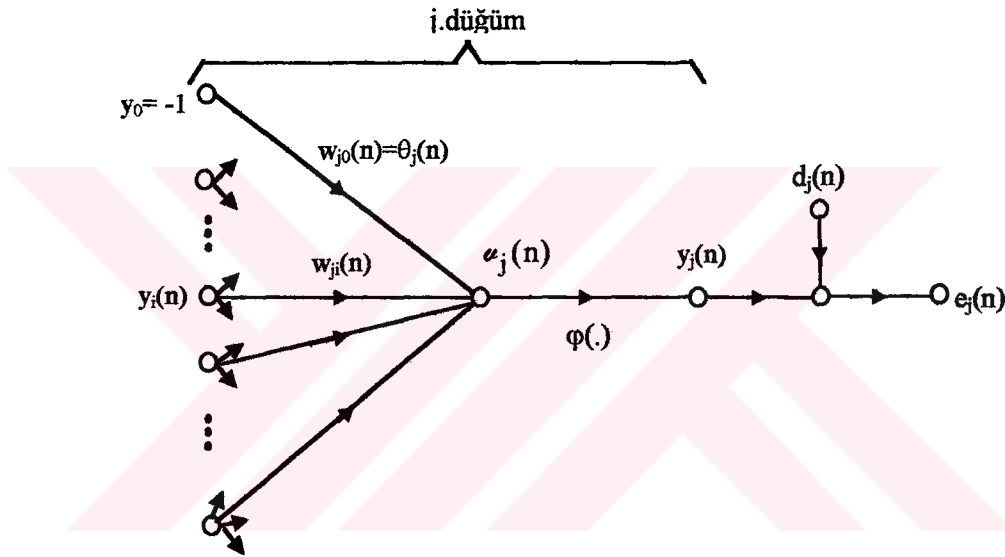


$$a_j(n) = \sum_{i=0}^p w_{ji}(n) y_i(n) \quad (2.4)$$

p eşik (yerel bellek) değeri hariç j. Dügüme giren giriş sayısıdır.  $w_{j0}$  ağırlığı ( $y_0 = -1$  girişine tekabül eden) j. düğüme uygulanan  $\theta_j$  eşik değerine eşittir. j. düğüm çıkışında oluşan sinyal  $y_j(n)$  n. iterasyon için

$$y_j(n) = \varphi_j(v_j(n)) \quad (2.5)$$

yazılabilir.

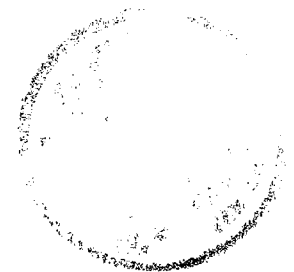


Şekil 2.7 j. Dügüme (nörona) ait detayları da gösteren işaret akış diagramı.

En küçük karesel hata (LMS) algoritmasına benzer şekilde ,GYA'sı  $w_{ji}(n)$  ağırlığını  $\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)}$  ani gradyantu ile orantılı olarak hesaplanan  $\Delta w_{ji}(n)$  kadar değıştirerek düzeltir. Bu kısmi türev terimi zincir kuralı geređi ařađıdaki gibi yazılabilir.

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial a_j(n)} \frac{\partial a_j(n)}{\partial w_{ji}(n)} \quad (2.6)$$

$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)}$  gradyantu ,  $w_{ji}$  ağırlığı için ağırlık uzayında araştırma yönünü belirler.



(2.2)'nin her iki tarafının  $e_j(n)$ 'e göre türevini alırsak

$$\frac{\partial \mathcal{E}(n)}{\partial e_j(n)} = \partial e_j(n) \quad (2.7)$$

elde ederiz benzer şekilde (2.1)'in her iki tarafını  $y_j(n)$ 'e göre türevini alırsak aşağıdaki sonucu buluruz.

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \quad (2.8)$$

Bundan sonra (2.5)'in  $u_j(n)$ 'e göre türevi alınır

$$\frac{\partial y_j(n)}{\partial u_j(n)} = \varphi'_j(u_j(n)) \quad (2.9)$$

bulunur. Son olarak (2.4)  $w_{ji}(n)$ 'e göre türevi alınır

$$\frac{\partial u_j(n)}{\partial w_{ji}(n)} = y_i(n) \quad (2.10)$$

Bulunur. Böylelikle, (2.7) den (2.10) 'a kadar elde edilen eşitlikler (2.6)'da yerine konulursa

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = -e_j(n) \varphi'_j(u_j(n)) u_j(n) y_i(n) \quad (2.11)$$

elde edilir.

$w_{ji}(n)$ 'e uygulanan  $\Delta w_{ji}(n)$  düzeltme terimi delta kuralı olarak tanımlanır.

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} \quad (2.12)$$

$\eta$  bir sabit olup öğrenme hızını belirler. GYA'nın öğrenme katsayısı olarak adlandırılır. Eksi işareti ağırlık uzayında adım adım azaltma suretiyle hedefe yaklaşmayı temsil eder. (2.11), (2.12)'de kullanılırsa

$$\Delta w_{ji}(n) = -\eta \delta_j(n) y_i(n) \quad (2.13)$$

elde edilir.Ki burdaki  $\delta_j(n)$  lokal gradyantı aşağıdaki gibi tanımlanır.

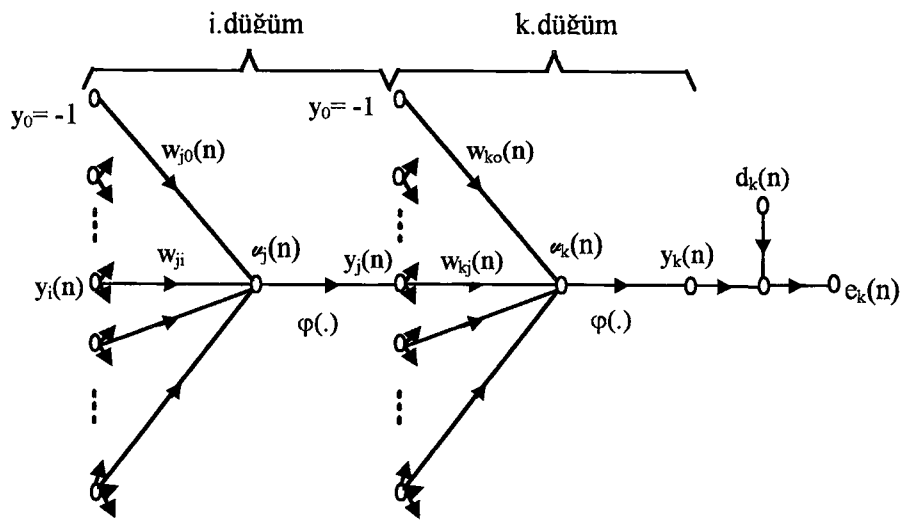
$$\delta_j(n) = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial u_j(n)} = e_j(n) \varphi'_j(u_j(n)) \quad (2.14)$$

Lokal gradyantlar ağırlıklardaki gerekli değişikliklere işaret eder.

(2.13) ve (2.14)'deki eşitliklerde anahtar faktör  $\Delta w_{ji}(n)$  ağırlık ayarlama miktarının hesaplanmasındaki j. Çıkış düğümüne ait hata sinyali  $e_j(n)$ 'dir.Eğer j. Düğüm bir çıkış düğümü ise hat sinyali (2.1) ile kolayca hesaplanır.Fakat j. düğüm bir gizli katman düğümü olduğunda ise,gizli katman düğümüne doğrudan ulaşılmasına rağmen,YSA devresinin çıkışında yapılan hatanın sorumluluğunu paylaşırlar.Buradaki soru,gizli katman düğümlerinin çıkışta oluşan hatanın sorumluluğunu ne ölçüde paylaştığı ve nasıl cezalandırılacağı veya ödüllendirileceği problemdir.Bu sorun hat sinyallerinin YSA devresi üzerinden geriye yayılması ile çözülür.

Eğer j. düğüm gizli katmanda bulunuyorsa ,bu düğüm için belirlenmiş arzulan değer bilgisi mevcut değildir.Bu gizli katman düğümüne ait hata sinyali,sözkonusu düğümüne doğrudan bağlı düğümlerin hata sinyallerinin terimi olarak hesaplanır.Şekil 2.8'de tasvir edilen durumu gözönüne alalım.(2.14)'e göre j. gizli katman düğümü için lokal gradyant  $\delta_j(n)$  yeniden belirlenmelidir.

$$\begin{aligned} \delta_j(n) &= \frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial u_j(n)} \\ &= \frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \varphi'_j(u_j(n)) \end{aligned} \quad \text{bir gizli katman düğümüdür} \quad (2.15)$$



Şekil 2.8 k çıkış düğümünün j gizli katman düğümüne bağlantısının ayrıntılı işaret akış diagramı.

$\frac{\partial \mathcal{E}(\mathbf{n})}{\partial y_j(\mathbf{n})}$  kısmi türevini hesaplamak için Şekil 2.8'den hareketle

$$E(\mathbf{n}) = \frac{1}{2} \sum_{k \in C} e_k^2(\mathbf{n}) \quad k \text{ bir çıkış katmanı düğümüdür} \quad (2.16)$$

(2.16) eşitliği  $j$  indeksi yerine  $k$  indeksi kullanılmasının dışında (2.2)'nin yeniden yazılmasından ibarettir. (Bu işlem  $j$ . Düğümün çıkış katmanı düğümü olma durumu ile karıştırılmaması için  $j$  yerine  $k$  indeksi kullanılmıştır.)

Herhangi bir durumda ,(2.16) eşitliği  $y_j(\mathbf{n})$  sinyaline göre türevi alınarak yazılacak olursa

$$\frac{\partial \mathcal{E}(\mathbf{n})}{\partial y_j(\mathbf{n})} = \sum_k e_k \frac{\partial e_k(\mathbf{n})}{\partial y_j(\mathbf{n})} \quad (2.17)$$

Daha sonra  $\frac{\partial e_k(\mathbf{n})}{\partial y_j(\mathbf{n})}$  için zincir kuralı kullanılarak (2.17) yeniden yazılacak olursa

$$\frac{\partial \mathcal{E}(\mathbf{n})}{\partial y_j(\mathbf{n})} = \sum_k e_k \frac{\partial e_k(\mathbf{n})}{\partial u_k(\mathbf{n})} \frac{\partial u_k(\mathbf{n})}{\partial y_j(\mathbf{n})} \quad (2.18)$$

Bununla birlikte ,Şekil 2.8'den hareketle şunu belirtmeliyiz ki

$$\begin{aligned} e_k(\mathbf{n}) &= d_k(\mathbf{n}) - y_k(\mathbf{n}) \\ &= d_k(\mathbf{n}) - \phi_k(v_k(\mathbf{n})) \quad k \text{ bir çıkış düğümüdür} \end{aligned} \quad (2.19)$$

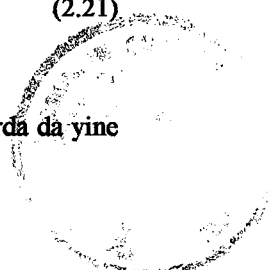
Bundan dolayı

$$\frac{\partial e_k(\mathbf{n})}{\partial u_k(\mathbf{n})} = -\phi_k'(u_k(\mathbf{n})) \quad (2.20)$$

Şekil 2.8'den hareketle şunu da belirtmeliyiz ki  $k$  çıkış düğümü için

$$u_k(\mathbf{n}) = \sum_{i=0}^q w_{kj}(\mathbf{n}) y_j(\mathbf{n}) \quad (2.21)$$

Burada  $q$   $k$ . çıkış düğümüne uygulanan (eşik değeri hariç) toplam giriş elemanı sayısıdır. Burada da yine



$w_{ko}(n), k$ . Düğümün  $\theta_k(n)$  eşik değeridir. Ki bu ağırlığa ait giriş değeri yine  $y_0 = -1$  'dir. Herhangi bir durum için (2.21)  $y_j(n)$  'e göre türev alınacak olursa

$$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n) \quad (2.22)$$

bulunur. Böylelikle (2.20) ve (2.22), (2.18)'de yerine konulursa kısmi türevi elde ederiz.

$$\begin{aligned} \frac{\partial \mathcal{E}(n)}{\partial y_j(n)} &= - \sum_k e_k \phi'_k(u_k(n)) w_{kj}(n) \\ &= - \sum_k \delta_k(n) w_{kj}(n) \end{aligned} \quad (2.23)$$

ikinci satırda (2.14)'de verilen lokal gradyant tanımını kullandık. Tabii ki  $j$  indeksi yerine  $k$  indeksi kullanıldı.

Sonuç olarak (2.23), (2.15)'de yerine konulursa  $j$ . gizli katman düğümü için gradyant  $\delta_j(n)$  aşağıdaki gibi verilir.

$$\delta_j(n) = \phi'_j(u_j(n)) \sum_k \delta_k(n) w_{kj}(n) \quad j \text{ gizli katman düğümü} \quad (2.24)$$

Şimdi GYA ile çıkardığımız ifadeleri özetleyecek olursak. İlk olarak  $i$ . düğümü  $j$ . düğüme bağlayan ağırlığa uygulanan düzeltme terimi  $\Delta w_{ji}(n)$  delta kuralı ile belirlidir.

$$\begin{pmatrix} \text{Ağırlık düzeltme} \\ \Delta w_{ji}(n) \end{pmatrix} = \begin{pmatrix} \text{Öğrenme Hızı} \\ \eta \end{pmatrix} \begin{pmatrix} \text{Lokal Gradyent} \\ \delta_j(n) \end{pmatrix} \begin{pmatrix} j. \text{ düğüme ait giriş sinyali} \\ y_j(n) \end{pmatrix} \quad (2.25)$$

İkinci olarak, lokal gradyant  $\delta_j(n)$   $j$ . düğümün gizli katman düğümü yada çıkış katmanı düğümü olmasına bağlıdır.

1.  $j$ . Düğüm çıkış katmanı düğümü ise  $\delta_j(n), \phi'_j(u_j(n))$  türev terimi ise hata sinyali  $e_j(n)$ 'in çarpımıdır. [(2.14)'e bak]

2. Eğer  $j$  bir gizli katman düğümü ise,  $\delta_j(n), \phi'_j(u_j(n))$  türev terimi ve  $j$ . düğüme bağlı olan bir sonraki (sağdaki, bakınız Şekil 2.8) gizli katman veya çıkış katmanı düğümü için hesaplanan  $\delta$ 'ların ağırlıklı toplamıdır. [(2.24)'e bak]

Öğrenme hızını artırırken kararsızlık riskinden sakınmak için (2.13)'le verilen delta kuralına momentum terimi Rumelhart (1986) tarafından aşağıdaki gibi eklenmiştir.

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n) \quad (2.26)$$

$\alpha$  momentum katsayısıdır. ( $0 < \alpha < 1$ ) (2.26) genelleştirilmiş delta kuralı olarak adlandırılır.  $\alpha=0$  olması durumunda (2.26), (2.13) ile verilen delta kuralına dönüşür.

Buraya kadar elde ettiğimiz sonuçların ışığında aktivasyon fonksiyonu olarak sigmoidal aktivasyon fonksiyonu kullanılması durumunda genelleştirilmiş delta kuralını üç katmanlı YSA'na uygulayalım. Öncelikle kullanılacak notasyonu özeteyelim.

$W^{(\ell)}$  :  $\ell$ . Katmandaki ağırlık vektörü ;  $\theta^{(\ell)}$  :  $\ell$ . katmandaki eşik vektörü ;

$y^{(\ell)}$  :  $\ell$ . katmandaki çıkış sinyal vektörü ;  $a^{(\ell)}$  :  $\ell$ . katmandaki net iç aktivite seviye vektörü

$\delta^{(\ell)}$  :  $\ell$ . katmandaki lokal gradyent vektörü ;  $e$  : çıkış katmanı hata vektörü ;

$\ell = 0$  'sa giriş katmanı ve  $\ell = L$  çıkış katmanını gösterir. (Bu şekilde  $L=3$ 'dür.)

$y_j(n) = \varphi_j(a_j(n))$  olduğu hatırlanırsa sigmoid tip aktivasyon fonksiyonu için

$$y_j(n) = \frac{1}{1 + e^{-a_j(n)}} \quad -\infty < v_j(n) < \infty \quad (2.27)$$

(2.27) her iki tarafı  $a_j(n)$ 'e göre türevi alınırsa

$$\begin{aligned} \frac{\partial y_j(n)}{\partial a_j(n)} &= \varphi_j'(a_j(n)) \\ &= \frac{e^{-a_j(n)}}{\left(1 + e^{-a_j(n)}\right)^2} = \frac{1}{1 + e^{-a_j(n)}} \frac{1 + e^{-a_j(n)} - 1}{1 + e^{-a_j(n)}} \\ &= \frac{1}{1 + e^{-a_j(n)}} \left(1 - \frac{1}{1 + e^{-a_j(n)}}\right) \end{aligned} \quad (2.28)$$

(2.28)'den aşikar olarak  $y_j' = \frac{\partial y_j(n)}{\partial a_j(n)} = \varphi_j'(a_j(n))$  aşağıdaki gibi yazılabilir.

$$\frac{\partial y_j(n)}{\partial a_j(n)} = y_j(n)(1 - y_j(n)) \quad (2.29)$$

Çıkış katmanında bulunan  $j$  düğümü için  $y_j(n) = o_j(n)$ 'dir. Bu durumda  $j$ . düğüm için lokal gradyent

$$\begin{aligned}\delta_j(n) &= e_j(n) \phi_j'(e_j(n)) \\ &= [d_j(n) - o_j(n)] o_j(n) [1 - o_j(n)] \quad \text{j çıkış düğümü}\end{aligned}\quad (2.30)$$

$o_j(n)$  j. düğümün çıkışındaki sinyaldir ve  $d_j(n)$  sözkonusu çıkış düğümü için arzu edilen çıkıştır. Diğer taraftan , herhangi j. gizli katman düğümü için , lokal gradyant yeniden yazılacak olursa

$$\begin{aligned}\delta_j(n) &= \phi_j'(e_j(n)) \sum_k \delta_k(n) w_{kj}(n) \\ \delta_j(n) &= y_j(n) [1 - y_j(n)] \sum_k \delta_k(n) w_{kj}(n) \quad \text{j gizli katman düğümü}\end{aligned}\quad (2.31)$$

(2.29)'dan  $\phi_j'(e_j(n))$  türev terimi maksimum değerini  $y_j(n) = 0.5$  ve minimum değerini  $y_j(n) = 0$  veya  $y_j(n) = 1$  için alır. Ağırlıkların değiştirilmesi miktarı  $\phi_j'(e_j(n))$ 'nin bir oranı olduğuna göre, sigmoid aktivasyon fonksiyonu kullanılması durumunda ağırlıklar, düğüm çıkışları alabileceği en büyük değer yarısına gelmesi durumunda , en fazla değiştirilir. Rumelhart (1986) ya göre GYA'nın bu özelliği bir öğrenme algoritması olarak kararlılığına (sonuca yakınsamada) katkıda bulunur.

Bu durumda ideal gradyantlar L çıkış katmanı j. çıkış düğümü için

$$\delta_j^{(L)}(n) = e_j^{(L)}(n) o_j(n) [1 - o_j(n)] \quad (2.32)$$

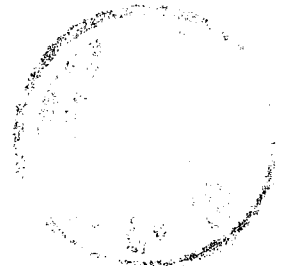
1. gizli katman j. gizli katman düğümü için

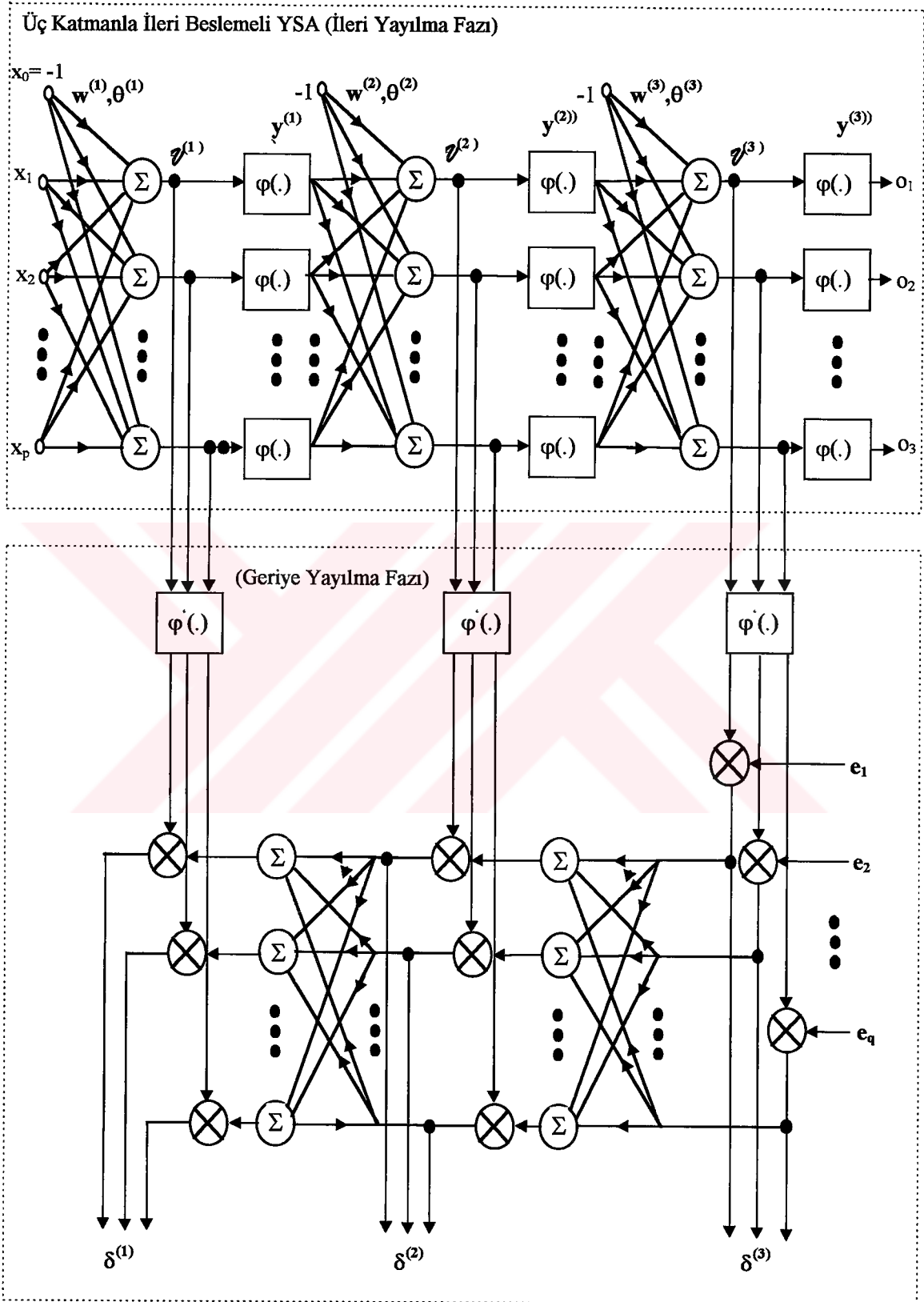
$$\delta_j^{(\ell)}(n) = y_j^{(\ell)}(n) [1 - y_j^{(\ell)}(n)] \sum_k \delta_j^{(\ell+1)}(n) w_{kj}^{(\ell+1)}(n) \quad (2.33)$$

Sonuç olarak ağırlıklar genelleştirilmiş delta kuralı ile aşağıdaki gibi ayarlanır.

$$w_{ji}^{(\ell)}(n+1) = w_{ji}^{(\ell)}(n) + \alpha \left[ w_{ji}^{(\ell)}(n) - w_{ji}^{(\ell)}(n-1) \right] + \eta \delta_j^{(\ell)}(n) y_i^{(\ell-1)}(n) \quad (2.34)$$

Ki burda  $\alpha$  momentum,  $\eta$  ise öğrenme katsayısıdır.





Şekil 2.9 Hatanın geriye yayılım algoritmasının üç katmanlı YSA'ya uygulanması.



### 2.6.1 Öğrenme ve Momentum Katsayıları

YSA ile ilgili bir başka sorunda düzgün bir öğrenme katsayısının ( $\eta$ ) ayarlanmasıdır. Ağırlıkları çok yüksek tutmak davranışın bozulmasına neden olabilir. O nedenle öğrenme katsayısını böyle bir davranışı önlemek için küçük tutmak gereklidir. Öğrenme katsayısı, genelde  $0 < \eta < 10$  aralığında seçilen sabit bir sayıdır. Öte yandan çok küçük bir öğrenme oranında, öğrenme işleminin yavaşlamasına yol açar.

Daha öncede bahsettiğimiz gibi momentum ( $\alpha$ ) fikri bu noktadan hareketle ortaya atılmıştır. Momentum mevcut delta ağırlığı üzerinden önceki delta ağırlığının belli bir kısmını besler. Böylece daha düşük öğrenme katsayısı ile daha hızlı öğrenme elde edilir. Momentum katsayısı genellikle  $0 < \alpha < 1$  aralığında değişen sabit bir sayıdır. EK-1'de optimum alfa katsayısını bulan program verilmiştir.

## 2.7 Yapay Sinir Ağlarını Elektronik Devre Olarak Gerçeklenmesi

### 2.7.1 Giriş

Yapay sinir ağları şekil sınıflandırma, modelleme, işaret işleme ve kontrol gibi mühendislik problemlerini şu iki yoldan biriyle çözer.

Geleneksel metodlarla mukayese edildiğinde, YSA'larının kullanımı gerçek-dünya uygulamalarında sistem performansında önemli derecede bir fark sağlar veya uygulamanın maliyetinde önemli bir miktarda azalmayı performansında bir bozulma olmadan sağlar.

YSA'ları başka çözüm yolu olmayan problemleri çözmeye imkanı verir.

Bu şekilde YSA yaklaşımı ile mühendislik problemlerine başarı ile uygulanabilecek çözüm verilmiştir. İkinci basamak olarak YSA'nın donanım (Hardware) olarak gerçekleştirilmesi gerekir ve çalışan düzeneğin içine bir parça donanım yerleştirilmelidir. Ancak çalışan bu sistemin çalışan bir modeli olduğunda onu tamamiyle anlayabildiğimizi söyleyebiliriz. Bu noktada düşünülmesi gereken anahtar soru bir YSA'nın donanım olarak uygulanabilmesi için bu noktada fiyat olarak en ucuz ve efektif ortam nedir? Tamamiyle sayısal (dijital) yaklaşım denince akla RISC işlemcisi gelir, RISC Reduced Instruction Set kelimesinin baş harflerinden türetilmiştir. (Cocke ve Marstein, 1990) Bu işlemci birkaç basit talimatı işletecek şekilde tasarlanmıştır. (tercihan her bilgisayar saykılında (adımında) bir işlevi (komutu) yürütecek) Gerçekten günümüzün modern RISC işlemcileri çok yüksek hızla sahip olmalarından dolayı bazı uygulamalarda YSA'larının emülasyonu için yeteri kadar hızlıdır. Mahmafih ses tanıma, optik karakter tanıma gibi RISC işlemcileri ile sağlanması mümkün olmayan performans seviyesi gerektiren bazı karmaşık uygulamalarda yetersiz kalır. Buna ilaveten proses kontrolü adaptif-ışın şekillendirme ve adaptif gürültü yok etme gibi istenilen öğrenme hızı standart işlemcilere göre çok hızlı olan uygulamalarda vardır. Bu tip uygulamalarda VLSI (Very Large Scale Integration) devreleri kullanılmak

zorundadır.Hızla gelişen bir teknolojiye sahip olması nedeniyle YSA'larının donanım olarak gerçekleştirilmelerinde ideal bir ortamdır.

VLSI teknolojisinde tek bir yarıiletken yongaya 10 milyonlarca transistör sığdırabilmesi kapasitesi vardır.Bu kapasite gelecekte fiziksel kanunların gerçekleştirdiği limite (Hoeneisen ve Mead(1972),Keyes(1987)) ulaşmadan önce bu miktar 100 kat daha artırılabilir.VLSI teknolojisini iki temel nedenden dolayı YSA'larına uygun buluyoruz.

1-Yüksek yoğunluğa (Birim yüzeye sığdırılan eleman sayısı) VLSI teknolojisinin müsaade etmesi nedeniyle tek bir yongaya çok sayıda birbirinin aynısı eş zamanlı çalışan nöronların uygulanmasına olanak tanır.Böylelikle YSA'larının doğal paralellliğini etkin olarak tamamiyle kullanmayı mümkün kılar.

2-YSA'larını düzenli topolojileri ve oldukça az sayıda iyi tanımlanan aritmetik işlemlerin eğitim algoritmalarına katılmış olması VLSI devrelerini tasarım ve üsten görünüş planı büyük ölçüde basitleştirir.

Bundan dolayı YSA'larının VLSI uygulamaları dünyada yaygın olarak büyük miktarda bir araştırma gayreti vardır.Bugün çok katmanlı algılayıcıların Boltzman makinesinin ortalama-alan teorisi makinelerinin ve kendi kendine organize olan YSA'larının gerçekleştirilmesi için genel amaçlı kırmıklara ulaşabilmektedir.Buna ilaveten özel bilgi işlem fonksiyonları için özel maksatlı çeşitli kırmıklar geliştirilmiştir.VLSI teknolojisinin başarılı kullanımı bilgi işlem bilimleri ile nörobilim arasında bir köprü kurulması şimdi sıralayacağımız yararlı etkilere sahip olacaktır.İnformasyon işlemeyi daha derinden anlamayı ve geleneksel bilgisayar teknikleri ile çözülmesi çok zor olan mühendislik problemlerinin çözümü için başlangıç metodu olmak.

### 2.7.2 Belli Başlı Tasarım Öğeleri

Kolay kullanım ve düşük maliyeti ve yüksek yoğunluklu olması nedeniyle CMOS transistörler YSA'larının VLSI uygulamalarında (Mead(1989)) kullanılır.YSA için geliştirmek için kullanılan kırmıkların genel amaçlı veya özel amaçlı olduğuna bakılmadan,bu teknolojiye gözönüne alınması gereken belli başlı tasarım öğeleri vardır.Özellikle aşağıda sıralayacağımız öğeleri tanımlamalıyız(Hammerstrom(1992))

1-Çarpımların toplamının hesaplanması.Bu bütün nöronlar için gerekli fonksiyonel bir gereksinimdir.Bu YSA'ya gelen giriş data vektörünün ağırlıklarla ayrı ayrı çarpıldıktan sonra toplanmasını ifade eder.Matematiksel olarak

$$y_j = \sum_i^p w_{ij} x_i \quad (2.28)$$

şeklinde ifade edilebilir. $w_{ij}$  ; i. nöronu j. nörona bağlayan ağırlık değeridir. $x_i$  giriş vektörü p ağırlıkların

YSA devrelerine uygulamak için iletişim biçimlerine önemli yenilikler getirmeliyiz. Bailey ve Hammerstrom (1988) tarafından hazırlanan bir makalede VLSI uygulamalarındaki özellikle bağlantı problemi üzerinde durulmuştur. Bu makalede çoğullamalı (Multiplexing) ana bağlantıların gerekliliği gösterilmiştir.

7-Uygulama maliyeti: Topyekün sistemin maliyeti YSA uygulama kırımının uygulama aşamasında göz önüne alınmalıdır. Hesaba katılması gereken faktörler aşağıdaki gibi sıralanabilir.

- Giriş/çıkış band genişliği gereksinimleri
- Güç tüketimi
- Uygulamaların kademesi ve esnek kullanım
- Analog ve sayısal teknolojinin kullanımı

### 2.7.3 VLSI Uygulamalarının Kategorileri

Analog uygulamalarda giriş işaretlerinin sürekli bir genliği vardır. Sayısal uygulamalarda ise, işaretler belirli bir aralıktaki ayrık genliklere kuantalanmıştır. Bir iki yaklaşımın karışımında (YSA'larının inşasında) kullanılan diğer yöntemin temelidir. (Hibrid Kombinezon)

#### Analog Teknikler:

Analog tekniklerin hassasiyet bakımından eksikliği olmasına rağmen klasik devre teorisine ve fizik kanunlarına dayandırılan verimli, etkin hesaplamalar bu eksikliği dengeler. Analog devreler geleneksel sayısal uygulamalarda zor veya fazla zaman tüketen belirli hesaplamaları kolaylıkla yapabilir. Buna ilaveten daha az güç harcar (Mead (1989))

Şekil 2.11'de Yük taşıyıcıları olarak elektron ve delikleri kullanan, n-kanal ve p-kanal MOS (Metal Oxide Silicon) transistörlerin devre sembolleri görülmektedir. Bu iki tip transistöre dayandırılan teknoloji tamamlayıcı MOS, CMOS (Complementary MOS) olarak anılır. MOS transistörünün işleyişi savak akımı  $I_d$ 'nin hesaba alınması ile anlaşılabilir. Ki buda geçit-kaynak geriliminin  $V_{gs}$  fonksiyonu olarak tanımlanmıştır. Savak geriliminin sabit tutulursa (Örneğin 2V) iki bölge aşağı belirtilen fonksiyonel bağımlılıkla tanımlanabilir.

-Eşik üstü bölgesi: Bu bölgede savak akımı  $I_d$ ,  $V_{gs}$  geçit-kaynak geriliminin ikinci dereceden polinomu şeklindedir.

-Eşik altı bölgesi: Bu bölgede transistör düşük geçit-kaynak geriliminde çalışır ve savak akımı  $I_d$  geçit-kaynak geriliminin eksponansiyel fonksiyonudur.

Bu bölgeler hemen hemen aynı işi görür, eksponansiyel nonlinearite tercihe daha çok sayandır. Çünkü birim akım başına daha çok geçiş iletkenliği sağlar. Buna ilave olarak, eşik altı bölgesinde, savak-kaynak gerilimine bağlı olarak aşağıda açıklanan iki kullanışlı fonksiyonu sağlar.

-Küçük savak-kaynak gerilimleri için (yaklaşık bir kaç yüz milivolttan düşük) transistör kaynak ve savak arasında mükemmel bir simetri ile kontrollü (ayarlı) iletkenlik gibi davranır. (Bir başka deyişle  $V_{gs}$  kontrollü direnç) Bu mod çalışma ohmik veya lineer bölge olarak anılır.

-Daha yüksek savak-kaynak gerilimlerinde, eleman gerilim kontrollü-akım kaynağı vazifesi görür.

Şu halde nörolojik fonksiyonlara benzeştirmek için MOS transistörler alt eşik bölgesinde çalıştırılmalıdır.

CMOS'un eşik altı bölgedeki çalıştırılması aşağıdaki sıralanan kullanışlı özellikleri gösterir.

-Genliğinin altı katından fazla eksponansiyel kazanç (10 pA→10µA)

-Çok etkin geriliminden akıma (eksponansiyel) çevirme veya akımından gerilime (logaritmik) çevirme (dönüştürme) bir tek transistör ile üretebilme

-Oldukça fazla düşük güç tüketimi (Tipik olarak  $10^{-12}$  den  $10^{-6}$  W'a kadar.)

Bunlar insan sinir ağına benzeştirilerek donanım olarak gerçekleşmesinin VLSI teknolojisi ile analog devreler kullanılarak gerçekleşmesinin ana tasarım düşüncesinin temelidir.

Böyle bir nörobilgisayarın analog hesaplamaları; zamanın ,uzayın, gerilimin, akımın ve yükün fonksiyonuna dayandırılarak gerçekleştirir. Fonksiyonlar devre elemanı seviyesi, devre seviyesi ve mimari seviye olarak aşağıdaki gibi tanımlanır.

-Devre elemanı seviyesi:  $I_d$  MOS transistörünün savak akımı gösterebilir.  $V_g$ ,  $V_s$  ve  $V_d$  sırasıyla geçit, kaynak ve savak gerilimleri olsun. Transistörün n-kanallı veya p-kanallı olmasına göre eşik altı bölgedeki davranışı

$$I_d = I_0 \exp\left(\frac{KV_g}{U_t}\right) \left[ \exp\left(-\frac{V_s}{U_t}\right) - \exp\left(-\frac{V_d}{U_t}\right) \right] \quad \text{n-kanal} \quad (2.32)$$

$$I_d = I_0 \exp\left(\frac{KV_g}{U_t}\right) \left[ \exp\left(\frac{V_s}{U_t}\right) - \exp\left(\frac{V_d}{U_t}\right) \right] \quad \text{p-kanal} \quad (2.33)$$

Burada  $I_0$  sıfır kutuplama akımı ve K yüzey etkisi katsayısıdır. Buda kullanılan transistör tipine bağlıdır.  $U_t$  termal gerilimdir. Aşağıdaki gibi tanımlanır.

$$U_t = \frac{k_B \cdot T}{q} \quad (2.34)$$

$k_B$  Boltzman sabiti,  $T$  Kelvin olarak ölçülen mutlak sıcaklık, ve  $q$  elektron yüküdür. Not etmek gerekir ki (2.32) ve (2.33) 'deki eksponansiyel fonksiyonlar Boltzman kanununa göre dir. Ve köşeli parantez içindeki eksponansiyel fonksiyonlar arası çıkarma işlemi Ohm kanununun gereğidir. Bu iki eşitliğin tanımladığı kombinezon CMOS teknolojisinde eşik altı bölgede analog hesaplamaların çeşitlerinin yerine getirilmesini, gerçekleşmesini, icra edilmesini tanımlar.

-Devre düzeyi: İkinci düzey yükün korunumu ve enerjinin korunumu ile yönetilir. Bunlarda

sırasıyla ,bilinen iki eşitliği verir.

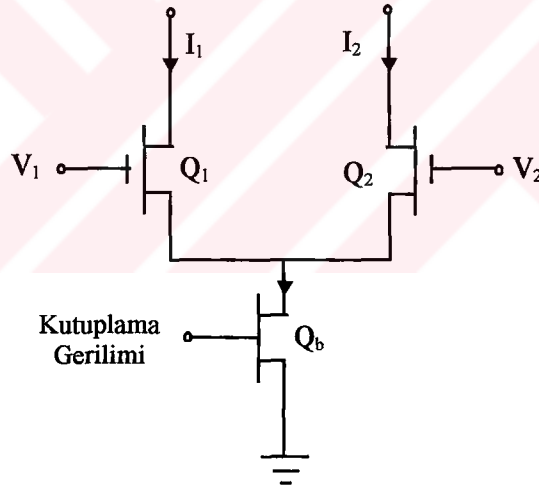
$$\sum_i I_i = 0 \quad (2.35)$$

$$\sum_i V_i = 0 \quad (2.36)$$

(2.35) Kirchoff akımlar kanunu (2.36) Kirchoff gerilimler kanunu olarak bilinir.

-Mimari düzeyi:Bu son düzeyde,matematiksel fizikten diferansiyel eşitlikler kullanışlı fonksiyonları uygulamak için kullanılır.Buda ilgilenilen uygulamaya bağlıdır.

Andreou (1992) tarafından tanımlanan analog yaklaşımında,en küçük yer kaplayan tasarım sitili uyarlanmıştır.Bu yaklaşımda,tek bir transistörün güçlü bir hesaplama elemanı olduğu inancı üzerine motive edilmiştir.Tasarım metodolojisi,akım-modu eşik altı CMOS devreleri üzerine dayandırılmıştır.İlgilenilen sinyaller akımlarla temsil edilir ve gerilimler daha az önemli rol üstlenirler.Akım-modu yaklaşımı mümkün olan en büyük band genişliğinde işaret işleme sağlar.



Şekil 2.10 Geçiş iletkenliği kuvvetlendiricisi olarak kullanılan farksal kuvvetlendirici çiftinin devre şeması

Bunun tersine,Mead (1989) tarafından tanımlanan bir analog yaklaşımda temel inşa bloğu olarak geçiş iletkenliği kuvvetlendiricisi kullanılmıştır.Bu kuvvetlendirici,Şekil 2.10'daki şeklin altındaki transistör  $Q_b$  akım kaynağı gibi çalışır.Sabit  $I_b$  akımına kaynaklık eder. $I_b$  akımı şeklin üstündeki  $Q_1$  ve  $Q_2$  transistörleri arasında bölüşülür; $V_1$ - $V_2$ ,farksal gerilimin yoluyla bu bölüşüm tayin edilir.Bu iki transistörün savak-kaynak geriliminin her ikisini de doyuma getirecek kadar büyük olduğu kabul edilirse,(2.32)'nin farksal geçiş iletkenliği kuvvetlendiricisine bir uygulamasını buluruz.Bu da

aşağıdaki sonucu bize verir.

$$I_{out} = I_b \tanh\left(\frac{KV_{in}}{2U_t}\right) \quad (2.37)$$

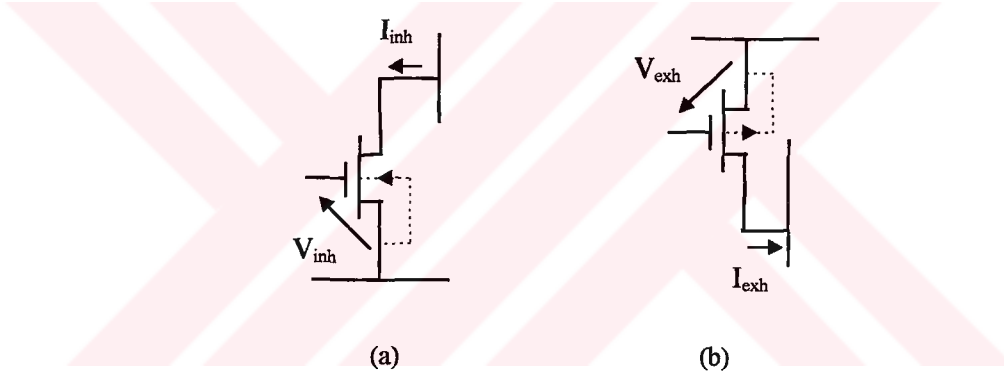
Burada

$$V_{in} = V_1 - V_2 \quad (2.38)$$

ve

$$I_{out} = I_1 - I_2 \quad (2.39)$$

dır. Böylece farksal gerilim  $V_{in}$  giriş işareti ve farksal akım  $I_{out}$  çıkış işareti gibi davranır. Görüyoruz ki Şekil 2.10'daki devre sigmoidal nonlinearite formunda "çıkış hesaplayan" eleman gibi davranır.



Şekil 2.11 (a)Yasaklayıcı girişli (b) Uyarıcı girişli synapsis ağırlıkları modellemek için sırasıyla n-kanallı ve p-kanallı transistörleri göstermektedir.

"Toplamaların çarpımını" hesaplama analog paradigmaya çok uygundur. CMOS teknolojisinde, synapsis (ağırlığın)'ın doğal seçeneği doyuma sürülmüş transistördür (Bohen (1989)). Özel olarak, transistörün geçitine uygulanan giriş gerilimi çıkışta düşük iletkenlik üretir. Bu ayarlama çok geniş fan-out'a izin verir. Şekil 2.11 sırasıyla yasaklayıcı girişli ve uyarıcı girişli synapsis ağırlıkları modellemek için sırasıyla n-kanallı ve p-kanallı transistörleri göstermektedir. Alışıldığı gibi, uyarma bir noktaya pozitif yükü beslemektir, ki pozitif yüklü savak yasaklamadır. Şekil 2.11'e de giriş-çıkış bağlantısı aşağıdaki gibi tanımlanır.

$$I_{inh} = I_0 \exp\left(\frac{KV_{inh}}{2U_t}\right)$$

ve Şekil 2.11b ise

$$I_{\text{exh}} = I_0 \exp\left(\frac{-KV_{\text{exh}}}{2U_t}\right)$$

olarak tanımlanır. Her iki durumda da sinaptik ağırlık bir geçiş iletkenliği ile modellenir. Son adım olarak “çarpımların toplamı” hesaplaması, Kirchoff’un akım yasası; buda bir sinir hücresinin çeşitli ağırlıklarına tekabül eden çıkış savak akımlarının toplamını gerektirir.

Gözönüne alınması gereken bir diğer önemli konu, saklama (kaydetme) meselesidir. Bu gereksinim, örneğin, geçitler arası gerilim farkını kullanarak, sinaptik ağırlığın kaydedilmesi (saklanması) halledilebilir.

Bu kısa incelemeden sonra şu önemli noktayı not etmeliyiz ki analog devreler, geleneksel CMOS teknolojisi veya eşik altı CMOS teknolojisine dayandırılan, yapay sinir ağlarının uygulanmasında hesaplama açısından çok etkin bir tekniktir.

#### Sayısal Teknikler:

Analog yaklaşıma göre sayısal (dijital) yaklaşımının iki anahtar avantajı vardır. (Hammerstrom(1992))

-Kolay Tasarım ve Üretim: VLSI teknolojisinin kullanımı yüksek hassasiyet, ağırlıkların kolay saklanması ve fiyat-performans avantajı sağlar. Analog VLSI’ye göre “programlanabilirlik” avantajı vardır. Bunlara ilaveten analog’a göre sayısal silikon işleme daha da kolayca elde edilebilir.

-Esneklik: Sayısal yaklaşımın ikinci ve en önemli üstünlüğü çok daha fazla esnek olmasıdır. Buda birçok daha kompleks algoritmanın kullanılmasına müsaade etmek suretiyle mümkün uygulama alanlarını çoğaltır. Bazı durumlarda, kompleks problemlerin çözümü yapay sinir ağı mimarisinde önemli derecede esneklik, ayarlanabilirlik gerektirir. Esneklik eksikliği analog sistemlerin gerçekten temel bir sınırlamasıdır. Özellikle karmaşıklık derecesi (teknolojinin üstesinden gelebileceği) analog teknoloji ile çözülebilecek problemlerin kademesini ve faaliyet alanını sık sık sınırlar.

Bununla birlikte, VLSI teknolojisinin çarpma işleminin sayısal uygulamalarında fazla güç tüketimine ve alana ihtiyaç gösterir. Alan ihtiyacı çoğullamalı iç bağlantı (Hammerstrom(1992)) kullanılarak giderilebilir.

Sayısal ve analog teknolojiden hangisinin kullanılacağına nihai kararı, yapay nöron ağı uygulaması için hangi özel algoritmanın kullanılacağı bilinmeden, verilemez. Eğer, genel amaçlı bir hedefimiz varsa, sayısal VLSI teknolojisine göre belirgin bir üstünlüğü vardır.

#### Hibrid Teknikler:

Analog hesaplamaların kullanımı YSA amaçlı VLSI’lar için, potansiyel hızları, az yer kaplamaları ve kuantalama etkilerinden uzak olmaları nedeniyle caziptirler. Diğer tarafından sayısal tekniklerin kullanımı uzun mesafeli haberleşmede sayısal işaretler kolayca iletilebilmesi ve yeniden üretilebilmesi nedeniyle, tercih edilir. Bu düşünceler YSA uygulamalarının VLSI tekniği ile uygulanmasında hibrid yaklaşımının (hem sayısal hem analog yaklaşımı kullanan) kullanımını

cesaretlendirmiştir.Ki bu yaklaşım her iki yaklaşımın yararlı tarafları üzerinde bina edilmiştir.(Murray(1991)) Haberleşme sistemlerde teori ve pratiği çok iyi bilinen darbe modülasyonu konusu hibrid yaklaşımda önemli bir rol oynar.(Haykın (1983),Black(1953)) Darbe modülasyonunda,YSA'nın bir konusu olarak bakıldığında,nöron hücresinin durumuna göre (çıkış değeri) taşıyıcı olarak kullanılan darbe hüzmesinin bazı karakteristikleri değiştirilir.Değiştirmekle kastedilen darbe genliği,darbe süresi ve darbe tekrarlama hızlarıdır.Burada tanımlandığı gibi üç temel darbe modülasyon tekniği birbirinden ayrılmalıdır.

-Darbe Genlik Modülasyonu:Burda darbenin genliği zamanla modüle edilir.(Nöron hücresinin çıkışının  $0 < y_j < 1$  arasında değer almasına göre) Özellikle YSA'larında bu teknik,bilgi analog gerilim seviyeleri olarak iletildiğinden ,yeterli değildir.

-Darbe Genişlik Modülasyonu:Bu durumda darbenin genişliği nöronun durumuna ( $y_j$ ) göre darbe genişliği değiştirilir.Bu durumda hibrid yaklaşımın üstünlüğü görünür hale gelir.Zira modüle edilen işaret içinde analog gerilim olmadığından bilgi zaman ekseninde boyunca kodlanır.Bu nedenle darbe-genişliği ile modüle edilen işaret çok daha sağlıklı ve güvenilirdir.Buna ilaveten demodülasyon integrasyon ile kolayca sağlanır.Bununla birlikte sabit işaretleme sinyali kullanıldığından modüle edilmiş sinyalin inen ve çıkan kenarı aynı anda iki farklı nöronun çıkış durumunu temsil edebilir.Bu senkronizasyon probleminin varlığı çok geniş paralel YSA VLSI devreleri için sakıncadır,bütün nöronlar ve ağırlıklar besleme hatlarına aynı anda sürülme eğiliminde olduğundan buda bu nedenle besleme hatlarının sayısını aşırı bir şekilde artmasını gerektirir.

Darbe frekans Modülasyonu:Darbe hüzmesinin frekansı nöronun durumuna ( $y_j$ ) göre değiştirilir.Frekans belli bir minimum değerden başlayıp yine belli bir maksimum değere kadar nöronun çıkışında aldığı değere göre değiştirilir.Bu durumda darbenin genliği ve darbe süresi (genişliği) sabit kalır.Bu durumda hibrid yaklaşımın kullanımı darbe-genişlik modülasyonunda bahsedilen nedenlerle avantajlıdır.Zira artık işaretleme frekansı değişkendir.Sonuç olarak besleme hatlarına olan talepten sakınılmış olunur ve darbe-frekans modülasyonunun kullanımının bir sonucu olarak güç tüketiminin hesabında zaman ortalaması alınır.

Bu incelemeden sonra,görülmektedir ki oldukça geniş paralel YSA VLSI devrelerinin işaretlenmesinde darbe frekans modülasyonu pratik bir teknik olarak ön plana çıkmaktadır.İlginç olarak not etmek gerekir ki bir yüzyıldan beri beyindeki nöronlar birbirleri arasında darbe-frekans modülasyonu kullandığı bilinmektedir.(Hecht,Nielsen(1990)).Böylece darbe-frekans modülasyonun yararları ,biyolojik nöronlardan da esinlenerek tanınmış olundu (Churcher(1993)) ,(Murray(1991)). Özellikle,devreler bilgi taşınmasında ve analog devrenin kontrolünde,analog bilginin zaman ekseninde saklanması sayısal işaretleri kullanır.Böylece burada tanımlanan VLSI YSA devreleri hibrid düzeneklerdir,sayısal domenden analog domene uygun bir şekilde geçilmek suretiyle ,sağlıklı çalışmayı,minimum alana maksimum elemanın sığdırılmasını ve hızı optimize eder.

YSA'larının VLSI tekniğinde kullanılan bir diğer hibrid tekniğide çarpıcı olarak kullanılan MDAC (Multiplying digital-to-analog converters) dır.Bu teknikte,analog durum (giriş sinyali gibi)



Belcore kırımığında olduđu gibi (Alspector(1991)) sayısal ağırlıkla veya sayısal durum AT&t ANNA kırımığında(çipinde) olduđu gibi (Sackinger(1992)) analog ağırlıkla çarpılabilir.Bu nedenle MDAC'lar YSA tasarımcılarına belirli bir hesaplama problemi,analog ve sayısal teknolojileri optimal bir şekilde kombine ederek ,çözmeye izin verir.

#### 2.7.4 YSA Donanımlarına Kısa Bir Bakış

YSA'nın VLSI uygulamalarına analog,sayısal ve hibrid yaklaşımlar açısından baktıktan ve avantaj ve dezavantajları belirledikten sonra artık YSA donanımlarına göz atabiliriz.Genel ve özel amaçlı nörobilgisayarlar kırımık ve sistemlerinin halihazırda elimizin altında olan listesi oldukça çeşitlilik arz etmekte ve halen genişlemektedir.Buda YSA'larının kullanıcı topluluđu tarafından ne kadar hızlı bir kabul gördüğünün göstergesidir.Genel amaçlı kırımık ve sistemleri ETANN analog kırımık (Holler(1989)),Edinburgh Üniversitesi EPSILON hibrid kırımığı (Murrey(1991),Hamilton(1992), Churcher(1993)),Adaptif çözümler CNAPS sayısal sistemi (Hammerstrom(1992),(1990)) , Siemens Sayısal sinyal prosesörü (Ramacher(1991),(1990)),Mitsubishi BNU sayısal kırımığı (Arima(1991)),Hitachi sayısal kırımığı (Watanabe(1993)),Bellcore Boltzmann ortalama-alan öğrenme kırımığı (Alspector(1991),(1992)), ve AT&T Bell Labaratuarı ANNA kırımığı (Boser(1992),Sackinger(1992)) olarak sayılabilir.Özel amaçlı kırımıklar Sinaptik OCR kırımığı,Kohonen'in kendi kendine organize olan haritası yaklaşımının analog uygulaması olan ve kırımık üzeri öğrenme imkanı da veren kırımık (Macq(1993)) ,Video makineleri dedeksiyonu için VLSI prosesörleri(Lee(1993)),Haberleşme alıcıları için programlanabilir analog VLSI YSA (Choi(1993)) ve Analog sayısal dönüştürücüler için çok seviyeli,YSA kırımığı (Yuh ve Newcomb(1993)) şeklinde sıralanabilir.

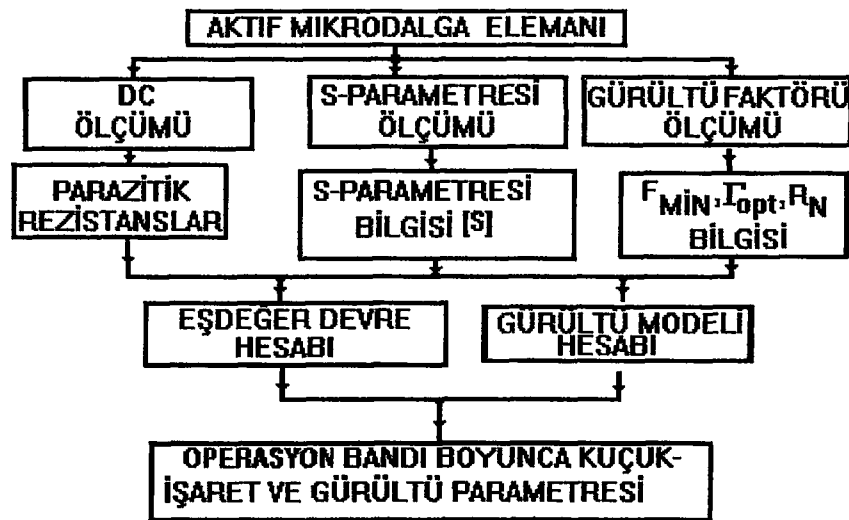
### 3 AKTİF MİKRODALGA ELEMANLARININ YSA İLE MODELLENMESİ

#### 3.1 Giriş

**Amaç:** Aktif mikrodalga elemanın küçük-ışaret ve gürültü davranışı çoklu kutuplama/konfigürasyon tipi için YSA yaklaşımıyla modellenmiştir. Burada ışıaret ve gürültü parametreleri YSA eşdeğeri tarafından hesaplanan eleman bir kara kutuyla modellenmiştir ve bu modelleme çoklu kutuplama ve farklı konfigürasyon tipleri için, bu iki parametre setinin ölçüm uzayına uydurulmasına dayandırılmıştır. Burada anlatılacak olan Eşdeğer YSA modeli elemanın optimizasyon sırasında eleman fiziki denklemlerinin çözülünü gerektirmez. Diğer modelleme teknikleri ile karşılaştırıldığında YSA yaklaşımının nonlinear elemanlar için çok boyutlu bir model olarak kullanılabilme kapasitesine sahip olduğu görülecektir.

Bir mikrodalga transistörünün kutuplama noktası civarındaki küçük ışıaret ve gürültüye karşı olan davranışları çalışma bandı boyunca sırasıyla saçılma parametreleri  $S_{11}, S_{22}, S_{21}, S_{12}$  ve gürültü parametreleri  $F_{opt}, \Gamma_{opt}, R_N$  ile belirlenebilir. S parametreleri ve gürültü parametrelerinin her ikisi de frekansa, kutuplama noktasına, konfigürasyon tipine ve elemanın fiziksel özelliklerine bağlıdır. S-parametreleri, elemanın sinyal güç kazançları ve giriş çıkış kapı uyumsuzluk kayıplarını tayin ederken, gürültü parametrelerinde elemanın giriş çıkışı arasında sinyal/gürültü oranı kötüleşmesini tayin eder.

Şekil 3.1'de gösterildiği gibi gürültü parametreleri de küçük ışıaret model parametreleri de küçük ışıaret modelleme bilgisine gereksinimi vardır. Çünkü her iki parametre seti de elemanın fiziksel ve geometrik parametrelerine, örneğin aktif katmanın geçit uzunluğuna sıkı sıkıya bağlıdır. Bu açıdan bakılacak olursa, eşdeğer devrenin elemanları hem küçük-ışıaret hem de gürültü parametrelerinden yararlanmak suretiyle tayin edilmelidir.



Şekil 3.1 Aktif Mikrodalga Elemanlarının Küçük-İşaret ve Gürültü Performans Parametrelerinin Hesaplanması için Genel Prosedür.

Bu çalışmanın hedefleri aşağıdaki gibi sıralanabilir.

(i) Eşdeğer işlevi görecek, tek gizli katmanlı, ileri beslemeli tipteki yapay sinir ağı devresini kurmak,

(ii) Geriye yayılım algoritması (BPA) ve lineer olmayan tipteki aktivasyon fonksiyonları kullanarak, yapay sinir ağı devresini herhangi bir tipteki aktif eleman için (Transistor gibi) çalışma bandı boyunca, çeşitli kutuplama noktalarında ve konfigürasyon tipleri için işaret-gürültü davranışının her ikisini de kullanarak, yapay sinir ağı devresini eğitmek,

(iii) Yapay sinir ağı devresinin performans ölçüsünü belirlemek,

(iv) Seçilen kutuplama noktası civarında, her hangi bir konfigürasyon tipi elemanın lineer olmayan karakteristiğine fonksiyon yaklaşımı yapan eğitilmiş yapay sinir ağı devresini kullanarak her hangi bir frekans için elemanın işaret-gürültü davranışını belirlemek.

Klasik modelleme bakış açısından küçük-ışaret-gürültü eşdeğer devreleri öz(intrinsic) devre ve dış(extrinsic) devre olmak üzere iki kısımda incelenir(Şekil 1.1). Bir MESFET'e ait eşdeğer devre düşünüldüğünde en önemli dış devre parametreleri ,yariletken yongayı dışarıyla bağlantısını sağlayan bağlantı telleri (genellikle altın kullanılır) nedeniyle oluşan, parazitik endüktans ve dirençlerdir. Dört ana öz(iç,yongaya ait)parametrede giriş kapasitesi  $C_{gs}$ , geçişiletkenliği  $g_m$ , çıkış iletkenliği  $g_{ds}$  ve geri besleme kapasitesi  $C_{gd}$ 'dir. Buna ilaveten, öz devrenin elektriksel davranışını tayin etmek için  $R_i$  öz direnci ve geçitten taşıyıcıların geçmesi için ihtiyaç duyulan zamana tekabül eden  $\tau$  geçiş iletkenliği gecikmesi sabiti olmak üzere iki parametreye daha ihtiyaç duyulur. Bu şekilde ayrıntılı modellemekten amaç ölçülmüş değerlerle hesaplanan değerler arasında mümkün olan en iyi uyumu yakalayabilmektir. Bu fiziksel yaklaşımın sık sık karşılaşılan iki önemli sakıncası vardır. Birincisi eşdeğer devre parametrelerinin seçilen elemanın elektriksel davranışının yegane çözümü olmaması, ikincisi ideal olmayan etkilerin yeterince doğru modellenememesinden kaynaklanır. İkinci sakınca küçük ölçme hatalarına göz yumulması durumunda bile mükemmel olmayan bir uyuma neden olabilir.

Bilgisayar simülasyonu Tümlleşik Devre tasarımında en önemli basamaklardan biridir. Tümlleşik devrelerin analiz ve tasarımının doğruluğu, devre elemanlarının bilgisayar simülasyonunun doğruluğu kadar olduğuna göre, yeterli doğrulukta modelleme Tümlleşik Devre Tasarımında gerekli bir faktördür. Alçak frekanslarda göreceli basit olan pasif elemanları karakterize edilmesi bile mikrodalga frekans bölgesinde zor olabilir. Çok sayıda parametre seti ve aralarındaki karmaşık ilişkileriyle yüksek dereceli nonlinear modellerle karakterize edilebilen yariletken elemanlar halinde ise ,parametrelerin uygun seçimi çok önemli bir iştir. Gerçekte, yetersiz bir uygulama ,simülasyon sonuçlarını önemli ölçüde bozabilir. Eleman nonlinearlikleri nedeniyle ,genellikle bu model parametreleri doğrudan ölçümlerle tayin edilemez.

Literatürdeki eleman modelleme, penaltı olarak nitelendirilebilecek iki temele dayandırılabilir.

(i) Genellikle küçük-ışaret performansı çalışmaları, gürültü performansı çalışmalarından ayrılmıştır. Bu konudaki yayınlar ya sadece küçük-ışaret modeli, yada ya da mevcut küçük-ışaret eşdeğer devrelerine ayandırılmış ve eleman gürültü özellikleriyle ilişkilendirilmemiş gürültü davranışı tasvirleridir



Sadece Hu v.d. (1993) ve Roux v.d.(1995)'de,bu iki davranışlar klasik bir eşdeğer devrede birleştirilmiştir.(ii)Bir tümleşik mikrodalga elemanı ve paketlenmesini karakterize etmenin standart yaklaşımı herbir bileşeni,elektriksel ölçmelerle uyum içinde olacak şekilde bir eşdeğer devreyle modellemektir.Böyle bir yaklaşımın ,paketlemenin(kılıflama) ,elemanın paketlenildiği durumdaki toplam elektriksel özelliklerine katkısını doğru olarak modellemediği,Bridge v.d.(1992) tarafından gösterilmiştir.

Ayrıca ,klasik eşdeğer devre eleman değerlerini bulmakta kullanılan optimizasyon temelli çıkarım teknikleri,yegane çözüm takımı yerine bir çok alternatif çözüm takımları üretirler.Kısımlara ayırarak ölçme yöntemi Curtice ve Camisa (1984) ve otomatik dekompozisyon tekniğiyle Kondoh (1986) bir kısım gelişmelere rağmen ,başlangıç değerinden kaynaklanan belirsizlik hala mevcuttur.

Yakın zamanda yaptığımız çalışmada Güneş v.d (1996), yayınlanan modellere göre ölçme sonuçları ile daha iyi bir uyum sergileyen,mikrodalga transistörünün hem işaret hem de gürültü özellikleri bir YSA modelinde birleştirilmiştir.Fizik-temelli FET modeller gibi hassas modeller kullanma trendi optimizasyon sonuçları ile ilişkilendirilmesi talep edildiğinde göz korkutucu işlemlerle karşı karşıya gelinmektedir.Bu tip modellerin standart optimizasyon ve istatistiksel yaklaşımlarda kullanımı hesaplama açısından çok yoğun bir işlemdir.Bundan dolayı devre simülatörünün her adımında fizik-temelli denklemlerin çözülmesi prosedürü hesaba katılmalıdır.Bu tip var olan optimizasyon metodları off-line tipi hesaplamalardan esinlenerek geliştirilmiştir.Tasarımcıların tarifnamedeki ,hatta devre topolojilerinde modifikasyonlar yapıldıktan sonra devreyi yeniden optimize etme gereksinimi duymaları durumu gibi interaktif tasarıma uygun değildir. Bu problemlerin üstesinden gelebilmek için,iki tip yaklaşımın çok boyutlu polinomlar yaklaşımı Barby v.d. (1988),Biernacki v.d. (1989) ve look-up tablosu yaklaşımının Meijer (1990) tarafından uygulandığı çalışmaların yerine bizim önerdiğimiz YSA modeli yaklaşımı eleman modellemeye Güneş v.d.(1996),devre optimizasyonuna ve istatistiksel tasarımda henüz uygulanmaya başlanmamıştır (Zaab v.d. (1995)).

Bu tezdeki çalışma bir transistörün YSA modeli kurulmasına yoğunlaşmış;ayrıca tektaş tümleşik mikrodalga devre tasarımında kullanımına örnek verilmiştir.(Bak bölüm 4.9)

### 3.2 Aktif Mikrodalga Elemanının Küçük-İşaret ve Gürültü Davranışının Belirlenmesi

Belli bir konfigürasyon tipinde bir kutuplama noktası civarındaki mikrodalga transistörünün küçük-ışaret ve gürültü performansı  $\omega$ -domeninde saçılma S ve gürültü N parametre vektörleri sıklıkla verilir.Ölçülmüş performans parametre datası tablo-formunda aşağıdaki gibi verilebilir:

$$\begin{bmatrix} f_1 & : & \mathbf{S}^{(1)} & \mathbf{N}^{(1)} \\ f_2 & : & \mathbf{S}^{(2)} & \mathbf{N}^{(2)} \\ \vdots & & \vdots & \vdots \\ f_N & : & \mathbf{S}^{(N)} & \mathbf{N}^{(N)} \end{bmatrix} \quad (3.1)$$

Burada  $S^{(1)}, N^{(1)}, \dots, S^{(N)}, N^{(N)}$  sırasıyla, saçılma ve gürültü vektörleridir ve  $f_1, \dots, f_N$  örnek çalışma frekansları ve  $S^{(N)}$  and  $N^{(N)}$  performans vektörleri aşağıdaki gibi verilebilir.

$$\left[ S^{(N)} \right]^{-1} = \begin{bmatrix} |S_{11}|^{(N)} & \varphi_{11}^{(N)} & |S_{12}|^{(N)} & \varphi_{12}^{(N)} & |S_{21}|^{(N)} & \varphi_{21}^{(N)} & |S_{22}|^{(N)} & \varphi_{22}^{(N)} \end{bmatrix} \quad (3.2)$$

$$\left[ N^{(N)} \right]^{-1} = \begin{bmatrix} F_{\min}^{(N)} & |\Gamma_{\text{opt}}|^{(N)} & \varphi_{\text{opt}}^{(N)} & R_N^{(N)} \end{bmatrix}$$

(3.1) ve (3.2) 'de verilen ifadeler YSA modelini eğitmek için kullanılmıştır. Sonra, performans-parametre vektörleri  $S^{(k)}, N^{(k)}$  arzu edilen bir  $f_k$  frekansında devrenin çıkışı  $f_k$  frekansı girilerek elde edilebilir. Ölçülen performans parametre çok boyutlu fonksiyonundan hareketle, sonsuz boyutlu diğer ifadeyle çalışma koşullarını içeren performans parametre fonksiyonu kestirimi yapılmaktadır.

$S^{(k)}$  ve  $N^{(k)}$  performans vektörleri bir kere hesaplandığında, aktif mikrodalga elemanı Şekil 3.5'deki gibi  $f_k$  frekansında bir kara kutu ile temsil edilebilir. Bu da Wedge ve Ruthledge (1992) tarafından verilen formda aşağıdaki gibi gösterilebilir.

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \quad (3.3)$$

Burada  $c_1$  and  $c_2$  zamanla değişen kompleks rastgele değişen gürültü dalgalarıdır. Bunlarda korelasyon matrisi  $C_s$  ile belirlidir ve aşağıdaki gibi verilebilir.

$$\left[ C_s \right] = \begin{bmatrix} |c_1|^2 & \overline{c_1 c_2^*} \\ \overline{c_2 c_1^*} & |c_2|^2 \end{bmatrix} \quad (3.4)$$

mutlak ergodiklik ve ortak geniş-anlamda stasyonær işlev kabulü altında üst çizgi zaman ortalamasını gösterir.  $C_s$  'in diagonaldeki terimler 1 Hz band genişliğindeki sonlandırmalara aktarılacak gürültü gücüdür. Çapraz-diagonal terimler ise çapraz ilişki terimleridir. Gürültü korelasyon (ilişki) matrisi  $C_s$  Hermitian matristir. Bu parametreler de  $S, N$  performans-parametrelerinin fonksiyonu olarak aşağıdaki şekilde ifade edilebilir.

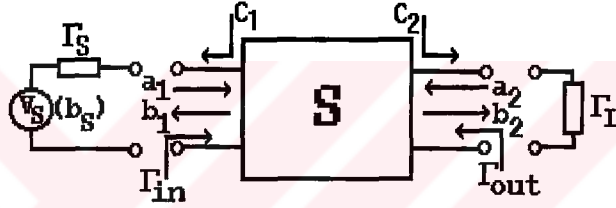
$$\overline{|c_1|^2} = kT_{\text{opt}} \left( |s_{11}|^2 - 1 \right) + \frac{kT |1 - s_{11} \Gamma_{\text{opt}}|^2}{|1 + \Gamma_{\text{opt}}|^2} \quad (3.5)$$

$$\overline{|c_2|^2} = |s_{21}|^2 (kT_{\text{opt}} + \frac{kt|\Gamma_{\text{opt}}|^2}{|1 + \Gamma_{\text{opt}}|^2}) \quad (3.6)$$

$$\overline{c_1 c_2^*} = \frac{-s_{21}^* \Gamma_{\text{opt}}^* kt}{|1 + \Gamma_{\text{opt}}|^2} + \frac{s_{11}}{s_{21}} \overline{|c_2|^2} \quad (3.7)$$

burada  $k$  Boltzmann's sabiti,  $Z_0$  normalizasyon empedansı,  $kt$  normalize ısı enerjidir.  $kt$  ve  $T_{\text{opt}}$  sırasıyla aşağıdaki gibi verilebilir.

$$kt = \frac{4kT_0 R_n}{Z_0}; \quad T_{\text{opt}} = (F_{\text{min}} - 1)T_0 \quad (3.6)$$



Şekil 3.2 Mikrodalga Elemanının kara kutu ile temsil edilmesi

Bir aktif elemanın dönüştürücü güç kazancı, yüke aktarılan güçle kaynaktan çekilebilecek en büyük güç oranı şeklinde hesaplanır. Dönüştürücü güç kazancı  $S$  parametreleri,  $\Gamma_s$  ve  $\Gamma_L$  sonlandırmalarının fonksiyonu olarak aşağıdaki gibi verilebilir (Vendelin v.d (1990)).

$$G_T = \frac{P_L}{P_{A,S}} = G_T(\Gamma_s, \Gamma_L, S) = \frac{(1 - |\Gamma_L|^2) |s_{21}|^2 (1 - |\Gamma_s|^2)}{|1 - s_{22} \Gamma_L|^2 |1 - \Gamma_{in} \Gamma_s|^2}$$

veya

$$G_T(\Gamma_s, \Gamma_L, S) = \frac{(1 - |\Gamma_L|^2) |s_{21}|^2 (1 - |\Gamma_s|^2)}{|1 - \Gamma_{out} \Gamma_L|^2 |1 - s_{22} \Gamma_s|^2} \quad (3.7)$$

Aktif elemanın sırasıyla giriş ve çıkış konjüge uygunsuzluk kaybı olan  $M_{in}$  ve  $M_{out}$  aşağıdaki gibi verilebilir.

$$M_{in} = \frac{P_{A,S}}{P_{in}} = M_{in}(\Gamma_S, \Gamma_L, S) = \frac{|1 - \Gamma_{in}\Gamma_S|^2}{(1 - |\Gamma_{in}|^2)(1 - |\Gamma_S|^2)} \quad (3.8)$$

$$M_{out} = \frac{P_{Aout}}{P_L} = M_{out}(\Gamma_S, \Gamma_L, S) = \frac{|1 - \Gamma_L\Gamma_{out}|^2}{(1 - |\Gamma_L|^2)(1 - |\Gamma_{out}|^2)} \quad (3.9)$$

burada

$$\Gamma_{out} = s_{22} + \frac{s_{12}s_{21}\Gamma_S}{1 - s_{11}\Gamma_S}, \quad \Gamma_{in} = s_{11} + \frac{s_{12}s_{21}\Gamma_L}{1 - s_{22}\Gamma_L} \quad (3.10)$$

Bir aktif mikrodalga elemanın gürültü figürü, giriş ve çıkıştan elde edilebilecek işaret-gürültü oranının oranı şeklinde tanımlanır. N vektörü gürültü figürü F'in kaynak sonlandırma yansımaya katsayısı  $\Gamma_s$  'e bağlılığını tanımlar. Bunlar arasındaki bağlantı Vendelin v.d. (1990) tarafından verilen çalışmadaki şekliyle aşağıdaki gibidir:

$$F(\Gamma_S, N) = F_{min} + 4 \frac{R_N}{Z_0} \frac{|\Gamma_S - \Gamma_{opt}|^2}{(1 - |\Gamma_S|^2)|1 + \Gamma_{opt}|^2} \quad (3.11)$$

Son yıllarda subnetwork-growth (SGM) olarak adlandırılan metod Gupta v.d. (1981) ,aktif veya pasif iki kapılıların rasgele birleştirilmelerinden oluşturulmuş, çok kapılı devrelerin işaret ve gürültü analizinin bilgisayar destekli tasarımı (CAD) için uygulanmıştır. SGM'ye dayandırılan bir çok genel amaçlı CAD mikrodalda devreleri analiz programı mikrodalga tümleşik devrelerinin analizi için uygulanmıştır. Bir sonuç olarak aktif mikrodalga elemanlarının bütün çalışma frekanslarında kara kutu ile karakterizasyonu özellikle önemli olmaya başlamıştır. İki kapılı aktif elemanın bir çalışma frekansında S ve N vektörlerini belirledikten sonra ,  $(\Gamma_S, \Gamma_L)$  sonlandırma çifti, çalışma bandında  $G_T, F, M_{in}, M_{out}$  performans fonksiyonlarını uzlaştırmak suretiyle belirlenebilir. Bir tipik örnek güneş v.d (1994) tarafından verilmiştir. Burada  $(\Gamma_S, \Gamma_L)$  çifti bir çalışma frekansında istenilen F ve  $M_{in}$  kararlı en büyük kazanç  $G_T$  için hesaplanmıştır.

### 3.3 Eşdeğer YSA Modelleri

Yukarıda kısaca mikrodalga mühendisliğinde aktif mikrodalga elemanlarının analizinin temeli verildikten ,S ve gürültü parametrelerinin rolü anlaşıldıktan sonra bir kara kutu işlevi görecektir olan eşdeğer YSA modellerini tanıtmaya başlayalım. Çalışmalarımızın başlangıcında kullandığımız eşdeğer YSA modellerimiz sadece S-parametrelerini, gürültü parametrelerini veya hem gürültü hemde S-

Parametrelerini belirli bir frekans bandında belirlemeye yöneliktir. Daha sonraları çok boyutlu bir küçük-ışaret-gürültü modeli oluşturmak için kutuplama noktası ve konfigürasyon tipi girişi ilave edilmiştir(Şekil 3.6)(Hamid TORPİ'ye ait makalelere bakınız.).

Bu çalışmada sürekli değerli girişe sahip , ileri beslemeli bir yapay sinir ağı olan çok katmanlı algılayıcı (ÇKA) kullanılmıştır. Eğitim algoritması olarak da geriye yayılma algoritması (GYA) kullanılmıştır. Geriye yayılım eğitim algoritması çok katmanlı algılayıcı'nın aktif çıkışıyla arzu edilen çıkış (Hedef uzayla belirlenen) arasındaki karesel ortalama hatanın minimize edilmesine dayanan bir gradyant (Türev temelli arama) algoritmasıdır. Dolayısıyla bu teknik sürekli türevi alınabilir nonlineer transfer fonksiyonuna ihtiyaç gösterir. Öğrenme işlemi GYA ile , ağırlıklar ayarlanarak gerçekleştirilir.

Lipmann (1987), çalışmasında doğru ile ayrılabilen bölgelerin tanıtılabilmesi için bir, konveks bölgelerin tanıtılabilmesi için iki, en karmaşık karar bölgelerinin tanıtılabilmesi için üç katmanlı algılayıcı 'un yeterli olduğunu göstermektedir.

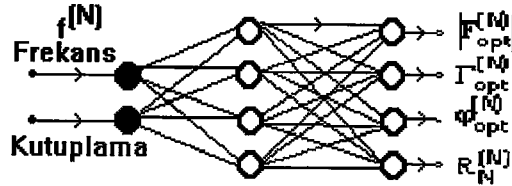
Bu çalışma için geliştirilen YSA için giriş katmanı, bir gizli katman ve çıkış katmanı olmak üzere iki katmanlı algılayıcı olarak adlandırılan yapının yeterli olduğu görülmüştür. Ağırlıkların başlangıç değerleri seçiminin, sonucu etkilemediği gözlenmiştir, dolayısıyla rasgele alınmıştır.

Bu aşamada geliştirdiğimiz eşdeğer YSA modelleri aşağıda sıralanmıştır. Her bir modelde YSA olarak Çok Katmanlı Algılayıcı (MLP: Multi Layer Perceptron) kullanılmıştır. Şekil 3.3'den Şekil 3.5'e kadar sıralanan eşdeğer YSA modellerindeki frekansa ilave kutuplama girişi sonuca daha hızlı yakınsaması için ikinci bir giriş olarak kullanılmıştır(Karlık(1994)). Bu kutuplama girişinin transistörün kutuplaması ile alakası yoktur. Bu ilk modellere kısaca değinildikten sonra diğerlerinin kapsayan çok boyutlu modelleme daha ayrıntılı olarak tanıtılacaktır.

$$\text{Kutuplama} = \sqrt{\frac{1}{\sum_{i=1}^{\text{Ö.S}} f_i^2}} \quad f_i = i. \text{Örnek frekansı} ; \text{Ö.S} : \text{Örnek Sayısı} \quad (3.1)$$

Bu şekilde kutuplama noktası adı altında ikinci bir giriş ilave etmek öğrenme hızını arttırmıştır. Örneğin 1,2,3,4,5,6,8,10,12,14 GHz noktalarındaki S parametrelerinden oluşan hedef uzayla yapılan sabit iterasyon sayılı eğitim için , bütün frekanslarda sabit olan ikinci bir giriş olarak yukarıdaki formülle hesap edilen değer (7.714) ilave kutuplamasıyla kullanılarak bulunan hata yarıya indirilmiştir.

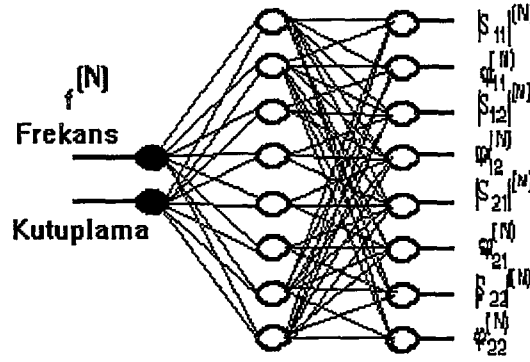
#### Eşdeğer Gürültü YSA Modeli:



Şekil-3.3 Gürültü parametreleri için kullanılan ÇKA yapısı

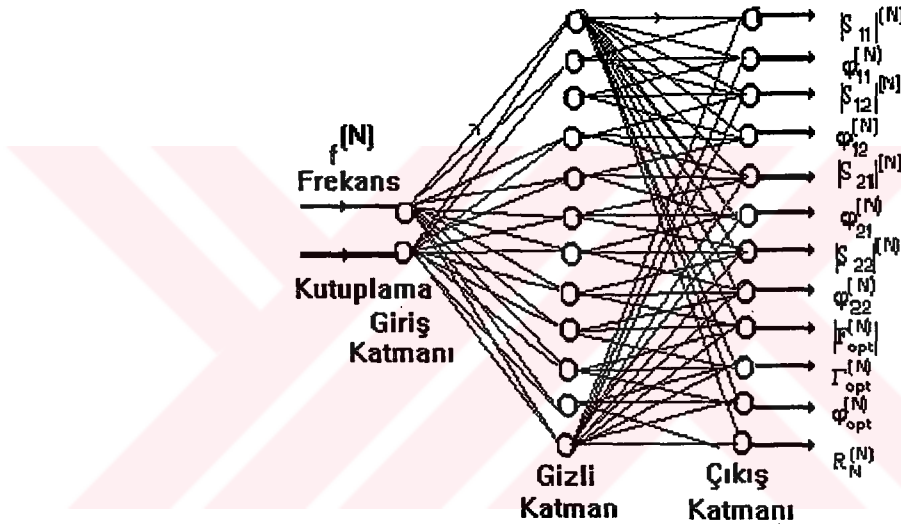


### Eşdeğer İşaret Parametreleri YSA Modeli:



Şekil 3.4 İşaret parametreleri için kullanılan ÇKA yapısı

### Eşdeğer İşaret&Gürültü Parametreleri YSA Modeli:



Şekil 3.5 Hem gürültü hem de sinyal parametreleri için kullanılan ÇKA yapısı

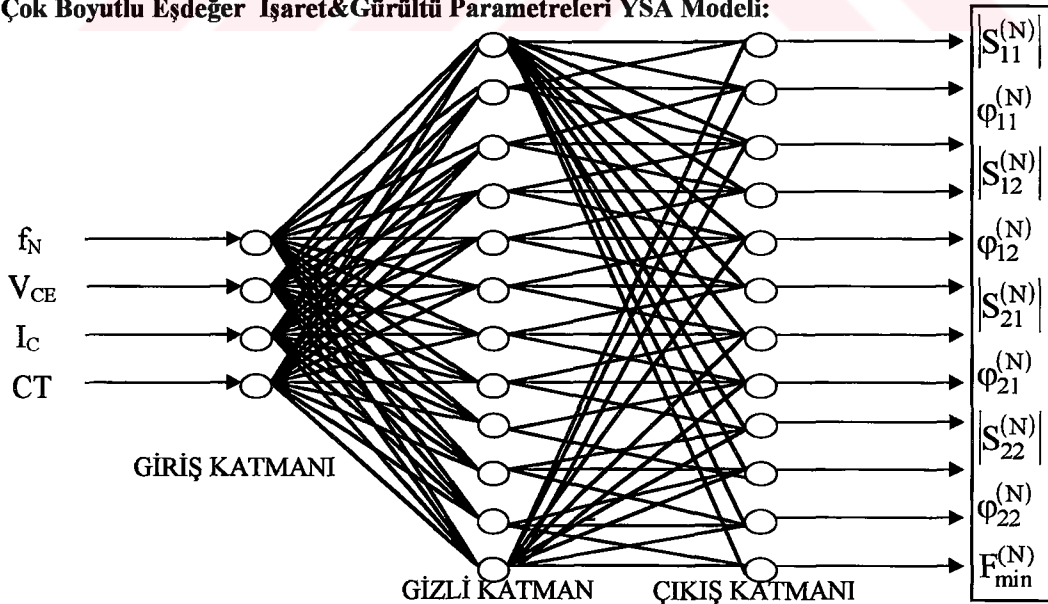
Tablo-4.6 incelendiğinde görülüyor ki YSA 'nın gösterdiği performans açısından Normalize hata teriminden hareketle en iyi sonuçlar Şekil 3.4'deki ([S]) yapı için sonra da Şekil 3.5'deki yapıyla elde edildiği görülüyor.(Bak.Tablo-4.7) Bu sonuçlar normalize değerler gözönüne alındığında elde edilmiştir.Bütün parametreler için gerçek değerler düzleminde (ki bizim için önemli olan budur) durum ayrıntılı ( yani hem sinyal hemde gürültü parametrelerinin hataları ayrı ayrı gözönüne alınarak) olarak incelenirse Şekil 3.5 'deki yapıyla sinyal parametrelerinin sonuç olarak en iyi performansla ( en az hatayla ) öğretilbildiği görülmüştür. Örneğin 1,2,3,4,5 GHz'de alınan örneklerle teşkil edilen hedef uzayının eğitilmesiyle elde edilen sonuçta Şekil-3.5'deki yapı için  $E_T=0.0044$  (Bak. Tablo-4.7) Şekil-3.4'de ki YSA yapısı için  $E_T=0.0077$  (Bak. Tablo-4.7) lik hatayla sonuçlar elde edilmiştir.(Bak. Tablo-4.7) Ve yine daha belirgin bir örnek olarak 1,2,3,4,5,6,8 GHz

'de alınan örneklerle teşkil edilen hedef uzayının öğretilmesi ile elde edilen sonuçta Şekil 3.5 'deki yapı için  $E_T=0.0064$ , Şekil 3.4'deki yapı için ise  $E_T=0.01289$  (%1.29)'lık hatalarla öğretilmiştir. Sonuç olarak Şekil 3.5'deki yapıyla yani Gürültü ve Sinyal parametrelerinin birlikte öğretilmeye çalışıldığı durumda sinyal parametreleri daha iyi eğitilebilmiş gürültü parametrelerinin tek başına eğitildiği yapıyla elde edilen sonuçlardan biraz daha kötü olmasına rağmen onlara yaklaşmıştır. Bu durumda sinyal ve gürültü parametrelerinin eğitilmesi için Şekil 3.5'deki YSA yapısından yararlanılarak sonuca gitmeye çalışmak daha avantajlı olduğunu söyleyebiliriz. Ama biz sadece Gürültü veya sadece Sinyal parametrelerini öğretmek istiyorsak Şekil 3.3 ve Şekil 3.4 'deki YSA yapılarını tercih etmek ( Zaman açısından da ayrıca avantajlı) muhakkak ki daha avantajlıdır. Bu sonuçlar Hu v.d. (1993) ve Roux v.d. (1995) tarafından belirtilen sinyal ve gürültü parametrelerinin ortak eşdeğer bir devreden hareketle hesaplanmasıyla daha iyi bir model kurulacağını saptayan makaleyle elde ettiğimiz sonuç itibarıyla bir paralellik arz etmektedir.

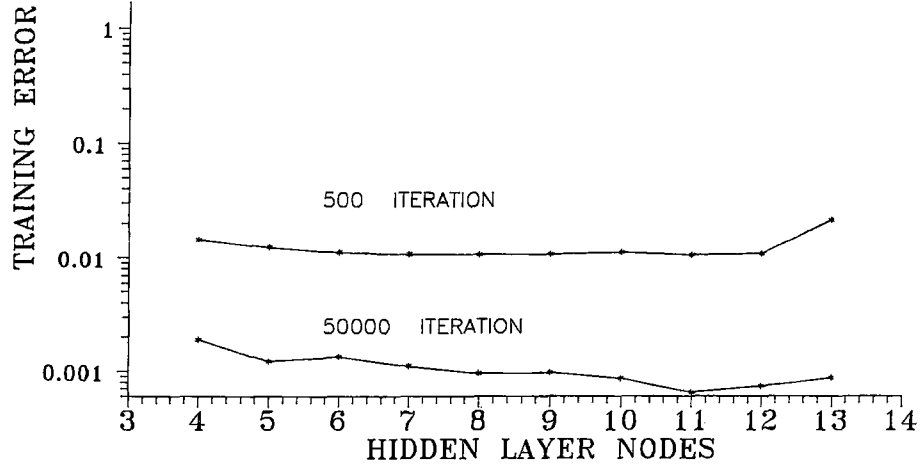
Zaman-hata optimum ilişkisini sağlamak için her bir yapıda çıkış katmanı düğüm sayısı gizli katman düğüm sayısına eşit alınmıştır. Şekil 3.7'de bunu teyid eden hatanın gizli katman düğüm sayısı ile değişimi çizdirilmiştir.

N72000aFET için öğrenme katsayısı  $\epsilon=0.5$  momentum katsayısı  $\alpha=0.8$  için en optimum öğrenme gerçekleşmiştir. Burada özellikle momentum teriminin önemine değinmek gerekiyor. Zira öğrenme katsayısı küçüldükçe öğrenme işlemi ( zaman açısından ) yavaşlar fakat hassas bir öğrenme ( doğruluk artar) gerçekleşir. Momentum katsayısı ( $0 < \alpha < 1$ ) fikri bu noktadan hareketle ortaya atılmıştır. Momentum mevcut delta ağırlığı üzerinden önceki delta ağırlığının belli bir kısmını besler. Böylece daha düşük öğrenme katsayısı ile daha hızlı öğrenme temin edilir.

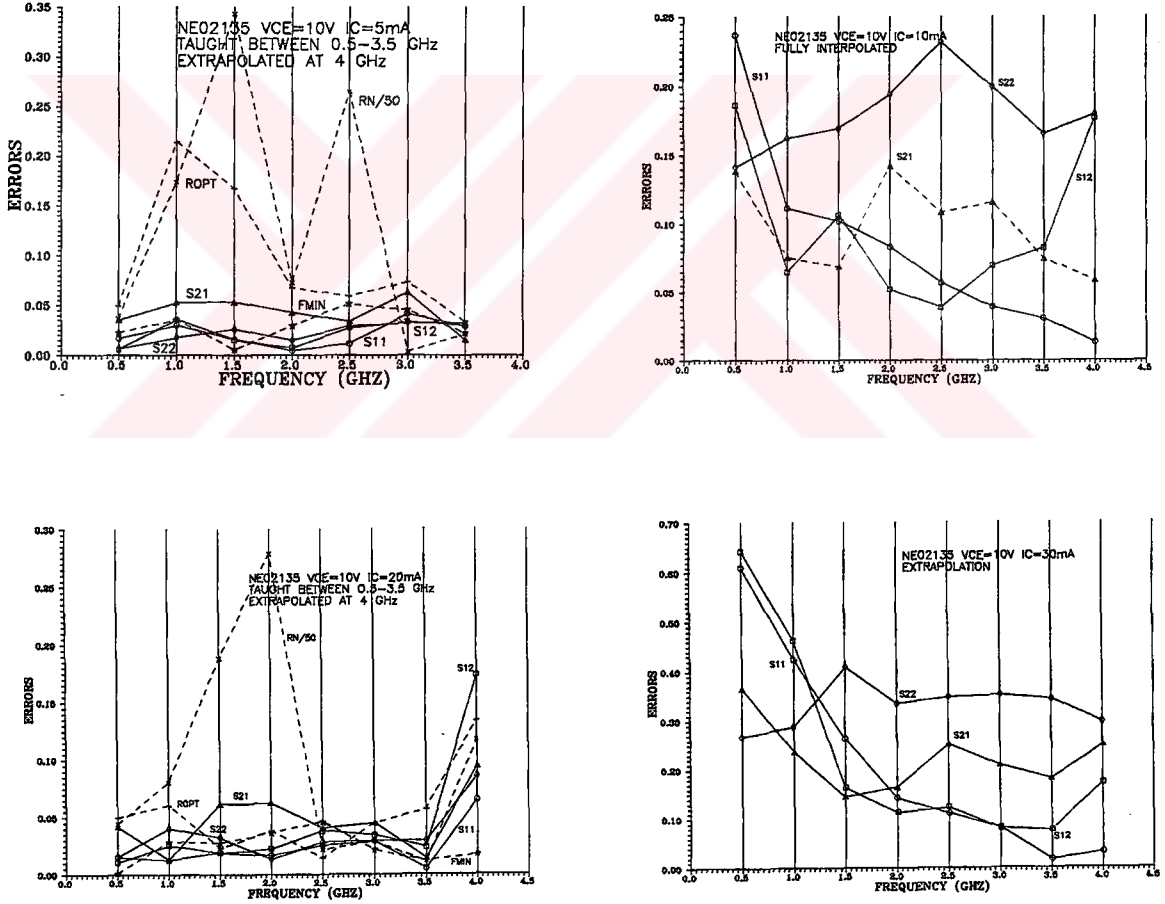
#### Çok Boyutlu Eşdeğer İşaret&Gürültü Parametreleri YSA Modeli:



Şekil 3.6 Hem gürültü hem de sinyal parametrelerini öğretmek için kullanılan çok boyutlu modellemeyi temin eden MLP yapısı



Şekil 3.7 İterasyon sayısı parametre olmak üzere hatanın gizli katman düğüm sayısı ile değişimi (NE02135)



Şekil 3.8 Çoklu kutuplamalı NE02135 transistörünün işaret ve gürültü parametrelerinin hata dağılımı grafiği.



Bu çok boyutlu modellemeyi temin eden YSA'nın eğitilmesi için üretici firmanın verdiği muhtelif kutuplama ve konfigürasyon tipindeki performans datasının kullanılarak hedef uzayının teşkil edilmesi ile sağlanmıştır. Eğitim algoritması olarak geriye yayılım algoritması kullanılmıştır. Hedef uzayını teşkil ederken kullanılacak datanın adedini modelden optimum performansı elde edilecek şekilde belirlenmiştir (Bakınız bölüm 4.3). Eğitim işlemi tamamlandıktan sonra, istenilen  $f_k$  frekansında,  $(V_{DS}, I_{DS})$  ile belirli kutuplama noktasında, her hangi bir konfigürasyon tipiyle belirli giriş vektörü için aktif mikrodalga elemanın çıkışı olarak İşaret ve gürültü parametre vektörleri  $S^{(k)}$  ve  $N^{(k)}$  kestirilebilir. Eşdeğer YSA'da konfigürasyon tipi belirli sayılarla temsil edilmiştir. Eğer  $S^{(k)}$ ,  $N^{(k)}$  hedef uzayı içinde değilse YSA'nın extrapolasyon özelliği ile bu değerle tabii olarak hesaplanabilecektir. Bu YSA'nın eğitilmemiş girişler içinde güzel çıkış verme özelliğinin bir yansımasıdır. Bizim uygulamamızda YSA işaret-gürültü eşdeğeri sadece uzayını kapsayan frekans ve kutuplama kademesinde değil (interpolasyon) bu frekans ve kutuplama kademesinin dışında da kestirim (extrapolasyon) yapabilme yeteneği vardır (Şekil 3.8). Bunlara ait tüm uygulamalara 4. bölümde yer verilmiştir.

### 3.4 Hedef Uzayının Belirlenmesi ve Genelleştirme

Bu bölümde tüm bandı öğretebilmek için acaba kaç örnek alınarak (teşkil edilen hedef uzayının öğretilmesi) en az hatayla eğitim yapılabilir? Sorusuna cevap arayacağız. Ayrıca eğitmenin yapıldığı örneklerin dışındaki frekanslarda geliştirilen MLP'nin davranışı bizim uygulamamız için en önemli kriterdir. Genelleştirme ile de kastedilen budur.

Bu bölümde elimizdeki bilgisayar alt dizinlerindeki (Directory) transistör örnekleri içinde en fazla ölçülmüş değere sahip olan N32684AA FET baz alınmıştır. Zira ara frekanslarda devrenin davranışının test edilebilmesi için mümkün olduğu kadar çok ölçülmüş örneğin elimizde bulunması gerekir. Bunun için sadece Sinyal parametreleri ele alınmış yani sadece Şekil 3.4'deki yapı kullanılmıştır. Bu sefer  $\alpha = 0.1$  olarak değiştirilmiştir. Epsilon öğrenme katsayısı değişmemiştir.

N32684AA FET 'i tüm kullanma bandında eğitmek amacıyla sırasıyla 0.5, 1, 2, 4 ve 5 GHz aralıklarla alınan düzgün örnekler sırasıyla 41, 21, 11, 6 ve 5 örnekle teşkil edilen hedef uzayları öğretilmiştir. Eğitim sonunda elde edilen neticeler tüm çalışma bandı için test edilmiştir. Elde edilen neticeler Şekil 4.12'da Sinyal parametreleri için toplam bağıl hata  $E_T$  düşey eksen örnek sayısı yatay eksen ve iterasyon sayısı parametre olmak üzere çizdirilmiştir. Ayrıca Şekil 4.13'de iterasyon sayısı değişken hedef uzayları parametre olmak üzere sinyal parametrelerine ait toplam bağıl hata  $E_T$  hesap ettirilerek çizdirilmiştir.

Elde edilen tüm bu sonuçlar değerlendirildiğinde görülmüştür ki 21 örnekle (1 GHz aralıklarla örnekler alınan) belirlenen hedef uzayının kullanılması ile tüm bandın en az hatayla MLP tarafından doğru bir şekilde temsil edilebildiği görülmüştür. Bunu 11 örnekle (2 GHz aralıklarla) hedef uzayının kullanıldığı kullanılması ile eğitilen MLP takip etmektedir. Bu sonuçlar bize bizim örneğimizdeki

MESFET 'i ( Yada herhangi bir Sinyal parametreleri ile temsil edilebilen mikrodalga elemanı ) tüm kullanma bandı boyunca Sinyal parametreleri açısından temsil edecek MLP 'un eğitilmesi için optimum bir örnekleme sayısının olduğunu ve eldeki bütün ölçülmüş örneklerle tüm bandı öğretmeye çalışmanın yanlış olduğunu göstermektedir.

### 3.5 Çok Boyutlu Eşdeğer İşaret&Gürültü YSA Modelinin Matematiksel Olarak Tanımlanması

#### 3.5.1 Devrenin Yapısı

YSA biyolojik sinir ağının matematiksel modelidir. Birbirine bağlantılı nöronlar topluluğundan ibarettir.

$$\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^t \quad (3.2)$$

$$\mathbf{y} = [y_1 \ y_2 \ \dots \ y_p]^t \quad (3.3)$$

YSA işaret ve gürültü eşdeğerinde, sırasıyla n ve p boyutlu giriş ve çıkış vektörleri olsun.  $x$   $f_k$  frekansını,  $(V_{DS}, I_{DS})$  kutuplama noktasını ve konfigürasyon tipini içeren 4-boyutlu giriş vektörü,  $y$  ise S- ve N- parametrelerini veren 12-boyutlu çıkış vektörüdür.  $x$  ve  $y$  arasındaki ilişki çok boyutlu ve nonlineerdir. Böyle çok boyutlu nonlineer bir ilişkiyi modellemek için, üç katmanlı bir YSA'ı kullanılmıştır. Şekil 3.6 temsil edildiği gibi giriş katmanında n adet , çıkış katmanında p adet ve gizli katmanda q adet işlem elemanı (nöron, düğüm) bulunduğunu düşünelim. İşlem elemanının yapısı Şekil 2.2' de verilmiştir.

$$\mathbf{a}^{(k)} = [a_1^{(k)} \ a_2^{(k)} \ \dots \ a_n^{(k)}]^t \quad (3.4)$$

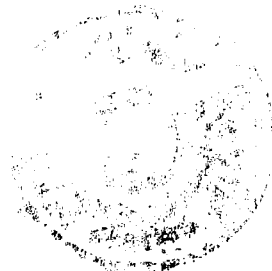
$$\mathbf{b}^{(k)} = [b_1^{(k)} \ b_2^{(k)} \ \dots \ b_p^{(k)}]^t \quad (3.5)$$

burada vektörler sırasıyla giriş ve çıkışın k ıncı örneğini temsil etmektedir.  $k=1,2,\dots,N_s$  'e kadar üzere  $N_s$  toplam örnek sayısıdır. Gizli katmanla giriş düğümü arasındaki bağlantı ağırlık matrisi  $\mathbf{W}$ , ve gizli katmanla çıkış katmanı arasındaki bağlantı ağırlık matrisi  $\mathbf{V}$  olmak üzere aşağıdaki gibi yazılabilir.

$$\mathbf{W} = [ \mathbf{W}_1, \mathbf{W}_2 \ \dots \ \mathbf{W}_h \ \dots \ \mathbf{W}_q ] \quad (3.6)$$

$$\mathbf{V} = [ \mathbf{V}_1 \ \mathbf{V}_2 \ \dots \ \mathbf{V}_h \ \dots \ \mathbf{V}_p ] \quad (3.7)$$

Burada  $\mathbf{W}_h$  h. gizli katman düğümü ile giriş katmanı arasındaki ağırlık vektörüdür:



$$\mathbf{W}_h = [ W_{1h} \ W_{2h} \ \dots \ W_{nh} ]^t \quad (3.8)$$

$V_j$  ise j. Çıkış düğümü ile gizli katman arasındaki ağırlık vektörüdür.

$$\mathbf{V}_j = [ V_{1j} \ V_{2j} \ \dots \ V_{qj} ]^t \quad (3.9)$$

Gizli katman çıkışından j. Çıkış düğümüne iletilen işare aşağıdaki formda yazılabilir.

$$\Phi_j(\mathbf{x}, \mathbf{W}, \mathbf{V}_j, \theta) = \sum_{h=1}^q V_{hj} g_h(\mathbf{x}, \mathbf{W}_h, \theta_h) \quad (3.10)$$

j. Çıkış düğümünden elde edilen net çıkış aşağıdaki gibi yazılabilir.

$$y_j(\mathbf{x}, \mathbf{W}, \mathbf{V}_j, \theta, T_j, \Theta_j) = T_j f_j(\Phi_j + \Theta_j) \quad (3.11)$$

burada  $g_h$  ve  $f_j$  sırasıyla h. Gizli katman düğümü ile j. Çıkış düğümünün aktivasyon fonksiyonudur (transfer fonksiyonu), Ki bizim uygulamamızda genellikle sigmoid tip nonlinear aktivasyon fonksiyonları kullanılmıştır.  $g_h(\mathbf{x}, \mathbf{W}_h, \theta_h)$  aşağıdaki formda ifade edilebilir:

$$g_h(\mathbf{x}, \mathbf{W}_h, \theta_h) = \frac{1}{1 + \exp\left(-\left(\sum_{i=1}^n x_i W_{ih}\right) - \theta_h\right)} \quad (3.12)$$

(3.11) ve (3.12) ifadelerinde,  $\Theta_j$  ve  $\theta_h$  sırasıyla j. Çıkış ve h. Gizli katman düğümlerinin yerel belleğidir.(eşik)

### 3.5.2 Devre Matrisi P 'nin Hesaplanması

Eğer mimarisinin parametreleri P matrisi ile ifade edilecek olursa, devre parametre matrisi P , sırasıyla giriş katmanı-gizli katman ve gizli katman-çıkış katmanı arası ağırlık faktörlerini, gizli katman ve çıkış katmanı düğümlerinin yerel belleklerini içeren  $n \times q + p \times q + q + p$  tane elemanı olan bir matrisdir. Eğitim prosedürü devre parametresi matrisi P nin aşağıdaki tanımlanan hata fonksiyonun minimize edilmesi suretiyle hesaplanmasından ibarettir.

$$E(\mathbf{P}) = \sum_{k=1}^{N_S} E(k) = \sum_{k=1}^{N_S} \left[ \frac{1}{2} \sum_{j=1}^p \left( y_j^{(k)} - b_j^{(k)} \right)^2 \right] \quad (3.13)$$

burada  $y_j^{(k)}$  ve  $b_j^{(k)}$  sırasıyla  $f_k$  eğitime frekansında  $j$ . Çıkış katmanı düğümünün kestirilen (tahmin edilen) ve ölçülen değerleridir. Bu eğitime işlemi geriye yayılım (BP) olarak adlandırılan ki iki değişik tip algoritmaya sahiptir. Bunlardan biri her bir çıkış örneği için devre parametre matrisi  $\mathbf{P}$  nin yeniden hesaplandığı 'on-line' eğitime olarak adlandırılır diğesinde ise bütün hesap uzayı değerlendirildikten sonra devre parametre matrisinin değiştirildiği için 'off-line' eğitime olarak adlandırılır. Bizim çalışmamızda bir çok uygulamada daha etkin olan 'on-line' eğitime kullanılmıştır. Devre parametrelerinin  $V_{hj}, W_{ih}, \theta_h$  içi değiştirilme ifadeleri aşağıdaki gibi verilebilir:

$$V_{hj}^{(k+1)} = V_{hj}^{(k)} - \eta \frac{\partial E^{(k)}}{\partial V_{hj}} + \alpha \left( V_{hj}^{(k)} - V_{hj}^{(k-1)} \right) \quad (3.14)$$

$$W_{ih}^{(k+1)} = W_{ih}^{(k)} - \eta \frac{\partial E^{(k)}}{\partial W_{ih}} + \alpha \left( W_{ih}^{(k)} - W_{ih}^{(k-1)} \right) \quad (3.15)$$

$$\theta_h^{(k+1)} = \theta_h^{(k)} - \eta \frac{\partial E^{(k)}}{\partial \theta_h} + \alpha \left( \theta_h^{(k)} - \theta_h^{(k-1)} \right) \quad (3.16)$$

benzer ifadeler  $\Theta_j$  ve  $T_j$  içinde yazılabilir. (3.14)-(3.16) arası ifadelerde  $\eta$  sıfırdan büyük olmak üzere öğrenme hızı ve  $\alpha$  0 ile 1 arasında değer alan momentum terimidir. Bu çalışmada momentum terimi öğrenme sırasında yerel minimumlara yakalanmamak ve en kısa zamanda eğitimi tamamlamak için ilave edilmiştir. Optimum alpha momentum terimini tesbit edebilmek için basit bir program geliştirilmiştir (EK 1.3). Öğrenme katsayısı ise eşdeğer YSA modeli simülatorü içinde otomatik olarak ayarlanmaktadır. Öyleki öğrenme katsayısı terimi hata arttıkça; belirli bir oranda artmakta, buna bağlı olarak öğrenme hızı artmakta, öğrenme hassasiyeti ise düşmektedir. Bunun aksine hata azaldıkça; öğrenme katsayısı yine belirli bir oranda azalmakta, buna bağlı olarak öğrenme hızı düşmekte, öğrenme duyarlılığı ise artmaktadır. Her hangi bir set devre parametre matrisi ile işe başladıktan sonra

her parametre (3.14)-(3.16) ifade edildiği gibi  $\frac{\partial E^{(k)}}{\partial V_{hj}}, \frac{\partial E^{(k)}}{\partial W_{ih}}, \frac{\partial E^{(k)}}{\partial \theta_h}$  duyarlılık terimleriyle orantılı

olarak değiştirilmektedir. Bu değiştirilme işlemi eğitime işleminde istenilen hassasiyete erişilinceye veya hata terimindeki azalma duruncaya kadar devam edilir. Böylelikle YSA'nın  $V_{hj}, W_{ih}, \theta_h, T_j$  and  $\Theta_j$  dan oluşan nihai devre parametresi seti tesbit edilmiş olur.  $V_{hj}, W_{ih}, \theta_h$  'ya bağlı olan YSA devresinin duyarlılık terimlerini  $Z_h$  ve  $F_j$  aşağıdaki gibi tanımlanır:



$$Z_h = g_h(\mathbf{x}, \mathbf{W}_h, \theta_h) = \frac{1}{1 + \exp\left(-\left(\sum_{i=1}^n x_i W_{ih}\right) - \theta_h\right)} \quad (3.17)$$

$$F_j = f_j(\Phi_j + \Theta_j) = \frac{1}{1 + \exp(-\Phi_j - \Theta_j)} \quad (3.18)$$

$$\frac{\partial E^{(k)}}{\partial V_{hj}} = \frac{\partial \left[ \frac{1}{2} \sum_{j=1}^p (y_j - b_j)^2 \right]}{\partial V_{hj}} \quad (3.19)$$

$$\frac{\partial E^{(k)}}{\partial V_{hj}} = (y_j - b_j) T_j \frac{\partial f_j}{\partial \Phi_j} \frac{\partial \Phi_j}{\partial V_{hj}} \quad \text{Final}$$

$$\frac{\partial E^{(k)}}{\partial V_{hj}} = (y_j - b_j) T_j F_j (1 - F_j) Z_h = \delta_j^{(3)} Z_h$$

$$\frac{\partial E^{(k)}}{\partial W_{ih}} = \sum_{j=1}^p \frac{\partial E^{(k)}}{\partial y_j} \frac{dy_j}{dF_j} \frac{dF_j}{d\Phi_j} \frac{\partial \Phi_j}{\partial Z_h} \frac{\partial Z_h}{\partial W_{ih}} \quad (3.20)$$

$$\frac{\partial E^{(k)}}{\partial W_{ih}} = \left[ \sum_{j=1}^p (y_j - b_j) T_j F_j (1 - F_j) V_{hj} \right] Z_h (1 - Z_h) a_i^{(k)}$$

$$\frac{\partial E^{(k)}}{\partial W_{ih}} = \sum_{j=1}^{N_p} \delta_j^{(3)} V_{hj} Z_h (1 - Z_h) a_i^{(k)} = \delta_h^{(2)} a_i^{(k)}$$

$$\frac{\partial E^{(k)}}{\partial \theta_h} = \sum_{j=1}^p \frac{\partial E^{(k)}}{\partial y_j} \frac{dy_j}{dF_j} \frac{dF_j}{d\Phi_j} \frac{\partial \Phi_j}{\partial Z_h} \frac{\partial Z_h}{\partial \theta_h}$$

$$\frac{\partial E^{(k)}}{\partial \theta_h} = \sum_{j=1}^p (y_j - b_j) T_j F_j (1 - F_j) V_{hj} Z_h (1 - Z_h) \quad (3.21)$$

$$\frac{\partial E^{(k)}}{\partial \theta_h} = \sum_{j=1}^{N_p} \delta_j^{(3)} V_{hj} Z_h (1 - Z_h) = \delta_h^{(2)}$$

$$\frac{\partial E^{(k)}}{\partial T_j} = \frac{\partial E^{(k)}}{\partial y_j^{(k)}} \frac{\partial y_j^{(k)}}{\partial T_j} = (y_j - b_j) F_j$$

$$\frac{\partial E^{(k)}}{\partial T_j} = \delta_j^{(3)} T_j^{-1} (1 - F_j)^{-1} \quad (3.22)$$





$$\frac{\partial E^{(k)}}{\partial \Theta_j} = \frac{\partial E^{(k)}}{\partial y_j} \frac{\partial y_j}{\partial F_j} \frac{\partial F_j}{\partial \Theta_j} = (y_j - b_j) T_j F_j (1 - F_j) \quad (3.23)$$

$$\frac{\partial E^{(k)}}{\partial \Theta_j} = \delta_j^{(3)}$$

burada  $\delta_h^{(2)}$  ve  $\delta_j^{(3)}$  sırasıyla gizli ve çıkış katmanlarındaki her bir düğüme ait gradyanları temsil eder.  $\delta_h^{(2)}$  ve  $\delta_j^{(3)}$  (3.19) ve (3.21) kullanılarak aşağıdaki gibi ifade edilebilir.

$$\delta_h^{(2)} = \sum_{j=1}^N \delta_j^{(3)} V_{hj} Z_h (1 - Z_h) \quad (3.24)$$

$$\delta_j^{(3)} = (y_j - b_j) T_j F_j (1 - F_j) \quad (3.25)$$

### 3.6 Başarım Ölçütü

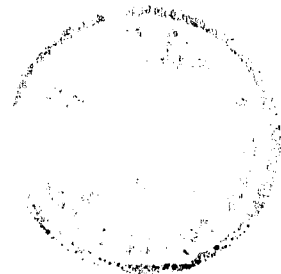
Kestirilen değer uzayının hedef uzayına yaklaşım ölçüsünü aşağıdaki kriterlerle tanımlamak uygun bulunmuştur:

$$E_{S_{ij}} = \frac{1}{N_S} \sum_{k=1}^{N_S} \frac{|S_{ij}^k \text{ölçülen} - S_{ij}^k \text{kestirilen}|}{|S_{ij}^k \text{ölçülen}|} \quad (3.26)$$

$$E_{N_i} = \frac{1}{N_S} \sum_{i=1}^{N_S} \frac{|N_i^k \text{ölçülen} - N_i^k \text{kestirilen}|}{|N_i^k \text{ölçülen}|} \quad (3.27)$$

$$E_T = \frac{1}{4} \sum_{i=1}^4 E_{i \text{işaret}} + \frac{1}{3} \sum_{i=1}^3 E_{i \text{gürültü}} \quad (3.28)$$

Burada n örnek sayısıdır.  $E_{S_{ij}}$  sinyal parametreleri için  $E_{N_i}$  gürültü parametreleri için ortalama bağıl hatayı göstermektedir.  $N_i$  sırasıyla  $F_{\min}, \Gamma_{\text{opt}}$  ve  $R_N$ ;  $S_{ij}$  sırasıyla  $S_{11}, S_{22}, S_{21}, S_{12}$  alınmalıdır.  $E_T$  Sinyal parametreleri ve gürültü parametreleri için toplam bağıl hatadır.



#### 4. YSA MODELİNİN AKTİF MİKRODALGA ELEMANLARINA UYGULANMASI

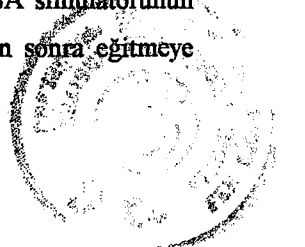
Bu bölümde EK1'de dökümü verilen bizim tarafımızdan geliştirilen PASCAL programlama dilinde yazılmış olan YSA simülatörünün çeşitli aktif mikrodalga elemanı için elde edilen simülasyon neticeleri gösterilecektir. Geliştirilen YSA simülatörünün performansı ölçülürken üretici firmaların verileri (ölçülen değerler olarak) esas alınmıştır. Üretici firma verileri üretici firmalarının (NEC, Siemens, Motorola, ... gibi) yayınlamış olduğu veri kitaplarından veya bu veri kitaplarının diskette dosya olarak depolanmış olarak temin edilen çeşitli konfigürasyon tipi ve/veya kutuplama koşulları ve frekansa bağlı olarak sunulan işaret [S] ve/veya gürültü parametrelerini içerir. Bu bölümde YSA'nın performansı hakkında fikir veren grafiklerde ölçülen değerler aksi belirtilmedikçe sürekli, modelleme sonucu elde edilen neticeler ise kesikli çizgilerle gösterilmiştir.

YSA modelimizin çok geniş bir bandı öğrenme yeteneğine sahiptir. Bunu için bölüm 4.1'de N24200A transistörüne ait simülasyon neticelerine (Şekil 4.1-4.3) bakıldığında 0,1-30 GHz 'lik oldukça geniş bir bandda işaret ve gürültü parametrelerinin oldukça iyi bir şekilde YSA'na öğretilbildiği görülür. Bölüm 4.2'de ise muhtelif konfigürasyon tipi ve muhtelif kutuplama noktalarının modele katılmasıyla elde edilen simülasyon neticeleri (Şekil 4.4-4.9) verilmiştir. Her bir frekans için ayrıntılı hata analizi (Şekil 4.5) de görmek mümkündür.

Üretici firma verileri tam istediğimiz zenginlikte olmadığı için örneğin N72000A transistörü için 0,1-18 GHz arasında sinyal parametresi verisine sahipken ancak 1-14 GHz arasında gürültü verisi ilave olarak verilmiştir. (Bakınız Tablo 4.5) Bu nedenle Şekil 4.10 ve Şekil 4.11'de verilen bu transistöre ait simülasyon neticeleri ancak 1-14 GHz aralığında gösterilebilmiştir.

Bölüm 4.3'de Şekil 4.12 ve Şekil 4.13 ile NE32684A transistörü üzerinde yapılan denemelerle hedef uzayının tesbitinde optimum örnek sayısının varlığına dikkat çekilmiştir. Bu transistörün bu amaçla kullanılmasının sebebi oldukça iyi sıklıkta hem gürültü hem de sinyal parametrelerinin üretici firma tarafından verilmiş olmasıdır. (Bakınız Tablo 4.8) Bölüm 4.4 ve bölüm 4.5'de ise tek kutuplama gerilimi ve muhtelif kutuplama akımları için sırasıyla sadece sinyal, işaret ve gürültü parametreleri birlikte olmak üzere eşdeğer YSA ile modellenmiştir. Bölüm 4.6'da ise muhtelif kutuplama akım ve gerilimleri için eşdeğer YSA modelinin performansı BFR28 transistörü üzerinde s-parametrelerinin kutuplama ile değişimi verilmiştir. (Şekil 4.30-4.37) Bu şekiller incelenecek olursa düşük frekanslarda işaret parametrelerinin kutuplamaya bağlılığının daha belirgin olduğu yüksek frekanslarda ise kutuplamaya bağlılığın etkisinin azaldığı anlaşılır.

Bölüm 4.8'de ise YSA simülatörünün muhtelif transistörlerle, iterasyon sayısını değişken olmak üzere, eğitilmesi için gerekli zaman hakkında fikir vermektedir. (Şekil 4.39) Ayrıca iterasyon sayısı ile hatanın nasıl değiştiğini gösteren grafiğe (Şekil 4.38) bakılacak olursa YSA simülatörünün hızla hatayı azaltmak suretiyle hedefe yakınsadığı ve belirli bir iterasyon sayısından sonra eğitime devam edilmesi durumunda ise öğrenme hızının düştüğü görülür.



#### 4.1 N24200A Üzerinde Geliştirilen Bir Sabit Kutuplamalı Eşdeğer Sinyal-Gürültü YSA Modeli İçin Elde Edilen Simülasyon Neticeleri .

Bu bölümde N24200A için tek bir kutuplama noktası civarında işaret ve gürültü parametreleri tek bir eşdeğer YSA'ı ile modellenmiştir.Band genişliği 0.1-30 GHz'dir.

##### 4.1.1 N24200A İçin Üretici Firma Verileri ve Simülasyon Neticeleri

Tablo 4.1 N24200A FET için verilen üretici firma sinyal ve gürültü parametreleri değerleri

! FILENAME: N24200A.S2P ! BIAS CONDITIONS: $V_{DS}=2V$ , $I_{DS}=10mA$													
! NOTE: NOISE PARAMETERS FOR 28 & 30 GHZ ARE EXTRAPOLATED, NOT MEASURED.													
#	S <sub>11</sub>		S <sub>21</sub>		S <sub>12</sub>		S <sub>22</sub>		F <sub>min</sub>	Γ <sub>opt</sub>		R <sub>N</sub> /50	
#GHZ	mag.	ang.	mag.	ang.	mag.	ang.	mag.	ang.	mag.	ang.	mag.	ang.	
Z													
0.1	.999	-1	5.04	179	.002	89	.62	-1					
0.2	.999	-3	5.02	178	.004	89	.62	-1					
0.5	.999	-6	4.97	175	.008	87	.62	-4					
1.0	.997	-12	4.88	170	.016	84	.62	-8	0.30	0.81	10	0.39	
2.0	.990	-23	4.70	161	.030	77	.61	-15	0.31	0.79	17	0.36	
3.0	.980	-34	4.54	152	.042	71	.61	-22					
4.0	.970	-44	4.38	144	.052	65	.61	-29	0.33	0.75	31	0.33	
5.0	.950	-53	4.22	136	.062	59	.60	-36					
6.0	.930	-62	4.08	128	.071	53	.59	-41	0.38	0.72	45	0.30	
7.0	.910	-71	3.93	120	.079	48	.59	-46					
8.0	.890	-79	3.80	113	.086	43	.58	-51	0.43	0.70	59	0.27	
9.0	.870	-87	3.67	106	.092	38	.57	-56					
10.0	.860	-94	3.54	99	.099	34	.56	-61	0.50	0.68	77	0.24	
11.0	.840	-102	3.42	92	.104	30	.55	-65					
12.0	.820	-108	3.30	86	.109	27	.54	-70	0.60	0.66	92	0.22	
13.0	.800	-115	3.19	80	.114	24	.53	-74					
14.0	.790	-121	3.08	74	.119	21	.51	-78	0.71	0.64	108	0.19	
15.0	.770	-128	2.97	68	.123	18	.50	-83					
16.0	.750	-134	2.87	63	.127	16	.49	-87	0.85	0.62	126	0.18	
17.0	.740	-139	2.77	57	.131	14	.48	-91					
18.0	.720	-145	2.68	52	.135	12	.47	-95	1.00	0.58	140	0.15	
19.0	.710	-150	2.59	47	.138	10	.46	-98					
20.0	.690	-155	2.50	42	.142	8	.45	-102	1.20	0.55	153	0.13	
22.0	.660	-165	2.32	32	.148	6	.43	-109	1.50	0.52	164	0.11	
24.0	.640	-175	2.16	23	.153	4	.42	-116	1.80	0.49	175	0.10	
26.0	.610	177	2.01	15	.159	3	.41	-122	2.10	0.48	-176	0.08	
28.0	.590	168	1.87	7	.163	1	.41	-128	2.40	0.46	-168	0.07	
30.0	.570	160	1.73	-1	.168	.001	.41	-134	2.80	0.46	-160	0.05	

Tablo 4.2 N24200A İçin Ölçülen Değerler ve YSA Eşdeğerinden Elde Edilen Nümerik Neticeler

HEDEF UZAYI												
f[GHz]	S <sub>11</sub>		S <sub>21</sub>		S <sub>12</sub>		S <sub>22</sub>		F <sub>min</sub>	Γ <sub>opt</sub>	R <sub>N</sub> /50	
1.0	0.997	-12	4.880	170	0.016	84	0.620	-8	0.300	0.810	10	0.390
2.0	0.990	-23	4.700	161	0.030	77	0.610	-15	0.310	0.790	17	0.360
4.0	0.970	-44	4.380	144	0.052	65	0.610	-29	0.330	0.750	31	0.330
6.0	0.930	-62	4.080	128	0.071	53	0.590	-41	0.380	0.720	45	0.300
8.0	0.890	-79	3.800	113	0.086	43	0.580	-51	0.430	0.700	59	0.270
10.0	0.860	-94	3.540	99	0.099	34	0.560	-61	0.500	0.680	77	0.240
12.0	0.820	-108	3.300	86	0.109	27	0.540	-70	0.600	0.660	92	0.220
14.0	0.790	-121	3.080	74	0.119	21	0.510	-78	0.710	0.640	108	0.190
16.0	0.750	-134	2.870	63	0.127	16	0.490	-87	0.850	0.620	126	0.180
18.0	0.720	-145	2.680	52	0.135	12	0.470	-95	1.000	0.580	140	0.150
20.0	0.690	-155	2.500	42	0.142	8	0.450	-102	1.200	0.550	153	0.130
22.0	0.660	-165	2.320	32	0.148	6	0.430	-109	1.500	0.520	164	0.110
24.0	0.640	-175	2.160	23	0.153	4	0.420	-116	1.800	0.490	175	0.100
26.0	0.610	177	2.010	15	0.159	3	0.410	-122	2.100	0.480	-176	0.080
28.0	0.590	168	1.870	7	0.163	1	0.410	-128	2.400	0.460	-168	0.070
30.0	0.570	160	1.730	-1	0.168	0	0.410	-134	2.800	0.460	-160	0.050

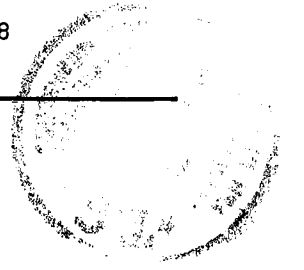
## KESTİRİLEN HEDEF UZAYI İŞARET VE GÜRÜLTÜ PARAMETRELERİ

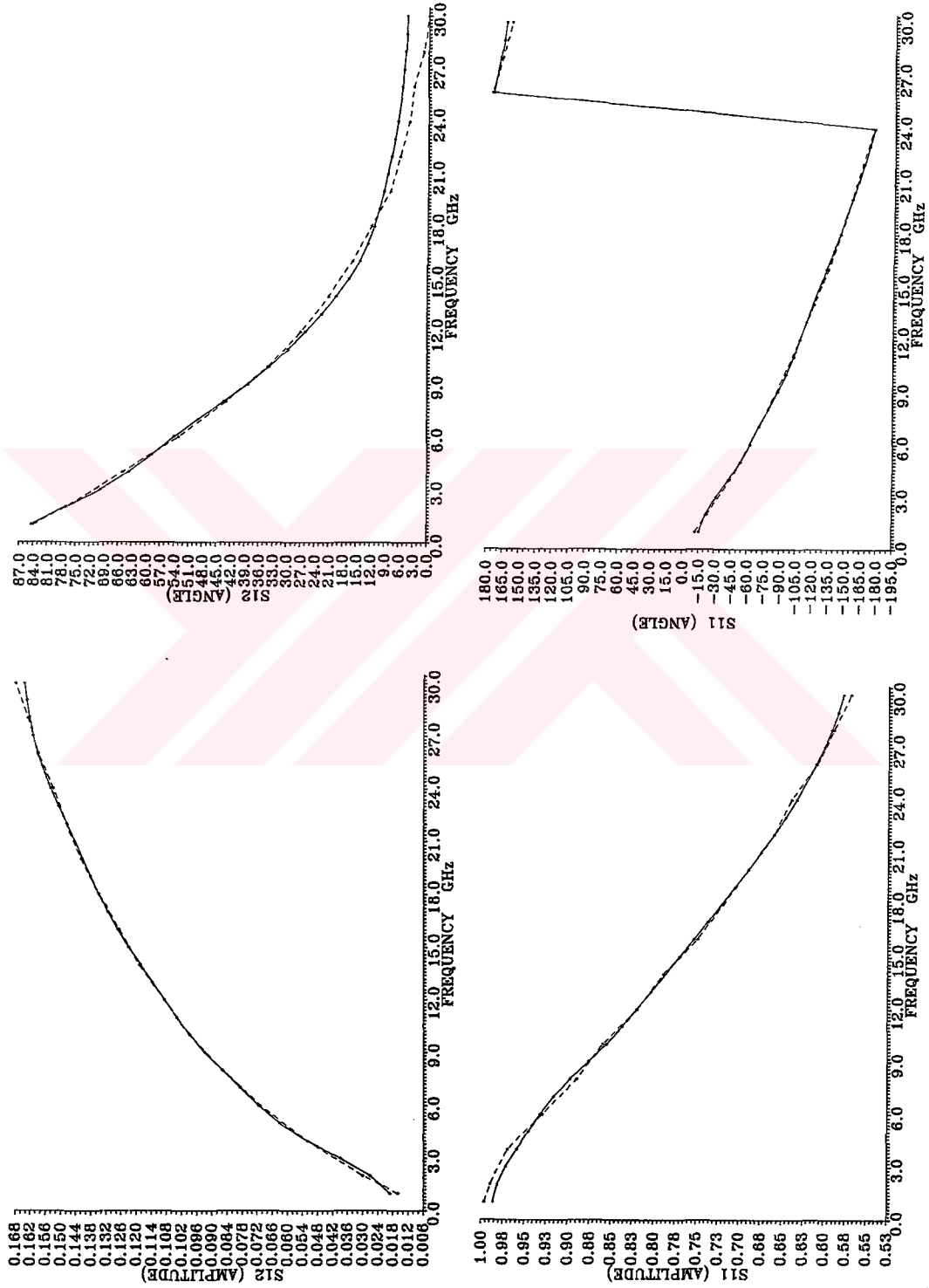
f[GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>	F <sub>min</sub>	Γ <sub>opt</sub>	R <sub>N</sub> /50					
1	0.992	-13.82	4.875	169.71	0.017	83.46	0.617	-9.13	0.297	0.812	10.91	0.386
2	0.985	-22.41	4.707	161.41	0.028	77.71	0.614	-14.91	0.305	0.786	16.51	0.365
4	0.959	-44.37	4.382	144.51	0.053	65.51	0.606	-29.67	0.333	0.745	30.97	0.328
6	0.931	-60.96	4.080	127.82	0.069	53.15	0.593	-40.07	0.380	0.724	44.33	0.297
8	0.895	-78.93	3.781	111.54	0.086	42.05	0.577	-51.16	0.441	0.703	60.65	0.267
10	0.857	-94.85	3.533	98.41	0.099	33.60	0.560	-61.16	0.508	0.683	76.69	0.242
12	0.822	-108.12	3.314	86.94	0.109	26.60	0.539	-69.75	0.590	0.661	91.84	0.219
14	0.788	-120.82	3.092	75.32	0.118	20.35	0.514	-78.23	0.701	0.635	107.9	0.195
16	0.752	-133.38	2.872	63.60	0.127	15.32	0.488	-86.83	0.844	0.607	124.8	0.172
18	0.719	-144.84	2.675	52.24	0.134	11.76	0.465	-94.81	1.012	0.579	140.1	0.151
20	0.689	-155.12	2.497	40.93	0.141	9.23	0.448	-102.0	1.214	0.553	153.0	0.132
22	0.662	-165.10	2.322	29.85	0.147	7.27	0.434	-109.0	1.475	0.526	164.5	0.113
24	0.635	-175.07	2.149	20.54	0.153	5.71	0.422	-116.0	1.795	0.499	175.1	0.095
26	0.610	175.99	2.000	14.31	0.158	4.59	0.412	-122.4	2.119	0.474	-175.	0.081
28	0.591	168.94	1.885	15.28	0.162	3.65	0.405	-127.6	2.391	0.455	-168.	0.070
30	0.570	159.90	1.729	-6.5	0.168	1.0	0.410	-134.0	2.801	0.460	-160.	0.05

## Hata Analizi

$$E_{11}=0.011053 \quad E_{21}=0.024918 \quad E_{12}=0.027162 \quad E_{22}=0.009100 \quad E_t=0.018058$$

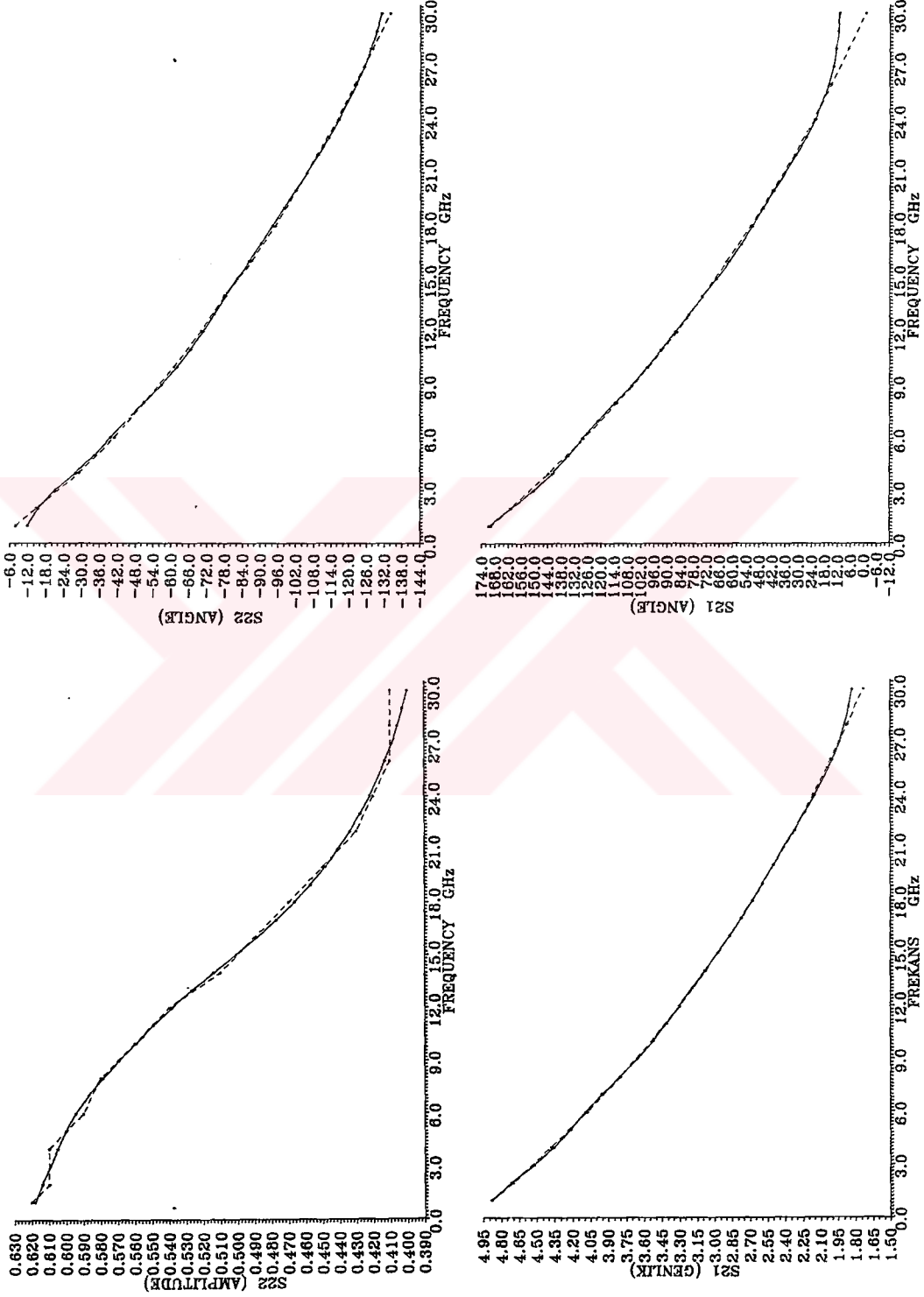
$$F_1=0.011198 \quad F_2=0.012857 \quad F_3=0.016921 \quad F_t=0.013659$$





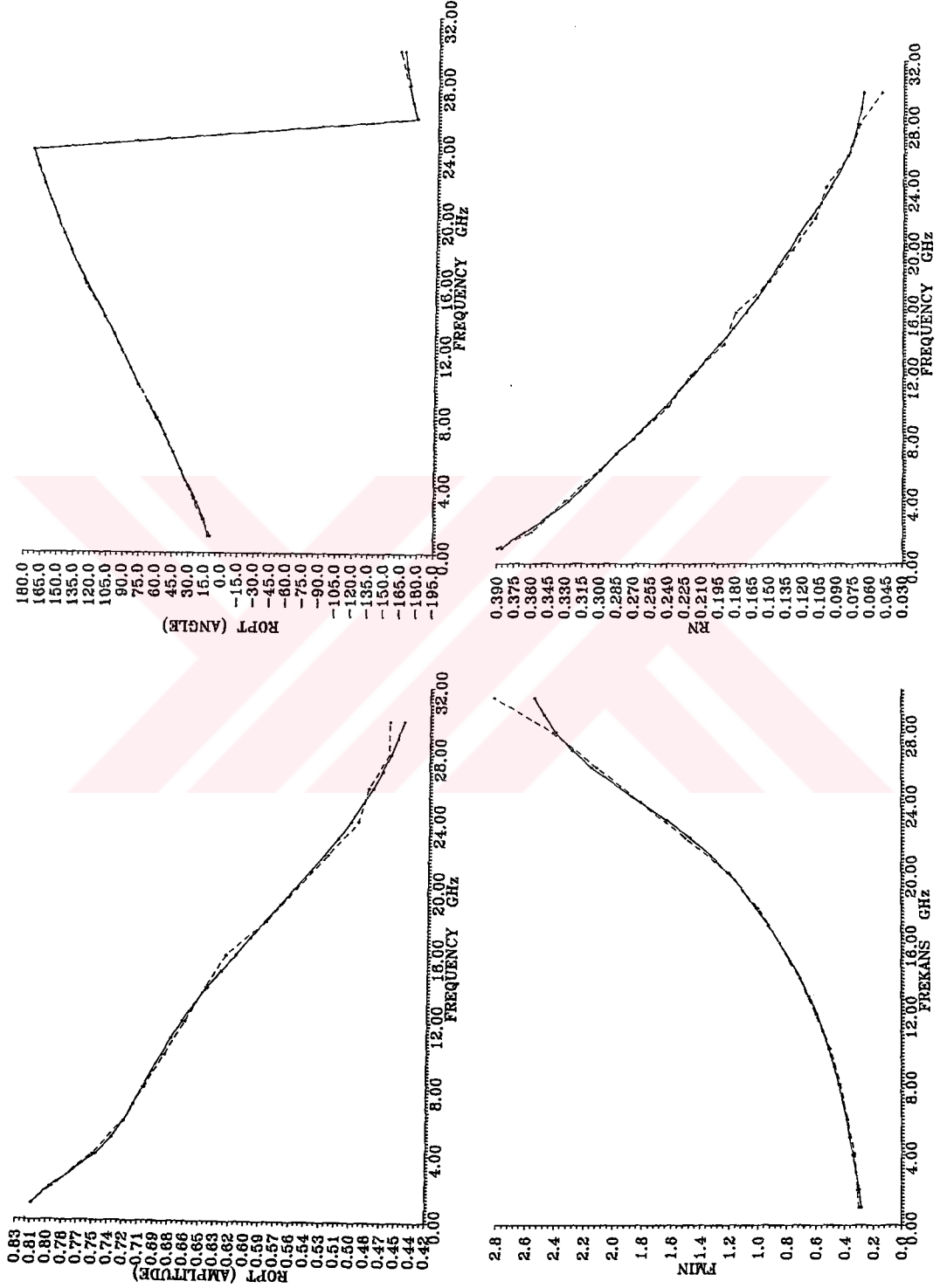
Şekil 4.1 N24200A için (S<sub>11</sub>, S<sub>12</sub>)





Şekil 4.2 N24200A için (S<sub>22</sub>, S<sub>21</sub>)





Şekil 4.3 N24200A için ( $F_{min}, R_{N_2}, \Gamma_{opt}$ )



#### 4.2 NE21900 İçin Üretici Firma Verileri ve Elde Edilen Simülasyon Neticeleri

Bu bölümde NE21900 için muhtelif kutuplama noktaları ve konfigürasyon tipleri için işaret parametreleri eşdeğer YSA ile modellenmiştir. Band genişliği 2-6 GHz'dir.

Tablo 4.3 NE21900 İçin Çoklu Kutuplama Noktası ve Konfigürasyon Tipi İçin Ölçülen Değerler

$V_{CE}=8$ $I_C=10mA$ CE Configuration					$V_{CE}=8$ $I_C=30mA$ CE Configuration								
f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>	f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>				
2	.70	-155	2.78	89 .08	72 .32	-44	2	.69	-159	3.03	88 .07	81 .26	-46
2.5	.68	-167	2.31	78 .09	72 .27	-45	2.5	.67	-170	2.51	78 .09	79 .21	-44
3	.72	-173	1.99	70 .11	74 .24	-42	3	.72	-175	2.15	70 .11	79 .21	-39
3.5	.74	-174	1.77	67 .12	77 .20	-58	3.5	.74	-176	1.90	67 .13	81 .14	-58
4	.70	176	1.57	55 .14	71 .13	-80	4	.69	175	1.68	56 .15	74 .08	-90
4.5	.72	176	1.38	52 .15	74 .15	-101	4.5	.73	174	1.46	53 .16	77 .11	-115
5	.76	172	1.26	46 .17	73 .19	-128	5	.76	172	1.34	47 .18	74 .16	-143
5.5	.72	168	1.13	41 .19	71 .26	-143	5.5	.72	167	1.21	42 .19	73 .23	-154
6	.73	168	.99	38 .19	72 .33	-148	6	.73	167	1.07	40 .20	74 .30	-155
$V_{CE}=8$ $I_C=30mA$ CE Configuration					$V_{CE}=8$ $I_C=10mA$ CC Configuration								
f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>	f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>				
2	.69	-161	3.07	87 .07	84 .25	-45	2	.88	-56	1.57	-29 .44	47 .64	143
2.5	.67	-171	2.53	77 .09	82 .20	-42	2.5	.87	-67	1.50	-34 .52	41 .60	139
3	.72	-176	2.16	70 .11	82 .18	-36	3	.83	-77	1.39	-42 .58	31 .54	128
3.5	.74	-177	1.92	67 .13	83 .13	-55	3.5	.82	-85	1.26	-44 .60	27 .46	131
4	.70	174	1.68	56 .15	75 .07	-86	4	.80	-93	1.18	-49 .64	21 .43	124
4.5	.73	174	1.49	52 .16	78 .09	-115	4.5	.79	-98	1.12	-52 .67	14 .38	125
5	.77	171	1.36	47 .18	75 .15	-144	5	.79	-103	1.02	-53 .68	11 .32	130
5.5	.73	167	1.22	41 .19	74 .22	-154	5.5	.77	-109	1.00	-55 .72	7 .31	126
6	.73	166	1.08	40 .20	74 .29	-155	6	.78	-113	0.93	-57 .72	4 .25	142
$V_{CE}=8$ $I_C=20mA$ CC Configuration					$V_{CE}=8$ $I_C=30mA$ CC Configuration								
f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>	f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>				
2	.91	-53	1.66	-29 .40	51 .75	147	2	.91	-52	1.67	-29 .39	52 .74	148
2.5	.90	-64	1.59	-34 .47	45 .68	143	2.5	.91	-64	1.61	-34 .47	46 .70	144
3	.87	-74	1.47	-42 .53	35 .62	132	3	.87	-74	1.49	-42 .52	36 .64	133
3.5	.86	-82	1.35	-43 .55	32 .54	135	3.5	.87	-82	1.36	-44 .55	33 .57	136
4	.84	-90	1.26	-49 .60	26 .52	128	4	.84	-90	1.28	-50 .59	27 .54	129
4.5	.84	-95	1.20	-52 .63	19 .47	129	4.5	.85	-95	1.22	-53 .62	20 .50	129
5	.83	-101	1.09	-54 .64	16 .41	131	5	.84	-100	1.10	-55 .64	17 .43	131
5.5	.82	-107	1.07	-56 .64	12 .40	127	5.5	.83	-107	1.08	-57 .69	13 .42	127
6	.82	-111	0.99	-59 .69	9 .34	138	6	.84	-111	1.00	-59 .69	11 .36	137





Tablo 4.4 Çoklu Kutuplama Noktası ve Konfigürasyon Tipindeki NE21900 için Hata Analizi Sonuçları

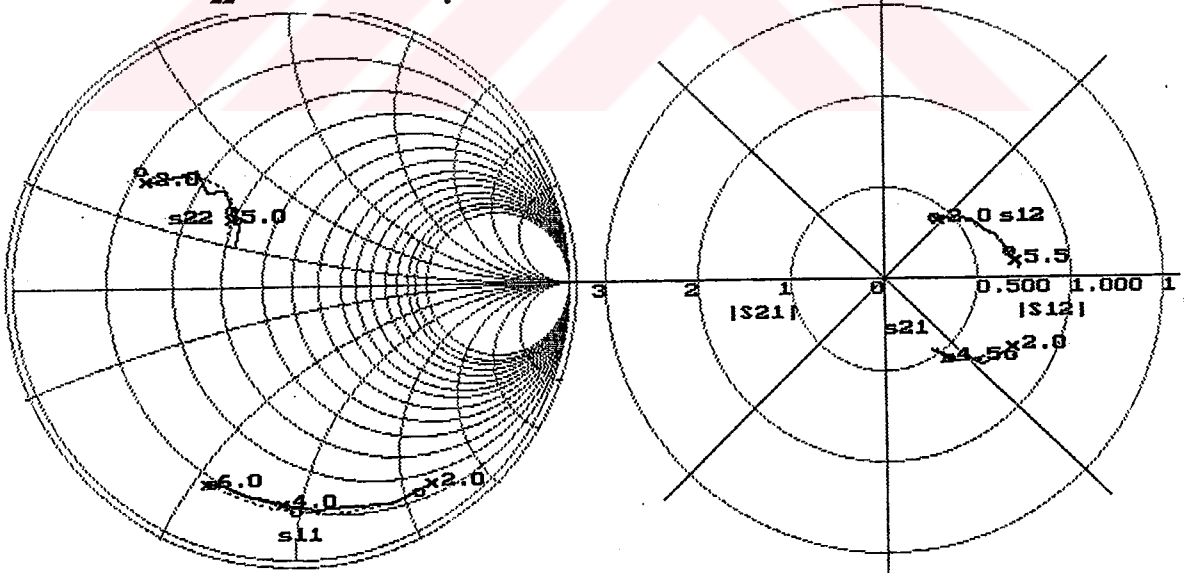
Operation Frequency Bandwidth: 2 GHz - 6 GHz				
Operations Bias Conditions				
Configuration	Bias Voltage ( $V_{CE}$ )[V]	Bias Currents ( $I_C$ ) [mA]		
		10	20	30
CC	8	SFG	WFBG	SFG
		$E_t=0.050105$	$E_t=0.044207$	$E_t=0.032068$
		$E_{11}=0.027857$	$E_{11}=0.019730$	$E_{11}=0.016128$
		$E_{21}=0.037930$	$E_{21}=0.035988$	$E_{21}=0.036308$
		$E_{12}=0.037051$	$E_{12}=0.049394$	$E_{12}=0.029213$
		$E_{22}=0.097582$	$E_{22}=0.071714$	$E_{22}=0.046625$
CE	8	Şekil 4.4	Şekil 4.5	Şekil 4.6
		$E_t=0.073243$	$E_t=0.065140$	$E_t=0.058195$
		$E_{11}=0.045472$	$E_{11}=0.039773$	$E_{11}=0.028880$
		$E_{21}=0.034876$	$E_{21}=0.044730$	$E_{21}=0.032119$
		$E_{12}=0.061781$	$E_{12}=0.060726$	$E_{12}=0.042287$
		$E_{22}=0.150843$	$E_{22}=0.115331$	$E_{22}=0.0129495$
		Şekil 4.7	Şekil 4.8	Şekil 4.9

Bias Condition:  $V_{CE}=8.0$  [Volt]  $I_C=10.0$  [mA]

COMMON COLLECTOR CONFIG.

$E_{11}=0.027857$   $E_{21}=0.037930$   $E_{12}=0.037051$

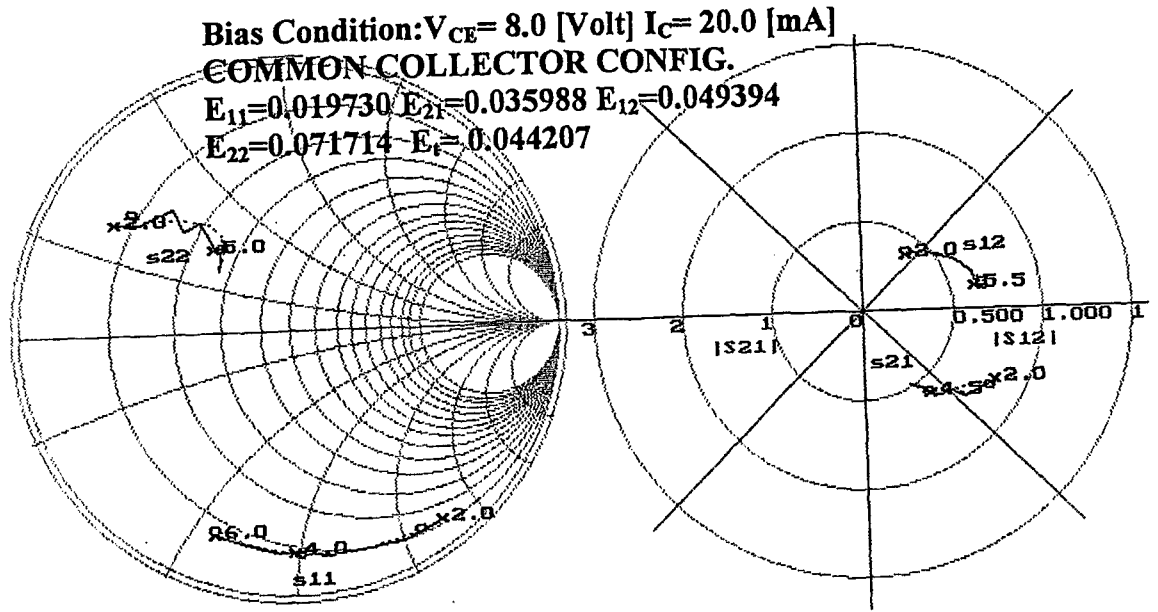
$E_{22}=0.097582$   $E_t=0.050105$



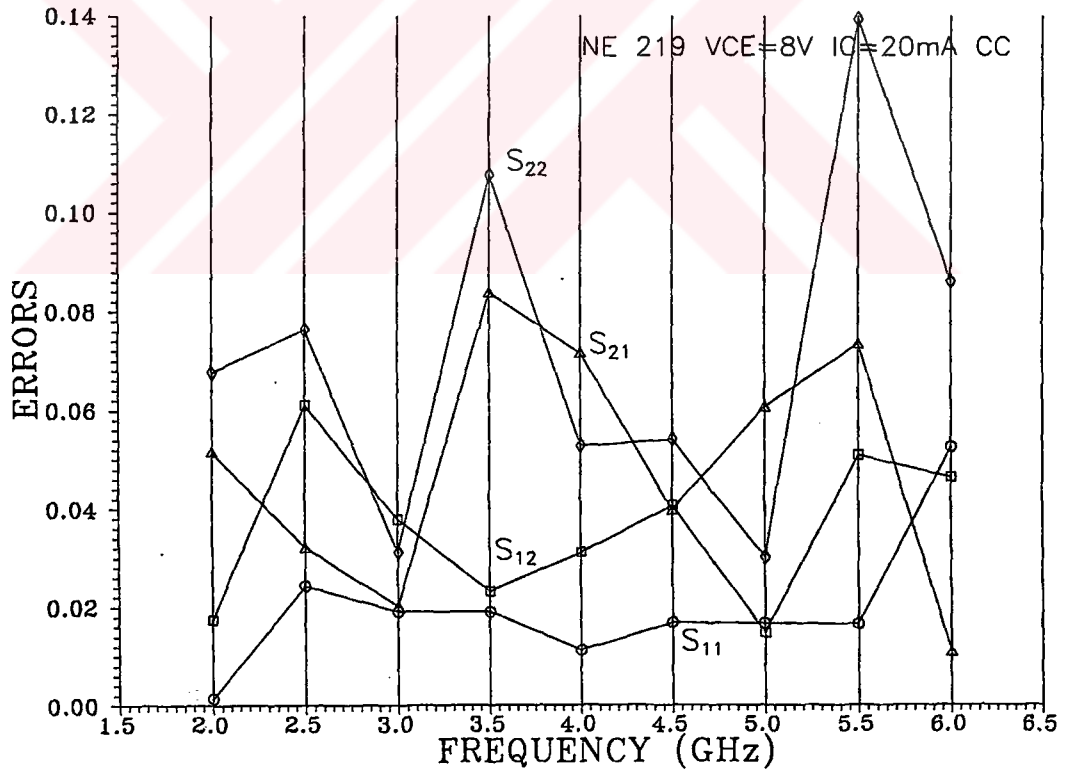
Ölçülen değer x — x hesaplanan değer 0 - - - - 0

Şekil 4.4 Ortak Kollektörlü konfigürasyonda,  $V_{CE}=8.0$  V  $I_C=10$  mA kutuplama noktasında ölçülen ve hesaplanan gürültü ve sinyal parametrelerinin frekansla değişimi.

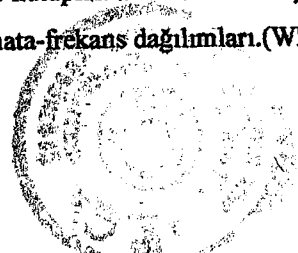


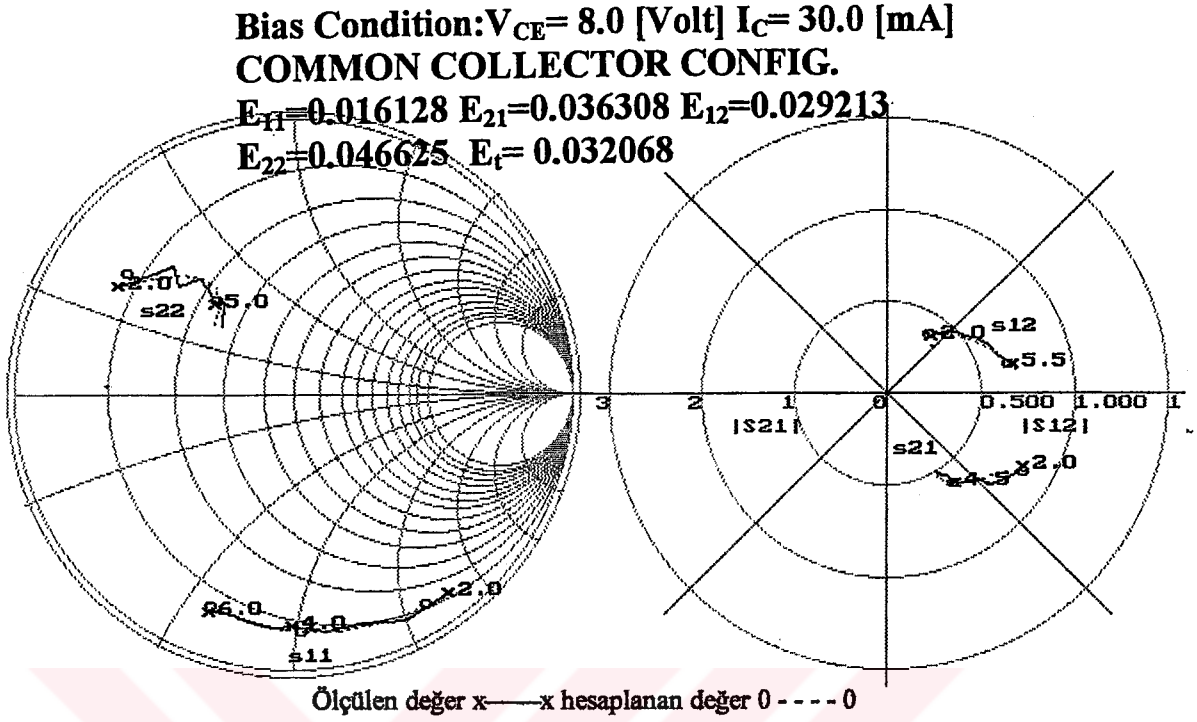


Ölçülen değer x — x hesaplanan değer o ---- o

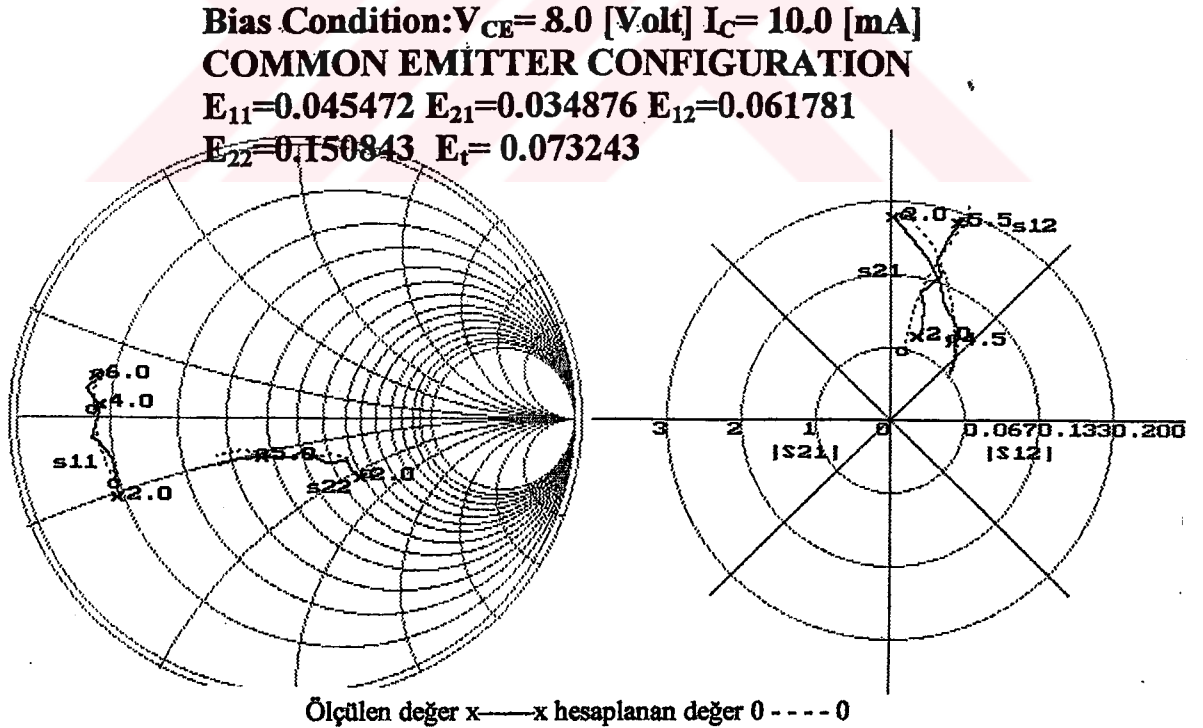


Şekil 4.5 Ortak Kollektörlü konfigürasyonda,  $V_{CE} = 8.0$  V  $I_C = 20$  mA kutuplama noktasında ölçülen ve hesaplanan gürültü ve sinyal parametrelerinin frekansla değişimi ve hata-frekans dağılımları.(WFBG)





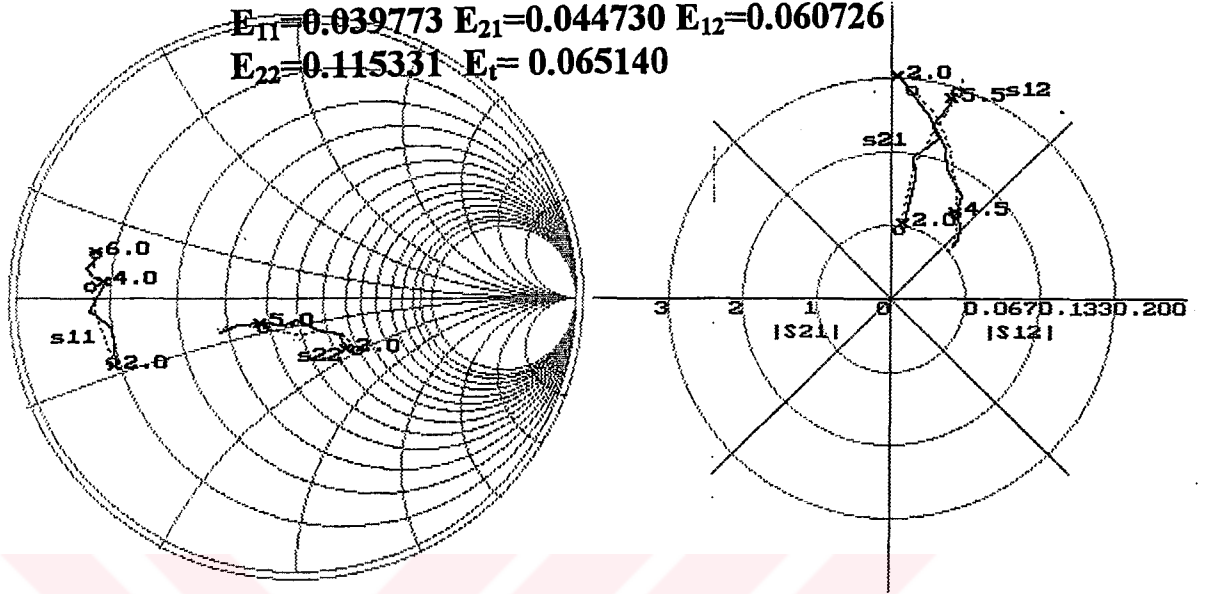
Şekil 4.6 Ortak Kollektörlü konfigürasyonda,  $V_{CE} = 8.0$  V  $I_C = 30$  mA kutuplama noktasında ölçülen ve hesaplanan gürültü ve sinyal parametrelerinin frekansla değişimi.



Şekil 4.7 Ortak emetörlü konfigürasyonda,  $V_{CE} = 8.0$  V  $I_C = 10$  mA kutuplama noktasında ölçülen ve hesaplanan gürültü ve sinyal parametrelerinin frekansla değişimi.

**Bias Condition:  $V_{CE}= 8.0$  [Volt]  $I_C= 20.0$  [mA]**  
**COMMON EMITTER CONFIGURATION**

$E_{11}=0.039773$   $E_{21}=0.044730$   $E_{12}=0.060726$   
 $E_{22}=0.115331$   $E_t= 0.065140$

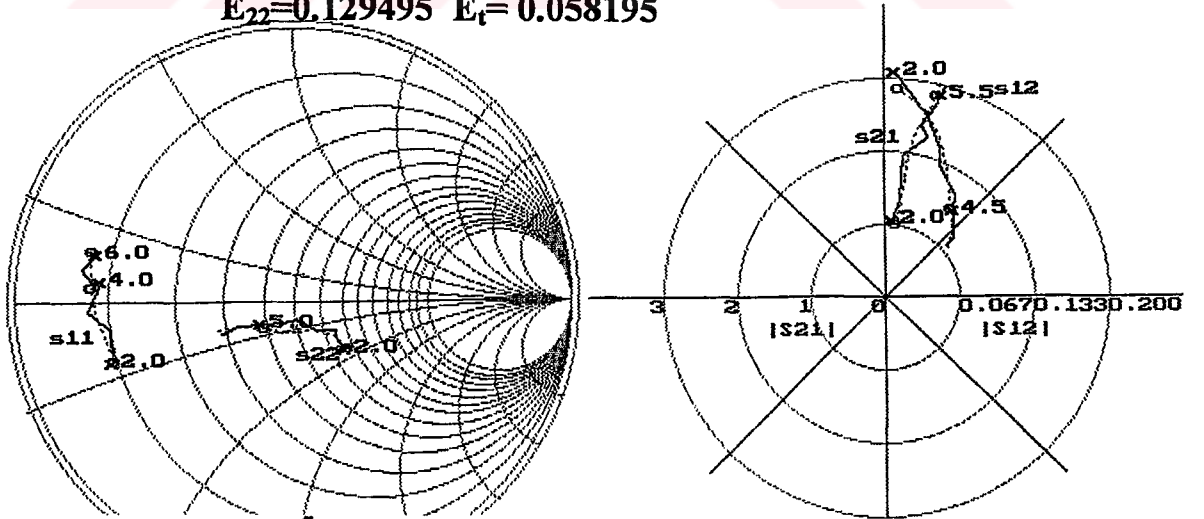


Ölçülen değer x — x hesaplanan değer 0 - - - - 0

Şekil 4.8 Ortak emetörlü konfigürasyonda,  $V_{CE}=8.0$  V  $I_C=20$  mA kutuplama noktasında ölçülen ve hesaplanan gürültü ve sinyal parametrelerinin frekansla değişimi.(WFBG)

**Bias Condition:  $V_{CE}= 8.0$  [Volt]  $I_C= 30.0$  [mA]**  
**COMMON EMITTER CONFIGURATION**

$E_{11}=0.028880$   $E_{21}=0.032119$   $E_{12}=0.042287$   
 $E_{22}=0.129495$   $E_t= 0.058195$



Ölçülen değer x — x hesaplanan değer 0 - - - - 0

Şekil 4.9 Ortak emetörlü konfigürasyonda,  $V_{CE}=8.0$  V  $I_C=30$  mA kutuplama noktasında ölçülen ve hesaplanan gürültü ve sinyal parametrelerinin frekansla değişimi.

### 4.3 N72000A ve NE32684A Transistörü Üzerinde Geliştirilen Çeşitli Tipteki Eşdeğer YSA Modellerinin İncelenmesi

Bu bölümde N72000A ve NE32684A üzerinde hem işaret hem de gürültü parametreleri tek bir kutuplama noktası civarında eşdeğer YSA ile modellenmiştir. N72000A için band genişliği 0.1-18 GHz ve NE32684A için ise band genişliği 0.1-20GHz'dir. Özellikle bu bölümde kullanılan transistörler üzerinde optimum örnekleme sayısı tesbit edilmesinin gerekliliği grafiklerle ispatlanmıştır. (Şekil 4.12)

Tablo 4.5 N72000A FET için verilen üretici firma sinyal ve gürültü parametreleri değerleri

! FILE NAME: N72000A.S2P BIAS CONDITIONS: $V_{DS}=3V$ , $I_{DS}=10mA$													
#	SIGNAL PARAMETERS								NOISE PARAMETERS (3/15/91)				
#	$S_{11}$	$S_{21}$	$S_{12}$	$S_{22}$	$F_{min}$	$\Gamma_{opt}$	$R_N/50$						
#GHZ	mag.	ang.	mag.	ang.	mag.	ang.	mag.	ang.	mag.	ang.	mag.	ang.	mag.
Z-----													
0.1	.999	-2	2.943	178	.004	84	.735	-1					
0.2	.999	-4	2.964	176	.008	86	.736	-2					
0.5	.993	-10	2.977	171	.021	83	.736	-6					
1.0	.982	-21	2.935	162	.042	77	.728	-12	0.55	.87	15	.65	
1.5	.979	-31	2.971	155	.060	71	.717	-17					
2.0	.961	-41	2.907	148	.077	65	.703	-23	0.60	.78	36	.50	
3.0	.923	-59	2.717	133	.107	55	.673	-33	0.87	.71	53	.48	
4.0	.881	-75	2.505	119	.130	45	.640	-42	1.00	.66	64	.45	
5.0	.839	-90	2.292	107	.147	36	.606	-49	1.20	.62	72	.43	
6.0	.802	-102	2.089	96	.159	31	.587	-56	1.35	.59	83	.39	
7.0	.776	-113	1.932	87	.167	24	.570	-62					
8.0	.744	-124	1.793	77	.171	18	.555	-69	1.70	.52	105	.38	
9.0	.722	-132	1.669	70	.171	13	.545	-74					
10.0	.711	-141	1.579	62	.174	10	.545	-79	2.15	.45	128	.32	
11.0	.691	-149	1.493	54	.176	5	.544	-85					
12.0	.677	-157	1.411	46	.171	1	.545	-89	2.50	.41	147	.27	
13.0	.671	-165	1.333	38	.167	-2	.550	-94					
14.0	.664	-172	1.263	32	.163	-4	.554	-98	2.95	.38	166	.32	
15.0	.657	-179	1.212	26	.159	-5	.564	-100					
16.0	.651	175	1.154	21	.159	-5	.568	-104					
17.0	.648	169	1.108	15	.152	-5	.576	-107					
18.0	.642	165	1.063	10	.152	-4	.572	-111					

Tablo 4.6 N72000a transistörünün MLP ile simülasyonundan elde edilen toplu neticeler.

Freq. GHz	Calculated S Parameters				& S <sub>22</sub>	Noise Parameters						
	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>		F <sub>min</sub>	Γ <sub>opt</sub>	R <sub>n</sub> /50				
1	0.984	0.942	0.294	0.450	0.209	0.214	0.727	0.966	0.107	0.871	0.044	0.647
2	0.958	0.887	0.290	0.412	0.387	0.181	0.705	0.936	0.130	0.779	0.101	0.511
3	0.924	0.836	0.273	0.369	0.532	0.153	0.672	0.908	0.164	0.711	0.145	0.469
4	0.882	0.790	0.250	0.330	0.651	0.124	0.638	0.884	0.203	0.660	0.177	0.449
5	0.838	0.751	0.229	0.298	0.735	0.101	0.608	0.864	0.240	0.620	0.201	0.434
Freq. GHz	Desired S Parameters				& S <sub>22</sub>	Noise Parameters						
	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>		F <sub>min</sub>	Γ <sub>opt</sub>	R <sub>n</sub> /50				
1	0.982	0.942	0.294	0.450	0.210	0.214	0.728	0.967	0.110	0.870	0.042	0.650
2	0.961	0.886	0.291	0.411	0.385	0.181	0.703	0.936	0.120	0.780	0.100	0.500
3	0.923	0.836	0.272	0.369	0.535	0.153	0.673	0.908	0.174	0.710	0.147	0.480
4	0.881	0.792	0.251	0.331	0.650	0.125	0.640	0.883	0.200	0.660	0.178	0.450
5	0.839	0.750	0.229	0.297	0.735	0.100	0.606	0.864	0.240	0.620	0.200	0.430

iteration= 200000 eps= 0.002416 error\_1= 0.0000023 error\_t= 0.0000023

Freq. GHz	Calculated S Parameters				& S <sub>22</sub>	Desired S Parameters										
	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>		S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>							
1	0.982	0.939	0.294	0.450	0.212	0.214	0.728	0.964	0.982	0.942	0.294	0.450	0.210	0.214	0.728	0.967
2	0.959	0.889	0.289	0.411	0.381	0.182	0.704	0.937	0.961	0.886	0.291	0.411	0.385	0.181	0.703	0.936
3	0.923	0.836	0.273	0.370	0.537	0.150	0.673	0.908	0.923	0.836	0.272	0.369	0.535	0.153	0.673	0.908
4	0.882	0.790	0.251	0.330	0.652	0.124	0.639	0.884	0.881	0.792	0.251	0.331	0.650	0.125	0.640	0.883
5	0.839	0.751	0.229	0.297	0.733	0.103	0.607	0.864	0.839	0.750	0.229	0.297	0.735	0.100	0.606	0.864

iteration= 200000 eps= 0.001248 error\_1= 0.000006 error\_t= 0.000006

Freq. GHz	Calculated F Parameters			Desired F Parameters		
	F <sub>min</sub>	Γ <sub>opt</sub>	R <sub>n</sub> /50	F <sub>min</sub>	Γ <sub>opt</sub>	R <sub>n</sub> /50

1	0.107	0.871	0.043	0.648	0.110	0.870	0.042	0.650
2	0.130	0.778	0.102	0.511	0.120	0.780	0.100	0.500
3	0.164	0.712	0.144	0.468	0.174	0.710	0.147	0.480
4	0.204	0.659	0.177	0.448	0.200	0.660	0.178	0.450
5	0.239	0.621	0.201	0.436	0.240	0.620	0.200	0.430

iteration= 250000 eps= 0.004254 error\_1= 0.000072 error\_t= 0.000072

learn & test finished at time alpha =0.800 epsilon =0.50 Bias Point=3.317

n72000a Transistoruna ait üç farklı tipteki YSA yapısı için eğitim sonuçları yukarıda verilmiştir.

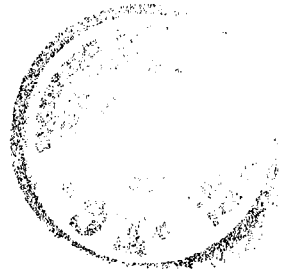
Not:Hesaplanan ve arzulanan İşaret ve Gürültü parametrelerinin normalize edilmiş olduğuna dikkat ediniz.

Tablo 4.7 250000 adet iterasyon için N72000A transistörüne ait 1,2,3,4 ve 5 GHz'de alınan örneklerle teşkil edilen hedef uzayıyla kestirilen değer uzayı arasındaki farklardan hata kriterleri yardımıyla elde edilen neticeler.

	Şekil 3.4'deki Yapı için	Şekil-3.5'deki Yapı için	Şekil-3.3'deki Yapı için
Gerçek Mutlak Toplam Hata	0.006819	0.006961	0.016462
E11	0.010758	0.004621	-
E12	0.012340	0.006393	-
E21	0.003366	0.003825	-
E22	0.004638	0.002851	-
ET	0.007776	0.004422	-
F1	-	0.036043	0.037645
F2	-	0.007320	0.009009
F3	-	0.010846	0.012717
FT	-	0.018070	0.019790

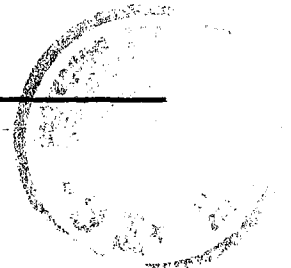
Tablo 4.8 N32684AA FET için verilen üretici firma sinyal ve gürültü parametreleri değerleri

```
! FILENAME: N32684AA.S2P VERSION:5.0
! NEC PART NUMBER: NE32684A DATE:04/92
! BIAS CONDITION: VDS=2.0V, IDS=10.0mA, VGS= -0.400V
! NOTE: DATA AT 20GHZ IS EXTRAPOLATED; NOT MEASURED.
# NOISE PARAMETERS (04/92)
# S11 S21 S12 S22 Fmin Γopt RN/50
#GHZ mag. ang. mag. ang. mag. ang. mag. ang. mag. ang.
Z-----
0.1 .999 -2.0 4.879 178.4 .002 88.8 .555 -1.9
0.2 .999 -3.6 4.872 176.7 .003 88.1 .554 -2.8
0.5 .999 -9.0 4.859 171.5 .008 85.9 .556 -6.0
1.0 .991 -17.6 4.796 162.2 .015 77.7 .552 -12.0 0.28 .90 17 .45
1.5 .976 -25.7 4.834 154.4 .023 73.6 .544 -17.6
2.0 .960 -33.9 4.750 146.0 .029 70.4 .541 -23.1 0.30 .85 32 .37
2.5 .943 -41.7 4.696 138.2 .037 64.8 .526 -28.1
3.0 .922 -49.4 4.618 130.4 .042 60.0 .520 -33.0
3.5 .898 -56.8 4.453 123.1 .047 57.2 .510 -37.5
```

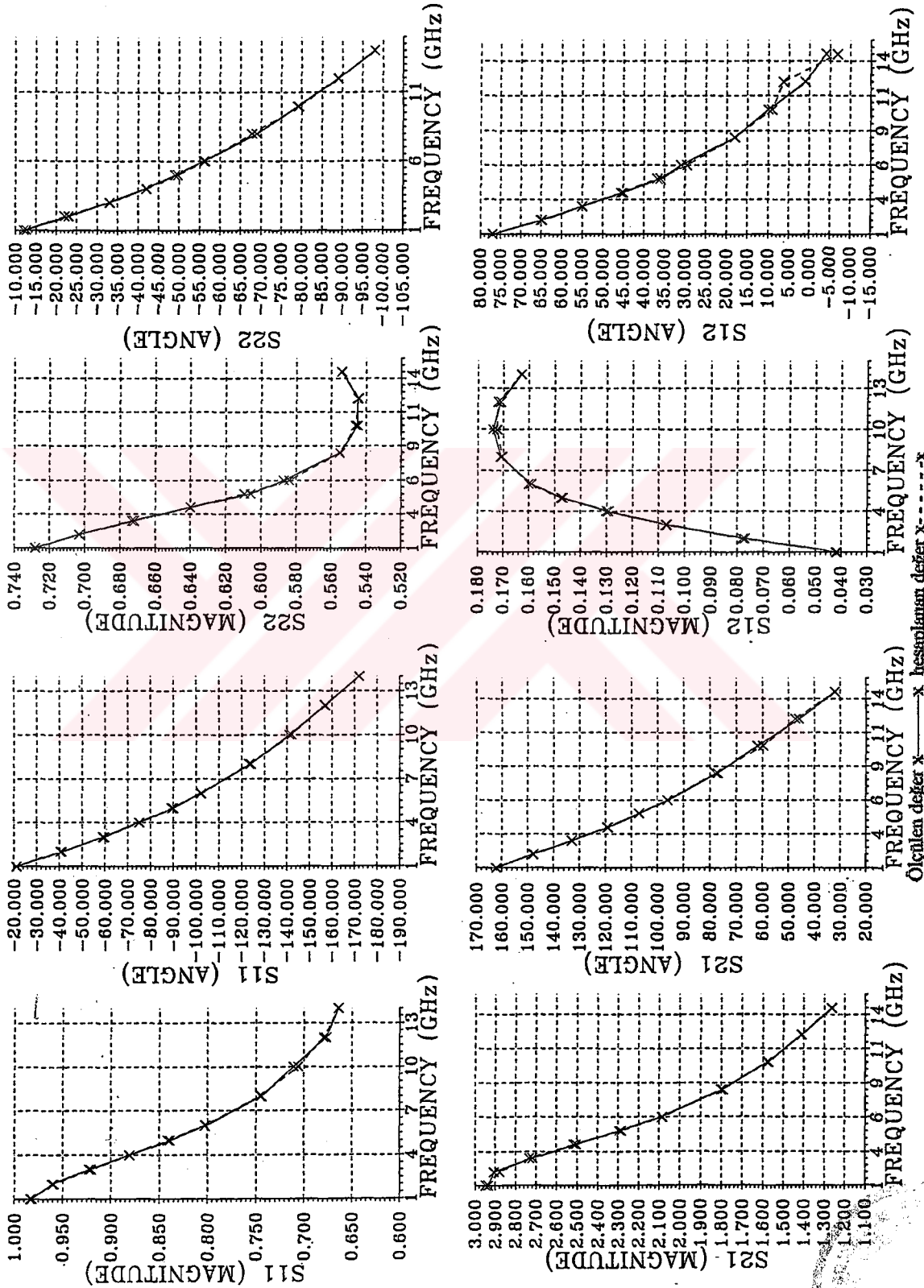


Tablo 4.8 N32684AA FET için verilen üretici firma sinyal ve gürültü parametreleri değerleri  
(Devamı)

4.0	.873	-64.5	4.370	115.8	.054	52.4	.505	-42.8	0.33	.72	64	.27
4.5	.843	-71.7	4.253	108.5	.058	49.0	.494	-47.5				
5.0	.816	-79.0	4.179	101.5	.062	45.2	.478	-52.5				
5.5	.788	-85.9	4.049	94.4	.067	41.5	.469	-57.8				
6.0	.758	-92.9	3.962	87.9	.070	38.9	.454	-62.2	0.37	.62	91	.21
6.5	.734	-99.5	3.826	81.2	.074	35.6	.447	-67.4				
7.0	.712	-106.0	3.720	74.7	.077	32.5	.437	-71.8				
7.5	.690	-111.4	3.618	69.7	.081	30.7	.430	-74.9				
8.0	.667	-117.2	3.527	63.3	.084	27.9	.425	-79.7	0.40	.54	116	.15
8.5	.648	-123.0	3.433	57.5	.087	25.3	.421	-84.4				
9.0	.629	-128.8	3.348	51.3	.090	22.7	.421	-88.7				
9.5	.607	-134.5	3.291	45.6	.092	20.6	.418	-92.4				
10.0	.592	-140.1	3.218	39.9	.095	18.5	.418	-96.8	0.45	.48	138	.10
10.5	.570	-146.1	3.153	33.9	.100	16.0	.414	-100.7				
11.0	.549	-152.4	3.104	28.3	.103	13.5	.410	-105.2				
11.5	.530	-159.1	3.054	22.3	.106	10.8	.402	-109.2				
12.0	.513	-165.5	2.994	17.1	.110	8.1	.396	-114.2	0.50	.42	164	.07
12.5	.499	-172.2	2.948	11.5	.111	5.8	.389	-118.5				
13.0	.487	-179.2	2.901	5.8	.115	3.4	.382	-123.2				
13.5	.474	174.1	2.868	0.2	.117	-0.3	.376	-127.9				
14.0	.464	168.1	2.825	-4.5	.121	-1.2	.368	-132.4	0.62	.38	169	.07
14.5	.452	160.9	2.802	-10.6	.125	-4.6	.368	-137.4				
15.0	.443	154.0	2.763	-16.3	.130	-7.7	.369	-143.0				
15.5	.435	146.9	2.709	-22.2	.134	-11.4	.376	-148.5				
16.0	.423	139.4	2.707	-27.6	.138	-13.9	.373	-154.5	0.75	.34	139	.08
16.5	.418	131.6	2.666	-33.4	.143	-17.8	.375	-160.2				
17.0	.415	123.9	2.638	-40.1	.144	-22.2	.374	-166.2				
17.5	.412	115.6	2.626	-46.3	.150	-26.4	.373	-171.1				
18.0	.414	107.4	2.614	-52.5	.152	-30.1	.371	-176.5	0.91	.34	101	.09
18.5	.405	98.3	2.607	-59.5	.158	-35.3	.372	178.2				
19.0	.413	89.2	2.581	-65.4	.161	-38.6	.360	171.9				
19.5	.423	82.6	2.547	-72.2	.164	-44.0	.348	166.4				
20.0	.432	74.1	2.528	-78.0	.167	-49.2	.341	158.7	1.10	.38	77	.10

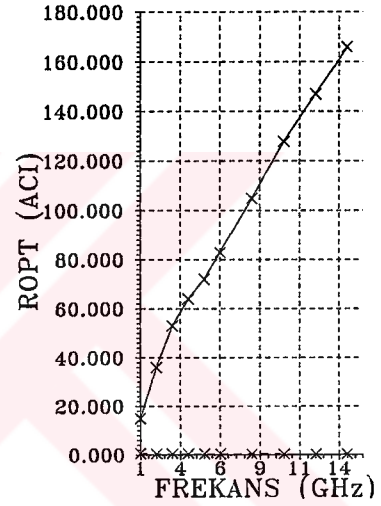
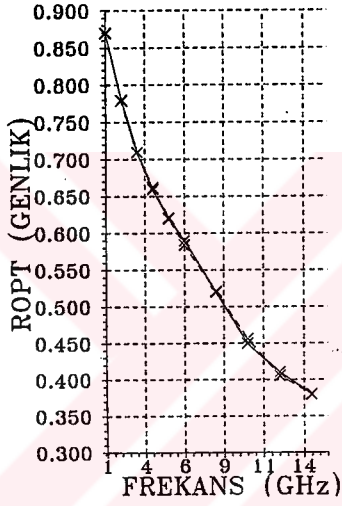
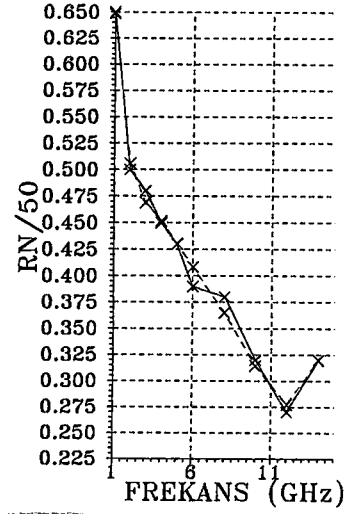
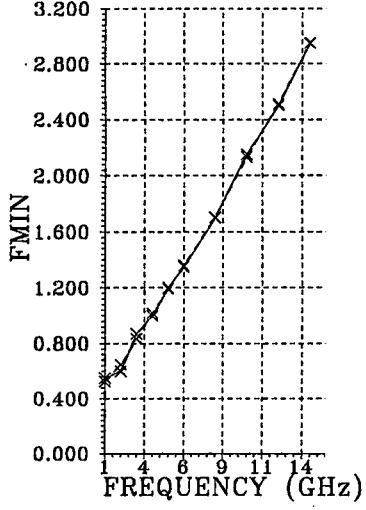






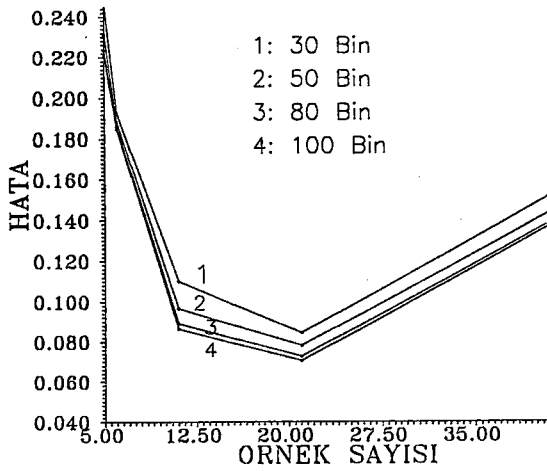
Şekil 4.10 N72000A için (S<sub>11</sub>, S<sub>12</sub>, S<sub>21</sub>, S<sub>22</sub>) ölçülen değer x → x hesaplanan değer x-----x



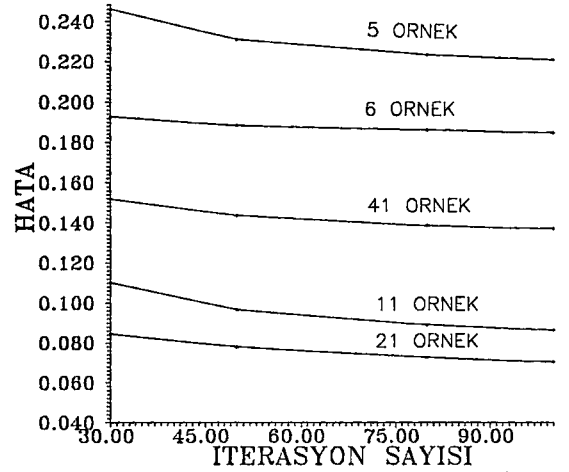


Öçülen değeri x — x hesaplanan değeri x - - - x

#### 4.11 N72000A FET'e ait 1-14 GHz bandında gürültü parametrelerinin simülasyonundan elde edilen neticelerden bir kaç örneğin grafiksel olarak gösterilmesi



4.12 Örnek sayısı değişken iterasyon sayısı parametre olmak üzere sinyal parametrelerine ait toplam bağıl hatayı göstermektedir.



4.13 İterasyon sayısı değişken hedef uzayları parametre olmak üzere sinyal parametrelerine ait toplam bağıl hatayı göstermektedir.



#### 4.4 Ortak Emetör Konfigürasyonundaki BFQ74 Transistörü Üzerinde Geliştirilen Tek Kutuplama Gerilimli ,farklı Kutuplama Akımları için Eşdeğer Sinyal-Gürültü YSA Modeli

Bu bölümde BFQ74 üzerinde muhtelif kutuplama noktaları için işaret parametreleri eşdeğer YSA ile modellenmiştir.Band genişliği 0,1-5GHz'dir.

##### 4.4.1 BFQ74 İçin Elde Edilen Simülasyon Neticeleri

Tablo 4.9 BFQ74 için farklı kutuplama noktalarındaki simülasyon neticelerinin hata analizi

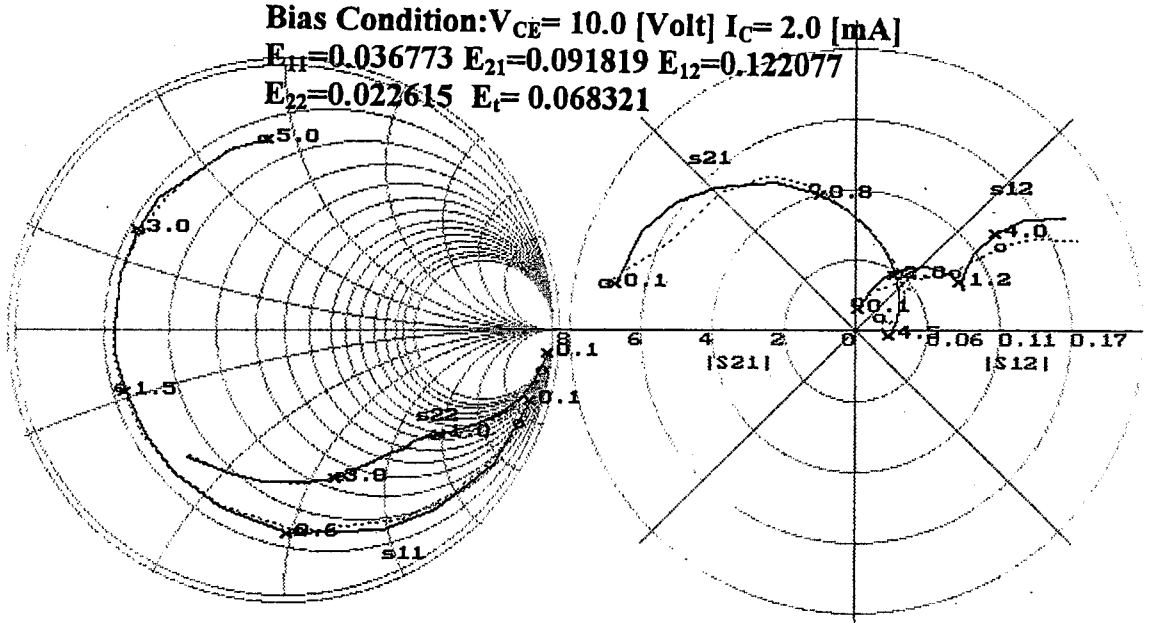
Name of the transistor: BFQ74						
Operation Frequency Bandwidth: 0.1 GHz - 5 GHz						
Operations Bias Conditions						
Bias Voltage (V <sub>CE</sub> ) [V]	Bias Currents (I <sub>c</sub> ) [mA]					
	2.0	5.0	10.0	15.0	20.0	25.0
10.0	SFG	WFBG	SFG	WFBG	SFG	WFBG
	E <sub>r</sub> =0.068321	E <sub>r</sub> =0.230778	E <sub>r</sub> =0.059663	E <sub>r</sub> =0.057487	E <sub>r</sub> =0.093752	E <sub>r</sub> =0.095066
	E <sub>11</sub> =0.03677	E <sub>11</sub> =0.21744	E <sub>11</sub> =0.04925	E <sub>11</sub> =0.04017	E <sub>11</sub> =0.05729	E <sub>11</sub> =0.06417
	E <sub>21</sub> =0.09181	E <sub>21</sub> =0.35323	E <sub>21</sub> =0.09826	E <sub>21</sub> =0.10369	E <sub>21</sub> =0.20905	E <sub>21</sub> =0.12021
	E <sub>12</sub> =0.12207	E <sub>12</sub> =0.14776	E <sub>12</sub> =0.05228	E <sub>12</sub> =0.06139	E <sub>12</sub> =0.06902	E <sub>12</sub> =0.12544
	E <sub>22</sub> =0.02261	E <sub>22</sub> =0.20467	E <sub>22</sub> =0.03884	E <sub>22</sub> =0.02468	E <sub>22</sub> =0.03962	E <sub>22</sub> =0.07042
	Şekil 4.14	Şekil 4.15	Şekil 4.16	Şekil 4.17	Şekil 4.18	Şekil 4.19

Tablo 4.10 BFQ74 Transistörü için verilen nümerik simülasyon neticelerden hareketle iterasyon sayısına bağlılığında gözlemlenmesi

Bias V <sub>CE</sub> , I <sub>c</sub>	Freq. GHz	Calculated S Parameters				Desired S Parameters										
		S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>							
1.0 0.2 0.1	0.954	0.956	0.191	0.476	0.071	0.218	0.988	0.980	0.960	0.956	0.195	0.469	0.068	0.219	0.990	0.986
1.0 0.2 0.3	0.874	0.867	0.175	0.395	0.194	0.180	0.929	0.957	0.880	0.861	0.177	0.400	0.196	0.172	0.920	0.956
1.0 0.2 0.6	0.766	0.750	0.147	0.329	0.307	0.113	0.812	0.930	0.770	0.753	0.137	0.328	0.304	0.119	0.810	0.928
1.0 0.2 1.0	0.677	0.649	0.094	0.264	0.352	0.069	0.707	0.905	0.680	0.647	0.097	0.258	0.356	0.081	0.700	0.906
1.0 0.2 1.2	0.652	0.609	0.079	0.236	0.361	0.062	0.672	0.895	0.650	0.608	0.084	0.233	0.364	0.069	0.670	0.897
1.0 0.2 1.5	0.632	0.561	0.066	0.198	0.366	0.061	0.639	0.882	0.630	0.561	0.070	0.200	0.364	0.061	0.640	0.886
1.0 0.2 1.8	0.626	0.522	0.058	0.164	0.366	0.065	0.620	0.871	0.630	0.522	0.060	0.172	0.368	0.058	0.630	0.872
1.0 0.2 2.0	0.626	0.500	0.055	0.144	0.367	0.069	0.613	0.863	0.630	0.497	0.055	0.153	0.364	0.058	0.610	0.864
1.0 0.2 3.0	0.650	0.406	0.043	0.068	0.406	0.089	0.593	0.811	0.660	0.403	0.039	0.078	0.396	0.081	0.590	0.803
1.0 0.2 4.1	0.694	0.322	0.031	0.037	0.545	0.094	0.577	0.724	0.680	0.328	0.031	0.008	0.544	0.097	0.580	0.731
1.0 0.2 4.5	0.708	0.295	0.029	0.975	0.644	0.085	0.585	0.691	0.710	0.297	0.027	0.978	0.640	0.094	0.580	0.692
1.0 0.2 5.0	0.725	0.265	0.028	1.000	0.753	0.075	0.594	0.646	0.720	0.264	0.024	0.950	0.760	0.081	0.600	0.647
1.0 1.0 0.1	0.828	0.899	0.656	0.420	0.059	0.190	0.918	0.963	0.830	0.903	0.647	0.431	0.060	0.194	0.920	0.964
1.0 1.0 0.3	0.648	0.751	0.415	0.331	0.134	0.147	0.689	0.933	0.630	0.753	0.428	0.336	0.136	0.147	0.700	0.925
1.0 1.0 0.6	0.559	0.636	0.267	0.279	0.191	0.127	0.535	0.915	0.550	0.639	0.256	0.272	0.188	0.128	0.540	0.911
1.0 1.0 1.0	0.533	0.550	0.163	0.231	0.239	0.125	0.461	0.905	0.530	0.553	0.161	0.225	0.232	0.125	0.460	0.903
1.0 1.0 1.2	0.531	0.520	0.136	0.210	0.261	0.128	0.442	0.900	0.520	0.525	0.136	0.208	0.256	0.125	0.440	0.897
1.0 1.0 1.5	0.534	0.485	0.110	0.181	0.292	0.131	0.423	0.892	0.520	0.489	0.111	0.181	0.288	0.128	0.430	0.889
1.0 1.0 1.8	0.542	0.458	0.095	0.153	0.326	0.132	0.411	0.883	0.530	0.464	0.093	0.158	0.332	0.128	0.420	0.878
1.0 1.0 2.0	0.548	0.443	0.087	0.136	0.350	0.132	0.406	0.875	0.550	0.447	0.084	0.144	0.356	0.131	0.410	0.869
1.0 1.0 3.0	0.591	0.375	0.062	0.068	0.504	0.123	0.386	0.824	0.600	0.375	0.059	0.081	0.508	0.119	0.380	0.814
1.0 1.0 4.0	0.640	0.312	0.043	0.035	0.686	0.106	0.368	0.747	0.630	0.311	0.046	0.017	0.692	0.100	0.370	0.744
1.0 1.0 4.5	0.661	0.283	0.040	0.979	0.793	0.091	0.374	0.704	0.670	0.286	0.042	0.986	0.776	0.089	0.370	0.706
1.0 1.0 5.0	0.681	0.256	0.037	1.000	0.871	0.077	0.381	0.661	0.670	0.256	0.038	0.958	0.868	0.069	0.380	0.658
1.0 2.0 0.1	0.735	0.858	0.921	0.413	0.046	0.193	0.863	0.950	0.730	0.858	0.938	0.408	0.052	0.192	0.860	0.950
1.0 2.0 0.3	0.543	0.685	0.523	0.304	0.098	0.143	0.583	0.920	0.560	0.681	0.515	0.308	0.104	0.139	0.580	0.919

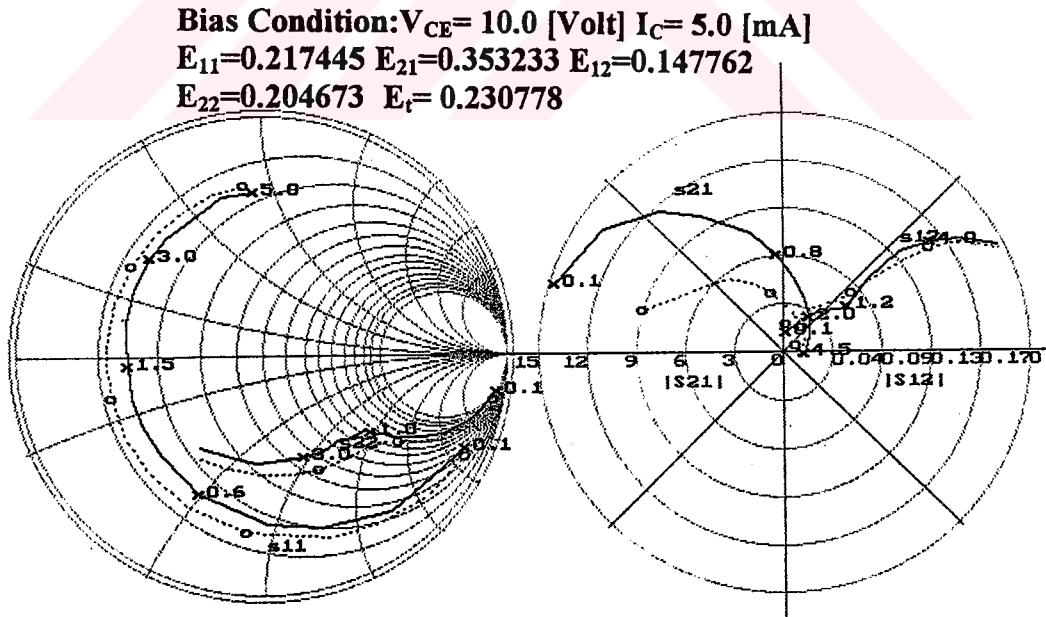
Tablo 4.10 BFQ74 Transistörü için verilen nümerik simülasyon neticelerden hareketle iterasyon sayısına bağlılığında gözlemlenmesi (Devamı)

1.0	2.0	0.6	0.497	0.582	0.273	0.257	0.149	0.141	0.463	0.910	0.510	0.581	0.280	0.253	0.144	0.142	0.460	0.917
1.0	2.0	1.0	0.503	0.516	0.165	0.218	0.206	0.149	0.419	0.904	0.510	0.517	0.171	0.211	0.200	0.150	0.410	0.914
1.0	2.0	1.2	0.509	0.493	0.139	0.200	0.233	0.151	0.406	0.900	0.510	0.489	0.143	0.197	0.228	0.153	0.400	0.908
1.0	2.0	1.5	0.519	0.466	0.114	0.173	0.274	0.151	0.393	0.893	0.510	0.464	0.115	0.172	0.272	0.156	0.390	0.900
1.0	2.0	1.8	0.530	0.444	0.099	0.148	0.316	0.148	0.385	0.883	0.530	0.442	0.097	0.153	0.320	0.153	0.390	0.886
1.0	2.0	2.0	0.539	0.431	0.091	0.132	0.346	0.145	0.380	0.876	0.550	0.428	0.087	0.139	0.348	0.150	0.380	0.878
1.0	2.0	3.0	0.585	0.369	0.064	0.067	0.522	0.127	0.361	0.824	0.590	0.367	0.061	0.078	0.520	0.133	0.360	0.822
1.0	2.0	4.0	0.635	0.308	0.044	0.034	0.710	0.107	0.343	0.745	0.630	0.306	0.048	0.014	0.712	0.108	0.350	0.750
1.0	2.0	4.5	0.657	0.279	0.041	0.982	0.812	0.091	0.349	0.703	0.670	0.281	0.043	0.986	0.796	0.094	0.340	0.708
1.0	2.0	5.0	0.677	0.253	0.038	1.000	0.885	0.077	0.356	0.659	0.680	0.253	0.039	0.958	0.896	0.075	0.360	0.661
<u>Iteration= 50000</u>			eps= 0.016451			error_1= 0.001083			error_t= 0.001067									
Bias Freq.		Calculated S Parameters								Desired S Parameters								
V <sub>CE</sub> , I <sub>C</sub>	GHz	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>									
1.0	0.2	0.1	0.956	0.956	0.193	0.477	0.071	0.219	0.990	0.981	0.960	0.956	0.195	0.469	0.068	0.219	0.990	0.986
1.0	0.2	0.3	0.872	0.867	0.176	0.394	0.194	0.178	0.929	0.956	0.880	0.861	0.177	0.400	0.196	0.172	0.920	0.956
1.0	0.2	0.6	0.766	0.750	0.142	0.328	0.309	0.118	0.808	0.929	0.770	0.753	0.137	0.328	0.304	0.119	0.810	0.928
1.0	0.2	1.0	0.681	0.649	0.093	0.264	0.352	0.071	0.708	0.905	0.680	0.647	0.097	0.258	0.356	0.081	0.700	0.906
1.0	0.2	1.2	0.654	0.609	0.079	0.236	0.361	0.063	0.673	0.896	0.650	0.608	0.084	0.233	0.364	0.069	0.670	0.897
1.0	0.2	1.5	0.631	0.561	0.067	0.198	0.365	0.060	0.639	0.882	0.630	0.561	0.070	0.200	0.364	0.061	0.640	0.886
1.0	0.2	1.8	0.624	0.522	0.059	0.165	0.366	0.063	0.620	0.870	0.630	0.522	0.060	0.172	0.368	0.058	0.630	0.872
1.0	0.2	2.0	0.624	0.499	0.056	0.144	0.366	0.066	0.613	0.862	0.630	0.497	0.055	0.153	0.364	0.058	0.610	0.864
1.0	0.2	3.0	0.650	0.406	0.043	0.068	0.405	0.088	0.596	0.811	0.660	0.403	0.039	0.078	0.396	0.081	0.590	0.803
1.0	0.2	4.1	0.695	0.322	0.030	0.037	0.544	0.095	0.578	0.724	0.680	0.328	0.031	0.008	0.544	0.097	0.580	0.731
1.0	0.2	4.5	0.709	0.295	0.029	0.975	0.643	0.087	0.584	0.691	0.710	0.297	0.027	0.978	0.640	0.094	0.580	0.692
1.0	0.2	5.0	0.725	0.265	0.028	1.000	0.754	0.077	0.592	0.647	0.720	0.264	0.024	0.950	0.760	0.081	0.600	0.647
1.0	1.0	0.1	0.831	0.899	0.652	0.419	0.060	0.192	0.922	0.964	0.830	0.903	0.647	0.431	0.060	0.194	0.920	0.964
1.0	1.0	0.3	0.644	0.751	0.418	0.331	0.131	0.144	0.688	0.931	0.630	0.753	0.428	0.336	0.136	0.147	0.700	0.925
1.0	1.0	0.6	0.558	0.637	0.268	0.280	0.188	0.125	0.534	0.914	0.550	0.639	0.256	0.272	0.188	0.128	0.540	0.911
1.0	1.0	1.0	0.533	0.552	0.164	0.232	0.238	0.124	0.462	0.904	0.530	0.553	0.161	0.225	0.232	0.125	0.460	0.903
1.0	1.0	1.2	0.531	0.521	0.137	0.210	0.261	0.126	0.442	0.900	0.520	0.525	0.136	0.208	0.256	0.125	0.440	0.897
1.0	1.0	1.5	0.534	0.486	0.111	0.180	0.293	0.129	0.423	0.892	0.520	0.489	0.111	0.181	0.288	0.128	0.430	0.889
1.0	1.0	1.8	0.541	0.459	0.095	0.153	0.326	0.131	0.412	0.883	0.530	0.464	0.093	0.158	0.332	0.128	0.420	0.878
1.0	1.0	2.0	0.548	0.443	0.088	0.136	0.350	0.131	0.406	0.875	0.550	0.447	0.084	0.144	0.356	0.131	0.410	0.869
1.0	1.0	3.0	0.591	0.375	0.061	0.069	0.503	0.123	0.386	0.825	0.600	0.375	0.059	0.081	0.508	0.119	0.380	0.814
1.0	1.0	4.0	0.640	0.312	0.042	0.035	0.686	0.106	0.368	0.747	0.630	0.311	0.046	0.017	0.692	0.100	0.370	0.744
1.0	1.0	4.5	0.661	0.283	0.039	0.980	0.792	0.091	0.374	0.705	0.670	0.286	0.042	0.986	0.776	0.089	0.370	0.706
1.0	1.0	5.0	0.681	0.256	0.037	1.000	0.871	0.077	0.382	0.661	0.670	0.256	0.038	0.958	0.868	0.069	0.380	0.658
1.0	2.0	0.1	0.733	0.858	0.927	0.412	0.052	0.193	0.860	0.950	0.730	0.858	0.938	0.408	0.052	0.192	0.860	0.950
1.0	2.0	0.3	0.544	0.685	0.522	0.305	0.099	0.142	0.584	0.920	0.560	0.681	0.515	0.308	0.104	0.139	0.580	0.919
1.0	2.0	0.6	0.499	0.581	0.269	0.258	0.148	0.142	0.465	0.910	0.510	0.581	0.280	0.253	0.144	0.142	0.460	0.917
1.0	2.0	1.0	0.504	0.515	0.165	0.218	0.205	0.151	0.419	0.904	0.510	0.517	0.171	0.211	0.200	0.150	0.410	0.914
1.0	2.0	1.2	0.509	0.493	0.139	0.200	0.233	0.153	0.405	0.901	0.510	0.489	0.143	0.197	0.228	0.153	0.400	0.908
1.0	2.0	1.5	0.519	0.466	0.115	0.173	0.274	0.153	0.392	0.893	0.510	0.464	0.115	0.172	0.272	0.156	0.390	0.900
1.0	2.0	1.8	0.530	0.444	0.099	0.148	0.317	0.150	0.383	0.884	0.530	0.442	0.097	0.153	0.320	0.153	0.390	0.886
1.0	2.0	2.0	0.538	0.431	0.092	0.132	0.347	0.147	0.378	0.876	0.550	0.428	0.087	0.139	0.348	0.150	0.380	0.878
1.0	2.0	3.0	0.584	0.369	0.063	0.067	0.522	0.128	0.360	0.824	0.590	0.367	0.061	0.078	0.520	0.133	0.360	0.822
1.0	2.0	4.0	0.635	0.308	0.043	0.034	0.710	0.106	0.343	0.746	0.630	0.306	0.048	0.014	0.712	0.108	0.350	0.750
1.0	2.0	4.5	0.657	0.279	0.040	0.982	0.811	0.090	0.349	0.703	0.670	0.281	0.043	0.986	0.796	0.094	0.340	0.708
1.0	2.0	5.0	0.676	0.253	0.038	1.000	0.884	0.076	0.358	0.659	0.680	0.253	0.039	0.958	0.896	0.075	0.360	0.661
<u>Iteration= 100000</u>			eps= 0.015907			error_1= 0.001012			error_t= 0.000999									
learn & test finished at time alpha =0.930 epsilon =0.50																		
BFQ74 Transistoruna ait eğitime sonuçları yukarıda verilmiştir.																		
Not:Hesaplanan ve arzulanın İşaret ve Gürültü parametrelerinin normalize edilmiş olduğuna dikkat ediniz.																		



Ölçülen değer x — x hesaplanan değer 0 ---- 0

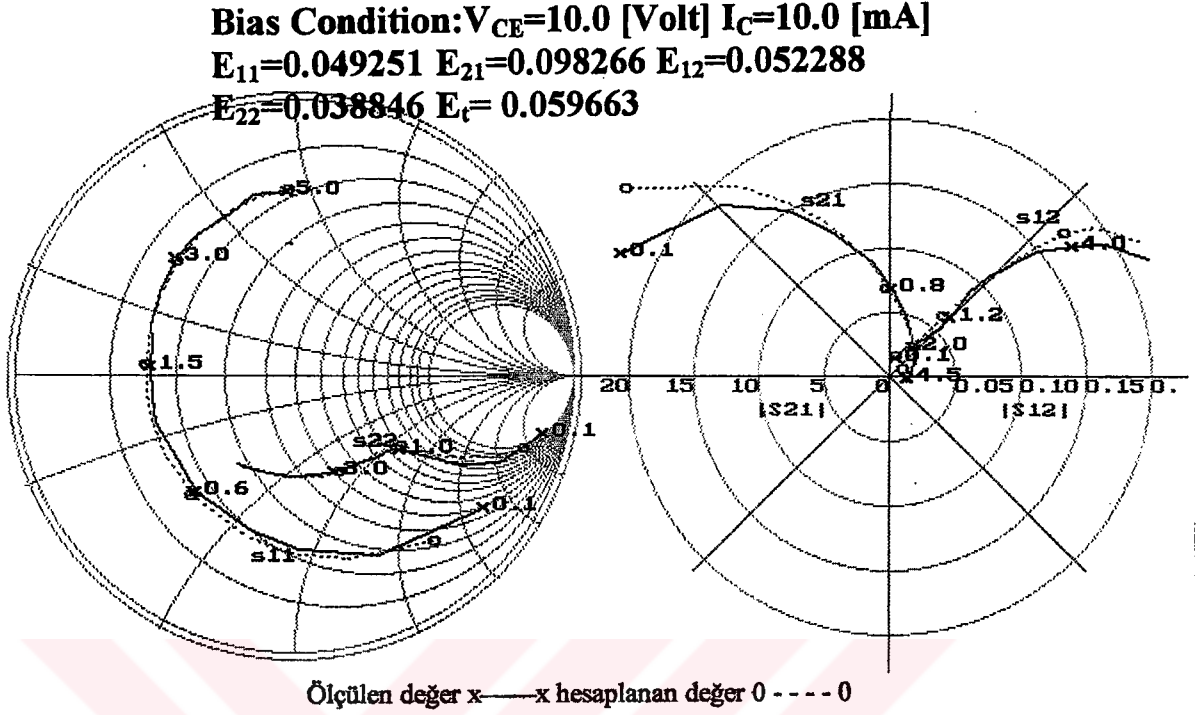
Şekil 4.14 Ortak emetörlü konfigürasyonunda,  $V_{CE}=10$  V  $I_C=2.0$  mA kutuplama noktasında ölçülen ve hesaplanan S-parametrelerinin frekansla değişimi.



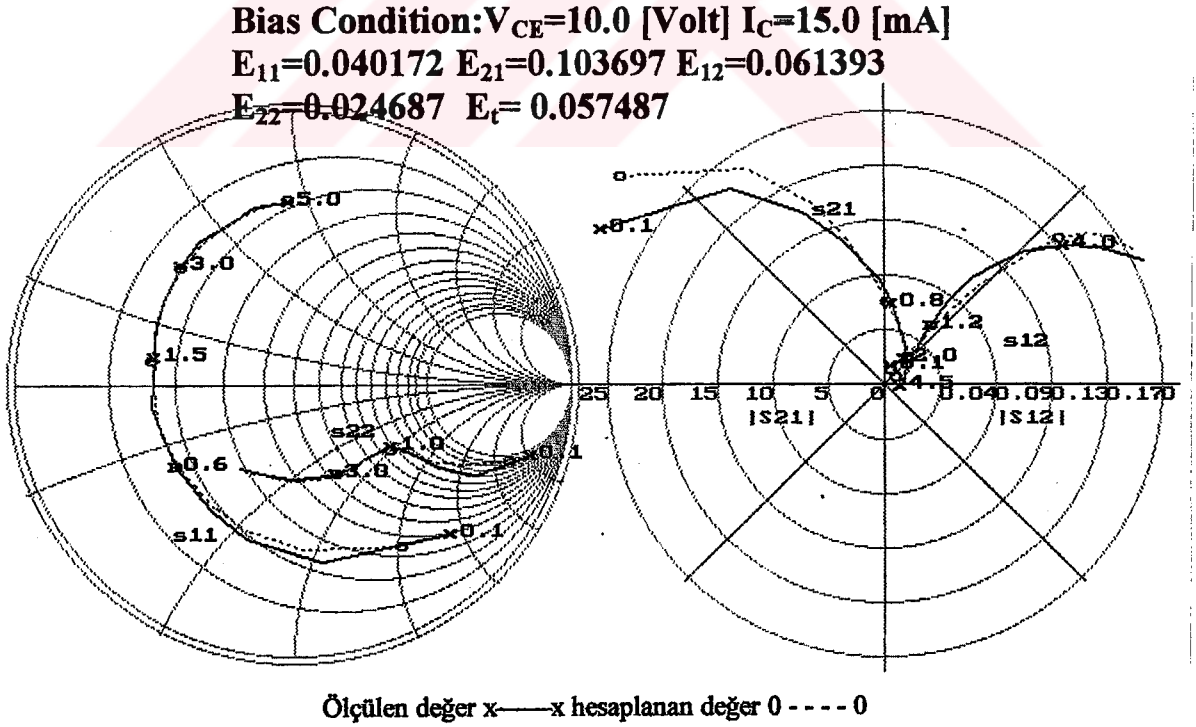
Ölçülen değer x — x hesaplanan değer 0 ---- 0

Şekil 4.15 Ortak emetörlü konfigürasyonunda,  $V_{CE}=10$  V  $I_C=5.0$  mA kutuplama noktasında ölçülen ve hesaplanan S-parametrelerinin frekansla değişimi.(WFBG)





Şekil 4.16 Ortak emetörlü konfigürasyonunda,  $V_{CE}=10$  V  $I_C=10$  mA kutuplama noktasında ölçülen ve hesaplanan S-parametrelerinin frekansla değişimi.

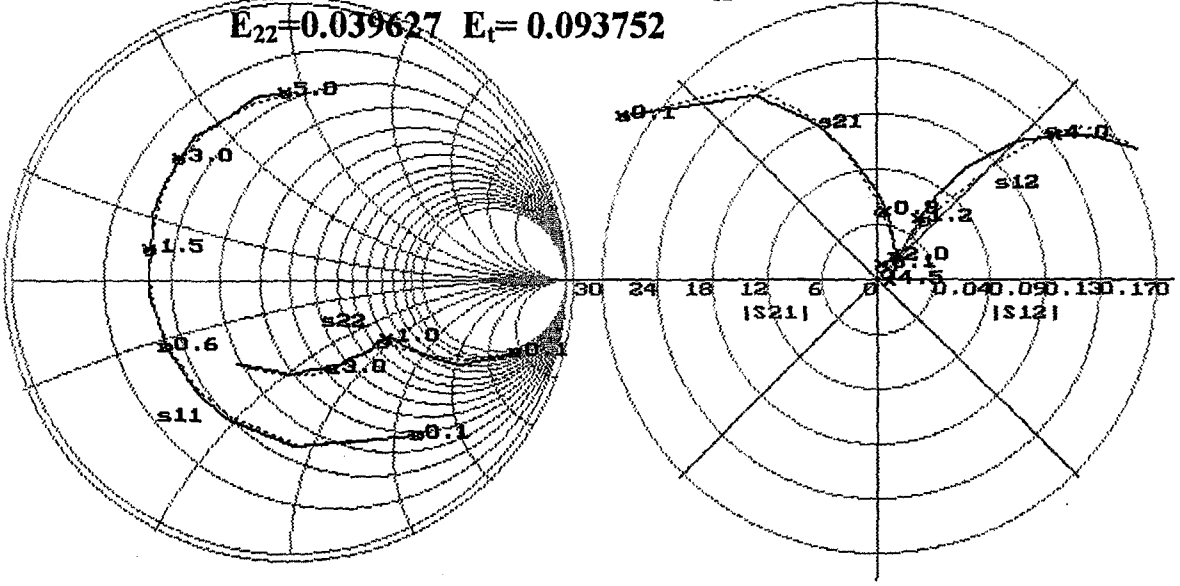


Şekil 4.17 Ortak emetörlü konfigürasyonunda,  $V_{CE}=10$  V  $I_C=15$  mA kutuplama noktasında ölçülen ve hesaplanan S-parametrelerinin frekansla değişimi.(WFBG)

**Bias Condition:  $V_{CE}=10.0$  [Volt]  $I_C=20.0$  [mA]**

$E_{11}=0.057297$   $E_{21}=0.209056$   $E_{12}=0.069029$

$E_{22}=0.039627$   $E_t=0.093752$



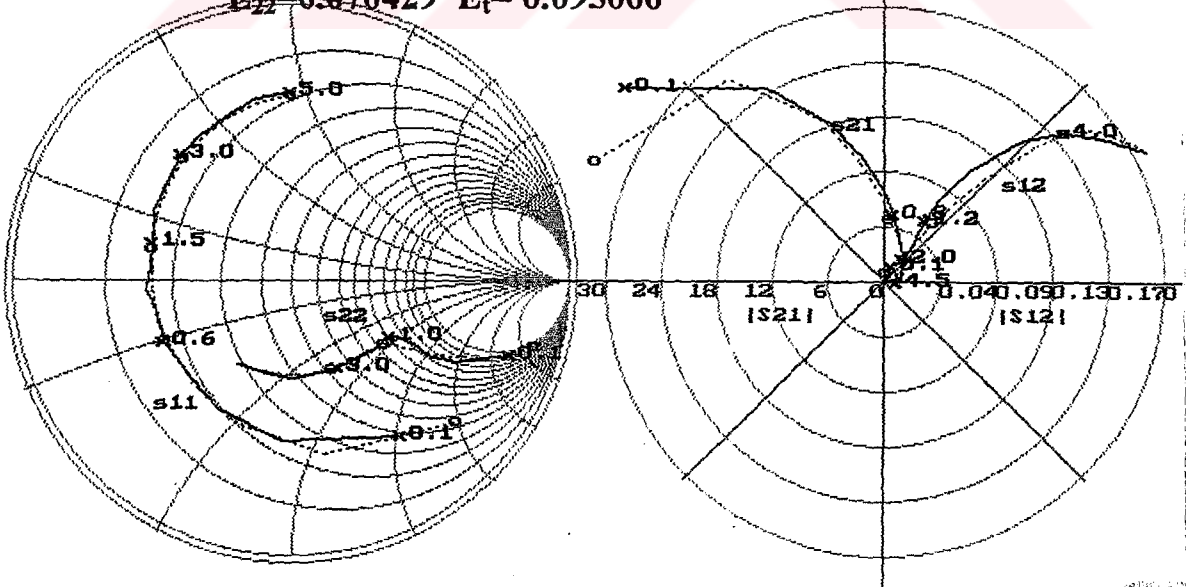
Ölçülen değer x — x hesaplanan değer o ---- o

Şekil 4.18 Ortak emetörlü konfigürasyonunda,  $V_{CE}=10$  V  $I_C=20$  mA kutuplama noktasında ölçülen ve hesaplanan S-parametrelerinin frekansla değişimi.

**Bias Condition:  $V_{CE}=10.0$  [Volt]  $I_C=25.0$  [mA]**

$E_{11}=0.064176$   $E_{21}=0.120213$   $E_{12}=0.125444$

$E_{22}=0.070429$   $E_t=0.095066$



Şekil 4.19 Ortak emetörlü konfigürasyonunda,  $V_{CE}=10$  V  $I_C=25$  mA kutuplama noktasında ölçülen ve hesaplanan S-parametrelerinin frekansla değişimi.

## 4.4.2 BFQ74 için Üretici Firma Verileri

Tablo 4.11 BFQ74 için verilen üretici firma sinyal ve gürültü parametreleri değerleri

$V_{CE}=10\text{ V } I_C=2\text{ mA CE}$					$V_{CE}=10\text{ V } I_C=5\text{ mA CE}$				
f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>	f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>
0.1	0.96	-16	6.83	169	0.017	79	0.99	-5	
0.2	0.93	-33	6.61	155	0.034	70	0.96	-11	
0.3	0.88	-50	6.18	144	0.049	62	0.92	-16	
0.4	0.84	-64	5.62	134	0.060	54	0.88	-20	
0.6	0.77	-89	4.78	118	0.076	43	0.81	-26	
0.8	0.71	-110	3.98	104	0.085	34	0.74	-31	
1	0.68	-127	3.41	93	0.089	29	0.70	-34	
1.2	0.65	-141	2.95	84	0.091	25	0.67	-37	
1.5	0.63	-158	2.45	72	0.091	22	0.64	-41	
1.8	0.63	-172	2.10	62	0.092	21	0.63	-46	
2	0.63	179	1.91	55	0.091	21	0.61	-49	
2.5	0.64	161	1.58	41	0.092	24	0.59	-60	
3	0.66	145	1.36	28	0.099	29	0.59	-71	
3.5	0.68	133	1.20	15	0.113	34	0.58	-83	
4	0.68	118	1.07	3	0.136	35	0.58	-97	
4.5	0.71	107	0.96	-8	0.160	34	0.58	-111	
5	0.72	95	0.85	-18	0.190	29	0.60	-127	

$V_{CE}=10\text{ V } I_C=10\text{ mA CE}$					$V_{CE}=10\text{ V } I_C=15\text{ mA CE}$				
f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>	f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>
0.1	0.83	-35	22.64	155	0.015	70	0.92	-13	
0.2	0.71	-65	18.55	134	0.026	59	0.80	-23	
0.3	0.63	-89	14.98	121	0.034	53	0.70	-27	
0.4	0.58	-105	12.22	112	0.039	48	0.63	-30	
0.6	0.55	-130	8.96	98	0.047	46	0.54	-32	
0.8	0.52	-148	6.91	89	0.053	44	0.49	-34	
1	0.53	-161	5.64	81	0.058	45	0.46	-35	
1.2	0.52	-171	4.76	75	0.064	45	0.44	-37	
1.5	0.52	176	3.87	65	0.072	46	0.43	-40	
1.8	0.53	167	3.25	57	0.083	46	0.42	-44	
2	0.55	161	2.95	52	0.089	47	0.41	-47	
2.5	0.57	148	2.41	40	0.107	46	0.39	-56	
3	0.60	135	2.06	29	0.127	43	0.38	-67	
3.5	0.62	125	1.82	18	0.148	41	0.37	-78	
4	0.63	112	1.62	6	0.173	36	0.37	-92	
4.5	0.67	103	1.46	-5	0.194	32	0.37	-106	
5	0.67	92	1.32	-15	0.217	25	0.38	-123	

$V_{CE}=10\text{ V } I_C=20\text{ mA CE}$					$V_{CE}=10\text{ V } I_C=25\text{ mA CE}$				
f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>	f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>
0.1	0.73	-51	32.84	147	0.013	69	0.86	-18	
0.2	0.61	-89	24.03	124	0.021	55	0.69	-27	
0.3	0.56	-115	18.02	111	0.026	50	0.58	-29	
0.4	0.53	-130	14.07	103	0.030	49	0.52	-30	
0.6	0.51	-151	9.80	91	0.036	51	0.46	-30	
0.8	0.50	-165	7.40	83	0.043	53	0.43	-30	
1	0.51	-174	5.97	76	0.050	54	0.41	-31	
1.2	0.51	176	5.01	71	0.057	55	0.40	-33	
1.5	0.51	167	4.04	62	0.068	56	0.39	-36	
1.8	0.53	159	3.38	55	0.080	55	0.39	-41	
2	0.55	154	3.06	50	0.087	54	0.38	-44	
2.5	0.57	143	2.50	39	0.108	52	0.36	-53	
3	0.59	132	2.13	28	0.130	48	0.36	-64	
3.5	0.62	123	1.87	17	0.152	45	0.34	-76	
4	0.63	110	1.67	5	0.178	39	0.35	-90	
4.5	0.67	101	1.50	-5	0.199	34	0.34	-105	
5	0.68	91	1.35	-15	0.224	27	0.36	-122	



#### 4.5 Ortak Emetör Konfigürasyonundaki NE02135 Transistörü Üzerinde Geliştirilen Tek Kutuplama Gerilimli ,farklı Kutuplama Akımları için Eşdeğer YSA Sinyal-Gürültü Modeli

Bu bölümde NE02135 için muhtelif kutuplama noktaları için işaret ve gürültü parametreleri eşdeğer YSA ile modellenmiştir.Band genişliği 0.1-4 GHz'dir.

##### 4.5.1 NE02135 için Üretici Firma Verileri

Tablo 4.12 NE02135 için verilen üretici firma sinyal ve gürültü parametreleri değerleri

$V_{CE}=10\text{ V } I_C= 5\text{ mA CE}$												
f [GHz]	$S_{11}$	$S_{21}$	$S_{12}$	$S_{22}$	$F_{min}$	$\Gamma_{opt}$	$R_N/50$					
0.1	.84	-36	13.82	156	.02	73	.94	-18				
0.5	.68	-126	7.18	106	.08	35	.51	-53	1.2	.36	69	.14
1	.66	-163	4.02	81	.09	27	.34	-66	1.5	.31	124	.12
1.5	.65	178	2.75	64	.10	27	.31	-74	2.0	.50	164	.05
2	.65	163	2.10	52	.12	30	.31	-83	2.4	.44	-175	.06
2.5	.66	151	1.68	39	.13	26	.31	-95	2.6	.52	-161	.10
3	.66	141	1.46	27	.14	26	.33	-106	3.6	.68	-141	.14
3.5	.67	129	1.24	17	.16	26	.36	-116	3.7	.71	-139	.21
4	.68	121	1.14	5	.17	23	.38	-127				
$V_{CE}=10\text{ V } I_C= 10\text{ mA CE}$												
f [GHz]	$S_{11}$	$S_{21}$	$S_{12}$	$S_{22}$								
.1	.73	-55	22.55	148	.02	65	.87	-27				
.5	.64	-148	8.90	98	.06	37	.36	-66				
1	.64	-176	4.71	77	.07	39	.23	-82				
1.5	.64	169	3.19	63	.09	40	.21	-87				
2	.65	156	2.42	52	.11	42	.21	-97				
2.5	.65	145	1.95	40	.13	37	.22	-108				
3	.66	135	1.69	29	.15	35	.25	-118				
3.5	.66	125	1.43	19	.17	32	.27	-127				
4	.68	117	1.34	8	.19	28	.29	-137				
$V_{CE}=10\text{ V } I_C= 20\text{ mA CE}$												
f [GHz]	$S_{11}$	$S_{21}$	$S_{12}$	$S_{22}$	$F_{min}$	$\Gamma_{opt}$	$R_N/50$					
0.1	.62	-80	31.13	139	.01	60	.77	-37				
0.5	.64	-163	9.88	93	.04	46	.26	-79	1.8	.16	149	.15
1	.65	176	5.07	75	.06	49	.16	-95	1.9	.33	169	.13
1.5	.64	164	3.45	63	.08	50	.15	-101	2.4	.46	-179	.09
2	.65	154	2.60	52	.11	51	.16	-111	2.9	.53	-167	.08
2.5	.66	142	2.10	40	.13	43	.18	-121	3.2	.57	-154	.14

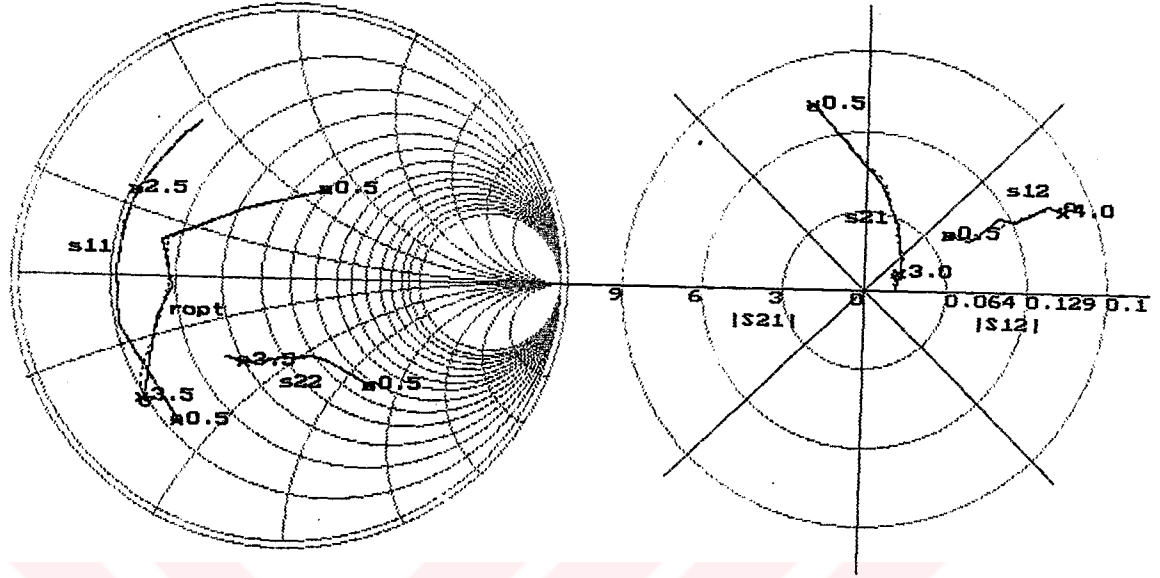
Tablo 4.12 NE02135 için verilen üretici firma sinyal ve gürültü parametreleri değerleri (Devamı)

3	.66	133	1.81	30	.15	40	.20	-129	3.9	.62	-139	.27
3.5	.66	122	1.55	20	.17	36	.22	-136	4.3	.67	-134	.42
4	.68	117	1.34	8	.19	28	.29	-137				
V <sub>CE</sub> =10 V I <sub>C</sub> = 30 mA CE												
f [GHz]	S <sub>11</sub>		S <sub>21</sub>		S <sub>12</sub>		S <sub>22</sub>					
.1	.58	-95	35.35	134	.01	59	.72	-40				
.5	.64	-169	10.11	91	.03	50	.22	-82				
1	.65	173	5.15	74	.06	55	.14	-97				
1.5	.65	162	3.49	62	.08	53	.14	-103				
2	.66	152	2.63	52	.11	54	.15	-112				
2.5	.66	141	2.10	39	.13	46	.17	-122				
3	.66	132	1.82	29	.15	42	.19	-129				
3.5	.67	122	1.54	20	.17	38	.22	-137				
4	.68	115	1.44	9	.20	31	.24	-146				

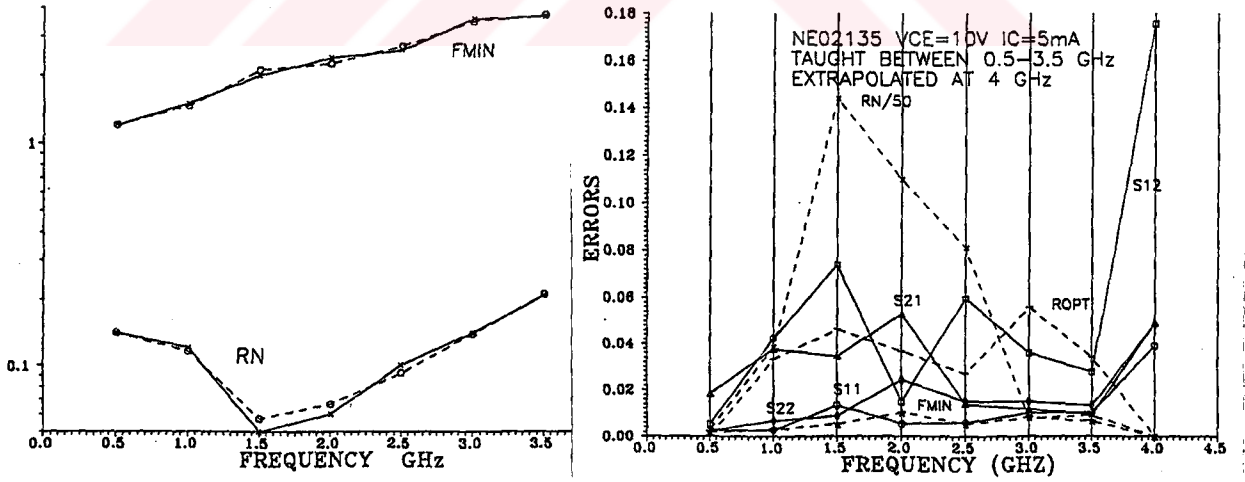
Tablo 4.13 NE02135 için verilen üretici firma sinyal ve gürültü parametreleri değerleri

Name of the transistor: NE02135				
Operation Frequency Bandwidth: 0.1 GHz - 4 GHz				
Operations Bias Conditions				
Bias Voltage (V <sub>CE</sub> )	NMDA: No Measured Data Available			
[V]	Bias Currents (I <sub>C</sub> ) [mA]			
10.0	5.0	10.0	20.0	30.0
	SFG	WFBG	SFG	WFBG
	E <sub>t</sub> =0.020487	E <sub>t</sub> =0.206208	E <sub>t</sub> =0.01798	E <sub>t</sub> =0.323723
	E <sub>11</sub> =0.007232	E <sub>11</sub> =0.169230	E <sub>11</sub> =0.008618	E <sub>11</sub> =0.337621
	E <sub>21</sub> =0.037046	E <sub>21</sub> =0.143008	E <sub>21</sub> =0.025877	E <sub>21</sub> =0.279333
	E <sub>12</sub> =0.025439	E <sub>12</sub> =0.301374	E <sub>12</sub> =0.020676	E <sub>12</sub> =0.302810
	E <sub>22</sub> =0.012227	E <sub>22</sub> =0.211221	E <sub>22</sub> =0.016775	E <sub>22</sub> =0.375153
	F <sub>1</sub> =0.031645	F <sub>1</sub> =NMDA	F <sub>1</sub> =0.011418	F <sub>1</sub> =NMDA
	F <sub>2</sub> =0.032288	F <sub>2</sub> =NMDA	F <sub>2</sub> =0.048255	F <sub>2</sub> =NMDA
	F <sub>3</sub> =0.056156	F <sub>3</sub> =NMDA	F <sub>3</sub> =0.014451	F <sub>3</sub> =NMDA
	F <sub>t</sub> =0.040363	F <sub>t</sub> =NMDA	F <sub>t</sub> =0.024708	F <sub>t</sub> =NMDA
	Şekil 4.20	Şekil 4.21	Şekil 4.23	Şekil 4.22

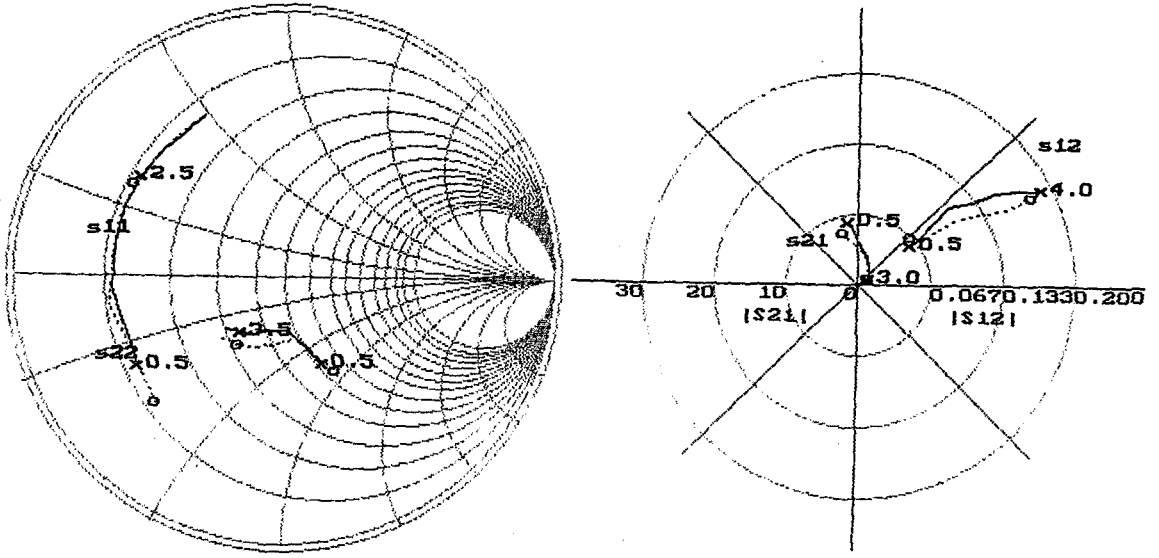
## 4.5.2 NE02135 İin Elde Edilen Simülasyon Neticeleri



Ölülen deęer x — x hesaplanan deęer 0 - - - - 0

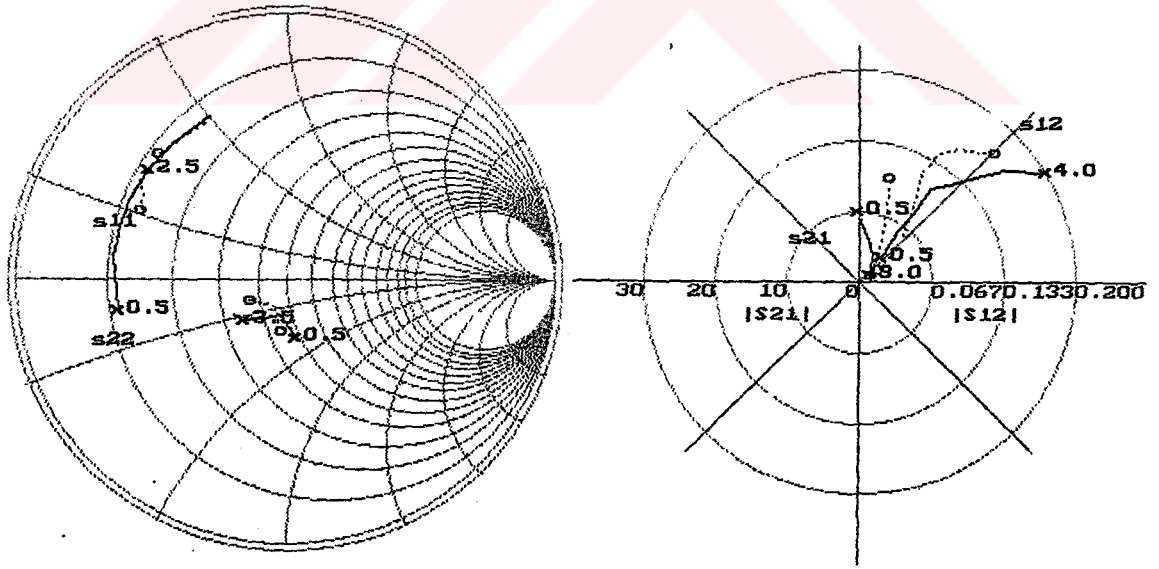


Şekil 4.20 OE konfigürasyonunda  $V_{CE}=10$  V  $I_C=5$  mA kutuplama noktasında ölçülmüş ve hesaplanmış S&N-parametresi-frekans deęişimi ve Hata-frekans dağılımları(WBFG).



Ölçülen değer x — x hesaplanan değer 0 - - - - 0

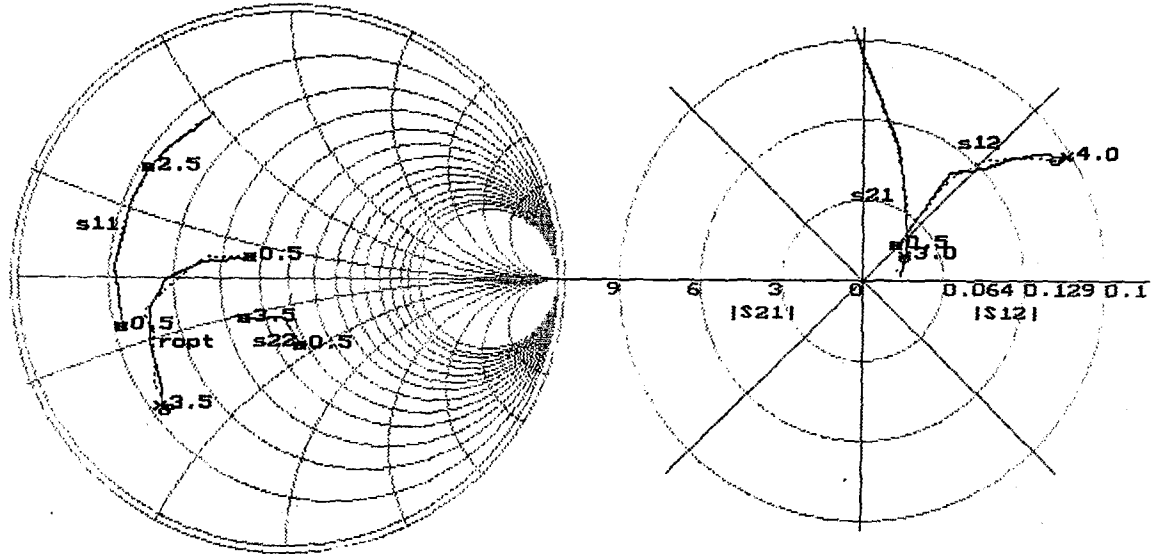
Şekil 4.21 OE konfigürasyonunda  $V_{CE}=10\text{ V}$   $I_C=10\text{ mA}$  kutuplama noktasında ölçülmüş ve hesaplanmış S&N-parametresi-frekans değişimi.



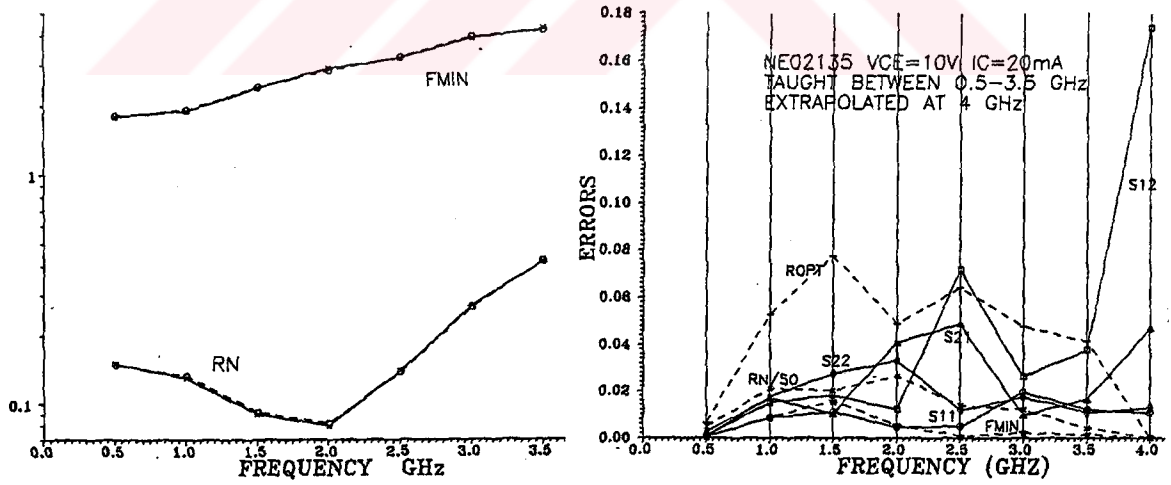
Ölçülen değer x — x hesaplanan değer 0 - - - - 0

Şekil 4.22 OE konfigürasyonunda  $V_{CE}=10\text{ V}$   $I_C=30\text{ mA}$  kutuplama noktasında ölçülmüş ve hesaplanmış S&N-parametresi-frekans değişimi.





Ölçülen değer x — x hesaplanan değer 0 - - - - 0



Şekil 4.23 OE konfigürasyonunda  $V_{CE}=10$  V  $I_C=20$  mA kutuplama noktasında ölçülmüş ve hesaplanmış S&N-parametresi-frekans değişimi ve Hata-frekans dağılımları(WBFG).

#### 4.6 Ortak Emetör Konfigürasyonundaki BFR28 Transistörü Üzerinde Geliştirilen farklı Kutuplama Noktaları için Eşdeğer Sinyal YSA Modeli

Bu bölümde BFR28 transistörü için muhtelif kutuplama noktalarında işaret parametreleri eşdeğer YSA ile modellenmiştir. Band genişliği 0.1-3 GHz'dir.

##### 4.6.1 BFR28 İçin Elde Edilen Simülasyon Neticeleri

Tablo 4.14 BFR28 için verilen üretici firma sinyal ve gürültü parametreleri değerleri

Name of the transistor: BFR28				
Operation Frequency Bandwidth: 0.1 GHz - 3 GHz				
Operations Bias Conditions				
Bias Voltage ( $V_{CE}$ ) [V]	Bias Currents ( $I_C$ ) [mA]			
1.0	0.5	0.75	1.0	
	SFG	WFBG	SFG	
	$E_t=0.025252$	$E_t=0.030763$	$E_t=0.024568$	
	$E_{11}=0.023428$	$E_{11}=0.032515$	$E_{11}=0.026304$	
	$E_{21}=0.026441$	$E_{21}=0.041766$	$E_{21}=0.018630$	
	$E_{12}=0.042841$	$E_{12}=0.033339$	$E_{12}=0.038530$	
	$E_{22}=0.008296$	$E_{22}=0.015432$	$E_{22}=0.014807$	
	Şekil 4.24	Şekil 4.25	Şekil 4.26	
3.0	2.0			
	WFBG			
	$E_t=0.072924$	$E_{11}=0.13815$	$E_{21}=0.03893$	$E_{12}=0.08422$ $E_{22}=0.03037$
Şekil 4.27				
5	3	8		
	SFG	WFBG		
	$E_t=0.026458$	$E_t=0.772174$		
	$E_{11}=0.049159$	$E_{11}=0.971943$		
	$E_{21}=0.006185$	$E_{21}=0.768650$		
	$E_{12}=0.036394$	$E_{12}=0.709091$		
	$E_{22}=0.014095$	$E_{22}=0.639012$		
	Şekil 4.28	Şekil 4.29		

## 4.6.2 BFR28 için Üretici Firma Verileri

Tablo 4.15 BFR28 için verilen üretici firma sinyal parametreleri değerleri

$V_{CE}=1\text{ V}$ $I_C=0.5\text{ mA}$ CE					$V_{CE}=1\text{ V}$ $I_C=1\text{ mA}$ CE												
f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>	f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>								
.10	.968	-5.3	1.79	174.2	.021	86.0	.996	-3.1	.10	.943	-7.2	3.35	172.6	.021	85.0	.993	-4.1
.20	.961	-10.5	1.78	168.2	.041	82.5	.991	-6.1	.20	.927	-14.2	3.32	165.1	.040	80.7	.982	-8.0
.30	.948	-15.7	1.77	162.1	.061	78.6	.984	-9.1	.30	.903	-21.0	3.25	157.7	.059	76.1	.967	-11.8
.40	.932	-20.7	1.75	156.1	.080	75.1	.975	-12.0	.40	.873	-27.6	3.16	150.5	.077	72.1	.946	-15.4
.50	.915	-25.6	1.73	150.6	.098	71.7	.963	-14.8	.50	.841	-33.9	3.06	144.0	.093	68.2	.923	-18.7
.60	.895	-30.4	1.70	145.1	.115	68.6	.951	-17.5	.60	.804	-39.8	2.95	137.8	.107	64.9	.899	-21.7
.70	.876	-35.2	1.67	139.7	.131	65.4	.936	-20.1	.70	.767	-45.5	2.84	131.8	.121	61.8	.874	-24.5
.80	.847	-40.1	1.63	134.7	.146	62.2	.922	-25.5	.80	.728	-51.2	2.72	126.3	.132	58.7	.849	-27.0
.90	.825	-44.7	1.63	129.6	.160	59.2	.903	-24.9	.90	.691	-56.5	2.64	121.0	.143	56.0	.821	-29.3
.95	.808	-47.2	1.62	127.0	.166	57.7	.894	-26.0	.95	.668	-59.3	2.59	118.3	.147	54.8	.808	-3.3
1	.797	-49.3	1.60	124.5	.172	56.3	.885	-27.2	1	.652	-61.8	2.54	115.8	.152	53.7	.796	-31.3
1.2	.733	-58.6	1.55	114.6	.192	51.2	.849	-31.1	1.20	.572	-71.7	2.35	106.1	.166	50.0	.750	-34.6
1.4	.672	-66.7	1.48	105.7	.209	46.9	.819	-34.8	1.40	.049	-80.2	2.16	97.6	.178	47.3	.713	-37.6
1.6	.617	-74.1	1.41	98.0	.222	43.4	.790	-38.2	1.60	.440	-87.5	2.00	90.6	.188	45.5	.684	-40.1
1.7	.590	-78.4	1.39	94.5	.228	41.6	.776	-39.9	1.70	.414	-91.8	1.94	87.3	.193	44.6	.670	-41.4
1.75	.580	-80.3	1.37	92.6	.231	40.9	.769	-40.7	1.75	.401	-93.9	1.91	85.6	.195	44.3	.663	-42.1
2	.520	-91.5	1.33	83.7	.242	37.1	.732	-44.2	2	.349	-106.1	1.78	77.6	.206	42.7	.629	-44.8
2.5	.401	-115.5	1.21	67.2	.250	32.3	.673	-51.2	2.50	.254	-133.9	1.54	63.2	.224	41.7	.578	-50.5
3	.332	-139.0	1.12	54.6	.252	30.5	.636	-57.5	3	.210	-161.2	1.39	51.8	.244	41.6	.549	-55.9

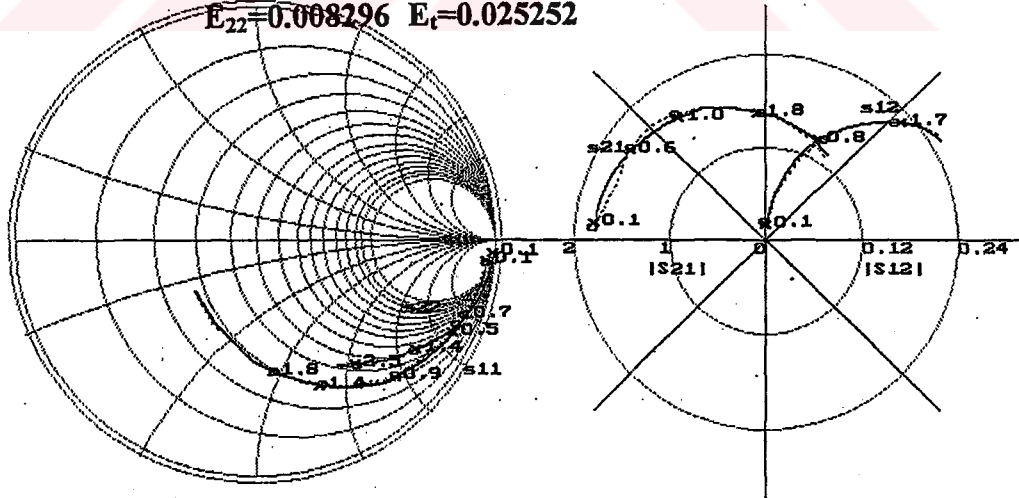
  

$V_{CE}=1\text{ V}$ $I_C=0.75\text{ mA}$ CE					$V_{CE}=5\text{ V}$ $I_C=3\text{ mA}$ CE												
f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>	f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>								
.1	.955	-6.3	2.6	173.4	.021	85.6	.995	-3.6	.1	.869	-11.0	8.22	168.2	.017	83.6	.980	-6.1
.2	.944	-12.4	2.58	166.7	.041	81.5	.988	-7.1	.2	.828	-21.4	7.87	157.0	.032	77.9	.949	-11.7
.3	.926	-18.5	2.55	159.9	.06	77.2	.976	-10.5	.3	.77	-30.7	7.38	146.8	.046	72.9	.906	-16.5
.4	.902	-24.3	2.5	153.3	.078	73.4	.961	-13.8	.4	.706	-39.0	6.81	137.7	.058	69.2	.860	-20.3
.5	.878	-30	2.45	147.3	.095	69.8	.944	-16.9	.5	.644	-46.1	6.27	130.0	.068	66.3	.815	-23.3
.6	.849	-35.4	2.38	141.4	.111	66.5	.925	-19.9	.6	.586	-52.0	5.74	123.3	.077	64.2	.775	-25.7
.7	.819	-40.8	2.32	135.7	.126	63.3	.905	-22.6	.7	.530	-57.6	5.29	117.2	.085	62.6	.739	-27.5
.8	.786	-46.1	2.23	130.3	.139	60.1	.885	-25.1	.8	.482	-62.6	4.87	112.0	.093	61.4	.709	-28.9
.9	.753	-51.2	2.20	125.1	.151	57.2	.861	-27.5	.9	.439	-67.1	4.53	107.2	.100	60.5	.681	-30.0
.95	.735	-53.8	2.17	122.5	.156	55.8	.849	-28.6	.95	.416	-69.5	4.37	104.9	.103	60.1	.668	-30.4
1	.72	-56.2	2.14	120.0	.161	54.5	.837	-29.6	1	.398	-71.5	4.22	102.8	.107	59.9	.657	-30.9
1.2	.645	-65.8	2.01	110.1	.177	50.1	.795	-33.3	1.2	.325	-79.3	3.70	94.9	.119	59.2	.620	-32.2
1.4	.575	-74.3	1.88	101.4	.191	46.6	.760	-36.7	1.4	.263	-85.2	3.27	88.2	.131	58.9	.595	-33.5
1.6	.516	-81.7	1.76	94.0	.202	44.0	.729	-39.7	1.6	.217	-88.8	2.93	82.7	.144	59.0	.579	-34.9
1.7	.489	-86.1	1.72	90.6	.207	42.7	.714	-41.2	1.7	.196	-91.9	2.80	80.1	.151	58.8	.571	-35.7
1.75	.477	-98.1	1.69	88.9	.209	42.3	.708	-41.9	1.75	.189	-93.1	2.73	78.8	.155	58.8	.568	-36.2
1.8	.469	-90.2	1.68	87.0	.211	41.8	.700	-42.5	1.8	.181	-95.5	2.68	77.5	.158	58.7	.563	-36.6
2	.421	-99.6	1.6	80.4	.219	39.9	.671	-44.9	2	.152	-104.8	2.48	72.8	.171	58.3	.547	-38.0
2.5	.311	-125.6	1.41	65.0	.232	37.5	.616	-51.1	2.5	.087	-143.3	2.07	61.6	.205	57.3	.515	-42.6
3	.254	-150.8	1.29	53.1	.244	37.2	.583	-56.8	3	.070	173.9	1.82	52.4	.240	55.5	.508	-42.5

Tablo 4.15 BFR28 için verilen üretici firma sinyal parametreleri değerleri (Devamı)

$V_{CE}= 3 \text{ V } I_C= 2 \text{ mA CE}$					$V_{CE}= 5 \text{ V } I_C= 8 \text{ mA CE}$						
f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>	f [GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>		
.1	.900	-9.3 6.06	170.2 .018	84.2 .987	-5.1	.1	.716	-18.4 14.57	16.9 .015	8.8 .949	-9.5
.2	.872	-18.2 5.9	160.7 .034	79.2 .966	-9.9	.2	.633	-33.7 12.93	144.6 .028	74.3 .874	-16.6
.3	.830	-26.6 5.64	151.7 .049	74.2 .937	-14.2	.3	.538	-45.8 11.12	131.9 .039	7.3 .795	-21.1
.4	.780	-34.2 5.34	143.3 .063	70.3 .902	-18	.4	.454	-54.8 9.53	122.0 .048	68.5 .730	-23.5
.5	.728	-41.2 5.02	136 .075	66.9 .866	-21.2	.5	.387	-62.0 8.24	114.5 .055	67.6 .681	-24.8
.6	.677	-47.4 4.7	129.3 .085	64.2 .831	-24	.6	.334	-67.2 7.22	108.5 .063	67.3 .645	-25.6
.7	.624	-53.3 4.4	123.2 .094	61.9 .798	-26.2	.7	.287	-72.5 6.40	103.3 .070	67.4 .618	-26.1
.8	.578	-58.6 4.12	117.2 .103	59.8 .768	-28.1	.8	.250	-76.9 5.75	98.9 .078	67.3 .597	-26.5
.9	.535	-63.7 3.89	112.7 .11	58.3 .738	-29.7	.9	.220	-81.0 5.22	95.0 .085	67.3 .579	-26.8
.95	.512	-66.2 3.77	110.2 .113	57.6 .725	-30.3	.95	.205	-83.4 4.98	93.2 .089	67.4 .572	-26.8
1	.491	-68.5 3.66	107.9 .117	57.1 .713	-31	1.0	.192	-85.0 4.77	91.5 .092	67.4 .566	-26.9
1.2	.412	-77.3 3.26	99.2 .128	55.5 .670	-32.9	1.2	.142	-93.9 4.08	85.2 .107	67.4 .546	-27.5
1.4	.343	-84.6 2.92	91.9 .140	54.7 .640	-34.7	1.4	.100	-101.6 3.55	79.7 .122	67.2 .535	-28.6
1.6	.29	-89.7 2.64	85.8 .151	54.6 .619	-36.4	1.6	.071	-103.3 3.15	75.1 .137	67.0 .529	-3.0
1.7	.267	-93.5 2.53	83.0 .157	54.3 .610	-37.3	1.7	.059	-110.6 2.99	73.0 .145	66.6 .525	-3.9
1.75	.257	-95.2 2.48	81.6 .160	54.3 .605	-37.8	1.75	.055	-112.6 2.91	72.0 .149	66.5 .524	-31.4
1.8	.249	-97.3 2.43	80.1 .163	54.3 .600	-35.3	1.8	.051	-118.2 2.84	7.9 .153	66.3 .521	-31.8
2	.213	-106.5 2.26	74.9 .174	53.9 .580	-39.8	2.0	.042	-145.5 2.61	66.9 .168	65.5 .511	-33.6
2.5	.139	-139.5 1.91	62.6 .204	53.6 .543	-44.6	2.5	.065	138.7 2.17	57.3 .208	63.2 .489	-38.6
3	.109	-173.1 1.69	52.7 .235	52.7 .525	-49.5	3	.091	117.0 1.88	48.9 .247	6.2 .479	-44.1

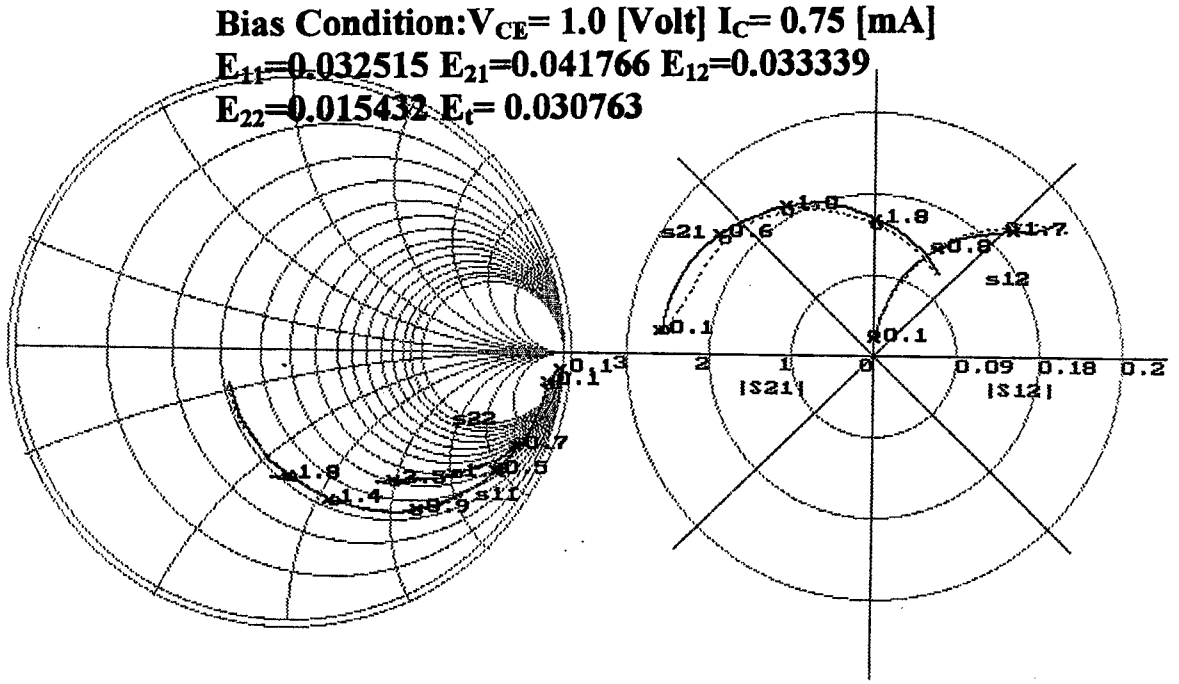
Bias Condition:  $V_{CE}= 1.0 \text{ [Volt] } I_C= 0.5 \text{ [mA]}$   
 $E_{11}=0.023428$   $E_{21}=0.026441$   $E_{12}=0.042841$   
 $E_{22}=0.068296$   $E_t=0.025252$



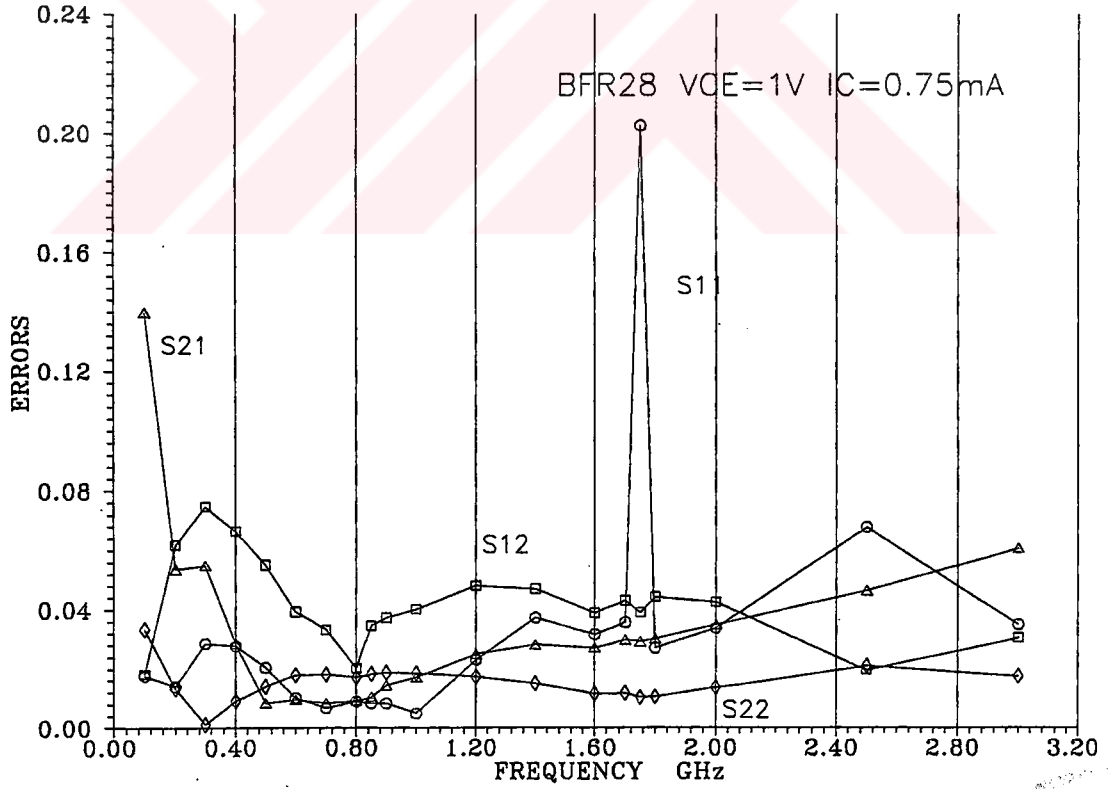
Ölçülen değer x — x hesaplanan değer 0 - - - - 0

Şekil 4.24 OE konfigürasyonda  $V_{CE}=1.0 \text{ V } I_C=0.5 \text{ mA}$  kutuplama noktasında ölçülmüş ve hesaplanmış S-parametresi-frekans değişimleri.



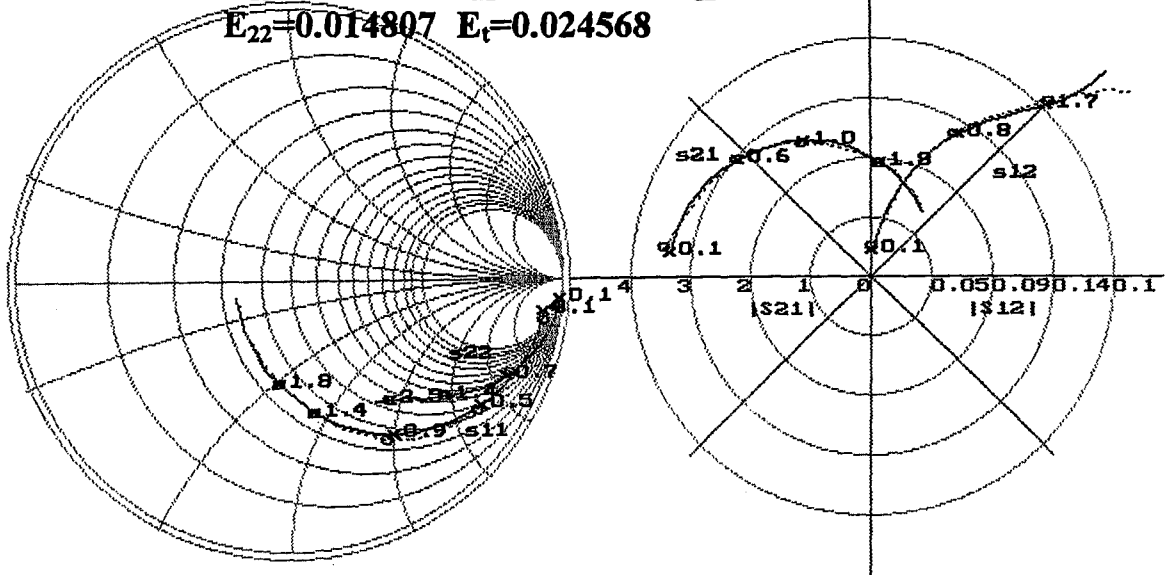


Ölçülen değer x — x hesaplanan değer 0 ---- 0



Şekil 4.25 OE konfigürasyonda  $V_{CE} = 1.0$  V  $I_C = 0.75$  mA kutuplama noktasında ölçülmüş ve hesaplanmış S-parametresi-frekans değişimleri ve hata-frekans dağılımları.(WFBG)

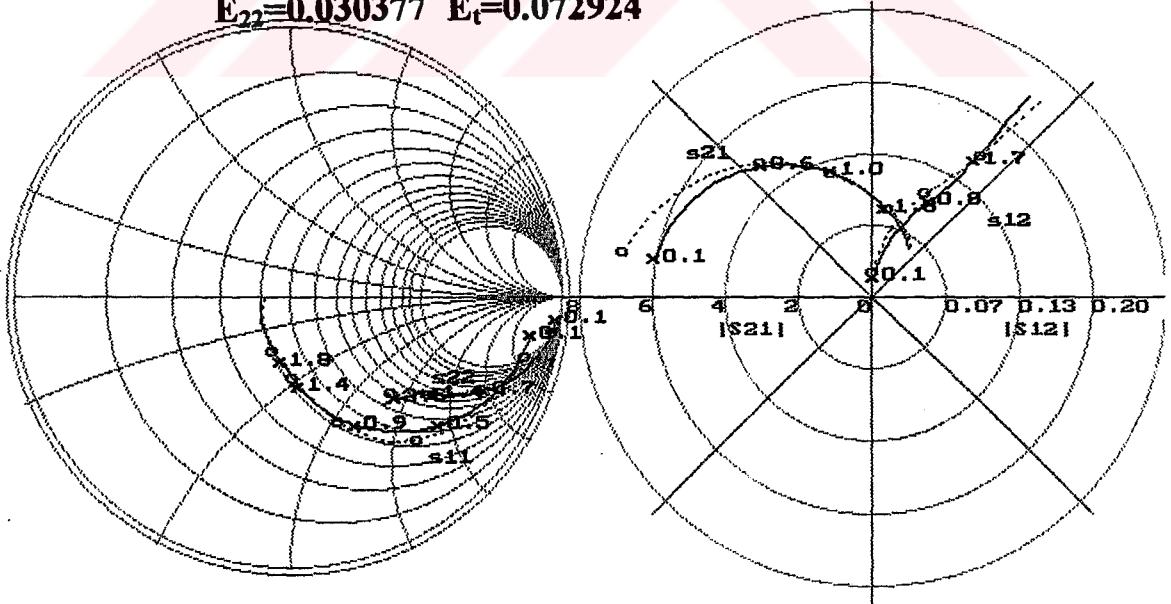
**Bias Condition:  $V_{CE}= 1.0$  [Volt]  $I_C= 1.0$  [mA]**  
 $E_{11}=0.026304$   $E_{21}=0.018630$   $E_{12}=0.038530$   
 $E_{22}=0.014807$   $E_t=0.024568$



Ölçülen değer x — x hesaplanan değer o - - - - 0

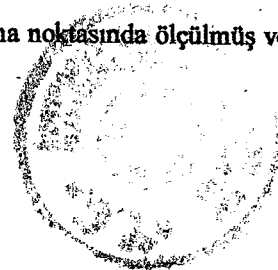
Şekil 4.26 OE konfigürasyonda  $V_{CE}=1.0$  V  $I_C=1.0$  mA kutuplama noktasında ölçülmüş ve hesaplanmış S-parametresi-frekans değişimleri.

**Bias Condition:  $V_{CE}= 3.0$  [Volt]  $I_C= 2.0$  [mA]**  
 $E_{11}=0.138153$   $E_{21}=0.038937$   $E_{12}=0.084228$   
 $E_{22}=0.030377$   $E_t=0.072924$



Ölçülen değer x — x hesaplanan değer o - - - - 0

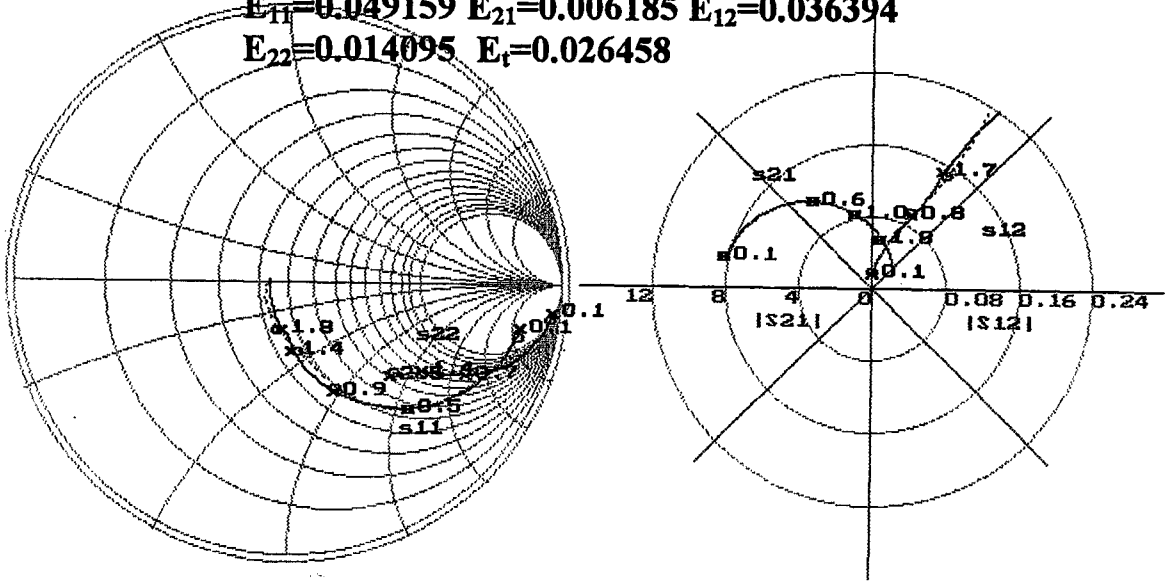
Şekil 4.27 OE konfigürasyonda  $V_{CE}=3.0$  V  $I_C=2.0$  mA kutuplama noktasında ölçülmüş ve hesaplanmış S-parametresi-frekans değişimleri.(WFBG)



**Bias Condition:  $V_{CE}= 5.0$  [Volt]  $I_C= 3.0$  [mA]**

$E_{11}=0.049159$   $E_{21}=0.006185$   $E_{12}=0.036394$

$E_{22}=0.014095$   $E_t=0.026458$



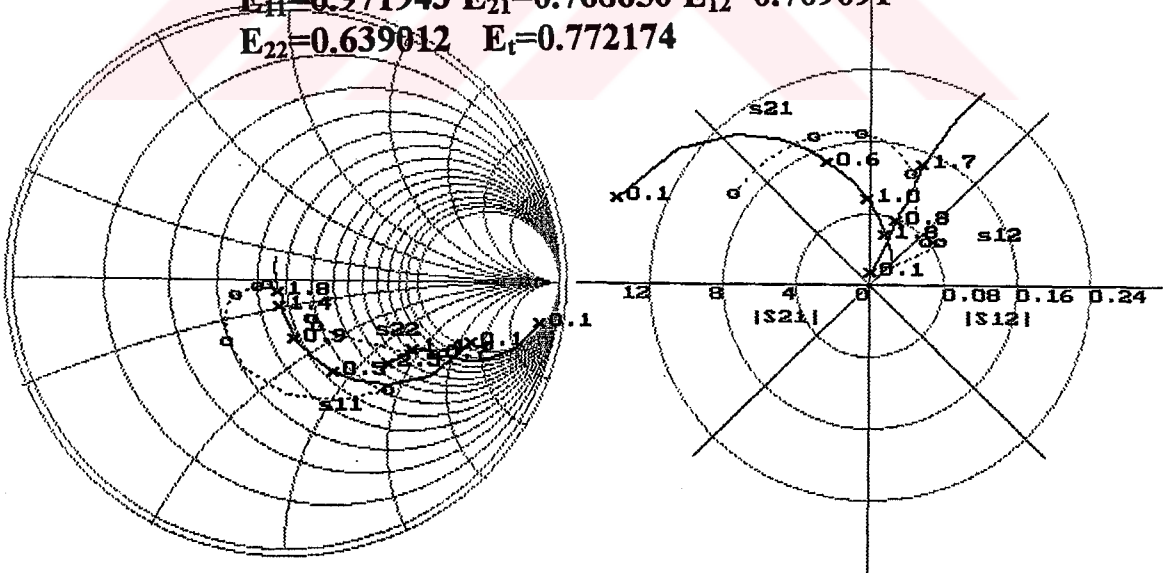
Ölçülen değer x — x hesaplanan değer 0 - - - - 0

Şekil 4.28 OE konfigürasyonda  $V_{CE}=5.0$  V  $I_C=3.0$  mA kutuplama noktasında ölçülmüş ve hesaplanmış S-parametresi-frekans değişimleri.

**Bias Condition:  $V_{CE}= 5.0$  [Volt]  $I_C= 8.0$  [mA]**

$E_{11}=0.971943$   $E_{21}=0.768650$   $E_{12}=0.709091$

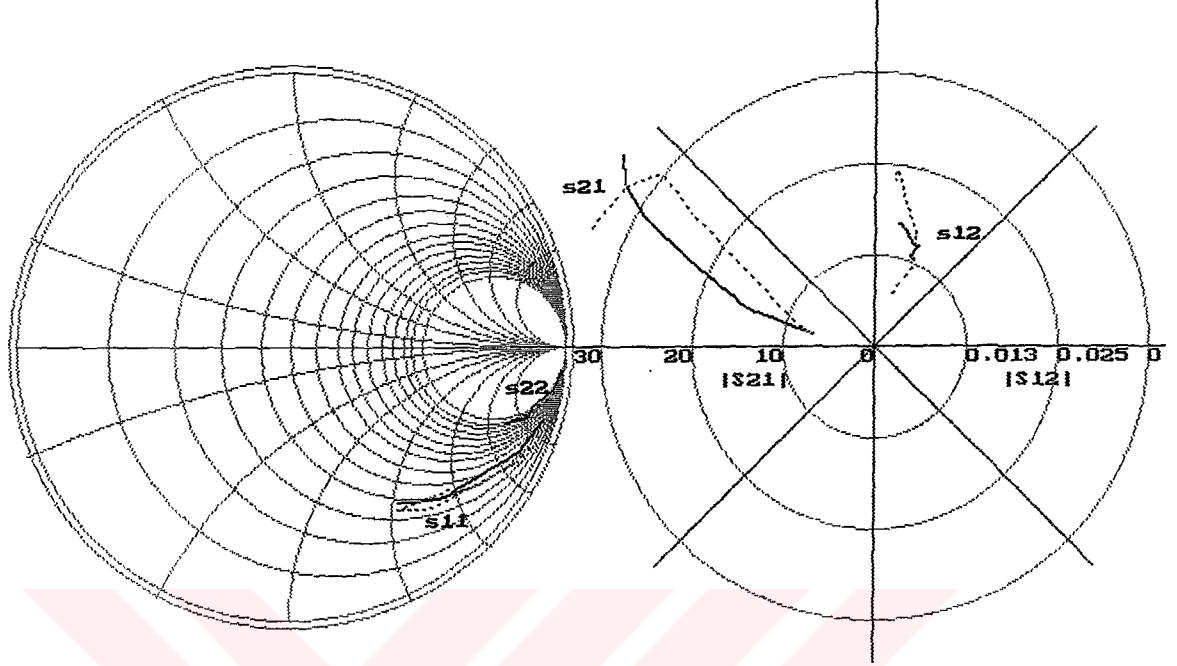
$E_{22}=0.639012$   $E_t=0.772174$



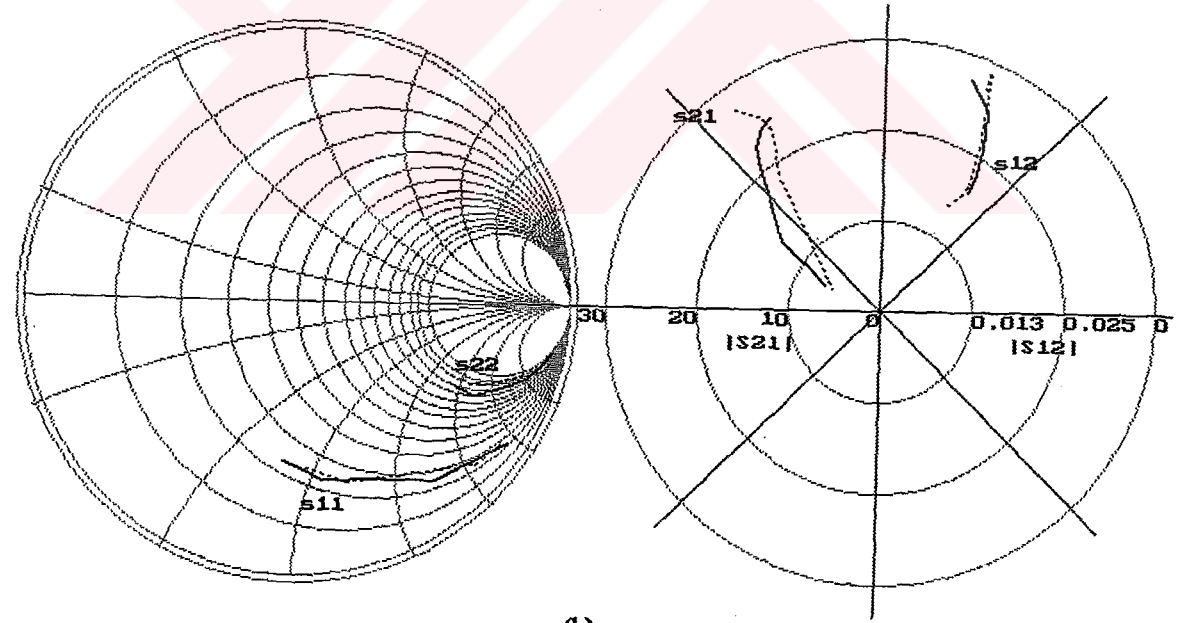
Ölçülen değer x — x hesaplanan değer 0 - - - - 0

Şekil 4.29 OE konfigürasyonda  $V_{CE}=5.0$  V  $I_C=8.0$  mA kutuplama noktasında ölçülmüş ve hesaplanmış S-parametresi-frekans değişimleri.(WFBG)

**4.7 Eşdeğer YSA Yardımıyla Herhangi Bir Frekansta ( $f=0.1 - 5$  GHz) BFQ74 Transistörünün S-parametrelerinin Kutuplama İle ( $V_{CE}=10$  V , $I_C=2-25$  mA) Değişiminin Hesaplanması Uygulamasına Ait Sonuçların Yansıma Katsayısı Düzleminde Çizdirilmesi.**



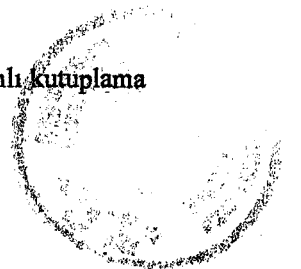
(a)

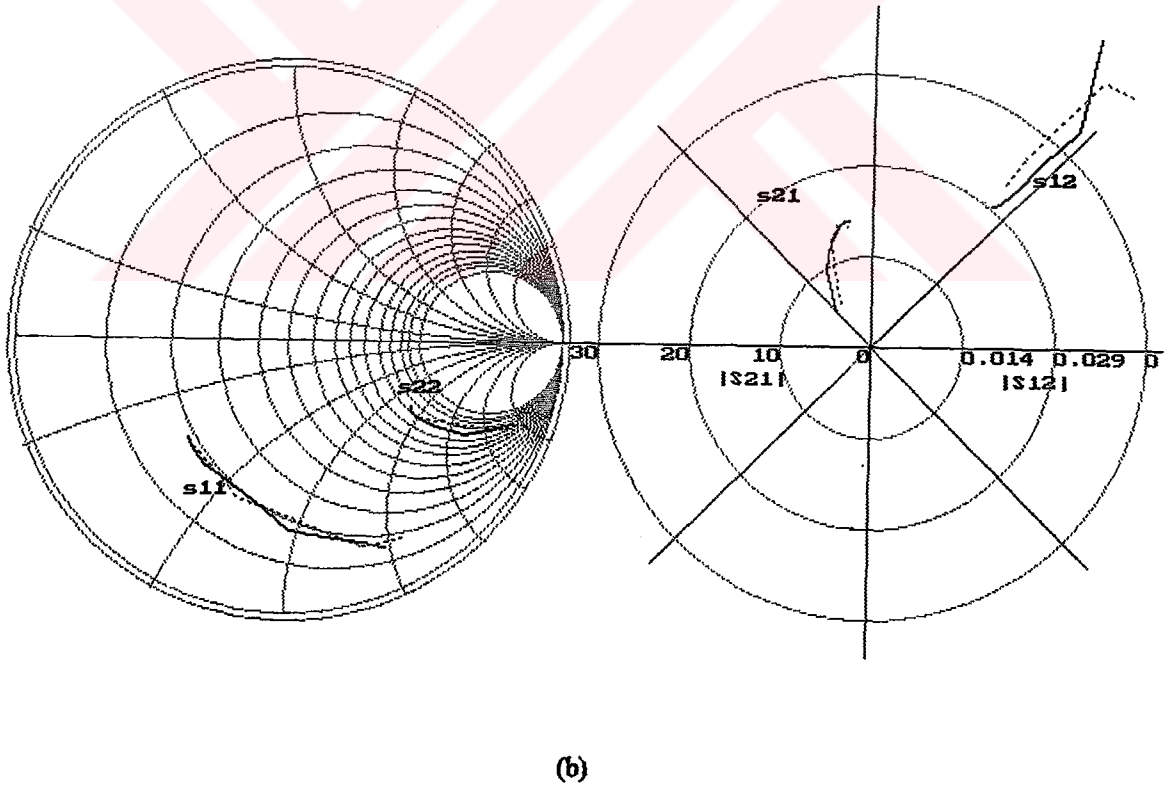
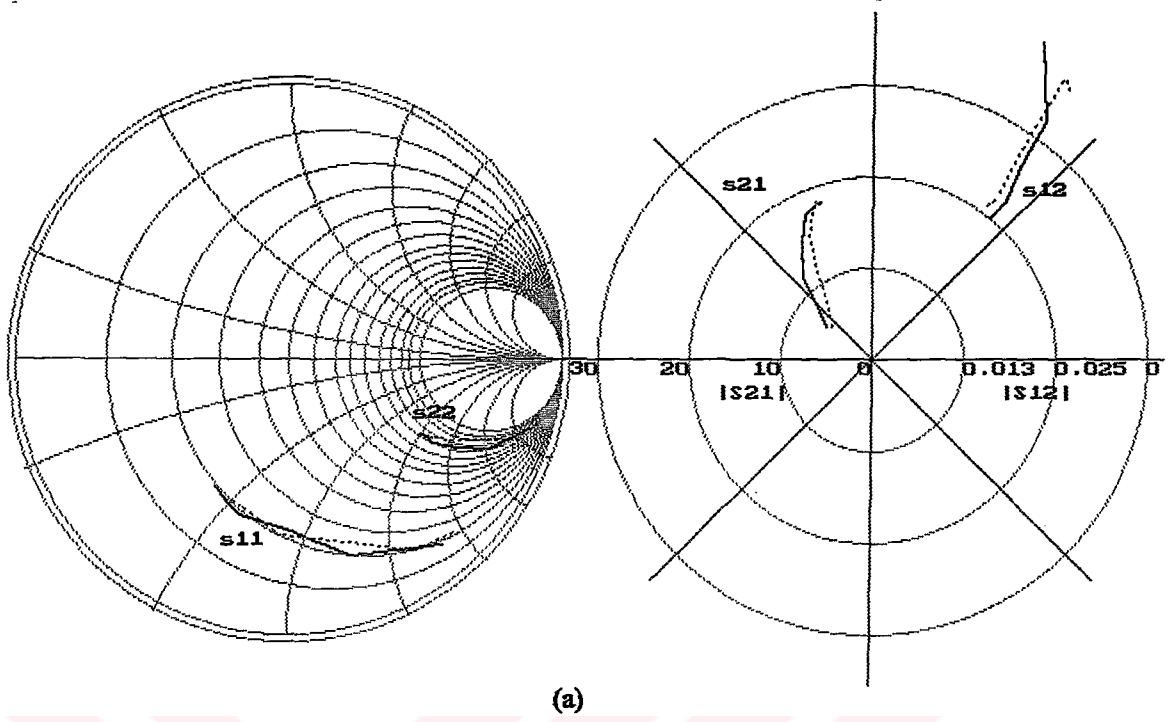


(b)

Öçtülen deęer — hesaplanan deęer - - - - -

Şekil 4.30 a-0.1 b-0.2 GHz frekansında  $V_{CE}=10$  Vsabit ve  $I_C=2-25$  mA deęişken ile tanımlı kutuplama noktaları ile NE02135 transistörünün S-parametrelerinin deęişimi.

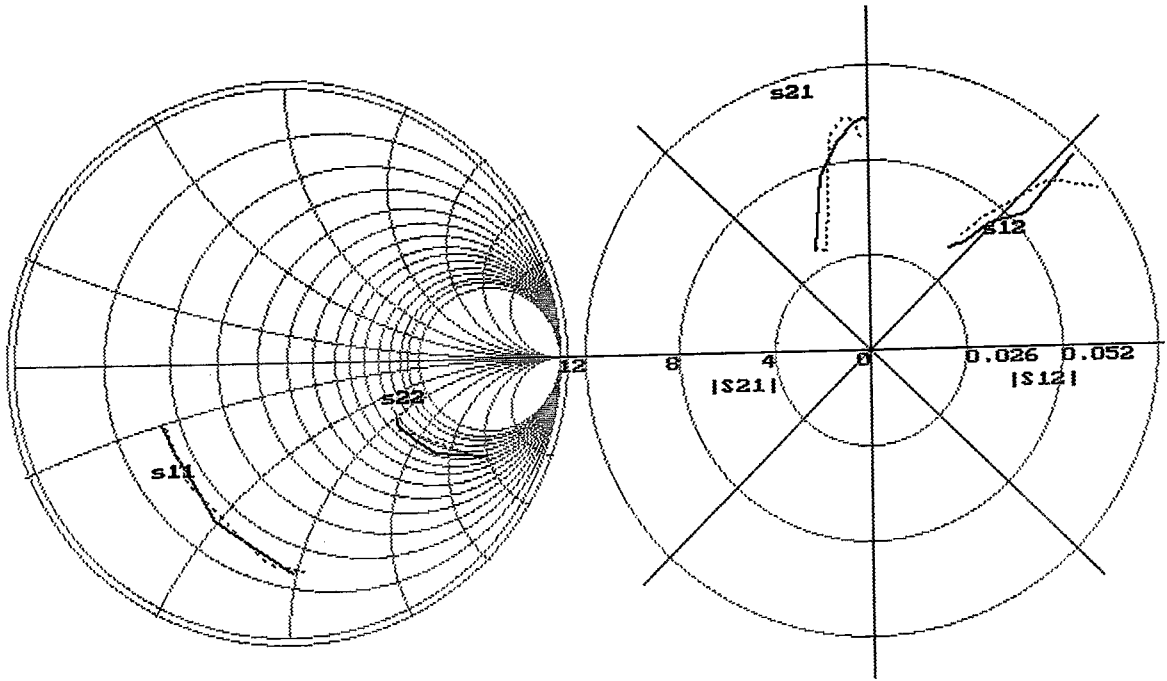




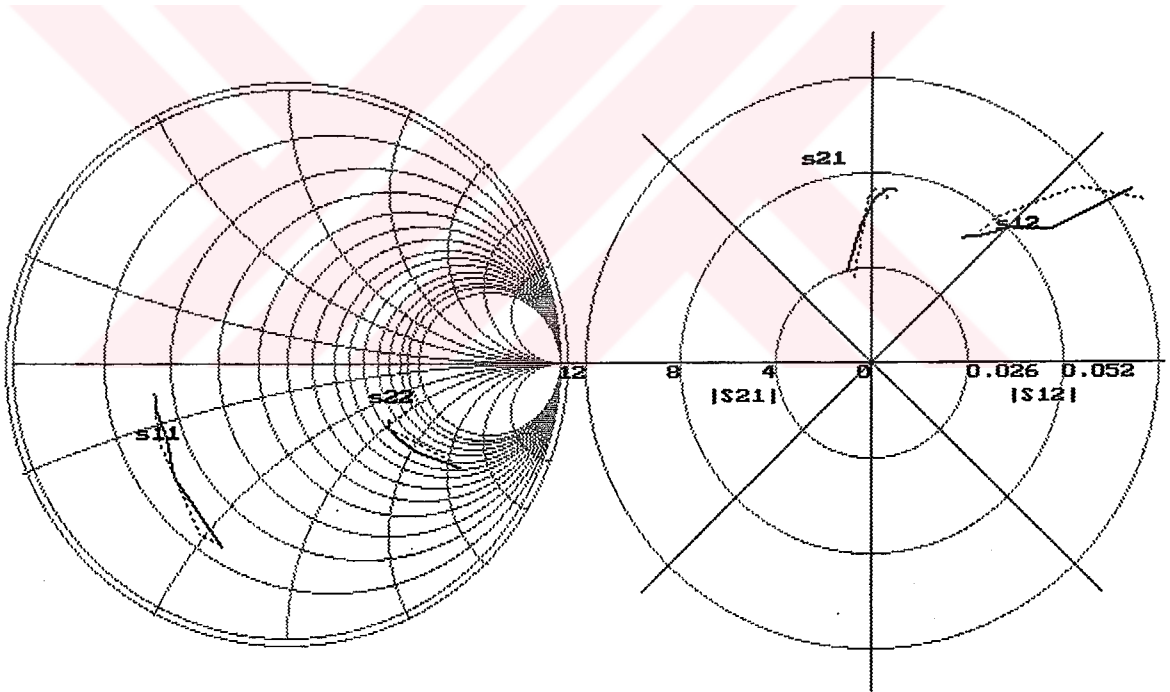
Ölçülen değer — hesaplanan değer - - - - -

Şekil 4.31 a-0.3-b-0.4 GHz frekansında  $V_{CE}=10$  Vsabit ve  $I_C=2-25$  mA değişken ile tanımlı kutuplama noktaları ile NE02135 transistörünün S-parametrelerinin değişimi.





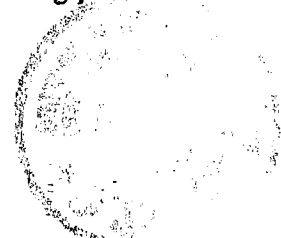
(a)

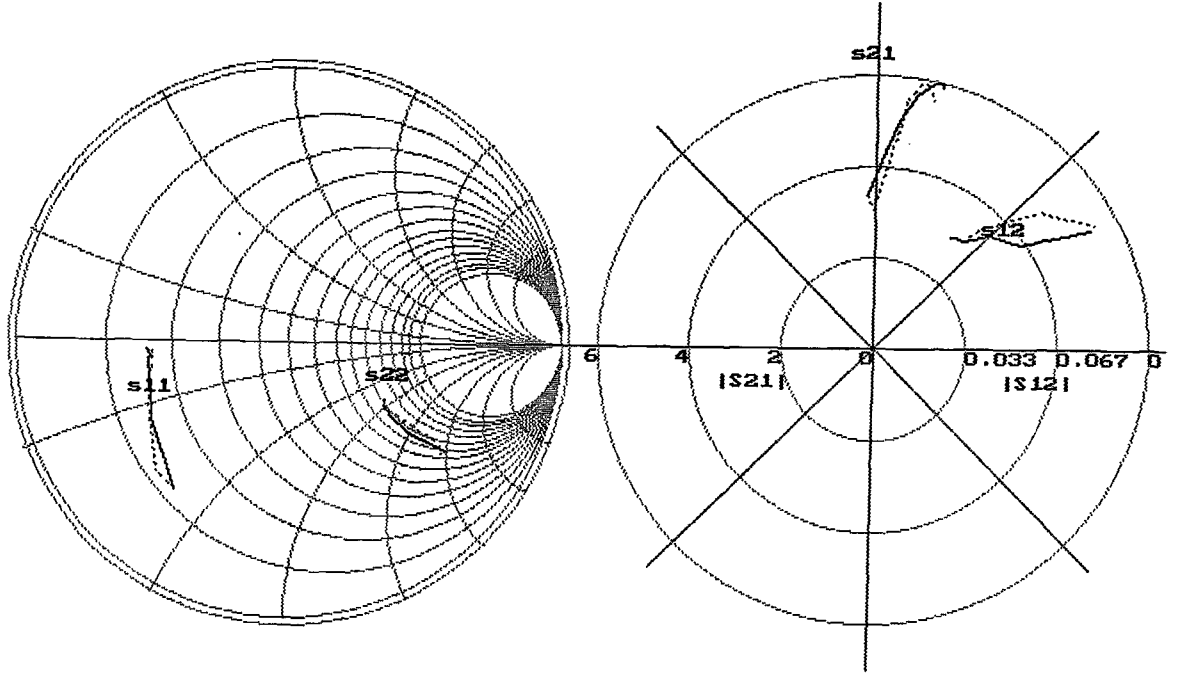


(b)

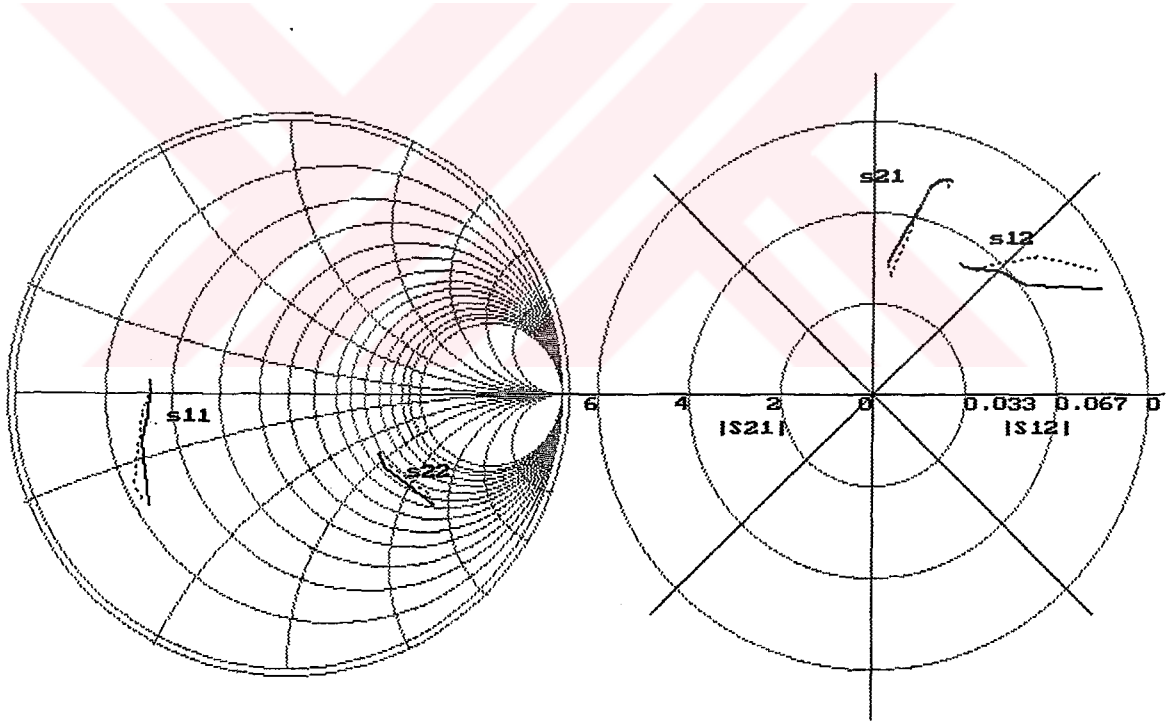
Ölçülen değer — hesaplanan değer - - - -

Şekil 4.32 a-0.6 b-0.8 GHz frekansında  $V_{CE}=10$  Vsabit ve  $I_C=2-25$  mA değişken ile tanımlı kutuplama noktaları ile NE02135 transistörünün S-parametrelerinin değişimi.





(a)

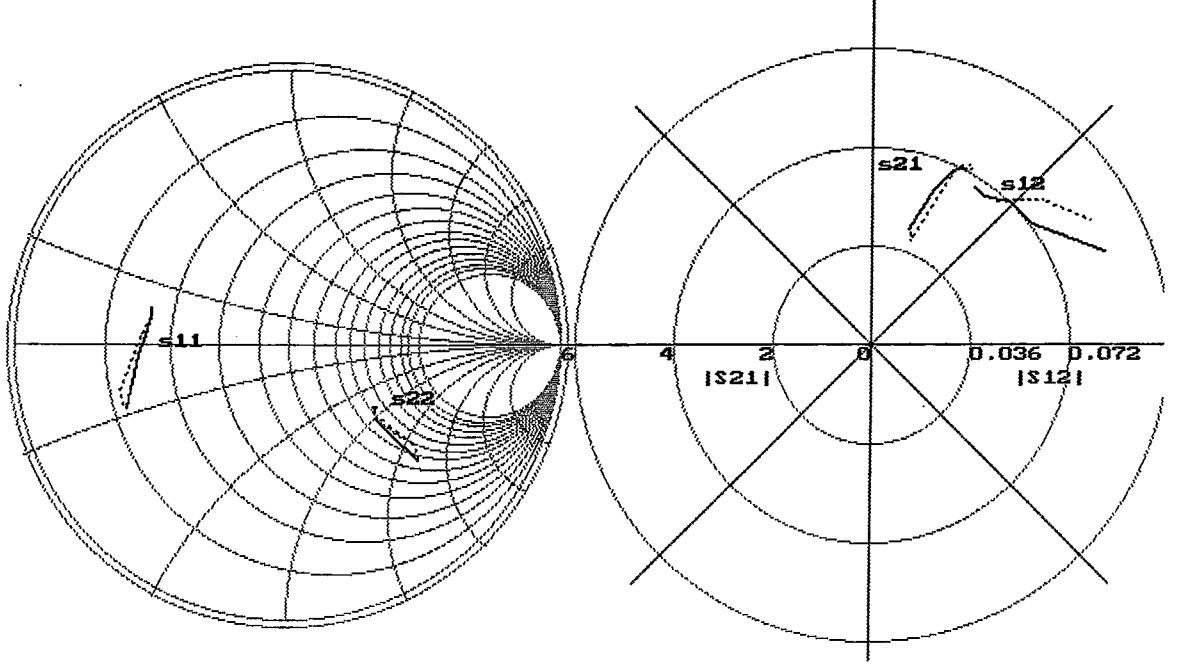


(b)

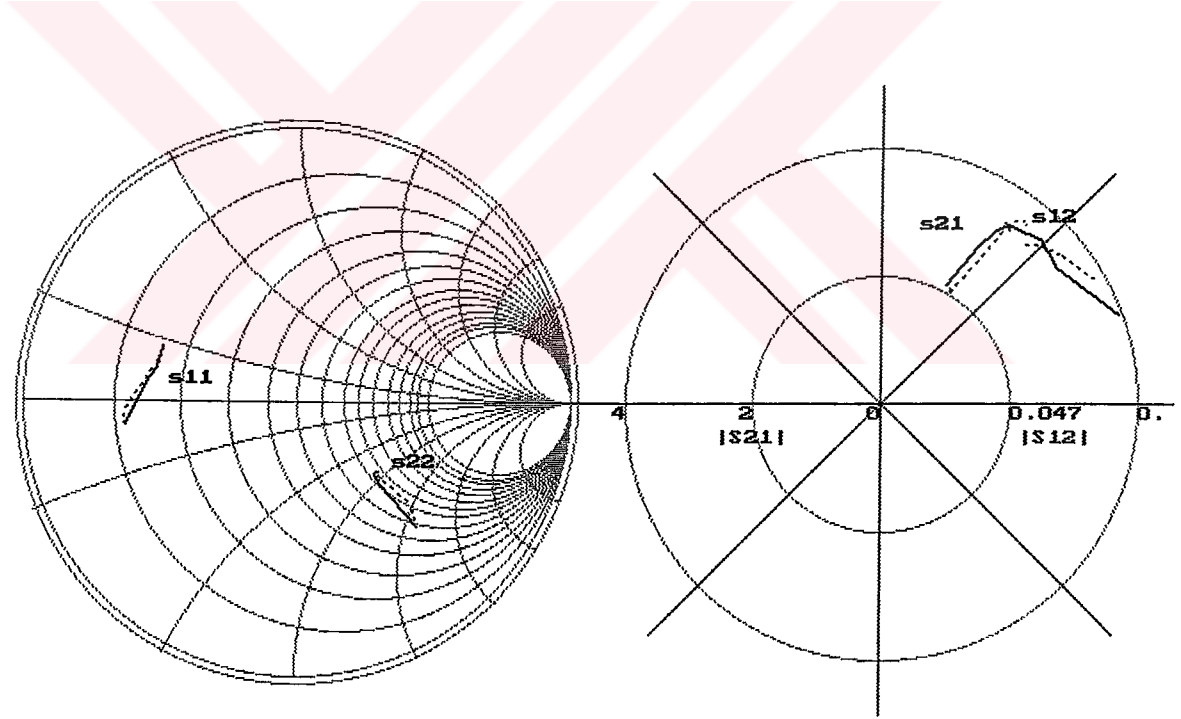
Ölçülen değer — hesaplanan değer - - - - -

Şekil 4.33 a-1.0 b-1.2 GHz frekansında  $V_{CE}=10$  Vsabit ve  $I_C=2-25$  mA değişken ile tanımlı kutuplama noktaları ile NE02135 transistörünün S-parametrelerinin değişimi.





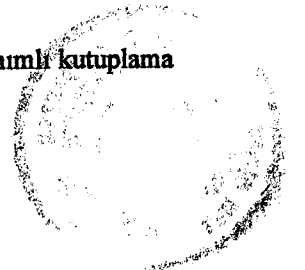
(a)



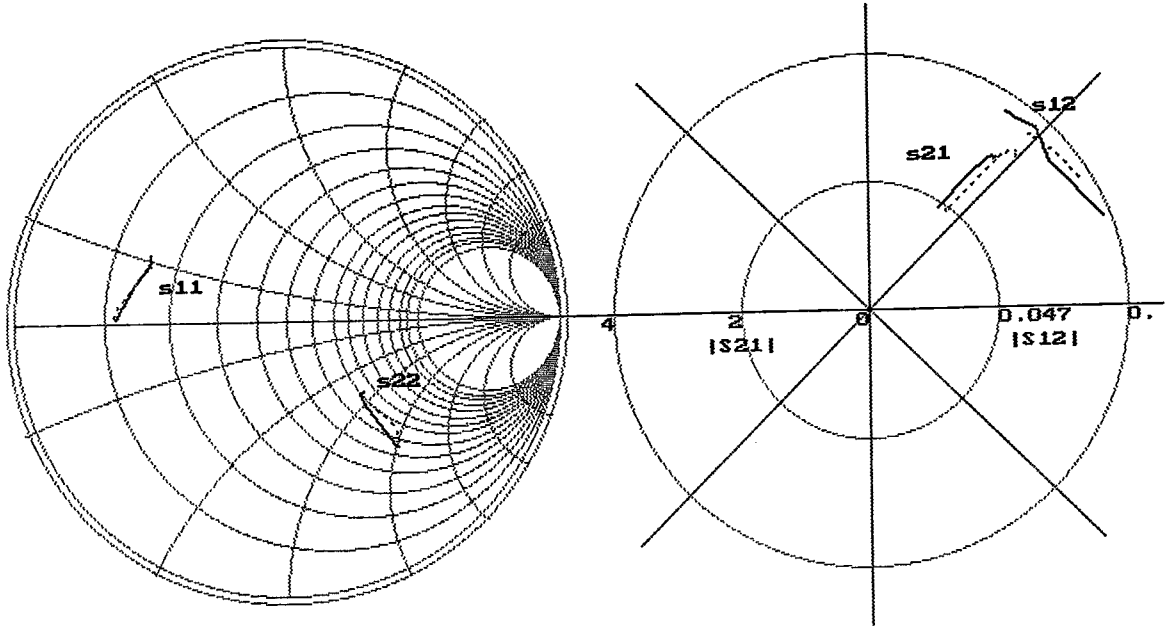
(b)

Ölçülen değer — hesaplanan değer - - - - -

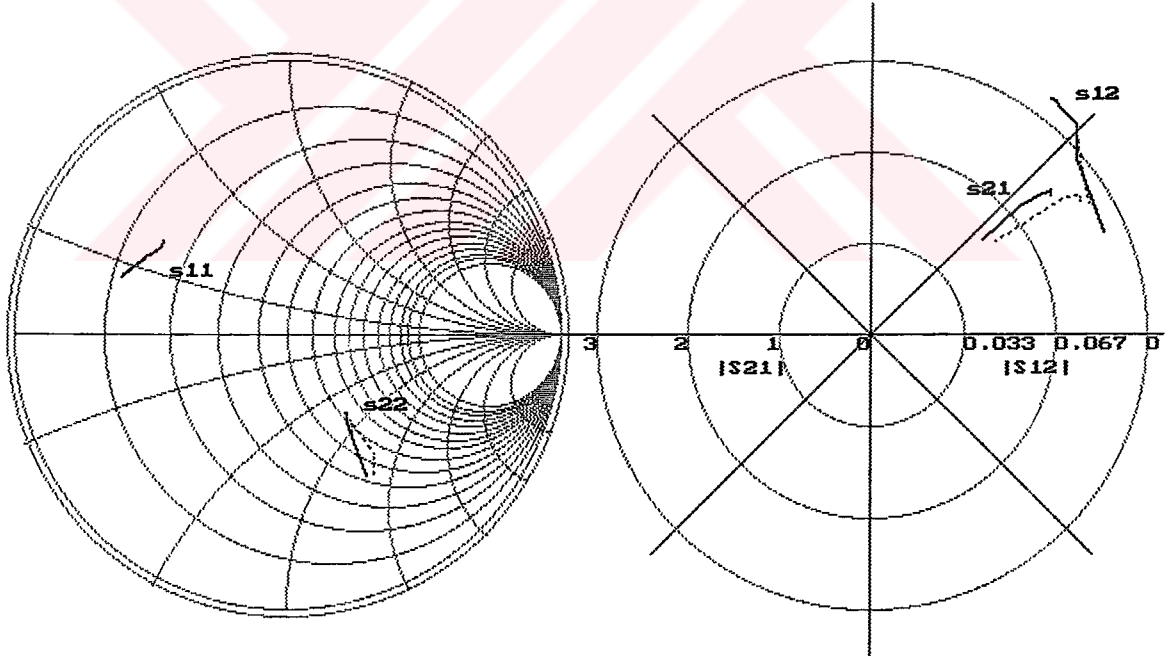
Şekil 4.34 a-1.5 b-1.8 GHz frekansında  $V_{CE}=10$  Vsabit ve  $I_C=2-25$  mA değişken ile tanımlı kutuplama noktaları ile NE02135 transistörünün S-parametrelerinin değişimi.







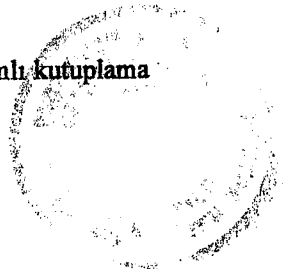
(a)

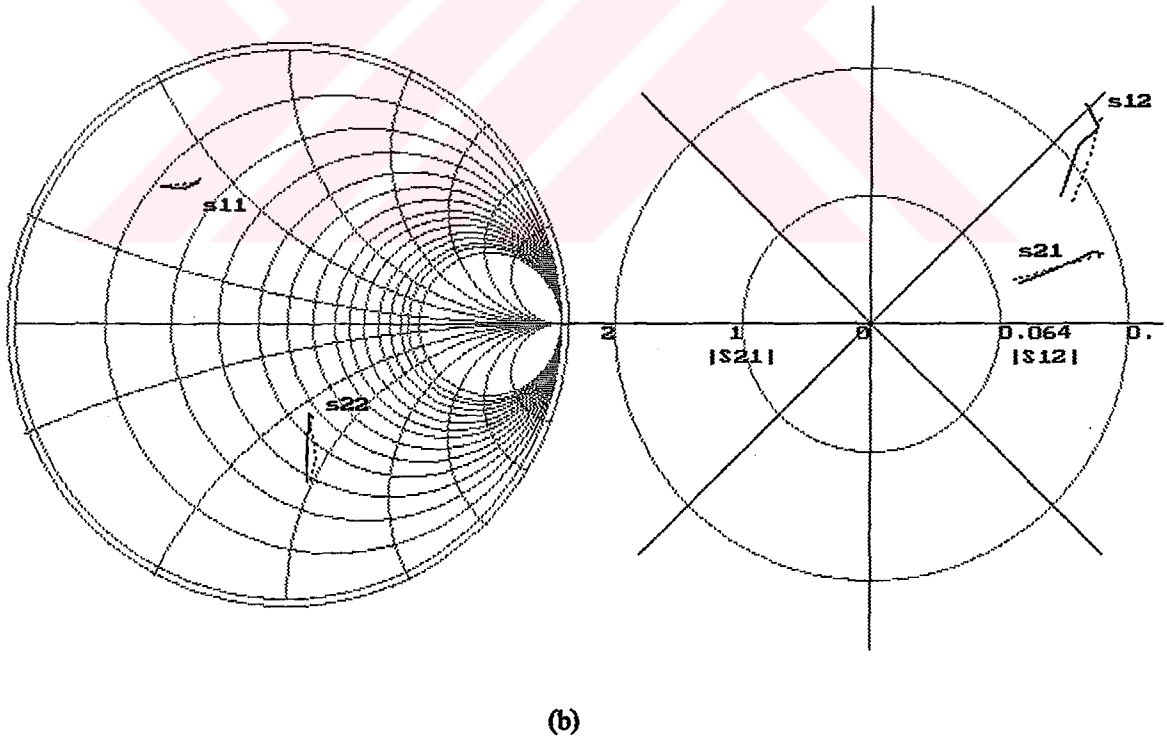
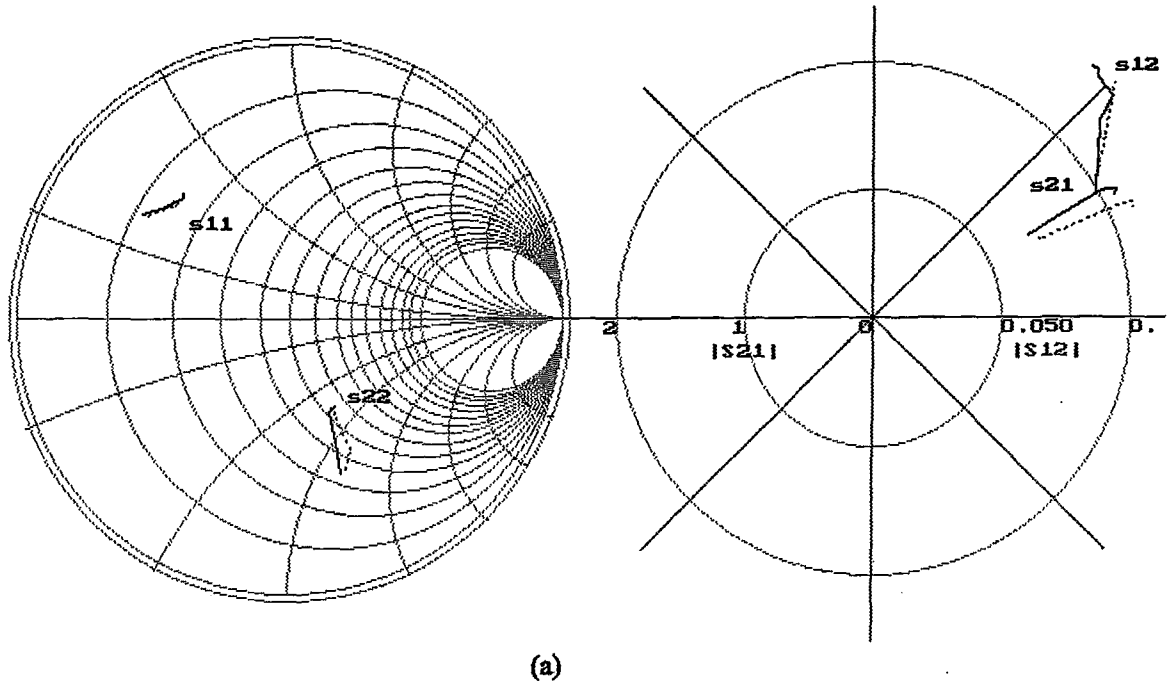


(b)

Ölçülen değer — hesaplanan değer - - - -

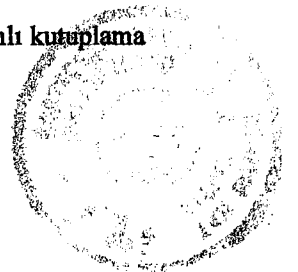
Şekil 4.35 a-2.0 b-2.5 GHz frekansında  $V_{CE}=10$  Vsabit ve  $I_C=2-25$  mA değişken ile tanımlı kutuplama noktaları ile NE02135 transistörünün S-parametrelerinin değişimi.

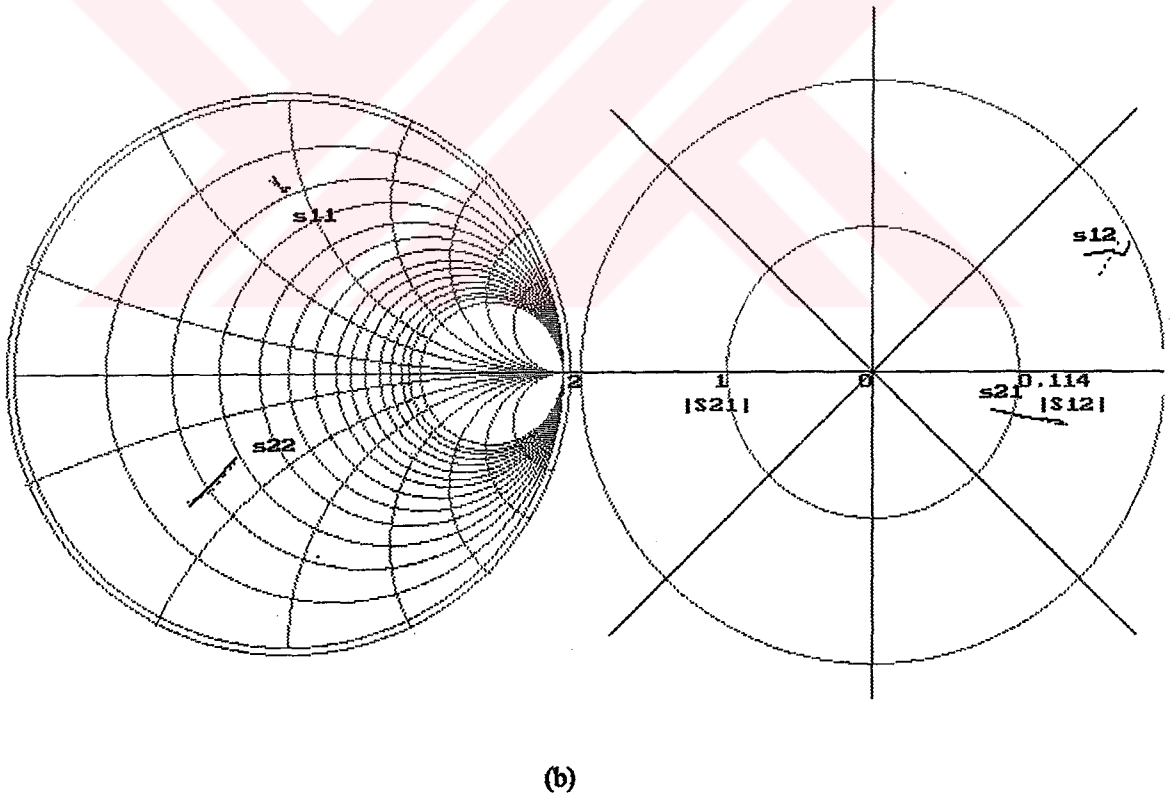
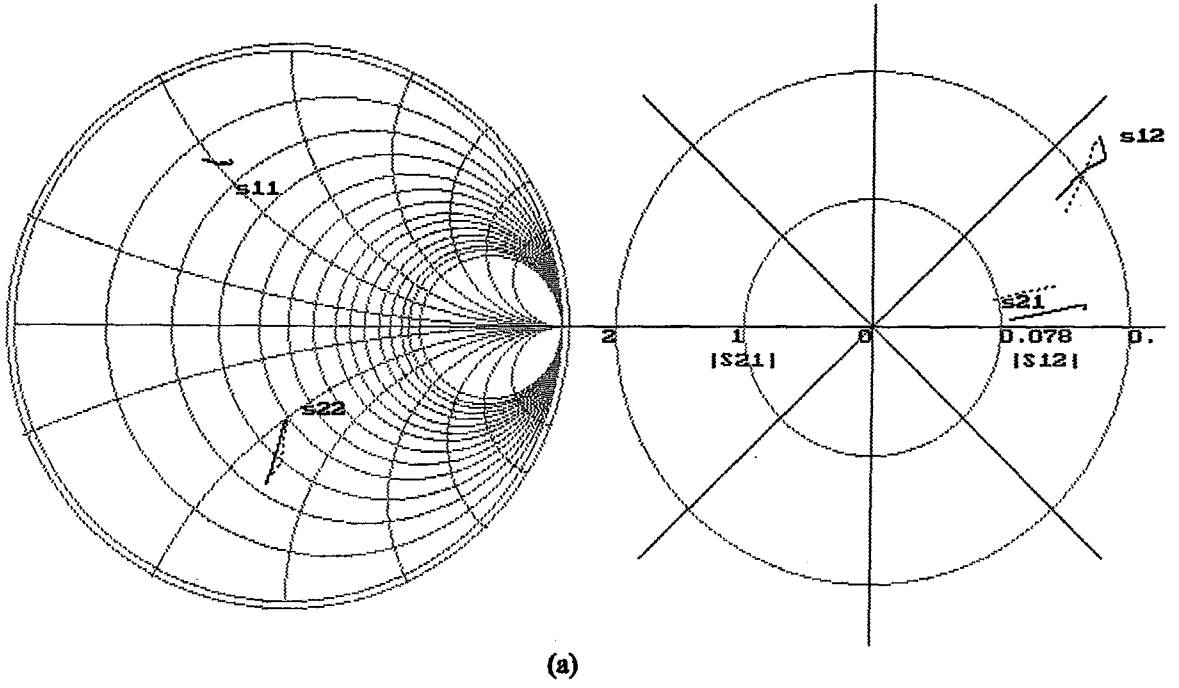




Ölçülen değer — hesaplanan değer - - - - -

Şekil 4.36 a-3.0 b-3.5 GHz frekansında  $V_{CE}=10$  Vsabit ve  $I_C=2-25$  mA değişken ile tanımlı kutuplama noktaları ile NE02135 transistörünün S-parametrelerinin değişimi.



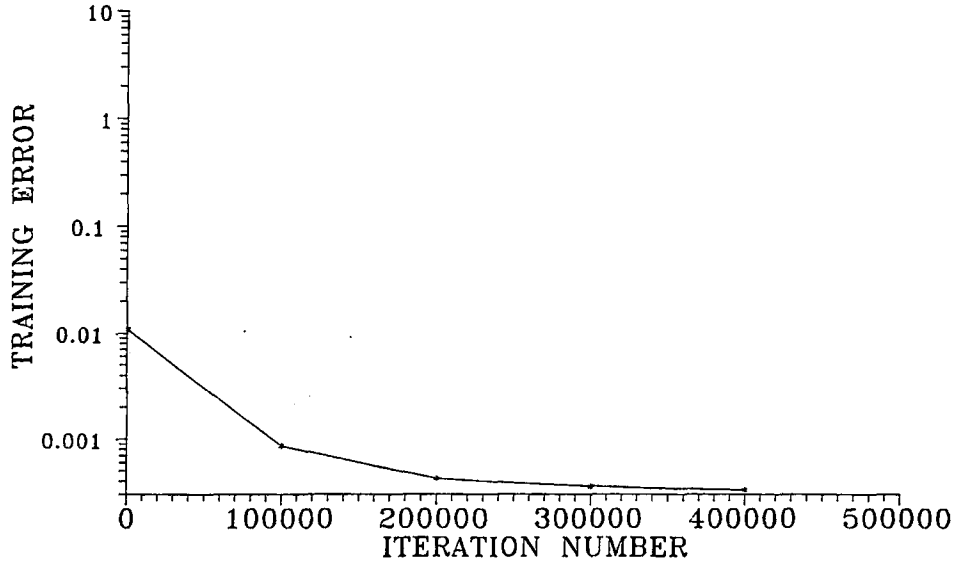


Ölçülen değer — hesaplanan değer - - - - -

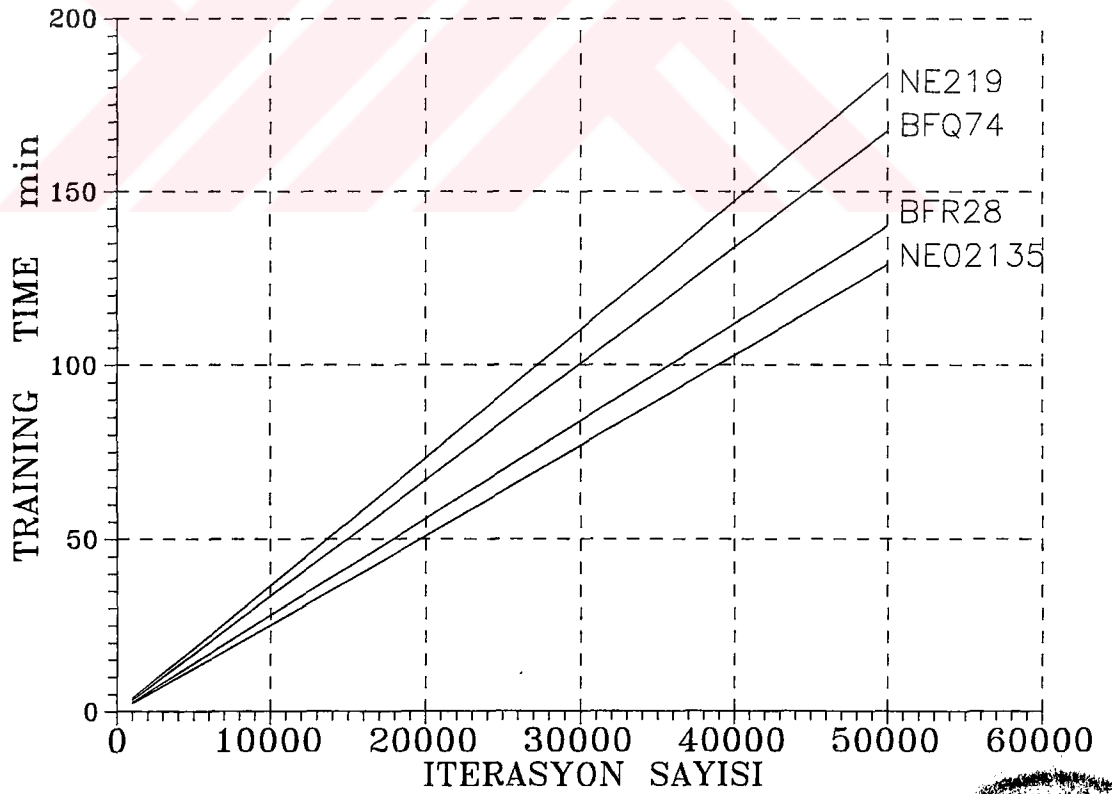
Şekil 4.37 a-4 b-5 GHz frekansında  $V_{CE}=10$  V sabit ve  $I_C=2-25$  mA değişken ile tanımlı kutuplama noktaları ile NE02135 transistörünün S-parametrelerinin değişimi.



#### 4.9 Muhtelif Transistörler İçin Eğitim İçin Gerekli Bilgisayar Zamanının İncelenmesi



Şekil 4.38 Tipik bir transistör tipik öğrenme hatası-iterasyon sayısı değişimi



Şekil 4.39 Muhtelif transistörler için iterasyon sayısı-zaman grafiği



## 5 SONUÇ VE TARTIŞMA

- DC kutuplamaya,konfigürasyon tipine işaret ve gürültü parametrelerinin bağılılığıda ilave edilerek devre tasarımcıları için büyük kolaylık sağlandı.Böylelikle 12 performans parametre (İşaret ve gürültü) fonksiyonunun kutuplama koşulu ,frekans ve konfigürasyon tipine bağılılığı ölçmeyle gayet iyi sayılabilecek uyuşma içinde yaklaşılmıştır.Tek bir eşdeğer devre transistörün işaret ve gürültü performansının bütün çalışma koşulları için kullanılabilir kılınmıştır. Literatürde ise ancak işaret ve gürültü parametrelerini belirleyebilen transistör modelleri mevcuttur.

- Bu sayede de geliştirilen modelin yardımıyla hangi kutuplama noktası civarında çalışayım ki max. Kazanç veya min. gürültü figürü elde edebileyim,aktif elemanın giriş yansıma katsayısı veya çıkış yansıma katsayısı kutuplamayla nasıl değişiyor,kutuplamanın işaret ve gürültü parametrelerine etkisi nedir? ...gibi sorulara bir cevap verme imkanını bize veriyor.

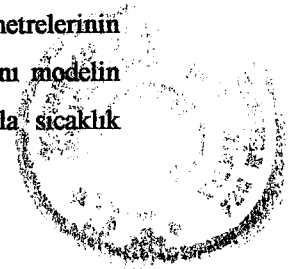
- Transistörün giriş-çıkış karakteristیکlerinin de YSA ile hesaplanabilmesi için oldukça başarılı denemeler yapıldı.(Bakınız Şekil 5.1)

- Bu çalışmamızın devre tasarımcıları için ve özellikle tümleşik mikrodalga devre tasarımında faydalı olacağı kanaatindeyiz.Zira tümleşik mikrodalga devre tasarımında sistem performansını etkileyen parametrelerinin tümünün kestirimi ve simülasyonu çok büyük önem taşımaktadır. Bir mikrodalga transistörü için gerekli tüm bilgiler tek bir YSA eşdeğer modeli üzerinden hesaplanabilecektir. Buna ait bir uygulama örneği olarak EK 2’de (sf. 273-278) verilen ECCTD’97 de sunulan “Neural Network Approach for Active Microwave Devices” adlı çalışmamızı verebiliriz.

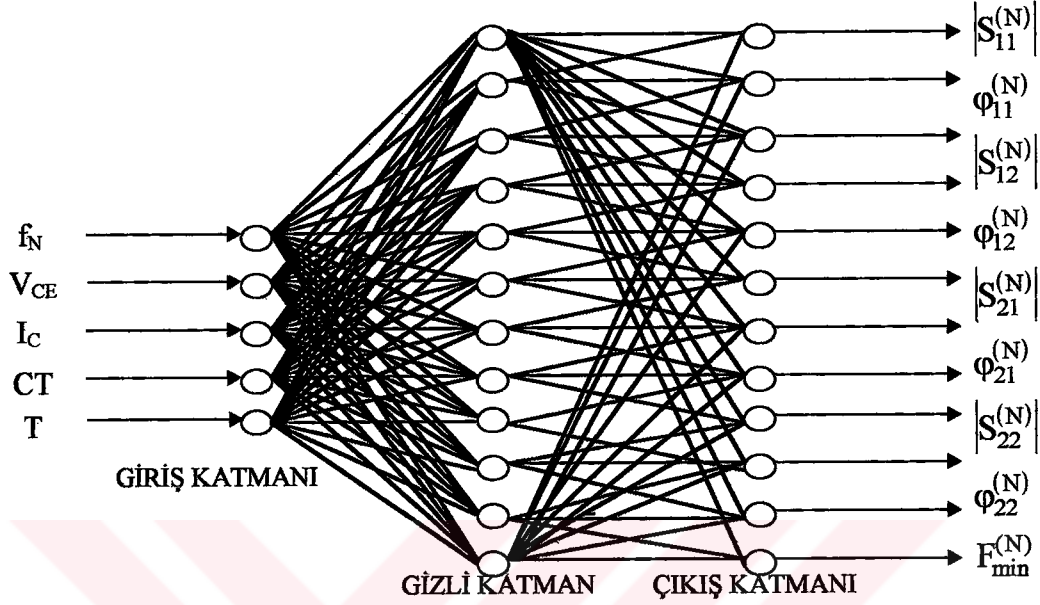
- Bu tezde geliştirilen YSA modeli öncü bir çalışma olup her hangi bir devre elemanına da uygulanabilir.Ancak daha ziyade çok boyutlu modelleme problemi için alternetifsiz bir modeldir.Zira sadece İşaret veya sadece gürültüyü modellemek için (yada her ikisi birarada) sabit kutuplama için modelleme yapan eşdeğer devre modelleri mevcuttur.Bu tezde geliştirdiğimiz model doğrudan ölçülen değerlerden hareketle geliştirilmiştir ve bunlara ilişkin ayrıntılı hata analizleri verilmiştir.Bu sayede diğer mevcut tekniklerle burada geliştirilen model arasında bir mukayese yapma imkanı da mevcuttur.Fakat bu çalışmada mevcut eşdeğer modellerle gerçekleştirilemeyen bir mikrodalga devre elemanının bütün özelliklerinin bir tek eşdeğer devre modeli ile gerçekleştirilmesi yoluna gidilmiştir.

- Bu çalışmada klasik optimizasyon teknikleri yardımıyla eşdeğer devre modeli eleman parametrelerinin kestiriminde başlangıç değerlerinin belirsizliğinden kaynaklanan problemler de bu yeni modellemeyle giderilmiş oluyor.

- Bu tezde geliştirilen model bir simülator (benzetim programı) ile gerçekleşmiştir.(Bakınız EK 1)EK 1 ‘de verilen program çalışmamızın çekirdeğini oluşturmaktadır.Eşdeğer YSA modeli programlamaya dayalı olmasının üstünlüğü giriş ve çıkış büyüklüklerinin kolaylıkla istenilen özellikler doğrultusunda modifiye edilebilmesidir.Örneğin elimizin altında gürültü ve sinyal parametrelerinin kutuplamayla ve frekansla bağılılığının yanında sıcaklıkla değişimine ait veriler olsaydı aynı modelin girişine sıcaklık parametresi için bir giriş ilave etmek suretiyle YSA modeli kolaylıkla sıcaklık



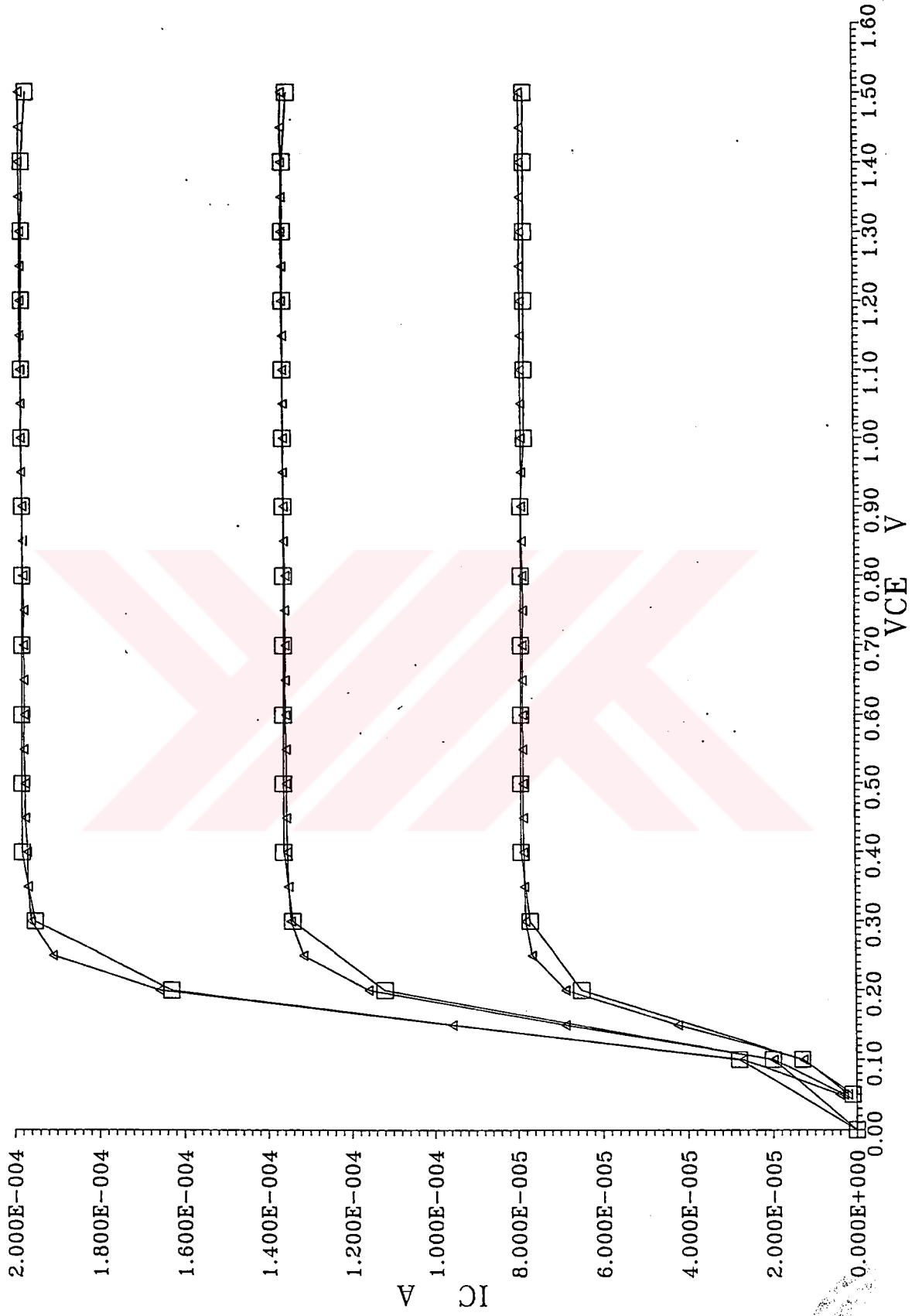
parametresini de içermek suretiyle modifiye edilebilir. Elde edilen yeni modelle devre elemanının sıcaklıkla analizinde yapmak mümkün olur. Örnek olarak verdiğimiz eşdeğer YSA modeline ait mimari yapıda aşağıdaki gibi olacaktır.



Sıcaklık parametresinin eklenmesiyle oluşan değişikliğin program üzerinde nasıl bir modifikasyon yapmak gerektiğini anlamak için EK 1.3.1 'e bir göz gezdirmek yeterli olacaktır. Ayrıca EK 1.1.1 ve EK 1.1.2 'de verilen program kodlarını mukayese ederek gerekli değişim için yapılması gerekenler anlaşılabilir. Bu modifikasyon hedef uzayının modifikasyonundan başka birşey değildir.

- Optimum alpha momentum terimini tesbit edebilmek büyük bir önem arz etmektedir. Aksi takdirde yerel minimuma yakalanma ihtimali artmakta öğrenme hızı ise düşmektedir. Her bir aktif mikrodalga elemanının kendine özgü bir optimum alpha değerinin olduğunu belirtmek gerekir. Optimum alpha momentum terimini belirlemek için basit bir program geliştirilmiştir (EK 1.3).  $\eta$  sıfırdan büyük olmak üzere öğrenme hızı ve  $\alpha$  0 ile 1 arasında değer alan momentum terimidir. Öğrenme katsayısı ise eşdeğer YSA modeli simülatörü içinde otomatik olarak ayarlanmaktadır. Öyleki öğrenme katsayısı terimi hata arttıkça; belirli bir oranda artmakta, buna bağlı olarak öğrenme hızı artmakta, öğrenme hassasiyeti ise düşmektedir. Bunun aksine hata azaldıkça; öğrenme katsayısı yine belirli bir oranda azalmakta, buna bağlı olarak öğrenme hızı düşmekte, öğrenme duyarlılığı ise artmaktadır.  $\eta$  (epsilon) program içinde yandaki formülle tesbit edilmektedir.  $\text{eps} = 0.5 * \text{sqrt}(\text{error})$ ;

- Modelimizin bir üstünlüğü de frekans ve kutuplama gibi sürekli reel sayılarla temsil edilebilecek büyüklüklerle birlikte, konfigürasyon tipi gibi ayrık sayıyla temsil edilebilecek bir parametreyi aynı model içine katılabilmesidir. Konfigürasyon tipinin YSA tarafından algılanabilmesi için [0,1] aralığında ayrık üç sayı tesbit edilmesi gerekir. Bu üç sayının birbirine uzak olması hâliyle konfigürasyon tipinin YSA tarafından ayırdedilmelerini prensip olarak kolaylaştıracağı aşikardır. Yalnız



Şekil 5.1 Eşdeğer YSA modeli ile eğitilen bir transistöre ait  $V_{CE}$ - $I_C$  (Çıkış Karakteristiği) değişimi.

YSA'nın algılama hassasiyeti düşünülecek olursa fazlaca bir önemide yoktur.Örneğin NE219 için konfigürasyon tipinin katılması ile elde edilen hedef uzayı EK 1.3.1'de aşağıdaki gibi verilmiştir.

⋮

begin

case isylla of

```

0: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=2; END;
1: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=2.5;END;
2: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=3; END;
3: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=3.5;END;
4: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=4; END;
5: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=4.5;END;
6: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=5; END;
7: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=5.5;END;
8: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=6; END;
9: BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=2; END;
10:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=2.5;END;
11:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=3; END;
12:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=3.5;END;
13:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=4; END;
14:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=4.5;END;
15:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=5; END;
16:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=5.5;END;
17:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=6; END;
18:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=2; END;
19:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=2.5;END;
20:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=3; END;
21:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=3.5;END;
22:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=4; END;
23:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=4.5;END;
24:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=5; END;
25:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=5.5;END;
26:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=6; END;
27:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=2; END;
28:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=2.5;END;
29:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=3; END;
30:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=3.5;END;
31:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=4; END;

```





```

32:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=4.5;END;
33:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=5; END;
34:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=5.5;END;
35:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=6; END;
end;
:

```

Burada out[0,0]  $V_{CE}$ 'yi,out[0,1]  $I_{CE}$ 'yi, out[0,2] konfigürasyon tipini göstermektedir,out[0,3] ise çalışma frekansını göstermektedir.Bu transistör için out[0,2]'nin 0.2 değeri alması ortak emetörlü olmasını 0.8 değeri alması ise ortak kollektörlü olması durumunu temsil etmektedir.Ortak bazlı olma durumu veri tabanında olmadığı için ortak bazlı konfigürasyonu için ayrı bir sayı tesbit edilmemiştir.

-YSA'nın performansını üstünlüğünü bu arada vurgulamadan geçemeyeceğim.Bölüm 4.3'de N72000a için verilen Tablo 4.6'dan alacağımız aşağıda gösterilen parça üzerinde göstermek istiyorum.

Freq. GHz	Calculated S Parameters				& S <sub>22</sub>	Noise Parameters						
	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>		F <sub>min</sub>	$\Gamma_{opt}$	R <sub>n</sub> /50				
1	0.984	0.942	0.294	0.450	0.209	0.214	0.727	0.966	0.107	0.871	0.044	0.647
2	0.958	0.887	0.290	0.412	0.387	0.181	0.705	0.936	0.130	0.779	0.101	0.511
3	0.924	0.836	0.273	0.369	0.532	0.153	0.672	0.908	0.164	0.711	0.145	0.469
4	0.882	0.790	0.250	0.330	0.651	0.124	0.638	0.884	0.203	0.660	0.177	0.449
5	0.838	0.751	0.229	0.298	0.735	0.101	0.608	0.864	0.240	0.620	0.201	0.434
Freq. GHz	Desired S Parameters				& S <sub>22</sub>	Noise Parameters						
	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>		F <sub>min</sub>	$\Gamma_{opt}$	R <sub>n</sub> /50				
1	0.982	0.942	0.294	0.450	0.210	0.214	0.728	0.967	0.110	0.870	0.042	0.650
2	0.961	0.886	0.291	0.411	0.385	0.181	0.703	0.936	0.120	0.780	0.100	0.500
3	0.923	0.836	0.272	0.369	0.535	0.153	0.673	0.908	0.174	0.710	0.147	0.480
4	0.881	0.792	0.251	0.331	0.650	0.125	0.640	0.883	0.200	0.660	0.178	0.450
5	0.839	0.750	0.229	0.297	0.735	0.100	0.606	0.864	0.240	0.620	0.200	0.430

iteration= 200000 eps= 0.002416 error\_1= 0.0000023 error\_t= 0.0000023

error\_t burada bütün hedef uzayı üzerinden ortalama karesel hatayı göstermektedir.Bu ortalama olarak hedef uzayının her bir elemanının nümerik olarak virgülden sonra 5 hanesini hatasız olarak tesbit edebildiğini göstermektedir.Örneğin 3 GHz'de YSA öğrenmesi gereken 4 parametreyi aynen , 4 parametreyide onbinde birlik bir hatayla öğrenebildiği görülmektedir.Burada onbinde bir olarak gözükken bu hata gerçekte daha da düşüktür.Bunun böyle gözükmesinin nedeni virgülden sonra altı hanenin gösterilmesine sayfa genişliğinin müsaade etmemesidir.Geri kalan diğer dört parametrede oldukça iyi yaklaşıklarla öğrenebilmektedir.Zaten 0,0000023 değeri tek başına bu bilgiyi vermektedir.Bu rakam bize YSA'nın hedef uzayını yaklaşık milyonda ikilik bir hatayla öğrenebildiğinin göstergesidir.Bu zannedirim klasik eşdeğer devre modelleri ile hayal edilemeyecek bir rakamdır.



Yalnız bu kriter YSA'nın performansının üstünlüğünü göstermekle birlikte gerçekte aktif mikrodalga elemanın performansını doğrudan temsil etmemektedir. Zira aktif mikrodalga elemanına ait hedef uzayındaki gerçek değerleri değil bunlara tekabül eden normalize değerleri sözkonusu doğrulukta öğrendiğini göstermektedir. Zira hedef uzayındaki aktif mikrodalga elemanın performansını temsil eden tüm parametreler sıfırla bir arasına normalizedirler. Zira aktivasyon fonksiyonunun çalışma bölgesi 0 ile 1 arasındadır. Denormalizasyon sırasında bu hata denormalizasyon katsayıları kadar artmaktadır. Normalizasyon işlemi simülasyonumuzda "hedef\_belle" prosedürünün içinde görülebileceği ve de aşağıdaki program parçasından anlaşılacağı gibi lineer bir normalizasyon işleminden geçirilmektedir. Burada  $dess[k,l]$  hedef uzayının  $k$ . satırındaki  $l$ . parametreyi temsil etmektedir.  $l=0..11$ 'e giderken  $dess[k,l]$  sırasıyla  $S_{11}$ ,  $S_{21}$ ,  $S_{12}$ ,  $S_{22}$ ,  $F_{min}$ ,  $\Gamma_{opt}$ ,  $R_{ref}/50$ 'i temsil etmektedir. Şu halde  $l=1,3,5,7$  s parametrelerinin açılış değerlerini,  $l=10$  ise  $\Gamma_{opt}$ 'un açılış değerini temsil etmektedir.  $S_{11}$  ve  $S_{22}$  ise zaten her zaman  $[0,1]$  aralığında değerler aldığı için normalizasyon işlemine tabi tutulmamışlardır. Şu halde  $l=2$   $S_{21}$  genliğini  $l=4$  ise  $S_{12}$  nin genliğini temsil etmektedir. Sona kalan  $l=8$  ise  $F_{min}$ 'i temsil etmektedir.

```

:
for l:= 0 to output_nodes-1 do
  begin
    case (l) of
      1,3,5,7,10:begin
        if dess[k,l] < 0 then
          begin
            dess[k,l]:=dess[k,l]+360;    dess[k,l]:=dess[k,l]/360;
          end
        else
          begin
            dess[k,l]:=dess[k,l]/360;
          end;
        end;
      4:dess[k,l]:=dess[k,l]*s12max;
      2:dess[k,l]:=dess[k,l]/s21max;
      8:dess[k,l]:=dess[k,l]/fminmax;
    end;
  end;
:
  - Denormalizasyon ise EK 1.1'de "CALCULTE_SF_BIAS_CT" içerisinde aşağıda
alıntıladığımız program bölümünde görüldüğü gibi yapılır.
:
for l:= 0 to outputs-1 do

```



```

begin
  str(out[2,1]:5:3,st1);
  case (l) of
    1,3,5,7,10:begin
      if (out[2,1] > 0.5) then
        begin
          out[2,1]:=out[2,1]*360-360;          str(out[2,1]:7:2,st1);
        end
      else
        begin
          out[2,1]:=out[2,1]*360;          str(out[2,1]:7:2,st1);
        end;
      end;
    4:begin
      out[2,1]:=out[2,1]/s12max;          str(out[2,1]:5:3,st1);
    end;
    8:begin
      out[2,1]:=out[2,1]*Fminmax;          str(out[2,1]:5:3,st1);
    end;
    2:begin
      out[2,1]:=out[2,1]*s21max;          str(out[2,1]:5:3,st1);
    end;
  end;
end;

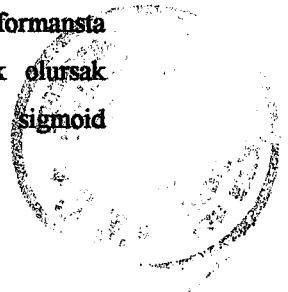
```

:

Buradan da anlaşılacağı üzere mesala açılarının denormalizasyonu sırasında 360 ile normalize değerler çarpılmak suretiyle gerçek değerlere varılmaktadır. Bu denormalizasyon işlevinin tabii sonucu olarakta gerçek düzlemdeki hatalar artmaktadır. Aynı örnek transistör için tablo 4-7’de görülebileceği gibi gerçek mutlak hata S-parametreleri için  $E_T=0.004422$  ve gürültü parametreleri için  $F_T= 0.018070$  olmaktadır.

- Eşdeğer YSA modelimizle elde edilen sonuçlar modifiye edilebilir. Eğer YSA’dan elde ettiğiniz performans yeterli olmuyorsa eğitime kaldığınız yerden devam etmek suretiyle performans iyileştirilebilir. Bunu için eğitim sonucu elde edilen ağırlıkların saklanması gerekir. (Bu işlem “write\_weights” prosedürü ile yapılır.) Deneme yaptığımız transistörler için eğitim sonucu elde edilen ağırlıklardan yararlanılarak bir veri tabanı oluşturulmuştur.

- Denemelerim sonucu sigmoid tip aktivasyon fonksiyonu ile optimum performansta modelleme yapılabildiği sonucuna vardım. Sigmoid fonksiyonunu yeniden hatırlayacak olursak  $y:=1/(1+\exp(-\beta*(x)))$  şeklinde yazılabilir. Buradaki beta parametresi yardımıyla sigmoid



fonksiyonunun eğimi değiştirilerek eğitimler yapılmıştır. Bunların sonucunda  $\beta=1$  için optimum öğrenmenin gerçekleştiği tesbit edilmiştir.

- Tanjant hiperbolik tip aktivasyon fonksiyonu bazı aktif mikrodalga elemanları için çok hızlı bir şekilde hedef uzayını öğrendiği görülmüştür.(örneğin N32684AA için)Fakat diğer aktif mikrodalga elemanları için aynı performansı gösterememiştir.Muhtemeldir ki yerel minimuma yakalanmıştır.Sigmoid tip aktivasyon fonksiyonu ile çalışmada böyle bir sorunla karşılaşılmamıştır.Bu PALUBINKAS'ın ifade ettiği aksine bir sonuç olmaktadır.

- Ağırlıkların başlangıç değerleri YSA literatüründe rasgele olarak başlatılmakta fakat bunun sınırları ile ilgili kesin bir bilgi bulunmamaktadır.Bizim çalışmamızda da ağırlıkların başlangıç değerleri  $[-0.3,0.3]$  arasında değişen rasgele sayılarla başlatılmıştır.Eğitme sonucu elde edilen ağırlık değerleri gözden geçirildiğinde ağırlık değerlerinin genellikle sıfır civarında olmasıda seçimimizin doğruluğunu göstermektedir.

- Aktivasyon fonksiyonu olarak Gauss dağılımını kullanan (Radial Basis Function) olarak adlandırılan tipteki yapay sinir ağırları ile de bazı denemeler yaptım.Ancak iyi bir öğrenme sağlanamadı.Daha çok belli bir frekans noktası civarında kümelenme oluştu.Yani sadece bir frekans için elemanın performansını öğrenebilmekte diğer frekanslarda da sözkonu frekansa ait sonuçlar tekrarlanmaktaydı.Yalnız bu sonucun elimizdeki RBF simülatörünün sınıflamaya yönelik geliştirilmesinden kaynaklanabilir.

- YSA 'ları bilindiği gibi ayrı bir bilim disiplini olarak hızla gelişmeye devam etmektedir.Yapay sinir ağı modelimiz tabii ki yapay sinir ağılarındaki gelişmelere paralel olarak geliştirilebilecektir.Henüz YSA teorisi yerli yerine oturmamış olup birbiriyle çelişkili sonuçlar ortaya çıkabilmektedir.Zamanla bu problemler çözüldükçe YSA' nı daha etkin bir şekilde kullanmak mümkün olabilecektir.Şimdi yapılan çalışmalar deneme yanılmaya metoduyla geliştirilmektedir.

- Çok geniş bir bandda modelimizin gerçek değerlerle iyi bir uyum içinde olduğunu dördüncü bölümde verdiğimiz uygulamalardan görebiliriz.

- Tabii ki hedef uzayı genişledikçe eğitim süresi uzayacaktır.

- Hedef uzayının belirlenmesi eşdeğer YSA modelinin başarımını doğrudan etkilemektedir.hedef uzayını oluşturan parametrelerin değişim hızına göre örnekleme alınacak nokta sayısı değişecek; hızlı değişim olduğu bölgelerde çok örnek, yavaş değişim olan bölgelerde az örnek almak gerekmektedir.Bunun aksine rasgele örnekler alınarak teşkil edilen hedef uzayı YSA modelinin performansını bozacaktır.(Aşırı örnekleme yapmak veya yetersiz örnekleme yapmak)Bunun için Şekil 4.12 ve Şekil 4.13'e bakınız.Görülebileceği gibi hedef uzayının belirlenmesi için optimum bir örnekleme sayısı mevcuttur.Bunun tesbit edilmesi eşdeğer YSA modelinin performansını artıracığı aşikardır.Bölüm 4'te verilen bir çok aktif mikrodalga elemanında böyle bir incelemeyi yaptıktan sonra eğitmeye geçme imkanı olmadığından eldeki verilerle eğitim işlemi yapılmıştır.Bu şartlar altında alınan sonuçlar oldukça ümit vericidir.

- Belirli bir kutuplama noktası civarında işaret ve gürültü parametrelerinin öğretilmesi için bir iki saat yeterli olurken ;kutuplama noktasıda ve konfigürasyon tipi değişken olarak YSA eşdeğerini eğitmek için beş altı saat gerekmektedir.Bu süreler 486 DX-40 MHz PC kullanılarak elde edilmiştir.Pentium bir PC ile bu süre bir kaç dakikaya kadar düşebilecektir.(Şekil 4.38 ve Şekil 4.39'a bakınız)

- Geliştirdiğimiz yapay sinir ağı eşdeğerinin tabii olarak hedef uzayının teşkil edildiği frekans bandı veya kutuplama noktası dışındada cevap verme yeteneği olduğu gösterilmektedir(Extrapolasyon ve İnterpolasyon).(Bakınız Bölüm 4'deki örnekler)

- Optimum öğrenme gizli katman düğüm sayısı çıkış katmanı düğüm sayısına eşitken elde edilebildi.(Bakınız Şekil 3.7) Bu sonuçlar BURR,D.J. \* (1992) çalışmasında belirttiği gizli katman düğüm sayısını(N) sırasıyla giriş ve çıkış katmanı düğüm sayılarının ( $N_i, N_o$ ) fonksiyonu olarak veren  $N \geq (2N_i N_o - N_o) / (N_i + N_o + 1)$  formülüyle ters düşmemekle birlikte bire bir aynı sonucuda vermemektedir.Tabii ki gizli katman düğüm sayısı ile ilgili bu sonuç fonksiyon yaklaşımı uygulamaları için geçerlidir.Mesala veri sıkıştırma probleminde gizli katmandaki düğüm sayısı minimum olmalıdır ki maksimum sıkıştırma gerçekleştirilebilsin.Zira gizli katman düğümlerinin çıkışları ile sıkıştırılmış data saklanmaktadır.Bu tip uygulamalarda giriş katmanına uygulanan işaret çıkış katmanından aynen elde edilmesi prensibine dayanır. Dolayısıyla giriş ve çıkış katmanı düğüm sayısı birbirine eşittir.

- Bundan sonraki çalışmamızda istenilen gürültü ve sinyal parametrelerinden hareketle tümleşik mikrodalga devre elemanının geçit genişliği,kanal boyu,kullanılan tabakanın dielektrik sabiti... vs. gibi fiziki parametrelerinin de YSA eşdeğeriyle hesaplanmasına çalışacağız.Buda eleman fiziğinin elektriksel etkilerinin hesaplanmasına dayanan oldukça karmaşık olan(1.1.1.2'ye bakınız) fiziksel temelli modellemeye alternatif bir YSA modeli geliştirilmesi üzerinde incelemelerimiz devam etmektedir. (Reverse modeling)Fizik temelli modellemede eleman fiziğini ifade eden denklemlerin karmaşıklığından dolayı bir çok yaklaşımlar altında denklemler çözülebilmektedir.Bu ise modellemenin hatasını arttırmaktadır.

\* BURR,D.J. :”Comparison of Gaussian...”,IEEE ICA SSP Vol.2,1992



## EK.1 AKTİF MİKRODALGA ELEMANLARI İÇİN EŞDEĞER YSA SİMÜLATÖRLERİ

### EK.1.1 Çeşitli Kutuplama ve Çeşitli Konfigürasyon Tipi için Gürültü ve Sinyal Parametreleri Simülatörü

```
PROGRAM MULTILAYERPERCEPTRON;{DİKKAT AYREAD VE YREAD SEÇİMİNİ YAPMAYI
UNUTMA} {SF_BIAS_Multi Configuration} {$N+}
```

```
uses dos,crt,printer;
```

```
CONST
```

```
Layers_Max = 3; { Maximum number of layers }
Nodes_Max = 12; { Maximum number of node in any layer }
Outputs_Max = 14; { Number of input combinations }
Input_Nodes = 4; { Number of input layer nodes }
Output_Nodes = 12; { Number of output layer nodes }
Hidden_Nodes = 4; { Number of hidden layer nodes }
Epsilon = 0.5;
Alpha = 0.92;
Iterative_Max = 600000; { Maximum iteration number }
Every = 600000; { After every iteration print results }
Fminmax = 4.5;
s21max = 36.0; { Normalization coefficient40}
s12max = 4.8; { Normalization coefficient3.7}
```

```
TYPE
```

```
adizi = array [ 0..outputs_Max-1 ] of real;
```

```
VAR
```

```
h1,h2,m1,m2,s1,s2,hund1,hund2 : Word;
```

```
f:adizi;
```

```
count,j:longint;
```

```
samplecount,i,ii,tt,isylla, num,dcount:byte;
```

```
fmax,fmin,step,eps, alp, err, error, terr, terror, total_rate,des,beta: REAL;
```

```
Layers, Inputs, Outputs : byte;
```

```
Node : array [0..Layers_Max-1] of integer;
```

```
matrix : array [0..Outputs_Max-1, 0..(Input_Nodes+Output_Nodes-1)] of REAL;
```

```
matrix_tmp: array [0..trunc(Iterative_Max/Every), 0..Outputs_Max-1,0..(Input_Nodes+Output_Nodes-1)] of REAL;
```

```
out : array [0..Layers_Max-1, 0..Nodes_Max-1] of REAL;
```

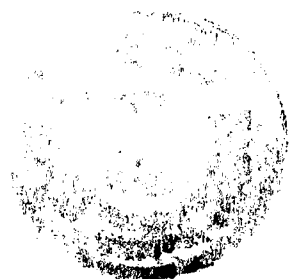
```
der : array [0..Layers_Max-1, 0..Nodes_Max-1] of REAL;
```



```

theta : array [0..Layers_Max-1, 0..Nodes_Max-1] of REAL;
dtheta : array [0..Layers_Max-1, 0..Nodes_Max-1] of REAL;
weight : array [0..Layers_Max-1, 0..Nodes_Max-1, 0..Nodes_Max-1] of REAL;
dweight : array [0..Layers_Max-1, 0..Nodes_Max-1, 0..Nodes_Max-1] of REAL;
dess : array [0..Outputs_Max-1, 0..output_nodes-1] of REAL;
st1: string[7]; st5, st6: string[6];
d, st2, st3, st4: string[12];
weightfile, idatafile, datafile, graphfile, outfile :text;
tercih, pass:char;
ad, hiz: string[8];
PROCEDURE ARRANGE;
var
  dosya, ydosya: text;
  A: string[1];
  B: string[10];
  q: string[80];
begin
  write(' OKUNACAK DOSYANIN ADINI GİRİNİZ. '); READLN(d);
  assign(dosya, 'c:\appcad\spara\necspar'+d+'.s2p'); assign(ydosya, d+'.dat'); reset(dosya);
  repeat
    readln(dosya, q); a:=copy(q, 1, 1); b:=copy(q, 2, 11);
  until (a='Z') or (a='z');
  rewrite(ydosya);
  while not eof(dosya) do
    begin
      readln(dosya, q); a:=copy(q, 1, 1);
      if (a='Z') or (a='z') then
        begin
          b:=copy(q, 2, 11); readln(dosya, q);
        end;
      writeln(ydosya, b, ' ', q);
    end;
  close(ydosya); close(dosya);
end; {Procedure arrange}
PROCEDURE HEDEF_BELLE;
type
  dizi=array[0..outputs_max-1, 0..output_nodes] of real;

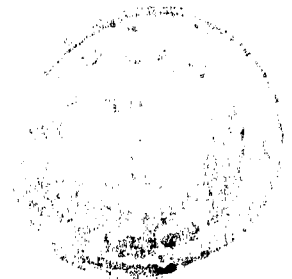
```



```

var
  tampon:text;
  yanıt:char;
  fdess:dizi;
  x,y:real;
  k,l,m,n:byte;
  ST:string[8];
begin
  assign(tampon,'c:\tp\hamid\n2135tmp.dat'); {assign(tampon,'tampon.dat');}
  write(' EN SON KULLANILAN FREKANS ÖRNEKLERİNİ KULLANMAK İSTER MİSİNİZ?
[E/H]');readln(yanıt); yanıt:=upcase(yanıt);
  if yanıt='E' then
  begin
    reset(tampon);
    for l:=0 to outputs_max-1 do
      begin
        READ(tampon,F[l]);
      end;
    end
  else
  begin
    rewrite(tampon); writeln(' HANGİ FREKANSLARDA ÖRNEK ALACAKSANIZ ');
    writeln(' SIRAYLA [GHz] OLARAK GİRİNİZ ');
    for l:=0 to outputs_max-1 do
      begin
        write(' ');write (l+1);write(' ÖRNEK = ');READ(F[l]);write(tampon,f[l]);write(' GHz de');
        writeln;
      end;
    writeln(' ALINACAKTIR. ');
  end;
  close(tampon); assign(datafile,d+'.dat'); reset(datafile);m:=0;
  assign(idatafile,'i'+d+'.dat'); rewrite(idatafile);
  while not eof(datafile) do
  begin
    for n:=0 to input_nodes-1 do
      read(datafile,x);
    for l:=m to outputs_max-1 do

```





```

begin
  while not (x=f[l]) do
    begin
      while not eoln(datafile) do
        read(datafile,x);
        for n:=0 to input_nodes-1 do
          read(datafile,x);
        end;
      for k:=0 to output_nodes do
        begin
          fdess[m,k]:=x;      read(datafile,x);
        end;
      for k:=0 to output_nodes do
        begin
          y:=fdess[m,k];str(Y:6:3,ST);      write(idatafile,' ',st);
        end;
      writeln(idatafile,' ');      m:=m+1;
      for n:=0 to input_nodes-2 do
        read(datafile,x);
      end;
    end;
  close(datafile);close(idatafile);
  for m:=0 to outputs_max-1 do
    for k:=0 to output_nodes-1 do
      dess[m,k]:=fdess[m,k+1]; clrscr;
    for k:=0 to outputs_max-1 do
      for l:= 0 to output_nodes-1 do
        begin
          case (l) of
            1,3,5,7,10:begin
              if dess[k,l] < 0 then
                begin
                  dess[k,l]:=dess[k,l]+360;  dess[k,l]:=dess[k,l]/360;
                end
              else
                begin
                  dess[k,l]:=dess[k,l]/360;

```



```

        end;
    end;
    4:dess[k,l]:=dess[k,l]*s12max;
    2:dess[k,l]:=dess[k,l]/s21max;
    8:dess[k,l]:=dess[k,l]/fminmax;
end;
end;
end; {Procedure hedef_belle}
PROCEDURE WRAND;
{ Initializes node offsets & weights at the beginning of the program }
var
    li, ni, ni_end, no, no_end : integer;
    drand48, power: REAL;
begin
    { Randomize offsets }
    for li:=1 to Layers_Max-1 do
    begin
        ni_end:=node[li];
        for ni:=0 to ni_end-1 do
        begin
            drand48:=Random(30)/100;    theta[li,ni]:= drand48;    dtheta[li,ni]:=0;
        end;
    end;
    { Randomize weights }
    for li:=Layers_Max-1 downto 1 do
    begin
        ni_end:=node[li];    no_end:=node[li-1];
        for ni:=0 to ni_end-1 do
            for no:=0 to no_end-1 do
            begin
                drand48:=Random(31)/100;    weight[li,ni,no]:=drand48;    dweight[li,ni,no]:=0;
            end;
        end;
    end;
end; { procedure WRAND }
PROCEDURE INITIAL;
{ sets zero the output of each node before every iteration }
VAR

```



```

l, n, n_end: integer;
begin
  for l:= 1 to Layers-1 do {Dikkat l:=0 idi}
    begin
      n_end:= node[l];
      for n:= 0 to n_end-1 do
        out[l,n]:= 0;
      end;
    end;
  end; { procedure INITIAL }
PROCEDURE YREAD ( isylla: integer );
{ read input pattern ( for several different input combination ) }
begin
  case isylla of
    0: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=0.5;END;
    1: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=1 ;END;
    2: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=1.5;END;
    3: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=2 ;END;
    4: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=2.5;END;
    5: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=3 ;END;
    6: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=3.5;END;
    7: BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=0.5;END;
    8: BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=1 ;END;
    9: BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=1.5;END;
    10:BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=2 ;END;
    11:BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=2.5;END;
    12:BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=3 ;END;
    13:BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=3.5;END;
  end;
end; { procedure YREAD }
PROCEDURE AYREAD(ISYLLA:BYTE);
{ read input pattern ( for several different input combination ) }
var
  a:string[80];
  x:real;
  l,m:byte;
  odosya:text;
begin

```



```

Assign(odosya,'i'+d+'.dat'); reset(odosya);
if (isylla=0) then read(odosya,out[0,0])
else
begin
for l:=1 to isylla do
begin
if (l=isylla) then
begin
for m:=0 to outputs do
read(odosya,x);      read(odosya,out[0,0]);
end;
readln(odosya,a);
end;
end;
close(odosya);
end; {Procedure AYread}
FUNCTION SIGMOID ( x: extended ): EXTENDED;
{ sigmoid function }
var
y: EXTENDED;
begin
Beta:=1;
if x >= 2000 then y:=1
else
begin
if x <= -2000 then y:=0
else y:=1/(1+exp(-beta*(x)));
end;
sigmoid:=y;
end; { function SIGMOID }
FUNCTION TANHIPER ( x: REAL ): REAL;
{ tanhiper function }
var
y: REAL;
begin
Beta:=1;
if x >= 1000000000 then y:=1

```



```

else
begin
  if x <= -1000000000 then y:=1
    else y:=(exp(beta*(x))-exp(-beta*(x)))/(exp(beta*(x))+exp(-beta*(x)));
end;

tanhiper:=y;
end; { function TANHIPER }
{FUNCTION SIGMOID ( x: REAL ): REAL;
sigmoid function
var
  y: REAL;
begin
  y:=1/(1+exp(-(x)));    sigmoid:=y;
end; function SIGMOID }
PROCEDURE WFORWARD ;
{ wforward propagation }
var
  li, ni, ni_end, no, no_end: integer;
  x, th: REAL;
begin
  for li:=1 to Layers-1 do
  begin
    ni_end:= node[li];    no_end:= node[li-1];
    for ni:= 0 to ni_end-1 do
    begin
      x:= 0;
      for no:= 0 to no_end-1 do
        x:= x + weight[li,ni,no] * out[li-1,no];out[li,ni]:= sigmoid( x-theta[li,ni] );
      end;
    end;
  end;
end; { procedure FORWARD }
FUNCTION BACK: REAL;
{ back propagation }
var
  li, ni, ni_end, no, no_end: integer;
  e, err, x, d, w, y: REAL;
begin

```



```

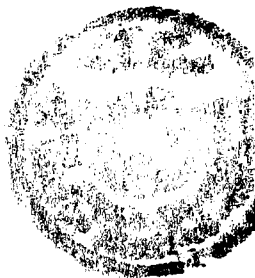
err:=0;
for ni:=0 to outputs-1 do
begin
  y:= out[2,ni];des:=dess[isylla,ni];      e:= des - y;      der[2,ni]:= e*y*(1-y);      err:=err+ e*e;
end;
for li:=2 downto 1 do
begin
  no_end:= node[li-1];
  ni_end:= node[li];
  for no:= 0 to no_end-1 do
  begin
    x:= 0; y:= out[li-1,no];
    for ni:= 0 to ni_end-1 do
    begin
      d:= der[li,ni];      w:= weight[li,ni,no];      x:= x + d*w;
    end;
    der[li-1,no]:= y*(1-y)*x;
  end;
end;
back:= 0.5*err;
end; { function BACK }
PROCEDURE LEARNING (alp, eps: REAL );
{ update offsets & weights }
var
  li, lo, ni, ni_end, no, no_end: integer;
  di, yo, ew, et: REAL;
begin
  for li:= 1 to layers-1 do
  begin
    ni_end:= node[li];
    for ni:= 0 to ni_end-1 do
    begin
      et:= der[li,ni];      dtheta[li,ni]:= -eps*et + alp*dtheta[li,ni];
      theta[li,ni]:= theta[li,ni] + dtheta[li,ni];
    end;
  end;
end;
for li:=1 to layers-1 do

```

```

begin
  lo:= li-1;   ni_end:= node[li];   no_end:= node[lo];
  for ni:= 0 to ni_end-1 do
    begin
      di:= der[li,ni];
      for no:= 0 to no_end-1 do
        begin
          yo:= out[lo,no];   ew:= di*yo;   dweight[li,ni,no]:= eps*ew + alp*dweight[li,ni,no];
          weight[li,ni,no]:= weight[li,ni,no] + dweight[li,ni,no];
        end;
      end;
    end;
  end;
end; { procedure LEARNING }
PROCEDURE MAKE_MATRIX( it, isylla: integer );
var
  i,j: integer;
begin
  for i:= 0 to inputs-1 do
    matrix_tmp[it,isylla,i]:= out[0,i];
  for i:= 0 to outputs-1 do
    matrix_tmp[it,isylla,inputs+i]:= out[Layers_Max-1,i];
  end; { procedure MAKE_MATRIX }
PROCEDURE READ_WEIGHTS;
var
  data,data1,data2,data3,data4,data5:string[100];
  li,ni,no,no_end,ni_end: byte;
begin
  write ('OKUNACAK AĞIRLIK DOSYASININ ADINI GİRİNİZ. ');readln(ad);
  assign(weightfile,'c:\tp\hamid\'+ad+'.txt'); reset(weightfile); readln(weightfile,data1);
  readln(weightfile,data2); readln(weightfile,data3); readln(weightfile,data4);
  for li:=Layers_Max-1 downto 1 do
    begin
      ni_end:=node[li];   no_end:=node[li-1];
      for no:=0 to no_end-1 do
        begin
          for ni:=0 to ni_end-1 do
            begin

```



```

        read(weightfile,weight[li,ni,no]); read(weightfile,dweight[li,ni,no]);{dweight[li,ni,no]:=0;}
    end;
end;
end;
for li:=1 to layers-1 do
begin
    ni_end:=node[li];
    for ni:=0 to ni_end-1 do
    begin
        readln(weightfile,theta[li,ni]);    readln(weightfile,dtheta[li,ni]);
    end;
end;
read(weightfile,alp);read(weightfile,eps); close(weightfile);
end;
PROCEDURE WRITE_WEIGHTS;
var
    li,ni,no,no_end,ni_end:byte;
begin
    GetTime(h2,m2,s2,hund2); assign(weightfile,'a'+d+'.txt'); rewrite(weightfile);
    writeln(weightfile,d,' Transistörüne ait ağırlıklar ve offsetler.'j,' iterasyon için aşağı-');
    writeln(weightfile,'da verilmiştir.Sinyal ve gürültü parametreleri için toplam hata',terror,' dir. ');
    writeln(weightfile,'Başlama Zamanı=',h1,'.',m1,'.',s1,'Bitiş zamanı=',h2,'.',m2,'.',s2,
    'Fark=',h2-h1,' saat ',m2-m1,' dakika ',s2-s1,' Saniyedir. ');
    writeln(weightfile,'Gizli Katman Düğüm Sayısı=',node[1],' Alpha=',Alp,' Epsilon=',Eps);
    for li:=Layers_Max-1 downto 1 do
    begin
        ni_end:=node[li];    no_end:=node[li-1];
        for no:=0 to no_end-1 do
        begin
            for ni:=0 to ni_end-1 do
            begin
                write(weightfile,' ',weight[li,ni,no]);    write(weightfile,' ',dweight[li,ni,no]);
            end;
            writeln(weightfile,' ');
        end;
    end;
end;
for li:=1 to layers-1 do

```

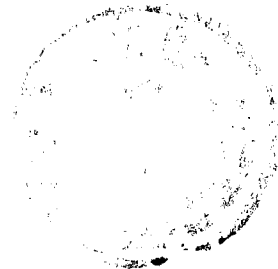




```

begin
  ni_end:=node[li];
  for ni:=0 to ni_end-1 do
    begin
      writeln(weightfile,theta[li,ni]);      writeln(weightfile,dtheta[li,ni]);
    end;
  end;
  writeln(weightfile,alp,eps); close(weightfile);
end;
PROCEDURE CALCULATE_SF_BIAS_CT;
type
  dizi=array[0..outputs_max-1,0..output_nodes] of real;
  sdizi=array[0..outputs_max-1,0..output_nodes] of string[8];
var
  tampon:text;
  secenek:char;
  device:string[3];
  f:array[0..50] of real;
  sout:sdizi;
  fdess:dizi;
  fmax,falt,step,a:real;
  k,l,m:byte;
begin
  writeln('HATA HESABI YAPILSIN İSTİYOR MUSUNUZ ?[E]VET/[H]AYIR!');READLN(pass);
  pass:=Upcase(pass);tt:=0;
  writeln('Test Başlıyor...');writeln;
  read_weights;{DİKKAT FORMAT AYNI DEĞİLSE UYGUNSUZ SONUÇLAR ÇIKAR}
  writeln('KUTUPLAMA NOKTASINI GİRİNİZ');writeln('VCE[V]=');READ(OUT[0,0]);
  writeln('IC[mA]=');READ(OUT[0,1]);writeln('Kutuplama Tipini Giriniz CE için 1 CC için 9 a basınız
');
  READ(OUT[0,2]);
  for k:=0 to inputs-2 do
    out[0,k]:=out[0,k]/10;
  writeln('TEK BİR FREKANSI TEST ETMEK İÇİN [T] YE');
  writeln('BELLİ BİR FREKANS BANDINI TEST ETMEK İÇİN [B] YE');
  writeln('RASGELE AYRIK FREKANSLARI HESAPLAMAK İÇİN [A] YA BASINIZ!!!');
  READLN(tercih);tercih:=Upcase(tercih);tercih:='A';

```



```

case tercih of
'T': begin write('FREKANSI [GHz] OLARAK GİRİNİZf=');readln(OUT[0,3]);samplecount:=1;END;
'B': begin
    write('FREKANSIN ÜST SINIRINI [GHz] OLARAK GİRİNİZ ');readln(fmax);
    write('FREKANSIN ALT SINIRINI [GHz] OLARAK GİRİNİZ ');readln(falt);
    write('FREKANS ADIMINI [GHz] OLARAK GİRİNİZ ');readln(step);clrscr;
    out[0,3]:=falt-step;
    samplecount:=round((fmax-falt)/step+1);
end;
'A': begin
    write('AYRIK FREKANSLARI BELİRLİ DOSYADAN OKUTMAK İSTER MİSİNİZ? [E/H]');
    secenek:=readkey;secenek:=Upcase(secenek);clrscr;
    if secenek='E' then
        begin
            assign(tampon,'c:\tp\hamid\2135tp2.dat');          reset(tampon);
            end;
            writeln('KAÇ AYRIK FREKANS İÇİN TEST ETTİRMEK İSTİYORSUNUZ?');
            readln(samplecount);
            for k:=0 to samplecount-1 do
                begin
                    if secenek='E' then read(tampon,f[k])
                    else
                        begin
                            write(k,'INCI ÖRNEĞİ GİRİNİZ ');          readln(f[k]);
                            end;
                        end;
                    end;
                if secenek='E' then close(tampon);
            end;
        end;
        writeln('SONUÇLARI PRİNTERDEN ALMAK İÇİN [P]');
        writeln('SONUÇLARI MONİTÖRDEN ALMAK İÇİN [M] E BASINIZ. ');
        readln(secenek);secenek:=Upcase(secenek);clrscr;
        case secenek of
            'M':device:="";
            'P':device:='lst';
        end;
        if device='lst' then

```



```

begin
    writeln (lst,'Freq. VCE IC Config      Calculated S&F Parameters ');
    writeln (lst,' GHz [V] [mA]      S11      S21      S12      S22 ');
    writeln (lst,'-----');
end
else
begin
    writeln ('Freq. VCE IC Config      Calculated S&F Parameters ');
    writeln (' GHz [V] [mA]      S11      S21      S12      S22 ');
    writeln ('-----');
end;
assign(graphfile,'c:\tp\hamid\h'+d+'.dat'); {d:=copy(d,1,7);} rewrite(graphfile);
for m:=0 to samplecount-1 do {Repeat}
begin
    case terciah of
        'A':out[0,3]:=f[m];
        'B':out[0,3]:=out[0,3]+step;
        'T':out[0,3]:=out[0,3];
    end;
    wforward; {out[0,2]:=out[0,2]+step;} str(out[0,3]:6:3,st1);{Frekans}
    str(out[0,1]*10:3:1,st2);{IC} str(out[0,0]*10:3:1,st3);{VCE}
    if device='lst' then write(lst,st1+' ') else write(st1+' ');
    if device='lst' then write(lst,st3+' ') else write(st3+' ');
    {write(lst,st1+' ');}
    if device='lst' then write(lst,st2+' ') else write(st2+' ');
    if out[0,2]-0.1<0.00001 then st4:='CE' else st4:='CC';
    if device='lst' then write(lst,' '+st4+' ') else write(' '+st4+' ');
    for l:= 0 to outputs-1 do
    begin
        str(out[2,l]:5:3,st1);
        case (l) of
            1,3,5,7,10:begin
                if (out[2,l] > 0.5) then
                begin
                    out[2,l]:=out[2,l]*360-360; str(out[2,l]:7:2,st1);
                end
                else

```

```

begin
    out[2,1]:=out[2,1]*360;          str(out[2,1]:7:2,st1);
end;
end;
4:begin
    out[2,1]:=out[2,1]/s12max;      str(out[2,1]:5:3,st1);
end;
8:begin
    out[2,1]:=out[2,1]*Fminmax;     str(out[2,1]:5:3,st1);
end;
2:begin
    out[2,1]:=out[2,1]*s21max;      str(out[2,1]:5:3,st1);
end;
end;
if device='lst' then write(lst,st1+' ') else write(st1+' ');
end;
begin
    str(out[0,0]:6:3,sout[0,0]);      str(out[0,1]:6:3,sout[0,1]);
    str(out[0,2]:2:1,sout[0,2]);      str(out[0,3]:6:3,sout[0,3]);
    for l:=0 to outputs-1 do
        str(out[2,1]:7:4,sout[2,1]);
        writeln(graphfile,sout[0,0],' ',sout[0,1],' ',sout[0,2],' ',sout[0,3],' ',sout[2,0],' ',sout[2,1],' ',
        ,sout[2,2],' ',sout[2,3],' ',sout[2,4],' ',sout[2,5],' ',sout[2,6],' ',sout[2,7],' ',sout[2,8],' ',sout[2,9],' ',
        ,sout[2,10],' ',sout[2,11]);    a:=1;
    end;
end;
{end;}
if device='lst' then writeln(lst,' ') else writeln;
end;{until (fmax <= out[0,0]);}
close(graphfile);
if device="" then writeln("DEVAM ETMEK İÇİN [enter] TUŞUNA BASINIZ!!! ");
if device="" then readln; if device='lst' then writeln(lst,' ') else writeln;
str(out[0,0]*10:3:1,st1); str(out[0,1]*10:3:1,st2);
if device='lst' then
begin
    writeln(lst,'Yukarıdaki sonuçlar '+ st1+' '+st2+' kutuplama noktasında '+st4+'konfigürasyonu için
elde edilmiştir. '); writeln(lst,' ');
    writeln (lst,'Freq. VCE IC Config      Calculated S&F Parameters ');

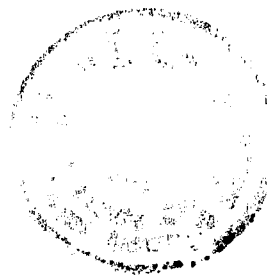
```



```

writeln (lst,' GHz [V] [mA]      S11      S21      S12      S22 ');
writeln (lst,'-----');
end
else
begin
writeln('Yukarıdaki sonuçlar '+ st1+' '+st2+' kutuplama noktasında '+st4+'konfigürasyonu için elde
edilmiştir. '); writeln;
writeln ('Freq. VCE IC Config      Calculated S&F Parameters ');
writeln (' GHz [V] [mA]      S11      S21      S12      S22 ');
writeln ('-----');
end;
reset(idatafile);
for l:=0 to outputs_max-1 do
for m:=0 to output_nodes do
read(idatafile,fdess[l,m]); clrscr;
for k:=0 to outputs_max-1 do
for l:= 0 to output_nodes-1 do
begin
case (l) of
1,3,5,7,10:begin
if dесс[k,l] > 0.5 then
begin
dесс[k,l]:=dесс[k,l]*360;          dесс[k,l]:=dесс[k,l]-360;
end
else
begin
dесс[k,l]:=dесс[k,l]*360;
end;
end;
8:dесс[k,l]:=dесс[k,l]*Fminmax;
4:dесс[k,l]:=dесс[k,l]/s12max;
2:dесс[k,l]:=dесс[k,l]*s21max;
end;
end;
for l:=0 to outputs_max-1 do
begin
str(fdess[l,0]:5:2,st1);

```

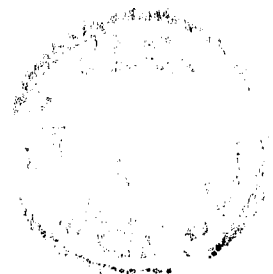


```

if device='lst' then write(lst,st1+' ') else write(st1+' ');
for m:=0 to output_nodes-1 do
begin
str(dess[l,m]:5:3,st1);
case (m) of
1,3,5,7,10:begin
str(dess[l,m]:7:2,st1);
end;
end;
if device='lst' then write(lst,st1+' ') else write(st1+' ');
end;
if device='lst' then writeln(lst,' ') else writeln(' ');
end;
if device="" then writeln('DEVAM ETMEK İÇİN [enter] TUŞUNA BASINIZ!!! ');if device="" then
readln; close(idatafile);
end; {Procedure Calculate_SF }
PROCEDURE HATA_RAPORU_SF_BIAS_CT;
type
dizi=array[0..outputs_max-1,0..output_nodes] of real;
sdizi=array[0..outputs_max-1,0..output_nodes] of string[8];
var
data:string[100];
fdess,calc,error:dizi;
terror:sdizi;
st1:string[4];
hata,hata1,hata2,hata3,hata4,hata5,hata6,hata7,shata,vce,ic,fhata:string[8];
thata:array [0..100] of REAL;
k,l,tt,m,N:byte;
a,s11,s12,s21,s22,fmin,ropt,rn,s_hata,f_hata,reel,imajinel,toplamhata,s,step:REAL;
odosya,hdosya,edosya:text;
begin
assign(hdosya,'h'+d+'.dat'); assign(odosya,'ne2135b'+'.dat'); assign(edosya,'e'+d+'.dat');
reset(hdosya); reset(odosya);tt:=0;k:=0; append(edosya);s11:=0;s12:=0;s21:=0;s22:=0;
fmin:=0;ropt:=0;rn:=0;
writeln(edosya,'FREKANS[GHz] S11 S12 S21 S22 Fmin ropt Rn/50');
for N:=0 to samplecount-1 do
begin

```

TÜRKİYE  
 İÇİŞLERİ BAKANLIĞI  
 SİBER GÜVENLİK VE VERİ KORUMA GENEL MÜDÜRLÜĞÜ  
 SİBER GÜVENLİK VE VERİ KORUMA GENEL MÜDÜRLÜĞÜ  
 SİBER GÜVENLİK VE VERİ KORUMA GENEL MÜDÜRLÜĞÜ



```

m:=0;
readln(hdosya,calc[0,1],calc[0,2],calc[0,3],calc[0,0],calc[2,0],calc[2,1],calc[2,2],calc[2,3],calc[2,4],
calc[2,5],calc[2,6],calc[2,7],calc[2,8],calc[2,9],calc[2,10],calc[2,11]);
calc[0,1]:=calc[0,1]*10;   calc[0,2]:=calc[0,2]*10;
repeat
  inc(m);
  if m>100 then begin
    writeln('DİKKAT TEST EDİLEN DEĞERİN ÖLÇÜLMÜŞ KARŞILIĞI YOK!!!');
    writeln('HATA HESAPLANAMAZ PROGRAM KIRILDI!!!');exit;
  END;
readln(odosya,fdess[0,1],fdess[0,2],fdess[0,3],fdess[0,0],fdess[2,0],fdess[2,1],fdess[2,2],fdess[2,3],fdess[
2,4],fdess[2,5],fdess[2,6],fdess[2,7],fdess[2,8],fdess[2,9],fdess[2,10],fdess[2,11]);
  if not((abs(fdess[0,1]-calc[0,1])<0.00001) and (abs(fdess[0,2]-calc[0,2])<0.00001) and
(abs(fdess[0,3]-calc[0,3])<0.00001)) then
    begin
      repeat
readln(odosya,fdess[0,1],fdess[0,2],fdess[0,3],fdess[0,0],fdess[2,0],fdess[2,1],fdess[2,2],fdess[2,3],
        fdess[2,4],fdess[2,5],fdess[2,6],fdess[2,7],fdess[2,8],fdess[2,9],fdess[2,10],fdess[2,11]);
        until (abs(fdess[0,1]-calc[0,1])<0.00001) and (abs(fdess[0,2]-calc[0,2])<0.00001) and
          (abs(fdess[0,3]-calc[0,3])<0.00001);
      end;
      until (abs(fdess[0,0]-calc[0,0])<0.00001);
      str(fdess[0,0]:4:1,st1);   a:=1;s:=0;write(edosya,' '+st1+' ');
      for k:=0 to outputs-1 do
        begin
          error[2,k]:=abs(abs(fdess[2,k])-abs(calc[2,k]))/abs(fdess[2,k]);
          s:=s+error[2,k];   str(error[2,k]:8:6,terror[2,k]);
        end;
      k:=0;
      for l:=0 to 6 do {3 for s parameters}
        begin
          if k<=9 then
            begin
              reel:=sqr(fdess[2,k]*cos(fdess[2,k+1]*pi/180)-calc[2,k]*cos(calc[2,k+1]*pi/180));
              imajinel:=sqr(fdess[2,k]*sin(fdess[2,k+1]*pi/180)-calc[2,k]*sin(calc[2,k+1]*pi/180));
              error[2,k]:=sqrt(reel+imajinel)/abs(fdess[2,k]);
            end;
          end;

```

```

case k of
  0:s11:=s11+error[2,k];
  2:s21:=s21+error[2,k];
  4:s12:=s12+error[2,k];
  6:s22:=s22+error[2,k];
  8:begin
    error[2,k]:=abs(fdess[2,k]-calc[2,k])/abs(fdess[2,k]); fmin:=fmin+error[2,k]; k:=k-1;
  end;
  9:ropt:=ropt+error[2,k];
  11:begin
    error[2,k]:=abs(fdess[2,k]-calc[2,k])/abs(fdess[2,k]); rn:=rn+error[2,k];
  end;
end;
str(error[2,k]:8:6,terror[2,k]); write(edosya,' ',terror[2,k]); k:=k+2;
end;
writeln(edosya,' '); thata[tt]:=s/outputs; inc(tt);
{writeln(edosya,'BU FREKANS AİT TOPLAM MUTLAK HATA= ',thata[tt]);}
end;{until (eof(odosya));} close(odosya); close(hdosya); s:=0;
for tt:=0 to samplecount-1 do s:=s+thata[tt];
toplamhata:=s/samplecount; s11:=s11/samplecount;s12:=s12/samplecount;
s21:=s21/samplecount; s22:=s22/samplecount; ropt:=ropt/samplecount; fmin:=fmin/samplecount;
rn:=rn/samplecount; str(s11:9:6,hata1); str(s21:9:6,hata2); str(s12:9:6,hata3); str(s22:9:6,hata4);
str(fmin:9:6,hata5); str(ropt:9:6,hata6); str(rn:9:6,hata7); str(toplamhata:9:6,hata);
s_hata:=(s11+s12+s21+s22)/4; str(s_hata:9:6,shata); f_hata:=(fmin+ropt+rn)/3;
str(f_hata:9:6,fhata); str(calc[0,1]:5:1,vce); str(calc[0,2]:5:1,ic); assign(weightfile,ad+'.txt');
reset(weightfile); readln(weightfile,data); data:=copy(data,51,6);
writeln(edosya,'VCE='+vce+' [Volt] IC='+ic+' [mA] kutuplama noktasında');
writeln(edosya,data+' adet iterasyon için '+d+' transistörüne ait');
writeln(edosya,'Gerçek Mutlak Toplam Hata = ',hata);
writeln(edosya,'E11=',hata1,' E21=',hata2,' E12=',hata3,' E22=',hata4); writeln(edosya,'Et= ',shata);
writeln(edosya,' F1=',hata5,' F2=',hata6,' F3=',hata7); writeln(edosya,'Ft= ',fhata); close(edosya);
close(weightfile);
end; {Procedure Hata_Raporu}
BEGIN { main }
  clrscr; arrange; hedef_belle;
  { print first header }
  node[2]:= Output_Nodes;

```

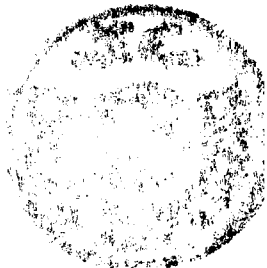




```

Outputs:= Output_Nodes;    { Çıkış Katmanı Düğüm Sayısı }
node[1]:= Hidden_Nodes;    { Gizli Katman Düğüm Sayısı }
node[0]:= Input_Nodes;
Inputs:= Input_Nodes;      { Giriş Katmanı Düğüm Sayısı }
Layers:= Layers_Max;       { Katman Sayısı }
num:= Outputs_Max;         { Örnek Sayısı }
alp:=alpha; eps:=epsilon;
{ *****
INITIALIZE MATRIX
***** }
for i:= 0 to num-1 do
for j:= 0 to ( inputs+Outputs-1 ) do
matrix[i,j]:= 0;
for ii:= 0 to ( Trunc(Iterative_Max/Every)-1 ) do
for i:= 0 to num-1 do
for j:= 0 to ( inputs+Outputs-1 ) do
matrix_tmp[ii,i,j]:=0;
for i:=1 to 24 do
writeln;
write('EĞİTME İÇİN (0) VEYA TEST İÇİN (1) E BASINIZ. '); readln(tt);
for i:=1 to 24 do writeln;
if tt=1 then calculate_sf_BIAS_CT
else
begin
writeln('Eğitmeyle İlgili Bilgileri Direkt Dosyadan Okutmak İsterseniz [D] ye');
writeln('Program İçinde Oluşturduğunuz Alt Programdan Okutmak İsterseniz [A] ya basınız. ');
readln(pass);pass:=upcase(pass);if pass='D' then hiz:='slow' else hiz:='fast'; pass:='H';
assign(outfile,'c:\tp\hamid\BSFoutfi.txt'); STR(EPSILON:4:2,ST5);STR(ALPHA:4:3,ST6);
writeln ('*****');
WRITELN (' EPSİLON=',st5, ' ', 'MIKRODALGA TRANSİSTORUNUN YNA İLE
SİMÜLASYONU..... ' );
WRITELN (' ALPHA =',st6, ' ' { HAMİD TORPİ AĞUSTOS\ 1994});
writeln ('*****');
writeln;append(outfile);
write('MEVCUT AĞIRLIKLARDAN YARARLANMAK İSTİYOR MUSUNUZ? [E/H] ');
readln(tercih);tercih:=Upcase(tercih);
if tercih='E' then read_weights else wrand;

```



```

GetTime(h1,m1,s1,hund1);  j:=0;dcount:=trunc(iterative_max/every)+1;
{*****}
LEARNING PHASE
***** }
while ( j < Iterative_Max ) do
begin
  for count:= 0 to Every-1 do
  begin
    error:= 0;
    for isylla:=0 to num-1 do
    begin
      initial;          if hiz='slow' then ayread( isylla ) else yread(isylla);          wforward;
      err:= back;          learning( alp, eps );          error:= error+err;
    end;
    error:= error / outputs;          eps:= 0.5 * sqrt( error );          Inc(j);
  end;
  {*****}
  TESTING PHASE
  ***** }
  terror:= 0;
  for isylla:= 0 to num-1 do
  begin
    if hiz='slow' then ayread( isylla ) else yread(isylla);          wforward;          terr:= back;
    terror:= terror + terr;          make_matrix( trunc( j/Every ) - 1, isylla );
  end;
  write (outfile, '          Freq.          Calculated S Parameters');
  writeln (outfile, '          &          F Parameters');
  write (outfile, 'NUM Bias Point Conf. GHz          S11          S21          S12          S22');
  write (outfile, 'Fmin ropt Rn/50          S11          S21          S12          S22 Fmin ropt');
  writeln (outfile, 'Rn/50');
  write (outfile, '-----');
  write (outfile, '-----');
  writeln (outfile, '----');
  for i:= 0 to num-1 do
  begin
    str( i:2, st1 );          write(outfile,st1+' ');
    for ii:= 0 to inputs-1 do

```



```

begin
  str( matrix_tmp[trunc(j/Every)-1,i,ii]:6:3, st1 );      write(outfile,st1+' ');
end;
write(outfile,' ');
for ii:= 0 to outputs-1 do
begin
  str( matrix_tmp[trunc(j/Every)-1,i,inputs+ii]:5:3, st1);      write( outfile,st1+' ');
end;
write(outfile,' ');
for ii:=0 to outputs-1 do
begin
  str( dess[i,ii]:5:3, st1 );      write( outfile, st1+' ');
end;
Writeln(outfile,' ');
end;
terror:= terror/outputs;
str ( j:7, st1 ); str ( eps:9:6, st2 );      str ( error:9:6, st3 ); str ( terror:9:6, st4 );
writeln( outfile, 'iteration=',st1,' eps=',st2,      ' error_1=', st3, ' error_t=', st4 );
if ( j = Iterative_Max ) then
begin
  writeln( outfile, 'learn & test finished at time alpha =',st6,' epsilon =',st5 );
  writeln( outfile,d,' Transistoruna ait eğitim sonuçları yukarıda verilmiştir. ');
end;
write_weights;
end; {İlk büyük döngü burada bitiyor}
close( outfile );
end;
if (pass='E') then Hata_Raporu_SF_BIAS_CT;
end.

```

### EK 1.1.1 Çeşitli Kutuplama İçin Sinyal Parametreleri Simülatörü

```

PROGRAM MULTILAYERPERCEPTRON;{DİKKAT AYREAD VE YREAD SEÇİMİNİ YAPMAYI
UNUTMA}
{$N+}
uses crt,printer,DOS;
CONST

```



```

Layers_Max = 3; { Maximum number of layers }
Nodes_Max = 8; { Maximum number of node in any layer }
Outputs_Max = 36; { Number of input combinations }
Input_Nodes = 3; { Number of input layer nodes }
Output_Nodes = 8; { Number of output layer nodes }
Hidden_Nodes = 8; { Number of hidden layer nodes }
Epsilon = 0.5;
Alpha = 0.93;
Iterative_Max = 1000; { Maximum iteration number }
Every = 1000; { After every iteration print results }
s21max = 35; { Normalization coefficient40}
s12max = 4; { Normalization coefficient3.7}

```

TYPE

```
adizi = array [ 0..outputs_Max-1 ] of real;
```

VAR

```

h1,h2,m1,m2,s1,s2,hund1,hund2 : Word;
j,count:longint;
f:adizi;
samplecount,i,ii,tt,isylla, num,dcount, Layers, Inputs, Outputs :byte;
fmax,fmin,step,eps, alp, err, error, terr, terror, total_rate,des,beta: REAL;
Node : array [0..Layers_Max-1] of integer;
matrix : array [0..Outputs_Max-1, 0..(Input_Nodes+Output_Nodes-1)] of REAL;
matrix_tmp: array [0..trunc(Iterative_Max/Every), 0..Outputs_Max-1,
                  0..(Input_Nodes+Output_Nodes-1)] of REAL ;
out : array [0..Layers_Max-1, 0..Nodes_Max-1] of REAL;
der : array [0..Layers_Max-1, 0..Nodes_Max-1] of REAL;
theta : array [0..Layers_Max-1, 0..Nodes_Max-1] of REAL;
dtheta : array [0..Layers_Max-1, 0..Nodes_Max-1] of REAL;
weight : array [0..Layers_Max-1, 0..Nodes_Max-1, 0..Nodes_Max-1] of REAL;
dweight : array [0..Layers_Max-1, 0..Nodes_Max-1, 0..Nodes_Max-1] of REAL;
dess : array [0..Outputs_Max-1,0..output_nodes-1] of REAL;

st1: string[7]; st5,st6: string[6];
d,st2, st3, st4: string[12];
idatafile,datafile,graphfile,outfile :text;
weightfile:text;
tercih,pass:char;

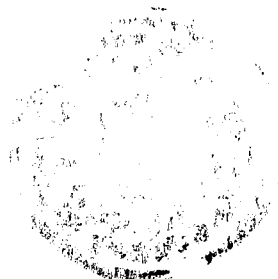
```



```

ad,hiz:string[8];
PROCEDURE ARRANGE;
var
  dosya,ydosya:text;
  A:string[1];
  B:string[10];
  q:string[80];
begin
  write(' OKUNACAK DOSYANIN ADINI GİRİNİZ. ');READLN(d);
  assign(dosya,'c:\appcad\spara\necspar'+d+'.s2p'); assign(ydosya,d+'.dat'); reset(dosya);
  repeat
    readln(dosya,q); a:=copy(q,1,1); b:=copy(q,2,9);
  until (a='Z') or (a='z');
  rewrite(ydosya);
  while not eof(dosya) do
  begin
    readln(dosya,q); a:=copy(q,1,1);
    if (a='Z') or (a='z') then
    begin
      b:=copy(q,2,9); readln(dosya,q);
    end;
    writeln(ydosya,b,' ',q);
  end;
  close(ydosya); close(dosya);
end; {Procedure arrange}
PROCEDURE HEDEF_BELLE;
type
  dizi=array[0..outputs_max-1,0..output_nodes] of real;
var
  tampon:text;
  yanit:char;
  fdess:dizi;
  x,y:real;
  k,l,m,n:byte;
  ST:string[8];
begin
  assign(tampon,'c:\tp\hamid\btampon.dat'); {assign(tampon,'tampon.dat');}

```



```

write(' EN SON KULLANILAN FREKANS ÖRNEKLERİNİ KULLANMAK İSTER MİSİNİZ?
[E/H]');readln(yanit); yanit:=upcase(yanit);
if yanit='E' then
begin
reset(tampon);
for l:=0 to outputs_max-1 do
begin
READ(tampon,F[l]);
end;
end
else
begin
rewrite(tampon); writeln(' HANGİ FREKANSLARDA ÖRNEK ALACAKSANIZ ');
writeln(' SIRAYLA [GHz] OLARAK GİRİNİZ ');
for l:=0 to outputs_max-1 do
begin
write(' ');write (l+1);write(' ÖRNEK = ');READ(F[l]);write(tampon,f[l]);write(' GHz de');
writeln;
end;
writeln(' ALINACAKTIR. ');
end;
close(tampon); assign(datafile,d+'.dat'); reset(datafile);m:=0;
assign(idatafile,'i'+d+'.dat'); rewrite(idatafile);
while not eof(datafile) do
begin
for n:=0 to input_nodes-1 do
read(datafile,x);
for l:=m to outputs_max-1 do
begin
while not (x=f[l]) do
begin
while not eoln(datafile) do
read(datafile,x);
for n:=0 to input_nodes-1 do
read(datafile,x);
end;
for k:=0 to output_nodes do

```

```

begin
  fdess[m,k]:=x;      read(datafile,x);
end;
for k:=0 to output_nodes do
begin
  y:=fdess[m,k];str(Y:6:3,ST);      write(idatafile,' ',st);
end;
writeln(idatafile,' ');      m:=m+1;
for n:=0 to input_nodes-2 do
  read(datafile,x);
end;
end;
close(datafile);close(idatafile);
for m:=0 to outputs_max-1 do
for k:=0 to output_nodes-1 do
dess[m,k]:=fdess[m,k+1]; clrscr;
for k:=0 to outputs_max-1 do
for l:= 0 to output_nodes-1 do
begin
case (l) of
1,3,5,7:begin
if dess[k,l] < 0 then
begin
dess[k,l]:=dess[k,l]+360;      dess[k,l]:=dess[k,l]/360;
end
else
begin
dess[k,l]:=dess[k,l]/360;
end;
end;
4:dess[k,l]:=dess[k,l]*s12max;
2:dess[k,l]:=dess[k,l]/s21max;
end;
end;
end; {Procedure hedef_belle}
PROCEDURE WRAND;
PROCEDURE INITIAL;

```

```

PROCEDURE YREAD ( isylla: integer );
{ read input pattern ( for several different input combination ) }
begin
  case isylla of
    0:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=0.1;END;
    1:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=0.3;END;
    2:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=0.6;END;
    3:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=1.0;END;
    4:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=1.2;END;
    5:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=1.5;END;
    6:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=1.8;END;
    7:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=2.0;END;
    8:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=3.0;END;
    9:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=4.1;END;
    10:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=4.5;END;
    11:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=5.0;END;
    12:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=0.1;END;
    13:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=0.3;END;
    14:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=0.6;END;
    15:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=1.0;END;
    16:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=1.2;END;
    17:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=1.5;END;
    18:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=1.8;END;
    19:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=2.0;END;
    20:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=3.0;END;
    21:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=4.0;END;
    22:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=4.5;END;
    23:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=5.0;END;
    24:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=0.1;END;
    25:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=0.3;END;
    26:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=0.6;END;
    27:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=1.0;END;
    28:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=1.2;END;
    29:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=1.5;END;
    30:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=1.8;END;
    31:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=2.0;END;
    32:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=3.0;END;

```





```

33:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=4.0;END;
34:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=4.5;END;
35:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=5.0;END;
end;
end; { procedure YREAD }
PROCEDURE AYREAD(ISYLLA:BYTE);
{ read input pattern ( for several different input combination ) }
var
  a:string[80];
  x:real;
  l,m:byte;
  odosya:text;
begin
  Assign(odosya,'i'+d+'.dat'); reset(odosya);
  if (isylla=0) then read(odosya,out[0,0])
  else
  begin
    for l:=1 to isylla do
    begin
      if (l=isylla) then
      begin
        for m:=0 to outputs do
          read(odosya,x);          read(odosya,out[0,0]);
        end;
        readln(odosya,a);
      end;
    end;
  end;
  close(odosya);
end; {Procedure AYread}
FUNCTION SIGMOID ( x: REAL ): EXTENDED;
PROCEDURE WFORWARD ;
FUNCTION BACK: REAL;
PROCEDURE LEARNING (alp, eps: REAL );
PROCEDURE MAKE_MATRIX( it, isylla: integer );
PROCEDURE READ_WEIGHTS;
var
  data,data1,data2,data3,data4,data5:string[100];

```



```

li,ni,no,no_end,ni_end: byte;
begin
write ('OKUNACAK AĞIRLIK DOSYASININ ADINI GİRİNİZ. ');readln(ad);
assign(weightfile,'c:\tp\hamid\'+ad+'.txt'); reset(weightfile); readln(weightfile,data);
readln(weightfile,data1); readln(weightfile,data2); readln(weightfile,data3);
readln(weightfile,data4);
for li:=Layers_Max-1 downto 1 do
begin
ni_end:=node[li]; no_end:=node[li-1];
for no:=0 to no_end-1 do
begin
for ni:=0 to ni_end-1 do
begin
read(weightfile,weight[li,ni,no]); read(weightfile,dweight[li,ni,no]);
end;
end;
end;
for li:=1 to layers-1 do
begin
ni_end:=node[li];
for ni:=0 to ni_end-1 do
begin
readln(weightfile,theta[li,ni]); readln(weightfile,dtheta[li,ni]);
end;
end;
read(weightfile,alp);read(weightfile,eps); close(weightfile);
end;
PROCEDURE WRITE_WEIGHTS;
var
li,ni,no,no_end,ni_end:byte;
begin
GetTime(h2,m2,s2,hund2); assign(weightfile,'v'+d+'.txt'); rewrite(weightfile);
writeln(weightfile,d,' Transistörüne ait ağırlıklar ve offsetler.'j,' iterasyon için aşağı-');
writeln(weightfile,'da verilmiştir.Sinyal ve gürültü parametreleri için toplam hata',terror,' dır. ');
writeln(weightfile,'Başlama Zamanı=',h1,',',m1,',',s1,'Bitiş zamanı=',h2,',',m2,',',s2,
'Fark= ',h2-h1,' saat ',m2-m1,' dakika ',s2-s1,' Saniyedir. ');
writeln(weightfile,'Gizli Katman Düğüm Sayısı=',node[1],' Alpha=',Alp,' Epsilon=',Eps);

```



```

for li:=Layers_Max-1 downto 1 do
begin
  ni_end:=node[li];  no_end:=node[li-1];
  for no:=0 to no_end-1 do
    begin
      for ni:=0 to ni_end-1 do
        begin
          write(weightfile,' ',weight[li,ni,no]);          write(weightfile,' ',dweight[li,ni,no]);
        end;
        writeln(weightfile,' ');
      end;
    end;
  end;
  for li:=1 to layers-1 do
  begin
    ni_end:=node[li];
    for ni:=0 to ni_end-1 do
      begin
        writeln(weightfile,theta[li,ni]);          writeln(weightfile,dtheta[li,ni]);
      end;
    end;
    writeln(weightfile,alp,eps);  close(weightfile);
  end;
PROCEDURE CALCULATE_S;
type
  dizi=array[0..outputs_max-1,0..output_nodes] of real;
  sdizi=array[0..outputs_max-1,0..output_nodes] of string[8];
var
  tampon:text;
  secenek:char;
  device:string[3];
  f,VCE,IC:array[0..50] of real;
  sout:sdizi;
  fdess:dizi;
  fmax,fmin,step,a:real;
  k,l,m:byte;
begin
  writeln('HATA HESABI YAPILSIN İSTİYOR MUSUNUZ ?[E]VET/[H]AYIR');READLN(pass);

```

```

pass:=Upcase(pass);tt:=0; writeln("Test Başlıyor...");writeln; read_weights;
writeln("KUTUPLAMA NOKTASINI GİRİNİZ");writeln("VCE[V]=");READ(OUT[0,0]);
writeln("IC[mA]=");READ(OUT[0,1]);
for k:=0 to inputs-2 do
    out[0,k]:=out[0,k]/10;
writeln("TEK BİR FREKANSI TEST ETMEK İÇİN [T] YE");
writeln("BELLİ BİR FREKANS BANDINI TEST ETMEK İÇİN [B] YE");
writeln("RASGELE AYRIK FREKANSLARI HESAPLAMAK İÇİN [A] YA BASINIZ!!!");
writeln("RASGELE AYRIK KUTUPLAMA NOKTALARINI HESAPLAMAK İÇİN [K] YA
BASINIZ!!!"); tercih:=readkey;tercih:=Upcase(tercih);
case tercih of
    'T': begin write("FREKANSI [GHz] OLARAK GİRİNİZ f=");readln(OUT[0,2]);samplecount:=1;END;
    'B': begin
        write("FREKANSIN ÜST SINIRINI [GHz] OLARAK GİRİNİZ ' ');readln(fmax);
        write("FREKANSIN ALT SINIRINI [GHz] OLARAK GİRİNİZ ' ');readln(fmin);
        write("FREKANS ADIMINI [GHz] OLARAK GİRİNİZ ' ');readln(step);clrscr;
        out[0,2]:=fmin-step;        samplecount:=round((fmax-fmin)/step+1);
    end;
    'A': begin
        write("AYRIK FREKANSLARI BELİRLİ DOSYADAN OKUTMAK İSTER MİSİNİZ? [E/H]");
        secenek:=readkey;secenek:=Upcase(secenek);clrscr;
        if secenek='E' then
            begin
                assign(tampon,'c:\tp\hamid\btampon1.dat');        reset(tampon);
            end;
            writeln("KAÇ AYRIK FREKANS İÇİN TEST ETTİRMEK İSTİYORSUNUZ? ");
            readln(samplecount);
            for k:=0 to samplecount-1 do
                begin
                    if secenek='E' then read(tampon,f[k])
                    else
                        begin
                            write(k,'INCI ÖRNEĞİ GİRİNİZ ');        readln(f[k]);
                        end;
                end;
            if secenek='E' then close(tampon);
        end;
end;

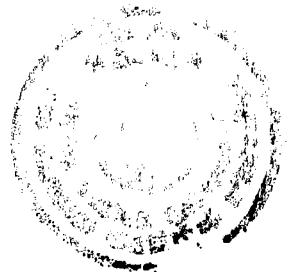
```



```

'K': begin
    write('ÇALIŞMA FREKANSINI GİRİNİZ [GHz] f=');READLN(OUT[0,2]);
    writeLN('AYRIK KUTUPLAMA NOKTALARINI BELİRLİ DOSYADAN OKUTMAK İSTER
MİSİNİZ? [E/H]');
    secenek:=readkey;secenek:=Upcase(secenek);clrscr;
    if secenek='E' then
        begin
            assign(tampon,'c:\tp\hamid\btamponK.dat');          reset(tampon);
            end;
        writeln('KAÇ AYRIK KUTUPLAMA NOKTASI İÇİN TEST ETTİRMEK İSTİYORSUNUZ?
'); readln(samplecount);
        for k:=0 to samplecount-1 do
            begin
                if secenek='E' then read(tampon,VCE[k],IC[k])
                    else
                        begin
                            write(k,'INCI ÖRNEĞİ GİRİNİZ.VCE= '); readln(VCE[k]);
                            write('IC='); readln(IC[k]);
                        end;
                end;
            if secenek='E' then close(tampon);
        end;
    end;
writeln('SONUÇLARI PRİNTERDEN ALMAK İÇİN [P]');
writeln('SONUÇLARI MONİTÖRDEN ALMAK İÇİN [M] E BASINIZ. ');
readln(secenek);secenek:=Upcase(secenek);clrscr;
case secenek of
    'M':device:='';
    'P':device:='lst';
end;
if device='lst' then
    begin
        writeln (lst,'Freq. VCE IC Calculated S Parameters ');
        writeln (lst,' GHz [V] [mA] S11 S21 S12 S22 ');
        writeln (lst,'-----');
    end
else

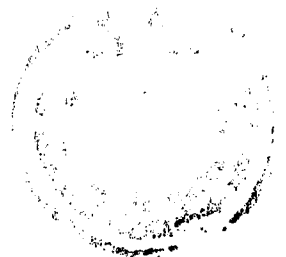
```



```

begin
    writeln ('Freq. VCE IC      Calculated S Parameters ');
    writeln (' GHz  [V] [mA]  S11      S21      S12      S22 ');
    writeln ('-----');
end;
assign(graphfile,'h'+d+'.dat');    {d:=copy(d,1,7);}  rewrite(graphfile);
for m:=0 to samplecount-1 do {Repeat}
begin
    case tercih of
        'A':out[0,2]:=f[m];
        'B':out[0,2]:=out[0,2]+step;
        'T':out[0,2]:=out[0,2];
        'K':Begin
            out[0,1]:=IC[m]/10;      out[0,0]:=VCE[m]/10;
        end;
    end;
    wforward;  str(out[0,2]:6:3,st1);  str(out[0,1]*10:3:1,st2);  str(out[0,0]*10:3:1,st3);
    if device='lst' then write(lst,st1+' ') else write(st1+' ');
    if device='lst' then write(lst,st3+' ') else write(st3+' ');
    if device='lst' then write(lst,st2+' ') else write(st2+' ');
    for l:= 0 to outputs-1 do
    begin
        str(out[2,l]:5:3,st1);
        case (l) of
            1,3,5,7:begin
                if (out[2,l] > 0.5) then
                begin
                    out[2,l]:=out[2,l]*360-360;          str(out[2,l]:7:2,st1);
                end
                else
                begin
                    out[2,l]:=out[2,l]*360;          str(out[2,l]:7:2,st1);
                end;
            end;
            4:begin
                out[2,l]:=out[2,l]/s12max;          str(out[2,l]:5:3,st1);
            end;
        end;
    end;
end;

```



```

2:begin
    out[2,1]:=out[2,1]*s21max;          str(out[2,1]:5:3,st1);
    end;
end;
if device='lst' then write(lst,st1+' ') else write(st1+' ');
end;
begin
    str(out[0,0]:6:3,sout[0,0]);      str(out[0,1]:6:3,sout[0,1]);      str(out[0,2]:6:3,sout[0,2]);
    for l:=0 to outputs-1 do
        str(out[2,1]:7:4,sout[2,1]);
        writeln(graphfile,sout[0,0],',',sout[0,1],',',sout[0,2],',',sout[2,0],',',sout[2,1],',',sout[2,2],',',
        ,sout[2,3],',',sout[2,4],',',sout[2,5],',',sout[2,6],',',sout[2,7]);          a:=1;
    end;
    if device='lst' then writeln(lst,' ') else writeln;
end;{until (fmax <= out[0,0]);} close(graphfile);
if device="" then writeln("DEVAM ETMEK İÇİN [enter] TUŞUNA BASINIZ!!! ");
if device="" then readln;
if device='lst' then writeln(lst,' ') else writeln; str(out[0,0]*10:3:1,st1); str(out[0,1]*10:3:1,st2);
if device='lst' then
begin
    writeln(lst,'Yukarıdaki sonuçlar '+ st1+' '+st2+' kutuplama noktası için elde edilmiştir. ');
    writeln(lst,' ');
    writeln (lst,'Freq. VCE IC          Calculated S Parameters ');
    writeln (lst,' GHz [V] [mA] S11      S21      S12      S22 ');
    writeln (lst,'-----');
end
else
begin
    writeln("Yukarıdaki sonuçlar '+ st1+' '+st2+' kutuplama noktası için elde edilmiştir. ');      writeln;
    writeln ('Freq. VCE IC          Calculated S Parameters ');
    writeln (' GHz [V] [mA] S11      S21      S12      S22 ');
    writeln ('-----');
end;
end;
reset(idatafile);
for l:=0 to outputs_max-1 do
for m:=0 to output_nodes do
    read(idatafile,fdess[l,m]);      clrscr;

```



```

for k:=0 to outputs_max-1 do
for l:= 0 to output_nodes-1 do
begin
case (l) of
1,3,5,7:begin
if dесс[k,l] > 0.5 then
begin
dесс[k,l]:=dесс[k,l]*360;           dесс[k,l]:=dесс[k,l]-360;
end
else
begin
dесс[k,l]:=dесс[k,l]*360;
end;
end;
4:dесс[k,l]:=dесс[k,l]/s12max;
2:dесс[k,l]:=dесс[k,l]*s21max;
end;
end;
for l:=0 to outputs_max-1 do
begin
str(fdесс[l,0]:5:2,st1);
if device='lst' then write(lst,st1+' ') else write(st1+' ');
for m:=0 to output_nodes-1 do
begin
str(dесс[l,m]:5:3,st1);
case (m) of
1,3,5,7:begin
str(dесс[l,m]:7:2,st1);
end;
end;
if device='lst' then write(lst,st1+' ') else write(st1+' ');
end;
if device='lst' then writeln(lst,' ') else writeln(' ');
end;
if device='' then writeln('DEVAM ETMEK İÇİN [enter] TUŞUNA BASINIZ!!! ');
if device='' then readln;   close(idatafile);
end; {Procedure Calculate_S }

```





PROCEDURE HATA\_RAPORU;

type

dizi=array[0..outputs\_max-1,0..output\_nodes] of real;  
sdizi=array[0..outputs\_max-1,0..output\_nodes] of string[8];

var

data:string[100];  
fdess,calc,error:dizi;  
terror:sdizi;  
toplamhata,s,step,a,s11,s12,s21,s22,s\_hata,reel,imajinel:REAL;  
st1:string[4];  
hata,hata1,hata2,hata3,hata4,shata,vce,ic:string[8];  
thata:array [0..100] of REAL;  
k,l,tt,m,N:byte;  
odosya,hdosya,edosya:text;

begin

assign(hdosya,'h'+d+'.dat'); assign(odosya,'bfq74'+'.dat'); assign(edosya,'e'+d+'.dat');  
reset(hdosya); reset(odosya);tt:=0;k:=0; append(edosya);s11:=0;s12:=0;s21:=0;s22:=0;  
writeln(edosya,'FREKANS[GHz]      S11              S12              S21              S22');  
for N:=0 to samplecount-1 do

begin

m:=0; readln(hdosya,calc[0,1],calc[0,2],calc[0,0],calc[2,0],calc[2,1],calc[2,2],calc[2,3],calc[2,4],  
calc[2,5],calc[2,6],calc[2,7]); calc[0,1]:=calc[0,1]\*10; calc[0,2]:=calc[0,2]\*10;

repeat

inc(m);

if m>100 then begin

writeln('DİKKAT TEST EDİLEN DEĞERİN ÖLÇÜLMÜŞ KARŞILIĞI YOK!!!');

writeln('HATA HESAPLANAMAZ PROGRAM KIRILDI!!!');exit;

END;

readln(odosya,fdess[0,1],fdess[0,2],fdess[0,0],fdess[2,0],fdess[2,1],fdess[2,2],fdess[2,3],fdess[2,4],  
fdess[2,5],fdess[2,6],fdess[2,7]);

if not((abs(fdess[0,1]-calc[0,1])<0.00001) and (abs(fdess[0,2]-calc[0,2])<0.00001)) then

begin

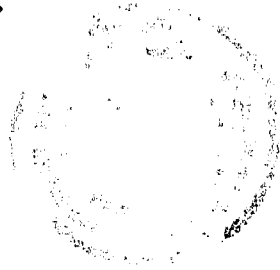
repeat

readln(odosya,fdess[0,1],fdess[0,2],fdess[0,0],fdess[2,0],fdess[2,1],fdess[2,2],fdess[2,3],

fdess[2,4],fdess[2,5],fdess[2,6],fdess[2,7]);

until (abs(fdess[0,1]-calc[0,1])<0.00001) and (abs(fdess[0,2]-calc[0,2])<0.00001);

end;



```

until (abs(fdess[0,0]-calc[0,0])<0.00001);
str(fdess[0,0]:4:1,st1);
a:=1;s:=0;write(odosya,' '+st1+' ');
for k:=0 to outputs-1 do
begin
    error[2,k]:=abs(abs(fdess[2,k])-abs(calc[2,k]))/abs(fdess[2,k]);    s:=s+error[2,k];
    str(error[2,k]:8:6,terror[2,k]);    {write(odosya,' ',terror[2,k]);}
end;
k:=0;
for l:=0 to 3 do {3 for s parameters}
begin
    reel:=sqr(fdess[2,k]*cos(fdess[2,k+1]*pi/180)-calc[2,k]*cos(calc[2,k+1]*pi/180));
    imajinel:=sqr(fdess[2,k]*sin(fdess[2,k+1]*pi/180)-calc[2,k]*sin(calc[2,k+1]*pi/180));
    error[2,k]:=sqrt(reel+imajinel)/abs(fdess[2,k]);
    case k of
        0:s11:=s11+error[2,k];
        2:s21:=s21+error[2,k];
        4:s12:=s12+error[2,k];
        6:s22:=s22+error[2,k];
    end;
    str(error[2,k]:8:6,terror[2,k]);    write(odosya,' ',terror[2,k]);    k:=k+2;
end;
writeln(odosya,' ');    thata[tt]:=s/outputs;
writeln(odosya,'BU FREKANSA AİT TOPLAM MUTLAK HATA= ',thata[tt]);inc(tt);
end;{until (eof(odosya));}
close(odosya); close(hdosya); s:=0;
for tt:=0 to samplecount-1 do s:=s+thata[tt]; toplamhata:=s/samplecount;
s11:=s11/samplecount;s12:=s12/samplecount; s21:=s21/samplecount;s22:=s22/samplecount;
str(s11:9:6,hata1); str(s21:9:6,hata2); str(s12:9:6,hata3); str(s22:9:6,hata4);
str(toplamhata:9:6,hata); s_hata:=(s11+s12+s21+s22)/4; str(s_hata:9:6,shata);
str(calc[0,1]:5:1,vce); str(calc[0,2]:5:1,ic); assign(weightfile,ad+'.txt'); reset(weightfile);
readln(weightfile,data); data:=copy(data,51,6); writeln(odosya,'VCE=' +vce+' [Volt] IC=' +ic+' [mA]
kutuplama noktasında'); writeln(odosya,data+' adet iterasyon için '+d+' transistörüne ait');
writeln(odosya,'Gerçek Mutlak Toplam Hata = ',hata); writeln(odosya,'E11=' ,hata1,' E21=' ,hata2,'
E12=' ,hata3,' E22=' ,hata4); writeln(odosya,'Et= ',shata); close(odosya); close(weightfile);
end; {Procedure Hata_Raporu}

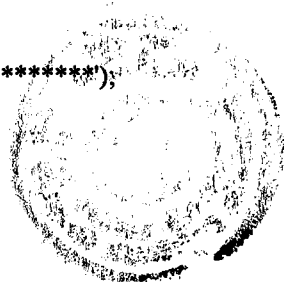
```



```

BEGIN { main }
  clrscr; arrange; hedef_belle;
  { print first header }
  node[2]:= Output_Nodes;
  Outputs:= Output_Nodes;   { Çıkış Katmanı Düğüm Sayısı }
  node[1]:= Hidden_Nodes;   { Gizli Katman Düğüm Sayısı }
  node[0]:= Input_Nodes;
  Inputs:= Input_Nodes;     { Giriş Katmanı Düğüm Sayısı }
  Layers:= Layers_Max;     { Katman Sayısı }
  num:= Outputs_Max;       { Örnek Sayısı }
  alp:=alpha; eps:=epsilon;
  { *****
  INITIALIZE MATRIX
  ***** }
  for i:= 0 to num-1 do
  for j:= 0 to ( inputs+Outputs-1 ) do
  matrix[i,j]:= 0;
  for ii:= 0 to ( Trunc(Iterative_Max/Every)-1 ) do
  for i:= 0 to num-1 do
  for j:= 0 to ( inputs+Outputs-1 ) do
  matrix_tmp[ii,i,j]:=0;
  for i:=1 to 24 do
  writeln;
  write('EĞİTME İÇİN (0) VEYA TEST İÇİN (1) E BASINIZ. '); readln(tt);
  for i:=1 to 24 do writeln;
  if tt=1 then calculate_s
  else
  begin
  writeln('Eğitimle İlgili Bilgileri Direkt Dosyadan Okutmak İsterseniz [D] ye');
  writeln('Program İçinde Oluşturduğunuz Alt Programdan Okutmak İsterseniz [A] ya basınız. ');
  readln(pass);pass:=upcase(pass);if pass='D' then hiz:='slow' else hiz:='fast';   pass:='H';
  assign(outfile,'C:\TPHAMID\BSoutfile.txt');   STR(EPSILON:4:2,ST5);STR(ALPHA:4:3,ST6);
  writeln ('*****');
  WRITELN (' EPSİLON=',st5, ' ', 'MIKRODALGA TRANSİSTORUNUN YNA İLE
SİMÜLASYONU..... ' );
  WRITELN (' ALPHA =',st6, ' '           { HAMİD TORPİ AĞUSTOS\ 1994});
  WRITELN ('*****');

```



```

writeln;append(outfile);
write('MEVCUT AĞIRLIKLARDAN YARARLANMAK İSTİYOR MUSUNUZ? [E/H] ');
readln(tercih);tercih:=Ucase(tercih); if tercih='E' then read_weights else wrand;
GetTime(h1,m1,s1,hund1); j:=0;dcount:=trunc(iterative_max/every)+1;
{*****}
LEARNING PHASE
***** }
while ( j < Iterative_Max ) do
begin
for count:= 0 to Every-1 do
begin
error:= 0;
for isylla:=0 to num-1 do
begin
initial;
if hiz='slow' then ayread( isylla ) else yread(isylla); wforward;
err:= back; learning( alp, eps ); error:= error+err;
end;
error:= error / outputs; eps:= 0.5 * sqrt( error ); Inc( j );
end;
{*****}
TESTING PHASE
***** }
terror:= 0;
for isylla:= 0 to num-1 do
begin
if hiz='slow' then ayread( isylla ) else yread(isylla); wforward; terr:= back;
terror:= terror + terr; make_matrix( trunc( j/Every ) - 1, isylla );
end;
write (outfile, ' Freq. Bias Calculated S Parameters ');
writeln (outfile, ' Desired S Parameters ');
write (outfile, 'NUM GHz Point S11 S21 S12 S22 ');
writeln (outfile, ' S11 S21 S12 S22 ');
write (outfile, '-----');
writeln (outfile, '-----');
for i:= 0 to num-1 do
begin

```



```

str( i:2, st1 );      write(outfile,st1+' ');
for ii:= 0 to inputs-1 do
begin
  str( matrix_tmp[trunc(j/Every)-1,i,ii]:6:3, st1 );      write(outfile,st1+' ');
end;
write(outfile,' ');
for ii:= 0 to outputs-1 do
begin
  str( matrix_tmp[trunc(j/Every)-1,i,inputs+ii]:5:3, st1);      write( outfile,st1+' ');
end;
write(outfile,' ');
for ii:=0 to outputs-1 do
begin
  str( dess[i,ii]:5:3, st1 );      write( outfile, st1+' ');
end;
Writeln(outfile,' ');
end;
terror:= terror/outputs;      str ( j:7, st1 ); str ( eps:9:6, st2 );
str ( error:9:6, st3 ); str ( terror:9:6, st4 );      writeln( outfile, 'iteration=',st1,' eps=',st2,
' error_1=', st3, ' error_t=', st4 );
if ( j = Iterative_Max ) then
begin
  writeln( outfile, 'learn & test finished at time alpha =',st6,' epsilon =',st5 );
  writeln( outfile,d,' Transistoruna ait eğitim sonuçları yukarıda verilmiştir. ');
end;
write_weights;
end; {İlk büyük döngü burada bitiyor}
close( outfile );
end;
if (pass='E') then Hata_Raporu;
end.

```

#### EK 1.1.2 Sabit Kutuplama İçin Sinyal ve Gürültü Parametreleri Simülatörü

```

PROGRAM MULTILAYERPERCEPTRON;
{$N+}
uses crt,printer;

```



CONST

```

Layers_Max = 3;    { Maximum number of layers }
Nodes_Max  = 12;   { Maximum number of node in any layer }
Outputs_Max = 16;  { Number of input combinations }
Input_Nodes = 2;   { Number of input layer nodes }
Output_Nodes = 12; { Number of output layer nodes }
Hidden_Nodes = 12; { Number of hidden layer nodes }
Epsilon    = 0.5;
Alpha      = 0.4;
Iterative_Max = 250000; { Maximum iteration number }
Every       = 50000; { After every iteration print results }

```

TYPE

```
adizi = array [ 0..outputs_Max-1 ] of real;
```

VAR

```

j,count:longint;
f:adizi;
samplecount,i,ii,tt,isylla, num,dcount,Layers, Inputs, Outputs:byte;
fmax,fmin,step,eps, alp, err, error, terr, terror, total_rate, des : double;
Node : array [0..Layers_Max-1] of byte;
matrix : array [0..Outputs_Max-1, 0..(Input_Nodes+Output_Nodes-1)] of double;
matrix_tmp: array [0..trunc(Iterative_Max/Every), 0..Outputs_Max-1,
0..(Input_Nodes+Output_Nodes-1)] of double;
out : array [0..Layers_Max-1, 0..Nodes_Max-1] of double;
der : array [0..Layers_Max-1, 0..Nodes_Max-1] of double;
theta : array [0..Layers_Max-1, 0..Nodes_Max-1] of double;
dtheta : array [0..Layers_Max-1, 0..Nodes_Max-1] of double;
weight : array [0..Layers_Max-1, 0..Nodes_Max-1, 0..Nodes_Max-1] of double;
dweight : array [0..Layers_Max-1, 0..Nodes_Max-1, 0..Nodes_Max-1] of double;
dess : array [0..Outputs_Max-1,0..output_nodes-1] of double;
st1: string[7]; st5,st6: string[6];
d,st2, st3, st4: string[12];
idatafile,datafile,graphfile,outfile,weightfile :text;
tercih,pass:char;
beta:real;
hiz,ad:string[8];

```

PROCEDURE ARRANGE;

var



```

dosya,ydosya:text;
A:string[1];
q:string[80];
begin
write(' OKUNACAK DOSYANIN ADINI GİRİNİZ. ');READLN(d);
assign(dosya,'c:\appcad\spara\necspar'+d+'.s2p'); assign(ydosya,d+'.dat'); reset(dosya);
repeat
  readln(dosya,q);  a:=copy(q,1,1);
until (a='Z');
rewrite(ydosya);
while not eof(dosya) do
begin
  readln(dosya,q);  writeln(ydosya,q);
end;
close(ydosya); close(dosya);
end; {Procedure arrange}
PROCEDURE HEDEF_BELLE;
type
  dizi=array[0..outputs_max-1,0..output_nodes] of real;
var
  tampon:text;
  yanit:char;
  fdess:dizi;
  x,y:real;
  k,l,m:byte;
  ST:string[8];
begin
  assign(tampon,'tampon.dat');
  write('EN SON KULLANILAN FREKANS ÖRNEKLERİNİ KULLANMAK İSTER MİSİNİZ?
[E/H]'); readln(yanit); yanit:=upcase(yanit);
  if yanit='E' then
  begin
    reset(tampon);
    for l:=0 to outputs_max-1 do
    begin
      READ(tampon,F[l]);
    end;
  end;

```



```

end
else
begin
rewrite(tampon); writeln(' HANGİ FREKANSLARDA ÖRNEK ALACAKSANIZ ');
writeln(' SIRAYLA [GHz] OLARAK GİRİNİZ ');
for l:=0 to outputs_max-1 do
begin
write(' ');write (l+1);write(' ÖRNEK = ');READ(F[l]);write(tampon,f[l]);write(' GHz de');
writeln;
end;
writeln(' ALINACAKTIR. ');
end;
close(tampon); assign(datafile,d+'.dat'); reset(datafile);m:=0;
assign(idatafile,'i'+d+'.dat'); rewrite(idatafile);m:=0;
while not eof(datafile) do
begin
read(datafile,x);
for l:=m to outputs_max-1 do
begin
while not (x=f[l]) do
begin
while not eoln(datafile) do
read(datafile,x); read(datafile,x);
end;
for k:=0 to output_nodes do
begin
fdess[m,k]:=x; read(datafile,x);
end;
for k:=0 to output_nodes do
begin
y:=fdess[m,k];str(Y:6:3,ST); write(idatafile,' ',st);
end;
writeln(idatafile,' ');
m:=m+1;
end;
end;
end;
m:=0;

```

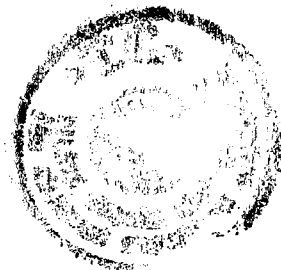




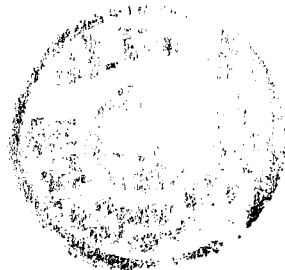
```

close(datafile); close(idatafile);
for m:=0 to outputs_max-1 do
for k:=0 to output_nodes-1 do
dess[m,k]:=fdess[m,k+1];
clrscr;
for k:=0 to outputs_max-1 do
for l:=0 to output_nodes-1 do
begin
case (l) of
1,3,5,7,10:begin
if dess[k,l] < 0 then
begin
dess[k,l]:=dess[k,l]+360; dess[k,l]:=dess[k,l]/360;
end
else
begin
dess[k,l]:=dess[k,l]/360;
end;
end;
4:dess[k,l]:=dess[k,l]*5;
2:dess[k,l]:=dess[k,l]/10;
8:dess[k,l]:=dess[k,l]/5;
end;
end;
end; {Procedure hedef_belle}
PROCEDURE WRAND;
PROCEDURE INITIAL;
PROCEDURE YREAD ( isylla: byte );
{ read input pattern ( for several different input combination ) }
begin
case isylla of
0: out[0,0]:=1;
1: out[0,0]:=2;
2: out[0,0]:=4;
3: out[0,0]:=6;
4: out[0,0]:=8;
5: out[0,0]:=10;

```



```
6: out[0,0]:=12;
7: out[0,0]:=14;
8: out[0,0]:=16;
9: out[0,0]:=18;
10: out[0,0]:=20;
11: out[0,0]:=22;
12: out[0,0]:=24;
13: out[0,0]:=26;
14: out[0,0]:=28;
15: out[0,0]:=30;
{ case isylla of
0: out[0,0]:=0.1 ;
1: out[0,0]:=0.5;
2: out[0,0]:=1;
3: out[0,0]:=1.5;
4: out[0,0]:=2;
5: out[0,0]:=2.5;
6: out[0,0]:=3;
7: out[0,0]:=3.5;
8: out[0,0]:=4;
9: out[0,0]:=4.5;
10: out[0,0]:=5;
11: out[0,0]:=5.5;
12: out[0,0]:=6;
13: out[0,0]:=6.5;
14: out[0,0]:=7;
15: out[0,0]:=7.5;
16: out[0,0]:=8;
17: out[0,0]:=8.5;
18: out[0,0]:=9;
19: out[0,0]:=9.5;
20: out[0,0]:=10;
21: out[0,0]:=10.5;
22: out[0,0]:=11;
23: out[0,0]:=11.5;
24: out[0,0]:=12;
25: out[0,0]:=12.5;
```



```

26:out[0,0]:=13;
27:out[0,0]:=13.5;
28:out[0,0]:=14;
29:out[0,0]:=14.5;
30:out[0,0]:=15;
31:out[0,0]:=15.5;
32:out[0,0]:=16;
33:out[0,0]:=16.5;
34:out[0,0]:=17;
35:out[0,0]:=17.5;
36:out[0,0]:=18;
37:out[0,0]:=18.5;
38:out[0,0]:=19;
39:out[0,0]:=19.5;
40:out[0,0]:=20;}
end;
end; { procedure YREAD }
PROCEDURE BIASPOINT;
{ read input pattern ( for several different input combination ) }
var
  s:real;
  t:real;
  l:byte;
  odosya:text;
begin
  Assign(odosya,'i'+d+'.dat'); reset(odosya); Read(odosya,s);T:=0;
  while not eof(odosya) do
    begin
      t:=t+sqr(s);
      for l:=0 to output_nodes do
        read(odosya,s);
      end;
      out[0,l]:=sqrt(t/outputs_max); close(odosya);
    end;
  END;
PROCEDURE AYREAD(ISYLLA:BYTE);
{ read input pattern ( for several different input combination ) }
var

```



```

a:string[80];
x:real;
l,m:byte;
odosya:text;
begin
  Assign(odosya,'i'+d+'.dat'); reset(odosya);
  if (isylla=0) then read(odosya,out[0,0])
  else
    begin
      for l:=1 to isylla do
        begin
          if (l=isylla) then
            begin
              for m:=0 to outputs do
                read(odosya,x);      read(odosya,out[0,0]);
            end;
            readln(odosya,a);
          end;
        end;
      close(odosya);
    end; {Procedure AYread}
  {OLD PROCEDURE AYREAD(ISYLLA:BYTE);
  read input pattern ( for several different input combination )
  var
    s:real;
    x,t:real;
    l,m:byte;
    odosya:text;
  begin
    Assign(odosya,'o'+d+'.dat'); reset(odosya); Read(odosya,s);T:=0;
    while not eof(odosya) do
      begin
        t:=t+sqr(s);
        for l:=0 to outputs do
          read(odosya,s);
        end;
        out[0,1]:=sqrt(t/num); reset(odosya);

```



```

if (isylla=0) then read(odosya,out[0,0])
else
begin
for l:=1 to isylla do
for m:=0 to outputs do
read(odosya,x);      read(odosya,out[0,0]);
end;
close(odosya);
end; Procedure AYread}
{OLD FUNCTION SIGMOID ( x: double ): double;
sigmoid function
var
y: double;
begin
y:=1/(1+exp(-(x))); sigmoid:=y;
end; function SIGMOID }
FUNCTION SIGMOID ( x: double ): double;
{ sigmoid function }
var
y: double;
begin
Beta:=1;
if x >= 1000000000 then y:=1
else
begin
if x <= -1000000000 then y:=0
else y:=1/(1+exp(-beta*(x)));
end;
sigmoid:=y;
end; { function SIGMOID }
FUNCTION TANHIPER ( x: double ): double;
{ tanhiper function }
var
y: double;
begin
Beta:=1;
if x >= 1000000000 then y:=1

```



```

else
begin
if x <= -1000000000 then y:=-1
else y:=(exp(beta*(x))-exp(-beta*(x)))/(exp(beta*(x))+exp(-beta*(x)));
end;
tanhiper:=y;
end; { function TANHIPER }
PROCEDURE WFORWARD ;
FUNCTION BACK: double;
PROCEDURE LEARNING (alp, eps: double );
PROCEDURE MAKE_MATRIX( it, isylla: integer );
PROCEDURE READ_WEIGHTS;
var
data:string[100];
li,ni,no,no_end,ni_end: byte;
begin
write ('OKUNACAK AĞIRLIK DOSYASININ ADINI GİRİNİZ. ');readln(ad);
assign(weightfile,ad+'.txt'); reset(weightfile); readln(weightfile,data); readln(weightfile,data);
for li:=Layers_Max-1 downto 1 do
begin
ni_end:=node[li]; no_end:=node[li-1];
for no:=0 to no_end-1 do
begin
for ni:=0 to ni_end-1 do
begin
read(weightfile,weight[li,ni,no]); read(weightfile,dweight[li,ni,no]);
end;
end;
end;
end;
for li:=1 to layers-1 do
begin
ni_end:=node[li];
for ni:=0 to ni_end-1 do
begin
readln(weightfile,theta[li,ni]); readln(weightfile,dtheta[li,ni]);
end;
end;
end;
end;

```



```

    read(weightfile,alp);read(weightfile,eps); close(weightfile);
end;
PROCEDURE WRITE_WEIGHTS;
var
    li,ni,no,no_end,ni_end:byte;
begin
    assign(weightfile,'v'+d+'.txt'); rewrite(weightfile);
    writeln(weightfile,d,' Transistörüne ait ağırlıklar ve offsetler.',j,' iterasyon için aşağı-');
    writeln(weightfile,'da verilmiştir.Sinyal ve gürültü parametreleri için toplam hata',terror,' dir. ');
    for li:=Layers_Max-1 downto 1 do
    begin
        ni_end:=node[li]; no_end:=node[li-1];
        for no:=0 to no_end-1 do
            begin
                for ni:=0 to ni_end-1 do
                    begin
                        write(weightfile,',weight[li,ni,no]); write(weightfile,',dweight[li,ni,no]);
                    end;
                    writeln(weightfile,');
                end;
            end;
        end;
        for li:=1 to layers-1 do
            begin
                ni_end:=node[li];
                for ni:=0 to ni_end-1 do
                    begin
                        writeln(weightfile,theta[li,ni]); writeln(weightfile,dtheta[li,ni]);
                    end;
                end;
            end;
        writeln(weightfile,alp,eps); close(weightfile);
    end;
PROCEDURE CALCULATE_SF;
type
    dizi=array[0..outputs_max-1,0..output_nodes] of real;
    sdizi=array[0..layers_max-1,0..output_nodes] of string[8];
var
    tercih,secenek:char;

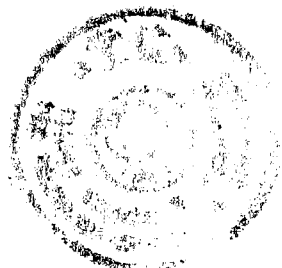
```



```

device:string[3];
f:array[0..50] of real;
sout:sdizi;
fdess:dizi;
fmax,fmin,step,a:real;
k,l,m:byte;
begin
  writeln('HATA HESABI YAPILSIN İSTİYOR MUSUNUZ ?[E]VET/[H]AYIR');READLN(pass);
  pass:=Ucase(pass);tt:=0; writeln('Test Başlıyor...');writeln; read_weights;
  writeln('TEK BİR FREKANSI TEST ETMEK İÇİN [T] YE');
  writeln('BELLİ BİR FREKANS BANDINI TEST ETMEK İÇİN [B] YE');
  writeln('RASGELE AYRIK FREKANSLARI HESAPLAMAK İÇİN [A] YA BASINIZ!!!');
  READ(tercih);tercih:=Ucase(tercih);
  case tercih of
    'T': begin write('FREKANSI [GHz]OLARAK GİRİNİZ f=');readln(OUT[0,0]);samplecount:=1;END;
    'B': begin
      write('FREKANSIN ÜST SINIRINI [GHz] OLARAK GİRİNİZ ');readln(fmax);
      write('FREKANSIN ALT SINIRINI [GHz] OLARAK GİRİNİZ ');readln(fmin);
      write('FREKANS ADIMINI [GHz] OLARAK GİRİNİZ ');readln(step);clrscr;
      out[0,0]:=fmin-step;      samplecount:=round((fmax-fmin)/step+1);
    end;
    'A': begin
      writeln('KAÇ AYRIK FREKANS İÇİN TEST ETTİRMEK İSTİYORSUNUZ? ');
      readln(samplecount);
      for k:=0 to samplecount-1 do
        begin
          write(k,'INCI ÖRNEĞİ GİRİNİZ ');readln(f[k]);
        end;
      end;
  end;
  writeln('SONUÇLARI PRİNTERDEN ALMAK İÇİN [P]');
  writeln('SONUÇLARI MONİTÖRDEN ALMAK İÇİN [M] E BASINIZ. ');
  readln(secenek);secenek:=Ucase(secenek);clrscr;
  case secenek of
    'M':device:='';
    'P':device:='lst';
  end;
end;

```

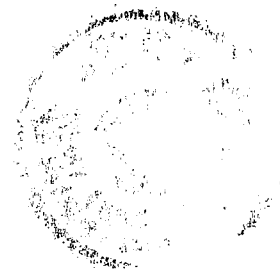




```

if device='lst' then
begin
  writeln (lst,'Freq.          Calculated S Parameters      &      Noise Parameters');
  writeln (lst,' GHz      S11      S21      S12      S22      Fmin      ropt      Rn/50 ');
  writeln (lst,'-----');
end
else
begin
  writeln ('Freq.          Calculated S Parameters      &      Noise Parameters');
  writeln (' GHz      S11      S21      S12      S22      Fmin      ropt      Rn/50 ');
  writeln ('-----');
end;
assign(graphfile,'h'+d+'.dat'); {d:=copy(d,1,7);} rewrite(graphfile);
for m:=0 to samplecount-1 do
begin
  case tercih of
    'A':out[0,0]:=f[m];
    'B':out[0,0]:=out[0,0]+step;
    'T':out[0,0]:=out[0,0];
  end;
  wforward;  str(out[0,0]:4:1,st1);
  if device='lst' then write(lst,st1+' ') else write(st1+' ');
  for l:= 0 to outputs-1 do
  begin
    str(out[2,l]:5:3,st1);
    case (l) of
      1,3,5,7,10:begin
        if (out[2,l] > 0.5) then
        begin
          out[2,l]:=out[2,l]*360-360;          str(out[2,l]:6:1,st1);
        end
        else
        begin
          out[2,l]:=out[2,l]*360;          str(out[2,l]:6:1,st1);
        end;
      end;
    end;
  4:begin

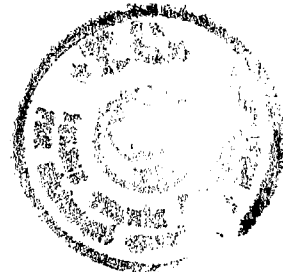
```



```

        out[2,1]:=out[2,1]/5;          str(out[2,1]:5:3,st1);
    end;
2:begin
    out[2,1]:=out[2,1]*10;          str(out[2,1]:5:3,st1);
    end;
8:begin
    out[2,1]:=out[2,1]*5;          str(out[2,1]:5:3,st1);
    end;
end;
if device='lst' then write(lst,st1+' ') else write(st1+' ');
end;
str(out[0,0]:6:3,sout[0,0]);
for l:=0 to outputs-1 do
    str(out[2,1]:7:4,sout[2,1]);
writeln(graphfile,sout[0,0],' ',sout[2,0],' ',sout[2,1],' ',sout[2,2],' ',sout[2,3],' ',sout[2,4],' ',sout[2,5],'
',sout[2,6],' ',sout[2,7],' ',sout[2,8],' ',sout[2,9],' ',sout[2,10],' ',sout[2,11]);
if device='lst' then writeln(lst,' ') else writeln;
end;
close(graphfile);
if device="" then writeln('DEVAM ETMEK İÇİN [enter] TUŞUNA BASINIZ!!! ');
if device="" then readln; reset(idatafile);
if device='lst' then writeln(lst,' ') else writeln;
str(out[0,1]:7:3,st1);
if device='lst' then
begin
    writeln(lst,'Yukarıdaki sonuçlar '+ st1+' kutuplama noktası için elde edilmiştir. '); writeln(lst,' ');
    writeln (lst,'Freq. The Used S Parameters & Noise Parameters For Learning');
    writeln (lst,' GHz S11 S21 S12 S22 Fmin ropt Rn/50 ');
    writeln (lst,'-----');
end
else
begin
    writeln('Yukarıdaki sonuçlar '+ st1+' kutuplama noktası için elde edilmiştir. '); writeln;
    writeln ('Freq. The Used S Parameters & Noise Parameters For Learning');
    writeln (' GHz S11 S21 S12 S22 Fmin ropt Rn/50 ');
    writeln ('-----');
end;
end;

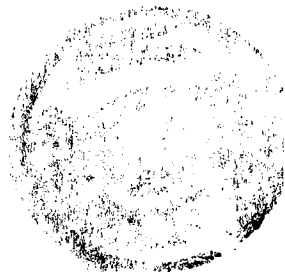
```



```

for l:=0 to outputs_max-1 do
  for m:=0 to output_nodes do
    read(idatafile,fdess[l,m]);  clrscr;
  for k:=0 to outputs_max-1 do
  for l:= 0 to output_nodes-1 do
  begin
    case (l) of
      1,3,5,7,10:begin
        if dесс[k,l] > 0.5 then
          begin
            dесс[k,l]:=dесс[k,l]*360;  dесс[k,l]:=dесс[k,l]-360;
          end
        else
          begin
            dесс[k,l]:=dесс[k,l]*360;
          end;
        end;
      4:dесс[k,l]:=dесс[k,l]/5;
      2:dесс[k,l]:=dесс[k,l]*10;
      8:dесс[k,l]:=dесс[k,l]*5;
    end;
  end;
for l:=0 to outputs_max-1 do
begin
  str(fdess[l,0]:4:1,st1);
  if device='lst' then write(lst,st1+' ') else write(st1+' ');
  for m:=0 to output_nodes-1 do
  begin
    str(dесс[l,m]:5:3,st1);
    case (m) of
      1,3,5,7,10:begin
        str(dесс[l,m]:6:1,st1);
      end;
    end;
  end;
  if device='lst' then write(lst,st1+' ') else write(st1+' ');
end;
if device='lst' then writeln(lst,' ') else writeln(' ');

```



```

end;
if device="" then writeln("DEVAM ETMEK İÇİN [enter] TUŞUNA BASINIZ!!! ");
if device="" then readln;
close(idatafile);
end; {Procedure Calculate_SF }
PROCEDURE HATA_RAPORU;
type
  dizi=array[0..outputs_max-1,0..output_nodes] of real;
  sdizi=array[0..outputs_max-1,0..output_nodes] of string[8];
var
  data:string[100];
  fdess,calc,error:dizi;
  terror:sdizi;
  toplamhata,s,step:double;
  ST1:string[4];
  hata,hata1,hata2,hata3,hata4,hata5,hata6,hata7,shata,fhata:string[8];
  thata:array [0..100] of double; {100 yerine output_nodes-1}
  k,l,tt,m,N,z:byte;
  a,s11,s12,s21,s22,fmin,ropt,rm,f_hata,s_hata,reel,imajinel:real;
  odosya,hdosya,edosya:text;
begin
  assign(hdosya,'h'+d+'.dat'); assign(odosya, d+'.dat'); assign(edosya,'e'+d+'.dat'); reset(hdosya);
  reset(odosya);tt:=0;k:=0; append(edosya);fmin:=0;ropt:=0;rm:=0;s11:=0;s12:=0;s21:=0;s22:=0;
  write(edosya,'FREKANS[GHz]   S11       S12       S21       S22       Fmin');
  writeln(edosya,'      ropt   Rn/50');
  for n:=0 to samplecount-1 do {repeat}
  begin
    m:=0;
    readln(hdosya,calc[0,0],calc[2,0],calc[2,1],calc[2,2],calc[2,3],calc[2,4],
    calc[2,5],calc[2,6],calc[2,7],calc[2,8],calc[2,9],calc[2,10],calc[2,11]);
    repeat
      inc(m);
      read(odosya,fdess[0,0]);      z:=0;
      repeat
        read(odosya,fdess[2,z]);      inc(z);
      until (eoln(odosya));
      if m>100 then begin

```



```

        writeln('DİKKAT TEST EDİLEN DEĞERİN ÖLÇÜLMÜŞ KARŞILIĞI YOK!!');
        writeln('HATA HESAPLANAMAZ PROGRAM KIRILDI!!!');exit;
    end;
until {(a=0) OR} (fdess[0,0]=calc[0,0]);
str(fdess[0,0]:4:1,st1);  a:=1;s:=0;write(edosya,' '+st1+' ');
for k:=0 to outputs-1 do
begin
    error[2,k]:=abs(abs(fdess[2,k])-abs(calc[2,k]))/abs(fdess[2,k]);    s:=s+error[2,k];
    str(error[2,k]:8:6,terror[2,k]);    { write(edosya,' ',terror[2,k]); }
end;
k:=0;
for l:=0 to 6 do {3 for s parameters}
begin
    if k<=8 then
        begin
            reel:=sqr(fdess[2,k]*cos(fdess[2,k+1]*pi/180)-calc[2,k]*cos(calc[2,k+1]*pi/180));
            imajinel:=sqr(fdess[2,k]*sin(fdess[2,k+1]*pi/180)-calc[2,k]*sin(calc[2,k+1]*pi/180));
            error[2,k]:=sqr(reel+imajinel)/abs(fdess[2,k]);
        end;
        case k of
            0:s11:=s11+error[2,k];
            2:s21:=s21+error[2,k];
            4:s12:=s12+error[2,k];
            6:s22:=s22+error[2,k];
            8:begin
                error[2,k]:=abs(fdess[2,k]-calc[2,k])/abs(fdess[2,k]);    fmin:=fmin+error[2,k];    k:=k-1;
            end;
            9:ropt:=ropt+error[2,k];
            11:begin
                error[2,k]:=abs(fdess[2,k]-calc[2,k])/abs(fdess[2,k]);    rn:=rn+error[2,k];
            end;
        end;
        str(error[2,k]:8:6,terror[2,k]);    write(edosya,' ',terror[2,k]);    k:=k+2;
    end;
    writeln(edosya,' ');    thata[tt]:=s/outputs;
    { writeln(edosya,'BU FREKANSA AİT TOPLAM MUTLAK HATA= ',thata[tt]); }    inc(tt);
end; {until (eof(odosya));}

```



```

close(odosya); close(hdosya); s:=0;
for tt:=0 to samplecount-1 do s:=s+thata[tt];
toplamlhata:=s/samplecount; s11:=s11/samplecount;s12:=s12/samplecount;
s21:=s21/samplecount;s22:=s22/samplecount; ropt:=ropt/samplecount; fmin:=fmin/samplecount;
rn:=rn/samplecount; str(s11:9:6,hata1); str(s21:9:6,hata2); str(s12:9:6,hata3); str(s22:9:6,hata4);
str(fmin:9:6,hata5); str(ropt:9:6,hata6); str(rn:9:6,hata7); str(toplamlhata:9:6,hata);
s_hata:=(s11+s12+s21+s22)/4; str(s_hata:9:6,shata); f_hata:=(fmin+ropt+rn)/3;
str(f_hata:9:6,fhata); assign(weightfile,ad+'.txt'); reset(weightfile); readln(weightfile,data);
data:=copy(data,52,6); writeln(edosya,data+' adet iterasyon için '+d+' transistörüne ait');
writeln(edosya,'Gerçek Mutlak Toplam Hata = ',hata);
writeln(edosya,' E11=',hata1,' E21=',hata2,' E12=',hata3,' E22=',hata4); writeln(edosya,'Et= ',shata);
writeln(edosya,' F1=',hata5,' F2=',hata6,' F3=',hata7); writeln(edosya,'Ft= ',fhata); close(edosya);
close(weightfile);
end; {Procedure Hata_Raporu}
BEGIN { main }
  clrscr; arrange; hedef_belle; biaspoint;
  { print first header }
  node[2]:= Output_Nodes;
  Outputs:= Output_Nodes; { Çıkış Katmanı Düğüm Sayısı }
  node[1]:= Hidden_Nodes; { Gizli Katman Düğüm Sayısı }
  node[0]:= Input_Nodes;
  Inputs:= Input_Nodes; { Giriş Katmanı Düğüm Sayısı }
  Layers:= Layers_Max; { Katman Sayısı }
  eps:= Epsilon; alp:= Alpha;
  num:= Outputs_Max; { Örnek Sayısı }
  { *****
  INITIALIZE MATRIX
  ***** }
  for i:= 0 to num-1 do
  for j:= 0 to ( inputs+Outputs-1 ) do
  matrix[i,j]:= 0;
  for ii:= 0 to ( Trunc(Iterative_Max/Every)-1 ) do
  for i:= 0 to num-1 do
  for j:= 0 to ( inputs+Outputs-1 ) do
  matrix_tmp[ii,i,j]:=0;
  for i:=1 to 24 do
  writeln; write('EĞİTME İÇİN (0) VEYA TEST İÇİN (1) E BASINIZ. '); readln(tt);

```



```

for i:=1 to 24 do writeln;
if tt=1 then calculate_sf
else
begin
  writeln('Eğitimle İlgili Bilgileri Direkt Dosyadan Okutmak İsterseniz [D] ye');
  writeln('Program İçinde Oluşturduğunuz Alt Programdan Okutmak İsterseniz [A] ya basınız. ');
  readln(pass);pass:=upcase(pass);if pass='D' then hiz:='slow' else hiz:='fast';   pass:='H';
  assign(outfile,'outfile.txt');   STR(EPSILON:4:2,ST5);STR(ALPHA:4:3,ST6);
  writeln ('*****');
  WRITELN (' EPSİLON=',st5,' ', 'MIKRODALGA TRANSİSTORUNUN YNA İLE
    SİMÜLASYONU..... ' );
  WRITELN (' ALPHA =',st6,' ' { HAMİD TORPİ AĞUSTOS\ 1994});
  WRITELN ('*****');
  writeln;append(outfile);
  write('MEVCUT AĞIRLIKLARDAN YARARLANMAK İSTİYOR MUSUNUZ? [E/H] ');
  readln(tercih);tercih:=Upcase(tercih);
  if tercih='E' then read_weights else wrand;
  j:=0;dcount:=trunc(iterative_max/every)+1;
  {*****
  LEARNING PHASE
  ***** }
  while ( j < Iterative_Max ) do
  begin
    for count:= 0 to Every-1 do
    begin
      error:= 0;
      for isylla:=0 to num-1 do
      begin
        initial;   if hiz='slow' then ayread(isylla) else yread(isylla);   wforward;
        err:= back;   learning( alp, eps );   error:= error+err;
      end;
      error:= error / outputs;   eps:= 0.5 * sqrt( error );   Inc( j );
    end;
    { *****
    TESTING PHASE
    ***** }
    terror:= 0;

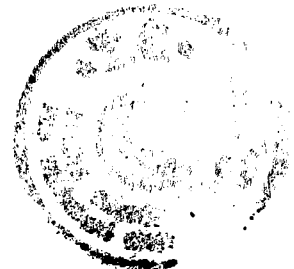
```



```

for isylla:= 0 to num-1 do
begin
  if hiz='slow' then ayread(isylla) else yread(isylla);
  wforward;      terr:= back;      terror:= terror + terr;
  make_matrix( trunc( j/Every ) - 1, isylla );
end;
write (outfile, ' Freq. Bias          Calculated S Parameters   &   Noise Parameters ');
writeln (outfile, '          Desired S Parameters   &   Noise Parameters');
write (outfile, 'NUM GHz Point  S11    S21    S12    S22  Fmin ropt  Rn/50');
writeln (outfile, '  S11    S21    S12    S22  Fmin ropt  Rn/50');
write (outfile, '-----');
writeln (outfile, '-----');
for i:= 0 to num-1 do
begin
  str( i:2, st1 );      write(outfile,st1+' ');
  for ii:= 0 to inputs-1 do
  begin
    str( matrix_tmp[trunc(j/Every)-1,i,ii]:5:3, st1 );      write(outfile,st1+' ');
  end;
  write(outfile,' ');
  for ii:= 0 to outputs-1 do
  begin
    str( matrix_tmp[trunc(j/Every)-1,i,inputs+ii]:5:3, st1 );      write( outfile,st1+' ');
  end;
  write(outfile,' ');
  for ii:=0 to outputs-1 do
  begin
    str( dess[i,ii]:5:3, st1 );      write( outfile, st1+' ');
  end;
  Writeln(outfile,' ');
end;
terror:= terror/outputs;      str ( j:7, st1 ); str ( eps:9:6, st2 );
str ( error:9:6, st3 ); str ( terror:9:6, st4 );
writeln( outfile, 'iteration=',st1, ' eps=',st2, ' error_1=', st3, ' error_t=', st4 );
if ( j = Iterative_Max ) then
begin
  writeln( outfile, 'learn & test finished at time alpha =',st6, ' epsilon =',st5 );

```





```

writeln( outfile, 'Hidden layer nodes ',node[1]);
writeln( outfile,d,' Transistoruna ait eğitim sonuçları yukarıda verilmiştir. ');
end;
write_weights;
end; {İlk büyük döngü burada bitiyor}
close( outfile );
end;
if (pass='E') then Hata_Raporu;;
end.

```

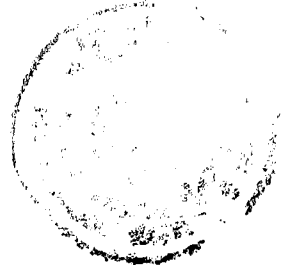
## EK.1.2 Simülasyon Sonuçlarını Smith Aşağına Bastırmak İçin Geliştirilen Program

### EK.1.2.1 Gürültü ve Sinyal Parametreleri İçin

```

Program Plot_SF_BW;
{$N+}
uses crt,dos,graph,printer;
const
  centerK=480; {320}
  centerM=300; {240}
  centerK1=160; {320}
  S21MAX=10;
  S12MAX=0.2;
  S12CARPAN=700;
  S21CARPAN=15;
  adim=3;
var
  Color,birim: Word;
  bilgi:string[30];
  birim1,errcode,l,gd,gm,C,D: integer;
  GrDriver,GrMode,ilk,aralik,kac1,kac2: integer;
  secim: byte;
PROCEDURE CIZIM;
var
  ilk,son : integer;
  arccoords:arccoordstype;
  reel,imajinel : real;

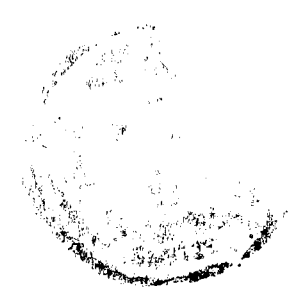
```



```

begin
  reel := 0;  imajinel:= 0.2;
  For ilk := 1 To 15 Do
    begin
      circle(centerk1+round(birim*reel/(1+reel)),centerm,round(birim/(1+reel)));
      reel := reel+0.2;
    end;
  circle(centerk1,centerm,birim+4);
  arc(centerk1+birim,centerm-round(birim/imajinel),247,270,round(birim/imajinel));
  arc(centerk1+birim,centerm+round(birim/imajinel),90,113,round(birim/imajinel));
  getarcoords(arccoords);  imajinel := imajinel+0.4;
  arc(centerk1+birim,centerm-round(birim/imajinel),207,270,round(birim/imajinel));
  arc(centerk1+birim,centerm+round(birim/imajinel),90,153,round(birim/imajinel));
  getarcoords(arccoords);  imajinel := imajinel+0.4;
  arc(centerk1+birim,centerm-round(birim/imajinel),180,270,round(birim/imajinel));
  arc(centerk1+birim,centerm+round(birim/imajinel),90,180,round(birim/imajinel));
  getarcoords(arccoords);  imajinel := imajinel+0.6;
  arc(centerk1+birim,centerm-round(birim/imajinel),151,270,round(birim/imajinel));
  arc(centerk1+birim,centerm+round(birim/imajinel),90,209,round(birim/imajinel));
  getarcoords(arccoords);  imajinel := imajinel+0.8;
  arc(centerk1+birim,centerm-round(birim/imajinel),130,270,round(birim/imajinel));
  arc(centerk1+birim,centerm+round(birim/imajinel),90,230,round(birim/imajinel));
  getarcoords(arccoords);  imajinel := imajinel+1.2;
  arc(centerk1+birim,centerm-round(birim/imajinel),116,270,round(birim/imajinel));
  arc(centerk1+birim,centerm+round(birim/imajinel),90,244,round(birim/imajinel));
  getarcoords(arccoords);  imajinel := imajinel+2;
  arc(centerk1+birim,centerm-round(birim/imajinel),105,270,round(birim/imajinel));
  arc(centerk1+birim,centerm+round(birim/imajinel),90,255,round(birim/imajinel));
  getarcoords(arccoords);  Line(160-birim,centerm,160+birim,centerm);
  setcolor(1);  outtextxy(400,10,'_____MEASURED');
  setcolor(9);  outtextxy(400,20,'_____CALCULATED');
end;
PROCEDURE DCIZIM;
var
  a,b,cap : integer;
  s12g:real;
  ss12g,scap: string[7];

```



```

begin
  cap:=0;   setcolor(0);
  repeat
    setcolor(10);   circle(centerk,centerm,S21CARPAN*CAP);
    s12g:=(s21carpan*cap/s12carpan);   str(cap:3,scap);   str ( s12g:4:3, ss12g);
    setcolor(8);   outtextxy(centerk-s21carpan*cap-23,centerm+2,scap);
    if s12g<>0 then outtextxy(centerk+s21carpan*cap,centerm+2,ss12g);
    cap:= cap+adim;
  until (cap >= s21max);
  b:=centerk+round((s21carpan*cap-60)/2);   a:=centerk-round((s21carpan*cap-30)/2);
  setcolor(8);   outtextxy(a,centerm+15,"S21");   outtextxy(b,centerm+15,"S12");
  Line(centerk-S21CARPAN*cap,CENTERM,centerk+S21CARPAN*cap,CENTERM);
  Line(CENTERK,CENTERM+S21CARPAN*cap,CENTERK,CENTERM-S21CARPAN*cap);
  Line(CENTERK-ROUND(0.6*S21CARPAN*cap),CENTERM+ROUND(0.6*S21CARPAN*cap)
,CENTERK+ROUND(0.6*S21CARPAN*cap),CENTERM-ROUND(0.6*S21CARPAN*cap));
  Line(CENTERK+ROUND(0.6*S21CARPAN*cap),CENTERM+ROUND(0.6*S21CARPAN*cap),
CENTERK-ROUND(0.6*S21CARPAN*cap),CENTERM-ROUND(0.6*S21CARPAN*cap));
end;
PROCEDURE SF_CIZIM;
const
  outputs_max=3;
  output_nodes=12;
type
  dizi=array[0..outputs_max-1,0..output_nodes] of real;
var
  fdess,calc:dizi;
  x_calc,y_calc,x_meas,y_meas,GP1:real;
  odosya,hdosya:text;
  tt,k,m,errorcode,n,i,j,ilk,hata,xm,ym,xc,yc:integer;
  xm11,ym11,xc11,yc11,xm22,ym22,xc22,yc22,roptcx,roptcy,roptmx,roptmy,t: integer;
  xm12,ym12,xc12,yc12,xm21,ym21,xc21,yc21,xm1,ym1,xc1,yc1:integer;
  o,yeni : char;
  gec,d,d1,VCE,IC : String[20];
  samplecount:byte;
  color:word;
  f:string[4];
begin

```



```

clrscr;
writeln('KAÇ AYRIK FREKANS İÇİN TEST ETTİRMEK İSTİYORSUNUZ?');
readln(samplecount);
write('HESAPLANAN DEĞERLER DOSYASININ ADINI GİRİNİZ. '); READLN(d);
write('ÖLÇÜLMÜŞ DEĞERLER DOSYASININ ADINI GİRİNİZ. '); READLN(d1); gotoxy(20,12);
write('Büyültme Katsayısını Giriniz. [10-200]---> '); readln(birim); GrDriver := detect;
initgraph(GrDriver,GrMode,"); {c:\tp\bgi idi.} hata := graphresult;
if hata <> grOK Then
  Begin
    WriteLn('Grafik Hatasi olusmustur. ');
    WriteLn('Graphics error: ',GraphErrorMsg(ErrorCode)); Halt;
  End;
color:=getmaxcolor; setbkcolor(15);setcolor(10); cleardevice;
settextstyle(smallFont,horizDir,1); cizim; dcizim;
assign(hdosya,'c:\tp\hamid'+d+'.dat'); assign(odosya,'c:\tp\hamid'+d1+'.dat');
reset(hdosya); reset(odosya); t:=0;
for n:=0 to samplecount-1 do
  begin
    m:=0;
    readln(hdosya,calc[0,1],calc[0,2],calc[0,3],calc[0,0],calc[2,0],calc[2,1],calc[2,2],calc[2,3],
    calc[2,4],calc[2,5],calc[2,6],calc[2,7],calc[2,8],calc[2,9],calc[2,10],calc[2,11]);
    calc[0,1]:=calc[0,1]*10;
    calc[0,2]:=calc[0,2]*10;
    repeat
      inc(m);
      if m>100 then begin
        writeln('DİKKAT TEST EDİLEN DEĞERİN ÖLÇÜLMÜŞ KARŞILIĞI YOK!!!');
        writeln('HATA HESAPLANAMAZ PROGRAM KIRILDI!!!');exit;
      END;
      readln(odosya,fdess[0,1],fdess[0,2],fdess[0,3],fdess[0,0],fdess[2,0],fdess[2,1],fdess[2,2],
      fdess[2,3],fdess[2,4],fdess[2,5],fdess[2,6],fdess[2,7],fdess[2,8],fdess[2,9],fdess[2,10],fdess[2,11]);
      if not((abs(fdess[0,1]-calc[0,1])<0.00001) and (abs(fdess[0,2]-calc[0,2])<0.00001)) then
        begin
          repeat
            readln(odosya,fdess[0,1],fdess[0,2],fdess[0,3],fdess[0,0],fdess[2,0],fdess[2,1],fdess[2,2],fdess[2,3],
            fdess[2,4],fdess[2,5],fdess[2,6],fdess[2,7],fdess[2,8],fdess[2,9],fdess[2,10],fdess[2,11]);
          until (abs(fdess[0,1]-calc[0,1])<0.00001) and (abs(fdess[0,2]-calc[0,2])<0.00001);
        end;
    end;
  end;

```

```

end;
until (abs(fdess[0,0]-calc[0,0])<0.00001);
k:=0;
for l:=0 to 4 do {3 for s parameters}
begin
x_meas:=fdess[2,k]*cos(fdess[2,k+1]*pi/180);x_calc:=calc[2,k]*cos(calc[2,k+1]*pi/180);
y_meas:=fdess[2,k]*sin(fdess[2,k+1]*pi/180);y_calc:=calc[2,k]*sin(calc[2,k+1]*pi/180);
if n=0 then
begin
CASE K OF
0:BEGIN
xm:=centerk1+round(birim*x_meas); xc:=centerk1+round(birim*x_calc);
ym:=centerm-round(birim*y_meas); yc:=centerm-round(birim*y_calc);
END;
2:BEGIN
xm1:=centerk+round(S21carpan*x_meas); xc1:=centerk+round(S21carpan*x_calc);
ym1:=centerm-round(S21carpan*y_meas); yc1:=centerm-round(S21carpan*y_calc);
END;
4:BEGIN
xm1:=centerk+round(S12carpan*x_meas); xc1:=centerk+round(S12carpan*x_calc);
ym1:=centerm-round(S12carpan*y_meas); yc1:=centerm-round(S12carpan*y_calc);
END;
6:BEGIN
xm:=centerk1+round(birim*x_meas); xc:=centerk1+round(birim*x_calc);
ym:=centerm-round(birim*y_meas); yc:=centerm-round(birim*y_calc);
END;
9:BEGIN
xm:=centerk1+round(birim*x_meas); xc:=centerk1+round(birim*x_calc);
ym:=centerm-round(birim*y_meas); yc:=centerm-round(birim*y_calc);
END;
END;
end;
case k of
0:begin{s11}
setcolor(1); setlinestyle(solidln,0,normwidth);
if n=0 then moveto(xm,ym) else moveto(xm11,ym11);
lineto(centerk1+round(birim*x_meas),centerm-round(birim*y_meas));

```

```

xm11:=getx;ym11:=gety;
if n=4*trunc(n/4) then
  begin
    str(fdess[0,0]:2:1,f);    outtextxy(getx-4,gety-4,'x');
    settextstyle(smallFont,horizDir,1);    outtextxy(getx+5,gety-5,f);
  end;
if N=3 then outtextxy(getx-30,gety+10,'s11');
setcolor(9);    setlinestyle(dottedln,0,normwidth);
if n=0 then moveto(xc,yc) else moveto(xc11,yc11);
lineto(centerk1+round(birim*x_calc),centerm-round(birim*y_calc));
xc11:=getx;yc11:=gety;
if n=4*trunc(n/4) then outtextxy(getx-4,gety-4,'o');
end;
2:begin{s21}
setcolor(1);    setlinestyle(solidln,0,normwidth);
if n=0 then moveto(xm1,ym1) else moveto(xm21,ym21);
lineto(centerk+round(s21carpan*x_meas),centerm-round(s21carpan*y_meas));
xm21:=getx;ym21:=gety;
if n=5*trunc(n/5) then
  begin
    str(fdess[0,0]:2:1,f);    outtextxy(getx-4,gety-4,'x');
    settextstyle(smallFont,horizDir,4);    outtextxy(getx+5,gety-5,f);
  end;
if N=3 then outtextxy(getx-30,gety-20,'s21');
setcolor(9);    setlinestyle(dottedln,0,normwidth);
if n=0 then moveto(xc1,yc1) else moveto(xc21,yc21);
lineto(centerk+round(s21carpan*x_calc),centerm-round(s21carpan*y_calc));
xc21:=getx;yc21:=gety;
if n=5*trunc(n/5) then outtextxy(getx-4,gety-4,'o');
end;
4:begin{s12}
setcolor(1);    setlinestyle(solidln,0,normwidth);
if n=0 then moveto(xm1,ym1) else moveto(xm12,ym12);
lineto(centerk+round(s12carpan*x_meas),centerm-round(s12carpan*y_meas));
xm12:=getx;ym12:=gety;
if n=7*trunc(n/7) then
  begin

```



```

str(fdess[0,0]:2:1,f);      outtextxy(getx-4,gety-4,'x');
settextstyle(smallFont,horizDir,1);      outtextxy(getx+5,gety-5,f);
end;
if N=3 then outtextxy(getx+5,gety-20,'s12');
setcolor(9);      setlinestyle(dottedln,0,normwidth);
if n=0 then moveto(xc1,yc1) else moveto(xc12,yc12);
lineto(centerk+round(s12carpan*x_calc),centerm-round(s12carpan*y_calc));
xc12:=getx;yc12:=gety;      if n=7*trunc(n/7) then outtextxy(getx-4,gety-4,'o');
end;
6:begin{s22}
setcolor(1);      setlinestyle(solidln,0,normwidth);
if n=0 then moveto(xm,ym) else moveto(xm22,ym22);
lineto(centerk1+round(birim*x_meas),centerm-round(birim*y_meas));
xm22:=getx;ym22:=gety;
if n=6*trunc(n/6) then
begin
str(fdess[0,0]:2:1,f);      outtextxy(getx-4,gety-4,'x');
settextstyle(smallFont,horizDir,1);      outtextxy(getx+5,gety-5,f);
end;
if N=3 then outtextxy(getx-10,gety+10,'s22');
setcolor(9);      setlinestyle(dottedln,0,normwidth);
if n=0 then moveto(xc,yc) else moveto(xc22,yc22);
lineto(centerk1+round(birim*x_calc),centerm-round(birim*y_calc));
xc22:=getx;yc22:=gety;
if n=6*trunc(n/6) then outtextxy(getx-4,gety-4,'o');
end;
9:begin{ropt}
setcolor(1);      setlinestyle(solidln,0,normwidth);
if n=0 then moveto(xm,ym) else moveto(roptmx,roptmy);
lineto(centerk1+round(birim*x_meas),centerm-round(birim*y_meas));
roptmx:=getx;roptmy:=gety;
if n=6*trunc(n/6) then
begin
str(fdess[0,0]:2:1,f);      outtextxy(getx-4,gety-4,'x');
settextstyle(smallFont,horizDir,1);      outtextxy(getx+5,gety-5,f);
end;
if N=3 then outtextxy(getx-1,gety+10,'ropt');

```



```

        setcolor(9);          setlinestyle(dottedln,0,normwidth);
        if n=0 then moveto(xc,yc) else moveto(roptcx,roptcy);
        lineto(centerk1+round(birim*x_calc),centerm-round(birim*y_calc));
        roptcx:=getx;roptcy:=gety;
        if n=6*trunc(n/6) then outtextxy(getx-4,gety-4,'o');
    end;
end;
k:=k+2;if k=8 then k:=9;
end;
end;
setcolor(5);  STR(FDESS[0,1]:3:1,VCE);  STR(FDESS[0,2]:3:1,IC);  d:=copy(d,1,7);
outtextxy(10,10,'VCE= '+VCE+'[V] IC='+IC+' [mA] KUTUPLAMA NOKTASINDA');
outtextxy(10,20,d+' TRANSISTORU S-PARAMETRELERI');  close(odosya);  close(hdosya);
Readln;  closegraph;
{ CLRSCR;  window(1,1,45,3);  textcolor(Black);  textbackground(White);  clrscr;}
end;
begin
    sf_cizim;
end.

```

### EK.1.2.2 Sinyal Parametreleri İçin

```

Program Plot_S_BW;
{$N+}
uses crt,dos,graph,printer;
const
    centerK=480; {320}
    centerM=300; {240}
    centerK1=160; {320}
    S21MAX=3.1;
    S12MAX=0.73;
    S12CARPAN=600;
    S21CARPAN=40;
    adim=1;
var
    Color,birim: Word;
    bilgi:string[30];

```





```

    birim1,errcode,l,gd,gm,C,D,GrDriver,GrMode,ilk,aralik,kac1,kac2: integer;
    secim: byte;
PROCEDURE CIZIM;
PROCEDURE DCIZIM;
PROCEDURE S_CIZIM;
const
    outputs_max=10;
    output_nodes=8;
type
    dizi=array[0..outputs_max-1,0..output_nodes] of real;
var
    fdess,calc:dizi;
    x_calc,y_calc,x_meas,y_meas,GP1:real;
    odosya,hdosya:text;
    tt,k,m,errorcode,n,i,j,ilk,hata,xm,ym,xc,yc,xm11,ym11,xc11,yc11,xm22,ym22,xc22,yc22,t:integer;
    xm12,ym12,xc12,yc12,xm21,ym21,xc21,yc21,xm1,ym1,xc1,yc1:integer;
    o,yeni : char;
    gec,d,VCE,IC : String[20];
    samplecount:byte;
    color:word;
    f:string[4];
begin
    clrscr;   writeln('KAÇ AYRIK FREKANS İÇİN TEST ETTİRMEK İSTİYORSUNUZ? ');
    readln(samplecount);   write(' OKUNACAK DOSYANIN ADINI GİRİNİZ. ');READLN(d);
    gotoxy(20,12);   write('Büyültme Katsayısını Giriniz. [10-200]---> ');readln(birim);
    GrDriver := detect;   initgraph(GrDriver,GrMode,""); {c:\tp\bgi idi.}   hata := graphresult;
    if hata <> grOK Then
        Begin
            WriteLn('Grafik Hatasi olusmustur. ');
            WriteLn('Graphics error: ',GraphErrorMsg(ErrorCode));   Halt;
        End;
    color:=getmaxcolor;   setbkcolor(15);setcolor(10);   cleardevice;
    settextstyle(smallFont,horizDir,1);   cizim;   dcizim;   assign(hdosya,'c:\tp\hamid\'+h'+d+'.dat');
    D:=COPY(D,2,7);   assign(odosya,'c:\tp\hamid\'+d+'.dat');   reset(hdosya);   reset(odosya);   t:=0;
    for n:=0 to samplecount-1 do
        begin
            m:=0;   readln(hdosya,calc[0,1],calc[0,2],calc[0,3],calc[0,0],calc[2,0],calc[2,1],calc[2,2],

```



```

calc[2,3],calc[2,4],calc[2,5],calc[2,6],calc[2,7]);
calc[0,1]:=calc[0,1]*10;      calc[0,2]:=calc[0,2]*10;
repeat
  inc(m);
  if m>100 then begin
    writeln("DİKKAT TEST EDİLEN DEĞERİN ÖLÇÜLMÜŞ KARŞILIĞI YOK!!!");
    writeln("HATA HESAPLANAMAZ PROGRAM KIRILDI!!!");exit;
  END;
  readln(odosya,fdess[0,1],fdess[0,2],fdess[0,3],fdess[0,0],fdess[2,0],fdess[2,1],fdess[2,2],
  fdess[2,3],fdess[2,4],fdess[2,5],fdess[2,6],fdess[2,7]);
  if not((abs(fdess[0,1]-calc[0,1])<0.00001) and (abs(fdess[0,2]-calc[0,2])<0.00001)) then
    begin
      repeat
        readln(odosya,fdess[0,1],fdess[0,2],fdess[0,3],fdess[0,0],fdess[2,0],fdess[2,1],fdess[2,2],fdess[2,3],
        fdess[2,4],fdess[2,5],fdess[2,6],fdess[2,7]);
        until (abs(fdess[0,1]-calc[0,1])<0.00001) and (abs(fdess[0,2]-calc[0,2])<0.00001);
      end;
    until (abs(fdess[0,0]-calc[0,0])<0.00001);
    k:=0;
    for l:=0 to 3 do {3 for s parameters}
      begin
        x_meas:=fdess[2,k]*cos(fdess[2,k+1]*pi/180);x_calc:=calc[2,k]*cos(calc[2,k+1]*pi/180);
        y_meas:=fdess[2,k]*sin(fdess[2,k+1]*pi/180);y_calc:=calc[2,k]*sin(calc[2,k+1]*pi/180);
        if n=0 then
          begin
            CASE K OF
              0:BEGIN
                xm:=centerk1+round(birim*x_meas); xc:=centerk1+round(birim*x_calc);
                ym:=centerm-round(birim*y_meas); yc:=centerm-round(birim*y_calc);
              END;
              2:BEGIN
                xm1:=centerk+round(S21carpan*x_meas); xc1:=centerk+round(S21carpan*x_calc);
                ym1:=centerm-round(S21carpan*y_meas); yc1:=centerm-round(S21carpan*y_calc);
              END;
              4:BEGIN
                xm1:=centerk+round(S12carpan*x_meas); xc1:=centerk+round(S12carpan*x_calc);
                ym1:=centerm-round(S12carpan*y_meas); yc1:=centerm-round(S12carpan*y_calc);

```



```

END;
6:BEGIN
  xm:=centerk1+round(birim*x_meas);    xc:=centerk1+round(birim*x_calc);
  ym:=centerm-round(birim*y_meas);    yc:=centerm-round(birim*y_calc);
END;
END;
end;
case k of
0:begin{s11}
  setcolor(1);
  setlinestyle(solidln,0,normwidth); if n=0 then moveto(xm,ym) else moveto(xm11,ym11);
  lineto(centerk1+round(birim*x_meas),centerm-round(birim*y_meas));
  xm11:=getx;ym11:=gety;
  if n=4*trunc(n/4) then
  begin
    str(fdess[0,0]:2:1,f);    outtextxy(getx-4,gety-4,'x');
    settextstyle(smallFont,horizDir,1);    outtextxy(getx+5,gety-5,f);
  end;
  if N=3 then outtextxy(getx-20,gety+10,'s11');
  setcolor(9);    setlinestyle(dottedln,0,normwidth);
  if n=0 then moveto(xc,yc) else moveto(xc11,yc11);
  lineto(centerk1+round(birim*x_calc),centerm-round(birim*y_calc));
  xc11:=getx;yc11:=gety;
  if n=4*trunc(n/4) then outtextxy(getx-4,gety-4,'o');
end;
2:begin{s21}
  setcolor(1);    setlinestyle(solidln,0,normwidth);
  if n=0 then moveto(xm1,ym1) else moveto(xm21,ym21);
  lineto(centerk+round(s21carpan*x_meas),centerm-round(s21carpan*y_meas));
  xm21:=getx;ym21:=gety;
  if n=5*trunc(n/5) then
  begin
    str(fdess[0,0]:2:1,f);    outtextxy(getx-4,gety-4,'x');
    settextstyle(smallFont,horizDir,4);    outtextxy(getx+5,gety-5,f);
  end;
  if N=3 then outtextxy(getx-45,gety-20,'s21');
  setcolor(9);    setlinestyle(dottedln,0,normwidth);

```



```

if n=0 then moveto(xc1,yc1) else moveto(xc21,yc21);
lineto(centerk+round(s21carpan*x_calc),centerm-round(s21carpan*y_calc));
xc21:=getx;yc21:=gety;
if n=5*trunc(n/5) then outtextxy(getx-4,gety-4,'o');
end;
4:begin{s12}
setcolor(1);      setlinestyle(solidln,0,normwidth);
if n=0 then moveto(xm1,ym1) else moveto(xm12,ym12);
lineto(centerk+round(s12carpan*x_meas),centerm-round(s12carpan*y_meas));
xm12:=getx;ym12:=gety;
if n=7*trunc(n/7) then
begin
str(fdess[0,0]:2:1,f);  outtextxy(getx-4,gety-4,'x');
setttextstyle(smallFont,horizDir,1);  outtextxy(getx+5,gety-5,f);
end;
if N=3 then outtextxy(getx+50,gety-40,'s12');
setcolor(9);      setlinestyle(dottedln,0,normwidth);
if n=0 then moveto(xc1,yc1) else moveto(xc12,yc12);
lineto(centerk+round(s12carpan*x_calc),centerm-round(s12carpan*y_calc));
xc12:=getx;yc12:=gety;
if n=7*trunc(n/7) then outtextxy(getx-4,gety-4,'o');
end;
6:begin{s22}
setcolor(1);      setlinestyle(solidln,0,normwidth);
if n=0 then moveto(xm,ym) else moveto(xm22,ym22);
lineto(centerk1+round(birim*x_meas),centerm-round(birim*y_meas));
xm22:=getx;ym22:=gety;
if n=6*trunc(n/6) then
begin
str(fdess[0,0]:2:1,f);  outtextxy(getx-4,gety-4,'x');
setttextstyle(smallFont,horizDir,1);  outtextxy(getx+5,gety-5,f);
end;
if N=3 then outtextxy(getx-100,gety+10,'s22');
setcolor(9);      setlinestyle(dottedln,0,normwidth);
if n=0 then moveto(xc,yc) else moveto(xc22,yc22);
lineto(centerk1+round(birim*x_calc),centerm-round(birim*y_calc));
xc22:=getx;yc22:=gety;

```



```

        if n=6*trunc(n/6) then outtextxy(getx-4,gety-4,'o');
    end;
    end;
    k:=k+2;
    end;
    end;
    setcolor(5);
    STR(FDESS[0,1]:3:1,VCE); STR(FDESS[0,2]:3:1,IC); d:=copy(d,1,7);
    outtextxy(10,10,'VCE= '+VCE+'[V] IC='+IC+' [mA] KUTUPLAMA NOKTASINDA');
    outtextxy(10,20,d+' TRANSISTORU S-PARAMETRELERI'); close(odosya); close(hdosya); Readln;
    closegraph; {CLRSCR; window(1,1,45,3); textcolor(Black); textbackground(White); clrscr;}
end;
begin
    s_cizim;
end.

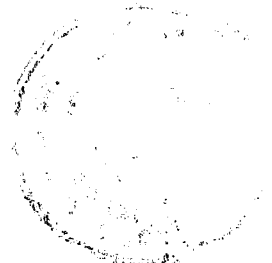
```

### EK.1.2.3 Renkli Baskılı Sinyal Parametreleri İçin

```

Program Plot_S_Coloured;
{$N+}
uses crt,dos,graph,printer;
const
    centerK=480; {320}
    centerM=300; {240}
    centerK1=160; {320}
    S21MAX=15;
    S12MAX=0.3;
    S12CARPAN=700;
    S21CARPAN=15;
var
    Color,birim : Word;
    bilgi:string[30];
    birim1,errcode,l,gd,gm,C,D: integer;
    GrDriver,GrMode,ilk,aralik,kac1,kac2: integer;
    secim: byte;
PROCEDURE RENKLICIZIM;
var

```



```

ilk,son : integer;
arccoords:arccoordstype;
reel,imajinel : real;
begin
reel := 0;
imajinel:= 0.2;
For ilk := 1 To 15 Do begin
circle(centerk1+round(birim*reel/(1+reel)),centerm,round(birim/(1+reel)));
reel := reel+0.2;
end;
circle(centerk1,centerm,birim+4);
arc(centerk1+birim,centerm-round(birim/imajinel),247,270,round(birim/imajinel));
arc(centerk1+birim,centerm+round(birim/imajinel),90,113,round(birim/imajinel));
getarccoords(arccoords);
imajinel := imajinel+0.4;
arc(centerk1+birim,centerm-round(birim/imajinel),207,270,round(birim/imajinel));
arc(centerk1+birim,centerm+round(birim/imajinel),90,153,round(birim/imajinel));
getarccoords(arccoords);
imajinel := imajinel+0.4;
arc(centerk1+birim,centerm-round(birim/imajinel),180,270,round(birim/imajinel));
arc(centerk1+birim,centerm+round(birim/imajinel),90,180,round(birim/imajinel));
getarccoords(arccoords);
imajinel := imajinel+0.6;
arc(centerk1+birim,centerm-round(birim/imajinel),151,270,round(birim/imajinel));
arc(centerk1+birim,centerm+round(birim/imajinel),90,209,round(birim/imajinel));
getarccoords(arccoords);
imajinel := imajinel+0.8;
arc(centerk1+birim,centerm-round(birim/imajinel),130,270,round(birim/imajinel));
arc(centerk1+birim,centerm+round(birim/imajinel),90,230,round(birim/imajinel));
getarccoords(arccoords);
imajinel := imajinel+1.2;
arc(centerk1+birim,centerm-round(birim/imajinel),116,270,round(birim/imajinel));
arc(centerk1+birim,centerm+round(birim/imajinel),90,244,round(birim/imajinel));
getarccoords(arccoords);
imajinel := imajinel+2;
arc(centerk1+birim,centerm-round(birim/imajinel),105,270,round(birim/imajinel));
arc(centerk1+birim,centerm+round(birim/imajinel),90,255,round(birim/imajinel));

```



```

getarccoords(arccoords);
Line(160-birim,centerm,160+birim,centerm);
setcolor(1);
outtextxy(400,10,'BLUE _____ S11(MEASURED)');
setcolor(9);
outtextxy(400,20,'LIGHT BLUE __ S11(CALCULATED)');
setcolor(2);
outtextxy(400,30,'GREEN _____ S21(MEASURED)');
setcolor(10);
outtextxy(400,40,'LIGHT GREEN __ S21(CALCULATED)');
setcolor(3);
outtextxy(400,50,'CYAN _____ S12(MEASURED)');
setcolor(11);
outtextxy(400,60,'LIGHT CYAN __ S12(CALCULATED)');
setcolor(4);
outtextxy(400,70,'RED _____ S22(MEASURED)');
setcolor(12);
outtextxy(400,80,'LIGHT RED __ S22(CALCULATED)');
end;
PROCEDURE DCIZIM;
PROCEDURE S_RENKLICIZIM;
const
  outputs_max=10;
  output_nodes=8;
type
  dizi=array[0..outputs_max-1,0..output_nodes] of real;
var
  fdess,calc:dizi;
  x_calc,y_calc,x_meas,y_meas, GP1 :real;
  odosya,hdosya:text;
  tt,k,m,errorcode,n,i,j,ilk,hata,xm,ym,xc,yc,xm11,ym11,xc11,yc11,xm22,ym22,xc22,yc22,t:integer;
  xm12,ym12,xc12,yc12,xm21,ym21,xc21,yc21,xm1,ym1,xc1,yc1:integer;
  o,yeni : char;
  gec,d,VCE,IC : String[20];
  samplecount:byte;
  color:word;
  f:string[4];

```



```

begin
  clrscr;
  writeln("KAÇ AYRIK FREKANS İÇİN TEST ETTİRMEK İSTİYORSUNUZ?
');readln(samplecount);
  write(' OKUNACAK DOSYANIN ADINI GİRİNİZ. ');READLN(d);
  gotoxy(20,12); write("Büyültme Katsayısını Giriniz. [10-200]----> ');readln(birim);
  GrDriver := detect;  initgraph(GrDriver,GrMode,""); {c:\tp\bgi idi.} hata := graphresult;
  if hata <> grOK Then
    Begin
      WriteLn('Grafik Hatasi olustur. '); WriteLn('Graphics error: ',GraphErrorMsg(ErrorCode));
      Halt;
    End;
  color:=getmaxcolor; setbkcolor(15);setcolor(14); cleardevice; settextstyle(smallFont,horizDir,4);
  renklicizim; dcizim; assign(hdosya,'c:\tp\hamid\'+h'+d+'.dat');assign(odosya,'c:\tp\hamid\'+d+'.dat');
  reset(hdosya); reset(odosya); t:=0;
  for n:=0 to samplecount-1 do
    begin
      m:=0;
      readln(hdosya,calc[0,1],calc[0,2],calc[0,0],calc[2,0],calc[2,1],calc[2,2],calc[2,3],calc[2,4],
      calc[2,5],calc[2,6],calc[2,7]);
      calc[0,1]:=calc[0,1]*10; calc[0,2]:=calc[0,2]*10;
      repeat
        inc(m);
        if m>100 then begin
          writeln("DİKKAT TEST EDİLEN DEĞERİN ÖLÇÜLMÜŞ KARŞILIĞI YOK!!!");
          writeln("HATA HESAPLANAMAZ PROGRAM KIRILDI!!!");exit;
        END;
      readln(odosya,fdess[0,1],fdess[0,2],fdess[0,0],fdess[2,0],fdess[2,1],fdess[2,2],fdess[2,3],fdess[2,4],
      fdess[2,5],fdess[2,6],fdess[2,7]);
      if not((abs(fdess[0,1]-calc[0,1])<0.00001) and (abs(fdess[0,2]-calc[0,2])<0.00001)) then
        begin
          repeat
            readln(odosya,fdess[0,1],fdess[0,2],fdess[0,0],fdess[2,0],fdess[2,1],fdess[2,2],fdess[2,3],
            fdess[2,4],fdess[2,5],fdess[2,6],fdess[2,7]);
            until (abs(fdess[0,1]-calc[0,1])<0.00001) and (abs(fdess[0,2]-calc[0,2])<0.00001);
          end;
        until (abs(fdess[0,0]-calc[0,0])<0.00001);
    end;
  until (abs(fdess[0,0]-calc[0,0])<0.00001);

```





```

k:=0;
for l:=0 to 3 do {3 for s parameters}
  begin
    x_meas:=fdess[2,k]*cos(fdess[2,k+1]*pi/180);x_calc:=calc[2,k]*cos(calc[2,k+1]*pi/180);
    y_meas:=fdess[2,k]*sin(fdess[2,k+1]*pi/180);y_calc:=calc[2,k]*sin(calc[2,k+1]*pi/180);
    if n=0 then
      begin
        CASE K OF
          0:BEGIN
            xm:=centerk1+round(birim*x_meas);   xc:=centerk1+round(birim*x_calc);
            ym:=centerm-round(birim*y_meas);   yc:=centerm-round(birim*y_calc);
          END;
          2:BEGIN
            xm1:=centerk+round(S21carpan*x_meas);   xc1:=centerk+round(S21carpan*x_calc);
            ym1:=centerm-round(S21carpan*y_meas);   yc1:=centerm-round(S21carpan*y_calc);
          END;
          4:BEGIN
            xm1:=centerk+round(S12carpan*x_meas);   xc1:=centerk+round(S12carpan*x_calc);
            ym1:=centerm-round(S12carpan*y_meas);   yc1:=centerm-round(S12carpan*y_calc);
          END;
          6:BEGIN
            xm:=centerk1+round(birim*x_meas);   xc:=centerk1+round(birim*x_calc);
            ym:=centerm-round(birim*y_meas);   yc:=centerm-round(birim*y_calc);
          END;
        END;
      end;
    case k of
      0:begin{s11}
        setcolor(1);      setlinestyle(solidln,0,normwidth);
        if n=0 then moveto(xm,ym) else moveto(xm11,ym11);
        lineto(centerk1+round(birim*x_meas),centerm-round(birim*y_meas));
        xm11:=getx;ym11:=gety;
        if n=4*trunc(n/4) then
          begin
            str(fdess[0,0]:2:1,f);   outtextxy(getx-4,gety-4,'x');
            settxtstyle(smallFont,horizDir,1);   outtextxy(getx+5,gety-5,f);
          end;
        end;
      end;
    end;
  end;
end;

```



```

end;
if N=3 then outtextxy(getx-10,gety+10,'s11'); setcolor(9);
setlinestyle(dottedln,0,normwidth); if n=0 then moveto(xc,yc) else moveto(xc11,yc11);
lineto(centerk1+round(birim*x_calc),centerm-round(birim*y_calc));
xc11:=getx;yc11:=gety;
if n=4*trunc(n/4) then outtextxy(getx-4,gety-4,'o');
end;
2:begin{s21}
setcolor(2); setlinestyle(solidln,0,normwidth);
if n=0 then moveto(xm1,ym1) else moveto(xm21,ym21);
lineto(centerk+round(s21carpan*x_meas),centerm-round(s21carpan*y_meas));
xm21:=getx;ym21:=gety;
if n=4*trunc(n/4) then
begin
str(fdess[0,0]:2:1,f); outtextxy(getx-4,gety-4,'x');
settextstyle(smallFont,horizDir,4); outtextxy(getx+5,gety-5,f);
end;
if N=3 then outtextxy(getx-10,gety+10,'s21');
setcolor(10); setlinestyle(dottedln,0,normwidth);
if n=0 then moveto(xc1,yc1) else moveto(xc21,yc21);
lineto(centerk+round(s21carpan*x_calc),centerm-round(s21carpan*y_calc));
xc21:=getx;yc21:=gety;
if n=4*trunc(n/4) then outtextxy(getx-4,gety-4,'o');
end;
4:begin{s12}
setcolor(3); setlinestyle(solidln,0,normwidth);
if n=0 then moveto(xm1,ym1) else moveto(xm12,ym12);
lineto(centerk+round(s12carpan*x_meas),centerm-round(s12carpan*y_meas));
xm12:=getx;ym12:=gety;
if n=6*trunc(n/6) then
begin
str(fdess[0,0]:2:1,f); outtextxy(getx-4,gety-4,'x');
settextstyle(smallFont,horizDir,1); outtextxy(getx+5,gety-5,f);
end;
if N=3 then outtextxy(getx+20,gety+10,'s12');
setcolor(11); setlinestyle(dottedln,0,normwidth);
if n=0 then moveto(xc1,yc1) else moveto(xc12,yc12);

```



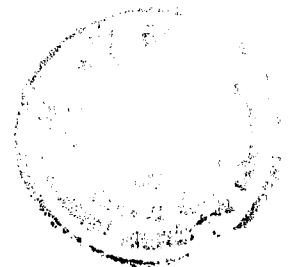
```

lineto(centerk+round(s12carpan*x_calc),centerm-round(s12carpan*y_calc));
xc12:=getx;yc12:=gety;
if n=6*trunc(n/6) then outtextxy(getx-4,gety-4,'o');
end;
6:begin{s22}
setcolor(4);      setlinestyle(solidln,0,normwidth);
if n=0 then moveto(xm,ym) else moveto(xm22,ym22);
lineto(centerk1+round(birim*x_meas),centerm-round(birim*y_meas));
xm22:=getx;ym22:=gety;
if n=5*trunc(n/5) then
begin
str(fdess[0,0]:2:1,f);      outtextxy(getx-4,gety-4,'x');
settextstyle(smallFont,horizDir,1);      outtextxy(getx+5,gety-5,f);
end;
if N=3 then outtextxy(getx-10,gety+10,'s22');
setcolor(12);      setlinestyle(dottedln,0,normwidth);
if n=0 then moveto(xc,yc) else moveto(xc22,yc22);
lineto(centerk1+round(birim*x_calc),centerm-round(birim*y_calc));
xc22:=getx;yc22:=gety;
if n=5*trunc(n/5) then outtextxy(getx-4,gety-4,'o');
end;
end;
k:=k+2;
end;
end;
setcolor(5);  STR(FDESS[0,1]:3:1,VCE);  STR(FDESS[0,2]:3:1,IC);  d:=copy(d,2,7);
outtextxy(10,10,'VCE= '+VCE+'[V] IC='+IC+' [mA] KUTUPLAMA NOKTASINDA');
outtextxy(10,20,d+' TRANSISTORUNA AIT S-PARAMETRELERİ'); close(odosya);close(hdosya);
Readln; closegraph; {CLRSCR;window(1,1,45,3);textcolor(Black);textbackground(White); clrscr;}
end;
begin
s_renklicizim;
end.

```

### EK 1.3 Optimum Alpha Değerini Bulan Program

PROGRAM OPTIMUM\_ALPHA\_BULUR



```

{$N+}
uses crt,printer;
CONST
  Layers_Max   = 3;   { Maximum number of layers }
  Nodes_Max    = 12;  { Maximum number of node in any layer }
  Outputs_Max  = 14;  { Number of input combinations }
  Input_Nodes  = 4;   { Number of input layer nodes }
  Output_Nodes = 12;  { Number of output layer nodes }
  Hidden_Nodes = 12;  { Number of hidden layer nodes }
  Epsilon      = 0.5;
  Iterative_Max = 10; { Maximum iteration number }
  Every        = 10;  { After every iteration print results }
  S22max       = 1;
  s21max       = 36;  { Normalization coefficient40}
  s12max       = 4.9; { Normalization coefficient3.7}
  fminmax      = 4.5;
  altsinir     = 0.0;
  ustsinir     = 0.99;
  adim         = 0.01;
TYPE
  adizi = array [ 0..outputs_Max-1 ] of real;
VAR
  f:adizi;
  j,son,count,i:longint;
  samplecount,ii,tt,isylla, num,dcount, Layers, Inputs, Outputs:byte;
  step,eps, alp,ust,optalpha, err, error, terr, terror, total_rate, beta,des: REAL;
  Node   : array [0..Layers_Max-1] of integer;
  matrix : array [0..Outputs_Max-1, 0..(Input_Nodes+Output_Nodes-1)] of REAL;
  out    : array [0..Layers_Max-1, 0..Nodes_Max-1] of REAL;
  der    : array [0..Layers_Max-1, 0..Nodes_Max-1] of REAL;
  theta  : array [0..Layers_Max-1, 0..Nodes_Max-1] of REAL;
  dtheta : array [0..Layers_Max-1, 0..Nodes_Max-1] of REAL;
  weight : array [0..Layers_Max-1, 0..Nodes_Max-1, 0..Nodes_Max-1] of REAL;
  dweight : array [0..Layers_Max-1, 0..Nodes_Max-1, 0..Nodes_Max-1] of REAL;
  dess   : array [0..Outputs_Max-1,0..output_nodes-1] of REAL;
  optalphas,errors,epsilons,st1: string[7]; st5,st6: string[6];

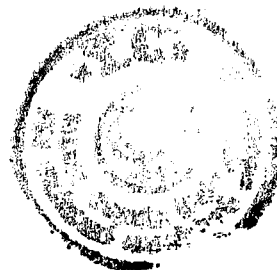
```

```

d,st2, st3, st4: string[12];
idatafile,datafile,graphfile,outfile,dosya,weightfile :text;
tercih,pass:char;
ad,hiz:string[8];
PROCEDURE ARRANGE;
var
  dosya,ydosya:text;
  A:string[1];
  B:string[10];
  q:string[80];
begin
  write(' OKUNACAK DOSYANIN ADINI GİRİNİZ. ');READLN(d);
  assign(dosya,'c:\appcad\spara\ncspar\''+d+'.s2p');
  assign(ydosya,d+'.dat');
  reset(dosya);
  repeat
    readln(dosya,q);
    a:=copy(q,1,1);
    b:=copy(q,2,11);
  until (a='Z') or (a='z');
  rewrite(ydosya);
  while not eof(dosya) do
  begin
    readln(dosya,q);
    a:=copy(q,1,1);
    if (a='Z') or (a='z') then
      begin
        b:=copy(q,2,11);
        readln(dosya,q);
      end;
    writeln(ydosya,b,' ',q);
  end;
  close(ydosya);
  close(dosya);
end; {Procedure arrange}

PROCEDURE HEDEF_BELLE;

```



```

type
  dizi=array[0..outputs_max-1,0..output_nodes] of real;
var
  tampon:text;
  yanit:char;
  fdess:dizi;
  x,y:real;
  l,m,n:byte;
  k:byte;
  ST:string[8];
begin
  assign(tampon,'c:\tp\hamid\n2135tmp.dat'); {assign(tampon,'tampon.dat');}
  write(' EN SON KULLANILAN FREKANS ÖRNEKLERİNİ KULLANMAK İSTER MİSİNİZ?
[E/H]');readln(yanit);
  yanit:=upcase(yanit);
  if yanit='E' then
  begin
    reset(tampon);
    for l:=0 to outputs_max-1 do
      begin
        READ(tampon,F[l]);
      end;
  end
  else
  begin
    rewrite(tampon);
    writeln(' HANGİ FREKANSLARDA ÖRNEK ALACAKSANIZ ');
    writeln(' SIRAYLA [GHz] OLARAK GİRİNİZ ');
    for l:=0 to outputs_max-1 do
      begin
        write(' ');write (l+1);write(' ÖRNEK = ');READ(F[l]);write(tampon,f[l]);write(' GHz de');
        writeln;
      end;
    writeln(' ALINACAKTIR. ');
  end;
  close(tampon);
  assign(datafile,d+'.dat');

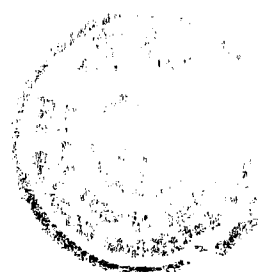
```



```

reset(datafile);m:=0;
assign(idatafile,'i'+d+'.dat');
rewrite(idatafile);
while not eof(datafile) do
begin
  for n:=0 to input_nodes-1 do
    read(datafile,x);
  for l:=m to outputs_max-1 do
  begin
    while not (x=f[l]) do
      begin
        while not eoln(datafile) do
          read(datafile,x);
        for n:=0 to input_nodes-1 do
          read(datafile,x);
        end;
      for k:=0 to output_nodes do
      begin
        fdess[m,k]:=x;      read(datafile,x);
      end;
      for k:=0 to output_nodes do
      begin
        y:=fdess[m,k];      str(Y:6:3,ST);      write(idatafile,' ',st);
      end;
      writeln(idatafile,' ');      m:=m+1;
      for n:=0 to input_nodes-2 do
        read(datafile,x);
      end;
    end;
  close(datafile);      close(idatafile);
  for m:=0 to outputs_max-1 do
  for k:=0 to output_nodes-1 do
  dесс[m,k]:=fdess[m,k+1];
  clrscr;
  for k:=0 to outputs_max-1 do -
  for l:= 0 to output_nodes-1 do
  begin

```



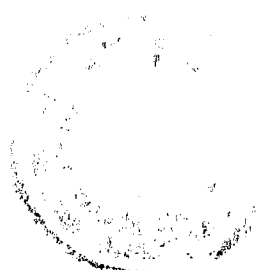
```

case (l) of
1,3,5,7,10:begin
    if dесс[k,l] < 0 then
    begin
        десс[k,l]:=десс[k,l]+360;   десс[k,l]:=десс[k,l]/360;
    end
    else
    begin
        десс[k,l]:=десс[k,l]/360;
    end;
end;
4:десс[k,l]:=десс[k,l]*s12max;
6:десс[k,l]:=десс[k,l]*s22max;
2:десс[k,l]:=десс[k,l]/s21max;
8:десс[k,l]:=десс[k,l]/fminmax;

end;
end;
end; {Procedure hedef_belle}
PROCEDURE WRAND;
PROCEDURE INITIAL;
PROCEDURE YREAD ( isylla: integer );
{ read input pattern ( for several different input combination ) }
begin
case isylla of
0: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=0.5;END;
1: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=1 ;END;
2: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=1.5;END;
3: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=2 ;END;
4: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=2.5;END;
5: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=3 ;END;
6: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=3.5;END;
7: BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=0.5;END;
8: BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=1 ;END;
9: BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=1.5;END;
10:BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=2 ;END;
11:BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=2.5;END;
12:BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=3 ;END;

```

TC YÜREKİNGİR  
 DOKÜMANIŞYON





```

13:BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=3.5;END;
end;
end; { procedure YREAD }
PROCEDURE AYREAD(ISYLLA:BYTE);
FUNCTION SIGMOID ( x: extended ): EXTENDED;
PROCEDURE WFORWARD ;
FUNCTION BACK: REAL;
PROCEDURE LEARNING (alp, eps: REAL );
BEGIN { main }
  clrscr; arrange; hedef_belle;
  { print first header }
  node[2]:= Output_Nodes;
  Outputs:= Output_Nodes; { Çıkış Katmanı Düğüm Sayısı }
  node[1]:= Hidden_Nodes; { Gizli Katman Düğüm Sayısı }
  node[0]:= Input_Nodes;
  Inputs:= Input_Nodes; { Giriş Katmanı Düğüm Sayısı }
  Layers:= Layers_Max; { Katman Sayısı }
  num:= Outputs_Max; { Örnek Sayısı }
  alp:=altsinir; ust:=ustsinir; eps:=epsilon; step:=adim;
  { *****
  INITIALIZE MATRIX
  ***** }
  for i:= 0 to num-1 do
  for j:= 0 to ( inputs+Outputs-1 ) do
  matrix[i,j]:= 0;
  for i:=1 to 24 do
  writeln;
  writeln('Eğitimle İlgili Bilgileri Direkt Dosyadan Okutmak İsterseniz [D] ye');
  writeln('Program İçinde Oluşturduğunuz Alt Programdan Okutmak İsterseniz [A] ya basınız. ');
  readln(pass);pass:=upcase(pass);if pass='D' then hiz:='slow' else hiz:='fast'; pass:='H'; terror:=1;
  son:=trunc((-1*abs(alp)+abs(ust))/step);
  for i:=0 to son do
  begin
  wrand; j:=0;dcount:=trunc(iterative_max/every)+1;
  { *****
  LEARNING PHASE
  ***** }

```



```

STR(EPSILON:4:2,ST5);STR(alp:4:3,ST6);
writeln ('*****');
WRITELN (' EPSİLON=',st5,' ', 'MIKRODALGA TRANSİSTORUNUN YNA İLE
SİMÜLASYONU.....' );
WRITELN (' ALPHA =',st6,' ' , { HAMİD TORPİ AĞUSTOS\ 1994});
WRITELN ('*****');
while ( j < Iterative_Max ) do
begin
  for count:= 0 to Every-1 do
  begin
    error:= 0;
    for isylla:=0 to num-1 do
    begin
      initial;          if hiz='slow' then ayread( isylla ) else yread(isylla);
      wforward;          err:= back;          learning( alp, eps );          error:= error+err;
    end;
    error:= error / outputs;
    if error < terror then
    begin
      terror:=error;          optalpha:=alp;
    end;
    eps:= 0.5 * sqrt( error );
    Inc( j );
  end;
end;
clrscr;  alp:=alp+step;
end;
assign(dosya,'optalpha.dat');
append(dosya); str(eps:4:2,epsilons); str(optalpha:4:2,optalphas); str(terror:6:5,terrors);
writeln(dosya,'İterasyon Sayısı=',iterative_max,' Epsilon=',epsilons,' Alpha=',optalphas,'
terror=',terrors);
close(dosya);
end.

```

Not: Yukarıda verilen programla optimum alpha bulmak için farklı transistor çeşitleri için aşağıdaki değişiklikler yapılması gereklidir. Her bir transistor için hedef uzayının değişmesi nedeniyle bu aşağıda bir kaç transistor için verilen parametrelerin değişmesi gerektiği aşıkardır.



### 1.3.1 Aktif Mikrodalga Elemanı Değiştikçe Programda Değişmesi Gereken Parametrelerin Bir kaç Örnek Üzerinde Gösterilmesi

#### NE02135 için

CONST

```

Layers_Max = 3; { Maximum number of layers }
Nodes_Max = 12; { Maximum number of node in any layer }
Outputs_Max = 14; { Number of input combinations }
Input_Nodes = 4; { Number of input layer nodes }
Output_Nodes = 12; { Number of output layer nodes }
Hidden_Nodes = 12; { Number of hidden layer nodes }
Epsilon = 0.5;
Iterative_Max = 10; { Maximum iteration number }
Every = 10; { After every iteration print results }
S22max = 1;
s21max = 36; { Normalization coefficient40}
s12max = 4.9; { Normalization coefficient3.7}
fminmax = 4.5;
altsinir = 0.0;
ustsinir = 0.99;
adim = 0.01;

```

PROCEDURE YREAD ( isylla: integer );

{ read input pattern ( for several different input combination ) }

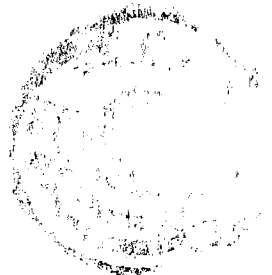
begin

case isylla of

```

0: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=0.5;END;
1: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=1 ;END;
2: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=1.5;END;
3: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=2 ;END;
4: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=2.5;END;
5: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=3 ;END;
6: BEGIN out[0,0]:=1;out[0,1]:=0.5;out[0,2]:=0.1;out[0,3]:=3.5;END;
7: BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=0.5;END;
8: BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=1 ;END;

```



```

9: BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=1.5;END;
10:BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=2 ;END;
11:BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=2.5;END;
12:BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=3 ;END;
13:BEGIN out[0,0]:=1;out[0,1]:=2 ;out[0,2]:=0.1;out[0,3]:=3.5;END;
end;
end; { procedure YREAD }

```

### NE219 için

CONST

```

Layers_Max   = 3;   { Maximum number of layers }
Nodes_Max    = 8;   { Maximum number of node in any layer }
Outputs_Max  = 36;  { Number of input combinations }
Input_Nodes  = 4;   { Number of input layer nodes }
Output_Nodes = 8;   { Number of output layer nodes }
Hidden_Nodes = 8;   { Number of hidden layer nodes }
Epsilon      = 0.5;
Iterative_Max = 100; { Maximum iteration number }
Every        = 100; { After every iteration print results }
S22max       = 1.34;
s21max       = 3.2;  { Normalization coefficient40}
s12max       = 1.35; { Normalization coefficient3.7}
altsinir     = 0.0;
ustsinir     = 0.9;
adim         = 0.1;

```

PROCEDURE YREAD ( isylla: integer );

{ read input pattern ( for several different input combination ) }

begin

case isylla of

```

0: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=2; END;
1: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=2.5;END;
2: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=3; END;
3: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=3.5;END;
4: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=4; END;
5: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=4.5;END;
6: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=5; END;

```

```

7: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=5.5;END;
8: BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.2;out[0,3]:=6; END;
9: BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=2; END;
10:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=2.5;END;
11:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=3; END;
12:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=3.5;END;
13:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=4; END;
14:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=4.5;END;
15:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=5; END;
16:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=5.5;END;
17:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.2;out[0,3]:=6; END;
18:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=2; END;
19:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=2.5;END;
20:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=3; END;
21:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=3.5;END;
22:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=4; END;
23:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=4.5;END;
24:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=5; END;
25:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=5.5;END;
26:BEGIN out[0,0]:=0.8;out[0,1]:=0.1;out[0,2]:=0.8;out[0,3]:=6; END;
27:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=2; END;
28:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=2.5;END;
29:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=3; END;
30:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=3.5;END;
31:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=4; END;
32:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=4.5;END;
33:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=5; END;
34:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=5.5;END;
35:BEGIN out[0,0]:=0.8;out[0,1]:=3; out[0,2]:=0.8;out[0,3]:=6; END;
end;
end; { procedure YREAD }

```

### BFR28 için

#### CONST

```

Layers_Max = 3; { Maximum number of layers }
Nodes_Max = 8; { Maximum number of node in any layer }

```



```

Outputs_Max = 30; { Number of input combinations }
Input_Nodes = 3; { Number of input layer nodes }
Output_Nodes = 8; { Number of output layer nodes }
Hidden_Nodes = 8; { Number of hidden layer nodes }
Epsilon = 0.5;
Iterative_Max = 2; { Maximum iteration number }
Every = 1; { After every iteration print results }
s21max = 15; { Normalization coefficient40}
s12max = 3.9; { Normalization coefficient3.7}
altsinir = 0.0;
ustsinir = 0.99;
adim = 0.01;
PROCEDURE YREAD ( isylla: integer );
{ read input pattern ( for several different input combination ) }
begin
  case isylla of
    0:BEGIN out[0,0]:=0.1;out[0,1]:=0.05;out[0,2]:=0.1;END;
    1:BEGIN out[0,0]:=0.1;out[0,1]:=0.05;out[0,2]:=0.2;END;
    2:BEGIN out[0,0]:=0.1;out[0,1]:=0.05;out[0,2]:=0.4;END;
    3:BEGIN out[0,0]:=0.1;out[0,1]:=0.05;out[0,2]:=0.6;END;
    4:BEGIN out[0,0]:=0.1;out[0,1]:=0.05;out[0,2]:=0.8;END;
    5:BEGIN out[0,0]:=0.1;out[0,1]:=0.05;out[0,2]:=1.0;END;
    6:BEGIN out[0,0]:=0.1;out[0,1]:=0.05;out[0,2]:=1.4;END;
    7:BEGIN out[0,0]:=0.1;out[0,1]:=0.05;out[0,2]:=1.7;END;
    8:BEGIN out[0,0]:=0.1;out[0,1]:=0.05;out[0,2]:=2.0;END;
    9:BEGIN out[0,0]:=0.1;out[0,1]:=0.05;out[0,2]:=3.0;END;
    10:BEGIN out[0,0]:=0.1;out[0,1]:=0.1;out[0,2]:=0.1;END;
    11:BEGIN out[0,0]:=0.1;out[0,1]:=0.1;out[0,2]:=0.2;END;
    12:BEGIN out[0,0]:=0.1;out[0,1]:=0.1;out[0,2]:=0.4;END;
    13:BEGIN out[0,0]:=0.1;out[0,1]:=0.1;out[0,2]:=0.6;END;
    14:BEGIN out[0,0]:=0.1;out[0,1]:=0.1;out[0,2]:=0.8;END;
    15:BEGIN out[0,0]:=0.1;out[0,1]:=0.1;out[0,2]:=1.0;END;
    16:BEGIN out[0,0]:=0.1;out[0,1]:=0.1;out[0,2]:=1.4;END;
    17:BEGIN out[0,0]:=0.1;out[0,1]:=0.1;out[0,2]:=1.7;END;
    18:BEGIN out[0,0]:=0.1;out[0,1]:=0.1;out[0,2]:=2.0;END;
    19:BEGIN out[0,0]:=0.1;out[0,1]:=0.1;out[0,2]:=3.0;END;
    20:BEGIN out[0,0]:=0.5;out[0,1]:=0.3;out[0,2]:=0.1;END;

```



```

21:BEGIN out[0,0]:=0.5;out[0,1]:=0.3;out[0,2]:=0.2;END;
22:BEGIN out[0,0]:=0.5;out[0,1]:=0.3;out[0,2]:=0.4;END;
23:BEGIN out[0,0]:=0.5;out[0,1]:=0.3;out[0,2]:=0.6;END;
24:BEGIN out[0,0]:=0.5;out[0,1]:=0.3;out[0,2]:=0.8;END;
25:BEGIN out[0,0]:=0.5;out[0,1]:=0.3;out[0,2]:=1.0;END;
26:BEGIN out[0,0]:=0.5;out[0,1]:=0.3;out[0,2]:=1.4;END;
27:BEGIN out[0,0]:=0.5;out[0,1]:=0.3;out[0,2]:=1.7;END;
28:BEGIN out[0,0]:=0.5;out[0,1]:=0.3;out[0,2]:=2.0;END;
29:BEGIN out[0,0]:=0.5;out[0,1]:=0.3;out[0,2]:=3.0;END;
end;
end; { procedure YREAD }

```

### **BFO74 için**

CONST

```

Layers_Max = 3; { Maximum number of layers }
Nodes_Max = 8; { Maximum number of node in any layer }
Outputs_Max = 36; { Number of input combinations }
Input_Nodes = 3; { Number of input layer nodes }
Output_Nodes = 8; { Number of output layer nodes }
Hidden_Nodes = 8; { Number of hidden layer nodes }
Epsilon = 0.5;
Iterative_Max = 10000; { Maximum iteration number }
Every = 5000; { After every iteration print results }
s21max = 35; { Normalization coefficient40}
s12max = 4; { Normalization coefficient3.7}
altsinir = 0.8;
ustsinir = 0.98;
adim = 0.01;

```

PROCEDURE YREAD ( isylla: integer );

{ read input pattern ( for several different input combination ) }

begin

case isylla of

```

0:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=0.1;END;
1:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=0.3;END;
2:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=0.6;END;
3:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=1.0;END;

```



```
4:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=1.2;END;
5:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=1.5;END;
6:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=1.8;END;
7:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=2.0;END;
8:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=3.0;END;
9:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=4.1;END;
10:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=4.5;END;
11:BEGIN out[0,0]:=1.0;out[0,1]:=0.2;out[0,2]:=5.0;END;
12:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=0.1;END;
13:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=0.3;END;
14:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=0.6;END;
15:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=1.0;END;
16:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=1.2;END;
17:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=1.5;END;
18:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=1.8;END;
19:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=2.0;END;
20:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=3.0;END;
21:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=4.0;END;
22:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=4.5;END;
23:BEGIN out[0,0]:=1.0;out[0,1]:=1;out[0,2]:=5.0;END;
24:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=0.1;END;
25:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=0.3;END;
26:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=0.6;END;
27:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=1.0;END;
28:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=1.2;END;
29:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=1.5;END;
30:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=1.8;END;
31:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=2.0;END;
32:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=3.0;END;
33:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=4.0;END;
34:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=4.5;END;
35:BEGIN out[0,0]:=1.0;out[0,1]:=2;out[0,2]:=5.0;END;
end;
end; { procedure YREAD }
```





## EK 2 YAZARIN DOKTORA TEZİ KONUSU İLE İLGİLİ YAYINLARI

### EK 2.1 Uluslararası Makaleler

1-GÜNEŞ, F., GÜRGEN, F., TORPİ, H.:"Signal-Noise neural network model for active microwave devices",IEE Proc.-Circuit Devices Syst., 1996, 143,(1), pp.1-8.

2-GÜNEŞ,F.,TORPİ,H.,GÜRGEN,F.,"Multidimensional Signal-Noise Neural Network Model for Microwave Transistors",submitted to IEE Proc.-Circuit Devices Systems.

### EK 2.2 Uluslararası Bildiriler

1-TORPİ,H.,GÜNEŞ,F.,GÜRGEN,F.,"Neural Network Simulators for Active Microwave Devices ", Fourth International Conference on Artificial Neural Networks,26-28 June 1995,Cambridge,England.

2-GÜNEŞ,F.,GÜRGEN,F.,TORPİ,H.," Neural Network Simulation of the Signal and Noise Parameters", Progress in Electromagnetic Research Symposium (PIERS),24-28 July 1995,Seattle Washington USA.

3-GÜNEŞ,F.,GÜRGEN,F.,TORPİ,H.,"Unified Small-Signal-Noise Neural Network for Active Microwave Devices",ECCTD'95 European Conference on Circuit Theory and Design, 27-31 August 1995, İstanbul-TURKEY.

4-GÜNEŞ,F.,TORPİ,H.,GÜRGEN,F.,"Multi-Bias/Configuration Neural Network Models for Microwave Active Devices", International Conference on Telecommunications, ICT'96, İstanbul-Türkiye, 13-17 April 1996.

5-GÜNEŞ,F.,TORPİ,H.,GÜRGEN,F.,"Efficient Model Parameter Extraction Using Neural Networks for Active Microwave Devices", Progress in Electromagnetic Research Symposium PIERS'96, 8-12 July 1996, Innsbruck, Austria.

6-TORPİ,H.,GÜNEŞ,F.,GÜRGEN,F.," Neural Network Approach for the Characterisation of the Active Microwave Devices", I.Symposium on Mathematical and Computational Applications,Vol. 1,No:1,19-21 November 1996,Manisa Turkey,pp.113-118.

7-GÜNEŞ,F.,TORPİ,H.,ÇETİNER,B.A.,"Neural Network Approach For The Active Device Characterization",accepted to submit in ECCTD'97 European Conference on Circuit Theory and Design, 1997, Budapeşte-HUNGARY.

### EK 2.3 Ulusal Bildiriler

1-TORPİ,H.,GÜNEŞ,F.,GÜRGEN,F.," Sinyal ve Gürültü Parametrelerinin Yapay Nöron Ağı Simülasyonu" SİU'95, Sinyal İşleme ve Uygulamaları Konferansı 26-28 April 1995, Nevşehir, Türkiye, pp.214-219.



2-TORPİ,H.,GÜNEŞ,F.,GÜRGEN,F., " Aktif Mikrodalga Elemanlarının Yapay Sinir Ağı Simulatörleri  
",Elektrik Mühendisliği 6.Ulusal Kongresi , 11-17 Sept. 1995, Bursa, TÜRKİYE, pp.676-679.

Bundan sonraki sayfalarda burda belirtilen sırada doktora tezinden çıkan bu yayınlarımızı sıraladık.



# Signal-noise neural network model for active microwave devices

F. Güneş  
F. Gürgen  
H. Torpi

*Indexing terms:* Neural network, Microwave devices

**Abstract:** A new method for concurrently modelling the small-signal and the noise performance of active microwave devices is proposed. Here, the device is modelled by a black box whose small signal and noise parameters are evaluated through a neural network, based upon the fitting of both of these parameters over the operational bandwidth of the device. On using the concurrent modelling procedure, it has been found that, not only can the small-signal performance be simulated accurately, but also the prediction of noise performance is in much better agreement with measurements than those of recent published models.

## 1 Introduction

Small-signal and noise behaviour of a microwave transistor around a bias point are usually determined by the scattering parameters  $S_{11}$ ,  $S_{22}$ ,  $S_{21}$ ,  $S_{12}$  and the noise parameters  $F_{opt}$ ,  $\Gamma_{opt}$ ,  $R_N$  over its operational bandwidth. Both  $S$ -parameters and noise parameters are frequency-dependent and intrinsic properties of the device.  $S$ -parameters are used to represent the device signal power gains and mismatch losses, whereas the noise figure describes the degradation of the signal-to-noise ratio between the input and output of the device.

Adequate representations of circuit elements for both the signal and noise behaviour over their whole operational ranges are essential for the design of monolithic microwave integrated circuits. Even the characterisation of passive elements, which is relatively simple for low frequencies, can be difficult for microwave frequencies. In the case of semiconductor devices, that are characterised by highly nonlinear models with a large set of parameters and complicated relationships between them, a proper selection of values for these parameters is a nontrivial task. Indeed, if performed inadequately this can significantly distort the simula-

tion results. Usually these model parameters cannot be determined by direct measurements because of device nonlinearities. Traditional device modelling is based upon the following two fundamentals which may be considered as penalties. (i) Studies of small-signal performance have been separated from those of noise performance. Published literature is either concerned with only the small-signal model or concentrates on the noise behaviour description based on existing small-signal equivalent circuits that have nothing to do with the device noise characteristics [1-5]. In [6] and [7] these two behaviours are combined in a unified classic circuit model. (ii) The standard approach for characterising an integrated microwave device and its enclosing package is to model each part separately by an equivalent circuit which is fitted to electrical measurements. This approach has been shown to lead to an inaccurate description of the contribution the package makes to the overall electrical characteristics of the packaged device described in [8].

Furthermore, the widespread optimiser-based extraction techniques suffer from the nonuniqueness of solutions [9]. Although, for example, some improvements have been aimed at taking into account additional measurements using a partitioning approach [10] or an automatic decomposition technique [11], uncertainties still exist with respect to the starting value problem.

In this work the device is modelled by a black box for which signal and noise parameters are evaluated through a neural network, based upon the fitting of both of these parameters to the corresponding measured data over the whole operational range from DC to more than 10GHz. The main purposes of the work are ordered as follows:

- (i) Establish a novel neural network of feedforward type with a single hidden layer.
- (ii) Using back-propagation and nonlinear types of activation functions, train the network for both the signal-noise behaviours over the operational bandwidth for any type of active device.
- (iii) Establish performance measure of the network.
- (iv) Predict the small-signal and noise behaviours at any operation frequency using the neural network which has already trained to make functional approximations of the device nonlinear characteristics in the vicinity of the chosen bias point.

## 2 Determination of small-signal and noise behaviours of active microwave devices

The signal and noise performance of an active microwave device around a bias point are usually given by

© IEE, 1996

*IEE Proceedings* online no. 19960150

Paper first received 26th June 1995 and in revised form 7th November 1995

F. Güneş and H. Torpi are with the Electronics and Communication Engineering Department, Yıldız Technical University, 80670 Maslak-Istanbul, Turkey

F. Gürgen is with the Computer Engineering Department, Boğaziçi University, 80815 Bebek-Istanbul, Turkey

the scattering  $S$  and noise  $N$  parameter vectors at the  $\omega$ -domain and the measured performance data over the operational band can be arranged in a table-form function as follows:

$$\begin{bmatrix} f_1: & S^{(1)} & N^{(1)} \\ f_2: & S^{(2)} & N^{(2)} \\ \vdots & \vdots & \vdots \\ f_N: & S^{(N)} & N^{(N)} \end{bmatrix} \quad (1)$$

where  $S^{(1)}, N^{(1)}, \dots, S^{(N)}, N^{(N)}$  are, respectively, the scattering and noise vectors at the  $f_1, \dots, f_N$  sample operation frequencies, and  $S^{(N)}$  and  $N^{(N)}$  can be given as follows:

$$\begin{aligned} [S^{(N)}]^T &= [s_{11}^{(N)} \ \varphi_{11}^{(N)} \ |s_{12}^{(N)} \ \varphi_{12}^{(N)} \ |s_{21}^{(N)} \ \varphi_{21}^{(N)} \ |s_{22}^{(N)} \ \varphi_{22}^{(N)}] \\ [N^{(N)}]^T &= [F_{opt}^{(N)} \ \Gamma_{opt}^{(N)} \ \varphi_{opt}^{(N)} \ R_N^{(N)}] \end{aligned} \quad (2)$$

The functions defined by eqns. 1 and 2 are utilised for training the neural network model of the device. Then, the performance vectors  $S^{(k)}$  and  $N^{(k)}$ , at a desired frequency,  $f_k$ , can be obtained from the network output by inputting the frequency  $f_k$ . If  $S^{(k)}$  and  $N^{(k)}$  are unmeasured, they are determined by the generalisation process of the neural network, which can be considered as the ability of the network to give good outputs to inputs it has not been trained on.

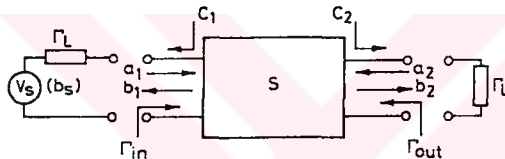


Fig. 1 Black box representation of an active microwave device

Once the performance vectors  $S^{(k)}$  and  $N^{(k)}$  are determined, an active microwave device can be represented as a black box at the frequency  $f_k$  (Fig. 1), which can be characterised in the following form [12]:

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \quad (3)$$

where  $c_1$  and  $c_2$  are the noise waves that are the time varying complex random variables characterised by a correlation matrix  $C_s$ , given by

$$[C_s] = \begin{bmatrix} |c_1|^2 & \overline{c_1 c_2^*} \\ \overline{c_2 c_1^*} & |c_2|^2 \end{bmatrix} \quad (4)$$

where the overbar indicates time-averaging with an implicit assumption of ergodicity and jointly wide-sense stationary processes. The diagonal terms of  $C_s$  give the noise power deliverable to the terminations in a 1 Hz bandwidth. The off-diagonal terms are correlation products. The noise wave correlation matrix  $C_s$  is Hermitian and its components are referred to as noise wave parameters which can be given in the terms of the performance vectors  $S^{(k)}, N^{(k)}$  by the following expressions:

$$\overline{|c_1|^2} = kT_{opt}(|s_{11}|^2 - 1) + \frac{kt|1 - s_{11}\Gamma_{opt}|^2}{|1 + \Gamma_{opt}|^2} \quad (5)$$

$$\overline{|c_2|^2} = |s_{21}|^2 \left( kT_{opt} + \frac{kt|\Gamma_{opt}|^2}{|1 + \Gamma_{opt}|^2} \right) \quad (6)$$

$$\overline{c_1 c_2^*} = \frac{-s_{21}^* \Gamma_{opt}^* kt}{|1 + \Gamma_{opt}|^2} + \frac{s_{11}}{s_{21}} \overline{|c_2|^2} \quad (7)$$

where  $k$  is the Boltzmann's constant,  $Z_0$  is the normalisation impedance,  $kt$  is the normalised temperature energy and  $kt$  and  $T_{opt}$  are, respectively, given by

$$kt = \frac{4kT_0 R_N}{Z_0} \quad \text{and} \quad T_{opt} = (F_{opt} - 1)T_0 \quad (8)$$

the transducer power gain of an active device is defined as the ratio of the power delivered to the load to the available power from the source. It is expressed in terms of the scattering parameters  $S$  and  $\Gamma_s$  and  $\Gamma_L$  terminations in the following forms [13]:

$$\begin{aligned} G_T &\triangleq \frac{P_L}{P_{A,S}} = G_T(\Gamma_s, \Gamma_L, S) \\ &= \frac{(1 - |\Gamma_L|^2)|s_{21}|^2(1 - |\Gamma_s|^2)}{|1 - s_{22}\Gamma_L|^2|1 - \Gamma_{in}\Gamma_s|^2} \end{aligned}$$

or

$$G_T(\Gamma_s, \Gamma_L, S) = \frac{(1 - |\Gamma_L|^2)|s_{21}|^2(1 - |\Gamma_s|^2)}{|1 - \Gamma_{out}\Gamma_L|^2|1 - s_{11}\Gamma_s|^2} \quad (9)$$

$M_{in}$  and  $M_{out}$  conjugate mismatch loss, respectively, at the input and output ports of the active device. They are given as

$$\begin{aligned} M_{in} &\triangleq \frac{P_{A,S}}{P_{in}} = M_{in}(\Gamma_s, \Gamma_L, S) \\ &= \frac{|1 - \Gamma_{in}\Gamma_s|^2}{(1 - |\Gamma_{in}|^2)(1 - |\Gamma_s|^2)} \end{aligned} \quad (10)$$

$$\begin{aligned} M_{out} &\triangleq \frac{P_{A,out}}{P_L} = M_{out}(\Gamma_s, \Gamma_L, S) \\ &= \frac{|1 - \Gamma_L\Gamma_{out}|^2}{(1 - |\Gamma_L|^2)(1 - |\Gamma_{out}|^2)} \end{aligned} \quad (11)$$

where

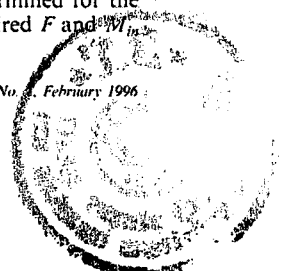
$$\Gamma_{out} = s_{22} + \frac{s_{12}s_{21}\Gamma_s}{1 - s_{11}\Gamma_s} \quad (12a)$$

$$\Gamma_{in} = s_{11} + \frac{s_{12}s_{21}\Gamma_L}{1 - s_{22}\Gamma_L} \quad (12b)$$

The noise figure of an active device is defined as the ratio of signal-to-noise ratios available at input and output; the  $N$  vector describes the dependence of the transistor noise figure  $F$  on the input termination (source) reflection coefficient  $\Gamma_s$ . These are linked through the relationship [13]

$$F(\Gamma_s, N) = F_{opt} + 4 \frac{R_N}{Z_0} \frac{|\Gamma_s - \Gamma_{opt}|^2}{(1 - |\Gamma_s|^2)|1 + \Gamma_{opt}|^2} \quad (13)$$

Today, a method called subnetwork-growth (SGM) [14], has been utilised for the CAD signal and noise analysis of the multiport networks. These are arbitrarily configured by all sorts of passive and active two-port devices. Many general purpose microwave CAD programs based on the SGM are implemented to analyse the microwave integrated circuits. As a result, black box characterisation of the active microwave devices over their whole operational frequencies has become especially important. After having fixed the  $S$  and  $N$  vectors of the two-port active device at an operation frequency, the  $(\Gamma_s, \Gamma_L)$  termination couple can be determined by making compromises among the performance functions  $G_T, F, M_{in}, M_{out}$  in some operational bandwidth. A typical application has been given in [15], where the  $(\Gamma_s, \Gamma_L)$  couple is determined for the maximum stable gain  $G_T$  under the required  $F$  and  $M_{in}$  at an operation frequency.



### 3 Neural network model

The multilayer perceptron (MLP), with a single hidden layer having the same number of units as the output layer, has been found to be sufficient to simulate an active microwave device: a back-propagation (BP) algorithm is utilised to train this network. [16-18] (Fig. 2).

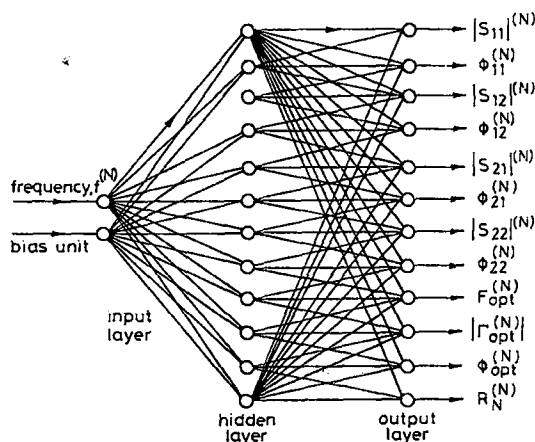


Fig.2 MLP for an active microwave device

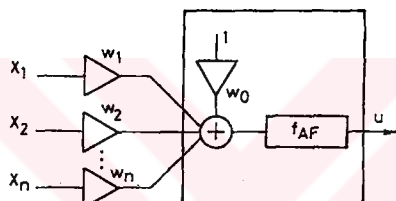


Fig.3 A perceptron node  
 $f_{AF}$  = activation function;  $w_1, \dots, w_n$  = weights;  $w_0$  = threshold (local memory);  $x_1, \dots, x_n$  = inputs

An additive bias is utilised as the second network input to ensure faster convergence which is taken as

$$Bias = \sqrt{\frac{1}{N_S} \sum_{i=1}^{N_S} f_i} \quad (14)$$

where  $N_S$  is the sample number. In Fig. 3 a block diagram of an operational node of an MLP is given. Thus the resulting signal from the hidden layer to the  $i$ th output node can be expressed in the form

$$\Phi_i(\mathbf{T}_i, \mathbf{W}, \mathbf{x}) = \sum_{h=1}^{N_h} T_{hi} g_h(\mathbf{W}_h, \mathbf{x}) + T_{ho} \quad (15)$$

and the net output of the  $i$ th output node is obtained as follows

$$\phi_i(\mathbf{T}_i, \mathbf{W}, \mathbf{x}) = T_{oi} g_i(\Phi_i) + T_i \quad (16)$$

where  $g_h$  and  $g_i$  are the basis functions for the  $h$ th hidden node and the  $i$ th output node, respectively, which are a sigmoid type of nonlinear function. In our case,  $g_h(\mathbf{W}_h, \mathbf{x})$  can be expressed in the following form:

$$g_h(\mathbf{W}_h, \mathbf{x}) = \frac{1}{1 + \exp\left(-\sum_{i=1}^n x_i W_{ih} - W_{oh}\right)} \quad (17)$$

In eqns. 15-17,  $\mathbf{x}$  is the input vector

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^t \quad (18)$$

$\mathbf{T}_i$  is the weighting vector between the  $i$ th output node and the hidden layer which can be written as

$$\mathbf{T}_i = [T_{1i}, T_{2i}, T_{3i}, T_{hi}, T_{Nh}]^t \quad (19)$$

$\mathbf{W}$  is the weighting matrix between the hidden and input layer which can be expressed as

$$\mathbf{W} = [W_1, W_2, \dots, W_h, \dots, W_{Nh}] \quad (20)$$

where  $\mathbf{W}_h$  is the weighting vector between the input layer and the  $h$ th hidden node and is given by

$$\mathbf{W}_h = [W_{1h}, W_{2h}, \dots, W_{nh}]^t \quad (21)$$

In eqns. 15-17,  $T_{ho}$  and  $T_i$  are the thresholds of the  $h$ th hidden and  $i$ th output nodes, respectively,  $W_{oh}$  is the local memory belonging to the  $h$ th hidden node.

As the result,  $i$ th output can be expressed in the form of the function  $F(\mathbf{P}, \mathbf{x})$  defined by the network architecture, that is, the number of hidden layers and nodes of each layer, weights of the connectivities between the nodes etc.

In our application, the learning process corresponds to the computation of  $\mathbf{P}$  values to minimise the error between  $y_i$ , the measured value, and  $F(\mathbf{P}, \mathbf{x})$ , over all training example pairs  $\{[x_i], y_i\}$  using a distance measure, the sum of square errors, for example:

$$E(\mathbf{P}) = \sum_{\{[x_i], y_i\}} [y_i - F(\mathbf{P}, [x_i])]^2 \quad (22)$$

Thus, we start with any set of weights and repeatedly change each weight by an amount proportional to  $\partial E / \partial P_p$ .

$$\Delta P_p = -\eta \frac{\partial E}{\partial P_p} \quad (23)$$

where  $\eta$  is the stepsize in descent. We assume that the training is completed when the error fails to decrease any further. In this case we take the best so far.

### 4 Performance measure and results

To evaluate the quality of the fit to measured data the following error terms are found to be convenient:

$$E_{S_{ij}} = \frac{1}{n} \sum_{k=1}^n \frac{|S_{ij}^k(\text{meas}) - S_{ij}^k(\text{predict})|}{|S_{ij}^k(\text{meas})|} \quad \text{for } i, j = 1, 2 \quad (24)$$

$$E_{N_i} = \frac{1}{n} \sum_{k=1}^n \frac{|N_i^k(\text{meas}) - N_i^k(\text{predict})|}{|N_i^k(\text{meas})|} \quad (25)$$

where  $S_{ij}$  and  $N_i$  are, respectively the signal and noise parameters.  $n$  is the number of discrete frequencies used. Total average error can be defined as the average of the signal and noise errors,

$$E_T = \frac{1}{4} \sum_{i=1}^4 E_{i(\text{signal})} + \frac{1}{3} \sum_{i=1}^3 E_{i(\text{noise})} \quad (26)$$

In Table 1 the measured values for the  $S$  and noise parameters of FET N24200A are given from its catalogue data file. The outputs of the neural network model of N24200A are given in Table 2 for the band from 1 to 30GHz over which both signal and noise parameters are simultaneously present in the catalogue file. The errors defined for each signal and noise parameter by eqns. 22 and 23 and the total average error eqn. 24 are also added to the end of Table 2.

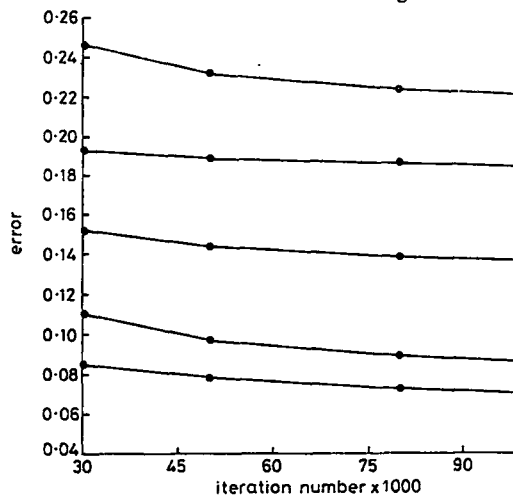
For each case of the simulation, the optimum number for sample and iteration used has been searched against the error. To give an example for their variations, the transistor N32684A was chosen due to

**Table 1: Manufacturer's values for signal and noise parameters for N23200A FET**

! FILENAME: N24200A.S2P VERSION: 5.0  
 ! NEC PART NUMBERS: NE24200 DATE: 6/91  
 ! BIAS CONDITIONS:  $V_{DS} = 2V$ ,  $I_{DS} = 10mA$   
 ! NOTE: S-PARAMETERS INCLUDE BOND WIRES.  
 ! GATE: TOTAL 2 WIRES, 1 PER BOND PAD, EACH WIRE 0.0132in. (335 $\mu$ m) LONG.  
 ! DRAIN: TOTAL 2 WIRES, 1 PER BOND PAD, EACH WIRE 0.0094in. (240 $\mu$ m) LONG.  
 ! SOURCE: TOTAL 4 WIRES, 2 PER SIDE, EACH WIRE 0.0070in. (178 $\mu$ m) LONG.  
 ! WIRE: 0.0007in. (17.8 $\mu$ m) DIAMETER, GOLD  
 ! NOTE: NOISE PARAMETERS FOR 28 AND 30GHz ARE EXTRAPOLATED, NOT MEASURED

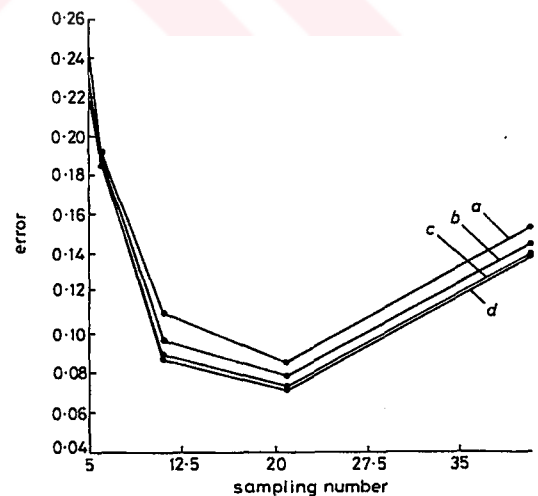
GHz	$S_{11}$		$S_{21}$		$S_{12}$		$S_{22}$		$F_{min}$	$\Gamma_{opt}$		$R_N/50$
	mag.	ang.	mag.	ang.	mag.	ang.	mag.	ang.		mag.	ang.	
0.1	0.999	-1	5.04	179	0.002	89	0.62	-1				
0.2	0.999	-3	5.02	178	0.004	89	0.62	-1				
0.5	0.999	-6	4.97	175	0.008	87	0.62	-4				
1.0	0.997	-12	4.88	170	0.016	84	0.62	-8	0.30	0.81	10	0.39
2.0	0.990	-23	4.70	161	0.030	77	0.61	-15	0.31	0.79	17	0.36
3.0	0.980	-34	4.54	152	0.042	71	0.61	-22				
4.0	0.970	-44	4.38	144	0.052	65	0.61	-29	0.33	0.75	31	0.33
5.0	0.950	-53	4.22	136	0.062	59	0.60	-36				
6.0	0.930	-62	4.08	128	0.071	53	0.59	-41	0.38	0.72	45	0.30
7.0	0.910	-71	3.93	120	0.079	48	0.59	-46				
8.0	0.890	-79	3.80	113	0.086	43	0.58	-51	0.43	0.70	59	0.27
9.0	0.870	-87	3.76	106	0.092	38	0.57	-56				
10.0	0.860	-94	3.54	99	0.099	34	0.56	-61	0.50	0.68	77	0.24
11.0	0.840	-102	3.42	92	0.104	30	0.55	-65				
12.0	0.820	-108	3.30	86	0.109	27	0.54	-70	0.60	0.66	92	0.22
13.0	0.800	-115	3.19	80	0.114	24	0.53	-74				
14.0	0.790	-121	3.08	74	0.119	21	0.51	-78	0.71	0.64	108	0.19
15.0	0.770	-128	2.97	68	0.123	18	0.50	-83				
16.0	0.750	-134	2.87	63	0.127	16	0.49	-87	0.85	0.62	126	0.18
17.0	0.740	-139	2.77	57	0.131	14	0.48	-91				
18.0	0.720	-145	2.68	52	0.135	12	0.47	-95	1.00	0.58	140	0.15
19.0	0.710	-150	2.59	47	0.138	10	0.46	-98				
20.0	0.690	-155	2.50	42	0.142	8	0.45	-102	1.20	0.55	153	0.13
22.0	0.660	-165	2.32	32	0.148	6	0.43	-109	1.50	0.52	164	0.11
24.0	0.640	-175	2.16	23	0.153	4	0.42	-116	1.80	0.49	175	0.10
26.0	0.610	-177	2.01	15	0.159	3	0.41	-122	2.10	0.48	-176	0.08
28.0	0.590	-168	1.87	7	0.163	1	0.41	-128	2.40	0.46	-168	0.07
30.0	0.570	-160	1.73	-1	0.168	0.1	0.41	-134	2.80	0.46	-160	0.05

its sufficient number of supplied signal and noise parameters. The variations are shown in Figs. 4 and 5.



**Fig.4** Error against iteration number for which various sampling numbers are taken as constant  
 Sampling numbers: a 5; b 6; c 41; d 11; e 21

The transistor N24200A was chosen to demonstrate the capacity of the neural network model for wide



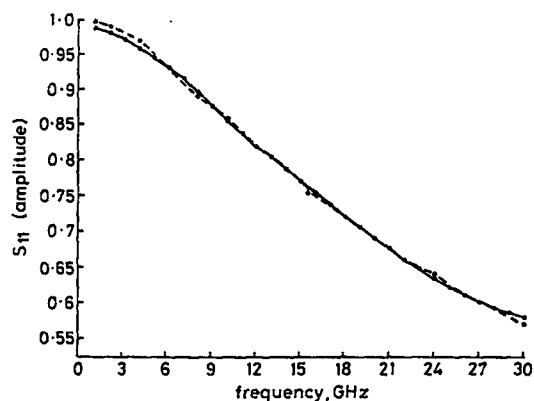
**Fig.5** Error against sampling number for which various iteration numbers are taken as constant  
 Iteration numbers (thousands): a 30; b 50; c 80; d 100

operational range of frequency for both signal and noise parameters (1 – 30GHz) where 16 samples are found to be sufficient for the network to learn. Simulation results for the N24200A transistor are also given

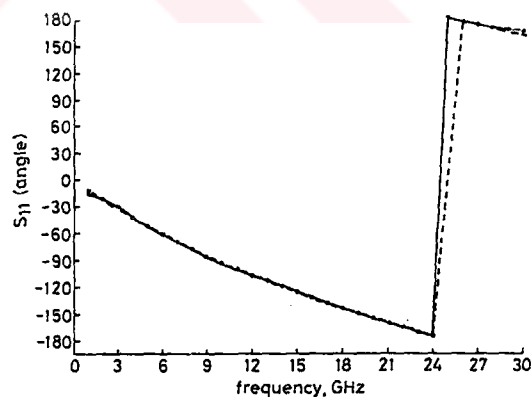
Table 2 Outputs of neural network model for N23200A

Signal and noise parameters for learning												
$f$ , GHz	$S_{11}$	$S_{21}$	$S_{12}$	$S_{22}$	$F_{min}$	$\Gamma_{opt}$	$R_N/50$					
1.0	0.997	-12.000	4.880	170.000	0.016	84.000	0.620	-8.000	0.300	0.810	10.000	0.390
2.0	0.990	-23.000	4.700	161.000	0.030	77.000	0.610	-15.000	0.310	0.790	17.000	0.360
4.0	0.970	-44.000	4.380	144.000	0.052	65.000	0.610	-29.000	0.330	0.750	31.000	0.330
6.0	0.930	-62.000	4.080	128.000	0.071	53.000	0.590	-41.000	0.380	0.720	45.000	0.300
8.0	0.890	-79.000	3.800	113.000	0.086	43.000	0.580	-51.000	0.430	0.700	59.000	0.270
10.0	0.960	-94.000	3.540	99.000	0.099	34.000	0.560	-61.000	0.500	0.680	77.000	0.240
12.0	0.820	-108.000	3.300	86.000	0.109	27.000	0.540	-70.000	0.600	0.660	92.000	0.220
14.0	0.790	-121.000	3.080	74.000	0.119	21.000	0.510	-78.000	0.710	0.640	108.000	0.190
16.0	0.750	-134.000	2.870	63.000	0.127	16.000	0.490	-87.000	0.850	0.620	126.000	0.180
18.0	0.720	-145.000	2.680	52.000	0.135	12.000	0.470	-95.000	1.000	0.580	140.000	0.150
20.0	0.690	-155.000	2.500	42.000	0.142	8.000	0.450	-102.000	1.200	0.550	153.000	0.130
22.0	0.660	-165.000	2.320	32.000	0.148	6.000	0.430	-109.000	1.500	0.520	164.000	0.110
24.0	0.640	-175.000	2.160	23.000	0.153	4.000	0.420	-116.000	1.800	0.490	175.000	0.100
26.0	0.610	177.000	2.010	15.000	0.159	3.000	0.410	-122.000	2.100	0.480	-176.000	0.080
28.0	0.590	168.000	1.870	7.000	0.163	1.000	0.410	-128.000	2.400	0.460	-168.000	0.070
30.0	0.570	160.000	1.730	-1.000	0.168	0.000	0.410	-134.000	2.800	0.460	-160.000	0.050
Predicted signal and noise parameters												
$f$ , GHz	$S_{11}$	$S_{21}$	$S_{12}$	$S_{22}$	$F_{min}$	$\Gamma_{opt}$	$R_N/50$					
1.000	0.992	-13.829	4.875	169.717	0.017	83.465	0.617	-9.134	0.297	0.812	10.917	0.386
2.000	0.985	-22.417	4.707	161.419	0.028	77.715	0.614	-14.916	0.305	0.786	16.516	0.365
4.000	0.959	-44.373	4.382	144.512	0.053	65.511	0.606	-29.678	0.333	0.745	30.971	0.328
6.000	0.931	-60.962	4.080	127.820	0.069	53.158	0.593	-40.074	0.380	0.724	44.336	0.297
8.000	0.895	-78.936	3.781	111.545	0.086	42.050	0.577	-51.161	0.441	0.703	60.658	0.267
10.000	0.857	-94.854	3.533	98.415	0.099	33.606	0.560	-61.167	0.508	0.683	76.692	0.242
12.000	0.822	-108.12	3.314	86.946	0.109	26.602	0.539	-69.757	0.590	0.661	91.846	0.219
14.000	0.788	-120.82	3.092	75.322	0.118	20.354	0.514	-78.232	0.701	0.635	107.991	0.195
16.000	0.752	-133.38	2.872	63.605	0.127	15.329	0.488	-86.831	0.844	0.607	124.823	0.172
18.000	0.716	-144.84	2.675	52.244	0.134	11.762	0.465	-94.819	1.012	0.579	140.145	0.151
20.000	0.689	-155.12	2.497	40.936	0.141	9.238	0.448	-102.017	1.214	0.553	153.077	0.132
22.000	0.662	-165.10	2.322	29.859	0.147	7.275	0.434	-109.009	1.475	0.526	164.537	0.113
24.000	0.635	-175.07	2.149	20.548	0.153	5.714	0.422	-116.052	1.795	0.499	175.182	0.095
26.000	0.610	175.998	2.000	14.310	0.158	4.593	0.412	-122.471	2.119	0.474	-175.64	0.081
28.000	0.591	168.947	1.885	15.288	0.162	3.659	0.405	-127.601	2.391	0.455	-168.58	0.070
30.000	0.570	159.9	1.729	-6.5	0.168	1.0	0.410	-134.0	2.801	0.460	-160.0	0.05
Error analysis												
$E_{11} = 0.011053, E_{21} = 0.024918, E_{12} = 0.027162, E_{22} = 0.009100, E_{\Gamma} = 0.018058$												
$F_1 = 0.011198, F_2 = 0.012857, F_3 = 0.016921, F_4 = 0.013659$												

in Figs. 6-9, Figs. 10-13 and Figs. 14-17 which show quite good agreement of the signal and noise parameters over the fairly large operational bandwidth.

Fig. 6 Amplitude of  $S_{11}$  against frequency

Frequency variations of the interpolation errors for the  $S$ -parameters are given in Figs. 18 and 19 where one can see that the network has a high capability to interpolate between the data points used for learning. Similar interpolation properties of the network can be

Fig. 7 Angle of  $S_{11}$  against frequency

said to be valid for the noise parameters too. Frequency variations of the extrapolation errors of the  $S$ -parameters and  $N$ -parameters are given, respectively, in Figs. 20 and 21 for the contracted band between 4-24 GHz, where the data points used for the training within the band are shown too. As seen from the related Figures, the neural network can fairly well extrapolate within the fairly large bandwidth outside the trained band region.

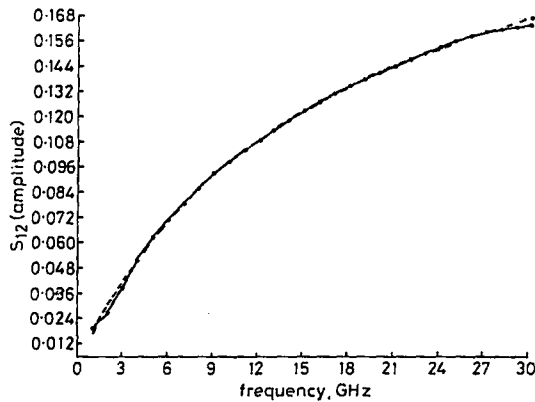


Fig. 8 Amplitude of  $S_{12}$  against frequency

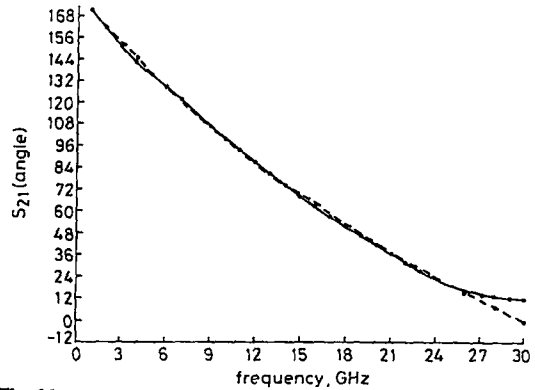


Fig. 11 Angle of  $S_{21}$  against frequency

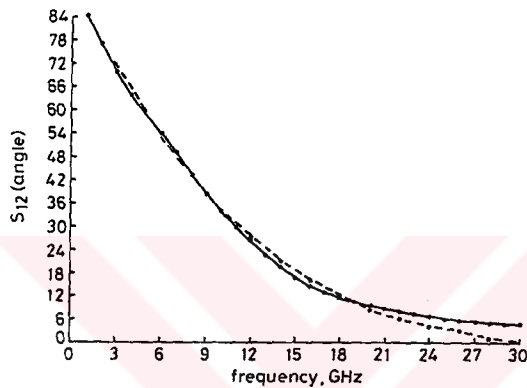


Fig. 9 Angle of  $S_{12}$  against frequency

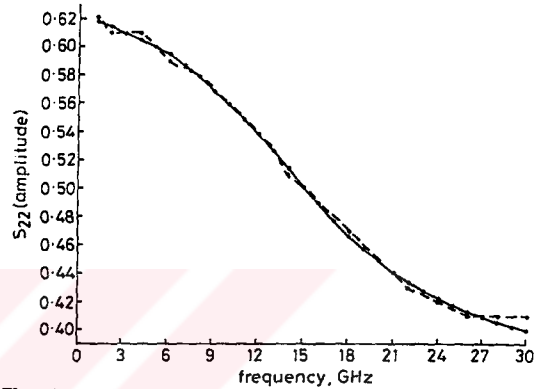


Fig. 12 Amplitude of  $S_{22}$  against frequency

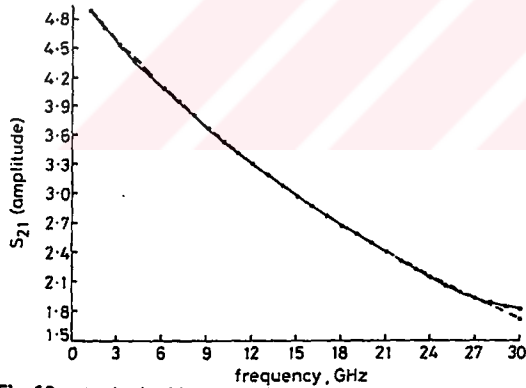


Fig. 10 Amplitude of  $S_{21}$  against frequency

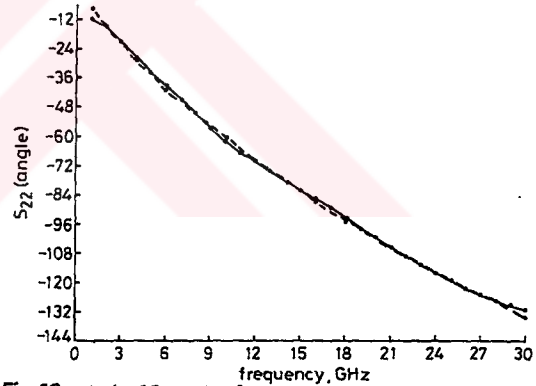


Fig. 13 Angle of  $S_{22}$  against frequency

5 References

- 1 FROELICH, R.K.: 'An improved model for noise characterization of microwave GaAs FETs', *IEEE Trans.*, 1990, MTT-38, (6), pp. 703-706
- 2 POSPIESZALSKI, M.W.: 'Modelling of noise parameters of MOSFETs and MODFETs and their frequency and temperature dependence', *IEEE Trans.*, 1989, MTT-37, (9), pp. 1340-1350
- 3 BERROTH, M., and BOSH, R.: 'Broad-band determination of the FET small-signal equivalent circuits', *IEEE Trans.*, 1990, MTT-38, (7), pp. 891-895
- 4 DAMBRINE, G., CAPPY, A., HELIDORE, F., and PLAYEZ, E.: 'A new method for determining the FET small-signal equivalent circuit', *IEEE Trans.*, 1988, MTT-36, (7), pp. 1151-1159
- 5 LADBROOKE, P.H.: 'MMIC design: GaSa FETs and HEMTs' (Artech House, Norwood, MA, 1989)

6

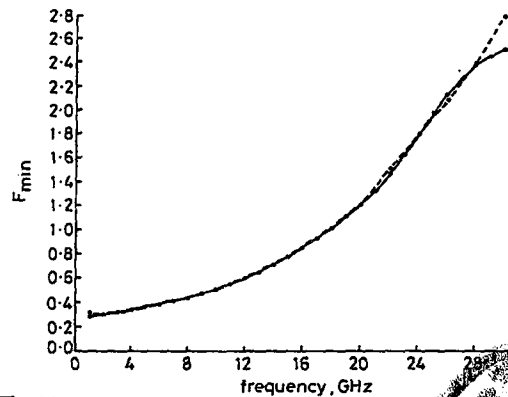
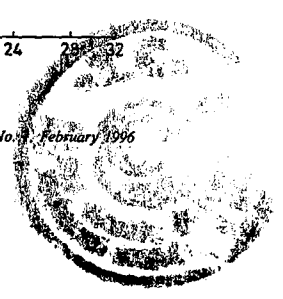


Fig. 14  $F_{min}$  against frequency





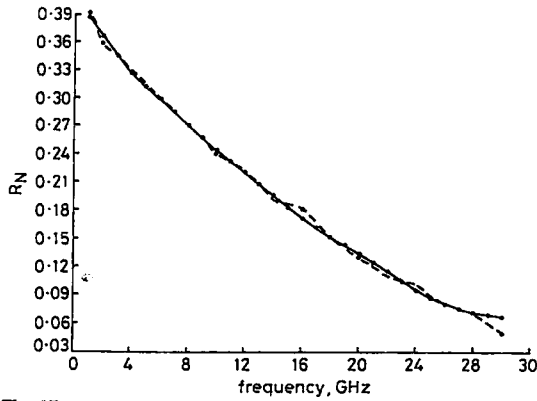


Fig. 15  $R_N$  against frequency

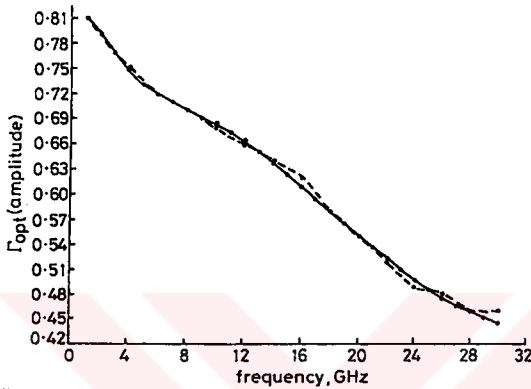


Fig. 16  $\Gamma_{opt}$  (amplitude) against frequency

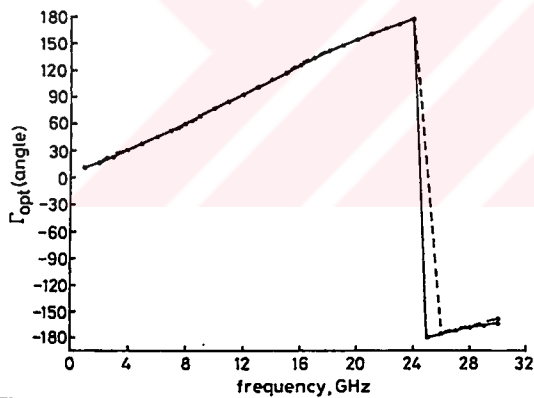


Fig. 17  $\Gamma_{opt}$  (angle) against frequency

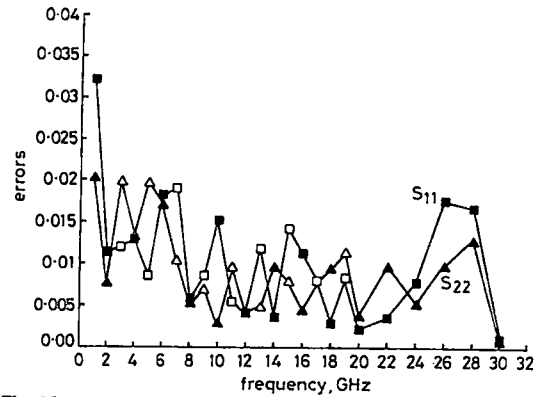


Fig. 18 Error-frequency variations for the interpolation of  $S_{11}$  and  $S_{22}$   
Values used in training: ■ ( $S_{11}$ ), ▲ ( $S_{22}$ )

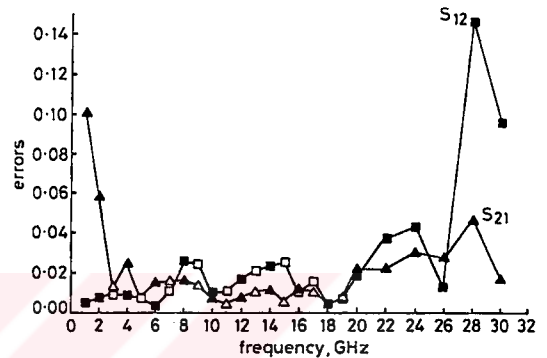


Fig. 19 Error-frequency variations for the interpolation of  $S_{12}$  and  $S_{21}$   
Values used in training: ■ ( $S_{12}$ ), ▲ ( $S_{21}$ )

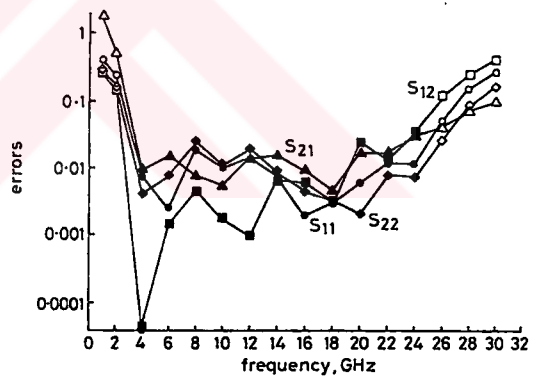
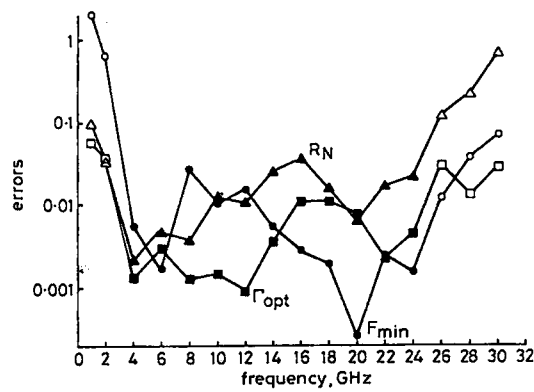


Fig. 20 Error-frequency variations for the extrapolation of  $S$  parameters for contracted band from 4 to 24 GHz  
Values used in training: ○ ( $S_{11}$ ), □ ( $S_{12}$ ), ▲ ( $S_{21}$ ), ◆ ( $S_{22}$ )

- 6 HU, Z.R., YANG, Z.M., FUSCO, V.F., and STEWART, J.A.C.: 'Unified small-signal-noise model for active microwave device', *IEE Proc. G*, 1993, 140, (1), pp. 55-60
- 7 ROUX, J.P., ESCOTTE, L., PLANA, R., GRAFFEUIL, and DELAGE, S.L.: 'Small-signal and noise model extraction technique for heterojunction bipolar transistor at microwave frequencies', *IEEE Trans.*, 1995, MTT-43, (2), pp. 293-298
- 8 BRIDGE, J.P., LADBROOKE, P.H., and HILL, A.J.: 'Characterisation of GaAs FET and HEMT chips and packages for accurate hybrid circuit design', *IEE Proc. H*, 1992, 139, (4), pp. 330-336
- 9 VAITKUS, R.L.: 'Uncertainty in the values of GaAs MESFET equivalent circuit elements extracted from measured two-port scattering parameters', Proceedings of the conference on *High speed semiconductor devices circuits*, Cornell University, Ithaca, New York, 1983, pp. 301-308

- 10 CURTICE, W.R., and CAMISA, : 'Self-consistent GaAs FET models for amplifier design and device diagnostics', *IEEE Trans.*, 1984, MTT-32, pp. 1573-1578
- 11 KONDOH, H.: 'An accurate FET modelling from measured S-parameters', IEEE MTT-S Int. Microwave Symp. Dig., Baltimore, 1986, pp. 377-380
- 12 WEDGE, W.S., and RUTLEDGE, D.B.: 'Wave techniques for noise modeling and measurement', *IEEE Trans.*, 1992, MTT-40, (11), pp. 2004-2012
- 13 VENDELIN, G.D., PAVIO, A.M., and ROHDE, U.L.: 'Microwave circuit design using linear and nonlinear techniques' (John Wiley & Sons, 1990)
- 14 GUPTA, K.C., GARG, R., and CHADRA, R.: 'Computer-aided design of microwave circuits' (Artech House, Dedham, MA, 1981)





**Fig.21** Error-frequency variations for the extrapolation of  $N$  parameters for contracted band from 4 to 24GHz  
 Values used in training: ○ ( $F_{min}$ ), ■ ( $\Gamma_{opt}$ ), ▲ ( $R_N$ )

- 15 GÜNEŞ F., GÜNEŞ M., and FIDAN, M.: 'Performance characterisation of a microwave device', *IEE Proc. Circuits, Devices, Syst.*, 1994, 141, (5), pp. 337-344
- 16 GÜRGEN, F., ALPAYDIN, R., ÜNLÜAKIN, U., and ALPAYDIN, E.: 'Distributed and local neural classifiers for phoneme recognition', *Pattern Recognit. Lett.*, 1994, pp. 1111-1118
- 17 HUSH, D.R., and HORNE, B.G.: 'Progress in supervised neural networks', *IEEE Signal Process. Mag.*, 1993, 10, pp. 8-36
- 18 HERTZ, J., KROGH, A., and PALMER, G.: 'Introductions to the theory of neural computation', vol. 1, (Addison-Wesley, 1991)

# MULTIDIMENSIONAL SIGNAL-NOISE NEURAL NETWORK MODEL FOR MICROWAVE TRANSISTORS

Filiz GÜNEŞ<sup>1</sup>

Hamid TORPİ<sup>1</sup>

Fikret GÜRGEN<sup>2</sup>

<sup>1</sup> Yıldız Technical University, Electronics & Communication Eng. Dept., 80670 MASLAK-İSTANBUL/TÜRKİYE

<sup>2</sup> Boğaziçi University, Computer Eng. Dept., 80815, BEBEK-İSTANBUL/TÜRKİYE

## Abstract:

Signal and noise behaviours of microwave transistors are modeled through the neural network approach for the whole operating ranges including frequency, bias and configuration types. Here, the device is modeled by a black box whose small-signal and noise parameters are evaluated through a neural network, based upon the fitting of both of these parameters for multiple bias and configuration. The concurrent modeling procedure does not require to solve device physics equations repeatedly during optimization and by this type of modeling the signal (S) and noise (N) parameters can be predicted not only at a single operation frequency around the chosen bias condition for a configuration, at the same time for the whole operation frequency band for the same operating conditions, with good agreement compared to the measurements.

The relationship between  $x$  and  $y$  is multidimensional and nonlinear. To model such a multi-dimensional nonlinear relationship, a three-layer neural network is employed, which has  $n$  processing nodes (PN) in the input layer,  $p$  PN's in the output layer, and  $q$  PN's in the hidden layer, as shown in Fig. 1. Let

$$a^{(k)} = [a_1^{(k)} \ a_2^{(k)} \ \dots \ a_n^{(k)}]^t \quad (2.1)$$

$$b^{(k)} = [b_1^{(k)} \ b_2^{(k)} \ \dots \ b_p^{(k)}]^t \quad (2.2)$$

be vectors representing the  $k$ th sample of the input and output, respectively,  $k=1,2,\dots,N_s$  where  $N_s$  is the total number of the data samples. The weighting matrix between the hidden and input layer is  $W$ , and between the hidden and output layer is  $V$ , which can be expressed as

$$W = [W_1, W_2 \ \dots \ W_h \ \dots \ W_q] \quad (3.1)$$

$$V = [V_1 \ V_2 \ \dots \ V_h \ \dots \ V_p] \quad (3.2)$$

where  $W_h$  vector is the weighting vector between the  $h$ th hidden node and the input layer:

$$W_h = [W_{1h} \ W_{2h} \ \dots \ W_{ih} \ \dots \ W_{nh}]^t \quad (3.3)$$

$V_j$  is the weighting vector between the  $j$ th output node and the hidden layer:

$$V_j = [V_{1j} \ V_{2j} \ \dots \ V_{qj}]^t \quad (3.4)$$

The signal resulted from the hidden layer to the  $j$ th output node can be expressed in the form of

$$\Phi_j(x, W, V_j, \theta) = \sum_{h=1}^q V_{hj} g_h(x, W_h, \theta_h) \quad (4.1)$$

and the net output of the  $i$ th output node is obtained as follows

$$y_i(x, W, V_j, \theta, T_j, \Theta_j) = T_j f_j(\Phi_j + \Theta_j) \quad (4.2)$$

where  $g_h$  and  $f_j$  are the basis functions for the  $h$ th hidden node and the  $j$ th output node, respectively, which are sigmoid type of nonlinear functions in our application, and  $g_h(x, W_h, \theta_h)$  can be expressed in the following form:

## I. DESCRIPTION OF THE WORK

In this work the signal-noise neural network in [1] is extended to include bias condition ( $V_{DS}, I_{DS}$ ) and configuration type (CT) so that performance parameters of the device can be generalized not only at a single operation frequency around the trained bias condition which may be named as Single Frequency Generalisation (SFG), at the same time in the whole operation frequency band around an untrained bias condition which may be named as Whole Frequency Band Generalisation (WFBG). The same performance measures as [1] are utilized for this model too. The literature is given extensively in [1]. The multiple bias-configuration signal-noise neural network is described in the second section in details, and worked examples and conclusion are given in the last section.

## II THE MULTIPLE BIAS & CONFIGURATION SIGNAL-NOISE NEURAL NETWORK

### Structure of the Network

A neural network is a simplified mathematical model of a biological neural network. It consists of a collection of interconnected neurons. Let

$$\begin{aligned} x &= [x_1 \ x_2 \ \dots \ x_n]^t \\ y &= [y_1 \ y_2 \ \dots \ y_p]^t \end{aligned} \quad (1.1)$$

be respectively input and output vectors of  $n$  and  $p$  dimensions, in the signal-noise neural network,  $x$  is a 4-dimensional vector containing frequency  $f_s$ , bias condition ( $V_{DS}, I_{DS}$ ) and configuration type (CT),  $y$  is a 12-dimensional vector which gives S- and N- parameters.



$$g_h(\mathbf{x}, \mathbf{W}_h, \theta_h) = \frac{1}{1 + \exp\left(-\left(\sum_{i=1}^n x_i W_{ih}\right) - \theta_h\right)} \quad (4.3)$$

In the eqns. (6.2) and (6.3),  $\Theta_j$  and  $\theta_h$  are the thresholds of the  $j$  th output and  $h$  th hidden nodes, respectively.

#### Determination of the Network Matrix P

If parameters of the network architecture is denoted by the matrix  $\mathbf{P}$ , the network parameter matrix  $\mathbf{P}$  will have  $n \times q + p \times q + q + p$  elements which consist of weighting factors between the input and hidden layers and the hidden and output layers, the local memories of the hidden and output nodes respectively. The training process can be defined as computation of the network parameter matrix  $\mathbf{P}$  so that the error function which is

$$E(\mathbf{P}) = \sum_{k=1}^{N_S} E^{(k)} = \sum_{k=1}^{N_S} \left[ \frac{1}{2} \sum_{j=1}^p \left( y_j^{(k)} - b_j^{(k)} \right)^2 \right] \quad (5)$$

is minimized, where  $y_j^{(k)}$  and  $b_j^{(k)}$  are respectively, the predicted and measured values of the  $j$  th output node at the training frequency  $f_k$ . This type of training process is also called backpropagation which is available in the two types of algorithms: 'on-line' training where neural network parameters are updated after each sample presentation, and 'off-line' training where neural network parameters are updated after all samples are presented. In this work, we chose the 'on-line' training approach since it is more efficient in most cases. The update equations for  $V_{hj}, W_{ih}, \theta_h$  can be given as follows:

$$V_{hj}^{(k+1)} = V_{hj}^{(k)} - \eta \frac{\partial E^{(k)}}{\partial V_{hj}} + \alpha \left( V_{hj}^{(k)} - V_{hj}^{(k-1)} \right) \quad (6.1)$$

$$W_{ih}^{(k+1)} = W_{ih}^{(k)} - \eta \frac{\partial E^{(k)}}{\partial W_{ih}} + \alpha \left( W_{ih}^{(k)} - W_{ih}^{(k-1)} \right) \quad (6.2)$$

$$\theta_h^{(k+1)} = \theta_h^{(k)} - \eta \frac{\partial E^{(k)}}{\partial \theta_h} + \alpha \left( \theta_h^{(k)} - \theta_h^{(k-1)} \right) \quad (6.3)$$

and the similar equations can be written for  $\Theta_j$  and  $T_j$ . In the eqns. (8.1)-(8.3)  $\eta$  and  $\alpha$  are positive-valued between 0 and 1 learning rate and momentum, respectively. In this work, a simple program is used to search their optimum values depending on type of the worked transistor where the strategy is to determine optimum values for  $\alpha$  and  $\eta$  to ensure rapid convergence to the satisfied performance measures of the model. Thus we start any set for the network parameters and then repeatedly change each parameter by an amount proportional to the related sensitivity terms such as  $\frac{\partial E^{(k)}}{\partial V_{hj}}, \frac{\partial E^{(k)}}{\partial W_{ih}}, \frac{\partial E^{(k)}}{\partial \theta_h}$

according to update equations and assume that the training is completed when the error fails to decrease any

further, the network parameters are then the final set of values of  $V_{hj}, W_{ih}, \theta_h, T_j$  and  $\Theta_j$ . By defining the sensitivities of the network with respect to  $V_{hj}, W_{ih}, \theta_h$  can be given as follows after  $Z_h$  and  $F_j$  letting be defined as:

$$Z_h = g_h(\mathbf{x}, \mathbf{W}_h, \theta_h) = \frac{1}{1 + \exp\left(-\left(\sum_{i=1}^n x_i W_{ih}\right) - \theta_h\right)} \quad (7.1)$$

$$F_j = f_j(\Phi_j + \Theta_j) = \frac{1}{1 + \exp(-\Phi_j - \Theta_j)} \quad (7.2)$$

$$\frac{\partial E^{(k)}}{\partial V_{hj}} = \frac{\left[ \frac{1}{2} \sum_{j=1}^p (y_j - b_j)^2 \right]}{\partial V_{hj}} \quad (7.3)$$

$$\frac{\partial E^{(k)}}{\partial V_{hj}} = (y_j - b_j) T_j \frac{\partial f_j}{\partial \Phi_j} \frac{\partial \Phi_j}{\partial V_{hj}}$$

$$\frac{\partial E^{(k)}}{\partial V_{hj}} = (y_j - b_j) T_j F_j (1 - F_j) Z_h = \delta_j^{(3)} Z_h$$

$$\frac{\partial E^{(k)}}{\partial W_{ih}} = \sum_{j=1}^p \frac{\partial E^{(k)}}{\partial y_j} \frac{dy_j}{dF_j} \frac{dF_j}{d\Phi_j} \frac{d\Phi_j}{dZ_h} \frac{dZ_h}{dW_{ih}}$$

$$\frac{\partial E^{(k)}}{\partial W_{ih}} = \left[ \sum_{j=1}^p (y_j - b_j) T_j F_j (1 - F_j) V_{hj} \right] Z_h (1 - Z_h) a_i^{(k)} \quad (7.4)$$

$$\frac{\partial E^{(k)}}{\partial W_{ih}} = \sum_{j=1}^{N_o} \delta_j^{(3)} V_{hj} Z_h (1 - Z_h) a_i^{(k)} = \delta_h^{(2)} a_i^{(k)}$$

$$\frac{\partial E^{(k)}}{\partial \theta_h} = \sum_{j=1}^p \frac{\partial E^{(k)}}{\partial y_j} \frac{dy_j}{dF_j} \frac{dF_j}{d\Phi_j} \frac{d\Phi_j}{dZ_h} \frac{dZ_h}{d\theta_h}$$

$$\frac{\partial E^{(k)}}{\partial \theta_h} = \sum_{j=1}^p (y_j - b_j) T_j F_j (1 - F_j) V_{hj} Z_h (1 - Z_h) \quad (7.5)$$

$$\frac{\partial E^{(k)}}{\partial \theta_h} = \sum_{j=1}^{N_o} \delta_j^{(3)} V_{hj} Z_h (1 - Z_h) = \delta_h^{(2)}$$

$$\frac{\partial E^{(k)}}{\partial T_j} = \frac{\partial E^{(k)}}{\partial y_j^{(k)}} \frac{\partial y_j^{(k)}}{\partial T_j} = (y_j - b_j) F_j \quad (7.6)$$

$$\frac{\partial E^{(k)}}{\partial T_j} = \delta_j^{(3)} T_j^{-1} (1 - F_j)^{-1}$$

$$\frac{\partial E^{(k)}}{\partial \Theta_j} = \frac{\partial E^{(k)}}{\partial y_j} \frac{\partial y_j}{\partial F_j} \frac{\partial F_j}{\partial \Theta_j} = (y_j - b_j) T_j F_j (1 - F_j) \quad (7.7)$$

$$\frac{\partial E^{(k)}}{\partial \Theta_j} = \delta_j^{(3)}$$



where  $\delta_h^{(2)}$  and  $\delta_j^{(3)}$  represent local gradients at individual node in the second and third layers, respectively and using (9.3) and (9.5) they can be expressed as

$$\delta_h^{(2)} = \sum_{j=1}^N \delta_j^{(3)} V_{hj} Z_h (1 - Z_h) \quad (8.1)$$

$$\delta_j^{(3)} = (y_j - b_j) T_j F_j (1 - F_j) \quad (8.2)$$

### III WORKED EXAMPLES AND CONCLUSION

In the foregoing sections a neural-based technique is described for modeling of signal and noise behaviour of microwave transistors over their operation frequency band, biasing ranges and configuration types. In the work, this multiple bias-configuration neural network model has been applied into many transistors for which the manufacturer's characterization data is used as the training data, the results show that the predicted parameters are generally in a good agreement with the desired parameters.

In the following subsections, two worked examples are presented with the performance measures of the neural network models, where amount of the data used for the training is optimized for each bias condition against the error and iteration number, a worked example is given in [1]. Fig. 2 gives variations of the hidden nodes of the neural network against the training error, the iteration number taken as the parameter, in our model, which is taken to be equal to the number of the outputs nodes, which is almost optimum. The graphs in Fig. 2 also show how rapid the training error converges, since the error settles down in the range of 0,01 within 500 iterations. Fig. 3 gives variation of the training time with respect to the iteration number for the various types of transistor, where it should be noted that once the neural network has been trained, it responds at once to the desired input. The generalization process can be considered into the two categories: (i) Single Frequency Generalization (SFG): It can be defined as the performance prediction at a single operation frequency of the trained bias condition. In the work [1], the interpolation and extrapolation capabilities of the network are demonstrated in the error-frequency planes over a fairly large operation bandwidth at a single bias condition. In this work, the SFG procedure is applied to a lot of bias conditions for the different configuration types, and the same conclusion as [1] is reached, which is that the network has a high capability to interpolate between the data points used for learning. (ii) The Whole Frequency Band Generalization (WFBG): This can be defined as prediction of the unknown multidimensional performance function for the whole operation frequency band, which has 8 signal-dimensions and 4 noise-dimensions given by (2), at an untrained bias condition. Fig. 5 gives the frequency distribution of error for the whole frequency band generalization (WFBG) of the scattering parameters at the bias condition  $V_{CE}=8V$ ,  $I_C=20mA$  for the common collector configuration. From the examination of the WFBG processes, one can conclude that the network can predict the multidimensional performance function of the

transistor fairly well, especially if it is trained at the sufficient numbers of bias condition in the neighborhood.



**Example 1: Multiple Bias & Configuration Signal Neural Network Model for the Transistor  
NE219 Common Collector & Common Emitter Configuration**

Name of the transistor: NE219

Operation Frequency Bandwidth: 2 GHz - 6 GHz

Operations Bias Conditions

Configuration	Bias Voltage ( $V_{CE}$ )[V]	Bias Currents ( $I_C$ ) [mA]		
		10	20	30
CC	8	SFG	WFBG	SFG
		$E_t=0.050105$	$E_t=0.044207$	$E_t=0.032068$
		$E_{11}=0.027857$	$E_{11}=0.019730$	$E_{11}=0.016128$
		$E_{21}=0.037930$	$E_{21}=0.035988$	$E_{21}=0.036308$
		$E_{12}=0.037051$	$E_{12}=0.049394$	$E_{12}=0.029213$
CE	8	Fig.4	Fig.5	Fig.6
		$E_t=0.073243$	$E_t=0.065140$	$E_t=0.058195$
		$E_{11}=0.045472$	$E_{11}=0.039773$	$E_{11}=0.028880$
		$E_{21}=0.034876$	$E_{21}=0.044730$	$E_{21}=0.032119$
		$E_{12}=0.061781$	$E_{12}=0.060726$	$E_{12}=0.042287$
	$E_{22}=0.150843$	$E_{22}=0.115331$	$E_{22}=0.0129495$	
	Fig.7	Fig.8	Fig.9	

**Bias Condition:  $V_{CE}=8.0$  [Volt]  $I_C=10.0$  [mA]  
COMMON COLLECTOR CONFIG.  
 $E_{11}=0.027857$   $E_{21}=0.037930$   $E_{12}=0.037051$   
 $E_{22}=0.097582$   $E_t=0.050105$**

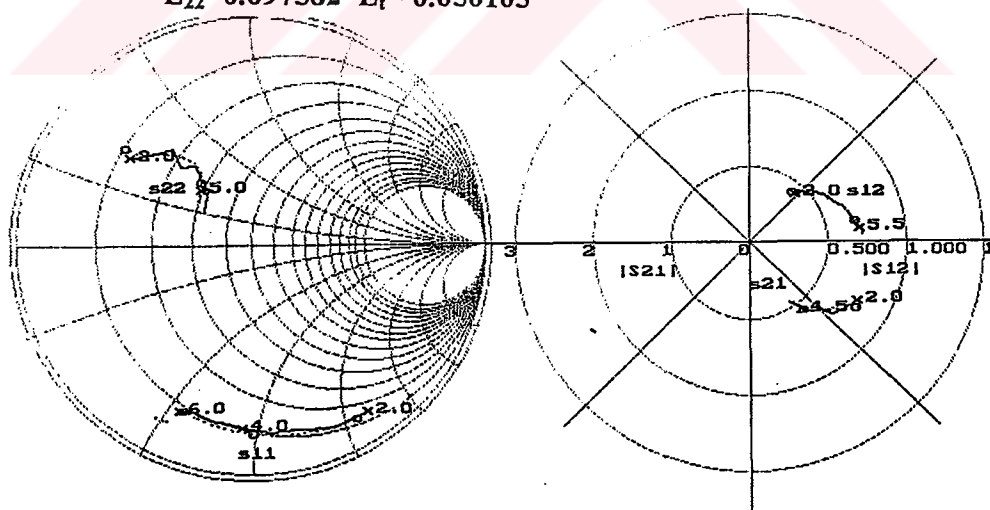


Fig.4 Predicted and measured S-parameter-frequency variations for  $V_{CE}=8.0$  V  $I_C=10$  mA at the common collector configuration.

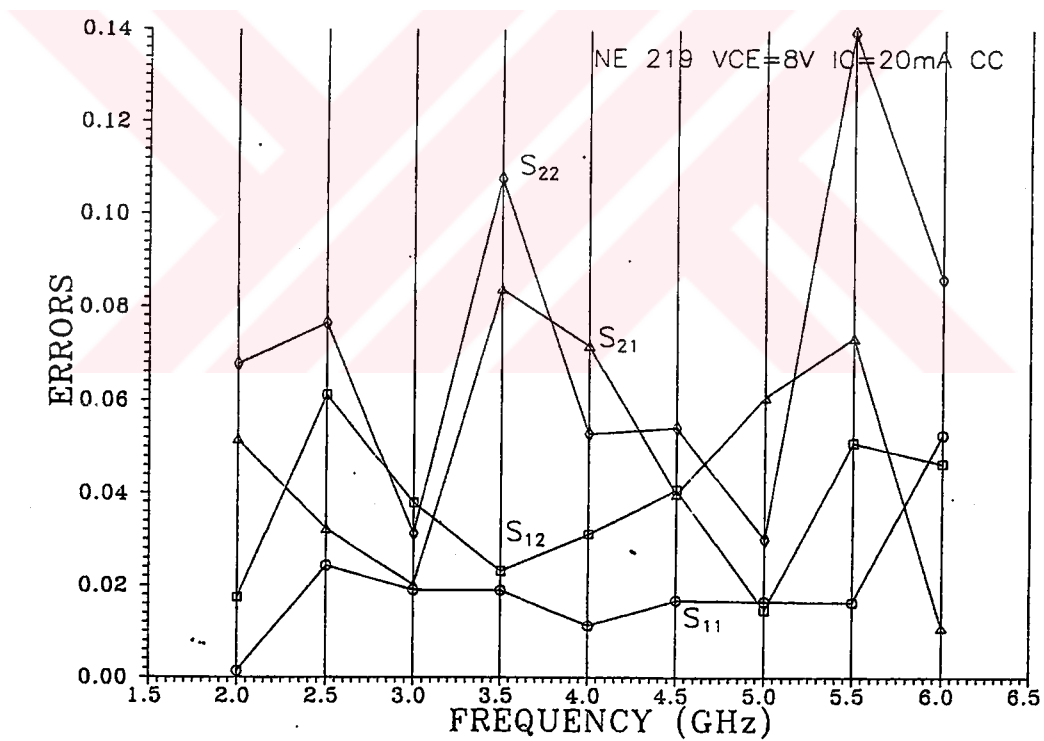
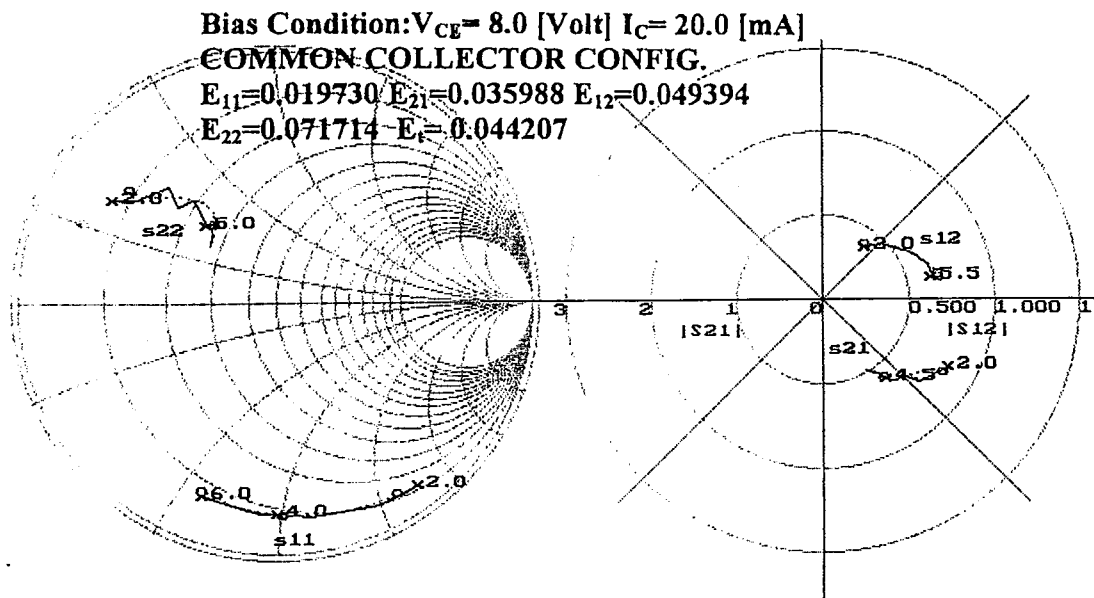
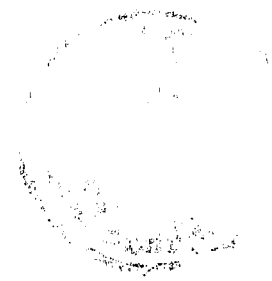


Fig.5 Predicted and measured S-parameter-frequency variations & error-frequency distributions for  $V_{CE} = 8V$   $I_C = 20mA$  at the common collector configuration (WFBG)



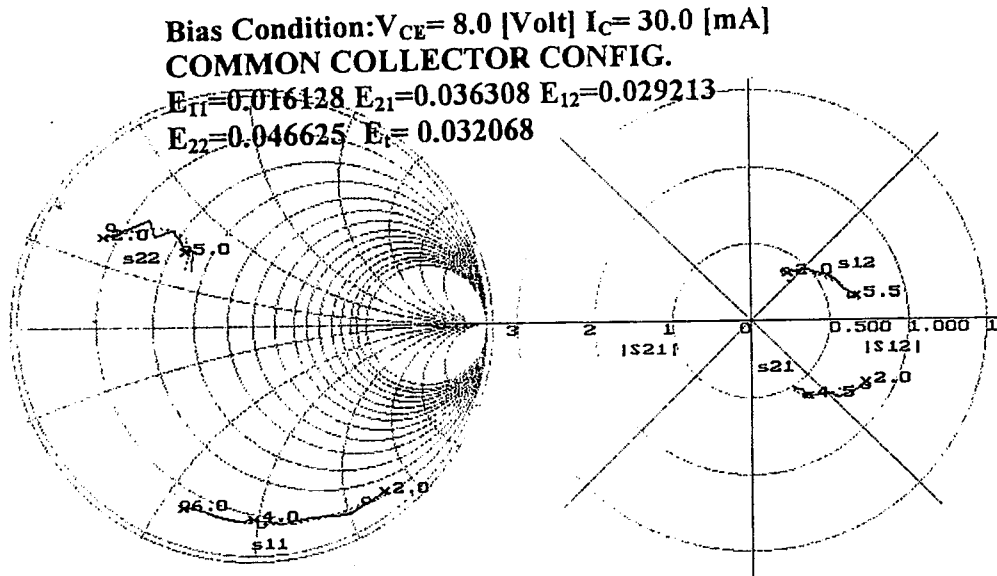


Fig.6 Predicted and measured S-parameter-frequency variations for  $V_{CE} = 8$  V  $I_C = 30$  mA at the common collector configuration.

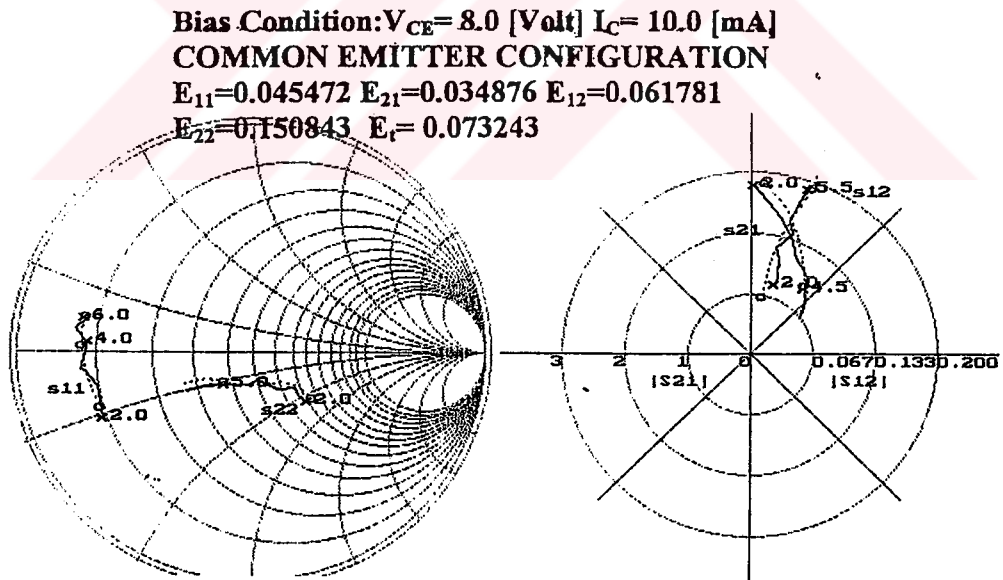


Fig.7 Predicted and measured S-parameter-frequency variations for  $V_{CE} = 8$  V  $I_C = 10$  mA at the common emitter configuration





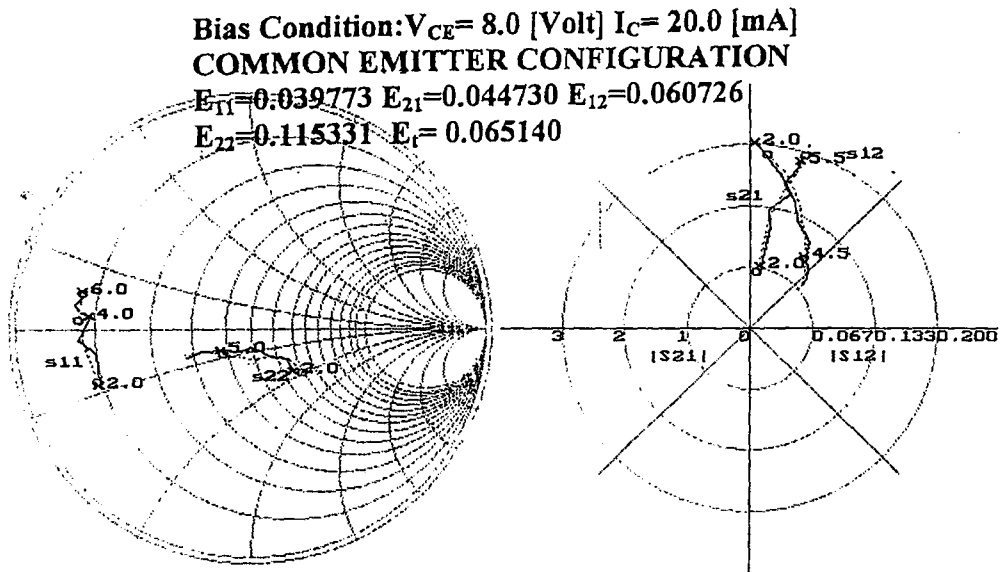


Fig.8 Predicted and measured S-parameter-frequency variations for  $V_{CE}=8$  V  $I_C=20$  mA at the common emitter configuration.(WFBG)

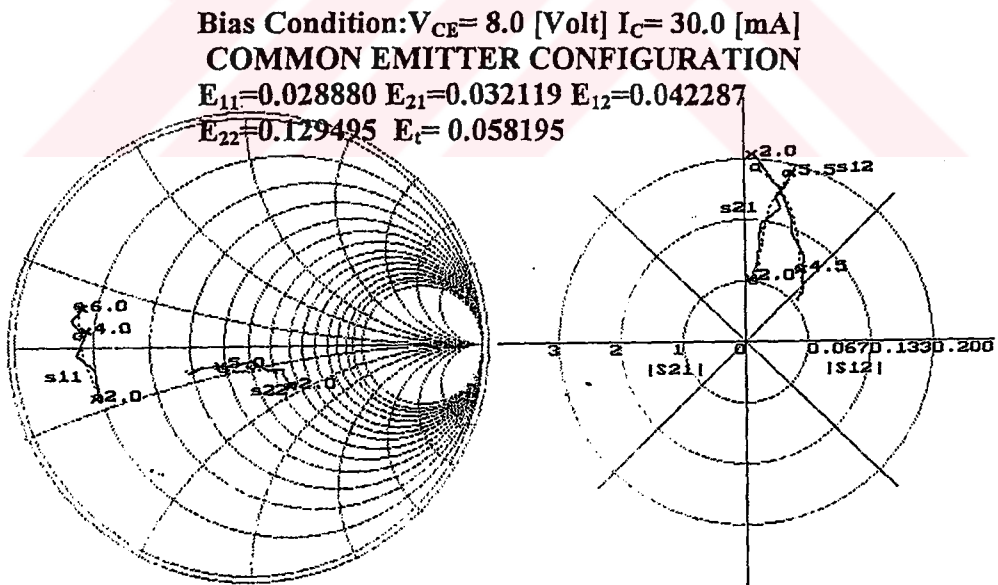


Fig.9 Predicted and measured S-parameter-frequency variations for  $V_{CE}=8$  V  $I_C=30$  mA at the common emitter configuration



**Example 2: Multiple Bias and Signal-Noise Neural Network Model for the Transistor NE02135 at the Common Emitter Configuration.**

Name of the transistor: NE02135		NMDA:No Measured Data Available.		
Operation Frequency Bandwidth: 0.1 GHz - 4 GHz				
Operations Bias Conditions				
Bias	Extrapolated values of Noise parameter for $V_{CE}=10V, I_C=30mA$ are used.			
Voltage ( $V_{CE}$ ) [V]	Bias Currents ( $I_C$ ) [mA]			
10.0	5.0	10.0	20.0	30.0
	SFG	WFBG	SFG	SFG
	$E_i=0.041287$	$E_i=0.079736$	$E_i=0.037003$	$E_i=0.044565$
	$E_{11}=0.033850$	$E_{11}=0.034377$	$E_{11}=0.023597$	$E_{11}=0.018405$
	$E_{21}=0.044067$	$E_{21}=0.056279$	$E_{21}=0.043421$	$E_{21}=0.049997$
	$E_{12}=0.043720$	$E_{12}=0.097377$	$E_{12}=0.046626$	$E_{12}=0.067227$
	$E_{22}=0.043512$	$E_{22}=0.130912$	$E_{22}=0.034370$	$E_{22}=0.042631$
	$F_1=0.030823$	$F_1=NMDA$	$F_1=0.023949$	$F_1=0.01918$
	$F_2=0.102794$	$F_2=NMDA$	$F_2=0.052824$	$F_2=0.07685$
	$F_3=0.123793$	$F_3=NMDA$	$F_3=0.096364$	$F_3=0.0659$
	$F_4=0.085804$	$F_4=NMDA$	$F_4=0.057712$	$F_4=0.0539$
	Fig.4	Fig.5	Fig.6	Fig.7

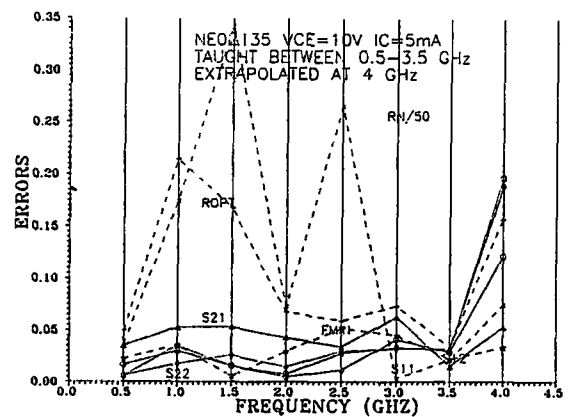
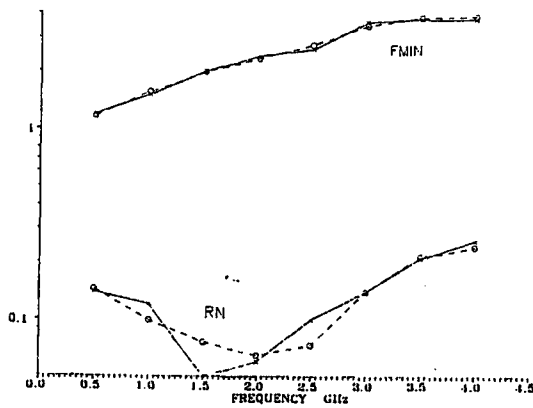
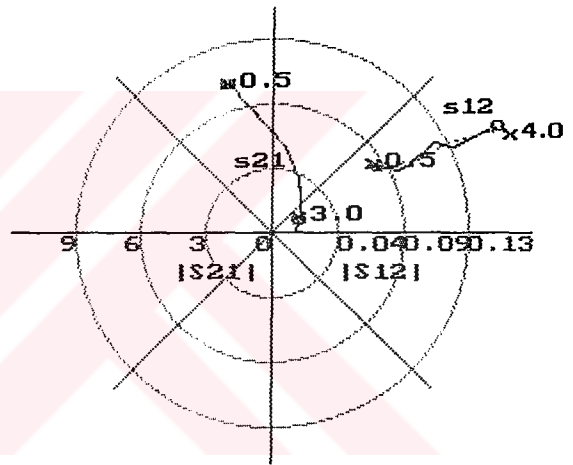
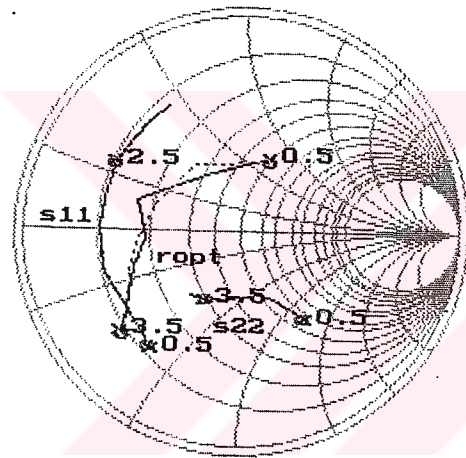
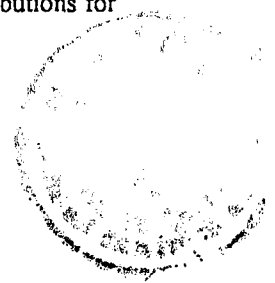


Fig.10 Predicted and measured S&N-parameters-frequency variations & error frequency distributions for  $V_{CE}=10 V, I_C=5 mA$  at the common emitter configuration of the transistor NE02135.(SFG)



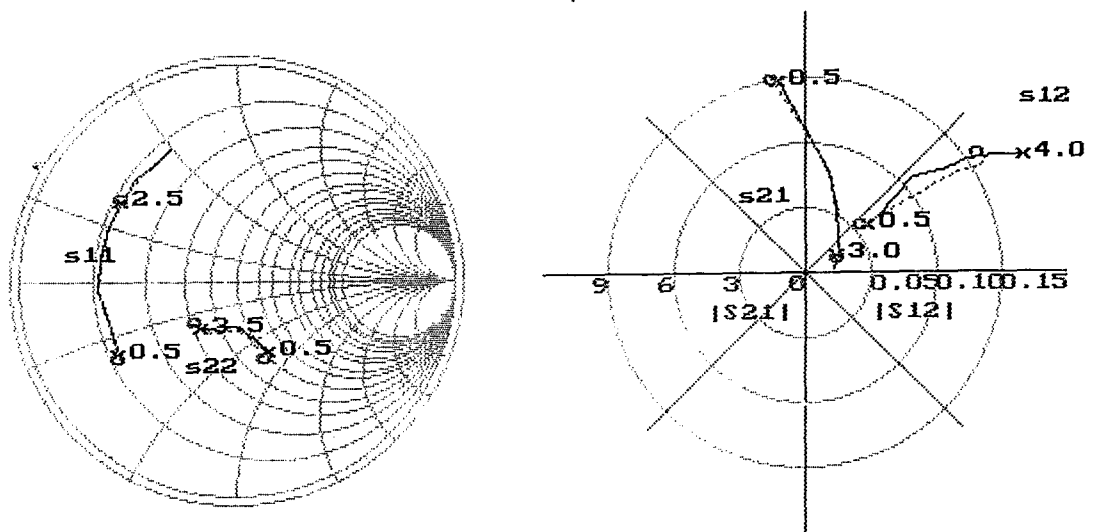


Fig. 11 Predicted and measured S-parameter-frequency variations for  $V_{CE}=10$  V  $I_C=10$  mA at the common emitter configuration of the transistor NE02135.(WFBG)

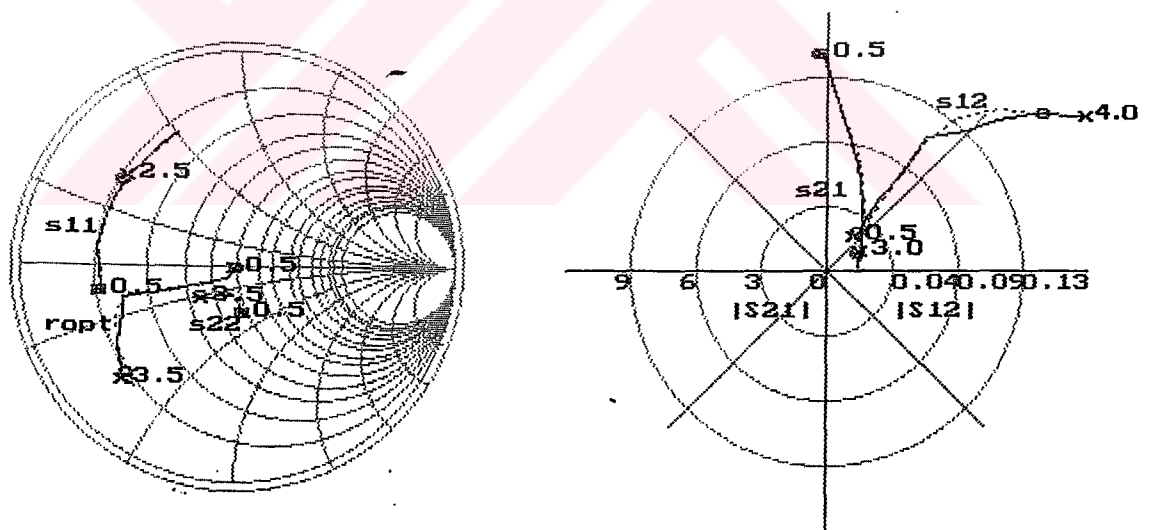
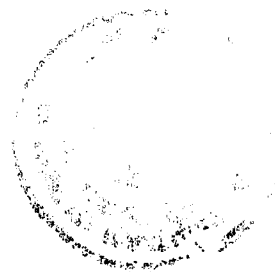


Fig. 12 Predicted and measured S-parameter-frequency variations for  $V_{CE}=10$  V  $I_C=30$  mA at the common emitter configuration of the transistor NE02135.(SFG)



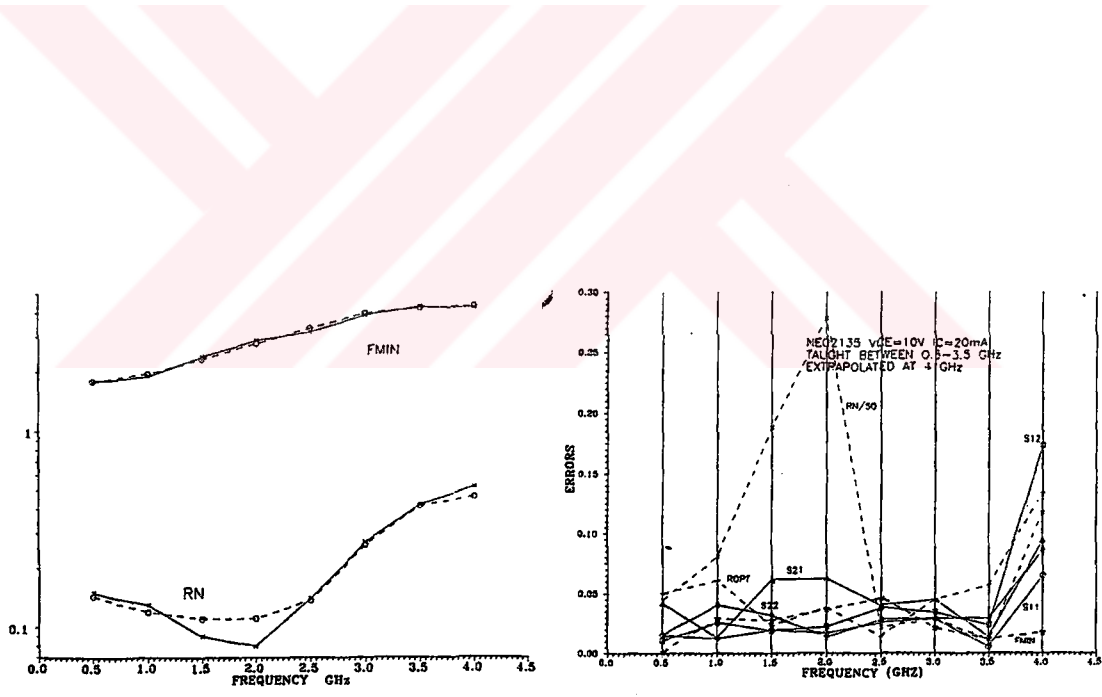
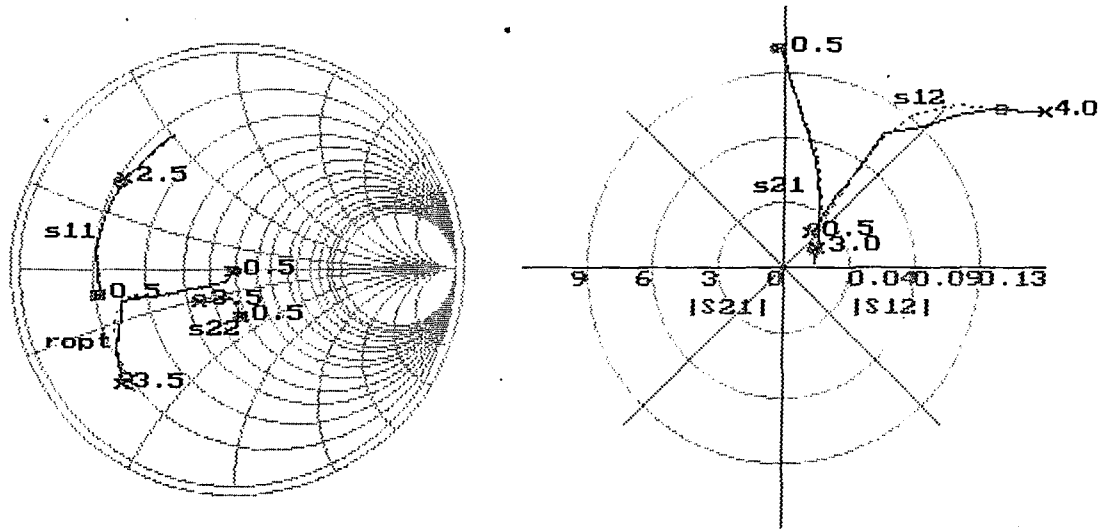


Fig.13 Predicted and measured S&N-parameter-frequency variations & error-frequency distributions for  $V_{CE}=10\text{ V}$   $I_C=20\text{ mA}$  at the common emitter configuration of the transistor NE02135.(SFG)



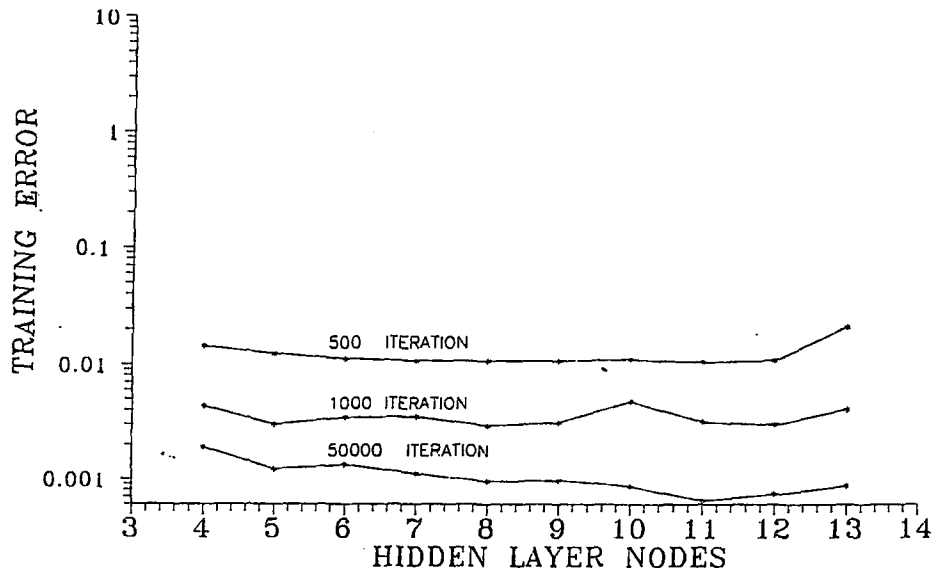


Fig. 2 Variations of the hidden nodes of the neural network against the training error for different Iteration number .

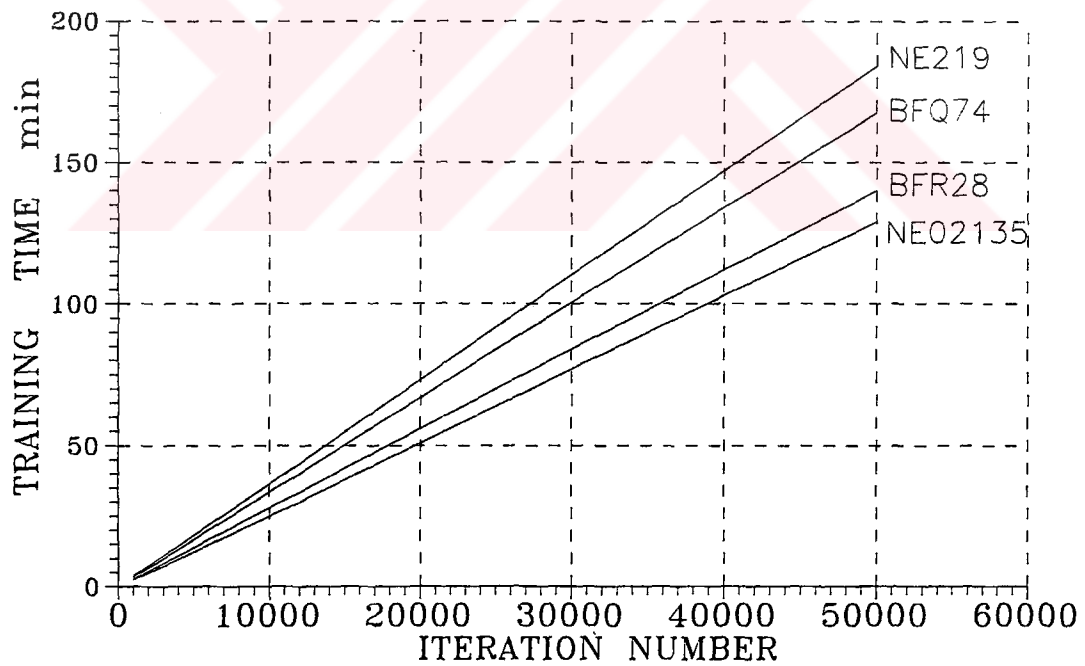


Fig. 3 Variation of the training time with respect to the iteration number for the various types of transistor



In this work, we have presented a nontraditional approach to modeling of microwave transistors, so that a multiple bias/configuration signal-noise neural network model is obtained, which is shown to be capable of giving results in a good agreement with the measured values. By exploiting the flexibility and generality of the neural network model, we have demonstrated its use for the device characterisation for its signal and noise performance over its whole operation ranges including frequency, bias conditions of the possible configuration types. At the same time, results of our work have demonstrated the feasibility and efficiency of using neural networks in the practical interactive design and optimisation methods for the physics-based device modeling.

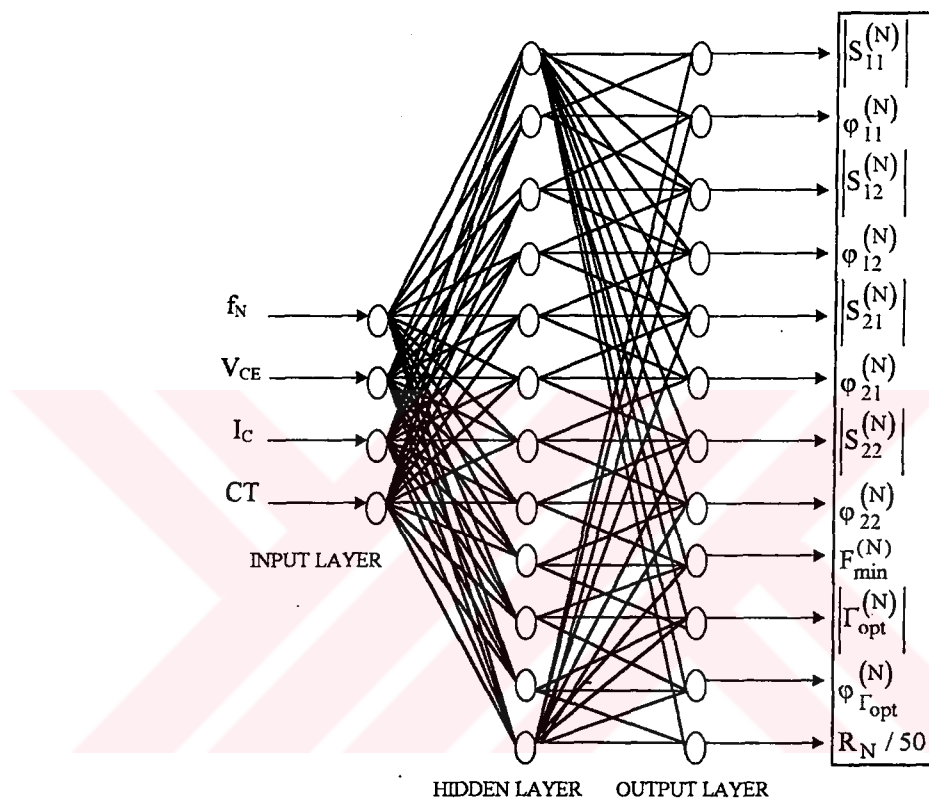
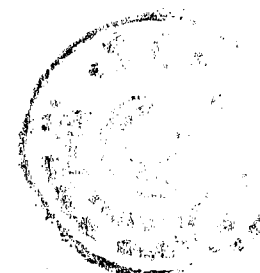


Fig.1 Signal-Noise Neural Network

### References

- [1] GÜNEŞ, F., GÜRGEN, F., TORPİ H.: "Signal-Noise neural network model for active microwave devices", IEE Proc.-Circuit Devices Syst., 1996, 143,(1), pp.1-8.
- [2] BARBY, J.A., VLACH, J., SINGHAL, K.: "Polynomial splines for MOSFET model approximation", IEEE Trans. Computer-Aided Design, 1988, 7, (5), pp. 557-567.
- [3] BIERNACKI, R.M., BANDLER, J.W., SONG, J. and ZHANG, Q.J.: "Efficient quadratic approximation for statistical design", IEEE Trans. Circuits Syst., 1989, 36, (11), pp.1449-1454.
- [4] MEIJER, P.B.L.: "Fast and smooth highly nonlinear multidimensional table models for device modeling", IEEE Trans. Circuit Syst., 1990, 37, (3), pp. 335-346.
- [5] ZAAB, H.A., ZHANG, Q.NAKHLA, M.: "A neural network modeling approach to circuit optimisation and statistical design", IEEE Trans. on Microwave Theory and Tech., 1995, 43, (6), pp.1349-1357.
- [6] BRIDGE, J.P., LADBROOKE, P.H. AND HILL, A.J., "Characterisation of GaAs FET and HEMT chips and packages for accurate hybrid circuit design", IEE Proceedings-H, 1992, Vol.139,(4), pp. 330-336.



## NEURAL NETWORK SIMULATORS FOR ACTIVE MICROWAVE DEVICES

Hamid TORPI\* Filiz GÜNEŞ\* Fikret GÜRGEN\*\*

\*Yıldız Technical University, Electronics & Communication Eng. Dept., MASLAK-ISTANBUL/TURKEY

\*\*Boğaziçi University, Computer Eng. Dept., BEBEK-ISTANBUL/TÜRKIYE

The small-signal and noise behaviours of a microwave transistor around bias point are usually determined by the scattering parameters  $S_{11}, S_{22}, S_{21}, S_{12}$  and the noise parameters  $F_{opt}, \Gamma_{opt}, R_N$  over its operating bandwidth. Both S-parameters and noise parameters are frequency dependent and intrinsic properties of a device. S-parameters are used to represent the device signal power gains and mismatch loss, whereas the noise figure describes the degradation of the signal-to-noise ratio between the input and output of a device.

As shown in the Fig.1 noise model parameters also require some small-signal modelling information, because both sets of parameters are related through the physical and geometrical parameters of the device e.g. both are strongly influenced by the gate length of the active layer. In view of this, and in order that the equivalent circuit can best predict both the small-signal and noise behaviours of the device simultaneously, its equivalent model should be extracted by both measured S-parameters and noise parameters. Published literature is either concerned with only the small-signal model or concentrates on the noise behaviour descriptions based on existing small-signal equivalent circuits that have nothing to do with the device noise characteristics. [1-4] A recent work gives an optimisation for the MESFET model parameters both S and noise parameters [5].

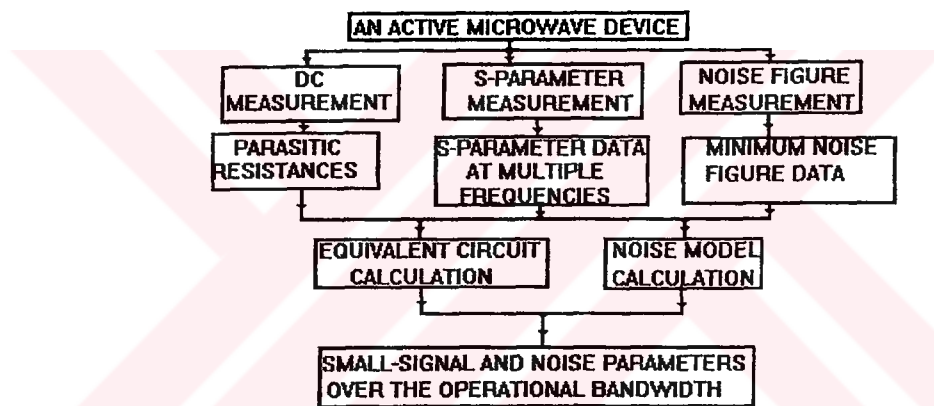


Fig. 1 General procedure for Determination of Small-Signal and Noise Performance of an Active Microwave device

The main purposes of this work can be ordered as follows:

- (i) Establish a novel neural network of feedforward type with a single hidden layer,
- (ii) Using back-propagation and nonlinear types of activation functions, train the network for both the signal-noise behaviours over the operational bandwidth for any type of active device,
- (iii) Establish performance measure of the network
- (iv) predict the small-signal and noise behaviours at any operation frequency concurrently using the trained network which makes functional approximations of the device nonlinear characteristics in the vicinity of the chosen bias point.

The signal and noise performance of an active microwave device around a bias point are usually given by the scattering  $[S]$  and noise  $[N]$  parameter vectors at the  $\omega$ -domain :

$$\begin{aligned}
 f_1 & : \begin{bmatrix} S^{(1)} & N^{(1)} \end{bmatrix} \\
 f_2 & : \begin{bmatrix} S^{(2)} & N^{(2)} \end{bmatrix} \\
 & \vdots \\
 f_N & : \begin{bmatrix} S^{(N)} & N^{(N)} \end{bmatrix}
 \end{aligned} \tag{1}$$

where  $[S^{(1)}], [N^{(1)}], \dots, [S^{(N)}], [N^{(N)}]$  are respectively, the scattering and noise vectors at  $f_1, f_2, \dots, f_N$  frequencies,  $[S^{(N)}]$  and  $[N^{(N)}]$  can be given as follows

$$\begin{bmatrix} S^{(N)} \end{bmatrix}^{-1} = \begin{bmatrix} |S_{11}|^{(N)} & \Psi_{11}^{(N)} & |S_{12}|^{(N)} & \Psi_{12}^{(N)} & |S_{21}|^{(N)} & \Psi_{21}^{(N)} & |S_{22}|^{(N)} & \Psi_{22}^{(N)} \end{bmatrix}$$

$$\begin{bmatrix} N^{(N)} \end{bmatrix}^{-1} = \begin{bmatrix} F_{opt}^{(N)} & |\Gamma_{opt}|^{(N)} & \Psi_{opt}^{(N)} & R_N^{(N)} \end{bmatrix}$$
(2)

The multilayer perceptron (MLP) with single hidden layer having the same number of units as the output layer has been found to be sufficient to simulate an active microwave device and back-propagation (BP) algorithm is utilized to train this network.[6]

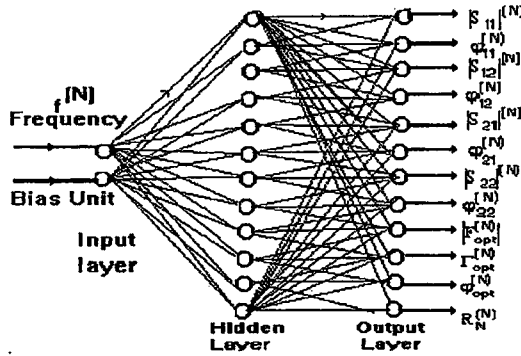


Fig. 2 MLP Simulator for an Active Microwave Device

An additive bias is utilized as the second network input to ensure faster convergence ,which is taken as

$$Bias = \sqrt{\frac{1}{N_s} \sum_{i=1}^{N_s} f_i}$$
(3)

Where  $N_s$  is the sample number. The resulted signal from the hidden layer to the  $i$ th output node can be expressed in the form of.

$$\phi_i(T, W, [x]) = \sum_{h=1}^{N_h} T_h g_h(W_h, [x]) + T_o$$
(4)

where  $g_h(w_h, [x])$  is the basis function , $T_h$  is its weight and  $T_o$  is a threshold , $[x]$  is the input vector. In our network the output is obtained by carrying out the similar activation function process to (4) by each output node. Sigmoid type of nonlinearity with  $W_{hi}$  weights of the connections from the input units to the hidden units are utilized :

$$g_h(x) = \frac{1}{1 + \exp\left(-\sum_i W_{hi} x_i - W_{ho}\right)}$$
(5)

where  $W_{ho}$  are the thresholds of the hidden units. Furthermore a tangent hyperbolic nonlinearity and radial basis function are also utilized as the hidden unit activation function.

Learning can be defined as a function approximation problem The function we use for approximation is of the following form

$$F(P, [x]) = [\phi_i^k] \quad i = 1, 2, 3, \dots, 12 ; \quad k = 1, \dots, N_s$$
(6)

where  $\phi_i^k$  corresponds to either a signal or noise parameter at the related frequency , $P$  is the connection weights of the network Learning process corresponds to the computation of  $P$  values to minimize the error between  $y_i$  - the measured value - and  $F(P, [x])$  over all training example pairs.  $\{[x_i], y_i\}$  using a distance measure , e.g., sum





of square errors:

$$E(P) = \sum_{\{[x_i], y_i\}} [y_i - F(P, [x])]^2 \tag{7}$$

So we start with any set of weights and repeatedly change each weight by an amount proportional to  $\frac{\partial E}{\partial P_p}$ :

$$\Delta P_p = -\eta \frac{\partial E}{\partial P_p} \tag{8}$$

where  $\eta$  is the stepsize in descent. We assume that the training is completed when error fails to decrease any further in which case we take the best so far.

To evaluate the quality of the fit to measured data the following error terms are found to be convenient: where  $S_{ij}$  and  $N_i$  are, respectively the signal and noise parameters, and  $n$  is the number of discrete frequencies

$$E_{S_{ij}} = \frac{1}{n} \sum_{k=1}^n \frac{|S_{\hat{u}_{meas.}}^k - S_{\hat{u}_{meas.}}^k|}{|S_{\hat{u}_{meas.}}^k|} \tag{9.1}$$

$$E_{N_i} = \frac{1}{n} \sum_{i=1}^n \frac{|N_{i_{meas.}}^k - N_{i_{predict.}}^k|}{|N_{i_{meas.}}^k|} \tag{9.2}$$

used. Total average error can be defined as the average of the signal and noise errors:

$$E_T = \frac{1}{4} \sum_{i=1}^4 E_{i_{signal}} + \frac{1}{3} \sum_{i=1}^3 E_{i_{noise}} \tag{9.3}$$

Table 1  
Manufacturer's Values for Signal and noise parameters for N72000a FET

FILE NAME: N72000A.S2P VER.4.0  
 ! NEC PART NUMBER: NE72000 DATE: 03/91  
 ! BIAS CONDITIONS: VDS=3V, IDS=10mA  
 ! NOTE: S-PARAMETERS INCLUDE BOND WIRES.  
 ! NOTE: NOISE PARAMETERS INCLUDE BOND WIRES

#	S11	S21	S12	S22	Fmin.	Gamma opt	N/50
#GHZ	mag. ang.	mag. ang.	mag. ang.	mag. ang.	mag. ang.	mag. ang.	mag. ang.
Z							
0.1	.999 -2	2.943 178	.004 84	.735 -1			
0.2	.999 -4	2.964 176	.008 86	.736 -2			
0.5	.993 -10	2.977 171	.021 83	.736 -6			
1.0	.982 -21	2.935 162	.042 77	.728 -12	0.55	.87 15	.65
1.5	.979 -31	2.971 155	.060 71	.717 -17			
2.0	.961 -41	2.907 148	.077 65	.703 -23	0.60	.78 36	.50
3.0	.923 -59	2.717 133	.107 55	.673 -33	0.87	.71 53	.48
4.0	.881 -75	2.505 119	.130 45	.640 -42	1.00	.66 64	.45
5.0	.839 -90	2.292 107	.147 36	.606 -49	1.20	.62 72	.43
6.0	.802 -102	2.089 96	.159 31	.587 -56	1.35	.59 83	.39
7.0	.776 -113	1.932 87	.167 24	.570 -62			
8.0	.744 -124	1.793 77	.171 18	.555 -69	1.70	.52 105	.38
9.0	.722 -132	1.669 70	.171 13	.545 -74			
10.0	.711 -141	1.579 62	.174 10	.545 -79	2.15	.45 128	.32
11.0	.691 -149	1.493 54	.176 5	.544 -85			
12.0	.677 -157	1.411 46	.171 1	.545 -89	2.50	.41 147	.27
13.0	.671 -165	1.333 38	.167 -2	.550 -94			
14.0	.664 -172	1.263 32	.163 -4	.554 -98	2.95	.38 166	.32
15.0	.657 -179	1.212 26	.159 -5	.564 -100			
16.0	.651 175	1.154 21	.159 -5	.568 -104			
17.0	.648 169	1.108 15	.152 -5	.576 -107			

The outputs of the NEURAL NETWORK MODEL for N72000A FET is given in table 1 for the sigmoid type of activation function, over the band from 1 GHz to 14 GHz over which both of signal and noise parameters are simultaneously present in catalog file. The errors defined each signal and noise parameter by (9.1),(9.2) and the total average error (9.3) are also added to the end of the Table 1. Finally, graphs of the network outputs with respect to the frequency will also be given which show quite good simulation of the signal and noise parameters over the fairly large operational bandwidth.

Table 2 Output of the NEURAL NETWORK MODEL for N72000A FET

Freq. GHz	Predicted S Parameters [Bias=7.714]								Measured S Parameters For Learning							
	S11	S21	S12	S22	S11	S21	S12	S22	S11	S21	S12	S22	S11	S21	S12	S22
1.0	0.984	-21.2	2.937	162.0	0.042	77.0	0.728	-12.7	0.982	-21.0	2.935	162.0	0.042	77.0	0.728	-12.0
2.0	0.962	-40.4	2.903	148.2	0.077	65.2	0.704	-22.1	0.961	-41.0	2.907	148.0	0.077	65.0	0.703	-23.0
3.0	0.919	-59.9	2.729	132.8	0.107	54.6	0.671	-33.3	0.923	-59.0	2.717	133.0	0.107	55.0	0.673	-33.0
4.0	0.880	-74.9	2.506	119.4	0.129	45.3	0.640	-41.7	0.881	-75.0	2.505	119.0	0.130	45.0	0.640	-42.0
5.0	0.842	-88.9	2.280	106.9	0.147	37.0	0.611	-49.1	0.839	-90.0	2.292	107.0	0.147	36.0	0.606	-49.0
6.0	0.804	-102.2	2.087	96.0	0.160	30.0	0.585	-56.3	0.802	-102.0	2.089	96.0	0.159	31.0	0.587	-56.0
8.0	0.745	-123.5	1.810	78.2	0.171	18.5	0.554	-68.2	0.744	-124.0	1.793	77.0	0.171	18.0	0.555	-69.0
10.0	0.706	-141.9	1.578	60.2	0.173	8.5	0.546	-79.4	0.711	-141.0	1.579	62.0	0.174	10.0	0.545	-79.0
12.0	0.680	-156.4	1.410	47.2	0.172	4.9	0.545	-88.8	0.677	-157.0	1.411	46.0	0.171	1.0	0.545	-89.0
14.0	0.664	-172.0	1.264	32.0	0.163	-5.9	0.554	-98.0	0.664	-172.0	1.263	32.0	0.163	-4.0	0.554	-98.0

Freq. GHz	Predicted Noise Parameters			Measured Noise Parameters				
	Fmin	ropt	Rn/50	Fmin	ropt	Rn/50		
1.0	0.540	0.870	16.3	0.650	0.550	0.870	15.0	0.650
2.0	0.600	0.780	36.0	0.500	0.623	0.780	35.4	0.503
3.0	0.835	0.710	51.8	0.471	0.870	0.710	53.0	0.480
4.0	1.000	0.660	64.0	0.450	1.020	0.663	63.2	0.452
5.0	1.193	0.622	73.8	0.430	1.200	0.620	72.0	0.430
6.0	1.350	0.590	83.0	0.390	1.364	0.584	84.2	0.408
8.0	1.696	0.521	103.6	0.367	1.696	0.521	103.6	0.367
10.0	2.124	0.457	126.8	0.316	2.150	0.450	128.0	0.320
12.0	2.500	0.410	147.0	0.270	2.511	0.406	147.7	0.276
14.0	2.948	0.380	165.9	0.320	2.950	0.380	166.0	0.320

ES11=0.009953 ES21=0.010105 ES12=0.019817 ES22=0.008045 E<sub>t</sub>= 0.011980  
EN1=0.015300 EN2=0.019178 EN3=0.014686 F<sub>t</sub>= 0.016388  
E<sub>T</sub>=0.0133306 (Total average error)

In the Table 3, the measured values for the S and noise parameters of FET N24200A are given from its catalog file data. The outputs of the NEURAL NETWORK MODEL of N24200A are given in Table 4, over the band from 1 GHz to 30 GHz over which both of the signal and noise parameters are simultaneously present in catalog file. The errors defined each signal and noise parameter by (9.1),(9.2) and the total average error (9.3) are also added to the end of the Table 4.

For each case of the simulation, optimum number for sample and iteration used has been searched against the error. To give an example for their variations the transistor N32684A was chosen due to its sufficient number of supplied signal and noise parameters. The variations are shown in Fig 4a and Fig 4b.

The transistor N24200A was chosen to demonstrate the capacity of the neural network model for wide operational range of frequency for both signal and noise parameters (1-30 GHz). The simulation results of N24200A transistor are also given in Fig.5 a-d, Fig 6 a-d and Fig.7 a-d which show quite good agreement of the signal and noise parameters over the fairly large operational bandwidth.

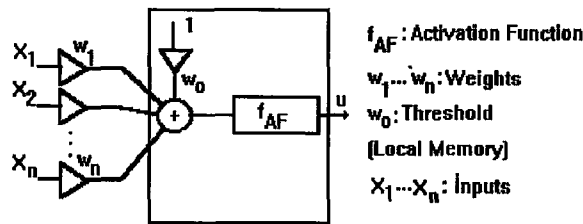


Fig.3 A perceptron node

Table 3  
 Manufacturer's Values for Signal and Noise parameters for N24200a FET

```

! FILENAME:          N24200A.S2P          VERSION: 5.0
! NEC PART NUMBERS: NE24200             DATE:6/91
! BIAS CONDITIONS:  VDS=2V, IDS=10mA
! NOTE:      S-PARAMETERS INCLUDES BOND WIRES.
! GATE:      TOTAL 2 WIRES, 1 PER BOND PAD, EACH WIRE 0.0132"(335um) LONG.
! DRAIN:     TOTAL 2 WIRES, 1 PER BOND PAD, EACH WIRE 0.0094"(240um) LONG.
! SOURCE:    TOTAL 4 WIRES, 2 PER SIDE,   EACH WIRE 0.0070"(178um) LONG.
! WIRE:      0.0007"(17.8um) DIAMETER, GOLD
! NOTE:      NOISE PARAMETERS FOR 28 & 30 GHZ ARE EXTRAPOLATED, NOT MEASURED.
#
#                               NOISE PARAMETERS
#          S11          S21          S12          S22          Fmin          aopt          RN/50
#GHZ mag.  ang.  mag.  ang. mag.  ang. mag.  ang.          mag.  ang.
Z-----
0.1  .999   -1   5.04 179 .002  89 .62   -1
0.2  .999   -3   5.02 178 .004  89 .62   -1
0.5  .999   -6   4.97 175 .008  87 .62   -4
1.0  .997  -12   4.88 170 .016  84 .62   -8   0.30   0.81   10   0.39
2.0  .990  -23   4.70 161 .030  77 .61  -15   0.31   0.79   17   0.36
3.0  .980  -34   4.54 152 .042  71 .61  -22
4.0  .970  -44   4.38 144 .052  65 .61  -29   0.33   0.75   31   0.33
5.0  .950  -53   4.22 136 .062  59 .60  -36
6.0  .930  -62   4.08 128 .071  53 .59  -41   0.38   0.72   45   0.30
7.0  .910  -71   3.93 120 .079  48 .59  -46
8.0  .890  -79   3.80 113 .086  43 .58  -51   0.43   0.70   59   0.27
9.0  .870  -87   3.67 106 .092  38 .57  -56
10.0 .860  -94   3.54  99 .099  34 .56  -61   0.50   0.68   77   0.24
11.0 .840 -102   3.42  92 .104  30 .55  -65
12.0 .820 -108   3.30  86 .109  27 .54  -70   0.60   0.66   92   0.22
13.0 .800 -115   3.19  80 .114  24 .53  -74
14.0 .790 -121   3.08  74 .119  21 .51  -78   0.71   0.64  108   0.19
15.0 .770 -128   2.97  68 .123  18 .50  -83
16.0 .750 -134   2.87  63 .127  16 .49  -87   0.85   0.62  126   0.18
17.0 .740 -139   2.77  57 .131  14 .48  -91
18.0 .720 -145   2.68  52 .135  12 .47  -95   1.00   0.58  140   0.15
19.0 .710 -150   2.59  47 .138  10 .46  -98
20.0 .690 -155   2.50  42 .142  8  .45 -102   1.20   0.55  153   0.13
22.0 .660 -165   2.32  32 .148  6  .43 -109   1.50   0.52  164   0.11
24.0 .640 -175   2.16  23 .153  4  .42 -116   1.80   0.49  175   0.10
26.0 .610  177   2.01  15 .159  3  .41 -122   2.10   0.48 -176   0.08
28.0 .590  168   1.87  7  .163  1  .41 -128   2.40   0.46 -168   0.07
30.0 .570  160   1.73 -1  .168  0  .41 -134   2.80   0.46 -160   0.05
    
```

Table 4  
 Outputs of NEURAL NETWORK MODEL FOR N24200A FET

**SIGNAL AND NOISE PARAMETERS FOR LEARNING**

f[GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>	F <sub>min</sub>	Γ <sub>opt</sub>	R <sub>N</sub> /50					
1.0	0.997	-12.000	4.880	170.000	0.016	84.000	0.620	-8.000	0.300	0.810	10.000	0.390
2.0	0.990	-23.000	4.700	161.000	0.030	77.000	0.610	-15.000	0.310	0.790	17.000	0.360
4.0	0.970	-44.000	4.380	144.000	0.052	65.000	0.610	-29.000	0.330	0.750	31.000	0.330
6.0	0.930	-62.000	4.080	128.000	0.071	53.000	0.590	-41.000	0.380	0.720	45.000	0.300
8.0	0.890	-79.000	3.800	113.000	0.086	43.000	0.580	-51.000	0.430	0.700	59.000	0.270
10.0	0.860	-94.000	3.540	99.000	0.099	34.000	0.560	-61.000	0.500	0.680	77.000	0.240
12.0	0.820	-108.000	3.300	86.000	0.109	27.000	0.540	-70.000	0.600	0.660	92.000	0.220
14.0	0.790	-121.000	3.080	74.000	0.119	21.000	0.510	-78.000	0.710	0.640	108.000	0.190
16.0	0.750	-134.000	2.870	63.000	0.127	16.000	0.490	-87.000	0.850	0.620	126.000	0.180
18.0	0.720	-145.000	2.680	52.000	0.135	12.000	0.470	-95.000	1.000	0.580	140.000	0.150
20.0	0.690	-155.000	2.500	42.000	0.142	8.000	0.450	-102.000	1.200	0.550	153.000	0.130
22.0	0.660	-165.000	2.320	32.000	0.148	6.000	0.430	-109.000	1.500	0.520	164.000	0.110
24.0	0.640	-175.000	2.160	23.000	0.153	4.000	0.420	-116.000	1.800	0.490	175.000	0.100
26.0	0.610	177.000	2.010	15.000	0.159	3.000	0.410	-122.000	2.100	0.480	-176.000	0.080
28.0	0.590	168.000	1.870	7.000	0.163	1.000	0.410	-128.000	2.400	0.460	-168.000	0.070
30.0	0.570	160.000	1.730	-1.000	0.168	0.000	0.410	-134.000	2.800	0.460	-160.000	0.050

Table 4  
 OUTPUTS OF NEURAL NETWORK MODEL FOR N24200A(Continued)

PREDICTED SIGNAL AND NOISE PARAMETERS												
1.000	0.992	-13.829	4.875	169.717	0.017	83.465	0.617	-9.134	0.297	0.812	10.917	0.386
2.000	0.985	-22.417	4.707	161.419	0.028	77.715	0.614	-14.916	0.305	0.786	16.516	0.365
4.000	0.959	-44.373	4.382	144.512	0.053	65.511	0.606	-29.678	0.333	0.745	30.971	0.328
6.000	0.931	-60.962	4.080	127.820	0.069	53.158	0.593	-40.074	0.380	0.724	44.336	0.297
8.000	0.895	-78.936	3.781	111.545	0.086	42.050	0.577	-51.161	0.441	0.703	60.658	0.267
10.000	0.857	-94.854	3.533	98.415	0.099	33.606	0.560	-61.167	0.508	0.683	76.692	0.242
12.000	0.822	-108.12	3.314	86.946	0.109	26.602	0.539	-69.757	0.590	0.661	91.846	0.219
14.000	0.788	-120.82	3.092	75.322	0.118	20.354	0.514	-78.232	0.701	0.635	107.991	0.195
16.000	0.752	-133.38	2.872	63.605	0.127	15.329	0.488	-86.831	0.844	0.607	124.823	0.172
18.000	0.719	-144.84	2.675	52.244	0.134	11.762	0.465	-94.819	1.012	0.579	140.145	0.151
20.000	0.689	-155.12	2.497	40.936	0.141	9.238	0.448	-102.017	1.214	0.553	153.077	0.132
22.000	0.662	-165.10	2.322	29.859	0.147	7.275	0.434	-109.009	1.475	0.526	164.537	0.113
24.000	0.635	-175.07	2.149	20.548	0.153	5.714	0.422	-116.052	1.795	0.499	175.182	0.095
26.000	0.610	-175.998	2.000	14.310	0.158	4.593	0.412	-122.471	2.119	0.474	-175.64	0.081
28.000	0.591	168.947	1.885	15.288	0.162	3.659	0.405	-127.601	2.391	0.455	-168.58	0.070
30.000	0.570	159.9	1.729	-6.5	0.168	1.0	0.410	-134.0	2.801	0.460	-160.0	0.05

#### Error Analysis

$E_{11}=0.011053$   $E_{21}=0.024918$   $E_{12}=0.027162$   $E_{22}=0.009100$

$E_t = 0.018058$

$F_1=0.011198$   $F_2=0.012857$   $F_3=0.016921$

$F_t = 0.013659$

#### References

1. FROELICH,R.K.: "An improved model for noise characterization of microwave GaAs FET's" IEEE Trans.,1990,MTT-38 (6),pp. 703-706.
2. POSPIESZALSKI,M.W.: "Modelling of noise parameters of MESFET's and MODFET's and frequency and temperature dependence",IEEE Trans.,1989,MTT-37,(9),pp1340-1350.
3. BERROTH,M. and BOSH R.: "Broad band determination of the FET small signal equivalent circuits ",IEEE Trans.1990,MTT-38,(7),pp891-895.
4. DAMBRINE,G.CAPPY,A.HELIDORE,F. AND PLAYEZ,E.: "A new method for determining the FET small-signal equivalent circuit",IEEE Trans.1988,MTT-36,(7),pp.1151-1159.
5. Z.R.HU,Z.M.YANG,V.F.FUSCO,J.A.C.STEWART,"Unified small-signal-noise model for active microwave device",IEE Proceedings-G, Vol.140,No1,Feb.1993 ,pp.55-60.

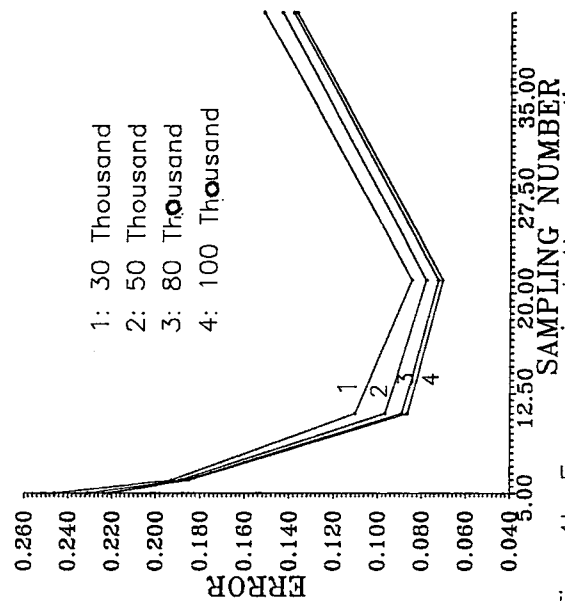


Fig. 4b Error against the sampling number

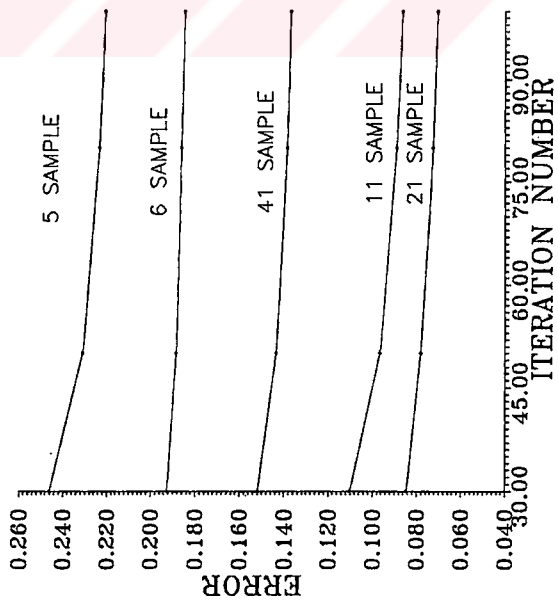


Fig 4a Error against the iteration number



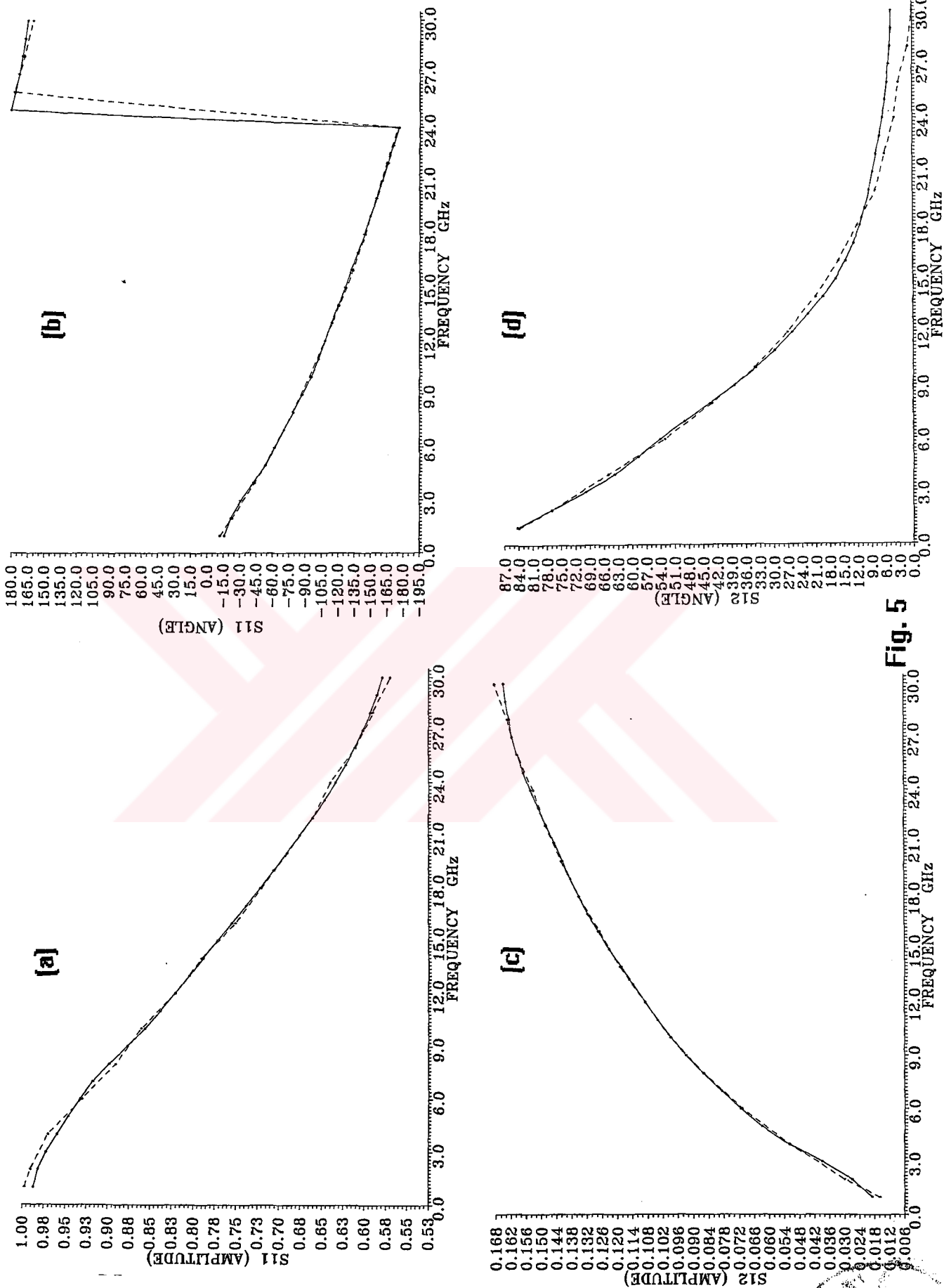


Fig. 5



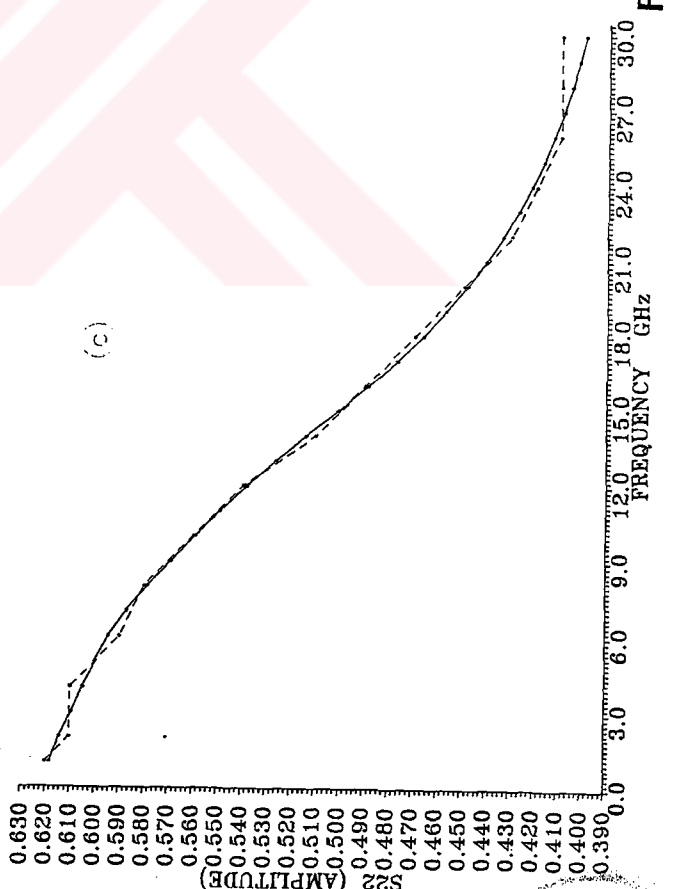
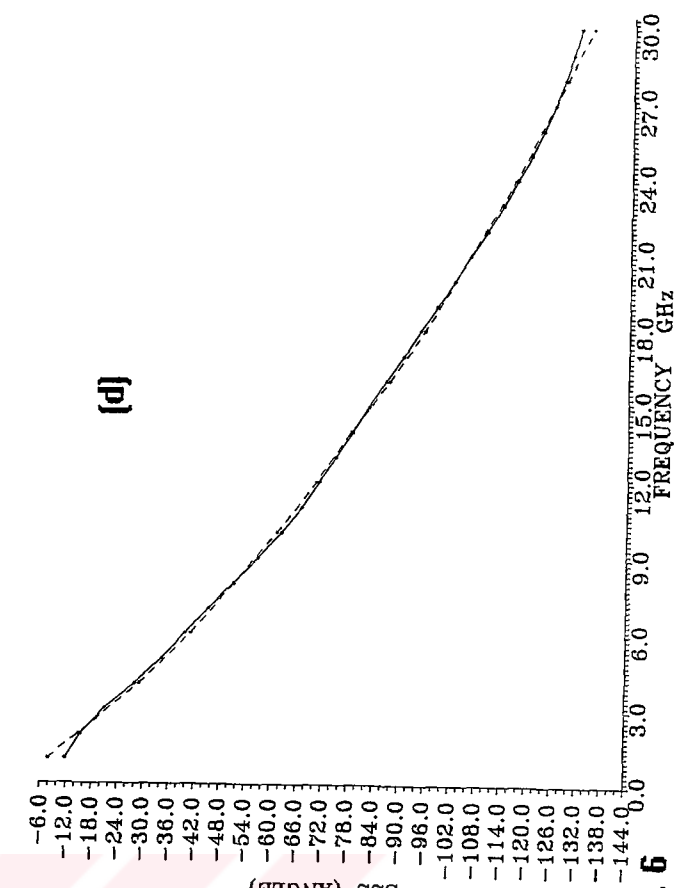
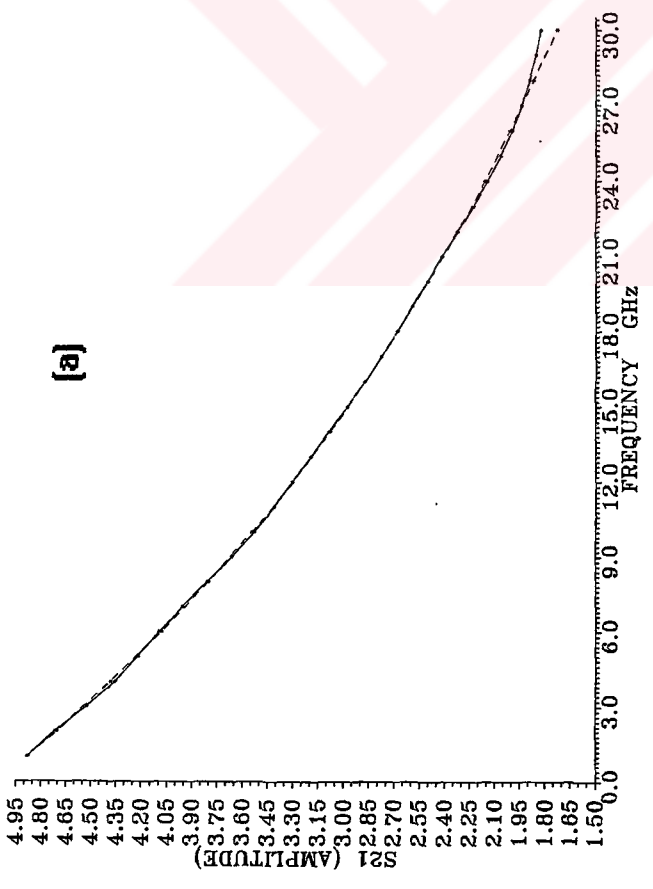
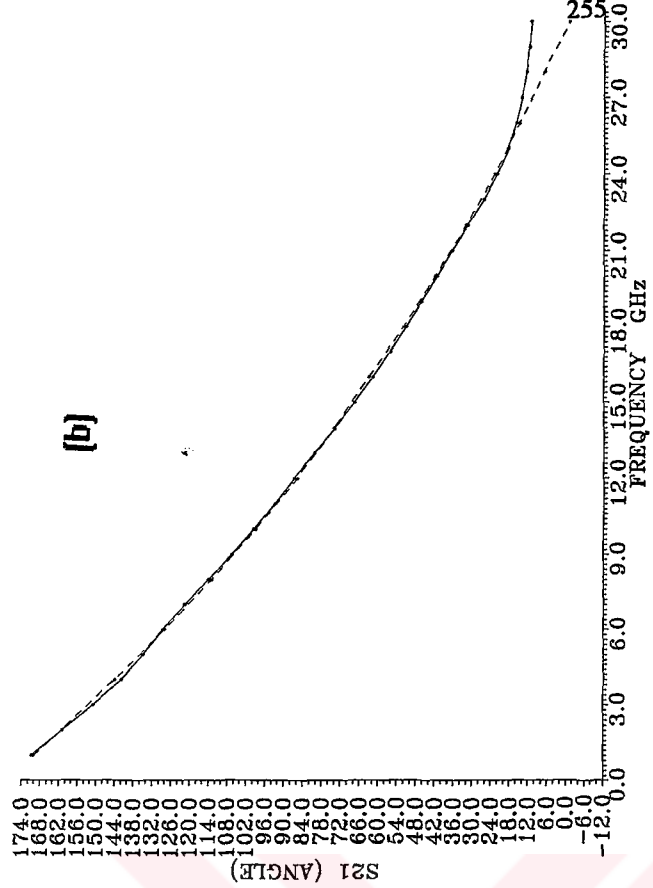


Fig. 6



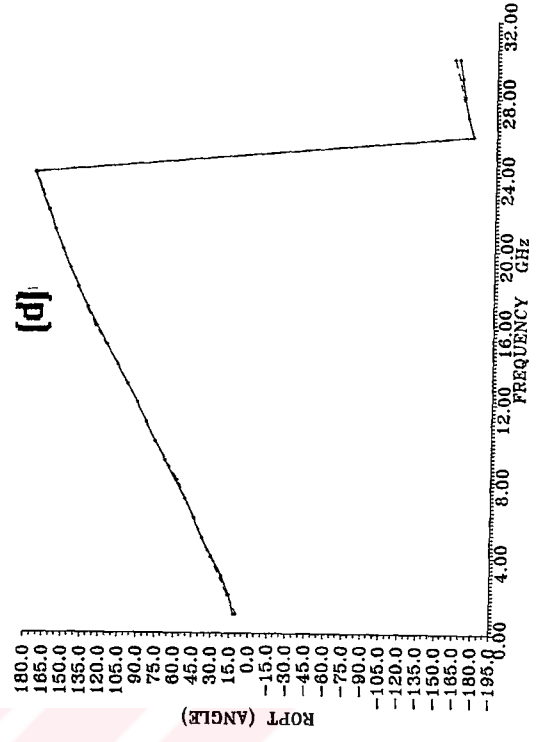
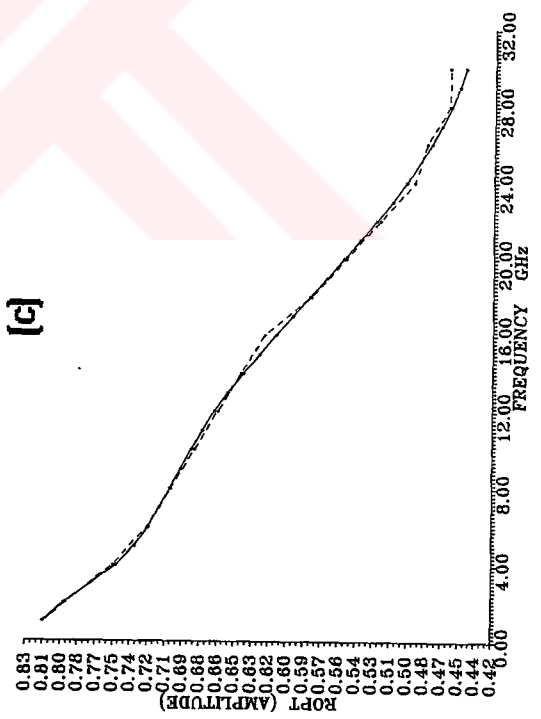
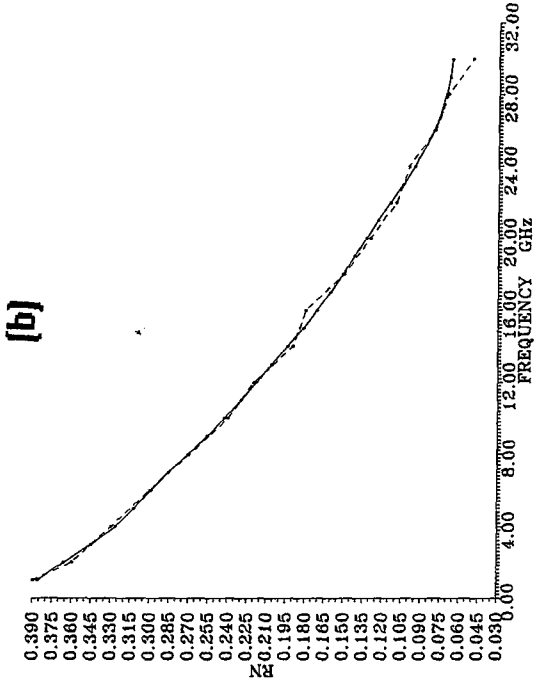
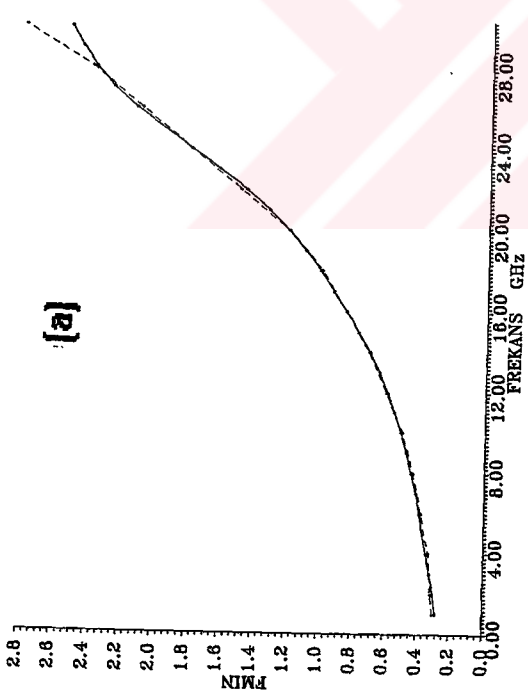


Fig. 7



## NEURAL NETWORK SIMULATION OF THE SIGNAL AND NOISE PARAMETERS

Filiz GÜNEŞ\* Fikret GÜRGEN\*\* Hamid TORPİ\*

\* Yıldız Technical University , Electronics & Communication Eng. Dept., 80670 MASLAK-  
ISTANBUL/TURKEY

\*\*Boğaziçi University, Computer Eng. Dept. ,BEBEK-ISTANBUL/TÜRKİYE

The small-signal and noise behaviours of a microwave transistor around bias point are usually determined by the scattering parameters  $S_{11}, S_{22}, S_{21}, S_{12}$  and the noise parameters  $F_{opt}, \Gamma_{opt}, R_N$  over its operating bandwidth. Both S-parameters and noise parameters are frequency dependent and intrinsic properties of a device. S-parameters are used to represent the device signal power gains and mismatch loss, whereas the noise figure describes the degradation of the signal-to-noise ratio between the input and output of a device. Since both sets of parameters are related through the physical and geometrical parameters of the device, in order that the equivalent can best predict both the small-signal and noise behaviours of the device simultaneously, its equivalent model should be extracted by both measured S-parameters and noise parameters. Published literature is either concerned with only the small-signal model or concentrates on the noise behaviour descriptions based on existing small-signal equivalent circuits that have nothing to do with the device noise characteristics. [1-4] A recent work gives an optimisation for the MESFET model parameters both S and noise parameters [5].

The main purposes of this work can be ordered as follows:

- (i) Establish a novel neural network of feedforward type with a single hidden layer,
- (ii) Using back-propagation and nonlinear types of activation functions, train the network for both the signal-noise behaviours over the operational bandwidth for any type of active device,
- (iii) Establish performance measure of the network
- (iv) predict the small-signal and noise behaviours at any operation frequency concurrently using the trained network which makes functional approximations of the device nonlinear characteristics in the vicinity of the chosen bias point.

At the conference in a worked example for particular active device an equivalent neural network with its performance measures will be discussed and the frequency dependencies of the measured and calculated signal and noise parameters will be given both numerically and graphically. Furthermore criteria to evaluate the quality of the fit to measured data will be defined and discussed.

### References

1. FROELICH, R.K.: "An improved model for noise characterization of microwave GaAs FET's" IEEE Trans., 1990, MTT-38 (6), pp. 703-706.
2. POSPIESZALSKI, M. W.: "Modelling of noise parameters of MESFET's and MODFET's and frequency and temperature dependence", IEEE Trans., 1989, MTT-37, (9), pp. 1340-1350.
3. BERROTH, M. and BOSCH R.: "Broad band determination of the FET small signal equivalent circuits", IEEE Trans. 1990, MTT-38, (7), pp. 891-895.
4. DAMBRINE, G. CAPPY, A. HELIDORE, F. AND PLAYEZ, E.: "A new method for determining the FET small-signal equivalent circuit", IEEE Trans. 1988, MTT-36, (7), pp. 1151-1159.
5. Z. R. HU, Z. M. YANG, V. F. FUSCO, J. A. C. STEWART, "Unified small-signal-noise model for active microwave device", IEE Proceedings-G, Vol. 140, No. 1, Feb. 1993, pp. 55-60.



## UNIFIED SMALL -SIGNAL - NOISE NEURAL NETWORK MODEL FOR ACTIVE MICROWAVE DEVICES

Filiz GÜNEŞ<sup>1</sup> , Fikret GÜRGEN<sup>2</sup> and Hamid TORPİ<sup>1</sup>

<sup>1</sup>Yıldız Technical University, Electronics & Communication Eng. Dept., MASLAK-ISTANBUL, TÜRKİYE

<sup>2</sup>Boğaziçi University, Computer Eng. Dept., BEBEK-ISTANBUL, TÜRKİYE

**Abstract** - A new method for modelling, concurrently the small-signal and the noise performance of active microwave devices is proposed. Here the device is modelled by a black box whose small signal and noise parameters are evaluated through a neural-network, based upon the fitting of both of these parameters over the operational bandwidth of the device. On using the concurrent modelling procedure suggested in this paper, it has been found that, not only can the small-signal performance be simulated accurately, but also the prediction of noise performance is in much better agreement with measurements than those of recent published models.

### I. INTRODUCTION

The small-signal and noise behaviours of a microwave transistor around bias point are usually determined by the scattering parameters  $S_{11}$ ,  $S_{22}$ ,  $S_{21}$ ,  $S_{12}$  and the noise parameters  $F_{opt}$ ,  $L_{opt}$ ,  $R_N$  over its operating bandwidth. Both S-parameters and noise parameters are frequency dependent and intrinsic properties of a device. S-parameters are used to represent the device signal power gains and mismatch loss whereas the noise figure describes the degradation of the signal-to-noise ratio between the input and output of a device.

As shown in the Fig. 1 noise model parameters also require some small-signal modelling information, because both sets of parameters are related through the physical and geometrical parameters of the device e.g. both are strongly influenced by the gate length of the active layer. In view of this, and in order that the equivalent circuit can best predict both the small-signal and noise behaviours of the device simultaneously, its equivalent model should be extracted by both measured S-parameters and noise parameters. Published literature is either concerned with only the small-signal model or concentrates on the noise behaviour descriptions based on existing small-signal equivalent circuits that have nothing to do with the device noise characteristics [1-3].

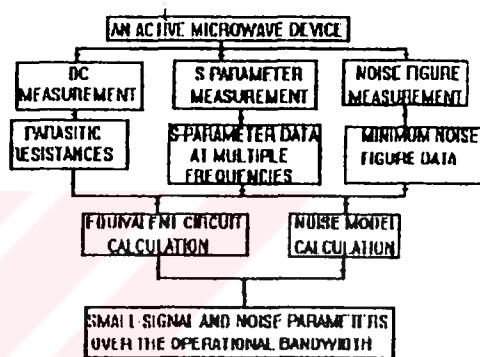


Fig. 1 General Procedure for Determination of Small-Signal and Noise Performance of an Active Device

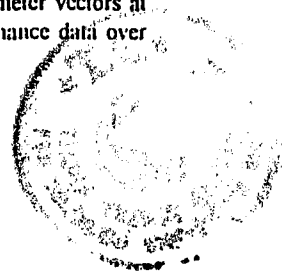
A recent work gives an optimisation for the MESFET model parameters both S and noise parameters [5].

The main purposes of this work can be ordered as follows:

- (i) Establish a novel neural network of feedforward type with a single hidden layer,
- (ii) using back-propagation and nonlinear types of activation functions, train the network for both the signal-noise behaviours over the operational bandwidth for any type of active device,
- (iii) establish performance measure of the network,
- (iv) predict the small-signal and noise behaviours at any operation frequency using the neural network which has already trained to make functional approximations of the device nonlinear characteristics in the vicinity of the chosen bias point.

### II. DETERMINATION OF SMALL-SIGNAL AND NOISE BEHAVIOURS OF AN ACTIVE MICROWAVE DEVICE

The signal and noise performance of an active microwave device around a bias point are usually given by the scattering [S] and noise [N] parameter vectors at the  $\omega$ -domain and the measured performance data over



the operational band can be arranged in a table-form function as follows:

$$\begin{bmatrix} f_1 & : & S^{(1)} & N^{(1)} \\ f_2 & : & S^{(2)} & N^{(2)} \\ \vdots & & \vdots & \vdots \\ f_N & : & S^{(N)} & N^{(N)} \end{bmatrix} \quad (1)$$

where  $S^{(1)}, N^{(1)}, \dots, S^{(N)}, N^{(N)}$  are respectively, the scattering and noise vectors at  $f_1, f_2, \dots, f_N$  sampling operating frequencies.  $S^{(N)}$  and  $N^{(N)}$  can be given as follows

$$\begin{aligned} [S^{(N)}]^{-1} &= \begin{bmatrix} |S_{11}^{(N)}| & \varphi_{11}^{(N)} & |S_{21}^{(N)}| & \varphi_{12}^{(N)} \\ |S_{21}^{(N)}| & \varphi_{21}^{(N)} & |S_{22}^{(N)}| & \varphi_{22}^{(N)} \end{bmatrix} \\ [N^{(N)}]^{-1} &= \begin{bmatrix} |c_1| & \varphi_{c1}^{(N)} \\ |c_2| & \varphi_{c2}^{(N)} \end{bmatrix} \end{aligned} \quad (2)$$

The functions defined by (1) and (2) are utilized for training the neural network model of device. Then performance vectors  $[S^{(k)}], [N^{(k)}]$  at a desired frequency  $f_k$  can be obtained from the network output by inputting the frequency  $f_k$ . If  $[S^{(k)}], [N^{(k)}]$  are unmeasured they are determined by the generalization process of the neural network, which can be considered as the ability of the network to give good outputs to inputs it has not been trained on.

Once the performance vectors  $[S^{(k)}]$  and  $[N^{(k)}]$  are determined, an active microwave device can be represented as a black box at the frequency  $f_k$ . Fig. 2, which can be characterized in the form of-frequency will be dropped hereafter

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \quad (3.1)$$

where  $c_1$  and  $c_2$  are the noise waves which are the time varying complex random variables characterized by a correlation matrix  $C_s$  given by

$$[C_s] = \begin{bmatrix} |c_1|^2 & \overline{c_1 c_2^*} \\ \overline{c_2 c_1^*} & |c_2|^2 \end{bmatrix} \quad (3.2)$$

where the overbar indicates time-averaging with an implicit assumption of ergodicity and jointly wide-sense stationary processes. The diagonal terms of  $C_s$  give the noise power deliverable to the terminations in a 1Hz bandwidth. The off-diagonal terms are correlation products. The noise wave correlation matrix  $C_s$  is Hermitian and its components are referred to as noise wave parameters which can be given in the terms of the

performance vectors  $[N^{(k)}]$  elements by the following expressions [5]

$$|c_1|^2 = kT_{opt} \left( |s_{11}|^2 - 1 \right) + \frac{kt |1 - s_{11} \Gamma_{opt}|^2}{|1 + \Gamma_{opt}|^2} \quad (3.3)$$

$$|c_2|^2 = |s_{21}|^2 (kT_{opt} + \frac{kt |1 - \Gamma_{opt}|^2}{|1 + \Gamma_{opt}|^2}) \quad (3.4)$$

$$\overline{c_1 c_2^*} = \frac{-s_{21}^* 1 - \Gamma_{opt}^* kt}{|1 + \Gamma_{opt}|^2} + \frac{s_{11}}{s_{21}} \overline{c_2} \quad (3.5)$$

where  $k$  is the Boltmann's constant,  $Z_0$  is the normalization impedance,  $kt$  is the normalized temperature energy and  $kt$  and  $T_{opt}$  are given, respectively by

$$kt = \frac{4kT_0 R_n}{Z_0}, \quad T_{opt} = (\Gamma_{opt} - 1)T_n \quad (3.6)$$

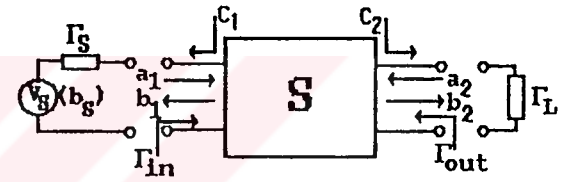


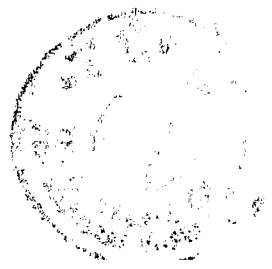
Fig. 2 Black box Representation of an Active Microwave Device

The transducer power gain of an active device is defined as the ratio of the power delivered to the load to the available power from source and can be expressed in terms of the scattering parameters  $S$  and  $\Gamma_s$  and  $\Gamma_L$  terminations in the following form [5]:

$$G_T \triangleq \frac{P_L}{P_{A,S}} = G_T(\Gamma_s, \Gamma_L, S) = \frac{(1 - |\Gamma_L|^2) |s_{21}|^2 (1 - |\Gamma_s|^2)}{|(1 - s_{22} \Gamma_L)(1 - s_{11} \Gamma_s)|} \quad (4.1)$$

$M_{in}$  and  $M_{out}$  conjugate mismatch loss respectively at the input and output ports of the active device can be given as [5]

$$M_{in} \triangleq \frac{P_{A,S}}{P_{in}} = M_{in}(\Gamma_s, \Gamma_L, S) = \frac{|1 - \Gamma_m \Gamma_s|^2}{(1 - |\Gamma_{in}|^2)(1 - |\Gamma_s|^2)} \quad (4.2)$$



$$M_{out} \Delta \frac{P_{Aout}}{P_i} = M_{out}(\Gamma_s, \Gamma_L, S) = \frac{|1 - \Gamma_L \Gamma_{out}|^2}{(1 - |\Gamma_L|^2)(1 - |\Gamma_{out}|^2)} \quad (4.3)$$

where

$$\Gamma_{out} = s_{22} + \frac{s_{12}s_{21}\Gamma_S}{1 - s_{22}\Gamma_S} \quad \Gamma_{in} = s_{11} + \frac{s_{12}s_{21}\Gamma_L}{1 - s_{22}\Gamma_L} \quad (4.4)$$

The noise figure of an active device is defined as the ratio of signal-to-noise ratios available at input and output and N vector describes the dependence of the transistor noise figure F on the input termination (source) reflection coefficient  $\Gamma_s$  through the relationship [5]

$$F(\Gamma_s, N) = F_{opt} + 4 \frac{R_N}{Z_0} \frac{|\Gamma_s - \Gamma_{opt}|^2}{(1 - |\Gamma_s|^2)(1 + |\Gamma_{opt}|^2)} \quad (5)$$

After having fixed S and N vectors of the device at any frequency,  $(\Gamma_s, \Gamma_L)$  termination couple can be determined by making compromise among the performance parameters  $G_T, F, M_{in}, M_{out}$  at any frequency or over an operational band, a recent typical application has been given in [6], where  $(\Gamma_s, \Gamma_L)$  couple is determined for the maximum stable gain  $G_T$  under the required F and  $M_{in}$  at any operation frequency.

### III. NEURAL NETWORK MODEL.

The multilayer perceptron (MLP) with single hidden layer having the same number of units as the output layer has been found to be sufficient to simulate an active microwave device and back-propagation (BP) algorithm is utilized to train this network [7], [8].

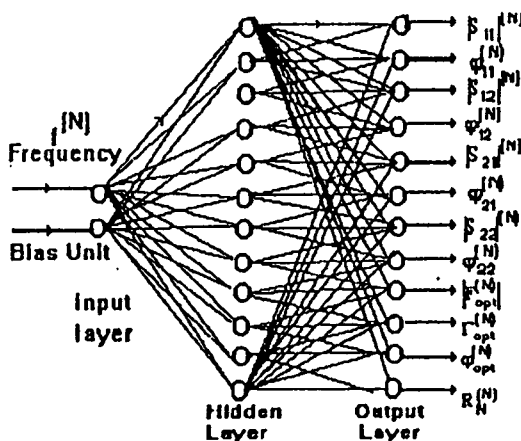


Fig. 3 MLP Simulator for an Active Microwave Device

An additive bias is utilized as the second network input to ensure faster convergence, which is taken as

$$\text{Bias} = \sqrt{\frac{1}{N_S} \sum_{i=1}^{N_S} \Gamma_i} \quad (6)$$

Where  $N_S$  is the sample number. The resulted signal from the hidden layer to the  $i$ th output node can be expressed in the form of

$$\phi_i(T, W, x) = \sum_{h=1}^{N_h} T_h g_h(W_h, x) + T_0 \quad (7)$$

where  $g_h(w_h, x)$  is the basis function,  $T_h$  is its weight and  $T_0$  is a threshold and  $x$  is the input vector. In our network the output is obtained by carrying out the similar activation function process to (7) by each output node. Sigmoid type of nonlinearity with  $W_{hi}$  weights of the connections from the input units to the hidden units is utilized due to its primary stability to the other types of nonlinear activation function e.g. tangent hyperbolic...

$$g_h(x) = \frac{1}{1 + \exp\left(-\sum_i W_{hi} x_i - W_{ho}\right)} \quad (8)$$

where  $W_{ho}$  are the thresholds of the hidden units so as the result,  $i$ th output can be expressed as  $F(P, x)$  where F is defined by the network architecture, i.e., the number of hidden layers, the number of hidden units, connectivity between layers, etc. P corresponds to the modifiable "weights" of the connections of this network.

In our application learning process corresponds to the computation of P values to minimize the error between  $y_i$  - the measured value - and  $F(P, x_i)$  over all training example pairs  $\{x_i, y_i\}$  using a distance measure, e.g., sum of square errors:

$$E(P) = \sum_{(x_i, y_i)} [y_i - F(P, x_i)]^2 \quad (9)$$

So we start with any set of weights and repeatedly change each weight by an amount proportional to  $\frac{\partial E}{\partial P_p}$ :

$$\Delta P_p = -\eta \frac{\partial E}{\partial P_p} \quad (10)$$

where  $\eta$  is the stepsize in descent. We assume that the training is completed when error fails to decrease any further in which case we take the best so far.



$$E_{sij} = \frac{1}{n} \sum_{k=1}^n \frac{|S_{ij}^k_{meas} - S_{ij}^k_{predict}|}{|S_{ij}^k_{meas}|} \quad (11.1)$$

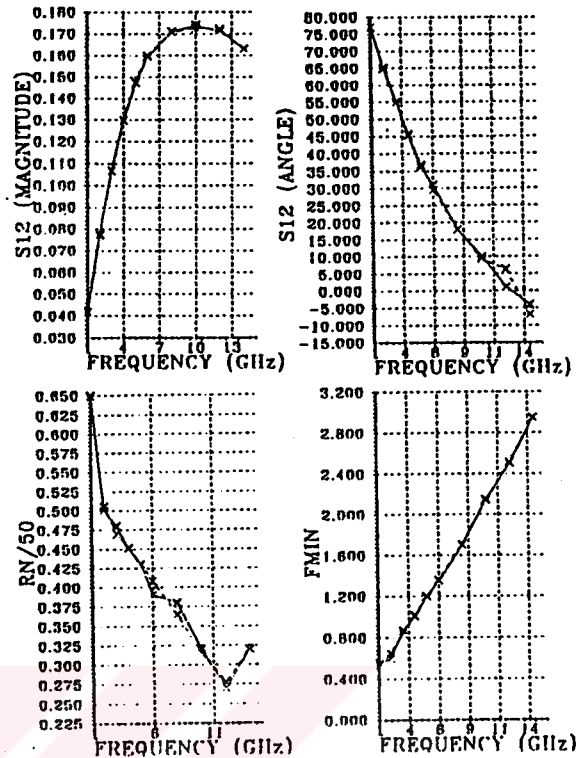
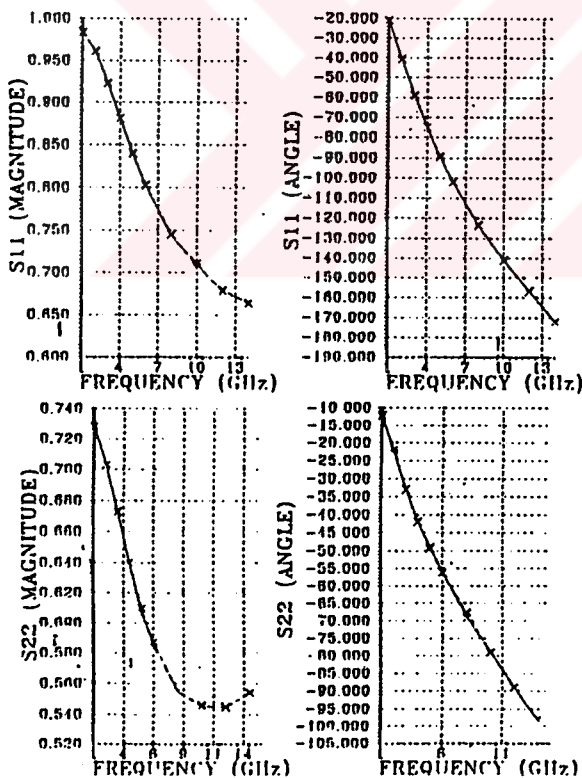
$$E_{Nij} = \frac{1}{n} \sum_{k=1}^n \frac{|N_{ij}^k_{meas} - N_{ij}^k_{predict}|}{|N_{ij}^k_{meas}|} \quad (11.2)$$

Where  $S_{ij}$  and  $N_{ij}$  are respectively the signal and noise parameters and  $n$  is the number of discrete frequencies used. Total average error can be defined as the average of the signal and noise errors:

$$E_{ij} = \frac{1}{4} \sum_{i=1}^4 E_{sij} + \frac{1}{3} \sum_{i=1}^3 E_{Nij} \quad (11.3)$$

Finally graphs of the network outputs with respect to the frequency are given in Figure-4 which show quite good simulation of the signal and noise parameters over the fairly large operational bandwidth

Figure 4  
Comparison between the predicted data and measured data for N72000A FET



Acknowledgement: We would like to thank TÜBİTAK supporting us with the projects EECAG-30 and EECAG-53.

REFERENCES

1. R.K. FROELICH: "An improved model for noise characterization of microwave GaAs FETs" *IEEE Trans.* 1990, MTT-38 (6), pp. 703-706.
2. M.W. POSPIESZALSKI: "Modelling of noise parameters of MESFETs and MODFETs and frequency and temperature dependence" *IEEE Trans.* 1989, MTT-37 (9), pp. 1340-1350.
3. M. BERROTTI and R. BOSI: "Broad band determination of the FET small signal equivalent circuits" *IEEE Trans.* 1990, MTT-38 (7), pp. 891-895.
4. Z. R. THUZ, M. YANG, V.F. FUSCO, J.A.C. STEWART: "Unified small-signal-noise model for active microwave device" *IEEE Proceedings-G*, Vol. 140, No.1, Feb. 1993, pp. 55-60.
5. H. VENDELIN, A.M. PAVIO, U.L. ROHDE: "Microwave Circuit Design using Linear and Nonlinear Techniques", John Wiley & Sons, 1990.
6. F. GÖNİŞ, M. GÖNİŞ, M. FİDAN: "Performance Characterisation of a Microwave Device" *IEE Proc. Circuits Devices Syst.*, Vol. 141, No.5, pp. 337-344, Oct. 1994.
7. F. GÜRGEN, R. ALPAYDIN, Ü. ÖLÇAKIN, E. ALPAYDIN: "Distributed and Local Neural Classifiers for Phoneme Recognition" *Pattern Recognition Letters*, pp. 1111-1118, Nov. 1994.
8. D.R. HUSLICH, G. HORNE: "Progress in Supervised Neural Networks" *IEEE Signal and Processing Magazine*, pp. 8-36, Jan. 1993.



# MULTI-BIAS AND CONFIGURATION NEURAL NETWORK MODELS FOR MICROWAVE ACTIVE DEVICES

Filiz GÜNEŞ<sup>1</sup>Hamid TORPİ<sup>1</sup>Fikret GÜRGEN<sup>2</sup><sup>1</sup>*Yıldız Technical University, Electronics & Communication Eng. Dept., 80670 MASLAK-ISTANBUL/TÜRKİYE*<sup>2</sup>*Boğaziçi University, Computer Eng. Dept., 80815 BEBEK-ISTANBUL/TÜRKİYE*

## Abstract

Small-signal and noise behaviour of an active microwave device is modeled through the neural network approach for multiple bias/configurations. Here, the device is modelled by a black box whose small signal and noise parameters are evaluated through a neural network, based upon the fitting of both of these parameters for the multiple bias or configuration. The concurrent modelling procedure does not require to solve device physics equations repeatedly during optimization. Compared to the existing device modelling techniques, the proposed approach has the capability to make high-dimensional models for highly nonlinear devices.

can be divided in to two parts: Extrinsic circuits and intrinsic circuit. The intrinsic circuit characterize the active region under the gate (or base) whose parameters are functions of bias conditions and device technological parameters, whereas the extrinsic parameters depend, at least to a first approximation, only on the technological parameters. If an unified circuit for a MESFET is considered, the most important extrinsic parameters are the gate, source and drain inductances due to the bond wires and the gate, source and drain resistances. The four main intrinsic parameters are the input capacitance  $C_{GS}$ , the transconductance  $g_m$ , the output conductance  $g_d$  and the feedback capacitance  $C_{GD}$ . In addition, the electrical behaviour of the intrinsic device requires the introduction of two more parameters: The intrinsic resistance  $R_i$ , which can be related to distributed nature of the RC input network, and the delay  $\tau$ , which is introduced in the expression for the current generator and corresponds to the time needed for the carriers to travel under the gate.

Briefly due to the intrinsic and extrinsic device properties, both the signal and noise parameters are the functions of the bias conditions, frequency, configuration types. The way to approximate these functions in the literature so far is considered in the forthcoming subsection.

## Review the literature

The problem of approximating measured device parameters or device response has been formulated as an optimization problem with respect to the equivalent circuit of a proposed model. The traditional approach in modeling is almost entirely directed at achieving the best possible match between measured and calculated parameters. This approach has serious shortcomings in two frequently encountered cases. The first case is when the equivalent circuit parameters are not unique with responses selected and the second is when the nonideal effects are not modelled adequately, the latter causing an imperfect match even if small measurement errors are allowed for. In both cases, a family of solutions for circuit model parameters may exist which produce reasonable and similar matches between measured and calculated responses. Besides, published literature is concerned with the equivalent circuit for the single-bias which are only

## I INTRODUCTION

### Aim of the work

The aim of this work is to model a microwave transistor by a black-box for which small-signal and noise parameters are evaluated through a neural network, based upon the fitting of both of these parameters to the corresponding measured data over the whole operational range from DC to more than 10 GHz for multiple bias and various types of configuration. So the stages of the work can be ordered as follows:

- (i) Establish a novel neural network of feedforward type with a single hidden layer,
- (ii) using back-propagation and nonlinear types of activation functions, train the network for both the signal-noise behaviours over the operational bandwidth for multiple bias and multiple configuration of any type of microwave transistor.
- (iii) Establish performance measure of the model,
- (iv) Predict the small-signal and noise behaviours at any operation frequency around any bias condition of any type of configuration using the neural network which has already been trained to make functional approximations of the device nonlinear characteristics in the vicinities of the chosen bias points.

From the classical point of view, unified small-signal-noise equivalent circuit for a microwave transistor

either small-signal models or the noise behaviour descriptions based on existing signal equivalent circuits that have nothing to do with the device noise characteristics .In [1] and [2] these two behaviours are combined in an unified classical circuit model for only a single-bias.A recent work [3] combines the signal and noise parameters in a neural network model over the fairly large operation band at a single bias point.

**II. NEURAL NETWORK MODEL**

*Signal-Noise Behaviour of a Microwave Transistor*

S and N parameter data measured at the multiple bias conditions ( $V_{DS}, I_{DS}$ ) for the configuration types (0.2,0.5,0.8) is all used to train the neural network.The amount of data used in the training and iteration number are altogether optimised against the error The measured S and N parameter data around a bias point for a type of configuration can be arranged in a table-form function as follows:

$$\begin{bmatrix} f_1 & : & S^{(1)} & N^{(1)} \\ f_2 & : & S^{(2)} & N^{(2)} \\ \vdots & & \vdots & \vdots \\ f_N & : & S^{(N)} & N^{(N)} \end{bmatrix} \quad (1)$$

where  $S^{(1)}, N^{(1)}, \dots, S^{(N)}, N^{(N)}$  are respectively, the scattering and noise vectors at the  $f_1, \dots, f_N$  measurement frequencies, and  $S^{(N)}$  and  $N^{(N)}$  performance vectors can be given as follows:

$$\begin{bmatrix} S^{(N)} \end{bmatrix}^t = \begin{bmatrix} |S_{11}^{(N)} & \varphi_{11}^{(N)} & |S_{12}^{(N)} & \varphi_{12}^{(N)} & |S_{21}^{(N)} & \varphi_{21}^{(N)} & |S_{22}^{(N)} & \varphi_{22}^{(N)} \end{bmatrix}$$

$$\begin{bmatrix} N^{(N)} \end{bmatrix}^t = \begin{bmatrix} F_{opt}^{(N)} & |\Gamma_{opt}|^{(N)} & \varphi_{opt}^{(N)} & F_N^{(N)} \end{bmatrix} \quad (2)$$

After having completed the training process, the performance vectors  $S^{(K)}, N^{(K)}$  at a desired frequency  $f_k$  at the conditions ( $V_{CE}, I_C, CT$  or  $V_{DS}, I_{DS}, CT$ ) for any configuration type among the trained ones can be obtained from the network output by inputting the frequency  $f_k$  bias configuration type which is defined by the numbers (0.2,0.5,0.8).If  $S^{(K)}, N^{(K)}$  are unmeasured they are determined by the generalization process of the neural network, which can be considered as the ability of the network to give good outputs to inputs it has not been trained on. In our application, the signal-noise neural network can generalize the performance not only at a single operation frequency of the trained bias condition, at the same time in the whole operation band of the untrained bias condition. The first may be named as the single frequency generalization, while the latter is called whole band generalization, worked examples of which

will be given in the result section.

*The multi-bias and configuration signal-noise neural network*

We use a novel neural network of feedforward type with a single hidden layer having the same number of nodes as the output layer. Let  $n, N_h$  and  $N_o$  be respectively the number of nodes in the input, hidden and output layers. In the signal-noise neural network  $n=4$  with the frequency, bias condition and the type of configuration  $CT, N_h=N_o=12$  which are the signal and noise vectors given by (2). (Fig.2)

The signal resulted from the hidden layer to the  $i$  th output node can be expressed in the form of

$$\Phi_i(x, T_i, W, \theta_h) = \sum_{h=1}^{N_h} T_{hi} g_h(x, W_h, \theta_h) \quad (3)$$

and the net output of the  $i$  th output node is obtained as follows

$$\phi_i(x, T_i, W, \Theta) = V_i f_i(\Phi_i + \Theta_i) \quad (4)$$

where  $g_h$  and  $f_i$  are the basis functions for the  $h$  th hidden node and the  $i$  th output node, respectively, which are sigmoid type of nonlinear functions in our case, e.g.  $g_h(W_h, x)$  can be expressed in the following form:

$$g_h(x, W_h, \theta_h) = \frac{1}{1 + \exp\left(-\sum_{i=1}^n x_i W_{ih} - \theta_h\right)} \quad (5)$$

In equations 3-5,  $x$  is the input vector of  $n$  dimensions:

$$x = [x_1, x_2, \dots, x_n]^t \quad (6)$$

$T_i$  is the weighting vector between the  $i$  th output node and the hidden layer:

$$T_i = [T_{i1} \ T_{i2} \ T_{i3} \ T_{ih} \ T_{iN_h}]^t \quad (7)$$

$W$  is the weighting matrix between the hidden and input layer:

$$W = [W_1, W_2 \ \dots \ W_h \ \dots \ W_{N_h}]^t \quad (8)$$

where  $W_h$  vector is the weight(ing) vector between the input layer and the  $h$  th hidden node and can be given by:

$$W_h = [W_{1h}, W_{2h}, \dots, W_{nh}]^t \quad (9.a)$$

$\theta_h, \Theta_i$  are the local memories belonging to the  $h$  th hidden and  $i$ .th output nodes, respectively. In the eqn: (4)

$V$  is the weighting factor of the output layer:

$$V = [V_1, V_2, \dots, V_{N_o}]^t \quad (9b)$$

**Determination of the network parameter matrix P**

If parameters of the network architecture is denoted **P**, the network parameter matrix **P** will have  $N_h \times N_i + N_h \times N_o + N_h + N_o$  elements which consist of weighting factors between the input and hidden layers and the hidden and output layers, the local memories of the hidden and output nodes. The training process can be defined as computation of the network parameter matrix **P** so that the error function which is

$$E(\mathbf{P}) = \sum_{k=1}^{N_S} E^{(k)} = \sum_{k=1}^{N_S} \left[ \frac{1}{2} \sum_{j=1}^{N_o} (y_j^{(k)} - \phi_j^{(k)})^2 \right] \quad (10)$$

is minimised, where  $y_j^{(k)}$  and  $\phi_j^{(k)}$  are respectively, the measured and predicted values of the  $j$  th output at the training frequency  $f_k$ . This training process is also called backpropagation and it is an 'on-line' process whose update equations for  $T_{hj}, W_{ih}, \theta_h$  can be given as follows:

$$T_{hj}^{(k+1)} = T_{hj}^{(k)} - \eta \frac{\partial E^{(k)}}{\partial T_{hj}} + \alpha (T_{hj}^{(k)} - T_{hj}^{(k-1)}) \quad (11.1)$$

$$W_{ih}^{(k+1)} = W_{ih}^{(k)} - \eta \frac{\partial E^{(k)}}{\partial W_{ih}} + \alpha (W_{ih}^{(k)} - W_{ih}^{(k-1)}) \quad (11.2)$$

$$\theta_h^{(k+1)} = \theta_h^{(k)} - \eta \frac{\partial E^{(k)}}{\partial \theta_h} + \alpha (\theta_h^{(k)} - \theta_h^{(k-1)}) \quad (11.3)$$

and the similar equations can be written for  $\Theta_i$  and  $V_i$ . In (11.1)-(11.3)  $\eta$  and  $\alpha$  are positive-valued learning rate and momentum, respectively. Thus we start any set the network parameters and then repeatedly change each parameter by an amount proportional to the terms  $\frac{\partial E^{(k)}}{\partial T_{hj}}$ ,

$\frac{\partial E^{(k)}}{\partial W_{ih}}$ ,  $\frac{\partial E^{(k)}}{\partial \theta_h}$  according to the eqns. (11.1-11.3) and assume that the training is completed when the error fails to decrease any further. In this case we take the best so far.

The sensitivity through the neural network with respect to  $T_{hj}, W_{ih}, \theta_h$  can be given as follows

$$\frac{\partial E^{(k)}}{\partial T_{hj}} = \frac{\partial \left[ \frac{1}{2} \sum_{j=1}^{N_o} (y_j^{(k)} - \phi_j^{(k)})^2 \right]}{\partial T_{hj}} = -V_j (y_j^{(k)} - \phi_j^{(k)}) f_j (1 - f_j) g_h$$

$$\frac{\partial E^{(k)}}{\partial T_{hj}} = \delta_j^{(3)} g_h \quad (12.1)$$

$$\frac{\partial E^{(k)}}{\partial W_{ih}} = \sum_{j=1}^{N_o} \frac{\partial E^{(k)}}{\partial \phi_j} \frac{d\phi_j}{df_j} \frac{df_j}{d\Phi_j} \frac{\partial \Phi_j}{\partial W_{ih}} = \sum_{j=1}^{N_o} -\delta_j^{(3)} \frac{\partial \Phi_j}{\partial W_{ih}} \quad (12.2)$$

$$\frac{\partial E^{(k)}}{\partial W_{ih}} = g_h (1 - g_h) \sum_{j=1}^{N_o} -\delta_j^{(3)} T_{hj} x_{ki} = \delta_h^{(2)} x_{ki}$$

$$\frac{\partial E^{(k)}}{\partial \theta_h} = \sum_{j=1}^{N_o} \frac{\partial E^{(k)}}{\partial \phi_j} \frac{d\phi_j}{df_j} \frac{\partial \Phi_j}{\partial \theta_h} \quad (12.3)$$

$$\frac{\partial E^{(k)}}{\partial \theta_h} = g_h (1 - g_h) \sum_{j=1}^{N_o} -\delta_j^{(3)} T_{hj} = \delta_h^{(2)}$$

$$\frac{\partial E^{(k)}}{\partial V_j} = \frac{\partial E^{(k)}}{\partial \phi_j^{(k)}} \frac{\partial \phi_j^{(k)}}{\partial V_j} = -(y_j^{(k)} - \phi_j^{(k)}) f_j (\Phi_j + \Theta_j)$$

$$\frac{\partial E^{(k)}}{\partial V_j} = \delta_j^{(3)} (1 - f_j)^{-1} \quad (12.4)$$

$$\frac{\partial E^{(k)}}{\partial \Theta_j} = -(y_j^{(k)} - \phi_j^{(k)}) f_j (1 - f_j) = \delta_j^{(3)} V_j^{-1} \quad (12.5)$$

where  $\delta_h^{(2)}$  and  $\delta_j^{(3)}$  represent local gradients at individual node in the second and third layer, respectively.

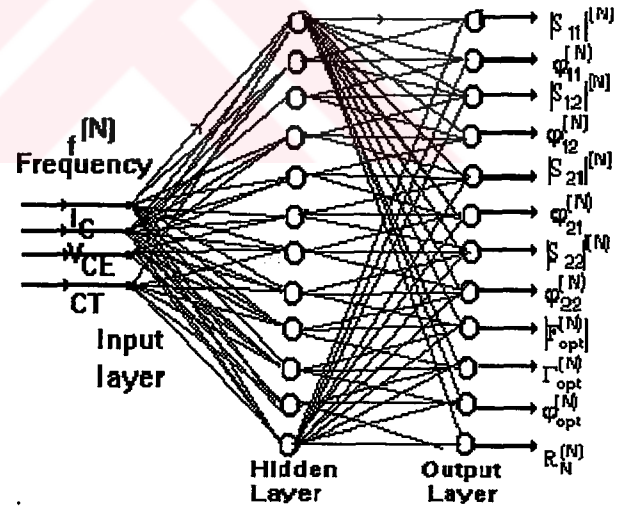


Fig. 1 Multi-bias and configuration signal-noise neural network

**III. PERFORMANCE MEASURE AND RESULTS**

To evaluate the quality of the fit to measured data the following error terms are found to be convenient





$$E_{S_{ij}} = \frac{1}{n} \sum_{k=1}^n \frac{|S_{ij,meas.}^k - S_{ij,meas.}^k|}{|S_{ij,meas.}^k|} \quad (13)$$

$$E_{N_i} = \frac{1}{n} \sum_{i=1}^n \frac{|N_{i,meas.}^k - N_{i,predict.}^k|}{|N_{i,meas.}^k|} \quad (14)$$

Where  $S_{ij}$  and  $N_i$  are ,respectively the signal and noise parameters , and  $n$  is the number of discrete frequencies used .Total average error can be defined as the average of the signal and noise errors:

$$E_T = \frac{1}{4} \sum_{i=1}^4 E_{i,signal} + \frac{1}{3} \sum_{i=1}^3 E_{i,noise} \quad (15)$$

Simulation results of NE219 transistor are given in Fig.4 a-f which shows variations of S parameters with frequency from 2 -6 GHz for the  $V_{CE}=8V$  and  $I_C=10,20$  and 30 mA at the common emitter configuration,the results for the same transistor,the same operation band and bias conditions but common collector configuration are obtained too.Distribution of errors with frequency for the whole band generalization at multiple bias points for the common collector configuration is given in Fig. 3. which show quite good agreement of the signal parameters over the operation bandwidth

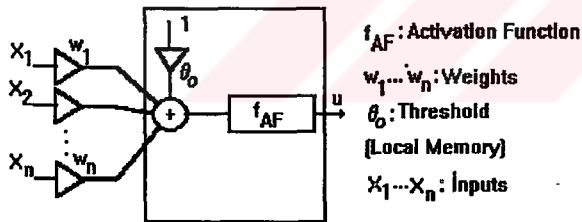


Fig. 2 A perceptron node

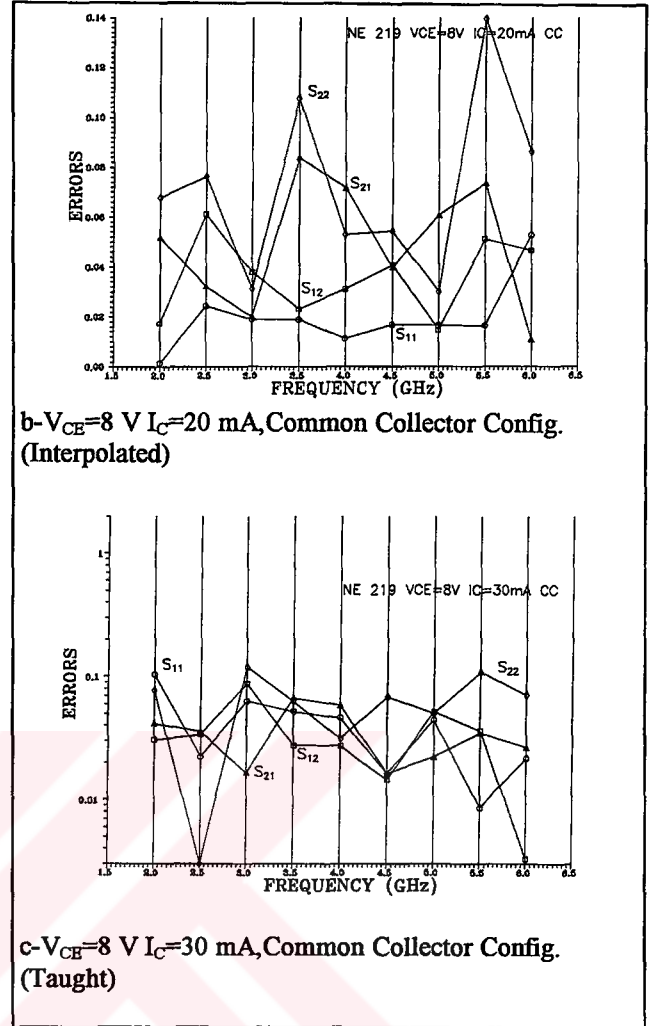
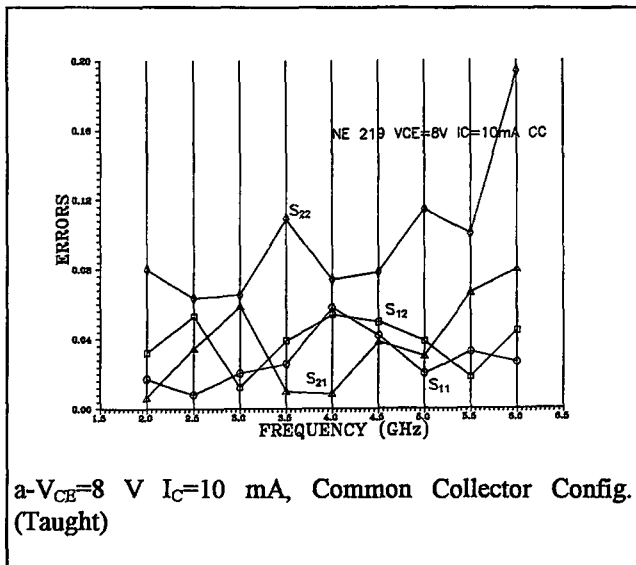


Fig.3 Distribution of interpolation errors with frequency for common collector configuration at multiple bias points for NE219 transistor.

Acknowledgement:This work was supported by the Yıldız Technical University Research Fund .Project number:92-A-04-03-12.

REFERENCES

[1].HU Z.R., YANG Z.M., FUSCO V.F., J.A.C.STEWART , "Unified small-signal-noise model for active microwave device",IEE Proceedings-G, Vol.140,(1),1993 ,pp.55-60.  
 [2].ROUX,J.P.,ESCOTTE L.,PLANA R., GRAFFEUIL, DELAGE S.L , "Small-Signal and Noise Model Extraction Technique for heterojunction Bipolar Transistor at Microwave Frequencies",IEEE Trans. 1995-MTT-43(2), pp293-298.  
 [3].F.GÜNEŞ,F.GÜRGEN and H.TORPI,"Signal-noise neural network model for active microwave devices",IEE Proc.-Circuits Devices Syst., 143, (1), pp. 1-8.



## EFFICIENT MODEL PARAMETER EXTRACTION USING NEURAL NETWORKS FOR ACTIVE MICROWAVE DEVICES

(1)F. Günes, (2)F. Gürgen, (1)H. Torpy

(1)Yıldız Technical University, Electronics & Communication Eng., Istanbul/Turkey

(2)Bodazici University, Computer Eng. Dept., Istanbul/Turkey

A new method of modeling, concurrently for the following properties of an active microwave device is proposed:(i) Small signal and noise parameters for multi-bias configuration;(ii) Output and dynamic characteristics. The signal and noise performance of an active microwave device around a bias point are usually given by the scattering S and N parameter vectors at the  $\omega$ -domain and the measured performance data over the operation bandwidth can be arranged in a table-form function whose the first column is the sample frequencies ,the second and third columns are respectively S and N vectors which can be expressed in the form of

$$[S^N]^t = [|S_{11}|^N \phi_{11}^N |S_{12}|^N \phi_{12}^N |S_{21}|^N \phi_{21}^N |S_{22}|^N \phi_{22}^N] \quad (1)$$

$$[N^N]^t = [F_{opt}^N | \Gamma_{opt}^N | o_{opt}^N R_N^N] \quad (2)$$

Here the device is modeled by a black box whose S and N parameters are evaluated through a neural network, based upon fitting of both of these parameters over the operation bandwidth around the chosen bias point on the device output characteristics. The measured performance data functions at the chosen bias conditions ( $V_{CE}, I_C$  or  $V_{DS}, I_{DS}$ ) of a configuration type (common gate, common drain, common source) are utilized for training the neural network model of the device. Besides, the training is also tried for the measured data functions of the possible configurations at the chosen bias points. Performance vectors  $S^{(k)}, N^{(k)}$  at desired frequency  $f_k$  and bias condition can be obtained from the network output by inputting the frequency  $f_k$  and that bias condition. If  $S^{(k)}$  and  $N^{(k)}$  are unmeasured they are determined by the generalization process of the neural network, which can be considered as the ability of the network to give good outputs to inputs it has not been trained on. The basis underlying to obtain equivalencies for the output and dynamic characteristics are the same as the signal and noise equivalencies, which is to train a neural network by a set of measured data, then is to get the desired response by the generalization. A novel neural network of feed-forward type with a single hidden layer is used, back-propagation and nonlinear types of activation functions are utilized in training the network. performance measure is established and the quality factors are evaluated in the generalization procedure for both types of equivalencies of a microwave active device.

In the symposium, many worked examples will be presented for the equivalencies of the active microwave devices.



## NEURAL NETWORK APPROACH FOR THE CHARACTERISATION OF THE ACTIVE MICROWAVE DEVICES

Hamid TORPİ<sup>1</sup>Filiz GÜNEŞ<sup>1</sup>Fikret GÜRGEN<sup>2</sup>

<sup>1</sup>*Yıldız Technical University, Electronics & Communication Eng. Dept., 80670 MASLAK-ISTANBUL/TÜRKİYE*

<sup>2</sup>*Boğaziçi University, Computer Eng. Dept., BEBEK-ISTANBUL/TÜRKİYE*

### Abstract

Small-signal and noise behaviour of an active microwave device is modeled through the neural network approach for multiple bias/configurations. Here, the device is modelled by a black box whose small signal and noise parameters are evaluated through a neural network, based upon the fitting of both of these parameters for the multiple bias or configuration. The concurrent modelling procedure does not require to solve device physics equations repeatedly during optimization. Compared to the existing device modelling techniques, the proposed approach has the capability to make high-dimensional models for highly nonlinear devices.

### I. INTRODUCTION

#### *Aim of the work*

The aim of this work is to model a microwave transistor by a black-box for which small-signal and noise parameters are evaluated through a neural network, based upon the fitting of both of these parameters to the corresponding measured data over the whole operational range from DC to more than 10 GHz for multiple bias of various types of configuration. So the stages of the work can be ordered as follows:

- (i) Establish a novel neural network of feedforward type with a single hidden layer,
- (ii) using back-propagation and nonlinear types of activation → functions, train the network for both the signal-noise behaviours over the operational bandwidth for multiple bias and multiple configuration of any type of microwave transistor.
- (iii) Establish performance measure of the model,
- (iv) Predict the small-signal and noise behaviours at any operation frequency around any bias condition of any type of configuration using the neural network which has already been trained to make functional approximations of the device nonlinear characteristics in the vicinities of the chosen bias points.

From the classical point of view, unified small-signal-noise equivalent circuit for a microwave transistor can be divided in to two parts: Extrinsic circuits and intrinsic circuit. The intrinsic circuit characterize the

active region under the gate (or base) whose parameters are functions of bias conditions and device technological parameters, whereas the extrinsic parameters depend, at least to a first approximation, only on the technological parameters. If a unified circuit for a MESFET is considered, the most important extrinsic parameters are the gate, source and drain inductances due to the bond wires and the gate, source and drain resistances. The four main intrinsic parameters are the input capacitance  $C_{GS}$ , the transconductance  $g_m$ , the output conductance  $g_d$  and the feedback capacitance  $C_{GD}$ . In addition, the electrical behaviour of the intrinsic device requires the introduction of two more parameters: The intrinsic resistance  $R_i$  which can be related to distributed nature of the RC input network, and the delay  $\tau$ , which is introduced in the expression for the current generator and corresponds to the time needed for the carriers to travel under the gate.

Briefly due to the intrinsic and extrinsic device properties, both the signal and noise parameters are the functions of the bias conditions, frequency, configuration types. The way to approximate these functions in the literature so far is considered in the forthcoming subsection.

#### *Review the literature*

The problem of approximating measured device parameters or device response has been formulated as an optimization problem with respect to the equivalent circuit of a proposed model. The traditional approach in modeling is almost entirely directed at achieving the best possible match between measured and calculated parameters. This approach has serious shortcomings in two frequently encountered cases. The first case is when the equivalent circuit parameters are not unique with responses selected and the second is when the nonideal effects are not modelled adequately, the latter causing an imperfect match even if small measurement errors are allowed for. In both cases, a family of solutions for circuit model parameters may exist which produce reasonable and similar matches between measured and calculated responses. Besides, published literature is concerned with the equivalent circuit for the single-bias which are only either small-signal models or the noise behaviour descriptions based on existing signal equivalent circuits that have nothing to do with the device noise characteristics. In [1] and [2] these two behaviours are combined in an unified classical circuit model for only a

single-bias. A recent work [3] combines the signal and noise parameters in a neural network model over the fairly large operation band at a single bias point.

## II. NEURAL NETWORK MODEL

### Signal-Noise Behaviour of a Microwave Transistor

S and N parameter data measured at the multiple bias conditions ( $V_{DS}, I_{DS}$ ) for the configuration types (0.2, 0.5, 0.8) is all used to train the neural network. The amount of data used in the training and iteration number are altogether optimised against the error. The measured S and N parameter data around a bias point for a type of configuration can be arranged in a table-form function as follows:

$$\begin{bmatrix} f_1 & : & \mathbf{S}^{(1)} & \mathbf{N}^{(1)} \\ f_2 & : & \mathbf{S}^{(2)} & \mathbf{N}^{(2)} \\ \vdots & & \vdots & \vdots \\ f_N & : & \mathbf{S}^{(N)} & \mathbf{N}^{(N)} \end{bmatrix} \quad (1)$$

where  $\mathbf{S}^{(1)}, \mathbf{N}^{(1)}, \dots, \mathbf{S}^{(N)}, \mathbf{N}^{(N)}$  are respectively, the scattering and noise vectors at the  $f_1, \dots, f_N$  measurement frequencies, and  $\mathbf{S}^{(N)}$  and  $\mathbf{N}^{(N)}$  performance vectors can be given as follows:

$$\begin{bmatrix} \mathbf{S}^{(N)} \end{bmatrix}^\dagger = \begin{bmatrix} |S_{11}|^{(N)} & \phi_{11}^{(N)} & |S_{12}|^{(N)} & \phi_{12}^{(N)} & |S_{21}|^{(N)} & \phi_{21}^{(N)} & |S_{22}|^{(N)} & \phi_{22}^{(N)} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{N}^{(N)} \end{bmatrix}^\dagger = \begin{bmatrix} F_{opt}^{(N)} & |\Gamma_{opt}|^{(N)} & \phi_{opt}^{(N)} & F_N^{(N)} \end{bmatrix} \quad (2)$$

After having completed the training process, the performance vectors  $\mathbf{S}^{(k)}, \mathbf{N}^{(k)}$  at a desired frequency  $f_k$  at the conditions ( $V_{CE}, I_C, CT$  or  $V_{DS}, I_{DS}, CT$ ) for any configuration type among the trained ones can be obtained from the network output by inputting the frequency  $f_k$  bias configuration type which is defined by the numbers (0.2, 0.5, 0.8). If  $\mathbf{S}^{(k)}, \mathbf{N}^{(k)}$  are unmeasured they are determined by the generalization process of the neural network, which can be considered as the ability of the network to give good outputs to inputs it has not been trained on. In our application, the signal-noise neural network can generalize the performance not only at a single operation frequency of the trained bias condition, at the same time in the whole operation band of the untrained bias condition. The first may be named as the single frequency generalization, while the latter is called whole band generalization, worked examples of which will be given in the result section.

### The multi-bias and configuration signal-noise neural network

We use a novel neural network of feedforward type with a single hidden layer having the same number of nodes as the output layer. Let  $n, N_h$  and  $N_o$  be respectively the number of nodes in the input, hidden and output layers. In the signal-noise neural network  $n=4$  with the frequency, bias condition and the type of configuration  $CT, N_h=N_o=12$  which are the signal and noise vectors given by (2). (Fig.2)

The signal resulted from the hidden layer to the  $i$  th output node can be expressed in the form of

$$\Phi_i(\mathbf{x}, \mathbf{T}_i, \mathbf{W}, \theta_h) = \sum_{h=1}^{N_h} T_{hi} g_h(\mathbf{x}, \mathbf{W}_h, \theta_h) \quad (3)$$

and the net output of the  $i$  th output node is obtained as follows

$$\phi_i(\mathbf{x}, \mathbf{T}_i, \mathbf{W}, \Theta) = V_i f_i(\Phi_i + \Theta_i) \quad (4)$$

where  $g_h$  and  $f_i$  are the basis functions for the  $h$  th hidden node and the  $i$  th output node, respectively, which are sigmoid type of nonlinear functions in our case, e.g.  $g_h(\mathbf{W}_h, \mathbf{x})$  can be expressed in the following form:

$$g_h(\mathbf{x}, \mathbf{W}_h, \theta_h) = \frac{1}{1 + \exp\left(-\sum_{i=1}^n x_i W_{ih} - \theta_h\right)} \quad (5)$$

In equations 3-5,  $\mathbf{x}$  is the input vector of  $n$  dimensions:

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^\dagger \quad (6)$$

$\mathbf{T}_i$  is the weighting vector between the  $i$  th output node and the hidden layer:

$$\mathbf{T}_i = [T_{1i} \ T_{2i} \ T_{3i} \ T_{hi} \ T_{N_{hi}}]^\dagger \quad (7)$$

$\mathbf{W}$  is the weighting matrix between the hidden and input layer:

$$\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2 \ \dots \ \mathbf{W}_h \ \dots \ \mathbf{W}_{N_h}]^\dagger \quad (8)$$

where  $\mathbf{W}_h$  vector is the weight(ing) vector between the input layer and the  $h$  th hidden node and can be given by:

$$\mathbf{W}_h = [W_{1h}, W_{2h}, \dots, W_{nh}]^\dagger \quad (9a)$$

$\theta_h, \Theta_i$  are the local memories belonging to the  $h$  th hidden and  $i$  th output nodes, respectively. In the eqn. (4)  $V$  is the weighting factor of the output layer:

$$\mathbf{V} = [V_1, V_2, \dots, V_{N_o}]^\dagger \quad (9b)$$

### Determination of the network parameter matrix P

If parameters of the network architecture is denoted  $\mathbf{P}$ , the network parameter matrix  $\mathbf{P}$  will have  $N_h \times n + N_h \times N_o + N_h + N_o$  elements which consist of weighting factors between the input and hidden layers and the hidden and output layers, the local memories of the hidden and output nodes. The training process can be defined as computation of the network parameter matrix  $\mathbf{P}$  so that the error function which is

$$E(\mathbf{P}) = \sum_{k=1}^{N_S} E^{(k)} = \sum_{k=1}^{N_S} \left[ \frac{1}{2} \sum_{j=1}^{N_o} (y_j^{(k)} - \phi_j^{(k)})^2 \right] \quad (10)$$

is minimised, where  $y_j^{(k)}$  and  $\phi_j^{(k)}$  are respectively the measured and predicted values of the  $j$  th output at the training frequency  $f_k$ . This training process is also called

backpropagation and it is an 'on-line' process whose update equations for  $T_{hj}, W_{ih}, \theta_h$  can be given as follows:

$$T_{hj}^{(k+1)} = T_{hj}^{(k)} - \eta \frac{\partial E^{(k)}}{\partial T_{hj}} + \alpha (T_{hj}^{(k)} - T_{hj}^{(k-1)}) \quad (11.1)$$

$$W_{ih}^{(k+1)} = W_{ih}^{(k)} - \eta \frac{\partial E^{(k)}}{\partial W_{ih}} + \alpha (W_{ih}^{(k)} - W_{ih}^{(k-1)}) \quad (11.2)$$

$$\theta_h^{(k+1)} = \theta_h^{(k)} - \eta \frac{\partial E^{(k)}}{\partial \theta_h} + \alpha (\theta_h^{(k)} - \theta_h^{(k-1)}) \quad (11.3)$$

and the similar equations can be written for  $\Theta_i$  and  $V_i$ . In (11.1)-(11.3)  $\eta$  and  $\alpha$  are positive-valued learning rate and momentum, respectively. Thus we start any set the network parameters and then repeatedly change each parameter by an amount proportional to the terms  $\frac{\partial E^{(k)}}{\partial T_{hj}}$ ,

$\frac{\partial E^{(k)}}{\partial W_{ih}}$ ,  $\frac{\partial E^{(k)}}{\partial \theta_h}$  according to the eqns. (11.1-11.3) and assume that the training is completed when the error fails to decrease any further. In this case we take the best so far.

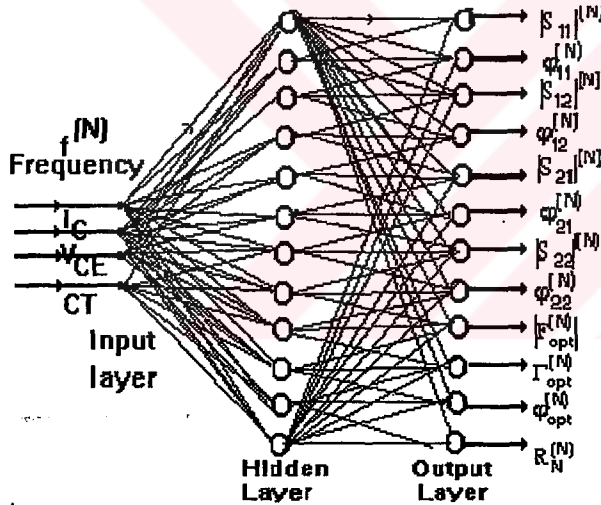


Fig. 1 Multi-bias and configuration signal-noise neural network

The sensitivity through the neural network with respect to  $T_{hj}, W_{ih}, \theta_h$  can be given as follows

$$\frac{\partial E^{(k)}}{\partial T_{hj}} = \frac{\partial \left[ \frac{1}{2} \sum_{j=1}^{N_o} (y_j^{(k)} - \phi_j^{(k)})^2 \right]}{\partial T_{hj}} = -V_j (y_j^{(k)} - \phi_j^{(k)}) f_j (1 - f_j) g_h$$

$$\frac{\partial E^{(k)}}{\partial T_{hj}} = \delta_j^{(3)} g_h \quad (12.1)$$

$$\frac{\partial E^{(k)}}{\partial W_{ih}} = \sum_{j=1}^{N_o} \frac{\partial E^{(k)}}{\partial \phi_j} \frac{d\phi_j}{df_j} \frac{df_j}{d\Phi_j} \frac{\partial \Phi_j}{\partial W_{ih}} = \sum_{j=1}^{N_o} -\delta_j^{(3)} \frac{\partial \Phi_j}{\partial W_{ih}} \quad (12.2)$$

$$\frac{\partial E^{(k)}}{\partial W_{ih}} = g_h (1 - g_h) \sum_{j=1}^{N_o} -\delta_j^{(3)} T_{hj} x_{ki} = \delta_h^{(2)} x_{ki}$$

$$\frac{\partial E^{(k)}}{\partial \theta_h} = \sum_{j=1}^{N_o} \frac{\partial E^{(k)}}{\partial \phi_j} \frac{d\phi_j}{df_j} \frac{df_j}{d\Phi_j} \frac{\partial \Phi_j}{\partial \theta_h} \quad (12.3)$$

$$\frac{\partial E^{(k)}}{\partial \theta_h} = g_h (1 - g_h) \sum_{j=1}^{N_o} -\delta_j^{(3)} T_{hj} = \delta_h^{(2)}$$

$$\frac{\partial E^{(k)}}{\partial V_j} = \frac{\partial E^{(k)}}{\partial \phi_j^{(k)}} \frac{\partial \phi_j^{(k)}}{\partial V_j} = -(y_j^{(k)} - \phi_j^{(k)}) f_j (\Phi_j + \Theta_j)$$

$$\frac{\partial E^{(k)}}{\partial V_j} = \delta_j^{(3)} (1 - f_j)^{-1} \quad (12.4)$$

$$\frac{\partial E^{(k)}}{\partial \Theta_j} = -(y_j^{(k)} - \phi_j^{(k)}) f_j (1 - f_j) = \delta_j^{(3)} V_j^{-1} \quad (12.5)$$

where  $\delta_h^{(2)}$  and  $\delta_j^{(3)}$  represent local gradients at individual node in the second and third layer, respectively.

### III. PERFORMANCE MEASURE AND RESULTS

To evaluate the quality of the fit to measured data the following error terms are found to be convenient:

$$E_{S_{ij}} = \frac{1}{n} \sum_{k=1}^n \frac{|S_{ij,meas}^k - S_{ij,meas}^k|}{|S_{ij,meas}^k|} \quad (13)$$

$$E_{N_i} = \frac{1}{n} \sum_{i=1}^n \frac{|N_{i,meas}^k - N_{i,predict}^k|}{|N_{i,meas}^k|} \quad (14)$$

Where  $S_{ij}$  and  $N_i$  are, respectively the signal and noise parameters, and  $n$  is the number of discrete frequencies used. Total average error can be defined as the average of the signal and noise errors:

$$E_T = \frac{1}{4} \sum_{i=1}^4 E_{i,signal} + \frac{1}{3} \sum_{i=1}^3 E_{i,noise} \quad (15)$$

Distribution of errors with frequency for the whole band generalization at multiple bias points for the common collector configuration is given in Fig.3. Simulation results of NE02135 (iter.num.400000) Transistor are given in Fig.4 a-d which shows Distribution of errors with frequency for the whole band generalization at multiple bias points for the common emitter configuration. In addition simulation results of NE219 transistor are given in Fig.5 a-f which shows variations of  $S$  parameters with frequency from 2 -6 GHz for the  $V_{CE}=8V$  and  $I_C=10,20$  and 30 mA at the common emitter configuration which show quite good agreement of the signal parameters over the operation band-width. The graphs include variations of  $S$  parameters and noise parameters with frequency from 0.5 -4 GHz for the  $V_{CE}=10V$  and  $I_C=5$ ,

10, 20 and 30 mA at the common emitter configuration. Finally variations of S parameters with respect to bias point is given for various constant frequencies. (Fig.6) *Acknowledgement: This work was supported by the Yildiz Technical University Research Fund. Project number: 92-A-04-03-12.*

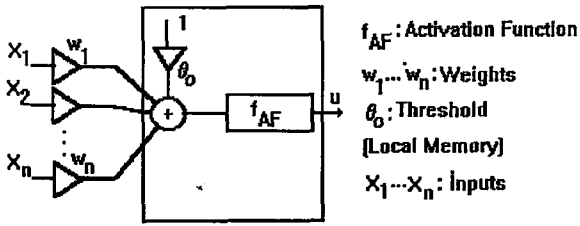


Fig. 2 A perceptron node

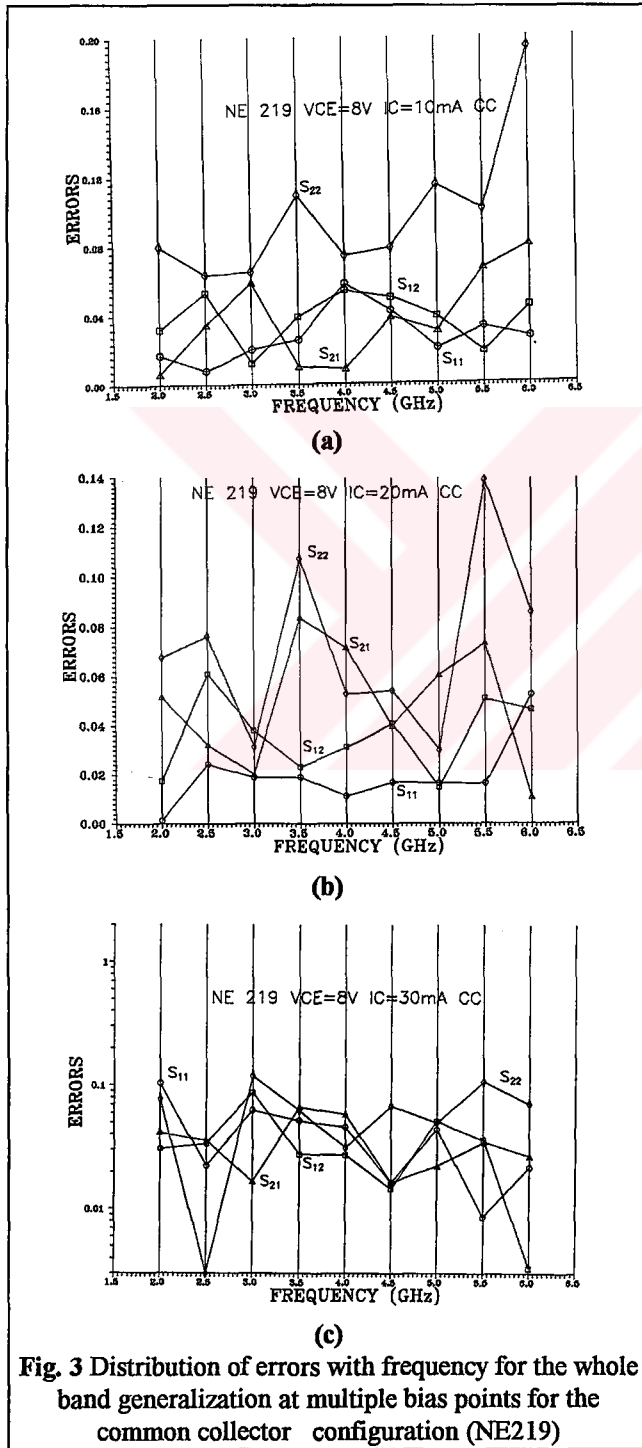


Fig. 3 Distribution of errors with frequency for the whole band generalization at multiple bias points for the common collector configuration (NE219)

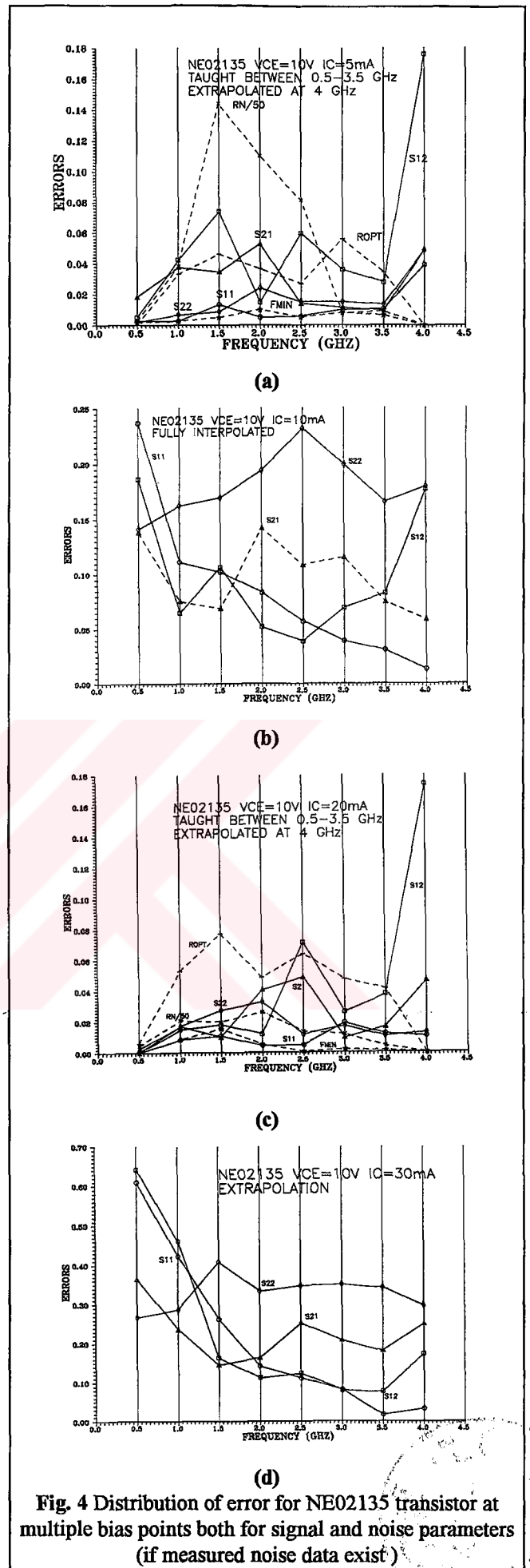
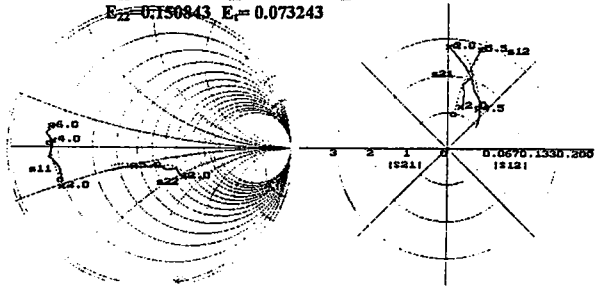


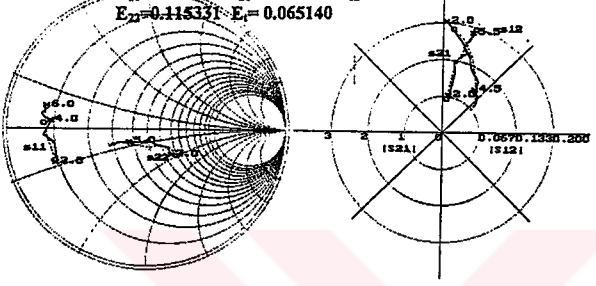
Fig. 4 Distribution of error for NE02135 transistor at multiple bias points both for signal and noise parameters (if measured noise data exist)

Bias Condition:  $V_{CE}=8.0$  [Volt]  $I_C=10.0$  [mA]  
 COMMON EMITTER CONFIGURATION  
 $E_{11}=0.045472$   $E_{21}=0.034876$   $E_{12}=0.061781$   
 $E_{22}=0.750843$   $E_r=0.073243$



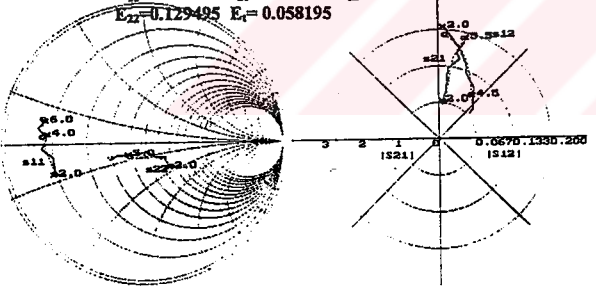
a-  $V_{CE}=8$  V  $I_C=10$  mA, Common Emitter Configuration (Taught)

Bias Condition:  $V_{CE}=8.0$  [Volt]  $I_C=20.0$  [mA]  
 COMMON EMITTER CONFIGURATION  
 $E_{11}=0.038773$   $E_{21}=0.044730$   $E_{12}=0.060726$   
 $E_{22}=0.115331$   $E_r=0.065140$



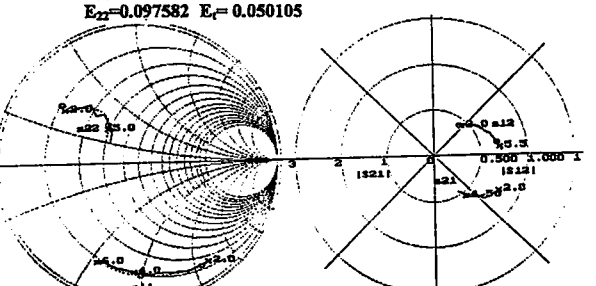
b-  $V_{CE}=8$  V  $I_C=20$  mA, Common Emitter Configuration (Taught)

Bias Condition:  $V_{CE}=8.0$  [Volt]  $I_C=30.0$  [mA]  
 COMMON EMITTER CONFIGURATION  
 $E_{11}=0.028880$   $E_{21}=0.032119$   $E_{12}=0.042287$   
 $E_{22}=0.129495$   $E_r=0.058195$



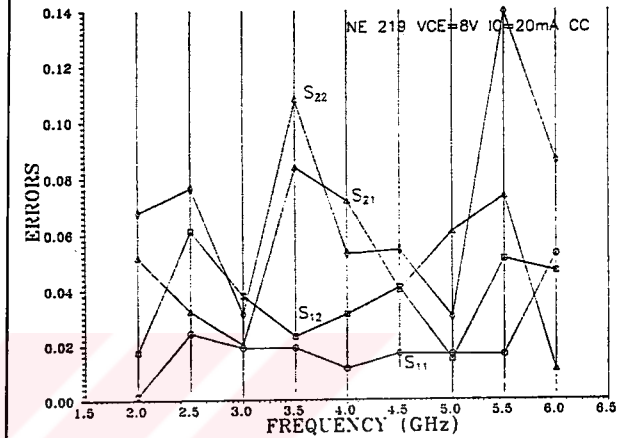
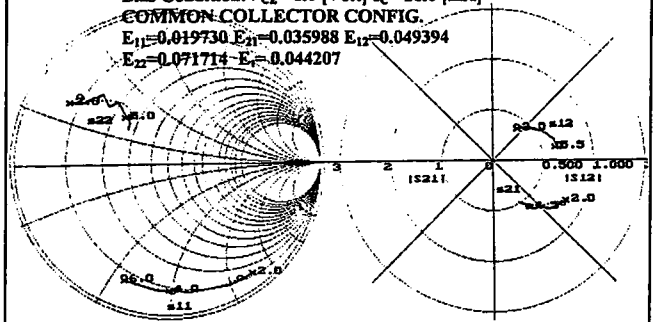
c-  $V_{CE}=8$  V  $I_C=30$  mA, Common Emitter Configuration (Interpolated)

Bias Condition:  $V_{CE}=8.0$  [Volt]  $I_C=10.0$  [mA]  
 COMMON COLLECTOR CONFIG.  
 $E_{11}=0.027857$   $E_{21}=0.037930$   $E_{12}=0.037051$   
 $E_{22}=0.097582$   $E_r=0.050105$



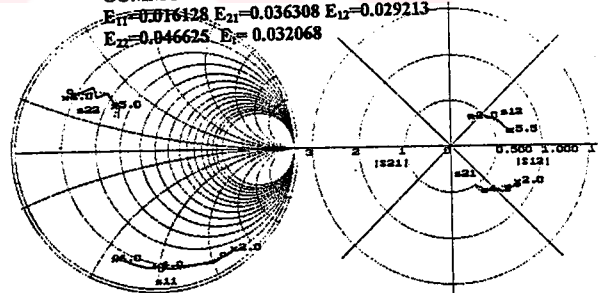
d-  $V_{CE}=8$  V  $I_C=10$  mA, Common Collector Configuration (Interpolated)

Bias Condition:  $V_{CE}=8.0$  [Volt]  $I_C=20.0$  [mA]  
 COMMON COLLECTOR CONFIG.  
 $E_{11}=0.019730$   $E_{21}=0.035988$   $E_{12}=0.049394$   
 $E_{22}=0.071714$   $E_r=0.044207$



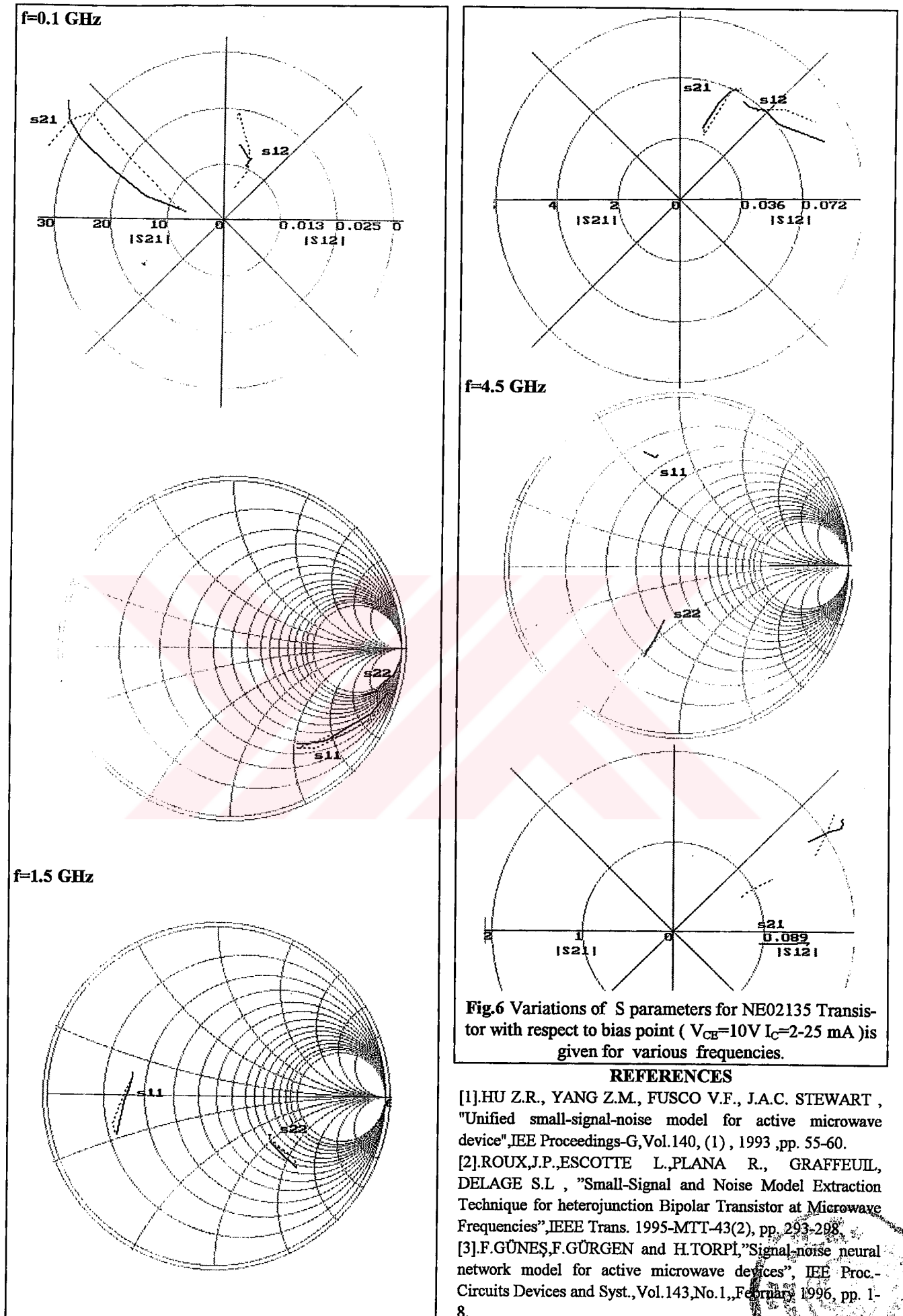
e-  $V_{CE}=8$  V  $I_C=20$  mA, Common Collector Configuration (Taught)

Bias Condition:  $V_{CE}=8.0$  [Volt]  $I_C=30.0$  [mA]  
 COMMON COLLECTOR CONFIG.  
 $E_{11}=0.016428$   $E_{21}=0.036308$   $E_{12}=0.029213$   
 $E_{22}=0.046625$   $E_r=0.032068$



f-  $V_{CE}=8$  V  $I_C=30$  mA, Common Collector Configuration (taught)

Fig. 5 Calculated (—) and measured (---) S parameter shown on Smith chart and polar coordinates for NE219 Transistor at various bias conditions. Error-frequency distribution for  $V_{CE}=8$  V  $I_C=20$  mA at the Common Emitter Configuration is also added.





# NEURAL NETWORK APPROACH FOR THE ACTIVE DEVICE CHARACTERIZATION

Filiz GÜNEŞ Hamid TORPİ Bedri A.ÇETİNER

YILDIZ TECHNICAL UNIVERSITY, Electronics&Communication Eng. Dept.  
80670 MASLAK-ISTANBUL /TÜRKIYE

## I. DESCRIPTION OF THE WORK

This work can be considered into two parts:(1) The first part is the multiple bias and configuration, signal-noise neural network through which small - signal and noise parameters of the device can be evaluated for the whole operating ranges including frequency, bias and configuration types-So multidimensional signal and noise modeling is achieved for the microwave transistors, as the result of this the signal S and noise N parameters are predicted not only at a single operation frequency around the measured bias condition for a configuration which can be called as the single frequency generalization (SFG), at the same time for the whole operation frequency band for an unmeasured bias condition, which may be named as the whole frequency band generalization (WFBG), with good agreement compared to the measurement. (2) The second part is computer simulation of the performance characterization through which the performance contours at an operating frequency of a bias condition can be obtained for a microwave transistor using cross- relations the possible (F<sub>i</sub>, V<sub>i</sub>, G<sub>imax</sub>) triplets.(Fig.1)

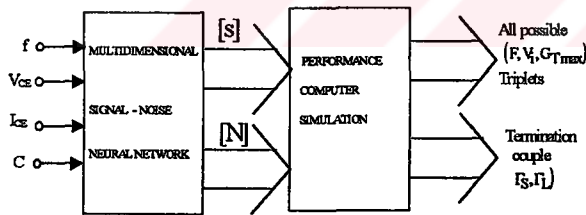


Fig.1 Neural Network Approach for The Active Device Characterization

## II. SIGNAL-NOISE NEURAL NETWORK

### A. Structure of the network

A neural network is a simplified mathematical model of biological neural network. It consists of a collection interconnecting neurons. Let

$$x=[x_1 \ x_2 \ \dots \ x_n]^t \quad , \quad y=[y_1 \ y_2 \ \dots \ y_p]^t \quad (1)$$

be respectively input and output vectors of n and p dimensions in the signal-neural network, x is a 4-dimensional vector containing frequency f<sub>k</sub>, bias condition (V<sub>DS</sub>, I<sub>DS</sub>) and configuration type (CT), y is

a 12-dimensional vector which gives S- and N-parameters(Fig. 2) and the relationship between x and y is multidimensional and nonlinear. To model such a typical multidimensional nonlinear relationship a three-layer neural network is employed, which has n processing nodes (PN) in the input layer, p PN's in the output layer, and q PN's in the hidden layer as shown Fig.2 and the block diagram of a PN is given Fig.3 .

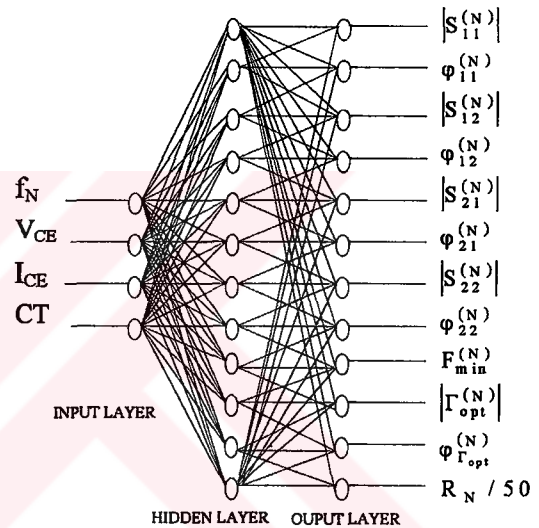


Fig.2 Signal-Noise Neural Network

Let

$$a^{(k)} = [a_1^{(k)} \ a_2^{(k)} \ \dots \ a_n^{(k)}]^t$$

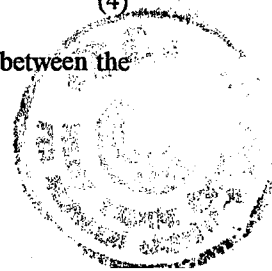
$$b^{(k)} = [b_1^{(k)} \ b_2^{(k)} \ \dots \ b_n^{(k)}]^t$$

be vectors representing, the kth sample of the inputs and outputs, respectively, k=1,2,..., and N<sub>s</sub> is the total number of the data samples. The weighting matrix between the hidden and input layer is W, and between the hidden and output layers is V, which can be expressed as

$$W = [ W_1, W_2 \ \dots \ W_h \ \dots \ W_q ] \quad (3)$$

$$V = [ V_1 \ V_2 \ \dots \ V_h \ \dots \ V_p ] \quad (4)$$

where W<sub>h</sub> vector is the weighting vector between the h th hidden node and the input layer:



$$\mathbf{W}_h = [W_{1h} \ W_{2h} \ \dots \ W_{nh}]^t \quad (5)$$

$\mathbf{V}_j$  is the weighting vector between the  $j$  th output node and the hidden layer:

$$\mathbf{V}_j = [V_{1j} \ V_{2j} \ \dots \ V_{qj}]^t \quad (6)$$

The signal resulted from the hidden layer to the  $j$  th output node can be expressed in the form of

$$\Phi_j(\mathbf{x}, \mathbf{W}, \mathbf{V}_j, \theta_h) = \sum_{h=1}^q V_{hj} g_h(\mathbf{x}, \mathbf{W}_h, \theta_h) \quad (7)$$

and the net output of the  $j$  th output node is obtained as follows

$$y_j(\mathbf{x}, \mathbf{W}, \mathbf{V}_j, \theta_h, T_j, \Theta_j) = T_j f_j(\Phi_j + \Theta_j) \quad (8)$$

where  $g_h$  and  $f_j$  are the basis functions for the  $h$  th hidden node and the  $j$  th output node, respectively, which are sigmoid type of nonlinear functions in our application, and  $g_h(\mathbf{x}, \mathbf{W}_h, \theta_h)$  can be expressed in the following form:

$$g_h(\mathbf{x}, \mathbf{W}_h, \theta_h) = \frac{1}{1 + \exp\left(-\left(\sum_{i=1}^n x_i W_{ih}\right) - \theta_h\right)} \quad (9)$$

If the network architecture is denoted by the matrix  $\mathbf{P}$ , the network parameter matrix  $\mathbf{P}$  will have  $n \times q + p \times q + q + p$  elements which consist of weighting factors between the input and hidden layers and the hidden and output layers, the local memories of the hidden and output nodes respectively. In the eqns. (7) and (8),  $\Theta_j$  and  $\theta_h$  are the thresholds of the  $j$  th output and  $h$  th hidden nodes, respectively.

#### B. Determination of the Network Matrix $\mathbf{P}$

If the training process can be defined as computation of the network parameter matrix  $\mathbf{P}$  so that the error function which is

$$E(\mathbf{P}) = \sum_{k=1}^{N_S} E^{(k)} = \sum_{k=1}^{N_S} \left[ \frac{1}{2} \sum_{j=1}^p \left( y_j^{(k)} - b_j^{(k)} \right)^2 \right] \quad (10)$$

is minimized, where  $y_j^{(k)}$  and  $b_j^{(k)}$  are respectively, the predicted and measured values of the  $j$  th output node at the training frequency  $f_k$ . This training process is also called backpropagation which is available in the two types of algorithms: 'on-line' training where neural network parameters are updated after each sample presentation, and 'off-line' training where neural network parameters are

updated after all samples are presented. In this work, we chose the 'on-line' training approach since it is more efficient in most cases.

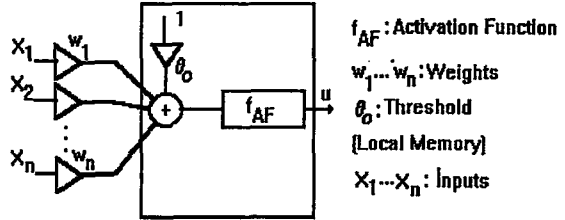


Fig. 3 A Processing Node (PN)

The update equations for  $V_{hj}, W_{ih}, \theta_h$  can be given as follows:

$$V_{hj}^{(k+1)} = V_{hj}^{(k)} - \eta \frac{\partial E^{(k)}}{\partial V_{hj}} + \alpha \left( V_{hj}^{(k)} - V_{hj}^{(k-1)} \right) \quad (11.1)$$

$$W_{ih}^{(k+1)} = W_{ih}^{(k)} - \eta \frac{\partial E^{(k)}}{\partial W_{ih}} + \alpha \left( W_{ih}^{(k)} - W_{ih}^{(k-1)} \right) \quad (11.2)$$

$$\theta_h^{(k+1)} = \theta_h^{(k)} - \eta \frac{\partial E^{(k)}}{\partial \theta_h} + \alpha \left( \theta_h^{(k)} - \theta_h^{(k-1)} \right) \quad (11.3)$$

and the similar equations can be written for  $\Theta_j$  and  $T_j$ . In the eqns. (11.1)-(11.3)  $\eta$  and  $\alpha$  are positive-valued between 0 and 1 learning rate and momentum, respectively. In this work, a simple program is used to search their optimum values depending on the worked transistor where the strategy is to determine optimum values for  $\alpha$  and  $\eta$  to ensure rapid convergence to the satisfied performance measures of the model. Thus we start any set for the network parameters and then repeatedly change each parameter by an amount proportional to the related

sensitivity terms such as  $\frac{\partial E^{(k)}}{\partial V_{hj}}, \frac{\partial E^{(k)}}{\partial W_{ih}}, \frac{\partial E^{(k)}}{\partial \theta_h}$

according to update equations and assume that the training is completed when the error fails to decrease any further, the network parameters are then the final set of values of  $V_{hj}, W_{ih}, \theta_h, T_j$  and  $\Theta_j$ .

The manufacturers characterization data is used for the training of the signal-noise neural.

#### C. Performance Measure of the Model

To evaluate the quality of the fit to measured data the following error terms are found to be convenient

$$E_T = \frac{1}{4} \sum_{i=1}^4 E_{i_{\text{signal}}} \quad F_T = \frac{1}{3} \sum_{i=1}^3 E_{i_{\text{noise}}}$$

$$E_{S_{ij}} = \frac{1}{N_S} \sum_{k=1}^{N_s} \frac{|S_{ij_{max}}^k - S_{ij_{predc.}}^k|}{|S_{ij_{max}}^k|}, E_{N_i} = \frac{1}{N_S} \sum_{i=1}^{N_s} \frac{|N_{i_{max}}^k - N_{i_{predc.}}^k|}{|N_{i_{max}}^k|} \quad (12)$$

### III PERFORMANCE SIMULATION PROGRAM

Performance simulation program is based on the novel Smith Chart formulation of the performance triplet ( $F_i, V_i, G_{tmax}$ )'s [1] and its flow chart is given in the Fig.12. For a [S] and [N] parameter set at an operating frequency around a bias condition, which are provided by the signal-noise neural network in our work, all the possible performance triplet ( $F_i, V_i, G_{tmax}$ )s and their terminations couple ( $\Gamma_S \Gamma_L$ )s can be obtained from the simulation program momentarily.

### IV WORKED EXAMPLES AND CONCLUSION

The first worked example is the multidimensional signal-noise neural network of the transistor NE02135 over its operating frequency range 0.1 GHz ... 4 GHz around the bias conditions  $V_{CE}=10V$  and  $I_C=5,7.5,10,15,20,25,30$  mA 's: It should be noted that once the neural network has been trained, it responds immediately to the desired input. The variations of the training time in minute against the iteration number and the training error against the iteration number have been investigated in [2]. The generalization process can be considered into the two categories: (i) Single Frequency Generalization (SFG) :It can be defined as the performance parameter prediction at a single operation frequency of the trained bias condition. frequency variations of the interpolation and extrapolation error for the S- and N- parameters have been obtained and it is seen that the network has a high capability to interpolate and extrapolate in and outside the data domain used for learning respectively. Figs.4,5,7 give the results of the SFG processes for the transistor NE02135 . (ii) The Whole Frequency Band Generalization (WFBG): This can be defined as prediction of the unknown multidimensional performance-parameter function for the whole operation frequency band at an untrained bias condition. For the transistor NE02135 at any desired bias condition, the WFBG processes can be achieved successfully and Fig.6 gives the results of WFBG process for the  $V_{CE}=10V, 10mA$ .

The second worked example is the performance characterization of the same transistor NE02135 over its whole operation range whose parameters have been obtained from its multidimensional signal-noise neural network. So

this work is a typical example of how a multidimensional signal-noise neural network can be utilized. Fig. 8 and 9 give variations of the maximum available gain  $G_{tmax}$  against noise F and VSWR, respectively, for a fixed frequency (0.5 GHz) around a fixed bias condition ( $V_{CE}=10V, I_{CE}= 5mA$ ). Figs 10 and 11 give variations of the maximum available gain  $G_{tmax}$  against operating frequency f and bias current  $I_{CE}$ , respectively, for a fixed bias voltage  $V_{CE}$  and noise F and input VSWR  $V_i$ .

Briefly it can be concluded that since optimization of the performance can be made interactively with respect to noise F, input VSWR, operating frequency f, bias condition ( $V_{CE}, I_{CE}$ ) the multidimensional signal-noise neural network model cascaded in the performance characterization computer simulation provides an excellent tool for the design of the monolithic microwave integrated circuit (MMIC).

#### Worked Example 1: Multiple Bias and Signal-Noise Neural Network Model for the Transistor NE02135 at the Common Emitter Configuration.

TABLE 1

Name of the transistor: NE02135  
 Operation Frequency Bandwidth: 0.1 GHz - 4 GHz  
 Operations Bias Conditions  
 Bias NMDA: No Measured Data Available  
 Volta Bias Currents ( $I_C$ ) [mA]  
 ge  
 ( $V_{CE}$ )  
 [V]  
 10

	5.0	10.0	20.0	30.0
SFG	SFG	WFBG	SFG	SFG
$E_k=0.0412$	$E_k=0.0797$	$E_k=0.0370$	$E_k=0.0445$	
$E_{11}=0.0338$	$E_{11}=0.0343$	$E_{11}=0.0235$	$E_{11}=0.0184$	
$E_{21}=0.0440$	$E_{21}=0.0562$	$E_{21}=0.0434$	$E_{21}=0.0499$	
$E_{12}=0.0437$	$E_{12}=0.0973$	$E_{12}=0.0466$	$E_{12}=0.0672$	
$E_{22}=0.0435$	$E_{22}=0.1309$	$E_{22}=0.0343$	$E_{22}=0.0426$	
$F_1=0.0308$	$F_1=NMDA$	$F_1=0.0239$	$F_1=0.0191$	
$F_2=0.1027$	$F_2=NMDA$	$F_2=0.0528$	$F_2=0.0768$	
$F_3=0.1237$	$F_3=NMDA$	$F_3=0.0963$	$F_3=0.0659$	
$F_i=0.0858$	$F_i=NMDA$	$F_i=0.0577$	$F_i=0.0539$	
Fig.7	Fig.6	Fig.5	Fig.4	

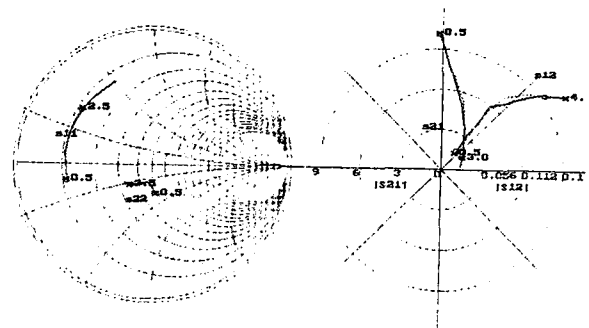


Fig.4 Predicted and measured S-parameter-frequency variations for  $V_{CE}=10 V, I_C=30 mA$  at the common emitter configuration.(SFG)

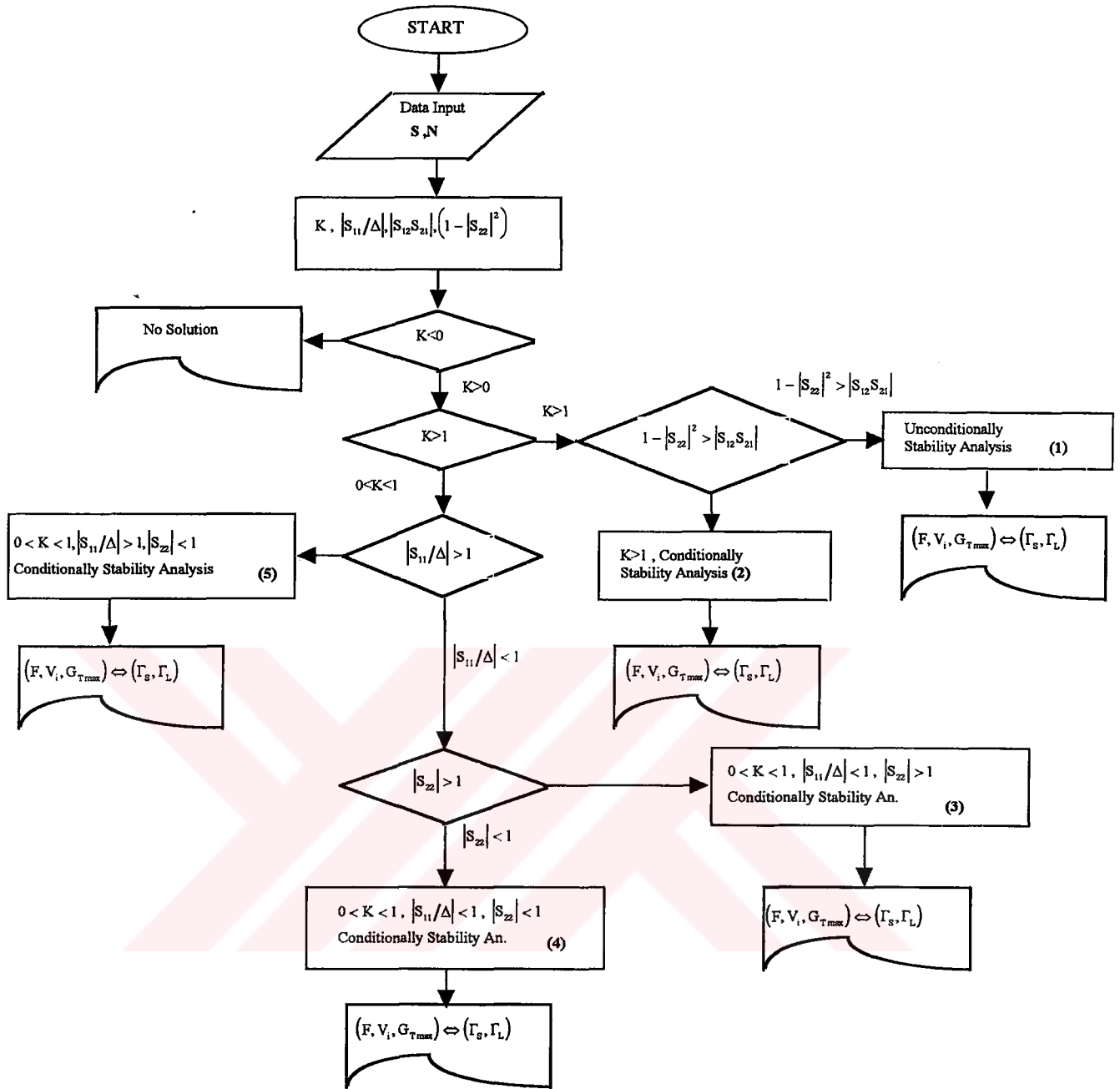


Fig.12 Main Flow Chart of the Performance Characterisation Programme



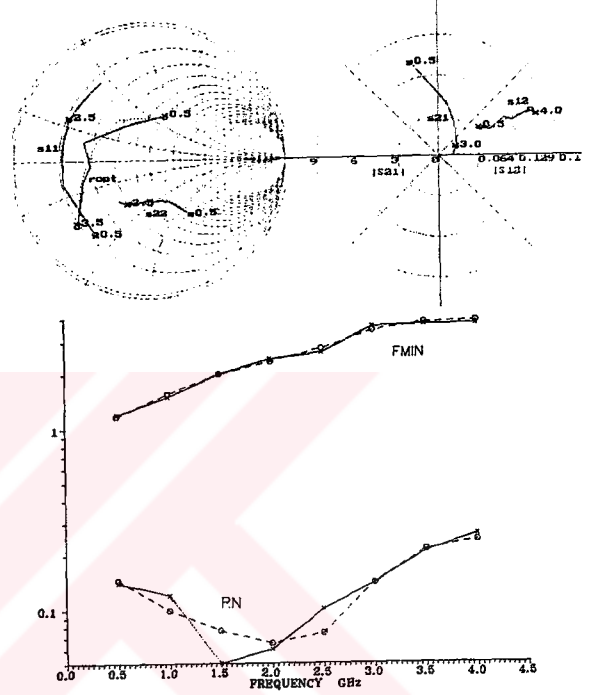
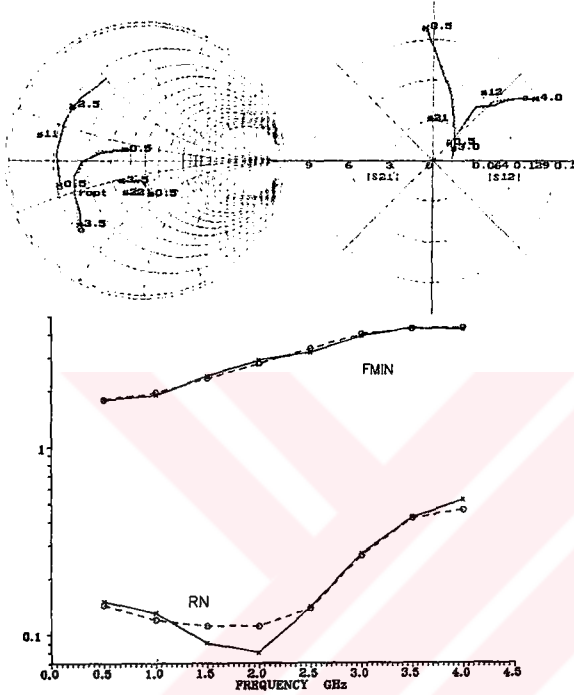
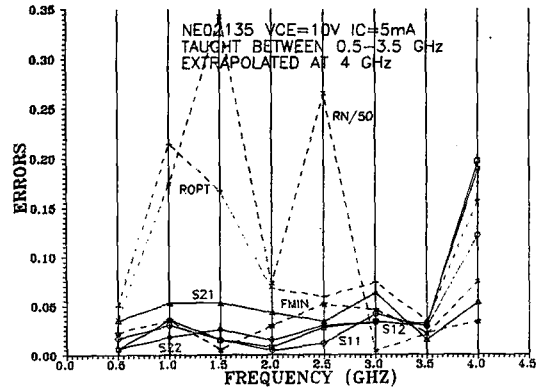
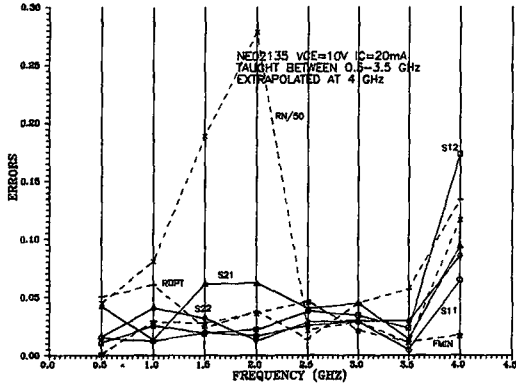


Fig.5 Predicted and measured S&N-parameter-frequency variations&error-frequency distributions for  $V_{CE}=10\text{ V}$   $I_C=20\text{ mA}$  at the common emitter configuration.(SFG)

Fig.7 Predicted and measured S&N-parameter-frequency variations&error-frequency distributions for  $V_{CE}=10\text{ V}$   $I_C=5\text{ mA}$  at the common emitter configuration.(SFG)

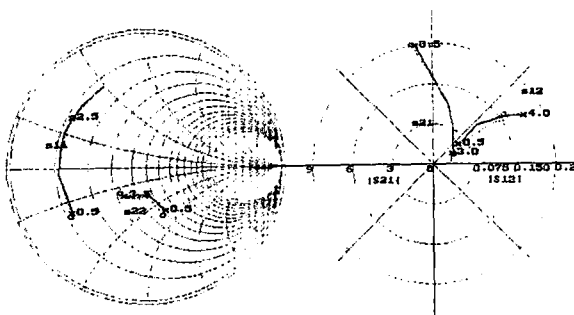


Fig.6 Predicted and measured S-parameter-frequency variations for  $V_{CE}=10\text{ V}$   $I_C=10\text{ mA}$  at the common emitter configuration.(WFBG)

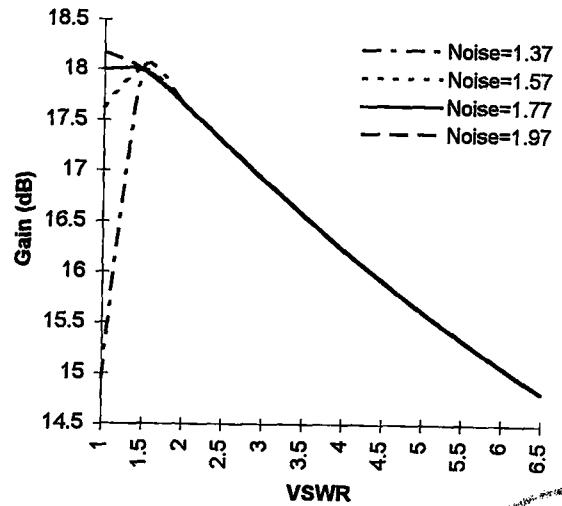
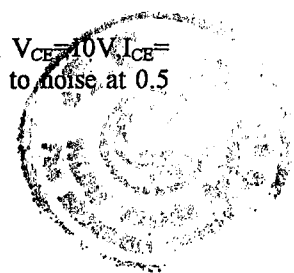


Fig.8 NE02135For the bias condition  $V_{CE}=10\text{ V}$   $I_C=5\text{ mA}$ , gain against VSWR with respect to noise at 0.5 GHz for the transistor NE02135.



REFERENCES

- [1] GÜNEŞ,F.,ÇETİNER B.A., "A NOVEL SMITH CHART FORMULATION OF THE POSSIBLE  $(F, V_i, G_{Tmax})$  TRIPLETS", submitted for ECCTD'97.
- [2] GÜNEŞ F., GÜRGEN F., TORPİ H., " SIGNAL-NOISE NEURAL NETWORK MODEL FOR ACTIVE MICROWAVE DEVICE" , IEE Proceedings-Circuits Devices and Systems Vol.143,No.1,,February 1996,pp.1-8.

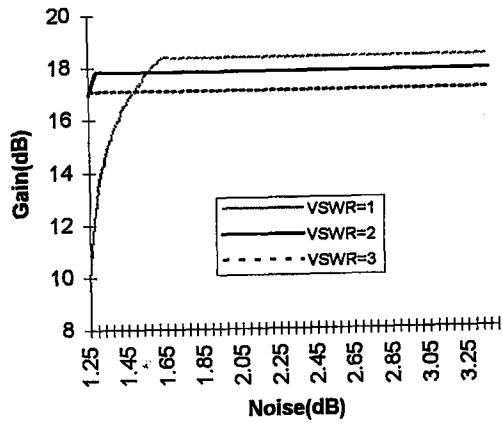


Fig.9 For the bias condition  $V_{CE}=10V, I_{CE}= 5mA$  gain against noise with respect to VSWR at 0.5 GHz for the transistor

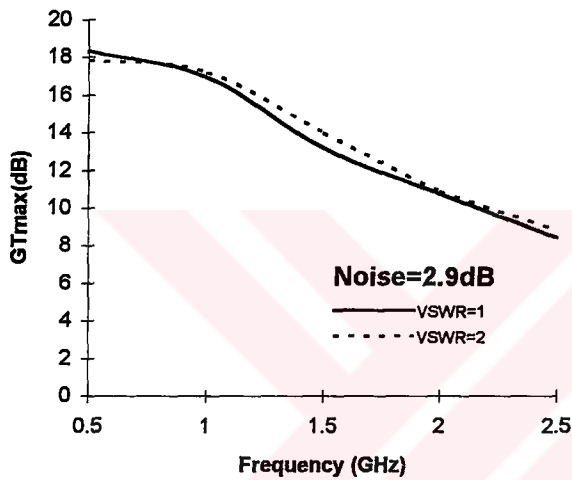


Fig.10 Variations of the gain against operating frequency for a bias condition  $V_{CE}=10V, I_{CE}= 5mA$  for the transistor NE02135.

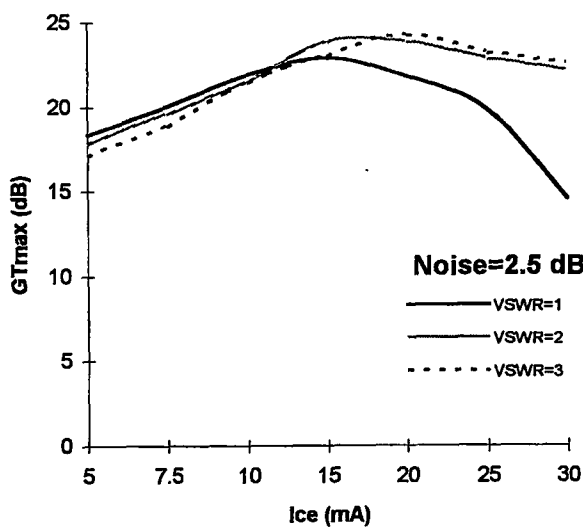
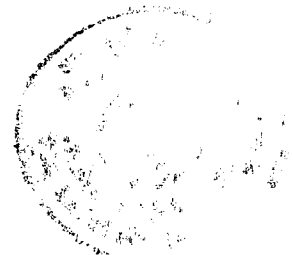


Fig.11 Variations of the gain against bias current,  $I_{CE}$  , at a fixed frequency  $f=0.5$  GHz, for the transistor NE02135.



## GÜRÜLTÜ VE SİNYAL PARAMETRELERİNİN YAPAY SİNİR AĞI İLE SİMÜLASYONU

Hamid TORPİ\*\* Filiz GÜNEŞ\*\* Fikret GÜRGEN\*

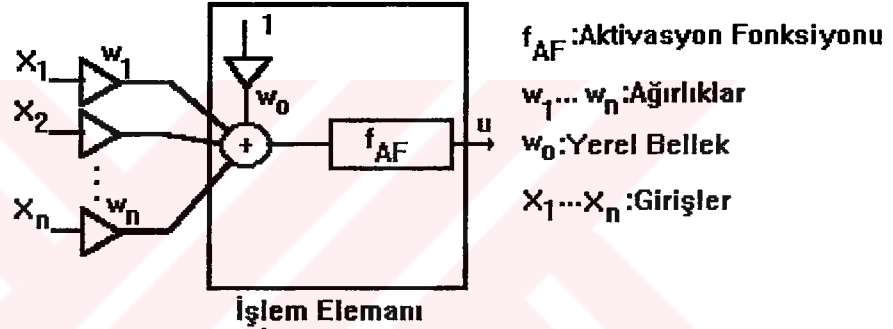
\*\* Yıldız Teknik Üniversitesi , Elektronik ve Haberleşme Müh. Böl., 80670 MASLAK-İSTANBUL/TÜRKİYE

\*Boğaziçi Üniversitesi, Bilgisayar Müh. Böl. ,BEBEK-İSTANBUL/TÜRKİYE

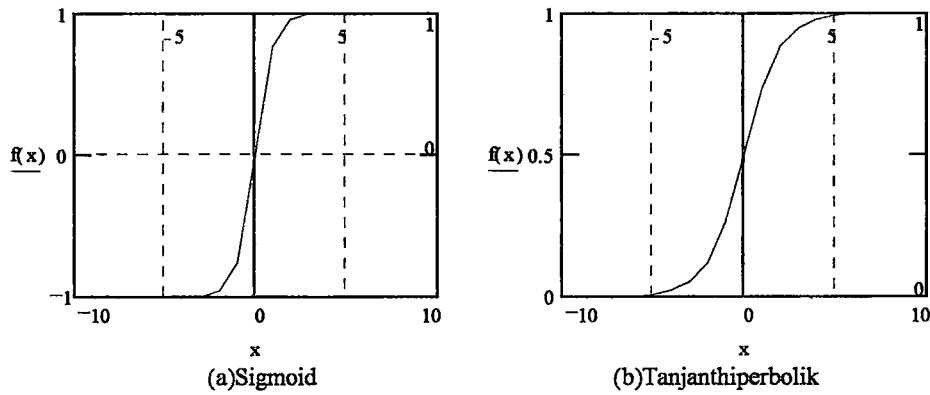
**ÖZET:** Bu çalışmayla eşdeğer devre modeli kullanmadan ve sonuçta doğrudan mikrodalga devre teorisi bilgisini gerektirmeden gürültü ve sinyal parametreleri, direkt olarak ölçülmüş gürültü ve sinyal parametrelerinden yararlanılarak her hangi bir frekans için ayrı ayrı veya ikisi de bir arada olmak üzere oldukça mükemmel bir hassasiyetle (düşük hatayla ) eşdeğer bir yapay sinir ağı (YSA), ile temsil edilmiştir. Bu amaç için belirli bir band içinde alınan muhtelif örneklerden oluşan band hedef uzayı teşkil edilmiştir. Daha sonra da bu hedef uzayı kullanarak YSA 'nın eğitilmesi (Supervised training) sağlanmıştır. Böylelikle YSA 'ı eğitildikten sonra hedef uzayını teşkil eden örneklerin dışındaki frekanslarda da gürültü ve sinyal parametreleri ayrı ayrı veya her ikisinde bir arada olmak üzere oldukça iyi bir performansta kestirilmiştir. Netice olarak eleman modellemede ilk olarak eşdeğer devre modellerinin yerine yapay sinir ağlarını kullanılması önerilmektedir.

### 1-YSA EŞDEĞERLİK TEMELLERİ

YSA'larında temel eleman işlem elemanları olup , bu işlem elemanlarının belirli bir düzen dahilinde değişken ağırlıklarla birbirlerine bağlanması ile teşkil edilmiştir. Her işlem elemanında aktivasyon fonksiyonu ile temsil edilen bir transfer fonksiyonu ve yerel belleği vardır. (Bakınız Şekil-1)[1]



Şekil 1'deki f transfer fonksiyonu olarak sigmoid ve tanjant hiperbolik tip nonlineer fonksiyonlar kullanılmıştır.(Şekil-2) Sigmoid tip çıkışlar 0 ila 1 arasında normalize olduğuna göre hedef uzayına normalizasyon işlemi uygulanmalıdır.



Şekil-2 Kullanılan Aktivasyon Fonksiyonları

Bu çalışmada sürekli değerli girişe sahip , ileri beslemeli bir yapay sinir ağı olan çok katmanlı algılayıcı(MLP) kullanılmıştır. Eğitim algoritması olarak da geriye yayılım algoritması (BPA) kullanılmıştır. Geriye yayılım eğitim algoritması çok katmanlı algılayıcı'nın aktif çıkışıyla arzu edilen çıkış (Hedef uzayla belirlenen) arasındaki karesel ortalama hatanın minimize edilmesine dayanan bir gradyant (Türev temelli arama) algoritmasıdır. [2] Dolayısıyla bu teknik sürekli türevi alınabilir nonlineer transfer fonksiyonuna ihtiyaç gösterir. (Sigmoid de olduğu gibi) Öğrenme işlemi

BPA ile , ağırlıklar ayarlanarak gerçekleştirilir.

Lipmann [2], çalışmasında doğru ile ayrılabilen bölgelerin tanıtılabilmesi için bir, konveks bölgelerin tanıtılabilmesi için iki, en karmaşık karar bölgelerinin tanıtılabilmesi için üç katmanlı algılayıcı 'un yeterli olduğunu göstermektedir.

Bu çalışma için geliştirilen YSA için giriş katmanı, bir gizli katman ve çıkış katmanı olmak üzere iki katmanlı algılayıcı olarak adlandırılan yapının yeterli olduğu görülmüştür. Ağırlıkların başlangıç değerleri seçiminin, sonucu etkilemediği gözlenmiştir, dolayısıyla rasgele alınmıştır.

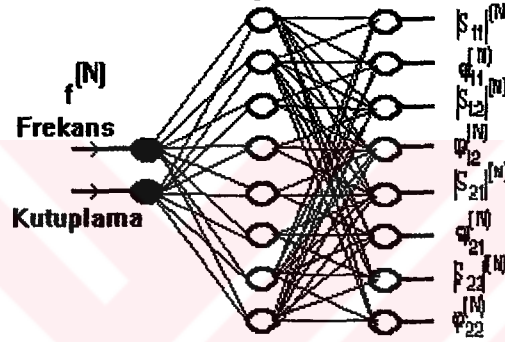
Geliştirdiğimiz YSA 'ı ile istenilen frekansta gürültü [F] yada sinyal [S] parametreleri veya her ikisini birden tayin edebilmektedir. Sonucu daha hızlı yakması için ilave bir kutuplama ikinci bir giriş olarak

$$Kutuplama = \sqrt{\frac{1}{O.S} \sum_{i=1}^{O.S} f_i^2} \quad \xi = i. \text{Örnek frekansı}, O.S : \text{Örnek Sayısı}$$

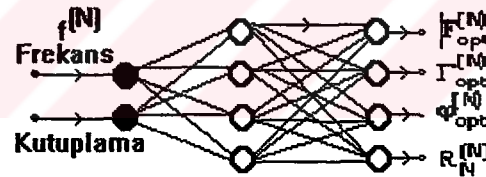
kullanılmıştır.

Bu şekilde kutuplama noktası adı altında ikinci bir giriş ilave etmek öğrenme hızını arttırmıştır. Örneğin 1,2,3,4,5,6,8,10,12,14 GHz noktalarındaki S parametrelerinden oluşan hedef uzayıyla yapılan sabit iterasyon sayılı eğitime için , bütün frekanslarda sabit olan ikinci bir giriş olarak yukarıdaki formülle hesap edilen değer (7.714) ilave kutuplamasıyla kullanılarak bulunan hata yarıya indirilmiştir.

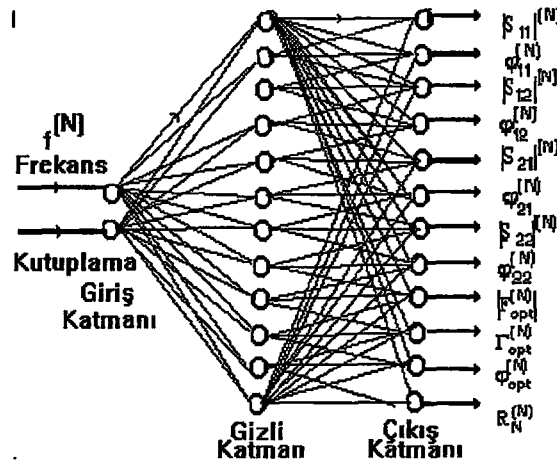
### 1.1 Simülasyonda Kullanılan Çok Katmanlı Algılayıcı Yapıları



Şekil-3 Sinyal parametrelerini öğretmek için kullanılan MLP yapısı



Şekil-4 Gürültü parametrelerini öğretmek için kullanılan MLP yapısı



Şekil-5 Hem gürültü hem de sinyal parametrelerini öğretmek için kullanılan MLP yapısı

Zaman-hata optimum ilişkisini sağlamak için her bir yapıda çıkış katmanı düğüm sayısı gizli katman düğüm sayısına eşit alınmıştır.



N72000aFET için öğrenme katsayısı epsilon=0.5 momentum katsayısı Alpha=0.8 için en optimum öğrenme gerçekleşmiştir. Burada özellikle momentum teriminin önemine değinmek gerekiyor. Zira öğrenme katsayısı küçüldükçe öğrenme işlemi ( zaman açısından ) yavaşlar fakat hassas bir öğrenme ( doğruluk artar) gerçekleşir. Momentum katsayısı (0<Alpha<1) fikri bu noktadan hareketle ortaya atılmıştır. Momentum mevcut delta ağırlığı

$$E_T = \frac{1}{4} \sum_{i=1}^4 E_{i_{\text{signal}}} \quad F_T = \frac{1}{3} \sum_{i=1}^3 E_{i_{\text{noise}}}$$

$$E_{S_{ij}} = \frac{1}{n} \sum_{k=1}^n \frac{|S_{ij_{\text{meas}}}^k - S_{ij_{\text{meas}}}^k|}{|S_{ij_{\text{meas}}}^k|} \quad E_{N_i} = \frac{1}{n} \sum_{i=1}^n \frac{|N_{i_{\text{meas}}}^k - N_{i_{\text{predict}}}^k|}{|N_{i_{\text{meas}}}^k|}$$

üzerinden önceki delta ağırlığının belli bir kısmını besler. Böylece daha düşük öğrenme katsayısı ile daha hızlı öğrenme temin edilir. [ 2]

## 1.2 Başarım Ölçütü

Kestirilen değer uzayının hedef uzayına yaklaşım ölçüsünü aşağıdaki kriterlerle tanımlamak uygun bulunmuştur:

Burada n örnek sayısıdır.  $E_{S_{ij}}$  sinyal parametreleri için  $E_{N_i}$  gürültü parametreleri için ortalama bağıl hatayı göstermektedir.  $N_i$  sırasıyla  $F_{\text{min}}$ ,  $\Gamma_{\text{opt}}$  ve  $R_N$  alınmalıdır.  $E_T$  Sinyal parametreleri,  $F_T$  gürültü parametreleri için toplam hatadır.

## 2 SİMÜLASYON SONUÇLARI

Bu açıklamalardan sonra geliştirdiğimiz bu üç ayrı tipteki YSA simülasyondan elde ettiğimiz neticeler aşağıdaki tabloda verilmiştir. En iyi sonucun alındığı örnek uzayı için ise yukarıda verdiğimiz kriterlere dayanılarak yapılan hesaplamalar Tablo-2'de verilmiştir.

Tablo-1 N72000a transistörünün MLP ile simülasyonundan elde edilen toplu neticeler.

(Hesaplanan ve arzulanan İşaret ve Gürültü parametrelerinin normalize edilmiş olduğuna dikkat ediniz.)

Bias Point=3.317

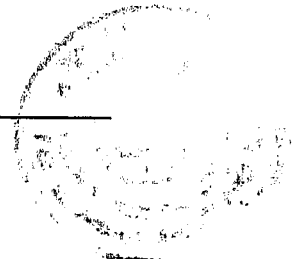
Freq. GHz	Calculated S Parameters & Noise Parameters											
	S11	S21	S12	S22	Fmin	ropt	Rn/50					
1.000	0.984	0.942	0.294	0.450	0.209	0.214	0.727	0.966	0.107	0.871	0.044	0.647
2.000	0.958	0.887	0.290	0.412	0.387	0.181	0.705	0.936	0.130	0.779	0.101	0.511
3.000	0.924	0.836	0.273	0.369	0.532	0.153	0.672	0.908	0.164	0.711	0.145	0.469
4.000	0.882	0.790	0.250	0.330	0.651	0.124	0.638	0.884	0.203	0.660	0.177	0.449
5.000	0.838	0.751	0.229	0.298	0.735	0.101	0.608	0.864	0.240	0.620	0.201	0.434

Freq. GHz	Desired S Parameters & Noise Parameters											
	S11	S21	S12	S22	Fmin	ropt	Rn/50					
1.000	0.982	0.942	0.294	0.450	0.210	0.214	0.728	0.967	0.110	0.870	0.042	0.650
2.000	0.961	0.886	0.291	0.411	0.385	0.181	0.703	0.936	0.120	0.780	0.100	0.500
3.000	0.923	0.836	0.272	0.369	0.535	0.153	0.673	0.908	0.174	0.710	0.147	0.480
4.000	0.881	0.792	0.251	0.331	0.650	0.125	0.640	0.883	0.200	0.660	0.178	0.450
5.000	0.839	0.750	0.229	0.297	0.735	0.100	0.606	0.864	0.240	0.620	0.200	0.430

iteration= 200000 eps= 0.002416 error\_1= 0.000023 error\_t= 0.000023

Freq. GHz	Calculated S Parameters								Desired S Parameters							
	S11	S21	S12	S22	S11	S21	S12	S22								
1.000	0.982	0.939	0.294	0.450	0.212	0.214	0.728	0.964	0.982	0.942	0.294	0.450	0.210	0.214	0.728	0.967
2.000	0.959	0.889	0.289	0.411	0.381	0.182	0.704	0.937	0.961	0.886	0.291	0.411	0.385	0.181	0.703	0.936
3.000	0.923	0.836	0.273	0.370	0.537	0.150	0.673	0.908	0.923	0.836	0.272	0.369	0.535	0.153	0.673	0.908
4.000	0.882	0.790	0.251	0.330	0.652	0.124	0.639	0.884	0.881	0.792	0.251	0.331	0.650	0.125	0.640	0.883
5.000	0.839	0.751	0.229	0.297	0.733	0.103	0.607	0.864	0.839	0.750	0.229	0.297	0.735	0.100	0.606	0.864

iteration= 200000 eps= 0.001248 error\_1= 0.000006 error\_t= 0.000006



Tablo-1 N72000a transistörünün MLP ile simülasyonundan elde edilen toplu neticeler.(Devamı )

Freq. GHz	Calculated F Parameters			Desired F Parameters		
	Fmin	ropt	Rn/50	Fmin	ropt	Rn/50
1.000	0.107	0.871	0.043	0.648	0.110	0.870
2.000	0.130	0.778	0.102	0.511	0.120	0.780
3.000	0.164	0.712	0.144	0.468	0.174	0.710
4.000	0.204	0.659	0.177	0.448	0.200	0.660
5.000	0.239	0.621	0.201	0.436	0.240	0.620
iteration= 250000 eps= 0.004254 error_1= 0.000072 error_t= 0.000072						
learn & test finished at time alpha =0.800 epsilon =0.50						
n72000a Transistoruna ait eğitim sonuçları yukarıda verilmiştir.						

Tablo-2 250000 adet iterasyon için N72000A transistörüne ait 1,2,3,4 ve 5 GHz'de alınan örneklerle teşkil edilen hedef uzayıyla kestirilen değer uzayı arasındaki farkın hata kriterleri yardımıyla elde edilen neticeler.

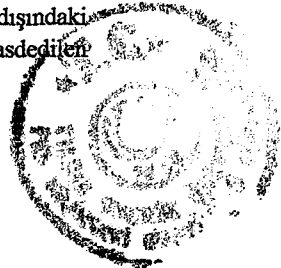
	Şekil-3'deki Yapı için	Şekil-4'deki Yapı için	Şekil-5'deki Yapı için
Gerçek Mutlak Toplam Hata	0.006819	0.006961	0.016462
E11	0.010758	0.004621	-
E12	0.012340	0.006393	-
E21	0.003366	0.003825	-
E22	0.004638	0.002851	-
ET	0.007776	0.004422	-
F1	-	0.036043	0.037645
F2	-	0.007320	0.009009
F3	-	0.010846	0.012717
FT	-	0.018070	0.019790

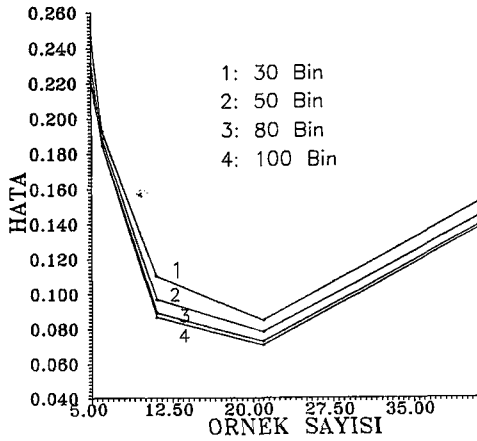
## 2.1 Simülasyon Sonuçlarının İrdelenmesi

Tablo-1 incelendiğinde görülmüyor ki YSA'nın gösterdiği performans açısından Normalize hata teriminden hareketle en iyi sonuçlar Şekil-3'deki ([S]) yapı için sonra da Şekil-4'deki yapıyla elde edildiği görülmüyor.(Bak.Tablo-2) Bu sonuçlar normalize değerler gözönüne alındığında elde edilmiştir.Bütün parametreler için gerçek değerler düzleminde (ki bizim için önemli olan budur) durum ayrıntılı ( yani hem sinyal hemde gürültü parametrelerinin hataları ayrı ayrı gözönüne alınarak) olarak incelenirse Şekil-5 'deki yapıyla sinyal parametrelerinin sonuç olarak en iyi performansla ( en az hatayla ) öğretilmediği görülmüştür. Örneğin 1,2,3,4,5 GHz'de alınan örneklerle teşkil edilen hedef uzayının eğitilmesiyle elde edilen sonuçta Şekil-5'deki yapı için  $E_T=0.0044$  (Bak. Tablo-2) Şekil-3'de ki YSA yapısı için  $E_T=0.0077$  (Bak. Tablo-2) lik hatayla sonuçlar elde edilmiştir.(Bak. Tablo-2) Ve yine daha belirgin bir örnek olarak 1,2,3,4,5,6,8 GHz 'de alınan örneklerle teşkil edilen hedef uzayının öğretilmesi ile elde edilen sonuçta Şekil-5 'deki yapı için  $E_T=0.0064$ ,Şekil-3'deki yapı için ise  $E_T=0.01289$ (%1.29)'lık hatalarla öğretilmiştir. Sonuç olarak Şekil-5'deki yapıyla yani Gürültü ve Sinyal parametrelerinin birlikte öğretilmeye çalışıldığı durumda sinyal parametreleri daha iyi eğitilebilmiş gürültü parametrelerinin tek başına eğitildiği yapıyla elde edilen sonuçlardan biraz daha kötü olmasına rağmen onlara yaklaşmıştır.Bu durumda sinyal ve gürültü parametrelerinin eğitilmesi için Şekil-5'deki YSA yapısından yararlanılarak sonuca gitmeye çalışmak daha avantajlı olduğunu söyleyebiliriz. Ama biz sadece Gürültü veya sadece Sinyal parametrelerini öğretmek istiyorsak Şekil-3 ve Şekil-4 'deki YSA yapılarını tercih etmek ( Zaman açısından da ayrıca avantajlı) muhakkak ki daha avantajlıdır. Bu sonuçlar [4] 'da belirtilen sinyal ve gürültü parametrelerinin ortak eşdeğer bir devreden hareketle hesaplanmasıyla daha iyi bir model kurulacağını saptayan makaleyle elde ettiğimiz sonuç itibarıyla bir paralellik arz etmektedir.

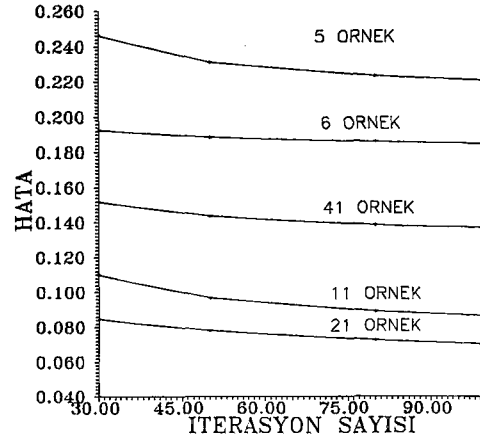
## 3 HEDEF UZAYININ BELİRLENMESİ VE GENELLEŞTİRME

Bu bölümde tüm bandı öğretebilmek için acaba kaç örnek alınarak (teşkil edilen hedef uzayının öğretilmesi) en az hatayla eğitim yapılabilir? Sorusuna cevap arayacağız. Ayrıca eğitmenin yapıldığı örneklerin dışındaki frekanslarda geliştirilen MLP'nin davranışı bizim uygulamamız için en önemli kriterdir.Genelleştirme ile de kastedilen budur.





Şekil-6 Örnek sayısı değişken iterasyon sayısı parametre olmak üzere sinyal parametrelerine ait toplam bağıl hatayı göstermektedir.



Şekil-7 İterasyon sayısı değişken hedef uzayları parametre olmak üzere sinyal parametrelerine ait toplam bağıl hatayı göstermektedir.

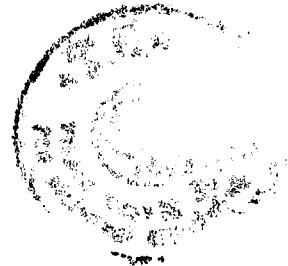
Bu bölümde elimizdeki bilgisayar alt dizinlerindeki (Directory) transistör örnekleri içinde en fazla ölçülmüş değere sahip olan N32684AA FET baz alınmıştır. Zira ara frekanslarda devrenin davranışının test edilebilmesi için mümkün olduğu kadar çok ölçülmüş örneğin elimizde bulunması gerekir. Bunun için sadece Sinyal parametreleri ele alınmış yani sadece Şekil-4'deki yapı kullanılmıştır. Bu sefer  $\alpha = 0.1$  olarak değiştirilmiştir. Epsilon öğrenme katsayısı değişmemiştir.

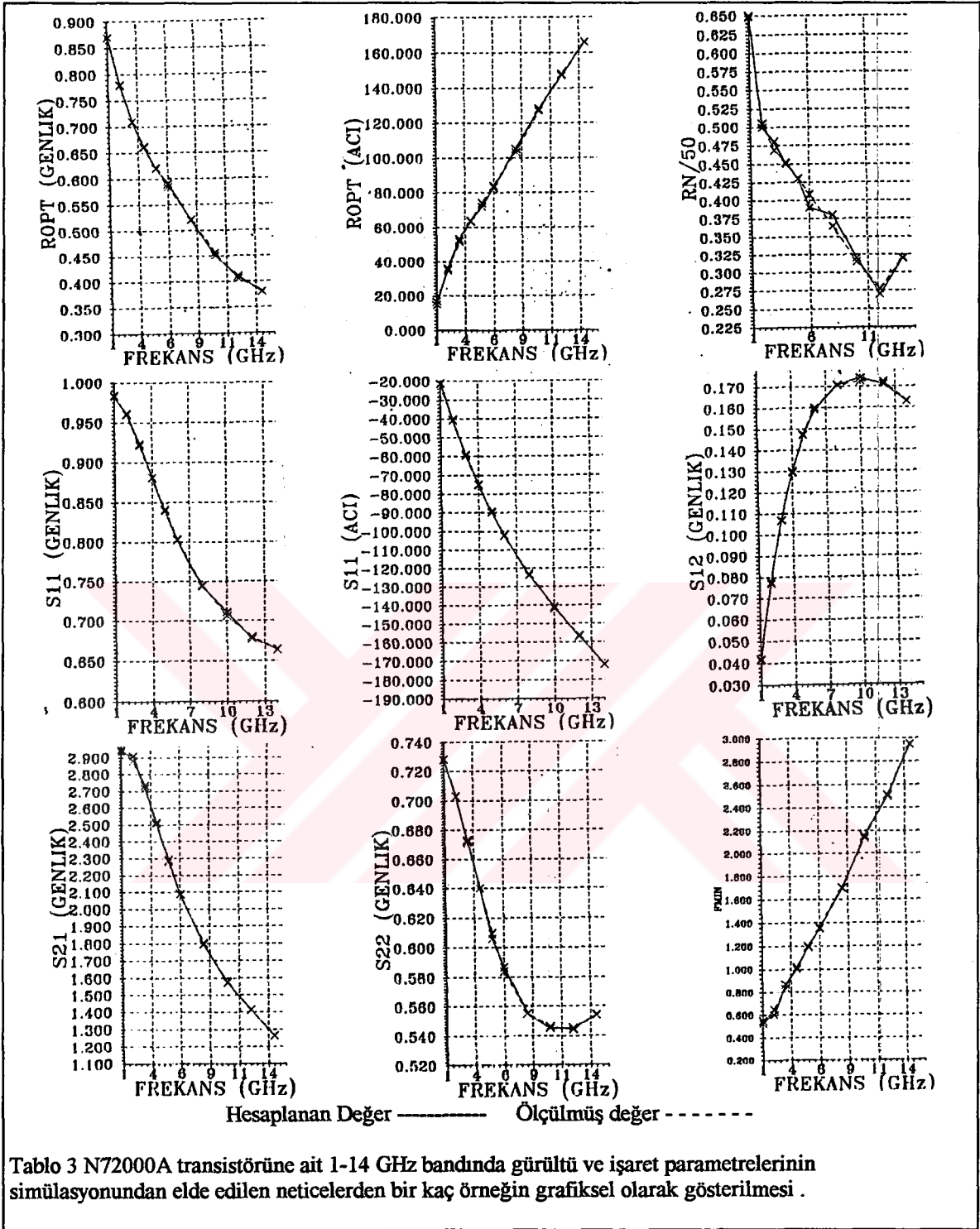
N32684AA FET 'i tüm kullanma bandında eğitmek amacıyla sırasıyla 0.5, 1, 2, 4 ve 5 GHz aralıklarla alınan düzgün örneklerler sırasıyla 41, 21, 11, 6 ve 5 örnekle teşkil edilen hedef uzayları öğretilmiştir. Eğitim sonunda elde edilen neticeler tüm çalışma bandı için test edilmiştir. Elde edilen neticeler Şekil-6'da Sinyal parametreleri için toplam bağıl hata  $E_T$  düşey eksen örnek sayısı yatay eksen ve iterasyon sayısı parametre olmak üzere çizdirilmiştir. Ayrıca Şekil-7'de iterasyon sayısı değişken hedef uzayları parametre olmak üzere sinyal parametrelerine ait toplam bağıl hata  $E_T$  hesap ettirilerek çizdirilmiştir.

Elde edilen tüm bu sonuçlar değerlendirildiğinde görülmüştür ki 21 örnekle (1 GHz aralıklarla örnekler alınan) belirlenen hedef uzayının kullanılması ile tüm bandın en az hatayla MLP tarafından doğru bir şekilde temsil edilebileceği görülmüştür. Bunu 11 örnekle (2 GHz aralıklarla) hedef uzayının kullanıldığı kullanılması ile eğitilen MLP takip etmektedir. Bu sonuçlar bize bizim örneğimizdeki MESFET 'i ( Yada herhangi bir Sinyal parametreleri ile temsil edilebilen mikrodalga elemanını ) tüm kullanma bandı boyunca Sinyal parametreleri açısından temsil edecek MLP 'un eğitilmesi için optimum bir örnekleme sayısının olduğunu ve eldeki bütün ölçülmüş örneklerle tüm bandı öğretmeye çalışmanın yanlış olduğunu göstermektedir.

## 5 SONUÇ VE TARTIŞMA

İki katmanlı algılayıcı ile gayet iyi bir yaklaşıklıkla kullanılan transistörün sinyal ve gürültü parametrelerini temsil edilebileceğini geliştirilen üç ayrı simülasyon programı ile gösterilmiştir. Transistörün kullanılabileceği tüm band boyunca dahi transistörün sinyal ve gürültü parametrelerini yüzde birkaçlık hatayla temsil edilebileceği görüldü. Yalnız daha az hata ile gürültü ve sinyal parametrelerinin öğretilmesi isteniyorsa (Binde bir'ler mertebesinde) bandın daha dar tutulması gerekmektedir. Bütün değerlerin 0 ile 1 arasında transfer edilmesi durumunda aktivasyon fonksiyonu tanhiperbolik seçilmesi durumu için iyi bir öğrenme gerçekleşmemiştir. Bu aktivasyon fonksiyonu için gürültü ve sinyal parametrelerinin -1 ile 1 arasında transfer edilerek hedef uzayının belirlendiği durum için [5] yerine göre çok daha iyi neticeler vermiştir. Yalnız bir çok durumda yerel minimuma yakalanmakta bir sonuç alınamamaktadır. Bir çok durumda ise sonuç alabilmek için  $\alpha$  değeri -1 ile 0 arasında aldığımızda iyi neticeler alınabilmektedir. Bu tür uygulamalar için daha iyi sonuç veren [1] "radial basis function" olarak adlandırılan aktivasyon fonksiyonu olarak Gauss dağılımını kullanan tip deki YSA'ları içinde henüz çalışmalarımız devam etmektedir. Sonuç olarak aktif mikrodalga devreleri için geliştirdiğimiz sinir ağı modeli gayet iyi bir performans göstermiştir.





## KAYNAKLAR

- [1] Don R.HUSH,Bill G.HORNE, IEEE Signal Processing Magazine,pg 8-22, January 1993 .
- [2] Richard P.LIPPMANN, IEEE ASSP Magazine, pg 5-38,April 1987.
- [3] Bekir KARLIK,'Çok Fonksiyonlu Protezler için Yapay Sinir Ağları Kullanılarak Miyoelektrik Kontrol ',YTÜ FBE Doktora Tezi,İstanbul TURKEY,Mart 1994.
- [4] Z.R.HU,Z.M.YANG,V.F.FUSCO,J.A.C.STEWART,IEE Proceedings-G Vol.140 ,No.1,February 1993.
- [5] PALUBINKAS G.,Pattern Recognition Letters,1994. "



# AKTİF MİKRODALGA ELEMANLARININ YAPAY SİNİR AĞI SİMÜLATÖRLERİ

Hamid TORPİ\*\*

Filiz GÜNEŞ\*\*

Fikret GÜRGEN\*

\*\* Yıldız Teknik Üniversitesi , Elektronik ve Haberleşme Müh. Böl., 80670 MASLAK-İSTANBUL/TÜRKİYE

\*Boğaziçi Üniversitesi, Bilgisayar Müh. Böl. ,BEBEK-İSTANBUL/TÜRKİYE

**Özet:** Mikrodalga transistörlerinin küçük-ışaret ve gürültü başarımının modellenmesi için yeni bir yöntem teklif edilmektedir. Bu yöntemde , transistör ışıaret ve gürültü parametreleri çalışma bandı boyunca , bir yapay sinir ağı (YSA) ile ölçüm uzayıyla uyuşacak şekilde hesaplanan bir kara kutuyla modellenmiştir. Bu modelleme yöntemi kullanılarak, sadece ışıaret parametrelerinin değil , aynı zamanda gürültü parametrelerinin de ölçüm uzayıyla , literatürdeki model sonuçlarından daha iyi bir uyuşum içerisinde olduğu bulunmuştur.

## 1-GİRİŞ

Bir mikrodalga transistörünün bir kutuplama noktası civarındaki küçük ışıaret ve gürültü davranışları çalışma bandı boyunca sırasıyla saçılma parametreleri  $S_{11}, S_{22}, S_{21}, S_{12}$  ve gürültü parametreleri  $F_{min}, \Gamma_{opt}, R_N$  ile tayin edilebilir. S parametreleri ve gürültü parametrelerinin her ikisi de frekansa ve elemanın yapısal özelliklerine bağlıdır.

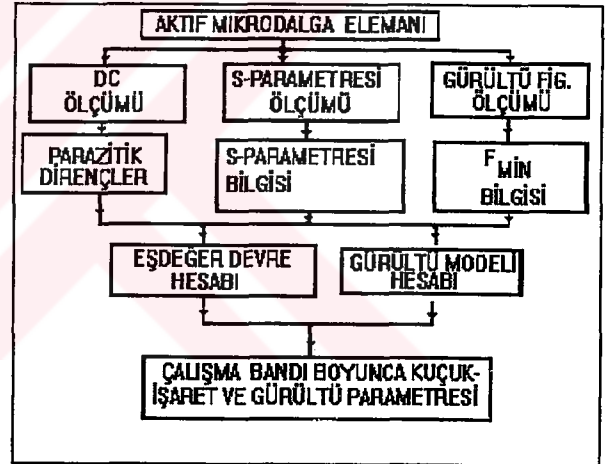
Şekil-1'de gösterildiği gibi gürültü parametrelerinin de küçük ışıaret model parametreleri gibi küçük ışıaret modelleme bilgisine gereksinimi vardır. Çünkü her iki parametre seti de elemanın fiziksel ve geometrik parametrelerine, örneğin GaAs MESFET'te aktif katmanın geçit uzunluğuna sıkı sıkıya bağlıdır. Bu açıdan bakılacak olursa, eşdeğer devre elemanı küçük-ışıaret ve gürültü parametrelerinden yararlanmak suretiyle belirlenmelidir. Mevcut literatürde /1-4/ sadece küçük-ışıaret modeli ile ilgilenilmekte ve küçük-ışıaret eşdeğer devrelerinin elemanın gürültü karakteristiğiyle ilişkisi olmadığı üzerinde yoğunlaşmaktadır. Bizim durumumuzda eşdeğerlik , ışıaret [S] ve gürültü parametrelerinin [N] her ikisini de birlikte kullanmak suretiyle belirlenmiştir. Bu çalışmanın başlıca amaçları aşağıdaki şekilde sıralanabilir.

(i) Eşdeğerlik işlevi görecektir, tek gizli katmanlı , ileri beslemeli tipte yapay sinir ağını kurmak,

(ii) Geriye yayılım algoritması (BPA) ve lineer olmayan tipte aktivasyon fonksiyonları kullanarak, yapay sinir ağını herhangi bir tipte aktif eleman için çalışma bandı boyunca ışıaret-gürültü davranışının her ikisi için de kullanarak, eğitmek,

(iii) Yapay sinir ağı performans ölçüsünü belirlemek,

(iv) Seçilen elemanın lineer olmayan karakteristiğinin kutuplama noktası civarındaki parçasına fonksiyonel yaklaşım yaparak eğitilmiş yapay sinir ağını kullanarak herhangi bir frekans için elemanın ışıaret-gürültü davranışını tayin etmek.



Şekil-1 Aktif Mikrodalga Elemanlarının Küçük-İşaret ve Gürültü Performansının Hesaplanması için Genel Prosedür.

## 2-YAPAY SİNİR AĞI EŞDEĞERİ

Çalışmada , transistörün çalışma bandı boyunca küçük-ışıaret ve gürültü davranışı simülasyonunda , çıkışıyla eşit sayıda işlem elemanına sahip tek gizli katmanlı bir algılama yapısı (ÇKA) yeterli bulunmuştur. (Şekil-2) YSA'nın yakınsama hızını artırmak amacıyla ilave bir kutuplama girişi kullanılmıştır.

$$Kutuplama = \sqrt{\frac{1}{N_s} \sum_{i=1}^{N_s} f_i^2}$$

(1)

Burada  $f_i$ , i. örnek frekansı  $N_s$  örnek sayısıdır. Şekil -2 'deki Çok Katmanlı Algılama yapısını (ÇKA=MLP), gizli katmanından i. çıkış elemanına gelen  $\Phi_i$  işareti

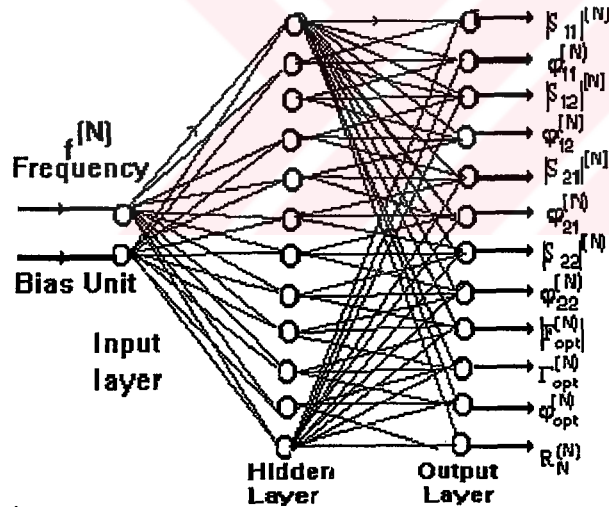
$$\Phi_i(\mathbf{T}_i, \mathbf{W}, \mathbf{x}) = \sum_{h=1}^{N_h} T_{hi} g_h(\mathbf{W}_h, \mathbf{x}) + T_{ho} \quad (2.1)$$

ifade edilebilir ve aynı düğümden net çıkış olarak aşağıdaki biçimde elde edilebilir.

$$\phi_i(\mathbf{T}_i, \mathbf{W}, \mathbf{x}) = T_{oi} g_i(\Phi_i) + T_i \quad (2.2)$$

Burada  $g_h$  ve  $g_i$ , sırasıyla h. gizli ve i. çıkış işlem elemanlarının nonlinear aktivasyon fonksiyonlarıdır. Çalışmada yaygın iki tip dağılmış devre baz fonksiyonu sigmoid tip ve tanjant hiperbolik tip kullanılmış kararlı nümerik hesaplama performansı nedeniyle sigmoid tip aktivasyon fonksiyonu tercih edilmiştir:

$$g_h(\mathbf{W}_h, \mathbf{x}) = \frac{1}{1 + \exp\left(-\sum_i x_i W_{ih} - W_{oh}\right)} \quad (3)$$



Şekil-2 Aktif Mikrodalga Elemanı için ÇKA yapısı

Burada  $W_{oh}$  h. işlem elemanının yerel belleğidir.. (4.1), (4.2) ve (5) deki  $\mathbf{x}$  giriş vektörüdür.

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^T \quad (4.1)$$

$T_i$  i. çıkış işlem elemanı ile gizli katman

arasındaki ağırlık vektörü de aşağıdaki biçimde ifade edilebilir.

$$\mathbf{T}_i = [T_{1i} \ T_{2i} \ T_{3i} \ \dots \ T_{hi} \ \dots \ T_{N_h i}]^T \quad (4.2)$$

$\mathbf{W}$ , gizli katman ile giriş katmanı arasındaki ağırlık matrisidir.

$$\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2 \ \dots \ \mathbf{W}_h \ \dots \ \mathbf{W}_{N_h}]^T \quad (4.3)$$

$T_{ho}$  ve  $T_i$  sırasıyla h. gizli ve i. çıkış işlem elemanlarının eşikleridir.

YSA'nın öğrenme mekanizması kısaca özetlemek amacıyla bir çıkış elemanının net çıkışının  $F(\mathbf{P}, \mathbf{x})$  formunda ifade edelim F, YSA mimarisi tarafından belirlenir ve YSA mimarisi kapsamında katman sayısı, her katmanın işlem elemanı sayısı, elemanlar arası bağlantılar gibi YSA yapı faktörleri içerilir. Bizim uygulamamızda öğrenme işlemi, P YSA bağlantı ağırlık vektörünün, ölçülmüş  $y_i$  ile hesaplanan  $F(\mathbf{P}, \mathbf{x}_i)$  arasındaki fark karelerinin örnekleme üzerinden toplamı şeklinde tanımlanan hata fonksiyonunun minimize edecek değerini tesbit etmeye karşı düşmektedir. Buna göre  $E(\mathbf{P})$  hata fonksiyonu

$$E(\mathbf{P}) = \sum_{\{x_i, y_i\}} [y_i - F(\mathbf{P}, [x_i])]^2 \quad (5.1)$$

şeklinde ifade edilebilir ve başlangıçta (YSA) bağlantılara rastgele ağırlıklar verilir ve her biri  $\frac{\partial E}{\partial P_p}$  ile orantılı biçimde tekrarlayan değişimlere tabi tutulur:

$$\Delta P_p = -\eta \frac{\partial E}{\partial P_p} \quad (5.2)$$

Burada  $\eta$  öğrenme katsayısıdır. (7.1) ile tanımlanan hata fonksiyonunun azalmasının durduğu zaman öğrenme işleminin tamamlandığı varsayılmaktadır. Öğrenme algoritması olarak geriye yayılım algoritması (BPA) kullanılmıştır.

### 3-BAŞARIM ÖLÇÜTÜ

Kestirim uzayının hedef uzayına -ölçülmüş değerler uzayı- yaklaşım ölçüsü olarak aşağıdaki kriterleri tanımlamak uygun

bulunmuştur.

$$E_{S_{ij}} = \frac{1}{n} \sum_{k=1}^n \frac{|S_{ij,meas.}^k - S_{ij,meas.}^k|}{|S_{ij,meas.}^k|} \quad (6.1)$$

$$E_{N_i} = \frac{1}{n} \sum_{i=1}^n \frac{|N_{i,meas.}^k - N_{i,predict.}^k|}{|N_{i,meas.}^k|} \quad (6.2)$$

$$E_T = \frac{1}{4} \sum_{i=1}^4 E_{i,signal} + \frac{1}{3} \sum_{i=1}^3 E_{i,noise} \quad (6.3)$$

Burada n örnek sayısıdır,ES<sub>ij</sub> işaret,EN<sub>i</sub> gürültü parametreleri için ortalama bağıl hatadır. N<sub>i</sub>

sırasıyla F<sub>min</sub>, Γ<sub>opt</sub>, R<sub>N</sub> alınmalıdır. E<sub>T</sub> işaret, F<sub>T</sub> gürültü parametreleri için toplam hatadır.

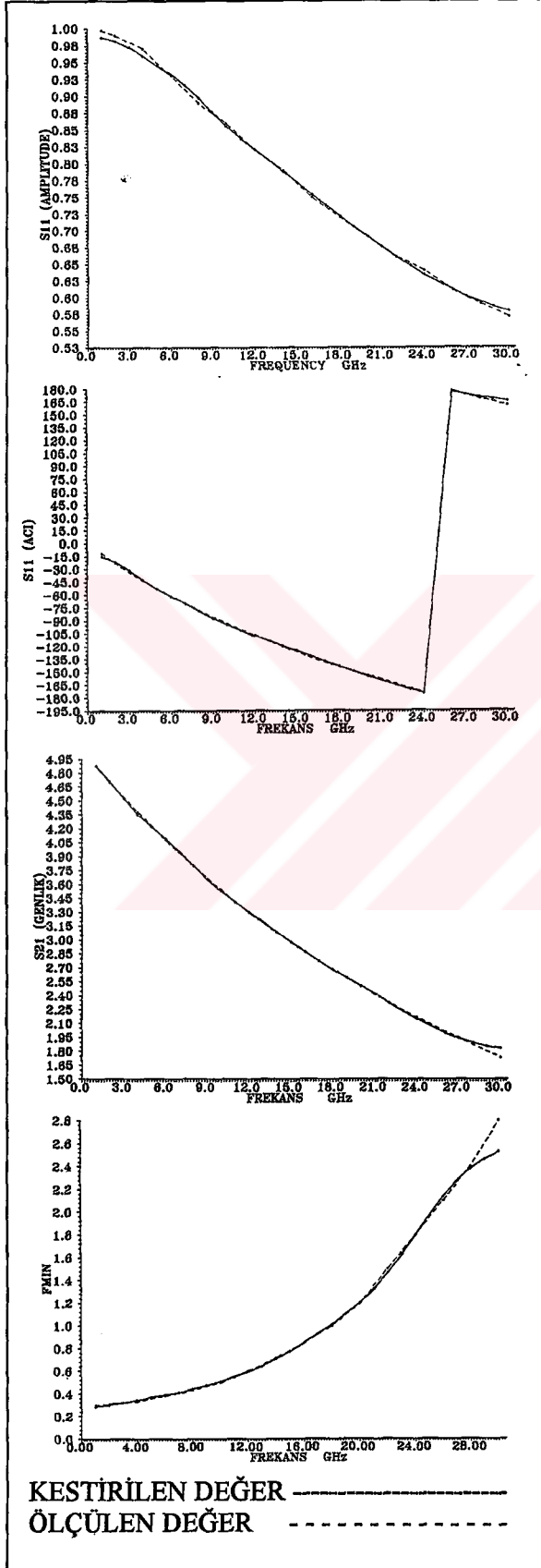
#### 4-SİMÜLASYON SONUÇLARI

Tablo I'de FET N24200A için ölçülmüş işaret ve gürültü parametreleri katalogdan verilmiştir, yine aynı tablo transistörün 1 Ghz'den 30 GHz'e kadar çalışma bandı boyunca YSA Simülatörünün nümerik sonuçları ve başarımlı ölçüleri verilmiştir. Tablo II'de ise muhtelif gürültü ve işaret parametrelerine ait [S<sub>11</sub>(Genlik ve açı), S<sub>21</sub>(Genlik), F<sub>min</sub>] grafikler verilmiştir.

Table I  
N24200A İÇİN YAPAY SİNİR AĞI MODELİ'NE AİT ÇIKIŞLAR

ÖĞRENME İÇİN KULLANILAN İŞARET VE GÜRÜLTÜ PARAMETRELERİ												
f[GHz]	S <sub>11</sub>	S <sub>21</sub>	S <sub>12</sub>	S <sub>22</sub>	F <sub>min</sub>	Γ <sub>opt</sub>	R <sub>N</sub> /50					
1.0	0.997	-12.000	4.880	170.000	0.016	84.000	0.620	-8.000	0.300	0.810	10.000	0.390
2.0	0.990	-23.000	4.700	161.000	0.030	77.000	0.610	-15.000	0.310	0.790	17.000	0.360
4.0	0.970	-44.000	4.380	144.000	0.052	65.000	0.610	-29.000	0.330	0.750	31.000	0.330
6.0	0.930	-62.000	4.080	128.000	0.071	53.000	0.590	-41.000	0.380	0.720	45.000	0.300
8.0	0.890	-79.000	3.800	113.000	0.086	43.000	0.580	-51.000	0.430	0.700	59.000	0.270
10.0	0.860	-94.000	3.540	99.000	0.099	34.000	0.560	-61.000	0.500	0.680	77.000	0.240
12.0	0.820	-108.000	3.300	86.000	0.109	27.000	0.540	-70.000	0.600	0.660	92.000	0.220
14.0	0.790	-121.000	3.080	74.000	0.119	21.000	0.510	-78.000	0.710	0.640	108.000	0.190
16.0	0.750	-134.000	2.870	63.000	0.127	16.000	0.490	-87.000	0.850	0.620	126.000	0.180
18.0	0.720	-145.000	2.680	52.000	0.135	12.000	0.470	-95.000	1.000	0.580	140.000	0.150
20.0	0.690	-155.000	2.500	42.000	0.142	8.000	0.450	-102.000	1.200	0.550	153.000	0.130
22.0	0.660	-165.000	2.320	32.000	0.148	6.000	0.430	-109.000	1.500	0.520	164.000	0.110
24.0	0.640	-175.000	2.160	23.000	0.153	4.000	0.420	-116.000	1.800	0.490	175.000	0.100
26.0	0.610	-177.000	2.010	15.000	0.159	3.000	0.410	-122.000	2.100	0.480	-176.000	0.080
28.0	0.590	168.000	1.870	7.000	0.163	1.000	0.410	-128.000	2.400	0.460	-168.000	0.070
30.0	0.570	160.000	1.730	-1.000	0.168	0.000	0.410	-134.000	2.800	0.460	-160.000	0.050
KESTİRİLEN İŞARET VE GÜRÜLTÜ PARAMETRELERİ												
1.000	0.992	-13.829	4.875	169.717	0.017	83.465	0.617	-9.134	0.297	0.812	10.917	0.386
2.000	0.985	-22.417	4.707	161.419	0.028	77.715	0.614	-14.916	0.305	0.786	16.516	0.365
4.000	0.959	-44.373	4.382	144.512	0.053	65.511	0.606	-29.678	0.333	0.745	30.971	0.328
6.000	0.931	-60.962	4.080	127.820	0.069	53.158	0.593	-40.074	0.380	0.724	44.336	0.297
8.000	0.895	-78.936	3.781	111.545	0.086	42.050	0.577	-51.161	0.441	0.703	60.658	0.267
10.000	0.857	-94.854	3.533	98.415	0.099	33.606	0.560	-61.167	0.508	0.683	76.692	0.242
12.000	0.822	-108.12	3.314	86.946	0.109	26.602	0.539	-69.757	0.590	0.661	91.846	0.219
14.000	0.788	-120.82	3.092	75.322	0.118	20.354	0.514	-78.232	0.701	0.635	107.991	0.195
16.000	0.752	-133.38	2.872	63.605	0.127	15.329	0.488	-86.831	0.844	0.607	124.823	0.172
18.000	0.719	-144.84	2.675	52.244	0.134	11.762	0.465	-94.819	1.012	0.579	140.145	0.151
20.000	0.689	-155.12	2.497	40.936	0.141	9.238	0.448	-102.017	1.214	0.553	153.077	0.132
22.000	0.662	-165.10	2.322	29.859	0.147	7.275	0.434	-109.009	1.475	0.526	164.537	0.113
24.000	0.635	-175.07	2.149	20.548	0.153	5.714	0.422	-116.052	1.795	0.499	175.182	0.095
26.000	0.610	175.998	2.000	14.310	0.158	4.593	0.412	-122.471	2.119	0.474	-175.64	0.081
28.000	0.591	168.947	1.885	15.288	0.162	3.659	0.405	-127.601	2.391	0.455	-168.58	0.070
30.000	0.570	159.9	1.729	-6.5	0.168	1.0	0.410	-134.0	2.801	0.460	-160.0	0.05
Error Analysis												
E <sub>11</sub> =0.011053	E <sub>21</sub> =0.024918	E <sub>12</sub> =0.027162	E <sub>22</sub> =0.009100									
E <sub>T</sub> = 0.018058												
F <sub>1</sub> =0.011198	F <sub>2</sub> =0.012857	F <sub>3</sub> =0.016921										
F <sub>T</sub> = 0.013659												

Tablo II N24200A İÇİN YAPAY NÖRON AĞI  
MODELİ'NE AİT ÖRNEK GRAFİKSEL  
ÇIKIŞLAR



## KAYNAKÇA

- 1/ R.K. FROELICH: "An improved model for noise characterization of microwave GaAs FET's" ,IEEE Trans., 1990, MTT-38 (6), pp. 703-706.
- 2/ M.W. POSPIESZALSKI: "Modelling of noise parameters of MESFET's and MODFET's and frequency and temperature dependence" , IEEE Trans. , 1989, MTT-37, (9), pp 1340-1350.
- 3/ M. BERROTH and R. BOSH: "Broad band determination of the FET small signal equivalent circuits" ,IEEE Trans.1990,MTT-38,(7),pp. 891-895.
- 4/Z.R.HU,Z.M.YANG,V.F.FUSCO,J.A.C.STEWART," Unified small-signal-noise model for active microwave device" , IEE Proceedings-G , Vol.140 , No1 , Feb. 1993 , pp. 55-60.
- 5/G. D. VENDELIN, A. M. PAVIO,U. L. ROHDE, "Microwave Circuit Design using Linear and Nonlinear Techniques" , John Wiley & Sons, 1990.
- 6/F.GÜNEŞ,M.GÜNEŞ,M.FIDAN:"Performance Characterisation of a Microwave Device",IEE Proc. Circuits Devices Syst., Vol.141 ,No.5, pp. 337-344, Oct. 1994.
- 7/F. GÜRGEN, R. ALPAYDIN,U. ÜNLÜAKIN. , E.. ALPAYDIN,"Distributed and Local Neural Classifiers for Phoneme Recognition",Pattern Recognition Letters, pp. 1111-1118, Nov. 1994.
- 8/D.R. HUSH,B.G. HORNE , "Progress in Supervised Neural Networks",IEEE Signal and Processing Magazine, pp-8-36, Jan. 1993.

## ÖZGEÇMİŞLER

Hamid TORPİ:YTÜ Elektronik ve haberleşme Mühendisliği bölümünden 1988'de mezun oldu.1989'dan beri aynı bölümde EM Alanlar ve Mikrodalga Tekniği Bilim dalında Araştırma Görevlisi olarak çalışmaktadır. Halen Doktora tezini hazırlama çalışmalarına devam etmektedir.

Filiz GÜNEŞ:Bilgi için yazarın yayınlanmış diğer bildirisine bakınız.

Fikret GÜRGEN:İTÜ Elektrik-Elektronik Fakültesi Elektronik ve Haberleşme Mühendisliği bölümünden 1981'de mezun oldu.Yazarımız 1983'de USA Ohio State Üniversitesinde Msc. ,1986'da USA Acron Üniversitesinde İşaret İşleme dalında Ph.D. derecesi aldı.1992'de Doç.Dr. ünvanını aldı.Halen Boğaziçi Üniversitesi Bilgisayar Mühendisliği bölümünde öğretim görevlisi olarak çalışmaktadır.



EK 3 İKİ KAPILI PARAMETRELERİ ARASI DÖNÜŞÜM FORMÜLLERİ

Tablo EK 1.1  $Z_0=1$ 'e normalize edilmiş iki kapılı parametreleri dönüşüm formleri ( $\Delta^K=K_{11}K_{22}-K_{12}K_{21}$ )

S	Z	Y	h	A
$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$	$S_{11} = \frac{(z_{11}-1)(z_{22}+1)-z_{12}z_{21}}{(z_{11}+1)(z_{22}+1)-z_{12}z_{21}}$ $S_{12} = \frac{z_{12}}{(z_{11}+1)(z_{22}+1)-z_{12}z_{21}}$ $S_{21} = \frac{z_{21}}{(z_{11}+1)(z_{22}+1)-z_{12}z_{21}}$ $S_{22} = \frac{(z_{11}+1)(z_{22}+1)-z_{12}z_{21}}{(z_{11}+1)(z_{22}+1)-z_{12}z_{21}}$	$S_{11} = \frac{(1-y_{11})(1+y_{22})+y_{12}y_{21}}{(1+y_{11})(1+y_{22})-y_{12}y_{21}}$ $S_{12} = \frac{-y_{12}}{(1+y_{11})(1+y_{22})-y_{12}y_{21}}$ $S_{21} = \frac{-y_{21}}{(1+y_{11})(1+y_{22})-y_{12}y_{21}}$ $S_{22} = \frac{(1+y_{11})(1+y_{22})-y_{12}y_{21}}{(1+y_{11})(1+y_{22})-y_{12}y_{21}}$	$S_{11} = \frac{(h_{11}-1)(h_{22}+1)-h_{12}h_{21}}{(h_{11}+1)(h_{22}+1)-h_{12}h_{21}}$ $S_{12} = \frac{h_{12}}{(h_{11}+1)(h_{22}+1)-h_{12}h_{21}}$ $S_{21} = \frac{h_{21}}{(h_{11}+1)(h_{22}+1)-h_{12}h_{21}}$ $S_{22} = \frac{(h_{11}+1)(h_{22}+1)-h_{12}h_{21}}{(h_{11}+1)(h_{22}+1)-h_{12}h_{21}}$	$\frac{A+B-C-D}{A+B+C+D} \quad \frac{2(AD-BC)}{A+B+C+D}$ $\frac{2}{A+B+C+D} \quad \frac{-A+B-C+D}{A+B+C+D}$
$Z_{11} = \frac{(1+S_{11})(1-S_{22})+S_{12}S_{21}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$ $Z_{12} = \frac{S_{12}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$ $Z_{21} = \frac{S_{21}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$ $Z_{22} = \frac{(1-S_{11})(1-S_{22})-S_{12}S_{21}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$	$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$	$\frac{Y_{22}}{\Delta^Y} \quad \frac{-Y_{12}}{\Delta^Y}$ $\frac{-Y_{21}}{\Delta^Y} \quad \frac{Y_{11}}{\Delta^Y}$	$\frac{\Delta^h}{h_{22}} \quad \frac{h_{12}}{h_{22}}$ $\frac{-h_{21}}{h_{22}} \quad \frac{1}{h_{22}}$	$\frac{A}{C} \quad \frac{\Delta^A}{C}$ $\frac{1}{C} \quad \frac{D}{C}$
$Y_{11} = \frac{(1-S_{11})(1+S_{22})+S_{12}S_{21}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$ $Y_{12} = \frac{S_{12}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$ $Y_{21} = \frac{S_{21}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$ $Y_{22} = \frac{(1+S_{11})(1+S_{22})-S_{12}S_{21}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$	$\frac{Z_{22}}{\Delta^Y} \quad \frac{-Z_{12}}{\Delta^Y}$ $\frac{-Z_{21}}{\Delta^Y} \quad \frac{Z_{11}}{\Delta^Y}$	$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$	$\frac{1}{h_{11}} \quad \frac{-h_{12}}{h_{11}}$ $\frac{h_{21}}{h_{11}} \quad \frac{\Delta^h}{h_{11}}$	$\frac{D}{B} \quad \frac{\Delta^A}{B}$ $\frac{-1}{B} \quad \frac{A}{B}$

Tablo EK 1.1 Z<sub>0</sub>=1'e normalize edilmiş iki-kaplı parametreleri dönüşüm formülleri ( $\Delta^K=K_{11}K_{22}-K_{12}K_{21}$ ) (Devamı)

	S	Z	Y	h	A
<b>h</b>	$h_{11} = \frac{(1+S_{11})(1+S_{22})-S_{12}S_{21}}{(1-S_{11})(1+S_{22})+S_{12}S_{21}}$ $h_{12} = \frac{2S_{12}}{(1-S_{11})(1+S_{22})+S_{12}S_{21}}$ $h_{21} = \frac{-2S_{21}}{(1-S_{11})(1+S_{22})+S_{12}S_{21}}$ $h_{22} = \frac{(1-S_{11})(1+S_{22})+S_{12}S_{21}}{(1-S_{22})(1-S_{11})-S_{12}S_{21}}$	$\frac{\Delta^z}{Z_{22}}$ $\frac{Z_{21}}{Z_{22}}$ $\frac{1}{Z_{22}}$	$\frac{1}{Y_{11}}$ $\frac{-Y_{12}}{Y_{11}}$	$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$	$\frac{D}{D}$ $\frac{\Delta^A}{D}$ $\frac{D}{D}$ $\frac{C}{D}$
<b>A</b>	$A = \frac{(1+S_{11})(1-S_{22})+S_{12}S_{21}}{2S_{21}}$ $B = \frac{(1+S_{11})(1+S_{22})-S_{12}S_{21}}{2S_{12}}$ $C = \frac{(1-S_{11})(1-S_{22})-S_{12}S_{21}}{2S_{21}}$ $D = \frac{(1-S_{11})(1+S_{22})+S_{12}S_{21}}{2S_{21}}$	$\frac{Z_{11}}{Z_{21}}$ $\frac{\Delta^z}{Z_{21}}$ $\frac{1}{Z_{21}}$	$\frac{-Y_{22}}{Y_{21}}$ $\frac{-1}{Y_{21}}$	$\frac{-\Delta^h}{h_{21}}$ $\frac{-h_{11}}{h_{21}}$	$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$

### KAYNAKLAR

1. ALSPECTOR,B.,ALLEN,R.B.,JAYAKUMAR,A,ZEPPEFELF,T. and MEIR,R.,”Relaxation networks for large supervised learning problems”,In Advances in Neural Information Processing Systems, San Mateo, CA:Morgan Kaufmann,1991,pp. 1015-1021.
2. ALSPECTOR,B.,JAYAKUMAR,A. and LUNA,S.,”Experimental evolution of learning in a neural microsystem”,In Advances in Neural Information Processing Systems 4,San Mateo,CA:Morgan Kaufmann,1992,pp. 871-878.
3. ANDREOU,A.G.,”Minimal circuit models of neurons,synapses and multivariable functions for analog VLSI neuromorphic computation”,Report JHU/ECE-92-13,Johns Hopkins University,Baltimore,MD,1992.
4. ARIMA,Y.K.,MASHIKLO,K.,OKADO,T.,YAMADA,T.,MACDA,A.,NONTANI,H.,KONDOH,H. and KAYANO,H.,”336-Neuron 28K- Synapseself-learning neural network chip with branch-neuron-unit-architecture”,IEEE Trans.Journal of Solid-State circuits, Vol.26,1992,pp.1637-1644.
5. BAILEY,J. And HAMMERSTROM,D.,”Why VLSI Implementation of associative VLCNs require connection multiplexing”,International Conference on Neural Networks,Vol.2,San Diego,CA,1988,pp. 173-180.
6. BANDLER,J.W. and ZHANG,Q.J.,”An automatic decomposition technique for device modeling and large circuit design”,in IEEE MTT-S Int. Microwave Symp.Dig. ,Las Vegas,NV,1987,pp. 709-712.
7. BARBY, J.A., VLACH, J., SINGHAL, K.:”Polynomial splines for MOSFET model approximation”, IEEE Trans. Computer-Aided Design, 1988, 7, (5), pp. 557-567.
8. BERROTH,M. and BOSH R.:”Broad-band determination of the FET small-signal equivalent circuits”,IEEE Trans.1990,MTT-38, (7),pp. 891-895.
9. BIERNACKI, R.M., BANDLER, J.W., SONG, J. and ZHANG, Q.J.:”Efficient quadratic approximation for statistical design”,IEEE Trans. Circuits Syst.,1989, 36, (11), pp.1449-1454.
10. BLACK,H.S.,Modulation Theory,Princeton,NY:Van Nostrand,1953.
11. BOHEN,K.A.,POULIQUEN,P.O.,ANDREOU,A.G. and JENKINS,R.E.,”A heteroassociative memory using current-mode analog VLSI Circuits”,IEEE Trans. on Circuits and Systems ,Vol.CAS-36,1989,pp. 747-755.
12. BOSER,B.E.,SACKINGER,E.,BROMLEY,J.,LECUN,Y. and JACKEL,L.D.,”Hardware requirements for neural network pattern classifiers”,IEEE Micro 12,1992,pp.32-40.
13. BRIDGE,J.P,LADBROOKE,P.H. AND HILL,A.J.,’Characterisation of GaAs FET and HEMT chips and packages for accurate hybrid circuit design’,IEE Proceedings-H,1992, Vol.139,(4),pp. 330-336.
14. CHOI,J.,BANG,S.H. and SHEU,B.J,”A programmable analog VLSI neural network processor for communication receivers”,IEEE Trans. on Neural Networks 4,1993,pp. 484-495.

15. CHURCHER,S.,BAXTER,D.J.,HAMILTON,A.,MURRAY,A.F. And REKIE,H.M.,”Generic analog neural computation-The EPSILON Chip”,In Advances in Neural Information Processing Systems,San Mateo,CA:Morgan Kaufmann,1993,pp. 773-780.
16. CHURCLAND,P.S. and SEJNOWSKI,T.J.,The Computational Brain,Cambridge,MA,MIT Press,1992.
17. CHURCLAND,P.S.,Neurophilosophy:Toward A Unified Science of the Mind/Brain,Cambridge, MA: MITT Press,1986.
18. COCKE,J. and MARSTEIN,V.,”The Evolution of RISC Technology at IBM”,IBM Journal of Research and Devolopment 34,1990,pp. 4-11.
19. CURTICE,V.R.,”A MESFET Model for Use in Design of GaAs Integrated Circuits”,IEEE Tran.Microwave Theory Tech.,vol. MTT-28,1980,pp.448-456.
20. CURTICE,W.R,CAMISA,’Self-consistent GaAs FET models for amplifier design and device diagnostics ‘,IEEE Trans. 1984, MTT-32, pp.1573-1578.
21. DAMBRINE,G.CAPPY,A.HELIDORE,F. AND PLAYEZ,E.:”A new method for determining the FET small-signal equivalent circuit”,IEEE Trans.1988,MTT-36,(7),pp.1151-1159.
22. FROELICH,R.K.:”An improved model for noise characterization of microwave GaAs FET's" IEEE Trans.,1990,MTT-38 (6),pp. 703-706.
23. G.D.VENDELIN,A.M.PAVIO,U.L.ROHDE,”Microwave Circuit Design using Linear and Nonlinear Techniques”,John Wiley & Sons , 1990.
24. GOLIO,J.M.,”Microwave MESFETs and HEMTs”,Artech House,Inc.,1991.
25. GOLIO,J.M.,MILLER,M.,MARACAS,G.N. And JOHNSON D.A.,”Frequency Dependent Electrical Characteristics of GaAs MESFET’s”,IEEE Trans.Elec. Dev., Vol.ED-37,May 1990,pp.1217-1227.
26. GÜNEŞ, F., GÜRGEN, F., TORPİ H.:”Signal-Noise neural network model for active microwave devices”,IEE Proc.-Circuit Devices Syst., 1996, 143,(1), pp.1-8.
27. GÜNEŞ,F.,GÜNEŞ,M. AND FİDAN,M.:’Performance Characterisation of a Microwave Device’,IEE Proceedings-G, 1994 Vol.141, (5),pp. 337-344.
28. GÜNEŞ,F.,GÜRGEN,F.,TORPİ,H.," Neural Network Simulation of the Signal and Noise Parameters", Progress in Electromagnetic Research Symposium (PIERS),24-28 July 1995,Seattle Washington USA.
29. GÜNEŞ,F.,GÜRGEN,F.,TORPİ,H.,"Neural Network Simulation for Active Microwave Devices ", Fourth International Conference on ANN,26-28 June 1995,Cambridge , England.
30. GÜNEŞ,F.,GÜRGEN,F.,TORPİ,H.,"Unified Small-Signal-Noise Neural Network for Active Microwave Devices",ECCTD'95 European Conference on Circuit Theory and Design, 27-31 August 1995, İstanbul-TURKEY.
31. GÜNEŞ,F.,TORPİ,H.,ÇETİNER,B.A.,”Neural Network Approach For The Active Device Characterization”,accepted to submit in ECCTD'97 European Conference on Circuit Theory and Design, 1997, Budapeşte-HUNGARY.

32. GÜNEŞ,F.,TORPİ,H.,GÜRGEN,F.,"Efficient Model Parameter Extraction Using Neural Networks for Active Microwave Devices", Progress in Electromagnetic Research Symposium PIERS'96, 8-12 July 1996, Innsbruck, Austria.
33. GÜNEŞ,F.,TORPİ,H.,GÜRGEN,F.,"Multi-Bias/Configuration Neural Network Models for Microwave Active Devices", International Conference on Telecommunications, ICT'96, İstanbul-Türkiye, 13-17 April 1996.
34. GUPTA,K.C.,GARG,R. and CHADRA,R., "Computer-Aided Design of Microwave Circuits",Dedham MA:Artech House,1981.
35. GÜRGEN,F.,ALPAYDIN,R.,ÜNLÜAKIN,U.,ALPAYDIN,E.: 'Distributed and local neural classifiers for phoneme recognition ',Pattern Recognition Letters,1994,pp. 1111-1118.
36. HAMMERSTROM,D.,"A VLSI architecture for high performance,low cost,or-chip learning " ,International Joint Conference on Neural Networks,Vol.2,San Diego,CA,1990,pp.537-544.
37. HAMMERSTROM,D.,"Electronic Neural Network Implementation",Tutorial No:5,International Joint Conference on Neural Networks,Baltimore,MD,1992.
38. HAMILTON,A.,MURRAY,A.F,BAXTER,D.J.,CHURCHER,S.REKIE,H.M. and TARESENKO, L., "Integrated pulse stream neural networks:Results,issues and pointers",IEEE Trans. on Neural Networks 3,1992,pp. 385-393.
39. HAYKIN,S.,"Neural Networks",Macmillan College Publishing Company,Inc.,1994.
40. HAYKIN,S.,Communication Systems,Second edition,New York,Wiley,1983.
41. HECHT-NIELSEN,R.,Neurocomputing,Reading,MA:Addison-Wesley,1990.
42. HERTZ,J.,KROGH,A., and PALMER,G.,"Introduction to the theory of neural networks",Vol.1,Addison-Wesley,1991.
43. HOENEISEN,B. And MEAD,C.A.,"Fundamental Limitations in Microelectronics.I.MOS Tecnology",Solid-State Electronics 15,1972,pp. 819-829.
44. HOLLER,M.A.,TAM,S.,CASTRO,H. and BENSON,R.,"An electrically trainable artificial neural network (ETANN) with 10240 'floating gates' synapses ",International Joint Conference on Neural Networks,Vol. 2,San Diego,CA,1989,pp. 191-196.
45. HSPICE User's Manual, Version H8907,Meta-Software,Campbell,CA,1989.
46. HU Z.R., YANG Z.M., FUSCO V.F.,J.A.C.STEWART,"Unified small-signal-noise model for active microwave device",IEE Proceedings-G,Vol.140,(1),1993 ,pp.55-60.
47. HUSH,D.R.,HORNE,B.G.,'Progress in supervised neural networks',IEEE Signal and Processing Magazine,1993,pp.8-36.
48. KACPRZAK,T.,MATERKA,A.,"Compact DC Model of GaAs FET's for Large-Signal Computer Calculation",IEEE J. of Solid- State Circuits,Vol. SC-18,April 1983,pp.211-213.
49. KARLIK,B.,'Çok Fonksiyonlu Protezler için Yapay Sinir Ağları Kullanılarak Miyoelektrik Kontrol ',YTÜ FBE Doktora Tezi,İstanbul TURKEY,Mart 1994.
50. KEYES,R.W.,The Physics of VLSI Systems,Reading,MA:Addison-Wesley,1987.

51. KONDOH,H., 'An accurate FET modelling from measured S-parameters ',IEEE MTT-S Int. Microwave Symp. Dig., Baltimore ,1986,pp. 377-380.
52. LADBROOKE,P.H.: 'MMIC design:GaAs FETs and HEMTs' (Artech House,Norwood,MA,1989).
53. LARSON,L.E., "An Improved GaAs MESFET Equivalent Circuit Model for Analog Integrated Circuit Applications", IEEE J. of Solid-State Circuits. Vol. SC-22, August 1987, pp.567-574.
54. LEE,J.-C.,SHEU,B.J.,FANG,W.C. and CHELLAPPA,R., "VLSI neuroprocessors for video motion dedector", IEEE Trans. on Neural Networks 4,1993,pp. 178-191.
55. LEVINE,M., Man and Machine Vision, New York, McGraw Hill, 1985.
56. LIPPMANN,R.P., IEEE ASSP Magazine, pg 5-38, April 1987.
57. MACQ,D., VERLEYSSEN,M., JESPERS,P. and LEGAT,J.-D., "Analog implementation of Cohonen map with on-chip learning", IEEE Trans. on Neural Networks 4,1993, pp. 456-461.
58. MARR,D., Vision, New York, Freeman, 1982.
59. MEAD,C.C., Analog VLSI and Neural Systems, Reading, MA: Addison-Wesley, 1989.
60. MEIJER, P.B.L.: "Fast and smooth highly nonlinear multidimensional table models for device modeling", IEEE Trans. Circuit Syst., 1990, 37, (3) , pp. 335-346.
61. MINASIAN,R.A., "Simplified GaAs MESFET model to 10 GHz", Electron Lett., Vol. 13, No. 8, 1977, pp. 541-549.
62. MOTCHENBACHER,C.D, and FITCHEN,F.C., "Low-Noise Electronic Design", John Wiley & Sons, 1973.
63. MURRAY,A.F., DEL CORSO,D. and TARRESENKO,L., "Pulse-stream VLSI neural networks mixing analog and digital techniques", IEEE Trans. on Neural Nets, London, UK, 1991, pp.27-32.
64. PALUBINKAS G., Pattern Recognition Letters, 1994.
65. POSPIESZALSKI,M.W.: "Modelling of noise parameters of MESFETs and MODFETs and their frequency and temperature dependence" , IEEE Trans., 1989, MTT-37,(9), pp1340-1350.
66. PUCEL,R.A., HAUS,H.A. ,and, STATZ,H., "Signal and Noise Properties of Gallium Arsenide Microwave Field-Effect Transistors", Advances in Electronics and Electron Physics, Vol.38, New York Academic Press, pp.195-265.
67. RAMACHER,U., "Hardware concepts of neural networks", In Advanced Neurocomputers, New York: Elsevier, 1990, pp. 209-218.
68. RAMACHER,U., BEICHTER,J. and BRULS,N., "Architecture of general-purpose neural signal processor", International Joint Conference on Neural Networks, Vol. 1, Seattle, WA, 1991, pp. 443-446.
69. ROUX,J.P., ESCOTTE L., PLANA R., GRAFFEUIL, DELAGE S.L., "Small-Signal and Noise Model Extraction Technique for heterojunction Bipolar Transistor at Microwave Frequencies", IEEE Trans. 1995-MTT-43(2), pp. 293-298.
70. RUMELHART,D.E., HINTON,G.E. and WILLIAMS,R.J., "Learning representations by back propagation errors" Nature (London), Vol. 323, 1986, pp 533-536.

71. SACKINGER,E.,BOSER,B.E.,BROMLEY,J.,LECUN,Y. and JACKEL,L.D.,"Application of the ANNA neural network chip to high-speed character recognition",IEEE Trans. on Neural Networks 3,1992,pp. 498-505.
72. SCHICHMANN,H. and HODGES,D.,"Modeling and Simulation of Insulated-gate FET Switching Circuits",IEEE Of Solid-State Circuits,Vol. SC-3,September 1968,pp.285-289.
73. SHIRAKAWA,K.,OIKAWA,H.,SHIMURA,T.,KAWASAKI,Y.,OHASHI,Y., SAITO,T. And DAIDO , Y."An Approach to Determining an Equivalent Circuit for HEMT's",IEEE Trans. On MTT,Vol. 43,N0. 3,March 1995,pp. 499-503.
74. STATZ,H.,NEWMAN,P.SMITH,I.,PUCEL,R. and HAUS,H.,"GaAs FET Device and Circuit Simulation in SPICE",IEEE Trans. Electron Devices,Vol, ED-34,1987,pp.160-169.
75. TAKI,T.,"Approximation of JFET Characteristics by Hyperbolic Functions",IEEE J. Of Solid-State Circuits,Vol. SC-13,October 1978,pp.724-726.
76. TORPİ,H.,'(LNA) Düşük Gürültülü Kuvvetlendirici Tasarımı' , Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Temmuz, 1991.
77. TORPİ,H.,GÜNEŞ,F.,GÜRGEN,F.," Neural Network Approach for the Characterisation of the Active Microwave Devices", I.Symposium on Mathematical and Computational Applications,Vol. 1,No:1,19-21 November 1996,Manisa Turkey,pp.113-118.
78. TORPİ,H.,GÜNEŞ,F.,GÜRGEN,F.,"Aktif Mikrodalga Elemanlarının Yapay Sinir Ağı Simulatörleri ",Elektrik Mühendisliği 6.Ulusal Kongresi , 11-17 Sept. 1995, Bursa, TÜRKİYE, pp.676-679.
79. TORPİ,H.,GÜNEŞ,F.,GÜRGEN,F.," Sinyal ve Gürültü Parametrelerinin Yapay Nöron Ağı Simülasyonu" SİU'95, Sinyal İşleme ve Uygulamaları Konferansı 26-28 April 1995, Nevşehir, Türkiye, pp.214-219.
80. TUYL,R.V. and LIECHTI,C.,"Gallium Arsenide Digital Integrated Circuits",Technical Report AFL-TR-74-40,Air Force Avionics Lab.,March 1974.
81. VAITKUS,R.L.'Uncertainty in the values of GaAs MESFET equivalent circuit elements extracted from measured two-port scattering parameters',Proc.1983 IEEE Conf.High Speed Semiconductor Devices Circuits,Cornell Univ. Ithaca,NY,pp. 301-308.
82. WATANABE,T.K.,KIMURA,K.,AOKI,M.,SAKATA,T. and ITO,K.,"A single 1.5 V digital chip for a  $10^6$  synapse neural network",IEEE Trans. on Neural Networks 4,1993,pp. 387-393.
83. WEDGE,W.S., and RUTLEDGE,D.B.,"Wave techniques for noise modeling and measurement",IEEE Trans. On MTT,Vol.40,No 11,November 1992,pp.2004-2012.
84. YUH,J.-D. and NEWCOMB,R.W.,"A multilevel neural network for A/D conversion",IEEE Trans. on Neural Networks 4,1993,pp. 470-483.
85. ZAAB, H.A., ZHANG, Q.NAKHLA, M.:"A neural network modeling approach to circuit optimisation and statistical design",IEEE Trans. on Microwave Theory and Tech., 1995, 43, (6), pp.1349-1357.

## ÖZGEÇMİŞ

- Hamid TORPi Elektronik ve Haberleşme Yüksek Mühendisi (1991 Yıldız Teknik Üniversitesi)
- Medeni Durumu Evli iki çocuk sahibi.
- Yabancı Dil İngilizce, Almanca
- Doğum Yeri Almanya, Stuttgart (Kircheim Unter Teck)
- Doğum Tarihi 7 Mart 1967
- 1973-1975 Özel İşcan Ana ve İlkokulu
- 1975-1976 Özel Yeşilyurt İlk ve Ortaokulu
- 1976-1978 Özel İdeal Ana ve İlkokulu (Mezuniyet)
- 1978-1980 Özel Fatih Koleji
- 1980-1981 Yalova Lisesi (Orta kısmından Mezuniyet)
- 1981-1984 Yalova Lisesi (Mezuniyet Ortalaması 8.09)
- 1984-1988 Yıldız Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü  
Mezuniyet Derecesi 1.(Dönem Birincisi), Mezuniyet Ortalaması 71.83  
Bitirme Tezi Konusu: "TVRO Uydu antenlerinin Reflektör ve Aktif Elemanlarının İncelenmesi", Tez Danışmanı Y.Doç.Dr.Cahit CANBAY
- 1988 YÜ FBE Elektronik ve Haberleşme Müh. Ana Bilimdalı Haberleşme Bilim dalında Yüksek Lisansa Girdi.
- 1990 YÜ F.B.Enstitüsü ' nde Arş.Gör. oldu.
- 1991 YÜ FBE Elektronik ve Haberleşme Müh. Ana Bilimdalı Haberleşme Bilim dalından Yüksek Mühendislik Diploması aldı. (Mezuniyet Ortalaması :82.52)  
Bitirme Tezi Konusu: "(LNA) Düşük Gürültülü Kuvvetlendirici Tasarımı",  
Tez Danışmanı: Prof.Dr.Filiz GÜNEŞ
- 1991 YTÜ Elektrik ve Elektronik Fakültesi Elektronik ve Hab. Müh. Bölümü EM. Alanlar ve Mikrodalga Tekniği A.B.D. Araştırma Görevliliği kadrosuna geçti.  
Yıldız Teknik Üniversitesi FBE Elektronik ve Haberleşme Müh. Anabilim dalı Haberleşme Bilimdalı Doktora Programına girdi.
- 1992 Ders Ortalaması: 82.85 ile Doktora derslerini başarı ile tamamladı.  
Doktora Yeterlilik Sınavını başarı ile bitirerek Prof.Dr. Filiz GÜNEŞ danışmanlığında Doktora tezine başladı.
- 1997 Doktora Tezini teslim etti.