

154363

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**TRANSFER SAÇILMA PARAMETRELERİ İLE
MİKRODALGA KUVVETLENDİRİCİ ANALİZ VE
SENTEZİ**

Elektronik ve Haberleşme Müh. Umut HINÇAL

FBE Elektronik ve Haberleşme Anabilim Dalı Haberleşme Programında
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof. Dr. Filiz GÜNEŞ (YTÜ)

Prof. Dr. Filiz Güneş
Y. Doç. Dr. Hamid TORPİ
Doç. Dr. Tayfun GÜNEL

İSTANBUL, 2004

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	v
KISALTMA LİSTESİ	vii
ŞEKİL LİSTESİ	viii
ÇİZELGE LİSTESİ	xi
ÖNSÖZ.....	xii
ÖZET	xiii
ABSTRACT	xiv
1. GİRİŞ.....	1
2. ÖNCEKİ ÇALIŞMALAR	2
3. ÇALIŞMA İLE SUNULAN YENİLİKLER	3
4. KASKAD BAĞLI İKİ-KAPILININ [S] PARAMETRELERİ İLE KARAKTERİZASYONU	4
4.1 [S] Parametreleri	4
4.2 Kaskad Bağlı İki-Kapılıların [ABCD] Parametreleri İle Karakterizasyonu.....	5
4.3 [A] Parametreleri	7
4.3.1 Kaskad Bağlı İki-Kapılıların [A] Parametreleri İle Karakterizasyonu	7
4.3.2 [A] - [S] Parametreleri Dönüşümleri	8
4.3.3 [S] - [A] Parametreleri Dönüşümleri	9
5. SİSTEM TANITIMI	10
5.1 Sistemde Kullanılan Temel Elemanlar Ve Karakterizasyonu	10
5.1.1 Seri Transmisyon Hat İki-Kapılısı.....	10
5.1.2 Paralel Transmisyon Hat İki-Kapılısı	15
5.2 “T”, “İT” VE “L” Uydurma Devreleri İle Tek Transistorlu Temel Kuvvetlendirici Devreleri	18
5.3 Uydurma Devreleri	21
5.3.1 “T” Devresi.....	21
5.3.2 “İT” Devresi	21
5.3.3 Paralel + Seri “L” Devresi	22
5.3.4 Seri + Paralel “L” Devresi	22
5.4 Aktif Eleman Karakterizasyonu Ve Performans Ölçü Fonksiyonları	23
5.4.1 Transistor İşaret Ve Gürültü Davranışı	23
5.4.2 Transistor Performans Ölçü Fonksiyonları.....	24

6.	MİKRODALGA KUVVETLENDİRİCİSİ GERÇEKLENEBİLİR TASARIM UZAYI.....	26
6.1	Mikrodalga Kuvvetlendiricisi Optimizasyonu İki Temel Problemi	26
6.2	Gerçeklenebilir Tasarım Hedef Uzayı Tayini: Performans Data Yaprakları	27
6.3	Transistor Performans Karakterizasyonu	30
6.4	Giriş Ve Çıkış Uydurma Devrelerinin Ayrı Ayrı Tasarlanması.....	31
6.4.1	Kaynak Empedansını Transistorun Giriş Empedansına Uygunlaştıran Devre.....	32
6.4.2	Yük Empedansını Transistorun Çıkış Empedansına Uygunlaştıran Devre.....	33
7.	MİKRODALGA KUVVETLENDİRİCİSİ TASARIM UZAYI VE TASARIM HEDEFLERİ	34
7.1	Mikrodalga Kuvvetlendirici Tasarım Uzayı.....	34
7.2	Mikrodalga Kuvvetlendirici Tasarım Hedefleri	36
8.	MİKRODALGA KUVVETLENDİRİCİ OPTİMİZASYONU.....	38
8.1	Temel Mühendislik Optimizasyon Problemleri	38
8.1.1	Tasarım Uzayı Ve Mikrodalga Kuvvetlendiricisine Uygulaması: Çok Hedefli Amaç Fonksiyonları	38
8.1.1.1	(F , V_i , G_T), B Hedef Uzayı Ve Kullanımı	38
8.1.1.2	(Z_S , Z_L) Ve B Hedef Uzayı Ve Kullanımı	39
8.1.2	Global Optimum Ve Algoritmaları.....	41
8.1.2.1	Gradient-Temelli Algoritmalar.....	41
	Dolaylı Metod.....	41
	Doğrudan Metod.....	42
8.1.2.2	Sezgisel Temelli Algoritmalar.....	43
8.1.2.3	Genetik Algoritma İle Diğer Geleneksel Yöntemler Arasındaki Farklar.....	43
9.	PROGRAM SONUÇLARI.....	47
9.1	Genetik Algoritma İle Gerçekleştirilen Kuvvetlendiriciler	48
9.1.1	NE329S01 Transistoru İle Gerçekleştirilen Kuvvetlendirici Sonuçları	48
9.1.1.1	Kuvvetlendirici Tasarım Uzayı	48
9.1.1.2	Giriş Ve Çıkış Uydurma Devreleri.....	50
9.1.1.3	“T” Tipi Kuvvetlendirici Devreleri	52
9.1.1.4	“Π” TİPİ Kuvvetlendirici Devreleri	56
9.1.1.5	T-Π Hibrid Mikrodalga Kuvvetlendiricisi.....	60
9.1.1.6	Π-T Hibrid Mikrodalga Kuvvetlendiricisi.....	62
9.1.1.7	T-L Hibrid Mikrodalga Kuvvetlendiricisi	64
9.1.1.8	Π-L Hibrid Mikrodalga Kuvvetlendiricisi.....	66
9.1.1.9	L Tipi Mikrodalga Kuvvetlendiricisi.....	68
10.	SONUÇLAR.....	70
	KAYNAKLAR.....	72
	EKLER	73

Ek 1 Mikrodalga Kuvvetlendiricisi Hedef Fonksiyonu Programları 74

Ek 2 Mikrodalga Kuvvetlendiricisi Toplam Devre Hesap Programları 92

Ek 3 Genetik Algoritma Optimizasyon Programları..... 123

ÖZGEÇMİŞ..... 155



SİMGE LİSTESİ

[A]	A Parametrelerinin Gösterimi
E	Hata Fonksiyonu
E_N	Gürültü Hata Fonksiyonu
E_{Vi}	VSWR Hata Fonksiyonu
$F(\omega)$	Gürültü Faktörü
F_M	Minimum Gürültü Faktörü
$F_{req}(\omega)$	İstenen Gürültü Faktörü
$G_T(\omega)$	Kuvvetlendirici Güç Kazancı
$G_{T_{max}}(\omega)$	Maksimum Kazanç Değeri
$G_{T_{min}}(\omega)$	Minimum Kazanç Değeri
$G_{T_{req}}(\omega)$	İstenen Kazanç Değeri
ℓ_n	Transmisyon Hattının Fiziksel Uzunluğu
N	Gürültü Parametresi
n	Frekans Örnekleme Sayısı
NF	Kuvvetlendirici Gürültü Faktörü
P_{AVS}	Maksimum Kaynak Gücü
P_L	Yüke Aktarılan Güç
R_L	Yük Empedansının Reel Kısmı
R_N	Eşdeğer Gürültü Direnci
R_S	Kaynak Empedansının Reel Kısmı
S	Saçılma Parametresi
V_{DS}, I_D, V_{CE}, I_C	Transistor Kutuplama Koşulları
$V_i(\omega)$	VSWR
$V_{i_{req}}(\omega)$	İstenen VSWR Değeri
W	Ağırlık Katsayısı
X_S	Kaynak Empedansının İmajiner Kısmı
X_L	Yük Empedansının İmajiner Kısmı
Z_i	Giriş Empedansı
Z_i^*	Giriş Empedansının Eşleniği
Z_{in}^{OMC}	Çıkış Uydurma Devresinin giriş empedansı
Z_L	Yük empedansı
Z_L^*	Yük Empedansının Eşleniği
$Z_{L_{IMC}}$	Transistorun Giriş Empedansı
$Z_{L_{ref}}(\omega_i)$	Yüke Bakıldığında Görülen Empedans Değeri
Z_O	Çıkış Empedansı
Z_{op}	Optimum Kaynak Empedansı
Z_{0n}	Transmisyon Hattı Empedansı
Z_{out}^{IMC}	Giriş Uydurma Devresinin çıkış empedansı
Z_S	Kaynak empedansı

Z_S^*	Kaynak Empedansının Eşleniği
$Z_{S_{OMC}}$	Transistorun Çıkış Empedansı
$Z_{S_{ref}}(\omega_i)$	Kaynağa Bakıldığında Görülen Empedans Değeri
β	Faz Kaydırma Sabiti
γ	Daralma Parametresi
σ	Büzülme Parametresi
ρ	Yansıma Parametresi
ρ_{in}	Giriş Yansıma Parametresi
$\rho_{in\,ref}$	Giriş Yansıma Parametresi Referans Değeri
χ	Genişleme Parametresi



KISALTMA LİSTESİ

VSWR	Voltage Standing Wave Ratio
NF	Noise Figure
CT	Configuration Type
CIS	Can Çevikbaş, Ilkay Köseli, Serhat Altunç
IEEE	Institute of Electrical and Electronics Engineers
YTU	Yıldız Teknik Üniversitesi
İTÜ	İstanbul Teknik Üniversitesi
IMC	Giriş Uydurma Devresi (Input Matching Circuit)
OMC	Çıkış Uydurma Devresi (Output Matching Circuit)
NE329S01	Katalogdan Seçilmiş bir Transistor Türü
FMINS	MATLAB 6 Programındaki Bir Toolbox
FCON	MATLAB 6 Programındaki Bir Toolbox
PS	Kaskad bağlı Paralel+Seri Transmisyon Hattı
SP	Kaskad bağlı Seri+ Paralel Transmisyon Hattı



ŞEKİL LİSTESİ

Şekil 4.1: N-kapılı bir Mikrodalga Devresi.....	4
Şekil 4.2: Bir İki- Kapılıda Gelen ve Yansıyan dalgalar.....	7
Şekil 4.3: Kaskad bağlı iki İki- Kapılıda Gelen ve Yansıyan dalgalar.....	7
Şekil 5.1.1.1 Seri transmisyon hat iki-kapılısı.....	10
Şekil 5.1.1.2 Z_N referans empedansı ile sonlandırılmış transmisyon hattı	11
Şekil 5.1.1.3 Z_{in} sonlandırmalı eşdeğer.....	11
Şekil 5.1.1.4 S_{21} 'i bulmak için kullanılan devre.....	12
Şekil 5.1.2 Paralel transmisyon hat iki-kapılısı.....	15
Şekil 5.2.1 T- tipi uydurma devreli tek transistorlu kuvvetlendirici.....	19
Şekil 5.2.2 Π - tipi uydurma devreli tek transistorlu kuvvetlendirici.....	19
Şekil 5.2.3 L- tipi (seri + paralel) uydurma devreli tek transistorlu kuvvetlendirici.....	19
Şekil 5.2.4 L- tipi (paralel + seri) uydurma devreli tek transistorlu kuvvetlendirici.....	20
Şekil 5.2.5 Giriş “T” tipi çıkış “L” tipi (seri+ paralel) uydurma devreli tek transistorlu kuvvetlendirici	20
Şekil 5.2.6 Giriş “ Π ” tipi çıkış “L” tipi (seri+ paralel) uydurma devreli tek transistorlu kuvvetlendirici	20
Şekil 5.3 “T” tipi uydurma devresi.....	21
Şekil 5.4 “ Π ” tipi uydurma devresi	21
Şekil 5.5 Paralel + Seri “L” uydurma devresi	22
Şekil 5.6 Seri + Paralel transmisyon hattı	22
Şekil 5.7 Mikrodalga transistorun iki-kapılı modeli.....	24
Şekil 6.1 Mikrodalga kuvvetlendiricisi tasarım uzayı “Kara-Kutu” modeli	27
Şekil 6.2 Mikrodalga transistoru için “Performans Data Yaprakları”nın blok diyagramı (Cengiz,2004)	28
Şekil 6.3 “Performans Data Yaprakları”nın “Yapay Sinir Ağı” blok diyagramı (Cengiz,2004)	29
Şekil 6.4 “Performans Data Yaprakları”nın bulanık mantık blok diyagramı (Cengiz,2004)	29
Şekil 6.5 Giriş empedans düzleminde koşullu kararlı halde bir V_{ireq} ile sınırlandırılmış kazanç daireleri (Güneş, 1994)	30
Şekil 6.6 Giriş ve çıkış uydurma devreli mikrodalga kuvvetlendiricisi	31
Şekil 6.7 Kaynak empedansını transistorun giriş empedansına uygunlaştıran devre.....	33
Şekil 6.8 Yük Empedansını Transistorun Çıkış Empedansına Uygunlaştıran Devre.....	33
Şekil 7.1 Mikrodalga kuvvetlendiricisi tasarım uzayı kara kutu modeli.....	36
Şekil 7.2 NE329S01 Transistorunun $F=0,46dB$, $V_i=1$ $G_{Tmax}(f)$ Değişimi	37
Şekil 7.3 NE329S01 Transistoru için (0,46dB ,1, 12dB) Üçlülerinin sonlandırma Fonksiyonları	37
Şekil 8.1 Pratikte karşılaşılan bir hedef fonksiyonu örneği (Erol, 1995)	42
Şekil 8.2 Amaç Fonksiyonları (Erol, 1995).....	44
Şekil 8.3 Genetik algoritmanın akış diyagramı (Cengiz,2004)	46
Şekil 9.1 NE329S01 Transistorunun $I_c=10$ mA, $V_{CE}=2$ V kutuplaması için $F_{min}(f)$ Değişimi (Güneş, 2002).....	48
Şekil 9.2 NE329S01 Transistorunun $F=0,46dB$, $V_i=1$ $G_{Tmax}(f)$ Değişimi (Güneş, 2002) .	49
Şekil 9.3 NE329S01 Transistoru için (0,46dB ,1, 12dB) uyumlu üçlülerinin sonlandırma fonksiyonları (Güneş, 2002)	49
Şekil 9.4 Kaynak empedansını transistorun giriş empedansına uygunlaştıran devre ($Z_s'=50\Omega$).....	50
Şekil 9.5 Yük Empedansını Transistorun Çıkış Empedansına Uygunlaştıran Devre ($Z_L'=50\Omega$)	50

Şekil 9.6 Tüm uydurma devrelerinin kazanç - frekans karakteristikleri	51
Şekil 9.7 Giriş ve çıkışı “T” tipi Uydurma devreli mikrodalga kuvvetlendiricisinin iki kapılı gösterimi	52
Şekil 9.8 Giriş uydurma devresi için kazanç - frekans karakteristiği	53
Şekil 9.9 Çıkış uydurma devresi için kazanç - frekans karakteristiği	54
Şekil 9.10 Girişi ve çıkışı “T” tipi uydurma devreli mikrodalga kuvvetlendiricisinin kazancının frekansa bağlı olarak değişimi	55
Şekil 9.11 Girişi ve çıkışı “T” tipi uydurma devreli mikrodalga kuvvetlendiricisinin VSWR – frekans değişimi	55
Şekil 9.12 Girişi ve çıkışı “T” tipi uydurma devreli mikrodalga kuvvetlendiricisinin gürültüsünün frekansa bağlı olarak değişimi	55
Şekil 9.13 Giriş ve çıkışı “Π” tipi Uydurma devreli mikrodalga kuvvetlendiricisinin iki kapılı gösterimi	56
Şekil 9.14 Giriş uydurma devresi için kazanç - frekans Karakteristiği	57
Şekil 9.15 Çıkış uydurma devresi için kazanç - frekans karakteristiği	58
Şekil 9.16 Girişi ve çıkışı “Π” tipi uydurma devreli mikrodalga kuvvetlendiricisinin kazancının frekansa bağlı olarak değişimi	59
Şekil 9.17 Girişi ve çıkışı “Π” tipi uydurma devreli mikrodalga kuvvetlendiricisinin VSWR – frekans değişimi	59
Şekil 9.18 Girişi ve çıkışı “Π” tipi uydurma devreli mikrodalga kuvvetlendiricisinin gürültüsünün frekansa bağlı olarak değişimi	59
Şekil 9.19 Giriş “T” tipi ve çıkış “Π” tipi Uydurma devreli mikrodalga kuvvetlendiricisinin iki kapılı gösterimi	60
Şekil 9.20 Girişi “T” çıkışı “Π” tipi uydurma devreli mikrodalga kuvvetlendiricisinin kazancının frekansa bağlı olarak değişimi	61
Şekil 9.21 Girişi “T” çıkışı “Π” tipi uydurma devreli mikrodalga kuvvetlendiricisinin VSWR – frekans değişimi	61
Şekil 9.22 Girişi “T” çıkışı “Π” tipi uydurma devreli mikrodalga kuvvetlendiricisinin gürültüsünün frekansa bağlı olarak değişimi	61
Şekil 9.23 Giriş “Π” tipi ve çıkış “T” tipi Uydurma devreli mikrodalga kuvvetlendiricisinin iki kapılı gösterimi	62
Şekil 9.24 Girişi “Π” çıkışı “T” tipi uydurma devreli mikrodalga kuvvetlendiricisinin kazancının frekansa bağlı olarak değişimi	63
Şekil 9.25 Girişi “Π” çıkışı “T” tipi uydurma devreli mikrodalga kuvvetlendiricisinin VSWR – frekans değişimi	63
Şekil 9.26 Girişi “Π” çıkışı “T” tipi uydurma devreli mikrodalga kuvvetlendiricisinin gürültüsünün frekansa bağlı olarak değişimi	63
Şekil 9.27 Giriş “T” tipi, çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistörli kuvvetlendirici	64
Şekil 9.28 Giriş “T” tipi, çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistörli kuvvetlendiricinin kazancının frekansa bağlı olarak değişimi	65
Şekil 9.29 Giriş “T” tipi, çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistörli kuvvetlendiricinin VSWR – frekans değişimi	65
Şekil 9.30 Giriş “T” tipi, çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistörli kuvvetlendiricinin gürültüsünün frekansa bağlı olarak değişimi	65
Şekil 9.31 Giriş “Π” ve çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistörli kuvvetlendirici	66
Şekil 9.32 Giriş “Π” tipi, çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistörli kuvvetlendiricinin kazancının frekansa bağlı olarak değişimi	67
Şekil 9.33 Giriş “Π” tipi, çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistörli	

kuvvetlendiricinin VSWR – frekans değişimi	67
Şekil 9.34 Giriş “Π” tipi, çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistör kuvvetlendiricinin gürültüsünün frekansa bağlı olarak değişimi.....	67
Şekil 9.35 Giriş ve çıkış “L” tipi Uydurma devreli tek transistör kuvvetlendirici.....	68
Şekil 9.36 Giriş ve çıkış “L” tipi uydurma devreli tek transistör kuvvetlendiricinin kazancının frekansa bağlı olarak değişimi.....	69
Şekil 9.37 Giriş ve çıkış “L” tipi uydurma devreli tek transistör kuvvetlendiricinin VSWR – frekans değişimi	69
Şekil 9.38 Giriş ve çıkış “L” tipi uydurma devreli tek transistör kuvvetlendiricinin gürültüsünün frekansa bağlı olarak değişimi	69



ÇİZELGE LİSTESİ

Çizelge 9.1 Tüm uydurma devrelerinin kazançlarının frekans ile değişimleri..... 51

Çizelge 9.2 Kaynak empedansı ile NE329S01 transistörünün giriş empedansı değerleri 52

Çizelge 9.3 Yük empedansı ile NE329S01 transistörünün çıkış empedansı değerleri..... 53

Çizelge 9.4 Kuvvetlendiricinin kazanç, gürültü ve giriş VSWR sonuçlarının simülasyon programı sonuçları ile birlikte gösterimi 54

Çizelge 9.5 Kaynak empedansı ile NE329S01 transistörünün giriş empedansı değerleri 56

Çizelge 9.6 Yük empedansı ile NE329S01 transistörünün çıkış empedansı değerleri..... 57

Çizelge 9.7 Kuvvetlendiricinin kazanç, gürültü ve giriş VSWR sonuçlarının simülasyon programı sonuçları ile birlikte gösterimi 58

Çizelge 9.8 Kuvvetlendiricinin kazanç, gürültü ve giriş VSWR sonuçlarının simülasyon programı sonuçları ile birlikte gösterimi 60

Çizelge 9.9 Kuvvetlendiricinin kazanç, gürültü ve giriş VSWR sonuçlarının simülasyon programı sonuçları ile birlikte gösterimi 62

Çizelge 9.10 Kuvvetlendiricinin kazanç, gürültü ve giriş VSWR sonuçlarının simülasyon programı sonuçları ile birlikte gösterimi 64

Çizelge 9.11 Kuvvetlendiricinin Kazanç, Gürültü ve VSWR sonuçlarının simülasyon programı sonuçları ile birlikte gösterimi 66

Çizelge 9.12 Kuvvetlendiricinin kazanç, gürültü ve giriş VSWR sonuçlarının simülasyon programı sonuçları ile birlikte gösterimi 68

Çizelge 10. 1 NE329S01 transistörü için ZI_Input.csv yardımcı data bilgileri 91

Çizelge 10. 2 NE329S01 transistörü için Zs_Input, csv yardımcı data bilgileri 91

ÖNSÖZ

Bu çalışma Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği anabilim dalı Haberleşme programında yüksek lisans tezi olarak hazırlanmıştır. Çalışmada temel parametreler arasında dönüşüm formülleri ve kullanılan temel hat yapılarının bu parametreler cinsinden karşılıkları sunulmuştur. Bunun yanında genetik algoritma tabanlı bir optimizasyon işlemi uydurma devreleri üzerine uygulanmış ve elde edilen sonuçlarla farklı uydurma devrelerine sahip mikrodalga devreleri oluşturulmuştur.

Bu tezin hazırlanmasında ve çalışmalarım sırasında değerli yardımlarını esirgemeyen, tezin şekillenmesi açısından yardımını esirgemeyen ve kendisini ailemden biri gibi gördüğüm sayın Hocam Prof. Dr. Filiz Güneş'e, hem çalışmalarımın en başında şahsen yardımcı olan hem de hazırladığı yüksek lisans tezi ile kaynak programlar da dahil olmak üzere çeşitli katkılarda bulunan sevgili arkadaşım Elektronik ve Haberleşme Y. Müh. Ilgar Aliyev'e, programlama ve genetik algoritma konusunda bana yardımcı olan sevgili arkadaşım Elektronik ve Haberleşme Y. Müh. Yavuz Cengiz'e, programlama konusunda sunduğu destekle ve sunduğu yeni fikirleriyle yardımcı olan Candaş Bilgin'e, tüm bunların yanında maddi ve manevi desteklerini hiçbir zaman esirgemeyen sevgili aileme teşekkürlerimi sunarım.



ÖZET

Mikroşerit transistörli kuvvetlendiriciler küçük boyutları, hafif olmaları ve üretim proseslerinin kolaylaşması gibi nedenlerden dolayı mikrodalga frekanslarda artan bir yaygınlıkla kullanılmaktadır. Ancak mikroşerit transistörli kuvvetlendiricilerin klasik yollardan tasarımı bazı güçlükler yaratmaktadır. Bunlar sınırlı Band Genişliği, Gürültü Faktörü olarak özetlenebilir. Bu yüzden tasarım aşamasında birçok deneysel adımın tekrarlanması ve değişik dizaynların denenmesi gerekebilir. Bu sebepten dolayı mikrodalga kuvvetlendiricilerin tasarımında bilgisayar destekli tasarım yaklaşımları öne çıkmıştır.

Bu aşamada kullanılan çeşitli parametreler içinde transfer saçılma parametreleri daha yaygın bir kullanıma sahiptirler. Özellikle [A] parametreleri kaskad bağlı devrelerin modellenmesinde matris çarpımı tabanlı basit bir hesaplama sunmaları açısından diğer parametrelere göre daha tercih edilebilir bir yapı sunuyorlar. Uydurma devrelerinin de bu yapıya kolayca adapte edilebiliyor olması bu parametrelerin işlevselliğini arttıran en temel etkenlerden birisi. [S] parametreleri ve [A] parametrelerinin basit aritmetik işlemlerle birbirine dönüştürülebilirliği devrelerin modelleme aşamasında büyük kolaylığı sağlamaktadır.

Bu çalışmada optimizasyon en temel olarak Genetik algoritma ile yapılmış, diğer optimizasyon teknikleri elde edilen bu sonuçlara uygulanarak yapının daha iyileştirilmesine çalışılmıştır. Uydurma devrelerinde duyarlılık da dahil olmak üzere kullanıcı açısından etken olabilecek özellikler de göz önünde tutularak hem 2 hem de 3 elemanlı yapılar kullanılmıştır.

Geliştirilen mikrodalga kuvvetlendiricilerinin gerçek işlemlerde uygulanabilirliğini göstermek amacıyla bir mikrodalga simülasyon programı ile devreler simule edilmiştir. Sınırlı bir çalışma bandı içerisinde mikrodalga kuvvetlendiricinin kazanç, gürültü ve giriş VSWR değerlerinin aynı anda optimize edilebileceği ortaya konulmuştur.

Anahtar Kelimeler: Mikrodalga Kuvvetlendirici, Genetik algoritma, Saçılma parametreleri, Uydurma devreleri, Optimizasyon.

ABSTRACT

Microwave circuits and microwave amplifiers are more preferable than the classic ones because they are easier to use. They are smaller, lighter and easy to produce. But surely, their modelling are not as easy as their usage. In this point, for modelling these circuits and amplifiers computer aid is clearly needed caused by the mass of the mathematical calculations. There are several kinds of parameters used in this step. The scattering parameters are most preferable ones in this kind of modelling. The transfer scattering parameters "[A]" are easier to use in the modelling of cascaded connected circuits because they convert the calculation of the total parameter calculation into an easier matrix multiplication. Adapting a matching circuit is easily be resembled by multiplying the total circuit [A] parameters by the matching circuit [A] parameters. And also having a very simple conversion with the [S] parameters, [A] are easiest way to use a microwave circuit.

The design of a microstrip transistor amplifiers have some problems when using the classical methods. These can be called as Bandwidth, Noise Factor and etc. So in the designing process several experimental steps are needed to be repeatedly used on different kinds of designs. This is the most important reason of the creation of the computer aided design of the microwave amplifiers.

In this study, the basis optimisation process is made by the genetic algorithm and after the other optimisation techniques are applied to the results in order to have a better design. Several needs can affect the design process as the sensitivity, so in order to include different kinds of designs the 2-part (L) matching circuits are also used with the classical 3-part matching circuits.

The circuits are simulated by a microwave simulation program in order to show the feasibility of the design in the real world. It's also shown that the gain, noise and input VSWR values can be optimised at the same time in a restricted working band.

Anahtar Kelimeler: Microwave amplifier, Genetic Algorithm, Scattering parameters, Matching Circuits, Optimisation.

1. GİRİŞ

Mikrodalga devreleri başta sadece bir alternatif olarak ortaya çıkıp daha sonra kendi başına ayrı bir araştırma alanı haline geldi. Günlük yaşama sokaktaki insan bazında yeterince girmemiş olmakla birlikte mikrodalga devrelerinin hem sanayi alanında hem de bireysel kullanım bazında önemli ve kesinlikle gözardı edilmemesi gerekiyor.

Mikrodalgaları temelde devre yapısının hantal yapısını büyük ölçüde ortadan kaldırdığı için sadece toplam sistem boyutunun küçültülmesini sağlamakla kalmayıp bazı sabit donanımların mobil kullanıma aktarılmasında büyük katkıda bulunmaktadır. Bu açıdan mikrodalga devrelerinin parlak bir geleceği olduğu net bir biçimde gözükmemektedir. Bu aşamada bu devre yapıları ön plana çıkmaktadır. Mikrodalga kuvvetlendiriciler ön plana çıkan bu temel yapılar içerisinde bizim açıımızdan en önemli olan birimlerden birisidir.

Mikrodalga kuvvetlendirici yapısının incelenmesi ve geliştirilebilmesi için bu birimlerin analiz ve sentezinin hızlı ve doğru bir biçimde yapılması gerekmektedir. Teknoloji çağında bulunduğumuz için bu işlemde bilgisayarların kullanımı kaçınılmaz bir beklenti olacaktır. Bilgisayar sistemlerinin genel olarak hâla durum bazlı mantık üzerinde çalışıyor olması mikrodalga kuvvetlendirici uygulamalarının sisteme parametre bazında aktarımını gerektirmektedir. Mikrodalga devre yapısının sadece operatör için değil aynı zamanda çoğu bilgisayar sistemi için bile karmaşık olması devre yapısının ardışık birimler halinde incelenmesi ile daha anlaşılabilir bir yapıya dönüştürülebilmektedir. Transfer saçılma parametrelerinden [A] parametreleri tam olarak bu istenileni yerine getirecek bir yapıya sahip olduğu için bu tez içerisinde özellikle bu parametrelere yer verilmiştir. Kaskad bağlı devre alt birimlere ait parametrelerin matris bazında çarpımı bize oldukça kolay bir şekilde tüm devrenin parametrelerine ulaşma imkanı tanımaktadır. Bu durumda sistem gelişimi sırasında eklenen yada değiştirilen alt birimlerin tüm devreye etkisi basit bir aritmetik işleme dönüşmektedir.

2. ÖNCEKİ ÇALIŞMALAR

Mikrodalga transistörleri ve saçılma parametreleri üzerine son 30 yıl içinde yazılmış bir çok çalışma bulunmaktadır. Bununa birlikte çalışmadaki yapının derinlemesine incelemesi yapıldığında önceki çalışmalar kapsamında sayabileceğimiz 3 adet çalışma bulunmaktadır. Bunlardan ilki çalışmanın [A] parametreleri ile ilgili kısmının temelini oluşturan ve 2001 senesinde sunmuş olduğum “Kaskad bağlı mikrodalga devrelerinin [A] parametreleri ile optimizasyonu” konulu lisans tezidir. Lisans tezindeki yaklaşım bu çalışmada yeniden ele alınmış, düzenlenmiş ve eksik yanlar giderilerek geliştirilmiştir.

Çalışmanın optimizasyon kısmının temelini ise iki tez çalışması oluşturmaktadır. Bu çalışmalardan ilki diğer bir çok teze de esin kaynağı olan 2001 yılında Aliyev tarafından hazırlanan “Performans (F , V_i , G_t) Üçlüleri Kullanılarak Geniş Bandlı Mikrodalga Kuvvetlendirici Tasarımı” konulu yüksek lisans tezidir. Bu yüksek lisans tezinde bulunan programlar daha sonradan CIS tarafından tekrar ele alınarak geliştirilmiştir.

Çalışmanın temelini oluşturan ikinci tez ise 2004 yılında Cengiz tarafından hazırlanan “Optimum Performanslı Mikrodalga kuvvetlendirici Tasarımı” konulu doktora tezidir. Elinizde bulunan tezde kısaca değinilen “Genetik algoritma”, “Performans Data yaprakları” ve “Kara kutu” gibi konular Cengiz’in doktora tezinde daha geniş olarak bulunmaktadır.

Yapının temelini oluşturan bu tezlerin dışında 1970 yılında Timothy N. Trick ve Jiri Vlach tarafından IEEE Transactions on Microwave Theory and Techniques’de yayımlanan “Computer Aided Design of Broadband Amplifiers with Complex Loads” isimli makele parametre kullanımı açısından sahip olduğu içerikle bu çalışma açısından öncü bir yapı içermektedir.

Bunun yanında 1994 yılında Dean A. Frickey tarafından IEEE Transactions on Microwave Theory and Techniques’de yayımlanan “Conversions Between S, Z, Y, h, ABCD and T Parameters which are Valid for Complex Source and Load Impedances” isimli makalesi elde edilen parametre formüllerinin daha önceden hesaplanan formüllerle karşılaştırılabilmesine imkan tanıyarak yapılan hesaplamaların doğruluğunun kontrol edilebilmesini sağlamıştır.

3. ÇALIŞMA İLE SUNULAN YENİLİKLER

Bu çalışma, daha evvelden de belirtildiği gibi sadece yeni bir çalışma sunmanın yanında aynı zamanda bilinen elde bulunan verilerin güncellenmesi amacını da taşımaktadır. Cengiz'in ve Aliyev'in tezlerindeki programlar bu tez içerisinde ele alınarak geliştirilmesi hedeflenmiştir.

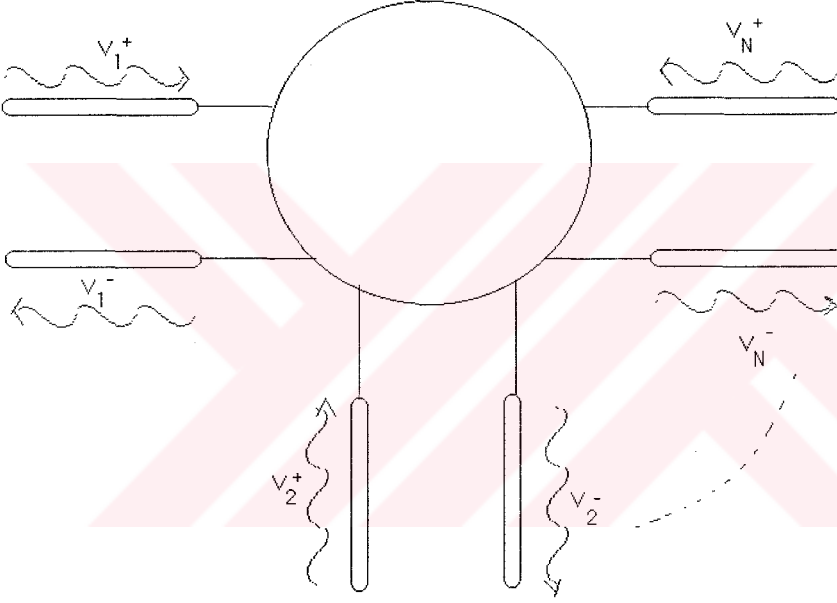
Bu çalışmadaki yenilikleri temel olarak ele alacak olursak:

- Genetik optimizasyon algoritması tamamen değiştirilmek yerine bazı küçük düzenlemelerle işlem süresi hızlandırılmıştır. Önceki çalışmalarda bulunan eşdeğerleri ile kıyaslandığında bu çalışmada bulunan program 6000 iterasyon komşuluğunda yaklaşık 38,6 kat daha hızlı sonuç verebilmektedir. Her iterasyon başına sonuç görüntülemek yerine sadece tüm işlemin sonucunu görüntüleme tabanlı bu düzenleme işlem yükünün hafifletilmesini sağlamıştır.
- Yapılan genetik optimizasyon aralığı hem hat uzunlukları hem de karakteristik empedanslar açısından tekrar ele alınmıştır. Mikro şerit hatlar için tercih edilmeyecek kadar düşük dirençler hesaplama dışı bırakılarak gereksiz işlemlerle süre kaybı ortadan kaldırılmıştır. Örneğin daha önceki çalışmalarda 0,5 ohm değerine kadar düşen aralık bu çalışmada 20 ohm ile alttan sınırlandırılmıştır. Bunu sağlamak için genetik optimizasyonda daha önce tek bir grup olarak ele alınan değişken kümesi uzunluk ve karakteristik empedans olarak ikiye ayrılmış ve genetik algoritma bu tamamen farklı sınır değerlerine sahip iki grup üzerine uygulanmıştır.
- Gerçeklenebilirlik ön plana alınmıştır. Hat uzunlukları 0,1-15 cm ve karakterteristik empedanslar 25-150 ohm aralıklarına çekilerek mikroşerit mimarisine daha uygun sonuçlar elde edilmiştir.
- Genetik optimizasyon ile bulunan sonuçlar MATLAB programının Nelder-Mead Simpleks tabanlı “fminsearch” ve Öntanımlı Konjüge Gradyant tabanlı “fmincon” komutlarından yararlanılarak optimizasyon teknikleri ile de frekans sonundan başına doğru genişleyen bir sistematik yapı ile ele alınmış ve daha da düzeltilmeye çalışılmıştır.

4. KASKAD BAĞLI İKİ-KAPILININ [S] PARAMETRELERİ İLE KARAKTERİZASYONU

4.1 [S] Parametreleri

Günümüzde mikrodalga devrelerinde hesaplamalarda kullandığımız en temel parametreler s-parametreleridir. Tez kapsamında kullandığımız diğer parametreler de bu parametreler cinsinden ifade edileceği için bu parametrelere kısaca değinilmelidir.



Şekil 4.1: N-kapılı bir Mikrodalga Devresi

V^+ : Gelen dalga

V^- : Yansıyan dalga

$$\begin{bmatrix} V_1^- \\ V_2^- \\ V_N^- \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{1N} \\ S_{21} & S_{21} & S_{2N} \\ S_{N1} & S_{N2} & S_{NN} \end{bmatrix} \begin{bmatrix} V_1^+ \\ V_2^+ \\ V_N^+ \end{bmatrix} \quad (4.1)$$

$$[V] = [V^+] [U + [S]] \quad (4.2)$$

$$[I] = \frac{[V^+]}{Z_0} [U - [S]] \quad (4.3)$$

$$[V] = [V_1 \quad V_2]^t \quad (4.4)$$

$$[V^+] = [V_1^+ \quad V_2^+]^t \quad (4.5)$$

$$[I^+] = [I_1^+ \quad I_2^+] \quad (4.6)$$

$$\begin{bmatrix} V_1^- \\ V_2^- \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} V_1^+ \\ V_2^+ \end{bmatrix} \quad (4.7)$$

4.2 Kaskad Bağlı İki-Kapılıların [ABCD] Parametreleri İle Karakterizasyonu

Bir iki-kapılının ABCD parametreleri ile aşağıdaki şekilde karakterize edilebilir:

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix} \quad (4.8a)$$

N-tane iki kapılıdan oluşan kaskad bağlı bir devrenin [ABCD] ile karakterizasyonu da

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}_1 \dots \begin{bmatrix} A & B \\ C & D \end{bmatrix}_N \begin{bmatrix} V_N \\ -I_N \end{bmatrix} \quad (4.8b)$$

şeklinde ifade edilebilir. Bu çalışmada kaskad bağlı devreler S-parametreleri ile karakterize edilmiştir. Klasik [ABCD] parametreleri ile S-parametreleri arasındaki dönüşüm aşağıdaki adımlarla elde edilir.

$$\begin{bmatrix} V_1^- \\ V_2^- \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \cdot \begin{bmatrix} V_1^+ \\ V_2^+ \end{bmatrix} \quad (4.9)$$

$$[V] = [V^+] + [V^-] \quad (4.10)$$

$$[I] = \frac{[V^+] - [V^-]}{Z_0} \quad (4.11)$$

$$V_1 = AV_2 - BI_2 \quad (4.12)$$

$$V_1^+ + V_1^- = AV_2^+ + AV_2^- - BV_2^+ / Z_0 + BV_2^- / Z_0 \quad (4.13)$$

$$V_1^- = (S_{11}V_1^+ + S_{12}V_2^+) \quad (4.14)$$

$$V_2^- = (S_{21}V_1^+ + S_{22}V_2^+) \quad (4.15)$$

$$V_1^+ + (S_{11}V_1^+ + S_{12}V_2^+) = AV_2^+ + A(S_{21}V_1^+ + S_{22}V_2^+) - BV_2^+ / Z_0 + B(S_{21}V_1^+ + S_{22}V_2^+) / Z_0 \quad (4.15)$$

$$1 + S_{11} = AS_{21} + BS_{21} / Z_0 \quad (4.16)$$

$$S_{12} = A + AS_{22} - B / Z_0 + BS_{22} / Z_0 \quad (4.17)$$

$$S_{12} = A + AS_{22} - B / Z_0 (1 - S_{22}) \quad (4.18)$$

$$(1 - S_{22})(1 + S_{11}) = AS_{21}(1 - S_{22}) + (1 - S_{22})BS_{21} / Z_0 \quad (4.19)$$

$$S_{21}S_{12} = S_{21}A + AS_{21}S_{22} - BS_{21} / Z_0 (1 - S_{22})$$

$$2AS_{21} = 1 - S_{22} + S_{11} - S_{11}S_{22} + S_{12}S_{21} \quad (4.20)$$

$$\Delta S = S_{11}S_{22} - S_{12}S_{21} \quad (4.21)$$

dersek, sonuçta;

$$A = \frac{1 - \Delta S + S_{11} - S_{22}}{2S_{21}} \quad (4.22)$$

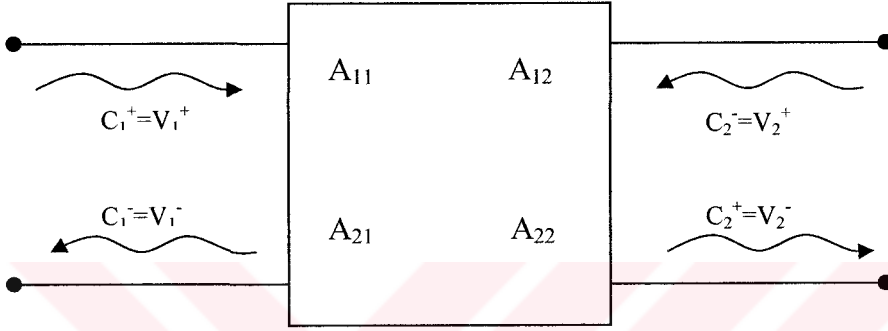
$$B = 50 \frac{(1 + \Delta S + S_{11} + S_{22})}{2S_{21}} \quad (4.23)$$

$$C = \frac{1 + \Delta S - S_{11} - S_{22}}{100S_{21}} \quad (4.24)$$

$$D = \frac{1 - \Delta S - S_{11} + S_{22}}{2S_{21}} \quad (4.25)$$

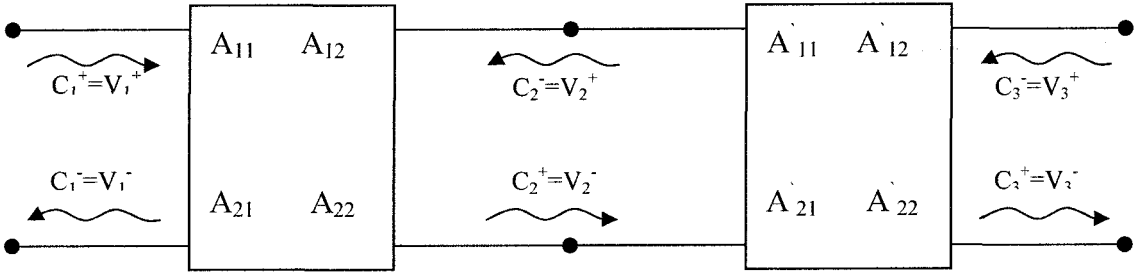
4.3 [A] Parametreleri

4.3.1 Kaskad Bağlı İki-Kapılıların [A] Parametreleri İle Karakterizasyonu



Şekil 4.2: Bir İki- Kapılıda Gelen ve Yansıyan dalgalar

$$\begin{bmatrix} V_1^+ \\ V_1^- \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} V_2^- \\ V_2^+ \end{bmatrix} \quad (4.26a)$$



Şekil 4.3: Kaskad bağlı iki İki- Kapılıda Gelen ve Yansıyan dalgalar

$$\begin{bmatrix} V_1^+ \\ V_1^- \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} A'_{11} & A'_{12} \\ A'_{21} & A'_{22} \end{bmatrix} \begin{bmatrix} V_3^- \\ V_3^+ \end{bmatrix} \quad (4.26b)$$

4.3.2 [A] - [S] Parametreleri Dönüşümleri

$$\begin{bmatrix} C_1^- \\ C_2^+ \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \cdot \begin{bmatrix} C_1^+ \\ C_2^- \end{bmatrix} \quad (4.27)$$

$$\begin{bmatrix} C_1^+ \\ C_1^- \end{bmatrix} = [A] \cdot \begin{bmatrix} C_2^+ \\ C_2^- \end{bmatrix} \quad (4.28)$$

burada s-matrisini çarpıp açarsak

$$C_1^- = S_{11} C_1^+ + S_{12} C_2^- \quad (4.29)$$

$$C_2^+ = S_{21} C_1^+ + S_{22} C_2^- \quad (4.30)$$

$$S_{21} C_1^+ = C_2^+ - S_{22} C_2^- \quad (4.31)$$

$$S_{21} C_1^- = S_{21} S_{11} C_1^+ + S_{21} S_{12} C_2^- \quad (4.32)$$

$$- S_{11} C_2^+ = - S_{11} S_{21} C_1^+ - S_{11} S_{22} C_2^- \quad (4.33)$$

Bu iki eşitliğin toplamından

$$S_{21} C_1^- = S_{11} C_2^+ + C_2^- [S_{21} S_{12} - S_{11} S_{22}] \quad (4.34)$$

$$C_1^- = \frac{S_{11} C_2^+}{S_{21}} + C_2^- \frac{[S_{21} S_{12} - S_{11} S_{22}]}{S_{21}} \quad (4.35)$$

$$C_1^+ = \frac{C_2^+}{S_{21}} + C_2^- \left[-\frac{S_{22}}{S_{21}} \right] \quad (4.36)$$

$$\begin{bmatrix} C_1^+ \\ C_1^- \end{bmatrix} = \begin{bmatrix} \frac{1}{S_{21}} & -\frac{S_{22}}{S_{21}} \\ \frac{S_{11}}{S_{21}} & \frac{S_{21} S_{12} - S_{11} S_{22}}{S_{21}} \end{bmatrix} \cdot \begin{bmatrix} C_2^+ \\ C_2^- \end{bmatrix} \quad (4.37)$$

buradan;

$$A_{11} = \frac{1}{S_{21}} \quad (4.38)$$

$$A_{12} = -\frac{S_{22}}{S_{21}} \quad (4.39)$$

$$A_{21} = \frac{S_{11}}{S_{21}} \quad (4.40)$$

$$A_{22} = \frac{S_{12}S_{21} - S_{11}S_{22}}{S_{21}} \quad (4.41)$$

4.3.3 [S] - [A] Parametreleri Dönüşümleri

(4.38)- (4.41) formüllerinden yararlanılarak aşağıdaki [S]- [A] parametre dönüşümleri elde edilebilir:

$$S_{11} = \frac{A_{21}}{A_{11}} \quad (4.42)$$

$$S_{21} = \frac{1}{A_{11}} \quad (4.43)$$

$$S_{12} = \frac{A_{22}A_{11} - A_{21}A_{12}}{A_{11}} \quad (4.44)$$

$$S_{22} = -\frac{A_{12}}{A_{11}} \quad (4.45)$$

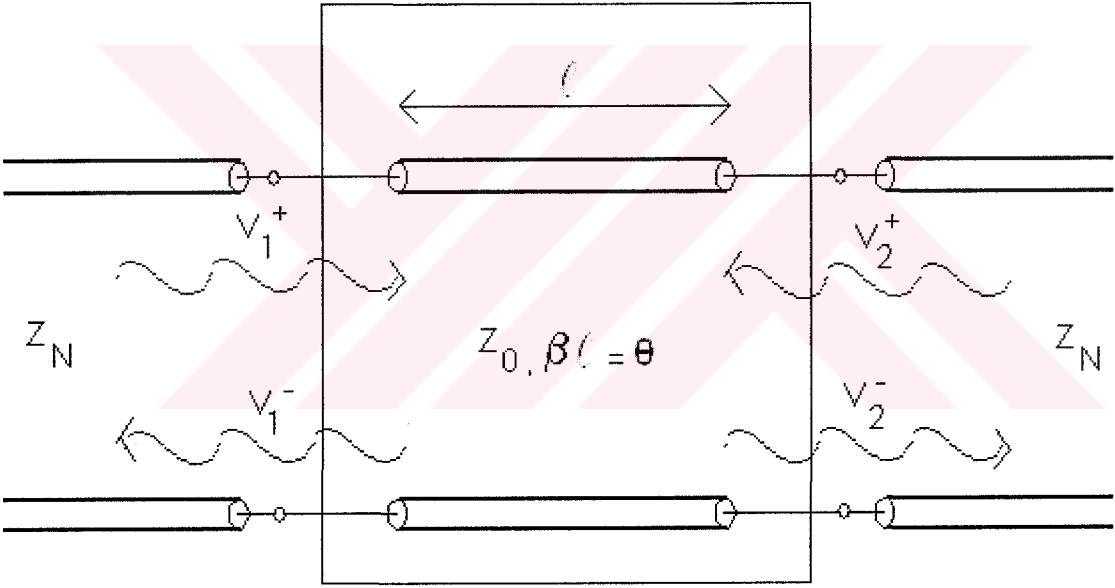
5. SİSTEM TANITIMI

5.1 Sistemde Kullanılan Temel Elemanlar Ve Karakterizasyonu

Her bir temel uydurma elemanı dağılmış parametrelili kayıpsız ve resiprok olup, Z_0 karakteristik empedansı ve fiziksel uzunluk ℓ parametreleri ile karakterize edilecektir. Buna göre iki tip temel transmisyon hat iki-kapılısı kullanılmıştır:

- 1: Seri transmisyon hat iki-kapılısı
- 2: Paralel transmisyon hat iki-kapılısı

5.1.1 Seri Transmisyon Hat İki-Kapılısı

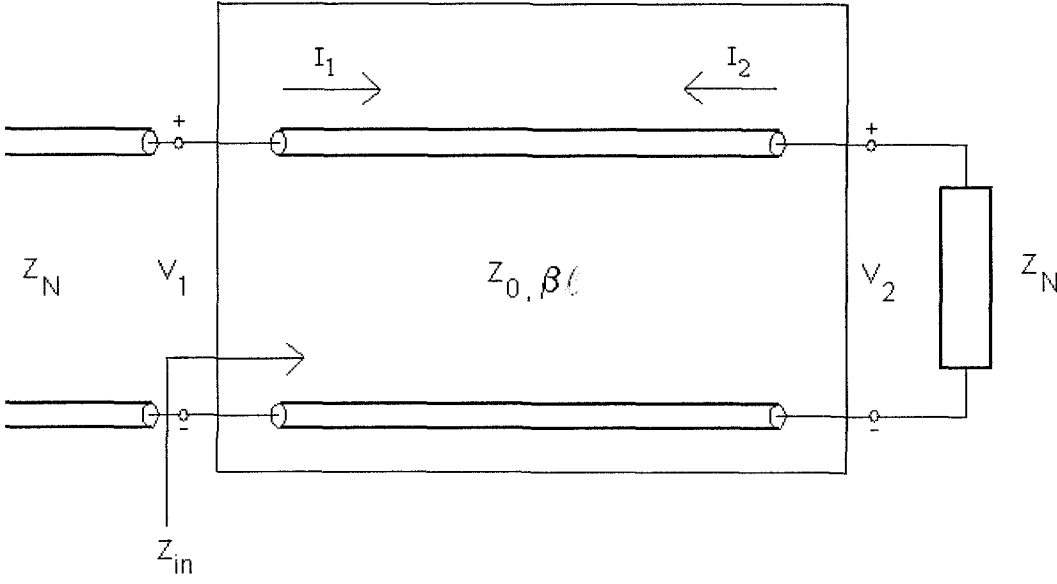


Şekil 5.1.1.1 Seri transmisyon hat iki-kapılısı

Eğer çıkış hattı uydurulmuşsa, o zaman:

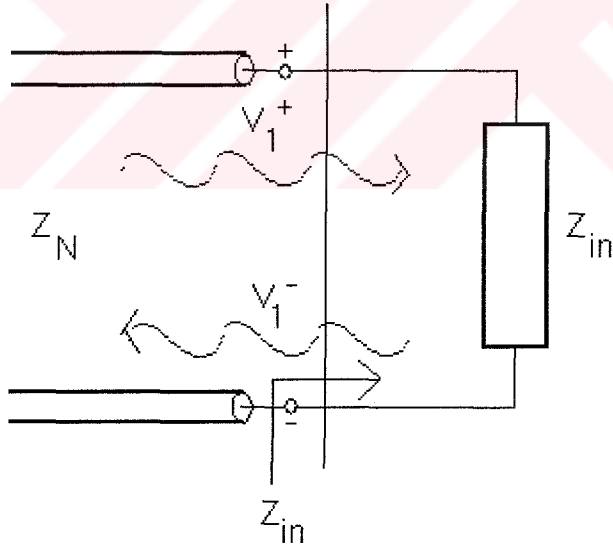
$$\frac{V_1^-}{V_1^+} = \frac{\overline{V_1^-}}{\overline{V_1^+}} = S_{11} = \frac{Z_{in} - Z_N}{Z_{in} + Z_N} \quad (5.1)$$

olarak bulunur. Burada Z_{in} , Z_N referans empedansınca sonlandırılan hattın giriş empedansıdır.



Şekil 5.1.1.2 Z_N referans empedansı ile sonlandırılmış transmisyon hattı

$$Z_{in} = Z_0 \frac{Z_N + jZ_0 \tan \beta\ell}{Z_0 + jZ_N \tan \beta\ell} \quad (5.2)$$



Şekil 5.1.1.3 Z_{in} sonlandırılmalı eşdeğer

$\Gamma_{in} = S_{11}$ olmak üzere

$$S_{11} = \frac{Z_0 \frac{Z_N + jZ_0 \tan \beta \ell}{Z_0 + jZ_N \tan \beta \ell} - Z_N}{Z_0 \frac{Z_N + jZ_0 \tan \beta \ell}{Z_0 + jZ_N \tan \beta \ell} + Z_N} = \frac{jZ_0^2 \tan \beta \ell - jZ_N^2 \tan \beta \ell}{jZ_0^2 \tan \beta \ell + jZ_N^2 \tan \beta \ell + 2Z_0 Z_N} \quad (5.3)$$

Pay ve paydayı $j \tan \beta \ell$ ($\beta \ell \neq 0$) ile böldüğümüz takdirde S_{11} :

$$S_{11} = \frac{Z_0^2 - Z_N^2}{Z_0^2 + Z_N^2 + j2Z_0 Z_N \cot \beta \ell} \quad (5.4)$$

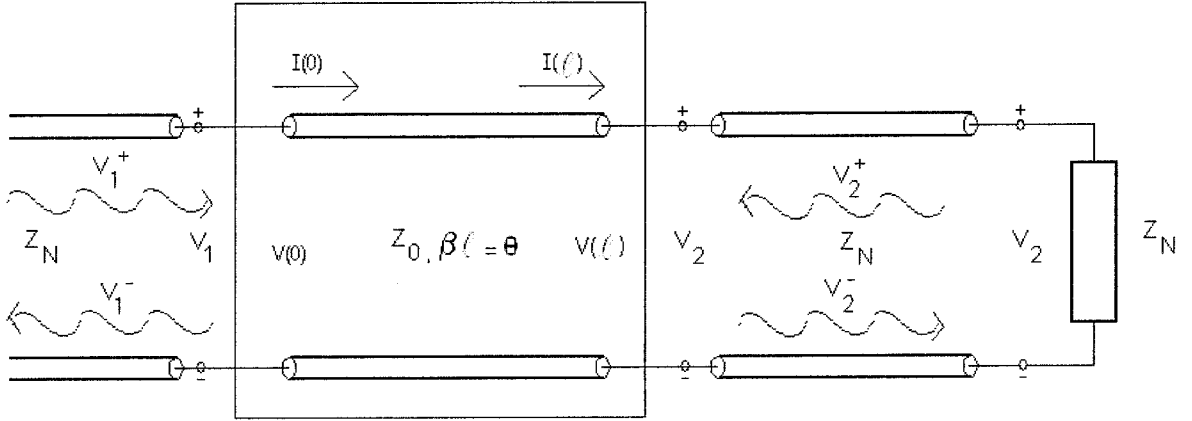
S_{22} için benzer bir mantık yürütülmektedir. Giriş hattı uydurulduğunda S_{22} , Z_N referans empedansı ile sonlandırılmış hattın yansıma katsayısı halini alır ve S_{11} 'e eşit olur. Bu durumda S_{22} :

$$S_{22} = S_{11} = \frac{Z_0^2 - Z_N^2}{Z_0^2 + Z_N^2 + j2Z_0 Z_N \cot \beta \ell} \quad (5.5)$$

S_{21} 'i bulmak için tekrar çıkış hattının uydurulmuş olduğu varsayılması gerekir. Bu durumda giriş hattında:

$$V_1 = V_1^+ + V_1^- = V_1^+ (1 + S_{11}) \quad (5.6)$$

$$I_1 = Y_N (V_1^+ - V_1^-) = Y_N V_1^+ (1 - S_{11}) \quad (5.7)$$



Şekil 5.1.1.4 S_{21} 'i bulmak için kullanılan devre

Giriş düğümü için

$$I_1 = I(0) = I^+ - I^- = I^+ (1 - \Gamma_L e^{-j2\beta \ell}) \quad (5.8)$$

$$I^+ = \frac{Y_N V_1^+ (1 - S_{11})}{1 - \Gamma_L e^{-j2\beta\ell}} \quad (5.9)$$

Sonuçta

$$I(\ell) = I^+ e^{-j\beta\ell} (1 - \Gamma_L) = \frac{Y_N V_1^+ (1 - S_{11}) e^{-j\beta\ell}}{1 - \Gamma_L e^{-j2\beta\ell}} (1 - \Gamma_L) \quad (5.10)$$

$$V_2 = V_2^- = I(\ell) Z_N = \frac{V_1^+ (1 - S_{11})}{e^{j\beta\ell} - \frac{Z_N - Z_0}{Z_N + Z_0} e^{-j\beta\ell}} \left(1 - \frac{Z_N - Z_0}{Z_N + Z_0} \right) \quad (5.11)$$

ki burada

$$1 - S_{11} = 1 - \frac{Z_0^2 - Z_N^2}{Z_0^2 + Z_N^2 - jZ_0 Z_N \cot \beta\ell} \quad (5.12)$$

$$1 - S_{11} = \frac{Z_0^2 + Z_N^2 - jZ_0 Z_N \cot \beta\ell - Z_0^2 + Z_N^2}{Z_0^2 + Z_N^2 - jZ_0 Z_N \cot \beta\ell} \quad (5.13)$$

$$1 - S_{11} = \frac{2Z_N^2 - jZ_0 Z_N \cot \beta\ell}{Z_0^2 + Z_N^2 - jZ_0 Z_N \cot \beta\ell} \quad (5.14)$$

$$S_{21} = S_{12} = \frac{V_2^-}{V_1^+} = \frac{2Z_N^2 - jZ_0 Z_N \cot \beta\ell}{Z_0^2 + Z_N^2 - jZ_0 Z_N \cot \beta\ell} \frac{2Z_0}{(Z_0 + Z_N)e^{j\beta\ell} - (Z_N - Z_0)e^{-j\beta\ell}} \quad (5.15)$$

$$A = Z_0^2 + Z_N^2 - jZ_0 Z_N \cot \beta\ell \quad (5.16)$$

$$B = 2Z_N^2 - jZ_0 Z_N \cot \beta\ell \frac{2Z_0}{(Z_0 + Z_N)e^{j\beta\ell} - (Z_N - Z_0)e^{-j\beta\ell}} \quad (5.17)$$

$$B = \frac{2Z_0 Z_N \left(2Z_N - jZ_0 \frac{\cos \beta\ell}{\sin \beta\ell} \right)}{Z_0 (e^{j\beta\ell} + e^{-j\beta\ell}) - Z_N (e^{j\beta\ell} - e^{-j\beta\ell})} \quad (5.18)$$

$$e^{\pm j\beta\ell} = \cos \beta\ell \pm j \sin \beta\ell \quad (5.19)$$

$$\cos \beta\ell = \frac{1}{2} (e^{j\beta\ell} + e^{-j\beta\ell}) \quad (5.20)$$

$$\sin \beta \ell = \frac{1}{2j} (e^{j\beta \ell} - e^{-j\beta \ell}) \quad (5.21)$$

$$\csc \beta \ell = \frac{1}{\sin \beta \ell} \quad (5.22)$$

Bu değerleri B eşitliğinde yerine koyduğumuz taktirde

$$B = \frac{2Z_0 Z_N \left(2Z_N + 2Z_0 \frac{e^{j\beta \ell} + e^{-j\beta \ell}}{e^{j\beta \ell} - e^{-j\beta \ell}} \right)}{Z_0 (e^{j\beta \ell} + e^{-j\beta \ell}) - Z_N (e^{j\beta \ell} - e^{-j\beta \ell})} \quad (5.23)$$

$$B = 4 \frac{Z_0 Z_N}{e^{j\beta \ell} - e^{-j\beta \ell}} \frac{Z_0 (e^{j\beta \ell} + e^{-j\beta \ell}) - Z_N (e^{j\beta \ell} - e^{-j\beta \ell})}{Z_0 (e^{j\beta \ell} + e^{-j\beta \ell}) - Z_N (e^{j\beta \ell} - e^{-j\beta \ell})} \quad (5.24)$$

$$B = \frac{4Z_0 Z_N}{2j \sin \beta \ell} = -2Z_0 Z_N j \csc \beta \ell \quad (5.25)$$

$$S_{21} = \frac{V_2^-}{V_1^+} = -e^{-j\beta \ell} (1 - S_{11}) \quad (5.26)$$

$$1 - S_{11} = \frac{Z_0^2 + Z_N^2 - jZ_0 Z_N \cot \beta \ell - Z_0^2 + Z_N^2}{Z_0^2 + Z_N^2 - jZ_0 Z_N \cot \beta \ell} \quad (5.27)$$

$$1 - S_{11} = \frac{2Z_N^2 - jZ_0 Z_N \cot \beta \ell}{Z_0^2 + Z_N^2 - jZ_0 Z_N \cot \beta \ell} \quad (5.28)$$

$$S_{12} = S_{21} = \frac{-2jZ_N Z_0 \csc \beta \ell}{Z_0^2 + Z_N^2 - j2Z_0 Z_N \cot \beta \ell} \quad (5.29)$$

Sonuçta seri transmisyon hat iki kapılısının S-parametreleri

$$S_{11} = S_{22} = \frac{Z_0^2 - Z_N^2}{Z_0^2 + Z_N^2 + j2Z_0 Z_N \cot \beta \ell} \quad (5.30)$$

$$S_{12} = S_{21} = \frac{-2jZ_N Z_0 \csc \beta \ell}{Z_0^2 + Z_N^2 - j2Z_0 Z_N \cot \beta \ell} \quad (5.31)$$

olarak hesaplanmıştır. Bu sonuçları S-A parametre değişim formüllerini kullanarak A-parametrelerini elde edebiliriz. Yapılan dönüşüm sonucunda

$$A_{11} = \frac{2 \cos \beta \ell Z_N + jZ_0 \sin \beta \ell + \frac{j \sin \beta \ell}{Z_0} Z_N^2}{2 Z_N} \quad (5.32)$$

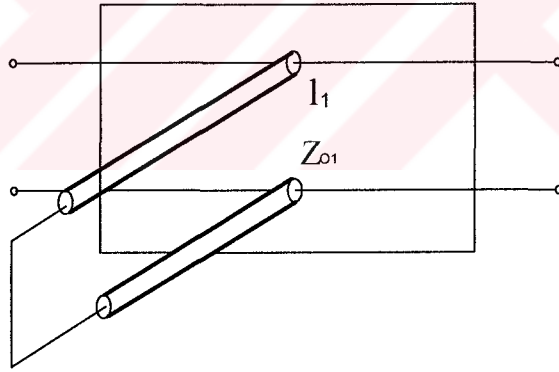
$$A_{12} = \frac{-jZ_0 \sin \beta \ell + \frac{j \sin \beta \ell}{Z_0} Z_N^2}{2 Z_N} \quad (5.33)$$

$$A_{21} = \frac{jZ_0 \sin \beta \ell - \frac{j \sin \beta \ell}{Z_0} Z_N^2}{2 Z_N} \quad (5.34)$$

$$A_{22} = \frac{2 \cos \beta \ell Z_N - jZ_0 \sin \beta \ell - \frac{j \sin \beta \ell}{Z_0} Z_N^2}{2 Z_N} \quad (5.35)$$

olarak karşımıza çıkmaktadır.

5.1.2 Paralel Transmisyon Hat İki-Kapılısı



Şekil 5.1.2 Paralel transmisyon hat iki-kapılısı

Burada Y sonu kısa devre edilmiş hattın giriş iletkenliği olarak ele alındığında:

$$Z = jZ_0 \tan \beta \ell \Rightarrow Y = -jY_0 \cot \beta \ell \quad (5.36)$$

$$S_{11} = \left. \frac{b_1}{a_1} \right]_{a_2 \equiv 0} \quad \text{için devre}$$

$$S_{11} = \Gamma_{in} = \frac{\frac{1}{Y_{in}} - \frac{1}{Y_N}}{\frac{1}{Y_{in}} + \frac{1}{Y_N}} \quad (5.37)$$

$$S_{11} = \frac{Y_N - Y_{in}}{Y_N + Y_{in}} = \frac{Y_N - Y - Y_N}{Y_N + Y + Y_N} = \frac{-Y}{2Y_N + Y} = \frac{jY_0 \cot \beta \ell}{2Y_N - jY_0 \cot \beta \ell} \quad (5.38)$$

S_{22} yi elde etmek için S_{11} 'i elde ettiğimiz mantıktan yola çıkarsak

Sonuçta

$$S_{22} = \left. \frac{b_2}{a_2} \right]_{a_1 \equiv 0} = S_{11} = \frac{jY_0 \cot \beta \ell}{2Y_N - jY_0 \cot \beta \ell} \quad (5.39)$$

$$a_1 = \frac{V_1^+}{\sqrt{Z_N}} \Rightarrow |a_1|^2 = \frac{|V_1^+|^2}{Z_N} \quad (5.40)$$

$$S_{21} = \frac{b_2}{a_2} = 1 + S_{11} \quad (5.41)$$

$$S_{21} = 1 + S_{11} = \frac{2Y_N}{2Y_N - jY_0 \cot \beta \ell} = S_{12} \quad (5.42)$$

Paralel transmisyon hat iki kapılısının S-parametreleri

$$S_{11} = S_{22} = \frac{jY_0 \cot \beta \ell}{2Y_N - jY_0 \cot \beta \ell} \quad (5.43)$$

$$S_{12} = S_{21} = \frac{2Y_N}{2Y_N - jY_0 \cot \beta \ell} \quad (5.44)$$

ve A-parametreleri

$$A_{11} = \frac{2 - \frac{jZ_N}{Z_0 \tan \beta \ell}}{2} \quad (5.45)$$

$$A_{12} = \frac{-\frac{jZ_N}{Z_0 \tan \beta \ell}}{2} \quad (5.46)$$

$$A_{21} = \frac{\frac{jZ_N}{Z_0 \tan \beta \ell}}{2} \tag{5.47}$$

$$A_{22} = \frac{2 + \frac{jZ_N}{Z_0 \tan \beta \ell}}{2} \tag{5.48}$$

olarak bulunmaktadır

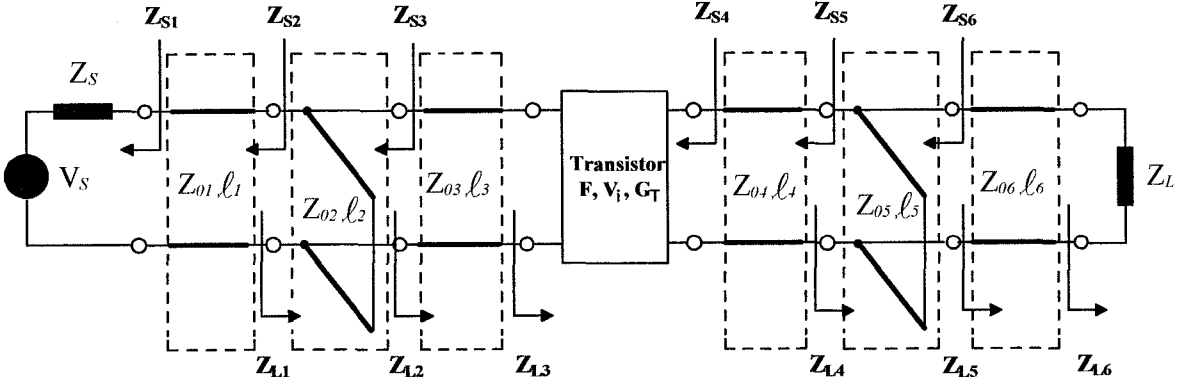
.



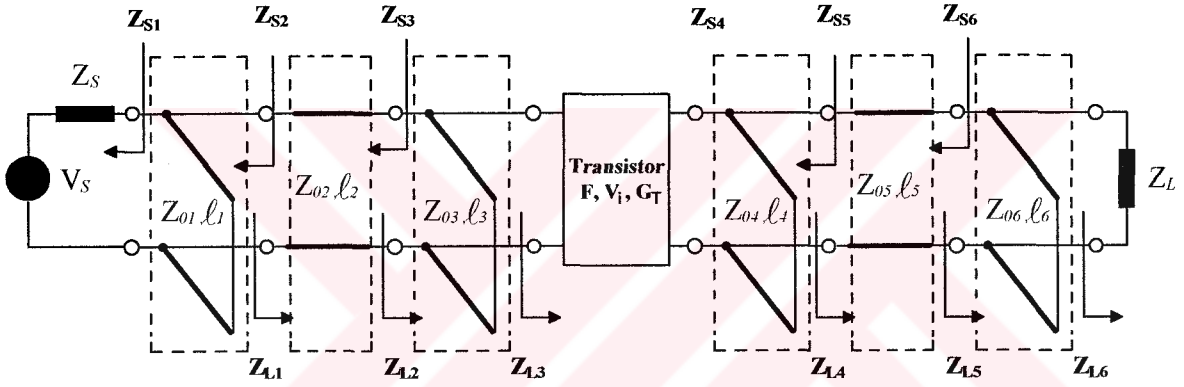
5.2 “T”, “Π” VE “L” Uydurma Devreleri İle Tek Transistorlu Temel Kuvvetlendirici Devreleri

Bu tezde ele alınan problem mikrodalga kuvvetlendirici devreleri iki-kapılı diyagramları Şekil 5.2.1-5.2.6’te verilmiştir ve her birinin giriş ve çıkışında sırasıyla aşağıdaki uydurma devreleri kullanılmıştır:

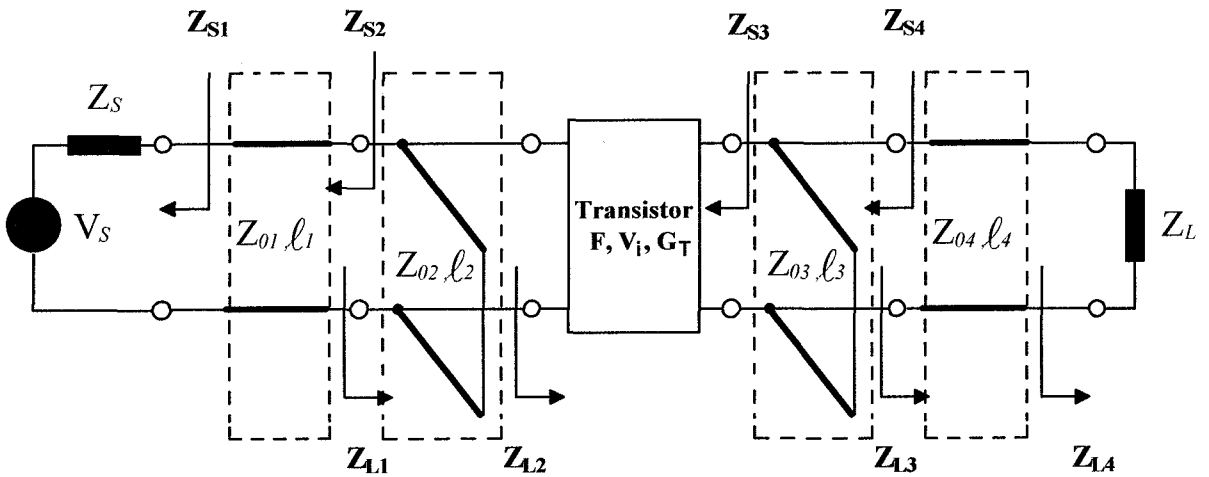
- 1: “T” tipi uydurma devreleri
- 2: “Π” tipi uydurma devreleri
- 3: “L” tipi seri+ paralel uydurma devreleri
- 4: “L” tipi paralel+seri uydurma devreleri
- 5: Giriş “T” tipi çıkış “L” tipi seri+ paralel uydurma devreleri
- 6: Giriş “Π” tipi çıkış “L” tipi seri+ paralel uydurma devreleri



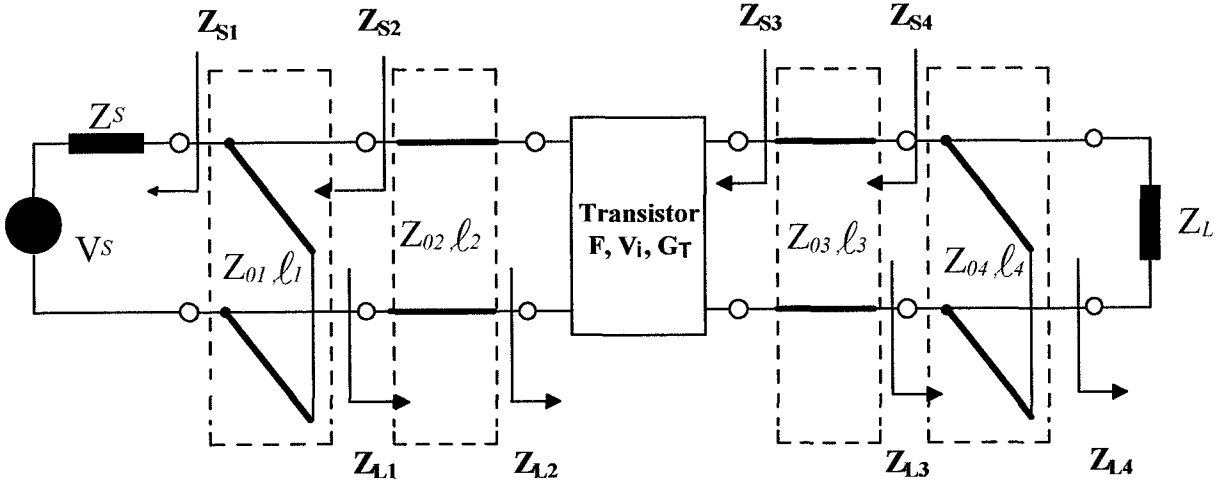
Şekil 5.2.1 T- tipi uydurma devreli tek transistorlu kuvvetlendirici



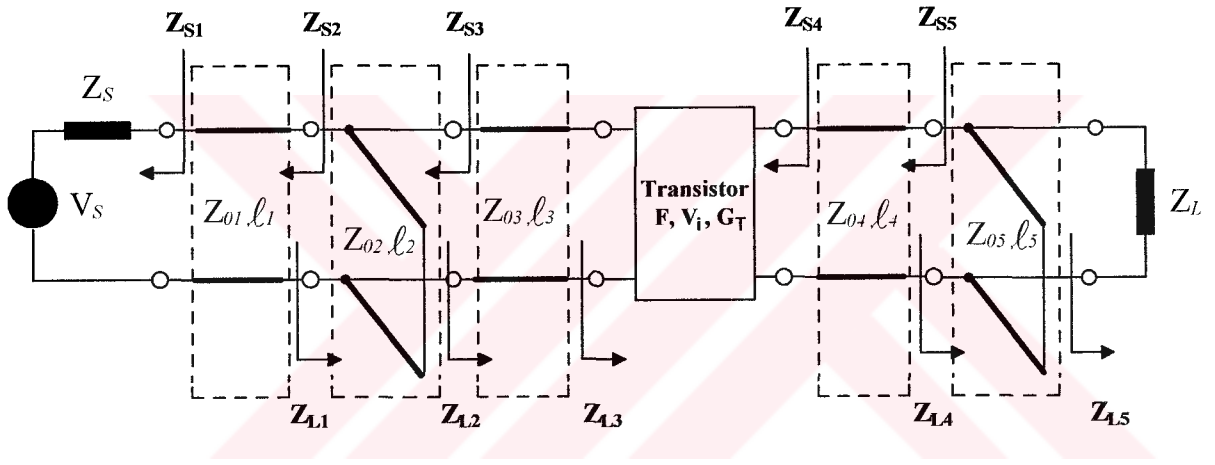
Şekil 5.2.2 Π - tipi uydurma devreli tek transistorlu kuvvetlendirici



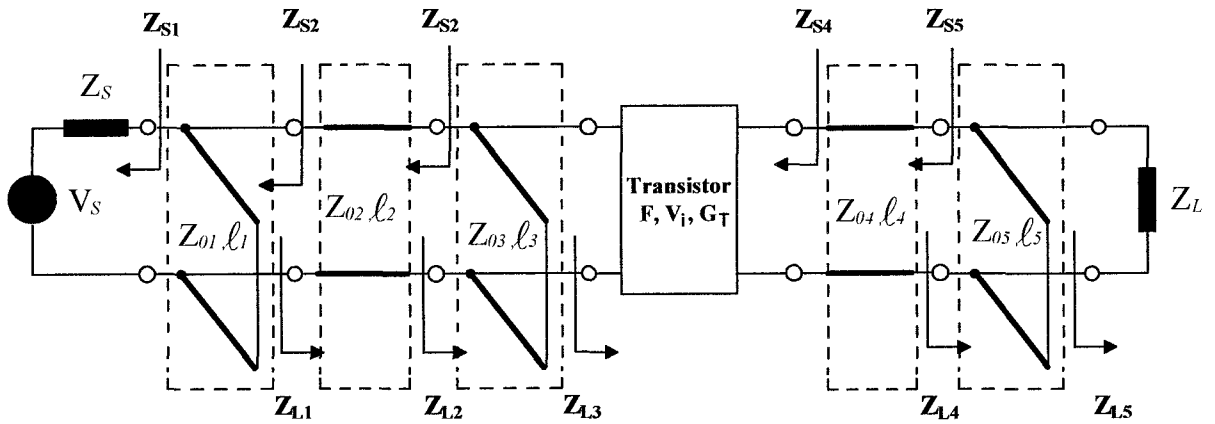
Şekil 5.2.3 L- tipi (seri + paralel) uydurma devreli tek transistorlu kuvvetlendirici



Şekil 5.2.4 L- tipi (paralel + seri) uydurma devreli tek transistörli kuvvetlendirici



Şekil 5.2.5 Giriş "T" tipi çıkış "L" tipi (seri+ paralel) uydurma devreli tek transistörli kuvvetlendirici



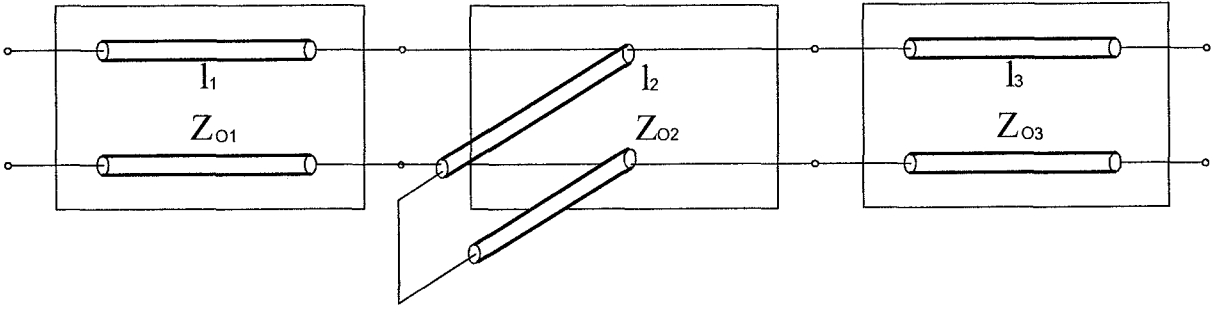
Şekil 5.2.6 Giriş "T" tipi çıkış "T" tipi (seri+ paralel) uydurma devreli tek transistörli kuvvetlendirici

5.3 Uydurma Devreleri

5.3.1 “T” Devresi

“T” devresi seri , şönt , seri hat iki-kapılılarının kaskad bağlanması sonucu oluşturulduğundan devrenin $[A]$ matrisi aşağıdaki gibi ifade edilebilir (Şekil 5.3):

$$[A]_T = [A]_{SD} \times [A]_{PD} \times [A]_{SD} \quad (5.49)$$

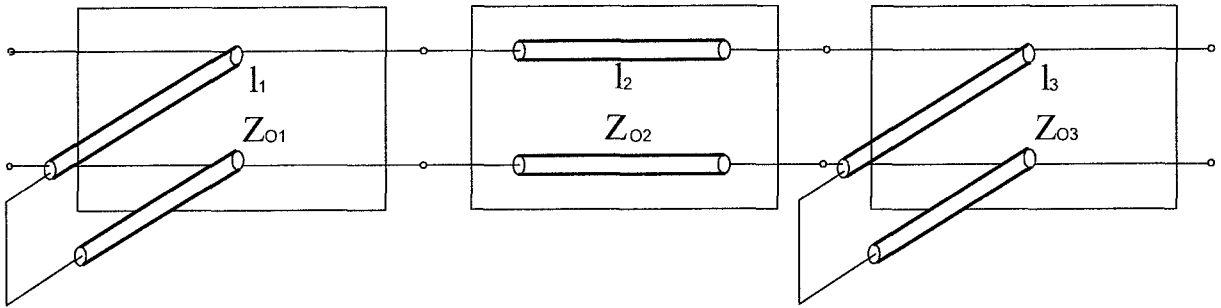


Şekil 5.3 “T” tipi uydurma devresi

5.3.2 “Π” Devresi

“Π” devresi paralel, seri, paralel hat iki-kapılılarının kaskad bağlanması sonucu oluşturulduğundan devrenin $[A]$ matrisi aşağıdaki gibi ifade edilebilir (Şekil 5.4):

$$[A]_{\Pi} = [A]_{PD} \times [A]_{SD} \times [A]_{PD} \quad (5.50)$$

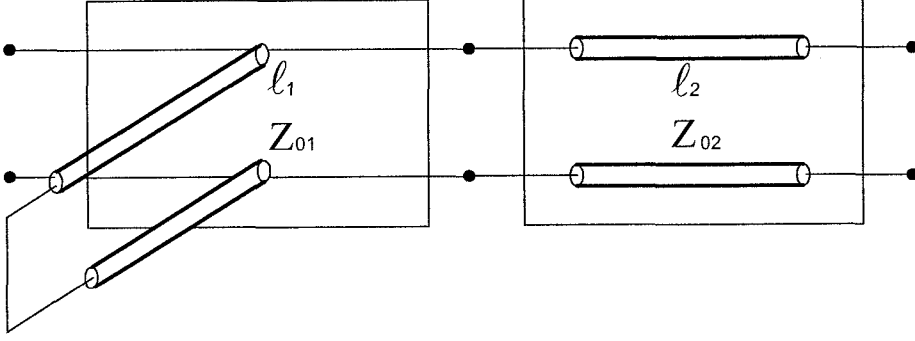


Şekil 5.4 “Π” tipi uydurma devresi

5.3.3 Paralel + Seri “L” Devresi

Paralel+seri “L” devresi, sırasıyla paralel ve seri transmisyon hat iki-kapılılarının kaskad bağlanmaları sonucu olduğundan devrenin $[A]$ matrisi aşağıdaki gibi ifade edilebilir (Şekil 5.5):

$$[A]_{L_{ps}} = [A]_{PD} \times [A]_{SD} \quad (5.51)$$

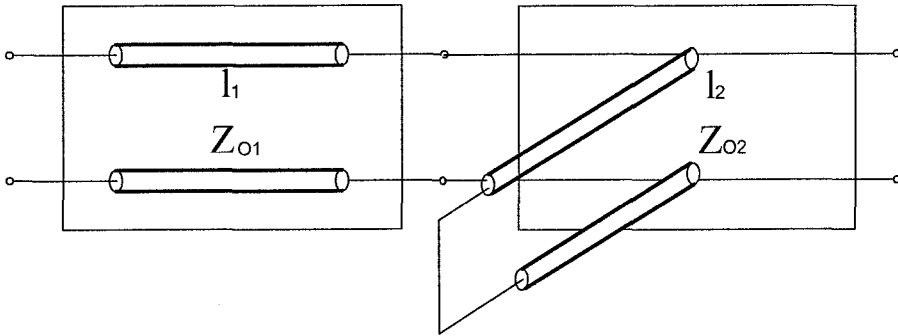


Şekil 5.5 Paralel + Seri “L” uydurma devresi

5.3.4 Seri + Paralel “L” Devresi

Paralel+seri “L” devresi, sırasıyla paralel ve seri transmisyon hat iki-kapılılarının kaskad bağlanmaları sonucu olduğundan devrenin $[A]$ matrisi aşağıdaki gibi ifade edilebilir (Şekil 5.6):

$$[A]_{L_{sp}} = [A]_{SD} \times [A]_{PD} \quad (5.52)$$



Şekil 5.6 Seri + Paralel transmisyon hattı

5.4 Aktif Eleman Karakterizasyonu Ve Performans Ölçü Fonksiyonları

5.4.1 Transistor İşaret Ve Gürültü Davranışı

Bu çalışmada aktif eleman olarak mikrodalga transistörü kullanılmıştır. Bir mikrodalga transistörünün küçük-ışaret ve gürültü davranışı, genellikle saçılma [S] ve gürültü [N] parametreleri ile karakterize edilen bir iki-kapılı ile modellenenir. [S] ve [N] parametreleri üretici firma tarafından transistörün konfigürasyon tipi CT, kutuplama koşulları V_{DS} ve I_{DS} ve çalışma frekansı f ’den oluşan çalışma uzayının sonlu sayıda noktalarındaki ölçümler sonucu “Data Yaprakları” halinde verilmektedir. “Soft Hesaplama” teknikleri ile [S] ve [N] parametreleri, bütün çalışma uzayına genişletilmelidir. Bu tezde transistörün yeniden soft modellemesini yapmak yerine Cengiz (2004) tarafından ortaya konulan yapı kabul edilmiştir. Cengiz transistörün soft modellenmesinde “Yapay Sinir Ağı” ve “Bulanık Mantık” tekniklerini kullanmıştır.

Transistörü ölçüm sonucu doğrudan karakterize eden [S] ve [N] parametreleri kullanılan devre konfigürasyonuna bağlı olarak uygun diğer tip karakterizasyon parametrelerine dönüştürülebilir. Örneğin bu çalışmada, transistör performans karakterizasyonu için [S] ve [N], [Z] (açık devre) parametrelerine, kaskad bağlı bir kuvvetlendirici devrenin analizi ve sentezi için de A ya da ABCD parametrelerine dönüştürülmüştür. Buna göre, Şekil 5.7’deki transistör iki-kapılısı için V-I denklemleri matrisiyel formda aşağıdaki şekilde ifade edilebilir:

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \quad (5.53)$$

Ayrıca transistörün sonlandırılma denklemleri

$$V_2 = -I_2 Z_L, \quad V_1 = V_s - I_1 Z_s, \quad (5.54)$$

ve Z_i giriş ve Z_0 çıkış empedansları da

$$\begin{aligned} Z_i &= \frac{V_1}{I_1} = z_{11} - \frac{z_{12} z_{21}}{z_{22} + Z_L}, \\ Z_0 &= \frac{V_2}{I_2} = z_{22} - \frac{z_{12} z_{21}}{z_{11} + Z_s} \end{aligned} \quad (5.55)$$

şeklinde ifade edilebilir.

5.4.2 Transistor Performans Ölçü Fonksiyonları

Bir transistor iki-kapılı performansı genellikle üç temel fonksiyon ile ölçülebilir: Transduser Güç Kazancı G_T , Gürültü faktörü F ve Giriş VSWR fonksiyonu V_i 'dir. Bu fonksiyonlar transistor $[Z]$ parametreleri, kaynak ve yük sonlandırmaları cinsinden aşağıdaki şekilde ifade edilebilir (Collins):

$$G_T = \frac{P_L}{P_{AVS}} \quad (5.56)$$

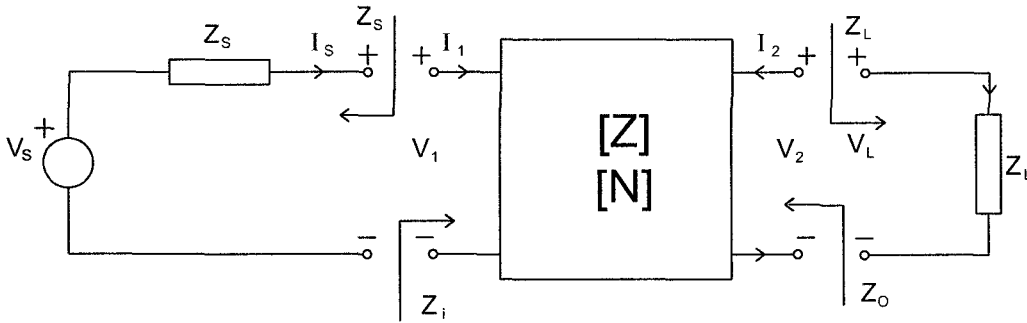
$$G_T \{R_s, X_s, R_L, X_L\} = \frac{4R_s R_L |Z_{21}|^2}{|(z_{11} + Z_s)(z_{22} + Z_L) - z_{12} z_{21}|^2}$$

$$F = \frac{(S/N)_i}{(S/N)_o} = F \{R_s, X_s\} = F_{\min} + \frac{R_N}{|Z_{opt}|^2} \frac{|Z_s - Z_{opt}|^2}{R_s} \quad (5.57)$$

$$\text{Giriş VSWR} = V_i \{R_s, X_s, R_L, X_L\} = \frac{1 + |\rho_i|^2}{1 - |\rho_i|^2}, \rho_i = \frac{Z_s - Z_i^*}{Z_s + Z_i} \quad (5.58)$$

Ayrıca, burada kaynak Z_s ve yük Z_L pasif sonlandırmaları transistorun kararlı çalışmasını sağlayacak şekilde tayin edilmelidir:

$$\text{Re}\{Z_i\} \geq 0, \text{ ve } \text{Re}\{Z_o\} \geq 0 \quad (5.59)$$



Şekil 5.7 Mikrodalga transistorun iki-kapılı modeli

Tayin edilmiş Z_S ve Z_L sonlandırmaları, transistörli kuvvetlendiricide verilen kaynak Z_g ve alıcı Z_r sonlandırmalarından sırası ile giriş ve çıkış uydurma devreleri ile elde edilirler. Bir kuvvetlendirici devresinde kullanılan uydurma devreleri aynı zamanda kayıpsız olmalı ve transistör için talep edilen çalışma bandını karşılamalıdır. Bu çalışmada uydurma devrelerinde kayıpsız ve resiprok dağılmış parametrelili elemanlar -transmisyon hat parçaları- kullanılmıştır. Sonuç olarak kuvvetlendiricinin performansını tamamiyle aktif eleman tayin ettiği söylenilebilir.



6. MİKRODALGA KUVVETLENDİRİCİSİ GERÇEKLENEBİLİR TASARIM UZAYI

6.1 Mikrodalga Kuvvetlendiricisi Optimizasyonu İki Temel Problemi

Bir mühendislik optimizasyon probleminde genellikle iki temel problem bulunmaktadır:

- “Gerçeklenebilir Tasarım Değişkenleri ve Hedefleri” cinsinden “Gerçeklenebilir Tasarım Uzayı”nın tayini;
- Optimizasyonu yöneten Hata (Amaç) fonksiyonunun “Gerçeklenebilir Tasarım Uzayı” içinde yer alan “Global Minimumu”nın tayini.

Bu çalışmadaki mikrodalga kuvvetlendiricisi optimizasyonunda “Tasarım Değişkenleri Uzayı”, uydurma devrelerinde kullanılan hat fiziksel uzunluk ve karakteristik empedanslarıdır ve $\mathbf{\Theta} = [\ell_1 \ell_2 \ell_3 \ell_4 \ell_5 \ell_6 Z_{01} Z_{02} Z_{03} Z_{04} Z_{05} Z_{06}]^T$ tasarım (optimizasyon) değişken vektörü şeklinde ifade edilebilir.

Problemde söz konusu olan “Gerçeklenebilir Tasarım” uzayı için “Tasarım Değişkenleri”;

- Transistorun hedef çalışma bandı B,
 - Devrenin gerçekleştirileceği teknolojinin alt ve üst sınırlamaları
- ele alınarak belirlenebilir.

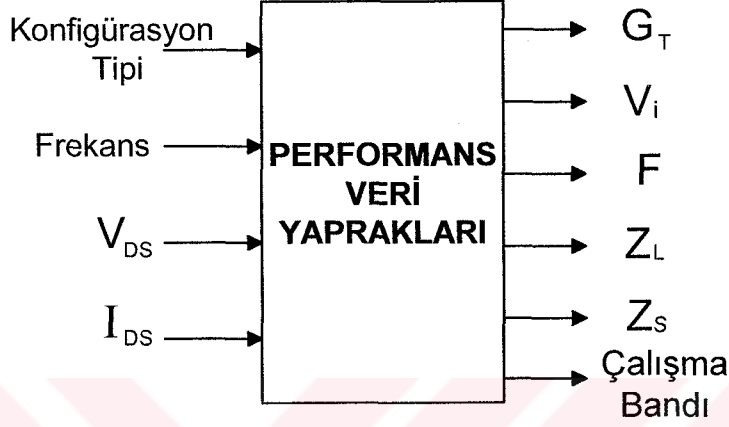
“Tasarım Hedef” alt-uzayı;

- Transistor “Performans Karakterizasyonu” (Güneş, 1994) teorisi uygulanması sonucu, (CT, V_{DS} , I_{DS} ve f) çalışma domeninde elde edilen uyumlu (F, V_i , G_T) üçlülere,
- (Z_S , Z_L) sonlandırmaları kullanılarak tayin edilebilmektedir.

Bu çalışmada “Tasarım Hedef” alt-uzayı daha spesifik bir yaklaşımla uyumlu (F, V_i , G_T) üçlülerinden çok (Z_S , Z_L) sonlandırmaları kullanılarak tayin edilmesi üzerine yoğunlaşmıştır.

6.2 Gerçeklenebilir Tasarım Hedef Uzayı Tayini: Performans Data Yaprakları

Bir mikrodalga transistor kuvvetlendiricisi için “Gerçeklenebilir Tasarım Hedef Uzayı”, aktif elemanın “Performans Data Yaprakları” isimlendirilebilecek aşağıdaki “Kara Kutu” modeli ile elde edilebilir:

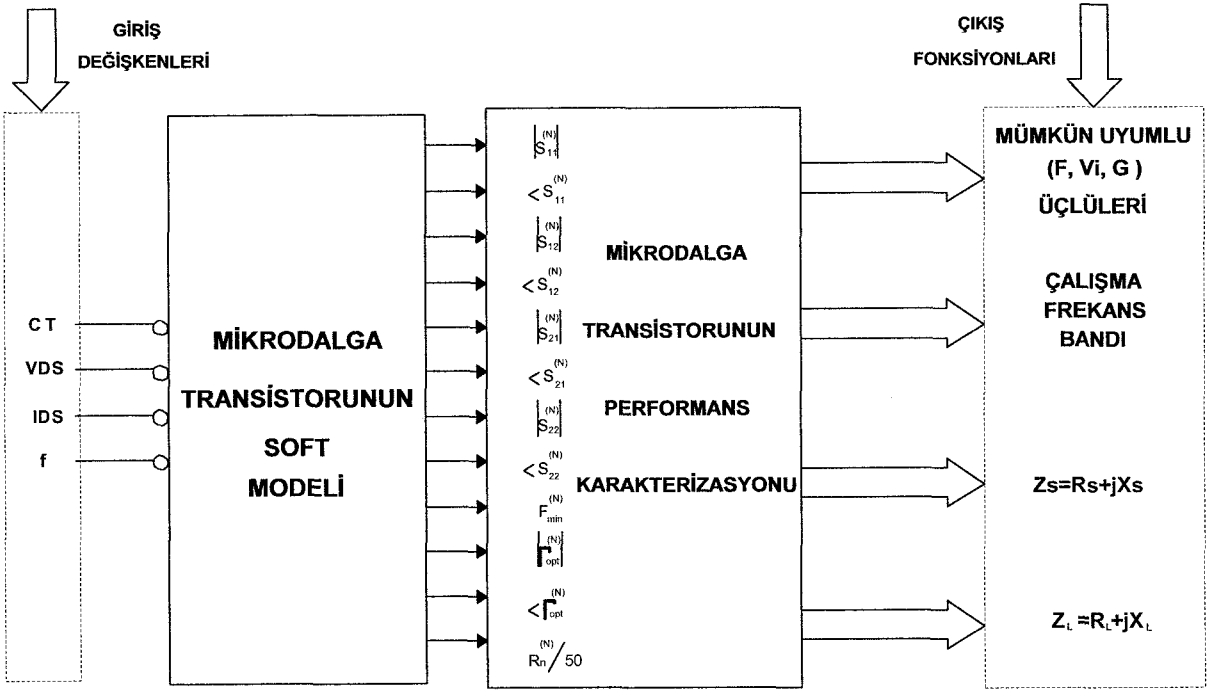


Şekil 6.1 Mikrodalga kuvvetlendiricisi tasarım uzayı “Kara-Kutu” modeli

Bu “Kara Kutu”,iki alt bloktan oluşturulabilir:

- i) Transistorun “Soft Modeli”,
- ii) “Performans Karakterizasyonu Teorisi”

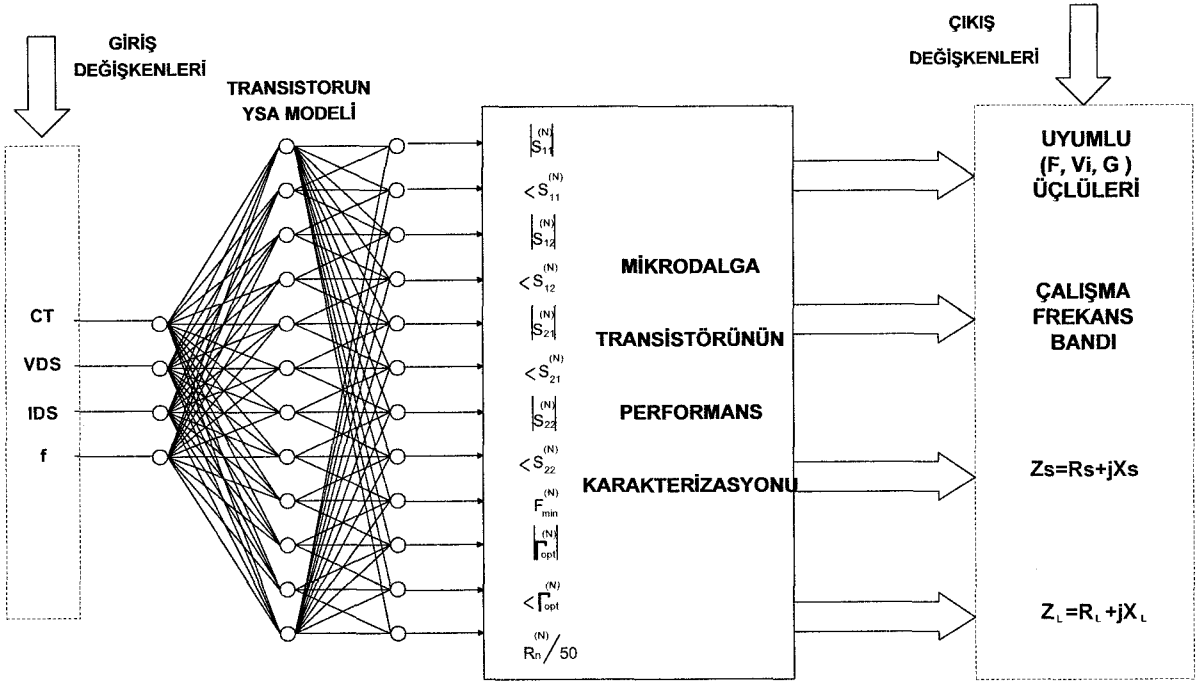
İlk blok, transistorun işaret ve gürültü davranışını çalışma uzayının her noktasında tanımlamak fonksiyonunu görür ve “Yapay Sinir Ağı” ve “Bulanık Mantık” gibi soft hesaplama yöntemleri ile gerçekleştirilmektedir.



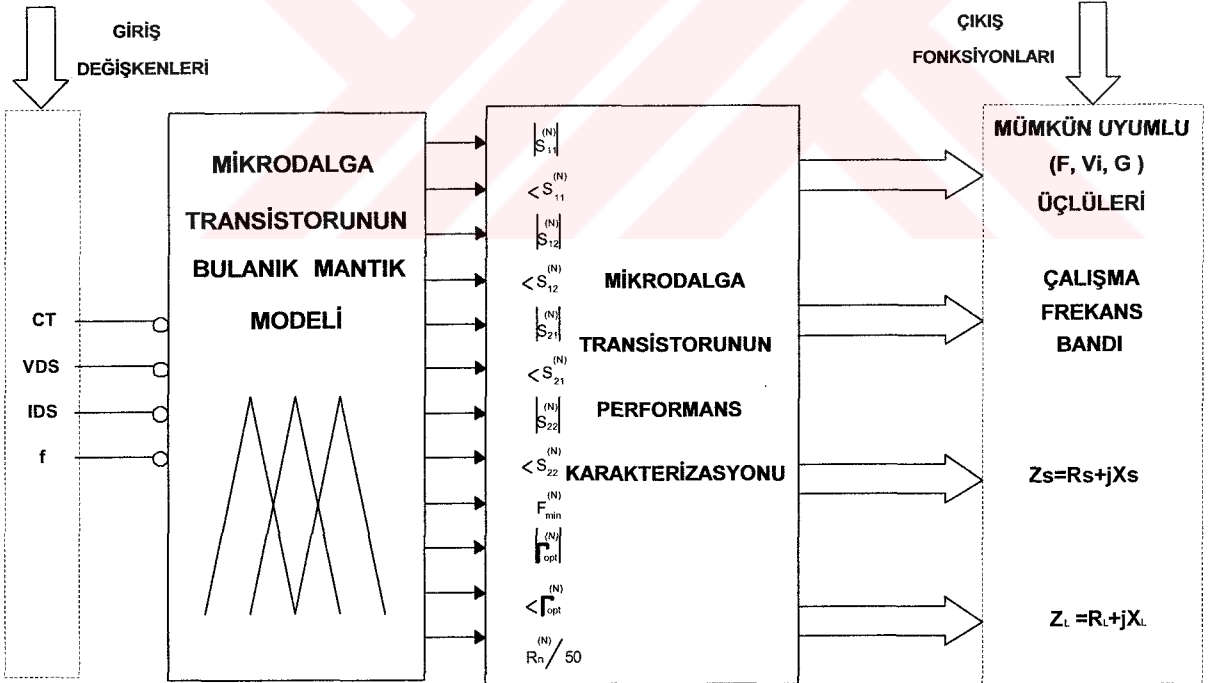
Şekil 6.2 Mikrodalga transistörü için “Performans Data Yaprakları”nın blok diyagramı (Cengiz,2004)

Referans olarak alınan Cengiz (2004) “Soft Hesaplama Yöntemine” göre iki tip “Performans Data Yaprakları” bloğu kullanılmıştır:

- Yapay Sinir Ağı ile Performans Data Yaprakları (Şekil 6.3);
- Bulanık Mantık ile Performans Data Yaprakları (Şekil 6.4);



Şekil 6.3 “Performans Data Yaprakları”nın “Yapay Sinir Ağı” blok diyagramı (Cengiz,2004)



Şekil 6.4 “Performans Data Yaprakları”nın bulanık mantık blok diyagramı (Cengiz,2004)

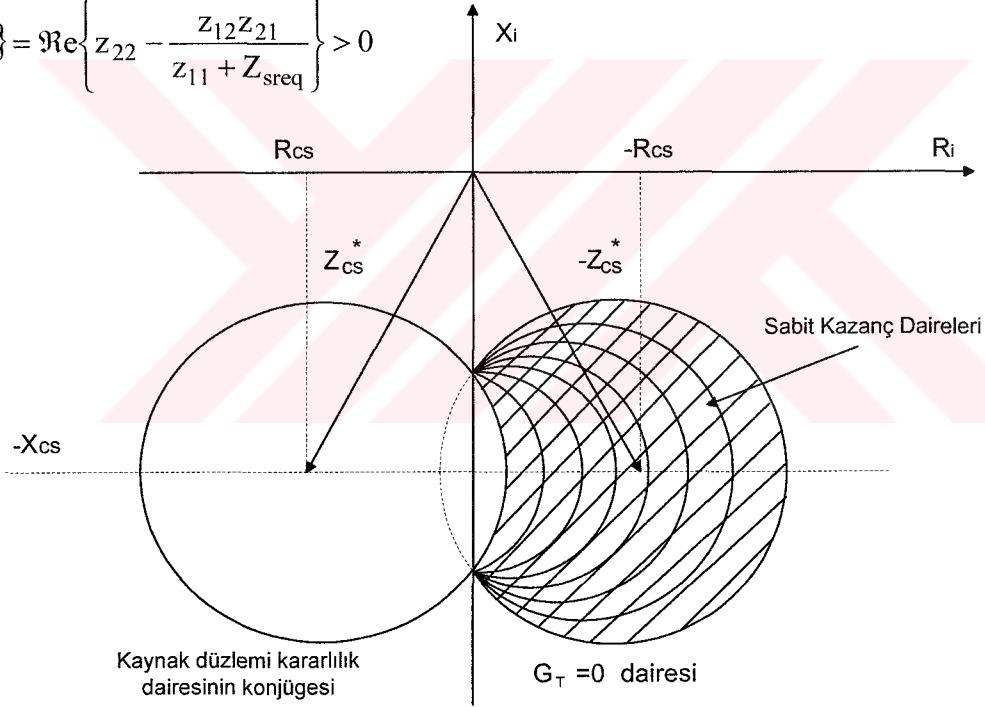
6.3 Transistor Performans Karakterizasyonu

5.56-5.58 denklemleri ile verilen transistor performans ölçü fonksiyonlarını eş zamanlı gerçekleyen (F, V_i, G_T) ve (Z_S, Z_L) fiziksel çözümleridir (Güneş, 1994, 1998, 2002). Fiziksel gerçekleştirilebilirlik koşulları aşağıda verilmiştir:

$$F_{req} \geq F_{min}, V_{ireq} \geq 1, G_{Tmin} \leq G_{Treq} \leq G_{Tmax} \quad (6.1)$$

Z_{Sreq} ve Z_{Lreq} sonlandırmaları transistörü “Kararlı Çalışma Bölgesi”ndeki pasif sonlandırmalardan seçilmelidir (Şekil 6.5):

$$\begin{aligned} \Re\{Z_{sreq}\} &\geq 0, \quad \Re\{Z_{Lreq}\} \geq 0 \\ \Re\{Z_{in}\} &= \Re\left\{Z_{11} - \frac{Z_{12}Z_{21}}{Z_{22} + Z_{Lreq}}\right\} > 0, \\ \Re\{Z_{out}\} &= \Re\left\{Z_{22} - \frac{Z_{12}Z_{21}}{Z_{11} + Z_{sreq}}\right\} > 0 \end{aligned} \quad (6.2)$$



Şekil 6.5 Giriş empedans düzleminde koşullu kararlı halde bir V_{ireq} ile sınırlandırılmış kazanç daireleri (Güneş, 1994)

Transistor performans karakterizasyonu kuvvetlendirici tasarımı için bir $B=f_1-f_2$ çalışma bandında, aşağıdaki uyumlu (F, V_i, G_T) üçlülerini ve (Z_S, Z_L) sonlandırmalarını çıktı olarak vermektedir:

$$(F_{\text{req}}, V_{i \text{ req}}, G_{T \text{ max}}) \Leftrightarrow Z_{S \text{ max}} = R_{S \text{ max}} + j X_{S \text{ max}}, \quad (6.3)$$

$$Z_{L \text{ max}} = R_{L \text{ max}} + j X_{L \text{ max}}; \quad (6.4)$$

$$(F_{\text{req}}, V_{i \text{ req}}, G_{T \text{ min}}) \Leftrightarrow Z_{S \text{ min}} = R_{S \text{ min}} + j X_{S \text{ min}}, \quad (6.5)$$

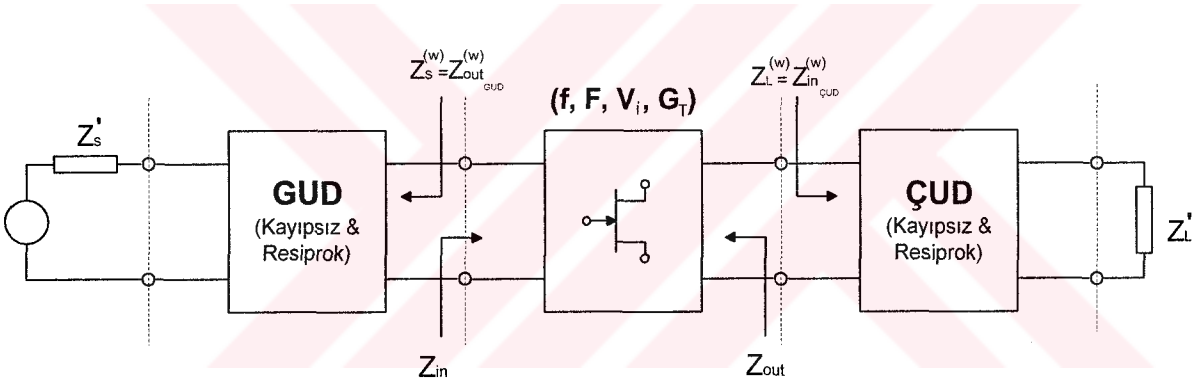
$$Z_{L \text{ min}} = R_{L \text{ min}} + j X_{L \text{ min}}; \quad (6.6)$$

$$(F_{\text{req}}, V_{i \text{ req}}, G_{T \text{ req}}) \Leftrightarrow Z_{S \text{ req}} = R_{S \text{ req}} + j X_{S \text{ req}}, \quad (6.7)$$

$$Z_{L \text{ req}} = R_{L \text{ req}} + j X_{L \text{ req}}. \quad (6.8)$$

6.4 Giriş Ve Çıkış Uydurma Devrelerinin Ayırık Ayırık Tasarlanması

Bir mikrodalga kuvvetlendiricisinin giriş ve çıkış uydurma devreleri ile birlikte genel görünümü Şekil 6.6’da gösterilmiştir. Transistorun “Performans Karakterasyonu” teorisi kullanılarak elde edilen uyumlu performans ($F_{\text{req}}, V_{i \text{ req}}, G_{T \text{ req}}$) üçlülerinin kuvvetlendirici içinde gerçekleştirilebilmesi için transistorun kaynak ($Z_{S \text{ req}}$) ve yük ($Z_{L \text{ req}}$) sonlandırılmalarının gerçekleştirilmesi gerekmektedir. Bu nedenlerden dolayı iki farklı devre tasarlanmıştır.



Şekil 6.6 Giriş ve çıkış uydurma devreli mikrodalga kuvvetlendiricisi

Kayıpsız ve resiprok iki kapılıların en önemli özellikleri şu şekilde tanımlanabilir;

Kayıpsız iki kapılıda net giriş gücü P_{in} , net çıkış gücüne P_{out} eşittir.

$$G_{op} = \frac{P_{out}}{P_{in}} = 1 \quad \Leftrightarrow \quad P_{in} = P_{out} \quad (6.9)$$

Giriş portundan elde edilebilecek güç, çıkış portundan elde edilebilecek güce eşittir.

$$P_{in \text{ AVS}} = P_{out \text{ AVS}} \quad (6.10)$$

Kayıpsız iki kapılı bir sistemin giriş ve çıkış portlarındaki Gerilim Duran dalga Oranı (VSWR) V_i sabittir.

6.4.1 Kaynak Empedansını Transistorun Giriş Empedansına Uygunlaştıran Devre

Giriş ve kaynak empedansı bilinen sistemin yansıtma katsayısı,

$$|\rho_{in}|^2 = \left| \frac{Z_{in} - Z_S^*}{Z_{in} + Z_S} \right|^2 = \frac{P_r}{P_{AVS}} \quad (6.11)$$

şeklinde hesaplanır. Burada;

Z_{in} = Giriş empedansı

Z_S = Kaynak empedansı

P_r = Yansıyan güç

P_{AVS} = Kaynaktan elde edilebilecek güçtür.

Herhangi bir iki kapılı için Güç Kazancı,

$$G_T = \frac{P_L}{P_{AVS}} = \frac{P_{in}}{P_{AVS}} \cdot \frac{P_L}{P_{in}} \quad (6.12)$$

$$G_{op} = \frac{P_L}{P_{in}} \quad (6.13)$$

$$G_T = \frac{P_{AVS} - P_r}{P_{AVS}} \cdot G_{op} \quad (6.14)$$

(6.11) ve (6.14)'den;

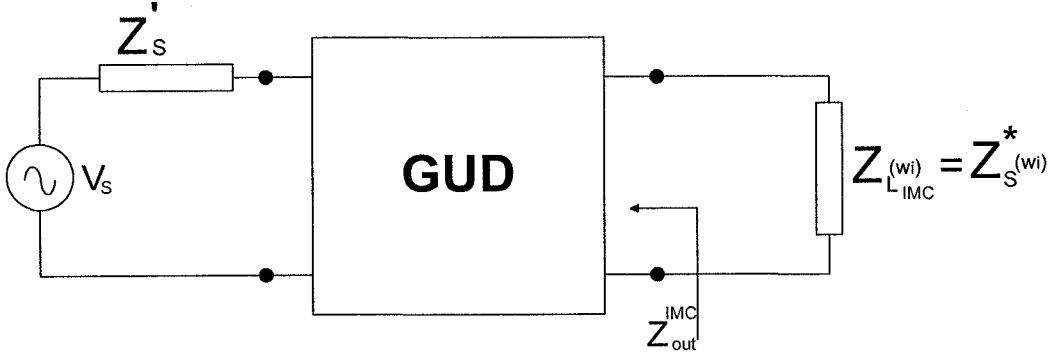
$$G_T = (1 - |\rho_{in}|^2) G_{op} \quad (6.15)$$

elde edilir. Kayıpsız ve resiprok bir iki kapılıda, $P_{in}=P_L$ olacağından,

$$G_T = 1 - |\rho_{in}|^2 \quad (6.16)$$

şeklinde olur.

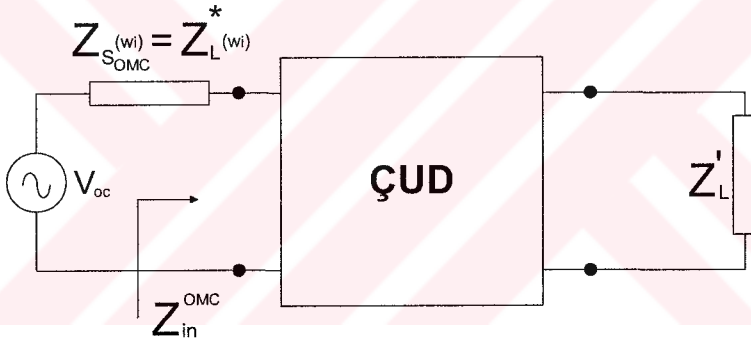
Eğer $\rho_{in}=0$ olursa $Z_{in}=Z_S^*$ ve dolayısıyla $G_T=1$ olur.



Şekil 6.7 Kaynak empedansını transistorun giriş empedansına uygunlaştıran devre

6.4.2 Yük Empedansını Transistorun Çıkış Empedansına Uygunlaştıran Devre

Aynen kaynak empedansını transistorun giriş empedansına uygunlaştıran devrede olduğu gibi transistorun çıkış empedansını, yük empedansına eşitleyen uydurma devresi de yukarıdaki bağıntılardan tasarlanabilir.



Şekil 6.8 Yük Empedansını Transistorun Çıkış Empedansına Uygunlaştıran Devre

Optimizasyon işleminde kullanılmak üzere giriş ve çıkış uydurma devreleri için hata ifadesi (6.17)'da tanımlanmıştır:

$$E = \sum_{i=1}^N 1 - e^{-W(1-G_T)^2} \quad (6.17)$$

W = Ağırlık katsayısı

7. MİKRODALGA KUVVETLENDİRİCİSİ TASARIM UZAYI VE TASARIM HEDEFLERİ

Optimizasyon, mühendislik problemlerinde kullanılan en temel uygulamalardan birisidir.

Genel olarak bir optimizasyon işlemi iki temel problem içerir.

- i) Tasarım değişkenleri ve hedefleri ile tanımlanan uygun tasarım uzayının belirlenmesi;
- ii) Uygun tasarım uzayında tasarım değişkenlerine bağlı hata fonksiyonunun global minimumunun bulunması.

7.1 Mikrodalga Kuvvetlendirici Tasarım Uzayı

Toplu ya da dağılmış parametrelili elemandan oluşan çok çeşitli konfigürasyonda giriş (GUD) ve çıkış (ÇUD) uydurma iki-kapılılardan oluşan kuvvetlendiriciler tasarlanabilir. Madem ki probleme sistem teorisi ile yaklaşılmaktadır, o halde metot her bir duruma kolaylıkla uygulanabilir.

Çalışmamızda uydurma devrelerinde iki tip eleman kullanılmaktadır:

- i) Birim Eleman (UE): Elektriksel uzunluğu $\theta = \beta \ell$ ve Z_0 karakteristik empedans ile transmisyon hat parçasıdır;
- ii) Reaktif Eleman: Kısa devre edilmiş transmisyon hat (θ, Z_0) parçası ile gerçekleştirilen kapasitif ya da endüktif eleman

bu iki tip eleman;

- a) Kaynak Empedansını Transistorun Giriş Empedansına Uyumlaştıran “T” ve “Π” tipi konfigürasyonlarda hat uzunlukları ve karakteristik empedansları optimizasyon değişkenleri olarak kullanılmıştır:

$$\mathbf{\theta} = [\ell_1 \ell_2 \ell_3 Z_{01} Z_{02} Z_{03}]^t$$

- b) Yük Empedansını transistorun çıkış empedansına uyumlaştıran “T” ve “Π” tipi

konfigürasyonlarda hat uzunlukları ve karakteristik empedansları optimizasyon değişkenleri olarak kullanılmıştır:

$$\mathbf{\Theta} = [\ell_4 \ell_5 \ell_6 Z_{04} Z_{05} Z_{06}]^t$$

- c) Kaynak Empedansını Transistorun Giriş Empedansına Uyumlaştıran “L” tipi konfigürasyonda hat uzunlukları ve karakteristik empedansları optimizasyon değişkenleri olarak kullanılmıştır:

$$\mathbf{\Theta} = [\ell_1 \ell_2 Z_{01} Z_{02}]^t$$

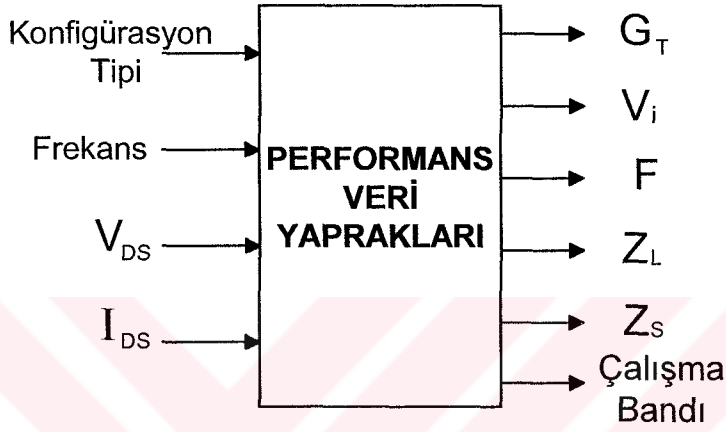
- d) Yük Empedansını transistorun çıkış empedansına uyumlaştıran “L” tipi konfigürasyonda hat uzunlukları ve karakteristik empedansları optimizasyon değişkenleri olarak kullanılmıştır:

$$\mathbf{\Theta} = [\ell_3 \ell_4 Z_{03} Z_{04}]^t$$

Genetik optimizasyon sürecinde hat uzunlukları ve karakteristik empedanslar program içinde iki ayrı gruba ayrılmış, her iki gruba ayrı sınır değerleri verilerek sonuç elde edilmiştir.

7.2 Mikrodalga Kuvvetlendirici Tasarım Hedefleri

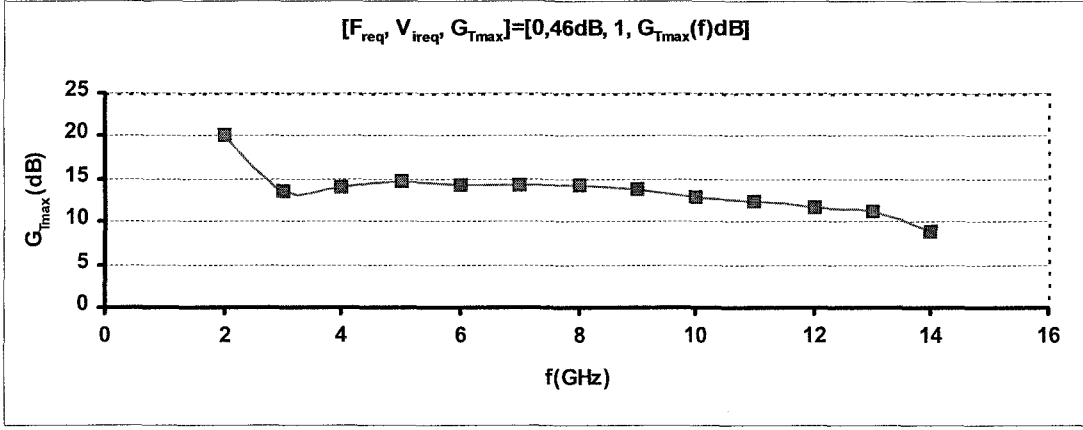
Bir mikrodalga kuvvetlendirici optimizasyonunda tasarım değişkenleri, genellikle alt ve üst sınırları kullanılan teknoloji ile belirlenen, uydurma devre parametreleridir. Bununla birlikte tasarım hedefleri kuvvetlendirici optimizasyonunun en önemli problemlerinden birisidir. Genellikle optimizasyon çalışmaları frekans bandı boyunca Gürültü (F) ve giriş duran dalga oranı (V_i) gibi performans kriterlerini göz önüne almadan sadece transduser güç kazancı (G_T) üzerinde yoğunlaşmıştır.



Şekil 7.1 Mikrodalga kuvvetlendiricisi tasarım uzayı kara kutu modeli

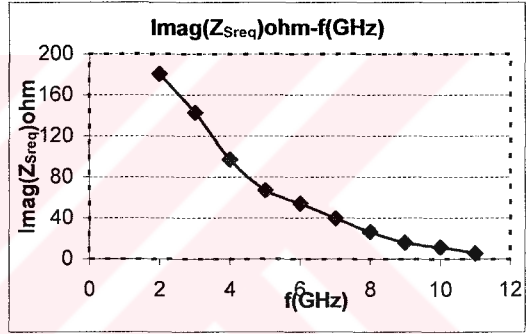
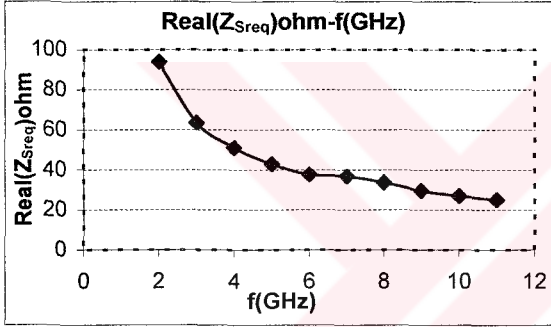
Optimizasyon prosesinin amacı $V_{ireq}(f)$ ve $F_{req}(f)$ sınırlayıcı fonksiyonlara uygun bir $G_{Tmin} < G_{Treq} < G_{Tmax}$ seviyeyi potansiyel bant genişliği boyunca ‘düz’ bir biçimde gerçekleştirmektedir. Buna göre minimize edilecek “Hata” (=Objektif) fonksiyonu, sadece Transduser kazanç G_T , giriş VSWR V_i ve gürültü faktörü F içermektedir ve bu ölçü fonksiyonları optimizasyon değişkenleri cinsinden analitik ifade edilmiştir.

NE32901 transistörü, $I_C=5mA$, $V_c=10V$ kutuplama koşullarında, uyumlu $F_{req}(\omega_i)=0.46dB$, $V_{ireq}(\omega_i)=1.0$, $G_{Treq}(\omega_i)=12dB$ üçlüsü için giriş ve çıkış uydurma devreleri ayrı ayrı tasarlanmıştır. NE32901 transistor karakteristiğine göre uyumlu $F_{req}(\omega_i)=0.46 dB$, $V_{ireq}(\omega_i)=1.0$, $G_{Treq}=12dB$ üçlüsü bant genişliği 2GHz ile 11GHz arasındadır. Transistörün bu performans potansiyeli, genetik data işlemesi kullanılan bir optimizasyon ile gerçekleştirilmiştir.

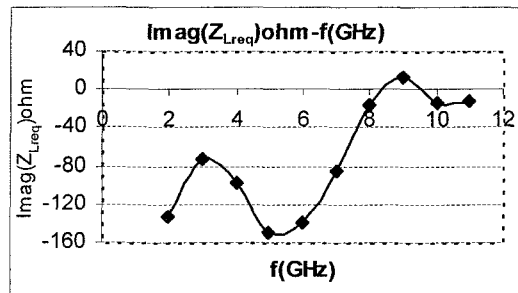
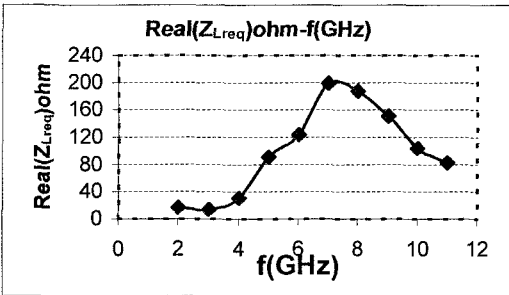


Şekil 7.2 NE329S01 Transistorunun $F=0,46\text{dB}$, $V_i=1$ $G_{Tmax}(f)$ Değişimi

(Koşullar: $V_{CE}=2\text{ V}$; $I_C=10\text{ mA}$ - Çalışma Band Genişliği : 2 - 18 GHz)



a) Kaynak Sonlandırmaları



b) Yük Sonlandırmaları

Şekil 7.3 NE329S01 Transistoru için (0,46dB ,1, 12dB) Üçlülerinin sonlandırma Fonksiyonları

8. MİKRODALGA KUVVETLENDİRİCİ OPTİMİZASYONU

8.1 Temel Mühendislik Optimizasyon Problemleri

Optimizasyon problemi, genel anlamda bir fonksiyonun en küçük değerini belirleme işlemi şeklinde değerlendirilebilir. Bu fonksiyon, bir hata fonksiyonu olarak değerlendirildiğinde tanımlanan hatayı en aza indirmeye işlemi gerçekleştirilecektir. Bu işlem tanımlanan hata fonksiyonunun bağlı olduğu parametre değerlerinin değiştirilmesi ile gerçekleştirilebilir. Bu değerlerin değişimi seçilen optimizasyon yöntemine göre farklılıklar gösterir. Optimizasyon yönteminin seçiminde hata fonksiyonunun tanımı önemlidir. Bu fonksiyona bağlı olarak belirlenecek minimum hata değeri optimizasyon yöntemine göre yerel veya genel minimum olabilmektedir (Kargın, 2000).

8.1.1 Tasarım Uzayı Ve Mikrodalga Kuvvetlendiricisine Uygulaması: Çok Hedefli Amaç Fonksiyonları

8.1.1.1 (F, V_i, G_T), B Hedef Uzayı Ve Kullanımı

Toplu ya da dağılmış parametrelili elemandan oluşan çok çeşitli konfigürasyonda giriş (GUD) ve çıkış (ÇUD) uydurma iki-kapılılardan oluşan kuvvetlendiriciler tasarlanabilir.

Uydurma devrelerinde daha önceden de belirttiğimiz gibi birim ve reaktif olmak üzere iki tip eleman kullanılmaktadır. Bu iki tip eleman;

- a) T, Π konfigürasyonunda hat uzunlukları ve karakteristik empedansları optimizasyon değişkenleri olarak seçilir:

$$\mathbf{\Theta} = [\ell_1 \ell_2 \ell_3 \ell_4 \ell_5 \ell_6 Z_{01} Z_{02} Z_{03} Z_{04} Z_{05} Z_{06}]^t$$

- b) İkinci bir yaklaşım olarak sabit hat uzunluğu ve sabit hat uzunluğuna bağlı olarak değişken karakteristik empedansları optimizasyon değişkeni olarak kullanılır.

$$\mathbf{\Theta} = [\ell Z_{01} Z_{02} Z_{03} Z_{04} Z_{05} Z_{06}]^t$$

- c) Giriş uydurma devresi T veya Π , çıkış uydurma devresi seri ile kısa devre edilmiş transmisyon hattından oluşan L tipi “Hibrid” uydurma devresi konfigürasyonunda, hat uzunlukları ve karakteristik empedansları optimizasyon değişkenleri olarak seçilir:

$$\mathbf{\theta} = [\ell_1 \ell_2 \ell_3 \ell_4 \ell_5 Z_{01} Z_{02} Z_{03} Z_{04} Z_{05}]^t$$

d) Giriş ve çıkış L tipi uydurma devreleri konfigürasyonunda hat uzunlukları ve karakteristik empedansları optimizasyon değişkenleri olarak kullanılır:

$$\mathbf{\theta} = [\ell_1 \ell_2 \ell_3 \ell_4 Z_{01} Z_{02} Z_{03} Z_{04}]^t$$

Çalışmamızda bu hedef uzayı kullanılmamıştır.

8.1.1.2 (Z_S, Z_L) Ve B Hedef Uzayı Ve Kullanımı

Transistor ün giriş empedansını, kaynak empedansına ve transistor ün çıkış empedansını yük empedansına uygunlaştıran uydurma devreleri için tasarım uzayı aşağıdaki şekilde belirlenmiştir:

a) Kaynak Empedansını Transistorun Giriş Empedansına Uygunlaştıran “T” ve “II” tipi konfigürasyonlarda hat uzunlukları ve karakteristik empedansları optimizasyon değişkenleri olarak kullanılmıştır:

$$\mathbf{\theta} = [\ell_1 \ell_2 \ell_3 Z_{01} Z_{02} Z_{03}]^t$$

b) Yük Empedansını transistorun çıkış empedansına uygunlaştıran “T” ve “II” tipi konfigürasyonlarda hat uzunlukları ve karakteristik empedansları optimizasyon değişkenleri olarak kullanılmıştır:

$$\mathbf{\theta} = [\ell_4 \ell_5 \ell_6 Z_{04} Z_{05} Z_{06}]^t$$

c) Kaynak Empedansını Transistorun Giriş Empedansına Uygunlaştıran “L” tipi konfigürasyonda hat uzunlukları ve karakteristik empedansları optimizasyon değişkenleri olarak kullanılmıştır:

$$\mathbf{\theta} = [\ell_1 \ell_2 Z_{01} Z_{02}]^t$$

d) Yük Empedansını transistorun çıkış empedansına uygunlaştıran “L” tipi konfigürasyonda hat uzunlukları ve karakteristik empedansları optimizasyon değişkenleri olarak kullanılmıştır:

$$\mathbf{\theta} = [\ell_3 \ \ell_4 \ Z_{o3} \ Z_{o4}]^t$$

Çalışmada iki farklı hedef uzayı üzerine çalışmak yerine (Z_S , Z_L) ve B hedef uzayı üzerine yoğunlaşmış ve hesaplamalar bu yönde yapılmıştır.



8.1.2 Global Optimum Ve Algoritmaları

Geleneksel en iyi (optimal noktayı) arama yöntemleri, iki ana başlık altında toplanabilir: Bunlar, gradient-temelli algoritmalar ve sezgisel yöntemlerdir. Gradient temelli algoritmalar, bugüne kadar literatürde yoğun bir biçimde incelenmişlerdir. Bunlar da kendi aralarında dolaylı ve doğrudan olmak üzere ikiye ayrılırlar.

8.1.2.1 Gradient-Temelli Algoritmalar

Matematiksel nonlinear programlama algoritmaları, mühendislik optimizasyon problemlerindeki uygulamalara bir çözüm olarak ortaya çıkmışlardır. Bu algoritmalar, eşit ve eşit olmayan sınırlamaların bir karışımı olan tek ve çok hedefli tasarım problemlerinin çözümü için genel bir yaklaşım sürerler. Bu algoritmaların en etkilileri, hedef fonksiyonunun en az birinci dereceden türevine ihtiyaç duyan gradient tabanlı olanlarıdır. Gradient tabanlı algoritmaların tepe tırmanma kabiliyeti, ilk tahminin optimum noktaya olan uzaklığı ile ilişkilidir. Tasarım uzayının iç bükey olduğu durumlarda global optimum noktayı garanti edemezler. Ayrıca bu metodlar tasarım uzayının süreksiz olduğu problemlerde de yetersiz kalmaktadırlar.

Dolaylı Metod

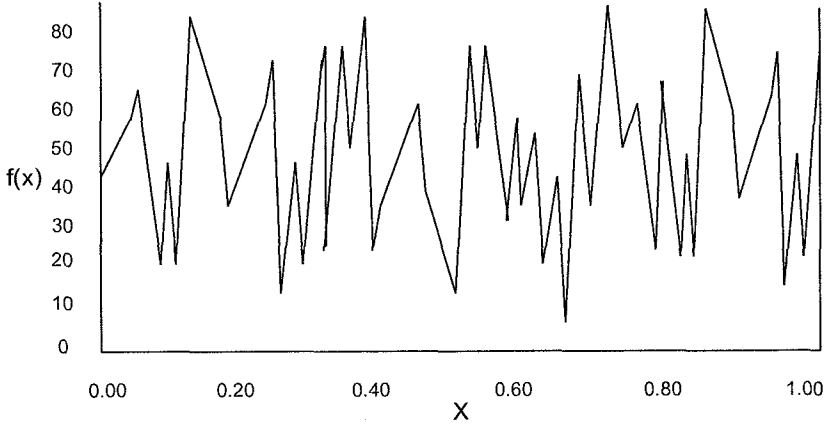
Dolaylı metoda kullanılan matematiksel ifadeler, genel olarak aşağıdaki formattadır:

$$\nabla F(x) = 0 \quad (8.3)$$

hedef fonksiyonunun gradyanının sıfıra eşit alınması sonucu ortaya çıkan doğrusal olmayan denklem takımının çözülmesi gerekir. Arama tepe veya çukur olması muhtemel noktalar civarında yapılır.

Bu yöntemin eksiklikleri şunlardır:

- 1) Doğrusal olmayan denklem takımlarının çözümü oldukça zordur. Başlangıç ve sınır şartları içerisinde çözüme ulaşılabilir.
- 2) Hedef fonksiyonları, türevi alınabilir olmalıdır. Teorik olarak hesabı kolay hedef fonksiyonlarının bulunabilmesine karşın, gerçek hayatta karşılaşılan problemler teoriden uzaktır. Parça parça süreksizlikler içerebileceği gibi gürültülü ortamlardan elde edildikleri için Şekil 8.1'deki gibi pek çok tepe noktalı olabilir.



Şekil 8.1 Pratikte karşılaşılan bir hedef fonksiyonu örneği (Erol, 1995)

- 3) Bu yöntemin en önemli eksikliği, yerel bakış açısıyla yürütülmesidir. Yöntem en iyi noktayı hareket noktasının yakınında arar. Örneğin Şekil 8.2.a'da gösterilen, uzayın sınırlı bir bölgesi olsun. Bu uzayın daha geniş bir bölgesi de, Şekil 8.2.b'de verilmiş olsun. Bu ikinci şekilden de görülebileceği üzere, küçük tepenin yakınında aramaya başlanılırsa, en iyi nokta olarak küçük tepenin üzerinde kalınacaktır. Halbuki, bulunacak bu noktadan daha iyi bir nokta vardır ki o da, daha yüksek olan ikinci tepe noktasıdır. Küçük tepenin yakınındaki bir noktada başlamak bile esas en iyi noktasını kaybettirecektir. Buna bir çözüm olarak, arama işlemine rasgele seçilmiş başka bir noktadan tekrar başlanılabilir. Şekil 8.2.a'daki gibi tek tepe noktalı fonksiyonlar, hesaba dayalı arama yöntemleri için kolay olmasına karşın Şekil 8.2.b'deki gibi fonksiyonlarda hangi tepeye tırmanılacağı ayrı bir problemidir.

Doğrudan Metod

Diğer bir yöntem ise alınan bir noktadan sadece yukarı ilerleyerek en iyi sonucu bulmayı hedefler. *Tepe Tırmanma* (Hill Climbing) adı verilen bu yöntem, fonksiyon grafiğinin tepelerine tırmanır. Bu arama şeklinde, fonksiyonun maksimum noktasının yeri hakkında bir tahminde bulunulur ve buna yakın herhangi bir noktadan arama sürdürülür. Burada yerel en iyi noktayı bulmak için mümkün olan en dik eğimli yönde ilerlenir.

Bu yöntemin eksiklikleri şunlardır:

- 1) Yöntem yerel bakış açısıyla ilerler. Çok sayıda dönme noktası içeren bir fonksiyonda çok sayıda tepe oluşur. Hangi tepenin en iyi çözüm olduğu bilinemez.
- 2) Hedef fonksiyonları türevlenebilir olmalıdır. Aksi taktirde arama yapamaz hale gelir.

8.1.2.2 Sezgisel Temelli Algoritmalar

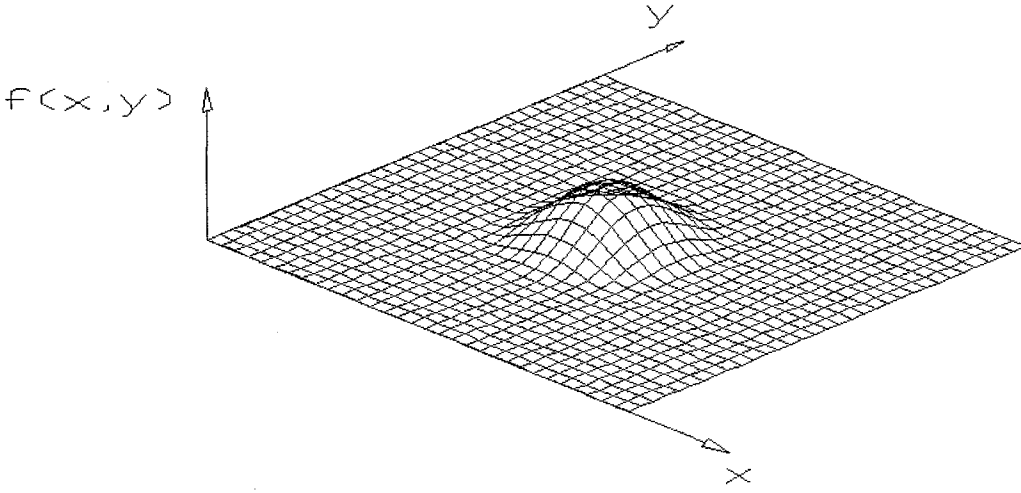
Bu algoritmalar geleneksel arama yöntemlerinin yetersiz kaldıkları yerlerde rahatlıkla kullanılabilir. Rastlantısal bir yönde tepe tırmanma veya rastgele noktalar seçerek bunların hedef fonksiyonlarını bulma gibi düşünceler, rastgele arama yöntemlerinin dışında tutulmalıdır. Sezgisel temelli algoritmalar, optimizasyon dizayn problemleri için en başarılı ve en basit arama metodlarıdır. Bu metodlar, her türlü tasarım uzayında, tasarım değişkenlerinde herhangi bir kısıtlamaya gitmeden kullanılabilirler.

Basit sayım teknikleri, rastgele yürüme metodları ile ilerlerler. Bu metodların en büyük dezavantajı, en basit problem için bile optimum noktaya ulaşmak için binlerce iterasyon yapmak zorunda olmalarıdır. Bundan dolayı optimal tasarım çözümleri için, sayım tekniklerinden daha az hesap gerektiren ve gradient tabanlı nonlinear programlama algoritmaları gibi bölgesel minimum noktaya takılmayan alternatif arama metodlarına ihtiyaç duyulmaktadır. Genetik algoritma bu konudaki hemen hemen tüm ihtiyaçları karşılayabilmektedir.

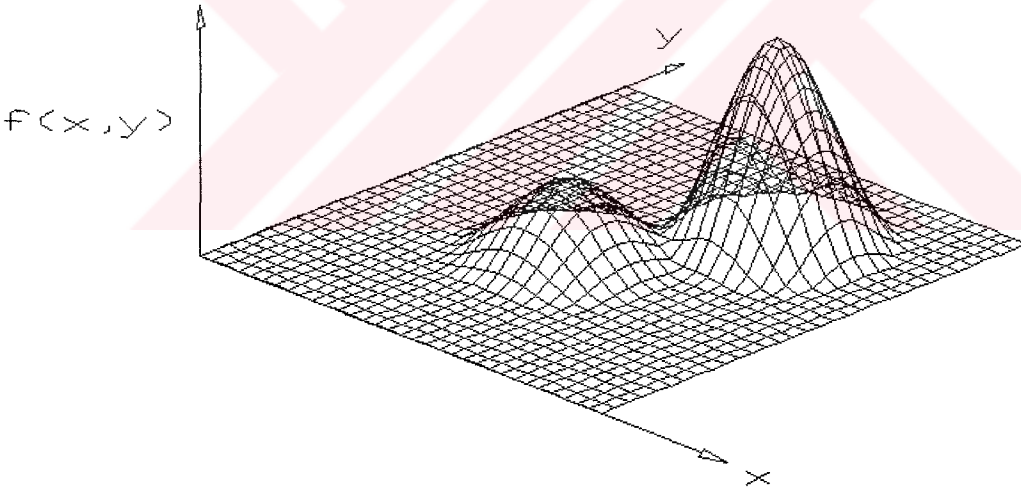
8.1.2.3 Genetik Algoritma İle Diğer Geleneksel Yöntemler Arasındaki Farklar

Genetik Algoritma ile diğer geleneksel yöntemler arasındaki farklar şu şekilde sıralanabilir:

- 1) Genetik algortmada, parametreler kodlanarak kullanılırlar. Genellikle kodlama biçimi ikilik sayı sistemindedir.
- 2) Genetik algortmada tek bir nokta yerine noktalar topluluğundan hareket edilir. Bu da Şekil 8.2b' deki geleneksel maksimuma ulaşmayı kolaylaştırır. Sistem yerel maksimuma takılmaz.
- 3) Genetik algortmada sadece hedef fonksiyonunun değeri kullanılır. Türev ve integral gibi daha farklı bilgilere gerek yoktur. Bu da yöntemi belirli kabul ve şartlardan uzak tutar.
- 4) Genetik algoritma belirlilik değil, olasılık kurallarına dayanır. Seçim işlemi bu olasılık doğrultusunda gerçekleştirilir.



(a)



(b)

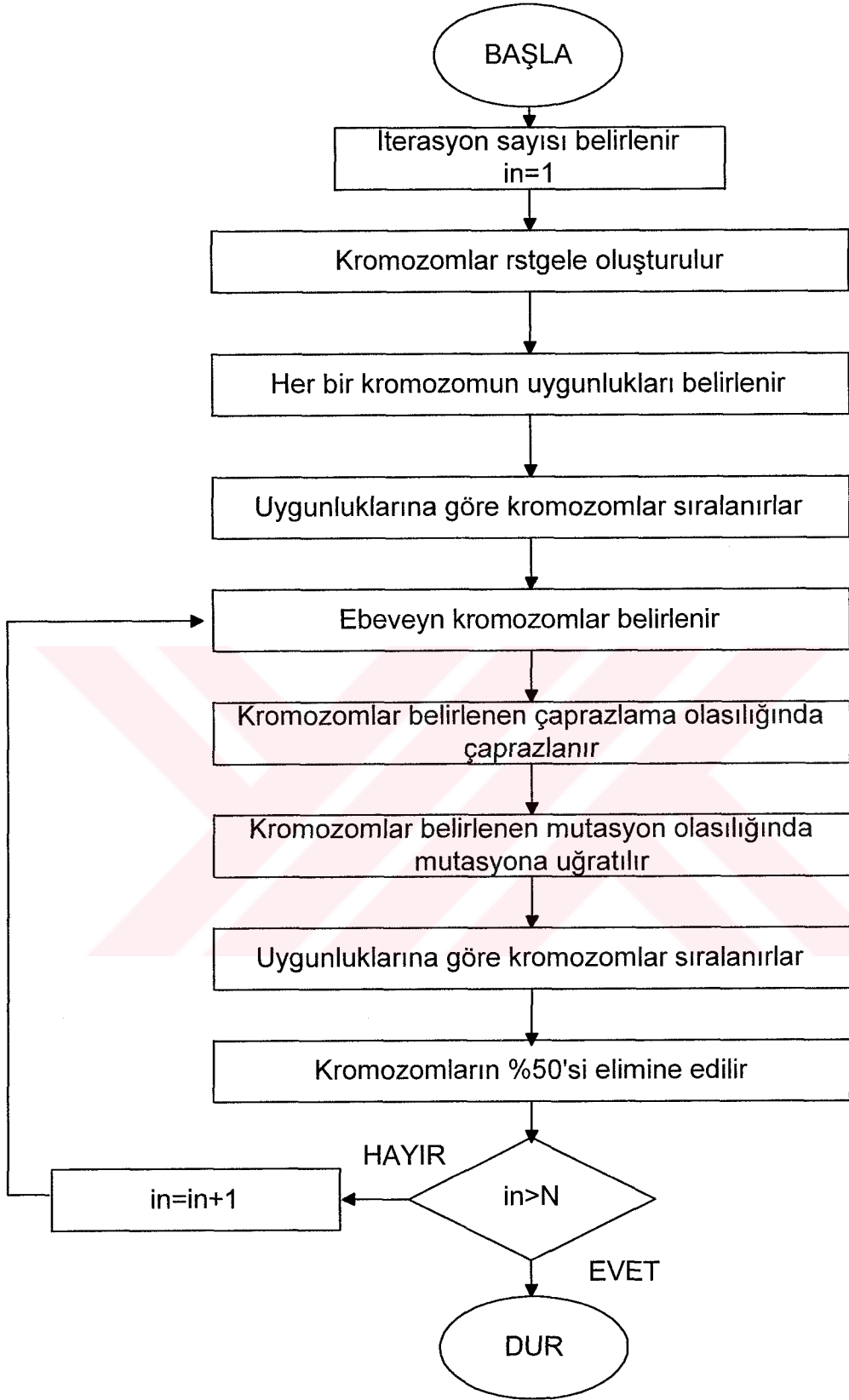
Şekil 8.2 Amaç Fonksiyonları (Erol, 1995)

Rastgeleliđi kullanmak, belirli bir mantıktan yoksun arama yapmak demek deđildir. Olasılıđı kullanmayan global arama yöntemleri de vardır. Ancak bunlarda da işlem sayısının fazlalığı çok boyutlu uzaylarda çözümü bulmayı zorlaştırmaktadır.

Tek bir nokta ele alıp, bu noktadan diđer bir noktaya belli bir kural kullanarak geçmek yanlış bir yerel en iyi noktaya yerleşmeye sebep olabilir. Genetik algoritma, tek bir nokta yerine, bir noktalar topluluğundan yararlandığı için, yapılacak ikinci adım bu noktalar topluluğunu belirlemektir. Böylece global en iyi noktaya yakınsama şansı artacaktır. Genetik algoritmada nesil adı verilen bu noktalar topluluğundan yeni bir nesil, yeni bir noktalar topluluđu elde edilir.

Genetik algoritma defalarca çalıştırılarak çok sayıda toplum oluşturulup hesaplanır. Toplumların hesaplanması sırasında en iyi bireyler saklandığı için o ana kadar bulunmuş en iyi çözüm çözümdür. Genetik algoritmanın yaptığı işlerin temeli akış diyagramı olarak Şekil 8.3’de verilmiştir.





Şekil 8.3 Genetik algoritmanın akış diyagramı (Cengiz,2004)

9. PROGRAM SONUÇLARI

Mikrodalga kuvvetlendiricisi optimizasyonu çalışmalarda NE329S01 mikrodalga transistörü üzerinde çalışılmıştır. Daha önce yapıldığı gibi tekrar bir modelleme işlemine gidilmek yerine yapılan optimizasyon işleminin hızlandırılmasına ve optimizasyon değişkeni olarak kabul ettiğimiz hat uzunluk ve karakteristik empedanslarının daha makul sınırlara çekilmesi üzerine yoğunlaşmıştır. Bunu sağlamak için hat uzunluk ve kalınlıkları program içerisinde iki farklı grup haline getirilip kullanılan değer aralıkları daha kontrol edilebilir bir yapıya sokulmuştur. Daha açık bir tanımla tek bir minimum değeri, iki farklı maksimum değeri bulunan 6 değişken yerine iki farklı minimum ve maksimum değerine sahip 3 değişkenli 2 grup oluşturulup mutasyon, cross-over gibi genetik işlemler bu grupların üzerine uygulanmıştır. Bun uygulamanın en önemli faydası gerçekleştirilebilirlikten uzak olan bir direnç aralığını optimizasyon işleminden çıkararak, programın daha dar bir aralıkta çalışması ve çok daha çabuk sonuca varabilmesini sağlamak olmuştur.

Hat uzunluk ve kalınlık değerleri Cengiz (2004) ile karşılaştırıldığında bu değerlerin mikroserit mimarisine çok daha uygun hale getirildiği gözükmektedir. Ayrıca daha önceki çalışmalarda elde edilen sonuçlar için 20-25 bin civarına iterasyon yürütülmesi gerekirken bu çalışmada maksimum 1500 iterasyonla bile istenene yakın sonuçlar elde edilebilmiştir.

Elde edilen genetik optimizasyon sonuçlarına MATLAB “fmincon” komutu ile 11 GHz’den 2 GHz’e kadar genişleyen bir bant üzerine adım adım tekrar optimizasyon prosesi uygulanmış ve sonuçlarda büyük bir değişikliğin olmadığı ve genetik optimizasyon işleminin son halinin Mikrodalga kuvvetlendirici optimizasyon işlemi için gayet yeterli olduğu sonucuna varılmıştır.

9.1 Genetik Algoritma İle Gerçekleştirilen Kuvvetlendiriciler

9.1.1 NE329S01 Transistoru İle Gerçekleştirilen Kuvvetlendirici Sonuçları

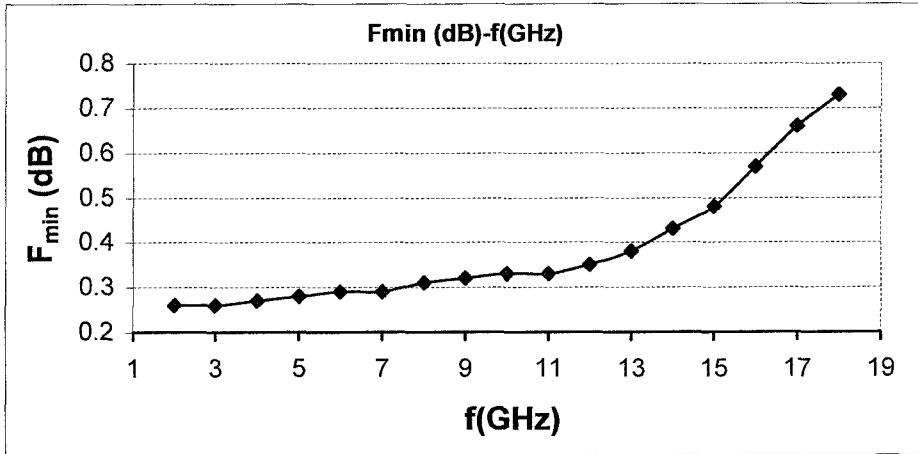
9.1.1.1 Kuvvetlendirici Tasarım Uzayı

En genel halde hat uzunluğu ℓ ve karakteristik empedansından Z_0 oluşan 12 değişkenli bir tasarım değişken uzayı ile $0.1\text{cm} < \ell < 15\text{cm}$, $20\Omega < Z_0 < 150\Omega$ sınırlamaları içinde çalışılmıştır. Bu sınırlamalar optimizasyon işlemi adım adım gözlenerek bazı devreler için daha da daraltılarak çok daha az iterasyonla daha iyi sonuçlar elde edilmesi sağlanmıştır. NE329S01 transistorunun $I_c=10\text{ mA}$, $V_{CE}=2\text{ V}$ kutuplaması için $F_{\min}(f)$ -f (Şekil 9.1) $F_{\text{req}}=0.46\text{dB}$ seçilmiş ve $V_i=1$ sınırlamaları içinde maksimum kazanç $G_{T\max}$ frekans değişimi Şekil 9.2’da verilmiştir. Buna göre “Tasarım Hedef Uzayı” olarak Şekil 9.2’deki $G_{T\max}$ -f değişim eğrisinden $F_{\text{req}}=0.46\text{dB}$, $V_{i\text{req}}=1$, $G_{T\text{req}}=12\text{dB}$ band genişliği $B=2\text{-}11\text{GHz}$ belirlenmiştir. Ayrıca bu performans hedeflerini gerçekleyen kaynak Z_S ve yük Z_L sonlandırmaları da Şekil 9.3’de verilmiştir.

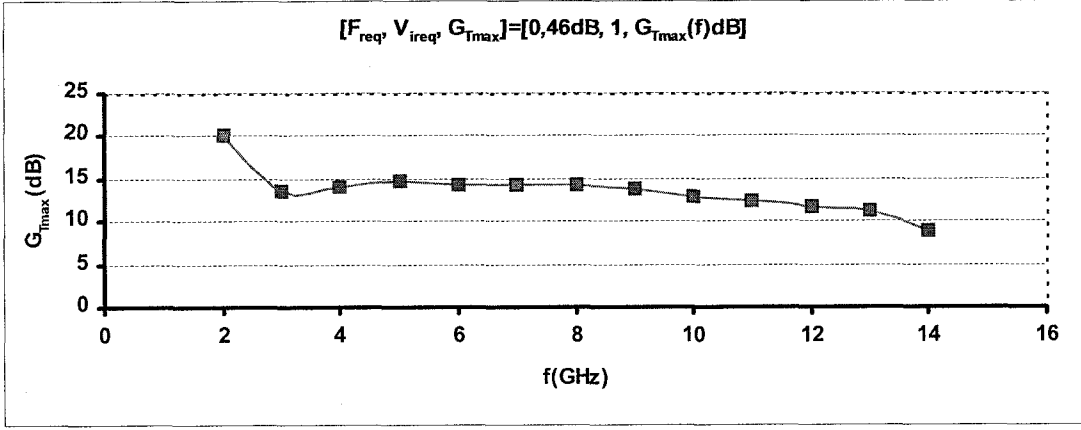
Transistorun Adı: NE329S01

Kutuplama Koşulu: $V_{CE}=2\text{V}$, $I_C=10\text{mA}$

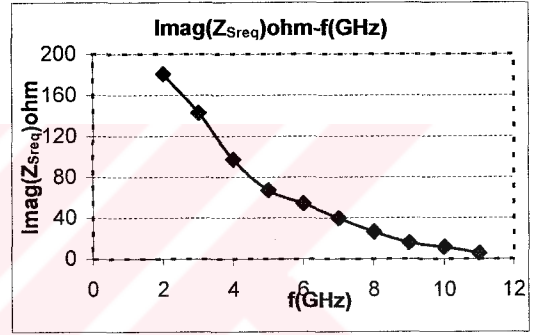
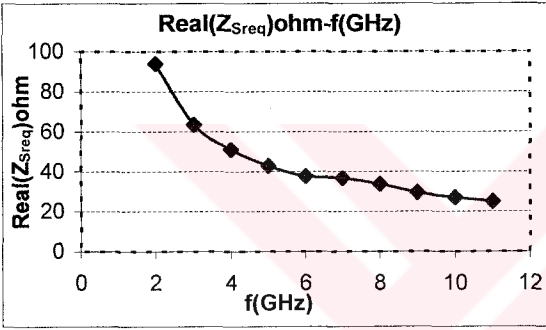
Çalışma Bandı: 2-18 GHz



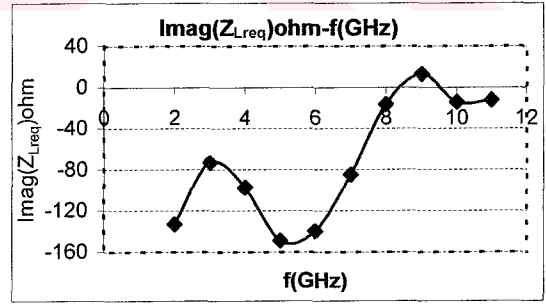
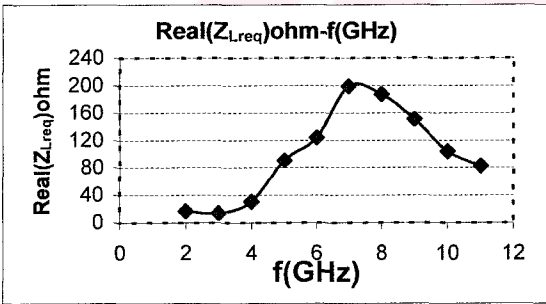
Şekil 9.1 NE329S01 Transistorunun $I_c=10\text{ mA}$, $V_{CE}=2\text{ V}$ kutuplaması için $F_{\min}(f)$ Değişimi (Güneş, 2002)



Şekil 9.2 NE329S01 Transistorunun $F=0,46\text{dB}$, $V_i=1$ $G_{Tmax}(f)$ Değişimi (Güneş, 2002)



a) Kaynak Sonlandırmaları

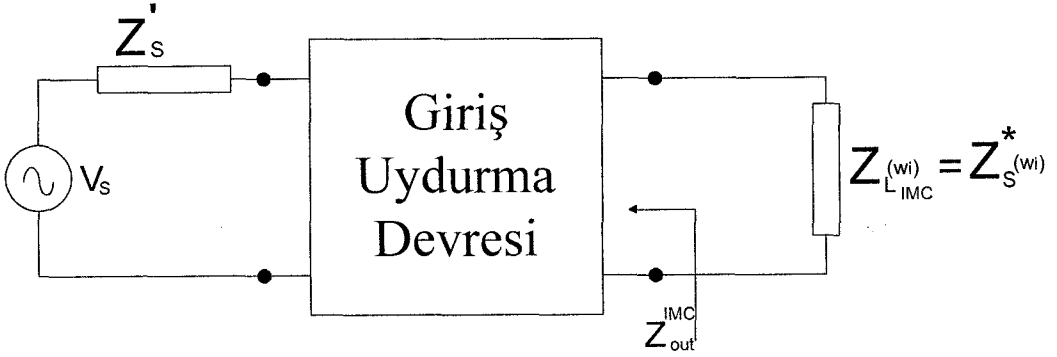


b) Yük Sonlandırmaları

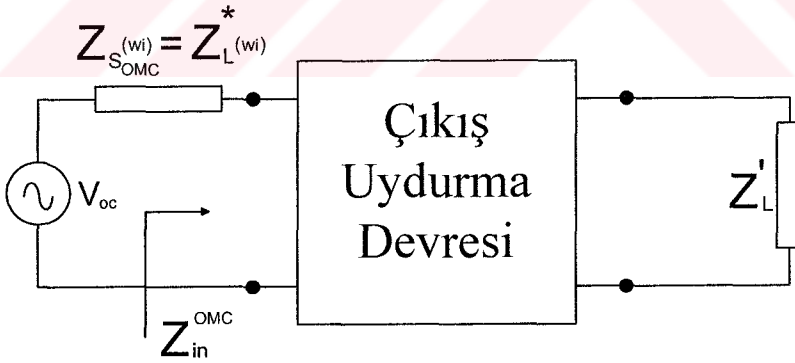
Şekil 9.3 NE329S01 Transistoru için (0,46dB ,1, 12dB) uyumlu üçlülerinin sonlandırma fonksiyonları (Güneş, 2002)

9.1.1.2 Giriş Ve Çıkış Uydurma Devreleri

Optimizasyon işlemi sırasında 2-11 GHz aralığı için genetik optimizasyon metodu kullanılarak 4 adet 3 elemanlı ve 4 adet 2 elemanlı olmak üzere toplam 8 adet uydurma devresi oluşturuldu.



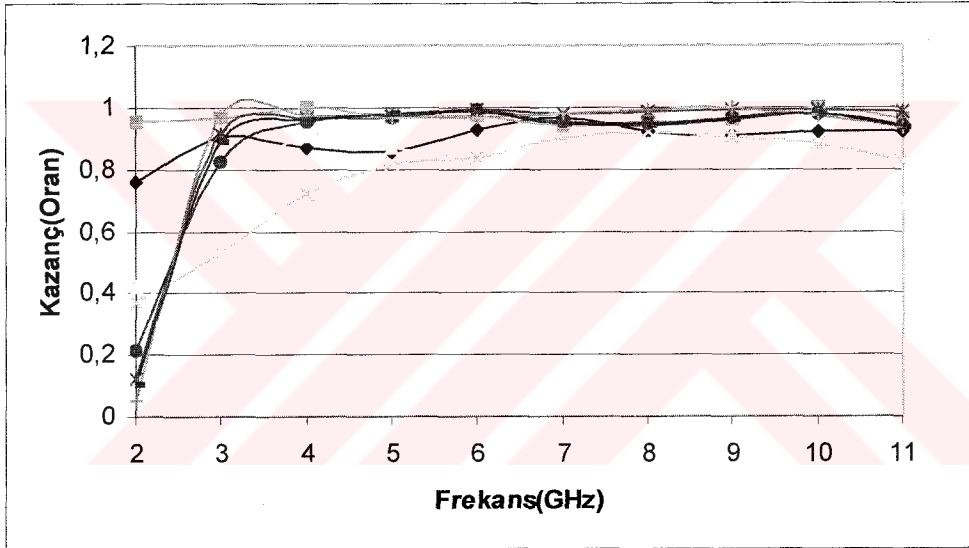
Şekil 9.4 Kaynak empedansını transistorun giriş empedansına uygunlaştıran devre ($Z'_s = 50\Omega$)



Şekil 9.5 Yük Empedansını Transistorun Çıkış Empedansına Uygunlaştıran Devre ($Z'_L = 50\Omega$)

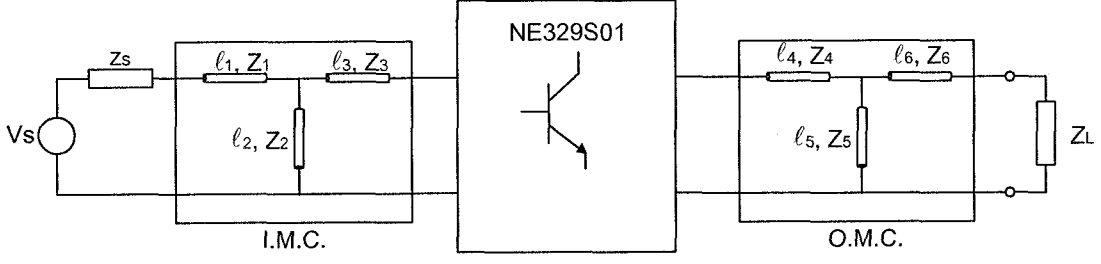
Çizelge 9.1 Tüm uydurma devrelerinin kazançlarının frekans ile değişimleri

Frekans (GHz)	Kazanç (Oran)							
	IMC-T	IMC -II	IMC-SP	IMC-PS	OMC-T	OMC -II	OMC-SP	OMC-PS
2	0,7565	0,9546	0,4197	0,3676	0,1231	0,2158	0,0545	0,1064
3	0,8986	0,9681	0,7261	0,5308	0,9151	0,8237	0,9734	0,8913
4	0,867	0,9982	0,9083	0,7178	0,9659	0,9556	0,97	0,9607
5	0,8578	0,9705	0,87	0,8199	0,9728	0,965	0,9635	0,9824
6	0,926	0,9726	0,8374	0,8349	0,9955	0,9901	0,9664	0,9838
7	0,9647	0,9416	0,8479	0,9028	0,9796	0,953	0,9817	0,9486
8	0,9232	0,9524	0,8851	0,9194	0,9853	0,9479	0,9925	0,9383
9	0,9067	0,9617	0,9059	0,9005	0,9896	0,9697	0,999	0,9628
10	0,9192	0,999	0,8653	0,8814	0,991	0,981	0,9713	0,9807
11	0,9241	0,9569	0,6881	0,8224	0,9863	0,9436	0,9335	0,9332



Şekil 9.6 Tüm uydurma devrelerinin kazanç - frekans karakteristikleri

9.1.1.3 “T” Tipi Kuvvetlendirici Devreleri



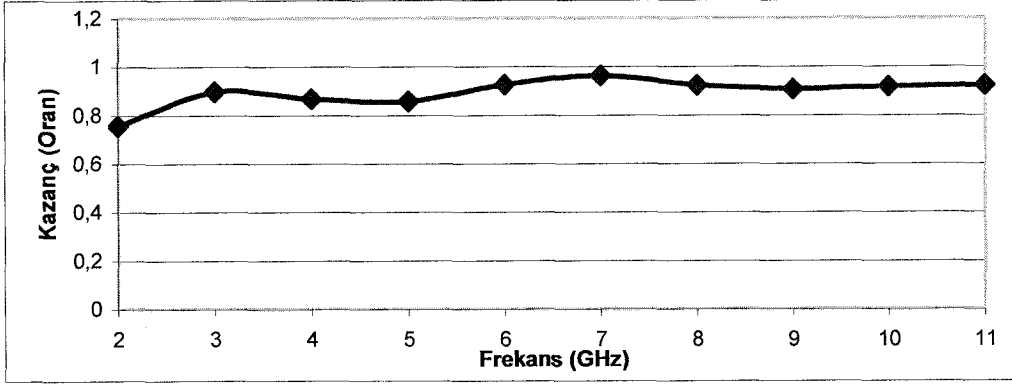
Şekil 9.7 Giriş ve çıkışı “T” tipi Uydurma devreli mikrodalga kuvvetlendiricinin iki kapılı gösterimi

Kaynak Empedansını NE329S01 Transistorunun Giriş Empedansına Uygunlaştıran “T” Tipi Giriş Uydurma Devresinde 3 adet ℓ uzunluğu ve 3 adet Z_0 karakteristik empedansı olmak üzere toplam 6 adet optimizasyon değişkeni kullanılmıştır. Elde edilen tüm $G_{Treq}(\omega_i)$ değerleri birlikte verilmiştir.

Çizelge 9.2 Kaynak empedansı ile NE329S01 transistorunun giriş empedansı değerleri

Frekans(GHz)	Kazanç	Re $\{Z_{out}\}$	Im $\{Z_{out}\}$	Re $\{Z_S\}$	Im $\{Z_S\}$
2	0.7433	78,88	84,08	93.99	180,61
3	0.9134	110,48	111,81	63.52	142.87
4	0.8551	105,85	114,92	51.04	97.3
5	0.8225	84,67	92,82	42.91	67.01
6	0.8952	64,84	60,01	37.57	54.02
7	0.9615	51,47	29,28	36.57	36.59
8	0.9539	43,94	7,02	33.57	26.58
9	0.9336	40,51	-3,31	29.43	15.87
10	0.9175	38,36	2,97	26.92	18.203
11	0.9425	30,06	34,29	24.89	19.480

$$\ell_1=12,6748 \text{ cm}, \ell_2=1,2092 \text{ cm}, \ell_3= 0,3163 \text{ cm}, Z_1=92,9533 \Omega, Z_2=150 \Omega, Z_3= 125,4577 \Omega,$$



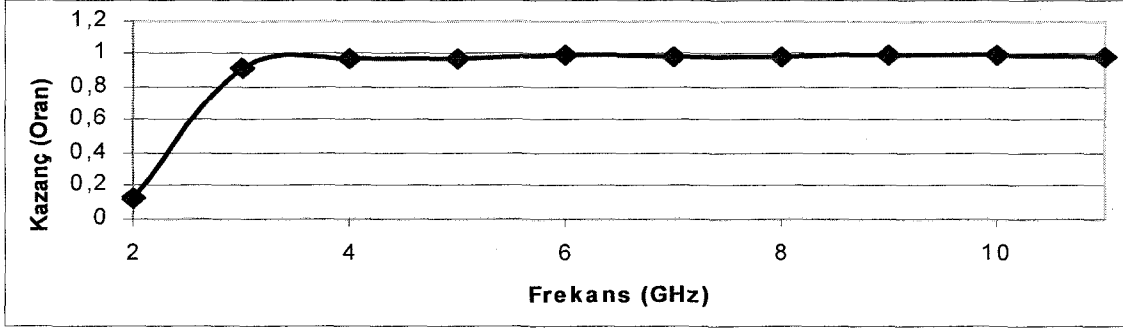
Şekil 9.8 Giriş uydurma devresi için kazanç - frekans karakteristiği

Yük Empedansını Transistorun Çıkış Empedansına Uygunlaştıran “T” Tipi Çıkış Uydurma Devresinde 3 adet ℓ uzunluğu ve 3 adet Z_0 karakteristik empedansı olmak üzere toplam 6 adet optimizasyon değişkeni kullanılmıştır. Elde edilen tüm $G_{Treq}(\omega_i)$ değerleri birlikte verilmiştir.

Çizelge 9.3 Yük empedansı ile NE329S01 transistorunun çıkış empedansı değerleri

Frekans (GHz)	Kazanç	$Re\{Z_{in}\}$	$Im\{Z_{in}\}$	$Re\{Z_S\}$	$Im\{Z_S\}$
2	0,1231	0.1609	-0.4489	16.87	-132.83
3	0,9151	0.2556	-0.7509	14.01	-73.29
4	0,9659	0.4050	-10.685	30.97	-97.55
5	0,9728	0.6746	-13.684	90.71	-148.77
6	0,9955	11.286	-15.128	124.07	-139.91
7	0,9796	16.325	-12.273	199.05	-84.99
8	0,9853	17.096	-0.5676	187.32	-16.18
9	0,9896	13.350	-0.1038	151.03	12.81
10	0,991	0.9311	+0.0131	103.88	-14.06
11	0,9863	0.6599	-0.0624	82.37	-12.13

$$\ell_1=13,8546 \text{ cm}, \ell_2=0,7932 \text{ cm}, \ell_3=0,9020 \text{ cm}, Z_1=150 \Omega, Z_2=139,2937 \Omega, Z_3=92,4048 \Omega$$



Şekil 9.9 Çıkış uydurma devresi için kazanç - frekans karakteristiği

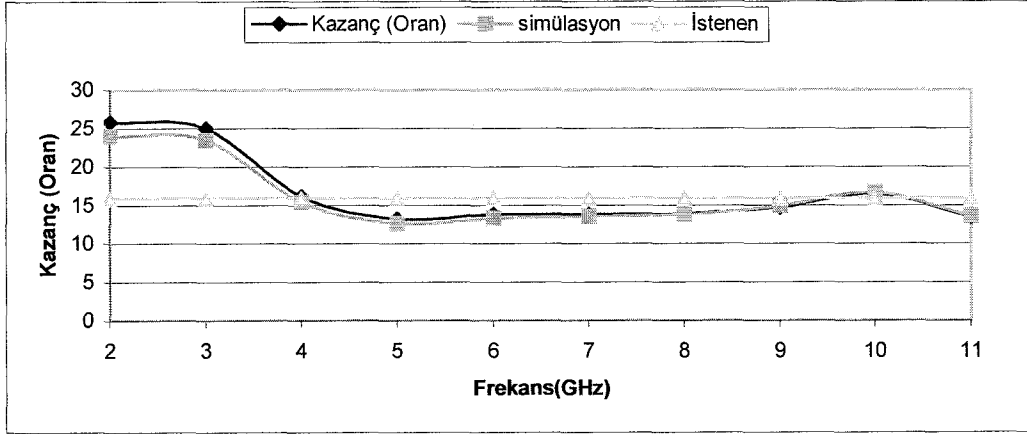
Kaynak Empedansını Transistorun Giriş Empedansına Uygunlaştıran “T” Tipi Giriş Uydurma Devresi sonucunda elde edilen ℓ uzunlukları ve Z_0 empedansları ile Yük Empedansını Transistorun Çıkış Empedansına Uygunlaştıran “T” Tipi Çıkış Uydurma Devresinde bulunan ℓ ve Z_0 değerleri NE329S01 transistörü ile birlikte Şekil 9.7’deki gibi bir devre ile çalıştırılmış ve Çizelge 9.4’deki sonuçlar elde edilmiştir.

Çizelge 9.4 Kuvvetlendiricinin kazanç, gürültü ve giriş VSWR sonuçlarının simülasyon programı sonuçları ile birlikte gösterimi

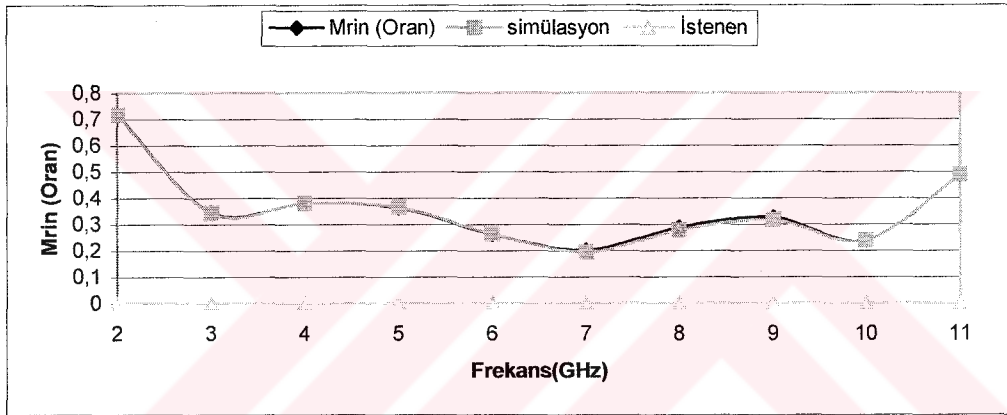
Tasarım Hedef Uzayı: $(F, V_i, G_T, B) \Leftrightarrow (1.112, 1, 15.86, 2-11\text{GHz})$

Frekans (GHz)	Kazanç (Oran)	Simülasyon Sonucu	Mrin (Oran)	Simülasyon Sonucu	Gürültü (Oran)	Simülasyon Sonucu
2	25,8572	24	0,7186	0,717	1,1962	1,122
3	24,9977	23,5	0,3452	0,345	1,1053	1,11
4	16,1604	15,4	0,381	0,381	1,0795	1,08
5	13,1794	12,6	0,3642	0,365	1,0759	1,08
6	13,7645	13,3	0,2638	0,265	1,086	1,09
7	13,7793	13,5	0,2027	0,196	1,1181	1,12
8	13,8485	13,8	0,288	0,277	1,1653	1,16
9	14,718	14,9	0,3269	0,316	1,169	1,17
10	16,4763	16,6	0,2387	0,238	1,1255	1,12
11	13,4308	13,5	0,4877	0,487	1,1013	1,1

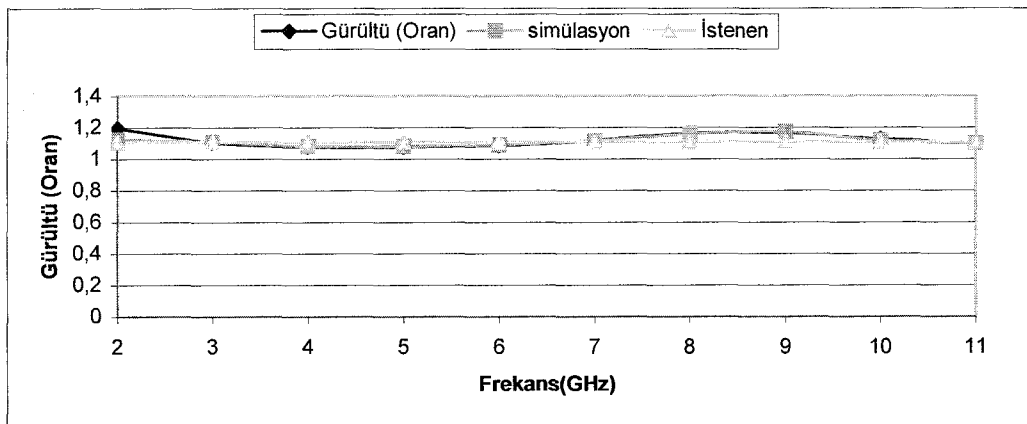
$\ell_1=12,6748$ cm, $\ell_2=1,2092$ cm, $\ell_3=0,3163$ cm, $\ell_4=13,8546$ cm, $\ell_5=0,7932$ cm,
 $\ell_6=0,9020$ cm, $Z_1=92,9533 \Omega$, $Z_2=150 \Omega$, $Z_3=125,4577 \Omega$, $Z_4=150 \Omega$, $Z_5=139,2937 \Omega$,
 $Z_6=92,4048 \Omega$



Şekil 9.10 Girişi ve çıkışı “T” tipi uydurma devreli mikrodalga kuvvetlendiricisinin kazancının frekansa bağlı olarak değişimi

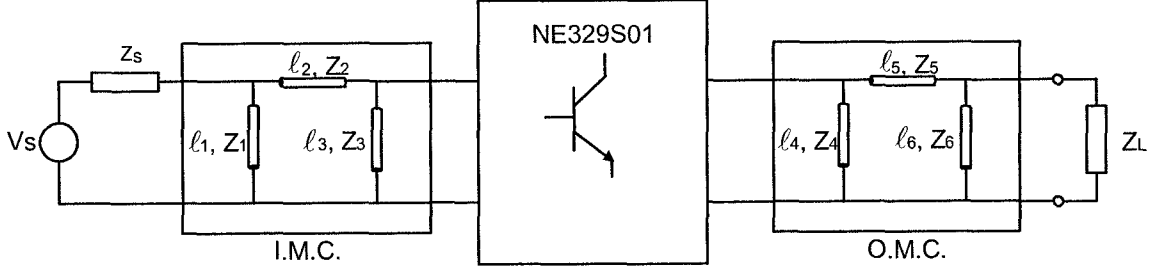


Şekil 9.11 Girişi ve çıkışı “T” tipi uydurma devreli mikrodalga kuvvetlendiricisinin VSWR – frekans değişimi



Şekil 9.12 Girişi ve çıkışı “T” tipi uydurma devreli mikrodalga kuvvetlendiricisinin gürültüsünün frekansa bağlı olarak değişimi

9.1.1.4 “II” TİPİ Kuvvetlendirici Devreleri



Şekil 9.13 Giriş ve çıkışı “II” tipi Uydurma devreli mikrodalga kuvvetlendiricinin iki kapılı gösterimi

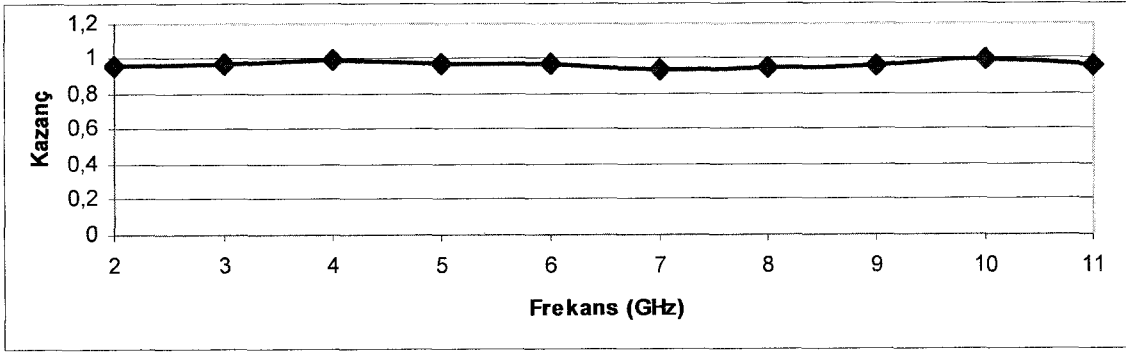
Kaynak Empedansını NE329S01 Transistorunun Giriş Empedansına Uygunlaştıran “II” Tipi Giriş Uydurma Devresinde 3 adet ℓ uzunluğu ve 3 adet Z_0 karakteristik empedansı olmak üzere toplam 6 adet optimizasyon değişkeni kullanılmıştır. Elde edilen tüm $G_{Treq}(\omega_i)$ değerleri birlikte verilmiştir.

Çizelge 9.5 Kaynak empedansı ile NE329S01 transistorunun giriş empedansı değerleri

Frekans(GHz)	Kazanç	$\text{Re}\{Z_{out}\}$	$\text{Im}\{Z_{out}\}$	$\text{Re}\{Z_{TRS}\}$	$\text{Im}\{Z_{TRS}\}$
2	0.9547	110.69	139.40	93.99	180,61
3	0.9681	76.54	121.14	63.52	142.87
4	0.9983	54.09	100.43	51.04	97.3
5	0.9705	40.57	81.36	42.91	67.01
6	0.9726	32.60	64.67	37.57	54.02
7	0.9416	28.10	50.13	36.57	36.59
8	0.9524	25.95	37.36	33.57	26.58
9	0.9617	25.58	26.13	29.43	15.87
10	0.9990	26.70	16.49	26.92	18.203
11	0.9569	28.99	8.84	24.89	19.480

$$\ell_1=0,5940 \text{ cm}, \ell_2=13,3534 \text{ cm}, \ell_3=0,6671 \text{ cm}, Z_1=31,3636 \Omega, Z_2=38,9743 \Omega,$$

$$Z_3= 67,0319 \Omega$$



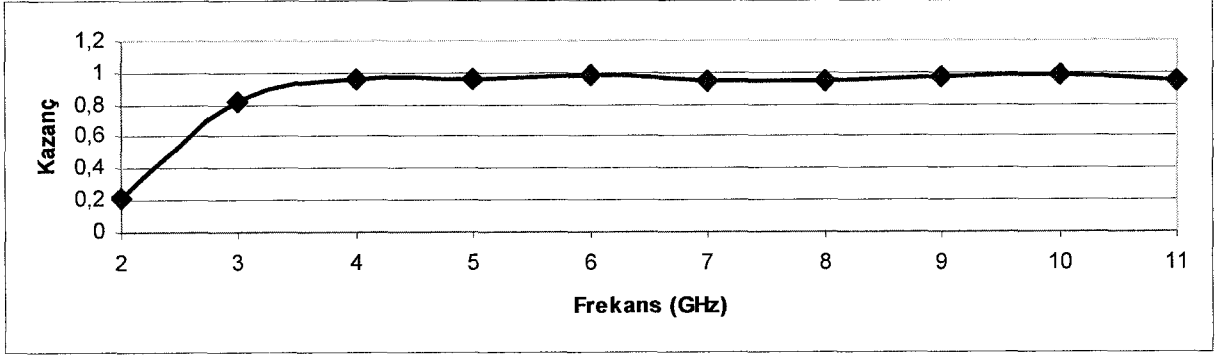
Şekil 9.14 Giriş uydurma devresi için kazanç - frekans Karakteristiği

Yük Empedansını Transistorun Çıkış Empedansına Uygunlaştıran “II” Tipi Çıkış Uydurma Devresinde 3 adet ℓ uzunluğu ve 3 adet Z_0 karakteristik empedansı olmak üzere toplam 6 adet optimizasyon değişkeni kullanılmıştır. Elde edilen tüm $G_{Treq}(\omega_i)$ değerleri birlikte verilmiştir.

Çizelge 9.6 Yük empedansı ile NE329S01 transistorunun çıkış empedansı değerleri

Frekans (GHz)	Kazanç	Re{Z _{in} }	Im{Z _{in} }	Re{Z _{TRL} }	Im{Z _{TRL} }
2	0.2158	25.22	- 54.63	16.87	-132.83
3	0.8237	34.22	- 76.38	14.01	-73.29
4	0.9556	46.95	- 101.41	30.97	-97.55
5	0.9650	68.33	- 128.76	90.71	-148.77
6	0.9901	105.18	- 152.79	124.07	-139.91
7	0.9530	160.75	- 154.63	199.05	-84.99
8	0.9479	207.08	- 106.40	187.32	-16.18
9	0.9697	189.76	- 32.92	151.03	12.81
10	0.9810	126.67	8.30	103.88	-14.06
11	0.9436	67.64	21.24	82.37	-12.13

$$\ell_1=14,02467 \text{ cm}, \ell_2=1,168 \text{ cm}, \ell_3=14,07369 \text{ cm}, Z_1=59,098 \Omega, Z_2=119,92227 \Omega, \\ Z_3=63,5157 \Omega$$



Şekil 9.15 Çıkış uydurma devresi için kazanç - frekans karakteristiği

Kaynak Empedansını Transistorun Giriş Empedansına Uygunlaştıran “II” Tipi Giriş Uydurma Devresi sonucunda elde edilen ℓ uzunlukları ve Z_0 empedansları ile Yük Empedansını Transistorun Çıkış Empedansına Uygunlaştıran “II” Tipi Çıkış Uydurma Devresinde bulunan ℓ ve Z_0 değerleri NE329S01 transistoru ile birlikte Şekil 9.13’deki gibi bir devre ile çalıştırılmış ve Çizelge 9.7’deki sonuçlar elde edilmiştir.

Çizelge 9.7 Kuvvetlendiricinin kazanç, gürültü ve giriş VSWR sonuçlarının simülasyon programı sonuçları ile birlikte gösterimi

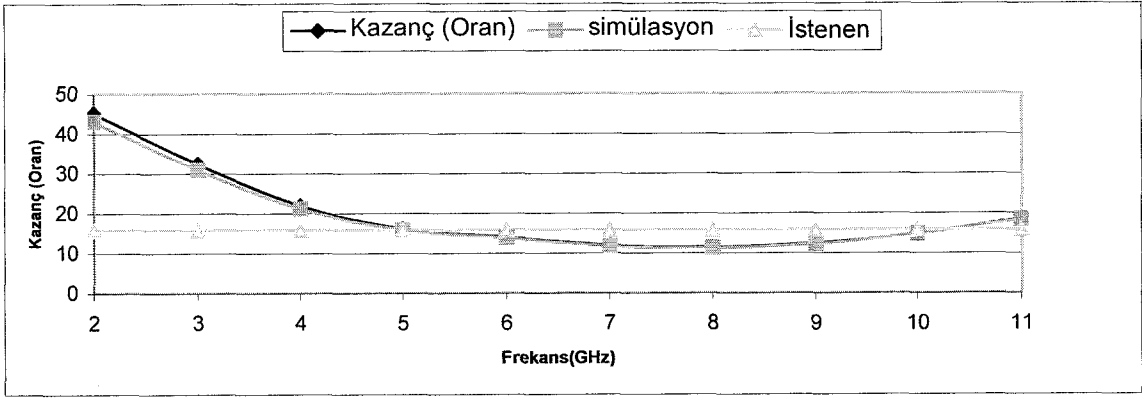
Tasarım Hedef Uzayı: (F, V_i , G_T , B) $\Leftrightarrow$ (1.112, 1, 15.86, 2-11GHz)

Frekans (GHz)	Kazanç (Oran)	Simülasyon Sonucu	MRin (Oran)	Simülasyon Sonucu	Gürültü (Oran)	Simülasyon Sonucu
2	45,2145	43	0,4657	0,468	1,1237	1,13
3	32,3574	31	0,1937	0,22	1,1174	1,11
4	21,9455	21,2	0,0816	0,111	1,104	1,1
5	16,1855	16	0,1376	0,139	1,0936	1,09
6	14,0267	13,8	0,165	0,16	1,1024	1,1
7	11,9048	11,8	0,2134	0,204	1,1102	1,11
8	11,3901	11,3	0,2346	0,228	1,1048	1,1
9	12,4404	12,3	0,2044	0,2	1,0957	1,09
10	14,8892	14,9	0,1299	0,132	1,0978	1,1
11	18,5898	18,4	0,1854	0,186	1,0931	1,09

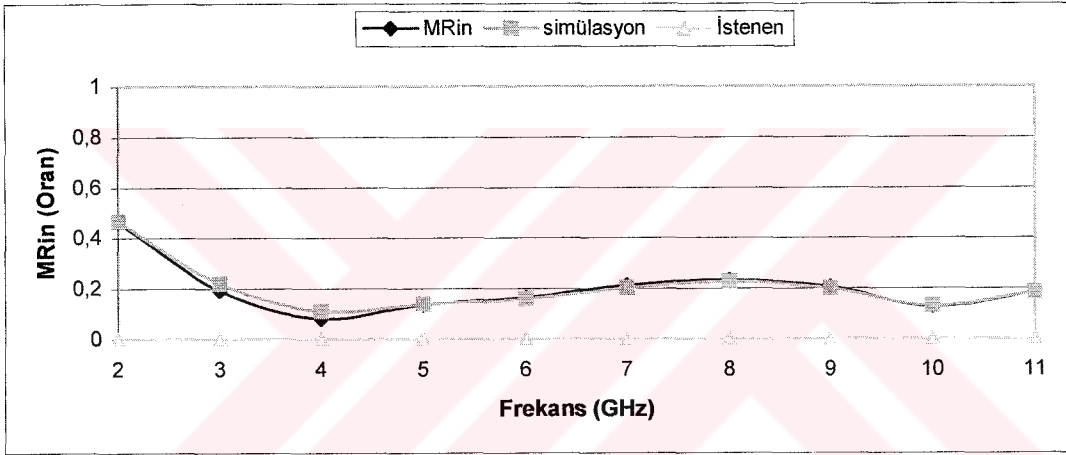
$\ell_1=0,5940$ cm, $\ell_2=13,3534$ cm, $\ell_3=0,6671$ cm, $\ell_4=14,02467$ cm, $\ell_5=1,168$ cm,

$\ell_6=14,07369$ cm $Z_1=31,3636 \Omega$, $Z_2=38,9743 \Omega$, $Z_3= 67,0319 \Omega$, $Z_4=59,098 \Omega$,

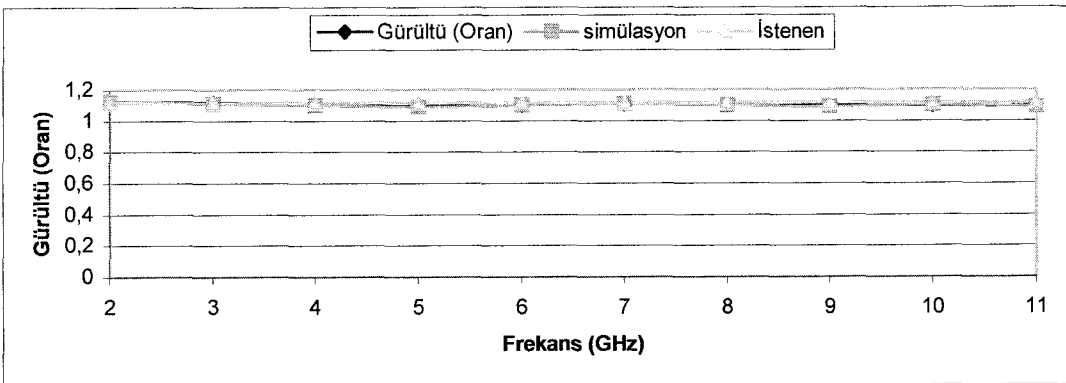
$Z_5=119,92227 \Omega$, $Z_6=63,5157 \Omega$



Şekil 9.16 Girişi ve çıkışı “Π” tipi uydurma devreli mikrodalga kuvvetlendiricisinin kazancının frekansa bağlı olarak değişimi

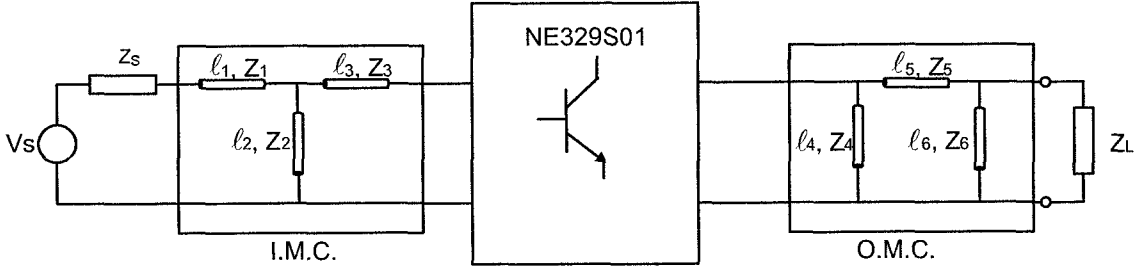


Şekil 9.17 Girişi ve çıkışı “Π” tipi uydurma devreli mikrodalga kuvvetlendiricisinin VSWR – frekans değişimi



Şekil 9.18 Girişi ve çıkışı “Π” tipi uydurma devreli mikrodalga kuvvetlendiricisinin gürültüsünün frekansa bağlı olarak değişimi

9.1.1.5 T-Π Hibrid Mikrodalga Kuvvetlendiricisi



Şekil 9.19 Giriş “T” tipi ve çıkış “Π” tipi Uydurma devreli mikrodalga kuvvetlendiricinin iki kapılı gösterimi

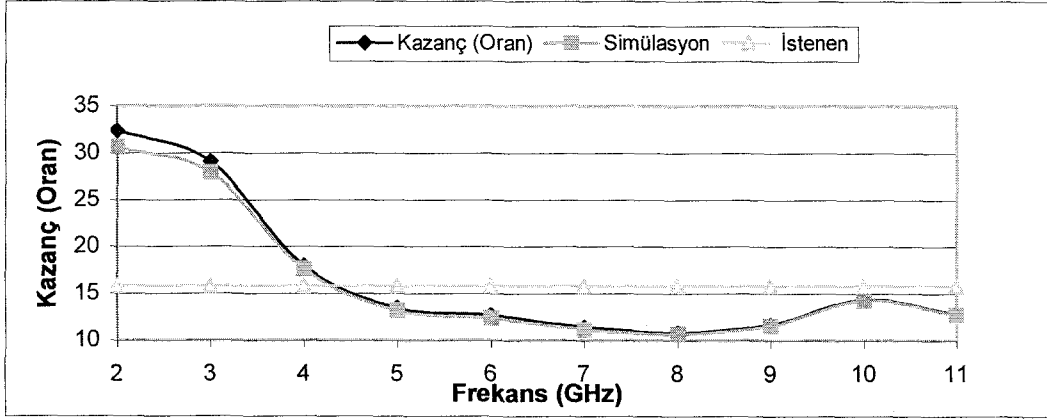
Kaynak Empedansını Transistorun Giriş Empedansına Uygunlaştıran “T” Tipi Giriş Uydurma Devresi sonucunda elde edilen ℓ uzunlukları ve Z_0 empedansları ile Yük Empedansını Transistorun Çıkış Empedansına Uygunlaştıran “Π” Tipi Çıkış Uydurma Devresinde bulunan ℓ ve Z_0 değerleri NE329S01 transistoru ile birlikte Şekil 9.19’deki gibi bir devre ile çalıştırılmış ve Çizelge 9.8’deki sonuçlar elde edilmiştir.

Çizelge 9.8 Kuvvetlendiricinin kazanç, gürültü ve giriş VSWR sonuçlarının simülasyon programı sonuçları ile birlikte gösterimi

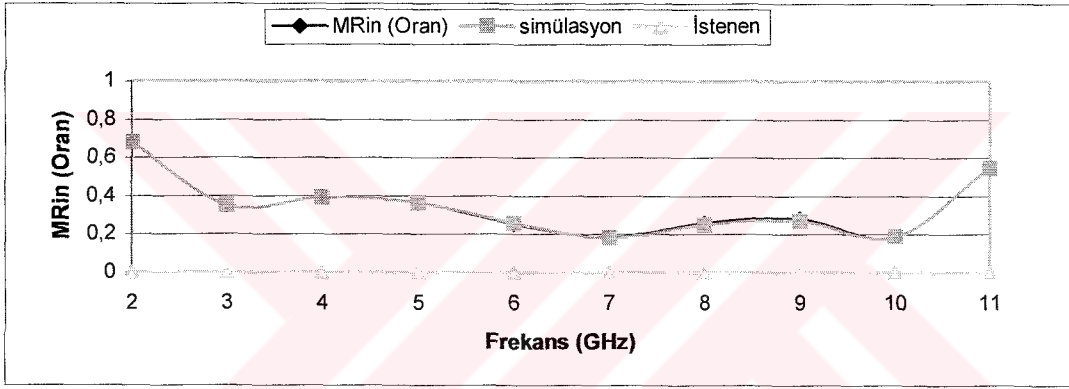
Tasarım Hedef Uzayı: $(F, V_i, G_T, B) \Leftrightarrow (1.112, 1, 15.86, 2-11\text{GHz})$

Frekans (GHz)	Kazanç (Oran)	Simülasyon Sonucu	MRin (Oran)	Simülasyon Sonucu	Gürültü (Oran)	Simülasyon Sonucu
2	32,4053	30,6	0,6814	0,682	1,1962	1,22
3	29,0681	27,9	0,3515	0,354	1,1053	1,11
4	17,933	17,6	0,3912	0,392	1,0795	1,08
5	13,5308	13,2	0,3649	0,367	1,0759	1,08
6	12,7228	12,4	0,255	0,257	1,086	1,09
7	11,4523	11,2	0,1829	0,18	1,1181	1,12
8	10,8279	10,7	0,2587	0,252	1,1653	1,16
9	11,6906	11,6	0,2823	0,273	1,169	1,17
10	14,3935	14,3	0,1887	0,189	1,1255	1,12
11	12,8722	12,8	0,5475	0,547	1,1013	1,1

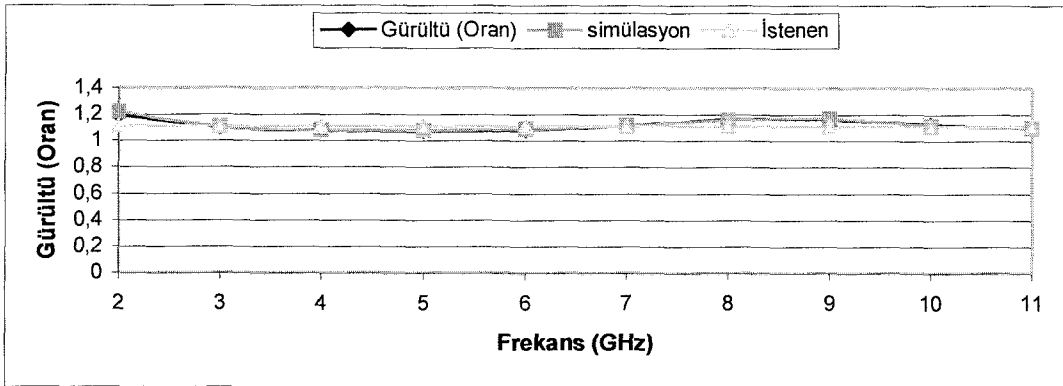
$$\ell_1=12.6748 \text{ cm}, \ell_2=1.2092 \text{ cm}, \ell_3=0.3163 \text{ cm}, \ell_4=14.0388 \text{ cm}, \ell_5=1.2254 \text{ cm}, \\ \ell_6=14.3562 \text{ cm}, Z_1=92.9533 \Omega, Z_2=150 \Omega, Z_3=125.4577 \Omega, Z_4=61.3766 \Omega, Z_5=115.198 \Omega, \\ Z_6=95.972 \Omega$$



Şekil 9.20 Girişi “T” çıkışı “Π” tipi uydurma devreli mikrodalga kuvvetlendiricisinin kazancının frekansa bağlı olarak değişimi

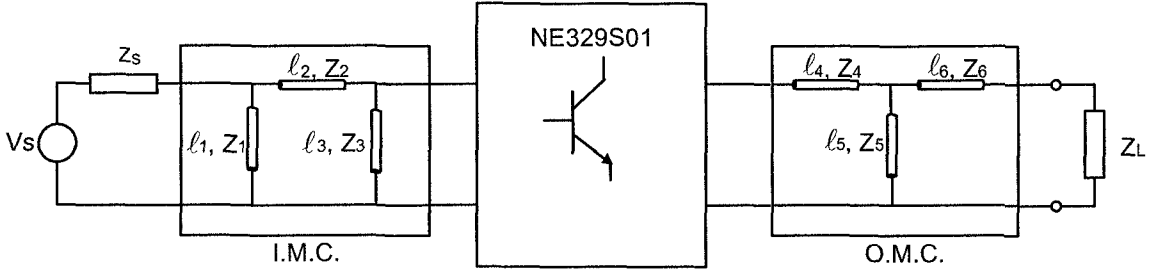


Şekil 9.21 Girişi “T” çıkışı “Π” tipi uydurma devreli mikrodalga kuvvetlendiricisinin VSWR – frekans değişimi



Şekil 9.22 Girişi “T” çıkışı “Π” tipi uydurma devreli mikrodalga kuvvetlendiricisinin gürültüsünün frekansa bağlı olarak değişimi

9.1.1.6 Π -T Hibrid Mikrodalga Kuvvetlendiricisi



Şekil 9.23 Giriş “ Π ” tipi ve çıkış “T” tipi Uydurma devreli mikrodalga kuvvetlendiricinin iki kapılı gösterimi

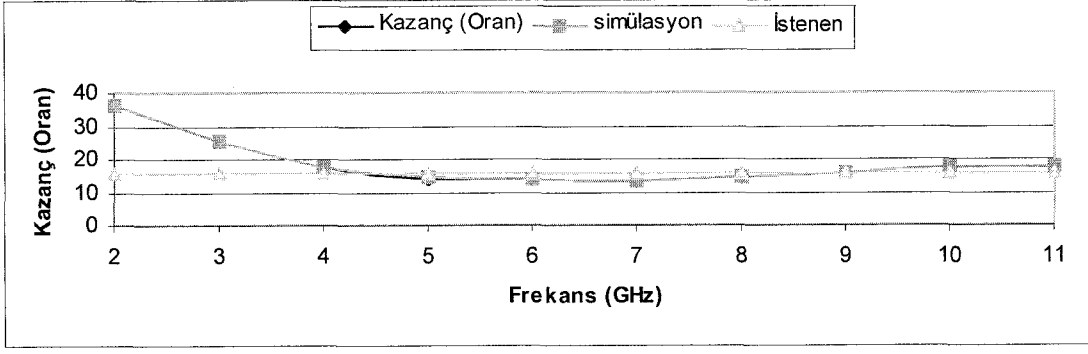
Kaynak Empedansını Transistorun Giriş Empedansına Uygunlaştıran “ Π ” Tipi Giriş Uydurma Devresi sonucunda elde edilen ℓ uzunlukları ve Z_0 empedansları ile Yük Empedansını Transistorun Çıkış Empedansına Uygunlaştıran “T” Tipi Çıkış Uydurma Devresinde bulunan ℓ ve Z_0 değerleri NE329S01 transistoru ile birlikte Şekil 9.23’deki gibi bir devre ile çalıştırılmış ve Çizelge 9.9’deki sonuçlar elde edilmiştir.

Çizelge 9.9 Kuvvetlendiricinin kazanç, gürültü ve giriş VSWR sonuçlarının simülasyon programı sonuçları ile birlikte gösterimi

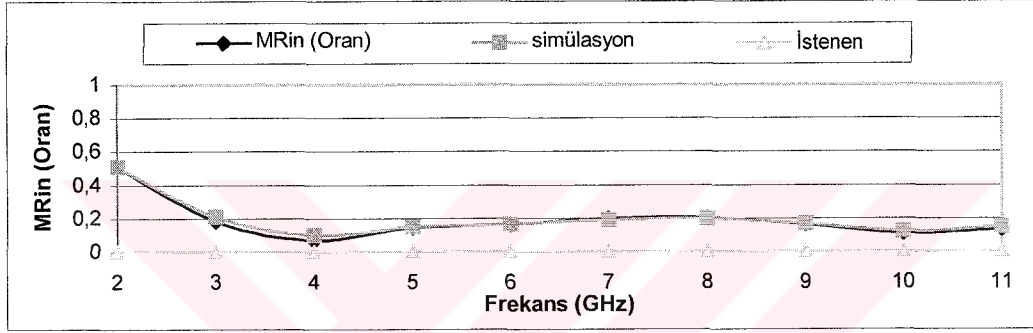
Tasarım Hedef Uzayı: $(F, V_i, G_T, B) \Leftrightarrow (1.112, 1, 15.86, 2-11\text{GHz})$

Frekans (GHz)	Kazanç (Oran)	Simülasyon Sonucu	MRin (Oran)	Simülasyon Sonucu	Gürültü (Oran)	Simülasyon Sonucu
2	36,1208	36,6	0,51	0,509	1,1237	1,13
3	25,5394	25,5	0,1826	0,209	1,1174	1,11
4	17,7076	17,8	0,0692	0,0986	1,104	1,1
5	14,1304	14,3	0,1456	0,148	1,0936	1,09
6	13,8081	13,9	0,1665	0,163	1,1024	1,1
7	13,4069	13,6	0,1989	0,191	1,1102	1,11
8	14,2453	14,4	0,204	0,199	1,1048	1,1
9	15,9589	16	0,1655	0,169	1,0957	1,09
10	17,3286	17,4	0,1107	0,124	1,0978	1,1
11	17,4515	17,5	0,135	0,149	1,0931	1,09

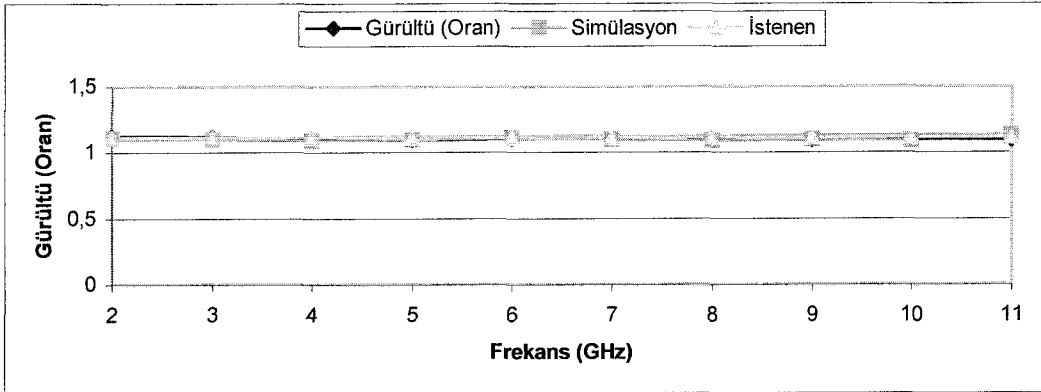
$$\ell_1=0.594 \text{ cm}, \ell_2=13.3534 \text{ cm}, \ell_3=0.6671 \text{ cm}, \ell_4=13.8546 \text{ cm}, \ell_5=0.7932 \text{ cm}, \\ \ell_6=0.902 \text{ cm}, Z_1=31.3636 \Omega, Z_2=38.9743 \Omega, Z_3=67.0319 \Omega, Z_4=150 \Omega, Z_5=139.2937 \Omega, \\ Z_6=92.4048 \Omega$$



Şekil 9.24 Girişi “Π” çıkışı “T” tipi uydurma devreli mikrodalga kuvvetlendiricisinin kazancının frekansa bağlı olarak değişimi

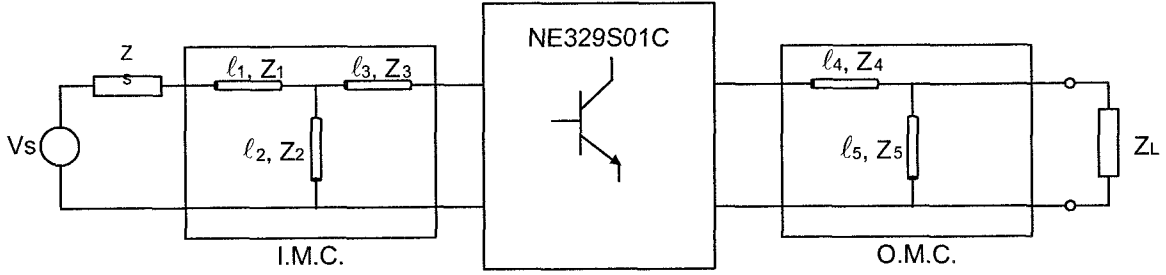


Şekil 9.25 Girişi “Π” çıkışı “T” tipi uydurma devreli mikrodalga kuvvetlendiricisinin VSWR – frekans değişimi



Şekil 9.26 Girişi “Π” çıkışı “T” tipi uydurma devreli mikrodalga kuvvetlendiricisinin gürültüsünün frekansa bağlı olarak değişimi

9.1.1.7 T-L Hibrid Mikrodalga Kuvvetlendiricisi



Şekil 9.27 Giriş “T” tipi, çıkış “L” tipi (Seri ve Paralel)uydurma devreli tek transistorlu kuvvetlendirici

Kaynak Empedansını Transistorun Giriş Empedansına Uyumlaştıran “T” Tipi Giriş Uydurma Devresi sonucunda elde edilen ℓ uzunlukları ve Z_0 empedansları ile Yük Empedansını Transistorun Çıkış Empedansına Uyumlaştıran “L” (Seri ve Paralel) Tipi Çıkış Uydurma Devresinde bulunan ℓ ve Z_0 değerleri NE329S01 transistörü ile birlikte Şekil 9.27’deki gibi bir devre ile çalıştırılmış ve Çizelge 9.10’daki sonuçlar elde edilmiştir.

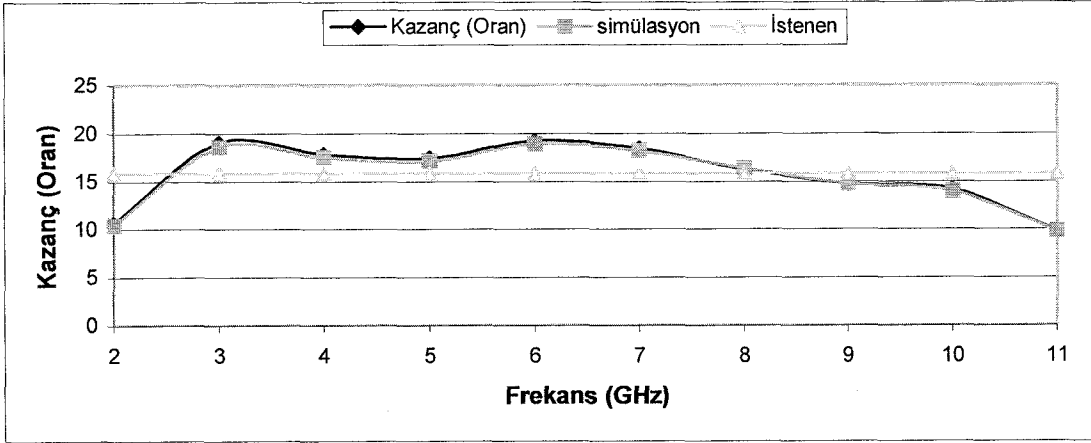
Çizelge 9.10 Kuvvetlendiricinin kazanç, gürültü ve giriş VSWR sonuçlarının simülasyon programı sonuçları ile birlikte gösterimi

Tasarım Hedef Uzayı: $(F, V_i, G_T, B) \Rightarrow (1.112, 1, 15.86, 2-11\text{GHz})$

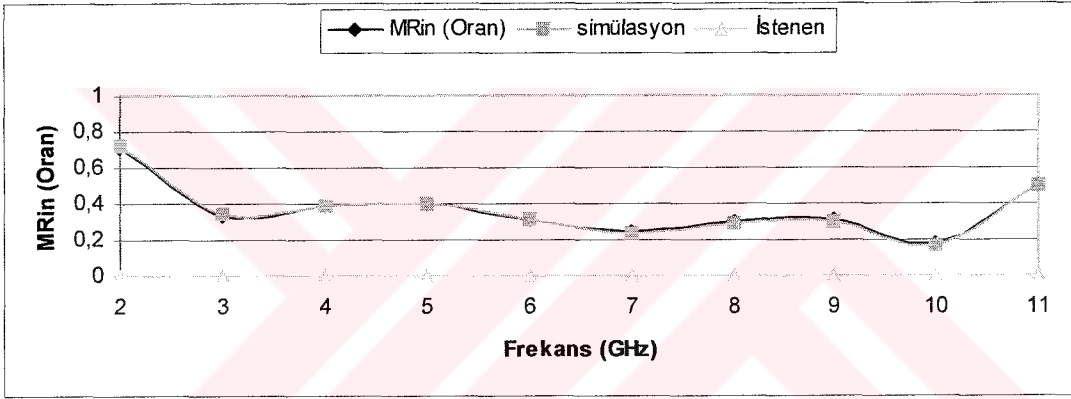
Frekans (GHz)	Kazanç (Oran)	Simülasyon Sonucu	MRin (Oran)	Simülasyon Sonucu	Gürültü (Oran)	Simülasyon Sonucu
2	10,5974	10,4	0,7149	0,717	1,1962	1,22
3	19,0174	18,6	0,3373	0,343	1,1053	1,1
4	17,8401	17,5	0,3894	0,392	1,0795	1,08
5	17,4598	17,1	0,3949	0,397	1,0759	1,08
6	19,2522	18,9	0,3129	0,314	1,086	1,09
7	18,4195	18,2	0,239	0,233	1,1181	1,12
8	16,2481	16,4	0,2995	0,289	1,1653	1,16
9	14,7769	14,8	0,3143	0,302	1,169	1,16
10	14,16	14	0,183	0,167	1,1255	1,12
11	9,8213	9,83	0,4973	0,498	1,1013	1,1

$$\ell_1 = 12.6748 \text{ cm}, \ell_2 = 1.2092 \text{ cm}, \ell_3 = 0.3163 \text{ cm}, \ell_4 = 14.2454 \text{ cm}, \ell_5 = 14.3077 \text{ cm}$$

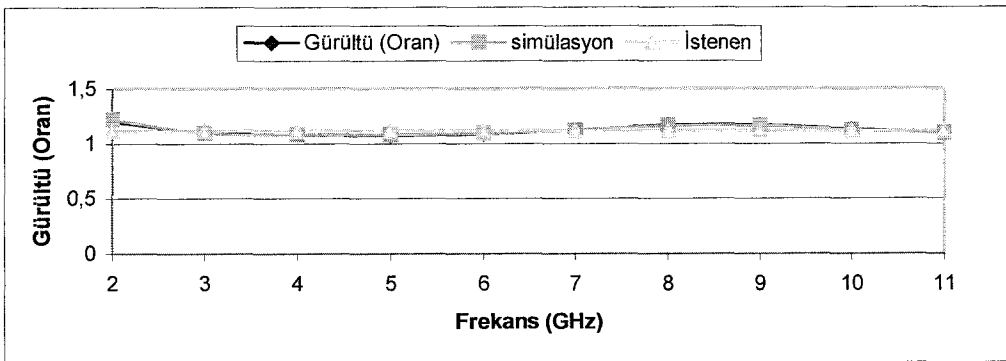
$$Z_1 = 92.9533 \Omega, Z_2 = 150 \Omega, Z_3 = 125.4577 \Omega, Z_4 = 81.2378 \Omega, Z_5 = 58.0025 \Omega$$



Şekil 9.28 Giriş “T” tipi, çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistorlu kuvvetlendiricinin kazancının frekansa bağlı olarak değişimi

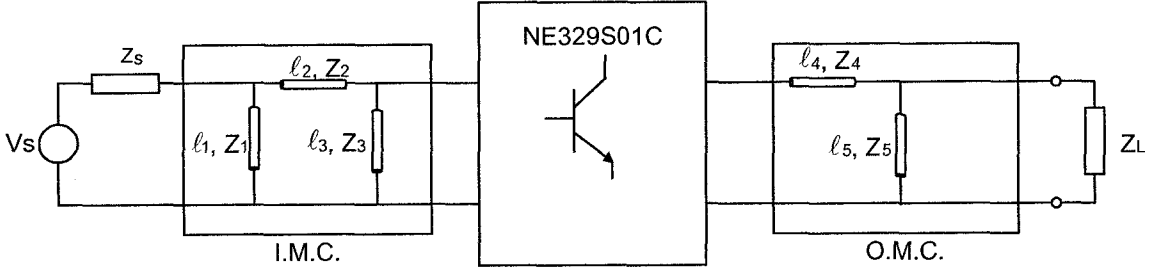


Şekil 9.29 Giriş “T” tipi, çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistorlu kuvvetlendiricinin VSWR – frekans değişimi



Şekil 9.30 Giriş “T” tipi, çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistorlu kuvvetlendiricinin gürültüsünün frekansa bağlı olarak değişimi

9.1.1.8 Π -L Hibrid Mikrodalga Kuvvetlendiricisi



Şekil 9.31 Giriş “ Π ” ve çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistorlu kuvvetlendirici

Kaynak Empedansını Transistorun Giriş Empedansına Uygunlaştıran “ Π ” Tipi Giriş Uydurma Devresi sonucunda elde edilen ℓ uzunlukları ve Z_0 empedansları ile Yük Empedansını Transistorun Çıkış Empedansına Uygunlaştıran “L” (Seri ve Paralel) Tipi Çıkış Uydurma Devresinde bulunan ℓ ve Z_0 değerleri NE329S01 transistörü ile birlikte Şekil 9.31’deki gibi bir devre ile çalıştırılmış ve Çizelge 9.11’deki sonuçlar elde edilmiştir

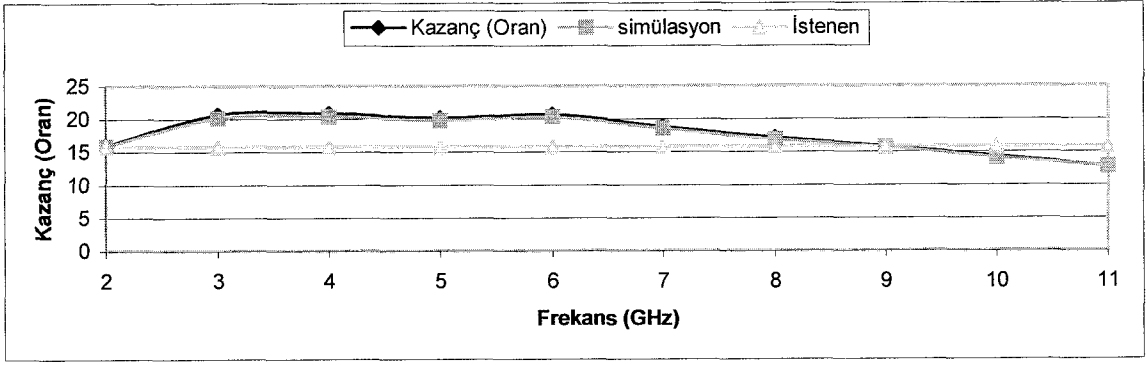
Çizelge 9.11 Kuvvetlendiricinin Kazanç, Gürültü ve VSWR sonuçlarının simülasyon programı sonuçları ile birlikte gösterimi

Tasarım Hedef Uzayı: $(F, V_i, G_T, B) \Leftrightarrow (1.112, 1, 15.86, 2-11\text{GHz})$

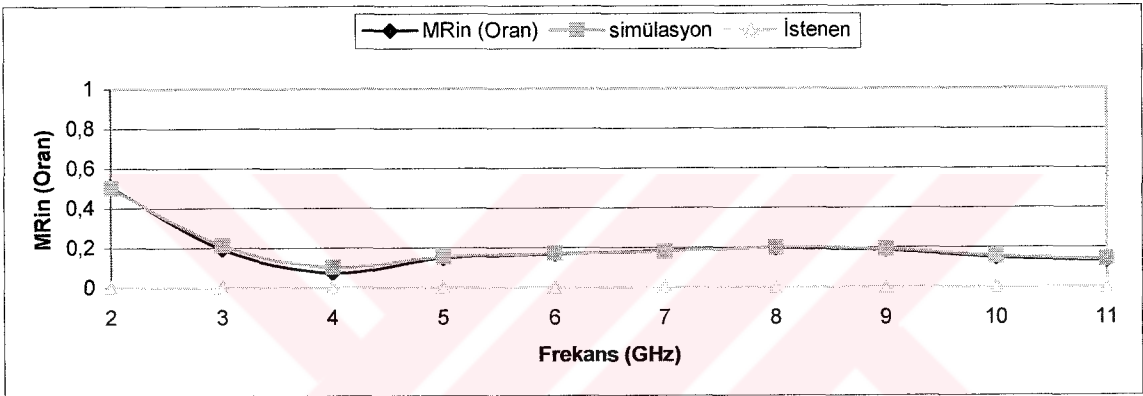
Frekans (GHz)	Kazanç (Oran)	Simülasyon Sonucu	MRin (Oran)	Simülasyon Sonucu	Gürültü (Oran)	Simülasyon Sonucu
2	16,0503	15,9	0,5094	0,503	1,1237	1,13
3	20,6561	20,1	0,1934	0,212	1,1174	1,11
4	20,917	20,4	0,0727	0,103	1,104	1,1
5	20,2303	19,8	0,1483	0,156	1,0936	1,09
6	20,7455	20,3	0,1671	0,171	1,1024	1,1
7	18,8858	18,6	0,1824	0,181	1,1102	1,11
8	17,1486	16,9	0,1981	0,199	1,1048	1,1
9	15,8234	15,7	0,1871	0,191	1,0957	1,09
10	14,3301	14,2	0,1479	0,154	1,0978	1,1
11	12,8164	12,8	0,1333	0,14	1,0931	1,09

$$\ell_1 = 0.594 \text{ cm}, \ell_2 = 13.3534 \text{ cm}, \ell_3 = 0.6671 \text{ cm}, \ell_4 = 14.2454 \text{ cm}, \ell_5 = 14.3077 \text{ cm}$$

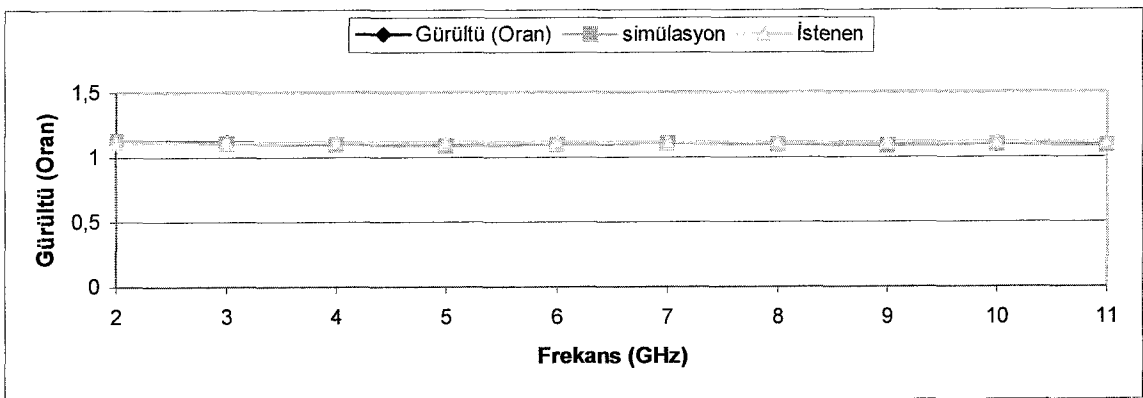
$$Z_1 = 31.3636 \Omega, Z_2 = 38.9743 \Omega, Z_3 = 67.0319 \Omega, Z_4 = 81.2378 \Omega, Z_5 = 58.0025 \Omega$$



Şekil 9.32 Giriş “Π” tipi, çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistorlu kuvvetlendiricinin kazancının frekansa bağlı olarak değişimi

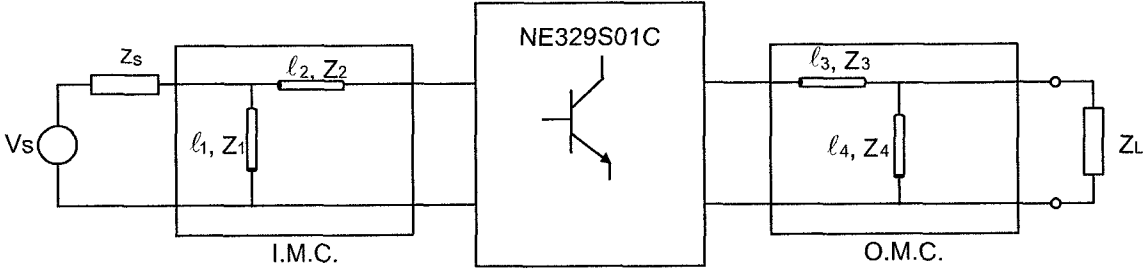


Şekil 9.33 Giriş “Π” tipi, çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistorlu kuvvetlendiricinin VSWR – frekans değişimi



Şekil 9.34 Giriş “Π” tipi, çıkış “L” tipi (Seri ve Paralel) uydurma devreli tek transistorlu kuvvetlendiricinin gürültüsünün frekansa bağlı olarak değişimi

9.1.1.9 L Tipi Mikrodalga Kuvvetlendiricisi



Şekil 9.35 Giriş ve çıkış “L” tipi Uydurma devreli tek transistorlu kuvvetlendirici

Kaynak Empedansını Transistorun Giriş Empedansına Uygunlaştıran “L” (Paralel ve Seri) Tipi Giriş Uydurma Devresi sonucunda elde edilen ℓ uzunlukları ve Z_0 empedansları ile Yük Empedansını Transistorun Çıkış Empedansına Uygunlaştıran “L” (Seri ve Paralel) Tipi Çıkış Uydurma Devresinde bulunan ℓ ve Z_0 değerleri NE329S01 transistoru ile birlikte Şekil 9.35’deki gibi bir devre ile çalıştırılmış ve Çizelge 9.12’deki sonuçlar elde edilmiştir

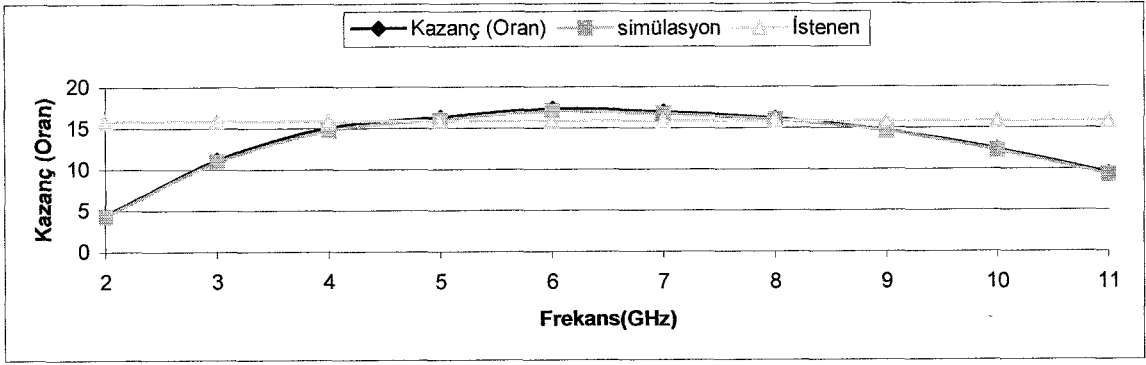
Çizelge 9.12 Kuvvetlendiricinin kazanç, gürültü ve giriş VSWR sonuçlarının simülasyon programı sonuçları ile birlikte gösterimi

Tasarım Hedef Uzaı: $(F, V_i, G_T, B) \Leftrightarrow (1.112, 1, 15.86, 2-11\text{GHz})$

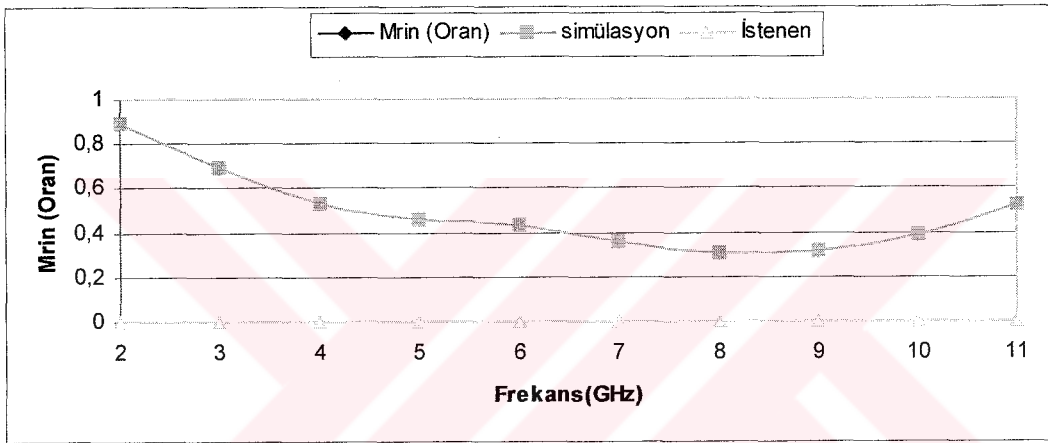
Frekans (GHz)	Kazanç (Oran)	Simülasyon Sonucu	Mrin (Oran)	Simülasyon Sonucu	Gürültü (Oran)	Simülasyon Sonucu
2	4,4407	4,33	0,8917	0,893	1,4675	1,55
3	11,1745	10,9	0,6923	0,695	1,2874	1,29
4	15,0457	14,7	0,5334	0,54	1,2189	1,22
5	16,3589	16	0,4573	0,459	1,1747	1,17
6	17,3371	17	0,4332	0,432	1,161	1,16
7	16,99	16,7	0,361	0,359	1,1421	1,14
8	16,1788	16	0,3059	0,31	1,1225	1,12
9	14,7229	14,6	0,3195	0,317	1,1009	1,1
10	12,4299	12,3	0,3893	0,383	1,0974	1,1
11	9,4284	9,29	0,5267	0,528	1,1224	1,12

$$\ell_1 = 1.1154 \text{ cm}, \ell_2 = 0.5108 \text{ cm}, \ell_3 = 14.2454 \text{ cm}, \ell_4 = 14.3077 \text{ cm}$$

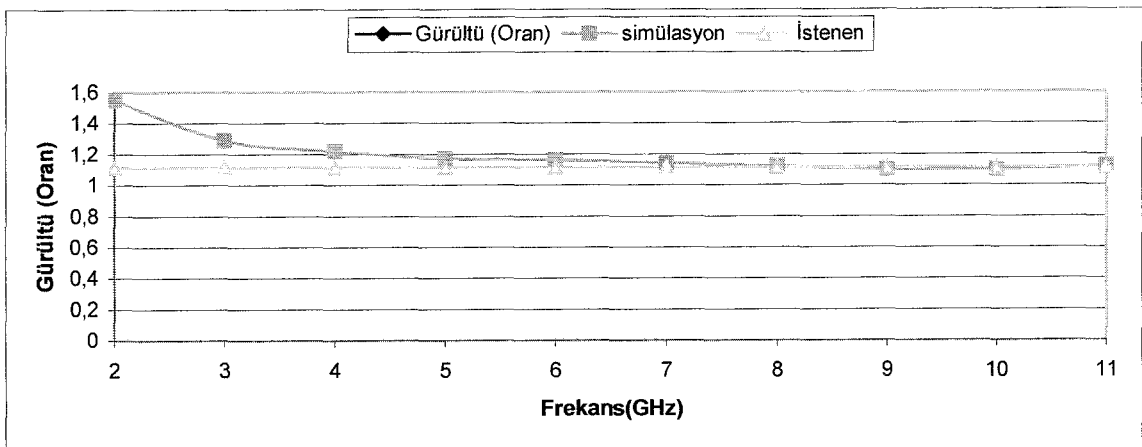
$$Z_1 = 119.705 \Omega, Z_2 = 66.5783 \Omega, Z_3 = 81.2378 \Omega, Z_4 = 58.0025 \Omega$$



Şekil 9.36 Giriş ve çıkış “L” tipi uydurma devreli tek transistorlu kuvvetlendiricinin kazancının frekansa bağlı olarak değişimi



Şekil 9.37 Giriş ve çıkış “L” tipi uydurma devreli tek transistorlu kuvvetlendiricinin VSWR – frekans değişimi



Şekil 9.38 Giriş ve çıkış “L” tipi uydurma devreli tek transistorlu kuvvetlendiricinin gürültüsünün frekansa bağlı olarak değişimi

10. SONUÇLAR

Bu çalışma da daha önceki çalışmalar gibi, tasarım değişkenleri ve tasarım hedefi alt uzaylarından oluşan gerçekleştirilebilir tasarım uzayı olarak adlandırdığımız, mikrodalga kuvvetlendiricilerin optimizasyonundaki en büyük sorunlardan biri çözülmesi üzerine kurulmuştur. Bununla birlikte tamamen yeni bir yapı sunmak yerine oldukça güncel çalışmaların analizi yapılarak bulunan sorunlar çerçevesinde yeni bir parametre takımı ile tekrar düzenlenmesi yapılmıştır. Daha önceki çalışmalarda kullanılan ABCD parametrelerinin yerini bu çalışmada transfer saçılma parametreleri yani [A] parametreleri almıştır. Birden fazla çalışmanın bir araya getirilmesine ek olarak kullanılan algoritmalar üzerinde yapılan geliştirmeler ile çok daha efektif bir optimizasyon süreci oluşturulması sağlanmıştır.

Bu çalışma iki aşamadan oluşmuştur:

- Giriş ve çıkış uydurma devrelerinin genetik algoritma ile optimizasyonu
- Elde edilen sonuçların klasik metodlarla tekrar gözden geçirildikten sonra tüm devreye uygulanması

Çalışmanın ilk aşaması tezin asıl kısmını ve yapılan çalışmanın hem emek hem de zaman açısından en büyük kısmını almıştır. Daha önceki çalışmaların anlaşılabilmesi ve bulunan bazı eksikliklerin tekrar düzenlenmesi aşamasında aylarca algoritmalar tekrarlanmış ve toplamda milyonlar mertebesinde iterasyonlar yürütülmüştür. Yapılan çalışmalar sonucunda optimizasyon sürecinin hızlandırılması ve hat uzunluk ve kalınlıklarının ayrı ayrı sınırlar içinde ele alınması sağlanmıştır. 0.5-25 ohm arasında bulunan ve mikroşerit mimarisi için pek de uygun sayılmayan bir bölgenin hesaplama alanından çıkartılması ile elde edilen optimizasyon süreci “Optimize” edilmiştir.

İkinci aşamada elde edilen sonuçlar göstermiştir ki genetik algoritma ile elde edilen sonuçlar diğer optimizasyon teknikleri ile tekrar optimize edildiklerinde çok büyük bir değişim göstermemektedirler. İterasyon sayısı ne kadar yüksek tutulursa değişim ve sonuçlar arasındaki fark o kadar azalmaktadır. Bu aşamada elde edilen sonuçlar şunu göstermektedir ki yeniden düzenlenen genetik optimizasyon süreci ek bir işleme ihtiyaç duymaksızın giriş ve çıkış uydurma devrelerini en iyi biçimde optimize etmektedir. Simülasyon sonuçlarının elde edilen sonuçlarla neredeyse tamamen çakışıyor olması yapılan işlemin doğruluğunu göstermektedir.

Elde bulunan genetik optimizasyon süreci şu anda getirilebileceği en üst noktalardan birinde

bulunmaktadır. Bu aşamada yapılacak yeni bir çalışmada iki yaklaşım sergilenebilir. Bunlardan ilki kullanılan mantığı değiştirip sonradan yapılan düzenlemeleri oluşturulacak yeni bir çekirdeğin içine yerleştirmektir. İkinci yaklaşım ise burada üzerinde durmadığımız diğer uydurma devre alternatifleri üzerine elde edilen mantığı uygulayıp yüksek sayıda iterasyon yardımı ile teknolojik ihtiyacın gerekleri doğrultusunda tasarımlar oluşturmak olacaktır. Algoritma yapısının esnekliği ve istenilen sayıda değişkenli devreler üzerine uygulanabilirliği farklı problemlerin optimizasyonu için de bu metodu uygun kılmaktadır.

Bu çalışma ile elde edilen sonuçlar, daha önce hazırlanan çalışmalar ile kıyaslandığında çok daha gerçekleştirilebilir bir yapıya oturtulmuştur. Bundan sonraki çalışmalardan beklenen, bu aralığın daha da daraltılarak bu çalışmada 14cm seviyelerine kadar çıkabilen uzunluk değerlerini ve 150 ohm'a varabilen empedansları daha da düşürerek mikroşerit mimarisine çok daha uygun hale getirmesi olacaktır. Bunun yanında M_{in} değerlerinin sıfır seviyesine çekilmesi de sağlanabilir.

Kullanılan kuvvetlendiricinin izin verdiği sınırlar içinde, mikroşerit mimarisinin gereklilikleri doğrultusunda transmisyon hatları için belirlenen sınırlar yapılacak yeni çalışmalarda değiştirilebilir. Bunun yanında bu çalışmada kullandığımız NE329S01 transistörünün günümüz şartları içinde üretilmiş olan bir transistör olup ileride sunulacak yeni teknolojiler doğrultusunda yerini çok daha gelişmiş transistörlere bırakabileceğini ve sınırların da bu transistörler doğrultusunda farklı noktalara gelebileceğini unutmamak gerekir.

KAYNAKLAR

- Aliyev, I., (2001), “Performans (F , V_i , G_i) Üçlüleri Kullanılarak Geniş Bandlı Mikrodalga Kuvvetlendirici Tasarımı”, Yüksek Lisans Tezi, YTÜ Fen Bilimleri Enstitüsü.
- Cengiz Y. (2004), “Optimum Performanslı Mikrodalga kuvvetlendirici Tasarımı”, Doktora Tezi, YTÜ Fen Bilimleri Enstitüsü.
- Collins R.E. ,“ Foundation For Microwave Engineering” Mcgraw-Hill 1992 Pp 334-339
- Erol O. K. (1995), “Bulanık Kontrolör Karar Tablosunun Genetik Algoritmalarla Oluşturulması” Yüksek Lisans Tezi İTÜ. Fen Bilimleri Enstitüsü, 1-13.
- Frickey Dean A., “Conversions Between S, Z, Y, h, ABCD and T Parameters which are Valid for Complex Source and Load Impedances IEEE Transactions on Microwave Theory and Techniques, Vol:42, No:2, February 1994
- Güneş, F., Cengiz, Y. ,”Optimization of a Microwave Amplifier Using Neural Performance Data Sheets With Genetic Algorithms”, International Conference on Artificial Neural Networks (ICANN) 2003, 630-637.
- Güneş, F., Torpi, H., Gürgen, F., “A Multidimensional Signal-Noise Neural Model for Microwave Transistor” IEE Proc-Circuits Devices And Systems Vol 145, No 2, 1998 Pp 111-117
- Güneş F., Güneş M. Ve Fidan M., “ Performance Characterisation Of A Microwave Transistör”, Ieee Proc. Circuits Devices System, Vol. 141, No:5, October 1994
- Güneş F., Çetiner B.A. , “Smith Chart Formulation Of Performance Characterisation For A Microwave Transistor”, IEEE Proc. Circuits Devices Syst. Vol. 145, No:6, December 1998
- Kargın, S., (2000), ” Genetik Algoritma Yardımı İle Yapay Açıklıklı Radarda Hedef Konumunun Belirlenmesi”, Doktora Tezi, İTÜ Fen Bilimleri Enstitüsü. 6-34.
- Trick Timoty N., Vlach Jiri, “Computer Aided Design of Broadband Amplifiers with Complex Loads”, IEEE Transactions on Microwave Theory and Techniques, Vol. MTT-18, No:9, September 1970.

EKLER

Ek 1	Mikrodalga Kuvvetlendiricisi Hedef Fonksiyonu Programları
Ek 2	Mikrodalga Kuvvetlendiricisi Toplam Devre Hesap Programları
Ek 3	Genetik Algoritma Optimizasyon Programları



Ek 1 Mikrodalga Kuvvetlendiricisi Hedef Fonksiyonu Programları

IMC_CIS

```
function E = imc_cis(l)
```

```
% Program ilk Ilgar ALIYEV tarafından hazırlanmış, sonradan GT'ye göre iyileştirme için
% CIS tarafından değiştirilmiş (CIS : Can Cevikbas, Ilkay Koseli, Serhat Altunc)
% Daha sonra A- parametreleri için Umut Hincal ve Yavuz Cengiz tarafından düzenlenmiştir.
%
% Danisman : Prof.Dr. Filiz GÜNES
%
```

```
% Giriş bilgileri '.csv' uzantılı dosyalardan okunur
```

```
global Zs_In Zoutsc Zinsec Zoutsd gmout gmin GT
Const = DLMREAD('Dosya_Const.csv',';',0,0,'A1..J1');
Zs_In = DLMREAD('Zs_Input.csv',';',0,0,'A1..E10');
```

```
% Kaynak ve Yük Empedansları
n=10;
```

```
Zs_const=50;
Zl_const=50;
```

```
E1 = zeros(2,2);
E2 = zeros(2,2);
E3 = zeros(2,2);
```

```
M = zeros(2,2);
```

```
Zoutsc=zeros(n,1);
Zinsec=zeros(n,1);
gmout=zeros(n,1);
gmin=zeros(n,1);
GT=zeros(n,1);
toplama=zeros(n,1);
```

```
% z, optimize edilecek hat parçalarının karakteristik empedansları
```

```
z1=l(4);
z2=l(5);
z3=l(6);
```

```
l1=l(1);
l2=l(2);
l3=l(3);
```

```
Theta = (2*pi/(Const(3)))*kron(l*1e-2,Zs_In(:,1)*1e9); % 1 cm. cinsinden ve frekans GHz
cinsinden olduğu için 0.01 ve 1*10^9 ile çarpılıyor.
```

```
Zoutsd = Zs_In(:,2)+i.*Zs_In(:,3);
```

```
Zn=50;
```

```
% bu parametreler A parametreleridir.
```

```
for n = 1:10
```

```
E1=[(((cos(Theta(n,1))).*Zn)+(i.*z1.*sin(Theta(n,1)))+(i.*sin(Theta(n,1))*Zn^2)./z1)+(cos(
    Theta(n,1))*Zn)./(2*Zn),(((cos(Theta(n,1))).*Zn)-
    (i.*z1.*sin(Theta(n,1)))+(i.*sin(Theta(n,1))*Zn^2)./z1)-
    (cos(Theta(n,1))*Zn)./(2*Zn);(((cos(Theta(n,1))).*Zn)+(i.*z1.*sin(Theta(n,1)))-
    (i.*sin(Theta(n,1))*Zn^2)./z1)-
    (cos(Theta(n,1))*Zn)./(2*Zn),(((cos(Theta(n,1))).*Zn)-(i.*z1.*sin(Theta(n,1)))-
    (i.*sin(Theta(n,1))*Zn^2)./z1)+(cos(Theta(n,1))*Zn)./(2*Zn)];
```

```
E2=[((2-(i.*Zn)./(z2.*tan(Theta(n,2))))/2,-
    ((i.*Zn)./(z2.*tan(Theta(n,2))))/2;((i.*Zn)./(z2.*tan(Theta(n,2))))/2,((2+(i.*Zn)./(z
    2.*tan(Theta(n,2))))/2)];
```

```
E3=[(((cos(Theta(n,3))).*Zn)+(i.*z3.*sin(Theta(n,3)))+(i.*sin(Theta(n,3))*Zn^2)./z3)+(cos(
    Theta(n,3))*Zn)./(2*Zn),(((cos(Theta(n,3))).*Zn)-
    (i.*z3.*sin(Theta(n,3)))+(i.*sin(Theta(n,3))*Zn^2)./z3)-
    (cos(Theta(n,3))*Zn)./(2*Zn);(((cos(Theta(n,3))).*Zn)+(i.*z3.*sin(Theta(n,3)))-
    (i.*sin(Theta(n,3))*Zn^2)./z3)-
    (cos(Theta(n,3))*Zn)./(2*Zn),(((cos(Theta(n,3))).*Zn)-(i.*z3.*sin(Theta(n,3)))-
    (i.*sin(Theta(n,3))*Zn^2)./z3)+(cos(Theta(n,3))*Zn)./(2*Zn)];
```

```
M = E1*E2*E3;
```

```
S(1,1)=(M(2,1)./M(1,1));
```

```
S(1,2)=1./M(1,1);
```

```
S(2,1)=1./M(1,1);
```

```
S(2,2)=-(M(1,2)./M(1,1));
```

```
Ros=(Zs_const-Zn)/(Zs_const+Zn);
```

```
%Zlimc(n) IMC devresinin yukudur ve transistor kaynak empedansinin eslenigi olarak
    alınmıştır
```

```
Zlimc(n) = conj(Zoutsd(n));
```

```
Rol(n)=((Zlimc(n)-Zn)/(Zlimc(n)+Zn));
```

```
Roout(n)=S(2,2)+(S(1,2).*S(2,1).*Ros)./(1-S(1,1).*Ros);
```

```
%Zoutimec(n)=Zn*((1+Roout(n))./(1-Roout(n)));
```

```
%Rooutgen(n)=((Zoutimec(n)-Zoutsd)./(Zoutimec(n)-Zlimc(n)));
```

```
%GT(n)=1-(abs(Rooutgen(n)).^2)
```

```
GT(n)=(((1-(abs(Ros)).^2).*(1-(abs(Rol(n))).^2).*(abs(S(2,1)).^2))./(((abs(1-
    S(1,1).*Ros)).^2).*((abs(1-Roout(n)).*Rol(n))).^2)))
```

```
end
```

```
% Hata Islevi
```

```
toplamlam = (1-exp(-1*((1 - GT).^2)));
```

E = sum(toplam)



IMC_CISP

```

function E = imc_cis(l)
% Program ilk Ilgar ALIYEV tarafından hazırlanmış, sonradan GT'ye göre iyileştirme için
% CIS tarafından değiştirilmiş (CIS : Can Cevikbas, Ilkay Koseli, Serhat Altunc)
% Daha sonra A- parametreleri için Umut Hincal ve Yavuz Cengiz tarafından düzenlenmiştir.
%
% Danisman : Prof.Dr. Filiz GÜNES
% Giriş bilgileri '.csv' uzantılı dosyalardan okunur

global Zs_In Zoutsc Zinsec Zoutsd gmout gmin GT
Const = DLMREAD('Dosya_Const.csv',';',0,0,'A1..J1');
Zs_In = DLMREAD('Zs_Input.csv',';',0,0,'A1..E10');

% Kaynak ve Yük Empedansları
n=10;

Zs_const=50;

Zl_const=50;

E1 = zeros(2,2);
E2 = zeros(2,2);
E3 = zeros(2,2);

M = zeros(2,2);

Zoutsc=zeros(n,1);
Zinsec=zeros(n,1);
gmout=zeros(n,1);
gmin=zeros(n,1);
GT=zeros(n,1);
toplam=zeros(n,1);

% z, optimize edilecek hat parçalarının karakteristik empedansları

z1=l(4);
z2=l(5);
z3=l(6);

l1=l(1);
l2=l(2);
l3=l(3);

Theta = (2*pi/(Const(3)))*kron(1*1e-2,Zs_In(:,1)*1e9); % l cm. cinsinden ve frekans GHz
cinsinden olduğu için 0.01 ve 1*10^9 ile çarpılıyor.

Zoutsd = Zs_In(:,2)+i.*Zs_In(:,3)

```

Zn=50;

% bu parametreler A parametreleridir.
for n = 1:10

```
E1=[((2-(i.*Zn)./(z1.*tan(Theta(n,1))))/2),-
      ((i.*Zn)./(z1.*tan(Theta(n,1))))/2;((i.*Zn)./(z1.*tan(Theta(n,1))))/2,((2+(i.*Zn)./(z
      1.*tan(Theta(n,1))))/2)]
E2=[(((cos(Theta(n,2))).*Zn)+(i.*z2.*sin(Theta(n,2)))+(i.*sin(Theta(n,2))*Zn^2)./z2)+(cos(
      Theta(n,2))*Zn)./(2*Zn),(((cos(Theta(n,2))).*Zn)-
      (i.*z2.*sin(Theta(n,2)))+(i.*sin(Theta(n,2))*Zn^2)./z2)-
      (cos(Theta(n,2))*Zn)./(2*Zn);(((cos(Theta(n,2))).*Zn)+(i.*z2.*sin(Theta(n,2)))-
      ((i.*sin(Theta(n,2))*Zn^2)./z2)-
      (cos(Theta(n,2))*Zn)./(2*Zn),(((cos(Theta(n,2))).*Zn)-(i.*z2.*sin(Theta(n,2)))-
      ((i.*sin(Theta(n,2))*Zn^2)./z2)+(cos(Theta(n,2))*Zn)./(2*Zn)]
E3=[((2-(i.*Zn)./(z3.*tan(Theta(n,3))))/2),-
      ((i.*Zn)./(z3.*tan(Theta(n,3))))/2;((i.*Zn)./(z3.*tan(Theta(n,3))))/2,((2+(i.*Zn)./(z
      3.*tan(Theta(n,3))))/2)]
```

M = E1*E2*E3;

```
S(1,1)=(M(2,1)./M(1,1));
S(1,2)=1./M(1,1);
S(2,1)=1./M(1,1);
S(2,2)=-(M(1,2)./M(1,1));
```

Ros=(Zs_const-Zn)/(Zs_const+Zn);

%Zlimc(n) IMC devresinin yukudur ve transistor kaynak empedansinin eslenigi olarak
alınmıştır

```
Zlimc(n) = conj(Zoutsd(n));
Rol(n)=((Zlimc(n)-Zn)/(Zlimc(n)+Zn));
Roout(n)=S(2,2)+(S(1,2).*S(2,1).*Ros)./(1-S(1,1).*Ros);
```

```
%Zoutime(n)=Zn*((1+Roout(n))./(1-Roout(n)));
%Rooutgen(n)=((Zoutime(n)-Zoutsd)./(Zoutime(n)-Zlimc(n)));
%GT(n)=1-(abs(Rooutgen(n)).^2)
```

```
GT(n)=(((1-(abs(Ros)).^2).*(1-(abs(Rol(n))).^2).*(abs(S(2,1)).^2))./(((abs(1-
      S(1,1).*Ros)).^2).*((abs(1-Roout(n)).*Rol(n)).^2)))
```

end

% Hata Islevi

```
toplamlam = (1-exp(-1*((1 - GT).^2)));
```

E = sum(toplamlam)

IMC_CIS_PS_L

```
function E = imc_cis(l)
```

```
% Program ilk Ilgar ALIYEV tarafından hazirlanmis, sonradan GT'ye gore iyilestirme icin
% CIS tarafından degistirilmis (CIS : Can Cevikbas, Ilkay Koseli, Serhat Altunc)
% Daha sonra A- parametreleri için Umut Hincal ve Yavuz Cengiz tarafından düzenlenmistir.
%
% Danisman : Prof.Dr. Filiz GÜNES
% Giriş bilgileri '.csv' uzantili dosyalardan okunur
```

```
global Zs_In Zoutsc Zinsc Zoutsd gmout gmin GT
Const = DLMREAD('Dosya_Const.csv',';',0,0,'A1..J1');
Zs_In = DLMREAD('Zs_Input.csv',';',0,0,'A1..E10');
```

```
% Kaynak ve Yük Empedansları
n=10;
```

```
Zs_const=50;
```

```
Zl_const=50;
```

```
E1 = zeros(2,2);
E2 = zeros(2,2);
E3 = zeros(2,2);
```

```
M = zeros(2,2);
```

```
Zoutsc=zeros(n,1);
Zinsc=zeros(n,1);
gmout=zeros(n,1);
gmin=zeros(n,1);
GT=zeros(n,1);
toplam=zeros(n,1);
```

```
% z, optimize edilecek hat parçalarının karakteristik empedansları
```

```
z2=l(3);
z3=l(4);
```

```
l2=l(1);
l3=l(2);
```

```
Theta = (2*pi/(Const(3)))*kron(l*1e-2,Zs_In(:,1)*1e9); % l cm. cinsinden ve frekans GHz
cinsinden olduğu için 0.01 ve 1*10^9 ile çarpılıyor.
```

```
Zoutsd = Zs_In(:,2)+i.*Zs_In(:,3);
```

```
Zn=50;
```

% bu parametreler A parametreleridir.

for n = 1:10

```
E2=[((2-(i.*Zn)./(z2.*tan(Theta(n,2))))/2),-
      ((i.*Zn)./(z2.*tan(Theta(n,2))))/2;((i.*Zn)./(z2.*tan(Theta(n,2))))/2,((2+(i.*Zn)./(z
      2.*tan(Theta(n,2))))/2)];
```

```
E3=[(((cos(Theta(n,3))).*Zn)+(i.*z3.*sin(Theta(n,3)))+(i.*sin(Theta(n,3))*Zn^2)./z3)+(cos(
      Theta(n,3))*Zn)./(2*Zn),(((cos(Theta(n,3))).*Zn)-
      (i.*z3.*sin(Theta(n,3)))+(i.*sin(Theta(n,3))*Zn^2)./z3)-
      (cos(Theta(n,3))*Zn)./(2*Zn);(((cos(Theta(n,3))).*Zn)+(i.*z3.*sin(Theta(n,3)))-
      ((i.*sin(Theta(n,3))*Zn^2)./z3)-
      (cos(Theta(n,3))*Zn)./(2*Zn),(((cos(Theta(n,3))).*Zn)-(i.*z3.*sin(Theta(n,3)))-
      ((i.*sin(Theta(n,3))*Zn^2)./z3)+(cos(Theta(n,3))*Zn)./(2*Zn)];
```

```
M = E2*E3;
```

```
S(1,1)=(M(2,1)./M(1,1));
```

```
S(1,2)=1./M(1,1);
```

```
S(2,1)=1./M(1,1);
```

```
S(2,2)=-(M(1,2)./M(1,1));
```

```
Ros=(Zs_const-Zn)/(Zs_const+Zn);
```

%Zlimc(n) IMC devresinin yukudur ve transistor kaynak empedansinin eslenigi olarak
alinmistir

```
Zlimc(n) = conj(Zoutsd(n));
```

```
Rol(n)=((Zlimc(n)-Zn)/(Zlimc(n)+Zn));
```

```
Roout(n)=S(2,2)+(S(1,2).*S(2,1).*Ros)/(1-S(1,1).*Ros);
```

```
%Zoutimc(n)=Zn*((1+Roout(n))./(1-Roout(n)));
```

```
%Rooutgen(n)=((Zoutimc(n)-Zoutsd)./(Zoutimc(n)-Zlimc(n)));
```

```
%GT(n)=1-(abs(Rooutgen(n)).^2)
```

```
GT(n)=(((1-(abs(Ros)).^2).*(1-(abs(Rol(n))).^2).*(abs(S(2,1)).^2))./(((abs(1-
      S(1,1).*Ros)).^2).*((abs(1-Roout(n)).*Rol(n))).^2)))
```

```
end
```

% Hata Islevi

```
toplamlam = (1-exp(-1*((1 - GT).^2)));
```

```
E = sum(toplamlam)
```

IMC_CIS_SP_L

```
function E = imc_cis(l)
```

```
% Program ilk Ilgar ALIYEV tarafından hazırlanmış, sonradan GT'ye göre iyileştirme için
% CIS tarafından değiştirilmiş (CIS : Can Cevikbas, Ilkay Koseli, Serhat Altunc)
% Daha sonra A- parametreleri için Umut Hincal ve Yavuz Cengiz tarafından düzenlenmiştir.
%
% Danışman : Prof.Dr. Filiz GÜNES
% Giriş bilgileri '.csv' uzantılı dosyalardan okunur
```

```
global Zs_In Zoutsc Zinsc Zoutsd gmout gmin GT
Const = DLMREAD('Dosya_Const.csv',';',0,0,'A1..J1');
Zs_In = DLMREAD('Zs_Input.csv',';',0,0,'A1..E10');
```

```
% Kaynak ve Yük Empedansları
n=10;
```

```
Zs_const=50;
Zl_const=50;
```

```
E1 = zeros(2,2);
E2 = zeros(2,2);
```

```
M = zeros(2,2);
```

```
Zoutsc=zeros(n,1);
Zinsc=zeros(n,1);
gmout=zeros(n,1);
gmin=zeros(n,1);
GT=zeros(n,1);
toplam=zeros(n,1);
```

```
% z, optimize edilecek hat parçalarının karakteristik empedansları
```

```
z1=l(3);
z2=l(4);
```

```
l1=l(1);
l2=l(2);
```

```
Theta = (2*pi/(Const(3)))*kron(1*1e-2,Zs_In(:,1)*1e9); % l cm. cinsinden ve frekans GHz
cinsinden olduğu için 0.01 ve 1*10^9 ile çarpılıyor.
```

```
Zoutsd = Zs_In(:,2)+i.*Zs_In(:,3)
```

```
Zn=50;
% bu parametreler A parametreleridir.
for n = 1:10
```

```

E1=[(((cos(Theta(n,1))).*Zn)+(i.*z1.*sin(Theta(n,1)))+(i.*sin(Theta(n,1))*Zn^2)./z1)+(cos(
    Theta(n,1))*Zn)./(2*Zn),(((cos(Theta(n,1))).*Zn)-
    (i.*z1.*sin(Theta(n,1)))+(i.*sin(Theta(n,1))*Zn^2)./z1)-
    (cos(Theta(n,1))*Zn)./(2*Zn);(((cos(Theta(n,1))).*Zn)+(i.*z1.*sin(Theta(n,1)))-
    (i.*sin(Theta(n,1))*Zn^2)./z1)-
    (cos(Theta(n,1))*Zn)./(2*Zn),(((cos(Theta(n,1))).*Zn)-(i.*z1.*sin(Theta(n,1)))-
    (i.*sin(Theta(n,1))*Zn^2)./z1)+(cos(Theta(n,1))*Zn)./(2*Zn)]
E2=[((2-(i.*Zn)./(z2.*tan(Theta(n,2))))/2),-
    ((i.*Zn)./(z2.*tan(Theta(n,2))))/2;((i.*Zn)./(z2.*tan(Theta(n,2))))/2,((2+(i.*Zn)./(z
    2.*tan(Theta(n,2))))/2)]

```

```

M = E1*E2;

```

```

S(1,1)=(M(2,1)./M(1,1));
S(1,2)=1./M(1,1);
S(2,1)=1./M(1,1);
S(2,2)=-(M(1,2)./M(1,1));

```

```

Ros=(Zs_const-Zn)/(Zs_const+Zn);

```

```

%Zlimc(n) IMC devresinin yukudur ve transistor kaynak empedansinin eslenigi olarak
    alinmistir

```

```

Zlimc(n) = conj(Zoutsd(n));
Rol(n)=((Zlimc(n)-Zn)/(Zlimc(n)+Zn));
Roout(n)=S(2,2)+(S(1,2).*S(2,1).*Ros)./(1-S(1,1).*Ros);

```

```

%Zoutimc(n)=Zn*((1+Roout(n))./(1-Roout(n)));
%Rooutgen(n)=((Zoutimc(n)-Zoutsd)./(Zoutimc(n)-Zlimc(n)));

```

```

%GT(n)=1-(abs(Rooutgen(n)).^2)

```

```

GT(n)=(((1-(abs(Ros)).^2).*(1-(abs(Rol(n))).^2).*(abs(S(2,1)).^2))./(((abs(1-
    S(1,1).*Ros)).^2).*((abs(1-Roout(n)).*Rol(n))).^2)))

```

```

end

```

```

% Hata Islevi

```

```

toplamlam = (1-exp(-1*((1 - GT).^2)));

```

```

E = sum(toplamlam)

```

OMC_CIS

```
function E = omc_cis(l)
```

```
% Program ilk Ilgar ALIYEV tarafından hazırlanmış, sonradan GT'ye göre iyileştirme için
% CIS tarafından değiştirilmiş (CIS : Can Cevikbas, Ilkay Koseli, Serhat Altunc)
% Daha sonra A- parametreleri için Umut Hincal ve Yavuz Cengiz tarafından düzenlenmiştir.
%
% Danışman : Prof.Dr. Filiz GÜNES
% Giriş bilgileri '.csv' uzantılı dosyalardan okunur
```

```
global Zl_In Zinlc Zoutlc gmin gmout GT
Const = DLMREAD('Dosya_Const.csv',';',0,0,'A1..J1');
Zl_In = DLMREAD('Zl_Input.csv',';',0,0,'A1..E10');
```

```
% Kaynak ve Yük Empedansları
n=10;
```

```
Zs_const=50;
Zl_const=50;
```

```
E1 = zeros(2,2);
E2 = zeros(2,2);
E3 = zeros(2,2);
```

```
M = zeros(2,2);
```

```
Zinlc=zeros(n,1);
Zoutlc=zeros(n,1);
gmin=zeros(n,1);
gmout=zeros(n,1);
GT=zeros(n,1);
toplaml=zeros(n,1);
```

```
% z, optimize edilecek hat parçalarının karakteristik empedansları
```

```
z1=l(4);
z2=l(5);
z3=l(6);
```

```
l1=l(1);
l2=l(2);
l3=l(3);
```

```
Theta=(2*pi/(Const(3)))*kron(l*1e-2,Zl_In(:,1)*1e9); % l cm. cinsinden ve frekans GHz
cinsinden olduğu için 0.01 ve 1*10^9 ile çarpılıyor.
```

```
Zinld=Zl_In(:,2)+i.*Zl_In(:,3);
```

```
Zn=50;
```

for n = 1:10

```
E1=[(((cos(Theta(n,1))).*Zn)+(i.*z1.*sin(Theta(n,1)))+(i.*sin(Theta(n,1))*Zn^2)./z1)+(cos(
    Theta(n,1)).*Zn)./(2*Zn),(((cos(Theta(n,1))).*Zn)-
    (i.*z1.*sin(Theta(n,1)))+(i.*sin(Theta(n,1))*Zn^2)./z1)-
    (cos(Theta(n,1)).*Zn)./(2*Zn);(((cos(Theta(n,1))).*Zn)+(i.*z1.*sin(Theta(n,1)))-
    (i.*sin(Theta(n,1))*Zn^2)./z1)-
    (cos(Theta(n,1)).*Zn)./(2*Zn),(((cos(Theta(n,1))).*Zn)-(i.*z1.*sin(Theta(n,1)))-
    (i.*sin(Theta(n,1))*Zn^2)./z1)+(cos(Theta(n,1)).*Zn)./(2*Zn)]
```

```
E2=[((2-(i.*Zn)./(z2.*tan(Theta(n,2))))/2),-
    ((i.*Zn)./(z2.*tan(Theta(n,2))))/2;((i.*Zn)./(z2.*tan(Theta(n,2))))/2,((2+(i.*Zn)./(z
    2.*tan(Theta(n,2))))/2)]
```

```
E3=[(((cos(Theta(n,3))).*Zn)+(i.*z3.*sin(Theta(n,3)))+(i.*sin(Theta(n,3))*Zn^2)./z3)+(cos(
    Theta(n,3)).*Zn)./(2*Zn),(((cos(Theta(n,3))).*Zn)-
    (i.*z3.*sin(Theta(n,3)))+(i.*sin(Theta(n,3))*Zn^2)./z3)-
    (cos(Theta(n,3)).*Zn)./(2*Zn);(((cos(Theta(n,3))).*Zn)+(i.*z3.*sin(Theta(n,3)))-
    (i.*sin(Theta(n,3))*Zn^2)./z3)-
    (cos(Theta(n,3)).*Zn)./(2*Zn),(((cos(Theta(n,3))).*Zn)-(i.*z3.*sin(Theta(n,3)))-
    (i.*sin(Theta(n,3))*Zn^2)./z3)+(cos(Theta(n,3)).*Zn)./(2*Zn)]
```

M=E1*E2*E3;

S(1,1)=(M(2,1)/M(1,1));

S(1,2)=1./M(1,1);

S(2,1)=1./M(1,1);

S(2,2)=-(M(1,2)/M(1,1));

Zinomc(n)=conj(Zinld(n));

Ros(n)=(Zinomc(n)-Zn)/(Zinomc(n)+Zn)

Rol=((Zl_const-Zn)/(Zl_const+Zn))

Roout(n)=S(2,2)+(S(1,2).*S(2,1).*Ros(n))./(1-S(1,1).*Ros(n));

Roin=S(1,1);

GT(n)=(((1-(abs(Ros(n))).^2).*(1-(abs(Rol))^2).*(abs(S(2,1)).^2))./(((abs(1-
 S(2,2).*Rol)).^2).*((abs(1-Roin.*Ros(n))).^2)))

end

% Hata Islevi

toplamlam = (1-exp(-2*((1 - GT).^2)));

E=sum(toplamlam)

OMC_CISP

```
function E = omc_cis(l)
```

```
% Program ilk Ilgar ALIYEV tarafından hazirlanmis, sonradan GT'ye gore iyilestirme icin
% CIS tarafından degistirilmis (CIS : Can Cevikbas, Ilkay Koseli, Serhat Altunc)
% Daha sonra A- parametreleri için Umut Hincal ve Yavuz Cengiz tarafından düzenlenmistir.
%
% Danisman : Prof.Dr. Filiz GÜNES
% Giriş bilgileri '.csv' uzantili dosyalardan okunur
```

```
global Zl_In Zinlc Zoutlc gmin gmout GT
Const = DLMREAD('Dosya_Const.csv',';',0,0,'A1..J1');
Zl_In = DLMREAD('Zl_Input.csv',';',0,0,'A1..E10');
```

```
% Kaynak ve Yük Empedanslari
n=10;
```

```
Zs_const=50;
Zl_const=50;
```

```
E1 = zeros(2,2);
E2 = zeros(2,2);
E3 = zeros(2,2);
```

```
M = zeros(2,2);
```

```
Zinlc=zeros(n,1);
Zoutlc=zeros(n,1);
gmin=zeros(n,1);
gmout=zeros(n,1);
GT=zeros(n,1);
toplam=zeros(n,1);
```

```
% z, optimize edilecek hat parçalarının karakteristik empedansları
```

```
z1=l(4);
z2=l(5);
z3=l(6);
```

```
l1=l(1);
l2=l(2);
l3=l(3);
```

```
Theta=(2*pi/(Const(3)))*kron(l*1e-2,Zl_In(:,1)*1e9); % l cm. cinsinden ve frekans GHz
cinsinden olduğu için 0.01 ve 1*10^9 ile çarpılıyor.
```

```
Zinld=Zl_In(:,2)+i.*Zl_In(:,3);
```

Zn=50;

for n = 1:10

```
E1=[((2-(i.*Zn)./(z1.*tan(Theta(n,1))))/2),-
      ((i.*Zn)./(z1.*tan(Theta(n,1))))/2;((i.*Zn)./(z1.*tan(Theta(n,1))))/2,((2+(i.*Zn)./(z
      1.*tan(Theta(n,1))))/2)];
E2=[(((cos(Theta(n,2))).*Zn)+(i.*z2.*sin(Theta(n,2)))+(i.*sin(Theta(n,2))*Zn^2)./z2)+(cos(
      Theta(n,2))).*Zn)./(2*Zn),(((cos(Theta(n,2))).*Zn)-
      (i.*z2.*sin(Theta(n,2)))+(i.*sin(Theta(n,2))*Zn^2)./z2)-
      (cos(Theta(n,2))).*Zn)./(2*Zn);(((cos(Theta(n,2))).*Zn)+(i.*z2.*sin(Theta(n,2)))-
      ((i.*sin(Theta(n,2))*Zn^2)./z2)-
      (cos(Theta(n,2))).*Zn)./(2*Zn),(((cos(Theta(n,2))).*Zn)-(i.*z2.*sin(Theta(n,2)))-
      ((i.*sin(Theta(n,2))*Zn^2)./z2)+(cos(Theta(n,2))).*Zn)./(2*Zn)];
E3=[((2-(i.*Zn)./(z3.*tan(Theta(n,3))))/2),-
      ((i.*Zn)./(z3.*tan(Theta(n,3))))/2;((i.*Zn)./(z3.*tan(Theta(n,3))))/2,((2+(i.*Zn)./(z
      3.*tan(Theta(n,3))))/2)];
```

M=E1*E2*E3;

```
S(1,1)=(M(2,1)./M(1,1));
S(1,2)=1./M(1,1);
S(2,1)=1./M(1,1);
S(2,2)=-(M(1,2)./M(1,1));
```

```
Zinomc(n)=conj(Zinld(n));
Ros(n)=(Zinomc(n)-Zn)/(Zinomc(n)+Zn);
Rol=((Zl_const-Zn)/(Zl_const+Zn));
Roout(n)=S(2,2)+(S(1,2).*S(2,1).*Ros(n))./(1-S(1,1).*Ros(n));
Roin=S(1,1);
```

```
GT(n)=(((1-(abs(Ros(n))).^2).*(1-(abs(Rol))^2).*(abs(S(2,1)).^2))./(((abs(1-
      S(2,2).*Rol)).^2).*((abs(1-Roin.*Ros(n))).^2)))
```

end

% Hata Islevi

```
toplam = (1-exp(-2*((1 - GT).^2)));
```

E=sum(toplam)

OMC_CIS_PS_L

```
function E = omc_cis(l)
```

```
% Program ilk Ilgar ALIYEV tarafından hazırlanmış, sonradan GT'ye göre iyileştirme için
% CIS tarafından değiştirilmiş (CIS : Can Cevikbas, Ilkay Koseli, Serhat Altunc)
% Daha sonra A- parametreleri için Umut Hincal ve Yavuz Cengiz tarafından düzenlenmiştir.
%
% Danışman : Prof.Dr. Filiz GÜNES
% Giriş bilgileri '.csv' uzantılı dosyalardan okunur
```

```
global Zl_In Zinlc Zoutlc gmin gmout GT
Const = DLMREAD('Dosya_Const.csv',';',0,0,'A1..J1');
Zl_In = DLMREAD('Zl_Input.csv',';',0,0,'A1..E10');
```

```
% Kaynak ve Yük Empedansları
n=10;
```

```
Zs_const=50;
Zl_const=50;
```

```
E1 = zeros(2,2);
E2 = zeros(2,2);
E3 = zeros(2,2);
```

```
M = zeros(2,2);
```

```
Zinlc=zeros(n,1);
Zoutlc=zeros(n,1);
gmin=zeros(n,1);
gmout=zeros(n,1);
GT=zeros(n,1);
toplaml=zeros(n,1);
```

```
% z, optimize edilecek hat parçalarının karakteristik empedansları
```

```
z2=l(3);
z3=l(4);
```

```
l2=l(1);
l3=l(2);
```

```
Theta=(2*pi/(Const(3)))*kron(l*1e-2,Zl_In(:,1)*1e9); % l cm. cinsinden ve frekans GHz
cinsinden olduğu için 0.01 ve 1*10^9 ile çarpılıyor.
```

```
Zinld=Zl_In(:,2)+i.*Zl_In(:,3);
```

```
Zn=50;
```

for n = 1:10

```
E2=[((2-(i.*Zn)./(z2.*tan(Theta(n,2))))/2),-
      ((i.*Zn)./(z2.*tan(Theta(n,2))))/2;((i.*Zn)./(z2.*tan(Theta(n,2))))/2,((2+(i.*Zn)./(z
      2.*tan(Theta(n,2))))/2)]
E3=[(((cos(Theta(n,3))).*Zn)+(i.*z3.*sin(Theta(n,3)))+(i.*sin(Theta(n,3))*Zn^2)./z3)+(cos(
      Theta(n,3))*Zn)./(2*Zn),(((cos(Theta(n,3))).*Zn)-
      (i.*z3.*sin(Theta(n,3)))+(i.*sin(Theta(n,3))*Zn^2)./z3)-
      (cos(Theta(n,3))*Zn)./(2*Zn);(((cos(Theta(n,3))).*Zn)+(i.*z3.*sin(Theta(n,3)))-
      ((i.*sin(Theta(n,3))*Zn^2)./z3)-
      (cos(Theta(n,3))*Zn)./(2*Zn),(((cos(Theta(n,3))).*Zn)-(i.*z3.*sin(Theta(n,3)))-
      ((i.*sin(Theta(n,3))*Zn^2)./z3)+(cos(Theta(n,3))*Zn)./(2*Zn)]
```

M=E2*E3;

```
S(1,1)=(M(2,1)./M(1,1));
S(1,2)=1./M(1,1);
S(2,1)=1./M(1,1);
S(2,2)=-(M(1,2)./M(1,1));
```

```
Zinomc(n)=conj(Zinld(n));
Ros(n)=(Zinomc(n)-Zn)/(Zinomc(n)+Zn)
Rol=((Zl_const-Zn)/(Zl_const+Zn))
Roout(n)=S(2,2)+(S(1,2).*S(2,1).*Ros(n))./(1-S(1,1).*Ros(n));
Roin=S(1,1);
```

```
GT(n)=(((1-(abs(Ros(n))).^2).*(1-(abs(Rol))^2).*(abs(S(2,1)).^2))./(((abs(1-
      S(2,2).*Rol)).^2).*((abs(1-Roin.*Ros(n))).^2)))
```

end

% Hata Islevi

```
toplamlam = (1-exp(-2*((1 - GT).^2)));
```

```
E=sum(toplam)
```

OMC_CIS_SP_L

```
function E = omc_cis(l)
```

```
% Program ilk Ilgar ALIYEV tarafından hazirlanmis, sonradan GT'ye gore iyilestirme icin
% CIS tarafından degistirilmis (CIS : Can Cevikbas, Ilkay Koseli, Serhat Altunc)
% Daha sonra A- parametreleri için Umut Hincal ve Yavuz Cengiz tarafından düzenlenmistir.
%
% Danisman : Prof.Dr. Filiz GÜNES
%
```

```
% Giriş bilgileri '.csv' uzantili dosyalardan okunur
```

```
global Zl_In Zinlc Zoutlc gmin gmout GT
Const = DLMREAD('Dosya_Const.csv',';',0,0,'A1..J1');
Zl_In = DLMREAD('Zl_Input.csv',';',0,0,'A1..E10');
```

```
% Kaynak ve Yük Empedanslari
n=10;
```

```
Zs_const=50;
```

```
Zl_const=50;
```

```
E1 = zeros(2,2);
E2 = zeros(2,2);
E3 = zeros(2,2);
```

```
M = zeros(2,2);
```

```
Zinlc=zeros(n,1);
Zoutlc=zeros(n,1);
gmin=zeros(n,1);
gmout=zeros(n,1);
GT=zeros(n,1);
toplam=zeros(n,1);
```

```
% z, optimize edilecek hat parçalarının karakteristik empedansları
```

```
z1=l(3);
z2=l(4);
```

```
l1=l(1);
l2=l(2);
```

```
Theta=(2*pi/(Const(3)))*kron(l*1e-2,Zl_In(:,1)*1e9); % l cm. cinsinden ve frekans GHz
cinsinden olduğu için 0.01 ve 1*10^9 ile çarpılıyor.
```

```
Zinld=Zl_In(:,2)+i.*Zl_In(:,3);
```

Zn=50;

for n = 1:10

```
E1=[(((cos(Theta(n,1))).*Zn)+(i.*z1.*sin(Theta(n,1)))+(i.*sin(Theta(n,1))*Zn^2)./z1)+(cos(
    Theta(n,1))*Zn)./(2*Zn),(((cos(Theta(n,1))).*Zn)-
    (i.*z1.*sin(Theta(n,1)))+(i.*sin(Theta(n,1))*Zn^2)./z1)-
    (cos(Theta(n,1))*Zn)./(2*Zn);(((cos(Theta(n,1))).*Zn)+(i.*z1.*sin(Theta(n,1)))-
    (i.*sin(Theta(n,1))*Zn^2)./z1)-
    (cos(Theta(n,1))*Zn)./(2*Zn),(((cos(Theta(n,1))).*Zn)-(i.*z1.*sin(Theta(n,1)))-
    (i.*sin(Theta(n,1))*Zn^2)./z1)+(cos(Theta(n,1))*Zn)./(2*Zn)]
```

```
E2=[((2-(i.*Zn)./(z2.*tan(Theta(n,2))))/2),-
    ((i.*Zn)./(z2.*tan(Theta(n,2))))/2;((i.*Zn)./(z2.*tan(Theta(n,2))))/2,((2+(i.*Zn)./(z
    2.*tan(Theta(n,2))))/2)]
```

M=E1*E2;

S(1,1)=(M(2,1)./M(1,1));

S(1,2)=1./M(1,1);

S(2,1)=1./M(1,1);

S(2,2)=-(M(1,2)./M(1,1));

Zinomc(n)=conj(Zinld(n));

Ros(n)=(Zinomc(n)-Zn)/(Zinomc(n)+Zn)

Rol=((Zl_const-Zn)/(Zl_const+Zn))

Roout(n)=S(2,2)+(S(1,2).*S(2,1).*Ros(n))./(1-S(1,1).*Ros(n));

Roin=S(1,1);

GT(n)=(((1-(abs(Ros(n))).^2).*(1-(abs(Rol))^2).*(abs(S(2,1)).^2))./(((abs(1-
 S(2,2).*Rol)).^2).*((abs(1-Roin.*Ros(n))).^2)))

end

% Hata Islevi

toplam = (1-exp(-2*((1 - GT).^2)));

E=sum(toplam)

ZI_Input.csv yardımcı data bilgileri (NE329S01)

Çizelge 10. 1 NE329S01 transistörü için ZI_Input.csv yardımcı data bilgileri

2	16,87	-132,83	1,00	1,00
3	14,12	-73,29	1,00	1,00
4	30,97	-97,55	1,00	1,00
5	90,71	-148,77	1,00	1,00
6	124,07	-139,91	1,00	1,00
7	199,05	-84,99	1,00	1,00
8	187,32	-16,18	1,00	1,00
9	151,03	12,81	1,00	1,00
10	103,88	-14,06	1,00	1,00
11	82,37	-12,13	1,00	1,00

Zs_Input.csv yardımcı data bilgileri (NE329S01)

Çizelge 10. 2 NE329S01 transistörü için Zs_Input.csv yardımcı data bilgileri

2	93,99	180,61	1,00	1,00
3	63,52	142,86	1,00	1,00
4	51,04	97,29	1,00	1,00
5	42,91	67,01	1,00	1,00
6	37,57	54,02	1,00	1,00
7	36,57	36,59	1,00	1,00
8	33,57	26,58	1,00	1,00
9	29,43	15,87	1,00	1,00
10	26,92	18,203	1,00	1,00
11	24,89	19,480	1,00	1,00

Ek 2 Mikrodalga Kuvvetlendiricisi Toplam Devre Hesap Programları

Bu bölümde verilen programlar optimizasyon işleminden geçirilmiş giriş ve çıkış uydurma devrelerine ait uzunluk ve karakteristik empedansları giriş değeri olarak alıp giriş uydurma devresi – transistor-çıkış uydurma devresi üçlüsünden oluşan toplam devrenin Kazanç, giriş VSWR ve gürültü değerlerini hesaplamada kullanılmıştır.

Bilgin31_NE329S01C2

```
function F=tez_NE329S01
```

```
%
```

```
%Program İlgar ALİYEV tarafından hazırlanmıştır
```

```
%Tez Danışmanı Prof. Dr. Filiz GÜNEŞ
```

```
%Tranzistör için Vce=10 V; Ic=20 mA olarak alınmıştır.
```

```
%Tüm Değerler Prof.Dr. Filiz GÜNEŞ'in Çalışmalarından Derlenmiştir.
```

```
% Giriş Bilgilerinin "bin" klasörü altındaki csv uzantılı dosyalardan okunması
```

```
Const = DLMREAD('Dosya_Const.csv',';',0,0,'A1..J1');
```

```
In = DLMREAD('Dosya_Input.csv',';',0,0,'A1..N12');
```

```
% Kaynak ve Yük Empedansları
```

```
Zs_const=50;
```

```
Rs_const=real(Zs_const);
```

```
Zl_const=50;
```

```
Rl_const=real(Zl_const);
```

```
% Gürültü Faktörü Referans Değeri
```

```
Fr=10^(Const(4)*0.1)
```

```
% z optimize edilecek hat parçalarının karakteristik empedansları
```

```
n=12;
```

```
z1=l(7);
```

```
z2=l(8);
```

```
z3=l(9);
```

```
z4=l(10);
```

```
z5=l(11);
```

```
z6=l(12);
```

```
l1=l(1);
```

```
l2=l(2);
```

```
l3=l(3);
```

```
l4=l(4);
```

```
l5=l(5);
```

l6=l(6);

% Referans Kazanç Değeri

Gref=10^(Const(10)*0.1);

% İlk Değerlerin Tanımlanması

n=12;

Gt=zeros(n,1);

Gav=zeros(n,1);

E1=zeros(2,2);

E2=zeros(2,2);

E3=zeros(2,2);

E4=zeros(2,2);

E5=zeros(2,2);

E6=zeros(2,2);

MRin=zeros(n,1)

EN=zeros(n,1);

GT=zeros(n,1);

LT=zeros(n,1);

Noise=zeros(n,1);

Zs=ones(n,7);

Zopt=zeros(n,1);

Zout=zeros(n,1);

Rout=zeros(n,1);

Zm=zeros(n,7);

Zi7=zeros(n,1);

A=zeros(n,1);

B=zeros(n,1);

C=zeros(n,1);

D=zeros(n,1);

AA=zeros(n,1);

BB=zeros(n,1);

CC=zeros(n,1);

DD=zeros(n,1);

EE=zeros(n,1);

FF=zeros(n,1);

GG=zeros(n,1);

HH=zeros(n,1);

```
roi=zeros(n,1);
```

```
EVi=zeros(n,1);
```

```
toplaml=zeros(n,1);
```

```
% S Parametrelerinin hesaplanması
```

```
S = MATCOMP(In(:,[3:2:9,12]),In(:,[4:2:10,13]));
```

```
G_opt=S(:,5);
```

```
S = S(:,1:4);
```

```
% ABCD Parametrelerinin Hesabı
```

```
A = ((1+S(:,1)).*(1-S(:,4))+S(:,2).*S(:,3))./(2.*S(:,2)));
```

```
B = 50.*((1+S(:,1)).*(1+S(:,4))-S(:,2).*S(:,3))./(2.*S(:,2)));
```

```
C = (1/50).*((1-S(:,1)).*(1-S(:,4))-S(:,2).*S(:,3))./(2.*S(:,2)));
```

```
D = ((1-S(:,1)).*(1+S(:,4))+S(:,2).*S(:,3))./(2.*S(:,2)));
```

```
% Transistörün Eşdeğer Gürültü Direnci
```

```
Rn =In(:,14).*Const(2);
```

```
% Minimum Gürültü Faktörü
```

```
Fmin =10.^(In(:,11).*0.1);
```

```
% Thetaların Hesabı
```

```
Theta=(2*pi/(Const(3)))*kron(1*1e-2,In(:,2)*1e9); % 1 cm. cinsinden ve frekans GHz  
cinsinden olduğu için 0.01 ve 1*10^9 ile çarpılıyor.
```

```
% Kazancın Hesaplanması
```

```
for n=1:12
```

```
E1=[cos(Theta(n,1)),i.*z1.*sin(Theta(n,1));i.*sin(Theta(n,1))./z1,cos(Theta(n,1))];
```

```
E3=[cos(Theta(n,3)),i.*z3.*sin(Theta(n,3));i.*sin(Theta(n,3))./z3,cos(Theta(n,3))];
```

```
E4=[cos(Theta(n,4)),i.*z4.*sin(Theta(n,4));i.*sin(Theta(n,4))./z4,cos(Theta(n,4))];
```

```
E6=[cos(Theta(n,6)),i.*z6.*sin(Theta(n,6));i.*sin(Theta(n,6))./z6,cos(Theta(n,6))];
```

```
E2=[1,0;-i./(z2.*tan(Theta(n,2))),1];
```

```
E5=[1,0;-i./(z2.*tan(Theta(n,5))),1];
```

```
M=E1*E2*E3*[A(n),B(n);C(n),D(n)]*E4*E5*E6;
```

```
Gt(n)=4*Rs_const*RI_const/(  
abs(M(1,1).*Zl_const+M(1,2)+Zs_const*(M(2,1).*Zl_const+M(2,2))) ).^2)
```

```
Zin(n)=(M(1,1).*Zl_const+M(1,2))./(M(2,1).*Zl_const+M(2,2))
```

```
MRin(n)=abs((Zs_const-conj(Zin(n)))/(Zs_const+Zin(n)))
```

```

Zout=(M(1,2)+M(2,2).*Zs_const)/(M(1,1)+M(2,1).*Zs_const)
Rout=real(Zout);

Gav(n)=4*Rs_const*Rout/(
    abs(M(1,1).*conj(Zout)+M(1,2)+Zs_const*(M(2,1).*conj(Zout)+M(2,2))).^2);

end
ortmr=MRin(2)+MRin(3)+MRin(4)+MRin(5)+MRin(6)+MRin(7)+MRin(8)+MRin(9)+MRin
    (10)+MRin(11);
ortmrin=ortmr/10
ortgt=Gt(1)+Gt(2)+Gt(3)+Gt(4)+Gt(5)+Gt(6)+Gt(7)+Gt(8)+Gt(9)+Gt(10);
ortgt=ortgt/10

% İlk Katın Kaynak Empedansı

Zs(:,1)=Zs_const*ones(n,1);

% İkinci Katın Kaynak Empedansı

Zs(:,2)=(i.*z1.*sin(Theta(:,1))+cos(Theta(:,1)).*Zs(:,1))./(cos(Theta(:,1))+i.*sin(Theta(:,1)).*
    Zs(:,1)./z1);

% Üçüncü Katın Kaynak Empedansı

Zs(:,3)=Zs(:,2)./(1+(-i./z2./(tan(Theta(:,2)))).*Zs(:,2));

% Dördüncü Katın Kaynak Empedansı

Zs(:,4)=(i.*z3.*sin(Theta(:,3))+cos(Theta(:,3)).*Zs(:,3))./(cos(Theta(:,3))+i.*sin(Theta(:,3)).*
    Zs(:,3)./z3);

% Transistör İçin Optimum Kaynak Empedansı

Zopt=50*(1+G_opt)/(1-G_opt);
Rs4=real(Zs(:,4));
Ropt=real(Zopt);

% Giriş Uydurma Devresi Pasif Ve Kayıpsız Olduğundaki Transduser Güç Kazancı

GT=4*Rs4.*Ropt./(abs(Zs(:,4))+conj(Zopt)).^2;
LT=1./GT;

Noise=(4*Rn.*Ropt.*(1./GT-1))./((abs(Zopt)).^2)+Fmin

% Gürültü Hata İfadesi

EN=1.2*ones(n,1)-Noise;
%EN=1+(Fr*ones(12,1)-Fmin).*((abs(Zopt)).^2)./(4*Rn.*Ropt)-(1./GT)

% İlk Katın Kaynak Empedansı

```

```

Zm(:,1)=-conj(Zs_const)*ones(12,1);
Zs(:,1)=Zs_const*ones(12,1);
AA=abs(
    (cos(Theta(:,1))+i.*sin(Theta(:,1)).*Zm(:,1)./z1)./(cos(Theta(:,1))+i.*sin(Theta(:,1)).*Zs(:,1)./z1));

% İkinci Katın Kaynak Empedansı

Zm(:,2)=(i.*z1.*sin(Theta(:,1))-cos(Theta(:,1)).*conj(Zs(:,1)))./(cos(Theta(:,1))-i.*sin(Theta(:,1)).*conj(Zs(:,1))./z1);
Zs(:,2)=(i.*z1.*sin(Theta(:,1))+cos(Theta(:,1)).*Zs(:,1))./(cos(Theta(:,1))+i.*sin(Theta(:,1)).*Zs(:,1)./z1);
BB=abs((1+(-i.*Zm(:,2))./z2./tan(Theta(:,2))) )./(1+(-i.*Zs(:,2))./z2./tan(Theta(:,2)))));

% Üçüncü Katın Kaynak Empedansı

Zm(:,3)=-conj(Zs(:,2))./(1-(-i.*conj(Zs(:,2))./z2./tan(Theta(:,2)))));
Zs(:,3)=(Zs(:,2))./(1+(-i.*Zs(:,2))./z2./tan(Theta(:,2)))));
CC=abs((cos(Theta(:,3))+i.*sin(Theta(:,3)).*Zm(:,3))./z3)./(cos(Theta(:,3))+i.*sin(Theta(:,3)).*Zs(:,3))./z3));

% Dördüncü Katın Kaynak Empedansı

Zm(:,4)=(i.*z3.*sin(Theta(:,3))-cos(Theta(:,3)).*conj(Zs(:,3)))./(cos(Theta(:,3))-i.*sin(Theta(:,3)).*conj(Zs(:,3))./z3);
Zs(:,4)=(i.*z3.*sin(Theta(:,3))+cos(Theta(:,3)).*Zs(:,3))./(cos(Theta(:,3))+i.*sin(Theta(:,3)).*Zs(:,3))./z3);
DD=abs((A+C.*Zm(:,4))./(A-C.*Zs(:,4)));

% Beşinci Katın Kaynak Empedansı

Zm(:,5)=(B-D.*conj(Zs(:,4)))./(A-C.*conj(Zs(:,4)));
Zs(:,5)=(B+D.*Zs(:,4))./(A-C.*Zs(:,4));
EE=abs((cos(Theta(:,4))+i.*sin(Theta(:,4)).*Zm(:,5))./z4)./(cos(Theta(:,4))+i.*sin(Theta(:,4)).*Zs(:,5))./z4));

% Altıncı Katın Kaynak Empedansı

Zm(:,6)=(i.*z4.*sin(Theta(:,4))-cos(Theta(:,4)).*conj(Zs(:,5)))./(cos(Theta(:,4))-i.*sin(Theta(:,4)).*conj(Zs(:,5))./z4);
Zs(:,6)=(i.*z4.*sin(Theta(:,4))+cos(Theta(:,4)).*Zs(:,5))./(cos(Theta(:,4))+i.*sin(Theta(:,4)).*Zs(:,5))./z4);
FF=abs((1+(-i.*Zm(:,6))./z5./tan(Theta(:,5))) )./(1+(-i.*Zs(:,6))./z5./tan(Theta(:,5)))));

% Yedinci Katın Kaynak Empedansı

Zm(:,7)=(-conj(Zs(:,6)))./(1-(-i.*conj(Zs(:,6))./z5./tan(Theta(:,5)))));
Zs(:,7)=(Zs(:,6))./(1+(-i.*Zs(:,6))./z5./tan(Theta(:,5)))));
GG=abs((cos(Theta(:,6))+i.*sin(Theta(:,6)).*Zm(:,7))./z6)./(cos(Theta(:,6))+i.*sin(Theta(:,6)).*Zs(:,7))./z6));

```

% Yedinci Katın Giriş Empedansı

```
Zi7=z6.*(Zl_const+i.*z6.*tan(Theta(:,6)))/(z6+i.*Zl_const*tan(Theta(:,6)));
HH=abs((Zi7+Zm(:,7))/(Zi7+Zs(:,7)));
```

```
roi=AA.*BB.*CC.*DD.*EE.*FF.*GG.*HH;
```

```
roir=(Const(5)-1)/(Const(5)+1);
```

```
%EVi=log(roir.^2)*ones(12,1)-log(MRin.^2)
```

```
EVi=0.46*ones(n,1)-MRin;
```

```
% Sistemin toplam hata fonksiyonunu hesaplayan program
```

```
options(14)=100*n;
```

```
Gr=15.6;
```

```
%toplamlam=exp(-Const(6).*EN)+exp(-Const(7).*EVi)+Const(8).*((Gt-Gref*ones(12,1)).^2);
```

```
%F=ortmrin+(exp(0.01*abs(ortgt-Gref)))*0.4
```

```
F=ortmrin+(exp(0.01*abs(3*(Gt(1)-Gr)+3*(Gt(2)-Gr)+3*(Gt(3)-Gr)+3*(Gt(4)-Gr)+5*(Gt(5)-Gr)+8*(Gt(6)-Gr)+8*(Gt(7)-Gr)+5*(Gt(8)-Gr)+5*(Gt(9)-Gr)+4*(Gt(10)-Gr))))*8
```

```
%F=sum(toplam)
```

Bilgin31_NE329S01C_PS_SP

```
function F=tez_NE329S01
```

```
%Program İlgar ALİYEV tarafından hazırlanmıştır
```

```
%Tez Danışmanı Prof. Dr. Filiz GÜNEŞ
```

```
%
```

```
%Tranzistör için Vce=10 V; Ic=20 mA olarak alınmıştır.
```

```
%Tüm Değerler Prof.Dr. Filiz GÜNEŞ'in Çalışmalarından Derlenmiştir.
```

```
% Giriş Bilgilerinin "bin" klasörü altındaki csv uzantılı dosyalardan okunması
```

```
Const = DLMREAD('Dosya_Const.csv',';',0,0,'A1..J1');
```

```
In = DLMREAD('Dosya_Input.csv',';',0,0,'A1..N12');
```

```
% Kaynak ve Yük Empedansları
```

```
Zs_const=50;
```

```
Rs_const=real(Zs_const);
```

```
Zl_const=50;
```

```
Rl_const=real(Zl_const);
```

```
% Gürültü Faktörü Referans Değeri
```

```
Fr=10^(Const(4)*0.1)
```

```
% z optimize edilecek hat parçalarının karakteristik empedansları
```

```
n=12;
```

```
%z1=l(7);
```

```
z1=l(5);
```

```
z2=l(6);
```

```
z3=l(7);
```

```
z4=l(8);
```

```
%z6=l(12);
```

```
%l1=l(1);
```

```
l1=l(1);
```

```
l2=l(2);
```

```
l3=l(3);
```

```
l4=l(4);
```

```
%l6=l(6);
```

```
% Referans Kazanç Değeri
```

```
Gref=10^(Const(10)*0.1);
```

```
% İlk Değerlerin Tanımlanması
```

```
n=12;
```

```
Gt=zeros(n,1);
```

```
Gav=zeros(n,1);
```

```
E1=zeros(2,2);
```

```
E2=zeros(2,2);
```

```
E3=zeros(2,2);
```

```
E4=zeros(2,2);
```

```
E5=zeros(2,2);
```

```
E6=zeros(2,2);
```

```
MRin=zeros(n,1)
```

```
EN=zeros(n,1);
```

```
GT=zeros(n,1);
```

```
LT=zeros(n,1);
```

```
Noise=zeros(n,1);
```

```
Zs=ones(n,7);
```

```
Zopt=zeros(n,1);
```

```
Zout=zeros(n,1);
```

```
Rout=zeros(n,1);
```

```
Zm=zeros(n,7);
```

```
Zi7=zeros(n,1);
```

```
A=zeros(n,1);
```

```
B=zeros(n,1);
```

```
C=zeros(n,1);
```

```
D=zeros(n,1);
```

```
AA=zeros(n,1);
```

```
BB=zeros(n,1);
```

```
CC=zeros(n,1);
```

```
DD=zeros(n,1);
```

```
EE=zeros(n,1);
```

```
FF=zeros(n,1);
```

```
GG=zeros(n,1);
```

```
HH=zeros(n,1);
```

```
roi=zeros(n,1);
```

```
EVi=zeros(n,1);
```

```
toplam=zeros(n,1);
```

```
% S Parametrelerinin hesaplanması
```

```
S = MATCOMP(In(:,[3:2:9,12]),In(:,[4:2:10,13]));
```

```
G_opt=S(:,5);
```

```
S = S(:,1:4);
```

```
% ABCD Parametrelerinin Hesabı
```

```
A = ((1+S(:,1)).*(1-S(:,4))+S(:,2).*S(:,3))./(2.*S(:,2)));
```

```
B = 50.*((1+S(:,1)).*(1+S(:,4))-S(:,2).*S(:,3))./(2.*S(:,2)));
```

```
C = (1/50).*((1-S(:,1)).*(1-S(:,4))-S(:,2).*S(:,3))./(2.*S(:,2)));
```

```
D = ((1-S(:,1)).*(1+S(:,4))+S(:,2).*S(:,3))./(2.*S(:,2)));
```

```
% Tranzistörün Eşdeğer Gürültü Direnci
```

```
Rn =In(:,14).*Const(2);
```

```
% Minimum Gürültü Faktörü
```

```
Fmin =10.^(In(:,11).*0.1);
```

```
% Thetaların Hesabı
```

```
Theta=(2*pi/(Const(3)))*kron(1*1e-2,In(:,2)*1e9); % 1 cm. cinsinden ve frekans GHz  
cinsinden olduğu için 0.01 ve 1*10^9 ile çarpılıyor.
```

```
% Kazancın Hesaplanması
```

```
for n=1:12
```

```
% % E1=[cos(Theta(n,1)),i.*z1.*sin(Theta(n,1));i.*sin(Theta(n,1))./z1,cos(Theta(n,1))];
```

```
%E3=[cos(Theta(n,3)),i.*z3.*sin(Theta(n,3));i.*sin(Theta(n,3))./z3,cos(Theta(n,3))];
```

```
%E4=[cos(Theta(n,4)),i.*z4.*sin(Theta(n,4));i.*sin(Theta(n,4))./z4,cos(Theta(n,4))];
```

```
% E6=[cos(Theta(n,6)),i.*z6.*sin(Theta(n,6));i.*sin(Theta(n,6))./z6,cos(Theta(n,6))];
```

```
%E2=[1,0;-i./(z2.*tan(Theta(n,2))),1];
```

```
%E5=[1,0;-i./(z2.*tan(Theta(n,5))),1];
```

```
% E1=[cos(Theta(n,1)),i.*z1.*sin(Theta(n,1));i.*sin(Theta(n,1))./z1,cos(Theta(n,1))];
```

```
E3=[cos(Theta(n,2)),i.*z2.*sin(Theta(n,2));i.*sin(Theta(n,2))./z2,cos(Theta(n,2))];
```

```
E4=[cos(Theta(n,3)),i.*z3.*sin(Theta(n,3));i.*sin(Theta(n,3))./z3,cos(Theta(n,3))];
```

```
% E6=[cos(Theta(n,6)),i.*z6.*sin(Theta(n,6));i.*sin(Theta(n,6))./z6,cos(Theta(n,6))];
```

```
E2=[1,0;-i./(z1.*tan(Theta(n,1))),1];
```

```
E5=[1,0;-i./(z4.*tan(Theta(n,4))),1];
```

```
% M=E1*E2*E3*[A(n),B(n);C(n),D(n)]*E4*E5*E6;
```

```

M=E2*E3*[A(n),B(n);C(n),D(n)]*E4*E5;
Mfront=E2*E3;
Gt(n)=4*Rs_const*Rl_const/(
    abs(M(1,1).*Zl_const+M(1,2)+Zs_const*(M(2,1).*Zl_const+M(2,2))).^2)

Zin(n)=(M(1,1).*Zl_const+M(1,2))./(M(2,1).*Zl_const+M(2,2))

MRin(n)=abs((Zs_const-conj(Zin(n)))/(Zs_const+Zin(n)))
%Zstr:Zstransistor

Zstr=(Mfront(1,2)+Mfront(2,2).*Zs_const)/(Mfront(1,1)+Mfront(2,1).*Zs_const)

Zout=(M(1,2)+M(2,2).*Zs_const)/(M(1,1)+M(2,1).*Zs_const)
Rout=real(Zout);

Gav(n)=4*Rs_const*Rout/(
    abs(M(1,1).*conj(Zout)+M(1,2)+Zs_const*(M(2,1).*conj(Zout)+M(2,2))).^2);

end

ortgt=sum(Gt(2:11));
ortgt=ortgt/10;
ortmrin=sum(MRin(2:11));
ortmrin=ortmrin/10;

% İlk Katın Kaynak Empedansı
Zs(:,1)=Zs_const*ones(n,1);

% İkinci Katın Kaynak Empedansı

%Zs(:,2)=(i.*z1.*sin(Theta(:,1))+cos(Theta(:,1)).*Zs(:,1))./(cos(Theta(:,1))+i.*sin(Theta(:,1)).
    *Zs(:,1)./z1);

% Üçüncü Katın Kaynak Empedansı
Zs(:,2)=Zs(:,1)./(1+(-i./z2./(tan(Theta(:,2)))).*Zs(:,1));

% Dördüncü Katın Kaynak Empedansı
Zs(:,3)=(i.*z3.*sin(Theta(:,3))+cos(Theta(:,3)).*Zs(:,2))./(cos(Theta(:,3))+i.*sin(Theta(:,3)).*
    Zs(:,2)./z3);

% Transistör İçin Optimum Kaynak Empedansı
Zopt=50*(1+G_opt)./(1-G_opt);
%Rs4=real(Zs(:,4));
Rs3=real(Zs(:,3));
Ropt=real(Zopt);

```

% Giriş Uydurma Devresi Pasif Ve Kayıpsız Olduğundaki Transduser Güç Kazancı

```
GT=4*Rs3.*Ropt./(abs(Zs(:,3))+conj(Zopt)).^2;
LT=1./GT;
```

```
%Rstr=real(Zstr);
%Noise=Fmin+(Rn.*(abs(Zstr-Zopt).^2)./(abs(Zopt).^2).*Rstr)
Noise=(4*Rn.*Ropt.*(1./GT-1))./((abs(Zopt)).^2)+Fmin
ortnoise=sum(Noise(2:11))/10;
```

% Gürültü Hata İfadesi

```
EN=1+(Fr*ones(12,1)-Fmin).*((abs(Zopt)).^2)./(4*Rn.*Ropt)-(1./GT)
```

% İlk Katın Kaynak Empedansı

```
%Zm(:,1)=-conj(Zs_const)*ones(12,1);
%Zs(:,1)=Zs_const*ones(12,1);
%AA=abs(
    (cos(Theta(:,1))+i.*sin(Theta(:,1)).*Zm(:,1)./z1)./(cos(Theta(:,1))+i.*sin(Theta(:,1)).*Zs(:,1)./z1));
```

% İkinci Katın Kaynak Empedansı

```
%Zm(:,2)=(i.*z1.*sin(Theta(:,1))-cos(Theta(:,1)).*conj(Zs(:,1)))./(cos(Theta(:,1))-i.*sin(Theta(:,1)).*conj(Zs(:,1))./z1);
%Zs(:,2)=(i.*z1.*sin(Theta(:,1))+cos(Theta(:,1)).*Zs(:,1))./(cos(Theta(:,1))+i.*sin(Theta(:,1)).*Zs(:,1)./z1);
%BB=abs((1+(-i.*Zm(:,2)./z2./tan(Theta(:,2)))))/(1+(-i.*Zs(:,2)./z2./tan(Theta(:,2)))));
```

% Üçüncü Katın Kaynak Empedansı

```
%Zm(:,3)=-conj(Zs(:,2))./(1-(-i.*conj(Zs(:,2))./z2./tan(Theta(:,2))));
%Zs(:,3)=(Zs(:,2))./(1+(-i.*Zs(:,2)./z2./tan(Theta(:,2))));
%CC=abs((cos(Theta(:,3))+i.*sin(Theta(:,3)).*Zm(:,3)./z3)./(cos(Theta(:,3))+i.*sin(Theta(:,3)).*Zs(:,3)./z3));
```

% Dördüncü Katın Kaynak Empedansı

```
%Zm(:,4)=(i.*z3.*sin(Theta(:,3))-cos(Theta(:,3)).*conj(Zs(:,3)))./(cos(Theta(:,3))-i.*sin(Theta(:,3)).*conj(Zs(:,3))./z3);
%Zs(:,4)=(i.*z3.*sin(Theta(:,3))+cos(Theta(:,3)).*Zs(:,3))./(cos(Theta(:,3))+i.*sin(Theta(:,3)).*Zs(:,3)./z3);
%DD=abs((A+C.*Zm(:,4))./(A-C.*Zs(:,4)));
```

% Beşinci Katın Kaynak Empedansı

```
%Zm(:,5)=(B-D.*conj(Zs(:,4)))./(A-C.*conj(Zs(:,4)));
%Zs(:,5)=(B+D.*Zs(:,4))./(A-C.*Zs(:,4));
%EE=abs((cos(Theta(:,4))+i.*sin(Theta(:,4)).*Zm(:,5)./z4)./(cos(Theta(:,4))+i.*sin(Theta(:,4)).*Zs(:,5)./z4));
```

```

).*Zs(:,5)./z4));

%      Altıncı Katın Kaynak Empedansı

%Zm(:,6)=(i.*z4.*sin(Theta(:,4))-cos(Theta(:,4)).*conj(Zs(:,5)))/(cos(Theta(:,4))-
i.*sin(Theta(:,4)).*conj(Zs(:,5)./z4));
%Zs(:,6)=(i.*z4.*sin(Theta(:,4))+cos(Theta(:,4)).*Zs(:,5))/(cos(Theta(:,4))+i.*sin(Theta(:,4)).
.*Zs(:,5)./z4);
%FF=abs( ( 1+(-i.*Zm(:,6)./z5./tan(Theta(:,5))) )./(1+(-i.*Zs(:,6)./z5./tan(Theta(:,5)))));

%      Yedinci Katın Kaynak Empedansı

%Zm(:,7)=(-conj(Zs(:,6)))/(1+(-i.*conj(Zs(:,6))./z5./tan(Theta(:,5))));
%Zs(:,7)=(Zs(:,6))./(1+(-i.*Zs(:,6)./z5./tan(Theta(:,5))));
%GG=abs((cos(Theta(:,6))+i.*sin(Theta(:,6)).*Zm(:,7)./z6)/(cos(Theta(:,6))+i.*sin(Theta(:,6))
).*Zs(:,7)./z6));

%      Yedinci Katın Giriş Empedansı

%Zi7=z6.*(Zl_const+i.*z6.*tan(Theta(:,6)))/(z6+i.*Zl_const*tan(Theta(:,6)));
%HH=abs((Zi7+Zm(:,7))./(Zi7+Zs(:,7)));

%roi=AA.*BB.*CC.*DD.*EE.*FF.*GG.*HH;

roir=(Const(5)-1)/(Const(5)+1);

%EVi=log(roir.^2)*ones(12,1)-log(MRin.^2)
EVi=0.46*ones(n,1)-MRin;
%      Sistemin toplam hata fonksiyonunu hesaplayan program

options(14)=100*n;

MRinref=0.1;

%toplamlam=exp(-Const(6).*EN)+exp(-Const(7).*EVi)+Const(8).*((Gt-Gref*ones(12,1)).^2);

%F=5*((2*(MRin(3)-MRinref)+2*(MRin(4)-MRinref)+2*(MRin(5)-MRinref)+2*(MRin(6)-
MRinref)+3*(MRin(7)-MRinref)+3*(MRin(8)-MRinref)+2*(MRin(9)-
MRinref)+3*(MRin(10)-MRinref)+2*(MRin(11)-
MRinref)))+(exp(0.01*abs((Gt(1)-Gref)+(Gt(2)-Gref)+(Gt(3)-Gref)+2.5*(Gt(4)-
Gref)+2.5*(Gt(5)-Gref)+2.5*(Gt(6)-Gref)+2.5*(Gt(7)-Gref)+2.5*(Gt(8)-
Gref)+2.5*(Gt(9)-Gref))))*0.7

%F=2*ortmrin+(exp(0.1*abs(ortgt-Gref))-1)*0.45
%F=sum(toplam)
F=(ortmrin-abs(roir))+0.01*(exp(0.4*abs((Gt(1)-Gref)+(Gt(2)-Gref)+(Gt(3)-Gref)+(Gt(4)-
Gref)+(Gt(5)-Gref)+(Gt(6)-Gref)+(Gt(7)-Gref)+(Gt(8)-Gref)+(Gt(9)-
Gref)+(Gt(10)-Gref))))

```

Bilgin31_NE329S01CP

```
function F=tez_NE329S01
```

```
%
```

```
%Program İlgar ALİYEV tarafından hazırlanmıştır
```

```
%Tez Danışmanı Prof. Dr. Filiz GÜNEŞ
```

```
%Tranzistör için Vce=10 V; Ic=20 mA olarak alınmıştır.
```

```
%Tüm Değerler Prof.Dr. Filiz GÜNEŞ'in Çalışmalarından Derlenmiştir.
```

```
% Giriş Bilgilerinin "bin" klasörü altındaki csv uzantılı dosyalardan okunması
```

```
Const = DLMREAD('Dosya_Const.csv',';',0,0,'A1..J1');
```

```
In = DLMREAD('Dosya_Input.csv',';',0,0,'A1..N12');
```

```
% Kaynak ve Yük Empedansları
```

```
Zs_const=50;
```

```
Rs_const=real(Zs_const);
```

```
Zl_const=50;
```

```
Rl_const=real(Zl_const);
```

```
% Gürültü Faktörü Referans Değeri
```

```
Fr=10^(Const(4)*0.1)
```

```
% z optimize edilecek hat parçalarının karakteristik empedansları
```

```
n=12;
```

```
z1=l(7);
```

```
z2=l(8);
```

```
z3=l(9);
```

```
z4=l(10);
```

```
z5=l(11);
```

```
z6=l(12);
```

```
l1=l(1);
```

```
l2=l(2);
```

```
l3=l(3);
```

```
l4=l(4);
```

```
l5=l(5);
```

```
l6=l(6);
```

```
% Referans Kazanç Değeri
```

```
Gref=10^(Const(10)*0.1);
```

% İlk Değerlerin Tanımlanması

```

n=12;
Gt=zeros(n,1);
Gav=zeros(n,1);

E1=zeros(2,2);
E2=zeros(2,2);
E3=zeros(2,2);
E4=zeros(2,2);
E5=zeros(2,2);
E6=zeros(2,2);
MRin=zeros(n,1)

EN=zeros(n,1);

GT=zeros(n,1);
LT=zeros(n,1);

Noise=zeros(n,1);

Zs=ones(n,7);

Zopt=zeros(n,1);
Zout=zeros(n,1);
Rout=zeros(n,1);

Zm=zeros(n,7);

Zi7=zeros(n,1);

A=zeros(n,1);
B=zeros(n,1);
C=zeros(n,1);
D=zeros(n,1);

AA=zeros(n,1);
BB=zeros(n,1);
CC=zeros(n,1);
DD=zeros(n,1);
EE=zeros(n,1);
FF=zeros(n,1);
GG=zeros(n,1);
HH=zeros(n,1);

roi=zeros(n,1);

EVi=zeros(n,1);

toplam=zeros(n,1);

```

% S Parametrelerinin hesaplanması

```
S = MATCOMP(In(:,[3:2:9,12]),In(:,[4:2:10,13]));
G_opt=S(:,5);
S = S(:,1:4);
```

% ABCD Parametrelerinin Hesabı

```
A = ((1+S(:,1)).*(1-S(:,4))+S(:,2).*S(:,3))./(2.*S(:,2));
B = 50.*((1+S(:,1)).*(1+S(:,4))-S(:,2).*S(:,3))./(2.*S(:,2));
C = (1/50).*((1-S(:,1)).*(1-S(:,4))-S(:,2).*S(:,3))./(2.*S(:,2));
D = ((1-S(:,1)).*(1+S(:,4))+S(:,2).*S(:,3))./(2.*S(:,2));
```

% Tranzistörün Eşdeğer Gürültü Direnci

```
Rn =In(:,14).*Const(2);
```

% Minimum Gürültü Faktörü

```
Fmin =10.^(In(:,11).*0.1);
```

% Thetaların Hesabı

```
Theta=(2*pi/(Const(3)))*kron(l*1e-2,In(:,2)*1e9); % l cm. cinsinden ve frekans GHz
cinsinden olduğu için 0.01 ve 1*10^9 ile çarpılıyor.
```

% Kazancın Hesaplanması

```
for n=1:12
```

```
E1=[1,0;-i./(z1.*tan(Theta(n,1))),1];
E3=[1,0;-i./(z3.*tan(Theta(n,3))),1];
E4=[1,0;-i./(z4.*tan(Theta(n,4))),1];
E6=[1,0;-i./(z6.*tan(Theta(n,6))),1];
E2=[cos(Theta(n,2)),i.*z2.*sin(Theta(n,2));i.*sin(Theta(n,2))./z2,cos(Theta(n,2))];
E5=[cos(Theta(n,5)),i.*z5.*sin(Theta(n,5));i.*sin(Theta(n,5))./z5,cos(Theta(n,5))];
```

```
M=E1*E2*E3*[A(n),B(n);C(n),D(n)]*E4*E5*E6;
```

```
Gt(n)=4*Rs_const*Rl_const/(
abs(M(1,1).*Zl_const+M(1,2)+Zs_const*(M(2,1).*Zl_const+M(2,2))).^2)
```

```
Zin(n)=(M(1,1).*Zl_const+M(1,2))./(M(2,1).*Zl_const+M(2,2))
```

```
MRin(n)=abs((Zs_const-conj(Zin(n)))/(Zs_const+Zin(n)))
```

```
Zout=(M(1,2)+M(2,2).*Zs_const)/(M(1,1)+M(2,1).*Zs_const)
Rout=real(Zout);
```

```
Gav(n)=4*Rs_const*Rout/(
```

```

abs(M(1,1).*conj(Zout)+M(1,2)+Zs_const*(M(2,1).*conj(Zout)+M(2,2))).^2);

end
ortgt=2*Gt(1)+2*Gt(2)+2*Gt(3)+2*Gt(4)+3*Gt(5)+4*Gt(6)+4*Gt(7)+4*Gt(8)+4*Gt(9)+4*G
t(10)+2*Gt(11)+Gt(12);
%ortgt=sum(Gt(2:11));
ortgt=ortgt/34;
%ortgt=ortgt/10;
ortmrin=MRin(1)+4*MRin(2)+4*MRin(3)+5*MRin(4)+5*MRin(5)+5*MRin(6)+4*MRin(7)
+4*MRin(8)+4*MRin(9)+3*MRin(10)+MRin(11)+MRin(12);
%ortmrin=sum(MRin(2:11));
ortmrin=ortmrin/40;

% İlk Katin Kaynak Empedansi

Zs(:,1)=Zs_const*ones(n,1);

% İkinci Katin Kaynak Empedansi

Zs(:,2)=Zs(:,1)./(1+(-i./(z1.*(tan(Theta(:,1)))))).*Zs(:,1));

% Üçüncü Katin Kaynak Empedansi

Zs(:,3)=(i.*z2.*sin(Theta(:,2))+cos(Theta(:,2)).*Zs(:,2))./(cos(Theta(:,2))+i.*sin(Theta(:,2)).*
Zs(:,2)./z2);

% Dördüncü Katin Kaynak Empedansı

Zs(:,4)=Zs(:,3)./(1+(-i./(z3.*(tan(Theta(:,3)))))).*Zs(:,3));

% Transistör İçin Optimum Kaynak Empedansı

Zopt=50*(1+G_opt)./(1-G_opt);
Rs4=real(Zs(:,4));
Ropt=real(Zopt);

% Giriş Uydurma Devresi Pasif Ve Kayıpsız Olduğundaki Transduser Güç Kazancı

GT=4*Rs4.*Ropt./(abs(Zs(:,4))+conj(Zopt)).^2;
LT=1./GT;

Noise=(4*Rn.*Ropt.*(1./GT-1))./((abs(Zopt)).^2)+Fmin

% Gürültü Hata İfadesi

EN=1+(Fr*ones(12,1)-Fmin).*((abs(Zopt)).^2)./(4*Rn.*Ropt)-(1./GT)

% İlk Katin Kaynak Empedansı

Zm(:,1)=-conj(Zs_const)*ones(12,1);

```

```

Zs(:,1)=Zs_const*ones(12,1);
AA=abs(
    (cos(Theta(:,1))+i.*sin(Theta(:,1)).*Zm(:,1)./z1)./(cos(Theta(:,1))+i.*sin(Theta(:,1)).*Zs(:,1)./z1));

% İkinci Katın Kaynak Empedansı

Zm(:,2)=(i.*z1.*sin(Theta(:,1))-cos(Theta(:,1)).*conj(Zs(:,1)))./(cos(Theta(:,1))-i.*sin(Theta(:,1)).*conj(Zs(:,1))./z1);
Zs(:,2)=(i.*z1.*sin(Theta(:,1))+cos(Theta(:,1)).*Zs(:,1))./(cos(Theta(:,1))+i.*sin(Theta(:,1)).*Zs(:,1)./z1);
BB=abs((1+(-i.*Zm(:,2)./z2./tan(Theta(:,2)))))/(1+(-i.*Zs(:,2)./z2./tan(Theta(:,2)))));

% Üçüncü Katın Kaynak Empedansı

Zm(:,3)=-conj(Zs(:,2))./(1-(-i.*conj(Zs(:,2))./z2./tan(Theta(:,2)))));
Zs(:,3)=(Zs(:,2))./(1+(-i.*Zs(:,2)./z2./tan(Theta(:,2)))));
CC=abs((cos(Theta(:,3))+i.*sin(Theta(:,3)).*Zm(:,3)./z3)./(cos(Theta(:,3))+i.*sin(Theta(:,3)).*Zs(:,3)./z3));

% Dördüncü Katın Kaynak Empedansı

Zm(:,4)=(i.*z3.*sin(Theta(:,3))-cos(Theta(:,3)).*conj(Zs(:,3)))./(cos(Theta(:,3))-i.*sin(Theta(:,3)).*conj(Zs(:,3))./z3);
Zs(:,4)=(i.*z3.*sin(Theta(:,3))+cos(Theta(:,3)).*Zs(:,3))./(cos(Theta(:,3))+i.*sin(Theta(:,3)).*Zs(:,3)./z3);
DD=abs((A+C.*Zm(:,4))./(A-C.*Zs(:,4)));

% Beşinci Katın Kaynak Empedansı

Zm(:,5)=(B-D.*conj(Zs(:,4)))./(A-C.*conj(Zs(:,4)));
Zs(:,5)=(B+D.*Zs(:,4))./(A-C.*Zs(:,4));
EE=abs((cos(Theta(:,4))+i.*sin(Theta(:,4)).*Zm(:,5)./z4)./(cos(Theta(:,4))+i.*sin(Theta(:,4)).*Zs(:,5)./z4));

% Altıncı Katın Kaynak Empedansı

Zm(:,6)=(i.*z4.*sin(Theta(:,4))-cos(Theta(:,4)).*conj(Zs(:,5)))./(cos(Theta(:,4))-i.*sin(Theta(:,4)).*conj(Zs(:,5))./z4);
Zs(:,6)=(i.*z4.*sin(Theta(:,4))+cos(Theta(:,4)).*Zs(:,5))./(cos(Theta(:,4))+i.*sin(Theta(:,4)).*Zs(:,5)./z4);
FF=abs((1+(-i.*Zm(:,6)./z5./tan(Theta(:,5)))))/(1+(-i.*Zs(:,6)./z5./tan(Theta(:,5)))));

% Yedinci Katın Kaynak Empedansı

Zm(:,7)=(-conj(Zs(:,6)))./(1-(-i.*conj(Zs(:,6))./z5./tan(Theta(:,5)))));
Zs(:,7)=(Zs(:,6))./(1+(-i.*Zs(:,6)./z5./tan(Theta(:,5)))));
GG=abs((cos(Theta(:,6))+i.*sin(Theta(:,6)).*Zm(:,7)./z6)./(cos(Theta(:,6))+i.*sin(Theta(:,6)).*Zs(:,7)./z6));

% Yedinci Katın Giriş Empedansı

```

```

Zi7=z6.*(Zl_const+i.*z6.*tan(Theta(:,6)))./(z6+i.*Zl_const*tan(Theta(:,6)));
HH=abs((Zi7+Zm(:,7))./(Zi7+Zs(:,7)));

roi=AA.*BB.*CC.*DD.*EE.*FF.*GG.*HH;

roir=(Const(5)-1)/(Const(5)+1);

EVi=log(roir.^2)*ones(12,1)-log(MRin.^2);

%           Sistemin toplam hata fonksiyonunu hesaplayan program

options(14)=100*n;

MRinref=0.1;
%toplamlam=exp(-Const(6).*EN)+exp(-Const(7).*EVi)+Const(8).*((Gt-Gref*ones(12,1)).^2);

Gr=15.86;
%F=sum(toplam)
%toplamlam=exp(-Const(6).*EN)+exp(-Const(7).*EVi)+Const(8).*((Gt-Gref*ones(12,1)).^2);
%F=ortmrin+(exp(0.01*abs(ortgt-Gref)))*0.4
%F=((MRin(2)-MRinref)+2*(MRin(3)-MRinref)+2*(MRin(4)-MRinref)+2*(MRin(5)-
MRinref)+2*(MRin(6)-MRinref)+2*(MRin(7)-MRinref)+2*(MRin(8)-
MRinref)+(MRin(9)-MRinref)+(MRin(10)-MRinref)+(MRin(11)-
MRinref)+(exp(0.01*abs((Gt(1)-Gref)+(Gt(2)-Gref)+(Gt(3)-Gref)+2.5*(Gt(4)-
Gref)+2.5*(Gt(5)-Gref)+3.5*(Gt(6)-Gref)+3*(Gt(7)-Gref)+3*(Gt(8)-
Gref)+2.5*(Gt(9)-Gref)+2*(Gt(10)-Gref)))))*80
%F=ortmrin+(exp(0.1*abs(ortgt-Gref))-1)*0.345
%F=ortmrin*1.1+(exp(0.1*abs(ortgt-Gref))-1)*0.4
F=ortmrin+(exp(0.01*abs(7*(Gt(1)-Gr)+10*(Gt(2)-Gr)+6*(Gt(3)-Gr)+5*(Gt(4)-
Gr)+5*(Gt(5)-Gr)+8*(Gt(6)-Gr)+8*(Gt(7)-Gr)+5*(Gt(8)-Gr)+5*(Gt(9)-
Gr)+4*(Gt(10)-Gr)))))*5.5

%F=sum(toplam)

```

Bilgin31_NE329S01CPT

function F=tez_NE329S01

```

%
%Program İlgar ALİYEV tarafından hazırlanmıştır
%Tez Danışmanı Prof. Dr. Filiz GÜNEŞ

%Tranzistör için Vce=10 V; Ic=20 mA olarak alınmıştır.
%Tüm Değerler Prof.Dr. Filiz GÜNEŞ'in Çalışmalarından Derlenmiştir.

% Giriş Bilgilerinin "bin" klasörü altındaki csv uzantılı dosyalardan okunması

Const = DLMREAD('Dosya_Const.csv',';',0,0,'A1..J1');
In = DLMREAD('Dosya_Input.csv',';',0,0,'A1..N12');

%          Kaynak ve Yük Empedansları

Zs_const=50;
Rs_const=real(Zs_const);

Zl_const=50;
Rl_const=real(Zl_const);

%          Gürültü Faktörü Referans Değeri

Fr=10^(Const(4)*0.1)

%          z optimize edilecek hat parçalarının karakteristik empedansları

n=12;

z1=l(7);
z2=l(8);
z3=l(9);
z4=l(10);
z5=l(11);
z6=l(12);

l1=l(1);
l2=l(2);
l3=l(3);
l4=l(4);
l5=l(5);
l6=l(6);

%          Referans Kazanç Değeri

Gref=10^(Const(10)*0.1);

%          İlk Değerlerin Tanımlanması

```

```

n=12;
Gt=zeros(n,1);
Gav=zeros(n,1);

E1=zeros(2,2);
E2=zeros(2,2);
E3=zeros(2,2);
E4=zeros(2,2);
E5=zeros(2,2);
E6=zeros(2,2);
MRin=zeros(n,1)

EN=zeros(n,1);

GT=zeros(n,1);
LT=zeros(n,1);

Noise=zeros(n,1);

Zs=ones(n,7);

Zopt=zeros(n,1);
Zout=zeros(n,1);
Rout=zeros(n,1);

Zm=zeros(n,7);

Zi7=zeros(n,1);

A=zeros(n,1);
B=zeros(n,1);
C=zeros(n,1);
D=zeros(n,1);

AA=zeros(n,1);
BB=zeros(n,1);
CC=zeros(n,1);
DD=zeros(n,1);
EE=zeros(n,1);
FF=zeros(n,1);
GG=zeros(n,1);
HH=zeros(n,1);

roi=zeros(n,1);

EVi=zeros(n,1);

toplam=zeros(n,1);

%           S Parametrelerinin hesaplanması

```

```
S = MATCOMP(In(:,[3:2:9,12]),In(:,[4:2:10,13]));
G_opt=S(:,5);
S = S(:,1:4);
```

```
% ABCD Parametrelerinin Hesabı
```

```
A = ((1+S(:,1)).*(1-S(:,4))+S(:,2).*S(:,3))./(2.*S(:,2));
B = 50.*((1+S(:,1)).*(1+S(:,4))-S(:,2).*S(:,3))./(2.*S(:,2));
C = (1/50).*((1-S(:,1)).*(1-S(:,4))-S(:,2).*S(:,3))./(2.*S(:,2));
D = ((1-S(:,1)).*(1+S(:,4))+S(:,2).*S(:,3))./(2.*S(:,2));
```

```
% Transistörün Eşdeğer Gürültü Direnci
```

```
Rn =In(:,14).*Const(2);
```

```
% Minimum Gürültü Faktörü
```

```
Fmin =10.^(In(:,11).*0.1);
```

```
% Thetaların Hesabı
```

```
Theta=(2*pi/(Const(3)))*kron(1*1e-2,In(:,2)*1e9); % l cm. cinsinden ve frekans GHz
cinsinden olduğu için 0.01 ve 1*10^9 ile çarpılıyor.
```

```
% Kazancın Hesaplanması
```

```
for n=1:12
```

```
E1=[1,0;-i./(z1.*tan(Theta(n,1))),1];
E3=[1,0;-i./(z3.*tan(Theta(n,3))),1];
E5=[1,0;-i./(z5.*tan(Theta(n,5))),1];
E2=[cos(Theta(n,2)),i.*z2.*sin(Theta(n,2));i.*sin(Theta(n,2))./z2,cos(Theta(n,2))];
E4=[cos(Theta(n,4)),i.*z4.*sin(Theta(n,4));i.*sin(Theta(n,4))./z4,cos(Theta(n,4))];
E6=[cos(Theta(n,6)),i.*z6.*sin(Theta(n,6));i.*sin(Theta(n,6))./z6,cos(Theta(n,6))];
```

```
M=E1*E2*E3*[A(n),B(n);C(n),D(n)]*E4*E5*E6;
```

```
Gt(n)=4*Rs_const*Rl_const/(
abs(M(1,1).*Zl_const+M(1,2)+Zs_const*(M(2,1).*Zl_const+M(2,2))).^2)
```

```
Zin(n)=(M(1,1).*Zl_const+M(1,2))./(M(2,1).*Zl_const+M(2,2))
```

```
MRin(n)=abs((Zs_const-conj(Zin(n)))/(Zs_const+Zin(n)))
```

```
Zout=(M(1,2)+M(2,2).*Zs_const)/(M(1,1)+M(2,1).*Zs_const)
Rout=real(Zout);
```

```
Gav(n)=4*Rs_const*Rout/(
abs(M(1,1).*conj(Zout)+M(1,2)+Zs_const*(M(2,1).*conj(Zout)+M(2,2))).^2);
```

```
end
```

```
ortmr=MRin(1)+MRin(2)+MRin(3)+MRin(4)+MRin(5)+MRin(6)+MRin(7)+MRin(8)+MRin(9)+MRin(10);
```

```
ortmrin=ortmr/10;
```

```
ortgt=Gt(1)+Gt(2)+Gt(3)+Gt(4)+Gt(5)+Gt(6)+Gt(7)+Gt(8)+Gt(9)+Gt(10);
```

```
ortgt=ortgt/10;
```

```
% İlk Katin Kaynak Empedansi
```

```
Zs(:,1)=Zs_const*ones(n,1);
```

```
% İkinci Katin Kaynak Empedansi
```

```
Zs(:,2)=Zs(:,1)./(1+(-i./(z1.*(tan(Theta(:,1)))))).*Zs(:,1));
```

```
% Üçüncü Katin Kaynak Empedansi
```

```
Zs(:,3)=(i.*z2.*sin(Theta(:,2))+cos(Theta(:,2)).*Zs(:,2))./(cos(Theta(:,2))+i.*sin(Theta(:,2))).*Zs(:,2)./z2);
```

```
% Dördüncü Katin Kaynak Empedansi
```

```
Zs(:,4)=Zs(:,3)./(1+(-i./(z3.*(tan(Theta(:,3)))))).*Zs(:,3));
```

```
% İlk Katin Kaynak Empedansi
```

```
%Zs(:,1)=Zs_const*ones(n,1);
```

```
% İkinci Katin Kaynak Empedansi
```

```
%Zs(:,2)=Zs(:,1)./(1+(-i./(z1.*(tan(Theta(:,1)))))).*Zs(:,1));
```

```
% Üçüncü Katin Kaynak Empedansi
```

```
%Zs(:,3)=Zs(:,2)./(1+(-i./z2./(tan(Theta(:,2))))).*Zs(:,2));
```

```
% Dördüncü Katin Kaynak Empedansi
```

```
%Zs(:,4)=(i.*z3.*sin(Theta(:,3))+cos(Theta(:,3)).*Zs(:,3))./(cos(Theta(:,3))+i.*sin(Theta(:,3))).*Zs(:,3)./z3);
```

```
% Transistör İçin Optimum Kaynak Empedansi
```

```
Zopt=50*(1+G_opt)./(1-G_opt);
```

```
Rs4=real(Zs(:,4));
```

```
Ropt=real(Zopt);
```

% Giriş Uydurma Devresi Pasif Ve Kayıpsız Olduğundaki Transduser Güç Kazancı

GT=4*Rs4.*Ropt./((abs(Zs(:,4))+conj(Zopt))).^2;
LT=1./GT;

Noise=(4*Rn.*Ropt.*(1./GT-1))./((abs(Zopt)).^2)+Fmin

% Gürültü Hata İfadesi

EN=1+(Fr*ones(12,1)-Fmin).*((abs(Zopt)).^2)./(4*Rn.*Ropt)-(1./GT)

% İlk Katın Kaynak Empedansı

Zm(:,1)=-conj(Zs_const)*ones(12,1);
Zs(:,1)=Zs_const*ones(12,1);
AA=abs(
 (cos(Theta(:,1))+i.*sin(Theta(:,1)).*Zm(:,1)./z1)./(cos(Theta(:,1))+i.*sin(Theta(:,1)).*Zs(:,1)./z1));

% İkinci Katın Kaynak Empedansı

Zm(:,2)=(i.*z1.*sin(Theta(:,1))-cos(Theta(:,1)).*conj(Zs(:,1)))./(cos(Theta(:,1))-
 i.*sin(Theta(:,1)).*conj(Zs(:,1))./z1);
Zs(:,2)=(i.*z1.*sin(Theta(:,1))+cos(Theta(:,1)).*Zs(:,1))./(cos(Theta(:,1))+i.*sin(Theta(:,1)).*
 Zs(:,1)./z1);
BB=abs((1+(-i.*Zm(:,2))./z2./tan(Theta(:,2))))./(1+(-i.*Zs(:,2))./z2./tan(Theta(:,2)))));

% Üçüncü Katın Kaynak Empedansı

Zm(:,3)=-conj(Zs(:,2))./(1+(-i.*conj(Zs(:,2))./z2./tan(Theta(:,2)))));
Zs(:,3)=(Zs(:,2))./(1+(-i.*Zs(:,2))./z2./tan(Theta(:,2)))));
CC=abs((cos(Theta(:,3))+i.*sin(Theta(:,3)).*Zm(:,3))./z3)./(cos(Theta(:,3))+i.*sin(Theta(:,3)).*
 Zs(:,3))./z3));

% Dördüncü Katın Kaynak Empedansı

Zm(:,4)=(i.*z3.*sin(Theta(:,3))-cos(Theta(:,3)).*conj(Zs(:,3)))./(cos(Theta(:,3))-
 i.*sin(Theta(:,3)).*conj(Zs(:,3))./z3);
Zs(:,4)=(i.*z3.*sin(Theta(:,3))+cos(Theta(:,3)).*Zs(:,3))./(cos(Theta(:,3))+i.*sin(Theta(:,3)).*
 Zs(:,3))./z3);
DD=abs((A+C.*Zm(:,4))./(A-C.*Zs(:,4)));

% Beşinci Katın Kaynak Empedansı

Zm(:,5)=(B-D.*conj(Zs(:,4)))./(A-C.*conj(Zs(:,4)));
Zs(:,5)=(B+D.*Zs(:,4))./(A-C.*Zs(:,4));
EE=abs((cos(Theta(:,4))+i.*sin(Theta(:,4)).*Zm(:,5))./z4)./(cos(Theta(:,4))+i.*sin(Theta(:,4)).*
 Zs(:,5))./z4));

% Altıncı Katın Kaynak Empedansı

```

Zm(:,6)=(i.*z4.*sin(Theta(:,4))-cos(Theta(:,4)).*conj(Zs(:,5)))/(cos(Theta(:,4))-
i.*sin(Theta(:,4)).*conj(Zs(:,5))./z4);
Zs(:,6)=(i.*z4.*sin(Theta(:,4))+cos(Theta(:,4)).*Zs(:,5))/(cos(Theta(:,4))+i.*sin(Theta(:,4)).*
Zs(:,5))./z4);
FF=abs( ( 1+(-i.*Zm(:,6))./z5./tan(Theta(:,5))) )/(1+(-i.*Zs(:,6))./z5./tan(Theta(:,5)))));

```

```
% Yedinci Katın Kaynak Empedansı
```

```

Zm(:,7)=(-conj(Zs(:,6)))/(1-(-i.*conj(Zs(:,6))./z5./tan(Theta(:,5))));
Zs(:,7)=(Zs(:,6))/(1+(-i.*Zs(:,6))./z5./tan(Theta(:,5))));
GG=abs((cos(Theta(:,6))+i.*sin(Theta(:,6)).*Zm(:,7))./z6)/(cos(Theta(:,6))+i.*sin(Theta(:,6)).*
Zs(:,7))./z6));

```

```
% Yedinci Katın Giriş Empedansı
```

```

Zi7=z6.*(Zl_const+i.*z6.*tan(Theta(:,6)))/(z6+i.*Zl_const*tan(Theta(:,6)));
HH=abs((Zi7+Zm(:,7))/(Zi7+Zs(:,7)));

```

```
roi=AA.*BB.*CC.*DD.*EE.*FF.*GG.*HH;
```

```
roir=(Const(5)-1)/(Const(5)+1);
```

```
EVi=log(roir.^2)*ones(12,1)-log(MRin.^2);
```

```
% Sistemin toplam hata fonksiyonunu hesaplayan program
```

```
options(14)=100*n;
```

```
toplamlam=exp(-Const(6).*EN)+exp(-Const(7).*EVi)+Const(8).*((Gt-Gref*ones(12,1)).^2);
```

```
Gr=15.86;
```

```
%F=sum(toplam)
```

```
%toplamlam=exp(-Const(6).*EN)+exp(-Const(7).*EVi)+Const(8).*((Gt-Gref*ones(12,1)).^2);
```

```
%F=ortmrin+(exp(0.01*abs(ortgt-Gref)))*0.4
```

```

%F=((MRin(2)-MRinref)+2*(MRin(3)-MRinref)+2*(MRin(4)-MRinref)+2*(MRin(5)-
MRinref)+2*(MRin(6)-MRinref)+2*(MRin(7)-MRinref)+2*(MRin(8)-
MRinref)+(MRin(9)-MRinref)+(MRin(10)-MRinref)+(MRin(11)-
MRinref)+(exp(0.01*abs((Gt(1)-Gref)+(Gt(2)-Gref)+(Gt(3)-Gref)+2.5*(Gt(4)-
Gref)+2.5*(Gt(5)-Gref)+3.5*(Gt(6)-Gref)+3*(Gt(7)-Gref)+3*(Gt(8)-
Gref)+2.5*(Gt(9)-Gref)+2*(Gt(10)-Gref)))))*80

```

```
%F=ortmrin+(exp(0.1*abs(ortgt-Gref))-1)*0.345
```

```
%F=ortmrin*1.1+(exp(0.1*abs(ortgt-Gref))-1)*0.4
```

```

%F=ortmrin+(exp(0.01*abs(7*(Gt(1)-Gr)+10*(Gt(2)-Gr)+6*(Gt(3)-Gr)+5*(Gt(4)-
Gr)+5*(Gt(5)-Gr)+8*(Gt(6)-Gr)+8*(Gt(7)-Gr)+5*(Gt(8)-Gr)+5*(Gt(9)-
Gr)+4*(Gt(10)-Gr)))))*5.5

```

```

F=ortmrin+(exp(0.01*abs(3*(Gt(1)-Gr)+3*(Gt(2)-Gr)+4*(Gt(3)-Gr)+5*(Gt(4)-Gr)+5*(Gt(5)-
Gr)+8*(Gt(6)-Gr)+8*(Gt(7)-Gr)+5*(Gt(8)-Gr)+5*(Gt(9)-Gr)+6*(Gt(10)-
Gr)))))*12

```

```
%F=sum(toplam)
```

Bilgin31_NE329S01CTP

function F=tez_NE329S01

```

%
%Program İlgar ALİYEV tarafından hazırlanmıştır
%Tez Danışmanı Prof. Dr. Filiz GÜNEŞ

%Tranzistör için Vce=10 V; Ic=20 mA olarak alınmıştır.
%Tüm Değerler Prof.Dr. Filiz GÜNEŞ'in Çalışmalarından Derlenmiştir.

% Giriş Bilgilerinin "bin" klasörü altındaki csv uzantılı dosyalardan okunması

Const = DLMREAD('Dosya_Const.csv',';',0,0,'A1..J1');
In = DLMREAD('Dosya_Input.csv',';',0,0,'A1..N12');

%          Kaynak ve Yük Empedansları

Zs_const=50;
Rs_const=real(Zs_const);

Zl_const=50;
Rl_const=real(Zl_const);

%          Gürültü Faktörü Referans Değeri

Fr=10^(Const(4)*0.1)

%          z optimize edilecek hat parçalarının karakteristik empedansları

n=12;

z1=l(7);
z2=l(8);
z3=l(9);
z4=l(10);
z5=l(11);
z6=l(12);

l1=l(1);
l2=l(2);
l3=l(3);
l4=l(4);
l5=l(5);
l6=l(6);

%          Referans Kazanç Değeri

Gref=10^(Const(10)*0.1);

%          İlk Değerlerin Tanımlanması

```

```
n=12;
Gt=zeros(n,1);
Gav=zeros(n,1);
```

```
E1=zeros(2,2);
E2=zeros(2,2);
E3=zeros(2,2);
E4=zeros(2,2);
E5=zeros(2,2);
E6=zeros(2,2);
MRin=zeros(n,1)
```

```
EN=zeros(n,1);
```

```
GT=zeros(n,1);
LT=zeros(n,1);
```

```
Noise=zeros(n,1);
```

```
Zs=ones(n,7);
```

```
Zopt=zeros(n,1);
Zout=zeros(n,1);
Rout=zeros(n,1);
```

```
Zm=zeros(n,7);
```

```
Zi7=zeros(n,1);
```

```
A=zeros(n,1);
B=zeros(n,1);
C=zeros(n,1);
D=zeros(n,1);
```

```
AA=zeros(n,1);
BB=zeros(n,1);
CC=zeros(n,1);
DD=zeros(n,1);
EE=zeros(n,1);
FF=zeros(n,1);
GG=zeros(n,1);
HH=zeros(n,1);
```

```
roi=zeros(n,1);
```

```
EVi=zeros(n,1);
```

```
toplam=zeros(n,1);
```

```
%          S Parametrelerinin hesaplanması
```

```
S = MATCOMP(In(:,[3:2:9,12]),In(:,[4:2:10,13]));
G_opt=S(:,5);
S = S(:,1:4);
```

```
%          ABCD Parametrelerinin Hesabı
```

```
A = ((1+S(:,1)).*(1-S(:,4))+S(:,2).*S(:,3))./(2.*S(:,2));
B = 50.*((1+S(:,1)).*(1+S(:,4))-S(:,2).*S(:,3))./(2.*S(:,2));
C = (1/50).*((1-S(:,1)).*(1-S(:,4))-S(:,2).*S(:,3))./(2.*S(:,2));
D = ((1-S(:,1)).*(1+S(:,4))+S(:,2).*S(:,3))./(2.*S(:,2));
```

```
%          Tranzistörün Eşdeğer Gürültü Direnci
```

```
Rn =In(:,14).*Const(2);
```

```
%          Minimum Gürültü Faktörü
```

```
Fmin =10.^(In(:,11).*0.1);
```

```
%          Thetaların Hesabı
```

```
Theta=(2*pi/(Const(3)))*kron(1*1e-2,In(:,2)*1e9); % 1 cm. cinsinden ve frekans GHz
cinsinden olduğu için 0.01 ve 1*10^9 ile çarpılıyor.
```

```
%          Kazancın Hesaplanması
```

```
for n=1:12
```

```
E1=[cos(Theta(n,1)),i.*z1.*sin(Theta(n,1));i.*sin(Theta(n,1))./z1,cos(Theta(n,1))];
E3=[cos(Theta(n,3)),i.*z3.*sin(Theta(n,3));i.*sin(Theta(n,3))./z3,cos(Theta(n,3))];
E5=[cos(Theta(n,5)),i.*z5.*sin(Theta(n,5));i.*sin(Theta(n,5))./z5,cos(Theta(n,5))];
E2=[1,0;-i./(z2.*tan(Theta(n,2))),1];
E4=[1,0;-i./(z4.*tan(Theta(n,4))),1];
E6=[1,0;-i./(z6.*tan(Theta(n,6))),1];
```

```
M=E1*E2*E3*[A(n),B(n);C(n),D(n)]*E4*E5*E6;
```

```
Gt(n)=4*Rs_const*Rl_const/(
abs(M(1,1).*Zl_const+M(1,2)+Zs_const*(M(2,1).*Zl_const+M(2,2))).^2)
```

```
Zin(n)=(M(1,1).*Zl_const+M(1,2))./(M(2,1).*Zl_const+M(2,2))
```

```
MRin(n)=abs((Zs_const-conj(Zin(n)))/(Zs_const+Zin(n)))
```

```
Zout=(M(1,2)+M(2,2).*Zs_const)/(M(1,1)+M(2,1).*Zs_const)
Rout=real(Zout);
```

```
Gav(n)=4*Rs_const*Rout/(
abs(M(1,1).*conj(Zout)+M(1,2)+Zs_const*(M(2,1).*conj(Zout)+M(2,2))).^2);
```

```
end
```

```

ortmr=MRin(1)+MRin(2)+MRin(3)+MRin(4)+MRin(5)+MRin(6)+MRin(7)+MRin(8)+MRin
(9)+MRin(10);
ortmrin=ortmr/10;
ortgt=Gt(1)+Gt(2)+Gt(3)+Gt(4)+Gt(5)+Gt(6)+Gt(7)+Gt(8)+Gt(9)+Gt(10);
ortgt=ortgt/10;

% İlk Katın Kaynak Empedansı
Zs(:,1)=Zs_const*ones(n,1);

% İkinci Katın Kaynak Empedansı
Zs(:,2)=(i.*z1.*sin(Theta(:,1))+cos(Theta(:,1)).*Zs(:,1))./(cos(Theta(:,1))+i.*sin(Theta(:,1)).*
Zs(:,1))./z1);

% Üçüncü Katın Kaynak Empedansı
Zs(:,3)=Zs(:,2)./(1+(-i./z2./(tan(Theta(:,2)))).*Zs(:,2));

% Dördüncü Katın Kaynak Empedansı
Zs(:,4)=(i.*z3.*sin(Theta(:,3))+cos(Theta(:,3)).*Zs(:,3))./(cos(Theta(:,3))+i.*sin(Theta(:,3)).*
Zs(:,3))./z3);

% İlk Katın Kaynak Empedansı
%Zs(:,1)=Zs_const*ones(n,1);

% İkinci Katın Kaynak Empedansı
%Zs(:,2)=(i.*z1.*sin(Theta(:,1))+cos(Theta(:,1)).*Zs(:,1))./(cos(Theta(:,1))+i.*sin(Theta(:,1)).*
*Zs(:,1))./z1);

% Üçüncü Katın Kaynak Empedansı
%Zs(:,3)=(i.*z2.*sin(Theta(:,2))+cos(Theta(:,2)).*Zs(:,2))./(cos(Theta(:,2))+i.*sin(Theta(:,2)).*
*Zs(:,2))./z2);

% Dördüncü Katın Kaynak Empedansı
%Zs(:,4)=Zs(:,3)./(1+(-i./z3.*(tan(Theta(:,3)))).*Zs(:,3));

% Transistör İçin Optimum Kaynak Empedansı
Zopt=50*(1+G_opt)./(1-G_opt);
Rs4=real(Zs(:,4));
Ropt=real(Zopt);

```

% Giriş Uydurma Devresi Pasif Ve Kayıpsız Olduğundaki Transduser Güç Kazancı

GT=4*Rs4.*Ropt./((abs(Zs(:,4))+conj(Zopt))).^2;
LT=1./GT;

Noise=(4*Rn.*Ropt.*(1./GT-1))./((abs(Zopt)).^2)+Fmin

% Gürültü Hata İfadesi

EN=1+(Fr*ones(12,1)-Fmin).*((abs(Zopt)).^2)./(4*Rn.*Ropt)-(1./GT)

% İlk Katın Kaynak Empedansı

Zm(:,1)=-conj(Zs_const)*ones(12,1);
Zs(:,1)=Zs_const*ones(12,1);
AA=abs((cos(Theta(:,1))+i.*sin(Theta(:,1)).*Zm(:,1)./z1)./(cos(Theta(:,1))+i.*sin(Theta(:,1)).*Zs(:,1)./z1));

% İkinci Katın Kaynak Empedansı

Zm(:,2)=(i.*z1.*sin(Theta(:,1))-cos(Theta(:,1)).*conj(Zs(:,1)))./(cos(Theta(:,1))-i.*sin(Theta(:,1)).*conj(Zs(:,1))./z1);
Zs(:,2)=(i.*z1.*sin(Theta(:,1))+cos(Theta(:,1)).*Zs(:,1))./(cos(Theta(:,1))+i.*sin(Theta(:,1)).*Zs(:,1)./z1);
BB=abs((1+(-i.*Zm(:,2))./z2./tan(Theta(:,2))))/(1+(-i.*Zs(:,2))./z2./tan(Theta(:,2)))));

% Üçüncü Katın Kaynak Empedansı

Zm(:,3)=-conj(Zs(:,2))./(1-(-i.*conj(Zs(:,2))./z2./tan(Theta(:,2)))));
Zs(:,3)=(Zs(:,2))./(1+(-i.*Zs(:,2))./z2./tan(Theta(:,2)))));
CC=abs((cos(Theta(:,3))+i.*sin(Theta(:,3)).*Zm(:,3))./z3)./(cos(Theta(:,3))+i.*sin(Theta(:,3)).*Zs(:,3))./z3));

% Dördüncü Katın Kaynak Empedansı

Zm(:,4)=(i.*z3.*sin(Theta(:,3))-cos(Theta(:,3)).*conj(Zs(:,3)))./(cos(Theta(:,3))-i.*sin(Theta(:,3)).*conj(Zs(:,3))./z3);
Zs(:,4)=(i.*z3.*sin(Theta(:,3))+cos(Theta(:,3)).*Zs(:,3))./(cos(Theta(:,3))+i.*sin(Theta(:,3)).*Zs(:,3))./z3);
DD=abs((A+C.*Zm(:,4))./(A-C.*Zs(:,4))));

% Beşinci Katın Kaynak Empedansı

Zm(:,5)=(B-D.*conj(Zs(:,4)))./(A-C.*conj(Zs(:,4))));
Zs(:,5)=(B+D.*Zs(:,4))./(A-C.*Zs(:,4));
EE=abs((cos(Theta(:,4))+i.*sin(Theta(:,4)).*Zm(:,5))./z4)./(cos(Theta(:,4))+i.*sin(Theta(:,4)).*Zs(:,5))./z4));

% Altıncı Katın Kaynak Empedansı

```

Zm(:,6)=(i.*z4.*sin(Theta(:,4))-cos(Theta(:,4)).*conj(Zs(:,5)))/(cos(Theta(:,4))-
i.*sin(Theta(:,4)).*conj(Zs(:,5))/z4));
Zs(:,6)=(i.*z4.*sin(Theta(:,4))+cos(Theta(:,4)).*Zs(:,5))/(cos(Theta(:,4))+i.*sin(Theta(:,4)).*
Zs(:,5))/z4);
FF=abs( ( 1+(-i.*Zm(:,6))/z5./tan(Theta(:,5))) )./(1+(-i.*Zs(:,6))/z5./tan(Theta(:,5)))));

```

```
% Yedinci Katın Kaynak Empedansı
```

```

Zm(:,7)=(-conj(Zs(:,6)))/(1-(-i.*conj(Zs(:,6))/z5./tan(Theta(:,5)))));
Zs(:,7)=(Zs(:,6))/(1+(-i.*Zs(:,6))/z5./tan(Theta(:,5)))));
GG=abs((cos(Theta(:,6))+i.*sin(Theta(:,6)).*Zm(:,7))/z6)/(cos(Theta(:,6))+i.*sin(Theta(:,6)).*
Zs(:,7))/z6));

```

```
% Yedinci Katın Giriş Empedansı
```

```

Zi7=z6.*(Zl_const+i.*z6.*tan(Theta(:,6)))/(z6+i.*Zl_const*tan(Theta(:,6)));
HH=abs((Zi7+Zm(:,7))/(Zi7+Zs(:,7)));

```

```
roi=AA.*BB.*CC.*DD.*EE.*FF.*GG.*HH;
```

```
roir=(Const(5)-1)/(Const(5)+1);
```

```
EVi=log(roir.^2)*ones(12,1)-log(MRin.^2);
```

```
% Sistemin toplam hata fonksiyonunu hesaplayan program
```

```
options(14)=100*n;
```

```
toplamlam=exp(-Const(6).*EN)+exp(-Const(7).*EVi)+Const(8).*((Gt-Gref*ones(12,1)).^2);
```

```
Gr=15.86;
```

```
%F=sum(toplam)
```

```
%toplamlam=exp(-Const(6).*EN)+exp(-Const(7).*EVi)+Const(8).*((Gt-Gref*ones(12,1)).^2);
```

```
%F=ortmrin+(exp(0.01*abs(ortgt-Gref)))*0.4
```

```

%F=((MRin(2)-MRinref)+2*(MRin(3)-MRinref)+2*(MRin(4)-MRinref)+2*(MRin(5)-
MRinref)+2*(MRin(6)-MRinref)+2*(MRin(7)-MRinref)+2*(MRin(8)-
MRinref)+(MRin(9)-MRinref)+(MRin(10)-MRinref)+(MRin(11)-
MRinref)+(exp(0.01*abs((Gt(1)-Gref)+(Gt(2)-Gref)+(Gt(3)-Gref)+2.5*(Gt(4)-
Gref)+2.5*(Gt(5)-Gref)+3.5*(Gt(6)-Gref)+3*(Gt(7)-Gref)+3*(Gt(8)-
Gref)+2.5*(Gt(9)-Gref)+2*(Gt(10)-Gref)))))*80

```

```
%F=ortmrin+(exp(0.1*abs(ortgt-Gref))-1)*0.345
```

```
%F=ortmrin*1.1+(exp(0.1*abs(ortgt-Gref))-1)*0.4
```

```

%F=ortmrin+(exp(0.01*abs(7*(Gt(1)-Gr)+10*(Gt(2)-Gr)+6*(Gt(3)-Gr)+5*(Gt(4)-
Gr)+5*(Gt(5)-Gr)+8*(Gt(6)-Gr)+8*(Gt(7)-Gr)+5*(Gt(8)-Gr)+5*(Gt(9)-
Gr)+4*(Gt(10)-Gr)))))*5.5

```

```

F=ortmrin+(exp(0.01*abs(3*(Gt(1)-Gr)+3*(Gt(2)-Gr)+4*(Gt(3)-Gr)+5*(Gt(4)-Gr)+5*(Gt(5)-
Gr)+8*(Gt(6)-Gr)+8*(Gt(7)-Gr)+5*(Gt(8)-Gr)+5*(Gt(9)-Gr)+6*(Gt(10)-
Gr)))))*12

```

```
%F=sum(toplam)
```

MATCOMP Yardımcı Programı

```
function Z=MATCOMP(A,Theta)
```

% Z=MATCOMP(A,Theta) Genlik ve Açı ile Kompleks sayıyı üretir.

% Theta derece cinsinden girilecek. A, Theta vektör veya matris te olabilir.

if (nargin==2)

```
error('wrong number of input arguments');
```

```
elseif (size(A)~=size(Theta))
```

```
error('matrix dimensions must agree');
```

else

$$Z=A.*\exp(i.*\text{Theta.}/180*\pi);$$

end

Dosya_Inp.csv yardımcı data bilgileri (NE329S01)

1; 2; 0.955;	-25.30;	5.347;	150.00;	0.027;	73.3;	0.546;	-21.3;	0.26;	0.93;	14.0;	0.38
2; 3; 0.91;	-38.50;	5.210;	135.70;	0.039;	64.8;	0.526;	-32.0;	0.26;	0.88;	19.6;	0.35
3; 4; 0.853;	-51.30;	5.015;	121.70;	0.049;	55.4;	0.498;	-41.2;	0.27;	0.80;	29.0;	0.33
4; 5; 0.791;	-63.50;	4.817;	108.70;	0.056;	47.4;	0.476;	-49.0;	0.28;	0.71;	39.6;	0.28
5; 6; 0.741;	-74.60;	4.688;	96.60;	0.061;	42.0;	0.453;	-54.9;	0.29;	0.65;	48.0;	0.25
6; 7; 0.657;	-85.80;	4.608;	83.20;	0.071;	34.5;	0.422;	-66.4;	0.29;	0.58;	56.7;	0.22
7; 8; 0.558;	-101.30;	4.484;	69.50;	0.076;	29.0;	0.368;	-76.0;	0.31;	0.49;	72.0;	0.18
8; 9; 0.497;	-122.70;	4.349;	55.70;	0.084;	21.5;	0.309;	-87.3;	0.32;	0.40;	89.4;	0.13
9; 10; 0.464;	-142.70;	4.183;	42.10;	0.092;	14.9;	0.248;	-104.1;	0.33;	0.36;	102.0;	0.11
10; 11; 0.443;	-163.10;	4.025;	28.70;	0.100;	6.2;	0.198;	-123.0;	0.33;	0.30;	116.8;	0.09
11; 12; 0.426;	172.30;	3.848;	15.00;	0.106;	-0.5;	0.154;	-140.5;	0.35;	0.27;	139.0;	0.08
12; 13; 0.433;	148.40;	3.627;	1.10;	0.109;	-8.3;	0.123;	-170.8;	0.38;	0.24;	163.5;	0.06
13; 14; 0.468;	127.40;	3.396;	-12.60;	0.111;	-15.0;	0.129;	145.9;	0.43;	0.24;	-176.0;	0.07
14; 15; 0.525;	106.80;	3.149;	-25.70;	0.112;	-22.7;	0.190;	116.0;	0.48;	0.25;	-153.0;	0.07
15; 16; 0.55;	93.00;	2.895;	-38.40;	0.112;	-28.5;	0.274;	97.8;	0.57;	0.30;	-122.0;	0.10
16; 17; 0.631;	82.50;	2.618;	-51.50;	0.110;	-35.0;	0.360;	82.6;	0.66;	0.39;	-82.6;	0.16
17; 17.5; 0.661;	76.50;	2.490;	-57.40;	0.111;	-37.6;	0.388;	77.4;	0.70;	0.43;	-68.6;	0.19
18; 18; 0.685;	71.00;	2.364;	-63.60;	0.110;	-38.9;	0.414;	72.4;	0.73;	0.47;	-58.0;	0.22

Ek 3 Genetik Algoritma Optimizasyon Programları

MURGAIMC_CIS

```

hata_bayragi = 0;
clc;
while hata_bayragi == 0,
    fprintf('tekrar uretim\n');
    clear;

    for r1=1:3
        enaz(r1)=0.2;
        encok(r1)=15;
    end
    for r1=4:6
        enaz(r1)=90;
        encok(r1)=150;
    end
    bs=28;
    eski=0;
    kromozom = zeros([20,7]);
    kromozomyed = zeros([60,7]);
    hata_bayragi=1;
    jenerasyon=0;
    rand('state',sum(100*clock));           %Randomizer baslatildi
    pdf = 1:20;
    pdf = pdf*0;
    for n= 1:20,
        pdf(n) = (21-n)/(10*21);
    end
    for n = 1:100,
        for r1=1:6
            kromozomi(n,r1) = enaz(r1)+(encok(r1)-enaz(r1))*rand;           % x
        end
        kromozomi(n,7) = imc_cis([kromozomi(n,1:6,:) ]); % hata fonksiyonu degeri
    end

    kromozomi = sortrows(kromozomi,7);
    kromozom=kromozomi(1:20,:);

    ortalama(1) = mean(kromozom(:,7));
    min_deger(1) = kromozom(1,7);

    for yyy=1:500
        fprintf('iterasyon');
        minimum_deger = kromozom(1,7,:);

        jenerasyon = jenerasyon+1;
        %rulet carki
        for os=1:20

```

```

    olasiliklar(os) = (((abs(kromozom(os,7))+0.0001)\sum(kromozom(:,7))));
end;
%rulet carki
for rca=1:20
    for rcaa=1:20
        rastgele(rcaa,2)=rand*olasiliklar(rcaa)*1000;
        rastgele(rcaa,1)=rcaa;
    end;
    rastgele=sortrows(rastgele,2);
    rula(rca)=rastgele(20,1);
end;
for rcb=1:20
    for rcbb=1:20
        rastgele(rcbb,2)=rand*olasiliklar(rcbb)*1000;
        rastgele(rcbb,1)=rcbb;
    end;
    rastgele=sortrows(rastgele,2);
    rulb(rcb)=rastgele(20,1);
end;

cri=0;

for cr=1:2:39
    cri=cri+1;
    carp=fix(rand*(6))+1;
    for cry=1:6
        if carp<cry

            kromozomcross(cr,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(c
ry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),ena
z(cry)));
        else

            kromozomcross(cr,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(c
ry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz
(cry)));

        end
    end
    kromozomcross(cr,7) = imc_cis([kromozomcross(cr,1:6,:) ]); % hata fonksiyonu
    degeri
    kromozomcross(cr+1,7) = imc_cis([kromozomcross(cr+1,1:6,:) ]); % hata fonksiyonu
    degeri
end
end

```

```

for mut=1:20
    carp1=fix(rand*(6))+1;
    carp2=fix(rand*(6))+1;

    kromozomyed(mut+60,:)=kromozom(rula(cry),:);
    kromozomyed(mut+60,carp1)=enaz(carp1)+(encok(carp1)-enaz(carp1))*rand;
    kromozomyed(mut+60,carp2)=enaz(carp2)+(encok(carp2)-enaz(carp2))*rand;

    kromozomyed(mut+60,7)=imc_cis([kromozomyed(mut+60,1:6,:) ]); % hata
    fonksiyonu degeri
end

kromozomyed(1:20,:)=kromozom;
kromozomyed(21:60,:)=kromozomcross;
kromozomyed = sortrows(kromozomyed,7);
kromozom = kromozomyed(1:20,:);
min_deger(jenerasyon)=kromozom(1,7);
fprintf('Hata : %f %f\n', kromozom(1,7,:),jenerasyon);
kromozom = sortrows(kromozom,7);

ortalama(jenerasyon) = mean(kromozom(:,7));
min_deger(jenerasyon) = kromozom(1,7,:);
if not(minimum_deger == min_deger(jenerasyon))
    eski=0;
end
if minimum_deger == min_deger(jenerasyon)
fprintf('*****');
    for r1=3:19;
s=fix(rand*6)+1;
        %for r2=1:6
            kromozom(r1,s)=enaz(s)+(encok(s)-enaz(s))*rand;
        %end;
        kromozom(r1,7)=imc_cis([kromozom(r1,1:6,:) ]);
    end
    eski = eski+1;
end;
if (minimum_deger == min_deger(jenerasyon)) & (eski>=3)
    for r1=3:19;
        for r2=1:6
            kromozom(r1,r2)=enaz(r2)+(encok(r2)-enaz(r2))*rand;
        end;
        kromozom(r1,7)=imc_cis([kromozom(r1,1:6,:) ]);
    end

%hata_bayragi = 0;
eski=0;
%break;
imc_cis([kromozom(1,1:6)]);
end;

hata_bayragi = 1;

```

```
end; % jenerasyon dongusunun bitisi

end; % hata kontrol dongusunun bitisi
imc_cis2([kromozom(1,1:6)])

fprintf('Optimizasyon Sonuclari imc_cis \n');

fprintf('L1 = %3.20f \n', (kromozom(1,1,:)));
fprintf('L2 = %3.20f \n', (kromozom(1,2,:)));
fprintf('L3 = %3.20f \n', (kromozom(1,3,:)));
fprintf('Z1 = %3.20f \n', (kromozom(1,4,:)));
fprintf('Z2 = %3.20f \n', (kromozom(1,5,:)));
fprintf('Z3 = %3.20f \n', (kromozom(1,6,:)));

fprintf('jenerasyon = %f \n', jenerasyon);
fprintf('hata = %3.20f \n', kromozom(1,7,:));
```



MURGAIMC_CISP

```

hata_bayragi = 0;
clc;
while hata_bayragi == 0,
    fprintf('tekrar uretim\n');
    clear;

    for r1=1:3
        enaz(r1)=0.5;
        encok(r1)=14;
    end
    for r1=4:6
        enaz(r1)=30;
        encok(r1)=70;
    end
    bs=28;
    eski=0;
    kromozom = zeros([20,7]);
    kromozomyed = zeros([60,7]);
    hata_bayragi=1;
    jenerasyon=0;
    rand('state',sum(100*clock));           %Randomizer baslatildi
    pdf = 1:20;
    pdf = pdf*0;
    for n= 1:20,
        pdf(n) = (21-n)/(10*21);
    end
    for n = 1:100,
        for r1=1:6
            kromozomi(n,r1) = enaz(r1)+(encok(r1)-enaz(r1))*rand;           % x
        end
        kromozomi(n,7) = imc_cisp([kromozomi(n,1:6,:) ]); % hata fonksiyonu degeri
    end
    kromozomi = sortrows(kromozomi,7);
    kromozom=kromozomi(1:20,:);

    ortalama(1) = mean(kromozom(:,7));
    min_deger(1) = kromozom(1,7);

    for yyy=1:500;
        fprintf('iterasyon');
        minimum_deger = kromozom(1,7,:);

        jenerasyon = jenerasyon+1;
        %rulet carki
        for os=1:20
            olasiliklar(os) = (((abs(kromozom(os,7))+0.0001)\sum(kromozom(:,7))));

```

```

end;
%rulet carki
for rca=1:20
    for rcaa=1:20
        rastgele(rcaa,2)=rand*olasiliklar(rcaa)*1000;
        rastgele(rcaa,1)=rcaa;
    end;
    rastgele=sortrows(rastgele,2);
    rula(rca)=rastgele(20,1);
end;
for rcb=1:20
    for rcbb=1:20
        rastgele(rcbb,2)=rand*olasiliklar(rcbb)*1000;
        rastgele(rcbb,1)=rcbb;
    end;
    rastgele=sortrows(rastgele,2);
    rulb(rcb)=rastgele(20,1);
end;

cri=0;

for cr=1:2:39
    cri=cri+1;
    carp=fix(rand*(6))+1;
    for cry=1:6
        if carp<cry

            kromozomcross(cr,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(cry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(cry)));
        else

            kromozomcross(cr,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(cry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(cry)));

        end
    end
    kromozomcross(cr,7) = imc_cisp([kromozomcross(cr,1:6,:) ]); % hata fonksiyonu
    degeri
    kromozomcross(cr+1,7) = imc_cisp([kromozomcross(cr+1,1:6,:) ]); % hata
    fonksiyonu degeri
end

for mut=1:20

```

```

carp1=fix(rand*(6))+1;
carp2=fix(rand*(6))+1;

kromozomyed(mut+60,:)=kromozom(rula(cry),:);
kromozomyed(mut+60,carp1)=enaz(carp1)+(encok(carp1)-enaz(carp1))*rand;
kromozomyed(mut+60,carp2)=enaz(carp2)+(encok(carp2)-enaz(carp2))*rand;

kromozomyed(mut+60,7)=imc_cisp([kromozomyed(mut+60,1:6,:) ]); % hata
fonksiyonu degeri
end

kromozomyed(1:20,:)=kromozom;
kromozomyed(21:60,:)=kromozomcross;
kromozomyed = sortrows(kromozomyed,7);
kromozom = kromozomyed(1:20,:);
min_deger(jenerasyon)=kromozom(1,7);
fprintf('Hata : %f %f \n', kromozom(1,7,:),jenerasyon);
kromozom = sortrows(kromozom,7);

ortalama(jenerasyon) = mean(kromozom(:,7));
min_deger(jenerasyon) = kromozom(1,7,:);
if not(minimum_deger == min_deger(jenerasyon))
    eski=0;
end
if minimum_deger == min_deger(jenerasyon)
fprintf('*****');
    for r1=3:19;
s=fix(rand*6)+1;
        %for r2=1:6
            kromozom(r1,s)=enaz(s)+(encok(s)-enaz(s))*rand;
            %end;
            kromozom(r1,7)=imc_cisp([kromozom(r1,1:6,:) ]);
        end
        eski = eski+1;
    end;
if (minimum_deger == min_deger(jenerasyon)) & (eski>=3)
    for r1=3:19;
        for r2=1:6
            kromozom(r1,r2)=enaz(r2)+(encok(r2)-enaz(r2))*rand;
        end;
        kromozom(r1,7)=imc_cisp([kromozom(r1,1:6,:) ]);
    end

%hata_bayragi = 0;
eski=0;
%break;
imc_cisp([kromozom(1,1:6)]);
end;

hata_bayragi = 1;

```

```
end; % jenerasyon dongusunun bitisi

end; % hata kontrol dongusunun bitisi
imc_cisp2([kromozom(1,1:6)])

fprintf('Optimizasyon Sonuclari imc cis p \n');

fprintf('L1 = %3.20f \n', (kromozom(1,1,:)));
fprintf('L2 = %3.20f \n', (kromozom(1,2,:)));
fprintf('L3 = %3.20f \n', (kromozom(1,3,:)));
fprintf('Z1 = %3.20f \n', (kromozom(1,4,:)));
fprintf('Z2 = %3.20f \n', (kromozom(1,5,:)));
fprintf('Z3 = %3.20f \n', (kromozom(1,6,:)));

fprintf('jenerasyon = %f \n', jenerasyon);
fprintf('hata = %3.20f \n', kromozom(1,7,:));
```



MURGAIMC_CIS_PS_L

```

hata_bayragi = 0;
clc;
while hata_bayragi ==0,
    fprintf('tekrar uretim\n');
    clear;

    enaz(1)=0;
    encok(1)=0;
    enaz(4)=50;
    encok(4)=50;
    for r1=2:3
        enaz(r1)=0.5;
        encok(r1)=5;
    end
    for r1=5:6
        enaz(r1)=50;
        encok(r1)=150;
    end

    bs=28;
    eski=0;
    kromozom = zeros([20,7]);
    kromozomyed = zeros([60,7]);
    hata_bayragi=1;
    jenerasyon=0;
    rand('state',sum(100*clock));    %Randomizer baslatildi
    pdf = 1:20;
    pdf = pdf*0;
    for n= 1:20,
        pdf(n) = (21-n)/(10*21);
    end
    for n = 1:100,
        for r1=2:6
            kromozomi(n,1)=0;
            kromozomi(n,r1) = enaz(r1)+(encok(r1)-enaz(r1))*rand;    % x
        end
        kromozomi(n,7) = imc_cis([kromozomi(n,1:6,:) ]); % hata fonksiyonu degeri
    end

    kromozomi = sortrows(kromozomi,7);
    kromozom=kromozomi(1:20,:);

    ortalama(1) = mean(kromozom(:,7));
    min_deger(1) = kromozom(1,7);

    for yyy=1:100
        fprintf('iterasyon');
        minimum_deger = kromozom(1,7,:);

```

```

jenerasyon = jenerasyon+1;
%rulet carki
for os=1:20
    olasiliklar(os) = (((abs(kromozom(os,7))+0.0001)\sum(kromozom(:,7))));
end;
%rulet carki
for rca=1:20
    for rcaa=1:20
        rastgele(rcaa,2)=rand*olasiliklar(rcaa)*1000;
        rastgele(rcaa,1)=rcaa;
    end;
    rastgele=sortrows(rastgele,2);
    rula(rca)=rastgele(20,1);
end;
for rcb=1:20
    for rcbb=1:20
        rastgele(rcbb,2)=rand*olasiliklar(rcbb)*1000;
        rastgele(rcbb,1)=rcbb;
    end;
    rastgele=sortrows(rastgele,2);
    rulb(rcb)=rastgele(20,1);
end;

cri=0;

for cr=1:2:39
    cri=cri+1;
    carp=fix(rand*(6))+1;
    for cry=1:6
        if carp<cry

            kromozomcross(cr,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(cry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(cry)));
        else

            kromozomcross(cr,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(cry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(cry)));

        end
    end
    kromozomcross(cr,7) = imc_cis([kromozomcross(cr,1:6,:) ]); % hata fonksiyonu
    degeri
    kromozomcross(cr+1,7) = imc_cis([kromozomcross(cr+1,1:6,:) ]); % hata fonksiyonu
    degeri

```

```

end

for mut=1:20
    carp1=fix(rand*(6))+1;
    carp2=fix(rand*(6))+1;

    kromozomyed(mut+60,:)=kromozom(rula(cry),:);
    kromozomyed(mut+60,carp1)=enaz(carp1)+(encok(carp1)-enaz(carp1))*rand;
    kromozomyed(mut+60,carp2)=enaz(carp2)+(encok(carp2)-enaz(carp2))*rand;

    kromozomyed(mut+60,7)=imc_cis([kromozomyed(mut+60,1:6,:)]); % hata
    fonksiyonu degeri
end

kromozomyed(1:20,:)=kromozom;
kromozomyed(21:60,:)=kromozomcross;
kromozomyed = sortrows(kromozomyed,7);
kromozom = kromozomyed(1:20,:);
min_deger(jenerasyon)=kromozom(1,7);
fprintf('Hata : %f %f \n', kromozom(1,7,:),jenerasyon);

kromozom = sortrows(kromozom,7);

ortalama(jenerasyon) = mean(kromozom(:,7));
min_deger(jenerasyon) = kromozom(1,7,:);
if not(minimum_deger == min_deger(jenerasyon))
    eski=0;
end
if minimum_deger == min_deger(jenerasyon)
fprintf('*****');
    for r1=3:19;
s=fix(rand*6)+1;
        %for r2=1:6
            kromozom(r1,s)=enaz(s)+(encok(s)-enaz(s))*rand;
        %end;
        kromozom(r1,7)=imc_cis([kromozom(r1,1:6,:)]);
    end
    eski = eski+1;
end;
if (minimum_deger == min_deger(jenerasyon)) & (eski>=3)
    for r1=3:19;
        for r2=1:6
            kromozom(r1,r2)=enaz(r2)+(encok(r2)-enaz(r2))*rand;
        end;
        kromozom(r1,7)=imc_cis([kromozom(r1,1:6,:)]);
    end

%hata_bayragi = 0;
eski=0;

```

```

%break;
imc_cis([kromozom(1,1:6)]);
end;

    hata_bayragi = 1;

end; % jenerasyon dongusunun bitisi

end; % hata kontrol dongusunun bitisi
imc_cis2 ([kromozom(1,1:6)])

fprintf('Optimizasyon Sonuclari imc cis ps l \n');

fprintf('L1*   = %3.20f \n', (kromozom(1,1,:)));
fprintf('L2   = %3.20f \n', (kromozom(1,2,:)));
fprintf('L3   = %3.20f \n', (kromozom(1,3,:)));
fprintf('Z1*   = %3.20f \n', (kromozom(1,4,:)));
fprintf('Z2   = %3.20f \n', (kromozom(1,5,:)));
fprintf('Z3   = %3.20f \n', (kromozom(1,6,:)));

fprintf('jenerasyon = %f \n', jenerasyon);
fprintf('hata     = %3.20f \n',kromozom(1,7,:));

```

MURGAIMC_CIS_SP_L

```

hata_bayragi = 0;
clc;
while hata_bayragi ==0,
    fprintf('tekrar uretim\n');
    clear;

    enaz(3)=0;
    encok(3)=0;
    enaz(6)=50;
    encok(6)=50;
    for r1=1:2
        enaz(r1)=0.5;
        encok(r1)=15;
    end
    for r1=4:5
        enaz(r1)=50;
        encok(r1)=150;
    end

    bs=28;
    eski=0;
    kromozom = zeros([20,7]);
    kromozomyed = zeros([60,7]);
    hata_bayragi=1;
    jenerasyon=0;
    rand('state',sum(100*clock));    %Randomizer baslatildi
    pdf = 1:20;
    pdf = pdf*0;
    for n= 1:20,
        pdf(n) = (21-n)/(10*21);
    end
    for n = 1:100,
        for r1=2:6
            kromozomi(n,1)=0;
            kromozomi(n,r1) = enaz(r1)+(encok(r1)-enaz(r1))*rand;    % x
        end
        kromozomi(n,7) = imc_cis([kromozomi(n,1:6,:) ]); % hata fonksiyonu degeri
    end

    kromozomi = sortrows(kromozomi,7);
    kromozom=kromozomi(1:20,:);

    ortalama(1) = mean(kromozom(:,7));
    min_deger(1) = kromozom(1,7);

    for yyy=1:800
        fprintf('iterasyon');
        minimum_deger = kromozom(1,7,:);

```

```

jenerasyon = jenerasyon+1;
%rulet carki
for os=1:20
    olasiliklar(os) = (((abs(kromozom(os,7))+0.0001)\sum(kromozom(:,7))));
end;
%rulet carki
for rca=1:20
    for rcaa=1:20
        rastgele(rcaa,2)=rand*olasiliklar(rcaa)*1000;
        rastgele(rcaa,1)=rcaa;
    end;
    rastgele=sortrows(rastgele,2);
    rula(rca)=rastgele(20,1);
end;
for rcb=1:20
    for rcbb=1:20
        rastgele(rcbb,2)=rand*olasiliklar(rcbb)*1000;
        rastgele(rcbb,1)=rcbb;
    end;
    rastgele=sortrows(rastgele,2);
    rulb(rcb)=rastgele(20,1);
end;

cri=0;

for cr=1:2:39
    cri=cri+1;
    carp=fix(rand*(6))+1;
    for cry=1:6
        if carp<cry

            kromozomcross(cr,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(cry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(cry)));
        else

            kromozomcross(cr,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(cry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(cry)));

        end
    end
    kromozomcross(cr,7) = imc_cis([kromozomcross(cr,1:6,:) ]); % hata fonksiyonu degeri
    kromozomcross(cr+1,7) = imc_cis([kromozomcross(cr+1,1:6,:) ]); % hata fonksiyonu degeri
end

```

end

for mut=1:20

carp1=fix(rand*(6))+1;

carp2=fix(rand*(6))+1;

kromozomyed(mut+60,:)=kromozom(rula(cry),:);

kromozomyed(mut+60,carp1)=enaz(carp1)+(encok(carp1)-enaz(carp1))*rand;

kromozomyed(mut+60,carp2)=enaz(carp2)+(encok(carp2)-enaz(carp2))*rand;

kromozomyed(mut+60,7)=imc_cis([kromozomyed(mut+60,1:6,:)]); % hata
fonksiyonu degeri

end

kromozomyed(1:20,:)=kromozom;

kromozomyed(21:60,:)=kromozomcross;

kromozomyed = sortrows(kromozomyed,7);

kromozom = kromozomyed(1:20,:);

min_deger(jenerasyon)=kromozom(1,7);

fprintf('Hata : %f %f\n', kromozom(1,7,:),jenerasyon);

kromozom = sortrows(kromozom,7);

ortalama(jenerasyon) = mean(kromozom(:,7));

min_deger(jenerasyon) = kromozom(1,7,:);

if not(minimum_deger == min_deger(jenerasyon))

eski=0;

end

if minimum_deger == min_deger(jenerasyon)

fprintf('*****');

for r1=3:19;

s=fix(rand*6)+1;

%for r2=1:6

kromozom(r1,s)=enaz(s)+(encok(s)-enaz(s))*rand;

%end;

kromozom(r1,7)=imc_cis([kromozom(r1,1:6,:)]);

end

eski = eski+1;

end;

if (minimum_deger == min_deger(jenerasyon)) & (eski>=3)

for r1=3:19;

for r2=1:6

kromozom(r1,r2)=enaz(r2)+(encok(r2)-enaz(r2))*rand;

end;

kromozom(r1,7)=imc_cis([kromozom(r1,1:6,:)]);

end

%hata_bayragi = 0;

eski=0;

```

%break;
%imc_cis([kromozom(1,1:6)]);
end;

    hata_bayragi = 1;

end; % jenerasyon dongusunun bitisi

end; % hata kontrol dongusunun bitisi
imc_cis2([kromozom(1,1:6)])

fprintf('Optimizasyon Sonuclari imc cis sp l \n');

fprintf('L1 = %3.20f\n', (kromozom(1,1,:)));
fprintf('L2 = %3.20f\n', (kromozom(1,2,:)));
fprintf('L3* = %3.20f\n', (kromozom(1,3,:)));
fprintf('Z1 = %3.20f\n', (kromozom(1,4,:)));
fprintf('Z2 = %3.20f\n', (kromozom(1,5,:)));
fprintf('Z3* = %3.20f\n', (kromozom(1,6,:)));

fprintf('jenerasyon = %f\n', jenerasyon);
fprintf('hata = %3.20f\n',kromozom(1,7,:));

```

MURGAOMC_CIS

```

hata_bayragi = 0;
clc;
while hata_bayragi ==0,
    fprintf('tekrar uretim\n');
    clear;

    for r1=1:3
        enaz(r1)=10;
        encok(r1)=15;
    end
    for r1=4:6
        enaz(r1)=88;
        encok(r1)=130;
    end
    bs=28;
    eski=0;
    kromozom = zeros([20,7]);
    kromozomyed = zeros([60,7]);
    hata_bayragi=1;
    jenerasyon=0;
    rand('state',sum(100*clock));    %Randomizer baslatildi
    pdf = 1:20;
    pdf = pdf*0;
    for n= 1:20,
        pdf(n) = (21-n)/(10*21);
    end
    for n = 1:100,
        for r1=1:6
            kromozomi(n,r1) = enaz(r1)+(encok(r1)-enaz(r1))*rand;    % x
        end
        kromozomi(n,7) = omc_cis([kromozomi(n,1:6,:) ]); % hata fonksiyonu degeri
    end

    kromozomi = sortrows(kromozomi,7);
    kromozom=kromozomi(1:20,:);

    ortalama(1) = mean(kromozom(:,7));
    min_deger(1) = kromozom(1,7);

    for yyy=1:1000 %hata 1 den buyukse jenerasyona devam et
        fprintf('iterasyon');
        minimum_deger = kromozom(1,7,:);

        jenerasyon = jenerasyon+1;
        %rulet carki
        for os=1:20
            olasiliklar(os) = (((abs(kromozom(os,7))+0.0001)\sum(kromozom(:,7))));
        end;
    end;

```

```

%rulet carki
for rca=1:20
    for rcaa=1:20
        rastgele(rcaa,2)=rand*olasiliklar(rcaa)*1000;
        rastgele(rcaa,1)=rcaa;
    end;
    rastgele=sortrows(rastgele,2);
    rula(rca)=rastgele(20,1);
end;
for rcb=1:20
    for rcbb=1:20
        rastgele(rcbb,2)=rand*olasiliklar(rcbb)*1000;
        rastgele(rcbb,1)=rcbb;
    end;
    rastgele=sortrows(rastgele,2);
    rulb(rcb)=rastgele(20,1);
end;

cri=0;

for cr=1:2:39
    cri=cri+1;
    carp=fix(rand*(6))+1;
    for cry=1:6
        if carp<cry

            kromozomcross(cr,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(cry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(cry)));
        else

            kromozomcross(cr,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(cry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(cry)));

        end
    end
    kromozomcross(cr,7) = omc_cis([kromozomcross(cr,1:6,:) ]); % hata fonksiyonu
    degeri
    kromozomcross(cr+1,7) = omc_cis([kromozomcross(cr+1,1:6,:) ]); % hata fonksiyonu
    degeri
end

for mut=1:20
    carp1=fix(rand*(6))+1;

```

```

carp2=fix(rand*(6))+1;

kromozomyed(mut+60,:)=kromozom(rula(cry),:);
kromozomyed(mut+60,carp1)=enaz(carp1)+(encok(carp1)-enaz(carp1))*rand;
kromozomyed(mut+60,carp2)=enaz(carp2)+(encok(carp2)-enaz(carp2))*rand;

kromozomyed(mut+60,7)=omc_cis([kromozomyed(mut+60,1:6,:) ]); % hata
fonksiyonu degeri
end

kromozomyed(1:20,:)=kromozom;
kromozomyed(21:60,:)=kromozomcross;
kromozomyed = sortrows(kromozomyed,7);
kromozom = kromozomyed(1:20,:);
min_deger(jenerasyon)=kromozom(1,7);
fprintf('Hata : %f %f \n', kromozom(1,7,:),jenerasyon);

kromozom = sortrows(kromozom,7);

ortalama(jenerasyon) = mean(kromozom(:,7));
min_deger(jenerasyon) = kromozom(1,7,:);
if not(minimum_deger == min_deger(jenerasyon))
    eski=0;
end
if minimum_deger == min_deger(jenerasyon)
fprintf('*****');
    for r1=3:19;
s=fix(rand*6)+1;
        %for r2=1:6
            kromozom(r1,s)=enaz(s)+(encok(s)-enaz(s))*rand;
        %end;
        kromozom(r1,7)=omc_cis([kromozom(r1,1:6,:) ]);
    end
    eski = eski+1;
end;
if (minimum_deger == min_deger(jenerasyon)) & (eski>=3)
    for r1=3:19;
        for r2=1:6
            kromozom(r1,r2)=enaz(r2)+(encok(r2)-enaz(r2))*rand;
        end;
        kromozom(r1,7)=omc_cis([kromozom(r1,1:6,:) ]);
    end

%hata_bayragi = 0;
eski=0;
%break;

end;

hata_bayragi = 1;

```

```
end; % jenerasyon dongusunun bitisi

end; % hata kontrol dongusunun bitisi
omc_cis2([kromozom(1,1:6)])

fprintf('Optimizasyon Sonuclari omc cis \n');

fprintf('L1 = %3.20f\n', (kromozom(1,1,:)));
fprintf('L2 = %3.20f\n', (kromozom(1,2,:)));
fprintf('L3 = %3.20f\n', (kromozom(1,3,:)));
fprintf('Z1 = %3.20f\n', (kromozom(1,4,:)));
fprintf('Z2 = %3.20f\n', (kromozom(1,5,:)));
fprintf('Z3 = %3.20f\n', (kromozom(1,6,:)));

fprintf('jenerasyon = %f\n', jenerasyon);
fprintf('hata = %3.20f\n', kromozom(1,7,:));
```



MURGAOMC_CISP

```

hata_bayragi = 0;
clc;
while hata_bayragi == 0,
    fprintf('tekrar uretim\n');
    clear;

    for r1=1:3
        enaz(r1)=1;
        encok(r1)=15;
    end
    for r1=4:6
        enaz(r1)=60;
        encok(r1)=120;
    end
    bs=28;
    eski=0;
    kromozom = zeros([20,7]);
    kromozomyed = zeros([60,7]);
    hata_bayragi=1;
    jenerasyon=0;
    rand('state',sum(100*clock));           %Randomizer baslatildi
    pdf = 1:20;
    pdf = pdf*0;
    for n= 1:20,
        pdf(n) = (21-n)/(10*21);
    end
    for n = 1:100,
        for r1=1:6
            kromozomi(n,r1) = enaz(r1)+(encok(r1)-enaz(r1))*rand;           % x
        end
        kromozomi(n,7) = omc_cisP([kromozomi(n,1:6,:) ]); % hata fonksiyonu degeri
    end

    kromozomi = sortrows(kromozomi,7);
    kromozom=kromozomi(1:20,:);

    ortalama(1) = mean(kromozom(:,7));
    min_deger(1) = kromozom(1,7);

    for yyy=1:500 %hata 1 den buyukse jenerasyona devam et
        fprintf('iterasyon');
        minimum_deger = kromozom(1,7,:);

        jenerasyon = jenerasyon+1;
        %rulet carki
        for os=1:20
            olasiliklar(os) = (((abs(kromozom(os,7))+0.0001)\sum(kromozom(:,7))));
        end;
    end;

```

```

%rulet carki
for rca=1:20
    for rcaa=1:20
        rastgele(rcaa,2)=rand*olasiliklar(rcaa)*1000;
        rastgele(rcaa,1)=rcaa;
    end;
    rastgele=sortrows(rastgele,2);
    rula(rca)=rastgele(20,1);
end;
for rcb=1:20
    for rcbb=1:20
        rastgele(rcbb,2)=rand*olasiliklar(rcbb)*1000;
        rastgele(rcbb,1)=rcbb;
    end;
    rastgele=sortrows(rastgele,2);
    rulb(rcb)=rastgele(20,1);
end;

cri=0;

for cr=1:2:39
    cri=cri+1;
    carp=fix(rand*(6))+1;
    for cry=1:6
        if carp<cry

            kromozomcross(cr,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(cry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(cry)));
        else

            kromozomcross(cr,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(cry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(cry)));

        end
    end
    kromozomcross(cr,7) = omc_cisP([kromozomcross(cr,1:6,:) ]); % hata fonksiyonu degeri
    kromozomcross(cr+1,7) = omc_cisP([kromozomcross(cr+1,1:6,:) ]); % hata fonksiyonu degeri
end

for mut=1:20
    carp1=fix(rand*(6))+1;

```

```

carp2=fix(rand*(6))+1;

kromozomyed(mut+60,:) =kromozom(rula(cry),:);
kromozomyed(mut+60,carp1) =enaz(carp1)+(encok(carp1)-enaz(carp1))*rand;
kromozomyed(mut+60,carp2) =enaz(carp2)+(encok(carp2)-enaz(carp2))*rand;

kromozomyed(mut+60,7) = omc_cisP([kromozomyed(mut+60,1:6,:) ]); % hata
fonksiyonu degeri
end

kromozomyed(1:20,:)=kromozom;
kromozomyed(21:60,:)=kromozomcross;
kromozomyed = sortrows(kromozomyed,7);
kromozom = kromozomyed(1:20,:);
min_deger(jenerasyon)=kromozom(1,7);
fprintf('Hata : %f %f \n', kromozom(1,7,:),jenerasyon);

kromozom = sortrows(kromozom,7);

ortalama(jenerasyon) = mean(kromozom(:,7));
min_deger(jenerasyon) = kromozom(1,7,:);
if not(minimum_deger == min_deger(jenerasyon))
    eski=0;
end
if minimum_deger == min_deger(jenerasyon)
fprintf('*****');
    for r1=3:19;
s=fix(rand*6)+1;
        %for r2=1:6
            kromozom(r1,s)=enaz(s)+(encok(s)-enaz(s))*rand;
        %end;
        kromozom(r1,7)=omc_cisP([kromozom(r1,1:6,:) ]);
    end
    eski = eski+1;
end;
if (minimum_deger == min_deger(jenerasyon)) & (eski>=3)
    for r1=3:19;
        for r2=1:6
            kromozom(r1,r2)=enaz(r2)+(encok(r2)-enaz(r2))*rand;
        end;
        kromozom(r1,7)=omc_cisP([kromozom(r1,1:6,:) ]);
    end

%hata_bayragi = 0;
eski=0;
%break;
omc_cisP([kromozom(1,1:6)]);
end;

hata_bayragi = 1;

```

```
end; % jenerasyon dongusunun bitisi

end; % hata kontrol dongusunun bitisi
omc_cisP2([kromozom(1,1:6)])

fprintf('Optimizasyon Sonuclari omc cis p \n');

fprintf('L1 = %3.20f \n', (kromozom(1,1,:)));
fprintf('L2 = %3.20f \n', (kromozom(1,2,:)));
fprintf('L3 = %3.20f \n', (kromozom(1,3,:)));
fprintf('Z1 = %3.20f \n', (kromozom(1,4,:)));
fprintf('Z2 = %3.20f \n', (kromozom(1,5,:)));
fprintf('Z3 = %3.20f \n', (kromozom(1,6,:)));

fprintf('jenerasyon = %f \n', jenerasyon);
fprintf('hata = %3.20f \n',kromozom(1,7,:));
```



MURGAOMC_CIS_PS_L

```

hata_bayragi = 0;
clc;
while hata_bayragi ==0,
    fprintf('tekrar uretim\n');
    clear;

    enaz(1)=0;
    encok(1)=0;
    enaz(4)=50;
    encok(4)=50;
    for r1=2:3
        enaz(r1)=1.1;
        encok(r1)=15;
    end
    for r1=5:6
        enaz(r1)=72;
        encok(r1)=135;
    end

    bs=28;
    eski=0;
    kromozom = zeros([20,7]);
    kromozomyed = zeros([60,7]);
    hata_bayragi=1;
    jenerasyon=0;
    rand('state',sum(100*clock));           %Randomizer baslatildi
    pdf = 1:20;
    pdf = pdf*0;
    for n= 1:20,
        pdf(n) = (21-n)/(10*21);
    end
    for n = 1:100,
        for r1=1:6
            kromozomi(n,r1) = enaz(r1)+(encok(r1)-enaz(r1))*rand;           % x
        end
        kromozomi(n,7) = omc_cis([kromozomi(n,1:6,:) ]); % hata fonksiyonu degeri
    end

    kromozomi = sortrows(kromozomi,7);
    kromozom=kromozomi(1:20,:);

    ortalama(1) = mean(kromozom(:,7));
    min_deger(1) = kromozom(1,7);

for yyy=1:1000 %hata 1 den buyukse jenerasyona devam et
    fprintf('.');
    minimum_deger = kromozom(1,7,:);

```

```

jenerasyon = jenerasyon+1;
%rulet carki
for os=1:20
    olasiliklar(os) = (((abs(kromozom(os,7))+0.0001)\sum(kromozom(:,7))));
end;
%rulet carki
for rca=1:20
    for rcaa=1:20
        rastgele(rcaa,2)=rand*olasiliklar(rcaa)*1000;
        rastgele(rcaa,1)=rcaa;
    end;
    rastgele=sortrows(rastgele,2);
    rula(rca)=rastgele(20,1);
end;
for rcb=1:20
    for rcbb=1:20
        rastgele(rcbb,2)=rand*olasiliklar(rcbb)*1000;
        rastgele(rcbb,1)=rcbb;
    end;
    rastgele=sortrows(rastgele,2);
    rulb(rcb)=rastgele(20,1);
end;

cri=0;

for cr=1:2:39
    cri=cri+1;
    carp=fix(rand*(6))+1;
    for cry=1:6
        if carp<cry

            kromozomcross(cr,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(cry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(cry)));
        else

            kromozomcross(cr,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(cry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(cry)));

        end
    end
    kromozomcross(cr,7) = omc_cis([kromozomcross(cr,1:6,:) ]); % hata fonksiyonu
    degeri
    kromozomcross(cr+1,7) = omc_cis([kromozomcross(cr+1,1:6,:) ]); % hata fonksiyonu
    degeri

```

```

end

for mut=1:20
    carp1=fix(rand*(6))+1;
    carp2=fix(rand*(6))+1;

    kromozomyed(mut+60,:)=kromozom(rula(cry),:);
    kromozomyed(mut+60,carp1)=enaz(carp1)+(encok(carp1)-enaz(carp1))*rand;
    kromozomyed(mut+60,carp2)=enaz(carp2)+(encok(carp2)-enaz(carp2))*rand;

    kromozomyed(mut+60,7)=omc_cis([kromozomyed(mut+60,1:6,:) ]); % hata
    fonksiyonu degeri
end

kromozomyed(1:20,:)=kromozom;
kromozomyed(21:60,:)=kromozomcross;
kromozomyed = sortrows(kromozomyed,7);
kromozom = kromozomyed(1:20,:);
min_deger(jenerasyon)=kromozom(1,7);
fprintf('Hata : %f %f \n', kromozom(1,7,:),jenerasyon);

kromozom = sortrows(kromozom,7);

ortalama(jenerasyon) = mean(kromozom(:,7));
min_deger(jenerasyon) = kromozom(1,7,:);
if not(minimum_deger == min_deger(jenerasyon))
    eski=0;
end
if minimum_deger == min_deger(jenerasyon)
fprintf('*****');
    for r1=3:19;
s=fix(rand*6)+1;
        %for r2=1:6
            kromozom(r1,s)=enaz(s)+(encok(s)-enaz(s))*rand;
        %end;
        kromozom(r1,7)=omc_cis([kromozom(r1,1:6,:) ]);
    end
    eski = eski+1;
end;
if (minimum_deger == min_deger(jenerasyon)) & (eski>=3)
    for r1=3:19;
        for r2=1:6
            kromozom(r1,r2)=enaz(r2)+(encok(r2)-enaz(r2))*rand;
        end;
        kromozom(r1,7)=omc_cis([kromozom(r1,1:6,:) ]);
    end

%hata_bayragi = 0;
eski=0;
%break;

```

```

omc_cis([kromozom(1,1:6)]);
end;

    hata_bayragi = 1;

end; % jenerasyon dongusunun bitisi

end; % hata kontrol dongusunun bitisi
omc_cis2([kromozom(1,1:6)])

fprintf('Optimizasyon Sonuclari omc cis ps l \n');

fprintf('L1  = %3.20f \n', (kromozom(1,1,:)));
fprintf('L2  = %3.20f \n', (kromozom(1,2,:)));
fprintf('L3  = %3.20f \n', (kromozom(1,3,:)));
fprintf('Z1  = %3.20f \n', (kromozom(1,4,:)));
fprintf('Z2  = %3.20f \n', (kromozom(1,5,:)));
fprintf('Z3  = %3.20f \n', (kromozom(1,6,:)));

fprintf('jenerasyon = %f \n', jenerasyon);
fprintf('hata    = %3.20f \n', kromozom(1,7,:));

```

MURGAOMC_CIS_SP_L

```

hata_bayragi = 0;
clc;
while hata_bayragi ==0,
    fprintf('tekrar uretim\n');
    clear;

    enaz(3)=0;
    encok(3)=0;
    enaz(6)=50;
    encok(6)=50;
    for r1=1:2
        enaz(r1)=14;
        encok(r1)=15;
    end
    for r1=4:5
        enaz(r1)=58;
        encok(r1)=87;
    end

    bs=28;
    eski=0;
    kromozom = zeros([20,7]);
    kromozomyed = zeros([60,7]);
    hata_bayragi=1;
    jenerasyon=0;
    rand('state',sum(100*clock));    %Randomizer baslatildi
    pdf = 1:20;
    pdf = pdf*0;
    for n= 1:20,
        pdf(n) = (21-n)/(10*21);
    end
    for n = 1:100,
        for r1=1:6
            kromozomi(n,r1) = enaz(r1)+(encok(r1)-enaz(r1))*rand;    % x
        end
        kromozomi(n,7) = omc_cis([kromozomi(n,1:6,:) ]); % hata fonksiyonu degeri
    end

    kromozomi = sortrows(kromozomi,7);
    kromozom=kromozomi(1:20,:);

    ortalama(1) = mean(kromozom(:,7));
    min_deger(1) = kromozom(1,7);

for yyy=1:1500 %hata 1 den buyukse jenerasyona devam et
    fprintf('.');
    minimum_deger = kromozom(1,7,:);

```

```

jenerasyon = jenerasyon+1;
%rulet carki
for os=1:20
    olasiliklar(os) = (((abs(kromozom(os,7))+0.0001)\sum(kromozom(:,7))));
end;
%rulet carki
for rca=1:20
    for rcaa=1:20
        rastgele(rcaa,2)=rand*olasiliklar(rcaa)*1000;
        rastgele(rcaa,1)=rcaa;
    end;
    rastgele=sortrows(rastgele,2);
    rula(rca)=rastgele(20,1);
end;
for rcb=1:20
    for rcbb=1:20
        rastgele(rcbb,2)=rand*olasiliklar(rcbb)*1000;
        rastgele(rcbb,1)=rcbb;
    end;
    rastgele=sortrows(rastgele,2);
    rulb(rcb)=rastgele(20,1);
end;

cri=0;

for cr=1:2:39
    cri=cri+1;
    carp=fix(rand*(6))+1;
    for cry=1:6
        if carp<cry

            kromozomcross(cr,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(cry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(cry)));
        else

            kromozomcross(cr,cry)=max((min((kromozom(rulb(cri),cry)),encok(cry))),enaz(cry)));

            kromozomcross(cr+1,cry)=max((min((kromozom(rula(cri),cry)),encok(cry))),enaz(cry)));

        end
    end
    kromozomcross(cr,7) = omc_cis([kromozomcross(cr,1:6,:) ]); % hata fonksiyonu
    degeri
    kromozomcross(cr+1,7) = omc_cis([kromozomcross(cr+1,1:6,:) ]); % hata fonksiyonu
    degeri

```

```

end

for mut=1:20
    carp1=fix(rand*(6))+1;
    carp2=fix(rand*(6))+1;

    kromozomyed(mut+60,:)=kromozom(rula(cry),:);
    kromozomyed(mut+60,carp1)=enaz(carp1)+(encok(carp1)-enaz(carp1))*rand;
    kromozomyed(mut+60,carp2)=enaz(carp2)+(encok(carp2)-enaz(carp2))*rand;

    kromozomyed(mut+60,7)=omc_cis([kromozomyed(mut+60,1:6,:) ]); % hata
    fonksiyonu degeri
end

kromozomyed(1:20,:)=kromozom;
kromozomyed(21:60,:)=kromozomcross;
kromozomyed = sortrows(kromozomyed,7);
kromozom = kromozomyed(1:20,:);
min_deger(jenerasyon)=kromozom(1,7);
fprintf('Hata : %f%f\n', kromozom(1,7,:),jenerasyon);

kromozom = sortrows(kromozom,7);

ortalama(jenerasyon) = mean(kromozom(:,7));
min_deger(jenerasyon) = kromozom(1,7,:);
if not(minimum_deger == min_deger(jenerasyon))
    eski=0;
end
if minimum_deger == min_deger(jenerasyon)
fprintf('*****');
    for r1=3:19;
s=fix(rand*6)+1;
        %for r2=1:6
            kromozom(r1,s)=enaz(s)+(encok(s)-enaz(s))*rand;
        %end;
        kromozom(r1,7)=omc_cis([kromozom(r1,1:6,:) ]);
    end
    eski = eski+1;
end;
if (minimum_deger == min_deger(jenerasyon)) & (eski>=3)
    for r1=3:19;
        for r2=1:6
            kromozom(r1,r2)=enaz(r2)+(encok(r2)-enaz(r2))*rand;
        end;
        kromozom(r1,7)=omc_cis([kromozom(r1,1:6,:) ]);
    end

%hata_bayragi = 0;
eski=0;
%break;
omc_cis([kromozom(1,1:6)]);

```

```
end;

hata_bayragi = 1;

end; % jenerasyon dongusunun bitisi

end; % hata kontrol dongusunun bitisi

omc_cis2([kromozom(1,1:6)])

fprintf('Optimizasyon Sonuclari omc cis sp l \n');

fprintf('L1 = %3.20f\n', (kromozom(1,1,:)));
fprintf('L2 = %3.20f\n', (kromozom(1,2,:)));
fprintf('L3 = %3.20f\n', (kromozom(1,3,:)));
fprintf('Z1 = %3.20f\n', (kromozom(1,4,:)));
fprintf('Z2 = %3.20f\n', (kromozom(1,5,:)));
fprintf('Z3 = %3.20f\n', (kromozom(1,6,:)));

fprintf('jenerasyon = %f\n', jenerasyon);
fprintf('hata = %3.20f\n', kromozom(1,7,:));
```

ÖZGEÇMİŞ

Doğum tarihi	21.04.1977	
Doğum yeri	Gelibolu	
Lise	1988-1995	Konya Meram Anadolu Lisesi
Lisans	1996-2001	Yıldız Teknik Üniversitesi Mühendislik Fak. Elektronik ve Haberleşme Müh. Anabilim Dalı
Yüksek Lisans	2002-2004	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Müh. Anabilim Dalı, Haberleşme Programı

