射频损耗是指射频信号在物理信道或接收机中受到的各种损耗,包 括信号在自由空间中的传输损耗、相位和频率偏移、相位噪声、热噪声, 以及接收机的非线性作用等。

## 5.1 射频概述

在 MATLAB 的 RF Toolbox 工具箱中提供一系列函数、对象和 App,用于射频(RF)组件网络的设计、建模、分析和可视化。此工具箱支持无线通信、雷达和信号完整性应用。

我们可以使用 RF Toolbox 构建包含滤波器、传输线、匹配网络、放 大器和混频器等射频组件的网络。要指定组件,可以使用 Touchstone 文 件等测量数据,也可以使用网络参数或物理属性。此工具箱提供了用于 射频数据分析、操作和可视化的函数。我们可以分析 S 参数,在 S、Y、Z、 T 和其他网络参数之间进行转换,还可借助矩形图、极坐标图以及史密 斯圆图将射频数据可视化。我们还可以去嵌入、检查和强制无源性,并 计算群和相位延迟。

此外,借助射频链路预算分析器,我们可以从噪声、功率和非线性方面分析收发机链路,并为电路包络仿真生成 RF Blockset 模型。可以使用有理函数拟合方法,构建背板、互连和线性组件模型,并导出为Simulink模块、SPICE 网表或 Verilog-A 模块,以用于时域仿真。

## 5.2 接收机

RF系统设计者在开始设计过程时,首先要确定整个系统必须满足的增益、噪声系数(NF)和非线性系数(IP3)。为了确保作为一个简单的 射频元件级联模型的架构的可行性,设计人员计算每级和级联增益、噪 声系数和 IP3(第三截距点)的值。

【例 5-1】 利用 MATLAB 构建接收机。

使用 RF 预算分析仪 App,可以做到:

- 建立一个射频元素级联。
- 计算系统的每级和级联输出功率、增益、噪声系数、信噪比和 IP3。
- 导出每个阶段和级联值到 MATLAB 工作空间。

第

• 将系统设计导出到 RF Blockset measurement testbench 作为 DUT(被测设备)子系 DesiredIF=400MHz OIP3=15 IL=1dB

• 导出系统设计到 RF Blockset 进行仿真。

统,通过 App 验证得到的结果。

其实现步骤如下。

1. 系统架构

使用 App 设计的接收系统架构如图 5-1 所示。



2. 构建超外差接收机

我们可以使用 MATLAB 命令行构建超外差接收机的所有组件,并使用 RF Budget Analyzer app 查看分析结果。

超外差接收机系统结构的第一个组成部分是天线和 TR 开关。我们用到达 TR 开关的 有效功率代替天线组。

(1) 该系统使用 TR 开关在发射机和接收机之间进行切换。交换机给系统增加 1.3dB 的损耗。代码中创建一个增益为一1.3dB、OIP3为37dBm的TRSwitch。为了匹配参考的 射频预算结果,假设噪声系数为2.3dB。

```
>> clear all;
elements(1) = rfelement('Name', 'TRSwitch', 'Gain', -1.3, 'NF', 2.3, 'OIP3', 37);
```

(2) 为了建立射频带通滤波器的模型,代码中使用 rfilter 函数来设计滤波器。为了进 行预算计算,每个阶段内部终止50Ω。因此,为了达到1dB的插入损耗,下一个元件即放大 器的 Zin 被设置为 132.896Ω。

```
>> Fcenter = 5.8e9;
Bwpass = 20e6;
Ζ
          = 132.986;
elements(2) = rffilter('ResponseType', 'Bandpass',
      'FilterType', 'Butterworth', 'FilterOrder', 6,
      'PassbandAttenuation', 10 * log10(2),
      'Implementation', 'Transfer function',
      'PassbandFrequency', [Fcenter - Bwpass/2 Fcenter + Bwpass/2], 'Zout', 50, ...
      'Name', 'RF Filter');
```

以上代码设计的滤波器的 S 参数并不理想,并且会自动在系统中插入大约-1dB 的

第 5 章

射

频损

耗

应 用 损耗。

(3)下面代码使用放大器对象来建模一个低噪声放大器块,增益为15dB,噪声系数为1.5dB,OIP3为26dBm。

>> elements(3) = amplifier('Name','LNA','Gain',15,'NF',1.5,'OIP3',26, 'Zin',Z)

(4) 模型 a 增益块的增益值设为 10.5dB,噪声系数设为 3.5dB,OIP3 设为 23dBm。

>> elements(4) = amplifier('Name', 'Gain', 'Gain', 10.5, 'NF', 3.5, '0IP3', 23);

(5) 接收机将 RF 频率下变频至 400MHz 的中频。使用调制器对象创建频率为 5.4GHz、 增益为-7dB、噪声系数为 7dB、OIP3 为 15dBm 的解调器块。

>> elements(5) = modulator('Name', 'Demod', 'Gain', - 7, 'NF', 7, 'OIP3', 15, 'L0', 5.4e9, 'ConverterType', 'Down');

(6) 为了建立射频带通滤波器的模型,使用 rfilter 函数来设计滤波器。

```
>> Fcenter = 400e6;
Bwpass = 5e6;
elements(6) = rffilter('ResponseType', 'Bandpass', ...
'FilterType', 'Butterworth', 'FilterOrder',4, ...
'PassbandAttenuation', 10 * log10(2), ...
'Implementation', 'Transfer function', ...
'PassbandFrequency', [Fcenter - Bwpass/2 Fcenter + Bwpass/2], 'Zout',50, ...
'Name', 'IF_Filter');
```

代码中的滤波器的 S 参数还是不理想,并且会自动在系统中插入大约-1dB 的损耗。 (7)将 a 型中频放大器,增益设为 40dB,噪声系数设为 2.5dB。

>> elements(7) = amplifier('Name', 'IFAmp', 'Gain', 40, 'NF', 2.5, 'Zin', Z);

(8) 接收机使用一个 AGC(自动增益控制)块,其中增益随可用的输入功率水平而变化。当输入功率为-80dB时,AGC 增益最大为 17.5dB。使用一个放大器模块来模拟 AGC。建立增益为 17.5dB、噪声系数为 4.3dB、OIP3 为 36dBm 的 AGC 模块。

>> elements(8) = amplifier('Name', 'AGC', 'Gain', 17.5, 'NF', 4.3, 'OIP3', 36);

(9) 根据以下系统参数计算超外差接收机的 rfbudget: 输入频率为 5.8GHz,可用输入 功率为-80dB,信号带宽为 20MHz。将天线元件替换为有效可用输入功率,估计达到 TR 开关的输入功率为-66dB。

```
>> superhet = rfbudget('Elements', elements, 'InputFrequency', 5.8e9, ...
'AvailableInputPower', - 66, 'SignalBandwidth', 20e6)
superhet =
    rfbudget with properties:
        Elements: [1x8 rf.internal.rfbudget.Element]
        InputFrequency: 5.8 GHz
        AvailableInputPower: - 66 dBm
        SignalBandwidth: 20 MHz
        Solver: Friis
        AutoUpdate: true
```

212

Analysis Results OutputFrequency: (GHz) [5.8 5.8 5.8 5.8 0.4 0.4 0.4 0.4] OutputPower: (dBm) [-67.3 -67.3 -53.3 -42.8 -49.8 - 49.8 - 10.8 6.7] TransducerGain: (dB)[-1.3 -1.3 12.7 23.2 16.2 16.2 55.2 72.7] NF: (dB) [2.3 2.3 3.531 3.657 3.693 3.693 3.728 3.728] IIP2: (dBm) [] OIP2: (dBm) [] IIP3: (dBm) [38.3 38.3 13.29 - 0.3904 - 3.824 - 3.824 - 3.824 - 36.7] OIP3: (dBm)[37 37 25.99 22.81 12.38 12.38 51.38 36] SNR: (dB) [ 32.66 32.66 31.43 31.31 31.27 31.27 31.24 31.24 ] %在射频预算分析软件中查看分析结果

>> show(superhet); %效果如图 5-2 所示

图 5-2 中 App 显示的级联值包括接收机输出频率、输出功率、增益、噪声系数、OIP3、信 噪比等。



图 5-2 射频预算分析效果图

图 5-3 中显示了 RF\_filter 级联对应的值。

- 3. 绘制级联传感器增益和级联噪声图
- (1)利用函数 rfplot 绘制接收端的级联传感器增益。

>> rfplot(superhet,'GainT') >> view(90,0) % 效果如图 5-4 所示

(2) 绘制接收机的级联噪声图。

```
>> rfplot(superhet,'NF')
>> view(90,0) %效果如图 5-5 所示
```

第5章

射频

损

耗

应

用

#### MATLAB无线通信系统建模与仿真



图 5-3 RF\_filter 级联对应的值



此外,我们还可以使用 RFBudgetAnalyzer 应用程序上的 Plot 按钮来绘制不同的输出值。

# 5.3 数据导入和网络参数

前面已对射频的相关概念及接收机进行了介绍,本节将对相关函数进行介绍,从而学 习怎样在射频中导入数据和设置网络参数。

### 5.3.1 数据导人参数

本节介绍一些数据导入的相关函数,主要对函数的调用格式及应用进行介绍。

1. sparameters 函数

在射频工具箱中,提供了 sparameters 函数用于设置 S 参数对象。函数的语法格式

如下。

sobj=sparameters(filename): 通过从 filename 指定的 Touchstone 文件导入数据来创 建一个 S 参数对象 sobj。

sobj=sparameters(data, freq):由S参数数据(data)和频率(freq)创建一个S参数 对象。

sobj=sparameters(data,freq,Z0):同时指定参考阻抗Z0。

sobj=sparameters(filterobj,freq): 计算一个滤波器对象 filterobj 的 S 参数。

sobj=sparameters(filterobj,freq,Z0):同时指定滤波器对象 filterobj 的参考阻抗 Z0。

sobj=sparameters(circuitobj,freq):使用默认参考阻抗计算一个电路对象 circuitobj 的 S 参数。

sobj=sparameters(circuitobj,freq,Z0):同时指定电路对象 circuitobj 的参考阻抗 Z0。

sobj=sparameters(netparamobj):将网络参数对象 netparamobj转换为具有默认参考 阻抗的 S 参数对象。

sobj=sparameters(netparamobj, Z0):将网络参数对象 netparamobj 转换为具有给定 参考阻抗 Z0 的 S 参数对象。

sobj=sparameters(rfdataobj):从 rfdataobj 中提取网络数据,并将其转换为 S 参数 对象。

sobj=sparameters(rfcktobj):从rfcktobj中提取网络数据,并将其转换为S参数对象。 sobj=sparameters(mnobj):返回最佳创建的匹配网络的S参数,根据源阻抗和负载阻 抗构建的频率列表进行评估。

sobj=sparameters(mnobj,freq):指定频率freq。

sobj=sparameters(mnobj,freq,Z0):指定参考阻抗Z0。

sobj=sparameters(mnobj,freq,Z0,circuitindices):指定每一个对象对应的 circuitindices 电路。

sobj=sparameters(antenna,freq,Z0): 计算天线对象在指定频率值和给定参考阻抗Z0上的复杂S参数。

sobj=sparameters(array,freq,Z0): 计算一个阵列对象在指定的频率值和给定的参考 阻抗 Z0 上的复杂 S 参数。

【例 5-2】 创建一个电阻元件 R50,将其添加到电路对象 example2 中,并计算 example2 的 S 参数。

```
>> hR1 = resistor(50, 'R50');
hckt1 = circuit('example2');
add(hckt1,[1 2], hR1)
setports(hckt1, [1 0],[2 0])
freq = linspace(1e3, 2e3, 100);
S = sparameters(hckt1, freq, 100);
disp(S)
```

```
运行程序,输出如下:
```

```
sparameters: S - parameters object
```

-- MATLAB无线通信系统建模与仿真

```
NumPorts: 2

Frequencies: [100 × 1 double]

Parameters: [2 × 2 × 100 double]

Impedance: 100

rfparam(obj,i,j) returns S - parameter Sij
```

2. yparameters 函数

在射频工具箱中,提供了 yparameters 函数用于创建 Y 参数对象。函数的调用格式如下。

hy = yparameters(filename): 通过从 filename 指定的 Touchstone 文件导入数据来创 建 Y 参数对象 hy。所有数据以真实/图像格式存储。

hy=yparameters(hnet):由 RF Toolbox 网络参数对象 hnet 创建 Y 参数对象。

hy=yparameters(data,freq):由Y参数数据(data)和频率(freq)创建Y参数对象。

hy=yparameters(rftbxobj):从rftbxobj 中提取网络数据并转换为Y参数数据。

【例 5-3】 在史密斯图上绘制 Y 参数。

```
%从默认值中提取 Y 参数,然后画在史密斯图上
```

```
>> Y = yparameters('default.s2p')
Y =
yparameters: Y - parameters object
    NumPorts: 2
    Frequencies: [191 × 1 double]
    Parameters: [2 × 2 × 191 double]
    rfparam(obj,i,j) returns Y - parameter Yij
>> figure;
smith(Y,1,1) %效果如图 5-6 所示
```

```
3. zparameters
```

在射频工具箱中,提供了 zparameters 函数用于创建 Z 参数对象。函数的调用格式

如下。

 $hz = zparameters(filename)_{\circ}$ 

```
hz = zparameters(hnet).
```

hz=zparameters(data,freq)。

```
hz=zparameters(rftbxobj)。
```

该函数的参数含义与 yparameters 函数的参数含义相同。

【例 5-4】 提取并绘制 Z11 的虚部。

```
% 读取文件默认值 s2p 作为 Z 参数,并提取 Z11
>> Z = zparameters('defaultbandpass.s2p')
Z =
    zparameters: Z - parameters object
        NumPorts: 2
        Frequencies: [1000 × 1 double]
        Parameters: [2 × 2 × 1000 double]
        rfparam(obj,i,j) returns Z - parameter Zij
>> z11 = rfparam(Z,1,1);
>> % 作图 Z11 的虚部
>> plot(Z.Frequencies, imag(z11)) %效果如图 5-7 所示
```



该函数的参数含义与 yparameters 函数的参数含义相同。

【例 5-5】 读取一个文件作为 ABCD 参数并提取 A。

```
% 读取文件默认值 s2p 作为 ABCD 参数
    >> abcd = abcdparameters('default.s2p')
    abcd =
      abcdparameters: ABCD - parameters object
           NumPorts: 2
         Frequencies: [191 × 1 double]
         Parameters: [2 \times 2 \times 191 \text{ double}]
      rfparam(obj,specifier) returns specified ABCD - parameter 'A', 'B', 'C', or 'D'
    8提取A参数
    >> A = rfparam(abcd, 'A')
    A =
      -0.1470 - 0.0698i
      -0.1421 - 0.0698i
      -0.1373 - 0.0696i
      -0.1325 - 0.0694i
      ...
      -0.1366 - 0.1883i
      -0.1379 - 0.1942i
      -0.1393 - 0.2001i
    5. gparameters 函数
    在射频工具箱中,提供了 gparameters 函数用于创建 hybrid-g 参数对象。函数的调用
格式如下。
```

```
hg=gparameters(filename).
```

```
-- MATLAB无线通信系统建模与仿真
```

```
hg=gparameters(hnet).
hg=gparameters(data,freq)。
hg=gparameters(rftbxobj)。
该函数的参数含义与 yparameters 函数的参数含义相同。
【例 5-6】 在读取的默认数据中提取 G11 参数。
>> g = gparameters('default.s2p')
a =
 gparameters: g - parameters object
      NumPorts: 2
     Frequencies: [191 × 1 double]
     Parameters: [2 \times 2 \times 191 \text{ double}]
 rfparam(obj, i, j) returns g - parameter gij
>> q11 = rfparam(q,1,1)
q11 =
   0.0158 + 0.0626i
  0.0167 + 0.0644i
  0.0176 + 0.0662i
  0.0185 + 0.0681i
```

```
6. hparameters 函数
```

...

0.0195 + 0.0700i 0.0206 + 0.0721i

在射频工具箱中,提供了 hparameters 函数用于创建混合 H 参数对象。函数的调用格 式如下。

```
hh=hparameters(filename).
```

```
hh=hparameters(hnet).
```

```
hh=hparameters(data,freq)。
```

hh=hparameters(rftbxobj)。

该函数的参数含义与 yparameters 函数的参数含义相同。

【例 5-7】 提取 H11。

```
% 读取文件默认值 s2p 作为 H 参数,并提取 H11
>> h = hparameters('default.s2p')
h =
  hparameters: h - parameters object
       NumPorts: 2
     Frequencies: [191 × 1 double]
     Parameters: [2 \times 2 \times 191 \text{ double}]
  rfparam(obj,i,j) returns h - parameter hij
>> h11 = rfparam(h,1,1)
h11 =
   1.0e+02 *
   0.0379 - 0.1501i
   0.0380 - 0.1472i
   0.0380 - 0.1442i
   0.0380 - 0.1412i
   0.0379 - 0.1383i
```

0.0377 - 0.1353i ...

7. tparameters 函数

在射频工具箱中,提供了 tparameters 函数用于创建 T 参数对象。函数的调用格式如下。

tobj=tparameters(filename): 通过从 filename 指定的 Touchstone 文件导入数据来创 建一个 T 参数对象 tobj。

tobj=tparameters(tobj\_old,z0):将 tobj\_old 中的 T 参数数据转换为新的阻抗 z0。z0 是可选的,如果没有提供,则利用复制代替转换。

tobj=tparameters(rftbx\_obj):从 rftbx\_obj对象中提取 S 参数网络数据,然后将数据 转换为 T 参数数据。

tobj=tparameters(hnet, z0):将 hnet 中的网络参数数据转换为 T 参数数据。

tobj=tparameters(paramdata,freq,z0):使用指定的频率和阻抗参数数据,创建 T 参数对象。

【例 5-8】 将文件中数据转换为 T 参数。

```
%从 Touchstone 文件中读取 S 参数数据,并将数据转换为 T 参数
>> T1 = tparameters('passive.s2p');
disp(T1)
  tparameters: T - parameters object
        NumPorts: 2
     Frequencies: [202 × 1 double]
     Parameters: [2 \times 2 \times 202 \text{ double}]
        Impedance: 50
  rfparam(obj, i, j) returns T - parameter Tij
%改变 t 参数的阻抗为 100 欧姆
>> T2 = tparameters(T1,100);
disp(T2)
  tparameters: T - parameters object
        NumPorts: 2
     Frequencies: [202 × 1 double]
     Parameters: [2 \times 2 \times 202 \text{ double}]
        Impedance: 100
  rfparam(obj, i, j) returns T - parameter Tij
```

8. rfdata.data 对象

在射频工具箱中,rfdata.data 对象用于将电路对象分析结果存储在页面中并全部展 开。有以下三种方法可以创建 rfdata。

• 可以使用 rfdata 从工作空间数据指定其属性来构造它。

• 可以使用 read 方法从文件数据创建它。

• 可以使用分析方法执行电路对象的频域分析,并将结果存储在 rfdata 中。

rfdata.data 对象的调用格式如下。

h=rfdata.data:返回一个数据对象,其属性都具有默认值。

h=rfdata.data('Property1',value1,'Property2',value2,...):使用一个或多个名称-值 对设置属性。可以指定多个名称-值对,但每个属性名称需用引号括起来。

#### - MATLAB无线通信系统建模与仿真

【例 5-9】 利用 rfdata. data 将来自. s2p 数据文件的射频数据用图形形式展开。

>> file = 'default.s2p'; h = read(rfdata.data,file); figure plot(h,'s21','db');

%将文件读入数据对象

% 在 XY 平面绘制 dB(S21)

运行程序,效果如图 5-8 所示。



图 5-8 读取的射频数据效果图

### 5.3.2 网络参数

5.3.1 节介绍了射频数据导入的相关函数,本节将介绍一些用于网络参数转换的相关 函数。

1. s2abcd 函数

在射频工具箱中,提供了 s2abcd 函数将 S 参数转换为 ABCD 参数。函数的调用格式 如下。

abcd\_params=s2abcd(s\_params,z0):将射频参数 s\_params 转换为 ABCD 参数 abcd\_ params。s\_params 输入是一个 2N×2N×M 数组,其中,M 表示 2N 端口的 S 参数。z0 为 参考阻抗,默认值是 50Ω。

【例 5-10】 将 S 参数转换为 ADCD 参数。

```
%定义一个S参数矩阵
>> s_11 = 0.61 * exp(j * 165/180 * pi);
s_21 = 3.72 * exp(j * 59/180 * pi);
s_12 = 0.05 * exp(j * 42/180 * pi);
s_22 = 0.45 * exp(j * (-48/180) * pi);
s_params = [s_11 s_12; s_21 s_22];
z0 = 50;
>> %转换为ABCD参数
>> abcd_params = s2abcd(s_params,z0)
abcd_params =
    0.0633 + 0.0069i    1.4958 - 3.9839i
    0.0022 - 0.0024i    0.0732 - 0.2664i
```

2. s2h 函数 在射频工具箱中,提供了 s2h 函数将 S 参数转换为混合 H 参数。函数的调用格式

第5章 射频损耗应用-------

如下。

h\_params=s2h(s\_params,z0): 将射频参数 s\_params 转换为混合参数 h\_params。 【例 5-11】 将 S 参数转换为混合 H 参数。

```
>> % 定义一个 S 参数矩阵
>> s_11 = 0.61 * exp(j * 165/180 * pi);
s_21 = 3.72 * exp(j * 59/180 * pi);
s_12 = 0.05 * exp(j * 42/180 * pi);
s_22 = 0.45 * exp(j * (-48/180) * pi);
s_params = [s_11 s_12; s_21 s_22];
z0 = 50;
>> % 转换为混合 H 参数
>> h_params = s2h(s_params, z0)
h_params =
15.3381 + 1.4019i 0.0260 + 0.0411i
-0.9585 - 3.4902i 0.0106 + 0.0054i
```

3. s2s 函数

在射频工具箱中,提供了 s2s 函数将 S 参数转换为不同阻抗的 S 参数。函数的调用格 式如下。

s\_params\_new = s2s(s\_params,z0): 将具有参考阻抗 z0 的射频参数 s\_params 转换 为具有默认参考阻抗 50Ω 的射频参数 s\_params\_new。

s\_params\_new = s2s(s\_params,z0,z0\_new): 将具有参考阻抗 z0 的射频参数 s\_ params 转换为具有参考阻抗 z0\_new 的射频参数 s\_params\_new。

【例 5-12】 利用 s2s 函数将定义的 S 参数矩阵转换为抗阻为 40 Ω 的 S 参数。

```
%定义一个S参数矩阵
>> s_11 = 0.61 * exp(1i * 165/180 * pi);
s_21 = 3.72 * exp(1i * 59/180 * pi);
s_12 = 0.05 * exp(1i * 42/180 * pi);
s_22 = 0.45 * exp(1i * (-48/180) * pi);
s_params = [s_11 s_12; s_21 s_22];
z0 = 50;
z0_new = 40;
% 转换为阻抗为 50Ω 的 S参数
>> s_params_new = s2s(s_params,z0,z0_new)
s_params_new =
        - 0.5039 + 0.1563i     0.0373 + 0.0349i
        1.8929 + 3.2940i     0.4150 - 0.3286i
```

4. s2scc 函数

在射频工具箱中,提供了 s2scc 函数将单端 S 参数转换为共模式 S 参数(Scc)。函数的 调用格式如下。

scc\_params=s2scc(s\_params):将2N端口单端S参数s\_params转换为N端口共模式S参数scc\_params。

scc\_params=s2scc(s\_params, option):根据选项中指定的端口顺序转换 S 参数。 【例 5-13】 利用 s2scc 函数将网络数据转换为共模式 S 参数。

#### -----MATLAB无线通信系统建模与仿真

```
>> s_params = sparameters('default.s4p');
s4p = s_params.Parameters;
s_cc = s2scc(s4p);
s_cc_new = s_cc(1:5)
s_cc_new =
    0.0267 + 0.0000i     0.9733 + 0.0000i     0.9733 + 0.0000i     0.0267 + 0.0000i
    0.1719 + 0.1749i
```

5. s2scd 函数

在射频工具箱中,提供了 s2scd 函数,用于将 4 端口、单端等 S 参数转换到 2 端口跨模式 S 参数(Scd)。函数的调用格式如下。

```
scd_params=s2scd(s_params):将 2N 端口单端 S 参数 s_params 转换为 N 端口跨模 式 S 参数 scd_params。
```

scd\_params=s2scd(s\_params, option): 根据可选选项 option 参数转换 S 参数。

【例 5-14】 使用默认端口顺序将网络数据转换为跨模式 S 参数。

```
>> ckt = read(rfckt.passive, 'default.s4p');
s4p = ckt.NetworkData.Data;
s_cd = s2scd(s4p);
s_cd_new = s_cd(1:5)
s_cd_new = 1 × 5 complex
0.0015 - 0.0029i 0.0003 - 0.0009i - 0.0005 + 0.0014i 0.0019 - 0.0027i
0.0030 - 0.0019i
```

6. s2sdd 函数

在射频工具箱中,提供了 s2sdd 函数,用于将 4 端口、单端等 S 参数转换到 2 端口差分 模式 S 参数(Sdd)。函数的调用格式如下。

```
sdd_params=s2sdd(s_params)。
sdd_params=s2sdd(s_params,option)。
该函数的参数含义与 s2scd 函数的参数含义相同。
【例 5-15】 利用 s2sdd 函数分析差分模式 S 参数。
```

```
%从数据文件创建一个电路对象
>> ckt = read(rfckt.passive, 'default.s4p');
data = ckt.AnalyzedResult;
8创建一个数据对象来存储不同的 S 参数
diffSparams = rfdata.network;
diffSparams.Freq = data.Freq;
diffSparams.Data = s2sdd(data.S Parameters);
diffSparams.Z0 = 2 * data.Z0;
%使用数据对象中的数据创建一个新的电路对象
diffCkt = rfckt.passive;
diffCkt.NetworkData = diffSparams;
*分析新的电路对象
frequencyRange = diffCkt.NetworkData.Freq;
ZL = 50;
ZS = 50;
Z0 = diffSparams.Z0;
analyze(diffCkt, frequencyRange, ZL, ZS, Z0);
diffData = diffCkt.AnalyzedResult;
```

```
% 将差分 S 参数写入 Touchstone 数据文件
write(diffCkt,'diffsparams.s2p')
ans = logical
1
```

7. s2smm 函数

在射频工具箱中,提供了 s2smm 函数用于将单端 S 参数转换为混合模式 S 参数。函数的调用格式如下。

[s\_dd,s\_dc,s\_cd,s\_cc]=s2smm(s\_params\_even,rfflag): 将单端 S 参数转换为混合 模式形式。

s\_mm=s2smm(s\_params\_odd):将单端奇 S 参数矩阵转换为混合模式矩阵。为了从 s\_params\_odd 创建混合模式矩阵,单端输入端口依次配对(端口 1 和端口 2,端口 3 和端口 4,等等),最后一个端口保持单端。

【例 5-16】 利用 s2smm 函数将 4 端口 S 参数转换为 2 端口混合模式 S 参数。

```
>> ckt = read(rfckt.passive, 'default.s4p');
s4p = ckt.NetworkData.Data;
[s_dd,s_dc,s_cd,s_cc] = s2smm(s4p);
s_dd1 = s_dd(1:5)
s_dd1 =
    0.0267 + 0.0000i     0.9733 + 0.0000i     0.9733 + 0.0000i     0.0267 + 0.0000i
    0.0335 - 0.0057i
```

8. s2rlgc 函数

在射频工具箱中,提供了 s2rlgc 函数将 S 参数转换为 RLGC 传输线路参数。函数的调用格式如下。

rlgc\_params=s2rlgc(s\_params,length,freq,z0):将多端口S参数数据转换为用频域 表示的 RLGC 传输线路参数。

rlgc\_params=s2rlgc(s\_params,length,freq):使用参考阻抗为 50Ω,将多端口 S 参数 数据转换为 RLGC 传输线路参数。

【例 5-17】 利用 s2rlgc 函数将创建的 S 参数转换为 RLGC 传输线路参数。

```
>>> s_11 = 0.000249791883190134 - 9.42320545953709e-005i;
s_12 = 0.999250283783862 - 0.000219770154524734i;
s_21 = 0.999250283783863 - 0.000219770154524756i;
s_22 = 0.000249791883190079 - 9.42320545953931e-005i;
s_params = [s_11,s_12; s_21,s_22];
%指定传输线的长度、频率和阻抗
length = 1e-3;
freq = 1e9;
z0 = 50;
%将S参数转换为rlgc参数
rlgc_params = s2rlgc(s_params,length,freq,z0)
```

运行程序,输出如下:

rlgc\_params = 包含以下字段的 struct: R: 50.0000

```
L: 1.0000e - 09
G: 0.0100
C: 1.0000e - 12
alpha: 0.7265
beta: 0.2594
Zc: 63.7761 - 14.1268i
```

9. s2t 函数

在射频工具箱中,提供了 s2t 函数将 S 参数转换为 T 参数。函数的调用格式如下。 t\_params=s2t(s\_params):将射频参数 s\_params 转换为射频参数 t\_params。 【例 5-18】 利用 s2t 函数将 S 参数转换为 T 参数。

```
>> %定义一个S参数矩阵
>> sl1 = 0.61 * exp(j * 165/180 * pi);
sl1 = 0.05 * exp(j * 59/180 * pi);
sl2 = 0.05 * exp(j * 42/180 * pi);
sl2 = 0.45 * exp(j * (-48/180) * pi);
s_params = [sl1 sl2; sl2 sl2];
>> t_params = slt(s_params)
t_params =
    0.1385 - 0.2304i    0.0354 + 0.1157i
    -0.0452 + 0.1576i    -0.0019 - 0.0291i
```

10. s2y 函数

在射频工具箱中,提供了 s2y 函数将 S 参数转换为 Y 参数。函数的调用格式如下。

y\_params=s2y(s\_params,z0): 将射频参数 s\_params 转换为射频参数 y\_params。 s\_params 输入是一个 N×N×M 数组,表示 M 个 N 端口 S 参数。z0 为参考阻抗,默认值是 50Ω。

【例 5-19】 利用 s2y 函数将 S 参数转换为 Y 参数。

```
%定义S参数和阻抗
>> s_11 = 0.61 * exp(1i * 165/180 * pi);
s_21 = 3.72 * exp(1i * 59/180 * pi);
s_12 = 0.05 * exp(1i * 42/180 * pi);
s_22 = 0.45 * exp(1i * (-48/180) * pi);
s_params = [s_11 s_12; s_21 s_22];
z0 = 50;
>> %将S参数转换为Y参数
>> y_params = s2y(s_params,z0)
y_params =
        0.0647 - 0.0059i - 0.0019 - 0.0025i
        -0.0826 - 0.2200i 0.0037 + 0.0145i
```

11. s2z 函数

在射频工具箱中,提供了 s2z 函数将 S 参数转换为 Z 参数。函数的调用格式为  $z_{params}=s2z(s_{params},z0)$ 。

该函数的参数含义与 s2t 函数的参数含义相同。

【例 5-20】 利用 s2z 函数将 S 参数转换为 Z 参数。

%定义一个 S 参数矩阵

```
>> s_11 = 0.61 * exp(j * 165/180 * pi);
s_21 = 3.72 * exp(j * 59/180 * pi);
s_12 = 0.05 * exp(j * 42/180 * pi);
s_22 = 0.45 * exp(j * (-48/180) * pi);
s_params = [s_11 s_12; s_21 s_22];
z0 = 50;
>> %将S参数转换为Z参数
>> z_params = s2z(s_params,z0)
z_params =
1.0e + 02 *
0.1141 + 0.1567i 0.0352 + 0.0209i
2.0461 + 2.2524i 0.7498 - 0.3803i
```

12. smm2s 函数

在射频工具箱中,提供了 smm2s 函数将混合模式 2N 端口 S 参数转换为单端 4N 端口 S 参数。函数的调用格式如下。

s\_params=smm2s(s\_dd,s\_dc,s\_cc):将混合模式,N端口S参数转换为单端, 2N端口S参数s\_params。Smm2s将混合模式端口的前半部分映射为奇数对单端端口,后 半部分映射为偶数对单端端口。

s\_params=smm2s(s\_dd,s\_dc,s\_cd,s\_cc,option):使用可选参数选项 option 转换 S 参数数据。还可以使用 snp2smp 函数重新排列 s\_params 中的端口。

【例 5-21】 实现混合模式和单端 S 参数之间的转换。

13. s2tf 函数

在射频工具箱中,提供了 s2tf 函数实现将 2 端口网络的 S 参数转换为电压或功率波传 递函数。函数的调用格式如下。

tf=s2tf(s\_params):将2端口网络的射频参数 s\_params 转换为该网络的电压传递函数。

tf=s2tf(s\_params,z0,zs,zl): 使用参考阻抗 z0、源阻抗 zs 和负载阻抗 zl 计算电压传 递函数。

tf=s2tf(hs):将2端口S参数对象hs转换为网络的电压传递函数。

tf=s2tf(hs,zs,zl):利用源阻抗 zs 和负载阻抗 zl 计算电压传递函数。

tf=s2tf(s\_params,z0,zs,zl,option)或tf=s2tf(hs,zs,zl,option):使用option指定的方法计算电压或功率波传递函数。

【例 5-22】 计算 S 参数阵列的电压传递函数。

```
>> ckt = read(rfckt.passive,'passive.s2p');
```

第5章

射频

损

耗

应

用

```
-----MATLAB无线通信系统建模与仿真
```

```
sparams = ckt.NetworkData.Data;
tf = s2tf(sparams)
tf =
    0.9964 - 0.0254i
    0.9960 - 0.0266i
    0.9956 - 0.0284i
    0.9961 - 0.0290i
    0.9960 - 0.0301i
```

```
14. snp2smp 函数
```

在射频工具箱中,提供了 snp2smp 函数将单端 N 端口 S 参数转换为单端 M 端口 S 参数,并重新排序。函数的调用格式如下。

s\_params\_mp = snp2smp(s\_params\_np): 将单端 N 端口 S 参数 s\_params\_np 转换为 单端 M 端口 S 参数 s\_params\_mp,并重新排序,且 M 必须小于或等于 N。

s\_params\_mp = snp2smp(s\_params\_np,Z0,n2m\_index,ZT): 使用控制转换的可选 参数 Z0、n2m\_index 和 ZT 来转换和重新排序 S 参数数据。

s\_params\_mp = snp2smp(s\_obj,n2m\_index,ZT): 将 S 参数对象 s\_obj 转换为单端 M 端口 S 参数 s\_params\_mp,并重新排序,且 M 必须小于或等于 N。

【例 5-23】 通过终止端口 3 与阻抗 Z0 将 3 端口 S 参数转换为 2 端口 S 参数。

```
>> ckt = read(rfckt.passive, 'default.s3p');
s3p = ckt.NetworkData.Data;
Z0 = ckt.NetworkData.Z0;
s2p = snp2smp(s3p,Z0);
s2p_new = s2p(1:5)
s2p_new =
    - 0.0073 - 0.8086i     0.0869 + 0.3238i - 0.0318 + 0.4208i     0.1431 - 0.7986i
    - 0.0330 - 0.8060i
```

```
15. rlgc2s 函数
```

在射频工具箱中,提供了 rlgc2s 函数将 RLGC 传输线路参数转换为 S 参数。函数的调用格式如下。

s\_params=rlgc2s(R,L,G,C,length,freq,z0):将 RLGC 传输线路参数数据转换为 S 参数。

s\_params=rlgc2s(R,L,G,C,length,freq):将RLGC传输线路参数数据转换为S参数,参考阻抗为50Ω。

【例 5-24】 将 RLGC 传输线路参数转换为 S 参数。

```
%定义传输线的变量
>> length = 1e-3;
freq = 1e9;
z0 = 50;
R = 50;
L = 1e-9;
G = .01;
C = 1e-12;
%计算的参数
s_params = rlgc2s(R,L,G,C,length,freq,z0)
```

```
运行程序,输出如下:
```

16. cascadesparams 函数

在射频工具箱中,提供了 cascadesparams 函数结合 S 参数形成级联网络。函数的调用格式如下。

 $s_params = cascadesparams(s1_params,s2_params,...,sk_params):将由S参数描述的k个输入网络的射频参数(S参数)形成级联。每个输入网络必须是一个由M个频率点的S参数组成的2N×2N×M阵列描述的2N端口网络。所有的网络必须有相同的参考阻抗。$ 

hs = cascadesparams(hs1, hs2, ..., hsk):将具有相等的阻抗和频率的对象进行级联,每个对象的参数包含  $2N \times 2N \times M$  的 S 参数数组的 M 个频率点。

s\_params = cascadesparams(s1\_params,s2\_params,...,sk\_params,Kconn): 根据 Kconn 指定的网络之间的级联连接数创建级联网络。

【例 5-25】 将一组 3 端口 S 参数和一组 2 端口 S 参数组装成一个 3 端口级联网络。

```
8创建一组3端口S参数和一组2端口S参数
>> ckt1 = read(rfckt.passive, 'default.s3p');
ckt2 = read(rfckt.amplifier, 'default.s2p');
freq = [2e9 2.1e9];
analyze(ckt1, freq);
analyze(ckt2, freq);
sparams 3p = ckt1.AnalyzedResult.S Parameters;
sparams_2p = ckt2.AnalyzedResult.S_Parameters;
%将两组设备通过一个端口级联
Kconn = 1;
sparams_cascaded_3p = cascadesparams(sparams_3p, sparams_2p, Kconn)
运行程序,输出如下:
sparams cascaded 3p(:,:,1) =
   0.1339 - 0.9561i 0.0325 + 0.2777i 0.0222 + 0.0092i
                                       0.0199 + 0.0255i
   0.3497 + 0.2449i
                      0.3130 - 0.9235i
  -4.0617 + 5.0914i -1.6296 + 4.7333i -0.7133 - 0.7305i
sparams_cascaded_3p(:,:,2) =
  -0.3023 - 0.7303i 0.0635 + 0.4724i 0.0005 - 0.0220i
                                       0.0198 - 0.0274i
   0.1408 + 0.2705i - 0.1657 - 0.7749i
```

5.7709 + 2.2397i 4.1929 - 0.2165i - 0.5092 + 0.4251i

# 5.4 射频滤波器设计

本节学习使用射频工具箱设计射频滤波器,如巴特沃斯,切比雪夫,逆切比雪夫等。还可以使用滤波器对象或 RFCKT 滤波器来设计具有不同实现类型的滤波器。

在射频工具箱中,提供了rffilter函数用于使用滤镜对象创建Butterworth、Chebyshev 或 Inverse Chebyshev RF滤波器。射频滤波器是一个2端口电路对象,我们可以把这个对 -- MATLAB无线通信系统建模与仿真

象作为电路的一个元素。函数的调用格式如下。

rfobj=rffilter: 创建具有默认属性的2端口滤波器。

rfobj=rffilter(Name, Value):使用一个或多个名称(Name)-值(Value)对设置属性。 例如,rfobj = rfilter('FilterType', 'Chebyshev')用于创建一个2端口 Chebyshev RF 滤波器。

【例 5-26】 创建并查看默认 RF 滤波器对象的属性。

```
>> rfobj = rffilter
rfobi =
  rffilter: Filter element
             FilterType: 'Butterworth'
            ResponseType: 'Lowpass'
          Implementation: 'LC Tee'
              FilterOrder: 3
       PassbandFrequency: 1.0000e + 09
     PassbandAttenuation: 3.0103
                      Zin: 50
                    Zout: 50
              DesignData: [1×1 struct]
         UseFilterOrder: 1
                    Name: 'Filter'
                NumPorts: 2
                Terminals: { 'p1 + ' 'p2 + ' 'p1 - ' 'p2 - '}
>> rfobj. DesignData
ans =
  包含以下字段的 struct:
              FilterOrder: 3
                Inductors: [7.9577e - 09 7.9577e - 09]
              Capacitors: 6.3662e - 12
                Topology: 'lclowpasstee'
       PassbandFrequency: 1.0000e + 09
     PassbandAttenuation: 3.0103
```

### 5.4.1 逆切比雪夫滤波器 I

【例 5-27】 此实例展示了如何确定一个五阶逆切比雪夫低通滤波器的传递函数,其通带衰减为 1dB,截止频率为 1rad/s,阻带最小衰减为 50dB,振幅响应为 2rad/s。

滤波器对象用于设计射频滤波器。一个滤波器需要一个最小的参数集来完全定义它。 滤波器对象初始化后,属性 DesignData 包含所设计的滤波器的完整解决方案。它是一种结构,它包含诸如用于构造传递函数的计算因式多项式等字段。

(1) 设计逆切比雪夫 I 型滤波器。

| >> N   | = 5;          | %滤波器阶数        |  |
|--|---------------|---------------|--|
| Fp   | = 1/(2 * pi); | %通带截止频率       |  |
| Ар   | = 1;          | <b>%</b> 通带衰减 |  |
| As   | = 50;         | %阻带衰减         |  |
| %使用滤波器对象创建所需的滤波器.逆切比雪夫的唯一实现类型是"传递函数"   |               |               |  |
| <pre>r = rffilter('FilterType', 'InverseChebyshev', 'ResponseType', 'Lowpass',</pre> |               |               |  |
| 'Implementation', 'Transfer function', 'FilterOrder', N,                             |               |               |  |
| 'PassbandFrequency', Fp, 'StopbandAttenuation', As,                                  |               |               |  |

'PassbandAttenuation', Ap); (2) 牛成和可视化传递函数多项式。 >> %利用 tf 函数生成传递函数多项式 [numerator, denominator] = tf(r); format long g %显示 Numerator{2,1}多项式系数 disp('传递函数的分子多项式系数'); %传递函数的分子多项式系数 disp(numerator{2,1}); 传递函数的分子多项式系数 0.672768334081369 0 0.0347736250821381 0 2.6032214373595 8显示分母的多项式系数 >> disp('传递函数的分母多项式系数'); %传递函数的分母多项式系数 >> disp(denominator); 传递函数的分母多项式系数 1 3.81150884154936 7.2631952221038 8.61344575257214 6.42982763112227 2.6032214373595 >> G s = tf(numerator, denominator) G\_s = From input 1 to output... s^5 1. \_\_\_\_\_ \_\_\_\_\_  $s^{5} + 3.812 s^{4} + 7.263 s^{3} + 8.613 s^{2} + 6.43 s + 2.603$  $0.03477 \text{ s}^{4} + 0.6728 \text{ s}^{2} + 2.603$ 2: \_\_\_\_\_  $s^{5} + 3.812 s^{4} + 7.263 s^{3} + 8.613 s^{2} + 6.43 s + 2.603$ From input 2 to output...  $0.03477 \, s^{4} + 0.6728 \, s^{2} + 2.603$ 1: \_\_\_\_\_ \_\_\_\_\_  $s^{5} + 3.812 s^{4} + 7.263 s^{3} + 8.613 s^{2} + 6.43 s + 2.603$ s^ 5 2. \_\_\_\_\_  $s^{5} + 3.812 s^{4} + 7.263 s^{3} + 8.613 s^{2} + 6.43 s + 2.603$ Continuous - time transfer function. (3) 可视化滤波器的振幅响应。

% 可视化滤波器的振幅响应
>> frequencies = linspace(0,1,1001);
Sparam = sparameters(r, frequencies);

注意: S参数使用二次(低通/高通)或四次(带通/带阻)分解形式计算传递函数。这些因子被用来构造多项式。多项式形式是数值不稳定的较大的滤波器阶,所以首选的形式是因式二次/四次形式。这些因数分解的部分存在于 r. DesignData 中。

>> rfplot(Sparam, 2, 1) %效果如图 5-9 所示

(4) 滤波器在指定频率下的振幅响应。

>> freq = 2/(2 \* pi); hold on; 第5章

射

频损耗

应

用

setrfplot('noengunits', false);

注意:要在同一个图上使用 rfplot 和 plot,请使用 setrfplot。在命令窗口中输入"help setrfplot"可获取其信息。

>> plot(freq \* ones(1,101),linspace(-120,20,101)); %效果如图 5-10 所示 >> % 在 2rad/s 处计算准确值 S freq = sparameters(r, freq); As freq = 20 \* log10(abs(rfparam(S freq, 2, 1))); sprintf('2rad/s的振幅响应为 %d dB', As freq) ans = '2rad/s的振幅响应为 - 3.668925e + 01 dB' %计算 As 处的阻带频率 >> Fs = r. DesignData. Auxiliary. Wx \* r. PassbandFrequency; sprintf('阻带频率为 - %d dB 是: %d Hz', As, Fs) ans = '阻带频率为 - 50 dB 是: 3.500241e - 01 Hz' 20 20 dB(S21) dB(S21) 0 0 -20-20(gp)-40 個) 一60 福度(dB) -40 -60 -80 -80-100-100-120L -1200.2 0.8 1 02 0.6 0.4 0.6 0.4 0.8 频率(Hz) 频率(Hz) 图 5-10 指定频率下的振幅响应效果图

图 5-9 滤波器的振幅响应效果图

### 5.4.2 中频巴特沃斯带通滤波器

【例 5-28】 此实例展示了如何设计一个中频(IF)巴特沃斯带通滤波器,中心频率为 400MHz,带宽为 5MHz,插入损耗(IL)为 1dB。

1

(1) 失配/插入损耗(IL)的原因。

实际电路存在一定程度的失配。当不匹配的电路连接到射频源时,就会发生失配,导 致反射,从而导致传递到电路的功率损失。我们可以使用 IL 来定义这种不匹配。根据给 定的 IL 计算输出阻抗失配, IL 与归一化输出阻抗(Z<sub>out</sub>)的关系如下:

IL(dB) =  $-10 \times lg(1 - |\gamma_{in}|^2) = -10 \times lg(4 \times Z_{out}/(1 + Z_{out})^2)$ 

上式得到的多项式的根返回归一化 Zout 的值。Zout 的非规格化值分别为 132.986Ω 和 18.799Ω。为滤波器设计选择较高的值。

>> Zout = 132.986;

(2)设计滤波器。

%使用滤波器对象设计所需规格的滤波器 >> Fcenter = 400e6; Bwpass = 5e6;



数据光标显示 1dB IL 在 Fcenter = 400MHz 效果图如图 5-12 所示。



(4) 将 Filter 表示为 Touchstone 文件。

使用 rfwrite 将 所 设 计 滤 波 器 的 参 数 写 入 所 需 的 频 率 范 围。我 们 可 以 将 这 个 Touchstone 文件读入一个 nport 对象, 而 nport 对象又可以插入 rfbudget 对象中。

```
filename = 'filterIF.s2p';
if exist(filename, 'file')
        delete(filename)
end
rfwrite(Sf,filename, 'format', 'MA')
```

## 5.4.3 带通滤波器的响应

【例 5-29】 此实例展示了如何计算一个简单带通滤波器的时域响应。 计算带通滤波器的时域响应的步骤如下。

• 采用经典的图像参数设计方法选择电感和电容值。

- 使用具有 add 函数的电路、电容和电感及其他元件,可编程构造一个巴特沃斯电路。
- 使用 setports 将电路定义为 2 端口网络。
- •利用参数提取宽频率范围的2端口网络的S参数。
- 使用 s2tf 计算输入到输出的电压传递函数。
- 使用 rational fit 生成 rational 拟合,捕获一个精度非常高的理想 RC 电路。
- 创建一个噪声输入电压波形。
- 使用 timeresp 计算噪声输入电压波形的瞬态响应。

(1) 通过图像参数设计带通滤波器。

图像参数设计方法是一种用于分析计算无源滤波器中串联和并联元件值的框架。 图 5-14 是一个由两个半部分组成的巴特沃斯带通滤波器。



图 5-14 巴特沃斯带通滤波器

下面的 MATLAB 代码生成一个带通滤波器的组件值与较低的 3dB 截止频率 2.4GHz 和较高的 3dB 截止频率 2.5GHz。

>> Ro = 50; f1C = 2400e6; f2C = 2500e6; Ls = (Ro / (pi \* (f2C - f1C)))/2; Cs = 2 \* (f2C - f1C)/(4 \* pi \* Ro \* f2C \* f1C); Lp = 2 \* Ro \* (f2C - f1C)/(4 \* pi \* f2C \* f1C); Cp = (1/(pi \* Ro \* (f2C - f1C)))/2;

(2) 通过编程构造电路。

在使用电感和电容对象构建电路之前,必须给如图 5-15 所示的电路节点编号。



图 5-15 添加到巴特沃斯带通滤波器的节点数

% 创建一个电路对象,并使用 add 函数填充电感和电容对象
>> ckt = circuit('butterworthBPF');
add(ckt,[3 2], inductor(Ls))

```
add(ckt,[4 3],capacitor(Cs))
add(ckt,[5 4],capacitor(Cs))
add(ckt,[6 5],inductor(Ls))
add(ckt,[4 1],capacitor(Cp))
add(ckt,[4 1],inductor(Lp))
add(ckt,[4 1],inductor(Lp))
add(ckt,[4 1],capacitor(Cp))
```

(3) 从 2 端口网络中提取 S 参数。

要从电路对象中提取 S 参数,首先使用 setports 函数将电路定义为 2 端口网络。一旦电路有了端口,即可使用参数提取感兴趣的频率 S 参数。

```
>> freq = linspace(2e9,3e9,101);
setports(ckt,[2 1],[6 1])
S = sparameters(ckt,freq);
```

(4) 将电路的传递函数拟合为有理函数。

使用 s2tf 函数从 S 参数对象生成一个传递函数,然后使用有理拟合将传递函数数据拟合为有理函数。

```
>> tfS = s2tf(S);
fit = rationalfit(freq,tfS);
```

(5) 验证有理拟合逼近。

利用频率响应函数验证有理拟合近似在拟合频率范围外的合理的行为。

```
>> widerFreqs = linspace(2e8,5e9,1001);
resp = freqresp(fit,widerFreqs);
figure
semilogy(freq,abs(tfS),widerFreqs,abs(resp),'--','LineWidth',2) %效果如图 5-16 所示
xlabel('频率(Hz)')
ylabel('幅度')
legend('数据','拟合')
title('有理拟合在拟合频率范围外表现效
果')
```

(6)构造输入信号来测试带通滤波器。该信号由于包含了零均值随机噪声和

2.35GHz的阻塞物而产生噪声,所以这个 带通滤波器应该能够恢复 2.45GHz的正弦 信号。下面的 MATLAB 代码从 4096 个样 本中构造这样一个信号。

```
>> fCenter = 2.45e9;
fBlocker = 2.35e9;
period = 1/fCenter;
sampleTime = period/16;
signalLen = 8192;
t = (0:signalLen - 1)' * sampleTime;
```



\*256 周期

第5章

射频

、損耗

应

用

#### ------ MATLAB无线通信系统建模与仿真

```
input = sin(2 * pi * fCenter * t); % 清除输入信号
rng('default')
noise = randn(size(t)) + sin(2 * pi * fBlocker * t);
noisyInput = input + noise; % 输入噪声信号
```

(7) 计算输入信号的瞬态响应。

timeresp 函数计算由有理拟合和输入信号定义的状态空间方程的解析解。

```
>> output = timeresp(fit, noisyInput, sampleTime);
```

(8) 在时域中查看输入信号和滤波器响应。

在图形窗口中绘制输入信号、噪声输入信号和带通滤波器输出。

```
>> xmax = t(end)/8;
figure
subplot(3,1,1)
plot(t, input)
axis([0 xmax -1.5 1.5])
title('输入信号')
subplot(3,1,2)
plot(t, noisyInput)
axis([0 xmax floor(min(noisyInput)) ceil(max(noisyInput))])
title('噪声输入信号')
ylabel('振幅(V)')
```

```
subplot(3,1,3)
plot(t,output)
axis([0 xmax -1.5 1.5])
title('带通滤波器输出')
xlabel('时间(s)')
```

```
运行程序,效果如图 5-17 所示。
```

```
(9) 在频域内查看输入信号和滤波器响应。
```

```
NFFT = 2<sup>nextpow2(signalLen);</sup> % Next power of 2 from length of y
Y = fft(noisyInput, NFFT)/signalLen;
samplingFreq = 1/sampleTime;
f = samplingFreq/2 * linspace(0,1, NFFT/2 + 1)';
0 = fft(output, NFFT)/signalLen;
```

```
figure
subplot(2,1,1)
plot(freq,abs(tfS),'b','LineWidth',2)
axis([freq(1) freq(end) 0 1.1])
legend('滤波器的传递函数')
ylabel('幅度')
```

```
subplot(2,1,2)
plot(f,2 * abs(Y(1:NFFT/2 + 1)),'g',f,2 * abs(O(1:NFFT/2 + 1)),'r','LineWidth',2)
axis([freq(1) freq(end) 0 1.1])
legend('输入 + 噪声','输出')
title('滤波器特性和噪声输入频谱.')
xlabel('频率(Hz)')
ylabel('电压(V)')
```

运行程序,效果如图 5-18 所示。



在频域中叠加噪声输入和滤波器响应解释了为什么滤波操作是成功的。2.35GHz的 阻塞信号和大部分噪声都被显著减弱。

### 5.4.4 设计匹配网络

【例 5-30】 此实例展示了如何使用射频工具箱来确定输入和输出匹配网络,以最大限 度地提高 50Ω 负载和系统的功率。

输入输出匹配网络的设计是放大器设计的重要组成部分。这个例子首先计算同时共 轭匹配的反射因子,然后确定在每个匹配网络中的一个分流存根的位置的指定频率。最 后,通过实例将匹配网络与放大器级联,并绘制结果。其实现步骤如下。

(1) 创建一个放大器对象。

创建一个放大器对象 rfckt。放大器对象用来表示放大器所在文件 samplebjt2. s2p 中测量频率的相关 S 参数数据。然后从 rfckt 中提取频率相关的 S 参数数据。

```
>> amp = read(rfckt.amplifier,'samplebjt2.s2p');
[sparams,AllFreq] = extract(amp.AnalyzedResult,'S_Parameters');
```

(2)检查放大器的稳定性。

在进行设计之前,确定放大器无条件稳定的测量频率。用稳定性函数计算每个频率上的 mu 和 muprime。然后,检查 mu 的返回值是否大于1。该准则是无条件稳定的充分必要条件。如果放大器不是无条件稳定的,打印相应的频率值。

```
>> [mu,muprime] = stabilitymu(sparams);
figure
plot(AllFreq/1e9,mu,'--',AllFreq/1e9,muprime,'r') %效果如图 5-19 所示
legend('MU','MU\prime','Location','Best')
title('稳定性参数 MU 和 MU\prime')
xlabel('频率(GHz)')
>> disp('测量频率,放大器不是无条件稳定:')
fprintf('\t 频率 = %.1e\n',AllFreq(mu<=1))
测量频率,放大器不是无条件稳定:</pre>
```

第5章

射频损耗

应

用

#### - MATLAB无线通信系统建模与仿真

频率 = 1.0e+09 频率 = 1.1e+09



图 5-19 放大器稳定性效果图

对于本例,放大器在除 1.0GHz 和 1.1GHz 以外的所有测量频率下都是无条件稳定的。 (3)确定同时共轭匹配的源和负载匹配网络。

通过将放大器接口上同时共轭匹配的反射系数转换为适当的源和负载导纳,设计输入 和输出匹配网络。本例使用如图 5-20 所示的无损传输线匹配模型。



该单一存根匹配方案的设计参数是存根的位置与放大器接口和存根的长度。该程序 使用了以下设计原则。

• 史密斯图的中心代表一个标准化的源或负载阻抗。

• 沿传输线运动相当于绕以史密斯图原点为圆心,半径等于反射系数大小的圆。

 当传输线导纳(传输线)与单位电导圆相交时,可以在传输线上插入单个传输线短 节。在这个位置,存根将抵消传输线电纳,导致电导率等于负载或源端。

这个例子使用了 YZ Smith 图表,因为使用这种类型的史密斯图表可以更容易地添加 存根和传输线。

(4) 计算并绘制复杂负载和源反射系数。

计算并绘制所有复杂负载和源反射系数,同时共轭匹配所有测量频率数据点的无条件 稳定。这些反射系数是在放大器接口上测量的。

>> AllGammaL = calculate(amp, 'GammaML', 'none');

```
AllGammaS = calculate(amp, 'GammaMS', 'none');
hsm = smithplot([AllGammaL{:} AllGammaS{:}]);
hsm.LegendLabels = {'#Gamma ML', '#Gamma MS'};
运行程序,效果如图 5-21 所示。
(5) 在单一频率下确定负载反射系数。
求设计频率 1.9GHz 下输出匹配网络的负载反射系数 GammaL。
>> freq = AllFreq(AllFreq == 1.9e9);
GammaL = AllGammaL{1}(AllFreq == 1.9e9)
GammaL =
-0.0421 + 0.2931i
```

(6) 绘制负载反射系数 GammaL 等幅圆。

画一个以归一化导纳史密斯图原点为中心的圆,其半径等于 GammaL 的大小。圆上的一 点表示传输线上某一特定位置的反射系数。放大器接口处传输线的反射系数为 GammaL,图 中心为归一化负载导纳 y L。该实例使用圆方法在史密斯图表上绘制所有适当的圆。

运行程序,效果如图 5-22 所示。



图 5-21 复杂负载和源反射系数图



图 5-22 负载反射系数 GammaL 等幅圆

(7) 画单位恒定电导圆,求交点。

为了确定短波长(电纳)及其相对于放大器负载匹配接口的位置,绘制归一化单位电导圆和等幅圆,并计算出两个圆的交点。

第5章

射频损耗

应

用

在 MATLAB 中,可使用数据游标交互式地找到交点,或者使用 helper 函数 find\_circle \_intersections\_helper 进行解析。本例使用 helper 函数,求圆相交于两点。这个例子使用第 三象限点,标记为"A"。单位电导圆的圆心为(-0.5,0),半径为 0.5。等幅圆以(0,0)为中 心,半径等于 GammaL 的幅值。

```
>> circle(amp, freq, 'G', 1, hsm);
hsm.ColorOrder(2,:) = [1 0 0];
[~,pt2] = imped_match_find_circle_intersections_helper([0 0], ...
     abs(GammaL), [-.50],.5);
GammaMagA = sqrt(pt2(1)^2 + pt2(2)^2);
GammaAngA = atan2(pt2(2),pt2(1));
plot(pt2(1), pt2(2), 'k.', 'MarkerSize', 16);
txtstr = sprintf('A = \\mid % s\\mid % s^\\circ', num2str(GammaMagA, 4), ...
     num2str(GammaAngA * 180/pi,4));
text(pt2(1),pt2(2) - .07,txtstr, 'FontSize',8, 'FontUnits', 'normalized', ...
     'FontWeight', 'Bold')
annotation('textbox', 'VerticalAlignment', 'middle',...
     'String',{'单位','电导','圆'},...
     'HorizontalAlignment', 'center', 'FontSize', 8, ...
     'EdgeColor', [0.04314 0.5176 0.7804],...
      'BackgroundColor', [1 1 1], 'Position', [0.1403 0.1608 0.1472 0.1396])
annotation('arrow', [0.2786 0.3286], [0.2778 0.3310])
annotation('textbox', 'VerticalAlignment', 'middle',...
     'String',{'常数','幅值','圆'},...
     'HorizontalAlignment', 'center', 'FontSize', 8, ...
     'EdgeColor', [0.04314 0.5176 0.7804],...
     'BackgroundColor', [1 1 1], 'Position', [0.8107 0.3355 0.1286 0.1454])
annotation('arrow', [0.8179 0.5761], [0.4301 0.4887]);
hold off
```



图 5-23 单位恒定电导圆及交点

运行程序,效果如图 5-23 所示。

(8) 计算输出匹配网络的 Stub 位置和 Stub 长度。

放大器负载接口的波长中开路的短波位 置是点 A 和 GammaL 之间顺时针角差的函 数。当点 A 出现在第三象限,伽马值出现在 第二象限时,短波在波长中的位置计算如下:

```
>> StubPositionOut = ((2 * pi + GammaAngA)
- angle(GammaL))/(4 * pi)
StubPositionOut =
    0.2147
```

存根值是将归一化负载导纳(史密斯图的中心)移动到等幅圆上指向 A 所需要的电纳 量。可以用一个开路短传输线来提供这个电纳值。它的波长定义为从史密斯图上的开路 导纳点(以下图中的点 M)到图外边缘所需的电纳点 N 的角旋转量。点 N 为常数电纳圆, 其值等于点 A 的电纳与单位圆相交。此外,下面使用的 StubLengthOut 公式要求 N 落在 第三或第四象限。

```
第5章 射频损耗应用------
```

```
>> GammaA = GammaMagA * exp(1j * GammaAngA);
bA = imag((1 - GammaA)/(1 + GammaA));
StubLengthOut = - atan2(-2 * bA/(1 + bA^2),(1 - bA^2)/(1 + bA^2))/(4 * pi)
StubLengthOut =
        0.0883
```

(9) 计算输入匹配网络的存根位置和存根长度。

下面代码计算了输出匹配传输网络所需的波长和位置,按照同样的方法,计算输入匹 配网络的行长度。

```
>> GammaS = AllGammaS{1}(AllFreq == 1.9e9)
GammaS =
    - 0.0099 + 0.2501i
>> StubLengthIn = - atan2(-2 * bA/(1 + bA^2),(1 - bA^2)/(1 + bA^2))/(4 * pi)
StubLengthIn =
        0.0759
>> [pt1,pt2] = imped_match_find_circle_intersections_helper([0 0], ...
        abs(GammaS),[-.5 0],.5);
GammaMagA = sqrt(pt2(1)^2 + pt2(2)^2);
GammaAngA = atan2(pt2(2),pt2(1));
GammaAngA = GammaMagA * exp(1j * GammaAngA);
bA = imag((1 - GammaA)/(1 + GammaAngA) - angle(GammaS))/(4 * pi)
StubPositionIn =
        0.2267
```

```
(10) 验证设计。
```

为验证设计,采用 50Ω 微带传输线为匹配网络组装电路。首先,通过分析 1.9GHz 设 计频率下的默认微带传输线,确定微带线是否是合适的选择。

```
>> stubTL4 = rfckt.microstrip;
analyze(stubTL4,freq);
Z0 = stubTL4.Z0;
```

该特性阻抗接近期望的 50Ω 阻抗,因此本例可以使用这些微带线进行设计。为了计算 存根放置所需的传输线长度(以米为单位),分析微带线以获得相位速度值。

```
>> phase_vel = stubTL4.PV;
```

利用相位速度值确定传输线波长和存根位置,为 TL2 和 TL3 两条微带传输线设置适 当的传输线长度。

```
>> TL2 = rfckt.microstrip('LineLength',phase_vel/freq*StubPositionIn);
TL3 = rfckt.microstrip('LineLength',phase_vel/freq*StubPositionOut);
% 再次使用相位速度为每个存根指定存根长度和存根模式
>> stubTL1 = rfckt.microstrip('LineLength',phase_vel/freq*StubLengthIn, ...
'StubMode','shunt','Termination','open');
set(stubTL4,'LineLength',phase_vel/freq*StubLengthOut, ...
'StubMode','shunt','Termination','open')
% 将电路元件串级,分析 1.5 ~ 2.3GHz 频率范围内有无匹配网络的放大器
>> matched_amp = rfckt.cascade('Ckts',{stubTL1,TL2,amp,TL3,stubTL4});
analyze(matched_amp,1.5e9:1e7:2.3e9);
```

#### -- MATLAB无线通信系统建模与仿真

```
analyze(amp, 1.5e9:1e7:2.3e9);
8为了验证放大器输入端的同时共轭匹配,绘制匹配和不匹配电路的 S11 参数(dB)
>> clf
plot(amp, 'S11', 'dB')
hold all
hline = plot(matched amp, 'S11', 'dB');
                                       %效果如图 5-24 所示
hline.Color = 'r';
legend('S_{11} - 原始的放大器', 'S_{11} - 匹配的放大器')
legend('Location', 'SouthEast')
hold off
%为了验证放大器输出的同时共轭匹配,绘制匹配和不匹配电路的 S22 参数(dB)
>> plot(amp, 'S22', 'dB')
hold all
hline = plot(matched amp, 'S22', 'dB');
                                       %效果如图 5-25 所示
hline.Color = 'r';
legend('S {22} - 原始的放大器', 'S {22} - 匹配的放大器')
legend('Location', 'SouthEast')
hold off
%最后,绘制传感器增益(Gt)和匹配电路的最大可用增益(Gmaq)的 dB 值
>> hlines = plot(matched amp, 'Gt', 'Gmag', 'dB'); %效果如图 5-26 所示
hlines(2).Color = 'r';
-5
                                         -5
-10
                                        -10
```



图 5-24 匹配和不匹配电路的 S11 参数效果图



图 5-25 匹配和不匹配电路的 S22 参数效果图



图 5-26 传感器增益和匹配电路的最大可用增益的 dB 值

由图 5-26 可以看出,传感器增益和最大可用增益在 1.9GHz 时非常接近。

## 5.5 射频电路对象

【例 5-31】 此实例展示了如何创建和使用射频工具箱电路对象。

在这个例子中,创建了三个电路(rfckt)对象:两条传输线和一个放大器。可以使用射 频工具箱函数可视化放大器数据,并检索从文件中读取到放大器 rfckt 对象的频率数据。 然后在不同的频率范围内分析放大器,并将结果可视化。接下来,将这三个电路串联起来, 创建一个级联的 rfckt 对象。然后分析级联网络,并在放大器的原始频率范围内可视化其 S 参数。最后,绘制级联网络的 S11、S22 和 S21 参数和噪声图。实现步骤如下。

(1) 创建 rfckt 对象。

创建三个电路对象:使用 amplifier 函数创建两条传输线和一个使用默认数据的放 大器。

```
>> FirstCkt = rfckt.txline;
SecondCkt = rfckt.amplifier('IntpType','cubic');
read(SecondCkt,'default.amp');
ThirdCkt = rfckt.txline('LineLength',0.025,'PV',2.0e8);
(2) 杳看 rfckt 对象的属性。
```

可以使用 get 函数来查看三个对象的属性。例如:

```
>> PropertiesOfFirstCkt = get(FirstCkt) %第一个对象属性
PropertiesOfFirstCkt =
  包含以下字段的 struct:
         LineLength: 0.0100
           StubMode: 'NotAStub'
       Termination: 'NotApplicable'
               Freq: 1.0000e + 09
                 Z0: 50.0000 + 0.0000i
                PV: 299792458
              Loss: 0
           IntpType: 'Linear'
             nPort: 2
     AnalyzedResult: []
              Name: 'Transmission Line'
>> PropertiesOfSecondCkt = get(SecondCkt) %第二个对象属性
PropertiesOfSecondCkt =
  包含以下字段的 struct:
         NoiseData: [1×1 rfdata.noise]
     NonlinearData: [1 × 1 rfdata.power]
           IntpType: 'Cubic'
       NetworkData: [1×1 rfdata.network]
             nPort: 2
     AnalyzedResult: [1×1 rfdata.data]
               Name: 'Amplifier'
>> PropertiesOfThirdCkt = get(ThirdCkt) % 第三个对象属性
PropertiesOfThirdCkt =
  包含以下字段的 struct:
         LineLength: 0.0250
```

第5章

射频损耗

应用

#### MATLAB无线通信系统建模与仿真

```
StubMode: 'NotAStub'
Termination: 'NotApplicable'
Freq: 1.0000e + 09
Z0: 50.0000 + 0.0000i
PV: 200000000
Loss: 0
IntpType: 'Linear'
nPort: 2
AnalyzedResult: []
Name: 'Transmission Line'
```

(3) 列出 rfckt 对象的方法。

可以使用 methods 函数来列出对象的方法。例如:

>> MethodsOfThirdCkt = methods(ThirdCkt);

(4) 更改 rfckt 对象的属性。

使用 get 函数或点符号来获得第一个传输线的线路长度。

```
>> DefaultLength = FirstCkt.LineLength;
%使用集合函数或点符号来更改第一个传输线的线路长度
>> FirstCkt.LineLength = .001;
NewLength = FirstCkt.LineLength;
```

(5) 绘制放大器 S11 和 S22 参数。

使用电路对象的 smithplot 方法将放大器(SecondCkt)的原始 S11 和 S22 参数绘制在 Z Smith 图上。放大器 S 参数的原始频率范围为 1.0~2.9GHz。

>> figure smithplot(SecondCkt,[11;22]); %效果如图 5-27 所示

(6) 绘制放大器 Amplifier 数据。

使用电路对象的绘图方法在 X-Y 平面上绘制 2.1GHz 的放大器(SecondCkt)数据,以 dBm 为单位。

>> plot(SecondCkt,'Pout','dBm') %效果如图 5-28 所示
>> legend('show','Location','northwest');



图 5-27 原始 S11 和 S22 参数效果图



图 5-28 放大器 Amplifier 数据效果图

(7) 原始频率数据和对放大器进行原始频率分析的结果。

当射频工具箱从默认值读取数据时,将放大器转换成放大器对象(SecondCkt),还分析 了放大器在默认频率下的网络参数。amplifier的存储结果在属性分析结果中,这是原始放 大器频率的分析结果。

```
>> f = SecondCkt.AnalyzedResult.Freq;
data = SecondCkt.AnalyzedResult
data =
    rfdata.data with properties:
        Freq: [191 × 1 double]
        S_Parameters: [2 × 2 × 191 double]
        GroupDelay: [191 × 1 double]
        NF: [191 × 1 double]
        OIP3: [191 × 1 double]
        Z0: 50.0000 + 0.0000i
        ZS: 50.0000 + 0.0000i
        ZL: 50.0000 + 0.0000i
        IntpType: 'Cubic'
        Name: 'Data object'
```

(8) 在一个新的频率范围分析放大器并绘制其新的 S11 和 S22 参数。

为了在不同的频率范围内可视化电路的 S 参数,必须首先在该频率范围内分析它。

```
>> analyze(SecondCkt,1.85e9:1e7:2.55e9);
smithplot(SecondCkt,[11;22],'GridType','ZY'); %效果如图 5-29 所示
```

(9) 创建和分析级联 rfckt 对象。

将三个电路对象级联形成一个级联电路对象,然后在1.0~2.9GHz的放大器原始频率 上进行分析。

>> CascadedCkt = rfckt.cascade('Ckts',{FirstCkt,SecondCkt,ThirdCkt});
analyze(CascadedCkt,f);

(10) 绘制级联电路的 S11 和 S22 参数。

使用电路对象的 smithplot 方法将级联电路(CascadedCkt)的 S11 和 S22 参数绘制在 Z Smith 图上。

>> smithplot(CascadedCkt,[11;22],'GridType','Z'); %效果如图 5-30 所示



图 5-29 新的 S11 和 S22 参数频率效果图



图 5-30 级联电路的 S11 和 S22 参数效果图

第5章

射

频损耗

应用

- MATLAB无线通信系统建模与仿真

(11) 绘制级联电路的 S21 参数。

使用电路对象的绘图方法在 X-Y 平面上绘制级联电路(CascadedCkt)的 S21 参数。

>> plot(CascadedCkt,'S21','dB') %效果如图 5-31 所示 >> legend show

(12) 绘制级联电路的 S21 参数和噪声图。

使用电路对象的绘图方法绘制出级联电路(CascadedCkt)在 X-Y 平面上的 S21 参数和 噪声图。

>> plot(CascadedCkt,'budget','S21','NF') % 效果如图 5-32 所示 >> legend show



