

Tutorial Uso toolbox RF Matlab – Prof Marcelo Perotoni UFABC

Deseja-se usar os recursos do toolbox RF Matlab em um exemplo prático. O transistor 2N3866 possui um datasheet na página .

<http://www.solidstateinc.com/specsearch/specs/2N3866-66A-ssi.pdf>

Supondo que a frequência de interesse seja a de 100 MHz, as figuras de 15-18 apresentam os valores do transistor para bias VCE de 15 V e corrente quiescente 80 mA. A fig. 1 apresenta a parte real e imaginária dos parâmetros Y (no caso y_{11}), para a frequência de 100 MHz.

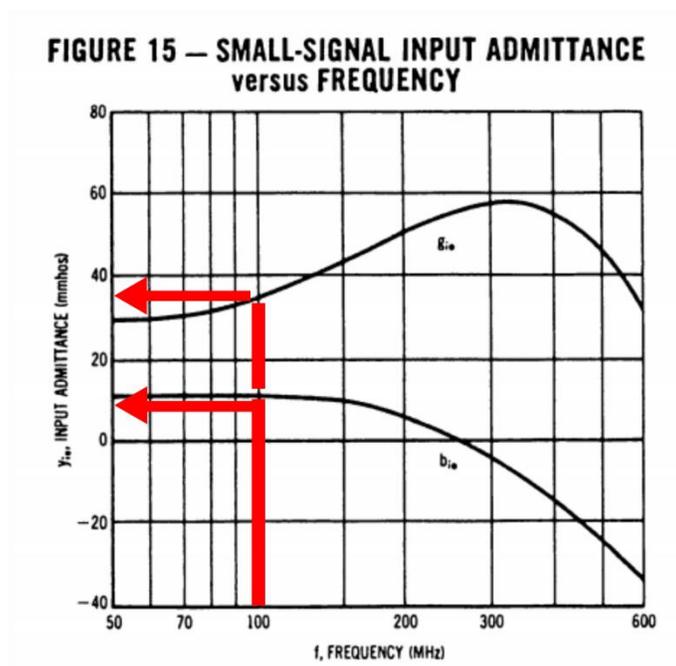
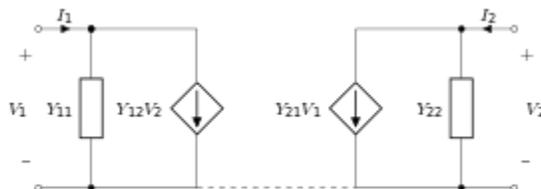


Fig. 1 Extração dos fator y_{11} .

Parâmetros Y (admitância) são definidos conforme a fig. 2 (fonte: wikipedia).



The Y-parameter matrix for the [two-port network](#) is probably the most common. In this case the relationship between the port voltages, port currents and the Y-parameter matrix is given by:

$$\begin{pmatrix} I_1 \\ I_2 \end{pmatrix} = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}.$$

where

$$Y_{11} = \left. \frac{I_1}{V_1} \right|_{V_2=0} \quad Y_{12} = \left. \frac{I_1}{V_2} \right|_{V_1=0}$$

$$Y_{21} = \left. \frac{I_2}{V_1} \right|_{V_2=0} \quad Y_{22} = \left. \frac{I_2}{V_2} \right|_{V_1=0}$$

Admittance relations [\[edit \]](#)

The input admittance of a two-port network is given by:

$$Y_{in} = Y_{11} - \frac{Y_{12}Y_{21}}{Y_{22} + Y_L}$$

where Y_L is the admittance of the load connected to port two.

Similarly, the output admittance is given by:

$$Y_{out} = Y_{22} - \frac{Y_{12}Y_{21}}{Y_{11} + Y_S}$$

where Y_S is the admittance of the source connected to port one.

Fig. 2 Parâmetros Y (fonte: https://en.wikipedia.org/wiki/Admittance_parameters)

O toolbox RF Matlab permite a definição dos parâmetros a partir da leitura de arquivos (por exemplo .S2p, de analisadores de rede) ou via direta, como no presente tutorial.

O programa exemplo Matlab a seguir realiza a conversão entre o parâmetro Y do datasheet e S.

Alternativamente, a página <http://www.daycounter.com/Calculators/S-Y-Z-Parameter-Converter-Calculator.phtml> permite a conversão online entre os parâmetros. Use-a para confirmar se os valores executados pelo programa Matlab estão corretos.

```
clc
close all

% 2N2866 input data from http://www.solidstateinc.com/specsearch/specs/2N3866-66A-ssi.pdf

% data extracted @ 80 mA, VCE = 15 VDC(figs 15,16,17,18), data for 100 MHz
freq = [1e8];
y = zeros(2,2);
y(1,1) = 35E-3 + 1i*10e-3;
y(2,1) = 80e-3 - 1i*360e-3;
y(1,2) = -1.2e-3 - 1i*1e-4;
y(2,2) = 1e-3 + 1i*4.5e-3;

hy = yparameters(y,freq)

% shows matrix Y
rfparam(hy,1,1)
rfparam(hy,1,2)
rfparam(hy,2,1)
rfparam(hy,2,2)
```

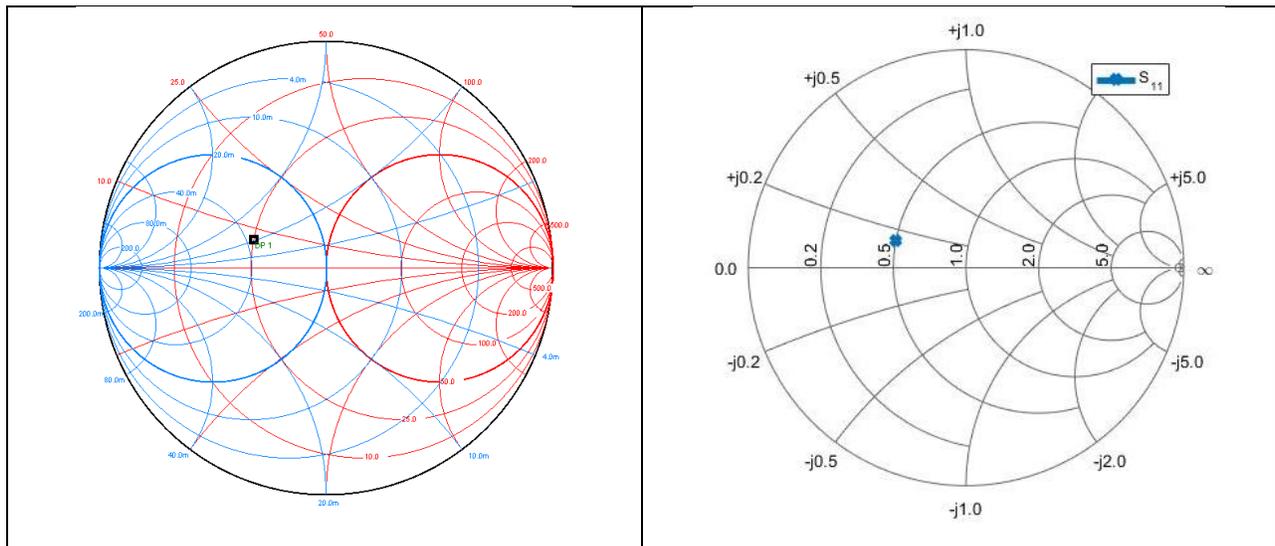
```

parametroS = sparameters(hy,50);
% mostra o componente S11
S11=rparams(parametroS,1,1);
disp('Real[S11]=')
real(S11)
disp('Imag[S11]=')
imag(S11)

figure(1);
t2=smith(parametroS,1,1);
set(t2,'LineWidth',4,'Marker','x')

```

A comparação da curva de Smith no programa Smith.exe com a carta do Matlab mostra que ambos resultados são corretos. A carta de Smith é alimentada com o ponto inicial do transistor (reflection coefficient $-0.3203+j.1266$, 100 MHz).



Uma vez disponibilizado o parâmetro S, pode-se partir para o programa da Carta de Smith de maneira a executar eventual casamento de impedância. O ponto inicial é aquele que o Matlab informa como $S_{11}=-0.32+j0.1266$. Perceba que S_{11} é tomado aqui como representando a impedância de entrada, o que é incorreto (valendo apenas para o caso em que o quadripolo é unilateral) – mas assumido como aproximado para casos em que S_{12} é pequeno. O valor rigoroso da impedância de entrada é (Pozar, Microwave Engineering):

$$\Gamma_{in} = S_{11} + \frac{S_{12}S_{21}\Gamma_L}{1 - S_{22}\Gamma_L} =$$

$$\Gamma_{out} = S_{22} + \frac{S_{12}S_{21}\Gamma_S}{1 - S_{11}\Gamma_S}$$

O transistor então está casado em 100 MHz com a inclusão de um capacitor de 50 pF em série mais uma linha de 50 Ohms em curto, em paralelo, com comprimento 0.124λ . O toolbox Matlab de RF permite a definição de demais componentes (tais como redes RLC, redes pi, T, etc). Dessa maneira o mesmo circuito analisado no programa da Carta de Smith pode ser analisado dentro do Matlab.

A função *cascadeparams* executa o cascatemaento série dos componentes. O bloco a seguir é continuação do anterior.

```
%% adds a capacitor series

c1 = capacitor(50e-12,'C1');
hckt = circuit('example2');
add(hckt,[1 2],c1)
setports(hckt, [1 0],[2 0])
capacitor_parametrosS = sparameters(hckt,freq);

%% cascade them both
s_paramsCascaded1 = cascadesparams(capacitor_parametrosS,parametroS)
figure(2);
t2=smith(s_paramsCascaded1,1,1);
set(t2,'LineWidth',4,'Marker','x')

%% add a stub in short

comprimentoLinha = 0.125*(3e8/100e6);
Stub = rfckt.txline('LineLength',comprimentoLinha,'StubMode','Shunt','Termination','Short')
Stub_Analise = analyze(Stub,freq);
Stub_parametroS = sparameters(Stub_Analise)
s_paramsCascaded2 = cascadesparams(Stub_parametroS,s_paramsCascaded1)

% cascade now the Stub to the capacitor + transistor
figure(3);
t3=smith(s_paramsCascaded2,1,1);
set(t3,'LineWidth',4,'Marker','x')
```

A figura mostra os dois passos, simultaneamente executados no Smith.exe e no Matlab. Ambos produzem o mesmo resultado.

