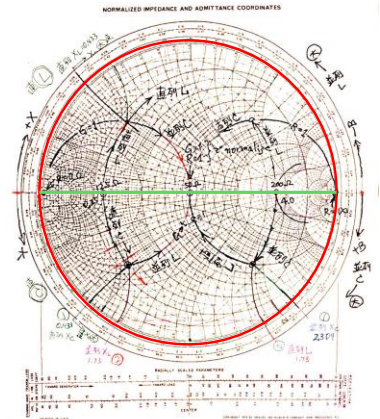


Immittance-Chart を考えながら インピーダンス・マッチングの計算 & 確認

高周波回路などでは、信号源インピーダンスと伝送路とのインピーダンス・マッチングや伝送路と終端の負荷とのインピーダンス・マッチングなど、効率よく信号電力を送るための設計が求められる。このとき良く利用されるのがイミッタンス・チャート(Impedance+Admittance)(右図)である。インピーダンス・チャートは「水橋・スミス・チャート」あるいは略して「スミス・チャート」とも呼ぶ。基本的には、マッチングと表裏一体をなす「反射」の解析に考案されたものであるが、複素インピーダンス(あるいはアドミタンス)解析においてはさまざまな場面に登場する。

本ピックアップではこのチャートの使い方だけでなく、この解析手法を根拠に置きながら、マッチング回路要素の計算とその定数の確認をLTspiceで行う。

このチャートそのものに関しては、多くの参考書が世の中にあるので、詳細な解説はそれらに譲る。

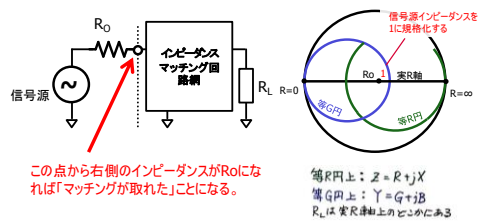


マッチングをとる手順

(信号源インピーダンスも負荷インピーダンスも純抵抗と仮定する)

インピーダンス・マッチングをとるとは...信号源(信号源インピーダンス= R_0 とする)と負荷抵抗(純抵抗で R_L とする)の間に、エネルギー・ロスのない回路網を挿入して、 R_L とこの回路網を合成したインピーダンスを R_0 に一致させればよい。

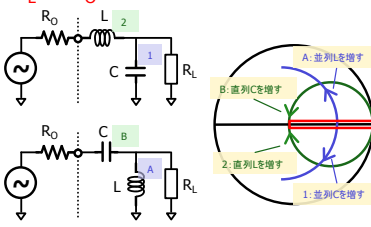
これを実現する手順は、イミッタンス・チャートの考え方に従えば、 R_L は「実R軸」上のどこかの点にあり、これを R_0 (1に規格化するので「実R軸」のチャート上の中点)にインピーダンス・チャート上で移動すればよい。移動するときには、等R円上あるいは等G円上を移動すれば、複素インピーダンス(あるいはアドミタンス)の実部は変化しないので、エネルギー・ロスはない。言い換えれば、虚数成分(リアクタンス(X)やサセプタンス(B))を適切に回路網に入れることで、実現できる。すなわち、CとLを組み合わせてマッチング回路網を作る。



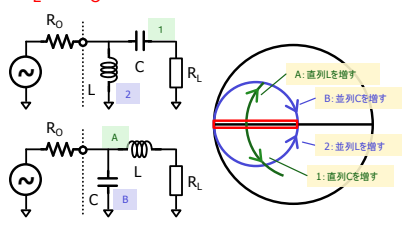
R_L が「実R軸」上の右半分か左半分かによって、CとLの構成が変わる。すなわち、等R円と等G円を組み合わせ、最終的に R_0 にたどり着くルートを探すことなので、最小のセグメント数は2...等R円と等G円を1回ずつ使って R_0 に到達するルートが選択される。

$R_L > R_0$ の場合は、初めに等G円(として、 R_L に並列)、次に等R円(Xとして、さらに直列)を使う。 $R_L < R_0$ の場合は、初めに等R円(Xとして R_L に直列)、次に等G円(さらにBとして並列)を使う。

$R_L > R_0$ のとき



$R_L < R_0$ のとき



LTspiceでXやBを計算し それらの値を実際に当てはめて確認

すでに見てきたように、 $RL > Ro$ の場合でも $RL < Ro$ の場合でも、CとLのどちらを先に配置するかによって、2通りずつ回路構成がある。どちらの場合でも、イミタンス・チャート上通るルートは上半分を通るか下半分を通るかの違いはあっても、セグメントの数は同じで、等R円と等G円の交点は互いに複素共役な位置にあることがわかる。したがって、RoとRLが決まり、さらにインピーダンスを決定する周波数が決まれば、リアクタンス(X)やサセプタンス(B)は計算できることになる。本トピックでは、計算過程の詳細は示さないが、参考回路の中に、MEASで作った計算式によって、これ等を求めるようにしてある。

<重要>

このシミュレーションは2段階で使うようになっている。

1a) パラメータを設定する。

Fo=周波数、

Ro=信号源インピーダンス、

RLL=負荷抵抗

設定した周波数を含む範囲で、

ACシミュレーション区間を設定

・・・まず[RUN]する。

1b) SPICE Error Logを開き、計算結果を読み取る。

2a) 読み取った値を、それぞれの部品名称に対応して値を書き換える。

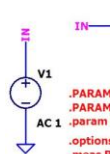
2b) シミュレーション結果をみて、所定の周波数で実部の大きさが-6.02dBで、虚部の値が0(位相が0°)であることを確認する。

参考回路図には $RL > Ro$ 用の Impedance-Matching_Cartesian_freq-1M_new_Large.asc

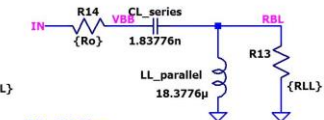
$RL < Ro$ 用の Impedance-Matching_Cartesian_freq-1M_new_Small.asc ...がある。

.ac dec 1000 900K 1.1Meg

In case of $RL > Ro$



```
.PARAM Fo=1Meg
.PARAM Ro 50
.PARAM RLL=200
.options measplxfmt=cartesian
.meas RK param mag(RLL/Ro)
.meas X_C param RK/SQRT(RK-1)
.meas BL_L param X_C*RK**2/(RK**2+X_C**2)
.meas CL_parallel param 1/(2*PI*Fo*Ro*X_C)
.meas LL_series param Ro*BL_L/(2*PI*Fo)
.meas LL_parallel param Ro*RK/(2*PI*Fo*BL_L)
.meas CL_series param 1/(2*PI*Fo*Ro*BL_L)
```



```
RLL=200 Ohm
cl_parallel: 1/(2*pi*fo*ro*x_c)=(1.37832e-09,0)
ll_series: ro*bl_l/(2*pi*fo)=(1.37832e-05,0)
ll_parallel: ro*rk/(2*pi*fo*bl_l)=(1.83776e-05,0)
cl_series: 1/(2*pi*fo*ro*bl_l)=(1.83776e-09,0)
RLL=300 Ohm
cl_parallel: 1/(2*pi*fo*ro*x_c)=(1.18627e-09,0)
ll_series: ro*bl_l/(2*pi*fo)=(1.77941e-05,0)
ll_parallel: ro*rk/(2*pi*fo*bl_l)=(2.13529e-05,0)
cl_series: 1/(2*pi*fo*ro*bl_l)=(1.42353e-09,0)
```



FAE : Michio Shibuya

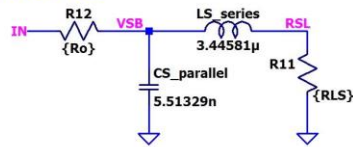
RL < Roの場合の例題

.ac dec 1000 900K 1.1Meg

In case of $RL < Ro$



```
.PARAM Fo=1Meg
.PARAM Ro 50
.PARAM RLS 12.5
.options measplxfmt=cartesian
.meas RK param mag(RLS/Ro)
.meas X_S param SQRT(RK-RK**2)
.meas BL_S param X_S/(RK**2+X_S**2)
.meas CS_series param 1/(2*PI*Fo*Ro*X_S)
.meas LS_parallel param Ro/(2*PI*Fo*BL_S)
.meas LS_series param Ro*X_S/(2*PI*Fo)
.meas CS_parallel param BL_S/(2*PI*Fo*Ro)
```



SPICE Error Logの結果をコメントとして張りつけた。

```
cs_series: 1/(2*pi*fo*ro*x_s)=(7.35105e-09,0)
ls_parallel: ro/(2*pi*fo*bl_s)=(4.59441e-06,0)
ls_series: ro*x_s/(2*pi*fo)=(3.44581e-06,0)
cs_parallel: bl_s/(2*pi*fo*ro)=(5.51329e-09,0)
```

基本的な考え方は、 $RL > Ro$ の場合と同様である。計算式に微妙な違いがある。SPICE Error Logの値の表示が10の3乗刻みではないので、位取りを間違えないように、部品の値を設定する。

<まとめ>

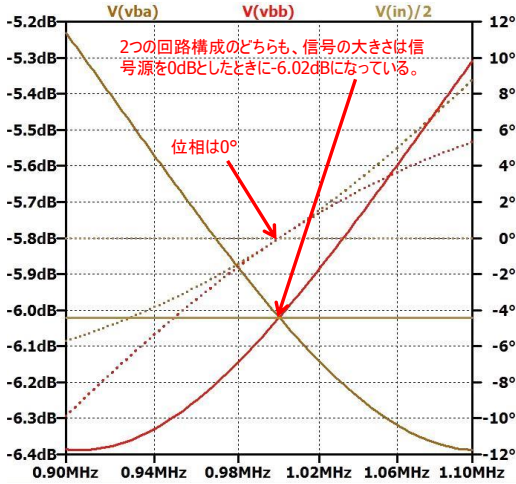
今回いままさながらイミタンス・チャートを取り上げたのは、実際に使ってみると、マッチングをとるべきインピーダンスを規格化することで(・・・というか1にしてしまうので)XやBの計算において、係数を掛けるのか割るのか、使い慣れなれていないと、うっかり間違いをする。さらに、チャートの目盛り線が見た目はものすごく細かく見えるが、実際に値を読み取ろうとすると、有効数字を2けた読み取るには結構まばらな印象がある。そこで、PCが身近に使える今日において、これを活用しない手はなかりう・・・ということで、「開数電卓」代わりにLTspiceを使ってみた次第である。



FAE : Michio Shibuya

シミュレーション結果

RL > Ro の場合 (Fo = 1MHz)



RL < Ro の場合 (Fo = 1MHz)

