Operating Point BODE100

Table of Contents

op=op_bode100(data,freq_op,freq_L,damp)

Author: JCCopyrights Summer 2019 Project: CRANE: Medical WPT for Deep Brain Stimulation Implants Creates a struct with power inductor model data extracted from raw data from Bode100 This function is created to work with the import_bode100 function. It is expeted to be used as data.op(i)=op_bode100(data,freq_op,freq_L,damp);

Parameters

- @param **data** Struct of dara extracted from Bode100.
- @param **freq_op** Frequency to be evaluated and working point of the model.(It really only affect R)
- @param **freq_L** Optional. If added is the frequency which Xs/w is exactily the L of the coil. If not added is asserted as freq_op
- @param **damp** Optional. If true the damping effect is added to the error and impdance checking of the model. If not added damping will not be considered

Code

```
function op=op_bode100(data,freq_op,freq_L,damp)

 %Look for closest frequency to freq_op
 [ trash, i_aprox ] = min(abs( data.raw.f-freq_op));
 f_aprox=data.raw.f(i_aprox);
 op.freq_op=f_aprox;
 op.raw.Rs=data.raw.Rs(i_aprox);
 op.raw.Ls=data.raw.Ls(i_aprox);
 op.raw.Z=data.raw.Z(i_aprox);
 op.raw.theta=data.raw.theta(i_aprox);
 op.raw.Q=data.raw.Q(i_aprox);
 text=sprintf('Aproximating f=%i to f_aprox=
%i',freq_op,f_aprox); %Because disp function sucks dick.
 disp(text);
 %%Resonance calculations
 [op.model.Rp, i_res]=max(data.raw.Rs); %Damping Resistance (Rfe).
 Notice that Rp is also dependent of frequency.
 if i_res<length(data.raw.f) %Check if resonance is in the measured
 range
  op.f_res=data.raw.f(i_res);
 else
  warning('Resonance freq not in measurement range. Cp is WHATEVER');
  op.f_res=1.3*data.raw.f(length(data.raw.f)); %Random shit.
 end;
```

```
% Notice that here we are ignoring the parasitic effects of Cp and R
when assuming Ls is L at freq_op
% This should be checked and studied. I am not going to. Because Fuck
You.
if nargin>2 %freq_L exist
 [ trash, i_L ] = min(abs( data.raw.f-freq_L));
 op.model.L=data.raw.Ls(i_L);
else
 op.model.L=data.raw.Ls(i_aprox);
end
op.model.Cp=1/((2*pi*op.f_res)^2*op.model.L);

%% Real Coil Model
% The parasitic effects of Cp and L over the Real impedance cannot be
ignored
% The parasitic effect of Rp near resonance cannot be ignored, but
adding Rp to the impedance eliminates
% any simple mathematical solution. So try not to work near resonance
maybe?
% @TODO: Introduce a numerical method to approximate model values
withoud analytical constrains.
w=2*pi*f_aprox;
a=w^2*op.model.Cp^2*op.raw.Rs;
b=-1;
c=op.raw.Rs*((1-w^2*op.model.L*op.model.Cp)^2);
% This equation for a R>0 has two solutions one bigger than the
other,
% The correct solution shall be the smaller one, because of math and
stuff (Convergence at 0+ when inf).
op.model.R=(-b-sqrt(b^2-4*a*c))/(2*a);
% Model impedance
Z_C=1/(j*w*op.model.Cp);
Zp=op.model.Rp;
Z_LR=op.model.R+j*w*op.model.L;
op.model.Z_model_nodamp=Z_C*Z_LR/(Z_C+Z_LR);
op.model.Z_model=op.model.Z_model_nodamp*Zp/(op.model.Z_model_nodamp
+Zp);
% Compare the model impedance with the measured impedance at freq_op
% If error>0 means model value is bigger
if nargin>3
 if damp==true
  op.error.err_rel_Real=(real(op.model.Z_model)-op.raw.Rs)/op.raw.Rs;
  op.error.err_rel_L=(imag(op.model.Z_model/w)-op.raw.Ls)/op.raw.Ls;
 else
  op.error.err_rel_Real=(real(op.model.Z_model_nodamp)-op.raw.Rs)/
op.raw.Rs;
  op.error.err_rel_L=(imag(op.model.Z_model_nodamp/w)-op.raw.Ls)/
op.raw.Ls;
 end
else
 op.error.err_rel_Real=(real(op.model.Z_model_nodamp)-op.raw.Rs)/
op.raw.Rs;
 op.error.err_rel_L=(imag(op.model.Z_model_nodamp/w)-op.raw.Ls)/
op.raw.Ls;
```

```
end
text=sprintf('Error model vs measured:\nReal %f\nImag
%f',op.error.err_rel_Real*100,op.error.err_rel_L*100); %Because disp
function sucks dick.
disp(text)
```

```
%    | - Name: Original csv file
%    |
%    |  |-f: frequency range
%    |  |-Z: Impedance in ohm
%    |- raw: |-theta: Impedance phase in degrees
%    |  |-Ls: Series Inductance
%    |  |-Rs: Real Impedance
%    |  |-Q: Quality factor
% data  |
%    |  |-freq_op: operating frequency
%    |  |-f_res: resonance frequency
%    |-op: |-raw: Contains the raw data measured at freq_op
%    |  |
%    |  |  |-L: Model inductance
%    |  |-model:|-R: Copper losses
%    |  |  |-Cp: Parasitic capactitor
%    |  |  |-Rp: Damping Resistance
%    |  |
%    |  |-error: Relative error between model and measured data
```

Published with MATLAB® R2018b