

# Speaker Response, Worked Example

Francis Deck

The purpose of this worked example is simply to check my math by comparing a design analysis with a well-respected program. I deliberately chose a bad design so its features would stand out in the graphs.

- ```
/* Thiele-Small parameters from datasheet for
Kappa Pro 10a driver, data from Eminence site */
Fs := 46: /* resonant frequency in Hz */
R := 6.5: /* DC resistance in Ohms */
Qes := 0.2: /* electrical damping factor */
Qms := 10.1: /* mechanical damping factor */
Qts := 1/(1/Qes + 1/Qms): /* total damping factor,
spec sheet says 0.32 */
"Vas in m^3"; Vas := float(52.2/1000);
"Frontal area in m^3"; Sd := float(344.9/10000);
"excursion limit in m"; Xmax := float(3.2/1000);
"inductance in H"; Le := float(1.15/1000);
```

"Vas in m^3"

0.0522

"Frontal area in m^3"

0.03449

"excursion limit in m"

0.0032

"inductance in H"

0.00115

- ```
/* Physical constants */
Patm := 101.3e3: /* Atmospheric pressure */
"adiabatic bulk modulus"; B := 1.4*Patm;
Rho := 1.18: /* Density of air in kg/m^3 */
C := 345: /* Speed of sound in m/s */
SplRef := 1e-12: /* 0 dB SPL reference in Pa */
```

"adiabatic bulk modulus"

141820.0

- ```
/* Derived electromechanical parameters, inferred from
```

```

formulas at Kyle Lahnakoski's page */
"resonant angular frequency in 1/s";
Ws := float(2*PI*Fs);
"spring constant of driver in N/m";
Km := B*Sd^2/Vas;
"mechanical resistance of driver in N/m/s";
Rm := Km/Qms/Ws;
"mass of driver in kg"; M := Km/Ws^2;
"field length product in N/A"; BL := (Km/Qes/Ws*R)^0.5;

```

"resonant angular frequency in 1/s"

289.0265241

"spring constant of driver in N/m"

3231.866157

"mechanical resistance of driver in N/m/s"

1.107118972

"mass of driver in kg"

0.03868815033

"field length product in N/A"

19.06336283

- /\* Parameters of the box \*/
 

```

"box volume in m^3"; Vbox := float(18/1000);
"port angular frequency in 1/s"; Wport := float(50*2*PI);
Rref := R; /* Reference impedance in Ohms */
"input voltage amplitude in V"; Vin := float(sqrt(1*Rref*2));
r := 1; /* Reference distance to listener, m */
"diameter of port in m"; Dport := 7.5*0.01;
"area of port in m^2"; Aport := float(PI*(Dport/2)^2);
EndCorrect := 0.732; /* Port end correction for one flush
end, stolen from WinISD */
"length of port in inches";
Zport := Aport*B/Rho/Vbox/Wport^2*EndCorrect/0.0254;
"spring constant of box in n/m"; Kbox := Sd^2*B/Vbox

```

"box volume in m^3"

0.018

"port angular frequency in 1/s"

314.1592654

6.5

"input voltage amplitude in V"

3.605551275

1

"diameter of port in m"

0.075

"area of port in m<sup>2</sup>"

0.004417864669

"length of port in inches"

8.613367867

"spring constant of box in n/m"

9372.411855

- ```
/* Reference efficiency from formulas */
"standard reference efficiency";
n0 := float(Ws^3*Vas/2/PI/C^3/Qes);
"reference SPL"; spl := 112 + 10*log(10,n0);
valim := Sd*Vin*BL/R/M;
/* Decibel SPL function */
dB := va -> 10*log(10,Rho*va^2/2/PI/C/4/PI/r/SplRef) :
"my calculated SPL"; float(dB(valim))
```

"standard reference efficiency"

0.02442398102

"reference SPL"

95.87816454

"my calculated SPL"

95.85419302

- ```
/* Excursion function, closed box system */
Xcb := proc(w)
begin
  Zcoil := R + I*w*Le;
```

```

    Kcb := Km + Kbox;
    BL*Vin/Zcoil/M/(Kcb/M - w^2 + I*w*(Rm + BL^2/Zcoil)/M)
end_proc:

```

- /\* Volume acceleration, closed box system \*/
 

```

VAcb := proc(f)
    begin
        w := float(f*2*PI);
        w^2*Sd*abs(Xcb(w))
    end_proc:

```
- /\* Excursion function, ported box system. The small imaginary term avoids a discontinuity at  $w = W_{port}$ , which causes graphing problems. \*/
 

```

Xport := proc(w)
    begin
        Zcoil := R + I*w*Le;
        Kpb := Km + Kbox*(1 + 1/(w^2/Wport^2 - 1 + I*0.000001));
        BL*Vin/Zcoil/M/(Kpb/M - w^2 + I*w*(Rm + BL^2/Zcoil)/M)
    end_proc:

```
- /\* Volume acceleration, ported system \*/
 

```

VAport := proc(f)
    begin
        w := float(f*2*PI);
        w^2*Sd*abs(Xport(w)*(1 + 1/(w^2/Wport^2 - 1)))
    end_proc:
    float(dB(VAport(40)))

```

## 82.19828813

- /\* Port air velocity as function of frequency \*/
 

```

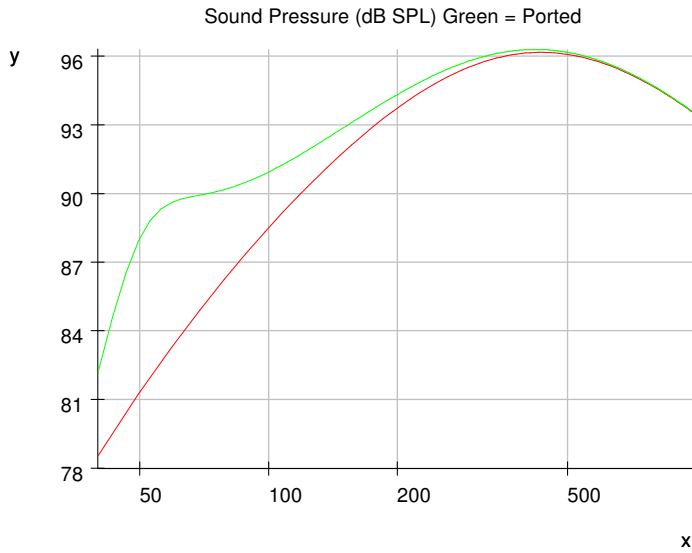
Vport := proc(f)
    begin
        w := float(f*2*PI);
        abs(w*Xport(w)*Sd/Aport/(w^2/Wport^2 - 1 + I*0.000001))
    end_proc:

```
- /\* Plots of closed and ported box frequency response functions \*/
 

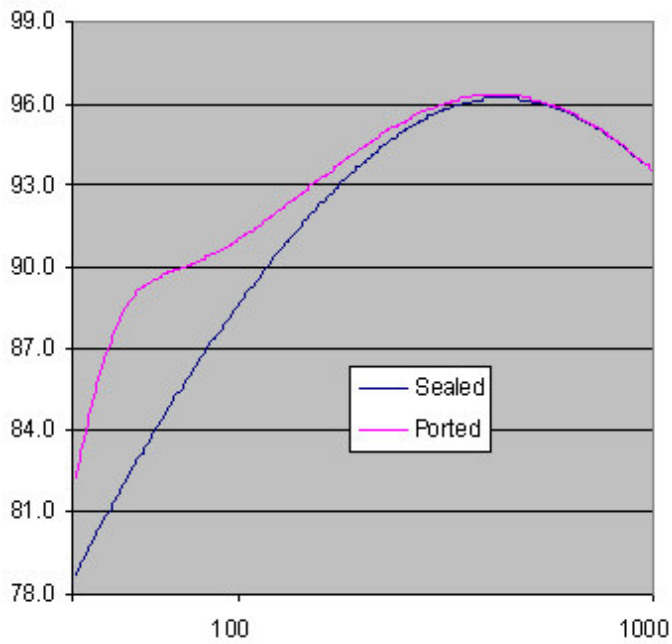
```

cbplot := plot::Function2d(dB(VAcb(f)), f=40 .. 1000,
    Color = [1,0,0]):
pbplot := plot::Function2d(dB(VAport(f)), f=40 .. 1000,
    Color = [0,1,0], Grid = [300]):
plot(cbplot, pbplot, AxesScaling = [Log,Lin],
    Title = "Sound Pressure (dB SPL) Green = Ported",
    Ticks = [[50 ,100, 200, 500], [78,81,84,87,90,93,96]],
    GridLines = [[50 ,100, 200, 500], [78,81,84,87,90,93,96]]);

```

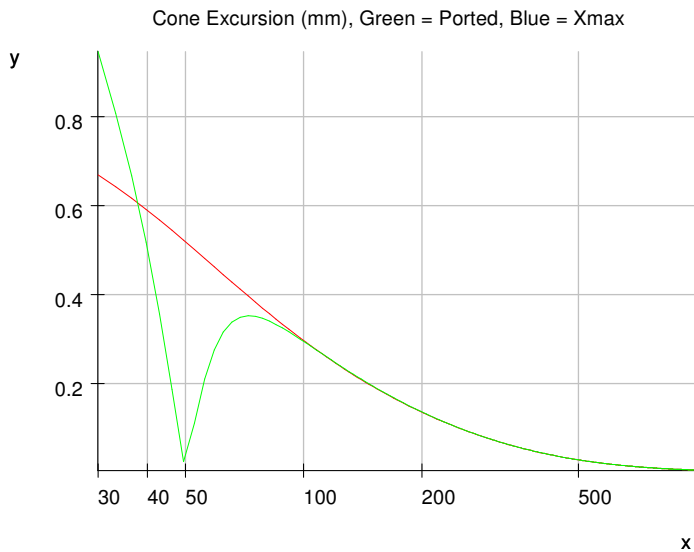


Here are the sensitivity curves for the same design, from the well-regarded UniBox program.



- ```
/* Plots of closed and ported box excursion amplitude
functions */
cxplot := plot::Function2d(1000*abs(Xcb(2*PI*f)),
f=30..1000, Color = [1,0,0], Grid = [300]):
pxplot := plot::Function2d(1000*abs(Xport(2*PI*f)),
f=30..1000, Color = [0,1,0], Grid = [300]):
/* xmpplot := plot::Function2d(1000*Xmax, f=30..1000,
Color = [0,0,1]): */
plot(cxplot, pxplot, /* xmpplot, */ AxesScaling = [Log, Lin],
```

```
Title = "Cone Excursion (mm), Green = Ported, Blue = Xmax",
Ticks = [[30, 40, 50 ,100, 200, 500],Automatic],
GridLines = [[30, 40, 50 ,100, 200, 500], Automatic]);
```



```
• /* Plot of port air speed */
asplot := plot::Function2d(Vport(f)/C, f=30..1000,
Color = [1,0,0], Grid = [300]):
plot(asplot, AxesScaling = [Log, Lin],
Title = "Port air speed (Mach)",
Ticks = [[30, 40, 50 ,100, 200, 500],Automatic],
GridLines = [[30, 40, 50 ,100, 200, 500], Automatic]);
```

