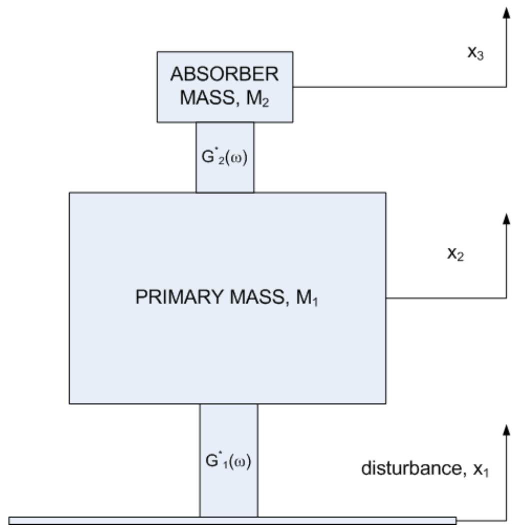
NMBD Dynamic Vibration Absorber

Derived from appendix T080194, and applied to the Non-Magnetic Blade-spring Damper (NMBD) for the Quad UIM internal blade-spring mode



coupled two-mass oscillator derivation

following J.C. Snowdon, Vibration and Shock in Damped Mechanical Systems, John Wiley & Sons, cr 1968, section 4.2, pp. 80-83.

derivation

The derivation below corrects Snowdon's equations. Snowdon's formulation for the amplitude of the transfer function from ground to primary mass motion is correct, but his real and imaginary terms are incorrect.

```
(Debug) In[1]:= eqn1[t_] := M1 x2''[t] - k1 G1[ω] (x1[t] - x2[t]) + k2 G2[ω] (x2[t] - x3[t]);

(Debug) In[2]:= eqn3 = LaplaceTransform[eqn1[t], t, s] /.
    {x1'[0] → 0, x1[0] → 0, LaplaceTransform[x1[t], t, s] → x1bar[s],
     x2'[0] → 0, x2[0] → 0, LaplaceTransform[x2[t], t, s] → x2bar[s],
     x3'[0] → 0, x3[0] → 0, LaplaceTransform[x3[t], t, s] → x3bar[s]}
```

(Debug) Out[2]= $s^2 M_1 \overline{x_2}[s] - k_1 \overline{x_1}[s] G_1[\omega] + k_1 \overline{x_2}[s] G_1[\omega] + k_2 \overline{x_2}[s] G_2[\omega] - k_2 \overline{x_3}[s] G_2[\omega]$

```
(Debug) In[3]:= eqn2[t_] := M2 x3''[t] - k2 G2[ω] (x2[t] - x3[t]);
```

(Debug) In[4]:=
```
eqn4 = LaplaceTransform[eqn2[t], t, s] /.
   {x1'[0] → 0, x1[0] → ξ, LaplaceTransform[x1[t], t, s] → x̄1[s],
    x2'[0] → 0, x2[0] → 0, LaplaceTransform[x2[t], t, s] → x̄2[s],
    x3'[0] → 0, x3[0] → 0, LaplaceTransform[x3[t], t, s] → x̄3[s]}
```

(Debug) Out[4]= $s^2 M_2 \overline{x_3}[s] - k_2 \overline{x_2}[s] G_2[\omega] + k_2 \overline{x_3}[s] G_2[\omega]$

(Debug) In[5]:=
```
soln = Solve[{eqn3 == 0, eqn4 == 0}, {x̄2[s], x̄3[s]}]
```

(Debug) Out[5]= $\{\{\overline{x_2}[s] \to -((k_1 \overline{x_1}[s] G_1[\omega] (s^2 M_2 + k_2 G_2[\omega])) / (k_2^2 G_2[\omega]^2 - (s^2 M_2 + k_2 G_2[\omega]) (s^2 M_1 + k_1 G_1[\omega] + k_2 G_2[\omega]))),$
$\overline{x_3}[s] \to (k_1 k_2 \overline{x_1}[s] G_1[\omega] G_2[\omega]) / (s^4 M_1 M_2 + s^2 k_1 M_2 G_1[\omega] + s^2 k_2 M_1 G_2[\omega] + s^2 k_2 M_2 G_2[\omega] + k_1 k_2 G_1[\omega] G_2[\omega])\}\}$

(Debug) In[6]:=
```
T[ω_] := Simplify[x̄2[s] / x̄1[s] /. soln /. s → I ω]
```

Snowdon's equation 4.7:

(Debug) In[7]:=
```
T[ω]
```

(Debug) Out[7]= $\{-((k_1 G_1[\omega] (-\omega^2 M_2 + k_2 G_2[\omega])) / (k_2^2 G_2[\omega]^2 - (-\omega^2 M_2 + k_2 G_2[\omega]) (-\omega^2 M_1 + k_1 G_1[\omega] + k_2 G_2[\omega])))\}$

Using the following substitutions / defintions

$\mu = \frac{M_1}{M_1 + M_2};$

$n = \frac{\omega_a}{\omega_0};$

$\omega_0 = \sqrt{\frac{k_1 Y_1[\omega_{00}]}{M_1 + M_2}};$

$\omega_a = \sqrt{\frac{k_2 Y_2[\omega_{00}]}{M_2}};$

$\Omega = \frac{\omega}{\omega_0};$

and thefollowing notation for the complex modulus :

(Debug) In[8]:=
```
G1[ω_] := Y1 (1 + I δ1);
G2[ω_] := Y2 (1 + I δ2);
```

The expression for the transfer function is as follows:

(Debug) In[10]:=
```
T[ω]
```

(Debug) Out[10]= $\{(k_1 Y_1 (-i + \delta_1) (\omega^2 M_2 + k_2 Y_2 (-1 - i \delta_2))) / (k_1 Y_1 (-i + \delta_1) (\omega^2 M_2 + k_2 Y_2 (-1 - i \delta_2)) + \omega^2 k_2 M_2 Y_2 (-i + \delta_2) + \omega^2 M_1 (i \omega^2 M_2 + k_2 Y_2 (-i + \delta_2)))\}$

Transfer function,: Real Numerator Term

(Debug) In[11]:= R_N = FullSimplify[Re[Expand[T[ω][[1, {1, 2, 3, 4}]]]], {k_1, k_2, Y_1, Y_2, δ_1, δ_2, μ, w0, wa, Ω, ω, M_1, M_2} ∈ Reals]

(Debug) Out[11]= $k_1 Y_1 \left(\omega^2 M_2 \delta_1 - k_2 Y_2 (\delta_1 + \delta_2)\right)$

(Debug) In[12]:= c = -n^2 / (k_1 k_2 Y_1 Y_2)

(Debug) Out[12]= $-\frac{n^2}{k_1 k_2 Y_1 Y_2}$

(Debug) In[13]:= Simplify[c R_N]

(Debug) Out[13]= $n^2 \left(\left(1 - \frac{\omega^2 M_2}{k_2 Y_2}\right) \delta_1 + \delta_2\right)$

(Debug) In[14]:= Simplify[Expand[c R_N] //. ω^2 n^2 M_2 / (k_2 Y_2) → Ω^2 (Y_a / Y_2)]

(Debug) Out[14]= $\left(n^2 - \frac{\Omega^2 Y_a}{Y_2}\right) \delta_1 + n^2 \delta_2$

This is the same as Snowdon's Equation 4.13 for I_N

Transfer function,: Imaginary Numerator Term

(Debug) In[15]:= I_N = FullSimplify[Im[Expand[T[ω][[1, {1, 2, 3, 4}]]]], {k_1, k_2, Y_1, Y_2, δ_1, δ_2, μ, w0, wa, Ω, ω, M_1, M_2} ∈ Reals]

(Debug) Out[15]= $-k_1 Y_1 \left(\omega^2 M_2 + k_2 Y_2 (-1 + \delta_1 \delta_2)\right)$

(Debug) In[16]:= Simplify[c I_N]

(Debug) Out[16]= $n^2 \left(-1 + \frac{\omega^2 M_2}{k_2 Y_2} + \delta_1 \delta_2\right)$

(Debug) In[17]:= Simplify[Expand[c I_N] //. ω^2 n^2 M_2 / (k_2 Y_2) → Ω^2 (Y_a / Y_2)]

(Debug) Out[17]= $\frac{\Omega^2 Y_a}{Y_2} + n^2 (-1 + \delta_1 \delta_2)$

This is the negative of Snowdon's Equation 4.12 for R_N

Transfer function,: Real Denominator Term

(Debug) In[18]:= R_D = FullSimplify[Re[1 / T[ω][[1, 5]]], {k_1, k_2, Y_1, Y_2, δ_1, δ_2, μ, w0, wa, Ω, ω, M_1, M_2} ∈ Reals]

(Debug) Out[18]= $\omega^2 k_2 (M_1 + M_2) Y_2 \delta_2 + k_1 Y_1 \left(\omega^2 M_2 \delta_1 - k_2 Y_2 (\delta_1 + \delta_2)\right)$

(Debug) In[19]:= Simplify[c R_D]

(Debug) Out[19]= $-\left(\left(n^2 \left(\omega^2 k_2 (M_1 + M_2) Y_2 \delta_2 + k_1 Y_1 \left(\omega^2 M_2 \delta_1 - k_2 Y_2 (\delta_1 + \delta_2)\right)\right)\right) / (k_1 k_2 Y_1 Y_2)\right)$

(Debug) In[20]:=
```
FullSimplify[Expand[Expand[c R_D //.
      {ω^2 (M₁ + M₂) → (k₁ Y₁) Ω^2 (Y₀ / Y₁), ω^2 n^2 M₂ / (k₂ Y₂) → Ω^2 (Yₐ / Y₂)}]] //.
  {ω^2 (M₁ + M₂) → (k₁ Y₁) Ω^2 (Y₀ / Y₁), ω^2 n^2 M₂ / (k₂ Y₂) → Ω^2 (Yₐ / Y₂)}]
```

(Debug) Out[20]= $\left(n^2 - \frac{\Omega^2 Y_a}{Y_2}\right) \delta_1 + \frac{n^2 \left(-\Omega^2 Y_0 + Y_1\right) \delta_2}{Y_1}$

This is the Snowdon's Equation 4.15 for I_D

Transfer function,: Imaginary Denominator Term

(Debug) In[21]:=
```
I_D = FullSimplify[Im[1 / T[ω][[1, 5]]],
  {k₁, k₂, Y₁, Y₂, δ₁, δ₂, μ, w0, wa, Ω, ω, M₁, M₂} ∈ Reals]
```

(Debug) Out[21]= $-\omega^2 k_2 M_2 Y_2 + M_1 \left(\omega^4 M_2 - \omega^2 k_2 Y_2\right) - k_1 Y_1 \left(\omega^2 M_2 + k_2 Y_2 (-1 + \delta_1 \delta_2)\right)$

(Debug) In[22]:=
```
Collect[Expand[c I_D] //.
   {ω^2 M₁ / (k₁ Y₁) + ω^2 M₂ / (k₁ Y₁) → Ω^2 (Y₀ / Y₁), ω^2 n^2 M₂ / (k₂ Y₂) → Ω^2 (Yₐ / Y₂),
    n^2 ω^4 M₁ M₂ / (k₁ k₂ Y₁ Y₂) → μ Ω^4 (Y₀ / Y₁) (Yₐ / Y₂)}, n] /.
 {ω^2 M₁ / (k₁ Y₁) + ω^2 M₂ / (k₁ Y₁) → Ω^2 (Y₀ / Y₁)}
```

(Debug) Out[22]= $\frac{\Omega^2 Y_a}{Y_2} - \frac{\mu \Omega^4 Y_0 Y_a}{Y_1 Y_2} + n^2 \left(-1 + \frac{\Omega^2 Y_0}{Y_1} + \delta_1 \delta_2\right)$

This is the negative of Snowdon's Equation 4.14 for R_D

---

coupled two-mass oscillator formulation

Same formulation as above but restated without subscripts.

general transmissibility equation

The UIM blade spring (D060237) has a mass of .367 kgm. I'll assume ~1/2 of this mass is involved in the 1st resonance.
The mass of the NMBD involved in its 1st elastic mode is ~2 x the mass of the dumb-bells (D1400296) = 2 x 17 = 34 gm
See NMBD damping results in T1600046

(Debug) In[50]:=
```
.055 / M1
```

(Debug) Out[50]= 0.299728

(Debug) In[23]:=
```
n = wa / w0;
M1 = .367 / 2;
M2 = eps M1;
mu = M1 / (M1 + M2);
```

(Debug) In[27]:=
```
d1[w_] := 1/720;
d2[w_] := d;
Y1[w_] := w0^2 (M1 + M2);
Y2[w_] := wa^2 M2;
```

Variables:

d = loss factor

eps = damping mass to primary mass ratio, M2/M1

wa = natural frequency of the damping element by itself, rad/sec

w0 = natural frequency of the primary mass without the damping element added, rad/sec

The total mass of the NMBD = 55 gm. However the mass of the dumb-bells involved in its 1st mode is ~34gm

So eps = .034/M1 = 0.185

wa was estimated in T1500237 to be ~120 Hz (indirectly by fitting to some data)

However using viton elastic stiffness, the cross-sectional area of the viton spring and the dumbbell mass, one gets wa = 344 Hz

From T1600046, w0 = 110 Hz and d1 = 1/720

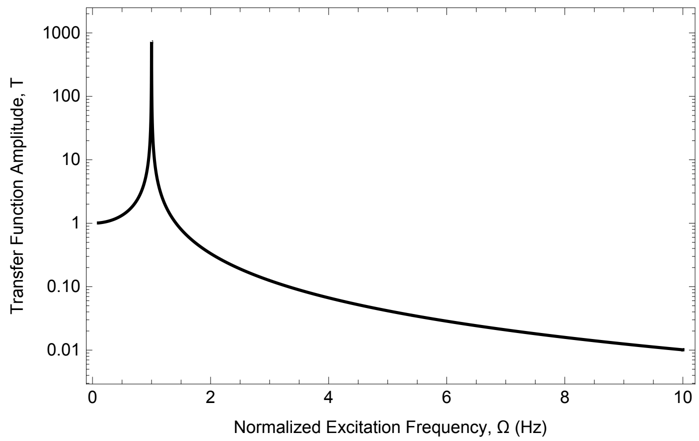
(Debug) In[31]:=
```
IN[w_] := (w/w0)^2 (Y2[wa]/Y2[w]) - n^2 (1 - d1[w] d2[w]);
RN[w_] := -(w/w0)^2 (Y2[wa]/Y2[w]) d1[w] + n^2 (d1[w] + d2[w]);
ID[w_] := -mu (w/w0)^4 (Y1[w0]/Y1[w]) (Y2[wa]/Y2[w]) +
    (w/w0)^2 (n^2 (Y1[w0]/Y1[w]) + (Y2[wa]/Y2[w])) - n^2 (1 - d1[w] d2[w]);
RD[w_] := -(w/w0)^2 (n^2 (Y1[w0]/Y1[w]) d2[w] + (Y2[wa]/Y2[w]) d1[w]) +
    n^2 (d1[w] + d2[w]);
T[w_] := (RN[w] + I IN[w]) / (RD[w] + I ID[w]);
```

NMBD application

case 1: UIM blade spring, no damper

(Debug) In[74]:=
```
plt1 = LogPlot[Abs[T[Ω w0]] /. {w0 -> 2 Pi 110, wa -> 2 Pi 344, d -> 0, eps -> 0},
  {Ω, .1, 10}, PlotRange -> All, PlotPoints -> 10 000,
  MaxRecursion -> 15, PlotStyle → {Black}, Frame → True,
  FrameLabel → {"Normalized Excitation Frequency, Ω (Hz)",
    "Transfer Function Amplitude, T"}]
```

(Debug) Out[74]=



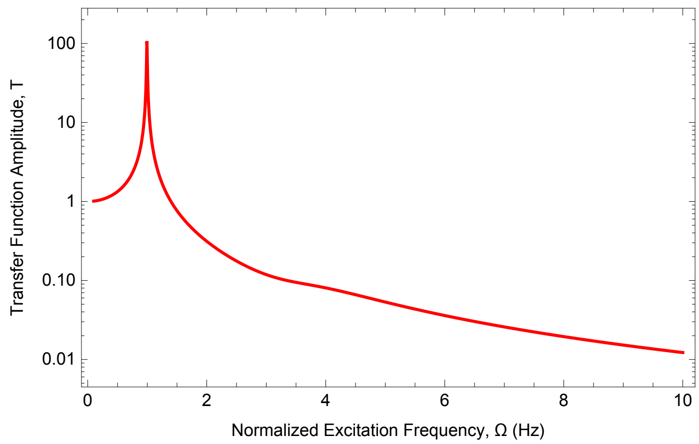
(Debug) In[75]:=
```
FindMaximum[Abs[T[ω]] /. {w0 → 2 Pi 110, wa → 2 Pi 344, d → 0, eps → 0}, {ω, 10, 500}]
```

(Debug) Out[75]=
```
{720.001, {ω → 691.15}}
```

case 2: NMBD, wa >> w0, 75 F (23.8 C)

(Debug) In[76]:=
```
plt2 =
 LogPlot[Abs[T[Ω w0]] /. {w0 -> 2 Pi 110, wa -> 2 Pi 344, d -> .68, eps -> .185},
  {Ω, .1, 10}, PlotRange -> All, PlotPoints -> 10 000,
  MaxRecursion -> 15, PlotStyle → {Red}, Frame → True,
  FrameLabel → {"Normalized Excitation Frequency, Ω (Hz)",
    "Transfer Function Amplitude, T"}]
```

(Debug) Out[76]=



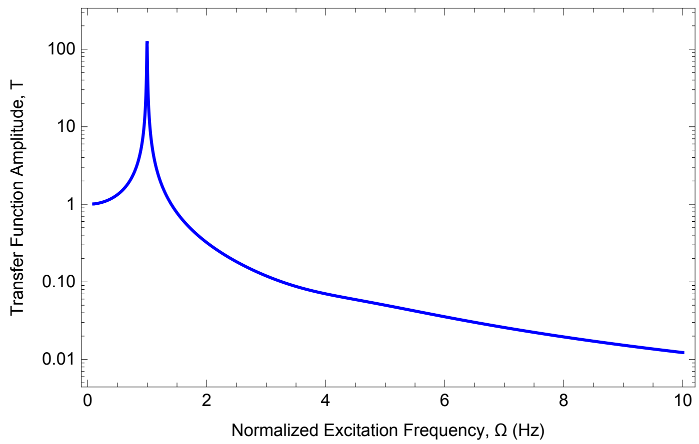
(Debug) In[77]:=
```
FindMaximum[
 Abs[T[ω]] /. {w0 → 2 Pi 110, wa → 2 Pi 344, d → 0.68, eps → 0.185}, {ω, 10, 500}]
```

(Debug) Out[77]= {102.624, {ω → 687.264}}

case 3: NMBD, wa >> w0, 65 F (18.3 C)

(Debug) In[78]:=
```
plt3 = LogPlot[
  Abs[T[Ω w0]] /. {w0 -> 2 Pi 110, wa -> 2 Pi 344 × 1.119, d -> .9, eps -> .185},
  {Ω, .1, 10}, PlotRange -> All, PlotPoints -> 10 000,
  MaxRecursion -> 15, PlotStyle → {Blue}, Frame → True,
  FrameLabel → {"Normalized Excitation Frequency, Ω (Hz)",
    "Transfer Function Amplitude, T"}]
```

(Debug) Out[78]=



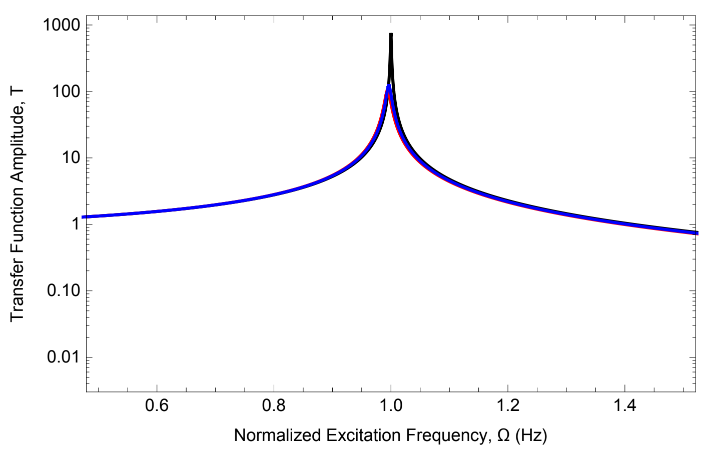
(Debug) In[79]:=
```
FindMaximum[
 Abs[T[ω]] /. {w0 → 2 Pi 110, wa → 2 Pi 344 × 1.119, d → 0.9, eps → 0.185}, {ω, 10, 500}]
```

(Debug) Out[79]= {121.554, {ω → 688.693}}

compare NMBD cases 1,2,3

(Debug) In[80]:=
```
Show[plt1, plt2, plt3, PlotRange → {{.5, 1.5}, All}]
```

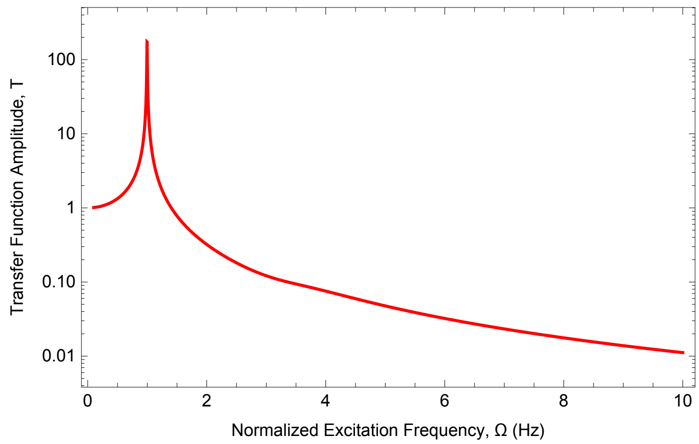
(Debug) Out[80]=



case 4: NMBD, wa >> w0, 75 F (23.8 C), d & M2 set to match T1600046-v2 Qdamped

(Debug) In[83]:=
```
plt4 = LogPlot[
  Abs[T[Ω w0]] /. {w0 -> 2 Pi 110, wa -> 2 Pi 344, d -> .6, eps -> .185/2},
  {Ω, .1, 10}, PlotRange -> All, PlotPoints -> 10 000,
  MaxRecursion -> 15, PlotStyle → {Red}, Frame → True,
  FrameLabel → {"Normalized Excitation Frequency, Ω (Hz)",
   "Transfer Function Amplitude, T"}]
```

(Debug) Out[83]=



(Debug) In[82]:=
```
FindMaximum[
 Abs[T[ω]] /. {w0 → 2 Pi 110, wa → 2 Pi 344, d → 0.6, eps → 0.185/2}, {ω, 10, 500}]
```

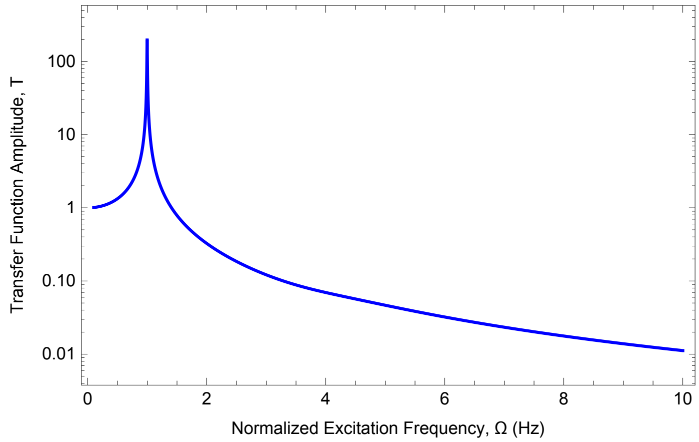
(Debug) Out[82]=
```
{172.987, {ω → 688.855}}
```

case 5: NMBD, wa >> w0, 65 F (23.8 C), d & M2 set to match T1600046-v2 Qdamped

Using d & M2 that match T160046-v2 Qdamped at 75F, but then estimating effect of temperature of 65 F, by increasing d by 1.32 factor and increasing wa frequency by 1.119 factor, predicted from Viton-B complex modulus variation with temperature

(Debug) In[84]:=
```
plt5 = LogPlot[Abs[T[Ω w0]] /.
    {w0 -> 2 Pi 110, wa -> 2 Pi 344 × 1.119, d -> .6 × 1.32, eps -> .185 / 2},
   {Ω, .1, 10}, PlotRange -> All, PlotPoints -> 10 000,
   MaxRecursion -> 15, PlotStyle → {Blue}, Frame → True,
   FrameLabel → {"Normalized Excitation Frequency, Ω (Hz)",
     "Transfer Function Amplitude, T"}]
```

(Debug) Out[84]=



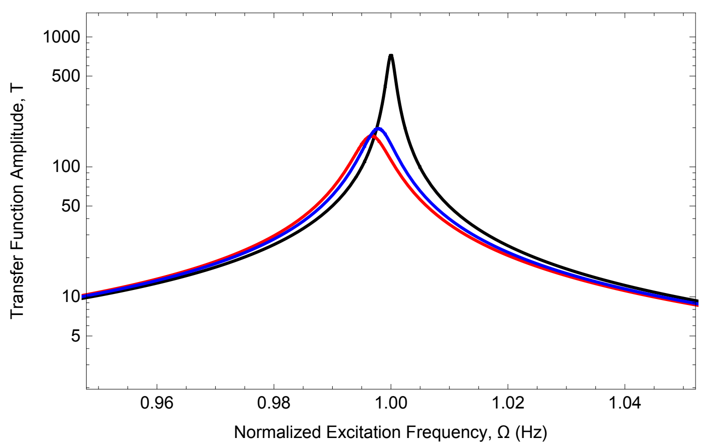
(Debug) In[85]:=
```
FindMaximum[Abs[T[ω]] /.
   {w0 → 2 Pi 110, wa → 2 Pi 344 × 1.119, d → 0.6 × 1.32, eps → 0.185 / 2}, {ω, 10, 500}]
```

(Debug) Out[85]= {196.895, {ω → 689.656}}

compare NMBD cases 1,4,5

(Debug) In[94]:=
```
Show[plt1, plt4, plt5, PlotRange → {{.95, 1.05}, {1, 7}}]
```

(Debug) Out[94]=



(Debug) In[96]:=
```
197 / 173 // N
```

(Debug) Out[96]= 1.13873

If the NMBD were assembled at 75