Notes and Reflections

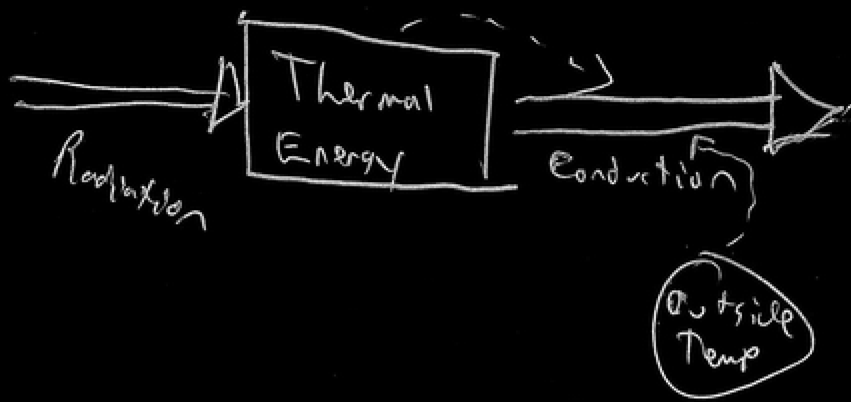
I used this bset to do several things:

1) **Check my understanding of frequency-domain reasoning by applying it to different types of systems**
2) **Practice working in the frequency domain in "solving mode" rather than "learning mode". In particular, I used computational tools to solve lots of the problems I solved on paper in previous bsets. Doing it on paper was nice, but in the real world, I'm not likely to use CTFT's if it requires pulling out a sheet of paper and transforming equations by hand.**
3) **Continue my attempts to "half-time" QEA this semester. I was able to finish all the assigned problems in around four hours, which is the ratio I've been trying to maintain. I'm completely fine with this arrangement.**
4) **Rediscover the beauty of *Mathematica*. Being able to throw DSolveValue[] at any problem involving a differential equation and get a symbolic answer out is *amazing*, and I'm reaffirming my commitment to always make sure I have a license to *Mathematica* or an equivalent.**

Table of Contents

Solutions begin on next page

1. Time-domain solar house

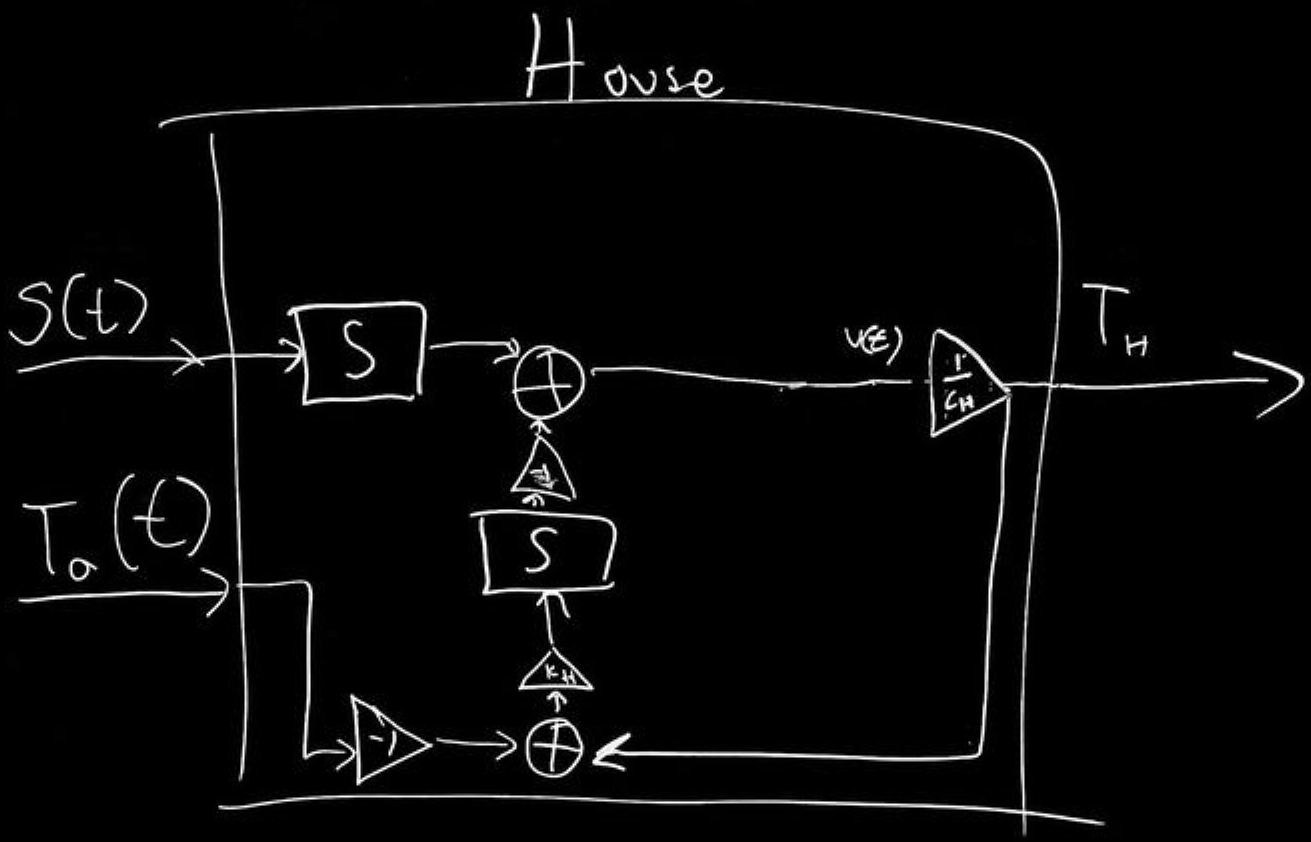


$$T_H(t) = \frac{U(t)}{C_H}$$

$$\frac{dU}{dt} = S(t) - k_H(T_H - T_o)$$

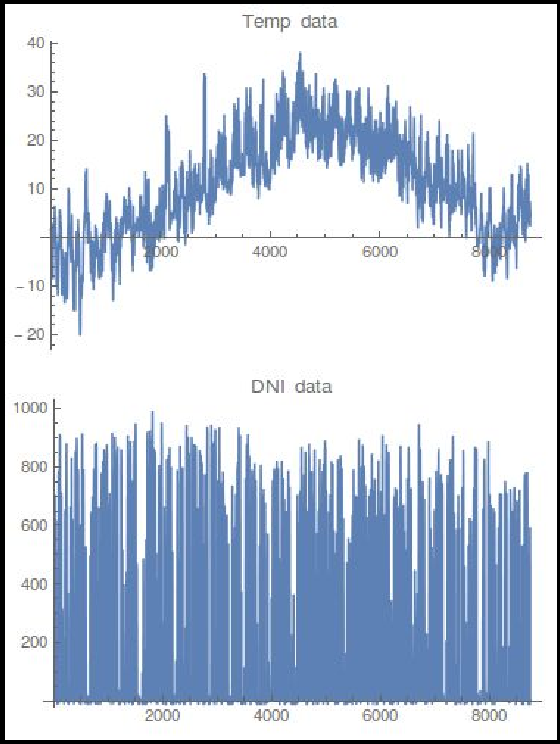
$$\frac{dU}{dt} = S(t) - k_H\left(\frac{U(t)}{C_H} - T_o(t)\right)$$

c. SigSys representation

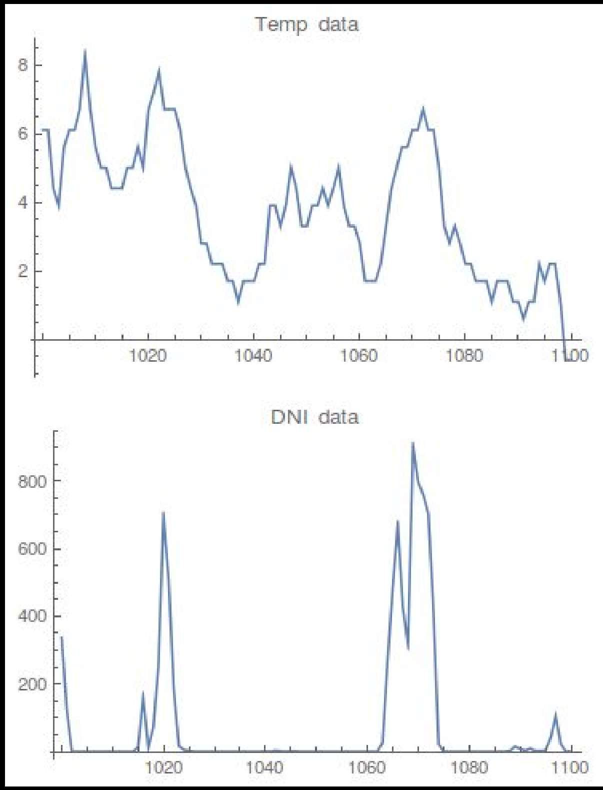


2. Frequency-domain solar house

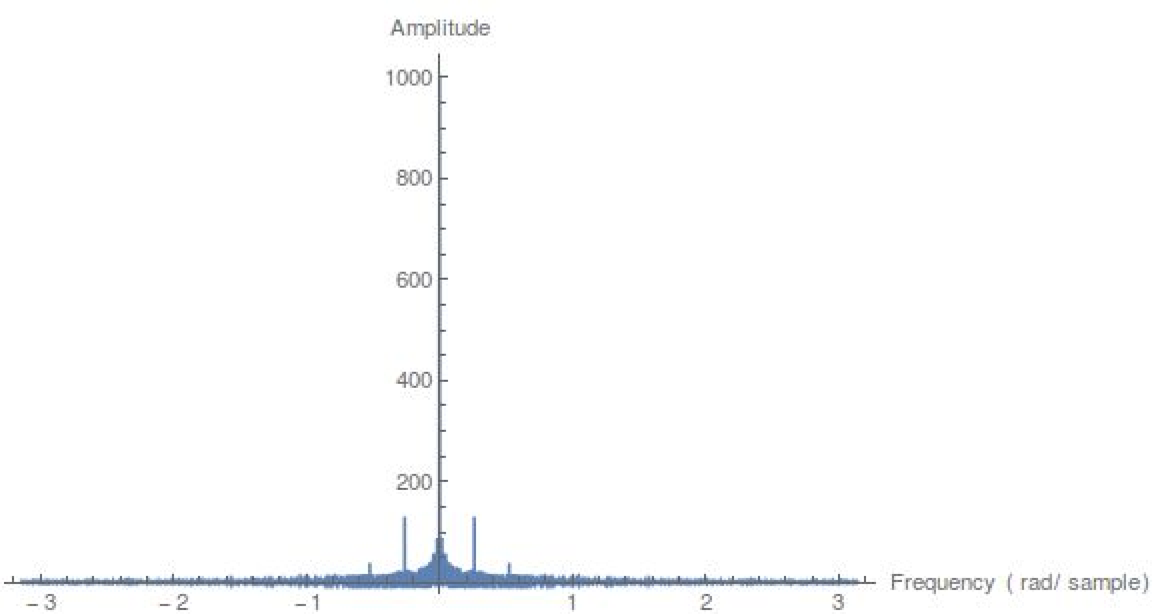
Yearly data

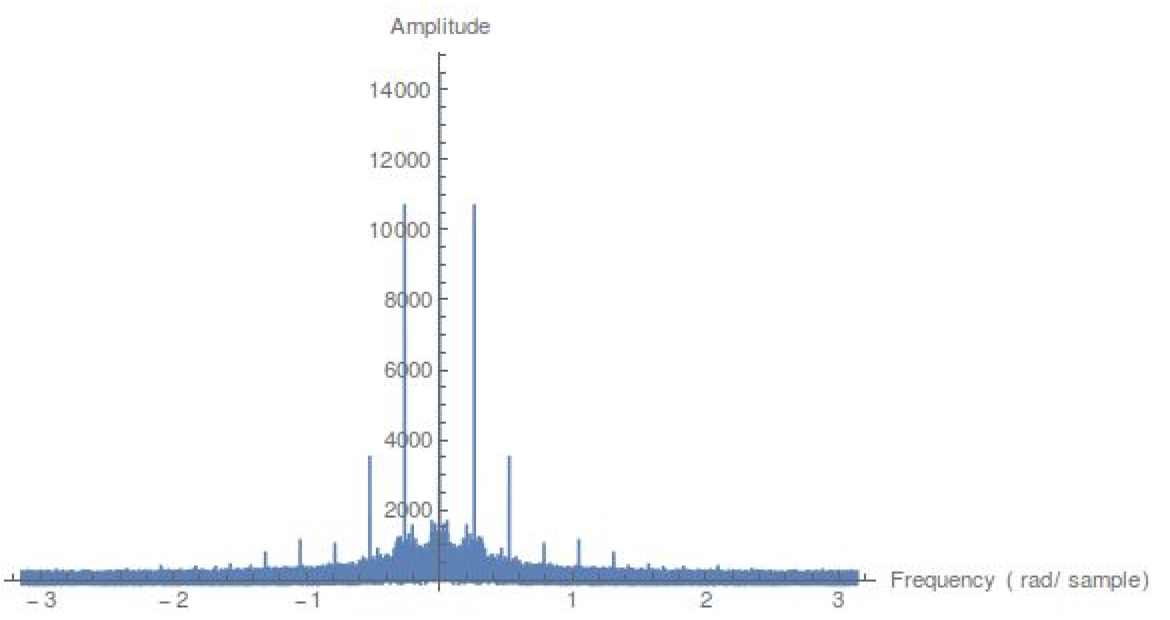


Daily data



Temperature data should have strong spikes on a yearly and daily timescale, the daily one being the biggest that will show up on a DFT. Incident solar power also has dominant frequencies at the daily level, but will have substantial high-frequency components because it is so spiky.





In direct defiance of directions, I didn't edit my plotFFT() function to label the axes properly. The correct x-axis label goes from -12 cycles / day to 12 cyles/day, with the 1/day value aligning to the spikes in both graphs living around ¼ radians/sample.

The DNI data has extra spikes at overtones of the 1/day frequency because the solar radiation skips a day much more often than temperature does. Skipping a day is manifested as a 2-day period cycle.

Converting to Frequency Domain

$$y'(t) = x(t) - \frac{k_H}{C_H} y(t)$$

$$j\omega Y(\omega) = X(\omega) - \frac{k_H}{C_H} Y(\omega)$$

$$(j\omega + \frac{k_H}{C_H}) Y(\omega) = X(\omega)$$

$$H(\omega) = \frac{Y(\omega)}{X(\omega)} = \frac{1}{j\omega + \frac{k_H}{C_H}}$$

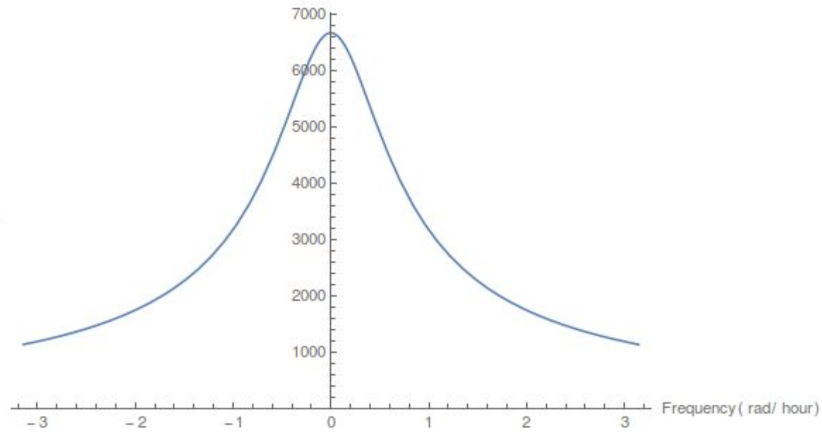
I spent a long time stuck at this point because I didn't believe that this transfer function was correct. In particular, I expected H -> 1 as ω -> 0, but this doesn't approach that. I now realize that it doesn't *need* to because of the units mismatch of exterior temperature and interior energy.

Calculating through frequency domain

Transfer function plot (vs frequency in radians/hour)

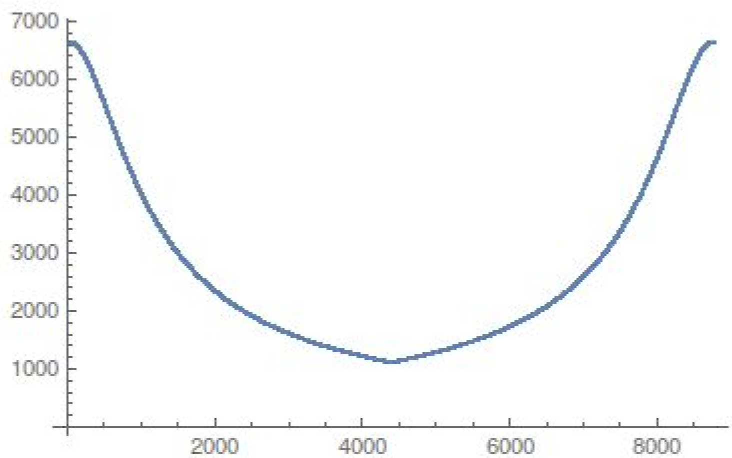
```
In[96]:= scalar = UnitSimplify[1 s / 1 h];
Plot[Abs@h[ω*scalar], {ω, -Pi, Pi}, AxesLabel → {"Frequency (rad/hour)"}
 PlotRange → {0, Automatic}]
```

Out[97]=




```
In[132]:= hlist = RotateRight[h /@ (scalar * Subdivide[-Pi, Pi
ListPlot[Abs@hlist]
```

Out[133]=

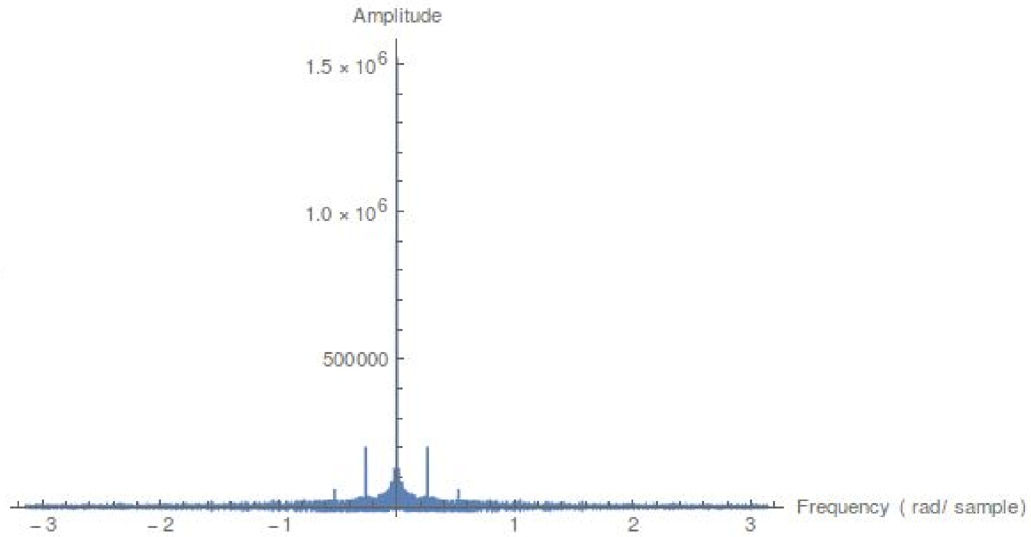


Note: it is important to make sure the H[] function and FFT are working on the same scale. Before performing the RotateRight, I was getting a highpass filter, which is the opposite of desired behavior.

Input data frequency domain plot

```
In[108]:= x = dni + kh temp;
plotFFT @ x
```

Out[109]=

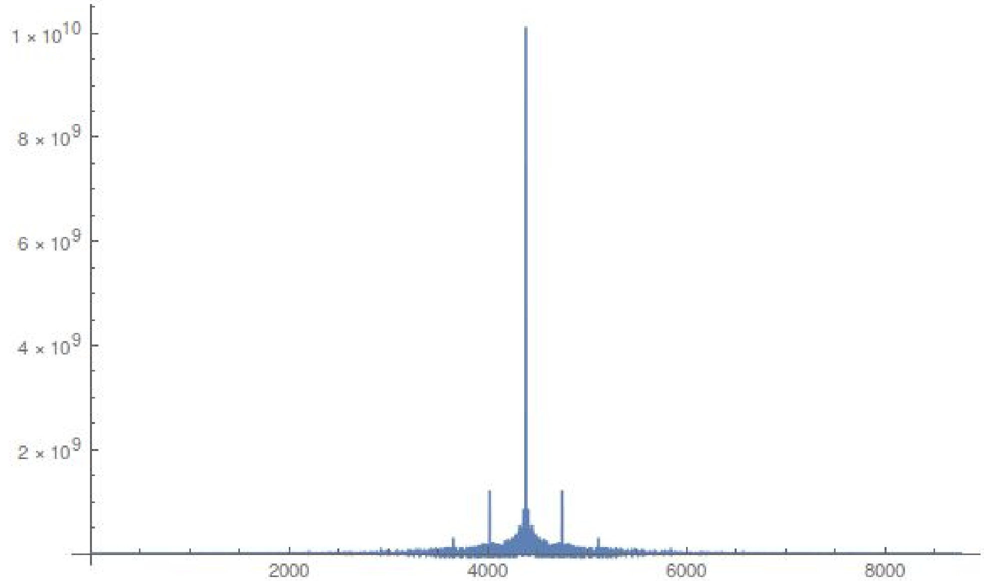


Compute and plot the product of the two

```
In[134]:= f = Fourier[x];
Length @ f
ftrans = hlist * f;
ListLinePlot[RotateRight[Abs @ ftrans, Length @ f / 2], PlotRange → Full, Image
```

```
Out[135]= 8760
```

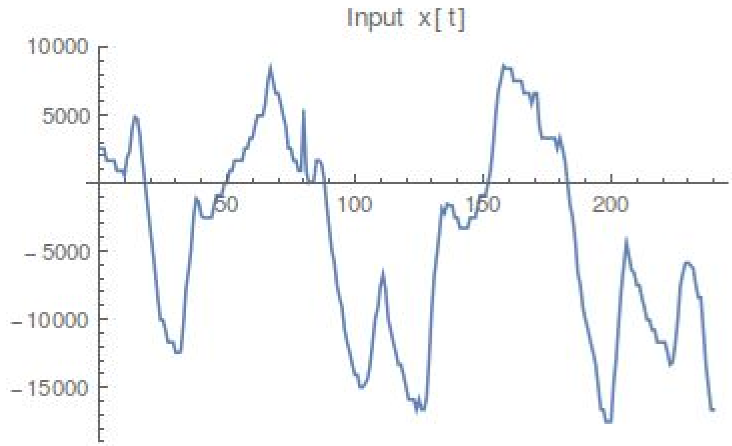
Out[137]=



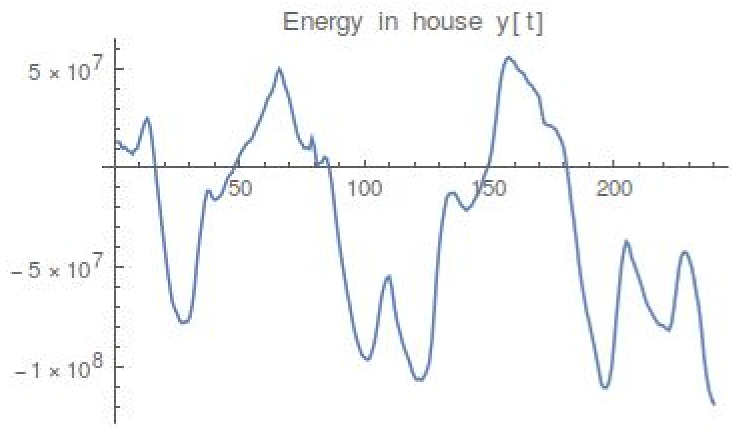
Convert back to time domain

In[141]:=
```
interiorEnergy = Re @ InverseFourier @ ftrans;
ListLinePlot[x[[1 ;; 24 * 10]], PlotLabel → "Input  x[t]"]
ListLinePlot[interiorEnergy[[1 ;; 24 * 10]], PlotLabel → "E
```

Out[142]=



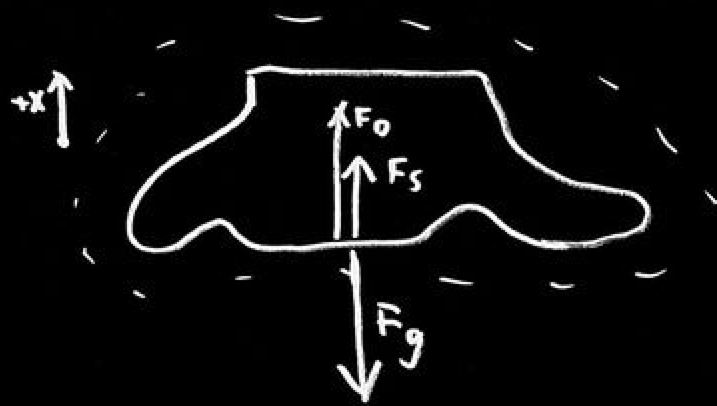
Out[143]=



This result implies that, as expected, the house behaves as a low-pass filter of its input level. In order to stabilize the temperature more, increasing thermal mass C_H would suffice, as derived from the transfer function. Unfortunately, because of the linear nature of the transfer function, no amount of smoothing will change the fact that x[t] is currently averaging <0C for the winter months. To fix this, insulation would need to increase or more sunlight would need to be absorbed.

It bothers me that we collapsed our two input signals into one outside of the "house" system rather than inside it. My block diagram formalism didn't do this, and I don't fully understand why yours did. Because of this decision, it became harder to reason about the effect of added insulation since it was baked into the input signal. Does the assumption that our system is LTI guarantee that we can always pull this combination out of the system? Do I need complex coefficients to do it properly?

3. Time-domain suspension

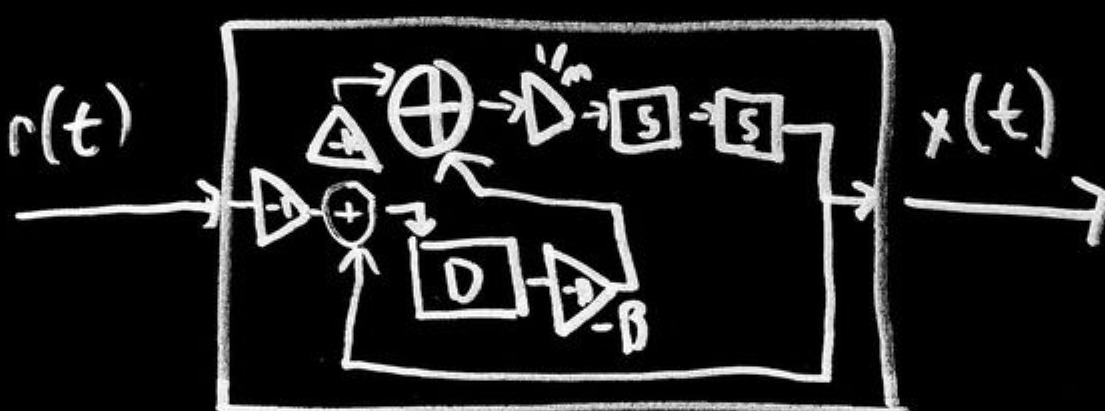


$$F_g = mg \qquad F_s = -k\left(x(t) - r(t)\right) + mg$$

$$F_D = -\beta \frac{d}{dt}\left(x(t) - r(t)\right)$$

Newton's 2nd $\rightarrow$ $ma = -F_g + F_s + F_D$

$$ma = -k\left(x(t) - r(t)\right) - \beta \frac{d}{dt}\left(x(t) - r(t)\right)$$



Simulation

Out[191]//TraditionalForm=

$$m\,x''(t) = -k\,(x(t) - r(t)) - B\,(x'(t) - r'(t))$$

Out[192]//TraditionalForm=

$$r(t) = a\cos\left(\frac{2\pi t v}{L}\right)$$

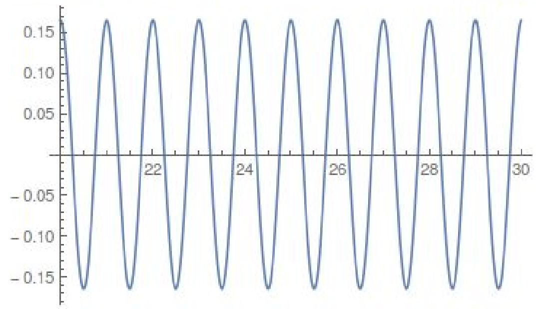
Setup values of declared variables

In[155]:=
```
ClearAll@consts
consts[speed_] := <| a → 1/10, L → 10, v → speed, k → 10^5, B → 10^3, m -> 10^3|>
consts[] = consts[10];
```

Solve the Differential Equation for given values (Note, this is entirely symbolic)

In[206]:=
```
sol = DSolveValue[{de1, de2, x'[0] == x[0] == 0} /. consts[], x[t], {t, 0, 30}];
Plot[sol /. t → x, {x, 20, 30}, PlotRange → Full]
fourier = FullSimplify@FourierTransform[sol, t, w]
```

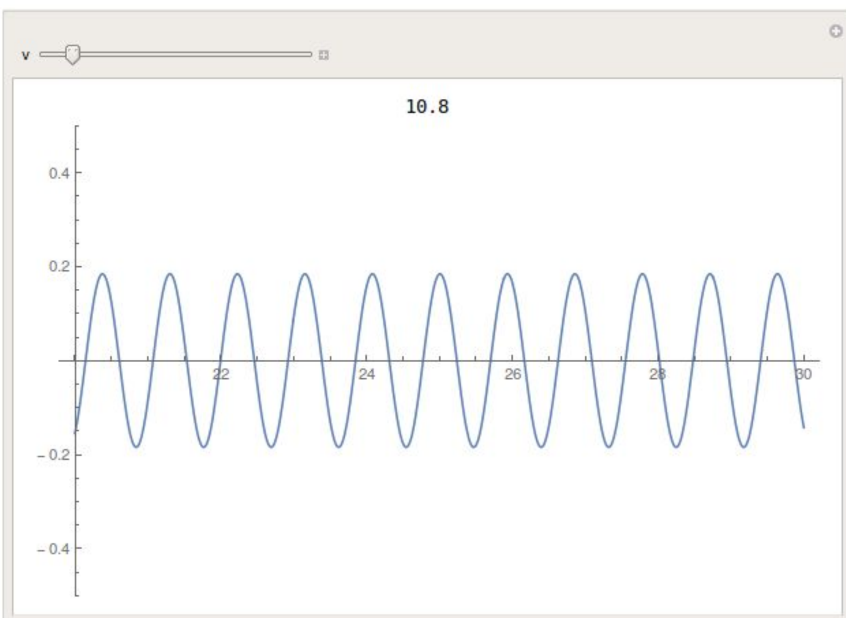
Out[207]=



Play with the value of v (Note, this is numeric because it makes the manipulate[] faster.

In[205]:=
```
Manipulate[
 sol = NDSolveValue[{de1, de2, x'[0] == x[0] == 0} /. consts[v], x[t], {t, 0, 30}];
 Grid[{{v}, {Plot[sol /. t → x, {x, 20, 30}, ImageSize → Large, PlotRange → .5 {-1, 1}]}}],
 {v, 0, 100}]
```

Out[205]=



The car's oscillations increase in amplitude with increasing speed until reaching a peak around 16m/s, after which they decrease in amplitude with increasing speed. This makes sense if 16hz is near the natural frequency of the system, because at high speeds the oscillation of the road is almost entirely absorbed by the suspension.

4. Frequency-domain suspension

$$H(\omega) = \frac{k + j\omega\beta}{k + j\omega\beta - m\omega^2}$$

```
In[209]:= del // TraditionalForm
```

Out[209]//TraditionalForm=

$$m\, x''(t) = -k\,(x(t) - r(t)) - \mathrm{B}\,(x'(t) - r'(t))$$

```
In[214]:= Block[{r, x}, r[t_] := Exp[I ω t];
  x[t_] := (k + I ω B) / (k + I ω B - m ω^2) Exp[I ω t];
  del // TraditionalForm]
 FullSimplify @ %
```

Out[214]//TraditionalForm=

$$-\frac{m\,\omega^2\, e^{i\,t\,\omega}\,(k + i\,\mathrm{B}\,\omega)}{i\,\mathrm{B}\,\omega + k - m\,\omega^2} = -k\left(\frac{e^{i\,t\,\omega}\,(k + i\,\mathrm{B}\,\omega)}{i\,\mathrm{B}\,\omega + k - m\,\omega^2} - e^{i\,t\,\omega}\right) - \mathrm{B}\left(\frac{i\,\omega\, e^{i\,t\,\omega}\,(k + i\,\mathrm{B}\,\omega)}{i\,\mathrm{B}\,\omega + k - m\,\omega^2} - i\,\omega\, e^{i\,t\,\omega}\right)$$

Out[215]= True

Qualitatively, this transfer function seems reasonable. It goes to 1 as omega -> 0 and 0 as omega -> inf, and has similar qualitative behavior to the previous analysis. To prove this, I asked *Mathematica* to simplify the initial differential equation with the assumption that $r(t) = e^{j\,\omega\,t}$ and $x(t) = H(\omega)\, e^{j\,\omega\,t}$, which returned True, validating that the equation is satisfied with these assumptions.

This problem is a great example of where Mathematica shines, I was able to try several different ways of plugging these things in to see how they behaved with much greater ease than otherwise possible.

Finding transfer function

Generate a transfer function for x[t]-r[t]

```
In[45]:= Block[{r, y},
  (*r[t_] := E^(I*t*ω);
  x[t_]:=c E^(I*t*ω);
  y[t_]:=x[t]-r[t];*)
  DSolveValue[de1, (x[t] - r[t]) / r[t], t]]
 FullSimplify[% /. C[_] → 0]
 ExportString[%, "TeXFragment"]
```

Out[45]=
$$\frac{c\, e^{i t \omega} - \frac{c\, e^{-\frac{k t}{B}+\frac{t (k+i B \omega)}{B}} (k+\omega (i B-m \omega))}{k+i B \omega} - e^{-\frac{k t}{B}} C[1]}{\frac{c\, e^{-\frac{k t}{B}+\frac{t (k+i B \omega)}{B}} (k+\omega (i B-m \omega))}{k+i B \omega} + e^{-\frac{k t}{B}} C[1]}$$

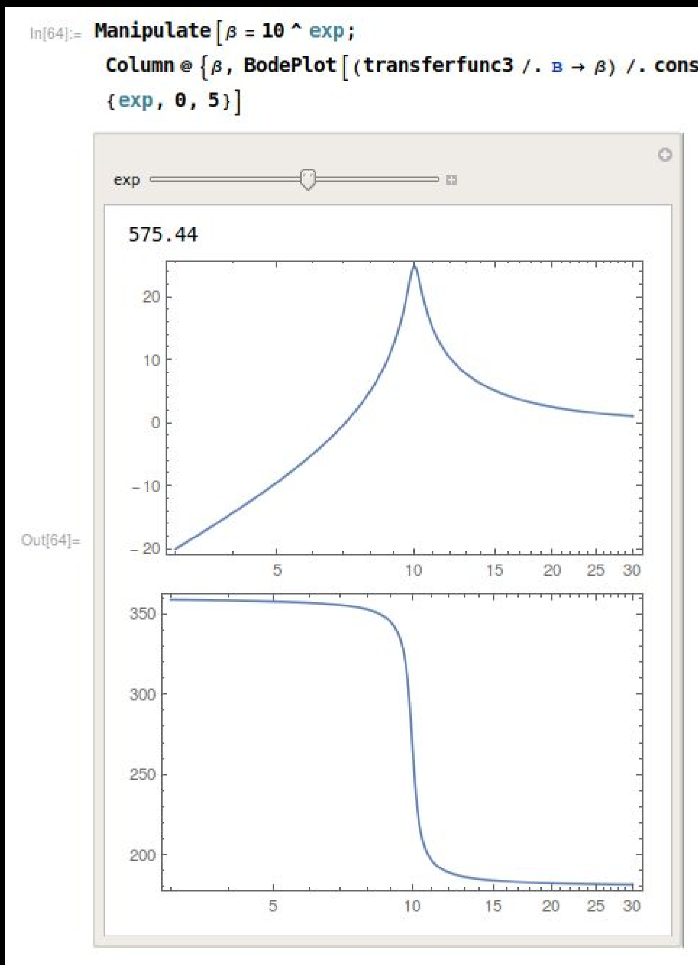
Out[46]=
$$\frac{m \omega^2}{k + i B \omega - m \omega^2}$$

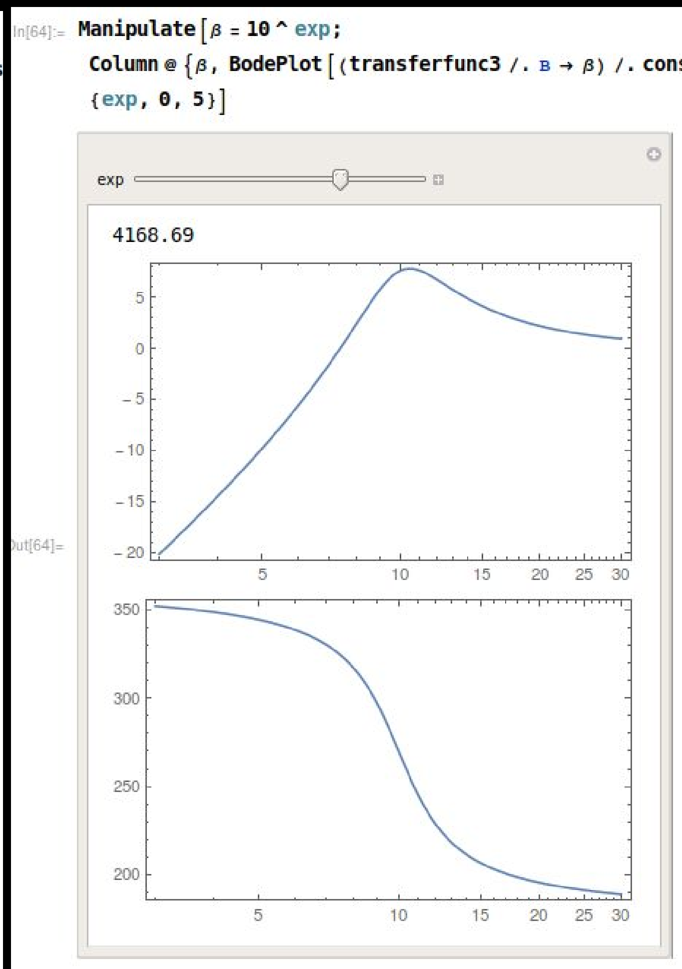
```
Out[47]= \[\frac{m \omega ^2}{k+i B \omega -m \omega ^2}\]
```

$$\frac{m\omega^2}{k + i\beta\omega - m\omega^2}$$

Qualitatively, this transfer function is almost the opposite of the previous one. At low velocities/frequencies, it evaluates to 0, implying the car tracks the road’s curve perfectly. At high velocities, it approaches -1, implying the phase of the car’s motion is offset 180° from the motion of the road, but of equal amplitude.

Values of β and Analysis





(the spike gets a lot spikier when beta --> 0)

High values of Beta (the threshold seems to lie around a few thousand) prevent the amplification effect that comes at ~10 m/s. This isn't surprising, as dampers tend to eliminate the power of resonant frequencies.

These results seem to agree with the NDSolveValue results from before, at least in amplitude space. Reading phase offsets from those is possible, but tricky, so this is more convenient.

The time-domain results are useful in to predict specific behavior (for example, maximum displacement) on a specific road, and provide good sanity checking. The frequency domain plots are more useful at predicting specific points, like urging you to avoid driving at exactly 10 m/s.

One possible design intent would be to decrease the maximum amplitude of suspension travel. This would allow lower ground clearance without scraping occasionally. Increasing the value of *k* would accomplish this, since it appears in the denominator of the transfer function for suspension travel, and would thus push that value toward 0.