T1300534-v1 (6/13/13) - Exploration of the best way of applying longitudinal force at the quad test mass using the top mass actuators

To run the calculations in this notebook, it needs to be installed in a calculation directory of a case of the *Mathematica* QuadLite2Lateral model. The DCC version was run as "^/trunk/Common/MathematicaModels/QuadLite2Lateral/mark.barton/20120601TMproductionTM/mark.barton/T1300534-v1 ASUS4XLLateralModelCalcLP.nb" in the SUS SVN.

Setup

Switches to enable loading of previously saved results instead of recalculating from scratch

```
useprecomputed = True; (* set to True to use saved results from precomputed subdirectory *)
If[useprecomputed,
    exceptdamping = False, (* False by default, True to recalculate just damping-dependent
    exceptdamping = True (* DON"T CHANGE *)
];
```

```
loadcasefromuser["ASUS4XLLateralCaseDefn.m"];
```

```
modelcase
```

```
{mark.barton, 20120601TMproductionTM}
```

```
modelcasecomment
```

```
20120601TMproductionTM, equivalent to
  ^trunk/QUAD/Common/MatlabTools/QuadModel_Production/quadopt_fiber.m
  r2731 of 6/1/12
```

Theory

Trying to get an L=longitudinal=x displacement of the optic (x3) by actuating in L at the top mass (x0) causes significant displacement in P=pitch as well due to L/P cross-coupling. This is especially true near the fundamental L mode at 0.434 Hz (see Mode Summary section below). We consider two approaches to mitigating this.

The "minimal" approach is just to add some P actuation at the top mass to take away the P displacement injected by the L-only approach.

The "maximal" approach looks at the exact combination of L and P required to get pure frequency-independent L actuation with no P. It turns out this is simpler and more robust to implement.

We start with the expected displacements L3=x3 and P3=pitch3 at the mass in terms of a matrix of transfer functions and the force/torque at the top mass, FL0 and FP0.

$$\begin{pmatrix} L3 \\ P3 \end{pmatrix} = \begin{pmatrix} L2L & P2L \\ L2P & P2P \end{pmatrix} \begin{pmatrix} FL0 \\ FP0 \end{pmatrix}$$

The minimal approach is to take FL0 as given and solve for FP0 with P3=0. This gives

$$FP0 = -\frac{P2P}{P2L} FL0$$

The maximal approach is to invert the full equation with P3=0 to get the optimum force/torque:

$$\begin{pmatrix} FL0 \\ FP0 \end{pmatrix} = \begin{pmatrix} L2L & P2L \\ L2P & P2P \end{pmatrix}^{-1} \begin{pmatrix} L3 \\ 0 \end{pmatrix}$$

See Conclusions section at end for take-away message.

Mode Summary

```
Join[
    {{"N","f","type"}},
    Table[
        Join[
            {i},
            {Hz2[[-i]]},
            important[e2ni.eigenvectors2[[-i]],0.7]
        ],
        {i,noofLF}
    ]
]//TableForm
```

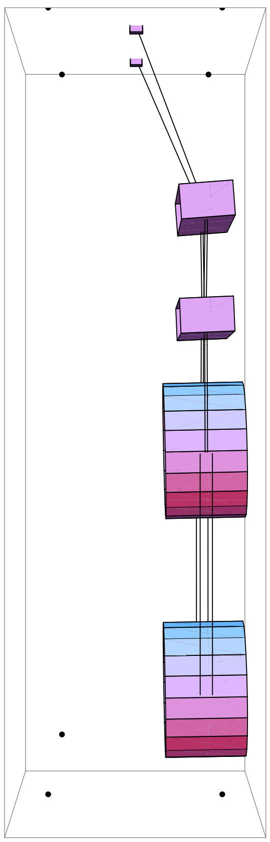
N	f	type		
1	0.434592	pitch3	pitch2	pitch1
2	0.463358	y3	roll3	roll2
3	0.550227	z3	z2	
4	0.563718	pitch3	pitch2	
5	0.600815	yaw3	yaw2	
6	0.875152	roll1	roll3	roll2
7	0.997385	pitch0	pitch1	x2
8	1.04738	y2	y1	y3
9	1.31272	pitch0	pitch1	
10	1.35431	yaw3	yaw1	
11	1.60487	pitch0	pitch1	
12	2.00573	x0	x1	pitch0
13	2.12242	roll1	y0	y1
14	2.21615	z0	z1	
15	2.39316	yaw0	yaw2	
16	2.63331	roll1	roll0	pitch1
17	2.81327	pitch1		
18	3.05151	yaw1	yaw2	
19	3.31788	pitch0	pitch1	roll0
20	3.41638	x1	x0	
21	3.55819	z1	z0	
22	5.07092	roll0	pitch0	
23	9.26975	z2	z3	
24	13.1739	roll2	roll3	

Illustration of cross-coupling at DC for pure L force at top mass

```
unitx0force = {
  pitch0 -> Re[calcTFf[eom2, makefinputvector[x0], makeoutputvector[pitch0], 0]],
  pitch1 -> Re[calcTFf[eom2, makefinputvector[x0], makeoutputvector[pitch1], 0]],
  pitch2 -> Re[calcTFf[eom2, makefinputvector[x0], makeoutputvector[pitch2], 0]],
  pitch3 -> Re[calcTFf[eom2, makefinputvector[x0], makeoutputvector[pitch3], 0]],
  x0 -> Re[calcTFf[eom2, makefinputvector[x0], makeoutputvector[x0], 0]],
  x1 -> Re[calcTFf[eom2, makefinputvector[x0], makeoutputvector[x1], 0]],
  x2 -> Re[calcTFf[eom2, makefinputvector[x0], makeoutputvector[x2], 0]],
  x3 -> Re[calcTFf[eom2, makefinputvector[x0], makeoutputvector[x3], 0]]
 }
```

```
{pitch0 → -0.000151034, pitch1 → -0.0000844887,
 pitch2 → -0.0000422785, pitch3 → -0.0000334987, x0 → 0.000346059,
 x1 → 0.000346446, x2 → 0.000346627, x3 → 0.000347385}
```

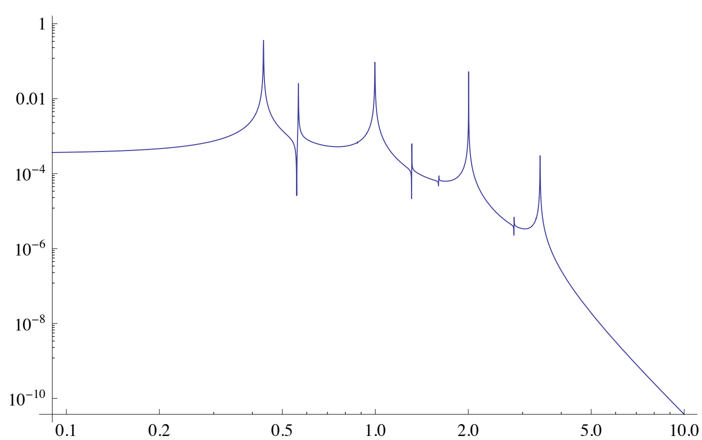
```
eigenplot[allvars /. unitx0force /. x_ :→ 0 /; MemberQ[allvars, x],
 500, {0, -3, 0}, floatmatrix2]
```



Elements of the LP coupling matrix

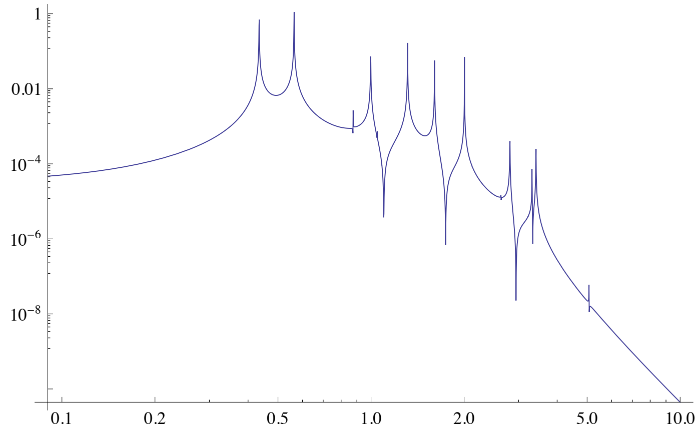
L2L

```
LogLogPlot[
 Abs[calcTFf[eom2, makefinputvector[x0], makeoutputvector[x3], f]], {f, 0.09, 10}]
```



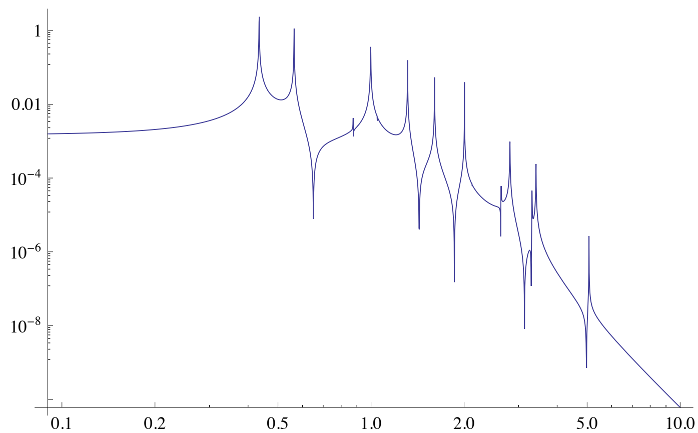
L2P

```
LogLogPlot[Abs[calcTFf[eom2, makefinputvector[x0], makeoutputvector[pitch3], f]],
 {f, 0.09, 10}]
```



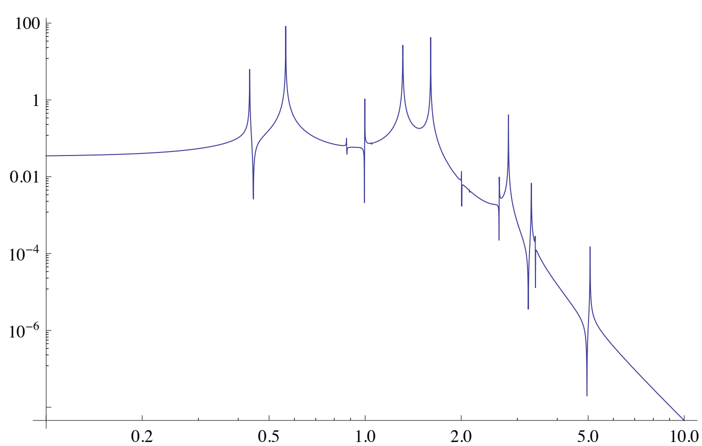
P2L

```
LogLogPlot[Abs[calcTFf[eom2, makefinputvector[pitch0], makeoutputvector[x3], f]],
 {f, 0.09, 10}]
```



P2P

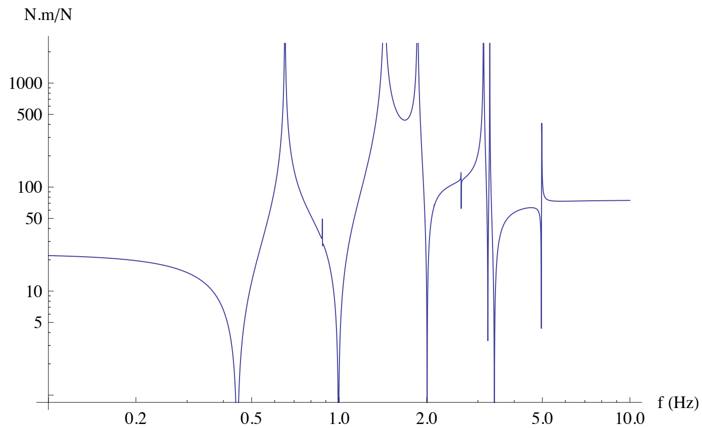
```
LogLogPlot[
 Abs[calcTFf[eom2, makefinputvector[pitch0], makeoutputvector[pitch3], f]],
 {f, 0.1, 10}]
```



Minimal Approach

Optimum P-cancelling filter (run L drive signal through this)

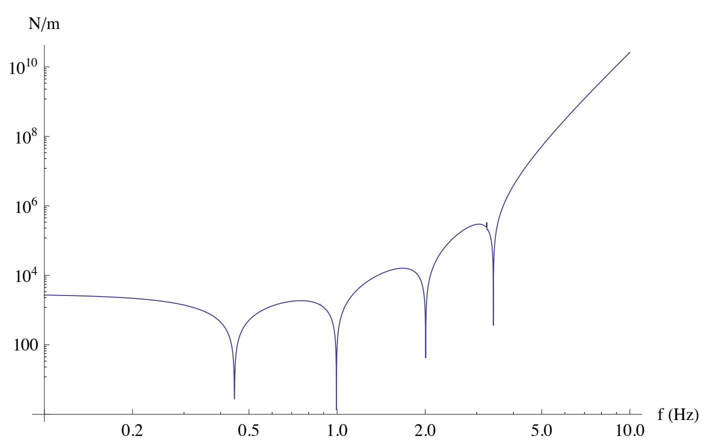
```
LogLogPlot[
 Abs[
  -calcTFf[eom2, makefinputvector[pitch0], makeoutputvector[pitch3], f]
   / calcTFf[eom2, makefinputvector[pitch0], makeoutputvector[x3], f]
 ],
 {f, 0.1, 10}, AxesLabel → {"f (Hz)", "N.m/N"}
]
```



Maximal Approach

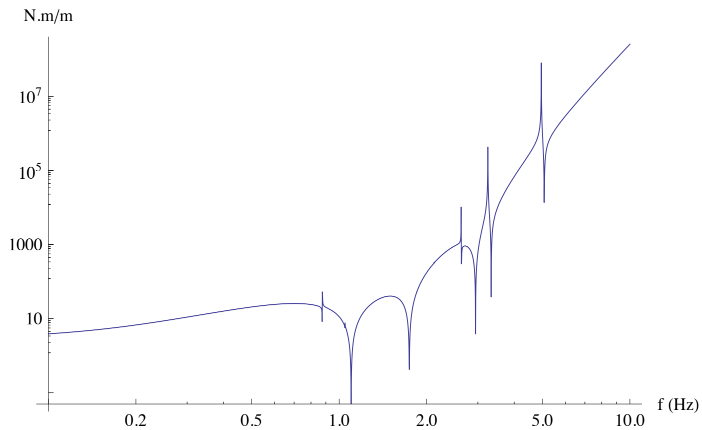
Optimum L filter for flat L output and no P (used in conjunction with P filter below)

```
LogLogPlot[
 Abs[
  (Inverse[
      {{calcTFf[eom2, makefinputvector[x0], makeoutputvector[x3], f],
        calcTFf[eom2, makefinputvector[pitch0], makeoutputvector[x3], f]},
       {calcTFf[eom2, makefinputvector[x0], makeoutputvector[pitch3], f],
        calcTFf[eom2, makefinputvector[pitch0], makeoutputvector[pitch3], f]}
      }
     ].{{1}, {0}}
   )[[1, 1]]
 ],
 {f, 0.1, 10}, AxesLabel → {"f (Hz)", "N/m"}
]
```



Optimum P filter for flat L output and no P (used in conjunction with L filter above)

```
LogLogPlot[
 Abs[
   (Inverse[
       {{calcTFf[eom2, makefinputvector[x0], makeoutputvector[x3], f],
         calcTFf[eom2, makefinputvector[pitch0], makeoutputvector[x3], f]},
        {calcTFf[eom2, makefinputvector[x0], makeoutputvector[pitch3], f],
         calcTFf[eom2, makefinputvector[pitch0], makeoutputvector[pitch3], f]}
       }
      ].{{1}, {0}}
    )[[2, 1]]
 ],
 {f, 0.1, 10}, AxesLabel → {"f  (Hz)", "N.m/m"}
]
```



Conclusions

The minimalist approach gives a messy filter that would be hard to implement and probably wouldn't do a great job. The maximalist filters turn out to be much simpler and could probably be approximated satisfactorily in the all-important low frequency region (below 1 Hz) with just a simple notch in L at 0.434 Hz, and a slight ramp up in P.

Appendix - Filters in table form

L

```
Table[{10^logf, Re[(Inverse[
         {{calcTFf[eom2, makefinputvector[x0], makeoutputvector[x3], 10^logf], calcTF
            eom2, makefinputvector[pitch0], makeoutputvector[x3], 10^logf]},
          {calcTFf[eom2, makefinputvector[x0], makeoutputvector[pitch3], 10^logf],
           calcTFf[eom2, makefinputvector[pitch0],
            makeoutputvector[pitch3], 10^logf]}
         }
        ].{{1}, {0}}
      )[[1, 1]]]
   }
   ,
   {logf, Log[10, 0.1], Log[10, 1], 0.01}
  ] // (TableForm[#, TableHeadings → {None, {"f  (Hz)", "L  (N/m)"}}] &)
```

f (Hz)	L (N/m)
0.1	2708.83
0.102329	2700.35
0.104713	2691.48
0.107152	2682.2
0.109648	2672.49
0.112202	2662.34

0.114815	2651.73
0.11749	2640.63
0.120226	2629.03
0.123027	2616.89
0.125893	2604.21
0.128825	2590.95
0.131826	2577.09
0.134896	2562.6
0.138038	2547.46
0.141254	2531.64
0.144544	2515.1
0.147911	2497.83
0.151356	2479.78
0.154882	2460.93
0.158489	2441.24
0.162181	2420.68
0.165959	2399.21
0.169824	2376.79
0.17378	2353.4
0.177828	2328.97
0.18197	2303.49
0.186209	2276.89
0.190546	2249.15
0.194984	2220.22
0.199526	2190.05
0.204174	2158.59
0.20893	2125.8
0.213796	2091.63
0.218776	2056.03
0.223872	2018.95
0.229087	1980.34
0.234423	1940.14
0.239883	1898.31
0.245471	1854.79
0.251189	1809.53
0.25704	1762.47
0.263027	1713.57
0.269153	1662.77
0.275423	1610.01
0.281838	1555.25
0.288403	1498.44
0.295121	1439.54
0.301995	1378.49
0.30903	1315.25
0.316228	1249.79
0.323594	1182.07
0.331131	1112.06
0.338844	1039.73
0.346737	965.071
0.354813	888.067
0.363078	808.714
0.371535	727.018
0.380189	642.996
0.389045	556.677
0.398107	468.102
0.40738	377.328
0.416869	284.429
0.42658	189.495
0.436516	92.6401
0.446684	-6.00123
0.457088	-106.269

0.467735	-207.97
0.47863	-310.894
0.489779	-414.784
0.501187	-519.352
0.512861	-624.268
0.524807	-729.161
0.537032	-833.61
0.549541	-937.144
0.562341	-1039.24
0.57544	-1139.3
0.588844	-1236.68
0.60256	-1330.66
0.616595	-1420.45
0.630957	-1505.15
0.645654	-1583.83
0.660693	-1655.43
0.676083	-1718.8
0.691831	-1772.7
0.707946	-1815.79
0.724436	-1846.61
0.74131	-1863.61
0.758578	-1865.09
0.776247	-1849.27
0.794328	-1814.25
0.812831	-1758.
0.831764	-1678.41
0.851138	-1573.31
0.870964	-1441.73
0.891251	-1275.47
0.912011	-1079.37
0.933254	-847.615
0.954993	-577.844
0.977237	-267.548
1.	85.7546

P

```
Table[{10^logf, Re[(Inverse[
        {{calcTFf[eom2, makefinputvector[x0], makeoutputvector[x3], 10^logf], calcTF
           eom2, makefinputvector[pitch0], makeoutputvector[x3], 10^logf]},
         {calcTFf[eom2, makefinputvector[x0], makeoutputvector[pitch3], 10^logf],
          calcTFf[eom2, makefinputvector[pitch0],
           makeoutputvector[pitch3], 10^logf]}
        }
       ].{{1}, {0}}
     )[[2, 1]]]
  }
  ,
  {logf, Log[10, 0.01], Log[10, 1], 0.1}
 ] // (TableForm[#, TableHeadings → {None, {"f (Hz)", "P (N.m/m)"}}] &)
```

f (Hz)	P (N.m/m)
0.01	2.90111
0.0125893	2.90706
0.0158489	2.91648
0.0199526	2.93142
0.0251189	2.95507
0.0316228	2.99253
0.0398107	3.05182
0.0501187	3.14561
0.0630957	3.29379
0.0794328	3.52747
0.1	3.89492
0.125893	4.46996
0.158489	5.3631
0.199526	6.7332
0.251189	8.79231
0.316228	11.7803
0.398107	15.8509
0.501187	20.7386
0.630957	24.9556
0.794328	24.2528
1.	11.2874