LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY

LIGO Laboratory / LIGO Scientific Collaboration

LIGO- T0900627-v2 *ADVANCED LIGO* 1 March 2010

Quad Suspension Cable Modeling in Mathematica™

Mark Barton

Distribution of this document:
DCC

This is an internal working note
of the LIGO Laboratory.

California Institute of Technology
LIGO Project – MS 18-34
1200 E. California Blvd.
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project – NW22-295
185 Albany St
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
P.O. Box 1970
Mail Stop S9-02
Richland WA 99352
Phone 509-372-8106
Fax 509-372-8137

LIGO Livingston Observatory
P.O. Box 940
Livingston, LA 70754
Phone 225-686-3100
Fax 225-686-7189

http://www.ligo.caltech.edu/

Table of Contents

1 Introduction

1.1 Purpose and Scope

The reaction chain of the quad will have many electrical cables to the OSEMs and ESD drive. The cables to the ESD drive are a particular concern because they bridge the final pendulum stage and so could defeat the isolation at a particularly crucial point. This document outlines models of the quad suspension with cables that were created to check whether any of the candidate cable types identified by Rich Abbott were too stiff to be acceptable. Previous drafts of this document exonerated all but the thickest cable type (Cooner Wire CW2040-2650F with Jacket). This draft exonerates all of the candidates.

1.2 References

T020205-02 Suspension models in Mathematica™

T080206-00 SURF 2008 Lab Notebook – J. Chen

T080341-00 Understanding Cabling Noise in LIGO, J. Chen

T080205-00 SURF 2008 Lab Notebook, J.E. Freed-Brown

T0900083-v1 Modeling Cable Noise in LIGO, J.E. Freed-Brown

T0900060-v1 OMC Suspension Final Design Document

T0900311-x0 Suspension Cable Modeling in Mathematica™ and Application to the OMC SUS

1.3 Version history

11/11/09: Pre-rev-v1 draft.

11/12/09: Pre-rev-v1 draft 2. Added case names.

12/14/09: Pre-rev-v1 draft3. Corrected year (was 2009 all along). Applied DCC number (T0900627). Added data for case `20090819TMcables8lossystiff` and revised conclusion to exonerate all cables. Added Appendix with parameters. Released as -v1.

3/1/10: -v2 with extra stiff cables at upper levels as proxy for multiple cables.

2 Quad Suspension Overview

The quad model used as the basis for this study is case 20090819TM of the “quad xtra-lite lateral” model. See T020205-02 for the Mathematica™ toolkit used to create this model. This is based on 20080326TM (Norna's conceptual design from T060283-02) with improved damping parameters from Alastair Heptonstall, larger d2 per M080134-00, and the latest monolithic parameters. This was chosen partly because it was better debugged than other cases and partly because the monolithic final stage would be a more sensitive test for coupling from the cables.

3 Modeling of Cables

See the forthcoming writeup T0900311-x0, for a description of how the modeling toolkit was extended to include cables and how numerical parameters were determined for samples of the cable for the OMC. To recap, the key things that need to be specified apart from the mass and MOI of cable segments are elasticities `{kx1,ky1,kz1,kyaw1,kpitch1,kroll1}` defined in a local coordinate system in which +x is along the tangent to the direction of the cable. The x component sets the longitudinal stiffness:

$$\texttt{kx} = \frac{(n+1)EA}{l}$$

where l is the length, n is the number of segments (so that $n+1$ is the number of springs) and EA is the effective product of Young's modulus and cross-sectional area:

$$EA = k_{long} L$$

where k_{long} is the longitudinal stiffness of a sample of length L.

Similarly, the roll component sets the torsional stiffness:

$$\texttt{kroll} = \frac{(n+1)GJ}{l}$$

where GJ is the effective product of shear modulus and torsional stiffness geometric factor:

$$GJ = k_{tors} L$$

where k_{tors} is the torsional stiffness of a sample of length L.

The yaw and pitch components set the bending stiffnesses:

$$\texttt{kyaw} = \frac{nEI}{l}$$ (and similarly for `kpitch`)

where EI is the effective Young's modulus*moment of area product. However from cantilever beam theory, this is not linear in anything interesting. Rather, the transverse stiffness at the end of a cantilevered sample of cable goes as the inverse cube of the length, so that

$$EI = \frac{k_{trans} L^3}{3}$$

where k_{trans} is the transverse stiffness at the tip. The non-linearity also explains the factor of n (rather than $n+1$) above: because the net compliance is set by lever arm effects in the cantilever, one gets a better approximation to the bulk properties by focusing on the lengths of the n segments rather than the compliances of the $n+1$ springs.

The y and z components do not correspond to any bulk property of the cable but merely keep the cable elements lined up nose-to-toe. In conjunction with the MOI of the cable segments they create high frequency eigenmodes, and so they should be chosen to place these comfortably above other eigenmodes of interest, but not so high as to bring on numerical stability issues. (This is especially important if the results are ever to be exported to Simulink and used for numerical simulation, because high eigenfrequencies will require small step-sizes and long runtimes to simulate.) Ideally they would be replaced by geometric constraints, but this currently thought to be infeasible due to

the complexity of the geometry (the constraint equations for a pair of segments at the end of the chain would involve the coordinates of all segments back to the beginning).

4 Cable parameters

Rich Abbott measured the transverse stiffnesses for samples of length 50.8 mm of different cables that were electrically suitable. See Table 1. Also included in the table are the parameters of the sample of OMC SUS cable measured by SURF student Chihyu "Jimmy" Chen, which had 18 conductors in a braided copper sleeve (T080341-00).

Table 1 - Cable Stiffnesses

COONER WIRE	OD	REF. LENGTH	WEIGHT	DEFLECTION	SPRING CONSTANT
PART #	in	m	grams	mm	N/m
CW2040-2650F with Jacket	**0.082**	**0.0508**	**0.912**	**0.298**	**30.027**
CW2040-2650F without Jacket		**0.0508**	**0.912**	**2.183**	**4.095**
CW2040-3050F with Jacket	**0.058**	**0.0508**	**0.228**	**0.695**	**3.217**
		0.0508	**0.456**	**1.588**	**2.815**
CW2040-3050F without Jacket		**0.0508**	**0.228**	**3.671**	**0.609**
CW2040-3250F with Jacket	**0.05**	**0.0508**	**0.228**	**0.992**	**2.252**
		0.0508	**0.456**	**2.084**	**2.145**
		0.0508	**0.684**	**3.572**	**1.877**
CW2040-3250F without Jacket		**0.0508**	**0.228**	**5.358**	**0.417**
		0.0508	**0.456**	**11.708**	**0.382**
CW2040-3650F with Jacket	**0.039**	**0.0508**	**0.228**	**2.778**	**0.804**
CW2040-3650F without Jacket		**0.0508**	**0.06849**	**9.128**	**0.074**
		0.0508	**0.13698**	**off the scale**	
OMC Cable 18 conductors + sleeve		0.05	1.6		**4.900**

The loss factor reported by Jimmy Chen was 0.001. This was measured in a torsion pendulum configuration, and probably includes significant dissipation dilution due to the low-loss elasticity from the winding-up of the internal conductors (which raises the mass against gravity). It is unrealistically low for cantilever geometry but no better value is available. Thus values of 0.01 and 0.1 have been used as plausible bounding values.

5 System modeled

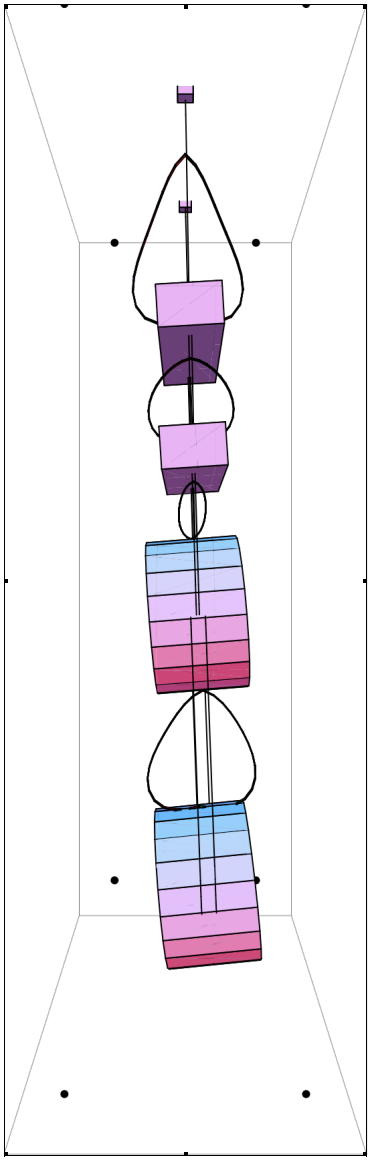
Because the OMC cable was stiffer and heavier than all but one wire type, and because the parameters were thoroughly debugged, it was used for initial simulations. (See Sections 6.1 and 6.2. Later the stiffest case was modeled separately - Section 6.3.) Two lengths of cable were added bridging each adjacent pair of levels in the suspension, as illustrated in Figure 1.

Ideally there would have been at 5 cables at the lowest level, being the number of ESD quadrants plus a ground, and considerably more at higher levels, reflecting the various OSEM cables.

Unfortunately, with 4 levels, 2 cables per level, and 15 segments per cable, the run time was becoming prohibitive (greater than 12 hours), so this is probably the best that can be done with available computing resources (and even with considerably more). Having 2 cables per level actually speeds things up because symmetry can be assumed, which reduces the number of variables that have to be optimized over when finding the equilibrium position. And having at least some cables at each level gives at least some prospect of detecting misbehaviour in which vibration is transmitted entirely through the cables.

From the top down the lengths of the segments were 50, 30, 15, and 40 cm, which were chosen by trial and error to drape nicely.

Figure 1 - Fundamental pendulum mode of quad pendulum with cables

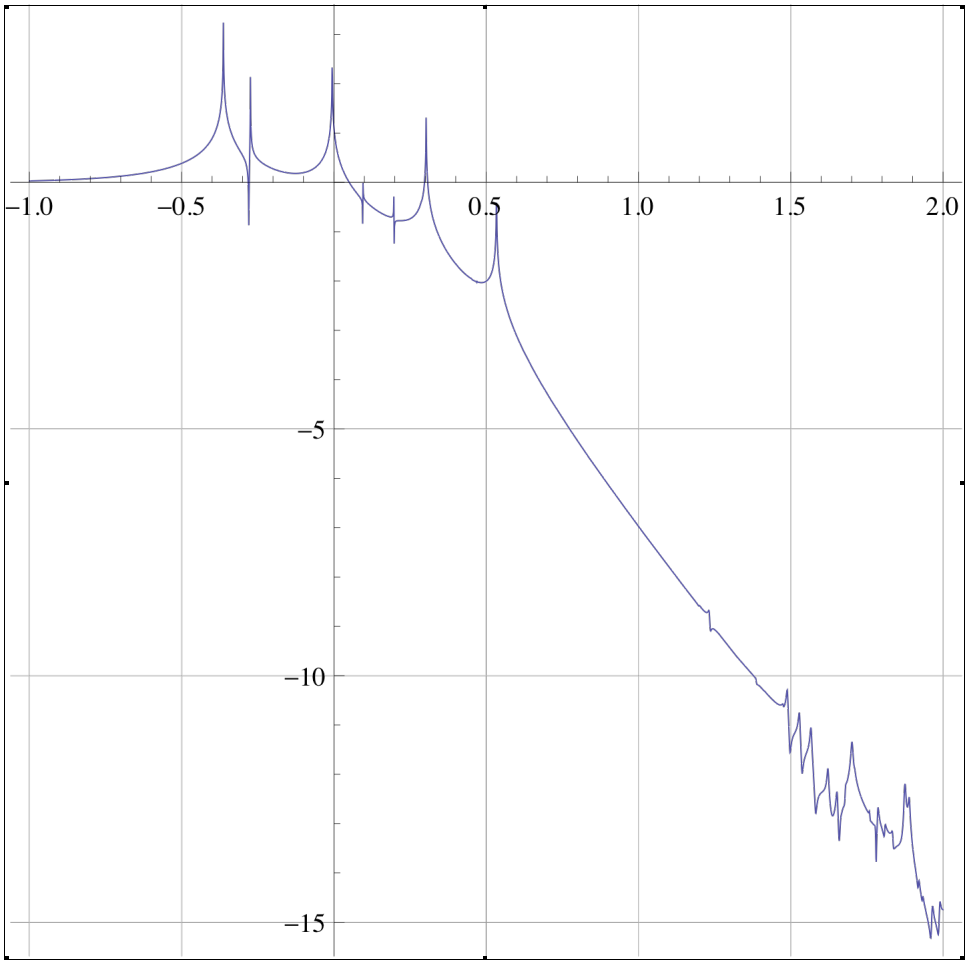


6 Results

6.1 Low-loss system

The transfer function for the low-loss (case `20090819TMcables8`; phi=0.01) system is shown in Figure 2. The main concern that might be anticipated with such a system is that high-Q peaks from the cable could compromise the isolation at higher frequencies. Sure enough, there is some structure above 30 Hz, but not enough to pose a problem. The overall trend of f^{-8} fall-off is uninterrupted.

Figure 2 - Transfer function from x of structure to x of optic for low-loss cables (phi=0.01)

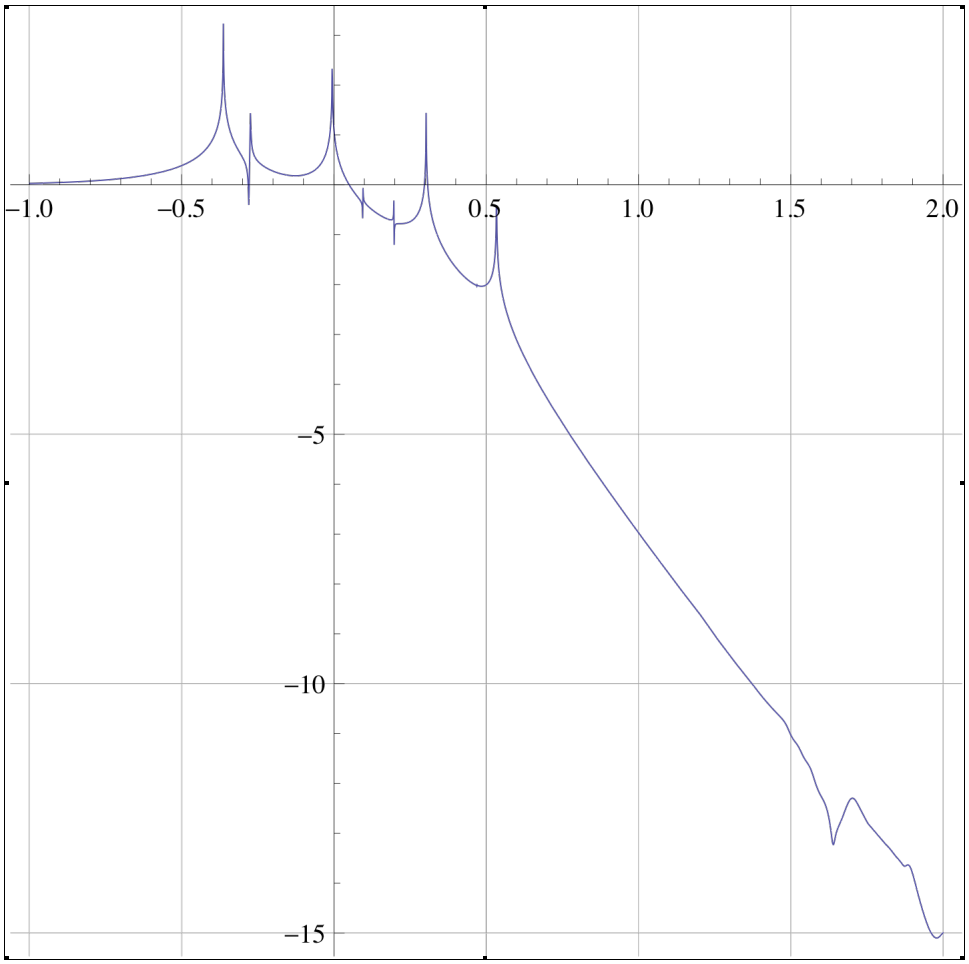


Vertical axis: log10(x-x transfer function); horizontal axis: log10(f in Hz)

6.2 High-loss system

The transfer function for the high-loss system (case `20090819TMcables8lossy`; phi=0.1) is shown in Figure 3. The problem that might be anticipated with high loss is for the f^{-8} fall-off expected for a 4-stage pendulum to be reduced to a lesser power of f, i.e., a slower fall-off. However there is even less structure than for the low loss case and the desired fall-off continues all the way to 100 Hz.

Figure 3 - Transfer function from x of structure to x of optic for high-loss cables (phi=0.1)

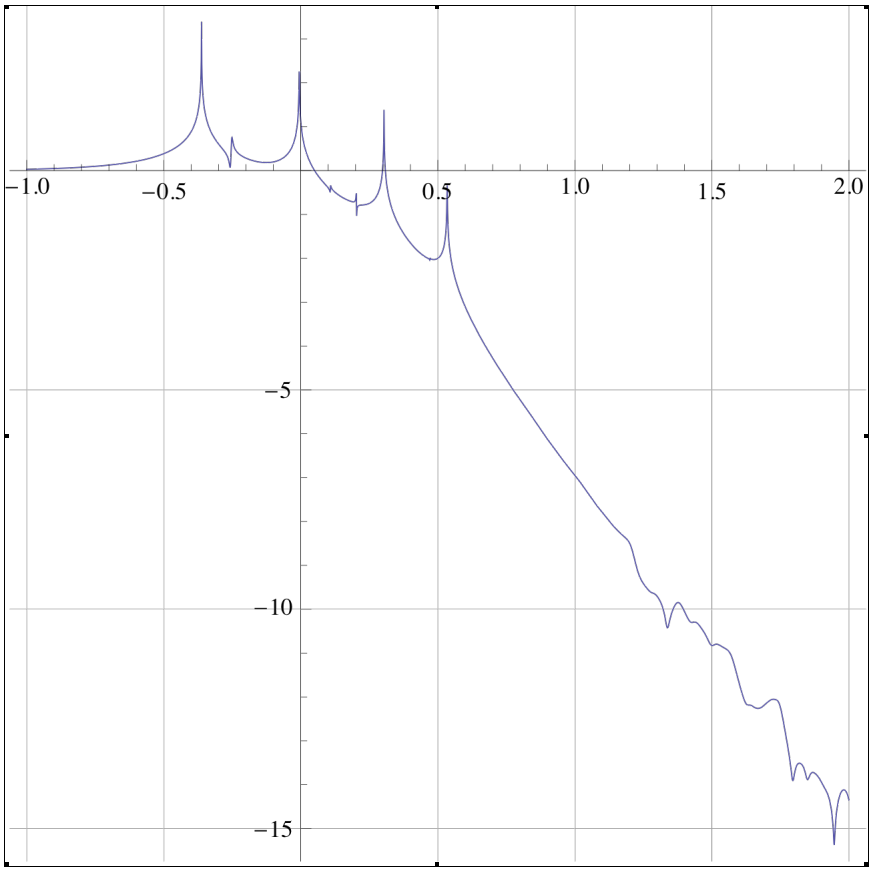


Vertical axis: log10(x-x transfer function); horizontal axis: log10(f in Hz)

6.3 Extra stiff case

The stiffest candidate was modeled as case 20090819TMcables8lossystiff, assuming a damping factor of 0.1. There was considerably more structure above 30 Hz than the other cases (see Figure 1), but still not enough to present a problem.

Figure 4 - Transfer function from x of structure to x of optic for stiffest candidate cable.

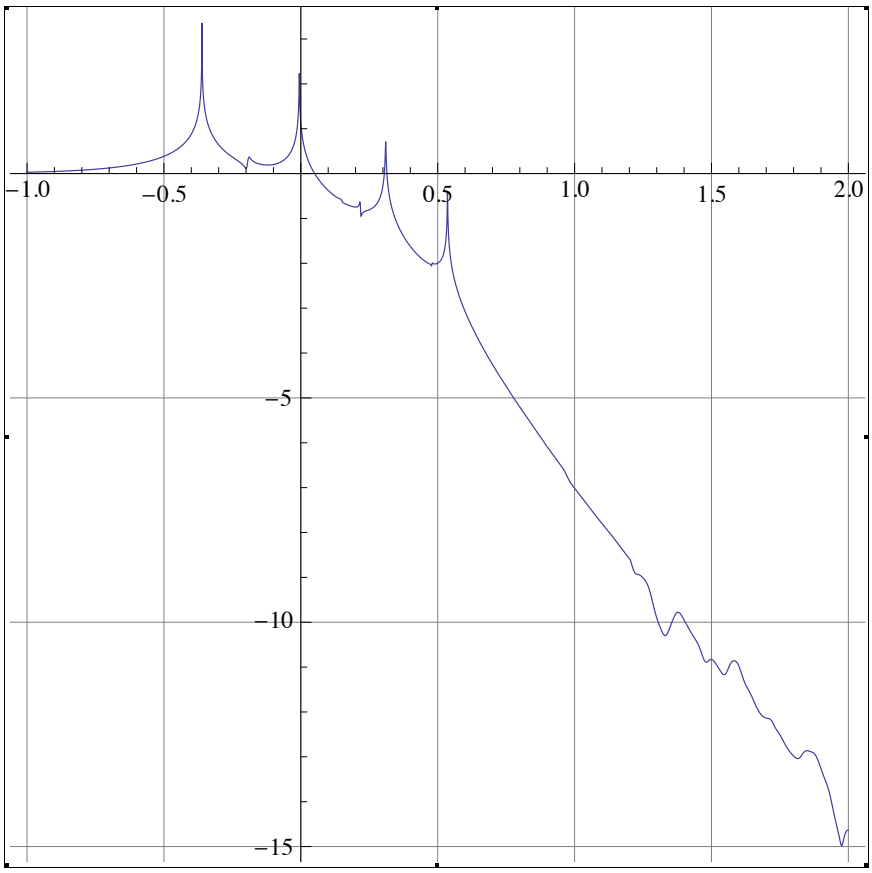


Vertical axis: log10(x-x transfer function); horizontal axis: log10(f in Hz)

6.4 Stiff cables at higher levels

Because the net degradation at the test mass is presumably something like the product of the degradation at each step in the chain, and because there are many additional non-ESD cables at higher levels, there could conceivably be an issue where ESD cables that would be fine in isolation are the straw that breaks the camel's back. Unfortunately, as noted above, eight cables is pushing the limits of the toolkit. However to get some idea of how serious a problem this might be, a case was prepared with the stiffness of each of the six upper cables increased by a factor of 10 (`20090819TMcables8lossystiff2`). The transfer function is shown in Figure 5. The structure in the curve is even less peaky than previously but still does not deviate significantly from the desired f^{-8} trend.

Figure 5: Transfer function from x of structure to x of optic for stiffest candidate cable at lowest level and 10 times that stiffness at the upper three levels as a proxy for multiple cables.



Vertical axis: log10(x-x transfer function); horizontal axis: log10(f in Hz)

7 Conclusion

All of Rich Abbot's candidate cables should be fine considered as ESD cabling in isolation. The transfer functions are sufficiently innocuous that even increasing the number of cables from 2 to 5 should be OK. The possibility of a problem from non-ESD cabling at higher levels in conjunction with the ESD cable at the lowest level was examined by using extra-stiff cable at the upper levels as a proxy for multiple cables and no problem was found.

8 Appendix

The "overrides list" defining one of the cases (`20090819TMcables8`) relative to the default values for the model is given for reference in Section 8.1. (In fact, nearly every definition is overridden.) Case `20090819TMcables8lossy` is identical except that all the cable damping functions,

```
damping[imag, cabletype0] -> (.01 &)
```

and the like, have 0.1 instead of 0.01.

The definition of the cables is given in Section 8.2.

8.1 Overrides for case `20090819TMcables8`

```
overrides0 = {
 (* AFCP = as for controls prototype *)
 ribbons -> False,
 nx -> 0.1300, (* AFCP, T040214-01 (display only) *)
 ny -> 0.5000, (* AFCP, T040214-01 (display only) *)
 nz -> 0.0840, (* AFCP, 040214-01 (display only) *)
 denn -> 4000, (* AFCP, T040214-01 (unused) *)
 mn -> 22.11, (* T010103-05 *)
 Inx -> 0.4558, (* T010103-05 *)
 Iny -> 0.0712, (* T010103-05 *)
 Inz -> 0.4547, (* T010103-05 *)
 Inxy -> 0, (* try to design symmetrically *)
 Inyz -> 0, (* try to design symmetrically *)
 Inzx -> 0, (* try to design symmetrically *)
 Inxz -> Inzx,
 Inzy -> Inyz,
 Inyx -> Inxy,
 ux -> 0.1300, (* AFCP, T040214-01 (display only) *)
 uy -> 0.5000, (* AFCP, T040214-01 (display only) *)
 uz -> 0.0840, (* AFCP, T040214-01 (display only) *)
 den1 -> 4000, (* AFCP, T040214-01 (unused) *)
 m1 -> 21.011, (* T010103-05 *)
 I1x -> 0.5174, (* T010103-05 *)
 I1y -> 0.0598, (* T010103-05 *)
 I1z -> 0.5205, (* T010103-05 *)
 ix -> 0.20, (* T060283 *)
 ir -> 0.17, (* T060283 *)
 if -> 0.095, (* M050397-02/T010103-05 *)
 den2 -> 2200, (* T060283 *)
 m2 -> den2 int[1,ir,ix,if],
 I2x -> den2 int[MOIintegrands[[1,1]],ir,ix,if],
 I2y -> den2 int[MOIintegrands[[2,2]],ir,ix,if],
 I2z -> den2 int[MOIintegrands[[3,3]],ir,ix,if],
 tx -> 0.20, (* T060283 *)
 tr -> 0.17, (* T060283 *)
 tf -> 0.095, (* M050397-02/T010103-05 *)
 den3 -> 2200, (* T060283 *)
 m3 -> den3 int[1,tr,tx,tf],
 I3x -> den3 int[MOIintegrands[[1,1]],tr,tx,tf],
 I3y -> den3 int[MOIintegrands[[2,2]],tr,tx,tf],
 I3z -> den3 int[MOIintegrands[[3,3]],tr,tx,tf],
 tlnspec -> 0.416, (* NR 4/3/06, T010103-05 *)
 tl1spec -> 0.277, (* NR 4/3/06, T010103-05 *)
 tl2spec -> 0.341, (* NR 4/3/06, T010103-05 *)
 tl3spec -> 0.602, (* NR 4/3/06, T010103-05 *)
 ln -> drop[], (* to be derived *)
 l1 -> drop[], (* to be derived *)
 l2 -> drop[], (* to be derived *)
 l3 -> drop[], (* to be derived *)
 rn -> 5.200 10^-04, (* AFCP, available size close to T010103-05 *)
 r1 -> 3.5000 10^-04, (* T010103-05 *)
 r2 -> 3.1000 10^-04, (* T010103-05 *)
 r3 -> If[ribbons, Indeterminate, 0.0004/2], (* 400 μm diameter middle
section *)
```

```
 t3 -> If[ribbons, 0.000115, Indeterminate], (* T060283 *)
 W3 -> If[ribbons, 0.00115, Indeterminate], (* T060283 *)
 optstress -> alphasilica*Ysilica/betasilica,
 t3m -> t3,
 W3m -> W3,
 t3n -> Sqrt[m3*g/optstress/10/4],
 W3n -> Sqrt[10*m3*g/optstress/4],
 r3m -> r3,
 r3n -> 0.0008/2, (* 800 μm diameter end sections *)
 M31 -> If[ribbons, W3n t3n^3/12, Pi*r3n^4/4],
 M32 -> If[ribbons, t3n W3n^3/12, Pi*r3n^4/4],
 A3n -> If[ribbons, W3n t3n, Pi r3n^2],
 A3m -> If[ribbons, W3m t3m, Pi r3m^2],
 nf -> 1/40.54, (* neck fraction relative to total fibre to give 1.5 cm
or 5 transverse flexure lengths *)
  kw3 -> 1/(1/kw3m+2/kw3n), (* net longitudinal elasticity of fibres *)
  kw3n -> Y3*A3n/(nf*l3), (* net longitudinal elasticity of one fibre
neck *)
  kw3m -> Y3*A3m/((1-2*nf)*l3), (* net longitudinal elasticity of fibre
midsection *)
 Yn -> 2.12 10^11, (* measured, MB, 11/18/05, via IFOModel v4.1; cf. 2.2
in T010103-05 *)
 Y1 -> 2.12 10^11, (* measured, MB, 11/18/05, via IFOModel v4.1; cf. 2.2
in T010103-05 *)
 Y2 -> 2.12 10^11, (* measured, MB, 11/18/05, via IFOModel v4.1; cf. 2.2
in T010103-05 *)
 Y3 -> Ysilica, (* use same value as damping model (was independent in
20061213TM) *)
 ffn -> 0.807, (* from Ian's data, 11/30/05, linear fit version *)
 ff1 -> 0.641, (* from Ian's data, 11/30/05, linear fit version *)
 ff2 -> 0.608, (* from Ian's data, 11/30/05, linear fit version *)
 kffn -> 1 + ffn*Tan[Pi*thetan/180],
 kff1 -> 1 + ff1*Tan[Pi*theta1/180],
 kff2 -> 1 + ff2*Tan[Pi*theta2/180],
 ufcVn -> 2.3700, (* AFCP, measured, CT et al., 11/06 *)
 ufcV1 -> 2.5800, (* AFCP, measured, CT et al., 11/06 *)
 ufcV2 -> 1.7900, (* AFCP, measured, CT et al., 11/06 *)
 ufcn -> ufcVn*Sqrt[kffn],
 ufc1 -> ufcV1*Sqrt[kff1],
 ufc2 -> ufcV2*Sqrt[kff2],
 kbuz -> 4*Pi^2*(59/60)^2*61*kffn, (* AFCP, measured, T040229-12 *)
 kbiz -> 4*Pi^2*(70/60)^2*50*kff1, (* AFCP, measured, T040229-12 *)
 kblz -> 4*Pi^2*(76/60)^2*39*kff2, (* AFCP, measured, T040229-12 *)
 dm -> 0.001-flexn,
 dn -> 0.001-flex1+(g*m13)/(2*kbix),
 d0 -> 0.001-flex1,
 d1 -> 0.001-flex2+(g*m23)/(2*kblx),
 d2 -> 0.010-flex2, (* larger d2 per M080134-00 *)
 d3 -> 0.001-flex3,
 d4 -> 0.001-flex3,
 twistlength -> 0,  (* T010103-05 (unused) *)
 d3tr -> 1.0000 10^-03, (* T010103-05 (unused) *)
 d4tr -> 1.0000 10^-03, (* T010103-05 (unused) *)
 sn -> 0,  (* T010103-05 (unused) *)
 su -> 0.003, (* T010103-05 (AFCP) *)
 si -> 0.003, (* T010103-05 (AFCP) *)
 sl -> 0.015, (* T060283 *)
 nn0 -> 0.250, (* T010103-05 (AFCP) *)
```

```
 nn1 -> 0.090, (* T010103-05 (AFCP) *)
 n0 -> 0.200, (* T010103-05 (AFCP) *)
 n1 -> 0.060, (* T010103-05 (AFCP) *)
 n2 -> 0.140, (* T010103-05 (AFCP) *)
 n3 -> 0.1762, (* T010103-05 *)
 n4 -> 0.1712, (* T010103-05 *)
 n5 -> 0.1712, (* T010103-05 *)
 nwn -> 2,
 nw1 -> 4,
 nw2 -> 4,
 nw3 -> 4,
 mn3 -> mn+m13,
 m13 -> m1+m23,
 m23 -> m2+m3,
 kbux -> 1.0 10^5, (* as for middle *)
 kbix -> 1.0 10^5, (* Justin 11/29/05 *)
 kblx -> 0.8 10^5, (* Ian 12/09/05 *)
 flexn -> Sqrt[nwn Mn1 Yn/(mn+m1+m2+m3)/g]*cn^(3/2),
 flex1 -> Sqrt[nw1 M11 Y1/(m1+m2+m3)/g]*c1^(3/2),
 flex2 -> Sqrt[nw2 M21 Y2/(m2+m3)/g]*c2^(3/2),
 flex3 -> Sqrt[nw3 M31 Y3/m3/g]*c3^(3/2),
 flex3tr -> Sqrt[nw3 M32 Y3/m3/g]*c3^(3/2),
 thetan -> 180 ArcSin[sin]/Pi,
 theta1 -> 180 ArcSin[si1]/Pi,
 theta2 -> 180 ArcSin[si2]/Pi,
 theta3 -> 180 ArcSin[si3]/Pi,
 DD -> Sqrt[M31 Y3/4/4/m3/g/sl3^2],

  rhosilica -> 2.2 10^3, (* IFOModel v4.1 *)
 Csilica -> 770., (* AH silica spec sheet, summary email by MB, 3/13/08
*)
 Ksilica -> 1.38, (* IFOModel v4.1 *)
  sigmasilica -> 0.17,
  Gsilica -> Ysilica/2/(1+sigmasilica), (* shear modulus *)
 alphasilica -> 3.9 10^-7, (* AH measurement, summary email by MB,
3/13/08 *)
 betasilica -> 1.52 10^-4, (* IFOModel v4.1 *)
 phisilica -> 4.1 10^-10, (* IFOModel v4.1 *)
 phissilica -> 3. 10^-11, (* surface *)
 Ysilica -> 7.2 10^10, (* AH spec sheet, summary email by MB, 3/13/08 *)
 dssilica -> 1.5 10^-2, (* IFOModel v4.1 *)
  sigmasilica -> 0.17,
  Gsilica -> Ysilica/2/(1+sigmasilica), (* shear modulus *)

 rhosteel-> 7800., (* IFOModel v4.1 *)
 Csteel-> 460., (* IFOModel v4.1 *)
 Ksteel-> 49., (* IFOModel v4.1 *)
 alphasteel-> 12. 10^-6, (* IFOModel v4.1 *)
 betasteel-> -2.5 10^-4, (* IFOModel v4.1 *)
 phisteel-> 1. 10^-4, (* IFOModel v4.1 *)
 Ysteel-> 2.12 10^11, (* measured, MB, 11/18/05, via IFOModel v4.1, oops
should be 2.119 *)

 rhomarag-> 7800., (* IFOModel v4.1 *)
 Cmarag-> 460., (* IFOModel v4.1 *)
 Kmarag-> 20., (* IFOModel v4.1 *)
 alphamarag-> 11. 10^-6, (* IFOModel v4.1 *)
 betamarag-> -2.5 10^-4, (* Geppo's value - Bench v4.1 is wrong *)
```

```
phimarag-> 1. 10^-4, (* IFOModel v4.1 *)
Ymarag-> 1.87 10^11, (* IFOModel v4.1 *)

(* Zener, 1938, Phys. Rev. 53:90-99 *)
magicnumber->1/4/FindRoot[0==D[BesselJ[1,x],x],{x,1.8}][[1,2]]^2,

tmU -> 0.0043, (* IFOModel v4.1 *)
deltabladeU -> Ymarag*alphamarag^2*temperature/(rhomarag*Cmarag),
taubladeU -> rhomarag*Cmarag*tmU^2/(Kmarag*N[Pi]^2),
damping[imag,bladeUtype] -> ((phimarag+
deltabladeU*(2*N[Pi]*#1*taubladeU)/(1+(2*N[Pi]*#1*taubladeU)^2))&),

tmI -> 0.0046, (* IFOModel v4.1 *)
deltabladeI -> Ymarag*alphamarag^2*temperature/(rhomarag*Cmarag),
taubladeI -> rhomarag*Cmarag*tmI^2/(Kmarag*N[Pi]^2),
damping[imag,bladeItype] -> ((phimarag+deltabladeI*(2*N[Pi]*#1*taubladeI)
/(1+(2*N[Pi]*#1*taubladeI)^2))&),

tmL -> 0.0042, (* IFOModel v4.1 *)
deltabladeL -> Ymarag*alphamarag^2*temperature/(rhomarag*Cmarag),
taubladeL -> rhomarag*Cmarag*tmL^2/(Kmarag*N[Pi]^2),
damping[imag,bladeLtype] -> ((phimarag+deltabladeL*(2*N[Pi]*#1*taubladeL)
/(1+(2*N[Pi]*#1*taubladeL)^2))&),

deltawireU -> Ysteel*temperature
*(alphasteel-betasteel*g*(mn+m1+m2+m3)/(nwn*N[Pi]*rn^2*Ysteel))^2
/(rhosteel*Csteel),
tauwireU -> magicnumber*rhosteel*Csteel*(2*rn)^2/Ksteel,
damping[imag,wireUtype] -> (phisteel&),
damping[imag,wireUatype] -> ((phisteel+deltawireU*(2*N[Pi]*#1*tauwireU)
/(1+(2*N[Pi]*#1*tauwireU)^2))&),

deltawireI -> Ysteel*temperature
*(alphasteel-betasteel*g*(m1+m2+m3)/(nw1*N[Pi]*r1^2*Ysteel))^2
/(rhosteel*Csteel),
tauwireI -> magicnumber*rhosteel*Csteel*(2*r1)^2/Ksteel,
damping[imag,wireItype] -> (phisteel&),
damping[imag,wireIatype] -> ((phisteel+deltawireI*(2*N[Pi]*#1*tauwireI)
/(1+(2*N[Pi]*#1*tauwireI)^2))&),

deltawireL -> Ysteel*temperature
*(alphasteel-betasteel*g*(m2+m3)/(nw2*N[Pi]*r2^2*Ysteel))^2
/(rhosteel*Csteel),
tauwireL -> magicnumber*rhosteel*Csteel*(2*r2)^2/Ksteel,
damping[imag,wireLtype] -> (phisteel&),
damping[imag,wireLatype] -> ((phisteel+deltawireL*(2*N[Pi]*#1*tauwireL)
/(1+(2*N[Pi]*#1*tauwireL)^2))&),

deltafibre -> Ysilica*temperature
*(alphasilica-betasilica*g*(m3)/(nw3*A3n*Ysilica))^2
/(rhosilica*Csilica),
taufibre -> If[
ribbons,
```

```
 rhosilica*Csilica*t3n^2/Ksilica/N[Pi]^2,
 magicnumber*rhosilica*Csilica*4*A3n/N[Pi]/Ksilica
 ],
 damping[imag,fibretype] -> If[
 ribbons,
 ((phisilica*(1+2*dssilica*(W3n+t3n)/(W3n*t3n)))&),
 ((phisilica*(1+2*dssilica/r3n))&)
 ],
 damping[imag,fibreatype] -> If[
 ribbons,

((phisilica*(1+2*dssilica*(3*nw3*W3n+t3n)*(W3n+t3n)/(nw3*W3n+t3n)/(W3n*t3
n))
 +deltafibre*(2*N[Pi]*#1*taufibre)/(1+(2*N[Pi]*#1*taufibre)^2))&),
 ((phisilica*(1+4*dssilica/r3n)
 +deltafibre*(2*N[Pi]*#1*taufibre)/(1+(2*N[Pi]*#1*taufibre)^2))&)
 ],

cablemaxangle -> p/3 ,(*The largest angle allowed by constraints*)
useNMinimize -> False,
usehardconstraints -> True,
iterations -> 10000,
cablegrav -> True,

kx5cm -> 500, (*The spring constants of 5cm of cable, for callibration*)
ky5cm -> 500,
kz5cm -> 500,
kyaw5cm -> 4.9,
kpitch5cm -> 4.9,
kroll5cm -> .003,
refleng -> 0.05, (* length of reference piece of cable, 5 cm *)

(*CABLE 3 - PM to optic*)

nc3 -> 15,
mpl3 -> .032, (*Mass per length of the cable*)
rog3 -> rcable3/Sqrt[2],
lcable3 -> 0.40, (*cable length, meters, guesstimate *)
rcable3 -> .0005, (*cable radius*)
mcable3 -> lcable3*mpl3, (*the total mass of the cable*)
Ixcable3 -> (1/2) (mcable3/nc3) rcable3^2,(*moment of inertia of a cable
element about x*)
Iycable3 -> (1/12) (mcable3/nc3) (lcable3/nc3)^2 + (1/4) (mcable3/nc3)
rcable3^2,(*moment of inertia of a cable element about y*)
Izcable3 -> (1/12) (mcable3/nc3) (lcable3/nc3)^2 + (1/4) (mcable3/nc3)
rcable3^2,(*moment of inertia of a cable element about z*)

kx3 -> refleng/(lcable3/(nc3 + 1)) kx5cm, (*The spring constants of the
cable elements*)
ky3 -> refleng/(lcable3/(nc3 + 1)) ky5cm,
kz3 -> refleng/(lcable3/(nc3 + 1)) kz5cm,
kyaw3 -> kyaw5cm refleng^3/3/(lcable3/nc3),
kpitch3 -> kpitch5cm refleng^3/3/(lcable3/nc3),
```

```
kroll3 -> refleng/(lcable3/(nc3 + 1)) kroll5cm,

damping[imag, cabletype3] -> (.01 &),

(*CABLE 2 - UIM to PM *)

nc2 -> 15,
mpl2 -> .032, (*Mass per length of the cable*)
rog2 -> rcable2/Sqrt[2],
lcable2 -> 0.15, (*cable length, meters, guesstimate *)
rcable2 -> .0005, (*cable radius*)
mcable2 -> lcable2*mpl2, (*the total mass of the cable*)
Ixcable2 -> (1/2) (mcable2/nc2) rcable2^2,(*moment of inertia of a cable
element about x*)
Iycable2 -> (1/12) (mcable2/nc2) (lcable2/nc2)^2 + (1/4) (mcable2/nc2)
rcable2^2,(*moment of inertia of a cable element about y*)
Izcable2 -> (1/12) (mcable2/nc2) (lcable2/nc2)^2 + (1/4) (mcable2/nc2)
rcable2^2,(*moment of inertia of a cable element about z*)

kx2 -> refleng/(lcable2/(nc2 + 1)) kx5cm, (*The spring constants of the
cable elements*)
ky2 -> refleng/(lcable2/(nc2 + 1)) ky5cm,
kz2 -> refleng/(lcable2/(nc2 + 1)) kz5cm,
kyaw2 -> kyaw5cm refleng^3/3/(lcable2/nc2),
kpitch2 -> kpitch5cm refleng^3/3/(lcable2/nc2),
kroll2 -> refleng/(lcable2/(nc2 + 1)) kroll5cm,

damping[imag, cabletype2] -> (.01 &),

(* CABLE 1 - top mass to UIM *)

nc1 -> 15,
mpl1 -> .032, (*Mass per length of the cable*)
rog1 -> rcable1/Sqrt[2],
lcable1 -> 0.3, (*cable length, meters, guesstimate *)
rcable1 -> .0005, (*cable radius*)
mcable1 -> lcable1*mpl1, (*the total mass of the cable*)
Ixcable1 -> (1/2) (mcable1/nc1) rcable1^2,(*moment of inertia of a cable
element about x*)
Iycable1 -> (1/12) (mcable1/nc1) (lcable1/nc1)^2 + (1/4) (mcable1/nc1)
rcable1^2,(*moment of inertia of a cable element about y*)
Izcable1 -> (1/12) (mcable1/nc1) (lcable1/nc1)^2 + (1/4) (mcable1/nc1)
rcable1^2,(*moment of inertia of a cable element about z*)

kx1 -> refleng/(lcable1/(nc1 + 1)) kx5cm, (*The spring constants of the
cable elements*)
ky1 -> refleng/(lcable1/(nc1 + 1)) ky5cm,
kz1 -> refleng/(lcable1/(nc1 + 1)) kz5cm,
kyaw1 -> kyaw5cm refleng^3/3/(lcable1/nc1),
kpitch1 -> kpitch5cm refleng^3/3/(lcable1/nc1),
kroll1 -> refleng/(lcable1/(nc1 + 1)) kroll5cm,
```

```
damping[imag, cabletype1] -> (.01 &),

(*CABLE 0 - structure to top mass *)

nc0 -> 15,
mpl0 -> .032, (*Mass per length of the cable*)
rog0 -> rcable0/Sqrt[2],
lcable0 -> 0.50, (*cable length, meters, guesstimate *)
rcable0 -> .0005, (*cable radius*)
mcable0 -> lcable0*mpl0, (*the total mass of the cable*)
Ixcable0 -> (1/2) (mcable0/nc0) rcable0^2,(*moment of inertia of a cable
element about x*)
Iycable0 -> (1/12) (mcable0/nc0) (lcable0/nc0)^2 + (1/4) (mcable0/nc0)
rcable0^2,(*moment of inertia of a cable element about y*)
Izcable0 -> (1/12) (mcable0/nc0) (lcable0/nc0)^2 + (1/4) (mcable0/nc0)
rcable0^2,(*moment of inertia of a cable element about z*)

kx0 -> refleng/(lcable0/(nc0 + 1)) kx5cm, (*The spring constants of the
cable elements*)
ky0 -> refleng/(lcable0/(nc0 + 1)) ky5cm,
kz0 -> refleng/(lcable0/(nc0 + 1)) kz5cm,
kyaw0 -> kyaw5cm refleng^3/3/(lcable0/nc0),
kpitch0 -> kpitch5cm refleng^3/3/(lcable0/nc0),
kroll0 -> refleng/(lcable0/(nc0 + 1)) kroll5cm,

damping[imag, cabletype0] -> (.01 &)
};
```

8.2 Definition of cables

```
cablelist = {
 {
 {x2, y2, z2, yaw2, pitch2, roll2},
 {0, 0, -ir},
 {0, 0, 0},
 {x3, y3, z3, yaw3, pitch3, roll3},
 {0, 0, +tr},
 {0, p, 0},
 cabletype3,
 DiagonalMatrix[{kx3, ky3, kz3, kyaw3, kpitch3, kroll3}],
 {0, 0, 0, 0, 0, 0},
 nc3,
 lcable3,
 mpl3,
 rog3,
 DiagonalMatrix[{Ixcable3, Iycable3, Izcable3}]
 },
 {
 {x2, y2, z2, yaw2, pitch2, roll2},
 {0, 0, -ir},
 {0, p, 0},
 {x3, y3, z3, yaw3, pitch3, roll3},
 {0, 0, +tr},
```

```
{0, 0, 0},
cabletype3,
DiagonalMatrix[{kx3, ky3, kz3, kyaw3, kpitch3, kroll3}],
{0, 0, 0, 0, 0, 0},
nc3,
lcable3,
mpl3,
rog3,
DiagonalMatrix[{Ixcable3, Iycable3, Izcable3}]
},
{
{x1, y1, z1, yaw1, pitch1, roll1},
{0, 0, -uz/2},
{0, 0, 0},
{x2, y2, z2, yaw2, pitch2, roll2},
{0, 0, +ir},
{0, p, 0},
cabletype2,
DiagonalMatrix[{kx2, ky2, kz2, kyaw2, kpitch2, kroll2}],
{0, 0, 0, 0, 0, 0},
nc2,
lcable2,
mpl2,
rog2,
DiagonalMatrix[{Ixcable2, Iycable2, Izcable2}]
},
{
{x1, y1, z1, yaw1, pitch1, roll1},
{0, 0, -uz/2},
{0, p, 0},
{x2, y2, z2, yaw2, pitch2, roll2},
{0, 0, +ir},
{0, 0, 0},
cabletype2,
DiagonalMatrix[{kx2, ky2, kz2, kyaw2, kpitch2, kroll2}],
{0, 0, 0, 0, 0, 0},
nc2,
lcable2,
mpl2,
rog2,
DiagonalMatrix[{Ixcable2, Iycable2, Izcable2}]
},
{
{x0, y0, z0, yaw0, pitch0, roll0},
{0, 0, -nz/2},
{0, 0, 0},
{x1, y1, z1, yaw1, pitch1, roll1},
{0, 0, +uz/2},
{0, p, 0},
cabletype1,
DiagonalMatrix[{kx1, ky1, kz1, kyaw1, kpitch1, kroll1}],
{0, 0, 0, 0, 0, 0},
nc1,
lcable1,
mpl1,
rog1,
DiagonalMatrix[{Ixcable1, Iycable1, Izcable1}]
},
```

```
{
{x0, y0, z0, yaw0, pitch0, roll0},
{0, 0, -nz/2},
{0, p, 0},
{x1, y1, z1, yaw1, pitch1, roll1},
{0, 0, +uz/2},
{0, 0, 0},
cabletype1,
DiagonalMatrix[{kx1, ky1, kz1, kyaw1, kpitch1, kroll1}],
{0, 0, 0, 0, 0, 0},
nc1,
lcable1,
mpl1,
rog1,
DiagonalMatrix[{Ixcable1, Iycable1, Izcable3}]
},
{
{x00, y00, z00, yaw00, pitch00, roll00},
{0, 0, 0},
{0, 0, 0},
{x0, y0, z0, yaw0, pitch0, roll0},
{0, 0, +nz/2},
{0, p, 0},
cabletype0,
DiagonalMatrix[{kx0, ky0, kz0, kyaw0, kpitch0, kroll0}],
{0, 0, 0, 0, 0, 0},
nc0,
lcable0,
mpl0,
rog0,
DiagonalMatrix[{Ixcable0, Iycable0, Izcable0}]
},
{
{x00, y00, z00, yaw00, pitch00, roll00},
{0, 0, 0},
{0, p, 0},
{x0, y0, z0, yaw0, pitch0, roll0},
{0, 0, +nz/2},
{0, 0, 0},
cabletype0,
DiagonalMatrix[{kx0, ky0, kz0, kyaw0, kpitch0, kroll0}],
{0, 0, 0, 0, 0, 0},
nc0,
lcable0,
mpl0,
rog0,
DiagonalMatrix[{Ixcable0, Iycable0, Izcable0}]
}
};
```