LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
-LIGO-
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note	**LIGO-T000149- 00- D**	8/22/00

Design Tool: LIGO Stage Designer 2

Dan DeBra, Brian Lantz

This is an internal working note
of the LIGO Project.

California Institute of Technology
LIGO Project – MS 51-33
Pasadena CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project – MS 20B-145
Cambridge, MA 01239
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

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

```
function []=LIGO_stage_designer();
% springs_for_ETF.m
% tapered blade spring and torsion bar springs scaling.
%
% DeBra 2000 aug 2
% modified by B. Lantz Aug 22 2000 to
% run parameters for the ETF HAM design
%
% updated with new masses on June 16, GH, BTL

% updated with design masses for final system - BTL Dec 4 2002
%close all

format short e
format compact
mfilename

E=1.86e11;
% corwin says
% 1/10 to 1/5
% yield 290kpsi for C300
C300yield=2e9;     %Pascals
stress= .25 * C300yield;  % 20% of yield is 4e8 Pascals
width_ratio = .5;       % base/length

g=9.8;
LBperNEWTON=2.204/9.8;  % pounds per Newton;

mfilename

% total load on the inner stage, including ballast
MStaticLoad  = 500;
MDynamicLoad = 500;

%fraction of payload which is dynamic (not the ballast)
Mouter   = 700;
Minner   = 750 + MStaticLoad
Mpayload = MDynamicLoad

ns=3;
max_force_inner = 3*10/LBperNEWTON;     % 3 actuators each with 10 lbs
max_force_outer = 3*50/LBperNEWTON;             % 3 actuators each with 50 lbs


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%             inner stage              %%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp(' ')
disp('           inner stage HAM')
disp(' ')

SupportedMass=Minner+Mpayload   % mass of outer, inner, and payload
Load=g*SupportedMass/ns;
inner_dist=1e-4 % 0.1mm
kmax = max_force_inner/inner_dist;

Freq=[2,3,4];  % res freq in Hertz

w0=2*pi*Freq;
kvert=Minner * w0.^2            % this is the total stiffness
Freq = [Freq, sqrt(kmax/Minner)/(2*pi)]
kvert = [kvert,kmax]
kinner=kvert
% one gets the equivalent force of 3 actuators to push verticaly
```

```
disp(sprintf('force per actuator (in pounds) to push the second stage vertically by
%g microns',1e6*inner_dist))
F = 1/3 * inner_dist  * LBperNEWTON * kinner  % answer in pounds

% stiffness per spring
kv3=kinner(1)/3;
kv4=kinner(2)/3;
kv5=kinner(3)/3;
kvmax = kinner(4)/3

L=logspace(-1,0);

B3 =      Base(E,Load,L,stress,kv3);
h3 = thickness(E,Load,L,stress,kv3);

B5 =      Base(E,Load,L,stress,kv4);
h5 = thickness(E,Load,L,stress,kv4);

B7 =      Base(E,Load,L,stress,kv5);
h7 = thickness(E,Load,L,stress,kv5);

Bm =      Base(E,Load,L,stress,kvmax);
hm = thickness(E,Load,L,stress,kvmax);

min_thickness_inner = sqrt(6/width_ratio)*sqrt(Load/stress)


figure
pp = loglog(L,B7,'b-', L,h7,'m-', ...
       L,B5,'b--', L,h5,'m--', ...
                 L,B3,'b:',L,h3,'m:', ...
                 L,Bm,'b-',L,hm,'m-');
set(pp(7),'LineWidth',2)
set(pp(8),'LineWidth',2)

axis([.1 1 1e-3 1]);
set(gca,'FontSize',12)
xlabel('blade spring length, L (m)')
ylabel('size (meters)')
title(sprintf('Inner stage blade design for ETF St=%1.1e (Pa);  ns=%1i',stress,ns))
legend('base','thickness')
grid on
axis;

indexInner=max(find(L<=.15));

text(.15,h3(indexInner),sprintf('%1.1f Hz',Freq(1)))
text(.15,h5(indexInner),sprintf('%1.1f Hz',Freq(2)))
text(.15,h7(indexInner),sprintf('%1.1f Hz',Freq(3)))
text(.15,hm(indexInner),sprintf('%1.1f Hz',Freq(4)))

%%%%%% draw some limits on the graph  %%%%%%%
hold on

pp = line([.1 .5],width_ratio*[.1 .5]);
set(pp,'Color','b','LineWidth',2)
text(.5,width_ratio*.5,'max base')

max_length = .5;
pp = line(max_length*[1 1],[.001 1]);
set(pp,'Color','k')
tt = text(max_length, .0011,'max length')
set(tt,'Rotation',90)

max_thick = .02;
```

```
pp = line([.1 1],max_thick*[1 1]);
set(pp,'Color','m','LineWidth',2)
tt = text(.11, max_thick,'max thickness')

pp = line([.1 1],min_thickness_inner*[1 1]);
set(pp,'Color','m','LineWidth',2)
tt = text(.11, min_thickness_inner,'min thickness')

finner=input('freq of inner stage? ');
w1=2*pi*finner;
kin=Minner*w1^2;
finlbs=1/3 * inner_dist  * LBperNEWTON * kin  % answer in pounds
disp(sprintf('the total vertical inner stage stiffness is %2.2e N/m',kin))
disp(sprintf('to push %g microns, each inner stage vertical actuator needs %g 
pounds',1e6*inner_dist,finlbs))

inner_length = input('choose the blade length (m): ');
inner_base   =      Base(E,Load,inner_length,stress,kin/3)
inner_thick  = thickness(E,Load,inner_length,stress,kin/3)
hold on
loglog(inner_length,inner_base,'bx')
loglog(inner_length,inner_thick,'mx')


%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%             outer stage               %%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp(' ')
disp('           outer stage HAM')
disp(' ')

SupportedMass=Mouter+Minner+Mpayload ;  % mass of outer, inner, and payload
Load=g*SupportedMass/ns;        % load per spring

freqMin=sqrt(kin/Mouter)/(2*pi);                        % minimum outer stage freq, 
if inner stage is fixed in space

Freq=freqMin+[.5,2,4];  % res freq in Hertz
w0=2*pi*Freq;   % natural freq of outer stage if inner stage is fixed, and inner and 
outer springs provide force
ktotal=Mouter*w0.^2
kvert=ktotal-kin                % total vertical stiffness of the outer stage 
springs
%kvert=1e6*[.5 1 2];
outer_dist=3e-4 ;
% 0.3mm
kmaxouter = max_force_outer/outer_dist;
kvert = [kvert,kmaxouter]
Freq = [Freq, sqrt((kmaxouter+kin)/Mouter)/(2*pi)];
kouter=kvert;

% one gets the equivalent force of 3 actuators to push vertically
disp(sprintf('force per actuator (in pounds) to push the outer stage vertically by 
%g microns',1e6*outer_dist))

F = 1/3 * outer_dist  * LBperNEWTON * kvert  % answer in pounds



kv3=kvert(1)/3;
kv4=kvert(2)/3;
kv5=kvert(3)/3;
```

```
kvmax = kvert(4)/3;

L=logspace(-1,0);



B3 =      Base(E,Load,L,stress,kv3);
h3 = thickness(E,Load,L,stress,kv3);

B5 =      Base(E,Load,L,stress,kv4);
h5 = thickness(E,Load,L,stress,kv4);

B7 =      Base(E,Load,L,stress,kv5);
h7 = thickness(E,Load,L,stress,kv5);

Bm =      Base(E,Load,L,stress,kvmax);
hm = thickness(E,Load,L,stress,kvmax);

min_thickness_outer = sqrt(6/width_ratio)*sqrt(Load/stress)

figure
pp = loglog(L,B7,'b-', L,h7,'m-', ...
       L,B5,'b--', L,h5,'m--', ...
                   L,B3,'b:',L,h3,'m:', ...
                   L,Bm,'b-',L,hm,'m-');
set(pp(7),'LineWidth',2)
set(pp(8),'LineWidth',2)


axis([.1 1 1e-3 1]);
set(gca,'FontSize',12)
xlabel('arm length, L (m)')
ylabel('size (meters)')
title(sprintf('Outer stage blade design for ETF St=%1.1e (Pa);  ns=%1i',stress,ns))
legend('base','thickness')
grid on
axis;

indexOuter=max(find(L<=.15));

text(.15,h3(indexOuter),sprintf('%1.1f Hz',Freq(1)))
text(.15,h5(indexOuter),sprintf('%1.1f Hz',Freq(2)))
text(.15,h7(indexOuter),sprintf('%1.1f Hz',Freq(3)))
text(.15,hm(indexOuter),sprintf('%1.1f Hz',Freq(4)))

%%%%%% draw some limits on the graph  %%%%%%%
hold on

pp = line([.1 .5],width_ratio*[.1 .5]);
set(pp,'Color','b','LineWidth',2)
text(.5,width_ratio*.5,'max base')

max_length = .5;
pp = line(max_length*[1 1],[.001 1]);
set(pp,'Color','k')
tt = text(max_length, .0011,'max length')
set(tt,'Rotation',90)

max_thick = .02;
pp = line([.1 1],max_thick*[1 1]);
set(pp,'Color','m','LineWidth',2)
tt = text(.11, max_thick,'max thickness')

pp = line([.1 1],min_thickness_outer*[1 1]);
set(pp,'Color','m','LineWidth',2)
```

```
tt = text(.11, min_thickness_outer,'min thickness')


% the eigenvalues of this matrix are w0^2,
% make 3 matricies, 1 for each freq
A=zeros(2,2,4);
A(1,1,:)=(kouter+kin)/Mouter;
A(1,2,:)=-kin/Mouter;
A(2,1,:)=-kin/Minner;
A(2,2,:)= kin/Minner
disp('The coupled resonant frequencies are:')
disp(sprintf('%1.1f Hz',Freq(1)))
sqrt(eig(A(:,:,1)))/(2*pi)
disp(sprintf('%1.1f Hz',Freq(2)))
sqrt(eig(A(:,:,2)))/(2*pi)
disp(sprintf('%1.1f Hz',Freq(3)))
sqrt(eig(A(:,:,3)))/(2*pi)
disp(sprintf('%1.1f Hz',Freq(4)))
sqrt(eig(A(:,:,4)))/(2*pi)

disp(' ')
fouter=input('freq of outer stage? ');
wout=2*pi*fouter;
kout=Mouter*wout^2 - kin;
foutlbs=1/3 * outer_dist  * LBperNEWTON * kout  % answer in pounds
disp(sprintf('the total vertical outer stage stiffness is %2.2e N/m',kout))
disp(sprintf('to push %g microns, each outer stage vertical actuator needs %g
pounds',1e6*outer_dist,foutlbs))

outer_length = input('choose the outer blade length (m): ');
outer_base   =      Base(E,Load,outer_length,stress,kout/3)
outer_thick  = thickness(E,Load,outer_length,stress,kout/3)
hold on
loglog(outer_length,outer_base,'bx')
loglog(outer_length,outer_thick,'mx')

% the eigenvalues of this matrix are w0^2,
% compute coupled freqs of final choice
Afinal=zeros(2,2);
Afinal(1,1)=(kout+kin)/Mouter;
Afinal(1,2)=-kin/Mouter;
Afinal(2,1)=-kin/Minner;
Afinal(2,2)= kin/Minner;
disp('The coupled resonant frequencies are (final design):')
sqrt(eig(Afinal(:,:)))/(2*pi)


%%%%%%%%%%%%%%%%%%%%%%%%%%
%      subfunctions      %
%%%%%%%%%%%%%%%%%%%%%%%%%%

function B=Base(ElasticMod,StatForce,Length,StressLevel,kspring)
E=ElasticMod;
F=StatForce;
L=Length;
St=StressLevel;
kv=kspring;
B=(6 * E^2 * F^3)./(L.^3 * St^3 * kv^2 );

function h=thickness(ElasticMod,StatForce,Length,StressLevel,kspring)
E=ElasticMod;
F=StatForce;
L=Length;
St=StressLevel;
kv=kspring;
```

```
h=(L.^2 * kv * St)/(E * F);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%                    results                          %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ÈETF_designer
% Mouter =
%    300
% Minner =
%    500
% Mpayload =
%    200
% stage =
% outer stage HAM
% SupportedMass =
%         1000
% Freq =
%      3     5     7
% kvert =
%    1.0659e+05   2.9609e+05   5.8033e+05
% dist =
%    1.0000e-03
% F =
%    7.9908e+00   2.2197e+01   4.3505e+01
% stage =
% inner stage HAM
% SupportedMass =
%    700
% Freq =
%      3     5     7
% kvert =
%    1.7765e+05   4.9348e+05   9.6722e+05
% dist =
%    1.0000e-04
% F =
%    1.3318e+00   3.6994e+00   7.2509e+00
```