```
function[pend] = quadopt
% Design of quad pendulum with blade spring optimisation v1.01
% Written by KAS  7/99
% edited by CIT 11/99
% changed again by KAS 12/99
% changes to use structure 12/99 KAS
% Calum's/Mike's/Matts blade equations added 12/99 KAS
% routine to optimise blades added 12/99 KAS
% pendulum updated 27/1/00 CIT & KAS
% QUAD PENDULUM MODEL WITH CANTILEVERS
% see pendulum diagram for definitions of parameters

% the parameters are now fixed to represent the REFERENCE DESIGN
% for the LIGO II main optics suspension
%
% coordinates x = longitudinal = u_LIGO roll about this axis
%              y = transverse   = v_LIGO pitch about this axis
%              z = vertical     = w_LIGO yaw about this axis

%*******************************************************************
% MASS (N) is REPRESENTED by a rectangular BLOCK
% in reality it will be larger and less dense.
global pend
 pend.g     = 9.81;
 pend.nx = 0.125;       %dimensions of MASS (N) (square)
 pend.ny = 0.40;
 pend.nz = 0.13;
 pend.denn  = 5600;        %density (steel)
 pend.mn = pend.denn* pend.ny* pend.nz* pend.nx;      %MASS (N)
 pend.Inx   =  pend.mn*( pend.ny^2+ pend.nz^2)/12; %moment of inertia (transverse roll)
 pend.Iny   =  pend.mn*( pend.nz^2+ pend.nx^2)/12; %moment of inertia (longitudinal pitch)
 pend.Inz   =  pend.mn*( pend.ny^2+ pend.nx^2)/12; %moment of inertia (yaw)

%*******************************************************************
% MASS (1) is REPRESENTED by a rectangular BLOCK
% in reality it will be larger and less dense.

 pend.ux = 0.125;       %dimensions MASS (1) (square)
 pend.uy    = 0.40;
 pend.uz = 0.13;
 pend.den1  = 5600;        %density (steel)

 pend.m1 = pend.den1* pend.uy* pend.uz* pend.ux ;  %MASS (1)

 pend.I1x   =  pend.m1*( pend.uy^2+ pend.uz^2)/12; %moment of inertia (transverse roll)
 pend.I1y   =  pend.m1*( pend.uz^2+ pend.ux^2)/12; %moment of inertia (longitudinal pitch)
 pend.I1z   =  pend.m1*( pend.uy^2+ pend.ux^2)/12; %moment of inertia (yaw)

%*******************************************************************
 pend.ix = 0.125;       %dimension of MASS (2) (cylinder)
 pend.ir = 0.16;
 pend.den2 = 5600;
 pend.m2 = pend.den2*pi* pend.ir^2* pend.ix;    %MASS (2)

 pend.I2x   = pend.m2*(pend.ir^2/2);            %moment of inertia (transverse roll)
 pend.I2y   = pend.m2*(pend.ir^2/4+pend.ix^2/12);  %moment of inertia (longitudinal pitch)
```

```
 pend.I2z   = pend.m2*(pend.ir^2/4+pend.ix^2/12);  %moment of inertia (yaw)

%**********************************************************************
 pend.tx    = 0.125;       %dimensions of MASS (3) (cylinder)
 pend.tr    = 0.16;
 pend.den3  = 4000;        %density (sapphire)

 pend.m3    = pend.den3*pi*pend.tr^2*pend.tx;   %MASS (3)

 pend.I3x   = pend.m3*(pend.tr^2/2);            %moment of inertia (transverse roll)
 pend.I3y   = pend.m3*(pend.tr^2/4+pend.tx^2/12);  %moment of inertia (longitudinal pitch)
 pend.I3z   = pend.m3*(pend.tr^2/4+pend.tx^2/12);  %moment of inertia (yaw)

%*********************************************************************
pend.ln = 0.54;      %wire length 1: short version
pend.l1  = 0.304;    %wire length 2: short version
pend.l2  = 0.302;    %wire length 3: short version
pend.l3  = 0.6;         %wire length 4


%**********************************************************************************************
 pend.nwn   = 2; %number of wires (= number of cantilevers if fitted) per stage (2 or 4)
 pend.nw1   = 4;
 pend.nw2   = 4;
 pend.nw3   = 4;

%**********************************************************************************
 pend.rn = 800e-6;   %radius of wire (N)
 pend.r1 = 805e-6;   %radius of wire (1)
 pend.r2 = 602e-6;   %radius of wire (2)
 pend.r3 = 200e-6;   %radius of wire (3)

%***********************************************************************************
 pend.Yn    = 1.65e11;  %Youngs Modulus of wire (N)   (s/steel 302)
 pend.Y1 = 1.65e11;  %Youngs Modulus of wire (1)   (s/steel 302)
 pend.Y2 = 1.65e11;  %Youngs Modulus of wire (2)   (s/steel 302)
 pend.Y3 = 7e10;     %Youngs Modulus of wire (3)   (fused silica)

 %blade design
 mntb = (pend.mn+pend.m1+pend.m2+pend.m3)/2;%total per blade
 mnb  = pend.mn/2;%uncoupled mass
 [uf,lnb,anb,hnb,stn] = opt(mnb,mntb,9e8,0.55,0.12);
 pend.lnb = lnb;
 pend.anb = anb;
 pend.hnb = hnb;
 pend.ufcn = uf;
 pend.stn  = stn;
 pend.intmode_n = 55*hnb*0.37^2/(0.002*lnb^2); %scaled from GEO blade

 %blade design
 m1tb = (pend.m1+pend.m2+pend.m3)/2;%total per blade
 m1b  = pend.m1/2;%uncoupled mass
 [uf,l1b,a1b,h1b,st1] = opt(m1b,m1tb,8.5e8,0.48,0.07);
 pend.l1b = l1b;
 pend.a1b = a1b;
 pend.h1b = h1b;
```

```
 pend.ufc1 = uf;
 pend.st1  = st1;
 pend.intmode_1 = 55*h1b*0.37^2/(0.002*l1b^2); %scaled from GEO blade
%blade design
 m2tb = (pend.m2+pend.m3)/2;%total per blade
 m2b  = pend.m2/2;%uncoupled mass
 [uf,l2b,a2b,h2b,st2] = opt(m2b,m2tb,8e8,0.4,0.07);
 pend.l2b = l2b;
 pend.a2b = a2b;
 pend.h2b = h2b;
 pend.ufc2 = uf;
 pend.st2  = st2;
 pend.intmode_2 = 55*h2b*0.37^2/(0.002*l2b^2); %scaled from GEO blade

%*******************************************************************************************
dees = 0.001;
pend.dm     = dees;  %height of wire break-off above c.of m. mass (N)
pend.dn     = dees;  %height of wire break-off below c.of m. mass (N)
pend.d0     = dees;  %height of wire break-off above c.of m. mass (1)
pend.d1     = dees;  %height of wire break-off below c.of m. mass (1)
pend.d2     = dees;  %height of wire break-off above c.of m. mass (2)
pend.d3     = dees;  %height of wire break-off below c.of m. mass (2)
pend.d4     = dees;  %height of wire break-off above c.of m. mass (3)
%additional information for ribbon breakoffs
%these are needed in translation/roll mode calculations
pend.twistlength = 0.006; %length of twist section in ribbon
pend.d3tr = dees - pend.twistlength;
pend.d4tr = dees - pend.twistlength;
%*******************************************************************************************
%X direction separation
pend.sn  = 0.00;     %1/2 separation of wires (N)
pend.su  = 0.002;    %1/2 separation of wires (1)
pend.si  = 0.002;    %1/2 separation of wires (2)
pend.sl  = 0.007;    %1/2 separation of wires (3)

%*******************************************************************************************
%Y direction separation
pend.nn0 = 0.25;     %1/2 separation of wires (N) at suspension point
pend.nn1 = 0.05;     %1/2 separation of wires (N) at mass (N)
pend.n0  = 0.2;      %1/2 separation of wires (1) at mass (N)
pend.n1 = 0.07;
pend.n2 = 0.12;
pend.n3  =pend.ir+0.0065;  %1/2 separation of wires (2) at mass (2)
pend.n4  =pend.tr+0.0015;  %1/2 separation of wires (3) at mass (2)
pend.n5  =pend.tr+0.0015;  %1/2 separation of wires (3) at mass (3)

%now we can work out the true lengths com to com etc.
pend.tln = sqrt(pend.ln^2 - (pend.nn0-pend.nn1)^2);
pend.tl1 = sqrt(pend.l1^2 - (pend.n0-pend.n1)^2);
pend.tl2 = sqrt(pend.l2^2 - (pend.n2-pend.n3)^2);
pend.tl3 = sqrt(pend.l3^2 - (pend.n4-pend.n5)^2);
pend.l_total  =  pend.tln +pend.tl1+pend.tl2+pend.tl3;

%*****************************************************************************
% represents small loss
pend.bd = 0.01; % makes phases of open loop plots look nicer
```

```
pend.lever_roll = 0.15;
pend.lever_pitch = 0.05;
pend.lever_yaw = 0.15;

%end
```