

Objetivo

Introducir al estudiante al estudio con múltiples cargas y que tome en consideración cuales son las implicaciones que esto conlleva

Estado del arte

El teleférico es un sistema de transporte no tripulado aéreo constituido por cabinas colgadas de una serie de cables que se encargan de hacer avanzar a las unidades a través de las estaciones. Cuando las cabinas van por tierra se denomina funicular.

Las primeras civilizaciones que hicieron uso del transporte por cable fueron las orientales (China, Japón e India) y la antigua civilización inca de Perú.

En Europa no aparece este tipo de transporte hasta la Edad Media, a principios del siglo XVI, empleándose para la construcción de castillos y fortificaciones. En aquella época el cable estaba fabricado por cuerdas de cáñamo y para su funcionamiento se utilizaba la tracción animal o humana.

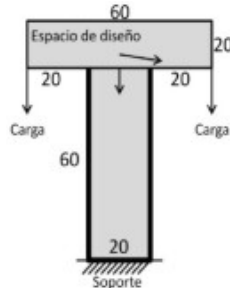
A partir del año 1500 comienza a utilizarse el cable de acero y desde el siglo XVI al XIX se va perfeccionando su trenzado de hilos, contribuyendo al desarrollo del transporte por cable, principalmente para su uso en las minas de carbón.

A partir de los años 30 del siglo XX proliferan los teleféricos con fines turísticos en lugares de montaña, sustituyendo en muchos casos a los funiculares y los trenes cremallera. Sin embargo, es en las décadas de los 60 y los 70 cuando se produce el mayor desarrollo de este tipo de transporte, impulsado principalmente por el auge del esquí y otros deportes de montaña.

Desde entonces hasta ahora los teleféricos han experimentado una gran evolución, construyéndose, hoy en día, instalaciones cada vez más modernas. En su diseño, además de garantizar el buen funcionamiento y la seguridad, se da gran importancia a la innovación, la estética y la comodidad.

Propuesta de diseño de la geometría, alcances y limitaciones

Se tomarán ciertas consideraciones para la solución de esta práctica: 2 cargas, los apoyos tendrán restricciones en Y" y el espacio de diseño para esta práctica será de:



El resultado de la optimización no tendrá en cuenta cargas como viento. El espacio de diseño es un área y que por tanto nuestro resultado final no nos arrojará información de sus dimensiones en el plano lateral.

Pasos del desarrollo de programación:

Declaración de vacío de la figura#1.

```
1      %%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, OCTOBER 1999 %%%
2      function new_pr42_f(nelx,nely,volfrac,penal,rmin);
3      % INITIALIZE
4      x(1:nely,1:nelx) = volfrac;
5      for ely = 1:nely
6          for elx = 1:nelx
7              if ely>21
8                  if elx<21
9                      passive(ely,elx) = 1;
10                     elseif elx>41
11                         passive(ely,elx)=1;
12                     else passive(ely,elx) = 0;
13                 end
14             end
15         end
16     end
```

Declaración de fuerzas:

```
100 % DEFINE LOADS AND SUPPORTS(HALF MBB-BEAM)
101 F(40,1) = -1.; F(9760,2)=1.;
102 fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
103 alldofs = [1:2*(nely+1)*(nelx+1)];
104 freedofs = setdiff(alldofs,fixeddofs);
```

```
%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, OCTOBER 1999 %%%
```

```
function new_pr42_f(nelx,nely,volfrac,penal,rmin);
```

```
% INITIALIZE
```

```
x(1:nely,1:nelx) = volfrac; for
```

```
ely = 1:nely
```

```
    for elx = 1:nelx if
```

```
        ely>21
```

```
            if elx<21
```

```
                passive(ely,elx) = 1;
```

```
            elseif elx>41
```

```
                passive(ely,elx)=1; else
```

```
                passive(ely,elx) = 0; end
```

```
            end
```

```
        end
```

```
end
```

```
x(find(passive))=0.001;
```

```
loop = 0;
```

```
change = 1.;
```

```
% START ITERATION
```

```
while change > 0.01 loop
```

```
    = loop + 1; xold = x;
```

```
% FE-ANALYSIS
```

```
[U]=FE(nelx,nely,x,penal);
```

```
% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS % [KE] =
```

```
lk;
```

```
c = 0.;
```

```
for ely = 1:nely for
```

```
    elx = 1:nelx
```

```
        n1 = (nely+1)*(elx-1)+ely; n2 =
```

```
        (nely+1)* elx +ely;
```

```
        dc(ely,elx)=0.;
```

```
for i=1:2
```

```
    Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1; 2*n2+2; 2*n1+1;2*n1+2],i);
```

```
    c = c + x(ely,elx)^penal*Ue'*KE*Ue;
```

```
    dc(ely,elx) = dc(ely,elx)-penal*x(ely,elx)^(penal-1)* Ue'*KE*Ue;
```

```
end
```

```
end
```

```
end
```

```
% FILTERING OF SENSITIVITIES %
```

```
[dc] = check(nelx,nely,rmin,x,dc);
```

```
% DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
```

```
[x] = OC(nelx,nely,x,volfrac,dc,passive);
```

```
% PRINT RESULTS
```

```
change = max(max(abs(x-xold)));
```

```
disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
```

```
    'Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ... ']
```

```
    'ch.: ' sprintf('%6.3f',change) ])
```

```
% PLOT DENSITIES
```

```
colormap(gray); imagesc(-x); axis equal; axis
```

```
tight;
```

```
axis off;
```

```
pause(1e-6);
```

```
end
```

```
%%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
```

```
function [xnew]=OC(nelx,nely,x,volfrac,dc,passive) l1
= 0;
l2 = 100000;
move = 0.2;
while (l2-l1 > 1e-4) lmid
    = 0.5*(l2+l1);
    xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
    xnew(find(passive))=0.001;
    if sum(sum(xnew)) - volfrac*nelx*nely > 0; l1 =
        lmid;
        else end
        l2 = lmid;
end
```

```
%%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%
```

```
function [dcn]=check(nelx,nely,rmin,x,dc)
dcn=zeros(nely,nelx);
for i = 1:nelx
    for j = 1:nely sum=0.0;
        for k = max(i-round(rmin),1): min(i+round(rmin),nelx)
            for l = max(j-round(rmin),1): min(j+round(rmin), nely) fac =
                rmin-sqrt((i-k)^2+(j-l)^2);
                sum = sum+max(0,fac);
                dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
            end
        end
    end
```

```
end
```

```
d
c
n
(
j
,
i
)
=
d
c
n
(
j
,
i
)
/
(
x
(
j
,
i
)
*
```

```

u                                     e
m                                     n
)                                     d

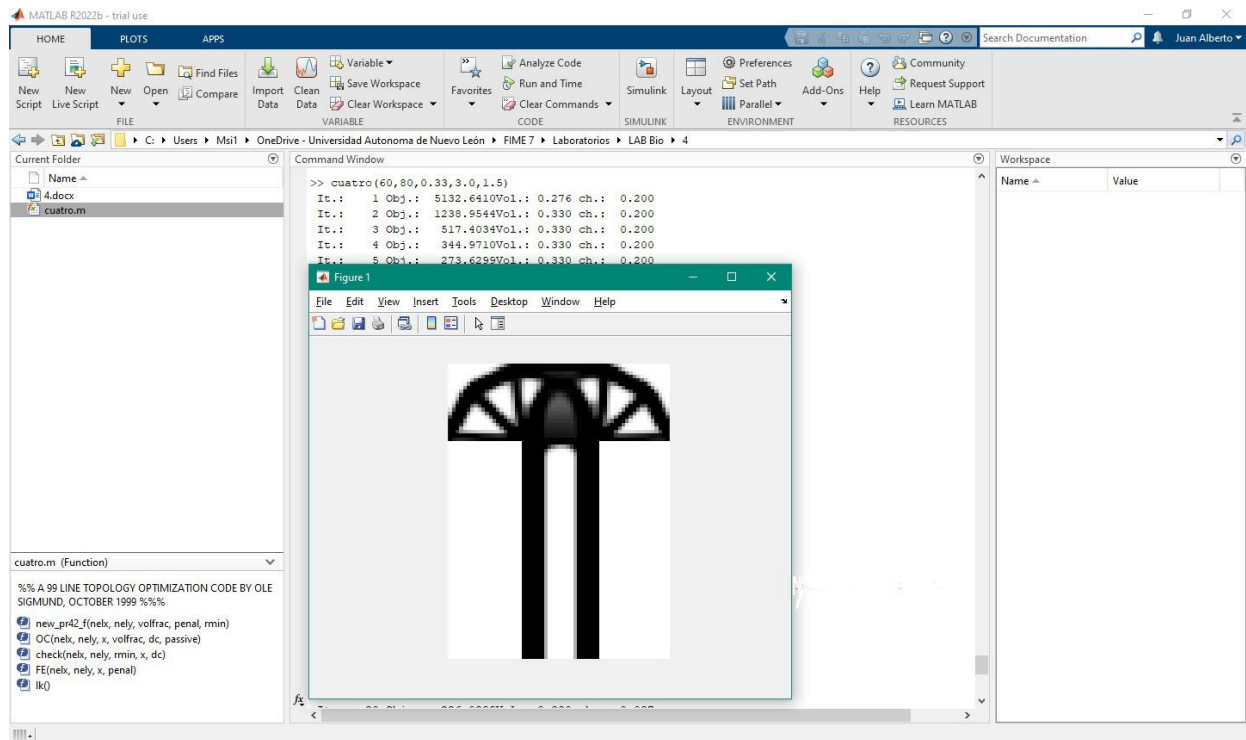
%%%%%%%%%% FE-ANALYSIS %%%%%%%%%%%%%%
function [U]=FE(nelx,nely,x,penal) [KE]
= lk;
K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1)); F =
sparse(2*(nely+1)*(nelx+1),2);
U = sparse(2*(nely+1)*(nelx+1),2); for
ely = 1:nely
    for elx = 1:nelx
        n1 = (nely+1)*(elx-1)+ely; n2 =
        (nely+1)* elx +ely;
        edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2;2*n1+1; 2*n1+2]; K(edof,edof) =
        K(edof,edof) + x(ely,elx)^penal*KE;
    end
end
% DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
F(40,1) = -1.; F(9760,2)=1.;
fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs,fixeddofs);
% SOLVING
U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:); U(fixeddofs,:) = 0;
%%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%
function [KE]=lk E =
1.;
nu = 0.3;

```

```

k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
    -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
KE = E/(1-nu^2)* [ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
    k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
    k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
    k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
    k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
    k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
    k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
    k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
  
```

Resultado de la optimización



Conclusiones:

Victor Cristopher Santiago Martinez: En conclusión con la práctica anterior, el desarrollo de la optimización topológica de la estructura o soporte de cable para un teleférico muestra como la geometría del mismo cambia respecto a cómo era originalmente. Dando como resultado su optimización, que a diferencia de su forma inicial se interpreta como una estructura mayor desarrollada gracias a su optimización.

Referencias:

99 Line Topology Optimization Code – O. Sigmund, Department of Solid Mechanics, Building 404, Technical University of Denmark, DK-2800 Lyngby, Denmark.

Nishiwaki, S., Frecker, M. I., Min, S. & Kikuchi, N. (1998). Topology optimization of compliant mechanisms using the homogenization method. Int. J. Numer. Meth. Engng. 42, 535 -559