



Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Laboratorio de Biomecánica

Reporte de Practica 4:

Refuerza de un cable teleférico

Docente: Dra. Yadira Moreno Vera

Día clase: martes

Hora: N3

Alumno	Matricula	Carrera
Sergio Jared Moreno Rodríguez	1880525	IMTC
Melanie Sofia Sánchez Barbosa	1902265	IMTC
Irving Raúl Garza Escobar	1910127	IMTC
Alejandro Gaytán Ayala	1991902	IMTC
Hernan Abif Castillo Mota	1992201	IMTC
Luis Gerardo Castillo Troncoso	1992317	IMTC

Número de equipo: 4

Brigada 202

Fecha: martes 7 de noviembre de 2022

Semestre agosto-diciembre 2022

Objetivo

El estudiante deberá presentar una propuesta de análisis de formas y de la programación para la ejecución de la optimización (descripción funcional) de características de trabajo específicas que presenta la(s) ventaja(s) (mencionar ventajas).

1.- Introducción

En esta práctica trabajaremos en una propuesta de análisis de formas y de la programación para la ejecución de la optimización.

El objetivo de las técnicas de optimización es mejorar el programa objeto para que nos dé un rendimiento mayor. La mayoría de estas técnicas vienen a compensar ciertas ineficiencias que aparecen en el lenguaje fuente, ineficiencias que son inherentes al concepto de lenguaje de alto nivel, el cual suprime detalles de la máquina objeto para facilitar la tarea de implementar un algoritmo.

Cada optimización está basada en una función de coste y en una transformación que preserve el significado del programa. Mediante la función de coste queremos evaluar la mejora que hemos obtenido con esa optimización y si compensa con el esfuerzo que el compilador realiza para poder llevarla a cabo. Los criterios más comunes que se suelen emplear son el ahorro en el tamaño del código, la reducción del tiempo de ejecución y la mejora de las necesidades del espacio para los datos del programa.

Algunas de sus ventajas: de estos procesos de optimización son la mitigación de riesgos, reducción de costos, mejores resultados, mayor eficiencia, gestión de tiempo optimizada, mejor visión general, mayor capacidad de resolución de problemas, cumplimiento bajo control.

2.- Nombre y definición de la forma geométrica

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. Este consta de varios elementos capaces de hacer funcionar su objetivo, para esta práctica se analizará su cableado o cable.

El **cable** da nombre a todos los sistemas de transporte por cable, los teleféricos. Los cables de acero están compuestos de hilos de cable que se retuercen alrededor del núcleo del cable. Empresas especializadas fabrican los cables y los montan en el lugar. A continuación, se presentarán las ventajas y desventajas de utilizar este cableado de acero.



Figura 1: Teleférico.

Ventajas:

- Es de manipulación más fácil que el cable Lang.
- Tiene tendencia a girar y a des cablearse.
- Por su estructura, es más resistente al aplastamiento y a las deformaciones.

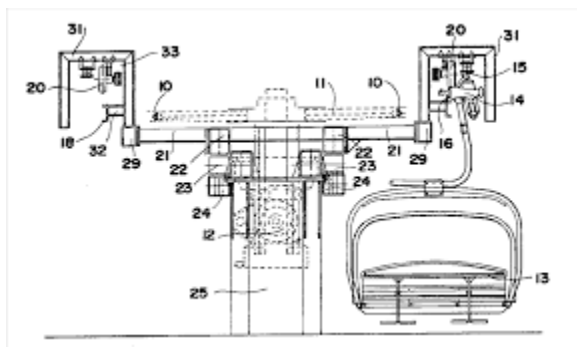
Inconvenientes:

- Menos resistencia al rozamiento, porque posee una menor superficie de contacto con las poleas.
- A igualdad de composición y diámetro, es menos flexible que el cable Lang.
- El cable cruzado es empleado normalmente en las instalaciones de ascensores accionados por poleas de arrastre con gargantas en V, y cuando el cable debe enrollarse en varias capas sobre el tambor a elevadas presiones.

Aplicaciones:

- Deporte y ocio.
- Alambre para vías de cuerda aérea.
- Alambres para cables para remolques de esquíes (elevadores de persona).
- Alambres para cables para elevadores de sillas y elevadores de góndola.
- Alambre para cables de transporte para funiculares.
- Alambre para teleféricos para transporte de personas.
- Alambre para cable para transporte de material (grúas de cuerda/vías de cuerda para fletes).

La forma geométrica para analizar se trata de la siguiente:



En la cual consiste en un soporte estructural en torres, las cuales son estructuras tubulares con la robustez suficiente para soportar el paso de las cabinas o sillas sujetas al cable. La altura de la piona varia en función del terreno donde este ubicada cada una de ellas, pudiendo ser de unos pocos metros, suficiente para elevar las sillas y cabinas del suelo evitando los obstáculos hasta varias decenas de metros según la demanda del recorrido. Las cabinas y el cable pasan por los dos lados de la piona en diferente sentido de la marcha.

3.- Estado del arte

Diseño de sistema de transporte por cable para productos agrícolas cosechados en terreno con pendiente	Christian Fabian Guerrero Ocampo	Christian Fabian Guerrero Ocampo (2017) Diseño de sistema de transporte .Facultad de Ingeniería en Mecánica y Ciencias de la Producción https://www.dspace.pol.edu.ec/retrieve/bc284b82-6027-4649-ae26-1fbc5b0a5496/D-CD88505.pdf	Capítulo de libro PDF	Este proyecto se desarrolló en cuatro etapas: investigación, diseño, modelado y prototipo virtual. Los componentes mecánicos del diseño tales como el sistema motriz, transmisión de potencia, torres de apoyo, garruchas y poleas en torres de apoyo fueron modelados mediante el uso del software Autodesk Inventor CAD	Entender mejor la función estructural del teleférico en su extenso análisis.
Criterios y alternativas para el diseño de un teleférico	Ing. Marcelo Romo Proaño, M.Sc. , David Checa C.y Santiago Zambrano E.	Marcelo Romo Proaño (2016) Criterios y alternativas para el diseño de un teleférico. Centro de Investigaciones Científicas Escuela Politécnica del Ejército. https://ingenieria.unach.mx/images/Articulos_revista/revistapakbal_43_pag69-77.pdf	Capítulo de libro PDF	El presente artículo estudia el comportamiento de las estructuras requeridas para viabilizar proyectos de teleféricos como medios de transporte; también incluye ciertos aspectos mecánicos del mismo. Se discute el papel de los diferentes componentes estructurales requeridos en los teleféricos.	Encontrar alternativas de diseño del teleférico.

Transporte por cable	Alfonso Orro Arcay Margarita Novales Ordax Miguel Rodríguez Bugarín	Alfonso Orro Arcay Margarita Novales Ordax Miguel Rodríguez Bugarín (2013) Transporte por cable Escuela Técnica Superior de Ingenieros de Caminos, Canales y Puertos. Universidad de la coruña. https://es.scribd.com/document/380959662/Transporte-Par-Cable-Teleferico	Capítulo de Libro PDF	El transporte por cable es un instrumento adecuado, en ocasiones el único, para resolver determinadas necesidades de transporte. Se emplea principalmente para transportes en pequeñas distancias con desniveles excepcionales u orografía especialmente complicada, así como en algunas aplicaciones turísticas por el propio atractivo de este tipo de transportes. Si bien existe un campo para el transporte por cable para mercancías, este texto se centra en el transporte por cable de viajeros.	Entender la función del transporte por teleférico y para entender los conceptos y aplicarlo en un modelo ideal.
----------------------	---	---	-----------------------	--	---

4.- Propuesta de diseño de la geometría, alcances y limitaciones

Los teleféricos tienen la característica de ejercer mucha fuerza sobre sus extremos, ya que de ellos depende la suspensión en el aire de la cabina, por lo que la propuesta de diseño se basa en el diseño de un refuerzo en su apoyo.

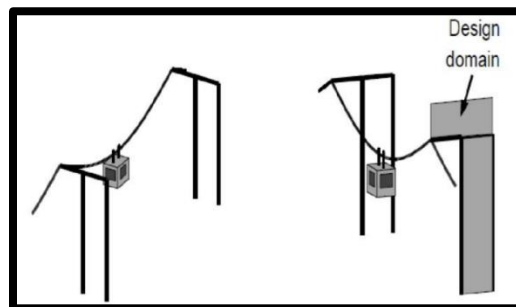


Figura 2: Punto de interés del teleférico.

Aunque hay que tener en cuenta que se realizaran dos propuestas de diseño, la primera es para el teleférico anterior donde está hecho para soportar una sola carga.

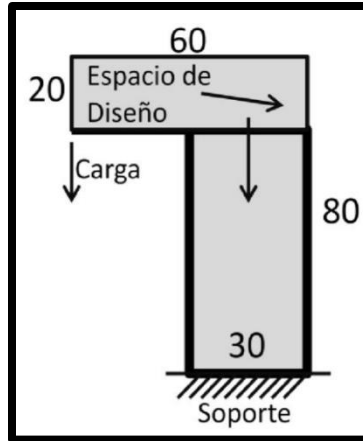


Figura 3: Espacio de diseño para una sola carga.

La segunda propuesta se basa en realizar mejoras a la estructura en cuestión, pero con la capacidad de llevar dos teleféricos a la vez, es decir que sea capaz de soportar múltiples cargas. Como se muestra en la siguiente figura.

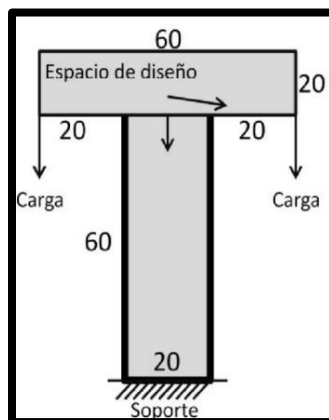


Figura 4 Espacio de diseño para dos cargas.

5.- Pasos del desarrollo de la programación

Se desarrollan dos códigos para la optimización de 2 casos.

Código de la primera figura

```

%%% Laboratorio de Biomecánica Equipo 4%%%
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<31
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
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
l2 = lmid;
end
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
dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
end
end

%%%%%%%%%% 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;
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)];

```

Código de la segunda figura

```
%%% Laboratorio de Biomecánica Equipo 4%%%  
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

```

```

l2 = lmid;
end
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
dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 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)];

```

6.- Resultados de la optimización

Capturas del código de la primera figura.

```
Editor - D:\Abnf\Archivos de programa\Mathlab\bin\Practica_4_Lab_Biomecanicam
Practica_4_Lab_Biomecanicam x +
1 %%% Laboratorio de Biomecánica Equipo 4%%
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<31
9 passive(ely,elx) = 1;
10 else
11 passive(ely,elx) = 0;
12 end
13 end
14 end
15 end
16 x(find(passive))=0.001;
17 loop = 0; change = 1.;
18 % START ITERATION
19 while change > 0.01
20 loop = loop + 1;
21 xold = x;
22 % FE-ANALYSIS
23 [U]=FE(nelx,nely,x,penal);
24 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
25 [KE] = lk;
26 c = 0.;
27 for ely = 1:nely
28 for elx = 1:nelx
29 n1 = (nely+1)*(elx-1)+ely;
30 n2 = (nely+1)* elx +ely;
31 dc(ely,elx)=0.;
32 for i=1:2
33 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);
34 c = c + x(ely,elx)*penal*Ue'*KE*Ue;
35 dc(ely,elx) = dc(ely,elx)-penal*x(ely,elx)^(penal-1)* Ue'*KE*Ue;
36 end
37 end
38 end
39 % FILTERING OF SENSITIVITIES
40 [dc] = check(nelx,nely,rmin,x,dc);
41 % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
42 [x] = OC(nelx,nely,x,volfrac,dc,passive);
43 % PRINT RESULTS
44 change = max(max(abs(x-xold)));
45 disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
46 'Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
47 ' ch.: ' sprintf('%6.3f',change )])
48 % PLOT DENSITIES
49 colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
50 end
51 %~~~~~ OPTIMALITY CRITERIA UPDATE ~~~~~
52 function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
53 l1 = 0; l2 = 100000; move = 0.2;
54 while (l2-l1 > 1e-4)
55 lmid = 0.5*(l2+l1);
56 xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid))));
57 xnew(find(passive))=0.001;
58 if sum(sum(xnew)) - volfrac*nelx*nely > 0.2
59 else
60 l2 = lmid;
61 end
62 end
63 end
64 %~~~~~ MESH-INDEPENDENCY FILTER ~~~~~
65 function [dcn]=check(nelx,nely,rmin,x,dc)
66 dcn=zeros(nely,nelx);
67 for i = 1:nelx
68 for j = 1:nely
69 sum=0.0;
70 for k = max(i-round(rmin),1): min(i+round(rmin),nelx)
71 for l = max(j-round(rmin),1): min(j+round(rmin), nely)
72 fac = rmin-sqrt((l-k)^2+(j-l)^2);
73 sum = sum+max(0,fac);
74 dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
75 end
76 end
77 dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
78 end
79 end
80 %~~~~~ FE-ANALYSIS ~~~~~
81 function [U]=FE(nelx,nely,x,penal)
82 [KE] = lk;
83 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
84 F = sparse(2*(nelx+1)*(nelx+1),2); U = sparse(2*(nely+1)*(nelx+1),2);
85 for ely = 1:nely
86 for elx = 1:nelx
87 n1 = (nely+1)*(elx-1)+ely;
88 n2 = (nely+1)* elx +ely;
```

```

Editor - D:\Abnf\Archivos de programa\Mathlab\bin\Practica_4\lab_Biomecanicam
Practica_4\lab_Biomecanicam
87 - n1 = (nely+1)*(elx-1)+1;
88 - n2 = (nely+1)*elx +1;
89 - edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
90 - K(edof,edof) = K(edof,edof) + x(elx,ely)^penal*KE;
91 - end
92 - end
93 - % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
94 - F(40,1) = -1;
95 - fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
96 - alldofs = [1:2*(nely+1)*(nelx+1)];
97 - freedofs = setdiff(alldofs,fixeddofs);
98 - % SOLVING
99 - U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
100 - U(fixeddofs,:) = 0;
101 - % ELEMNT STIFFNESS MATRIX
102 - function [KE]=lk
103 - E = 1.;
104 - nu = 0.3;
105 - k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
106 - -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
107 - KE = E/(1-nu^2)* [ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
108 - k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
109 - k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
110 - k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
111 - k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
112 - k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
113 - k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
114 - k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

Resultado de la optimización



Figura 5. Resultado de optimización primer caso.

Capturas del código de la segunda figura.

```

Editor - D:\Abnf\Archivos de programa\Matlab\bin\Practica_4_Lab_Biomecanicam*
Practica_4_Lab_Biomecanicam* x +
1  %%% Laboratorio de Biomecánica Equipo 4%%
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
13 passive(ely,elx) = 0;
14 end
15 end
16 end
17 end
18 x(find(passive))=0.001;
19 loop = 0; change = 1.;
20 % START ITERATION
21 while change > 0.01
22 loop = loop + 1;
23 xold = x;
24 % FE-ANALYSIS
25 [U]=FE(nelx,nely,x,penal);
26 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
27 [KE] = lk;
28 c = 0.;
29 for ely = 1:nely
30 for elx = 1:nelx
31 n1 = (nely+1)*(elx-1)+ely;
32 n2 = (nely+1)* elx +ely;
33 dc(ely,elx)=0.;
34 for i=1:2
35 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);
36 c = c + x(ely,elx)^penal*Ue'*KE*Ue;
37 dc(ely,elx) = dc(ely,elx)-penal*x(ely,elx)^(penal-1)* Ue'*KE*Ue;
38 end
39 end
40 end
41 % FILTERING OF SENSITIVITIES
42 [dc] = check(nelx,nely,rmin,x,dc);
43 % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
44 [x] = OC(nelx,nely,x,volfrac,dc,passive);
45 % PRINT RESULTS
46 change = max(max(abs(x-xold)));
47 disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
48 'Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
49 ' ch.: ' sprintf('%6.3f',change )])
50 % PLOT DENSITIES
51 colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
52 end
53 %%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%
54 function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
55 l1 = 0; l2 = 100000; move = 0.2;
56 while (l2-l1 > 1e-4)
57 lmid = 0.5*(l2+l1);
58 xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
59 xnew(find(passive))=0.001;
60 if sum(sum(xnew)) - volfrac*nelx*nely > 0;
61 l1 = lmid;
62 else
63 l2 = lmid;
64 end
65 end
66 %%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%
67 function [dcn]=check(nelx,nely,rmin,x,dc)
68 dcn=zeros(nely,nelx);
69 for i = 1:nelx
70 for j = 1:nely
71 sum=0.0;
72 for k = max(i-round(rmin),1): min(i+round(rmin),nelx)
73 for l = max(j-round(rmin),1): min(j+round(rmin), nely)
74 fac = rmin-sqrt((i-k)^2+(j-l)^2);
75 sum = sum+max(0,fac);
76 dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
77 end
78 end
79 dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
80 end
81 end
82 %%%%%%%%% FE-ANALYSIS %%%%%%%%%
83 function [U]=FE(nelx,nely,x,penal)
84 [KE] = lk;
85 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
86 F = sparse(2*(nely+1)*(nelx+1),2); U = sparse(2*(nely+1)*(nelx+1),2);
87 for ely = 1:nely

```

```

Editor - D:\Abnf\Archivos de programa\Matlab\bin\Practica_4_Lab_Biomecanicam
Practica_4_Lab_Biomecanicam
88 - for elx = 1:nelx
89 -     n1 = (nely+1)*(elx-1)+ely;
90 -     n2 = (nely+1)* elx +ely;
91 -     edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2;2*n1+1; 2*n1+2];
92 -     K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
93 - end
94 - end
95 - % DEFINE LOADSAND SUPPORTS (HALF MBB-BEAM)
96 - F(40,1) = -1.; F(9760,2)=1.;
97 - fixeddofs = 2*(nely+1):2*(nely+1):2*(nelx+1)*(nely+1);
98 - alldofs = [1:2*(nely+1)*(nelx+1)];
99 - freedofs = setdiff(alldofs,fixeddofs);
100 - % SOLVING
101 - U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
102 - U(fixeddofs,:)= 0;
103 - %ELEMENT STIFFNESS MATRIX %%%%%%%%%
104 - function [KE]=lk
105 -     E = 1.;
106 -     nu = 0.3;
107 -     k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
108 -        -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
109 -     KE = E/(1-nu^2)* [ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
110 -        k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
111 -        k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
112 -        k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
113 -        k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
114 -        k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
115 -        k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
116 -        k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

Resultado de la optimización

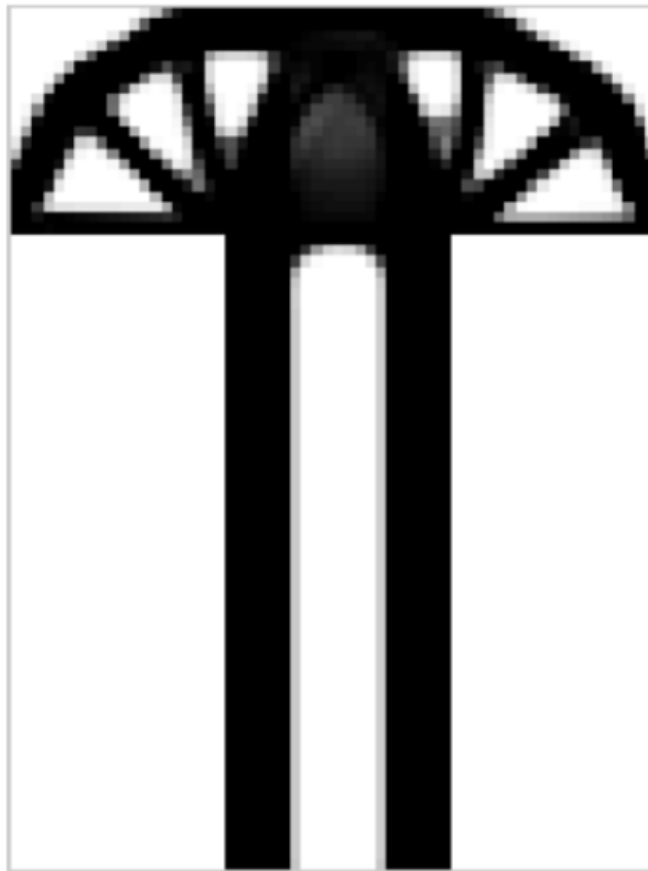


Figura 6. Resultado de optimización segundo caso.

7.- Conclusiones de los autores

Sergio Jared Moreno Rodríguez – 1880525

Melanie Sofia Sánchez Barbosa – 1902265

En esta práctica se realizó una optimización topológica para el refuerzo del soporte de los cables de un teleférico. Fue muy interesante ver una nueva aplicación de la optimización topológica, y de como un mismo código mediante algunas modificaciones es capaz de utilizarse en múltiples áreas, como se ha visto a lo largo de las prácticas. Así mismo, también fue de gran apoyo visualizar el análisis con una y dos cargas, lo que ayuda a entender mejor cómo funciona la optimización y cómo se puede adaptar a diversos requisitos.

Irving Raúl Garza Escobar – 1910127

Se pudo aprender ahora, como se efectúa un análisis topológico, para la optimización de un soporte de teleférico, en el cual fue interesante, ver como el código toma la forma optimizada, con tal de por medio del código, se efectúa diversas características, desde la forma del diseño estructural. Así como su cambio a una optimización, con los esfuerzos que son generados, y de esta dar la forma optimizada, así como las condiciones que son tomada en la alteración del código topológico y como las ecuaciones interactúan con los parámetros para generar esta forma, por lo que también esto resulta interesante de conocer.

Alejandro Gaytán Ayala – 1991902

Como conclusión es importante seleccionar el cable de acero adecuado ya que, dependiendo de su aplicación tiene que superar desafíos específicos. Un factor crítico de éxito es utilizar el cable correcto.

Marcas de cables de acero tales como TEUFELBERGER disponen de los conocimientos técnicos necesarios sobre cada tipo de teleférico por lo que pueden ofrecer el cable adecuado.

La estructura de un cable decide sobre su calidad. La sección y el acabado superficial del cable de acero son dos indicadores de la alta calidad de los cables de acero.

Hernán Abif Castillo Mota – 1992201

Esta práctica de laboratorio se optimizo el diseño de los refuerzos en los apoyos de los teleféricos. Esa estructura es de suma importancia para el funcionamiento correcto de los teleféricos, ya que son los encargados de soportar las cargas del mismo a su vez que permitir el desplazamiento. Por lo que en este trabajo se trabajó con dos espacios de diseño, uno para una carga y otro para dos.

Una prueba más de que incluso las cosas que menos notamos son de suma importancia, de igual forma existen muchos soportes que cumplen esta función en nuestra vida cotidiana pero no nos damos cuenta, por lo que esta práctica nos ayudó a entender mejor el proceso de optimización.

Sergio Jared Moreno Rodriguez-1880525

En esta práctica realizamos el método de optimización topológica para el mecanismo de los cables de un teleférico. Se realizó mediante la optimización ya que sirve para mejorar el programa objeto para que nos dé un rendimiento mayor, ya que por medio de este código se puede analizar el mecanismo. En conclusión pudimos observar que es necesario seleccionar el cable correcto y material correcto ya que debe soportar ciertos esfuerzos.

Luis Gerardo Castillo Troncoso – 1992317

Esta práctica se basa en los refuerzos y como entenderlos para soportar el peso y el cable de un teleférico, por cual es necesario tomar en cuenta el análisis estático de la estructura y a partir de ahí hacer un análisis topológico de la base, vemos ahí en la programación como va tomando forma, en este caso es un modelo ideal ya que no vamos a tomar varios factores que influyen en el peso y contexto del teleférico, viendo un poco algunos documentos del estado del arte los teleféricos sin duda es un transporte muy eficiente tomando en cuenta el tiempo y lugar ya que por sí mismo va de un punto “a” al “b” directo, saltando elevaciones, caminos complicados peligros etc, por lo regular lo vemos en montañas por esa misma condición

8.- Referencias bibliográficas

- -LEITNER. ELEMENTOS DE TELEFÉRICOS. Recuperado de
- <https://www.leitner.com/es/empresa/informaciones-utiles/elementos-de-telefericos/>
- -Benzabala. Cables de acero. Recuperado de
- <https://bezabala.es/wp-content/uploads/2019/04/Cables-de-acero.pdf>