Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

- *Tarea: Refuerzo del cable de un Teleférico.*

Maestro: Isaac Estrada.

Nombre:	**Matricula:**
Felipe Daniel Zamarripa Valdez	1755483
Miguel Angel Morales Arredondo	1987049
Severo Wenceslao Arreola Platas	1915391
Alvaro Alexis García Garza	1605716
Juan Angel Reyna Medrano	1669008

Materia: Laboratorio de Biomecánica Hora: Viernes N5

Cd. Universitaria, San Nicolás de los Garza, N.L.

Refuerzo del cable de un Teleférico

Objetivo:

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

Marco teórico:

Ventajas del cable teleférico:

- Alta resistencia de tensión, lo que resulta en el desempeño de cables de rendimiento superior (= alta carga de rotura para el diámetro de cable establecido).
- Excelente ductilidad del alambre, lo cual resulta en propiedades de torsión de la cuerda óptimas a la fatiga.
- Alambre adecuado para usos compactos y no compactos.
- Gran utilidad y confiabilidad de rendimiento.

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).

Desarrollo:

El teleférico de la figura 1 necesita un refuerzo en su apoyo. Sugiera un refuerzo según la información dada en la figura 2.

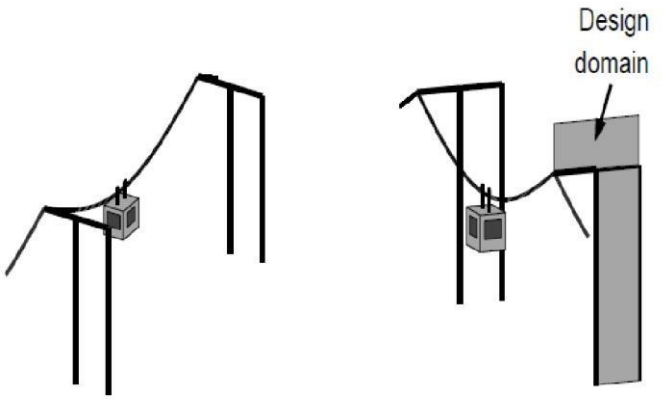


Figura 1: Teleférico

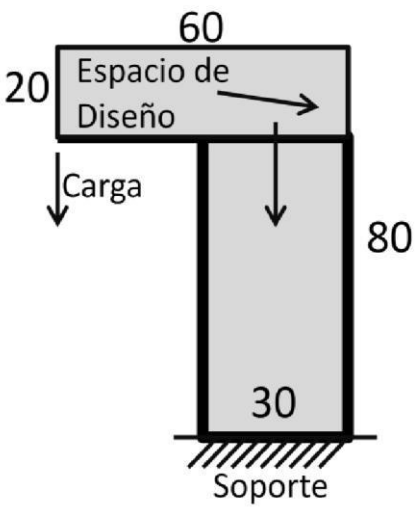


Figura 2: Espacio de diseño

Al cuidador del teleférico también le gustaría que se hicieran mejoras para que la estructura pueda llevar dos teleféricos a la vez, como se ilustra en la figura 3. Este último caso implica considerar múltiples cargas.

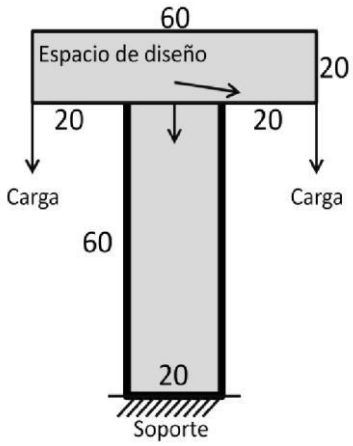


Figura 3: Espacio de diseño para dos cargas.

REPORTE

1. Realizar los ejercicios tanto de una sola carga como el de múltiples cargas.

Ejercicio 1.

Modificación en el código:

Declaración del vacío en la figura:

```
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
```

Declaración de fuerza:

```
F(40,1) = -1;
```

Ejercicio 2.

Modificaciones en el código:

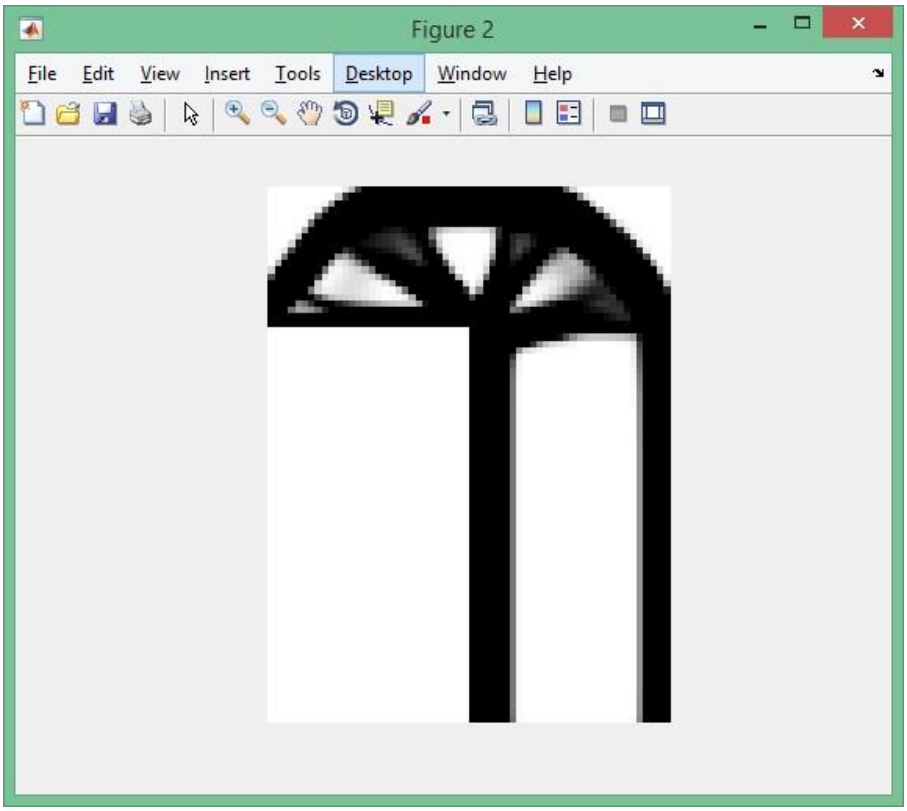
Declaración de vacío de la figura T:

```
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
```

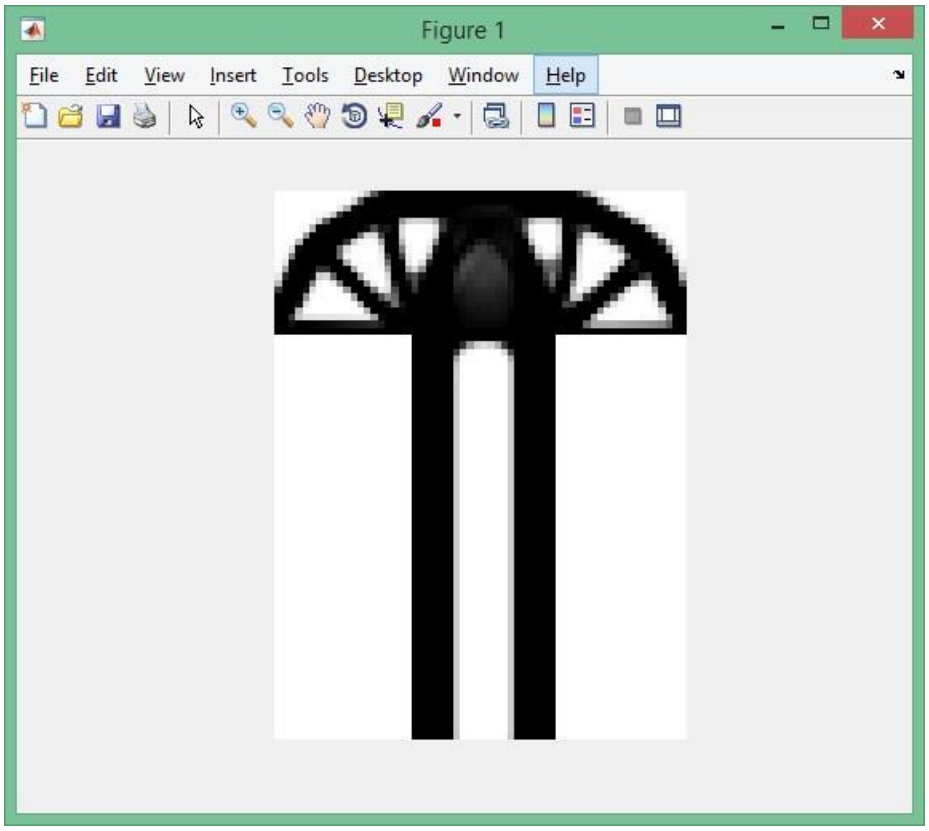
Declaracion de fuerzas:

```
F(40,1) = -1.; F(9760,2)=1.;
```

Resultado de la optimización de la figura 1:



Resultado optimización de la figura 2:



Código completo con las modificaciones finales

Figura 1

```
%%%% 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<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

% 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)];
```

Figura 2

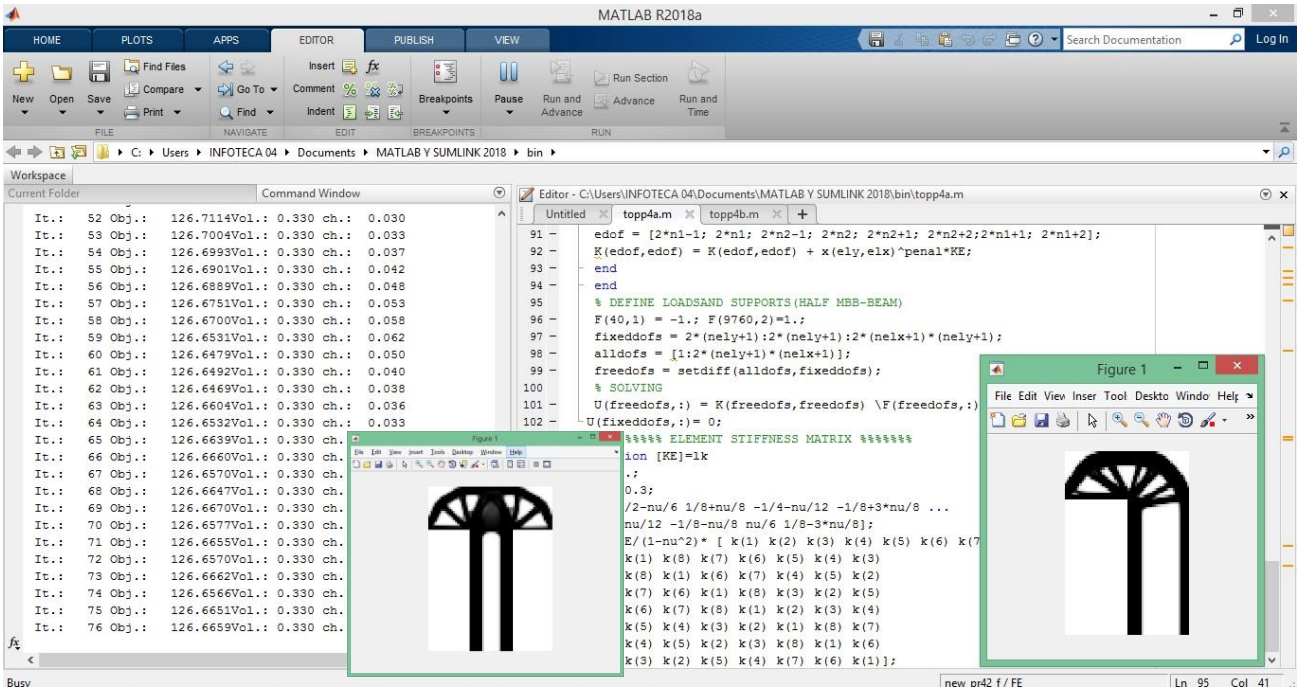
```
%%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND,
OCTOBER 1999 %%% function new_pr42_f(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
 x(1:nely,1:nelx) = volfrac; for ely =
1:nely    for elx = 1:nelx       if ely>21
if elx<21                passive(ely,elx) =
1;                           elseif  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 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)];
```

Impresión de pantalla



Conclusiones:

Juan Reyna
En esta práctica seguimos utilizando Matlab para modificar y optimizar algún elemento, en este caso el cable de un teleférico, también podemos ver cómo afectaría dicha modificación y si sería practico/seguro o no de implementarlo, porque no toda modificación es beneficiosa y aquí debemos de elegir una que no comprometa la integridad del cable.

Alexis García
Concluyo 1ue en esta práctica se uso un procedimiendo diferente para obtener los cálculos adecuados para poder asi realizarla y saber si el cable se complemete a la seguridad del teleferico y asi se pueda usar este como referencia.

Severo Arreola
En esta práctica se hizo uso de la aplicación MatLab, en el cual se tenía un planteamiento del problema de un teleférico y se buscaba la modificación y optimizacion del elemento, en el cual se quería analizar si sería deamera práctica y a su vez segura, en dónde aquí debemos escoger la que no sea de riesgo.

Daniel Zamarripa
En este reporte de laboratorio mostramos lo que se realizó a través de la práctica, esto usando Matlab, observamos que el tiempo para la realización de ésta fue mayor a la anteriores por el proceso que tuvo que llevar el software para optimizar los esfuerzos, además de ver los espacios en blanco que son elementos pasivos que necesitan ser tomados en cuenta para el diagrama.

Miguel Morales
Se puede observar que en los resultados de los casos propuestos se tiene una geométrica muy similar entre ellos. En el caso de dos cargas, este como las fuerzas son aplicadas en opuestos simétricos la forma de pieza es simetrica en el eje Y. Empezando con lo que hicimos podemos concluir que aunque se crea que algo no se toma en cuenta dentro de un sistema de esfuerzos por ser un espacio en blanco, esto no debe ser así, debemos darle la importancia para el diseño óptimo del diagrama.