UANL
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FIME
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

Laboratorio de Biomecánica

Práctica 4: Diseño de la estructura de un teleférico

INSTRUCTOR@: YADIRA MORENO VERA

MARTES V1 BRIGADA 204

Equipo #2:

Matricula	Nombre	Carrera
1991908	Covarrubias Becerril Brian Eduardo	IMTC
1991966	Luis Javier Rodríguez Vizcarra	IMTC
1925324	Rubén Cantú Espinoza	IMTC
1926098	Ángel Fernando Mexquitic Rodríguez	IMTC
1895460	Elmer Javier Delgadillo García	IMTC
1847932	Francisco Javier Velazco Rivas	IMTC

18 de octubre del 2022, San Nicolás de los Garza, Nuevo León

Objetivo

El estudiante deberá utilizar múltiples cargas y en base a estas realizar un estudio que tome múltiples cargas y que tome en consideración cuales son las implicaciones que esto conlleva.

Nombre y definición de la forma geometría

El teleférico transporta a personas o bienes en vehículos (cabinas, sillas o perchas) colgados en un cable por medio de torres/soportes de una estación a la otra. Los pasajeros pueden subir y bajar en las estaciones. El motor es eléctrico y está situado en una de las estaciones. Los componentes esenciales de un teleférico son: vehículos, cable, motor, estaciones y torres.

Hay varios sistemas de transporte por cable. Para áreas urbanas se recomiendan los teleféricos monocable (teleféricos desembragables) con cabinas para máx. 10 personas, teleféricos vaivén con cabinas para hasta 230 personas, teleféricos tricables con cabinas para hasta 34 personas, funiculares o el Cable Liner.

Los sistemas de teleférico requieren como mínimo dos estaciones. Una estación se construye en el punto de inicio del teleférico y el otro en su punto final. En cuanto a los aspectos técnicos de un teleférico, el motor, los frenos de servicio o el panel de mando no se sitúan en el vehículo sino directamente en la estación.

Los teleféricos para transporte de materiales de Doppelmayr/Garaventa se utilizan a escala mundial en los ámbitos de aplicación más diversos. El sistema de teleférico adecuado se selecciona en función de los requisitos concretos del cliente y se adapta perfectamente sus necesidades. Gracias a su flexibilidad, algunos sistemas de teleférico, como los de tipo vaivén y los funiculares, son especialmente adecuados para el transporte de materiales. Además de transportar pasajeros de un lugar a otro, también pueden trasladar palés, maquinaria diversa, bultos o mercancía a granel. Ya sea en terrenos montañosos o llanos, los teleféricos de transporte de materiales convencen en cualquier tipo de pendiente. Los teleféricos de transporte de materiales de Doppelmayr/Garaventa son el medio de transporte adecuado, en especial en terrenos accidentados. Los barrancos, valles, ríos, lagos, carreteras o bosques no suponen un obstáculo para el teleférico de transporte de materiales.

El teleférico de transportes de materiales, en resumen

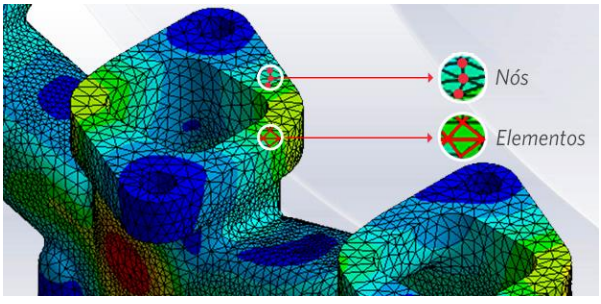
- Cargas individuales de hasta 40 t
- Capacidad de transporte de hasta 1.500 t/h
- Inclinación del transporte de hasta 45°
- Tramos de transporte de hasta 10 km
- Transporte de bultos, mercancía a granel y personas

Los componentes esenciales de un teleférico son: vehículos, cable, motor, estaciones y torres. Detalle: Vehículo: sirve para transportar a los pasajeros. Cable: sirve para soportar y mover los vehículos. Motor: sirve para mover el cable y, por lo tanto, los vehículos.Jul 8, 2020

Estado del arte

El artículo proporcionado por el instructor de laboratorio presenta una implementación de un código de implementación topológica para la minimización de la conformidad de estructuras cargadas estáticamente. El número total de líneas de entrada de Matlab es de 99, incluyendo el optimizador y la subrutina de elementos finitos. Las 99 líneas están divididas en 36 líneas para el programa principal, 12 líneas para el optimizador basado en criterios de optimización, 16 líneas para un filtro de dependencia de malla y, por último, 35 líneas para el código de elementos finitos.

Como paréntesis, el método de elementos finitos consiste en proponer que un número infinito de variables desconocidas, sean sustituidas por un número limitado de elementos de comportamiento bien definido. Esas divisiones pueden tener diferentes formas, tales como triangular, cuadrangular, entre otros, dependiendo del tipo y tamaño del problema. Como el número de **elementos** es limitado, son llamados de “elementos finitos” – palabra que da nombre al método. Los elementos finitos están conectados entre sí por puntos, que se llaman **nodos** o **puntos nodales**. Al conjunto de todos estos ítems – elementos y nodos – se lo denomina **malla**. Debido a las subdivisiones de la geometría, las ecuaciones matemáticas que rigen el comportamiento físico no se resolverán de una manera exacta, sino aproximada por este método numérico. La precisión de los Métodos dos Elementos Finitos depende de la cantidad de nodos y elementos, del tamaño y de los tipos de elementos de la malla. Por lo tanto, cuanto menor sea el tamaño y mayor el número de elementos en una malla, más precisos serán los **resultados del análisis.**



El código Matlab presentado en la página https://www.topopt.mek.dtu.dk/apps-and-software/a-99-line-topology-optimization-code-written-in-matlab está destinado a la

educación en ingeniería. Los estudiantes y los recién llegados al campo de la optimización topológica pueden encontrar el código aquí y descargarlo. El código se puede utilizar en cursos de optimización estructural en los que se puede asignar a los estudiantes la realización de extensiones, como casos de carga múltiples, esquemas alternativos de independencia de malla, áreas pasivas, etc. Esta versión del código está optimizada con respecto a la velocidad en comparación con la versión original y fue lanzada el 28 de mayo de 2002.

Propuesta de diseño de la geometría, alcances y limitaciones

Se parte con la idea de un teleférico el cual analizaremos mediante el siguiente código, que modificaremos con la idea de analizar la resistencia del mismo transporte. El análisis que llevaremos a cabo lo haremos en el simulador matlab ya que también lo intentamos en scilab y creemos que en matlab es mucho más fácil pero lo hicimos para comprobar mediante las capturas o el código en sí podremos ver desarrollado este análisis y podremos sacar las mismas conclusiones.

Al cuidador del teleférico 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 caso implica considerar múltiples cargas.

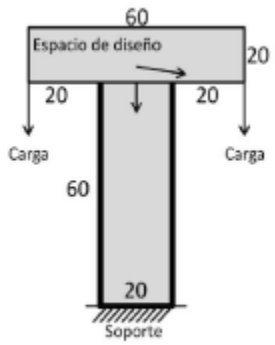
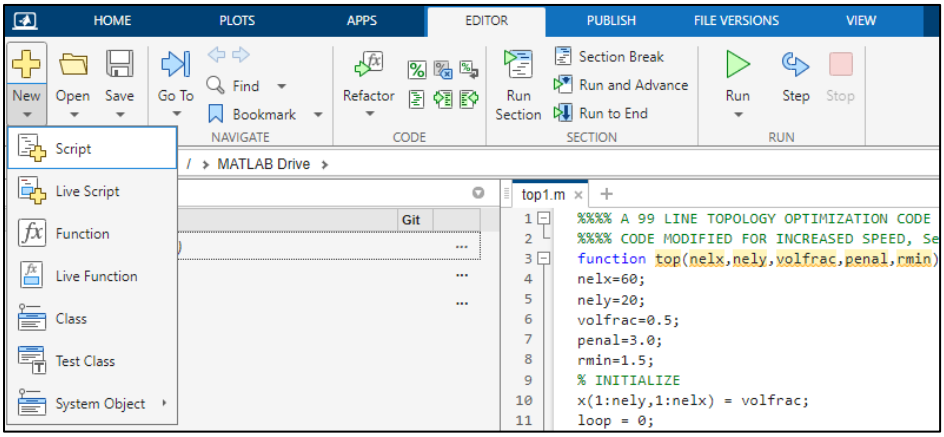


Figura 3: Espacio de diseño para dos cargas.

Pasos del desarrollo de la programación

Para simular en Matlab el código propuesto de optimización topológica, debemos llevar a cabo los siguientes pasos:

1) Abrir Matlab, ya sea la aplicación de escritorio o directamente desde la página oficial usando su repositorio online.
2) Crear un nuevo script, para ello seleccionamos la opción en la barra de herramientas.



3) Una vez hecho esto, solo tenemos que copiar el código proporcionado que se localiza en la página indicada por el instructor.

4) Ya con el código copiado (99 Line Topology Optimization Code), tenemos que guardar primero el documento, en este caso como se utilizó Matlab online, se guardará en la nube de la cuenta que esta iniciada.
5) Antes de ejecutar la simulación, es necesario modificar ciertas líneas del código, las cuales se mostrarán a continuación:

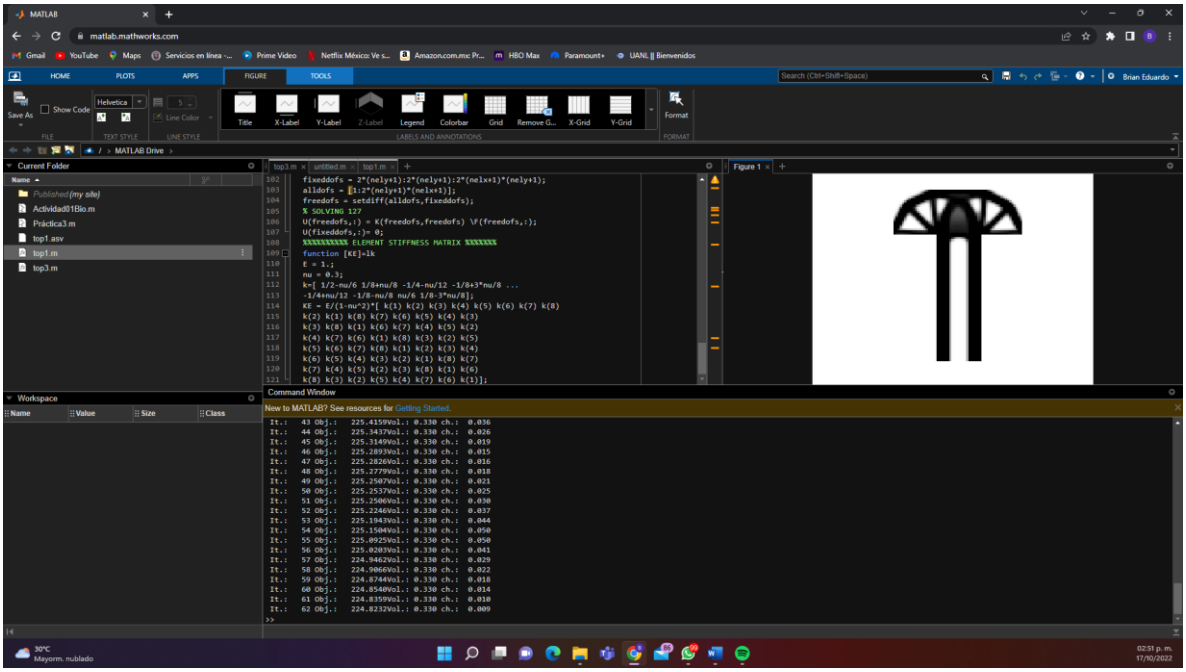
```
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
for ely = 1:nely
    for elx = 1:nelx
        if ely>21
            if elx<21
                passive(ely,elx) = 1;
            elseif elx>41
                passive(ely,elx)=1;
            else
                passive(ely,elx) = 0;
            end
        end
    end
end
x(find(passive))=0.001;
loop = 0; change = 1.;
```

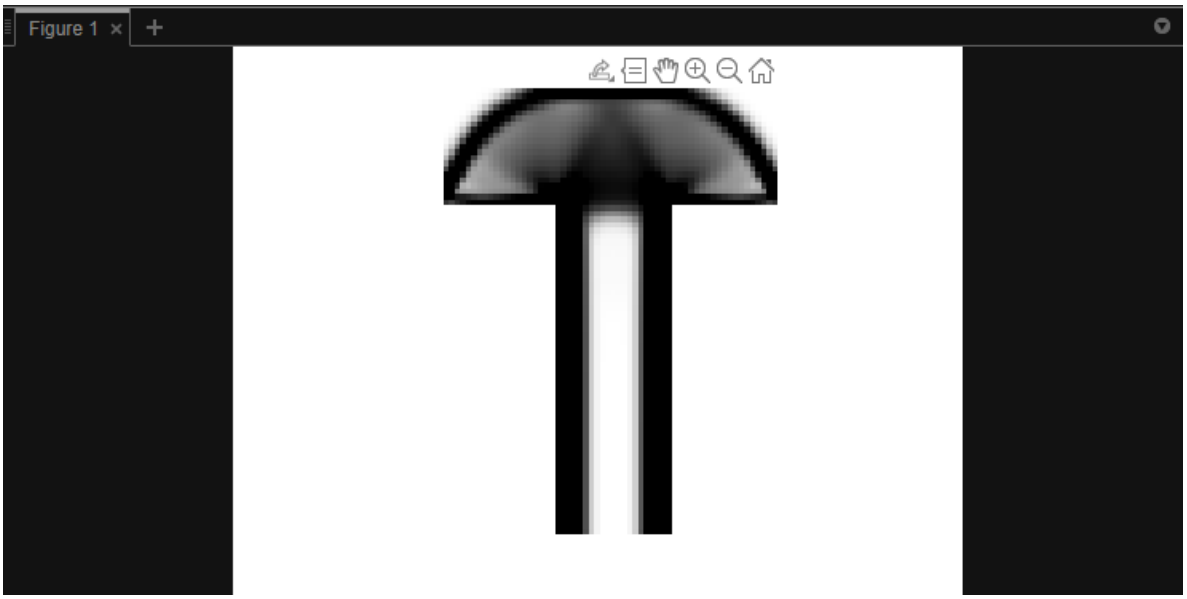
```
% START ITERATION
while change > 0.01
loop = loop + 1;
xold = x;
```

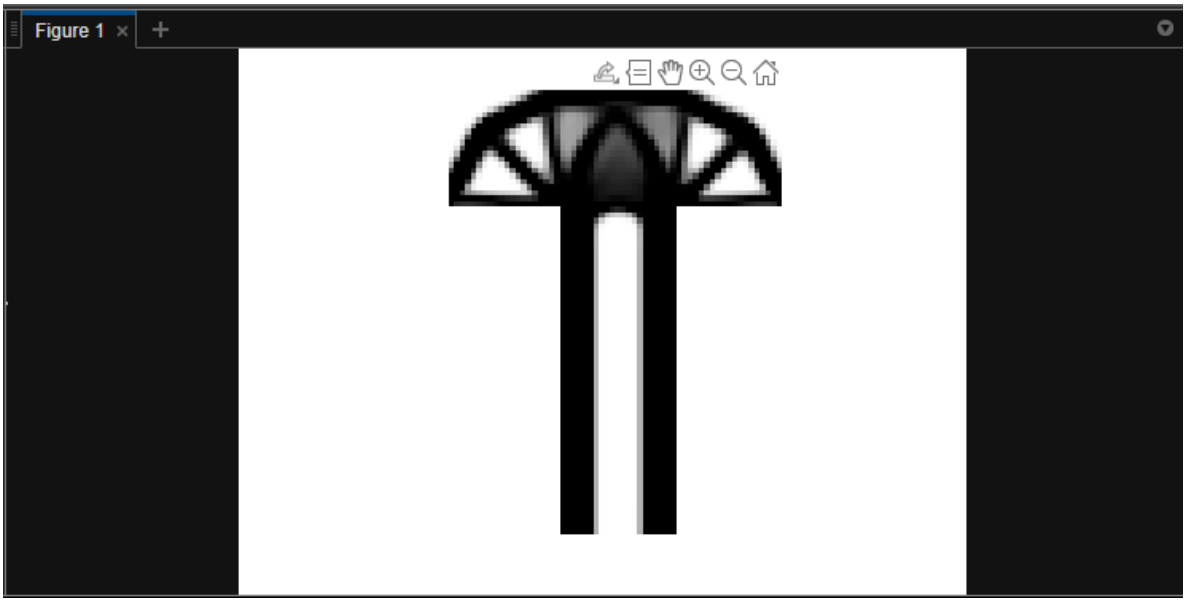
```
%29 PRINT RESULTS
change = max(max(abs(x-xold)));
disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
'Vol.:' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
' ch.: ' sprintf('%6.3f',change )])
% PLOT DENSITIES
colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-6);
end
%40 %%%%%%%%%% 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
```

```
% 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);
```

Resultados de la optimización







Código utilizado

```
%%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
function top3(nelx,nely,volfrac,penal,rmin);
nelx=60;
nely=80;
volfrac=0.33;
penal=3.0;
rmin=1.5;
% 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
%25 FILTERING OF SENSITIVITIES
[dc] = check(nelx,nely,rmin,x,dc);
%27 DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
[x] = OC(nelx,nely,x,volfrac,dc,passive);
%29 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
%40 %%%%%%%%% 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
%65 %%%%%%%%% 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),5); U =zeros(2*(nely+1)*(nelx+1),5);
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 127
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)];
```

Conclusiones

- **Covarrubias Becerril Brian Eduardo:** para esta práctica hicimos uso nuevamente del código de optimización topológica proporcionado por el profesor, con la ligera diferencia de que en el caso que se analizó se requirió de mayor tiempo para la realización debido a la optimización de esfuerzos, además de ver los espacios en blanco que fueron tomados en cuenta para el diagrama. El caso que se analizó fue donde a un cuidador del teleférico le gustaría que se hicieran mejoras para que la estructura pueda llevar dos teleféricos a la vez, este caso implicó considerar múltiples cargas.

- **Luis Javier Rodriguez Vizcarra:** En esta practica en lo que se trabajo fue en un teleférico, optimizándolo para que pueda satisfacer la necesidad del cuidador de estos, ya que se busca que en vez de un teleférico estén dos a la vez, para esto lo que se tiene que hacer es minimizar el peso del teleférico pero sin perder su resistencia mecánica, o sea haciendo uso de la optimización topológica, esto se realizo mediante el software Matlab, parecido a las practicas anteriormente realizadas, se cumplió el objetivo al lograr optimizar el teleférico de una manera correcta.

- **Francisco Javier Velazco Rivas:** En esta práctica se consultó el hecho de que buscamos optimizar además de analizar el movimiento de un teleférico lo hemos visto en prácticas anteriores pero en ésta nos basamos y nos representamos en este caso las bandas o las cuerdas que este mismo usan con el objetivo de buscar, resultados o las mismas fuerzas que éstas se ejercen sobre las cuerdas al tener el programa o un planteamiento similar nos resultó no tan complicado podré llevar a cabo la modificación de este mismo punto final.

- **Elmer Javier Delgadillo Garcia:** En base a lo observado en la presente práctica, se puede destacar el procedimiento del diseño de la estructura de un teleférico a través del software de Matlab, que si bien también se pudo haber desarrollado en scilab, la visualización del análisis de resistencias y la respectiva modificación de la estructura para el soporte de dos teleféricos puede apreciarse de mejor manera en el software que ya hemos manejado con anterioridad.

- **Angel Fernando Mexquitic Rodriguez:** Con esta practica se realizo un teleférico en el cual se optimizo para mejorar su funcionamiento, esto realizado mediante Matlab. Optimizando sus esfuerzos y mejorándolo para llevar dos teleféricos a la vez. Al ser parecido a las realizadas con anterioridad, fue bastante sencillo de realizar y se logro realizar el objetivo propuesto.

- **Rubén Cantú Espinoza:** para la elaboración de esta práctica se llevó a cabo el mismo procedimiento que se aplicó en la práctica pasada, primeramente se propusieron diversos análisis (teniendo en cuenta que el diseño viene siendo el de un teleférico), con el objetivo de optimizar los esfuerzos que recibe la pieza. Para esto se definió la forma geométrica de la pieza en el software de maltab, y posteriormente se declararon las variables a tomar en cuenta, como lo es el espacio del diseño y las fuerzas aplicadas. Se continuó con el estado del arte donde se analizó la pieza para la optimización de acuerdo a su estructura y se finalizó con la implementación del código.

Referencias

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

- IMME. (2007). *Una metodología para la optimización estructural de formas usando principios de evolución flexible distribuida*. Scientific Electronic Library Online. http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0376-723X2007000100002