



**UNIVERSIDAD AUTÓNOMA DE  
NUEVO LEÓN  
FACULTAD DE INGENIERÍA  
MECÁNICA Y ELÉCTRICA**



Laboratorio de biomecánica

**Maestro: Ing. Isaac Estrada**

**Practica#5: Optimización de una parte de una  
prótesis de pie**

**Periodo: Agosto – Diciembre 2022**

Brigada: 309

Hora Clase: N5

<b>Nombre:</b>	<b>Matrícula</b>	<b>Carrera</b>
Luis Lauro García Hernández	1803392	IMTC
Víctor Cristopher Santiago Martínez	1858524	IMTC
Julio Adrián Torres Cisneros	1903681	IMTC

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

## Nombre y definición de la geometría.

En el presente trabajo se propondrá una propuesta de análisis de formas y programación para la ejecución de la optimización del diseño de una prótesis de pie para diferentes “estados o fases” dentro del ciclo de una la marcha humana.

## Estado del Arte.

Durante las últimas décadas, los investigadores han explorado y mejorado la funcionalidad del pie protésico. Los amputados disponen de una amplia gama de prótesis de pie entre las que han de optar por la más adecuada. Se han diseñado diferentes modelos para que los amputados lleven la vida cotidiana y realicen diversas actividades como caminar, correr, bailar, andar en bicicleta, jugar al golf, nadar, etc. Anteriormente se utilizaron materiales de madera y acero, que han sido reemplazados con éxito por plásticos livianos, fibra de carbono compuestos y aleaciones de metales. Los elementos del pie protésico se han diseñado de tal manera que pueden almacenar energía cuando se cargan al deformar los componentes elásticos y liberar la energía durante la última fase de apoyo, lo que proporciona un empujón del tobillo para mover el cuerpo hacia adelante. El pie protésico ha ayudado a los amputados a recuperar la movilidad y les ha hecho independientes para llevar una vida normal como las demás personas sanas. Por medio de esta sección dedicada al estado del arte, revisaremos la importancia de imitar la biomecánica del pie-tobillo humano, el pie protésico de almacenamiento y retorno de energía (ESR) y el pie protésico convencional.



## Historia

Diversos investigadores mencionan en sus escritos que los dedos artificiales descubiertos en Egipto son las partes protésicas más antiguas conocidas del mundo.

En el año 2000 en el Cairo se encontró un dedo que se remonta a los años 1069 a 664 a. C. Los arqueólogos especularon que la mujer de 50 a 60 años de la que procedía la prótesis pudo haber perdido el dedo a causa de las complicaciones por la diabetes (suceso común en la actualidad con el pie diabético). Los materiales implementados fue madera tallada y cuero para sujetar el dedo al resto del pie.

Otro ejemplo de estas piezas protésicas es un dedo elaborado con cartonaje (mezcla de papel maché hecho con lino, pegamento y yeso entintado), esta pieza se encuentra en exhibición en el museo británico de Londres, que lleva el nombre de la coleccionista que lo adquirió para el museo en 1881, se denomina Greville Chester Great Toe.

Estos son claros ejemplos de cómo la creatividad del hombre crea soluciones, en la actualidad está más que comprobado que cuando se llega a realizar una amputación del dedo gordo del pie, la estabilidad y la marcha se modifican de manera importante. A simple vista el complemento que crearon del pie gordo puede parecer insignificante, pero es una representación de la historia de las prótesis que cumple las dos características primordiales:

- Función al permitir seguir usando el calzado característico de esa época.
- Como identidad al regresar la integridad de la usuaria.

A nivel general de la cultura de Egipto no marco un cambio el faltan de una parte del cuerpo, pudo haber sido más fácil diseñar algún otro tipo de zapato. Los egipcios laboran sus extremidades protésicas de manera rudimentarias con fibras, el factor de uso primordial era para la sensación de completitud.

## **Función**

Un pie protésico se trata de un aparato médico diseñado justo a la medida de cada paciente para suplantar estética y funcionalmente la parte faltante de la extremidad a causa de una amputación.

Esta tiene la función tanto de aparentar que existe la sección faltante como de ser un punto de apoyo para que el paciente pueda caminar, utilizar calzado y desempeñar sus labores de forma normal o lo más normal posible.

Esta prótesis puede suplantar desde unos cuantos dedos (prótesis de dedos de pie) el antepié, pie completo y pie con tobillo (todo depende del grado y lugar de la amputación).

Todas las prótesis de pie cuentan con una serie de mecanismos diseñados para poder adaptarse al muñón de forma adecuada y firme, además de brindar el soporte y movimiento necesario para que el paciente pueda sostener de pie, caminar e inclusive correr y desempeñar labores o deportes de alto rendimiento.

Esta sin duda es la prótesis más básica del miembro inferior, pues todas las prótesis de pierna, independientemente del tipo de amputación y el nivel en donde se encuentre, debe contar por obvias razones con la sección del pie prostético y el tobillo.

## **Elementos**

Dependiendo del tipo de amputación, la prótesis podrá estar compuesta de diversos elementos, como lo pueden ser:

- Socket o encaje con el muñón.
- Plantillas.
- Almohadillas.
- Estabilizadores para el talón.
- Elementos estéticos.

De igual manera, existen prótesis mucho más avanzadas que pueden tener dispositivos y elementos robóticos, mecánicos, bioeléctricos y biomecánicos para otorgar movimientos especiales en caso de ser necesario.

El pie protésico puede ser fabricado de uno o más materiales, los cuales son seleccionados para brindar el soporte, agarre y diseño necesario para suplantar la parte faltante del cuerpo humano.

Por lo general se utiliza:

- Silicona médica.
- Plantillas de carbono.
- Rellenos de materiales elásticos.
- Aluminio.
- Titanio.
- Acero inoxidable.
- Plásticos.

## **Propuesta de diseño de la geometría, alcances y limitaciones**

Se propone en este caso el diseñar una pierna protésica de un costo/vida útil menor a 100,000 [pesos/año] para pacientes con amputaciones transtibiales.

Este trabajo detallará el diseño de únicamente un pie articulado y una tibia. Además, este diseño contempla una prótesis pasiva. Esto debido a que ambos diseños, pasivo y activo, son inherentemente distintos en cuanto a rangos de movimiento y cargas aplicadas.

## **Pasos del desarrollo de la programación**

En esta sección se detalla la metodología empleada para la elaboración de la prótesis de pierna.

#### 4.1. Escaneo de pierna

Debido a que se pretende realizar una optimización de estructura de vigas para diseñar la tibia, se le debe ingresar al algoritmo una base de la que extraer puntos como nodos de los elementos y construir así una estructura. Dicha base viene en forma de una matriz de nodos posibles que están uniformemente distribuidos dentro de una geometría tridimensional con forma de pantorrilla. La base para la creación de la matriz de nodos es el obtener una nube de puntos que describa la geometría de una pantorrilla. Con este objetivo es que se realiza el escaneo tridimensional de una pierna.

El escáner en cuestión ocupa el método de luz estructurada. Así se genera una nube de puntos tridimensional que describe la forma exterior del objeto. El método de luz estructurada funciona con un proyector y una cámara: El proyector emite patrones de luz sobre el objeto de interés y según cómo la luz sea reflejada por el objeto (debido a su forma), la cámara interpreta los contornos del objeto con base en la distorsión del patrón de luz, la siguiente Figura muestra un esquema de funcionamiento de escaneo por luz estructurada.

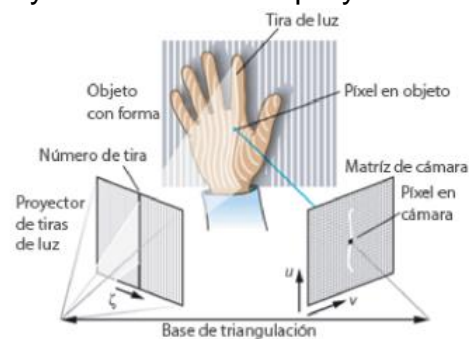
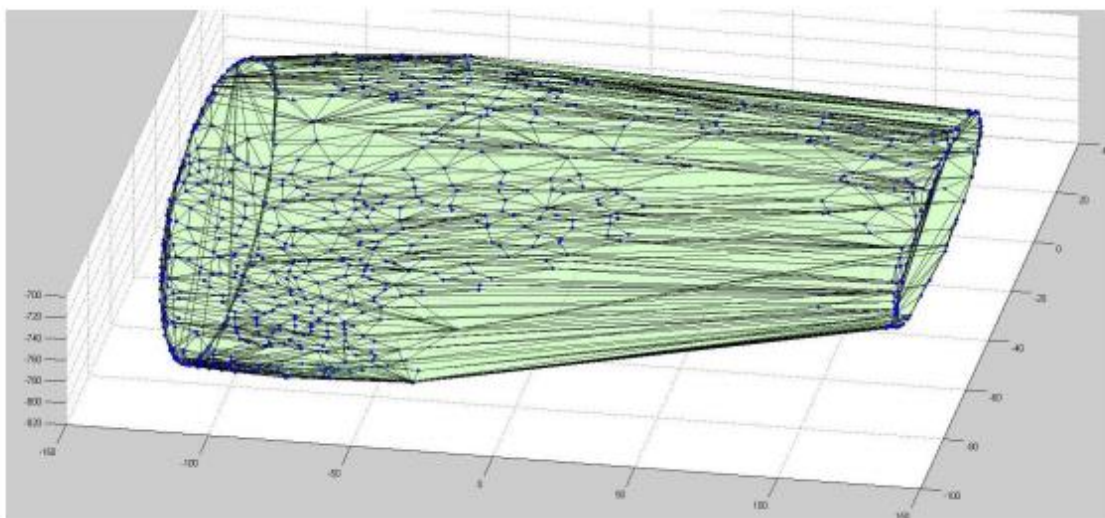


Figura 1.- Escaneo por luz estructurada.

#### Limpieza y calce de nube de puntos.

Debido a que el escáner obtiene sus datos mediante luz óptica, éste devuelve como resultado la escena completa que ha escaneado. Por lo anterior, primero se deben excluir aquellos puntos que no sean de interés para el estudio. Este proceso se hace con el software CloudCompare™. Posteriormente los puntos de todas las escenas deben ser calzados en un único espacio tridimensional. El resultado de dicho proceso no está perfectamente nivelado, tema que debe ser tratado antes de generar los puntos finales para la estructura.

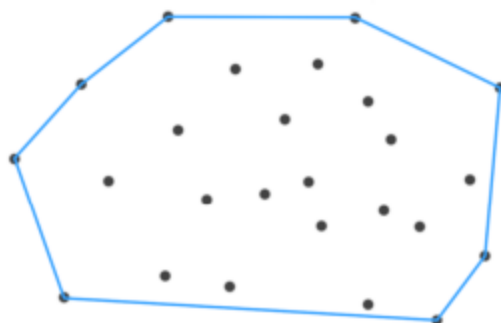
Para obtener un poliedro convexo a partir de una base de puntos en el espacio se genera una cobertura o casco convexo. Una cobertura es el poliedro convexo más grande que se puede construir basado en los puntos provistos al algoritmo. El uso de una cobertura resulta ser muy conveniente, ya que evita el problema del margen de error al tomar los puntos más exteriores y también genera un poliedro convexo. La Figura de adelante esquematiza cómo funciona la generación de una cobertura en un espacio 2D.



*Figura 2.- Casco convexo obtenido desde la nube de puntos.*

En el caso de la geometría de interés (la superficie escaneada) se genera un algoritmo, ubicado en el Anexo A, que pueda extraer los puntos de la nube que pertenecen al casco convexo, como se puede ver en la Figura 3-6 que muestra el resultado de dicho proceso.

El código que hace esta tarea se encuentra en el Anexo A.1



*Figura 3.- Esquema de funcionamiento de la cobertura.*

```

function [extPoints] = Cloud2CHVertex( PointCloud )
%-----
%This function extracts the convex hull from a Point Cloud
%and returns the vertexes of the convex hull as to clean up the point cloud
%-----
%Get the convex hull
T=convhulln(PointCloud);
%Get the vertex indexes for the triangles without repetitions
extIndex=unique([T(:,1);T(:,2);T(:,3)]);
%Extract coordinates from the original Point Cloud
extPoints=PointCloud(extIndex,:);
%Extract the coordinates of the interior points by removing the exterior
%ones
intPoints=setdiff(PointCloud,extPoints,'rows');
%Plot mesh, vertexes and interior points for graphic confirmation
figure
trisurf(T,PointCloud(:,1),PointCloud(:,2),PointCloud(:,3),1,'faceAlpha',0.2)
hold on
%scatter3(intPoints(:,1),intPoints(:,2),intPoints(:,3),25,'r','filled')
scatter3(extPoints(:,1),extPoints(:,2),extPoints(:,3),25,'b','filled')

end

function [ inpoints ] = generateInsidePoints( extPoints,numPoints )
%-----
%This function gets the external vertexes that comprise the convex hull of
%the point Cloud and returns numPoints internal points that reside inside
%the geometry
%-----
%Create a Delaunay triangulation for checking afterwards
DT=delaunayTriangulation(extPoints);
%Define geometric bounding box for geometry
lbx=min(extPoints(:,1));
ubx=max(extPoints(:,1));
lby=min(extPoints(:,2));
uby=max(extPoints(:,2));
lbz=min(extPoints(:,3));
ubz=max(extPoints(:,3));
%Generate proposed points
testPointsx=rand(numPoints,1)*(ubx-lbx)+lbx;
testPointsy=rand(numPoints,1)*(uby-lby)+lby;
testPointsz=rand(numPoints,1)*(ubz-lbz)+lbz;
testPoints=[testPointsx testPointsy testPointsz];
i=1;
%Regenerate points if they don't lie within the geometry
while i<=numel(testPointsx)
    inCheck=pointLocation(DT,testPoints(i,:));
    if isnan(inCheck)==1
        testPoints(i,:)=[rand*(ubx-lbx)+lbx rand*(uby-lby)+lby ...
            rand*(ubz-lbz)+lbz];
    else
        i=i+1;
    end
end
%Return required points
inpoints=testPoints;
end

end

```

## Obtención de matriz de nodos.

Una vez hecho todo lo anterior se pueden generar puntos aleatorios en el espacio que comprende la geometría, pudiendo ser chequeada la pertenencia del punto en cuestión dentro de la geometría. Esto es realizado con el código en el anexo A.2. Una vez generados los puntos interiores, estos son ordenados según la dirección de mayor dimensión de la geometría, permitiendo la rápida identificación de los

puntos superiores e inferiores. Esto permite forzar posteriormente las condiciones de borde de manera más fácil. El poder identificar los puntos superiores e inferiores también permite forzar la pertenencia de ellos en la geometría final, restringiendo el espacio de búsqueda a aquellas geometrías que tengan un largo en particular. Ya hecho este proceso, se tiene el espacio de búsqueda sobre el que el algoritmo genético busca optimizar la estructura.

```
TDScan=load('TreatedCloudm');
global extpoints
extpoints=TDScan.TreatedCloudm;
global intpoints
nintpoints=1000;
intpoints=generateInsidePoints(extpoints,nintpoints);
global NodesU
NodesU=sortrows([extpoints;intpoints],3);

% global DT
% DT=delaunayTriangulation([extpoints;intpoints]);
% global NodesU
% NodesU=DT.Points;
% global EdgesU
% EdgesU=edges(DT);

NIND = 1500;          % Number of individuals per subpopulations
MAXGEN = 250;         % maximum Number of generations
CGAP = .9;           % Generation gap, how many new individuals are created
NVAR = size(NodesU,1)-50;
PRECI = 1;           % Precision of binary representation

% Build field descriptor, not needed in this case
% FieldD = [rep([PRECI],[1, NVAR])); rep([-512;512],[1, NVAR]);...
%          rep([1; 0; 1; 1], [1, NVAR])];

% Initialise population
Chrom = crthp(NIND, NVAR*PRECI);

% Reset counters
Best = NaN*ones(MAXGEN,1); % best in current population
BestW = NaN*ones(MAXGEN,1); %weights of bests
BestD = NaN*ones(MAXGEN,1); %Displacements of bests
global gen
gen = 0; % generational counter
% Evaluate initial population
[ObjV,weight,meandisp] = evaluate(Chrom);
```

## Acoplamiento de método de elementos finitos y AG

Para hacer la optimización de la estructura se usa un algoritmo genético con una representación cromosómica binaria, es decir, que el material genético de cada individuo sólo puede tener dos valores: cero y uno. En este caso un 0 significa que el punto al que corresponde ese gen no es utilizado, mientras que un 1 significa que el punto es utilizado en la construcción de la estructura. El script que realiza esta tarea se encuentra en el anexo A.3.



---

```

% Track best individual and display convergence

[Best(gen+1),I] = min(ObjV);
%   BestW(gen+1)=weight(I);
%   BestD(gen+1)=meandisp(I);
plot(log10(Best),'ro');xlabel('generation'); ylabel('log10(f(x))');
text(0.5,0.95,['Best = ', num2str(Best(gen+1))],'Units','normalized');
drawnow;

% Generational loop
while gen < MAXGEN,
gen
    % Assign fitness-value to entire population
    FitnV = ranking(ObjV);

    % Select individuals for breeding
    SelCh = select('sus', Chrom, FitnV, GGAP);

    % Recombine selected individuals (crossover)
    SelCh = recomb('xovsp',SelCh,0.7);

    % Perform mutation on offspring
    SelCh = mut(SelCh);

    % Evaluate offspring, call objective function
    [ObjVSel,weight,meandisp] = evaluate(SelCh);

    % Reinsert offspring into current population
    [Chrom, ObjV]=reins(Chrom,SelCh,1,1,ObjV,ObjVSel);

    % Increment generational counter
    gen = gen+1;
    [Best(gen+1),I] = min(ObjV);
    % Update display and record current best individual
    %   if min(ObjV)<=min(ObjVSel)%if offspring is not better than best parent
    %       Best(gen+1) = Best(gen);
    %       BestW(gen+1)= BestW(gen);
    %       BestD(gen+1)= BestD(gen);
    %   else
    %       [Best(gen+1),I] = min(ObjV);
    %       BestW(gen+1)=weight(I);
    %       BestD(gen+1)=meandisp(I);
    %   end
    plot(log10(Best),'ro'); xlabel('generation'); ylabel('log10(f(x))');
    text(0.5,0.95,['Best = ', num2str(Best(gen+1))],'Units','normalized');
    drawnow;
end

```

Para la definición de los elementos se aplica una triangulación de Delaunay sobre los puntos cuyos valores respectivos en el cromosoma sean 1. La triangulación de Delaunay se caracteriza por que, en el caso 2-D, sin importar el triángulo escogido, la circunferencia que lo circunscribe no contiene ningún nodo que no sean los tres que componen el triángulo en cuestión. El uso de triangulaciones de Delaunay, como medio para generarlos elementos de una estructura, ha sido probado útil en la optimización de estructuras simples. Se escoge porque evita tres fenómenos altamente indeseados que causan errores en la evaluación de la estructura por elementos finitos

Un algoritmo genético requiere la existencia de una función objetivo de la que obtener un valor de aptitud. En este caso específico se interviene la función de aptitud para guiar a los individuos en espacio de búsqueda hacia el resultado de

interés. La evaluación se hace con el código encontrado en el anexo A.4. Como función objetivo a optimizar se ponderan por igual importancia el peso total de la estructura como la deformación promedio de la estructura.

```
function [stiffness]=...
formStiffness3Dframe(GDof,numberElements,...
elementNodes,numberNodes,nodeCoordinates,E,A,Iz,Iy,G,J)
stiffness=zeros(GDof);
% computation of the system stiffness matrix
for e=1:numberElements;
% elementDof: element degrees of freedom (Dof)
indice=elementNodes(e,:);
elementDof=[6*indice(1)-5 6*indice(1)-4 6*indice(1)-3 ...
6*indice(1)-2 6*indice(1)-1 6*indice(1)...
6*indice(2)-5 6*indice(2)-4 6*indice(2)-3 ...
6*indice(2)-2 6*indice(2)-1 6*indice(2)];
x1=nodeCoordinates(indice(1),1);
y1=nodeCoordinates(indice(1),2);
z1=nodeCoordinates(indice(1),3);
x2=nodeCoordinates(indice(2),1);
y2=nodeCoordinates(indice(2),2);
z2=nodeCoordinates(indice(2),3);
L = sqrt((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1) +...
(z2-z1)*(z2-z1));
k1 = E*A/L;
k2 = 12*E*Iz/(L*L*L);
k3 = 6*E*Iz/(L*L);
k4 = 4*E*Iz/L;
k5 = 2*E*Iz/L;
k6 = 12*E*Iy/(L*L*L);
k7 = 6*E*Iy/(L*L);
k8 = 4*E*Iy/L;
k9 = 2*E*Iy/L;
k10 = G*J/L;
a=[k1 0 0; 0 k2 0; 0 0 k6];
b=[ 0 0 0;0 0 k3; 0 -k7 0];
c=[k10 0 0;0 k8 0; 0 0 k4];
d=[-k10 0 0;0 k9 0;0 0 k5];
k = [a b -a b;b' c {-b}' d; {-a}' -b a -b;b' d' {-b}' c];
```

```

xx=nodeCoordinates(:,1);

yy=nodeCoordinates(:,2);
zz=nodeCoordinates(:,3);
elementNodes=edges(DT);
elementNodes([13,15,17,255,290,300,560,602,639],:)=[];
numberNodes=size(nodeCoordinates,1);
numberElements=size(elementNodes,1);
% for structure:
% displacements: displacement vector
% force : force vector
% stiffness: stiffness matrix
% GDof: global number of degrees of freedom
GDof=6*numberNodes;
U=zeros(GDof,1);
stress=zeros(numberElements);
force=zeros(GDof,1);
stiffness=zeros(GDof);
%force vector
forceI=[1:30]*5-3;
force(forceI)=24/1000;%kN
% calculation of the system stiffness matrix
% and force vector
% stiffness matrix
stiffness=...
formStiffness3Dframe(GDof,numberElements,...
elementNodes,numberNodes,nodeCoordinates,E,A,Iz,Iy,G,J);
% boundary conditions and solution
prescribedDof=((size(usedN,1)-20)*6+1:size(usedN,1)*6);
% solution
displacements=solution(GDof,prescribedDof,stiffness,force);
[stress,Le]=getstresses(displacements,numberElements,elementNodes,...
nodeCoordinates,E,Iy,Iz);
auxvec=zeros(3,1);
save('stress','stress','displacements');
[maxistress,pointer1]=max(stress)
[ministress,pointer2]=min(stress)
%extract the nodes movement for averaging as objective function
for e=1:(numberNodes-1)
auxvec=vertcat(auxvec,[e:e:e]);
end
dstripper=rep([1:2:3],[numberNodes,1]);
dstripper=dstripper+(6*auxvec);
weight(i)=sum(Le)*density*pi*r^2/1000000;
meandisp(i)=mean(displacements(dstripper));

ObjV(i)=0.4*weight(i)+0.6*100*meandisp(i);
m=m+1;

for e=1:numberElements

    if stress(e)>=ystressT*0.25 || stress(e)<=-ystressC*0.25
        m=m+1;
    end
    if stress(e)<=-0.25*E*Iz*(pi/Le(e))^2
        m=m+1;
    end
    if stress(e)>=0.5*18.9
        m=m+1;
    end
end
pfactor=0;
%10^9-(3*numberElements-m)*10^9/(3*numberElements)
%ObjV(i)=ObjV(i)+(0.5*gen)^2*pfactor;
% displacements
% disp('Displacements')
% jj=1:GDof; format long
% f=[jj;displacements];
% fprintf('node U\n')
% fprintf('%3d %12.8f\n',f)
% %drawing mesh and deformed shape
% U=displacements;
% clf
% drawingMesh(nodeCoordinates+500*[U(1:6:6*numberNodes)...
% U(2:6:6*numberNodes) U(3:6:6*numberNodes)],...
% elementNodes,'L2','k-');
% drawingMesh(nodeCoordinates,elementNodes,'L2','k--');
end
end

```

Es importante notar que las capacidades que tiene el algoritmo para optimizar están limitadas al espacio de búsqueda que se le provee. En el caso de la formulación del presente problema ese espacio es conformado por los nodos que puede utilizar la estructura para hacer la triangulación.

### Diseño de pie según modelo de forma de rodadura

El diseño del pie, segundo componente crítico, supone el modelamiento inicial del pie, en conjunto con su articulación que permita las rotaciones definidas anteriormente. Con esto en mente, se recurre al uso de la forma de rodadura del pie, así como el costo y facilidad de construcción como guías principales de diseño.

Para diseñar de manera preliminar la forma del pie, se asume que está hecho de varios perfiles rectangulares superpuestos, así como se muestra en la Figura siguiente, que es la situación donde el pie está hacia el final de un paso. En esta situación se tiene que sólo la primera capa es la que está en contacto con el suelo. Se toma en cuenta que las capas que van arriba de la primera son más gruesas, por lo que se asume que la primera capa está empotrada a una distancia igual a la que empieza la segunda, midiendo desde la punta del pie

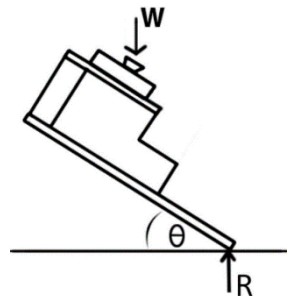


Figura 4.- Diagrama de fuerzas para pie.

Es importante precisar que este método no da un modelo definitivo, ya que la forma real que necesita adoptar el pie no es cuadrada, sino es más bien similar a la forma ilustrada en la Figura de adelante.

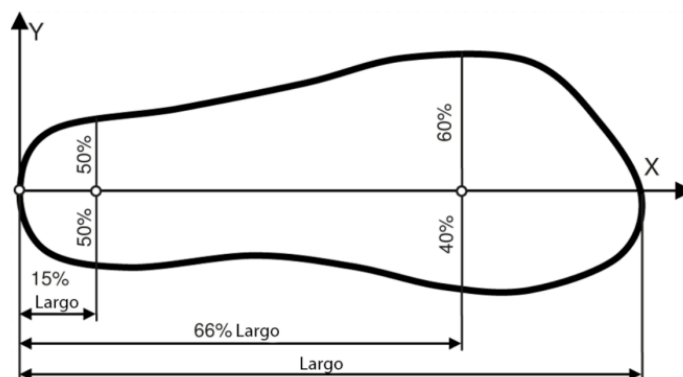


Figura 1.- Forma de planta de referencia.

## Análisis de piezas en software de elementos finitos

Una vez diseñadas las piezas del pie, se simulan sus comportamientos como ensamblaje bajo las cargas de uso normal. Lo anterior tiene por objetivo cerciorar que los diseños son aptos en una segunda aproximación para las condiciones de borde impuestas. Los softwares proveen facilidades que permiten aproximar mejor algunos valores como los esfuerzos y deformaciones, dado un perfil de esfuerzo aplicado, permite obtener modos normales y respuesta estática a carga. En el caso específico del análisis de vida a fatiga, se han obtenido las curvas S-N de fatiga para los polímeros considerados para las piezas. Debido a que el software no presenta una opción de uso de curvas S-Personalizadas para cada material, se hace una evaluación fuera de software, basado en las curvas ya mencionadas y los esfuerzos calculados.

Un método que ayuda a la convergencia en general del sistema es permitirle al software modificarla rigidez del contacto en cada iteración, en un intento de encontrar buena convergencia. Lo anterior puede ser hecho de manera normal, variando un poco, o de manera agresiva, permitiendo amplias variaciones.

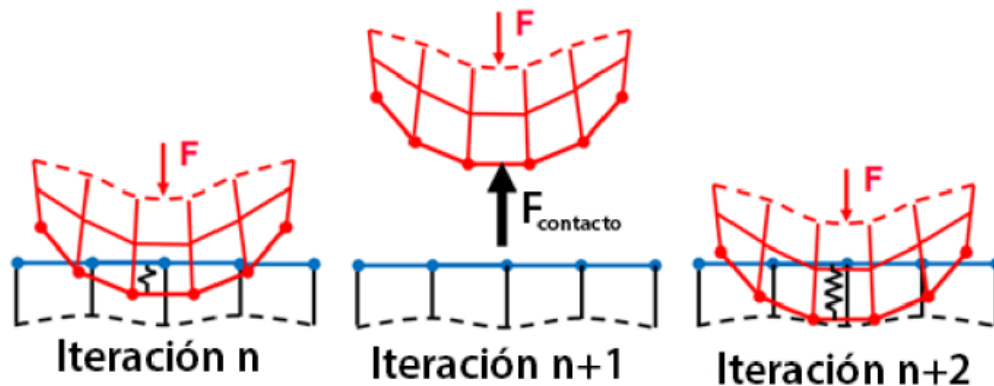


Figura 5.- Esquema de vibración en modelo no convergente.

## Análisis de articulación

En el caso de la articulación se usa la parte superior como empotrada para poder obtener el comportamiento de esta frente a los esfuerzos. Hay que notar que las fuerzas no son aplicadas directamente en el objeto a analizar, sino que son aplicadas lejos de él, sobre otro sistema, y es el contacto entre ambos lo que transmite las fuerzas al objeto de interés. Para emular este comportamiento se utiliza una fuerza remota en un punto remoto asociado a la cara de interfaz entre ambos sistemas. Un punto remoto, al aplicarle fuerzas, se comporta como un elemento rígido que une el punto en cuestión con el centro geométrico que ha sido asociado al punto. Esto permite transmitir tanto la fuerza como los momentos asociados a la geometría deseada. Es de interés de este trabajo el analizar la respuesta del sistema en las condiciones de contacto de talón y de levante de talón.

## Análisis de sistema pie

Para la simulación del comportamiento del sistema pie se utilizan fuerzas proyectadas sobre superficies pequeñas en la parte inferior del sistema, esto para simular la localización de la fuerza sobre las zonas respectivas de contacto.

Debido a que las diversas placas que conforman el sistema se encuentran unidas por pernos, las superficies de las placas deben ser capaces de deslizarse unas con respecto a las otras al deformarse, así como su separación en puntos lejanos a los lugares apernados. No obstante, si todas las superficies pudieran hacer eso, todas las placas presentarían movimiento de cuerpo rígido y se separarían. Para solucionar este problema se puede, o bien agregar los pernos como sólidos parte del modelo, o bien generar contacto de unión en ciertas zonas. Debido a la gran complejidad computacional agregada por los pernos, se usa la segunda opción.

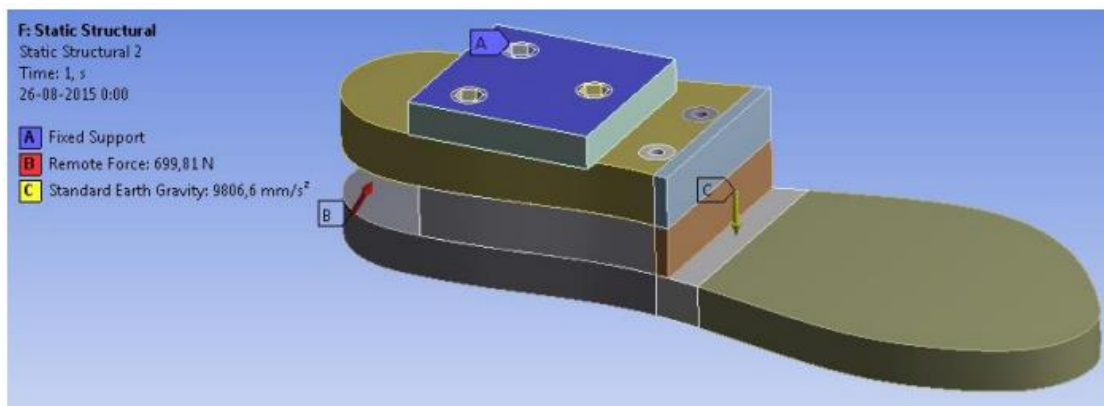


Figura 6.- Esquema de cargas para el sistema pie en carga trasera.

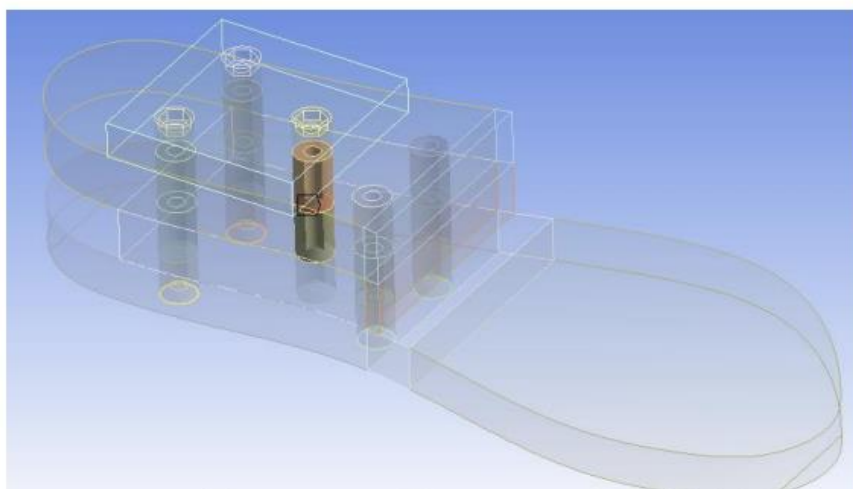


Figura 7.- Esquema de unión de placas vía cilindros.

## Resultados de la optimización

En el caso particular de la función objetivo mencionada, se ocupa una población inicial de 1500 individuos. Se considera como uso una sección circular constante de 5mm de diámetro. Cada uno de los individuos inicialmente puede utilizar tanto los 813 puntos de la envolvente convexa como los 1000 puntos internos generados de manera indistinta. Esto resulta en un cromosoma de 1813 variables por cada individuo. Como condición de término se impuso el llegar a 300 generaciones de individuos, con un tiempo de ejecución de 192 horas. Este proceso puede ser visto en la Figura siguiente. Cabe mencionar que este sistema no fue intervenido mediante penalizaciones.

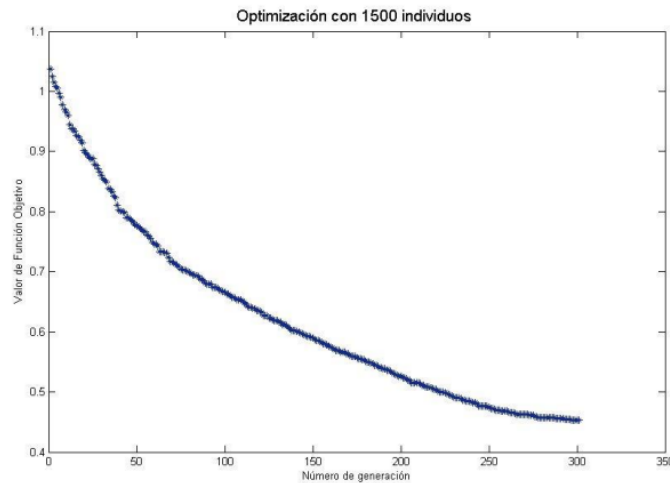


Figura 8.- Evolución de función objetivo en función de las generaciones.

Al evaluar la estructura campeona se obtienen los esfuerzos encontrados en la Figura siguiente. En esta figura los esfuerzos de compresión y tensión se encuentran en rojo y azul respectivamente, mientras que el límite de fatiga del material en esfuerzo cíclico se puede contemplar como líneas negras horizontales. La estructura en cuestión tiene un peso de 1.1[kg], con un desplazamiento promedio en todos sus elementos de  $2.4 \cdot 10^6$ [m].

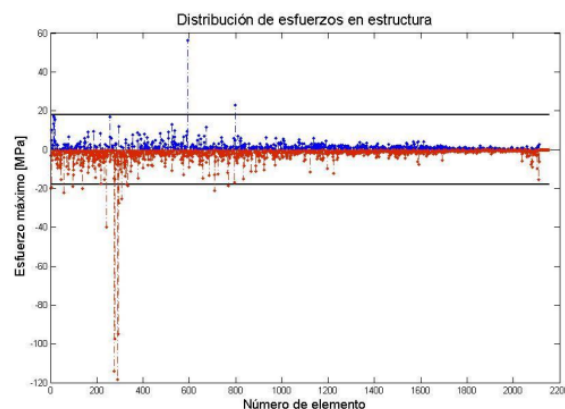
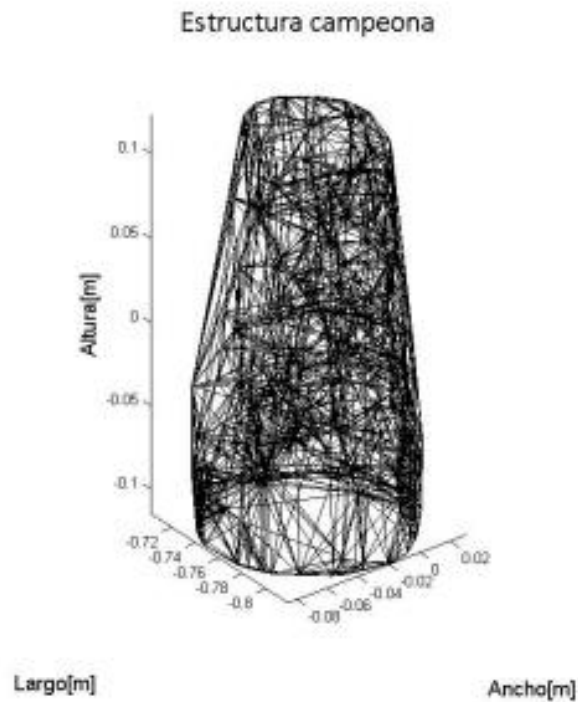
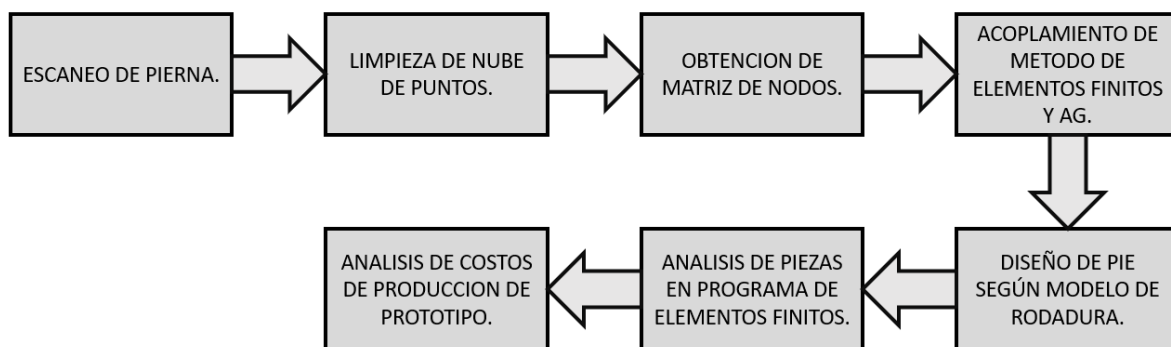


Figura 9.- Distribución de esfuerzos en estructura optimizada.

Se puede ver la geometría de la estructura campeona en la Figura de adelante Geometría de estructura campeona.



*Figura 10.- Geometría de la estructura optimizada.*





## Conclusiones

### **Víctor Cristopher Santiago Martínez 1858524**

En conclusión, con la practica anterior, el diseño de prótesis de pie depende de distintas variables en las cuales geometría, ángulo de inclinación entre otros. Además de eso también el estudio y análisis de la estructura a llevar a cabo como remplazo de esta misma. Finalmente. Se concluye que con la practica realizada se llegó a la idea de cómo se pude realizar de una forma más compleja una prótesis de este tipo.

### **Luis Lauro Garcia Hernandez 1803392**

En esta práctica se vio el diseño de una prótesis en este caso de un pie, con ello se pudo ver su geometría y ver cuál era el ángulo de inclinación de pie con todo eso se diseñó el pie en Matlab viendo bien los puntos aparte también hay que tener en consideración la longitud del pie, las articulaciones, etc. Finalmente se experimentó y se puede decir que se cumplió la experimentación y se puede entender que el desarrollar una prótesis de pie tiene una cierta dificultad y que hay que tomar muchos datos sobre ello para poder realizarlo con éxito.

### **Julio Adrián Torres Cisneros 1903681**

Para esta práctica se buscó desarrollar una prótesis de pierna de tal forma que se reduzca el peso a un mínimo y que sea lo suficientemente rígida y resistente para ser usada cómodamente.

Este caso es ideal para aplicar la optimización topológica, ya que una prótesis de pierna tiene una forma simple y no requiere de ser articulada, por lo que esta se puede hacer de una sola pieza rígida.

Al final aprendimos más acerca de cómo funciona la optimización topológica y como se puede aplicar a casos reales.

## Referencias Bibliográficas

- DOBERTI MARTÍNEZ, A. J. (2015, 13 octubre). DISEÑO DE UNA PRÓTESIS DE PIERNA PARA AMPUTADOS TRANSTIBIALES. DEPARTAMENTO DE INGENIERÍA MECÁNICA, UNIVERSIDAD DE CHILE. <https://repositorio.uchile.cl/bitstream/handle/2250/137154/Diseno-de-una-protesis-de-pierna-para-amputados-transtibiales.pdf?sequence=1>
- LTF Erick Lemus. (2021, August 15). ▷ Prótesis de Pie - Tipos, Funciones y Precios - Conoce más aquí. Mi Protesis de Pierna. <https://miprotesisdepierna.mx/protesis-de-pie/>
- Mediprax. (2018). Trayecto de las prótesis en la historia | Mediprax. Mediprax.mx. <https://mediprax.mx/trayecto-de-las-protesis-en-la-historia-primera-parte/>
- Un breve recorrido por la historia de la protésica - Amputee Coalition. (2018, October 26). Amputee Coalition. <https://www.amputee-coalition.org/resources/spanish-history-prosthetics/#:~:text=En%201863%2C%20Dubois%20Parmlee%20invent%C3%B3,fueran%20m%C3%A1s%20livianas%20y%20funcionales.>