

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

**Laboratorio de Biomecánica**

**Práctica 5. Optimización de una prótesis de pie**

<b>Nombre:</b>	<b>Matricula:</b>	<b>Carrera:</b>
Campos Miranda Julio Emilio	1917048	IMTC
Cruz Puente Sergio Eduardo	1903115	IMTC
Cynthia Belen Guerrero Pardo	1026215	IMTC
Jorge Luis Salinas Garza	1916367	IMTC
Acuña Rodríguez Christopher Russ	1894698	IMTC

**Maestro:** Ing Isaac Estrada

**Grupo:** 309

**Hora:** N5

Cd. Universitaria, San Nicolás de los Garza.

*16 de Noviembre de 2022*

## **Práctica 4. Refuerza del cable de un teleférico.**

### **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).

### **Nombre y definición de la Geometría**

Para hacer una prótesis de pie se deben de tomar en cuenta muchos factores, en el caso del soporte estructural se suelen usar tubos de aleaciones ligeras, por lo que es regular el uso de aluminio por su baja densidad y buenas propiedades comparado al acero convencional. En estas partes específicas se tiene mucha libertad en cuanto a forma y materiales, ya que el único trabajo de estas piezas es el resistir de buena manera los esfuerzos que se imprimen sobre ellas. Si se hacen optimizaciones, se pueden implementar algoritmos de luciérnagas de pocas variables que

permiten encontrar resultados satisfactorios con bajo número de iteraciones y bajo requerimiento computacional.

En el caso de los pies, éstos son clasificados según el grado de complejidad a medida que imitan de mejor manera la anatomía original de un pie natural. Una buena articulación afecta de gran manera la suavidad de la marcha y la eficiencia de ésta. Relacionado al movimiento del tobillo está la absorción de shock, aspecto crítico para evitar sobrecargas sobre el muñón.

La mayoría de los pies prostéticos son de “respuesta dinámica”, esto significa que el pie tiene una parte central deformable (que se comporta como resorte) que provee una sensación más intuitiva al caminar. Este tipo de diseño es llamado también como ESAR (Energy Storage and Return). La mayoría de los pies consisten de una quilla anterior de la que se desprenden una o más quillas posteriores que se deforman en el plano sagital. Durante la marcha la quilla posterior se deforma imitando la dorsiflexión plantar, luego ésta libera la energía a medida que el pie se carga hacia la parte anterior. Desde la mitad del apoyo hasta el despegue del pie, la quilla anterior se deforma imitando la flexión de los ortejos, liberando energía a medida que se levanta el pie. Algunos pies, en adición, tienen la quilla anterior partida en dos, para permitir un pequeño grado de pronunciación o de supinación. También es usual el uso de medios extras para absorber aún más el shock de la marcha mediante elementos pasivos como vigas curvas que, al aplicar un esfuerzo axial, se deflectan para absorber energía elástica

### **Estado del Arte.**

Una prótesis es básicamente una extensión de su cuerpo. Las prótesis individuales diferirán según el nivel de amputación, las capacidades físicas de una persona y sus objetivos y necesidades personales. Su protésico es responsable de hacer recomendaciones de diseño y de crear un dispositivo hecho a medida para adaptarse a su extremidad y adaptarse a su estilo de vida.

Los componentes básicos de una prótesis de miembro superior estándar son similares a los de una prótesis de miembro inferior (encajes, pilones, etc.). Mientras que una prótesis de miembro inferior incorpora un pie con un enfoque en la deambulación (caminar), una prótesis de miembro superior tendrá un «dispositivo terminal» como un gancho, una mano o una herramienta especializada, con un enfoque en la mejora funcional.

Una prótesis estándar se fabrica con componentes convencionales que crean la pierna y pie protésicos. Estas piezas están conectadas a un encaje que se ajusta por encima del muñón. El encaje permite que la prótesis se conecte al muñón. Las opciones más comunes de encaje son de succión, de vacío y con clavija.

La tecnología y los enfoques innovadores de dispositivos protésicos continúan brindando más y mejores opciones de atención. La provisión de una prótesis representa la culminación de la evaluación, la evaluación, el diseño, la fabricación, el ajuste, la capacitación y el seguimiento, lo que finalmente da como resultado un dispositivo personalizado. La profesión tiene una base clínica y está orientada al servicio, por lo que es importante que trabaje con un protésico que escuche sus necesidades y objetivos, y a quién tendrá un acceso relativamente fácil para ajustes y mantenimiento a lo largo del tiempo. Con los avances en técnicas, materiales y componentes, como rodillas con microprocesador y pies avanzados, manos con múltiples patrones de agarre y osteointegración, se continúan desarrollando nuevas oportunidades para ayudar a las personas con pérdida y diferencia de extremidades a vivir la vida que más desean vivir.

Tipos de prótesis:

- Pierna y pie. Hay varios pies protésicos disponibles para simular la acción de un pie natural después de una amputación por debajo de la rodilla. Al menos una prótesis de pie y tobillo disponible está controlada por un microprocesador. Utiliza la retroalimentación de los sensores para ajustar el movimiento de las articulaciones, lo que hace que caminar sea más eficiente y reduce el riesgo de caídas.
- Pierna con rodilla. Para las amputaciones por encima de la rodilla, la prótesis tiene una articulación de rodilla y tobillo. Actualmente existen más de 100 modelos protésicos de tobillo, pie y rodilla. Algunos usan dispositivos controlados por fluidos o hidráulicos que permiten a los usuarios variar su velocidad al caminar. Otros utilizan piezas computarizadas que permiten al usuario realizar ajustes rápidos en tiempo real mientras camina.

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

### **CODIGO CASO 1**

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

```

function top5(nelx,nely,volfrac,penal,rmin);
nelx=72;
nely=34;
volfrac=0.33;
penal=3.0;
rmin=1.2;
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
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=1:5
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);
% 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)
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)))));
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),5); U =sparse(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(3222,1) = -1;
F(3782,2) = -1;
F(2662,3) = -1;

```

```

F(2942,4) = -1;
F(3502,5) = -1;
fixeddofs = union([560:2*(nely+1):1260],[3920:2*(nely+1):4620]);
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)];

```

## **CODIGO CASO 2**

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

```

function top5(nelx,nely,volfrac,penal,rmin);
nelx=72;
nely=34;
volfrac=0.33;
penal=3.0;
rmin=1.2;
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
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=1:5
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);
% 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)
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)))));
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),5); U =sparse(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(3222,1) = -1;
F(3782,2) = -1;
F(2662,3) = -1;
F(2942,4) = -1;
F(3502,5) = -1;
fixeddofs = [3920:2*(nely+1):4620];
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)];
```

### **CODIGO CASO 3**

```
%%%%%% A 99 LINE TOPOLOGY
OPTIMIZATION CODE BY OLE SIGMUND, OCTOBER 1999 %%%%
function top5(nelx,nely,volfrac,penal,rmin);
nelx=72;
nely=34;
volfrac=0.33;
penal=3.0;
rmin=1.2;
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
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=1:5
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);
% 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)
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)))));
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),5); U =sparse(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 LOADS AND SUPPORTS (HALF MBB-BEAM)
F(3222,1) = -1;
F(3782,2) = -1;
F(2662,3) = -1;
F(2942,4) = -1;
F(3502,5) = -1;
fixeddofs = [560:2*(nely+1):1260];
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)];

```

## Resultados de la optimización

Caso 1



Caso 2



Caso 3



## Conclusiones

**Sergio Eduardo Cruz Puente 1903115**

Durante la elaboración de esta última práctica del laboratorio de biomecánica se estudió el comportamiento y el desarrollo de una prótesis de pie en la cual se visualizaban todos los

elementos, fuerzas que actúan sobre la prótesis y cómo ésta tiene que ser de un material muy resistente para que esta funcione adecuadamente. Se realizaron diversas optimizaciones en Matlab para así tener una mejor visualización de cómo sería una prótesis de pie.

### **Julio Emilio Camps Miranda 1917048**

En base a esta práctica de laboratorio de biomecánica, pudimos observar el comportamiento en este caso de una prótesis y de pie, pudiendo así notar el cómo las distintas cargas o las partes del cuerpo, se es necesario un tipo de prótesis, tal es el caso que tuvimos distintos puntos en el cual abordamos el problema y las distintas optimizaciones obtenidas. De igual manera se pudieron reforzar los conocimientos acerca de la optimización topológica en 19/20 Matlab por medio de la programación, más sin embargo ya aplicado a el área médica el cual nos será de mucha ayuda en dado caso que nos queramos enfocar en dicha rama de nuestra carrera

### **Christopher Russ Acuña Rodríguez 1894698**

Para finalizar la práctica, pude comprobar el potencial de Matlab, gracias a este software podemos rediseñar y optimizar una prótesis de pie. Esto nos hace ver el potencial que tiene esta aplicación. Se usó el algoritmo para la optimización topológica de esta prótesis y los resultados fueron sorprendentes.

### **Jorge Luis Salinas Garza 1916367**

A lo largo de este reporte pudimos observar y nos percatamos que se puede usar Matlab para generar un análisis de elemento finito para objetos de ámbito simple o más profesional y que se pueden usar para diferentes casos, además de generar un buen soporte que nos ayudará mucho en este caso. A partir de lo que aprendimos, nos damos cuenta que los softwares de hoy en día nos apoyan mucho con cálculos e impresiones que nos facilitan el poder generar nuevas ideas e ir más rápido en nuestras investigaciones. Asimismo, es importante destacar que la movilidad libre de las articulaciones y el trabajo que desempeñan los músculos es importante para lograr el mejor diseño posible. La falta de ciertas acciones, debe ser sustituida por otras, con el fin de mantener la estabilidad y la progresión. El uso de estas herramientas nos permite analizar nuestro diseño desde diferentes perspectivas.

### **Cynthia Belen Guerrero Pardo 1926215**

Para terminar con esta última actividad pude comprender que incluso para hacer cosas físicas como lo es en este caso una prótesis de pie, nos podemos apoyar de softwares de programación como en este caso lo fue Matlab; con este pudimos ver el comportamiento de nuestra prótesis y poder hacer su debido análisis gracias al elemento finito, con esto fuimos capaces de hacer nuestras propias conclusiones e idear que hacer para mejorar la prótesis, como en el principio de la actividad lo dice, una optimización.