

Universidad Autónoma de Nuevo León Facultad de Ingeniería Mecánica y Eléctrica



Laboratorio de Biomecánica

Práctica 5: Optimización de una prótesis de pie

Nombre	Matrícula	Carrera
Víctor Hugo Puente Álvarez	1929757	IMTC
Alejandro Hernández Navarro	1923272	IMTC
Angela Rodríguez Flores	1896624	IMTC
Diego Ávila González	1853396	IMTC
Osiris Acosta Cisneros	1992234	IMTC

Día y Hora: Martes V1

Brigada: 204

Docente: Isaac Estrada

Fecha de entrega: de 2022

1) Nombre y definición de la geometría

El objetivo de esta práctica es presentar una propuesta de análisis de formas y de la programación para la ejecución de la optimización (descripción funcional) de características de trabajo específicas y sus ventaja(s).

La geometría que se propuso a analizar y optimizar fue la de una prótesis de pie haciendo uso del algoritmo de optimización topológica con el fin de conseguir una optimización lo más apta posible de su estructura para un uso real, que soporte las cargas y distribuya el peso de la manera más óptima posible.

2) Estado del arte

Una prótesis es un dispositivo de extensión adaptable al organismo que reemplaza una parte del cuerpo que no se encuentra. Esta permite cumplir la función que ese segmento o estructura cumplía anteriormente.

Las prótesis se utilizan desde la antigüedad para reemplazar las estructuras faltantes en el cuerpo y de esta forma devolverle la funcionalidad luego de una amputación o malformación congénita, además del sentido estético a ese segmento afectado.

Es un elemento artificial que se adosa al cuerpo para brindar la función que anteriormente realizaba en el cuerpo, devolviendo la independencia y autonomía en las diferentes actividades de la vida diaria en caso de haber afectado esta área.



Imagen 5.1. Ejemplo de prótesis.

Tipos de prótesis

Endoprótesis - Son aquellas que requieren de una cirugía para su colocación, estas reemplazan una parte anatómica y suelen estar fabricadas con aleaciones metálicas como el cromo, titanio o cobalto.

Exoprótesis - Son aquellas que se ubican en la zona exterior, pueden ser retiradas y nuevamente colocadas por el paciente. Estas sustituyen parcial o totalmente un miembro o segmento del cuerpo por amputación, con el objetivo de restituir la función anatómica y la integración socio-laboral.

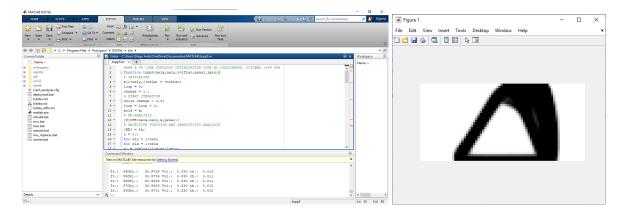
3) Propuesta de diseño de la geometría, alcances y limitaciones

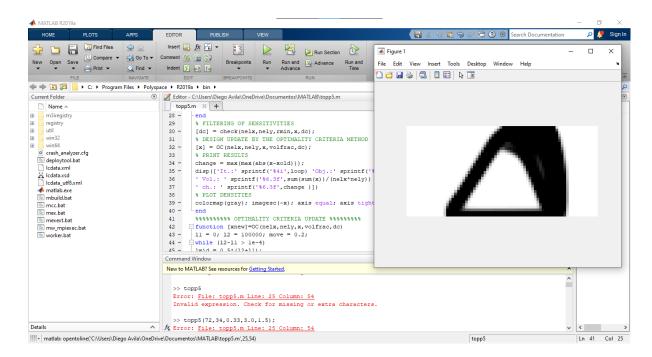
Para la elaboración de esta práctica se analizó el comportamiento de un solo pie dentro de las 3 fases que componen la marcha del ser humano.

- 1- Normal (el talón y el área metatarsial son los apoyos, la fuerza de aplica directamente sobre el tobillo).
- 2- Despliegue (El área metatarsial es el apoyo, la fuerza se aplica sobre el tobillo con un ángulo de 30°).
- 3- Apoyo (El área del talón es el apoyo, la fuerza de aplica sobre el tobillo con un ángulo de 50°).

4) Pasos del desarrollo de la Programación

Parte1





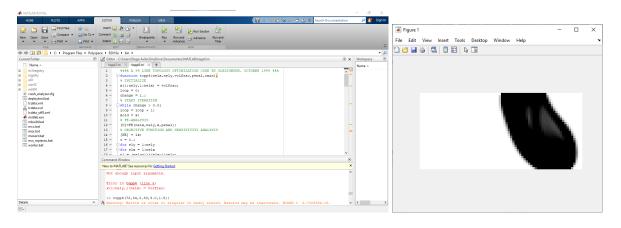
topp5(72,34,0.33,3.0,1.5);

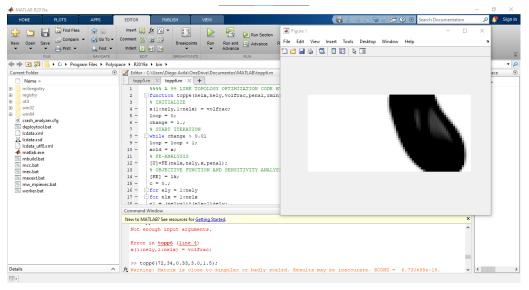
Código 1

```
%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY
                                                                 % PRINT RESULTS
OLESIGMUND, OCTOBER 1999 %%
                                                                 change = max(max(abs(x-xold)));
function topp5(nelx,nely,volfrac,penal,rmin);
                                                                 disp(['It.:' sprintf('%4i',loop) 'Obj.:' sprintf('%10.4f',c) ...
% INITIALIZE
                                                                 'Vol.: 'sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
x(1:nely,1:nelx) = volfrac;
                                                                 'ch.: 'sprintf('%6.3f',change)])
loop = 0;
                                                                 % PLOT DENSITIES
change = 1.;
                                                                 colormap(gray); imagesc(-x); axis equal; axis tight; axis
% START ITERATION
                                                                 off;pause(1e-6);
while change > 0.01
                                                                 end
                                                                 %%%%%%%%%% OPTIMALITY CRITERIA UPDATE
loop = loop + 1;
xold = x;
                                                                 %%%%%%%%%%%%
% FE-ANALYSIS
                                                                 function [xnew]=OC(nelx,nely,x,volfrac,dc)
                                                                 11 = 0; 12 = 100000; move = 0.2;
[U]=FE(nelx,nely,x,penal);
% OBJECTIVE FUNCTION
                                  AND
                                          SENSITIVITY
                                                                 while (I2-I1 > 1e-4)
ANALYSIS
                                                                 Imid = 0.5*(12+11);
[KE] = Ik;
                                                                 xnew
c = 0.:
                                                                 max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmi
for ely = 1:nely
                                                                 d)))));
for elx = 1:nelx
                                                                 if sum(sum(xnew)) - volfrac*nelx*nely > 0;
n1 = (nely+1)*(elx-1)+ely;
                                                                 I1 = Imid;
n2 = (nely+1)^* elx + ely;
                                                                 else
dc(ely,elx)=0.;
                                                                 12 = Imid:
for i=1:5
                                                                 end
Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1; 2*n2+2;
                                                                                      MESH-INDEPENDENCY FILTER
2*n1+1;2*n1+2],i);
                                                                 %%%%%%%%%%%%
c = c + x(ely,elx)^penal*Ue'*KE*Ue;
                                                                 %%%%%%%%%%%%%%%
dc(elv.elx)
                                                                 function [dcn]=check(nelx,nely,rmin,x,dc)
dc(ely,elx)-penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
                                                                 dcn=zeros(nely,nelx);
                                                                 for i = 1:nelx
                                                                 for j = 1:nely
end
                                                                 sum=0.0:
end
% FILTERING OF SENSITIVITIES
                                                                 for k = max(i-round(rmin),1):min(i+round(rmin),nelx)
[dc] = check(nelx,nely,rmin,x,dc);
                                                                 for I = max(j-round(rmin),1):min(j+round(rmin), nely)
% DESIGN UPDATE BY THE OPTIMALITY CRITERIA
                                                                 fac = rmin-sqrt((i-k)^2+(j-l)^2);
METHOD
                                                                 sum = sum + max(0,fac);
[x] = OC(nelx,nely,x,volfrac,dc);
                                                                 dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
```

```
F(2942,4) = -1;
end
                                                                   F(3502,5) = -1;
end
dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
                                                                   fixeddofs
                                                                   union([560:2*(nely+1):1260],[3920:2*(nely+1):4620]);
end
                                                                   alldofs = [1:2*(nely+1)*(nelx+1)];
end
%%%%%%%%%%%%%%
                                           FE-ANALYSIS
                                                                   freedofs = setdiff(alldofs,fixeddofs);
%%%%%%%%%%%%%%%%%
                                                                   % SOLVING 127
                                                                   U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
function [U]=FE(nelx,nely,x,penal)
                                                                   U(fixeddofs,:)= 0;
[KE] = Ik;
K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
                                                                   %%%%%%%%% ELEMENT STIFFNESS MATRIX
                  sparse(2*(nely+1)*(nelx+1),5);
                                                                   %%%%%%%%
=sparse(2*(nely+1)*(nelx+1),5);
                                                                   function [KE]=lk
for ely = 1:nely
                                                                   E = 1.;
for elx = 1:nelx
                                                                   nu = 0.3;
n1 = (nely+1)*(elx-1)+ely;
                                                                   k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
n2 = (nely+1)^* elx + ely;
                                                                   -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
edof = [2*n1-1;
                                                                   KE = E/(1-nu^2)^*[k(1)k(2)k(3)k(4)k(5)k(6)k(7)k(8)
                             2*n1;
                                       2*n2-1;
                                                    2*n2:
2*n2+1;2*n2+2;2*n1+1; 2*n1+2];
                                                                   k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
                                                                   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)
end
                                                                   k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
% DEFINE LOADSAND SUPPORTS(HALF MBB-BEAM)
                                                                   k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
F(3222,1) = -1;
                                                                   k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
F(3782,2) = -1;
                                                                   k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
F(2662,3) = -1;
```

Parte 2

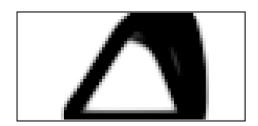


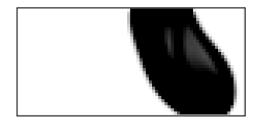


topp6(72,34,0.33,3.0,1.5);

5) Resultados

Ambas partes unidas:





Código 2

```
%%%% A 99 LINE TOPOLOGY OPTIMIZATION
CODE BY OLESIGMUND, OCTOBER 1999 %%%
function topp6(nelx,nely,volfrac,penal,rmin);
% 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] = Ik;
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)
11 = 0; 12 = 100000; move = 0.2;
while (12-11 > 1e-4)
Imid = 0.5*(12+11);
xnew
max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(
-dc./lmid)))));
if sum(sum(xnew)) - volfrac*nelx*nely > 0;
I1 = Imid;
else
12 = Imid:
end
end
%%%%%%%%%%%%%
                          MESH-INDEPENDENCY
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 I = 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
                                                            F(2662,3) = -1;
dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
                                                            F(2942,4) = -1;
                                                            F(3502,5) = -1;
                                                            fixeddofs = [3920:2*(nely+1):4620];
end
%%%%%%%%%%%%%
                                     FE-ANALYSIS
                                                            alldofs = [1:2*(nely+1)*(nelx+1)];
%%%%%%%%%%%%%%%%
                                                            freedofs = setdiff(alldofs,fixeddofs);
function [U]=FE(nelx,nely,x,penal)
                                                            % SOLVING 127
                                                            U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
[KE] = Ik;
                        sparse(2*(nelx+1)*(nely+1),
                                                            U(fixeddofs,:)= 0;
Κ
                                                            %%%%%%%%%%%%
                                                                                    ELEMENT
                                                                                                   STIFFNESS
2*(nelx+1)*(nely+1));
              sparse(2*(nely+1)*(nelx+1),5);
                                                            MATRIX %%%%%%%%
=sparse(2*(nely+1)*(nelx+1),5);
                                                            function [KE]=Ik
for ely = 1:nely
                                                            E = 1.;
for elx = 1:nelx
                                                            nu = 0.3:
                                                            k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
n1 = (nely+1)*(elx-1)+ely;
n2 = (nely+1)^* elx + ely;
                                                            -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
                         2*n1;
             [2*n1-1;
                                   2*n2-1;
                                              2*n2;
                                                            KE = E/(1-nu^2)^*[k(1) k(2) k(3) k(4) k(5) k(6) k(7)]
      =
2*n2+1;2*n2+2;2*n1+1; 2*n1+2];
                                                            k(8)
K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
                                                            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)
end
end
                                                            k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
     DEFINE LOADSAND SUPPORTS(HALF
                                                            k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
MBB-BEAM)
                                                            k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
F(3222,1) = -1;
                                                            k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
F(3782,2) = -1;
                                                            k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1);
```

6) Conclusiones

Alejandro Hernandez Navarro

Durante esta práctica se propuso una geometría para una prótesis de pie con la finalidad de aplicar la optimización topológica y poder optimizar la geometría propuesta, reduciendo la masa y manteniendo la rigidez de la pieza. Para lograr esto se utilizaron conocimientos previos respecto al código de optimización topológica en Matlab para adaptarlo al problema que se está trabajando, una vez que se propuso el diseño con sus respectivas restricciones y condiciones frontera, se adaptó el código de Matlab de optimización topológica para así obtener el diseño final.

Diego Avila González

Esta práctica es la última que realizaremos del laboratorio de biomecánica y por lo tanto como su nombre lo indica podremos realizar un diseño de un cuerpo biologico y mecanico que tienen movimiento, nada mejor que realizar el diseño de un pie humano, para esto pondremos en práctica lo que hemos aprendido en las pasadas prácticas ya que este es un diseño más completo y complejo, por lo tanto influenciaron más cosas para poderlo realizar correctamente, con esta práctica

concluimos que con la optimización topológica si podemos realizar usos en la biomecánica y podremos realizar prótesis de varias partes del cuerpo.

Victor Hugo Puente Alvarez

Por medio de esta practica se puede ver como diseñar piezas en diferentes etapas, para que al momento de hacer un ensamble tener como resultado una que en este caso fue una protesis para un pie, se utilizo de tal forma los conocimientos vistos en las practicas antiriores para optimizar dichas piezas y minimizar su masa y maximizar su rigidez para tener el mejor resultado posible.

Osiris Acosta Cisneros

En esta práctica vimos como diseñar un pie humano, dónde pudimos a prueba todo lo que hemos aprendido hasta el momento en el laboratorio. Pudimos aplicar el conocimiento de manera efectiva y se puede concluir que la optimización topológica también se puede utilizar es parte biológicas.

Referencias bibliográficas

- FisioOnline. (s.f.). *Definición - Qué es prótesis*. FisioOnline. Recuperado 24 de octubre de 2022, de https://www.fisioterapia-online.com/glosario/protesis