

Practica 5 Optimización de una prótesis de pie

Uriel G. Ernesto A. Marcela O. Ana L. Diego O

15 de noviembre de 2022

Resumen

Se realizará una optimización enfocada en el pie, por lo cual se realizara un breve estudio de la biomecánica del pie, se continuará trabajando con programación en Matlab.

1. Introducción

el pie está sometido a cargas biomecánicas transmitidas por sus estructuras que le dan funcionalidad, hay diferentes tipos de funciones: soporte, locomoción, percusión y ataque,¹ para llevar a cabo estas funciones biomecánicas, es necesario el funcionamiento de todas las estructuras que conforman la articulación. Cuando el atleta presenta dolor del tercio distal de la extremidad o sólo del pie sin antecedentes de trauma, hay que sospechar en anomalías biomecánicas

2. Anatomía Funcional

Recuerdo anatómico del pie El pie tiene 3 partes fundamentales: la bóveda plantar, su apoyo posterior o talón y su apoyo anterior o antepié.

Bóveda plantar

La parte superior de la bóveda, que soporta fuerzas a compresión, está formada por los huesos; la inferior, que resiste esfuerzos de tracción, está constituida por ligamentos aponeuróticos y músculos cortos, que son las estructuras preparadas mecánicamente para esta función. Mantiene su forma gracias a una serie de estructuras que la estabilizan. Estas estructuras son los huesos, las cápsulas y ligamentos y los músculos. Los 2 primeros lo hacen de forma pasiva, mientras que los últimos lo hacen de una forma activa.

Talón

Visto por detrás, el talón debe seguir la línea de Helbing (vertical que pasa por el centro del hueso poplíteo y por el centro del talón), o bien desviarse en unos 5° de valgo, lo cual contribuye a amortiguar el choque del talón con el suelo durante la marcha.

Antepié

El antepié es una entidad anatomofuncional amortiguadora y propulsiva compuesto por cinco cadenas cinemáticas libres denominadas radios. También es un sensor sensitivo y sensorial para la función del equilibrio. El antepié está unido al resto del pie por el complejo articular de Lisfranc, que forma una bóveda transversal alejada del suelo en su parte medial, mientras que las cabezas metatarsianas están en un mismo plano apoyadas en el suelo. Se trata de una estructura deformable viscoelástica de geometría variable. Puede soportar tensiones muy elevadas durante tiempos muy breves de unos milisegundos [1].

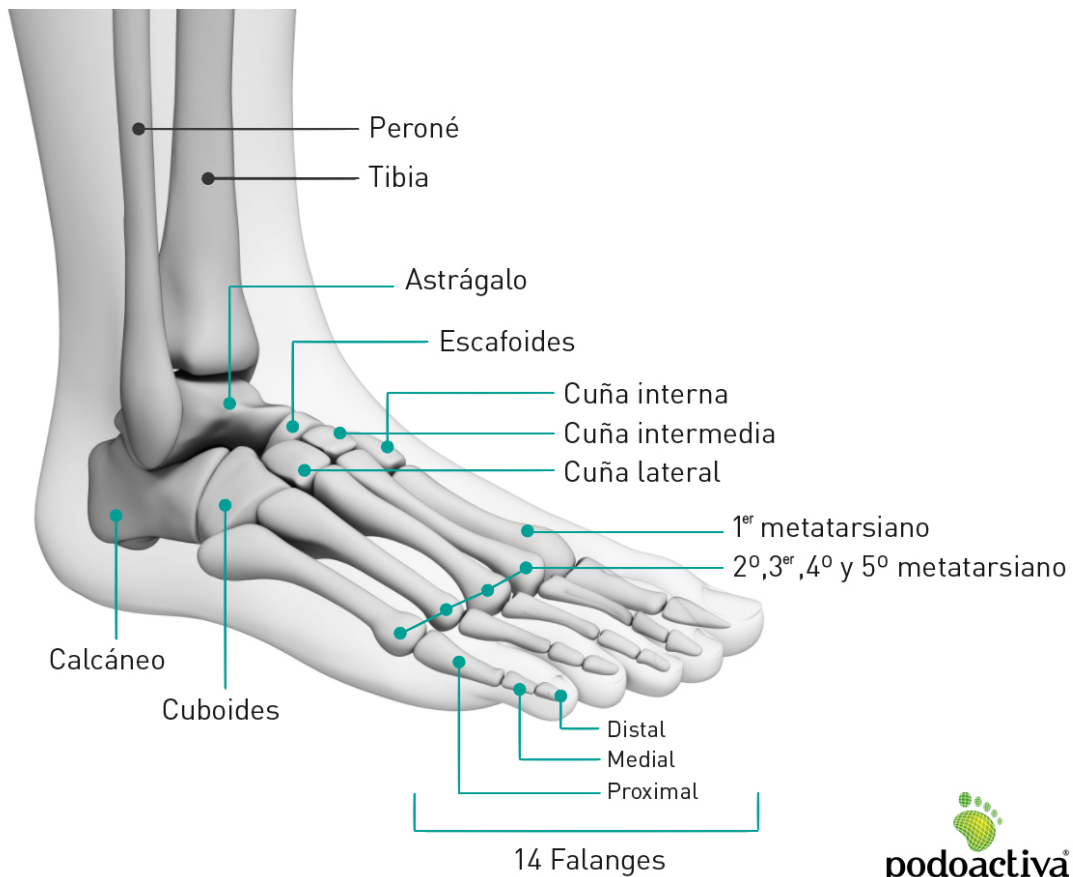
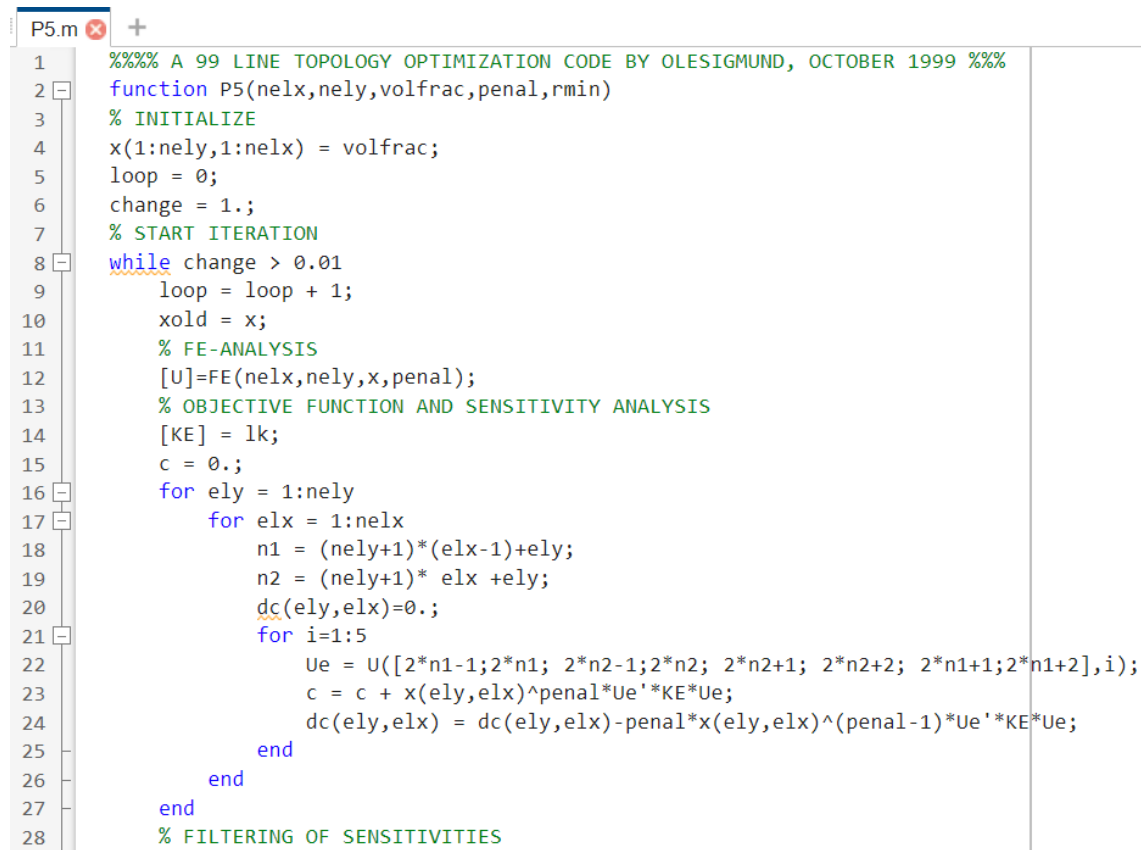


Figura 1: Partes del pie

3. Propuesta de diseño

Proponer el diseño de una prótesis de pie para los diferentes estados dentro de la marcha humana.

4. Código de programación



```
P5.m
1  %%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLESIGMUND, OCTOBER 1999 %%%
2  function P5(nelx,nely,volfrac,penal,rmin)
3  % INITIALIZE
4  x(1:nely,1:nelx) = volfrac;
5  loop = 0;
6  change = 1.;
7  % START ITERATION
8  while change > 0.01
9      loop = loop + 1;
10     xold = x;
11     % FE-ANALYSIS
12     [U]=FE(nelx,nely,x,penal);
13     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
14     [KE] = lk;
15     c = 0.;
16     for ely = 1:nely
17         for elx = 1:nelx
18             n1 = (nely+1)*(elx-1)+ely;
19             n2 = (nely+1)* elx +ely;
20             dc(ely,elx)=0.;
21             for i=1:5
22                 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);
23                 c = c + x(ely,elx)^penal*Ue'*KE*Ue;
24                 dc(ely,elx) = dc(ely,elx)-penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
25             end
26         end
27     end
28     % FILTERING OF SENSITIVITIES
```

Figura 2: Parte 1 del código

```

29     [dc] = check(nelx,nely,rmin,x,dc);
30     % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
31     [x] = OC(nelx,nely,x,volfrac,dc);
32     % PRINT RESULTS
33     change = max(max(abs(x-xold)));
34     disp(['It.: ' sprintf('%4i',loop) 'Obj.: ' sprintf('%10.4f',c) ...
35          ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
36          ' ch.: ' sprintf('%6.3f',change )])
37     % PLOT DENSITIES
38     colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
39 end
40 end
41
42 %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
43 function [xnew]=OC(nelx,nely,x,volfrac,dc)
44     l1 = 0; l2 = 100000; move = 0.2;
45     while (l2-l1 > 1e-4)
46         lmid = 0.5*(l2+l1);
47         xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
48         if sum(sum(xnew)) - volfrac*nelx*nely > 0
49             l1 = lmid;
50         else
51             l2 = lmid;
52         end
53     end
54 end
55
56 %%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%

```

Figura 3: Parte 2 del código

```

57 function [dcn]=check(nelx,nely,rmin,x,dc)
58     dcn=zeros(nely,nelx);
59     for i = 1:nelx
60         for j = 1:nely
61             sum=0.0;
62             for k = max(i-round(rmin),1):min(i+round(rmin),nelx)
63                 for l = max(j-round(rmin),1):min(j+round(rmin), nely)
64                     fac = rmin-sqrt((i-k)^2+(j-l)^2);
65                     sum = sum+max(0,fac);
66                     dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
67                 end
68             end
69             dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
70         end
71     end
72 end
73
74 %%%%%%%%%%% FE-ANALYSIS %%%%%%%%%%%
75 function [U]=FE(nelx,nely,x,penal)
76     [KE] = lk;
77     K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
78     F = sparse(2*(nely+1)*(nelx+1),5); U =sparse(2*(nely+1)*(nelx+1),5);
79     for ely = 1:nely
80         for elx = 1:nelx
81             n1 = (nely+1)*(elx-1)+ely;
82             n2 = (nely+1)* elx +ely;
83             edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1;2*n2+2;2*n1+1; 2*n1+2];
84             K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;

```

Figura 4: Parte 3 del codigo

```

85         end
86     end
87     % DEFINE LOADSAND SUPPORTS(HALF MBB-BEAM)
88     F(3222,1) = -1;
89     F(3782,2) = -1;
90     F(2662,3) = -1;
91     F(2942,4) = -1;
92     F(3502,5) = -1;
93     fixeddofs = union([560:2*(nely+1):1260],[3920:2*(nely+1):4620]);
94     alldofs = (1:2*(nely+1)*(nelx+1));
95     freedofs = setdiff(alldofs,fixeddofs);
96     % SOLVING 127
97     U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
98     U(fixeddofs,:)= 0;
99     end
100
101     %%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%
102     function [KE]=lk
103     E = 1.;
104     nu = 0.3;
105     k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
106       -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
107     KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
108       k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
109       k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
110       k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
111       k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
112       k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)

```

Figura 5: Parte 4 del código

```

P5.m x +
87 % DEFINE LOADS AND SUPPORTS (HAFT HBB-BEAM)
88 F(3222,1) = -1;
89 F(3782,2) = -1;
90 F(2662,3) = -1;
91 F(2942,4) = -1;
92 F(3502,5) = -1;
93 fixeddofs = union([560:2*(nely+1):1260],[3920:2*(nely+1):4620]);
94 alldofs = (1:2*(nely+1)*(nelx+1));
95 freedofs = setdiff(alldofs,fixeddofs);
96 % SOLVING 127
97 U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
98 U(fixeddofs,:)= 0;
99 end
100
101 %%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%
102 function [KE]=lk
103 E = 1.;
104 nu = 0.3;
105 k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
106 -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
107 KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
108 k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
109 k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
110 k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
111 k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
112 k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
113 k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
114 k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)]
115 end

```

Figura 6: Parte 5 del código

5. Resultado del Código de programación

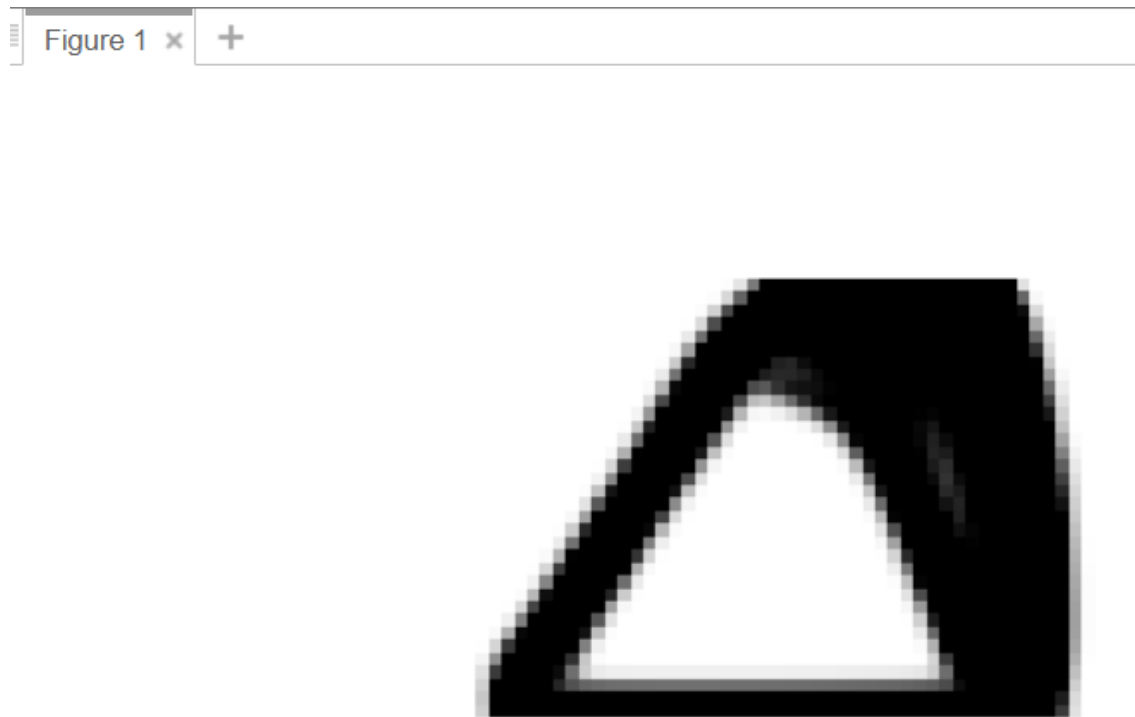


Figura 7: Resultado de la optimización

6. Código de programación parte 2

```
87 % DEFINE LOADSAND SUPPORTS(HALF MBB-BEAM)
88 F(3222,1) = -1;
89 F(3782,2) = -1;
90 F(2662,3) = -1;
91 F(2942,4) = -1;
92 F(3502,5) = -1;
93 fixeddofs = [560:2*(nely+1):1260];
94 alldofs = [1:2*(nely+1)*(nelx+1)];
95 freedofs = setdiff(alldofs,fixeddofs);
96 % SOLVING 127
97 U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
98 U(fixeddofs,:)= 0;
99 end
```

Figura 8: Parte 1 del código

7. Resultado de programación parte 2

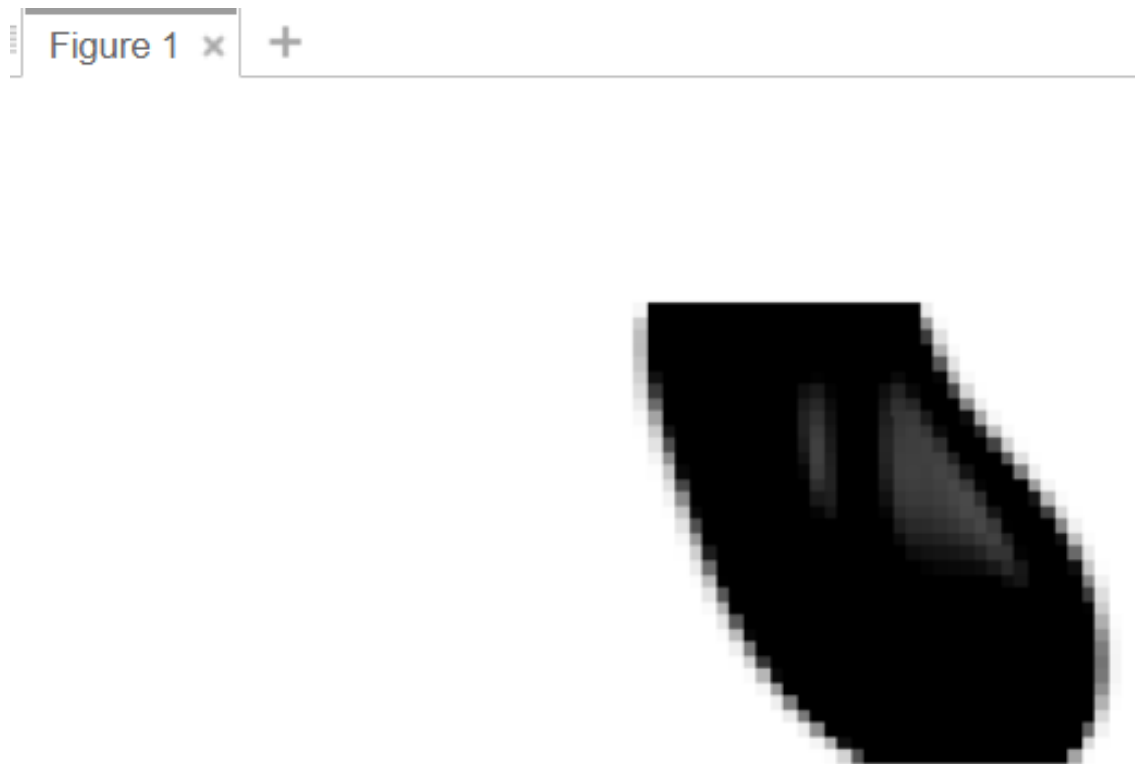


Figura 9: Resultado del código

8. Código de programación parte 3

```
87 % DEFINE LOADSAND SUPPORTS(HALF MBB-BEAM)
88 F(3222,1) = -1;
89 F(3782,2) = -1;
90 F(2662,3) = -1;
91 F(2942,4) = -1;
92 F(3502,5) = -1;
93 fixeddofs = [3920:2*(nely+1):4620];
94 alldofs = [1:2*(nely+1)*(nelx+1)];
95 freedofs = setdiff(alldofs,fixeddofs);
96 % SOLVING 127
97 U(freedofs,:) = K(freedofs,freedofs) \F(freedofs,:);
98 U(fixeddofs,:)= 0;
99 end
```

Figura 10: Parte 3 del código

9. Resultado de programación parte 2



Figura 11: Resultado del código

10. Conclusiones

■ Diego O.

Al término de esta práctica se pudo apreciar la gran importancia que tienen los diferentes lenguajes de programación siendo este caso matlab para poder realizar procedimientos algebraicos de una manera más óptima y rápida, así como la facilidad de poder mostrar gráficamente estos resultados, aplicados ahora hacia la prótesis Y es de gran importancia el manejo de software en este rubro de la biomecánica ya que son una necesidad cotidiana en una parte de la población mundial

■ Ana L.

Gracias a esta practica se pudo observar y realizar la optimizacion topologica a partir del software de modelado MATLAB, todo esto posible gracias al codigo de 99 lineas. Esta practica fue muy interesante de hacer y lo mas complicado fue el codigo.

■ Uriel G.

En esta practica se puede concluir que matlab es una gran herramienta para la simulacion y creacion de prototipos a la hora de crear algo nuevo, tambien puedo decir que estuvo un poco complicada debido a que eran 3 estructuras diferentes sobre la prótesis de pie, pero fue interesante el como crearlas por medio de un codigo que gracias a ello puede tomar distintas medidas cual sea la necesidad.

■ Marcela O.

En esta práctica pudimos aplicar más conocimiento del que vimos en clase respecto a las prótesis, pero en este caso lo pudimos aplicar y optimizar con MATLAB. Lo cual nos lleva a tener un punto de vista diferente y complementar nuestro trabajo. Gracias a lo que hicimos en la práctica pudimos repasar conocimientos de clases y semestres previos. Las prótesis son de gran ayuda en la vida cotidiana de muchas personas y es importante que sigamos desarrollando la tecnología en este aspecto, para así poder seguir ayudando y apoyando a más personas.

■ Ernesto A.

Gracias a las practicas realizadas con anterioridad, pudimos llegar al punto de optimizar lo que fue una prótesis de pie a lo cual llegamos a concluir como estas optimizaciones nos pueden servir para el diseño de una, tambien la importancia de tener una buena toma de mediciones y de consulta del paciente a colgar, dado que esto es de gran importancia por que cada ser humano es diferente, por lo cual es importante el estudio de la biomecánica. Y como el MATLAB es una herramienta de uso muy completa para todo tipo de actividades y trabajos.

Referencias

- [1] A. V. Voegeli. Functional anatomy and biomechanics of the ankle and foot. number =.