

Práctica 2

Diseño del marco de una bicicleta

Equipo 4
Omar Gutierrez, Pietro Castañeda, Lauro Garcia

19 de octubre de 2022

1. Introducción

La optimización topológica del marco de una bicicleta es muy importante ya que este es la parte central de la bicicleta, es prácticamente su columna vertebral y donde se unen todos sus componentes. Puede ser de diferentes materiales, las más comunes acero, aluminio o fibra de carbono, esto depende de la disciplina en la que se use la bicicleta, ya que de esto depende su ligereza.

Este trabajo propone el diseño de un marco de bicicleta y sobre este la aplicación de un código de 99 líneas de optimización topológica con el uso del criterio de optimalidad estándar, con el fin de asignar una geometría con valores propuestos, más cargas, huecos vacíos y soportes para representarlo en Matlab, este código fue realizado por el profesor Ole Sigmund, así mismo se aborda la teoría necesaria sobre el diseño óptimo de bicicletas, optimización topológica y el ambiente de programación en Matlab.

2. 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) ventajas(s) (mencionar ventajas)

La metodología que utilizará la propuesta se compondrá en los siguientes pasos:

- 1) Nombre y definición de la geometría
- 2) Estado del arte
- 3) Propuesta de diseño de la geometría, alcances y limitaciones
- 4) Pasos del desarrollo de la programación
- 5) Resultados de la optimización
- 6) Conclusiones

3. Nombre y definición de la forma geométrica

En esta práctica la forma geométrica que se propuso a optimizar fue un marco de bicicleta en la cual se hará uso y modificación del código de 99 líneas de optimización topológica con el fin de determinar la distribución de masa más eficiente para cumplir exactamente con los requisitos de diseño impuestos a la pieza.

Para definir la forma geométrica se examina sobre el cuadro o marco de una bicicleta común, por lo que en la figura 1 podemos observarlas partes que lo componen y más adelante abordaremos esto mismo y veremos algunas diferencias que pueden ser cruciales a la hora de elegir uno o querer añadir ciertos accesorios y piezas.



Figura 1: Componentes del marco de bicicleta

4. Estado del arte

4.1. Optimización topológica

La optimización de la topología (TO) es un método matemático que optimiza el diseño del material dentro de un espacio de diseño dado, para un conjunto dado de cargas, condiciones de contorno y restricciones con el objetivo de maximizar el rendimiento del sistema. TO es diferente de la optimización de la forma y la optimización del tamaño en el sentido de que el diseño puede alcanzar cualquier forma dentro del espacio de diseño, en lugar de tratar con configuraciones predefinidas.

La formulación convencional de TO utiliza un método de elementos finitos [FEM] para evaluar el desempeño del diseño [2]. El diseño se optimiza utilizando técnicas de programación matemática basadas en gradientes, como el algoritmo de criterios de optimalidad y el método de mover asíntotas o algoritmos no basados en gradientes, como los algoritmos genéticos.

En el proceso de optimización topológica, se deben de tener en cuenta varios aspectos; el espacio de diseño, el o los casos de carga que va a sufrir la pieza en cuestión, el material y la tecnología con que se va a realizar su fabricación, la reducción de costes mediante la minimización de soportes y aprovechamiento de la cuba de impresión, en caso de utilizar tecnologías aditivas, y muchos más.

Pasos Optimización Topológica:

1. Dibujar o Importar geometría
2. Simplificar la pieza y definir el espacio de diseño
3. Establecer uniones, juntas y contactos
4. Asignar materiales
5. Definir los casos de carga
6. Generar la optimización
7. Refinar la geometría
8. Exportar a CAD o generar STL
9. Verificar el rendimiento
10. Fabricar

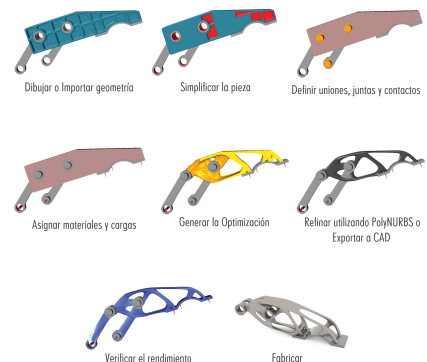


Figura 2: Flujo de trabajo

4.2. Geometría de un cuadro de bicicleta

En la figura 1, podremos distinguir dos “triángulos”. El frontal se conforma por el tubo del asiento, superior, inferior y el de la dirección. Por su parte, el triángulo trasero; está formado por los tirantes y el tubo del asiento[3].

- **Tubo del asiento:**

Va desde el eje de centro o pedalier, que es la parte donde van los pedales, hasta el asiento.

- **Tubo superior:**

Puede correr horizontalmente, inclinado o inclusive con algunas curvas, dependiendo del tipo de bici, y va desde el tubo de la dirección hasta el tubo del asiento. La distancia del tubo del asiento al de dirección es una medida importante al momento de elegir un cuadro de talla adecuada.

- **Tubo inferior:**

Corre diagonalmente desde el eje de centro hasta la parte baja del poste de la dirección.

- **Tubo de la dirección:**

Es muy corto en comparación con el resto y conecta el tubo superior con el inferior. Su longitud varía dependiendo del tipo de bici y es por donde entra el poste de dirección de la tijera.

- **Tirante superior/inferior:**

Dos juegos de tubos, más delgados que el resto. Los tirantes superiores corren desde el tubo del asiento hasta los terminales de donde se sujeta la rueda trasera. Los tirantes inferiores conectan el forkend al eje de centro.

- **Ángulo de dirección:**

Este es el que se forma entre la pipa de la dirección y suelo. Siendo un ángulo habitual 73° , podemos apreciar que cuanto más se acerca a 90° más nerviosa es la dirección, y cuanto más nos alejamos de 90° , encontramos una dirección más dócil y tenderá a corregir sola la trayectoria.



Figura 3: Ángulo de dirección

- **Ángulo de sillín:**

Es el formado entre suelo y tubo del sillín. Siendo un ángulo habitual 72° , podemos apreciar que cuanto más se acerca a 90° estaremos sentados más lejos del eje de la rueda trasera = menos tracción, por el contrario, cuanto más se aleja de 90° tendremos mayor tracción [1].

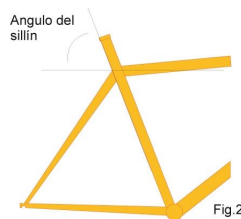


Figura 4: Ángulo de sillín

5. Propuesta de diseño de la gemetría, alcanes y limitaciones

La diferencia más importante entre las bicicletas de hombre y de mujer radica en la geometría de los cuadros. Unas variantes en el diseño de los cuadros atribuibles a las diferencias existentes entre la morfología masculina y la femenina. Hombres y mujeres tenemos similitudes físicas, pero indudablemente también diferencias importantes como para que el diseño del cuadro presente variantes. Así pues, aprovechamos para desmentir completamente el tópico de que la bicicleta de mujer es una bicicleta de hombre más pequeña.

El problema de diseño se ilustra en la figura 5. Nuestra tarea es construir la parte delantera del marco para que la bicicleta sea lo más rígido posible. Esta parte se conecta con el manubrio y el asiento.

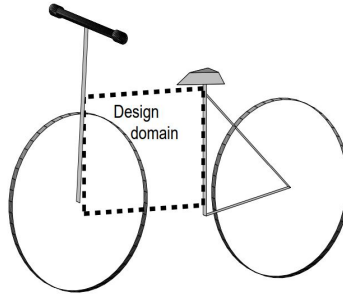


Figura 5: Dominio de diseño

El dominio de diseño se ilustra esquemáticamente en la figura 6, tomando las siguientes consideraciones.

- El manubrio produce una fuerza en dirección vertical
- Bastidor trasero actúa como soporte
- Tenemos que declarar una parte vacía del domino de diseño para hacer el espacio de la rueda delantera

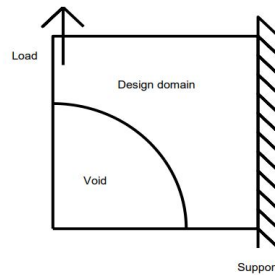


Figura 6: Vista esquemática

Las limitaciones que presenta este análisis, es que solo se analiza una parte del marco de la bicicleta debido a la complejidad que tomaría el programa para poder analizar correctamente el marco completo, ya que se agregarían más cargas, más puntos fijos y vacíos en la figura, además de esto es que existen más variables que pueden intervenir al momento de querer realizar la optimización topológica, como fenómenos naturales, los materiales de fabricación entre otros, desgaste, cargar variables, etc. Al tratar de simplificarla se busca que el alcance de este logre asimilarse al comportamiento que se le ejerce hacia una bicicleta.

6. Desarrollo de programación

Inicialmente, el programa cuenta con 99 líneas del código, estas están divididas en cinco secciones, las cuales son:

■ Función superior

La sintaxis de la función superior es **top(nelx,nely,volfrac,penal,rmin)** donde las variables denotan lo siguiente:

- nelx es el número de elementos finitos en la dirección horizontal.
- nely es el número de elementos finitos en la dirección vertical.
- vol frac es la fracción de volumen en el dominio de diseño.
- penal es la penalización de las densidades intermedias. Una penalización alta hará que la solución sea en blanco y negro, es decir, los elementos finitos estarán llenos o vacíos. Usualmente $\text{penal} = 3$ suficientes. Una penalización $\text{penal} = 1$ significa que no hay penalización de las densidades intermedias.
- rmin es un radio de filtro para un filtro que hace que el diseño sea independiente de la malla.

■ Inicialización

El programa principal empieza distribuyendo el material en partes iguales en el dominio del diseño. Después de otras inicializaciones, el bucle empieza llamando a la subrutina del elemento finito que regresa el desplazamiento del vector U ya que la matriz de rigidez de elemento para el material sólido es la misma para todos los elementos, la subrutina es llamada solo una vez. Después de esto, un bucle sobre todos los elementos determinará la función objetivo y las sensibilidades. Las variables $n1$ y $n2$ indican los números de nodos de elementos superiores izquierdo y derecho y son usados para extraer el vector de desplazamiento U_e del desplazamiento global del vector U . El análisis de sensibilidad es seguido al llamar al filtro de malla independiente y al optimizador. El bucle principal se termina si el cambio en las variables de diseño es menor al 1 por ciento.

■ Aplicación de criterios de optimalidad

Las variables de diseño actualizadas se encuentran en el optimizador. Sabiendo que el volumen del material ($\text{sum}(\text{sim}(\text{xnew}))$) es una función decreciente monótona de los multiplicadores de Lagrange (lag), el valor de los multiplicadores de Lagrange que satisface la restricción de volumen se puede encontrar mediante un algoritmo de bifurcación. Este algoritmo se inicializa adivinando un valor $l1$ inferior y un $l2$ superior para el lagrangiano. El tamaño de los elementos finitos se define como una vez por 1 unidad.

■ Filtro de malla independiente

Representa la implementación de Matlab de. Es importante notar que no todos los elementos en el dominio del diseño son buscados con el fin de encontrar los elementos que se encuentran dentro del radio r_{\min} pero solo los que se encuentren dentro de un cuadrado con lados de longitud dos veces redondas (r_{\min}) alrededor del elemento considerado. Al seleccionar menos r_{\min} que uno en la rutina, las sensibilidades filtradas serán iguales a las sensibilidades originales que hacen el filtro inactivo.

■ Análisis de elemento finito - Matriz de rigidez

En el código de elemento finito la matriz global rígida se forma por un bucle sobre todos los elementos. Como fue en el caso del programa principal, las variables $n1$ y $n2$ denotan los números de nodos de elementos superior izquierdo y derecha y son usados para insertar la matriz de rigidez de los elementos en los lugares correctos en la matriz global.

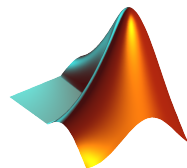


Figura 7: Matlab

Con esto, el programa realiza una optimización con forma, cargas y apoyos predefinidos, por lo que es necesario realizar cambios en el programa para realizar la optimización al marco de una bicicleta.

Primeramente, para realizar la optimización se coloca el apoyo y la fuerza correspondiente observada en la figura 6, se considera solo la carga ejercida por el manubrio y tomamos como fijo el bastidor trasero. Para representar esto en el código se editan las siguientes líneas:

- 78 $F(2,1) = -1$;
- 79 `fixeddofs = union([1:2:2*(nely+1)],[2*(nelx+1)*(nely+1)]);`

Por:

- 78 $F(2,1) = 1$;
- 79 `fixeddofs = 2*nelx*(nely+1)+1:2*(nelx+1)*(nely+1);`

Después de esto para representar correctamente la figura 6 hay que generar una región vacía para la rueda delantera, para corregir esto llamaremos pasivos a los elementos finitos que generaran el vacío y definamos una matriz con ceros en los elementos libres y unos en los pasivos, generando así el siguiente código que se coloca entre la línea 5 y 6:

```
for ely = 1:nely
    for elx = 1:nelx
        if ((elx)^2+(ely-nely)^2) < (0.612*nelx)^2
            passive(ely,elx) = 1;
        else
            passive(ely,elx) = 0;
        end
    end
end
x(find(passive))=0.001;
```

El último comando inicializa todos los elementos en la región vacía al valor bajo 0.001.

Y por último hay que hacer cambios en las variables y funciones que almacenan los valores iniciales para así poder realizar la modificación de generar un vacío, se modifican las siguientes líneas:

- 27 `[x] = OC(nelx,nely,x,vfrac,dc,passive);`
- 37 `function [xnew]=OC(nelx,nely,x,vfrac,dc,passive)`
- 41 `xnew(find(passive)) = 0.001;`

7. Resultados

Una vez realizados hay que comprobar que el vacío que se genero representa correctamente a la figura 6, para esto, en la consola de Matlab se introduce la siguiente línea del código **top(20,20,1,0.1,1.5)**, donde definimos los valores de la función superior pero solo generamos una iteración que nos permitirá ver la geometría generada.

Como podemos observar, en la figura 8 se generó correctamente el espacio vacío que representa el espacio que ocupara una parte de la rueda delantera.



Figura 8: Geometría generada

Con esto comprobado se procede a introducir nuevamente la línea de código, pero ahora modificamos volfrac y penal **top(20,20,0.33,3,1.5)** para que se produzcan las iteraciones y con los parámetros especificados al llamar la instrucción lo que se hace es discretizar 20 veces en 20 elementos finitos el dominio de diseño.

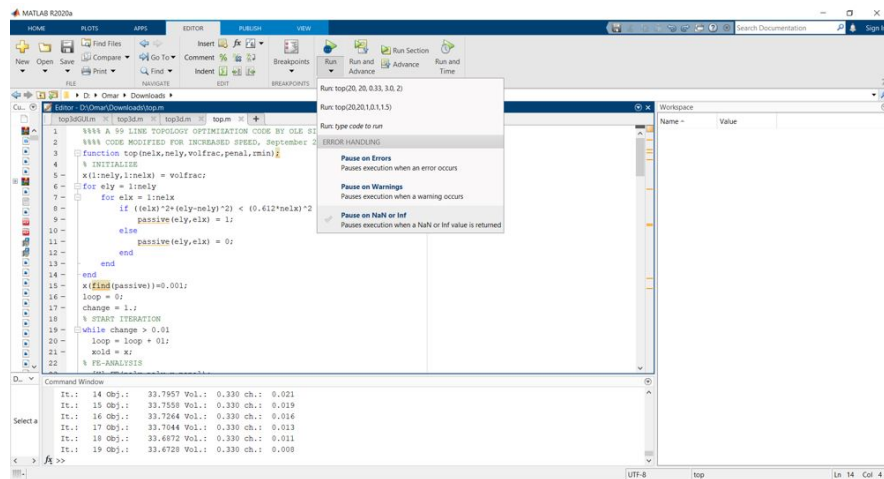


Figura 9: Modificación de función superior

Realizando la operación, finalmente obtenemos la optimización de la geometría, como podemos observar en la figura 10 el código propuesto determinó la distribución de masa más eficiente para cumplir exactamente con los requisitos de diseño impuestos a la pieza, tiene en cuenta el espacio permitido, las restricciones y condiciones de carga de la pieza y las tensiones máximas admisibles en el material, por lo que con esto se obtiene el resultado esperado y se da por concluida la optimización.

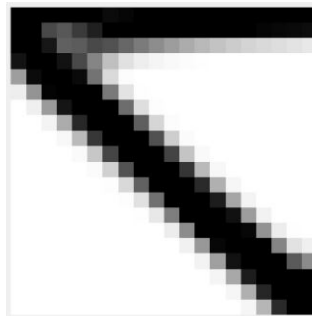


Figura 10: Geometría optimizada

A continuación, se anexa el código final después de las modificaciones realizadas.

```

%%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 %%%
function top(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
for ely = 1:nely
    for elx = 1:nelx
        if ((elx)^2+(ely-nely)^2) < (0.612*nelx)^2
            passive(ely,elx) = 1;
        else
            passive(ely,elx) = 0;
        end
    end
end
x(find(passive))=0.001;
loop = 0;
change = 1.;
% START ITERATION
while change > 0.01
    loop = loop + 01;
    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;
        Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
        c = c + x(ely,elx)^penal*Ue'*KE*Ue;
        dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
    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,passive);
% 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
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
l1 = 0; l2 = 100000; move = 0.2;
while ((l2-l1)/l2 > 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
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-floor(rmin) , 1):min(i+floor(rmin) , nelx)
            for l = max(j-floor(rmin) , 1):min(j+floor(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
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), 1); U = zeros(2*(nely+1)*(nelx+1), 1);
for elx = 1:nelx
    for ely = 1:nely
        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
end
% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
F(2,1) = 1;
fixeddofs = 2*nelx*(nely+1)+1:2*(nelx+1)*(nely+1);
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs , fixeddofs);
% SOLVING
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)];

```

8. Conclusiones

- Omar Alejandro Gutierrez Carrizales:

Con lo realizado en esta segunda práctica se reforzaron los conocimientos adquiridos en la práctica anterior referente a la optimización topológica, pues en este caso se necesitaba realizar la optimización a un marco de bicicleta. Y nuevamente, gracias a lo realizado se recalca la importancia de la implementación de las optimizaciones hoy en día dentro de la industria, desde el diseño automotriz, en la industria para mejorar las propiedades de alguna pieza de maquinaria, en el diseño y elaboración de prótesis, entre muchas otras áreas de desarrollo. Este tipo de optimizaciones ayuda sobre todo a ahorrar gastos al momento de elaboración de las piezas, debido a que es mucho menor el material necesario para las elaboraciones de las piezas. Por otro lado, la elaboración del código resultó mucho más sencilla, debido a que se utilizó el mismo código de la práctica pasada y únicamente fue necesario el realizar ciertos ajustes a el código para poder ajustar nuestra geometría a el marco de la bicicleta y de esta manera cumplir con el objetivo de la práctica.

- Pietro Augusto Castañeda Moreno:

La optimización de un diseño parte de un análisis de sus elementos, cuyo cálculo puede tomar una cantidad finita de valores, que, entre mayor sea, mayor precisión tendrá el estudio, pero a un alto costo de procesamiento computacional.

El programa utilizado en matlab no cuenta con gran precisión, y las "definiciones" que pueden determinarse en el mismo son inexactas, pero sirven para hacer una aproximación de un modelo, que si bien no será igual, bien puede desempeñarse a términos prácticos de una manera casi igual, o idónea.

La Optimización Topológica y de Forma, o en su defecto, el Diseño Generativo en sí, permite el aprovechamiento de recursos, a su vez que se pueden superar limitaciones propias del medio o necesarias para el funcionamiento óptimo de un mecanismo, sin duda herramientas que cada vez ganan mayor peso en la industria de la actualidad.

- Luis Lauro García Hernández:

En esta práctica vimos más que nada sobre la optimización topológica así mismo sobre la optimización de forma y con ello pude aprender el cómo analizar un componente una pieza mecánica y después viendo bien el análisis de elemento finito poder ver y analizar bien la pieza, esto ayuda mucho cuando se necesita resolver problemas estructurales en este caso piezas mecánicas todo esto lo podemos aplicar en la biomecánica a la hora de hacer una prótesis así poder ver el material si tiene complicaciones, si aguanta bien, etc. Por eso es importante bien entender estos conceptos para poder realizar bien una prótesis.

Referencias

- [1] Escuela La Bicicleta. Geometría de un cuadro de bicicleta. Recuperado de: <https://labicicleta.net/escuela/geometria-de-un-cuadro-de-bicicleta/>, Junio 2019. Acceso: 18-10-2022.
- [2] O. Begambre Carrillo C. Millán Páramo. Solución de problemas de optimización topológica empleando el algoritmo simulated annealing modificado. Recuperado de: <https://www.sciencedirect.com/science/article/pii/S0213131515000127>, 2016.
- [3] PEDALIA. Conoce el cuadro de tu bicicleta. Recuperado de: <https://pedalia.cc/conoce-cuadro-bicicleta/>, Diciembre 2020. Acceso: 18-10-2022.