

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA MECÁNICA



UNIVERSITAS  
Miguel Hernández



"DISEÑO DE UN SENSOR DE FUERZAS  
TRIDIMENSIONAL BASADO EN UNA  
PLATAFORMA PARALELA"

TRABAJO FIN DE GRADO

Septiembre 2024

AUTOR: Pedro Castro Benito

DIRECTOR/ES: Adrián Peidro Vidal

# ÍNDICE GENERAL

<b>1. INTRODUCCIÓN</b> .....	<b>1</b>
1.1. ANTECEDENTES .....	1
1.1.1 <i>Aplicaciones y clasificación</i> .....	1
1.2. MOTIVACIÓN .....	2
1.3. OBJETO DEL TRABAJO.....	3
1.4. ESTRUCTURA DE LA MEMORIA.....	3
<b>2. DISEÑO TEÓRICO DEL SENSOR</b> .....	<b>5</b>
2.1. ARQUITECTURA DE LA PLATAFORMA .....	6
2.2. FUNDAMENTOS TEÓRICOS.....	7
2.3. PROGRAMACIÓN EN MATLAB .....	10
2.4. CONFIGURACIONES .....	14
2.4.1. <i>Modelo comercial</i> .....	15
2.4.2. <i>Modelo tetraédrico</i> .....	19
2.4.3. <i>Conclusiones</i> .....	21
<b>3. MODELADO 3D DEL SENSOR</b> .....	<b>22</b>
3.1. PIEZAS PARA FABRICACIÓN EN TALLER.....	22
3.2. CÉLULAS DE CARGA .....	24
3.3. DISEÑO Y ENSAMBLAJE .....	25
3.3.1. <i>Modelo comercial</i> .....	25
3.3.2. <i>Modelo tetraédrico</i> .....	25
3.4. SIMULACIONES .....	27
3.4.1. <i>Modelo comercial</i> .....	28
3.4.2. <i>Modelo tetraédrico</i> .....	31
<b>4. CONCLUSIONES</b> .....	<b>34</b>
<b>5. ANEXOS</b> .....	<b>35</b>
5.1. CÓDIGO DE LA FUNCIONES EN MATLAB.....	35
5.2. ARCHIVOS DE INVENTOR .....	43
5.3. HOJA TÉCNICA DE LA CÉLULA DE CARGA .....	46

# 1. INTRODUCCIÓN

## 1.1. ANTECEDENTES

Las plataformas paralelas son estructuras mecánicas que utilizan múltiples brazos o elementos de soporte para mover una plataforma en el espacio tridimensional. Estas plataformas son muy valoradas en diversas aplicaciones debido a su alta precisión y rigidez.

Estas plataformas están en constante evolución, gracias a los avances en materiales, control y un diseño más refinado que aumentan su eficiencia y efectividad.

### 1.1.1. Aplicaciones y clasificación

Entre las plataformas paralelas más actuales podemos distinguir los siguientes modelos y sus aplicaciones:

**Delta Robots:** Utilizados principalmente en la industria de la automatización y el embalaje, los robots Delta son conocidos por su alta velocidad y precisión en el movimiento. Son ideales para tareas de pick-and-place y ensamblaje.

**Stewart Platforms:** También conocidas como plataformas hexapod, estas estructuras se utilizan en simuladores de vuelo y en aplicaciones de ingeniería para pruebas de vibración y movimiento. Su diseño permite un control preciso en seis grados de libertad.

**Parallel Kinematic Machines (PKM):** Estas máquinas se utilizan en la fabricación de alta precisión, como en la industria aeroespacial y automotriz. Su diseño permite un control preciso de la herramienta, lo que es esencial para procesos de mecanizado.

**Plataformas de Manipulación:** En el ámbito de la robótica, se están desarrollando plataformas paralelas que permiten la manipulación de objetos en entornos complejos, como en la cirugía robótica o en la exploración espacial.

**Plataformas de Medición:** Se utilizan en metrología para medir fuerzas y momentos en diferentes direcciones. Estas plataformas son esenciales en la investigación y desarrollo de nuevos materiales y estructuras.

**Plataformas de Realidad Virtual y Aumentada:** Algunas plataformas paralelas se están utilizando para crear experiencias inmersivas en entornos de realidad virtual, permitiendo un movimiento más natural y realista.

## 1.2. MOTIVACIÓN

Diseñar un sensor de fuerzas tridimensional basado en una plataforma paralela tiene varios motivos interesantes y beneficiosos. En primer lugar, este tipo de sensor permite medir fuerzas en múltiples direcciones simultáneamente, lo que es crucial en aplicaciones donde se requiere un análisis preciso de las cargas y tensiones en un objeto o sistema.

Además, como se mencionó anteriormente, estas plataformas paralelas ofrecen una alta rigidez y estabilidad, lo que se traduce en mediciones más precisas y confiables que las que se podrían obtener con otros sensores. Esto es especialmente importante en campos como la robótica, la biomecánica y la ingeniería estructural, donde la precisión de las mediciones de fuerzas puede influir en el rendimiento y la seguridad de los sistemas testeados.

Otro motivo para el desarrollo de este tipo de sensor es la posibilidad de miniaturización. Las plataformas paralelas pueden ser diseñadas para ocupar un reducido espacio, lo que facilita su integración en dispositivos más pequeños o su uso en entornos de espacio limitado.

Finalmente, el desarrollo de un sensor de este tipo puede abrir nuevas oportunidades en investigación y desarrollo, permitiendo innovaciones en el diseño de productos y en la mejora de procesos industriales, ayudando a impulsar la innovación en diversas aplicaciones.

## 1.3. OBJETO DE TRABAJO

El principal objetivo de este trabajo es diseñar un sensor de fuerzas tridimensional basado en una plataforma paralela. Diseñar un sensor implica considerar varias características clave para asegurar su funcionalidad, precisión y aplicabilidad. Las características deseables que debería cumplir el sensor a desarrollar, son las siguientes:

- ❖ **Precisión y Sensibilidad:** El sensor debe ser capaz de medir fuerzas con alta precisión y sensibilidad en las tres dimensiones (x, y, z).
- ❖ **Rango de Medición:** Definir un rango adecuado de fuerzas que el sensor puede medir sin saturarse o perder precisión.
- ❖ **Linealidad:** Asegurar que la respuesta del sensor sea lineal dentro del rango de operación establecido, lo que facilitará la interpretación de los datos y mejorará la calibración.
- ❖ **Robustez y Durabilidad:** Diseñar el sensor para que sea resistente a las acciones externas aplicadas a este dependiendo de su aplicación.
- ❖ **Tamaño y Peso:** Minimizar el tamaño y peso del sensor para facilitar su integración en diferentes sistemas o dispositivos.

- ❖ **Facilidad de Integración:** Diseñar el sensor con interfaces estándar que faciliten su integración con otros sistemas electrónicos o plataformas de control.
- ❖ **Costo-Efectividad:** Considerar el costo de producción y materiales para hacer que el sensor sea económicamente viable para su uso comercial o industrial.
- ❖ **Calibración y Mantenimiento:** Facilitar procesos de calibración sencillos y efectivos, así como minimizar la necesidad de mantenimiento frecuente.
- ❖ **Compatibilidad con Software:** Asegurar que el sensor pueda comunicarse fácilmente con software de análisis o control, permitiendo una fácil visualización e interpretación de los datos recogidos.
- ❖ **Multifuncionalidad:** Si es posible, diseñar el sensor para que pueda medir no solo fuerzas, sino también otros parámetros relevantes como momentos, ampliando así su utilidad.
- ❖ **Seguridad:** Implementar características que garanticen la seguridad tanto del dispositivo como del usuario durante su operación.

Este trabajo se centrará en la creación de un sensor compacto y robusto con la utilización de células de carga instaladas en las barras. Este sensor debe permitir realizar mediciones precisas hasta una carga máxima de compresión con un valor de 700Kg y no superar los 60x60x60cm.

## 1.4. ESTRUCTURA DE LA MEMORIA

La memoria está compuesta de 5 capítulos principales. El primero de ellos es la introducción que acabamos de leer, exponiendo los antecedentes, motivaciones y el objetivo de este trabajo de fin de grado.

El segundo capítulo trata sobre el diseño teórico del sensor, explicando la arquitectura y las configuraciones adoptadas, al igual que los fundamentos teóricos mediante los cuales se ha realizado la programación en Matlab.

El tercer capítulo está dedicado al modelado 3D del sensor mediante el programa informático Inventor, exponiendo las piezas que componen el sensor y cómo se planea su fabricación. También se mostrarán los resultados de las simulaciones a las que se han sometido los sensores diseñados y se comprobará si cumplen los objetivos.

El cuarto capítulo expondrá las conclusiones extraídas de las simulaciones y se debatirá qué configuración es la más beneficiosa y cuáles podrían ser los inconvenientes de dicho diseño.

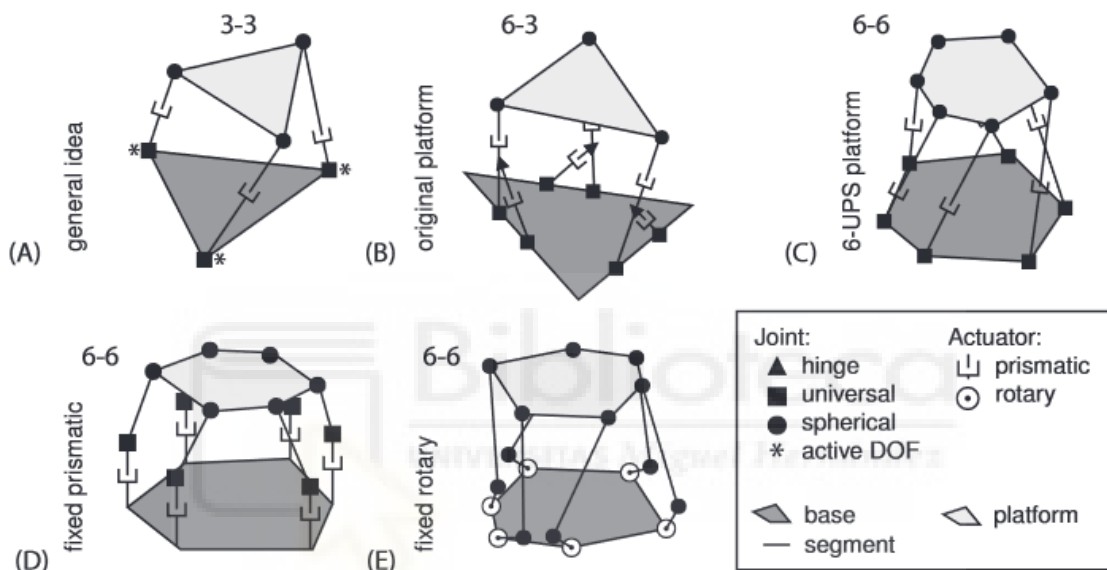
Finalmente, el quinto capítulo recogerá los anexos de esta memoria, los cuales contendrán los archivos generados en Inventor y la programación realizada en Matlab.



## 2. DISEÑO TEÓRICO DEL SENSOR

### 2.1. ARQUITECTURA DE LA PLATAFORMA

El sensor se diseñará tomando como base una plataforma Stewart, la cual consiste en una plataforma, normalmente triangular, soportada mediante articulaciones esféricas mediante patas de longitud ajustable conectadas a una plataforma inferior fija. La plataforma Stewart dispone de 6 grados de libertad consistente de dos cuerpos rígidos conectados mediante 6 restricciones lineales o angulares entre 6 pares de puntos.



**Figura 1.** Diferentes arquitecturas de Plataforma Stewart. Fuente: Szufnarowski, F. (2013). Stewart platform with fixed rotary actuators: a low cost design study. *Advances in Medical Robotics, Chapter 4, 1st Ed.*

El sensor estará conformado por dos plataformas con 6 puntos de unión cada una, a las cuales se le acoplarán mediante rótulas esféricas 6 barras de longitud no variable subdivididas en dos segmentos. Entre los dos segmentos que conforman cada barra se situará una célula de carga, la cual será la encargada de medir los esfuerzos aplicados a la plataforma superior.

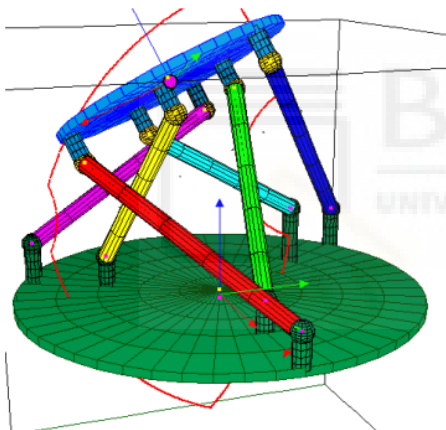
## 2.2. FUNDAMENTOS TEÓRICOS

Para la construcción del sensor y encontrar la configuración óptima que nos asegure una mayor rigidez y precisión es necesario programar una función que reciba como entrada una matriz "a" que contenga 3 filas y 6 columnas, donde cada columna contiene las coordenadas de posición de cada uno de los puntos de anclaje en los que se conectan las barras a la plataforma inferior, una matriz "b" del mismo tamaño y que contenga la misma información, pero refiriéndose a los puntos de anclaje de las barras con la plataforma superior, un vector "p" 3x1 que representa la posición de la plataforma superior respecto a la inferior y finalmente la orientación en forma de ángulos de Euler "alfa", "beta" y "gamma" que denominamos "o".

Las coordenadas de la matriz "b" se escribirán en base a un sistema de referencia local fijado a la plataforma superior mientras que la matriz de posición "a" y el vector "p" estarán definidos respecto al sistema de coordenadas global.

Ejemplo espacial: robot 6UPS  
(hexápodo, plataforma de Stewart)

Dado  $T = \begin{bmatrix} R & p \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$  que codifica la posición y orientación de la plataforma móvil



Para cada pata  $j$ :

$a_j$  : coordenadas de las articulaciones universales (U) en coordenadas globales

$b_j$  : coordenadas de las articulaciones esféricas (S) en coordenadas locales

$B_j$  : coordenadas de las articulaciones esféricas (S) en coordenadas globales

$$\begin{bmatrix} B_j \\ 1 \end{bmatrix} = T \begin{bmatrix} b_j \\ 1 \end{bmatrix} \rightarrow L_j = \|a_j - B_j\|$$

**Figura 2.** Robot hexápodo, cálculo de la longitud de las barras. Fuente: Apuntes de la asignatura de Cinemática de Robots del Máster en Robótica de la UMH

"T" es una matriz 4x4 que se construye juntando una matriz de rotación "R" de tamaño 3x3 y el vector de posición "p" de tamaño 3x1. La matriz de rotación "R" es la siguiente:

$$R = \begin{bmatrix} \cos(\beta) \cos(\gamma) & -\cos(\beta) \sin(\gamma) & \sin(\beta) \\ \cos(\alpha) \sin(\gamma) + \sin(\alpha) \sin(\beta) \cos(\gamma) & \cos(\alpha) \cos(\gamma) - \sin(\alpha) \sin(\beta) \sin(\gamma) & -\sin(\alpha) \cos(\beta) \\ \sin(\alpha) \sin(\gamma) - \cos(\alpha) \sin(\beta) \cos(\gamma) & \cos(\alpha) \sin(\beta) \sin(\gamma) + \sin(\alpha) \cos(\beta) & \cos(\alpha) \cos(\beta) \end{bmatrix}$$

$$p = \begin{bmatrix} px \\ py \\ pz \end{bmatrix}$$

$$o = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$



Con referencia a la figura 2, tenemos la ecuación de la cinemática inversa para cada barra, la cual nos proporciona la longitud de cada barra en función del diseño de la plataforma (dado por  $a_j$  y  $b_j$ ) y por el vector posición “p” y la matriz de rotación “R” de la plataforma superior.

$$\begin{aligned}
 L1 &= \|a1 - B1\| = \text{función de } px, py, pz, \text{ alfa, betta y gamma} \\
 L2 &= \|a2 - B2\| = \text{función de } px, py, pz, \text{ alfa, betta y gamma} \\
 L3 &= \|a3 - B3\| = \text{función de } px, py, pz, \text{ alfa, betta y gamma} \\
 L4 &= \|a4 - B4\| = \text{función de } px, py, pz, \text{ alfa, betta y gamma} \\
 L5 &= \|a5 - B5\| = \text{función de } px, py, pz, \text{ alfa, betta y gamma} \\
 L6 &= \|a6 - B6\| = \text{función de } px, py, pz, \text{ alfa, betta y gamma}
 \end{aligned}$$

$$\begin{bmatrix} L1 \\ L2 \\ L3 \\ L4 \\ L5 \\ L6 \end{bmatrix} = F(px, py, pz, \alpha, \beta, \gamma)$$

Si variamos ligeramente la posición, la orientación o ambas simultáneamente de la plataforma superior, las barras  $L_j$  sufren pequeños estiramientos o contracciones (Representaremos con  $\Delta$  dichos cambios). Estos cambios se relacionan por la matriz Jacobiana inversa según esta ecuación:

$$\Delta \begin{bmatrix} L1 \\ L2 \\ L3 \\ L4 \\ L5 \\ L6 \end{bmatrix} = J * \Delta \begin{bmatrix} px \\ py \\ pz \\ \alpha \\ \beta \\ \gamma \end{bmatrix}$$

Aunque “J” es la matriz Jacobiana inversa, por simplicidad, la denotamos como J a secas. Esta matriz está compuesta de las derivadas parciales, donde cada columna no es más que la derivada de “F” (siendo F la función encargada de calcular la cinemática inversa) con respecto a cada una de las variables.

La ecuación de pequeños incrementos tiene su contraparte o análogo en términos de fuerza con la siguiente expresión:

$$f = -J^T \tau$$

Donde “f” son las fuerzas externas aplicadas a la plataforma superior y  $\tau$  las fuerzas sentidas por las barras del sistema (por tanto, las medidas por las células de carga)

Para escoger la configuración óptima del sensor emplearemos el número de condición de la Jacobiana inversa y la iteración. El número de condición es una medida que se utiliza en matemáticas y en análisis numérico para evaluar la sensibilidad de una solución a un problema frente a cambios en los datos de entrada; por tanto, alterando los valores pre-establecidos de los puntos de unión y las orientaciones de ambas bases.

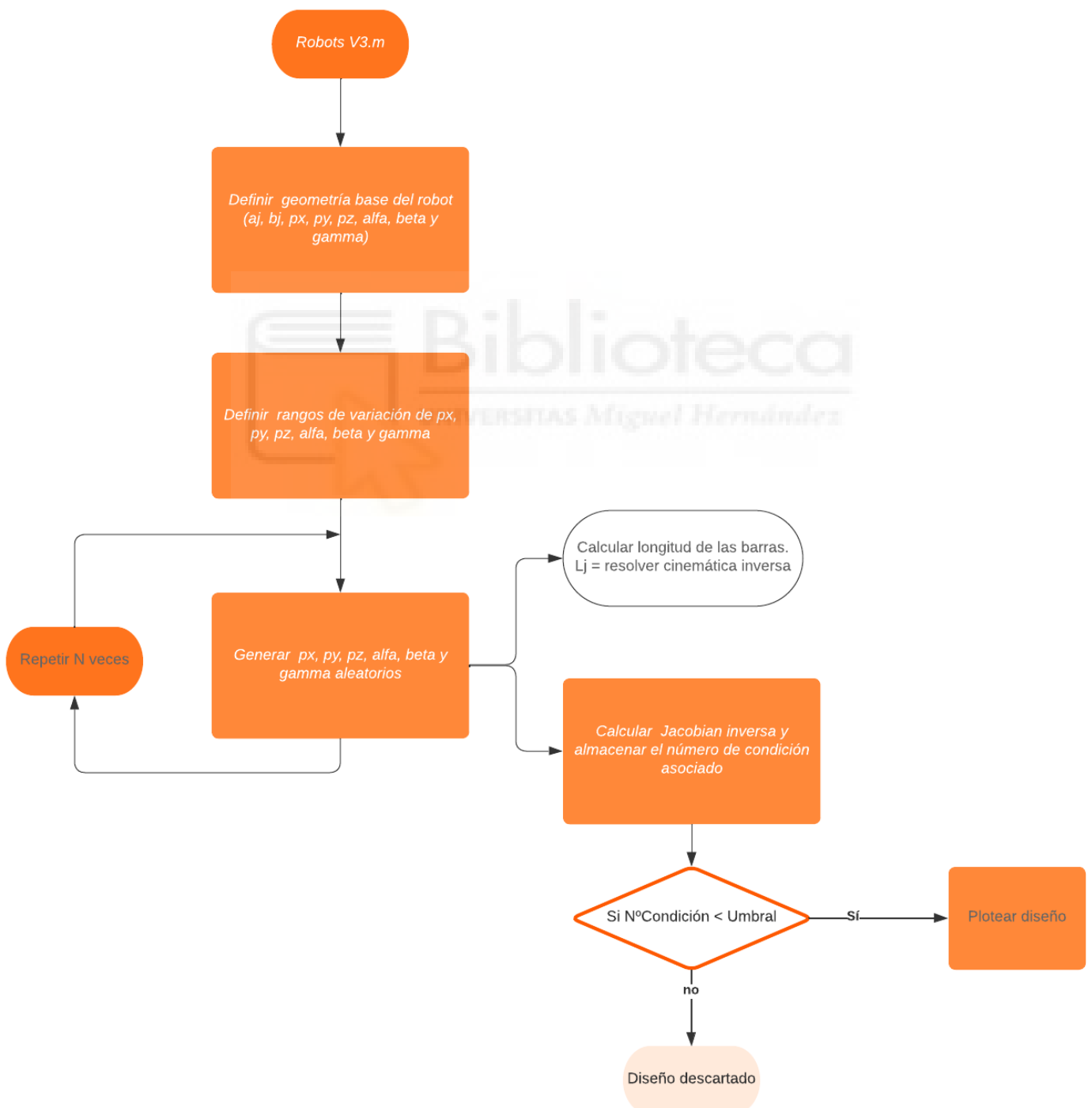
Un número de condición bajo indica que el sistema es más estable y menos sensible a errores, mientras que un número de condición alto sugiere que pequeñas perturbaciones pueden causar grandes cambios en la solución. Por lo general, un número de condición menor a 10 se considera bueno, mientras que un número mayor a 100 puede ser problemático.

El número de condición de la matriz Jacobiana inversa es una medida útil en análisis estático en mecánica porque representa la isotropía de la relación entre las fuerzas aplicadas a la plataforma superior y las fuerzas axiales medidas por las células de carga montadas en las barras del robot. Cuando el número de condición es pequeño, favorece una relación isótropa entre las fuerzas aplicadas a la plataforma superior del sensor y las fuerzas axiales medidas por las células de carga, es decir, el sensor medirá fuerzas externas de forma similar independientemente de la dirección de aplicación de las fuerzas "f". Por el contrario, un número de condición elevado denota proximidad a singularidad, esto presenta las siguientes implicaciones indeseables:

- Podría ocurrir que, aunque las fuerzas "f" sean nulas,  $\tau \neq 0$ . Es decir, las células de carga podrían estar midiendo fuerzas por tensiones internas del sistema.
- No toda fuerza externa f puede ser resistida por el sensor. Podría ocurrir que para ciertas fuerzas o momentos la plataforma ceda (Deje de ser un sistema rígido y obtenga grados de libertad no deseados)

## 2.3. PROGRAMACIÓN EN MATLAB

Los cálculos fueron realizados mediante el programa Matlab, generando 3 funciones encargadas de proporcionar todos los resultados. El primer paso es definir la geometría básica del sensor y el rango de las variaciones que se le aplicarán al diseño base. Posteriormente estableceremos un bucle que se repetirá N veces, aplicando las variaciones expuestas anteriormente y seguidamente calculando la longitud de las barras y la Jacobiana inversa. Finalmente se almacenarán los números de condición y procederemos a realizar un cribado, ploteando los diseños cuyo número de condición este por debajo del umbral deseado.



La función “Robots” contiene toda la información que concierne a los diseños de los sensores y los cálculos necesarios para la obtención de las longitudes de las barras, llamando a la función “cinematicainversa” al igual que el cálculo de la Jacobiana y su condición. A la función se le introducirá como entrada la disposición de ambas plataformas “a” y “b” al igual que el vector posición “p”. La configuración base se verá modificada mediante la alteración del vector posición “p” y la orientación de la base superior. Permitiendo que para una misma configuración obtengamos diseños con similar geometría, pero diferentes entre sí.

Matrices “a” y “b”:

```

a1 = [x1; y1; z1];
a2 = [x2; y2; z2];
a3 = [x3; y3; z3];
a4 = [x4; y4; z4];
a5 = [x5; y5; z5];
a6 = [x6; y6; z6];
b1 = [X1; Y1; Z1];
b2 = [X2; Y2; Z2];
b3 = [X3; Y3; Z3];
b4 = [X4; Y4; Z4];
b5 = [X5; Y5; Z5];
b6 = [X6; Y6; Z6];

```

```

a = [a1 a2 a3 a4 a5 a6];
b = [b1 b2 b3 b4 b5 b6];

```

Vector “p” y orientación “o”:

```

px_min = -0.01;
py_min = -0.01;
pz_min = 0.4-0.01;
px_max = 0.01;
py_max = 0.01;
pz_max = 0.4+0.01;

```

```

alpha_min = -0.15;
beta_min = -0.15;
gamma_min = -0.15;
alpha_max = 0.15;
beta_max = 0.15;
gamma_max = 0.15;

```

```

px = px_min + (px_max - px_min)*rand;
py = py_min + (py_max - py_min)*rand;
pz = pz_min + (pz_max - pz_min)*rand;
alpha = alpha_min + (alpha_max - alpha_min)*rand;
beta = beta_min + (beta_max - beta_min)*rand;
gamma = gamma_min + (gamma_max - gamma_min)*rand;

```

Esta función es la encargada de obtener la Jacobiana de cada diseño, llamando a la función “cinematicainversa” para la realización del cálculo.

Jacobiana “J”:

```
J = [ J1 J2 J3 J4 J5 J6];
```

```
J1=(cinematicainversa(a,b,px+d,py,pz,alpha,beta,gamma) - cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;  
J2=(cinematicainversa(a,b,px,py+d,pz,alpha,beta,gamma) - cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;  
J3=(cinematicainversa(a,b,px,py,pz+d,alpha,beta,gamma) - cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;  
J4=(cinematicainversa(a,b,px,py,pz,alpha+d,beta,gamma) - cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;  
J5=(cinematicainversa(a,b,px,py,pz,alpha,beta+d,gamma) - cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;  
J6=(cinematicainversa(a,b,px,py,pz,alpha,beta,gamma+d) - cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;
```

Finalmente, la función realiza una criba de diseños dependiendo del número de condición atribuido a este, separando los diseños defectuosos de los válidos.

```
por_debajo_de = [];  
umbral = 25;
```

```
for i=1:N  
    if datos{i,4} < umbral  
        por_debajo_de = [por_debajo_de;datos{i,1}];  
  
        px = datos{i,1}(1);  
        py = datos{i,1}(2);  
        pz = datos{i,1}(3);  
        alpha = datos{i,1}(4);  
        beta = datos{i,1}(5);  
        gamma = datos{i,1}(6);  
  
        handle = plotear_robot(a,b,px,py,pz,alpha,beta,gamma);  
        pause(2)  
        delete(handle)  
  
        view(20,40)  
    end
```

La función “plotear\_robot” se encarga de representar gráficamente los sensores creados en la función “Robots”. Para la correcta representación de los robots se emplea el comando “handle”, la matriz de transformación, un cambio de sistema de coordenadas y finalmente el cálculo de la longitud de las barras.

```
handle = [handle,fill3(a(1,:),a(2,:),a(3:),'r')];  
handle = [handle,fill3(B(1,:),B(2,:),B(3:),'b')];
```

```
for i=1:6  
    handle = [handle,plot3([a(1,i) B(1,i)],[a(2,i) B(2,i)],[a(3,i) B(3,i)],'k','LineWidth',2)];  
end
```

Finalmente, la función “cinematicainversa” engloba los cálculos necesarios para la obtención de las longitudes de las barras que componen cada diseño.

Matriz de transformación “T”:

$$p = [px; py; pz];$$

$$R = \begin{bmatrix} \cos(\beta) \times \cos(\gamma) & -\cos(\beta) \times \sin(\gamma) & \sin(\beta) \\ \cos(\alpha) \times \sin(\gamma) + \sin(\alpha) \times \sin(\beta) \times \cos(\gamma) & \cos(\alpha) \times \cos(\gamma) - \sin(\alpha) \times \sin(\beta) \times \sin(\gamma) & -\sin(\alpha) \times \cos(\beta) \\ \sin(\alpha) \times \sin(\gamma) - \cos(\alpha) \times \sin(\beta) \times \cos(\gamma) & \cos(\alpha) \times \sin(\beta) \times \sin(\gamma) + \sin(\alpha) \times \cos(\beta) & \cos(\alpha) \times \cos(\beta) \end{bmatrix}$$

$$T = [R, p; [0 \ 0 \ 0 \ 1]];$$

Cambio de sistema de coordenadas:

$$Aux = T * [b; [1 \ 1 \ 1 \ 1 \ 1]];$$

$$B = Aux(1:3, :);$$

Cálculo de la longitud de las barras:

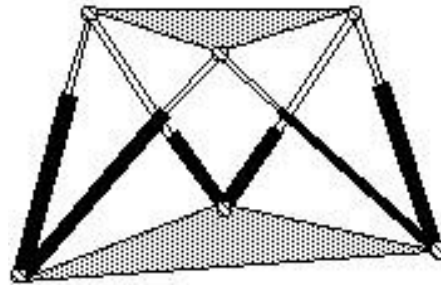
$$L = vecnorm(a-B)'$$

El código desarrollado de cada una de las funciones se colocará en un anexo al final de la memoria.

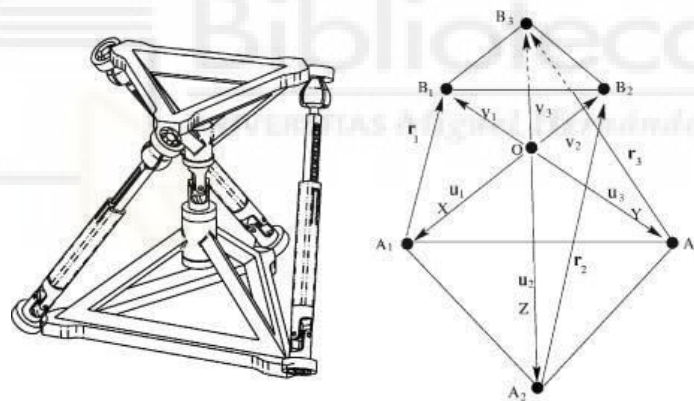


## 2.4. CONFIGURACIONES

Se han testado dos configuraciones principales para el sensor. La configuración que hemos llamado “Comercial” que debe su nombre al ser la disposición más común que se puede encontrar en el mercado (**Figura 3**), y la configuración “Tetraédrica” que ofrece un diseño inusual y complejo (**Figura 4**). Lo que se observa en la **Figura 4** es una base inferior tetraédrica, que nosotros hemos adaptado considerando que 3 de las aristas del tetraedro inferior son barras que incluirán células de carga.



**Figura 3.** Configuración “Comercial”. Fuente: Researchgate - Control visual de robots paralelos: análisis, desarrollo y aplicación a la plataforma RoboTennis.



**Figura 4.** Configuración “Tetraédrica”. Fuente: ASME

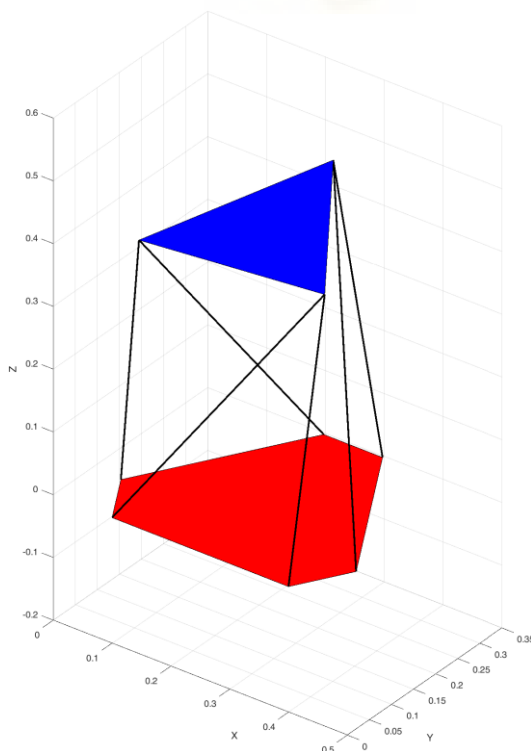
### 2.4.1. Modelo “Comercial”

Esta configuración dispone de una base inferior hexagonal con lados iguales 3 a 3 en cuyos vértices se han situado rotulas esféricas para el asentamiento de las barras. Los lados grandes miden 30cm mientras que los cortos presentan una longitud de 10cm. La plataforma superior, a diferencia de la inferior, tiene una geometría triangular equilátera de 30cm de lado y dos barras confluyen en cada uno de sus vértices, acopladas a rotulas de igual manera que en la plataforma inferior.

Al modelo “comercial” nominal mostrado en la figura 5 se le aplicarán variaciones aleatorias con un valor de  $\pm 0.01$  en los tres ejes, con la diferencia que en el eje Z la altura mínima será de 0’39m. Por el contrario, los ángulos alpha, betta y gamma presentan una desviación de  $\pm 0.15$  radianes. El valor final de cada iteración se calcula de la siguiente forma, en donde rand es un número comprendido entre 0 y 1.

$$\begin{aligned} px &= px\_min + (px\_max - px\_min)*rand; \\ py &= py\_min + (py\_max - py\_min)*rand; \\ pz &= pz\_min + (pz\_max - pz\_min)*rand; \\ alpha &= alpha\_min + (alpha\_max - alpha\_min)*rand; \\ beta &= beta\_min + (beta\_max - beta\_min)*rand; \\ gamma &= gamma\_min + (gamma\_max - gamma\_min)*rand; \end{aligned}$$

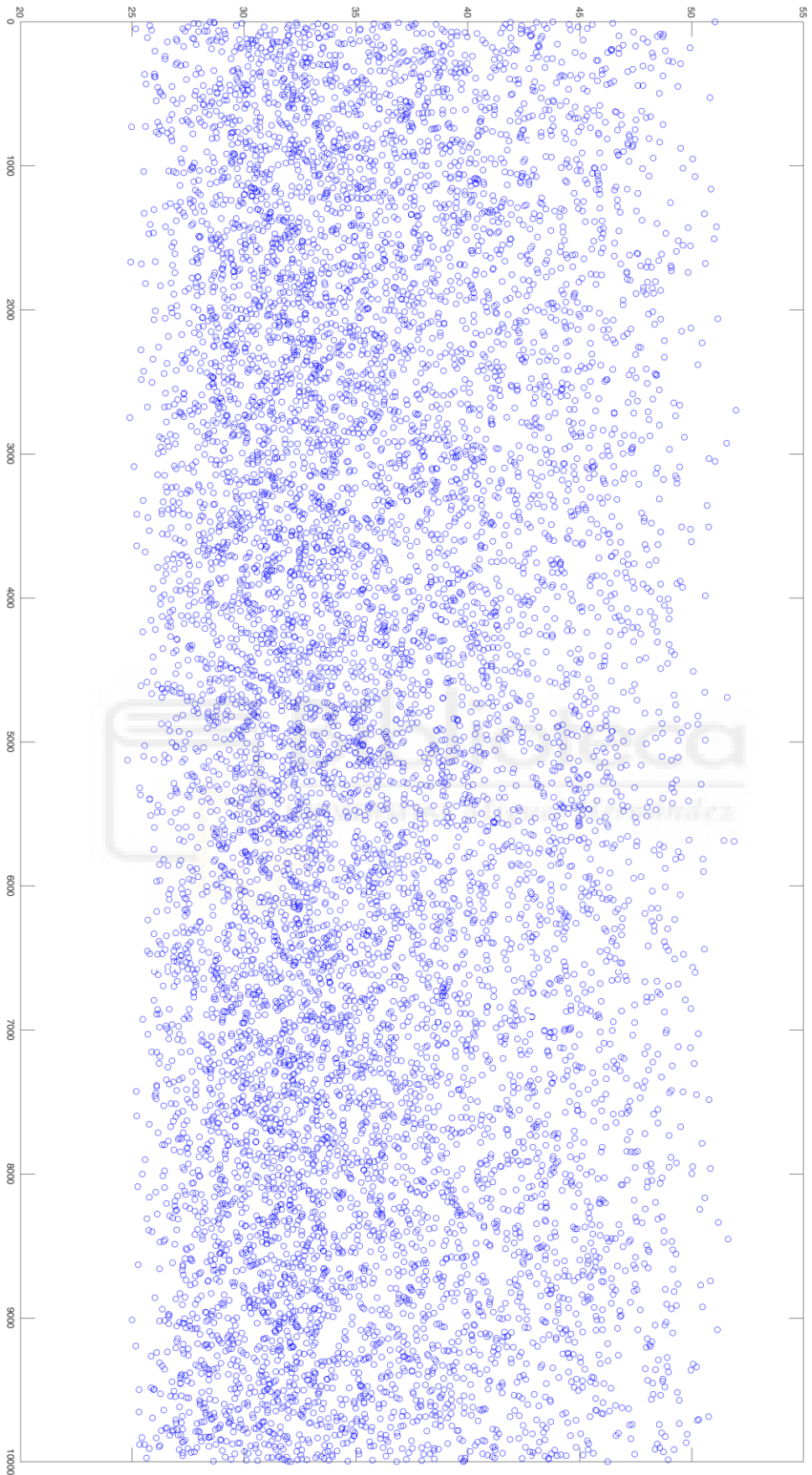
Tras 10000 iteraciones aleatorias descritas en la sección anterior, se obtienen los números de condición mostrados en la figura 6, que como se observa en dicha figura, alcanzan un valor mínimo de 25, sin embargo, este modelo presenta una gran complicación a la hora de su montaje debido a que dos barras confluyen en un único vértice.



**Figura 5.** Configuración teórica “Comercial”.  
Fuente: Elaboración propia



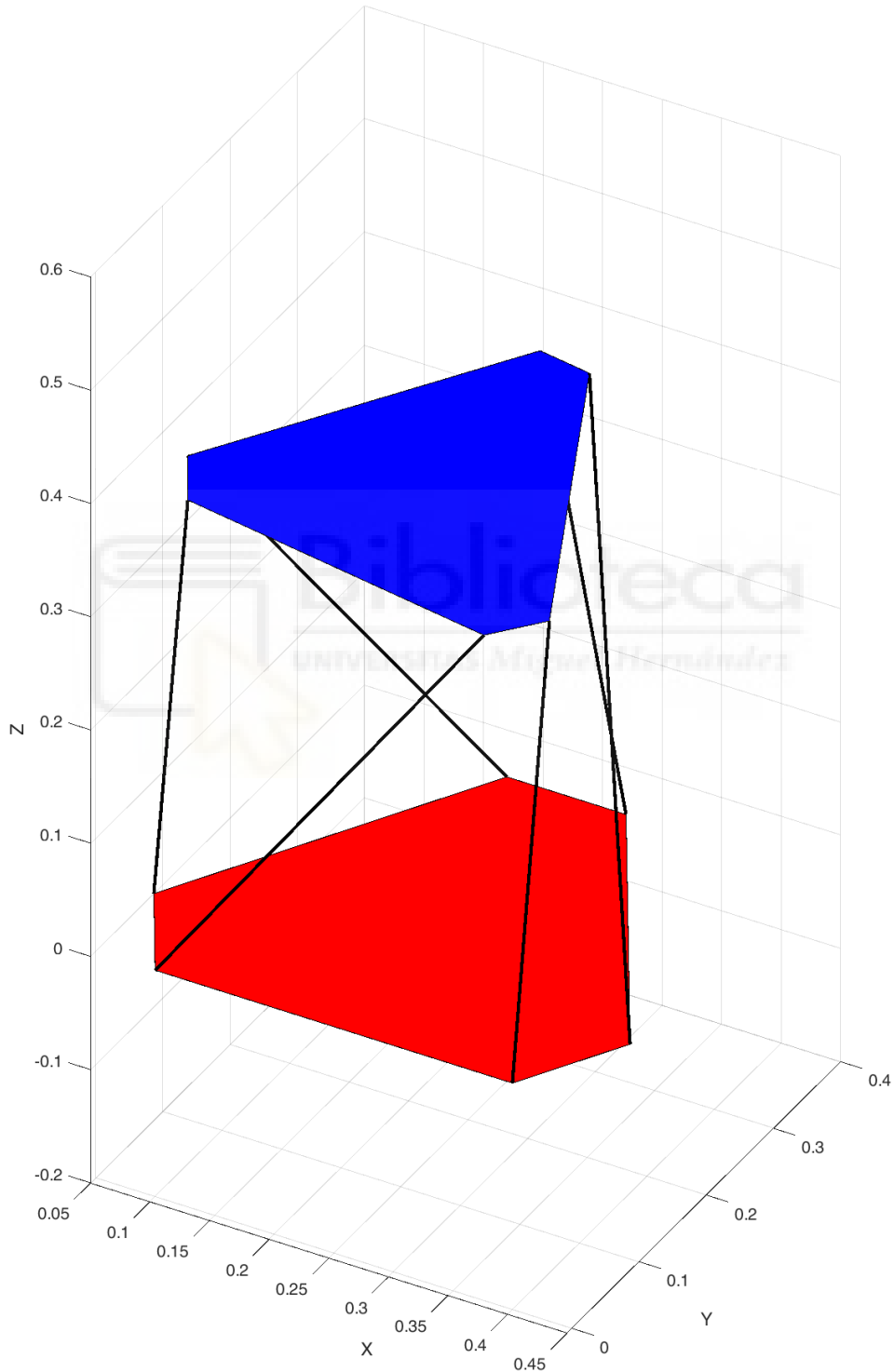
**LEYENDA**  
-Eje de ordenadas: Número de condición  
-Eje de abscisas: Prueba Nº



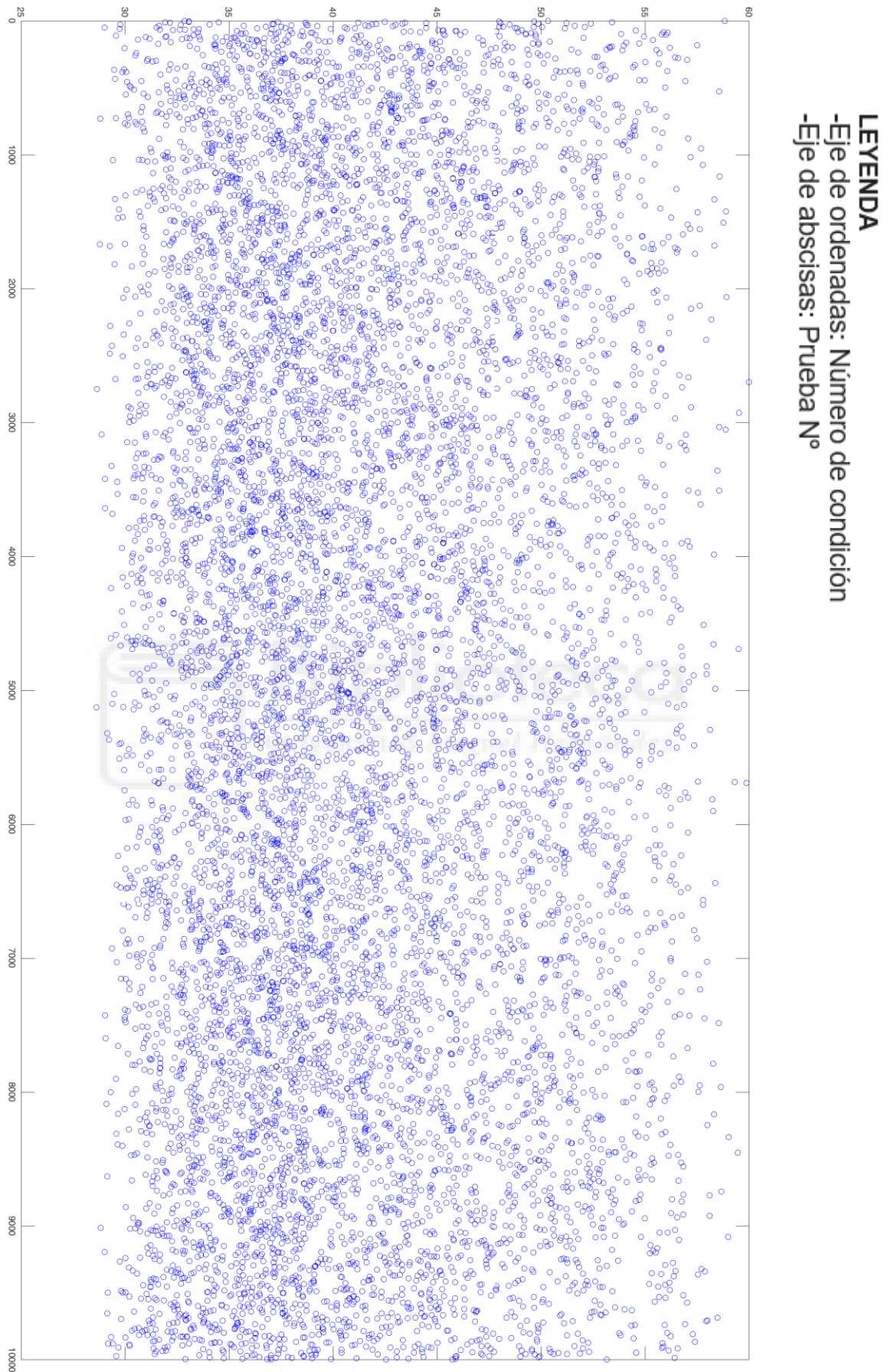
**Figura 6.** Número de condición de la configuración teórica "comercial". Fuente: Elaboración propia

Se decidió modificar la plataforma superior a un hexágono similar al de la plataforma inferior, únicamente dejando la separación mínima necesaria (alrededor de 4cm) para el montaje de dos rótulas, véase la figura 7.

Este cambio produjo un cambio en el número de condición mínimo, el cual tras 10000 nuevas iteraciones aumentó de 25 a 28'6 como se muestra en la figura 8.



**Figura 7.** Configuración real "comercial". Fuente: Elaboración propia



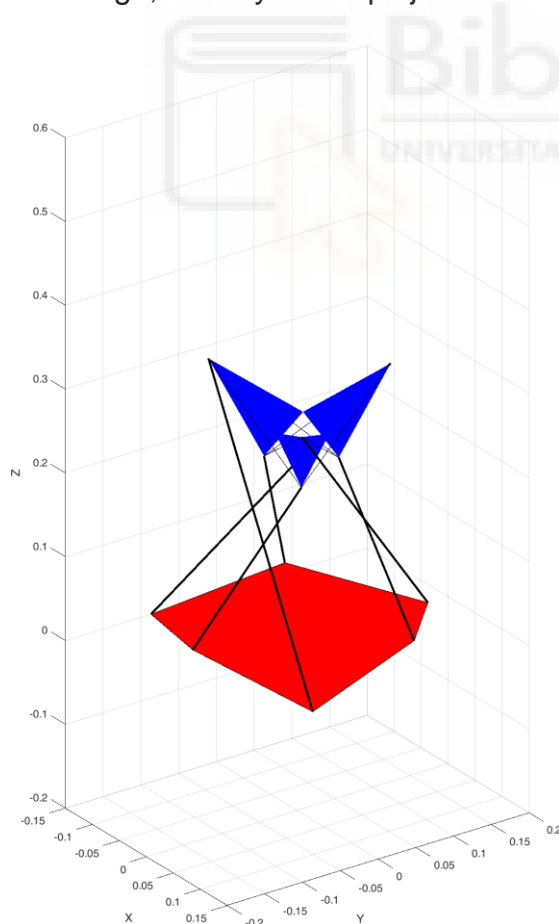
**Figura 8.** Número de condición de la configuración real “comercial”. Fuente: Elaboración propia

## 2.4.2. Modelo “Tetraédrico”

Esta configuración, véase figura 9, dispone de una base hexagonal, con tres de sus vértices elevados de forma intermitente respecto a los otros tres. Sin embargo, el cambio más significativo respecto a la configuración “comercial”, se encuentra en la base superior, la cual ahora tiene una geometría tetraédrica. La base superior consta de tres vértices superiores donde se acoplarán las barras más largas, que, a su vez, estarán conectadas a los vértices sin elevación de la base inferior. La porción inferior del tetraedro fue ‘cortada’ para obtener una sección triangular y por tanto poder colocar una rótula por vértice. Las barras más cortas se conectarán aquí y a los vértices elevados de la base inferior.

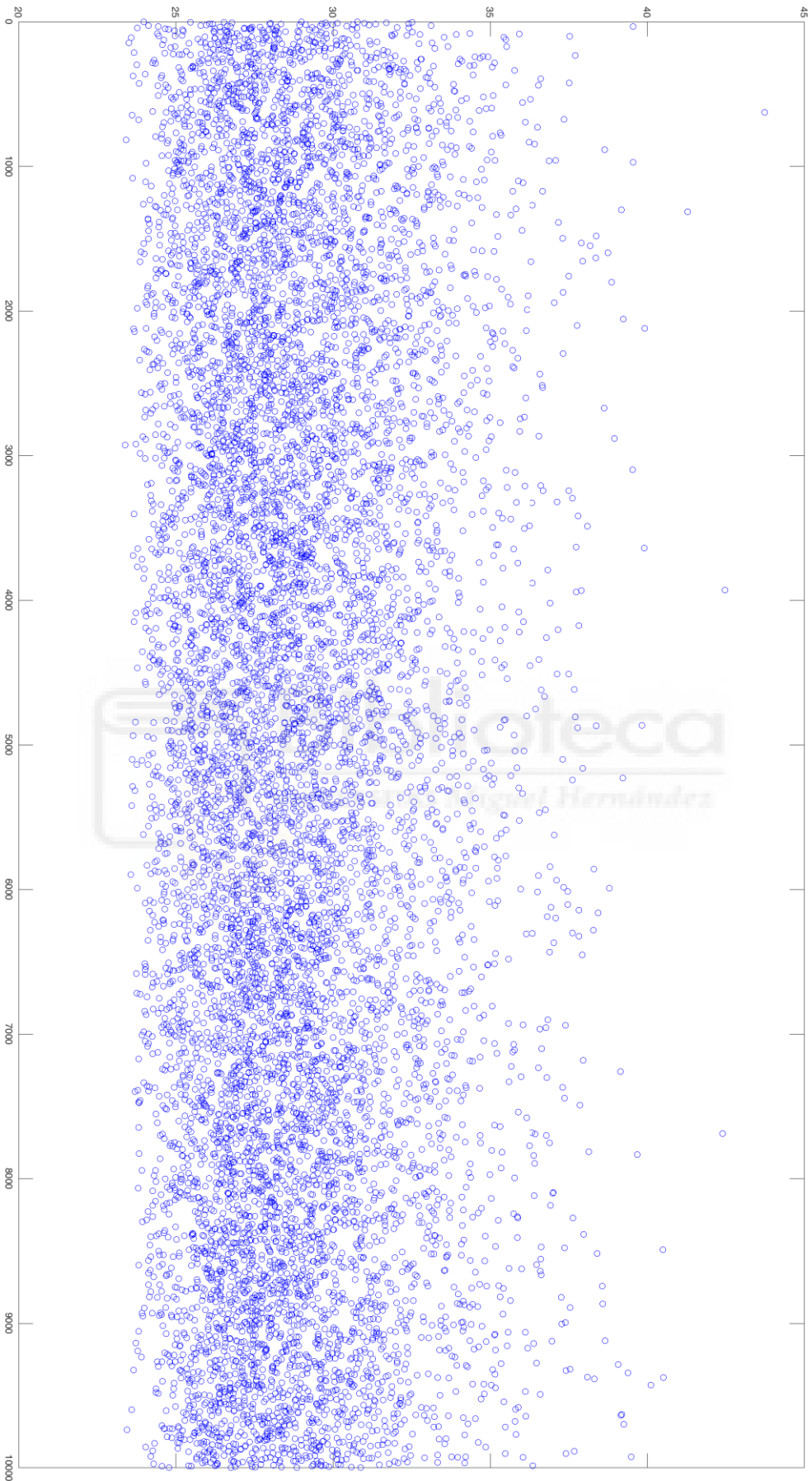
Al modelo “tetraédrico” mostrado se le aplicarán variaciones aleatorias de igual manera que al “comercial”, con un valor de  $\pm 0.01$  en los tres ejes, con la diferencia que en el eje Z la altura mínima será de 0’29m. Los ángulos alpha, betta y gamma presentan una desviación de  $\pm 0.15$  radianes al igual que en el modelo “comercial” y valor final de cada iteración se calcula de igual forma.

Tras 10000 iteraciones aleatorias, se obtienen los números de condición mostrados en la figura 10, que como se observa en dicha figura, alcanzan un valor mínimo de 23’39, siendo inferior al obtenido mediante la configuración “comercial”, sin embargo, su mayor complejidad dificultará su construcción.



**Figura 9.** Configuración “tetraédrica”. Fuente: Elaboración propia

**LEYENDA**  
-Eje de ordenadas: Número de condición  
-Eje de abscisas: Prueba N°



**Figura 10.** Número de condición de la configuración “tetraédrica”. Fuente: Elaboración propia

### 2.4.3. Conclusiones

Tras decenas de configuraciones y cientos de pruebas se ha podido llegar a las siguientes conclusiones. El cruzamiento de barras entre vértices ayuda a la reducción del número de condición a costa de aumentar la complejidad del diseño. Por otro lado, si se mantiene una relación de aspecto aproximada de 3/4 entre la base inferior y la altura del sensor, es decir, las barras son  $\frac{3}{4}$  del ancho de la base, el número de condición obtenido resulta menor. De igual manera, si la plataforma superior presenta una orientación diferente respecto a la inferior el número de condición obtiene un valor menor.

Con todo esto en mente, se puede llegar a la conclusión que una asimetría controlada y un diseño proporcionado favorece un número de condición bajo, lo que explicaría por qué la configuración “tetraédrica” presenta un número de condición inferior respecto a la “comercial”. La configuración “tetraédrica” presenta una geometría asimétrica al igual que barras de menor longitud en comparación a la “comercial”, presentándose como la favorita por el momento.



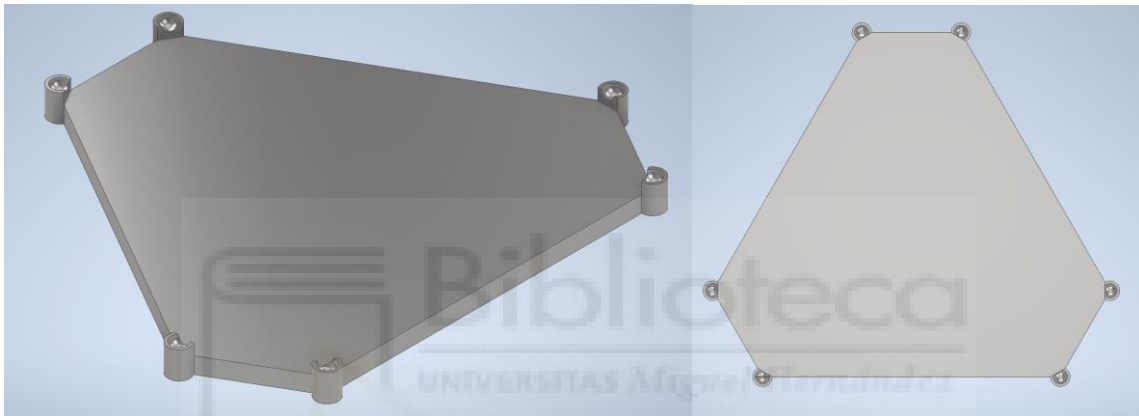
### 3. MODELADO 3D DEL SENSOR

Tomaremos los mejores diseños de ambas configuraciones y modelaremos en 3D sus componentes y posteriormente los uniremos en un ensamblaje. Estos ensamblajes serán testeados mediante simulaciones con el objetivo de ver cómo se comporta cada configuración bajo las cargas máximas de diseño.

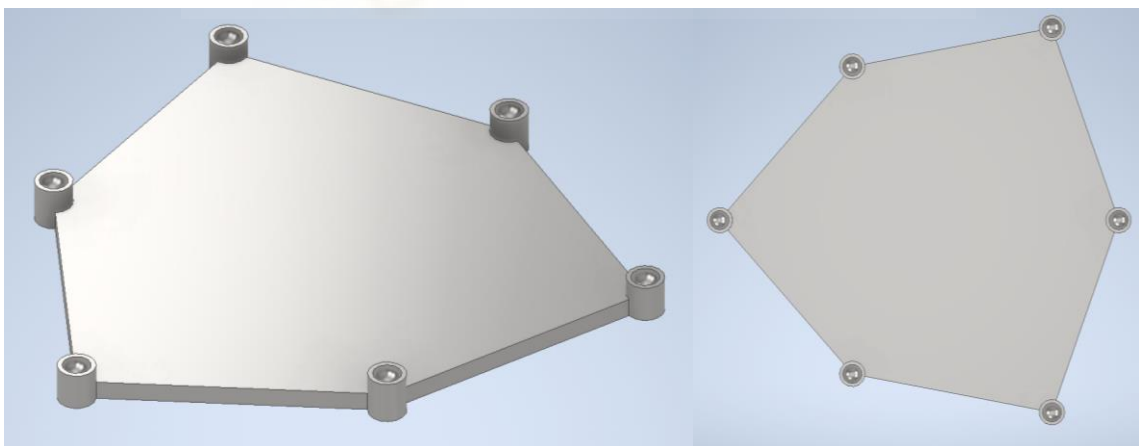
#### 3.1. PIEZAS PARA FABRICACIÓN EN TALLER

Se ha realizado el diseño de 3 piezas con el objetivo de ser fabricadas mediante mecanizado, fresado y corte térmico. Estas piezas son:

- **Base del sensor**, a la que se le conectarán las barras metálicas, con rótulas esféricas cuya cavidad tiene un diámetro de 1'5cm



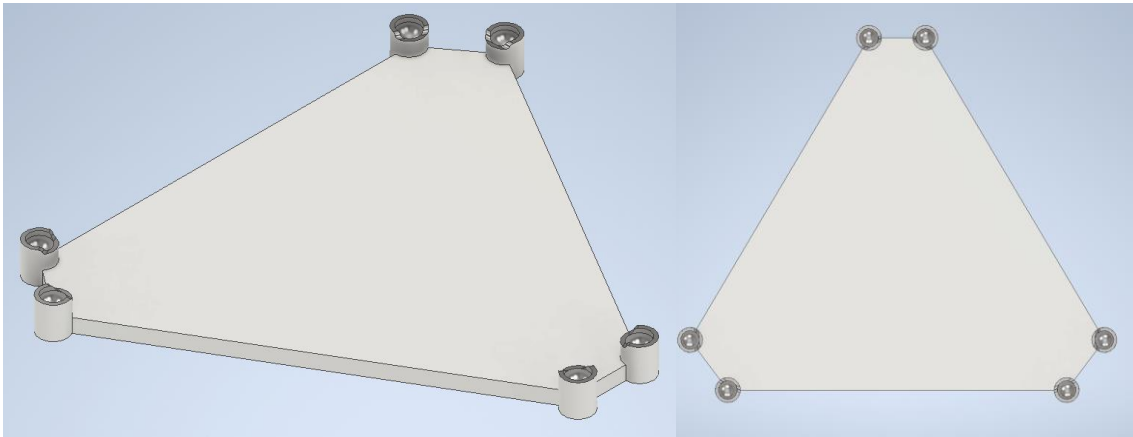
*Figura 11. Base configuración "Comercial". Fuente: Elaboración propia*



*Figura 12. Base configuración "Tetraédrica". Fuente: Elaboración propia*

- **Barras**, de longitud variable según la posición que ocupen en cada configuración. Cada sensor dispone de un total de 12 barras de 1'2cm de diámetro, con uno de sus extremos con métrica M12 y el otro con una esfera de 1'5cm. Para su diseño se ha tenido en cuenta que cada célula de carga ocupaba 6'4cm del total de esta.

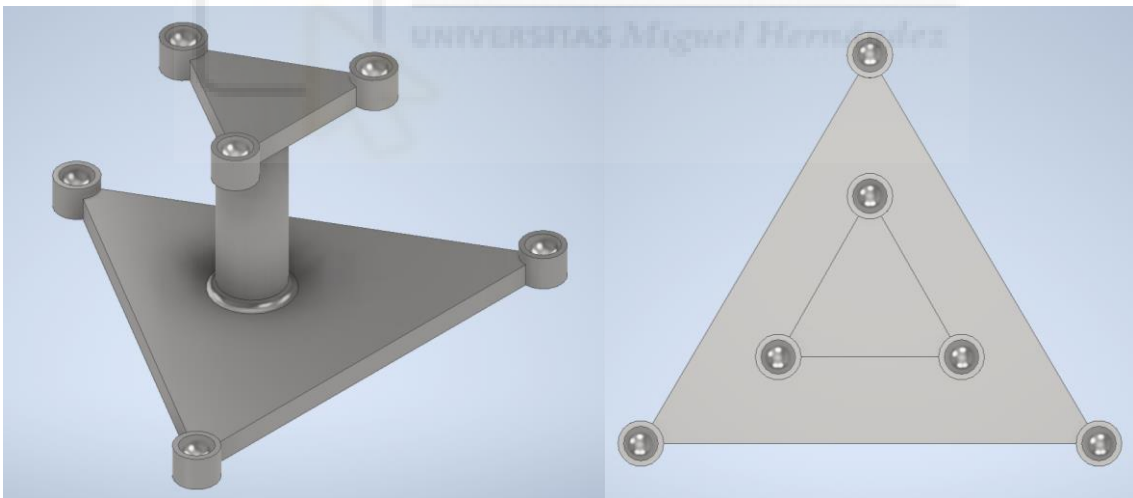
- **Plataforma**, a la que se conectarán el otro extremo de las barras metálicas y sobre la que se aplicarán las fuerzas y momentos al sensor. Al igual que la base del sensor dispone de 6 rótulas esféricas de 1'5cm de diámetro.



**Figura 13.** Plataforma configuración "Comercial". Fuente: Elaboración propia

Para la plataforma de la configuración "Tetraédrica" se ha decidido simplificar la arquitectura con el objetivo de facilitar la fabricación de la pieza y evitar de forma prematura interferencias con las barras o células de carga instaladas en estas.

Las dos bases que componen la plataforma se unen mediante un cilindro de 3cm de diámetro y 8cm de altura soldado por ambos extremos.

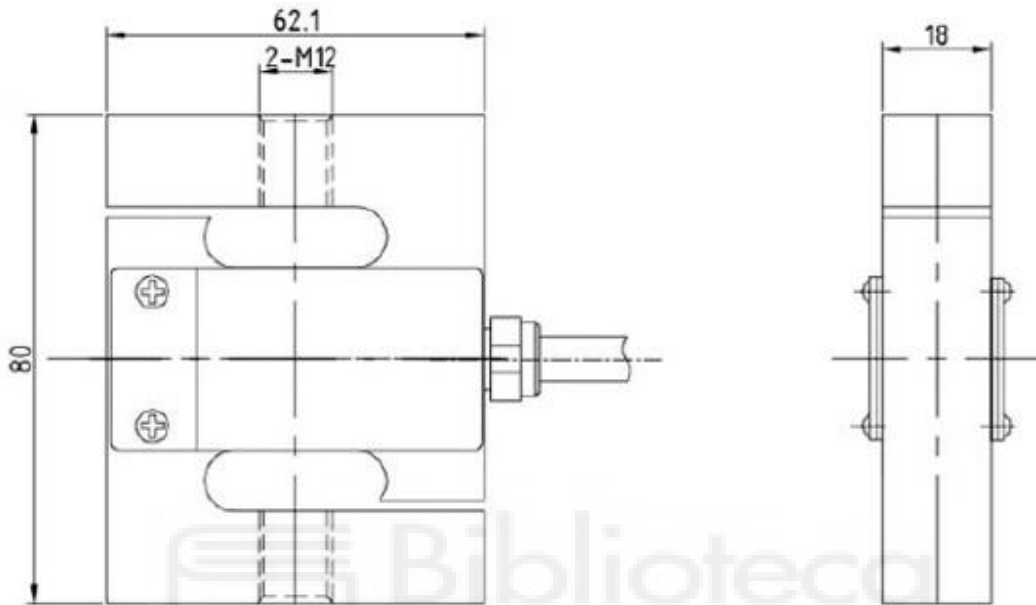


**Figura 14.** Plataforma configuración "Tetraédrica". Fuente: Elaboración propia



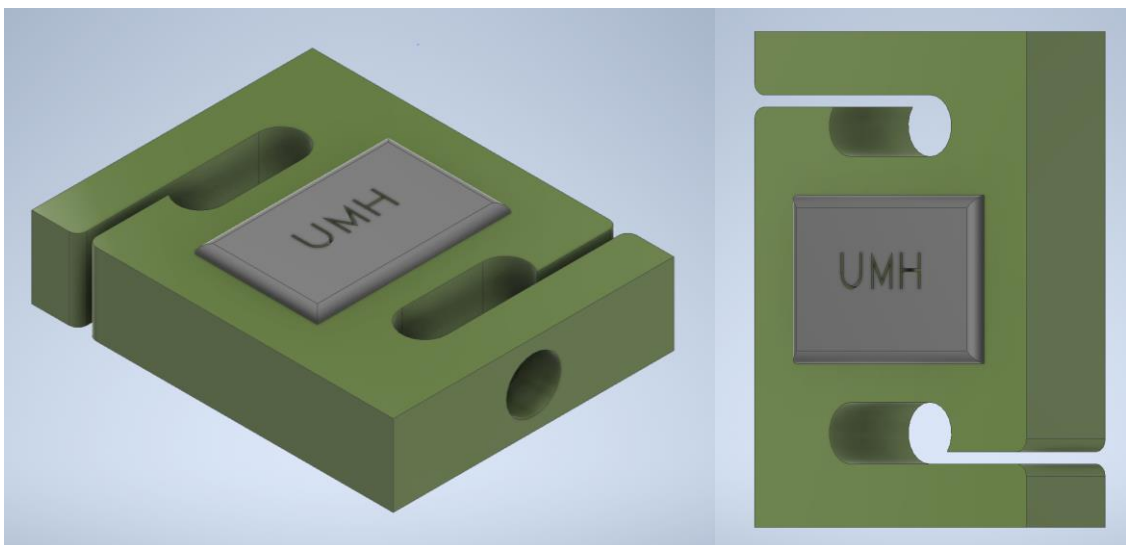
### 3.2. CÉLULAS DE CARGA

En el sensor se instalarán las células de carga “RS PRO Alloy Steel S Beam Tension Load Cell”, las cuales tienen unas dimensiones de 8x6’21cm, una profundidad de 1’8cm y dos orificios roscados M12 en los que se instalarán las barras de acero inoxidable. Su hoja técnica completa se adjuntará en un anexo al final de la memoria.



**Figura 11.** Plano célula de carga. Fuente: Ficha técnica RS PRO Alloy Steel S Beam Tension Load Cell

Para realizar el modelo 3D del sensor se han modelado de forma aproximada las células de carga atendiendo a su plano y atribuyéndoles el material de acero inoxidable.



**Figura 12.** Modelo 3D célula de carga. Fuente: Elaboración propia

### 3.3. DISEÑO y ENSAMBLAJE

El ensamblaje de los diferentes componentes se realizó mediante la restricción de coincidencia, haciendo coincidir ejes y superficies. La barra 1 ha sido coloreada de amarillo para su fácil identificación y facilitar el montaje del sensor.

#### 3.3.1. Configuración “Comercial”

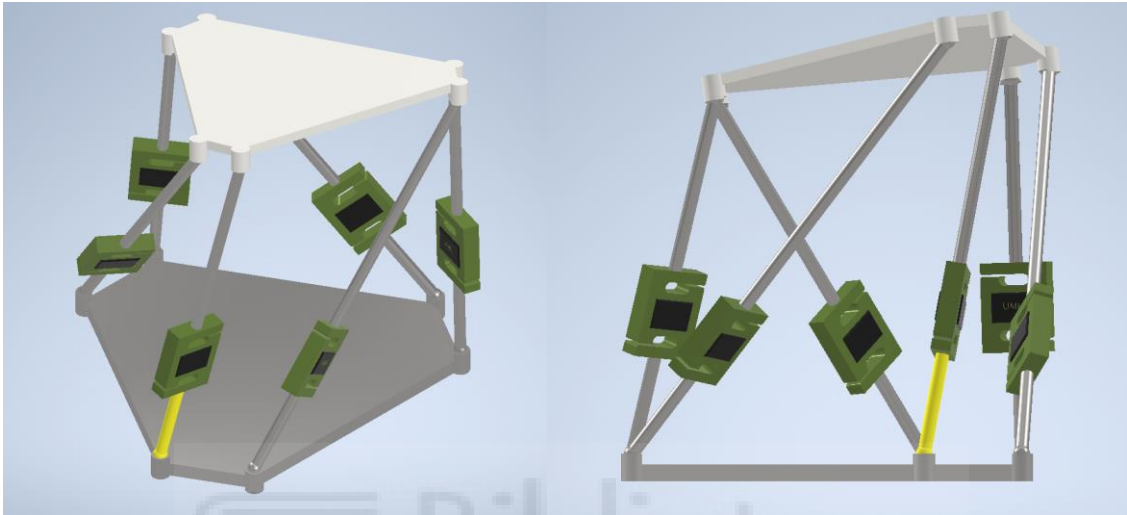


Figura 13. Modelo 3D configuración “comercial”. Fuente: Elaboración propia

#### 3.3.2. Configuración “Tetraédrica”

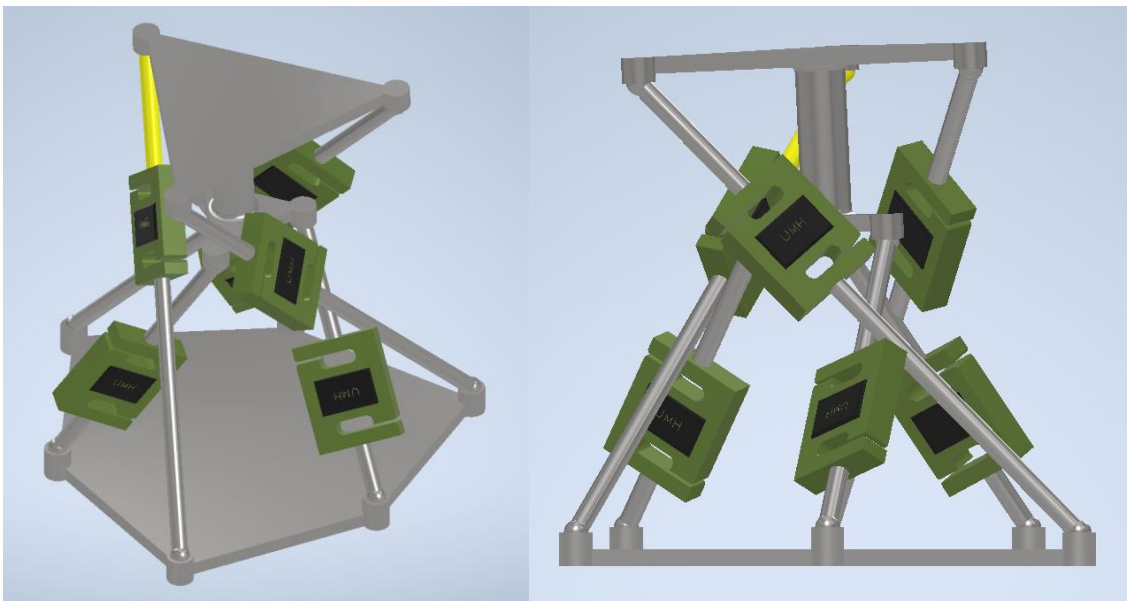
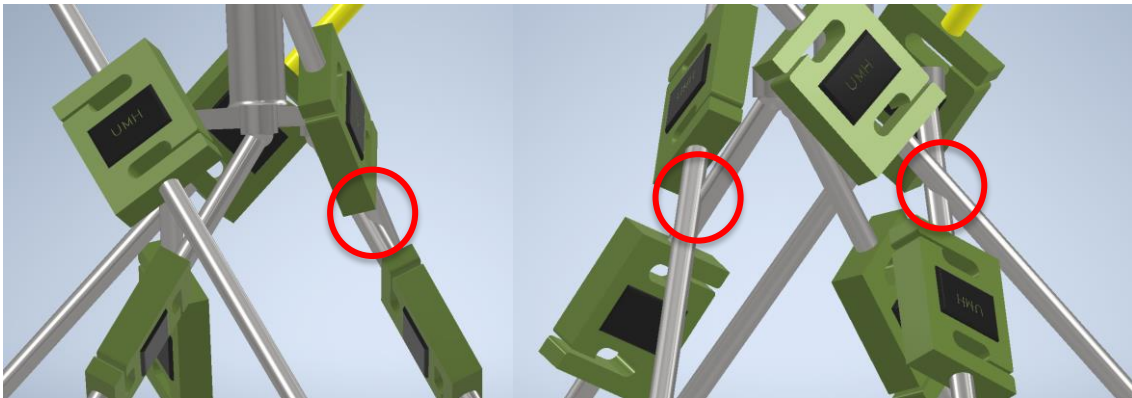


Figura 14. Modelo 3D configuración “tetraédrica”. Fuente: Elaboración propia

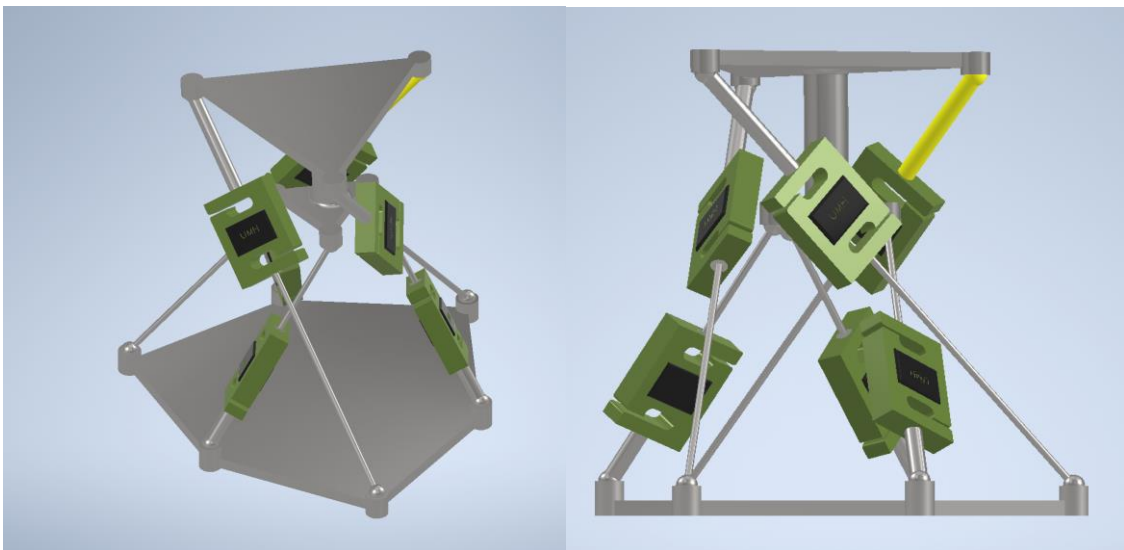
Al realizar el ensamblado de todos los componentes que conforman el sensor en su configuración “tetraédrica” se produjo una interferencia entre las barras exteriores e interiores.

El modelo matemático generado en Matlab proporcionaba barras unidimensionales en las cuales se ignoraba su grosor y solo se proporcionaba la longitud de estas. El grosor de las barras puede afectar al espacio disponible y si son demasiado gruesas, pueden ocupar más espacio del disponible.



**Figura 15.** Interferencias entre barras del modelo 3D configuración “tetraédrica”. Fuente: Elaboración propia

Para solucionar este problema se optó por reducir el grosor de las barras inferiores de 1'2cm de diámetro a 0'5cm. Esta reducción en las secciones de las barras inferiores causó preocupación sobre la resistencia de estas a las fuerzas a las que sería sometido el sensor. Para combatir esta reducción de la sección se barajó cambiar el material de las barras inferiores si estas no soportaran las fuerzas. El material cambiaría de acero inoxidable a titanio, un material mucho más caro, pero a su vez más resistente.



**Figura 16.** Modelo 3D configuración final “tetraédrica”. Fuente: Elaboración propia

### 3.4. SIMULACIONES

Las simulaciones se han realizado aplicando el efecto de la gravedad sobre todo el ensamblaje y una carga distribuida de compresión de  $1'78\text{Kg/cm}^2$  para la configuración "Comercial" y  $4'140\text{Kg/cm}^2$  para la configuración "Tetraédrica" que aplicada sobre la superficie total de la plataforma equivaldría a 700Kg.

En las simulaciones estudiaremos la tensión de Von Mises, el coeficiente de seguridad y finalmente los desplazamientos que sufren los componentes debido a las deformaciones.

La tensión de Von Mises es un concepto fundamental en la mecánica de materiales y la ingeniería estructural, ya que se utiliza para predecir el comportamiento de los materiales bajo cargas complejas. La tensión de Von Mises se utiliza como un criterio para determinar cuándo un material comenzará a fluir o deformarse plásticamente. Esto es crucial en el diseño de estructuras y componentes, ya que permite asegurarse de que las piezas no fallarán bajo condiciones de carga específicas. Además, mediante el análisis por elementos finitos (FEA), la tensión de Von Mises es ampliamente utilizada para evaluar el comportamiento estructural bajo diversas condiciones, lo que ayuda a optimizar diseños antes de la fabricación.

En resumen, la tensión de Von Mises es esencial para garantizar que los materiales y estructuras sean diseñados adecuadamente para soportar las cargas esperadas sin sufrir fallos prematuros.

Por otro lado, el coeficiente de seguridad (también conocido como factor de seguridad) es un concepto fundamental en ingeniería y diseño estructural que se utiliza para garantizar que una estructura o componente pueda soportar las cargas y condiciones a las que estará sometido durante su vida útil. Se define como la relación entre la resistencia máxima de un material o estructura y la carga real o máxima esperada que se aplicará sobre él.

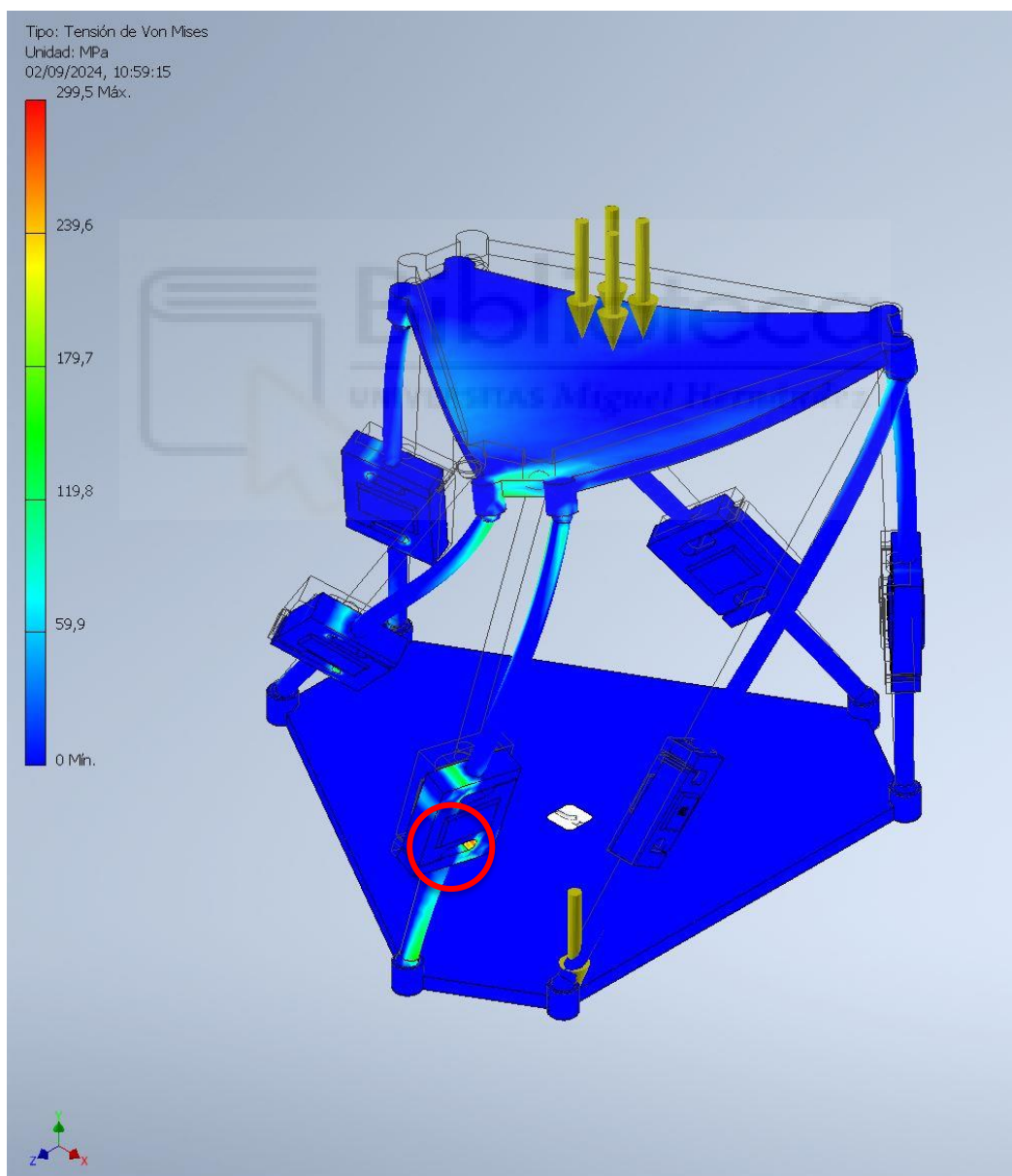
Finalmente, los desplazamientos permiten entender cómo se deformará una pieza bajo carga, lo que es crucial para garantizar que la estructura cumpla con los requisitos funcionales y estéticos.

### 3.4.1. Configuración “Comercial”

Tras someter el modelo a la gravedad y a la carga distribuida de  $1'78\text{Kg/cm}^2$  definidas anteriormente obtenemos los siguientes resultados:

**Tensión de Von Mises**, cuyo valor máximo es de  $299'5\text{Mpa}$  y se obtiene en una célula de carga como se muestra en la figura 17. Tres de las barras están tensionadas ligeramente con un valor de  $59'9\text{MPa}$  mientras que el resto del sensor no presenta tensión alguna.

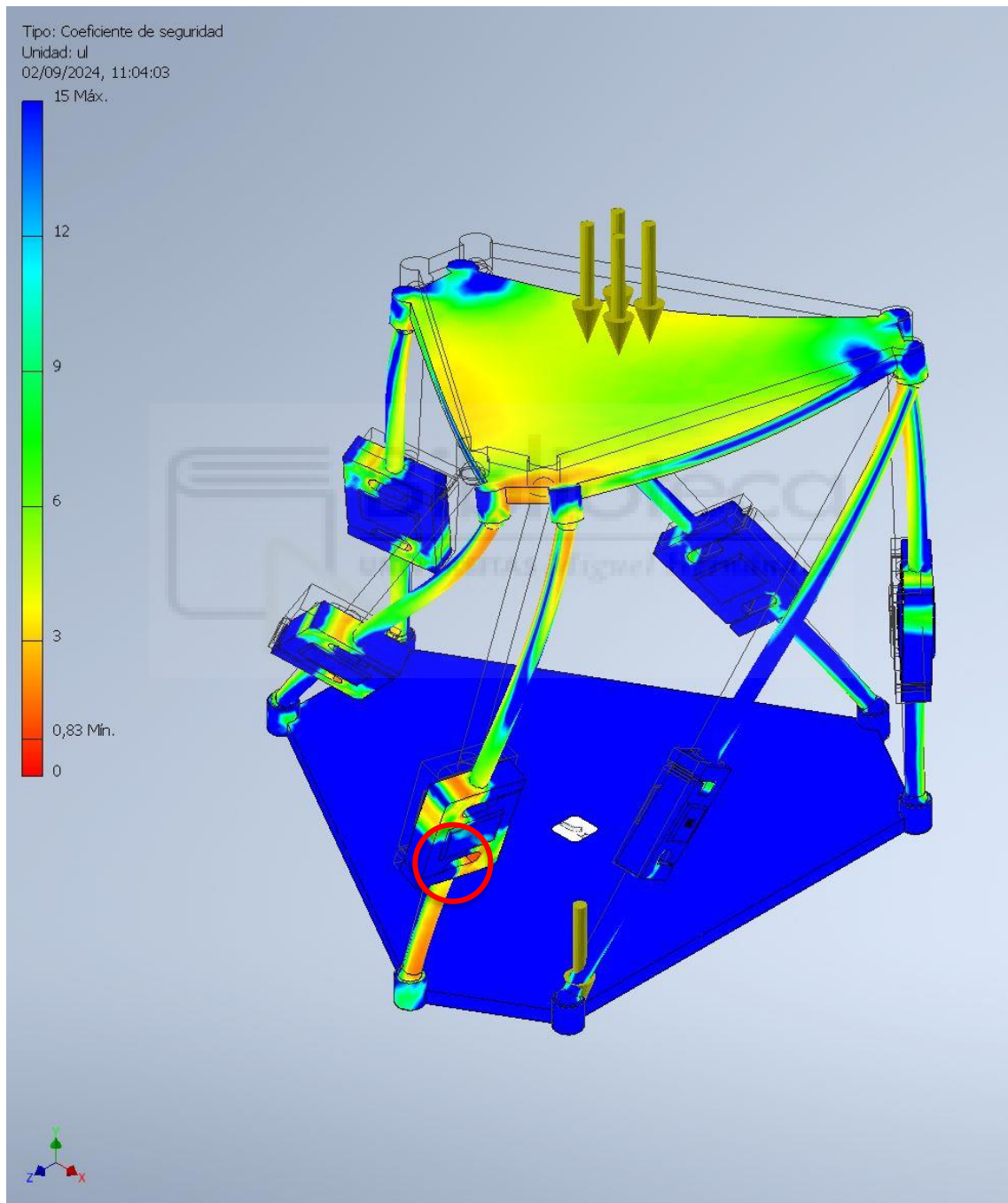
Es importante tomar con cautela esta tensión máxima, ya que las células de carga han sido modeladas de forma aproximada a partir de las dimensiones proporcionadas en el datasheet y posiblemente tengan un comportamiento diferente en el modelo real.



**Figura 17.** Resultados de la simulación. Tensión de Von Mises. Fuente: Elaboración propia

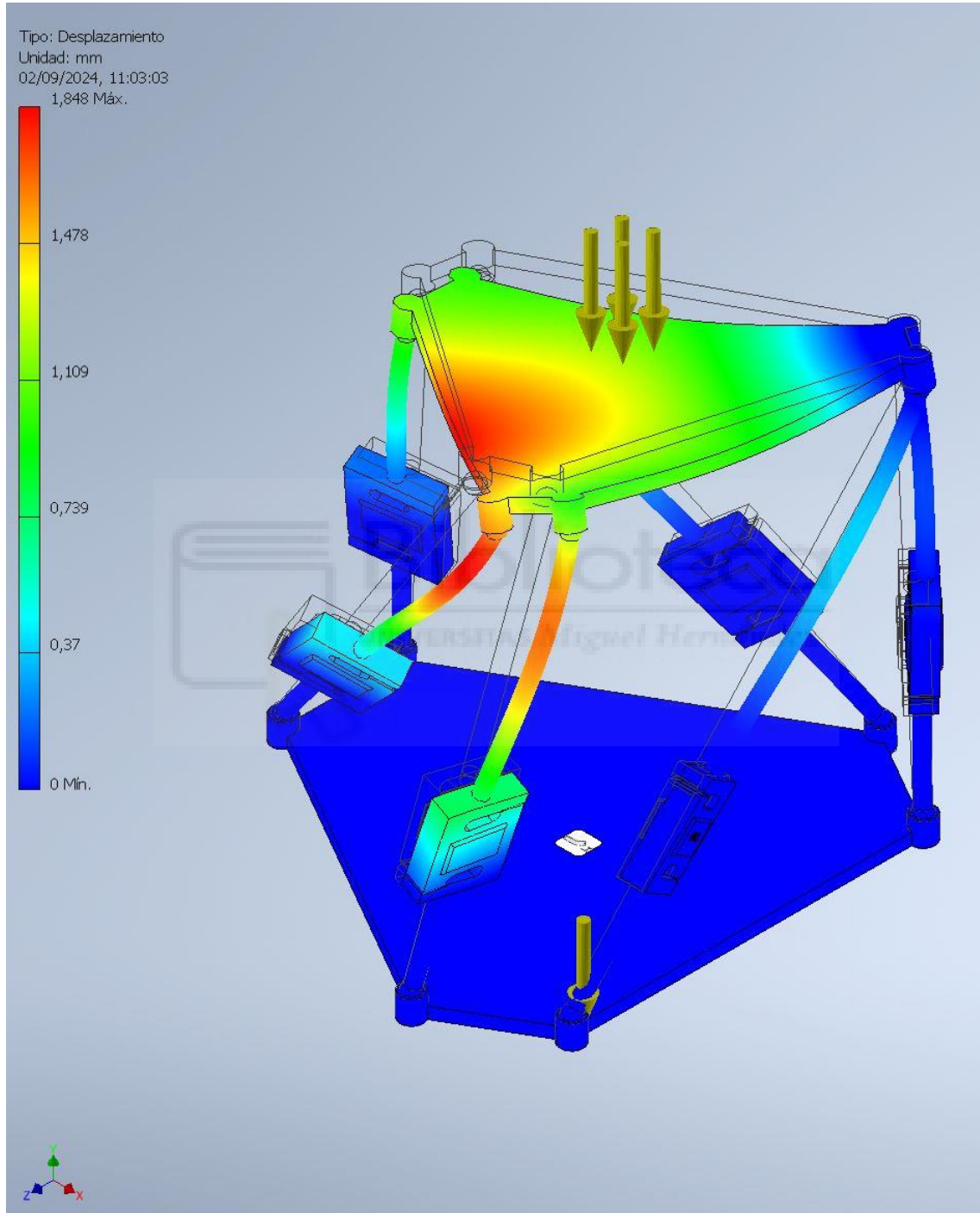
El **coeficiente de seguridad** mínimo se encuentra en 0'83 en la célula de carga sometida a la mayor tensión de Von Mises. Un valor por debajo de 1 indica que el modelo podría fallar y sucumbir ante la fuerza aplicada. Además, una gran porción de las barras en sus puntos de anclaje ve su factor de seguridad reducido a valores comprendidos entre 3 y 1'5.

Una posible solución para corregir este fenómeno sería aumentar el diámetro de las barras de 1'2cm a 1'5cm o cambiar el material de estas a titanio.



**Figura 18.** Resultados de la simulación. Factor de seguridad. Fuente: Elaboración propia

Finalmente, los **desplazamientos** máximos se producen en la parte superior de dos de las barras y en un costado de la plataforma con un valor máximo de 1'845mm. El resto de la plataforma se desplaza entre 1'1mm a 0'73mm y de igual manera la porción inferior de las barras que presentaban el mayor desplazamiento.

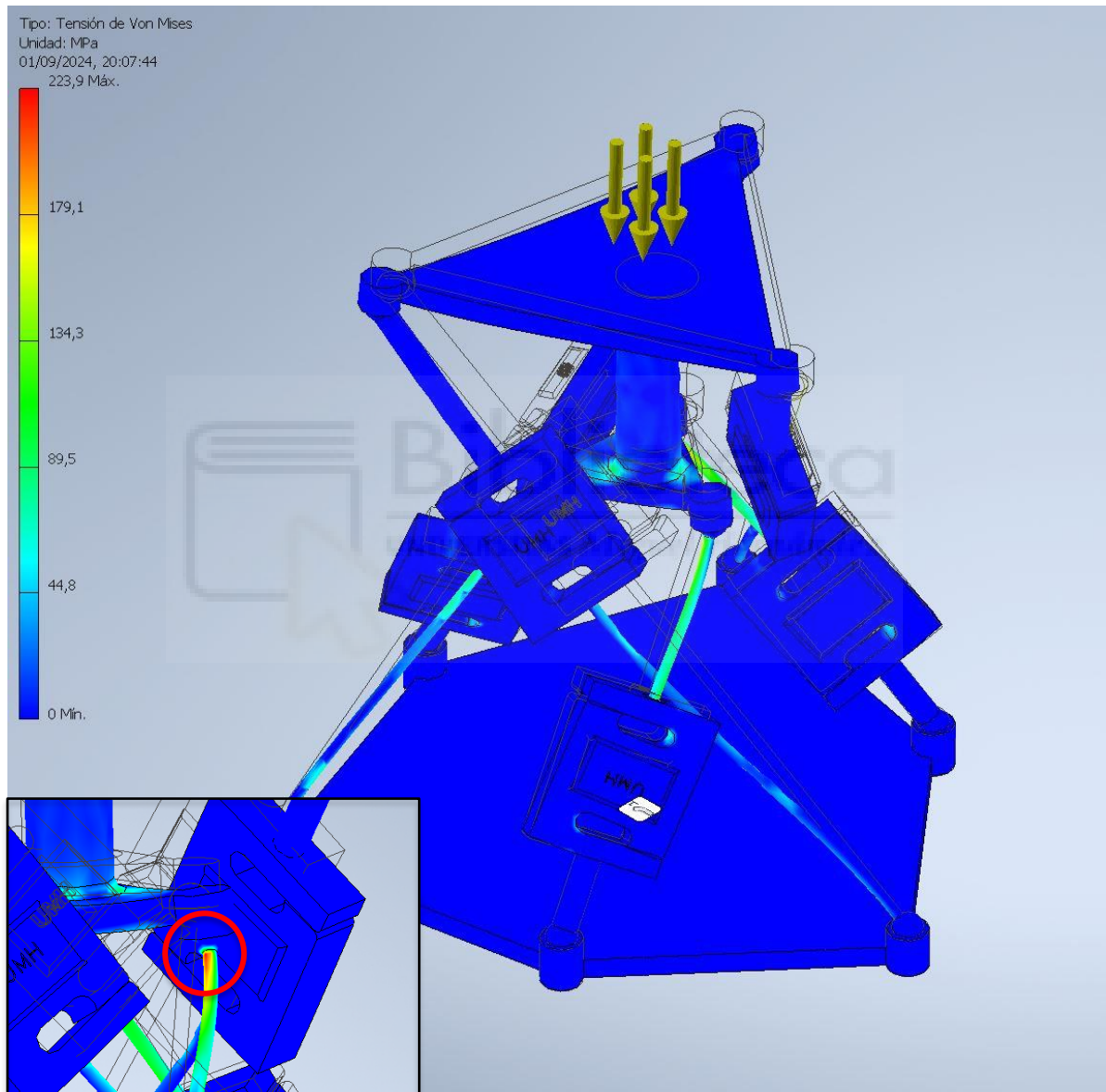


**Figura 19.** Resultados de la simulación. Factor de seguridad. Fuente: Elaboración propia

### 3.4.2. Configuración “Tetraédrica”

Tras someter el modelo a la gravedad y a la carga distribuida de  $4'140\text{Kg/cm}^2$  definidas anteriormente obtenemos los siguientes resultados:

**Tensión de Von Mises**, cuyo valor máximo es de  $223'9\text{Mpa}$  y se obtiene en una de las barras superiores interiores como se muestra en la figura 20. El resto de las barras interiores oscila entre una tensión de  $150\text{MPa}$  a  $40\text{MPa}$ .

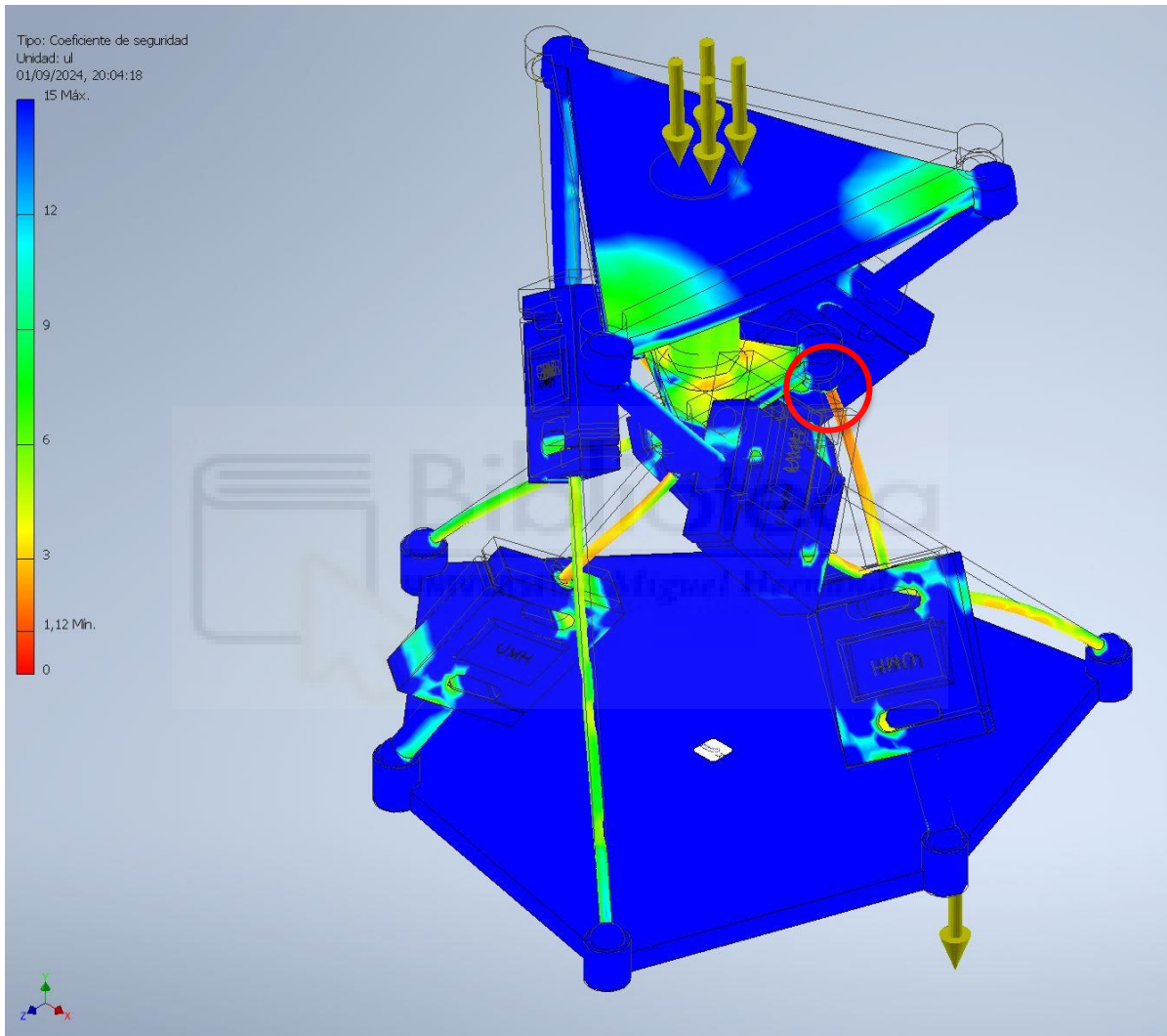


*Figura 20. Resultados de la simulación. Tensión de Von Mises. Fuente: Elaboración propia*



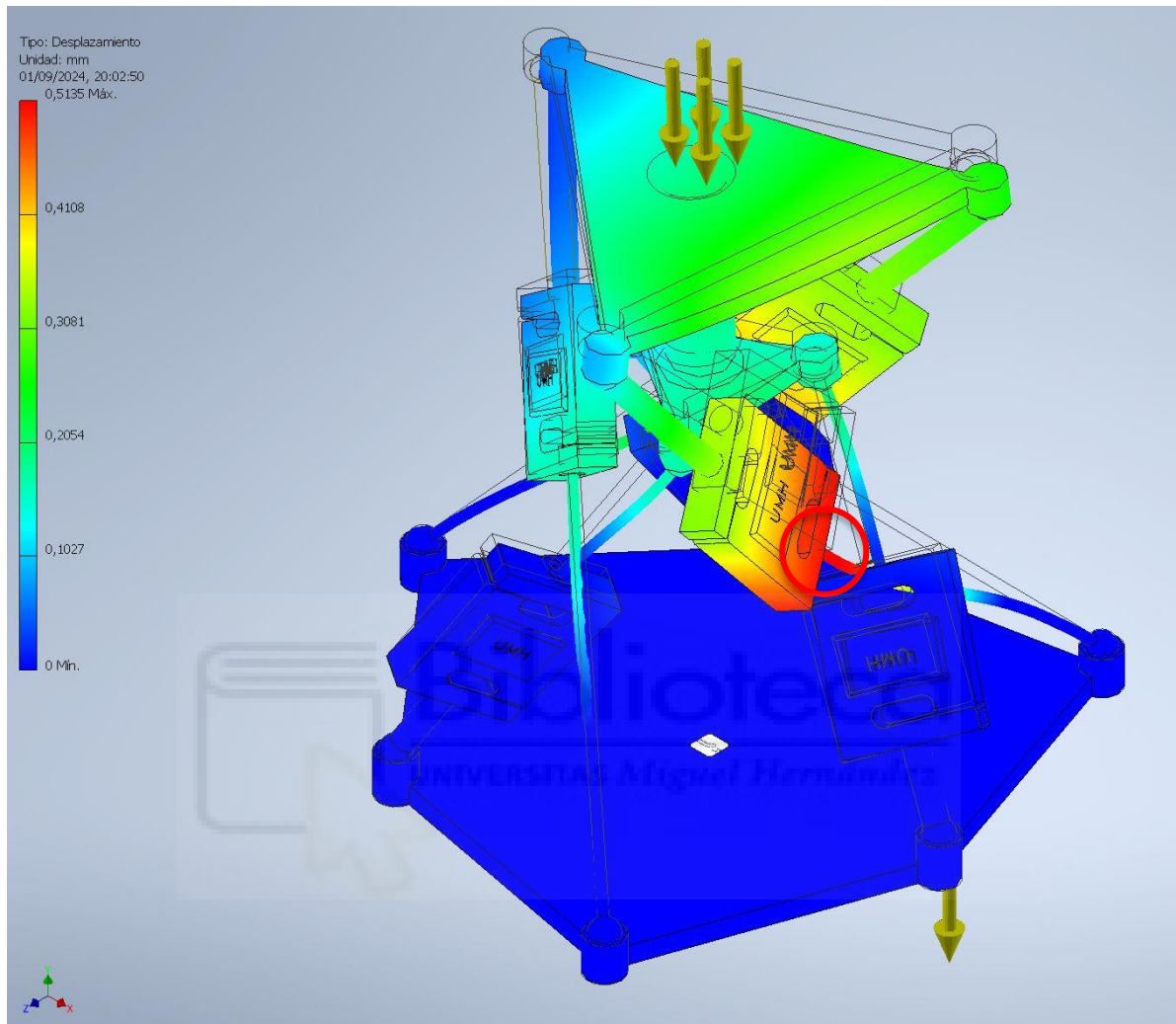
El **coeficiente de seguridad** mínimo se encuentra en 1'12, tanto en las barras interiores como exteriores, lo que en un principio garantiza que el sensor aguantaría a pesar de verse reducidas las secciones de las barras inferiores. Por tanto, se descartó cambiar el material a titanio.

La plataforma superior, en su cordón de soldadura con el cilindro que une ambas bases se presenta un factor de seguridad de 3 en algunos puntos. Cuando se fabrique esta pieza será necesario asegurarse que se realiza una buena soldadura para evitar que bajo la carga máxima pueda partir la unión.



**Figura 21.** Resultados de la simulacion. Factor de seguridad. Fuente: Elaboracion propia

Finalmente, los **desplazamientos** máximos se producen en la parte inferior de una célula de carga con un valor máximo de 0'5135mm. La plataforma se desplaza entre 0'1mm y 0'2mm al igual que las barras en sus puntos de conexión a las células de carga.



**Figura 22.** Resultados de la simulación. Desplazamientos. Fuente: Elaboración propia

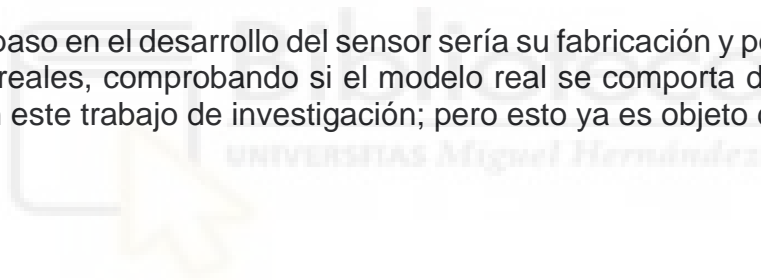
## 4. CONCLUSIONES

Tras analizar detalladamente las simulaciones y diseños de ambas configuraciones se ha llegado a las siguientes conclusiones.

La configuración “tetraédrica” se presenta como la más favorable ya que todos sus componentes presentan un factor de seguridad mayor de 1, además, el número de condición asociado era menor que el de la configuración comercial, haciendo esta configuración más precisa. Sin embargo, la configuración “comercial” también tiene sus ventajas, es una configuración más simple y por tanto más sencilla, económica de fabricar y cuya mayor debilidad es fácilmente subsanable ya sea incrementando el grosor de las barras o cambiando el acero inoxidable por titanio.

En resumen, ambas configuraciones son perfectamente factibles y aunque la configuración “tetraédrica” se postule como la preferida sería recomendable realizar pruebas de ambas con modelos reales. Al realizar los modelos en 3D y sus respectivas simulaciones no se tuvo en cuenta el factor humano involucrado en el montaje de los sensores. La mayor complejidad y número de piezas de la configuración “tetraédrica” puede ser más propensa a contener fallos, ya sean de tolerancias o de soldadura que acaben afectando al rendimiento del sensor.

El siguiente paso en el desarrollo del sensor sería su fabricación y posterior prueba bajo cargas reales, comprobando si el modelo real se comporta de acuerdo a lo estudiado en este trabajo de investigación; pero esto ya es objeto de otro trabajo.



## 5. ANEXOS

### 5.1. CÓDIGO DE LAS FUNCIONES EN MATLAB

*plotear\_robot:*

```
function handle = plotear_robot(a,b,px,py,pz,alpha,beta,gamma)
```

```
handle = [];
```

```
%Matriz de transformación
```

```
p=[px;py;pz];
```

```
R=[cos(beta)*cos(gamma) -cos(beta)*sin(gamma)
```

```
sin(beta);cos(alpha)*sin(gamma)+sin(alpha)*sin(beta)*cos(gamma)
```

```
cos(alpha)*cos(gamma)-sin(alpha)*sin(beta)*sin(gamma) -
```

```
sin(alpha)*cos(beta);sin(alpha)*sin(gamma)-cos(alpha)*sin(beta)*cos(gamma)
```

```
cos(alpha)*sin(beta)*sin(gamma)+sin(alpha)*cos(gamma) cos(alpha)*cos(beta)];
```

```
T=[R,p;[0 0 0 1]];
```

```
%Cambio de sistema de coordenadas
```

```
Aux=T*[b;[1 1 1 1 1 1]];
```

```
B=Aux(1:3,:);
```

```
%Cálculo de la longitud de las barras
```

```
L=vecnorm(a-B)';
```

```
hold on
```

```
grid on
```

```
handle = [handle,fill3(a(1,:),a(2,:),a(3:),'r')];
```

```
handle = [handle,fill3(B(1,:),B(2,:),B(3:),'b')];
```

```
for i=1:6
```

```
    handle = [handle,plot3([a(1,i) B(1,i)],[a(2,i) B(2,i)],[a(3,i)
```

```
    B(3,i)],'k','LineWidth',2)];
```

```
end
```

```
zlim([-0.2,0.6])
```

```
set(gca,'DataAspectRatio',[1 1 1])
```

```
xlabel('X')
```

```
ylabel('Y')
```

```
zlabel('Z')
```

```
end
```

### ***cinematicainversa:***

```
function L = cinematicainversa(a,b,px,py,pz,alpha,beta,gamma)
```

```
%Matriz de transformación
```

```
p=[px;py;pz];
```

```
R=[cos(beta)*cos(gamma) -cos(beta)*sin(gamma)
```

```
sin(beta);cos(alpha)*sin(gamma)+sin(alpha)*sin(beta)*cos(gamma)
```

```
cos(alpha)*cos(gamma)-sin(alpha)*sin(beta)*sin(gamma) -
```

```
sin(alpha)*cos(beta);sin(alpha)*sin(gamma)-cos(alpha)*sin(beta)*cos(gamma)
```

```
cos(alpha)*sin(beta)*sin(gamma)+sin(alpha)*cos(gamma) cos(alpha)*cos(beta)];
```

```
T=[R,p;[0 0 0 1]];
```

```
%Cambio de sistema de coordenadas
```

```
Aux=T*[b;[1 1 1 1 1 1]];
```

```
B=Aux(1:3,:);
```

```
%Cálculo de la longitud de las barras
```

```
L=vecnorm(a-B)';
```

```
end
```

### ***RobotsV3 (configuración "comercial"):***

```
%Variables a definir
```

```
%Coordenadas en la placa inferior: x1->x6, y1->y6, z1->z6
```

```
%Coordenadas en la placa superior: X1->X6, Y1->Y6, Z1->Z6
```

```
%Ángulos de acople: Alpha, Betta, Gamma
```

```
% Se debe elegir entre las matrices del modelo teórico o real
```

```
%Matrices en coordenadas locales (Modelo comercial teórico)
```

```
a1=[0.05;0.0866;0];
```

```
a2=[0.1;0;0];
```

```
a3=[0.4;0;0];
```

```
a4=[0.45;0.0866;0];
```

```
a5=[0.30;0.3464;0];
```

```
a6=[0.20;0.3464;0];
```

```
b1=[0.10;0.0567;0];
```

```
b2=[0.4;0.0567;0];
```

```
b3=[0.4;0.0567;0];
```

```
b4=[0.25;0.3175;0];
```

```
b5=[0.25;0.3175;0];
```

```
b6=[0.10;0.0567;0];
```

**%Matrices en coordenadas locales (Modelo comercial)**

```
a1=[0.05;0.0866;0];  
a2=[0.1;0;0];  
a3=[0.4;0;0];  
a4=[0.45;0.0866;0];  
a5=[0.30;0.3464;0];  
a6=[0.20;0.3464;0];
```

```
b1=[0.115;0.0367;0];  
b2=[0.385;0.0367;0];  
b3=[0.415;0.0767;0];  
b4=[0.2725;0.3175;0];  
b5=[0.2275;0.3175;0];  
b6=[0.085;0.0767;0];
```

```
a=[a1 a2 a3 a4 a5 a6];  
b=[b1 b2 b3 b4 b5 b6];
```

**%Jacobiana**

```
d=0.0001;
```

```
N = 10000;  
datos = cell(N,4);  
for i=1:N
```

**%Cálculos de arriba, generando valores aleatorios para px, py, pz, alpha, beta y gamma.**

```
px_min = -0.01;  
py_min = -0.01;  
pz_min = 0.4-0.01;  
px_max = 0.01;  
py_max = 0.01;  
pz_max = 0.4+0.01;
```

```
alpha_min = -0.15;  
beta_min = -0.15;  
gamma_min = -0.15;  
alpha_max = 0.15;  
beta_max = 0.15;  
gamma_max = 0.15;
```

```
px = px_min + (px_max - px_min)*rand;  
py = py_min + (py_max - py_min)*rand;  
pz = pz_min + (pz_max - pz_min)*rand;  
alpha = alpha_min + (alpha_max - alpha_min)*rand;  
beta = beta_min + (beta_max - beta_min)*rand;  
gamma = gamma_min + (gamma_max - gamma_min)*rand;
```

```
longitudes_barras = cinematicainversa(a,b,px,py,pz,alpha,beta,gamma);
```

```
J1=(cinematicainversa(a,b,px+d,py,pz,alpha,beta,gamma)-  
cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;
```

```
J2=(cinematicainversa(a,b,px,py+d,pz,alpha,beta,gamma)-  
cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;
```

```
J3=(cinematicainversa(a,b,px,py,pz+d,alpha,beta,gamma)-  
cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;
```

```
J4=(cinematicainversa(a,b,px,py,pz,alpha+d,beta,gamma)-  
cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;
```

```
J5=(cinematicainversa(a,b,px,py,pz,alpha,beta+d,gamma)-  
cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;
```

```
J6=(cinematicainversa(a,b,px,py,pz,alpha,beta,gamma+d)-  
cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;
```

```
J = [ J1 J2 J3 J4 J5 J6 ];
```

```
datos{i,1} = [px, py, pz, alpha, beta, gamma];
```

```
datos{i,2} = a;
```

```
datos{i,3} = b;
```

```
datos{i,4} = cond(J);
```

```
if cond(J) < 2
```

```
break
```

```
end
```

```
end
```

```
close all
```

```
save(strrep(datestr(datetime("now")),':','-'))
```

```
figure
```

```
plot([datos{:},4'],'ob')
```

```
%%
```

```
figure
```

```
por_debajo_de = [];
```

```
umbral = 25;
```

```

for i=1:N
    if datos{i,4} < umbral
        por_debajo_de = [por_debajo_de;datos{i,1}];

        px = datos{i,1}(1);
        py = datos{i,1}(2);
        pz = datos{i,1}(3);
        alpha = datos{i,1}(4);
        beta = datos{i,1}(5);
        gamma = datos{i,1}(6);

        handle = plotear_robot(a,b,px,py,pz,alpha,beta,gamma);
        pause(2)
        delete(handle)

        view(20,40)
    end
end

por_debajo_de

%%
% PRUEBA es la prueba a representar obtenida de los cálculos previos. Se obtiene
del eje X
px = datos{5126,1}(1);
py = datos{5126,1}(2);
pz = datos{5126,1}(3);
alpha = datos{5126,1}(4);
beta = datos{5126,1}(5);
gamma = datos{5126,1}(6);
figure
plotear_robot(a,b,px,py,pz,alpha,beta,gamma)

```



### **RobotsV3 (configuración “tetraédrica”):**

%Variables a definir

%Coordenadas en la placa inferior: x1->x6, y1->y6, z1->z6

%Coordenadas en la placa superior: X1->X6, Y1->Y6, Z1->Z6

%Ángulos de acople: Alpha, Betta, Gamma

%Matrices en coordenadas locales (Modelo tetraedrico)

a1=[-0.15;-0.0866;0];

a2=[0;-0.1386;0.03];

a3=[0.15;-0.0866;0];

a4=[0.12;0.0693;0.03];

a5=[0;0.1732;0];

a6=[-0.12;0.0693;0.03];

b1=[0;0.1115;0];

b2=[0.04;-0.02;-0.1];

b3=[-0.10;-0.0577;0];

b4=[0;0.05;-0.1];

b5=[0.1;-0.0577;0];

b6=[-0.04;-0.02;-0.1];

a=[a1 a2 a3 a4 a5 a6];

b=[b1 b2 b3 b4 b5 b6];

%Jacobiana

d=0.0001;

N = 10000;

datos = cell(N,4);

for i=1:N

%Cálculos de arriba, generando valores aleatorios para px, py, pz, alpha, beta y gamma.

px\_min = -0.01;

py\_min = -0.01;

px\_max = 0.01;

py\_max = 0.01;

pz\_min = 0.3-0.01;

pz\_max = 0.3+0.01;

alpha\_min = -0.15;

beta\_min = -0.15;

gamma\_min = -0.15;

alpha\_max = 0.15;

beta\_max = 0.15;

```

gamma_max = 0.15;

px = px_min + (px_max - px_min)*rand;
py = py_min + (py_max - py_min)*rand;
pz = pz_min + (pz_max - pz_min)*rand;
alpha = alpha_min + (alpha_max - alpha_min)*rand;
beta = beta_min + (beta_max - beta_min)*rand;
gamma = gamma_min + (gamma_max - gamma_min)*rand;

longitudes_barras = cinematicainversa(a,b,px,py,pz,alpha,beta,gamma);

J1=(cinematicainversa(a,b,px+d,py,pz,alpha,beta,gamma)-
cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;
J2=(cinematicainversa(a,b,px,py+d,pz,alpha,beta,gamma)-
cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;
J3=(cinematicainversa(a,b,px,py,pz+d,alpha,beta,gamma)-
cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;
J4=(cinematicainversa(a,b,px,py,pz,alpha+d,beta,gamma)-
cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;
J5=(cinematicainversa(a,b,px,py,pz,alpha,beta+d,gamma)-
cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;
J6=(cinematicainversa(a,b,px,py,pz,alpha,beta,gamma+d)-
cinematicainversa(a,b,px,py,pz,alpha,beta,gamma))/d;

J = [ J1 J2 J3 J4 J5 J6 ];

datos{i,1} = [px, py, pz, alpha, beta, gamma];
datos{i,2} = a;
datos{i,3} = b;

datos{i,4} = cond(J);
if cond(J) < 2
    break
end
end

close all
save(strrep(datestr(datetime("now")),':','-'))
figure
plot([datos{:},4], 'ob')

%%

figure

por_debajo_de = [];
umbral = 24;

```

```

for i=1:N
    if datos{i,4} < umbral
        por_debajo_de = [por_debajo_de;datos{i,1}];

        px = datos{i,1}(1);
        py = datos{i,1}(2);
        pz = datos{i,1}(3);
        alpha = datos{i,1}(4);
        beta = datos{i,1}(5);
        gamma = datos{i,1}(6);

        handle = plotear_robot(a,b,px,py,pz,alpha,beta,gamma);
        pause(2)
        delete(handle)

        view(20,40)
    end
end

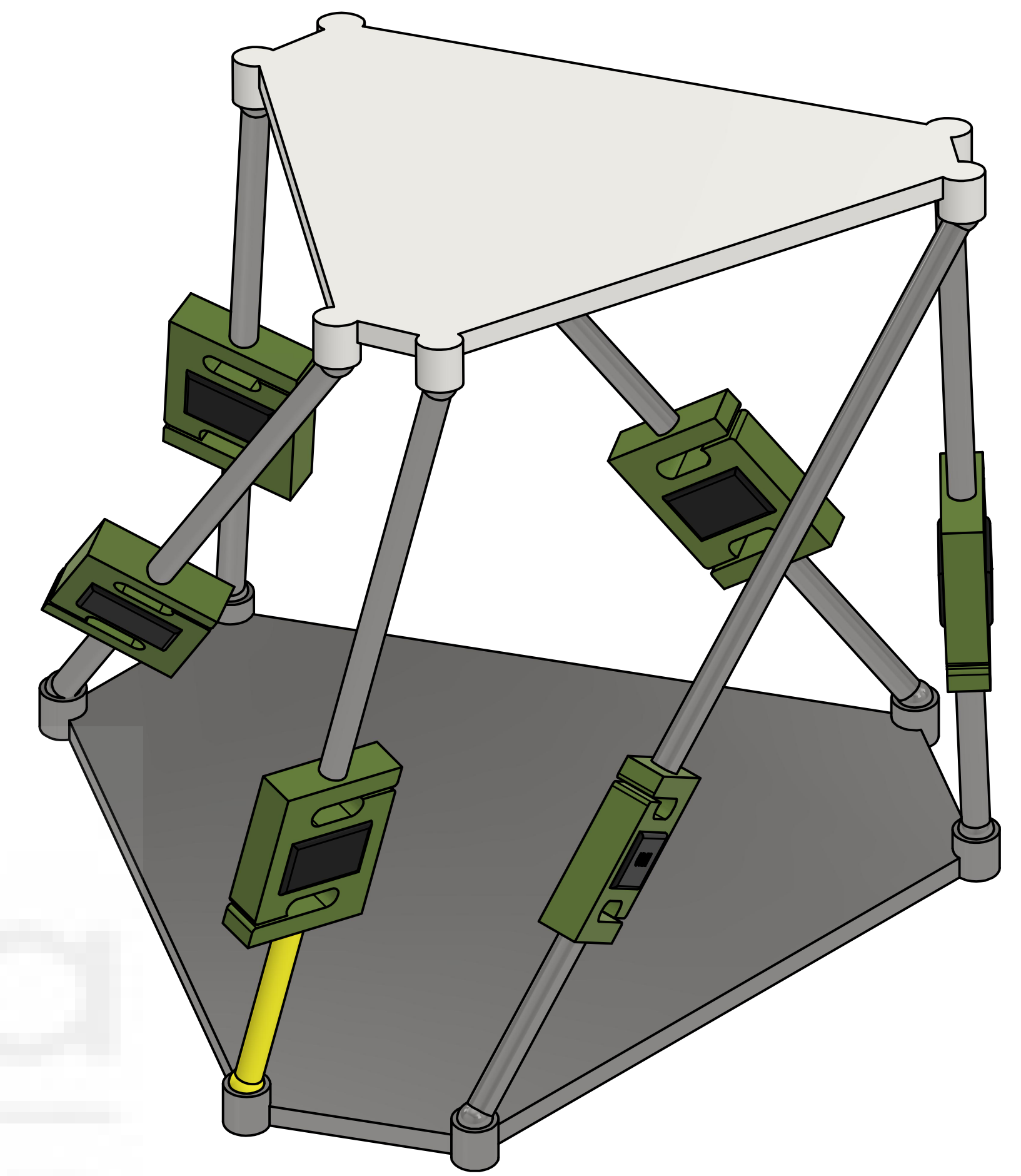
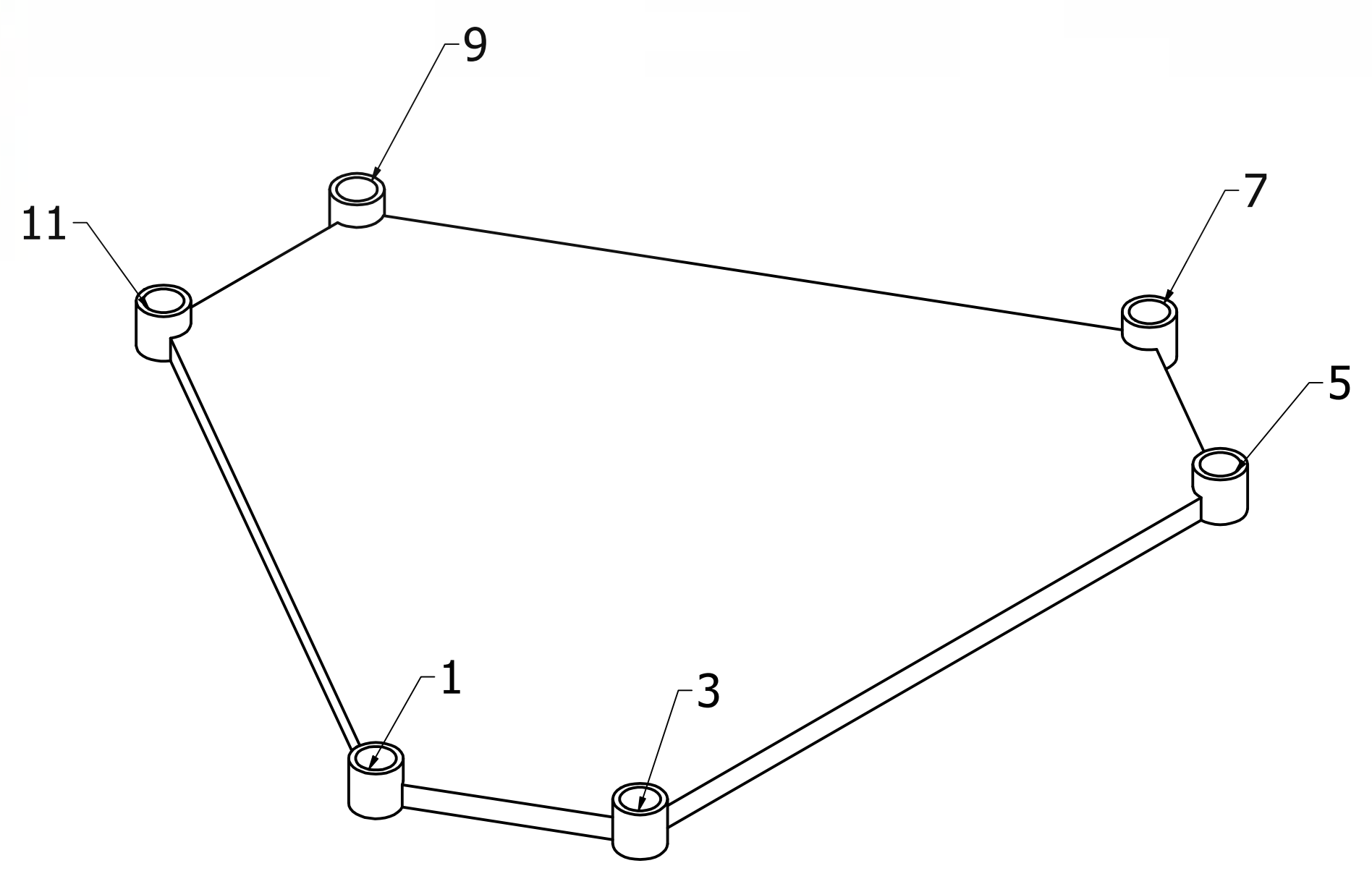
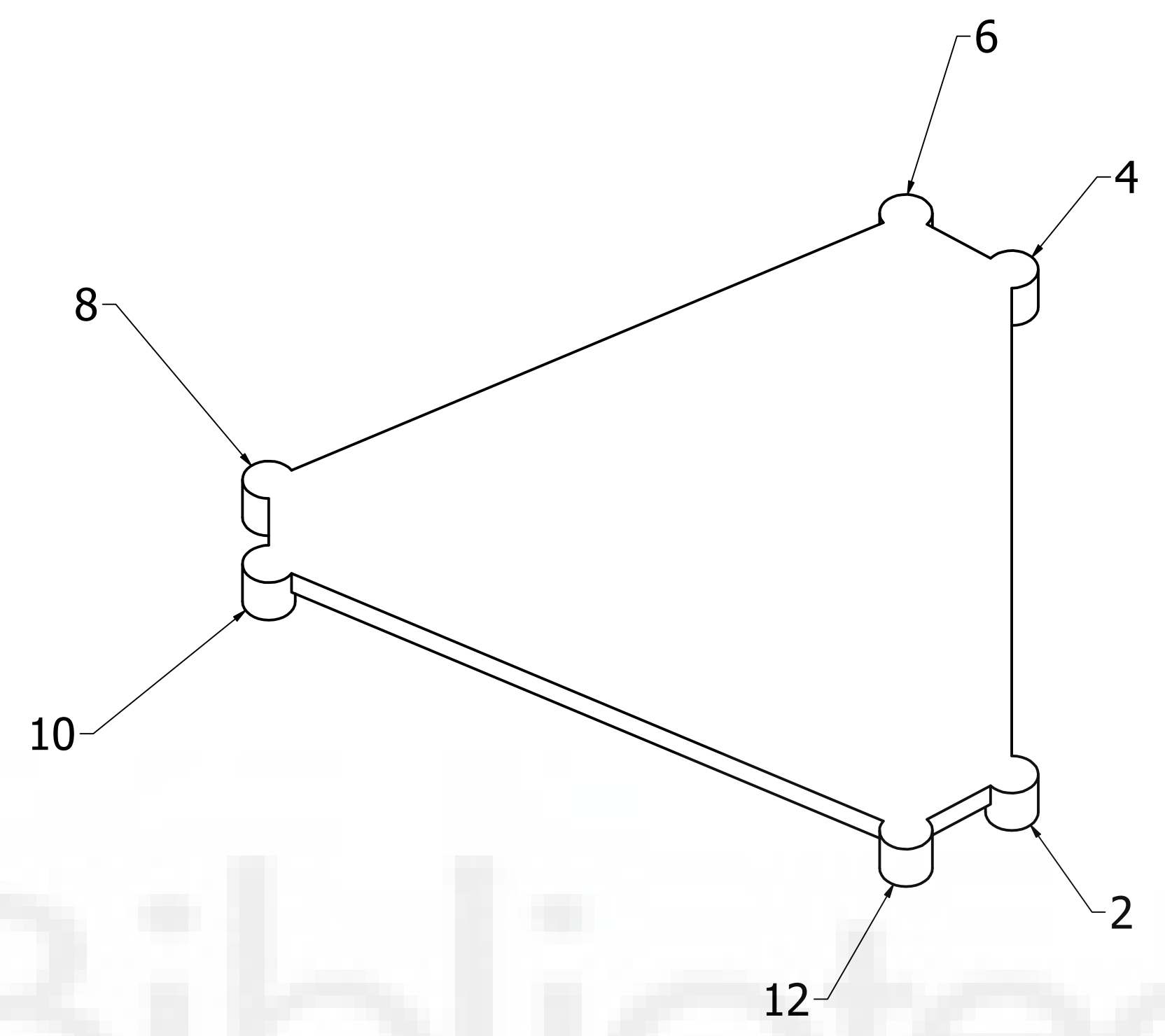
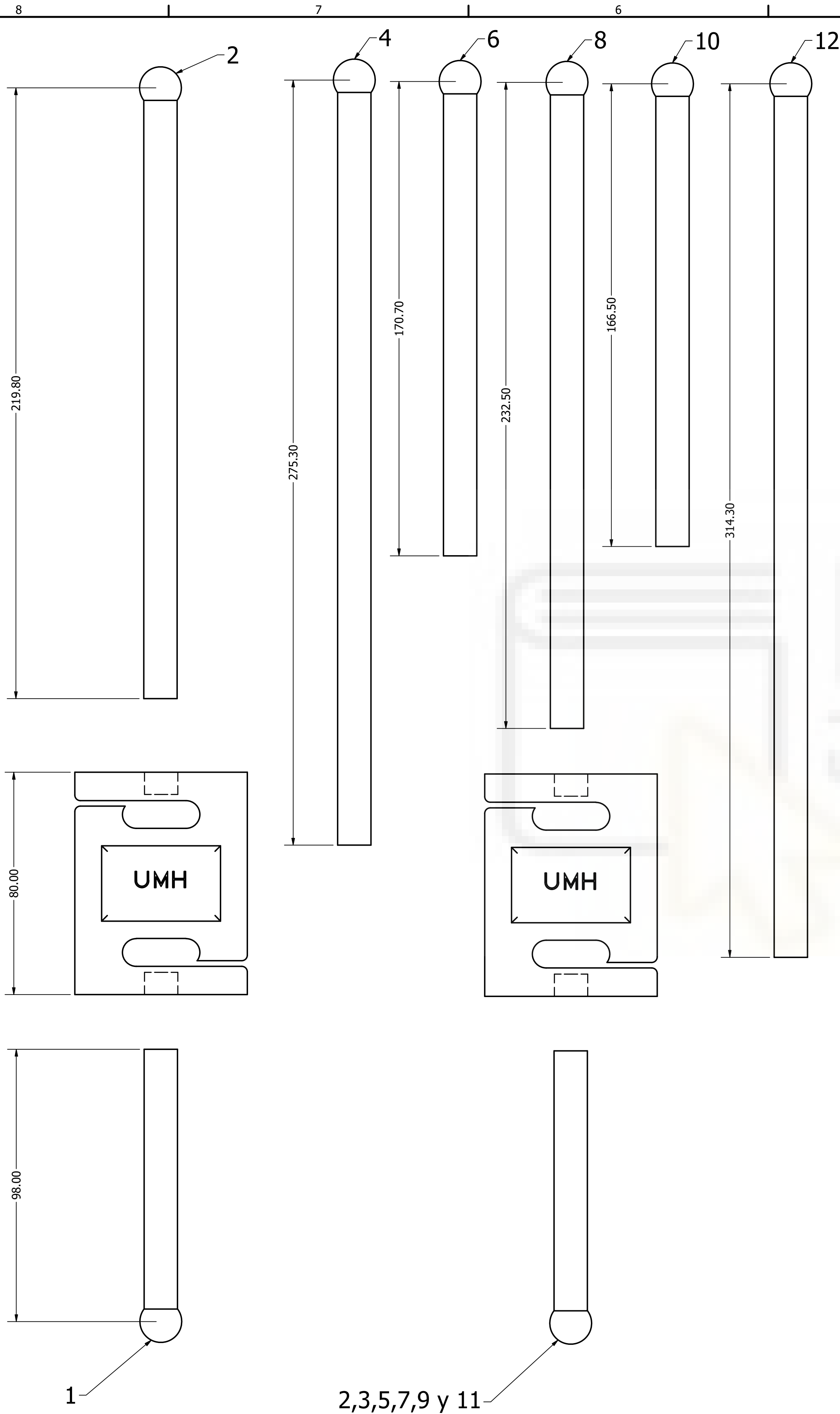
por_debajo_de

%%
% PRUEBA es la prueba a representar obtenida de los cálculos previos. Se
% obtiene del eje X
px = datos{5054,1}(1);
py = datos{5054,1}(2);
pz = datos{5054,1}(3);
alpha = datos{5054,1}(4);
beta = datos{5054,1}(5);
gamma = datos{5054,1}(6);
figure
plotear_robot(a,b,px,py,pz,alpha,beta,gamma)

```

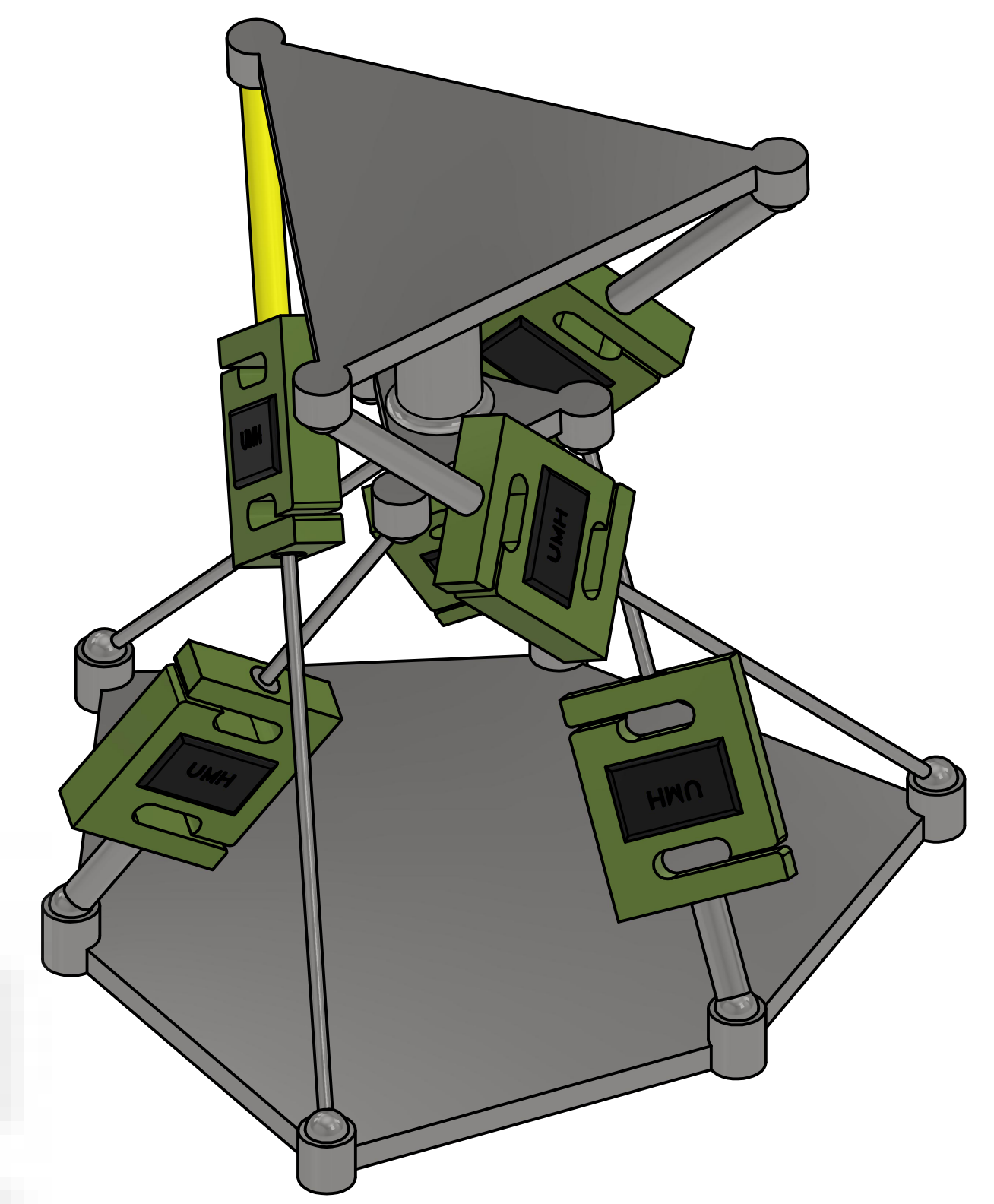
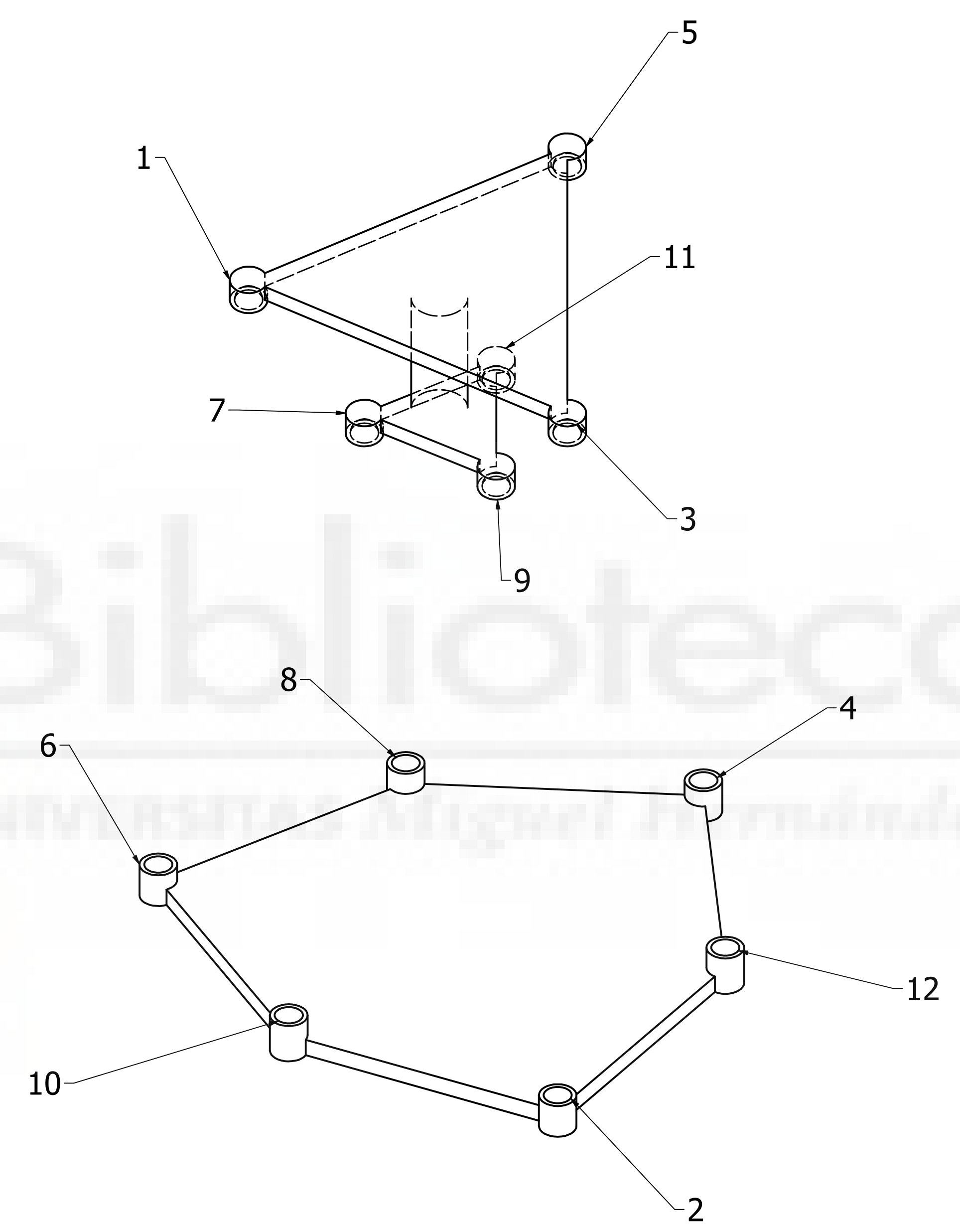
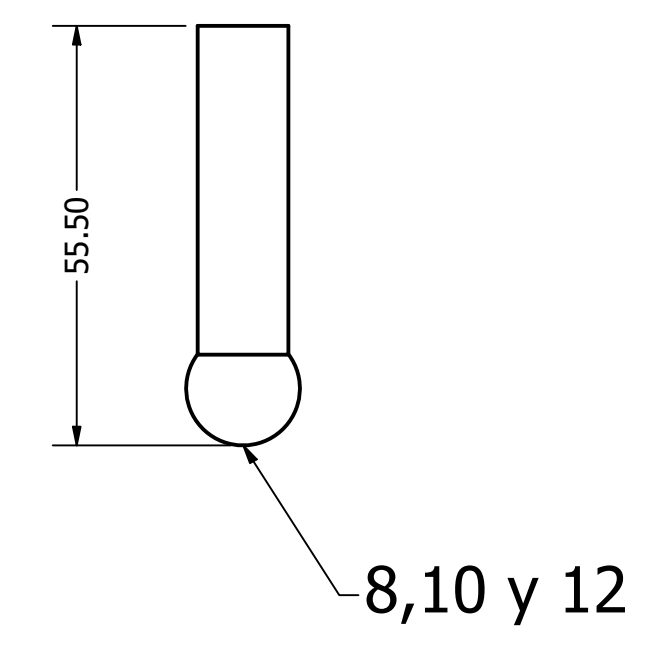
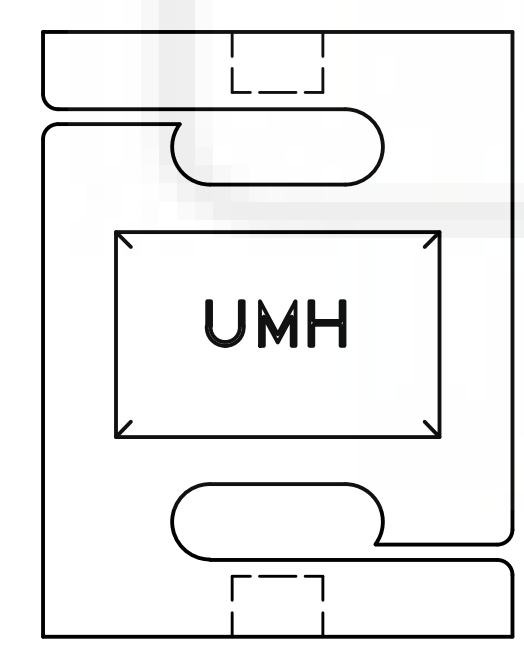
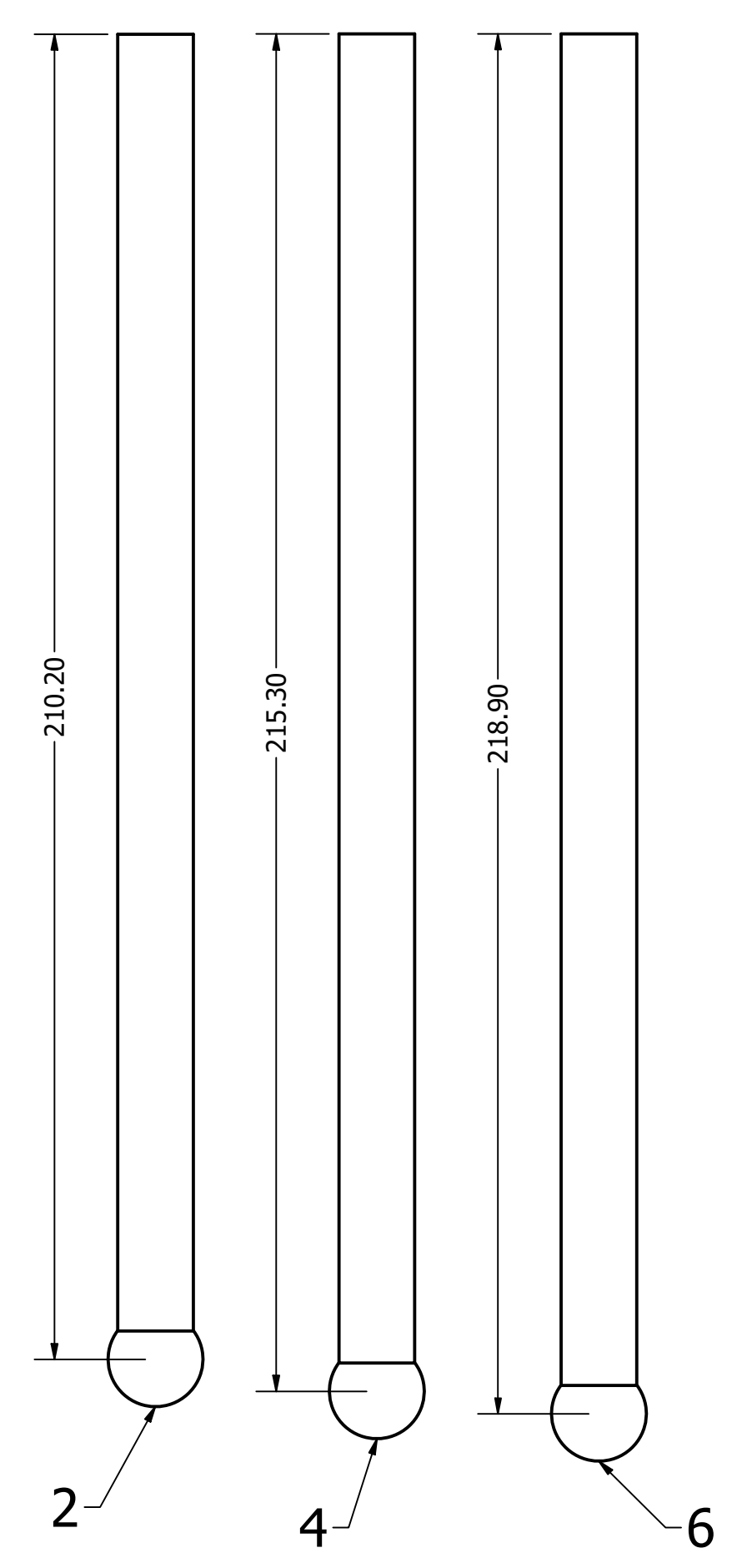
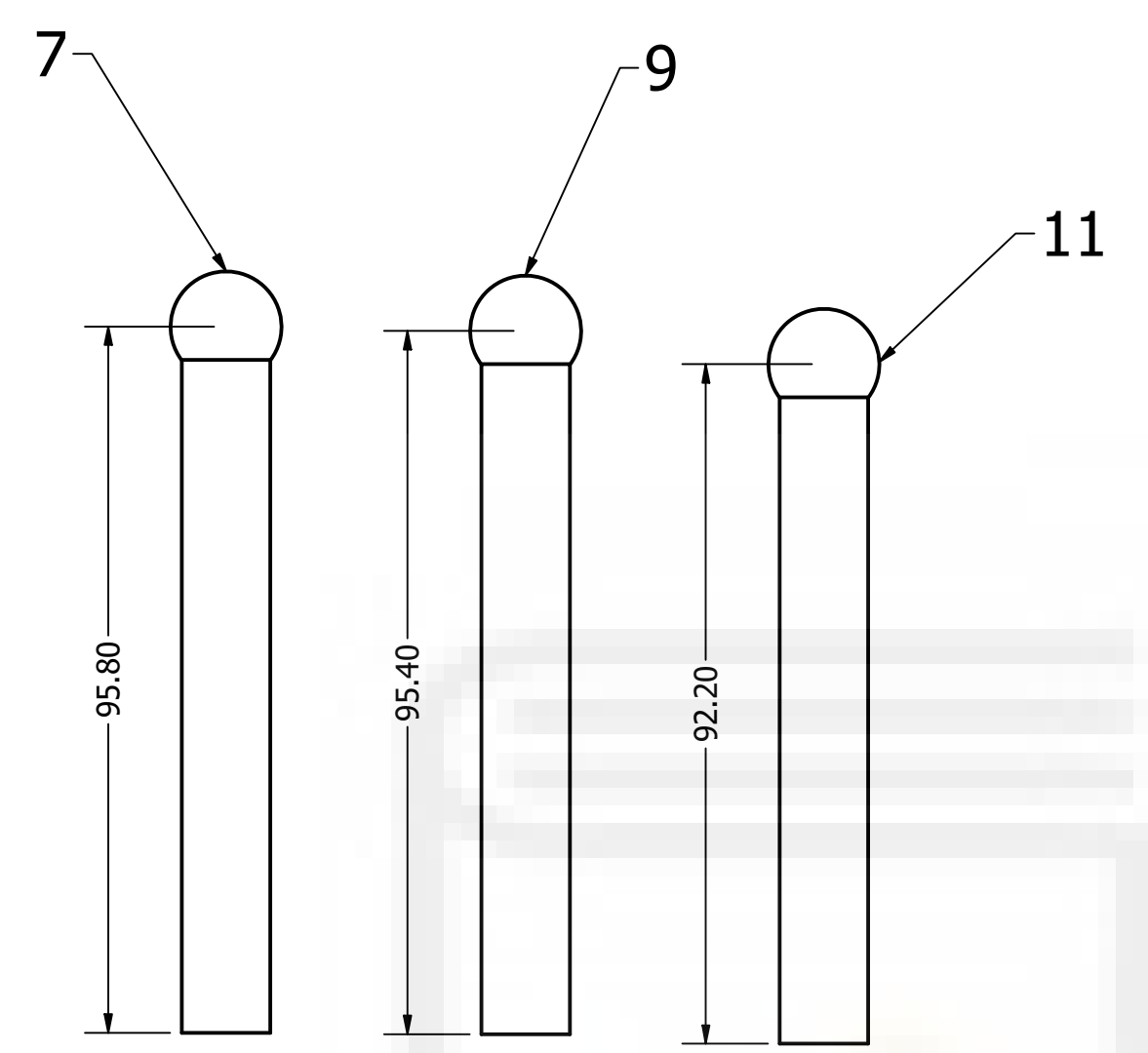
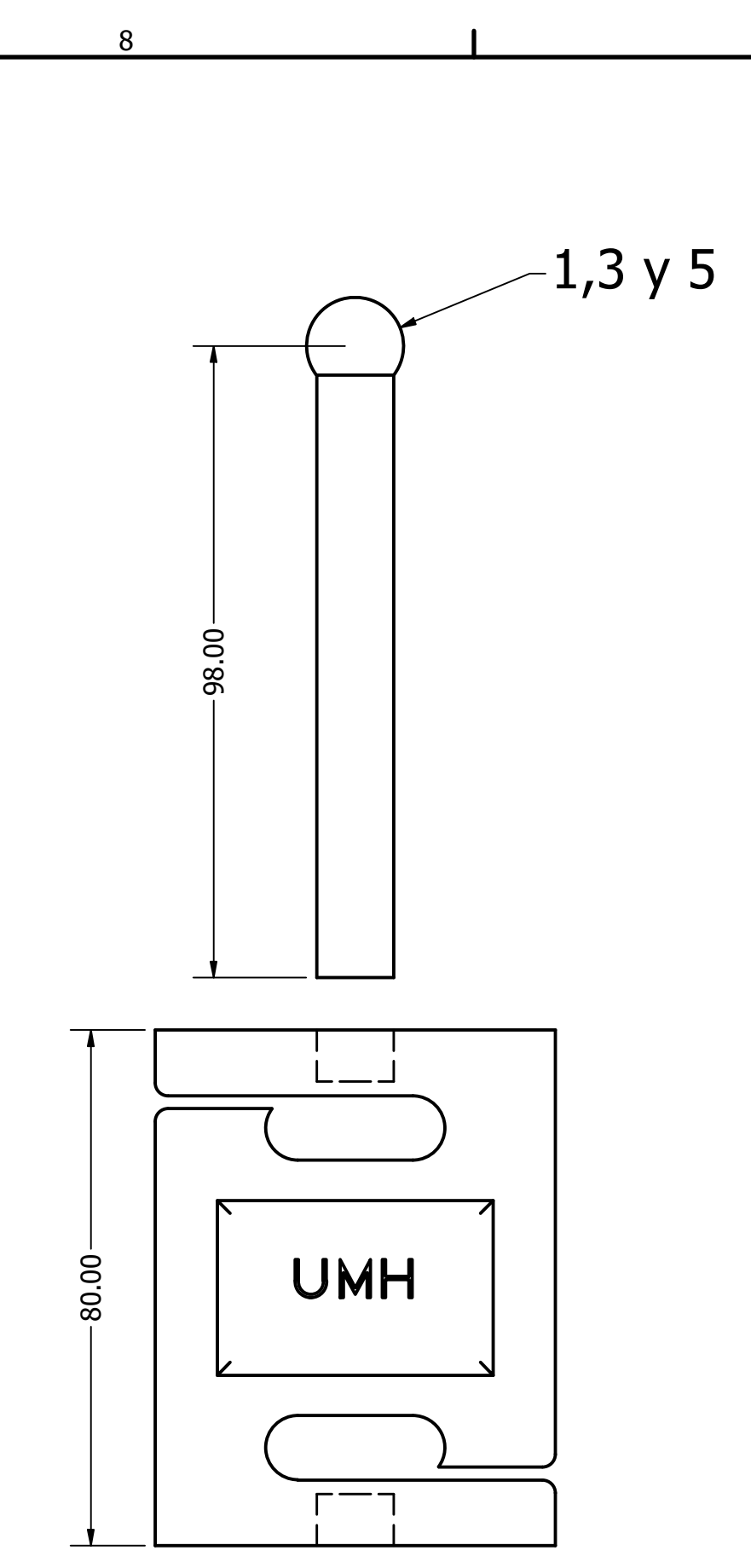
## 5.2. ARCHIVOS DE INVENTOR





DRAWN	<b>PEDRO CASTRO</b>	11/09/2024			
CHECKED	<b>ADRIÁN PEIDRO</b>		TITLE		
QA			<b>CONFIGURACIÓN COMERCIAL</b>		
MFG					
APPROVED			SIZE	DWG NO	REV
			D	Comercial	
			SCALE	1 / 2	SHEET 1 OF 1

Biblioteca  
UNIVERSITAS Miguel Hernández



DRAWN	<b>PEDRO CASTRO</b>	11/09/2024		
CHECKED	<b>ADRIÁN PEIDRO</b>		TITLE	
QA			<b>CONFIGURACIÓN TETRAÉDRICA</b>	
MFG				
APPROVED			SIZE	DWG NO
			D	<b>TETRAÉDRICA</b>
			SCALE	1 / 2
				REV
				SHEET 1 OF 1

### 5.3. HOJA TÉCNICA DE LA CÉLULA DE CARGA



## FEATURES

- Low profile
- Alloy steel construction with nickel plated treatment
- Capacity: 300kg
- Easy to install, stable & reliable

# RS PRO Alloy Steel S Beam Tension Load Cell

RS Stock No.: 204-2768



RS Professionally Approved Products bring to you professional quality parts across all product categories. Our product range has been tested by engineers and provides a comparable quality to the leading brands without paying a premium price.



### Product Description

S Beam Load Cell 614 offers a compact design for platforms, silos, or scales. The tension Load Cell has a wide range of capacities from 50kg to 1000kg, comes in steel alloy construction.

Capacities: 300kg

Material: Alloy Steel

Dimension: 62.1\*18\*80mm

Rated Output: 2.0mV/V

Protection Class: IP66

### Applications

Hopper scales, Crane scales, Tension control and machine testing Batching systems, blending & mixing systems, belt scales, etc.

### Electrical Specifications

Rated Capacities	300kg
Accuracy Class	C3
Rated Output	2.0±0.002 mV/V
Zero Balance	±0.0200 mV/V
Non-linearity	0.03%R.O.
Hysteresis	0.03%R.O.
Repeatability	0.03%R.O.
30mins Creep	0.03%R.O.
30mins Return	0.03%R.O.
Safe Overload	120 %R.O.
Ultimate Overload	200 %R.O.
Temperature Effect on Output	0.002 %R.O./°C
Temperature Effect on Zero	0.003 %R.O./°C
Input Impedance	400±20 Ω
Output Impedance	350±3 Ω

## Load Cells

Insulation Impedance	≥5000 MΩ/(50VDC)
Recommended Excitation	4~12 VDC
Maximum Excitation	15 VDC
Operating Temperature Range	-20~60 °C
Construction	Steel Alloy
Cable	φ5×3m
Mode of Connection	Red (EXC+) , Black (EXC-) , Green (SIG+) , White (SIG-)

### Protection Category

Waterproof class	IP66
------------------	------

### Classification

eCl@ss (Version)	C3
------------------	----

### Approvals

Declarations	ROHS, OIML
--------------	------------

### Additional Information

Custom Tariff Number	84239000
----------------------	----------

### Dimensions:

Unit (mm)

