



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL MECATRÓNICA

PROYECTO DE UNA PLATAFORMA DE STEWART PARA LA SIMULACIÓN DE VIBRACIONES MECÁNICAS

Memoria para optar al Título de
Ingeniero Civil Mecatrónico

Profesor Guía:
Ambrosio Martinich Leal

VÍCTOR MATÍAS CASTRO VARAS

CURICÓ-CHILE

2018

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Two circular official stamps and handwritten signatures in blue ink. The left stamp is from the 'DIRECCIÓN SISTEMA DE BIBLIOTECAS UNIVERSIDAD DE TALCA' and the right stamp is from the 'SISTEMA DE BIBLIOTECAS CAMPUS CURICO'.

Curicó, 2020

Resumen

En el presente documento se tuvo como fin proyectar una plataforma de tipo Stewart de 6 grados de libertad. Esto significó formular los modelos cinemático y dinámico del manipulador, además de elaborar los planos de construcción, interfaz gráfica para el control y por último presentar opciones de controladores para el movimiento de los actuadores de este.

En primer lugar, se presenta la descripción de una plataforma de Stewart, para luego dar paso al análisis cinemático de esta. Utilizando la cinemática inversa se calculan los largos de las piernas que generan los cambios de posición y orientación de la plataforma móvil superior, para así después obtener los ángulos de giro de los actuadores del manipulador y finalmente realizar una interfaz gráfica que facilite la interacción entre el usuario y la máquina para controlar la posición y orientación de la plataforma.

Entre los resultados a presentar, además del informe que da cuenta del diseño y la modelación del dispositivo, se encuentran la interfaz gráfica ya mencionada, los planos realizados en software CAD (Autodesk Inventor 2018) y la evidencia gráfica de algunas simulaciones de respuesta de movimiento de la plataforma superior al variar el ángulo de los actuadores, realizadas también con el software CAD Autodesk Inventor 2018.



Dedicado a mis padres, hermanos y seres queridos.

Agradecimientos

La realización de este documento representa el término de un ciclo, de una meta, la cual no podría haber logrado sin contar con el apoyo de muchas personas, familiares, amigos, compañeros, profesores, etc. Más directamente hablando:

- A mi padre Juan Carlos, mi madre Mariam, mis hermanos Carlos y Francisco. Apoyo incondicional y pilares para conseguir logros y metas en mi vida.
- A Fernando, Alex, Sergio, Juan Pablo, Mario y Danilo quienes siendo estudiantes mayores de la universidad me acogieron y acompañaron durante mis primeros años en esta.
- A mis amigos del colegio, quienes siguen presentes hoy en día, Rony, Daniel, Carolina, Fady, Diego A., Diego G., Nicolás, Ricardo, Karina y Carla quienes fueron parte fundamental de mi crecimiento como estudiante y formación como persona.
- A los amigos que me entrego la universidad Nataly, Luis, Felipe, Edmundo, Héctor, Mario, Carlos, Juan y Pedro, personas con los que compartí estudios, fiestas, trabajo, deportes y siempre tuvieron una palabra de apoyo.
- A mis profesores, don Ambrosio Martinich, quien me guio y apoyó en la realización de este proyecto. Dr. Johan Guzmán quien a pesar de demostrarle mi irresponsabilidad siempre confió en mis habilidades. Y por último, pero no menos importante, a mi entrenador don Jorge Muñoz quien siempre tuvo una palabra de apoyo y otra de reto ante los errores que cometí por mi irresponsabilidad.

Tabla de Contenido

RESUMEN	II
TABLA DE CONTENIDO	V
LISTA DE TABLAS	VII
LISTA DE FIGURAS	VIII
NOMENCLATURA	X
CAPÍTULO 1 INTRODUCCIÓN	1
INTRODUCCIÓN GENERAL	1
ESTADO DEL ARTE	3
1. <i>Robótica</i>	3
2. <i>Partes del robot</i>	8
3. <i>Grados de libertad</i>	10
4. <i>Manipuladores</i>	12
5. <i>Manipuladores paralelos</i>	13
6. <i>Ejemplos de aplicación de manipuladores paralelos:</i>	14
HIPÓTESIS DE TRABAJO.....	16
OBJETIVOS	16
1. <i>Objetivo general</i>	16
2. <i>Objetivos específicos</i>	16
ALCANCES Y LIMITACIONES	17
TEMARIO.....	17
CAPÍTULO 2: LA PLATAFORMA DE STEWART	18
INTRODUCCIÓN	18
DESCRIPCIÓN DE LA PLATAFORMA	18
DESCRIPCIÓN DE LA NECESIDAD.....	20
1. <i>Cinemática de la plataforma</i>	20
2. <i>Ecuaciones de posición</i>	22
3. <i>Mecanismo de movimiento</i>	27
4. <i>Secuencia de control</i>	32
CAPÍTULO 3: DISEÑO MECÁNICO DE LA PLATAFORMA E INTERFAZ GRÁFICA DE CONTROL	33
INTRODUCCIÓN	33
DISEÑO MECÁNICO.....	33
1. <i>Base y plataforma superior</i>	33
2. <i>Mecanismo de movimiento</i>	37
3. <i>Ensamblaje de la plataforma</i>	40
INTERFAZ GRÁFICA.....	45
1. <i>Tkinter</i>	45
2. <i>Label</i>	46
3. <i>Scale</i>	46
4. <i>Button</i>	47
CAPÍTULO 4: RESULTADOS DE LA SIMULACIÓN	50
INTRODUCCIÓN	50

CAPÍTULO 5: CONCLUSIONES60
 TRABAJOS FUTUROS.....60
REFERENCIAS.....61
ANEXO I – CÓDIGO DE LA INTERFAZ63
ANEXO II – PLANOS MECÁNICOS 71



Lista de Tablas

Tabla 1. Clasificación de los robots según T.M.KnaseI [8].	7
Tabla 2. Clasificación de los robots según la AFRI [9].	8
Tabla 3. Comparación de manipuladores [12]	13
Tabla 4. Resultados prueba simulación 1	56
Tabla 5, Resultados prueba simulación 2	56
Tabla 6. Resultados prueba simulación 3	56



Lista de Figuras

Fig. 1. Plataforma de Stewart.....	1
Fig. 2. Robot.....	3
Fig. 3. Gallo de Estrasburgo.....	4
Fig. 4. El pato, Jaces Vaucanson.....	4
Fig. 5. Muñeco de Pierre Jaquet.....	5
Fig. 6. Unimate de Devol y Engelberger.....	5
Fig. 7. Handy – Man de Mosher.....	6
Fig. 8. Robot SCARA.....	6
Fig. 9. Eslabón (A) – Articulación (B).....	9
Fig. 10. Tipos de articulaciones.....	11
Fig. 11. Manipulador paralelo (a) – Manipulador serial (b).....	12
Fig. 12. Closed – Kinematic Chain Mechanism (CKCM).....	14
Fig. 13. Telescopio UKIRT.....	15
Fig. 14. Endoscopio con 3 GDL.....	15
Fig. 15. Manipulador paralelo Delta.....	16
Fig. 16. Plataforma de Stewart.....	18
Fig. 17. Descripciones de distintas plataformas de Stewart.....	19
Fig. 18. Descripción de la plataforma de Stewart.....	19
Fig. 19. Vista superior de la placa base.....	22
Fig. 20. Vista superior de la plataforma superior.....	24
Fig. 21. Plataforma de Stewart convencional.....	26
Fig. 22. Plataforma de Stewart.....	27
Fig. 23. Mecanismo biela – manivela.....	27
Fig. 24. Plataforma de Stewart, sistema biela – manivela.....	28
Fig. 25. Mecanismo biela – manivela en 3 dimensiones.....	29
Fig. 26. Reflexión de manivelas pares (even) e impares (odd).....	30
Fig. 27. Plataforma base (vista isométrica).....	34
Fig. 28. Plataforma base (vista superior).....	35
Fig. 29. Plataforma superior (vista isométrica).....	36
Fig. 30. Plataforma superior (vista superior).....	36
Fig. 31. Modelo base para servomotor.....	37
Fig. 32. Modelo servomotor [20].....	37
Fig. 33. Modelo de la biela.....	38
Fig. 34. Modelo de la manivela.....	38
Fig. 35. Datos de construcción de la rótula.....	39
Fig. 36. Modelo de la rótula.....	39
Fig. 37. Ensamblaje base actuador – actuador – manivela.....	40
Fig. 38. Ensamblaje rótula – biela – rótula.....	41
Fig. 39. Ensamblaje base – actuador – biela – manivela.....	41
Fig. 40. Ensamblaje plataforma de Stewart.....	42
Fig. 41. Plataforma completa (nueva medida b).....	43
Fig. 42. Plataforma completa (nueva medida d).....	43
Fig. 43. Medición de ángulos β	44
Fig. 44. Ejmplo de Scale.....	46
Fig. 45. Interfaz gráfica.....	47

Fig. 46. Micro Maestro 6-channel.....	49
Fig. 47. Acceso al entorno de simulación dinámica.....	50
Fig. 48. Simulación dinámica.....	50
Fig. 49. Uniones estándar.....	51
Fig. 50. Opciones de revolución.....	51
Fig. 51. Posición impuesta.	52
Fig. 52. Menú del simulador.	53
Fig. 53. Plataforma superior M1.	53
Fig. 54. Plataforma de Stewart (modelo 1).	53
Fig. 55. Plataforma superior M2.	54
Fig. 56. Plataforma de Stewart (modelo 2).	54
Fig. 57. Plataforma superior M3.	55
Fig. 58. Plataforma de Stewart (modelo 3).	55
Fig. 59. Plataforma de Stewart.....	57
Fig. 60. Eje coordenado Autodesk Inventor.....	57
Fig. 61. Medidas prueba 3, superior M3.....	58
Fig. 62. Desplazamiento prueba 3, superior M3.....	58



Nomenclatura

B_i	:	Puntos de conexión de la pierna i con la base
T_i	:	Puntos de conexión de la pierna i con la plataforma superior
S_i	:	Largo de la pierna i que conecta la plataforma superior con la base
M_i	:	Conexión biela i con manivela i
m_i	:	Largo de la manivela i
b_i	:	Largo de la biela i
${}^P R_B$:	Matriz de transformación homogénea
α	:	Ángulo de la manivela con respecto a la horizontal
β	:	Ángulo del plano de la manivela relativo al eje x
W_i	:	Ancho de pulso para el servo actuador i
W_i^0	:	Ancho de pulso inicial para el servo actuador i
m	:	Número de grados de libertad del sistema
λ	:	Grados de libertad del entorno donde se encuentra el mecanismo
n	:	Número de eslabones fijos del mecanismo, incluyendo base y superior
j	:	Número de juntas del mecanismo
f_i	:	Grados del movimiento relativos por junta
I_f	:	Número de grados de libertad pasivos del mecanismo

Capítulo 1 Introducción

Introducción General

La plataforma de Stewart [1] corresponde a un manipulador robótico paralelo que consta de una placa base y una plataforma superior. Ambas con forma de hexágono semirregular, vinculadas entre sí mediante 6 piernas consistentes en 6 mecanismos biela – manivelas espaciales las que, a su vez, se componen de parejas barras rígidas (m_i y b_i).

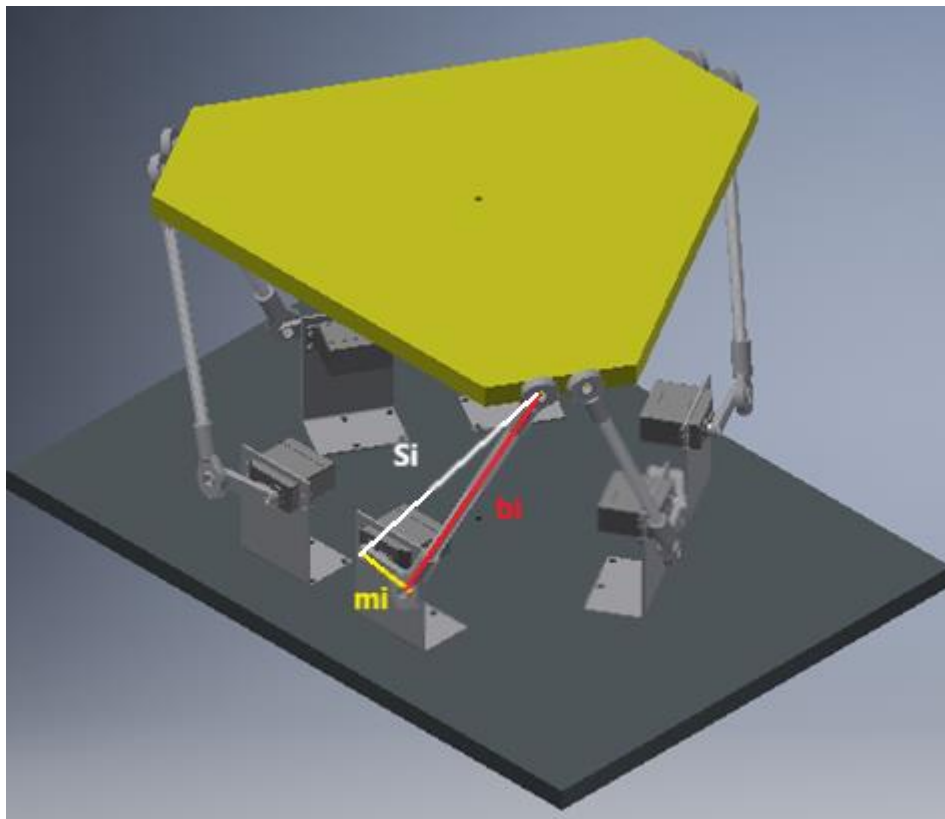


Fig. 1. Plataforma de Stewart.

Este manipulador tiene 6 grados de libertad y el método de control de posición y orientación de su plataforma superior, viene dado por la variación de las distancias entre los puntos de conexión de las piernas con la placa base y con la plataforma superior. Esas distancias se calculan mediante la cinemática del manipulador, y al tener las piernas constituidas por mecanismos de biela – manivelas, se requiere realizar un cálculo de los ángulos de rotación de los actuadores acoplados a estas.

Entre las aplicaciones existentes para las plataformas de Stewart, las cuales se analizarán más adelante, se pueden mencionar simuladores de vuelo, simuladores de manejo de vehículos y aplicaciones en manipuladores de cirugía mínimamente invasiva.

Para la construcción, y posterior simulación, del modelo 3D de la plataforma se utilizará el software Autodesk Inventor 2018 (Licencia de estudiante) con el que se realizarán pruebas a 3 distintos puntos de conexión de las piernas con la plataforma superior.

Finalmente se entregará una interfaz gráfica, generada en código Python, la cual realiza el cálculo de la longitud de las 6 piernas, usando la cinemática inversa del mecanismo que se estudiará más adelante. De esta manera se obtendrá en tiempo real el valor de la longitud de piernas y ángulos de rotación de los actuadores que las impulsan.



Estado del Arte

1. Robótica

Para comenzar a hablar de robótica es necesario considerar el significado de la palabra “robot”, para el que se encuentran las siguientes definiciones:

- **Diccionario Webster:** “Una máquina que se asemeja a una criatura viviente al ser capaz de moverse independientemente y realizar acciones complejas” [2]
- **Real academia:** “Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a personas” [3]

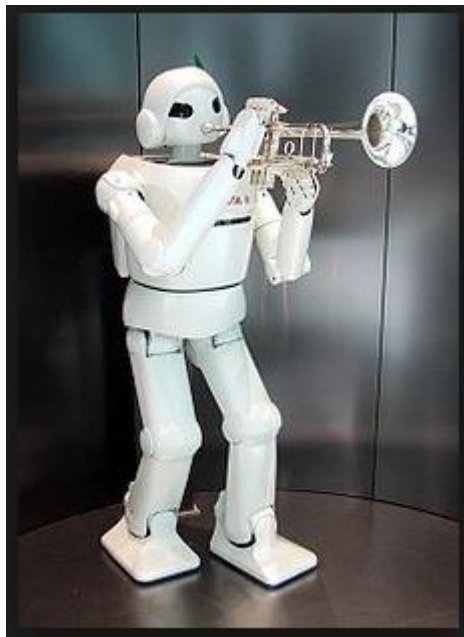


Fig. 2. Robot.

Aunque se piense lo contrario, los robots llevan mucho tiempo entre nosotros, prácticamente son anteriores a nuestro milenio. El primer “robot”, del que se tiene conocimiento, fue construido durante el siglo I por Herón de Alejandría. El cual simulaba ser un actor de teatro.

Dentro de los autómatas que marcaron un hito en la historia se pueden mencionar [4] [5]:

- El *Gallo de Estrasburgo*, el autómata más antiguo que se conserva en la actualidad. Formando parte de un reloj este gallo movía las alas y el pico al dar las horas.

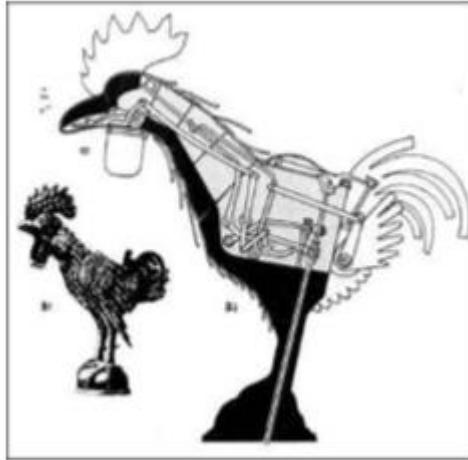


Fig. 3. Gallo de Estrasburgo.

- El *Pato de Vaucanson*, creado por Jaces Vaucanson. Este autómata podía alargar el cuello para tomar un grano de la mano y luego lo tragaba y digería. Además, podía beber y graznar. Los alimentos los digería por disolución y se conducían hacia el ano a través de tubos, para así poder evacuarlos.

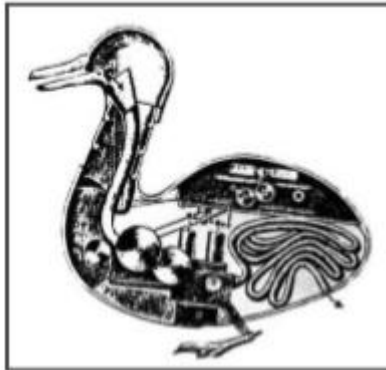


Fig. 4. El pato, Jaces Vaucanson.

- El relojero suizo Pierre Jaquet Droz y sus hijos construyeron diversos muñecos capaces de escribir, dibujar y tocar melodías.



Fig. 5. Muñeco de Pierre Jaquet.

A pesar de esto, el robot como “*máquina*” lleva un desarrollo independiente del término “*robot*”. Donde podemos mencionar:

- Telemanipulador con dispositivo de control mecánico (Goertz del Argonne National Laboratory)
- Telemanipulador con dispositivo de servo control eléctrico (Goertz)
- Devol y Engelberger comienzan a aplicar dispositivos robóticos en la industria e instalan el primer Unimate en una planta de General Motors.

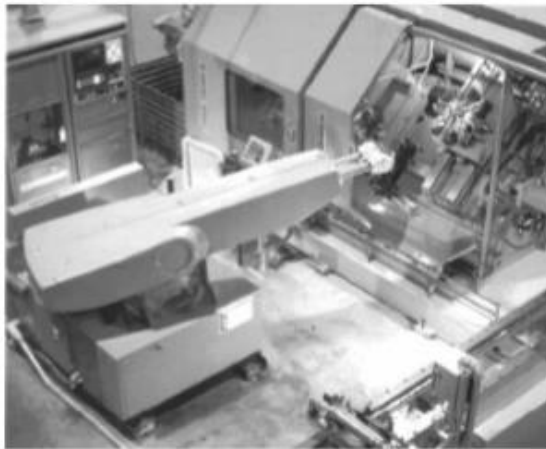


Fig. 6. Unimate de Devol y Engelberger.

- Handy – man, General Electric. Creado por Ralph Mosher, nace ante la necesidad de General Electric de tener un manipulador dedicado para manejar experimentos con un motor de avión atómico [6].

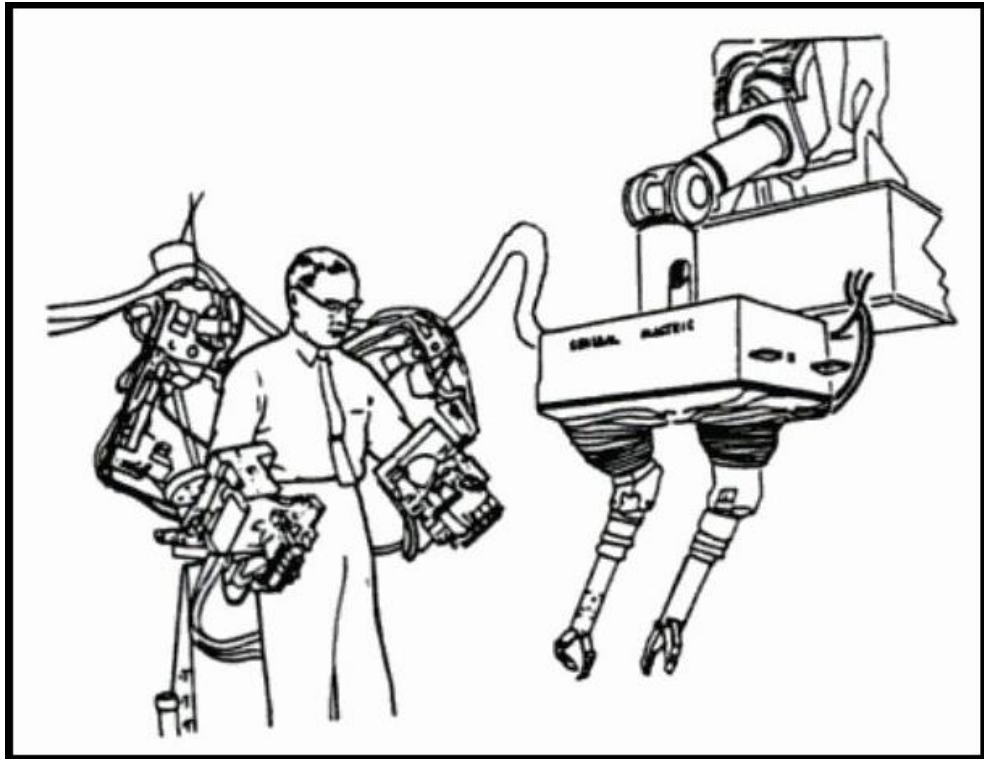


Fig. 7. Handy – Man de Mosher.

- El primer robot *SCARA* fue creado por el profesor Hiroshi Makino. La simplicidad del robot *SCARA* era brillante ya que al realizar menos movimiento podía lograr una mayor velocidad y precisión. El acrónimo significa “*Selective Complaint Assembly Robot Arm*” [7].

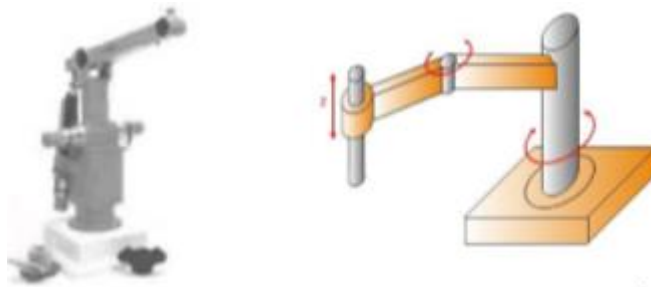


Fig. 8. Robot SCARA.

Los robots industriales se pueden clasificar en tres tipos (generaciones) según su automatización:

- **1ra Generación:** Repiten la tarea programada secuencialmente, sin importar las variaciones del entorno.
- **2da Generación:** Adquiere información limitada de su entorno y actúa en consecuencia.
- **3ra Generación:** Su programación se realiza mediante el empleo de un lenguaje natural. Posee capacidad para la planificación automática de tareas.

Tabla 1. Clasificación de los robots según T.M.Knasek [8].				
Generación	Nombre	Tipo de Control	Grado de Movilidad	Usos más frecuentes
1 (1982)	Pick & Place	Fines de carrera, aprendizaje	Ninguno	Manipulación, servicio de máquinas
2 (1984)	Servo	Servocontrol, trayectoria continua, progr. condicional	Desplazamiento por vía	Soldadura, pintura
3 (1989)	Ensamblado	Servos de precisión, visión, tacto	Guiado por vía	Ensamblado, desbardado
4 (2000)	Móvil	Sensores inteligentes	Patas, ruedas	Construcción, mantenimiento
5 (2010)	Especiales	Controlados con técnicas de IA	Andante, saltarín	Militar, espacial

En la Tabla 1 se puede observar otra forma de clasificación de robots, la cual llega hasta 5ta generación, tomando en cuenta el tipo de control y grado de movilidad de estos.

En la siguiente Tabla se muestra un tercer tipo de clasificación de robots la cual no se estandariza por generación, si no por tipo de robot (A – D).

Tabla 2. Clasificación de los robots según la AFRI [9].	
Tipo A	Manipulador con control manual o telemando
Tipo B	Manipulador automático con ciclos preajustados; regulación mediante fines de carrera o topes; control por PLC; accionamiento neumático, eléctrico o hidráulico.
Tipo C	Robot programable con trayectoria continua o punto a punto. Carece de conocimiento sobre su entorno.
Tipo D	Robot capaz de adquirir datos de su entorno, readaptando su tarea en función de estos.

2. Partes del robot

El cuerpo del robot se constituye de 2 partes, los eslabones y las articulaciones. Lo anterior considerando que la mayor parte de los manipuladores prácticos utilizan pares de revolución.

Los eslabones representan a las partes rígidas del robot que tienen movimiento, mientras que las articulaciones son las encargadas de unir los eslabones y además permitir el movimiento de estos. En la figura siguiente se pueden observar, señalados, un eslabón y una articulación de un robot.

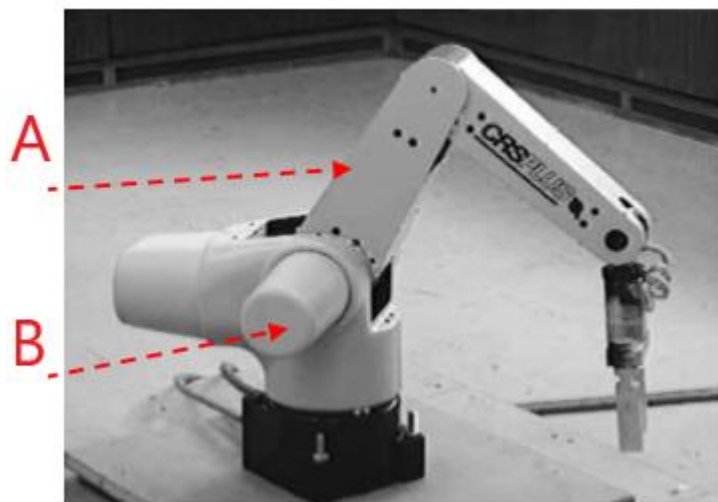


Fig. 9. Eslabón (A) – Articulación (B)

Además, para generar el movimiento del robot existen 2 partes importantes, los sensores (herramientas de percepción) y los actuadores (herramientas de ejecución).

Los sensores cumplen la función de recopilar información de lo que ocurre a su alrededor, esta información puede ser de tipo:

1. **Propio – percepción:** Permiten saber dónde se encuentran sus articulaciones (codificadores, odómetro, orientación, acelerómetro, tacómetro)
2. **Percepción de fuerza:** Permiten detectar fuerzas y obtener movimientos obedientes.
3. **Percepción táctil:** Permiten calcular deformación, utilizando un material elástico y midiendo la distorsión del material que está en contacto.
4. **Sonar:** Usando el tiempo que tarda un impulso sonoro producido por el sensor en llegar al objeto y ser reflejado por este, permite evitar obstáculos y seguir pistas de un blanco cercano.
5. **Cámara:** Se utiliza para captar escenas del medioambiente.

Los actuadores son herramientas para la ejecución, convirtiendo los comandos generados por el software en movimientos físicos para el robot. Por lo general son motores eléctricos o cilindros hidráulicos o neumáticos. Los actuadores se pueden clasificar de 2 maneras:

1. **Locomoción:** modifican la ubicación del robot respecto del ambiente (caminante estable, dinámicamente estables, ruedas y llantas).
2. **Manipulación:** desplazan objetos del entorno. Existen dos tipos de movimientos: giratorios (alrededor de un eje) y prismáticos (movimientos lineales).

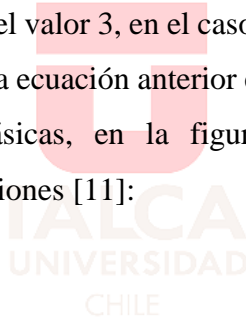
3. Grados de libertad

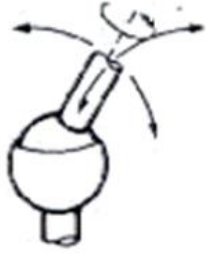
Como se mencionó anteriormente los robots generalmente están compuestos de eslabones unidos a través de articulaciones, conformando pares de revolución, que permiten un movimiento giratorio relativo entre dos eslabones contiguos.

Cada uno de los movimientos independientes que puede realizar cada articulación es denominada “*Grado de Libertad*” (GDL). El número de GDL corresponde al número de variables independientes que fijan la situación del órgano terminal. Al final el número de GDL se puede calcular mediante la siguiente ecuación, correspondiente al criterio de Kutzbach – Grübler [10]:

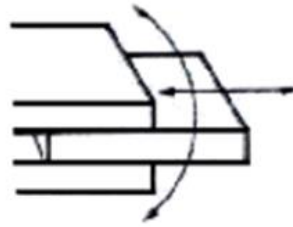
$$m = \lambda(n - j - 1) + \sum_{i=1}^j f_i - I_f \quad (1)$$

Donde el valor de λ puede tomar el valor 3, en el caso de manipuladores en 2D, o 6, en el caso de manipuladores espaciales o 3D. Con la ecuación anterior es posible calcular los GDL del robot. Se conocen 6 tipos de articulaciones básicas, en la figura siguiente se pueden observar más detalladamente los seis tipos de articulaciones [11]:

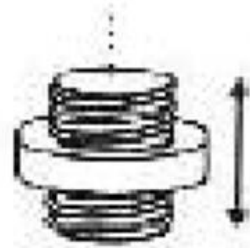




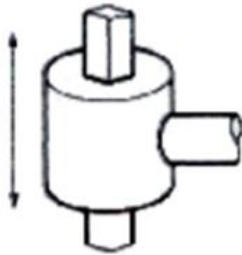
Esférica (rótula)
3 GL



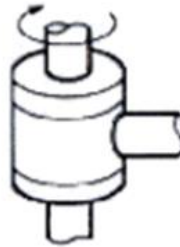
Planar
2 GL



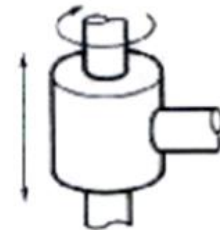
Tornillo
1 GL



Prismática
1 GL



Rotacional
1 GL



Cilíndrica
2 GL

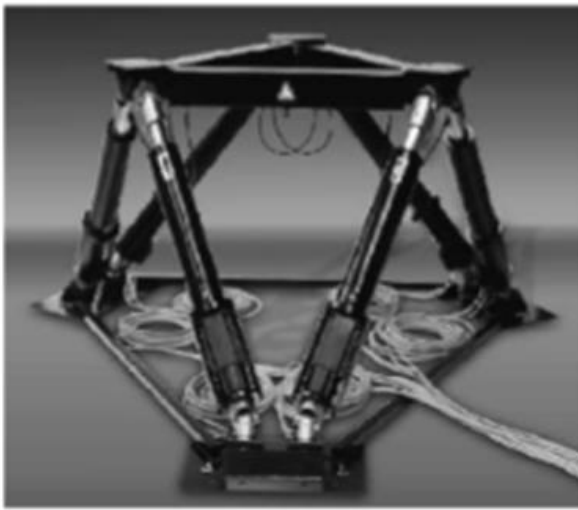
Fig. 10. Tipos de articulaciones.



4. Manipuladores

De acuerdo con la estructura cinemática, los manipuladores se pueden clasificar en seriales, paralelos o híbridos. Un manipulador en serie consta de una estructura cinemática que tiene forma de cadena abierta, es decir, un ensamble de eslabones y articulaciones que proporcionan un movimiento de salida en respuesta a un movimiento de entrada.

Un manipulador paralelo se encuentra formado por cadenas cinemáticas cerradas. Y por último un manipulador híbrido es formado por cadenas cinemáticas abiertas y cerradas.



(a)



(b)

Fig. 11. Manipulador paralelo (a) – Manipulador serial (b).

La comparación de estos tipos de manipuladores viene dada en términos de sus problemas mecánicos y de control. En términos mecánicos los manipuladores en serie se componen de actuadores y eslabones que dan lugar a momentos de inercia relativamente altos. En cambio, en los manipuladores paralelos todos los actuadores se encuentran en la base, permitiendo una posible reducción de la masa de sus partes móviles, implicando una característica dinámica superior a los manipuladores en serie.

Los errores de posicionamiento son acumulativos en los manipuladores en serie, mientras que en los manipuladores paralelos estos errores son el promedio de todos los errores. En la siguiente tabla se muestra una comparación cualitativa de las características mecánicas de ambos manipuladores.

Tabla 3. Comparación de manipuladores [12]				
Nivel	Características	Manipulador serial	Manipulador paralelo	
Mecanismo	Inercia	Grande	Pequeña	
	Volumen de trabajo	Grande	Pequeño	
	Apariencia	Antropomórfica	Base estructural	
	Fabricación	Difícil	Fácil	
Control	Control de posición en el espacio de trabajo	Difícil	Fácil	
	Control de fuerza en el espacio de trabajo	Fácil	Difícil	
	Detección de fuerzas	Difícil	Fácil	
	Error de posición	Acumulado	Media	
	Error de control de fuerzas	Media	Acumulado	
	Cerca de los puntos	Degeneración en el control de fuerza		Disminución de la precisión de posicionamiento
		Movimientos grandes en el actuador		Fuerza grande en el actuador
Dinámica		Complicada	Mucho más complicada	

5. Manipuladores paralelos

Un manipulador paralelo consiste en un generador de efectos con n grados de libertad y una base fija que se conectan entre sí por al menos dos cadenas cinemáticas independientes. Un

manipulador paralelo generalizado es un mecanismo de cierre de una cadena cinemática cuyo efector final está conectado a la base a través de varias cadenas cinemáticas independientes.

Las principales características de estos manipuladores son las siguientes:

- Al menos dos cadenas de efectores.
- Cada una de estas cadenas contiene un mínimo de un actuador simple.
- Hay un sensor adecuado para medir el valor de las variables relacionadas con el desempeño.
- Un mínimo de dos cadenas permite una mejor distribución de carga en las cadenas.

6. Ejemplos de aplicación de manipuladores paralelos:

- **Aplicaciones exclusivas:** El robot CKCM (closed-kinematic chain mechanism) estudiado por la NASA (Goddard Space Flight Center), por Nguyen y su equipo [13].

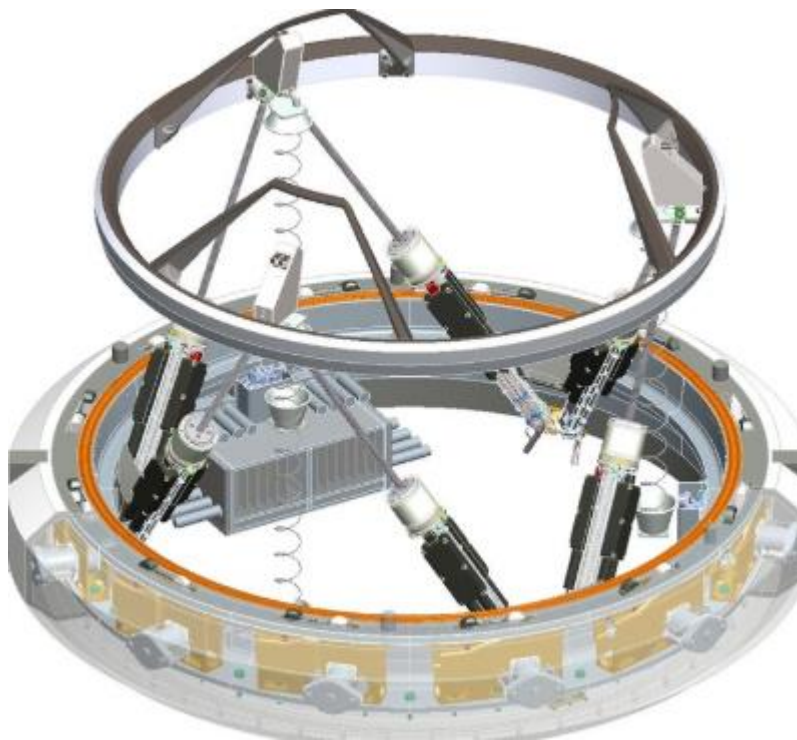


Fig. 12. Closed – Kinematic Chain Mechanism (CKCM).

Un robot hexápodo desarrollado por el instituto Max Planck para ser usado como telescopio UKIRT (United Kingdom Infra – Red Telescope) [14].

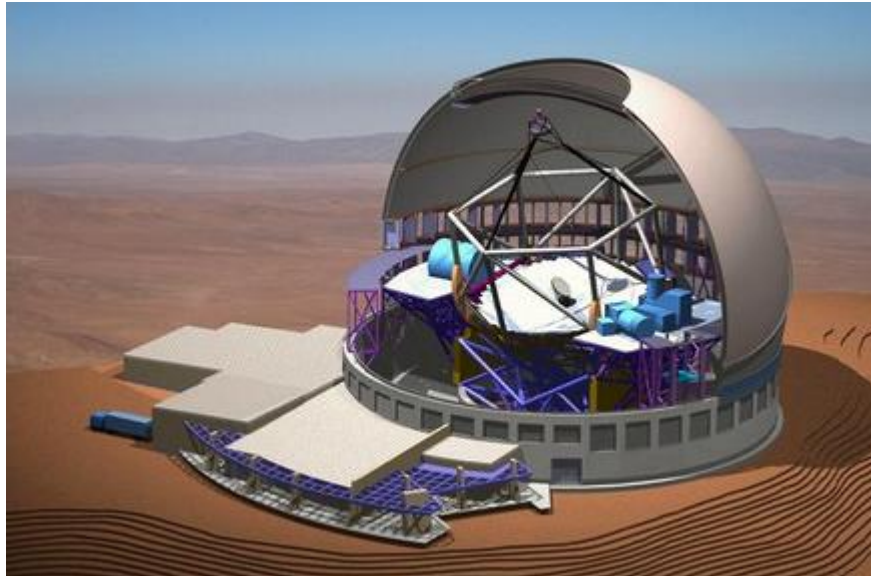


Fig. 13. Telescopio UKIRT.

- **Aplicaciones médicas [15]:** En esta área se puede destacar un endoscopio con un dispositivo de fijación activa conformado por un robo de tres grados de libertad, accionados por hilos; construido por Wendlat. Otra aplicación en la medicina es para asistencia de personas con deficiencias en el movimiento de sus brazos.



Fig. 14. Endoscopio con 3 GDL.

- **Aplicaciones industriales:** Gracias a su alta precisión de posicionamiento y gran rigidez, los manipuladores paralelos son bastante usados industrialmente, principalmente en operaciones de montaje y desmontaje, manipulación de piezas y aplicaciones de soldadura. El modelo Delta representa el mejor ejemplo de manipulador con estructura paralela utilizado industrialmente en tareas de manipulación rápida de objetos.



Fig. 15. Manipulador paralelo Delta

Hipótesis de Trabajo

Es posible proyectar una plataforma de Stewart y realizar de simulaciones de vibraciones mecánicas.

Objetivos

1. Objetivo general

Diseñar un manipulador paralelo, del tipo plataforma de Stewart, de 6 grados de libertad para la simulación de vibraciones mecánicas.

2. Objetivos específicos

- Definir necesidades a cubrir con el manipulador.
- Formular modelos matemáticos, cinemática y dinámica, de la plataforma de Stewart.
- Diseñar el sistema mecánico, con planos de fabricación, en 3D con ayuda de software CAD (Autodesk Inventor 2018).
- Crear interfaz usuario – manipulador para la realización de ensayos de vibraciones a través de

lenguaje Python (IDLE Eclipse Java Oxygen).

Alcances y Limitaciones

En esta Memoria de Título se compromete el diseño mecánico de la plataforma, realizado con los cálculos de construcción que se expondrán más adelante. Se compromete además una interfaz gráfica que muestre el valor de los ángulos que deben alcanzar los actuadores rotativos para lograr la orientación y posición deseada. No se realizará una selección del controlador del actuador, pero sí se darán a conocer algunas opciones viables para trabajos posteriores.

Temario

En el presente documento se muestran los pasos que se siguieron durante el desarrollo del proyecto. En el capítulo 1 se presenta una mirada global de la robótica y los manipuladores. En el capítulo 2 se abordará de manera más completa el caso del manipulador paralelo, del tipo plataforma de Stewart, junto con las distintas ecuaciones que corresponden a su modelamiento cinemático y dinámico.

En el capítulo 3 se presenta el desarrollo de la construcción (mediante software) de cada una de las partes que componen el manipulador, además de realizar los cálculos de la cinemática para una posición inicial (denominada “home”), la cual se implementa en la interfaz gráfica. Finalmente, en el capítulo 4, se mostrarán distintas respuestas de la plataforma, ante distintas variaciones de los ángulos de los actuadores rotativos, para así analizar los errores de movimiento de la plataforma superior comparándolos con los valores obtenidos a través de la interfaz gráfica.

Capítulo 2: La Plataforma de Stewart

Introducción

En el presente capítulo se presenta la plataforma de Stewart, su definición, configuraciones, aplicaciones, etc. Además, se presentan las ecuaciones que dan cuenta de su construcción y de su cinemática. Se define, generalmente, la plataforma de Stewart como un manipulador robótico paralelo con 6 GDL controlado por cualquier combinación de 6 motores solidarios a la base.



Fig. 16. Plataforma de Stewart.

Descripción de la plataforma

Como se ha mencionado anteriormente la plataforma de Stewart es un manipulador robótico paralelo el cual consta de una placa base fija y otra superior móvil. Cuyas piernas varían su geometría dependiendo de la configuración de estas, siendo el esquema de la idea general una plataforma 3 – 3, base y plataforma superior con 3 puntos de conexión para 3 piernas.

En la figura 17 se pueden observar 5 configuraciones de la plataforma de Stewart, las que varían en la geometría de su placa base y plataforma superior, además de variar en sus juntas y disposición de los actuadores.

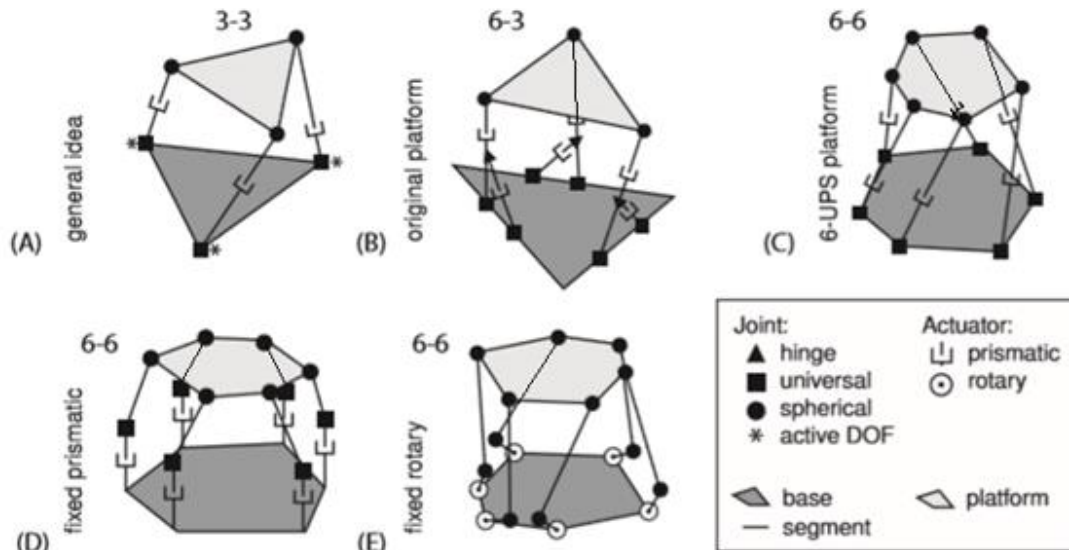


Fig. 17. Descripciones de distintas plataformas de Stewart.

De la figura anterior se puede observar:

- Idea original de Stewart.
- La primera realización de la plataforma, con 6 actuadores hidráulicos.
- La realización más típica de una plataforma con 6 grados de libertad.
- Ejemplo de plataforma con actuadores prismáticos.
- Ejemplo de plataforma con actuadores rotativos.

El control de la orientación y posición de la plataforma superior móvil viene dado por la variación de la longitud de las piernas que conectan ambas partes (base y plataforma superior). Dichas longitudes varían gracias al movimiento controlado de los actuadores empotrados en la base.

Para la realización de este proyecto se tomará como base el modelo de la figura siguiente:

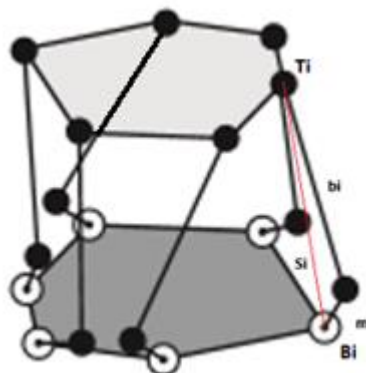


Fig. 18. Descripción de la plataforma de Stewart

Se puede observar que las “piernas” que vinculan las plataformas (S_i) se componen de mecanismos de biela – manivela espaciales, compuestos a su vez de 2 barras rígidas, m_i y b_i (manivela y biela respectivamente). Las manivelas y la placa base constituyen 6 pares de revolución, mientras que las bielas se vinculan con las manivelas y placa superior conformando 12 pares esféricos.

Al constituir de esta forma las piernas, se reemplazan los actuadores lineales usados en plataformas Stewart convencionales (Fig. 16 y Fig. 17.C).

Descripción de la necesidad

A pesar de que existe mucha documentación sobre este tipo de plataforma y su cinemática, en general se basan en el modelo original (un modelo 6 – 3 como se ve en la figura 17). El modelo 3 – 3 resulta ser demasiado teórico al no existir manera de conectar 2 piernas independientes a un mismo punto, un vértice del triángulo equilátero. Al analizar el siguiente, usando como base los modelos existentes para las plataformas 6 – 3, se busca generar un modelo funcional y práctico modificando la plataforma de una 6 – 3 a una 6 – 6 (6 actuadores, 6 vértices superiores).

1. Cinemática de la plataforma

Para hablar de la cinemática de la plataforma nos basaremos en las publicaciones de Kai Liu, John M. Fitzgerald y Frank L. Lewis [16].

Existen dos formas de resolver el problema cinemático de un manipulador [17]:

- **Problema cinemático directo:** Determinar la posición y orientación del extremo del manipulador, con respecto a un sistema de coordenadas de referencia, conocidos los valores de los giros de las articulaciones y los parámetros geométricos de los elementos del manipulador.
- **Problema cinemático inverso:** Determinar la configuración que deben adoptar las articulaciones del robot para alcanzar una posición y orientación del extremo dadas.

Para resolver el problema cinemático es necesario formular las ecuaciones de posición de la plataforma superior, con respecto a un sistema de ejes cartesiano fijo en la placa base (se analizará en la siguiente sección). Para esto es necesario recurrir al vector traslación y la matriz de rotación homogénea [18].

El vector traslación es sencillo de ver, ya que solo existen desplazamientos en los 3 ejes (X, Y, Z), por lo que el vector se verá compuesto por los desplazamientos que sufra cualquier punto de la plataforma superior (el más simple de ver es su centro). El vector traslación se puede escribir como:

$$T = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (2)$$

Además, la plataforma puede sufrir rotaciones en sus 3 ejes, por lo que se aplica la matriz de rotación completa, la cual se obtiene al multiplicar las matrices de rotación básicas entre sí, para representar una secuencia de rotación finita respecto del eje principal de sistema.

Para este proyecto se tomó en cuenta la orientación RPY (Roll, Pitch, Yaw), el cual consta de los siguientes pasos:

1. Girar el sistema en un ángulo ψ (yaw) con respecto al eje X .
2. Girar el sistema en un ángulo θ (pitch) con respecto al eje Y .
3. Girar el sistema en un ángulo ϕ (roll) con respecto al eje Z .

Como se mencionó anteriormente, la matriz de rotación completa se puede calcular multiplicando las matrices de rotación básicas, según los pasos anteriores. La multiplicación se realiza del paso 3 al paso 1, de la siguiente forma:

$${}^P R_B = R(\psi) \times R(\theta) \times R(\phi) \quad (3)$$

Lo que, considerando $C = \cos()$ y $S = \sin()$ se expresa como:

$${}^P R_B = \begin{bmatrix} C\phi & -S\phi & 0 \\ S\phi & C\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\psi & -S\psi \\ 0 & S\psi & C\psi \end{bmatrix} \quad (3.1)$$

Obteniendo:

$${}^P R_B = \begin{bmatrix} C\phi C\theta & C\phi S\theta S\psi - S\phi C\psi & C\phi S\theta C\psi + S\phi S\psi \\ S\phi C\theta & S\phi S\theta S\psi - S\phi C\psi & S\phi S\theta C\psi - S\phi S\psi \\ -S\theta & C\theta S\psi & C\theta C\psi \end{bmatrix} \quad (3.2)$$

Utilizando el vector de traslación T y la matriz de rotación ${}^P R_B$, se pueden escribir las coordenadas de la plataforma superior con respecto a la placa base, como veremos más adelante.

2. Ecuaciones de posición

Para obtener los puntos de conexión en la placa base B_i , debemos observar la figura siguiente:

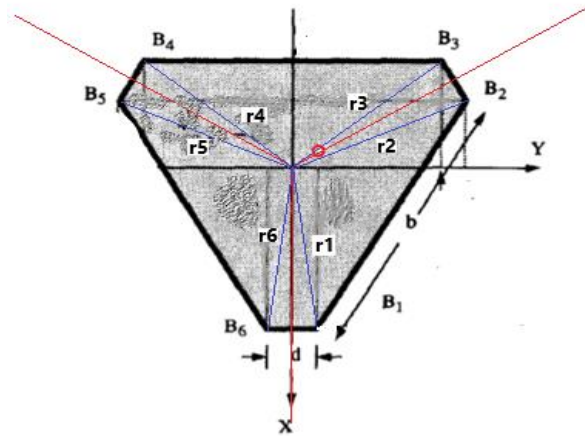


Fig. 19. Vista superior de la placa base.

Donde se puede observar que corresponde a un triángulo equilátero truncado en los vértices, con centro en O y un sistema de ejes cartesianos $[X, Y, Z]$ ubicado en la plataforma base. Además, se pueden observar los vectores r_i , los cuales tienen una longitud:

$$r_i = \frac{1}{2} \sqrt{b^2 + d^2} \quad (4)$$

La cual es idéntica para cada uno de los vectores r_i . Además, se observa que en el extremo de cada uno de los vectores se indica la posición fija de la conexión de cada una de las 6 piernas del sistema. Cada vector se escribe de la forma:

$$r_i = \begin{bmatrix} X_{B_i} \\ Y_{B_i} \\ Z_{B_i} \end{bmatrix}, \quad i = 1, \dots, 6 \quad (4.1)$$

Y, mediante un análisis geométrico, se pueden obtener las coordenadas de cada punto B_i , que se utilizarán para la construcción, de la placa base:

$$B_1 = \begin{bmatrix} X_{B_1} \\ Y_{B_1} \\ Z_{B_1} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{6} (2b + d) \\ \frac{1}{2} d \\ 0 \end{bmatrix} \quad (5.1)$$

$$B_2 = \begin{bmatrix} X_{B_2} \\ Y_{B_2} \\ Z_{B_2} \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{3}}{6}(b-d) \\ \frac{1}{2}(b+d) \\ 0 \end{bmatrix} \quad (5.2)$$

$$B_3 = \begin{bmatrix} X_{B_3} \\ Y_{B_3} \\ Z_{B_3} \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{3}}{6}(b+2d) \\ \frac{1}{2}b \\ 0 \end{bmatrix} \quad (5.3)$$

$$B_4 = \begin{bmatrix} X_{B_4} \\ Y_{B_4} \\ Z_{B_4} \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{3}}{6}(b+2d) \\ -\frac{1}{2}b \\ 0 \end{bmatrix} \quad (5.4)$$

$$B_5 = \begin{bmatrix} X_{B_5} \\ Y_{B_5} \\ Z_{B_5} \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{3}}{6}(b-d) \\ -\frac{1}{2}(b+d) \\ 0 \end{bmatrix} \quad (5.5)$$

$$B_6 = \begin{bmatrix} X_{B_6} \\ Y_{B_6} \\ Z_{B_6} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{6}(2b+d) \\ -\frac{1}{2}d \\ 0 \end{bmatrix} \quad (5.6)$$

Donde b y d son las longitudes de los lados de hexágono semirregular (Fig. 19). Como estas coordenadas fijas se expresan según el centro de la plataforma base O , la componente Z de cada punto B_i es nula.

Para el caso de la plataforma superior, al observar la figura 17A la forma de la plataforma superior es un triángulo equilátero. En la práctica, para poder realizar la conexión de las piernas con esta plataforma, se debe trincar sus vértices dándole la forma de hexágono, que se muestra en la figura 20, y que corresponde a un espejo de la placa base, con centro en O' y eje cartesiano $[x, y, z]$.

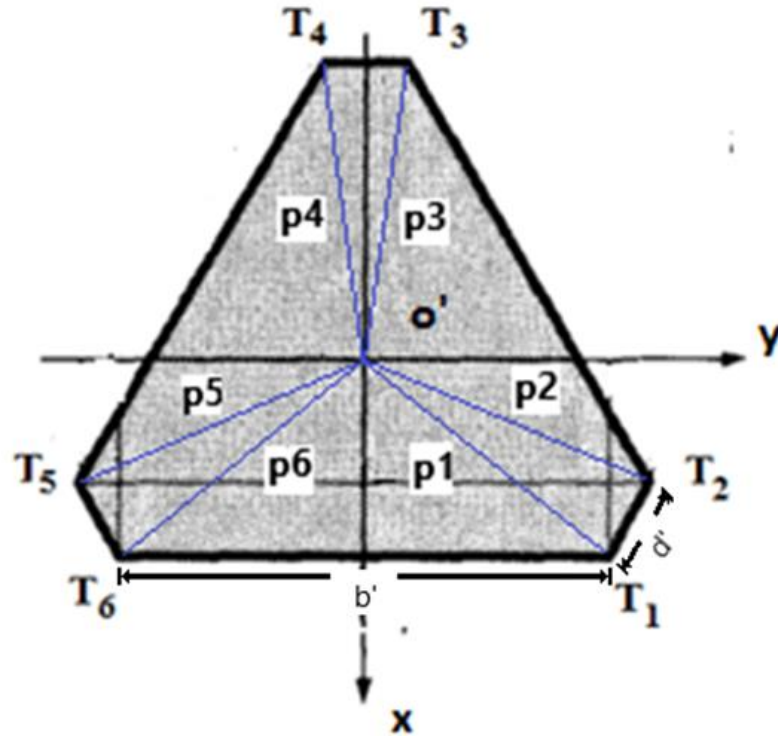


Fig. 20. Vista superior de la plataforma superior.

Al igual que en la placa base, la plataforma superior tiene vectores p_i dispuestos como se observa en la figura 20, los cuales tienen una longitud:

$$p_i = \frac{1}{2} \sqrt{b'^2 + d'^2} \quad (6)$$

Cuyos extremos coinciden en cada punto T_i de la plataforma superior, donde se conectan las piernas S_i con los puntos B_i de la placa base. Cada vector se compone de la siguiente manera:

$$p_i = \begin{bmatrix} x_{T_i} \\ y_{T_i} \\ z_{T_i} \end{bmatrix}, \quad i = 1, \dots, 6 \quad (6.1)$$

Al igual que con la placa base, mediante un análisis geométrico, se obtienen las coordenadas de cada punto T_i de la plataforma superior:

$$T_1 = \begin{bmatrix} x_{T_1} \\ y_{T_1} \\ z_{T_1} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{6} (b' + 2d') \\ \frac{1}{2} b' \\ 0 \end{bmatrix} \quad (7.1)$$

$$T_2 = \begin{bmatrix} x_{T_2} \\ y_{T_2} \\ z_{T_2} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{6}(b' - d') \\ \frac{1}{2}(b' + d') \\ 0 \end{bmatrix} \quad (7.2)$$

$$T_3 = \begin{bmatrix} x_{T_3} \\ y_{T_3} \\ z_{T_3} \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{3}}{6}(2b' + d') \\ \frac{1}{2}d' \\ 0 \end{bmatrix} \quad (7.3)$$

$$T_4 = \begin{bmatrix} x_{T_4} \\ y_{T_4} \\ z_{T_4} \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{3}}{6}(2b' + d') \\ -\frac{1}{2}d' \\ 0 \end{bmatrix} \quad (7.4)$$

$$T_5 = \begin{bmatrix} x_{T_5} \\ y_{T_5} \\ z_{T_5} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{6}(b' - d') \\ -\frac{1}{2}(b' + d') \\ 0 \end{bmatrix} \quad (7.5)$$

$$T_6 = \begin{bmatrix} x_{T_6} \\ y_{T_6} \\ z_{T_6} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{6}(b' + 2d') \\ -\frac{1}{2}b' \\ 0 \end{bmatrix} \quad (7.6)$$

Donde b' y d' son las longitudes de los lados de hexágono semirregular (Fig. 20). Como estas coordenadas fijas se obtienen según el centro de la plataforma superior O' , la componente z de cada punto T_i será nula.

La siguiente figura muestra una plataforma de Stewart convencional, donde se pueden observar cada uno de los puntos que intervienen en los cálculos necesarios para el control de la orientación y posición de la plataforma superior.

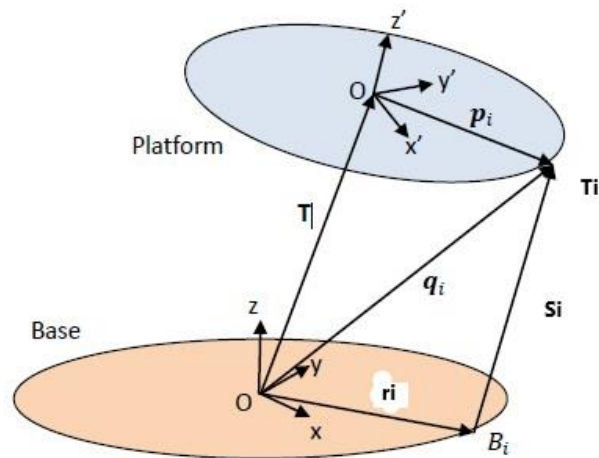


Fig. 21. Plataforma de Stewart convencional.

Teniendo en cuenta todo lo anterior, se puede calcular el vector q_i que representa la ubicación del punto T_i de la plataforma superior, con respecto a la placa base. Existen 2 sumas de vectores que conducen a la solución de q_i las cuales son:

$$q_i = r_i + S_i \quad (8)$$

$$q_i = T + p_i \quad (9)$$

Es necesario recordar que se debe aplicar la matriz de rotación ${}^P R_B$ en el vector p_i para poder obtener la posición después de las rotaciones de la plataforma superior, con respecto a la placa base. Reescribiendo las ecuaciones:

$$q_i = r_i + S_i \quad (8.1)$$

$$q_i = T + {}^P R_B \times p_i \quad (9.1)$$

Se observa que con la ecuación (9.1) se obtiene la posición del punto T_i de la plataforma superior con respecto a la placa base. Luego, igualando ambas ecuaciones se pueden obtener los vectores de cada pierna para obtener la orientación y posición deseada de la plataforma superior. Con lo anterior se obtiene la siguiente ecuación:

$$S_i = T + {}^P R_B \times p_i - r_i \quad (10)$$

$$i = 1, \dots, 6$$

Para obtener el largo de cada pierna se debe calcular el módulo del vector de la ecuación (10).

En la figura siguiente se presenta el modelo de la plataforma utilizada para calcular la cinemática:

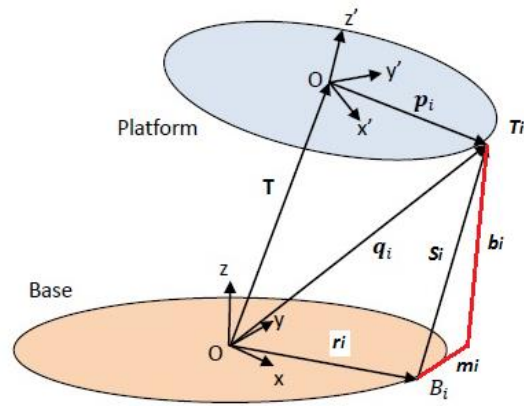


Fig. 22. Plataforma de Stewart.

Como se puede observar, en la figura 22, que el largo de la pierna S_i viene relacionado directamente con los vectores m_i y b_i (manivela y biela respectivamente). Donde además el punto B_i corresponde al extremo del eje de cada actuador del sistema.

3. Mecanismo de movimiento

Para el mecanismo de movimiento se considera un sistema biela – manivela, accionado por servo actuadores, como el mostrado en la figura siguiente:

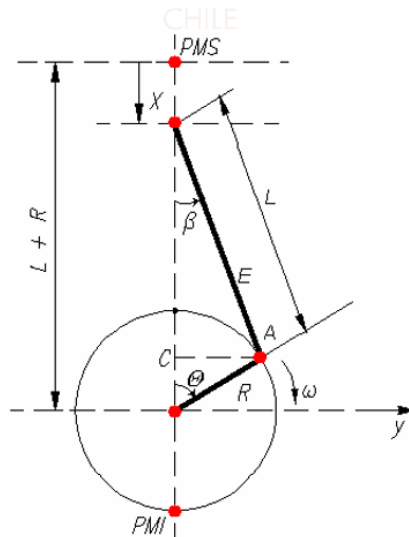


Fig. 23. Mecanismo biela – manivela.

Como se puede observar, en la figura 23, este modelo del mecanismo no considera que las juntas de los puntos A y X son juntas esféricas, por lo que este modelo no se puede utilizar. Razón por la cual se utilizará el modelo que se representa en la siguiente figura [19]:

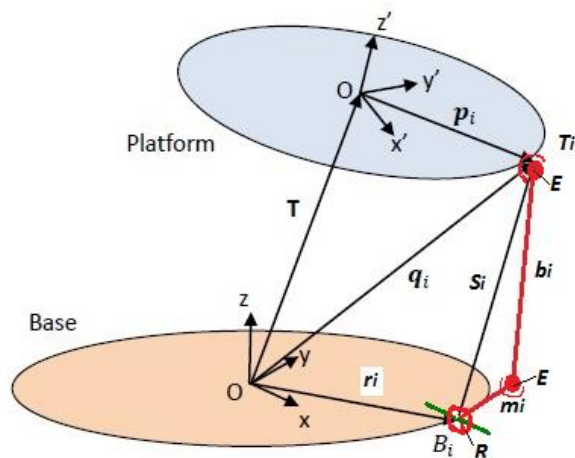


Fig. 24. Plataforma de Stewart, sistema biela – manivela.

En la figura anterior se puede observar que:

- En el vértice del vector r_i se encuentra el eje del servo actuador, el cual es una articulación rotativa (1 grado de libertad) representado en la imagen con la letra **R**.
- En el vértice del vector m_i , correspondiente al punto de conexión de la manivela con la biela, se encuentra una articulación esférica (3 grados de libertad) representado en la imagen con la letra **E**.
- En el vértice de vector b_i , correspondiente al punto de conexión de la biela con la plataforma superior, se encuentra una segunda articulación esférica también representada por la letra **E**.

Al utilizar el modelo presentado en la figura 24, es necesario calcular el ángulo de giro de cada servo actuador para obtener la variación de S_i deseada. Detallando el diagrama vectorial como muestra la figura 25, se observa el siguiente mecanismo a analizar.

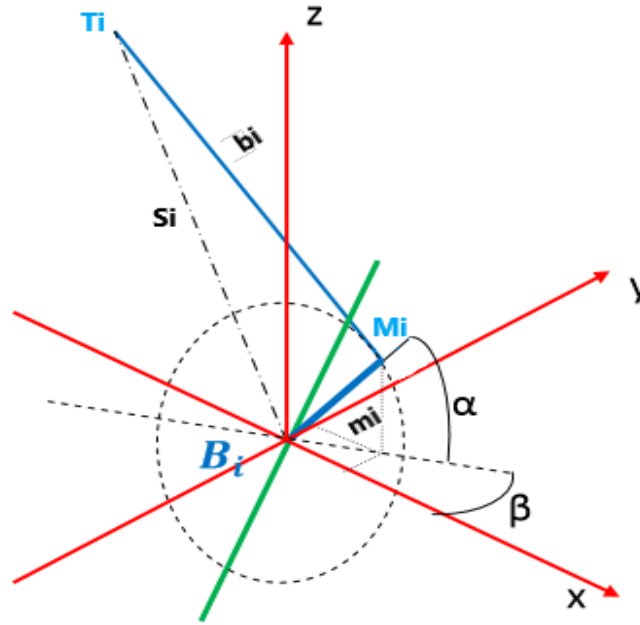


Fig. 25. Mecanismo biela – manivela en 3 dimensiones.

$M_i \rightarrow$ Conexión manivela con biela

$$M_i = [X_M \ Y_M \ Z_M]^T \quad (11)$$

$B_i \rightarrow$ Eje de rotacion de los servomotores, con coordenadas

$$B_i = [X_B \ Y_B \ Z_B]^T \quad (12)$$

$T_i \rightarrow$ Punto de la junta de la pierna con la plataforma, con coordenadas

$$T_i = [X_T \ Y_T \ Z_T]^T \quad (13)$$

$b_i \rightarrow$ Longitud de la pierna operativa

$S_i \rightarrow$ Longitud de la i^{esima} pierna calculado con (10)

$m_i \rightarrow$ Longitud de la manivela operativa

$\alpha \rightarrow$ Angulo de la manivela con respecto a la horizontal

$\beta \rightarrow$ Angulo del plano de la manivela relativo al eje x

El arreglo de los servomotores indica que las manivelas pares e impares son una reflexión la una de la otra, como se muestra en la figura siguiente.

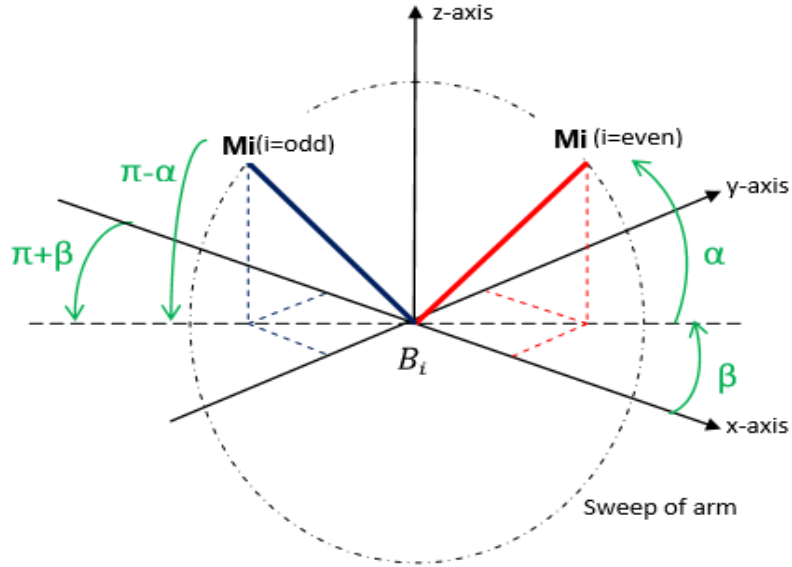


Fig. 26. Reflexión de manivelas pares (even) e impares (odd)

De la figura anterior se puede obtener que, para las piernas pares:

$$X_M = m_i \cos \alpha \cos \beta + X_B \quad (14)$$

$$Y_M = m_i \cos \alpha \sin \beta + Y_B \quad (15)$$

$$Z_M = m_i \sin \alpha + Z_B \quad (16)$$

Y para las piernas impares:

$$X_M = m_i \cos(\pi - \alpha) \cos(\pi + \beta) + X_B \quad (17)$$

$$Y_M = m_i \cos(\pi - \alpha) \sin(\pi + \beta) + Y_B \quad (18)$$

$$Z_M = m_i \sin(\pi - \alpha) + Z_B \quad (19)$$

Pero:

$$\sin(\pi - \alpha) = \sin \alpha, \cos(\pi - \alpha) = -\cos \alpha \quad (20)$$

$$\sin(\pi + \beta) = -\sin \beta, \cos(\pi + \beta) = -\cos \beta \quad (21)$$

Sustituyendo (20) y (21) en las ecuaciones (17), (18) y (19), se obtiene que éstas son iguales a las ecuaciones (14), (15) y (16), que son las ecuaciones para las piernas pares.

A continuación, se tiene por el teorema de Pitágoras:

$$m_i^2 = (X_M - X_B)^2 + (Y_M - Y_B)^2 + (Z_M - Z_B)^2$$

$$m_i^2 = (X_M^2 + Y_M^2 + Z_M^2) + (X_B^2 + Y_B^2 + Z_B^2) - 2(X_M X_B + Y_M Y_B + Z_M Z_B) \quad (22)$$

$$S_i^2 = (X_T - X_B)^2 + (Y_T - Y_B)^2 + (Z_T - Z_B)^2$$

$$S_i^2 = (X_T^2 + Y_T^2 + Z_T^2) + (X_B^2 + Y_B^2 + Z_B^2) - 2(X_T X_B + Y_T Y_B + Z_T Z_B) \quad (23)$$

$$b_i^2 = (X_T - X_M)^2 + (Y_T - Y_M)^2 + (Z_T - Z_M)^2$$

$$b_i^2 = (X_T^2 + Y_T^2 + Z_T^2) + (X_M^2 + Y_M^2 + Z_M^2) - 2(X_T X_M + Y_T Y_M + Z_T Z_M) \quad (24)$$

Sustituyendo (22) y (23) en (24), se obtiene:

$$b_i^2 = S_i^2 - (X_B^2 + Y_B^2 + Z_B^2) + 2(X_T X_B + Y_T Y_B + Z_T Z_B) + m_i^2 - (X_B^2 + Y_B^2 + Z_B^2) + 2(X_M X_B + Y_M Y_B + Z_M Z_B) - 2(X_T X_M + Y_T Y_M + Z_T Z_M)$$

Ordenando:

$$S_i^2 - (b_i^2 - m_i^2) = 2(X_B^2 + Y_B^2 + Z_B^2) 2X_M(X_T - X_B) + 2Y_M(Y_T - Y_B) + 2M(Z_T - Z_B) - 2(X_T X_B + Y_T Y_B + Z_T Z_B) \quad (24.1)$$

Por último, reemplazando (14), (15) y (16) en la ecuación (24.1):

$$\begin{aligned} S_i^2 - (b_i^2 - m_i^2) &= 2(X_B^2 + Y_B^2 + Z_B^2) \\ &+ 2(m_i \cos \alpha \cos \beta + X_B)(X_T - X_B) + 2(m_i \cos \alpha \sin \beta + Y_B)(Y_T - Y_B) \\ &+ 2(m_i \sin \alpha + Z_B)(Z_T - Z_B) - 2(X_T X_B + Y_T Y_B + Z_T Z_B) \quad (24.2) \end{aligned}$$

Lo que se reduce a:

$$S_i^2 - (b_i^2 - m_i^2) = 2m_i \sin \alpha (Z_T - Z_B) + 2m_i \cos \alpha [\cos \beta (X_T - X_B) + \sin \beta (Y_T - Y_B)] \quad (24.3)$$

8Ecuación que tiene la forma:

$$L = M \sin \alpha + N \cos \alpha \quad (25)$$

Utilizando la identidad trigonométrica:

$$a \sin x + b \cos x = c \sin(x + v) \quad (25.1)$$

$$c = \sqrt{a^2 + b^2} \quad y \quad v = \tan^{-1} \left(\frac{b}{a} \right) \quad (25.2)$$

Entonces se obtiene una nueva función seno de α con cambio de fase δ :

$$L = \sqrt{M^2 + N^2} \sin(\alpha + \delta) \quad (25.3)$$

$$\sin(\alpha + \delta) = \frac{L}{\sqrt{M^2 + N^2}} \quad (25.4)$$

$$\delta = \tan^{-1} \left(\frac{N}{M} \right) \quad (25.5)$$

Finalmente se obtiene que:

$$\alpha = \sin^{-1} \left(\frac{L}{\sqrt{M^2 + N^2}} \right) - \tan^{-1} \left(\frac{N}{M} \right) \quad (26)$$

Donde:

$$L = S_i^2 - (b_i^2 - m_i^2) \quad (26.1)$$

$$M = 2m_i(Z_T - Z_B) \quad (26.2)$$

$$N = 2m_i[\cos \beta (X_T - X_B) + \sin \beta (Y_T - Y_B)] \quad (26.3)$$

Por último, los anchos de pulso de la señal necesaria para que los servomotores varíen su ángulo de giro α según se requiera es:

$$W_i = W_i^0 + (\alpha_i - \alpha_0)r \quad \text{para } i = 2, 4, 6 \quad (27.1)$$

$$W_i = W_i^0 - (\alpha_i - \alpha_0)r \quad \text{para } i = 1, 3, 5 \quad (27.2)$$

Con:

$$r = \frac{400}{45} \times \frac{360}{2\pi} = \frac{1600 \mu\text{segundos}}{\pi \text{ radianes}} \quad (28)$$

Que corresponde al tiempo de cambio del ancho de pulso.

4. Secuencia de control

La secuencia de eventos para realizar el control del giro de los ejes de los servomotores es:

1. Ingresar la información posicional de la plataforma B_i, T_i, b, m, β_i . Todos son constantes de la construcción de la plataforma.
2. Ingresar las constantes de los servomotores, W_i^0 y r (Ancho de pulso para la $\alpha = 0^\circ$ y reloj del servomotor).
3. Calcular los valores de h_0 y α_0 .
4. Ingresar las variables para $(p_x, p_y, p_z, \psi, \phi, \theta)$ para la posición deseada de la plataforma.
5. Calcular la matriz de rotación ${}^P R_B$.
6. Calcular el largo de las piernas S_i .
7. Calcular los ángulos α_i requeridos para cada servomotor.
8. Calcular los anchos de pulso W_i requeridos para cada servomotor.
9. Enviar los valores de W_i al controlador de los servos.
10. Regresar al paso 4 y repetir el proceso.

Capítulo 3: Diseño mecánico de la plataforma e interfaz gráfica de control

Introducción

En el presente capítulo se realizarán los cálculos necesarios para construir el modelo 3D de la plataforma, además de mostrar la interfaz gráfica creada a través de lenguaje Python.

Diseño mecánico

Para la realización del modelo mecánico se aplicarán las ecuaciones presentadas en el capítulo 2, pero primero se utilizará la ecuación (1) para así demostrar que la plataforma si cumple con los 6 GDL. Utilizando los valores:

$$\lambda = 6 \quad (28.1)$$

$$n = 14 \quad (28.2)$$

$$j = 18 \quad (28.3)$$

$$f_i = 3 \quad (28.4)$$

$$I_f = 6 \quad (28.5)$$

Recordando que λ corresponde al espacio de trabajo tridimensional (con 6 GDL). n corresponde al número de eslabones fijos del mecanismo, donde se deben considerar los 6 pares biela – manivelas y la placa base y superior. j es el número de juntas que tiene el mecanismo (12 esféricas y 6 rotativas) f_i corresponde a los grados de libertad del movimiento relativos por junta (12 juntas con 3GDL y 6 con 1GDL) e I_f corresponde a los grados de libertad pasivos.

Reemplazando estos datos en la ecuación (1):

$$m = 6(14 - 18 - 1) + \sum_{i=1}^{12} 3 + \sum_{i=1}^6 1 - 6 = 6 \quad (29)$$

Se obtiene como resultado que el manipulador si tiene 6 GDL.

1. Base y plataforma superior

Para la construcción del modelo 3D de la base, es necesario definir las longitudes de los lados del hexágono semirregular. Para esto se utilizaron los valores:

$$b = 200 \text{ mm}; d = 80 \text{ mm} \quad (30)$$

Y utilizando las coordenadas mostradas en el capítulo 2, se obtienen los puntos B_i correspondientes al eje de los servoactuadores.

$$B_1 = (138,56; 40; 0) \quad (31.1)$$

$$B_2 = (-34,64; 140; 0) \quad (31.2)$$

$$B_3 = (-103,92; 100; 0) \quad (31.3)$$

$$B_4 = (-103,92; -100; 0) \quad (31.4)$$

$$B_5 = (-34,64; -140; 0) \quad (31.5)$$

$$B_6 = (138,56; -40; 0) \quad (31.6)$$

Utilizando los valores indicados anteriormente, se construyó el modelo 3D que se observa a continuación:

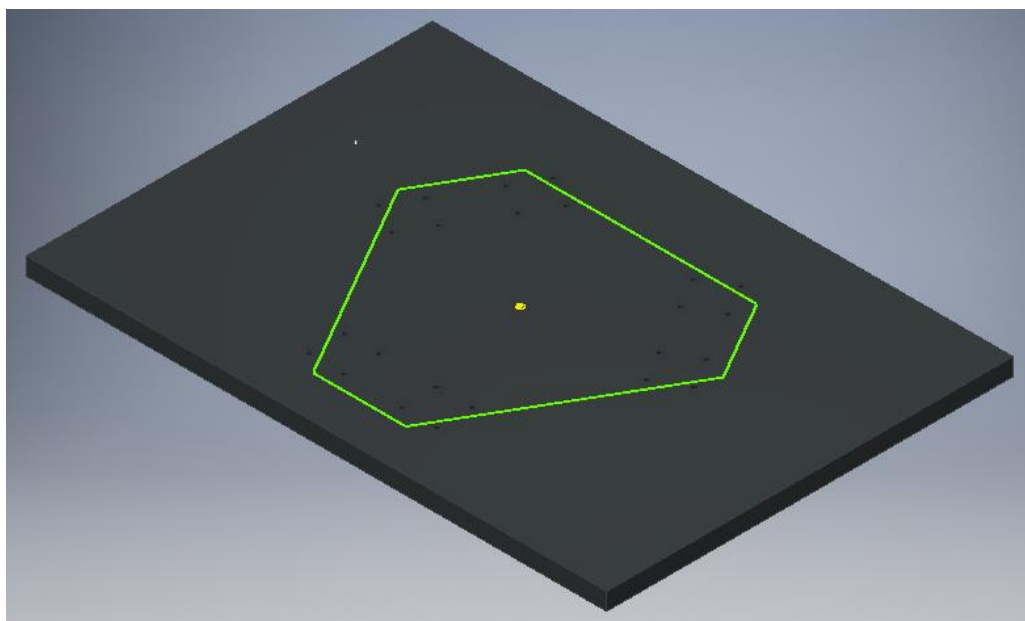


Fig. 27. Plataforma base (vista isométrica).

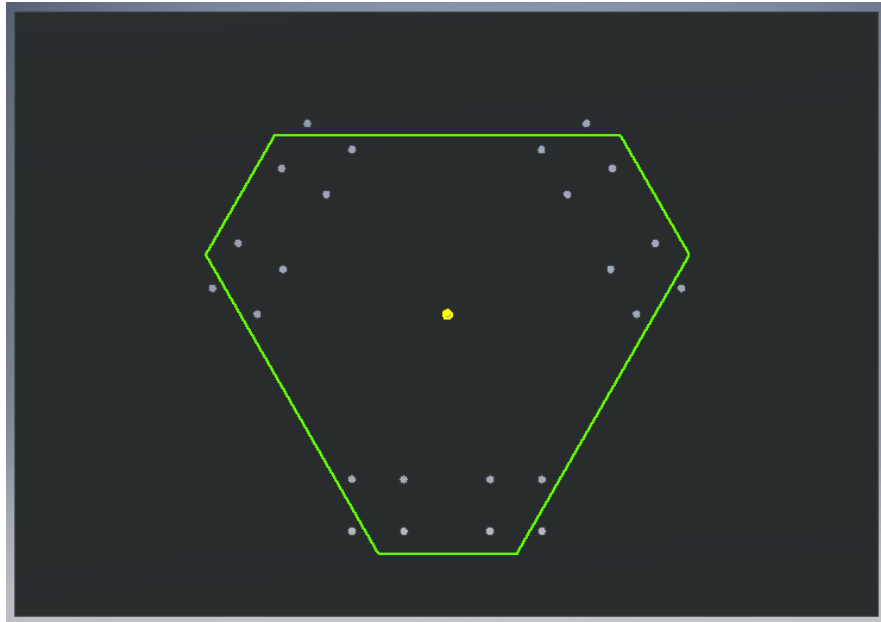


Fig. 28. Plataforma base (vista superior).

Se puede observar, en la figura anterior, que la plataforma base no necesariamente tiene que tener una forma estrictamente de hexágono semirregular, sino que, el hexágono se puede contener dentro de una placa aún más grande. Además, se observa que los puntos B_i que corresponden al eje de los servoactuadores se encuentran en el centro de los agujeros hechos para la ubicación de estos actuadores.

Por otro lado, para la plataforma superior se obtuvieron los siguientes puntos T_i , utilizando los valores:

$$b' = 300 \text{ mm} ; d' = 100 \text{ mm} \quad (32)$$

$$T_1 = (144,34; 150; 0) \quad (33.1)$$

$$T_2 = (57,73; 200; 0) \quad (33.2)$$

$$T_3 = (-202,07; 50; 0) \quad (33.3)$$

$$T_4 = (-202,07; -50; 0) \quad (33.4)$$

$$T_5 = (57,73; -200; 0) \quad (33.5)$$

$$T_6 = (144,34; -150; 0) \quad (33.6)$$

Con lo cual se construyó el siguiente modelo 3D, de la plataforma superior.

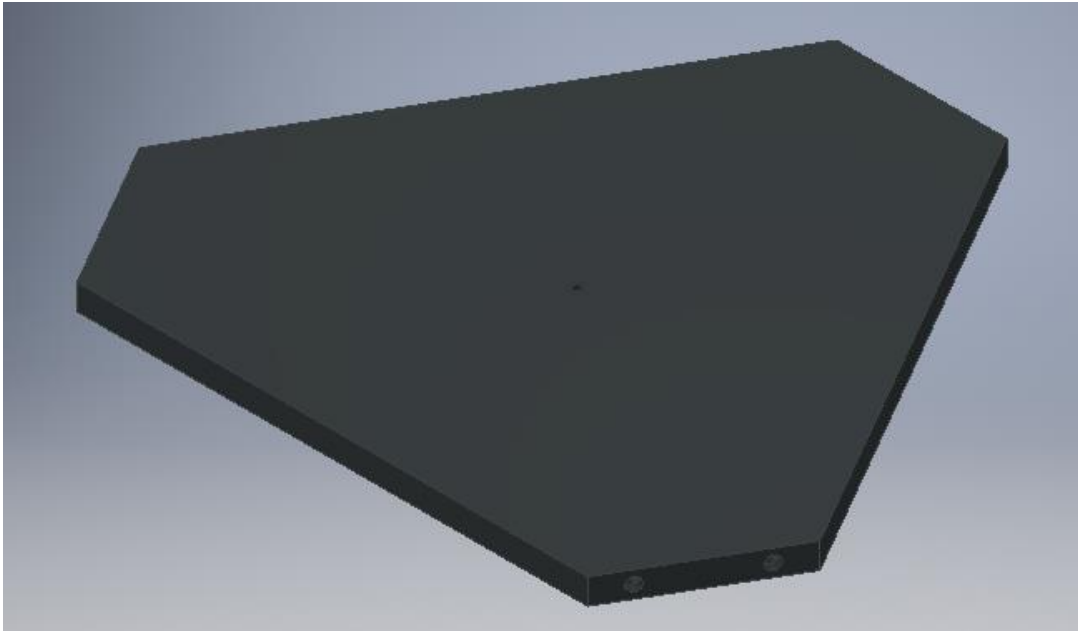


Fig. 29. Plataforma superior (vista isométrica).

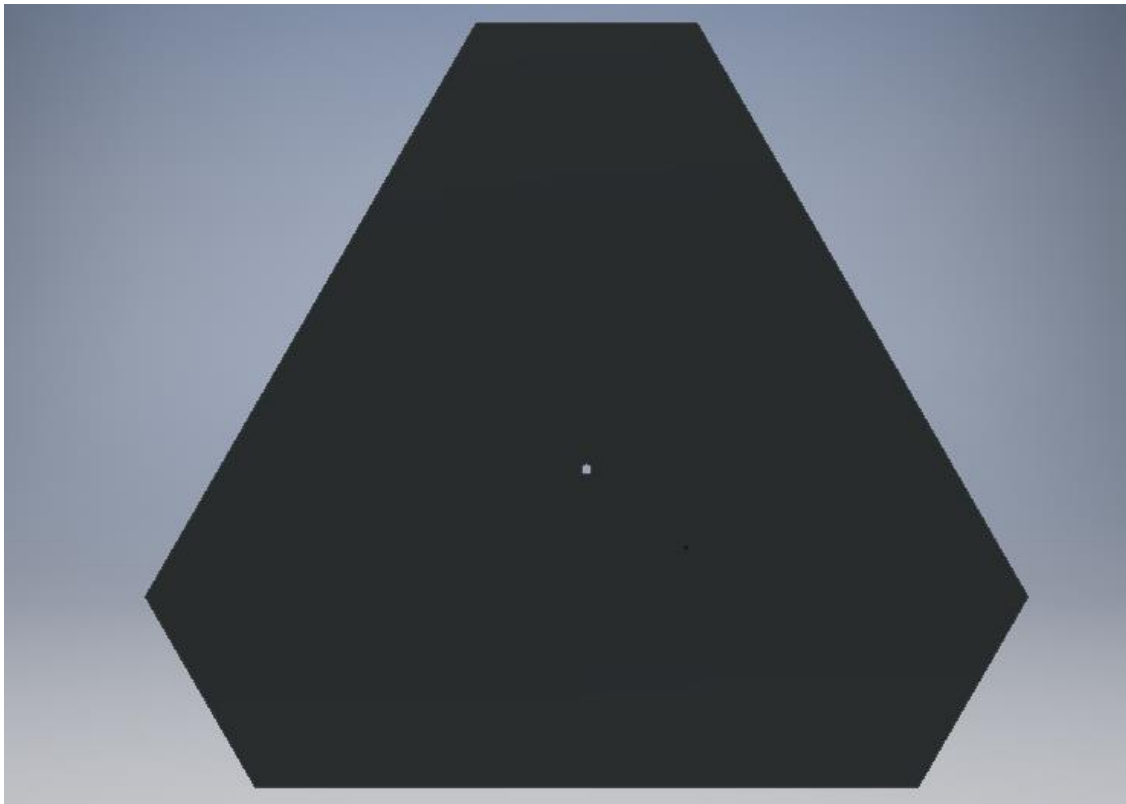


Fig. 30. Plataforma superior (vista superior).

2. Mecanismo de movimiento

Para realizar el movimiento de las piernas se utilizará un mecanismo biela – manivela accionado por servomotores. La ventaja de utilizar servomotores radica en que, al seleccionar un servomotor con reductor que mantenga una relación de 1:8 o mayor, permite controlar la dinámica de la plataforma sólo con los movimientos de los servomotores.

Para realizar la fijación de los actuadores a la base, se creó un modelo 3D de una base adecuada a las dimensiones del modelo 3D del servomotor.

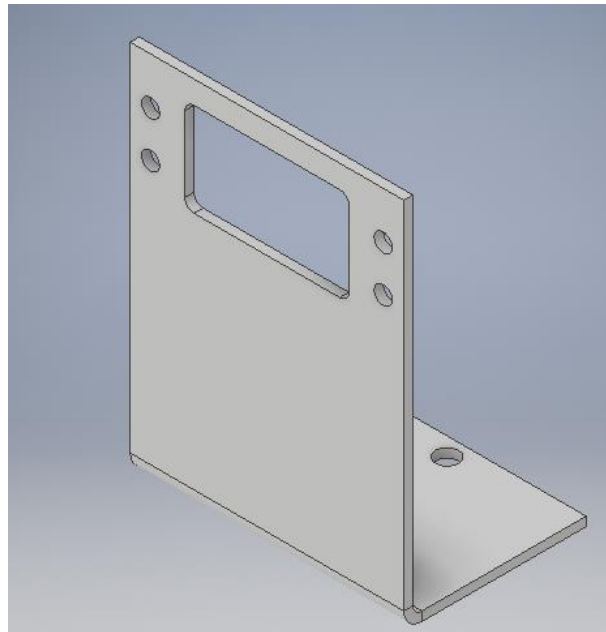


Fig. 31. Modelo base para servomotor.

El modelo del servomotor se muestra a continuación:

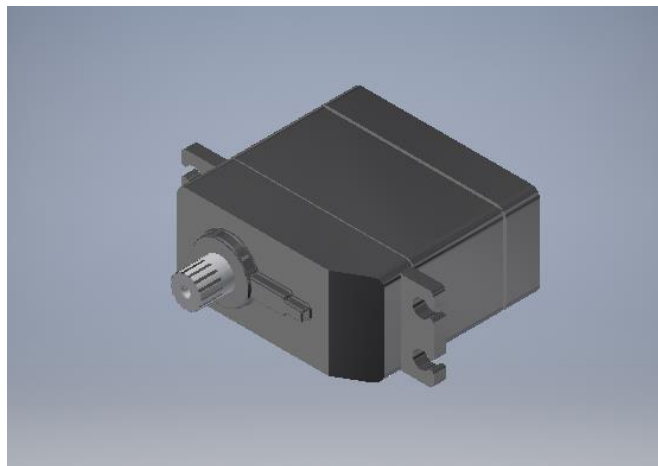


Fig. 32. Modelo servomotor [20].

Por otro lado, los eslabones que componen el mecanismo se crearon con los siguientes valores:

$$m_i = 45 \text{ mm}; b_i = 200 \text{ mm} \quad (34)$$

Con lo cual se crearon los siguientes modelos:



Fig. 33. Modelo de la biela.

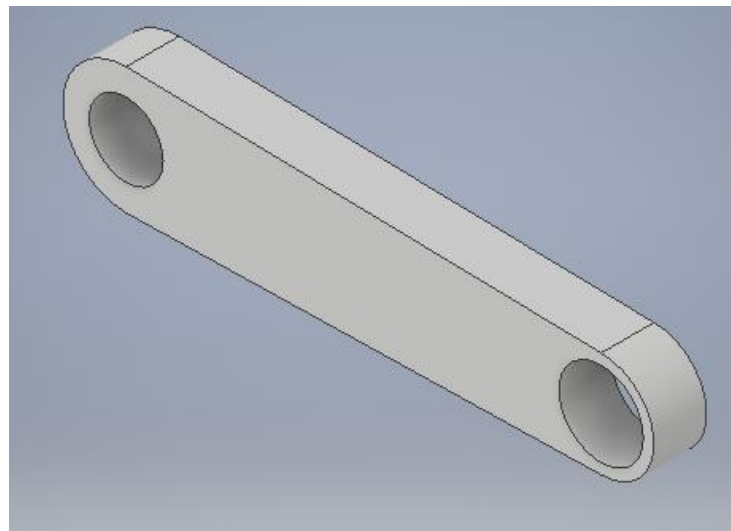
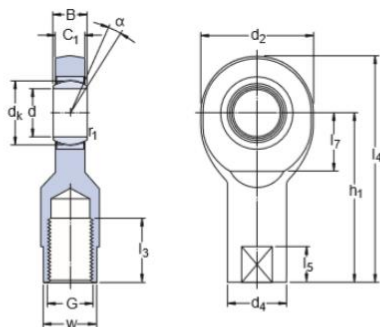


Fig. 34. Modelo de la manivela.

Cabe destacar que, la biela presentada en la figura 33 no cumple con el valor $b_i = 200mm$ ya que este valor se mide entre los ejes de las rotulas que se conectan en las puntas de la misma.

Para las uniones esféricas, se crearon modelos en base a los mostrados en el catálogo RÓTULAS Y CABEZAS DE ARTICULACIÓN SKF [21]. La junta se debe escoger según la carga que deba soportar la plataforma, como en este proyecto se centró en el diseño sin una carga fija se seleccionó una junta sin un criterio específico. Con lo cual se obtuvo el siguiente modelo 3D.



Dimensiones principales				Ángulo de inclinación	Capacidad de carga	Masa	Designaciones												
d	d ₂ máx	G 6H	B				C ₁ máx	h ₁	α	Capacida dinám. C	estát. C ₀	Cabeza articulación rosca a derechas	rosca a izquierdas	dk	d ₂	l ₃ mín	l ₄ máx	l ₅ ≈	l ₇ mín
mm						grados	N		kg	–									
6	22	M 6	6	4,5	30	13	3 600	8 150	0,023	SI 6 C	SIL 6 C	10	11	11	43	8	10	0,3	9
8	25	M 8	8	6,5	36	15	5 850	12 900	0,036	SI 8 C	SIL 8 C	13	13	15	50	9	11	0,3	11

Fig. 35. Datos de construcción de la rótula.

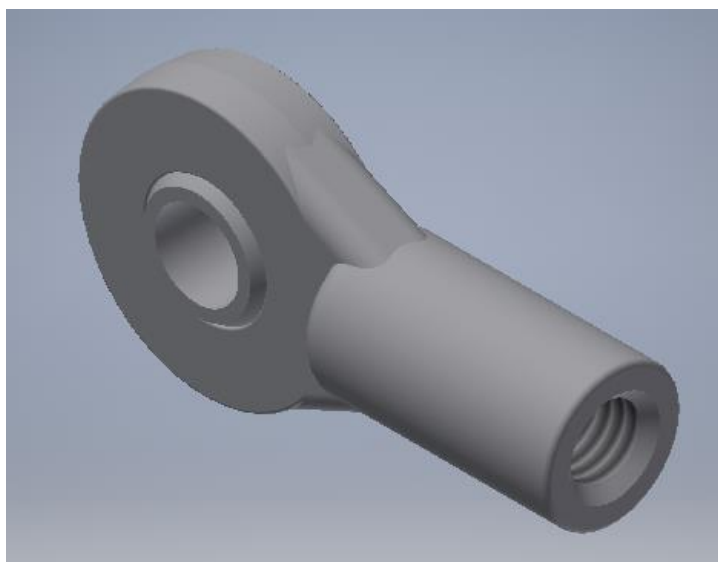


Fig. 36. Modelo de la rótula.

3. Ensamblaje de la plataforma

Una vez presentados los elementos que componen el modelo 3D de la plataforma, es necesario realizar el ensamblaje de cada uno de los mismos. Comenzando con el ensamblaje de la base del actuador – actuador – manivela, se obtiene el siguiente resultado:

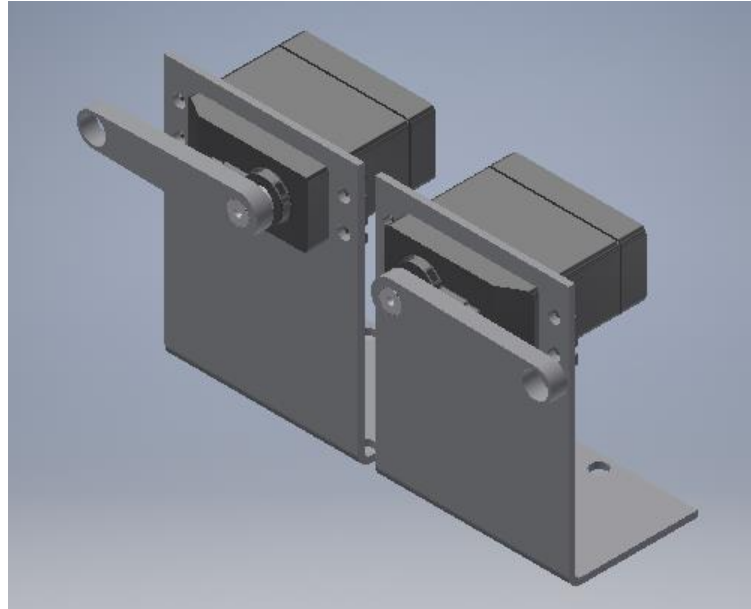


Fig. 37. Ensamblaje base actuador – actuador – manivela

Se puede observar, en la figura anterior, que las manivelas se opuestas una de la otra. Esto se explicó en el capítulo 2, cuando se mencionó que para los puntos B_i impares, es decir $i = 1,3,5$ la posición de las manivelas de los puntos B_i pares, $i = 2,4,6$ serán espejos unos de los otros.

Al conectar la biela, mostrada en la figura 33, con las rótulas se obtiene la pierna de largo $b_i = 200 \text{ mm}$, como se observa a continuación:

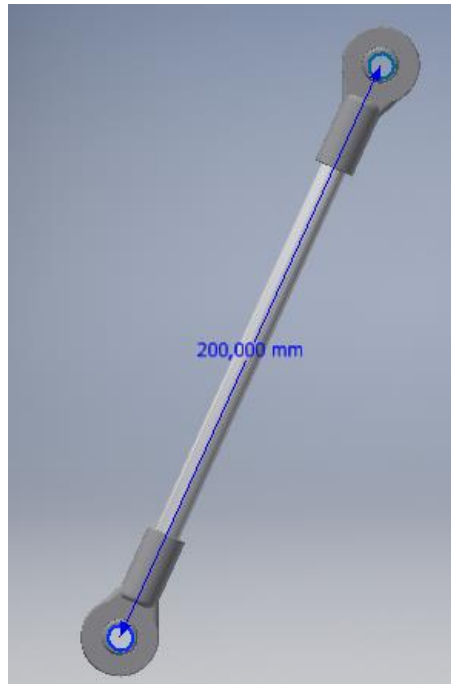


Fig. 38. Ensamblaje rótula – biela – rótula.

Utilizando estos elementos y ensamblándolos en la base de la plataforma obtenemos lo siguiente:

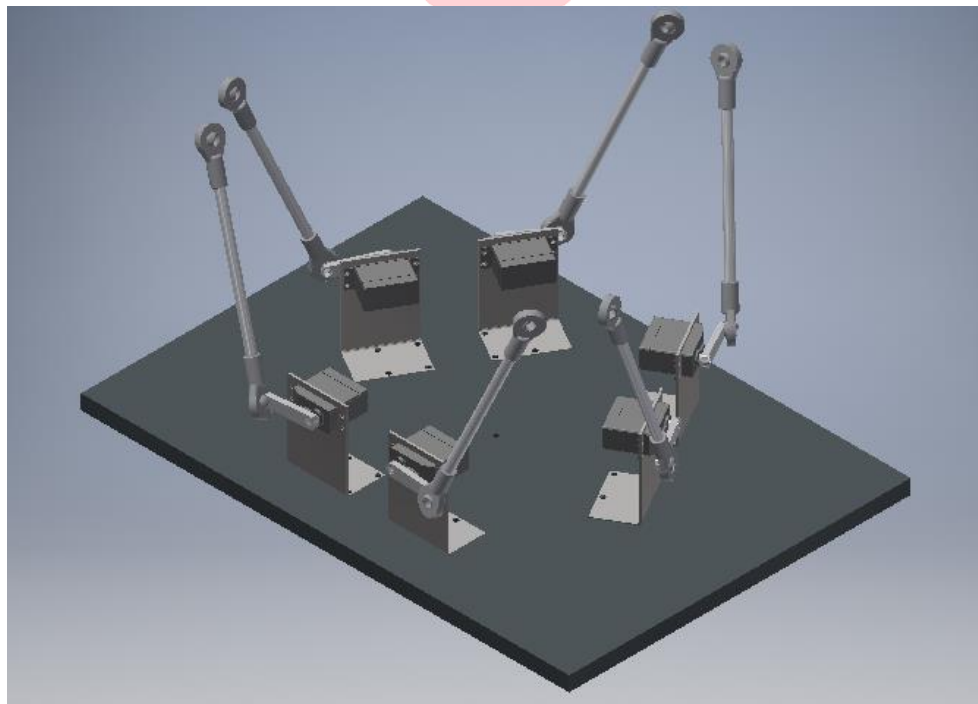


Fig. 39. Ensamblaje base – actuador – biela – manivela.

A lo que solo le hace falta ensamblar la plataforma superior, mostrada en la figura 29. El modelo completo de la plataforma se muestra a continuación:

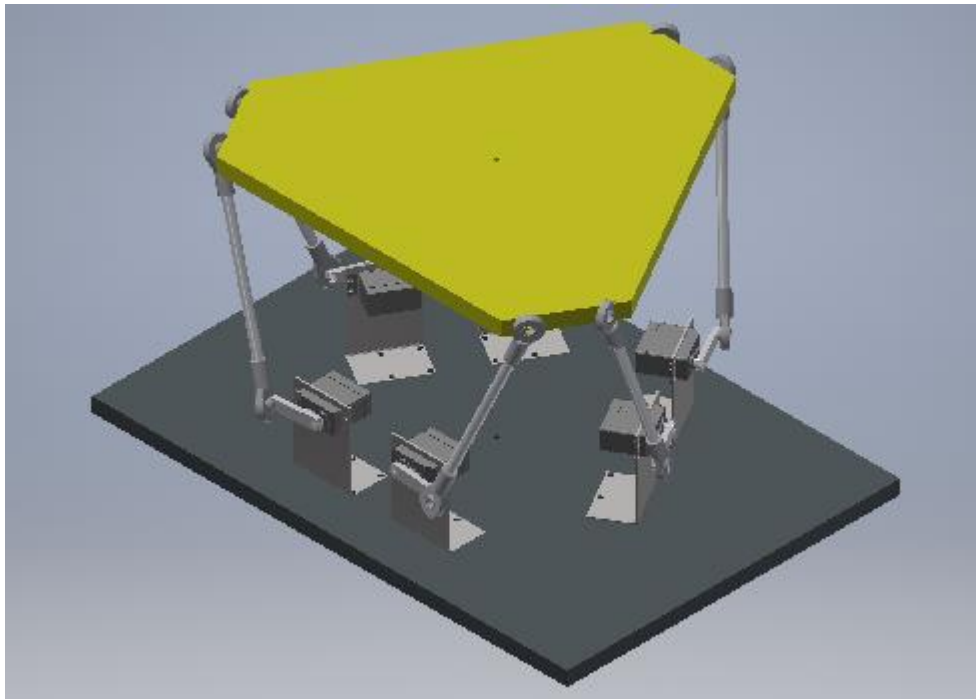


Fig. 40. Ensamblaje plataforma de Stewart.

De la figura anterior hay que mencionar que los ejes de los actuadores apuntan hacia afuera debido a que la superficie de la plataforma superior es mayor que la de la plataforma base. En el caso contrario, es decir que la plataforma base tenga mayor superficie que la superior los ejes de los actuadores apuntarían hacia el centro de la plataforma base.

Además, como se mencionó anteriormente, los puntos B_i que componen la plataforma base también corresponden a los ejes de los actuadores. Lo que hace necesario comprobar que las longitudes b y d no hayan sufrido una variación considerable. Utilizando la función *medir* del software Autodesk Inventor 2018 se puede comprobar las medidas de estas.

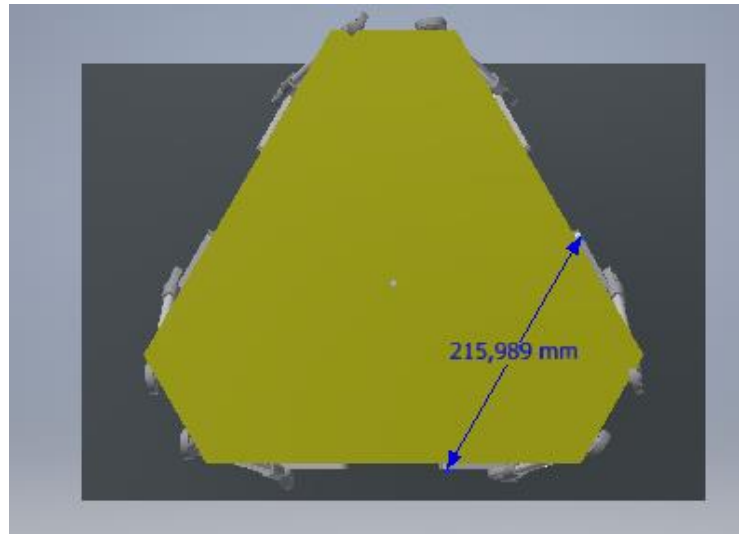


Fig. 41. Plataforma completa (nueva medida b).

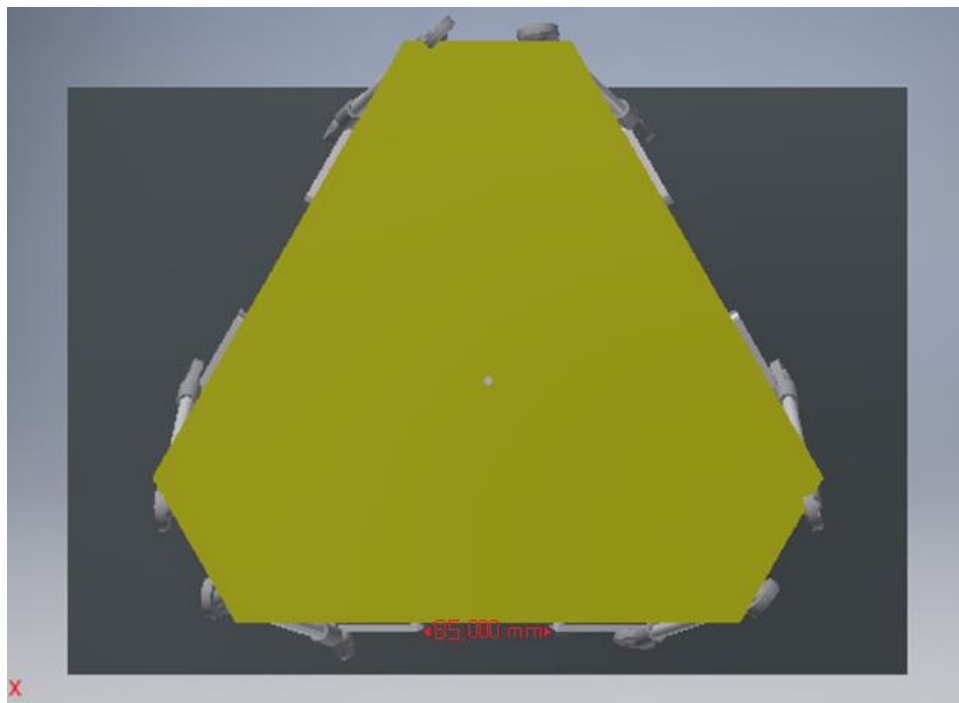


Fig. 42. Plataforma completa (nueva medida d).

De las figuras anteriores se observan que los nuevos valores son:

$$b \approx 216 \text{ mm}; d = 85 \text{ mm (35)}$$

Estos valores son importantes ya que afectan directamente a la cinemática del manipulador, y por lo tanto, al control de este.

Otro dato relevante que se puede obtener es β_i valor que corresponde al ángulo del plano de la manivela, con respecto al eje x . Entonces, utilizando los nuevos valores de b y d se puede dibujar un nuevo hexágono con el cual se pueden obtener los valores de β_i , como se muestra en la figura siguiente:

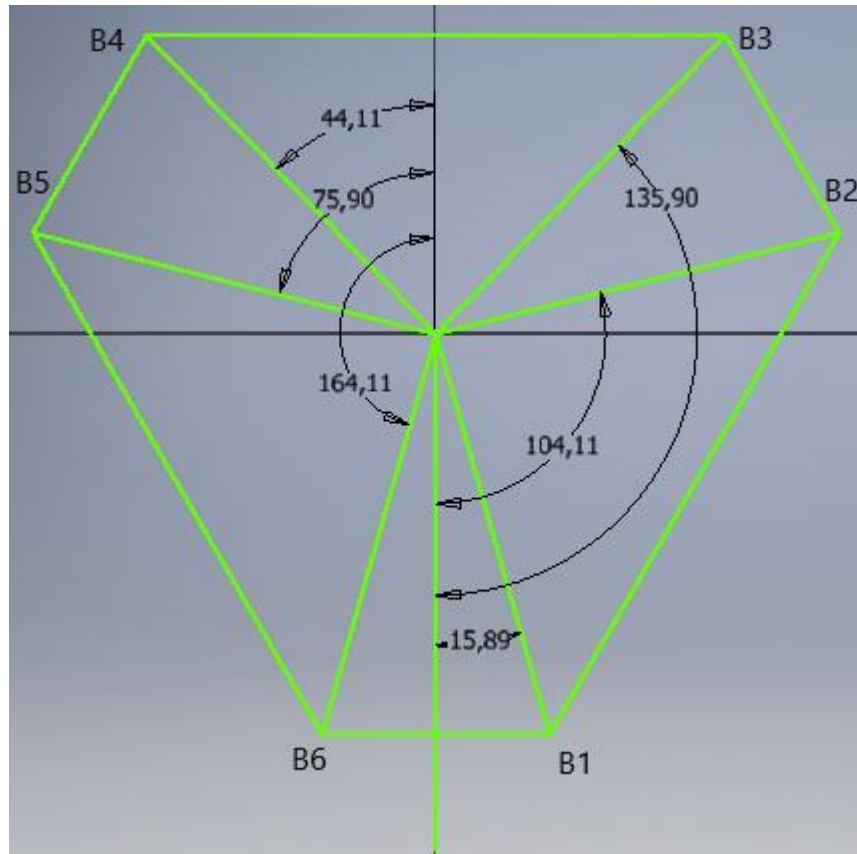


Fig. 43. Medición de ángulos β .

Como se observa en la figura anterior se trazan vectores desde el centro hasta el punto B_i y se mide el ángulo con respecto al eje x . Además, se puede ver que para $\beta_{1,2,3}$ los valores se muestran completos, mientras que para $\beta_{4,5,6}$ es necesario sumar al valor mostrado en la figura el valor 180. A continuación, se muestran los valores de β_i :

$$\beta_1 = 15,89^\circ; \beta_2 = 104,11^\circ; \beta_3 = 135,9^\circ; \beta_4 = 224,11^\circ; \beta_5 = 255,9^\circ; \beta_6 = 344,11^\circ$$

Interfaz Gráfica

Para crear la interfaz gráfica se utilizó el lenguaje Python [22] (programación orientada a objetos). La interfaz se creó utilizando el programa de software libre “Eclipse Java Oxygen” [23] el cual soporta lenguajes como Java, C y Python, entre otros.

En el ambiente de programación de Python, existe una biblioteca que permite la creación de GUIs (graphical user interfaces) o interfaces gráficas de usuario, llamada “Tkinter”.

1. Tkinter

Esta biblioteca de Python permite la creación de botones, etiquetas, barras deslizantes, etc. Que nos permitirán realizar los cálculos de los ángulos de giro de los actuadores a través de las ecuaciones presentadas en el capítulo 2. Antes de comenzar con *Tkinter* es necesario realizar algunas definiciones:

- **Ventana:** Este término tiene diferentes definiciones según el contexto, pero en general se refiere a un área rectangular en alguna parte de su pantalla de visualización.
- **Widget:** Es un término genérico para cualquier bloque de construcción que componen una aplicación en una interfaz gráfica de usuario. Algunos ejemplos son: botones, campos de textos, marcos, y etiquetas de textos.
- **Marco:** En *Tkinter* el widget marco es la unidad básica de organización de diseño complejo. Un marco es un área rectangular la cual contiene los widgets.
- **Child, parent:** Cuando se crea algún widget, se genera una relación “padre – hijo”. Por ejemplo, si se ubica una etiqueta de texto dentro de un marco, el marco es el “parent” de la etiqueta.

Para crear la ventana (raíz) es necesario incluir la siguiente línea de código al comienzo del programa:

```
Interfaz = Tk()
Interfaz.geometry("1000x400")
Interfaz.title("Stewart")
```

Las tres líneas de código representan la creación de la ventana (raíz), el tamaño de la ventana y el título que tendrá la ventana. Además, para que la ventana se continúe ejecutando es necesario que al final del programa se incluya la siguiente línea de código:

Interfaz.mainloop()

Dentro del programa creado se utilizaron los siguientes widgets: **Label** (etiqueta), **Scale** (escalares) y **Button** (botón).

2. Label

Este widget permite mostrar una o más líneas de texto en un mismo estilo. Para crear un *Label* en la ventana raíz o el marco se utiliza el siguiente código:

$$w = tk.Label(parent, option, ...)$$

Donde *tk* puede ser omitido, *parent* corresponde al lugar donde se encontrará el *Label*, *w* es el nombre asignado al *Label* y por último *option* corresponde a las opciones que se le asignarán al *Label* (tamaño, fuente, etc).

3. Scale

El propósito de la *Scale* es permitir al usuario “setear” un valor entero o flotante dentro de un rango especificado. En la figura siguiente se presentan 2 *Scale*:

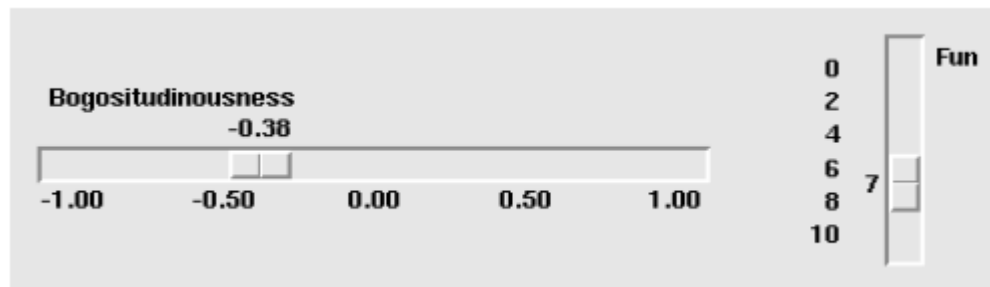


Fig. 44. Ejmplo de *Scale*.

Como se observa en la figura anterior cada *Scale* muestra un *Slider* donde el usuario puede arrastrar el cursor para cambiar el valor. Para crear un *Scale* se utiliza un código similar al mostrado para los *Label*:

$$w = tk.Scale(parent, option, ...)$$

Como se observa, la creación del *Scale* es idéntica a la del *Label*, pero la diferencia radica en las opciones de creación que tiene cada una. Por ejemplo, para la *Scale* las opciones utilizadas son *command* la cual permite llamar a una función cada vez que el *Scale* se mueve, *from_* que permite indicar el valor menor del rango del *Scale* y *to* que corresponde al valor mayor del *Scale*.

4. Button

Por último, el widget *Button* se utiliza para crear botones que permitirán realizar las acciones de algunos movimientos predeterminados a ser calculados por la interfaz. Al igual que los casos mencionados anteriormente el *Button* se crea de la siguiente manera:

$$w = tk.Button(parent, option, ...)$$

Entre las opciones de los *Button* también existe el *command* utilizado para ejecutar una función específica al presionarlo, además se puede escoger el estado inicial del botón entre *Normal* o *Disable*.

Para escoger la ubicación de cada uno de los widgets presentados anteriormente se utiliza el comando *Grid* el cual recibe como valores *Row* (fila) y *Column* (columna) donde se ubicará el widget. El código de la interfaz se encuentra en **ANEXO I – CODIGO DE LA INTERFAZ**.

La figura siguiente muestra la interfaz creada:

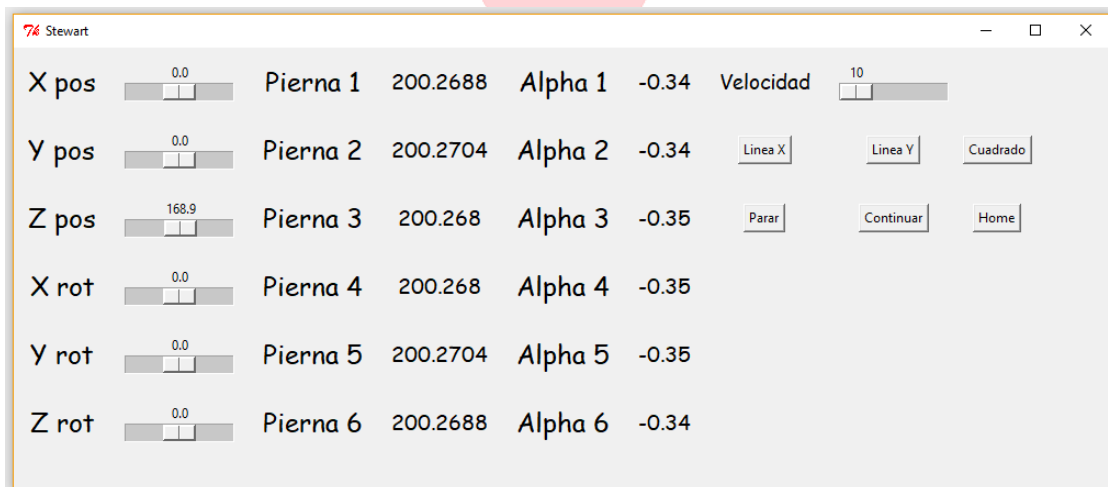


Fig. 45. Interfaz gráfica.

De la figura anterior se puede observar que la interfaz contiene 7 *Scale*, los cuales se utilizan para ingresar los valores de $(p_x, p_y, p_z, \psi, \phi, \theta)$ y la velocidad para los movimientos predeterminados, respectivamente. Luego se crearon 30 *Label*, 18 utilizados para mostrar simples nombres y 12 para presentar el largo de las piernas y los ángulos de rotación que se calculan al variar los valores de rotación y posición de la plataforma superior. Finalmente se utilizaron 6 botones, 3 para movimientos

predeterminados, 1 para detener los movimientos, 1 para que continúen los movimientos (una vez detenidos) y el último para hacer volver a la plataforma a la posición “home”, el cual se calcula a continuación.

Utilizando la ecuación (10) se puede obtener la posición de la plataforma cuando longitud de la pierna toma un valor igual al de la biela (200mm), es decir $S = 200mm$. Reemplazando los valores de B_1 y T_1 en la ecuación 10:

$$200^2 = (144,34 - 149,25)^2 + (150 - 42,5)^2 + (Z_T - 0)^2 \quad (36)$$

$$Z_T = \sqrt{28419,64} \quad (36.1)$$

$$Z_T = 168,58 \text{ mm} \quad (36.2)$$

Valor que corresponde a la posición de la plataforma superior, en el eje z, con respecto a la plataforma base. También es necesario destacar que, por la posición de los actuadores, los valores mostrados en la interfaz representan que la manivela debe bajar (signo negativo) o que debe subir (signo positivo), lo cual se realiza al enviar los pulsos a los actuadores a través de las ecuaciones (27.1) y (27.2).

Cabe mencionar, que en el código no se incluyó la conexión y envío de datos hacia el controlador, esto porque el controlador no será seleccionado dentro de esta memoria (no se encuentra entre los alcances). Los controladores que se podrían utilizar son:

- **Microcontrolador Arduino:** El más común al momento de hablar de controladores. Se presenta ya que, se puede aprovechar la cantidad de salidas PWM que tiene el controlador. Además de ser programado de manera simple y existir una gran cantidad de documentos de control de servo motores con Arduino.
- **Micro Maestro 6-Channel USB Servo Controller [24]:** Este controlador permite controlar al mismo tiempo 6 servomotores. Contando con un programa de control y configuración gratis disponible para Windows, con el cual se pueden realizar pruebas de movimiento de los servos. Además, se pueden enviar datos a este a través del puerto COM donde se encuentre conectado.

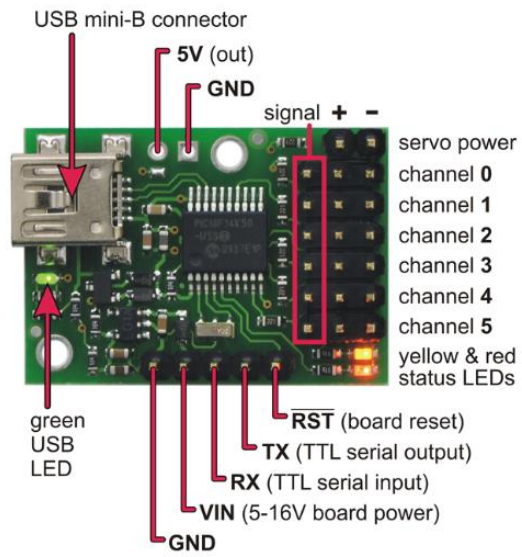


Fig. 46. Micro Maestro 6-channel.



Capítulo 4: Resultados de la Simulación

Introducción

El presente capítulo muestra los resultados de las simulaciones consideradas en el proyecto, los cuales se obtuvieron usando Autodesk Inventor 2018. Para realizarlas se utilizaron los valores de α que se obtienen a través de la interfaz gráfica. Además, se realizaron comparaciones entre 3 plataformas superiores, las cuales varían en el punto de conexión de las piernas con esta misma.

Para poder realizar las simulaciones se debe ingresar, luego de tener la plataforma completamente ensamblada, a la sección de simulación dinámica. A la cual se ingresa siguiendo la ruta:

Entornos → Simulación dinámica

Como se observa en la figura siguiente:

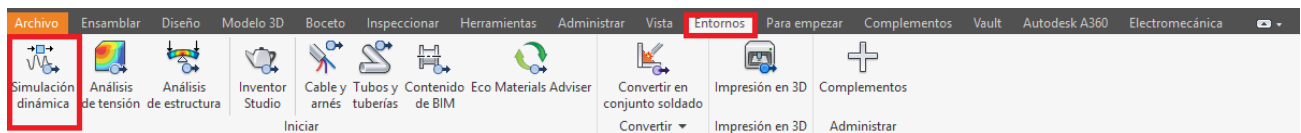


Fig. 47. Acceso al entorno de simulación dinámica

Al ingresar a esta opción de entorno, se mostrará la siguiente pantalla. Donde las partes que afectan la simulación se separan en partes fijas, grupos móviles, uniones estándar y cargas externas.

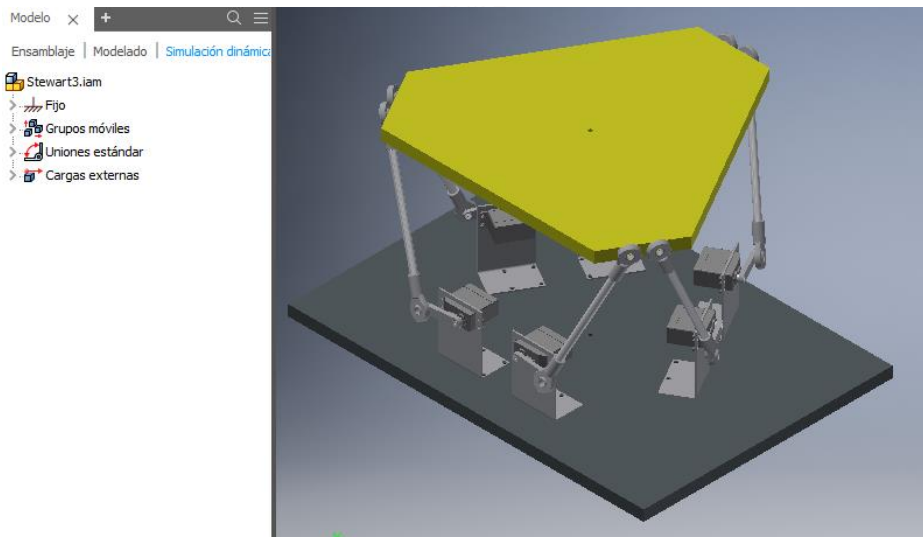


Fig. 48. Simulación dinámica.

Para realizar las simulaciones es necesario desplegar la pestaña “Uniones estándar” la cual permite acceder a todos las uniones que existen en el modelo y a su vez, modificar sus parámetros para generar algún movimiento deseado.

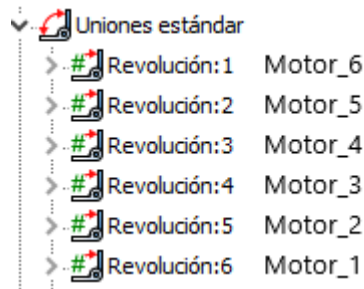


Fig. 49. Uniones estándar.

En la figura anterior se pueden observar las uniones a las cuales se les modificarán los valores para generar el movimiento. Dichas uniones corresponden a la rotación de los servomotores. Además, en la figura se puede observar qué motor modifica cada una de las uniones.

Al ingresar a las propiedades de cada unión se muestra una pestaña general, en donde se encuentran casillas para modificar la fuerza y torsión de la revolución, así también como la opción de bloquear sus grados de libertad. Ingresando a la segunda pestaña “grado de libertad” se le permite al usuario modificar los valores iniciales del ángulo de rotación del motor, además de su velocidad y establecer los límites de giro de dicho motor. Para este caso se debe ingresar a la opción de movimiento impuesto.

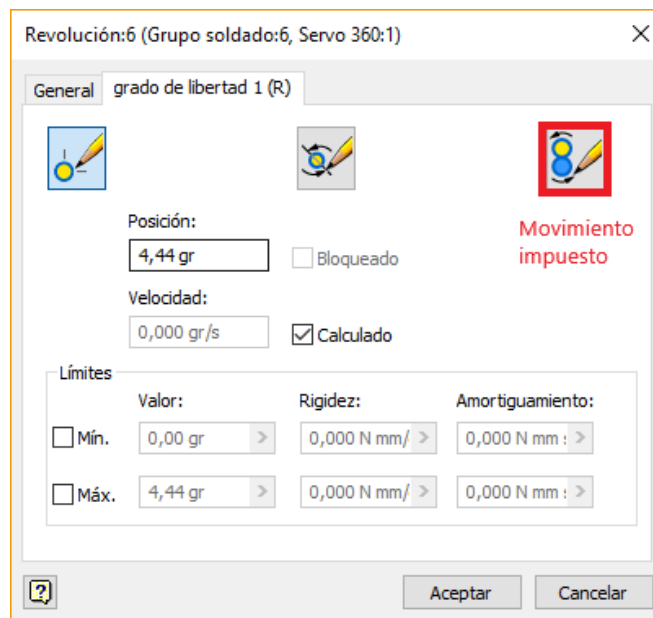


Fig. 50. Opciones de revolución.

Ingresando a esta opción, se presentan 3 movimientos que se pueden imponer al girp (revolución): posición, velocidad y aceleración angulares. Para estas simulaciones se impondrá la posición que debe tomar el ángulo de giro del motor utilizando el valor de α_i calculado a través de la interfaz gráfica utilizando la cinemática inversa.

Al momento de ingresar la posición que se quiere imponer, se presentan varias curvas para generar la variación del ángulo. Para las simulaciones de prueba se utilizará el tramo lineal, el cual requiere un tiempo inicial, ángulo inicial, tiempo final y ángulo final.

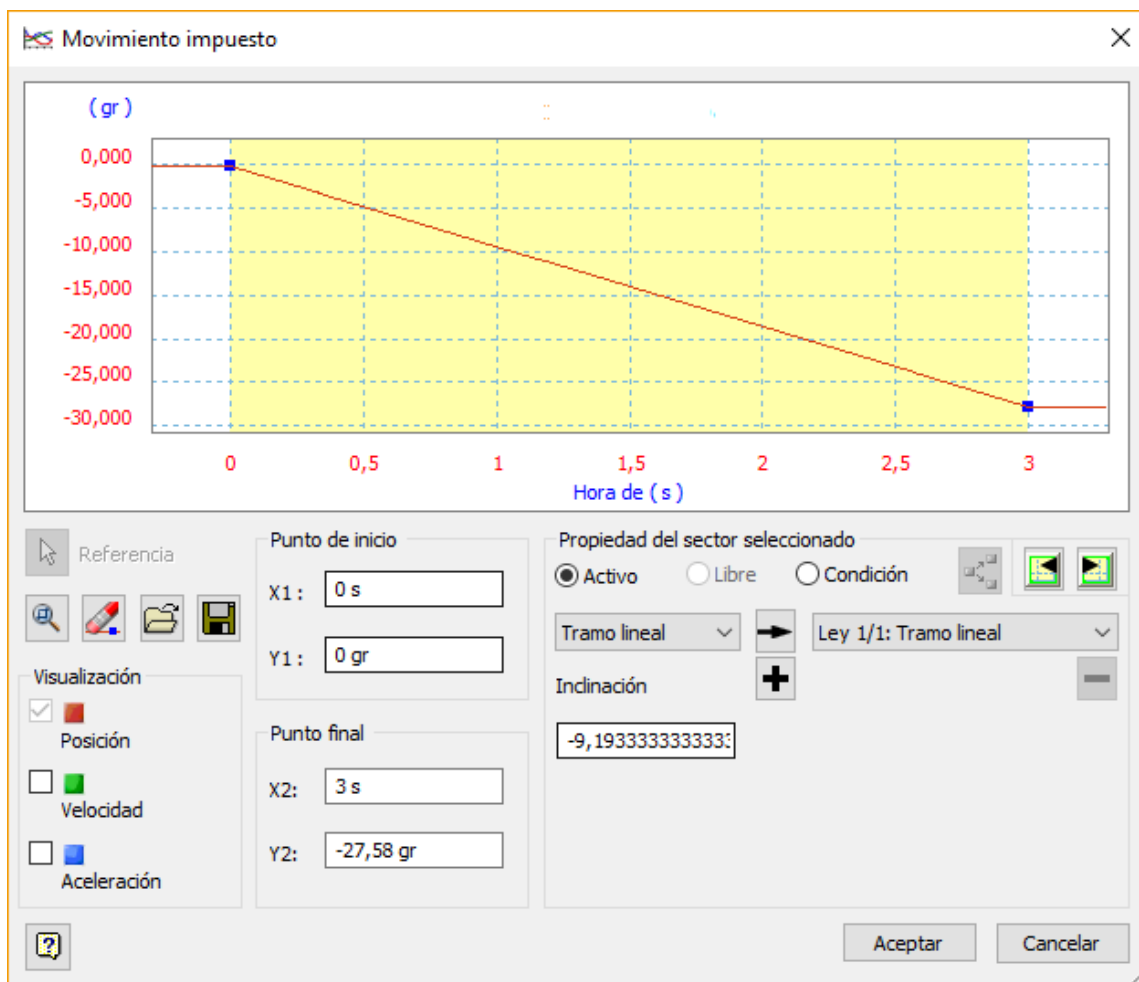



Fig. 51. Posición impuesta.

Se puede observar, en la figura anterior, que al ingresar los valores requeridos para generar el movimiento impuesto se genera una gráfica en la parte superior de la pantalla. Una vez ingresados estos datos, se presiona  para dar inicio a la simulación.

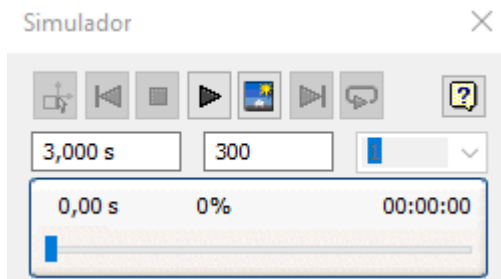


Fig. 52. Menú del simulador.

Ya que, la forma de realizar las simulaciones ha sido explicada, las plataformas superiores creadas, y las tres plataformas de Stewart ensambladas, se muestran a continuación:

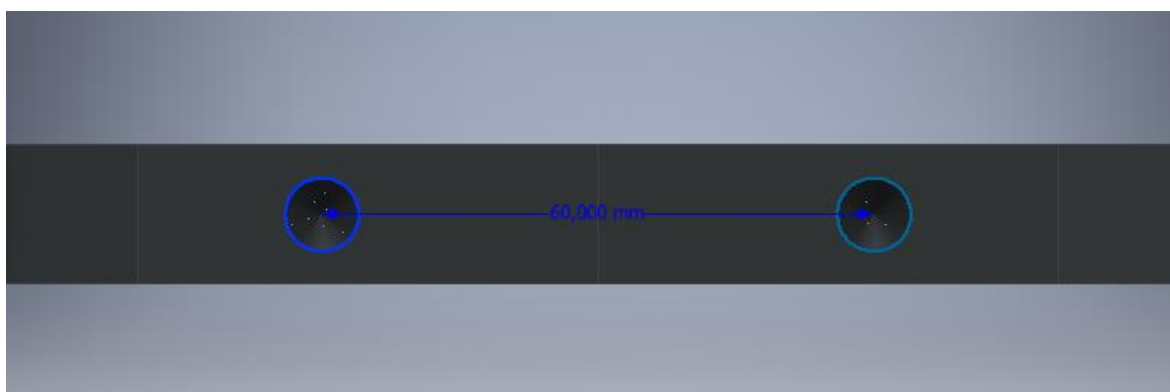


Fig. 53. Plataforma superior M1.

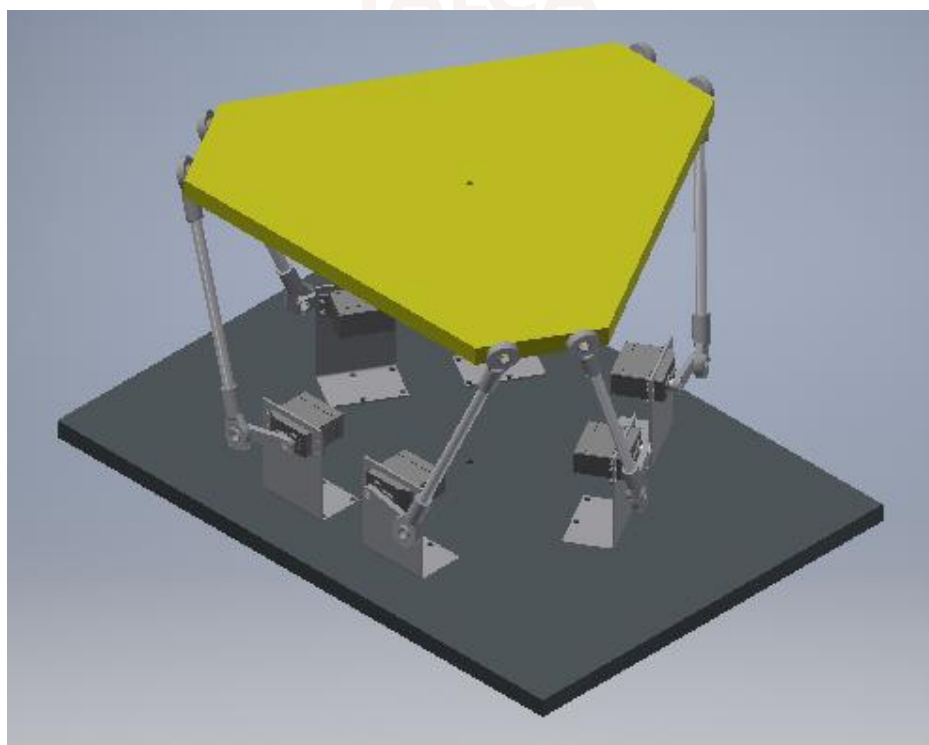


Fig. 54. Plataforma de Stewart (modelo 1).

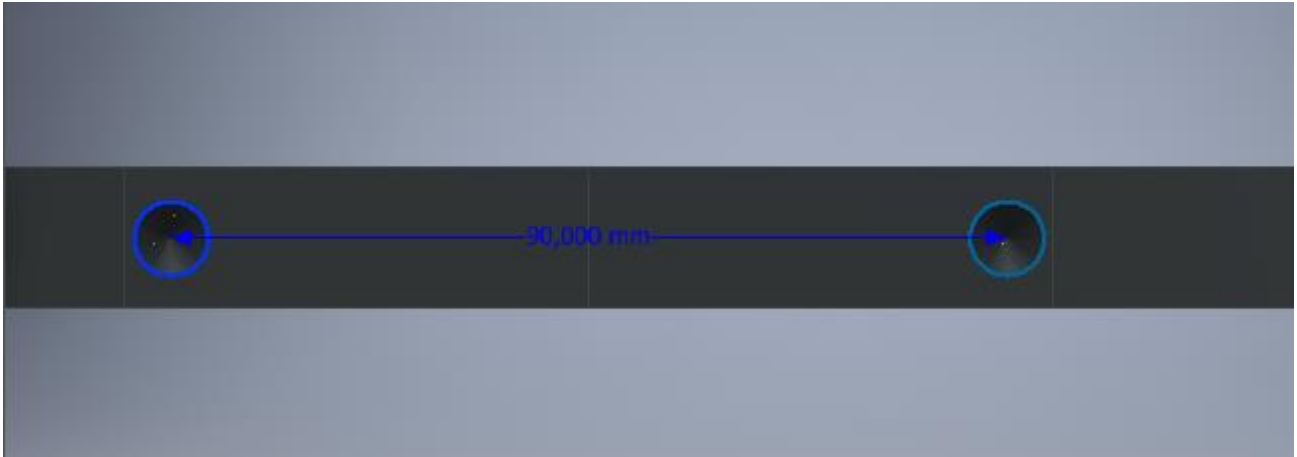


Fig. 55. Plataforma superior M2.

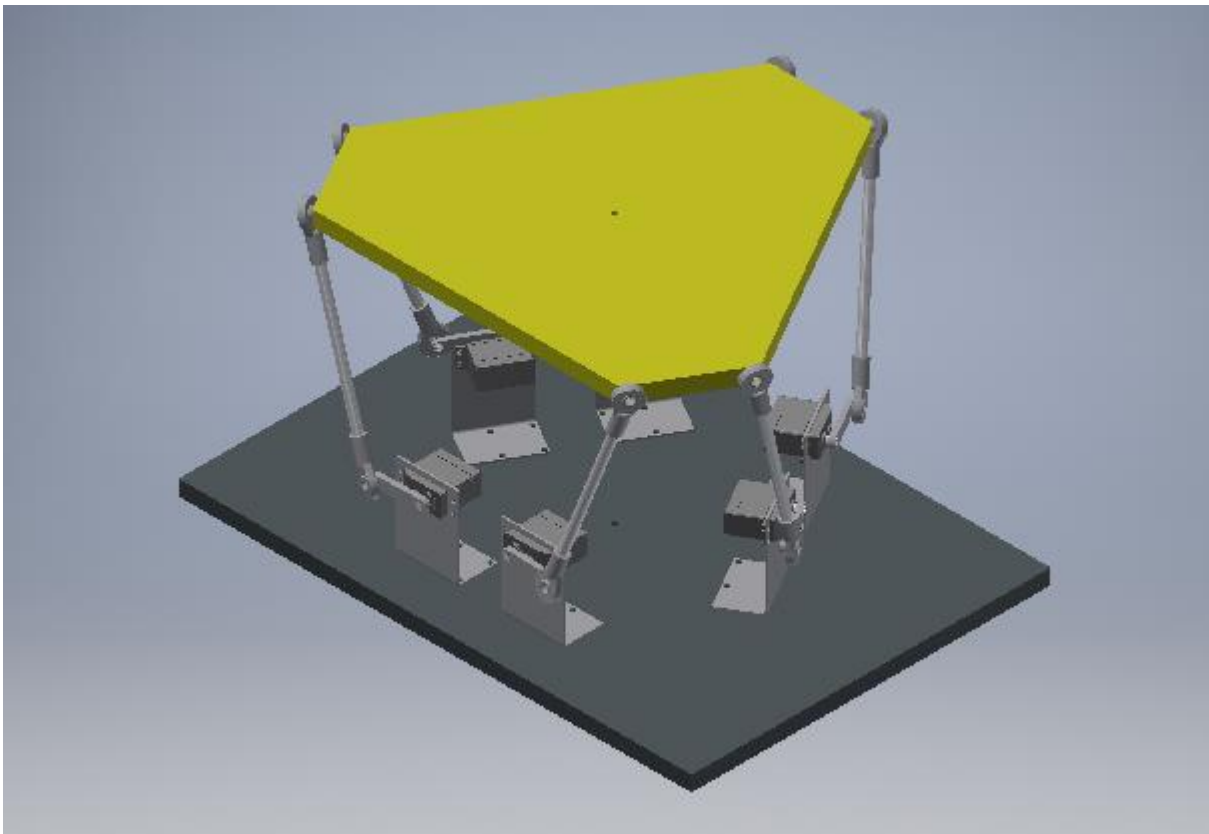


Fig. 56. Plataforma de Stewart (modelo 2).



Fig. 57. Plataforma superior M3.

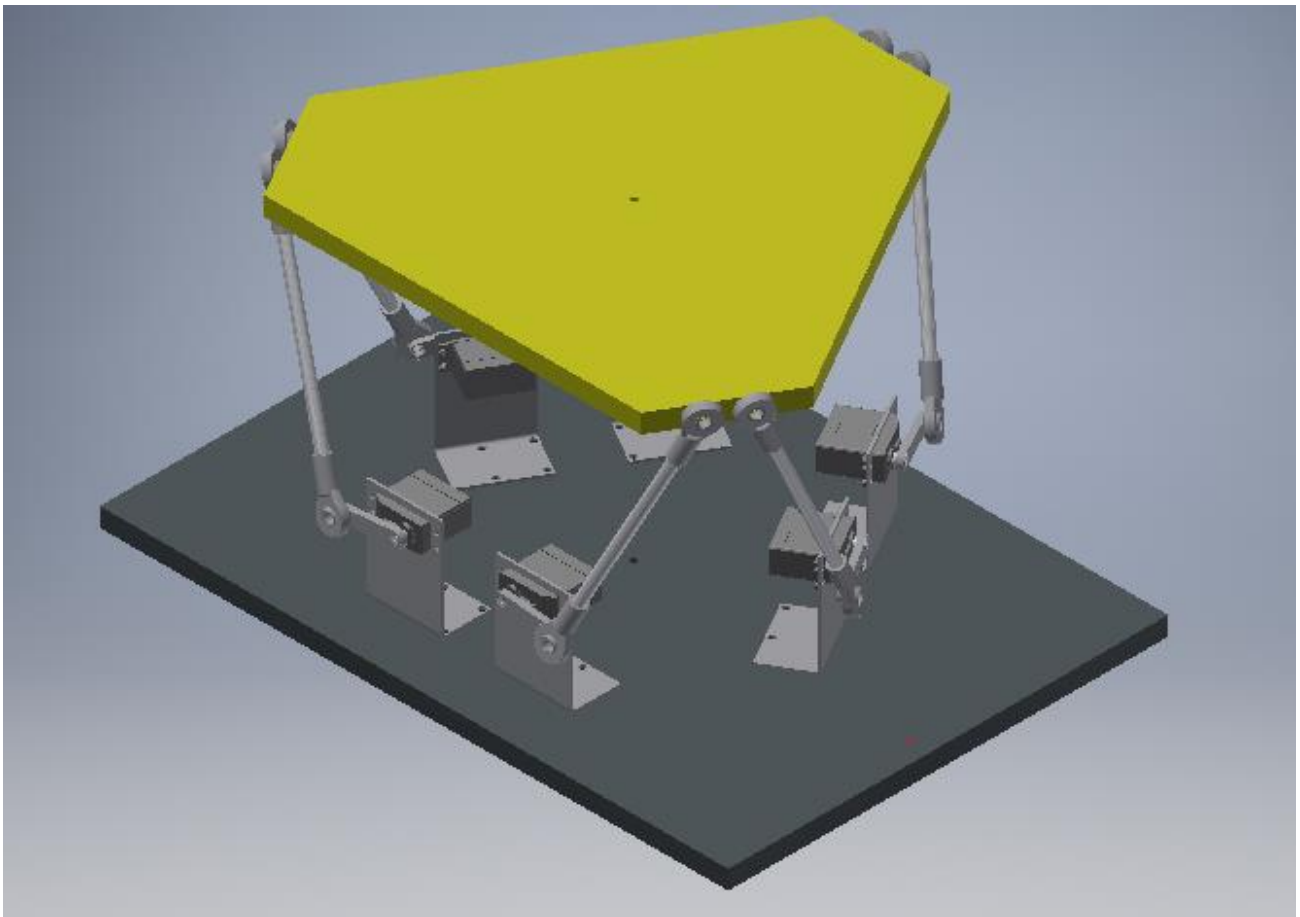


Fig. 58. Plataforma de Stewart (modelo 3).

En las figuras 54, 56 y 58 se observa que la diferencia de cada uno de los modelos de plataforma ensamblados viene dada por los puntos de conexión de las piernas con la plataforma superior, los cuales varían según las distancias mostradas en las figuras 53, 55 y 57 (60 mm, 90 mm y 30 mm respectivamente). Se realizaron 3 pruebas de movimiento a cada plataforma (9 pruebas en total) con las que se obtuvieron los siguientes resultados:

Tabla 4. Resultados prueba simulación 1						
Posición	$X = 40,4 \text{ mm}$		$Y = 0 \text{ mm}$		$Z = 168,6 \text{ mm}$	
α	$\alpha_1 = 9,04$	$\alpha_2 = 43,33$	$\alpha_3 = 12,5$	$\alpha_4 = -12,5$	$\alpha_5 = -43,32$	$\alpha_6 = -9,04$
M1	$X = 46,52 \text{ mm}$		$Y = 0 \text{ mm}$		$Z = 172,24 \text{ mm}$	
Error	$E_X = +6,12 \text{ mm}$		$E_Y = 0 \text{ mm}$		$E_Z = +3,64 \text{ mm}$	
M2	$X = 68,15 \text{ mm}$		$Y = 15,03 \text{ mm}$		$Z = 164,12 \text{ mm}$	
Error	$E_X = +27,75 \text{ mm}$		$E_Y = +15,03 \text{ mm}$		$E_Z = -4,46 \text{ mm}$	
M3	$X = 37,44 \text{ mm}$		$Y = 0 \text{ mm}$		$Z = 168,86 \text{ mm}$	
Error	$E_X = -2,96 \text{ mm}$		$E_Y = 0 \text{ mm}$		$E_Z = +0,26 \text{ mm}$	

Tabla 5, Resultados prueba simulación 2						
Posición	$X = 0 \text{ mm}$		$Y = -22,6 \text{ mm}$		$Z = 168,6 \text{ mm}$	
α	$\alpha_1 = -15$	$\alpha_2 = 0,2$	$\alpha_3 = 16,81$	$\alpha_4 = -13,78$	$\alpha_5 = 2,21$	$\alpha_6 = 18,29$
M1	$X = 2 \text{ mm}$		$Y = -34,75 \text{ mm}$		$Z = 174,51 \text{ mm}$	
Error	$E_X = +2 \text{ mm}$		$E_Y = -12,15 \text{ mm}$		$E_Z = +5,91 \text{ mm}$	
M2	$X = -3,85 \text{ mm}$		$Y = -21,08 \text{ mm}$		$Z = 184,23 \text{ mm}$	
Error	$E_X = -3,85 \text{ mm}$		$E_Y = +1,52 \text{ mm}$		$E_Z = +15,63 \text{ mm}$	
M3	$X = 1,76 \text{ mm}$		$Y = -27,86 \text{ mm}$		$Z = 170,24 \text{ mm}$	
Error	$E_X = +1,76 \text{ mm}$		$E_Y = -5,26 \text{ mm}$		$E_Z = +1,64 \text{ mm}$	

Tabla 6. Resultados prueba simulación 3						
Posición	$X = -22,6 \text{ mm}$		$Y = 21 \text{ mm}$		$Z = 168,6 \text{ mm}$	
α	$\alpha_1 = 27,58$	$\alpha_2 = -14,01$	$\alpha_3 = -1,35$	$\alpha_4 = 29,41$	$\alpha_5 = -16,33$	$\alpha_6 = -4,01$
M1	$X = -33,09 \text{ mm}$		$Y = 26,7 \text{ mm}$		$Z = 173,15 \text{ mm}$	
Error	$E_X = -10,49 \text{ mm}$		$E_Y = +5,7 \text{ mm}$		$E_Z = +4,55 \text{ mm}$	
M2	$X = -41,5 \text{ mm}$		$Y = 33,6 \text{ mm}$		$Z = 174,32 \text{ mm}$	
Error	$E_X = -18,9 \text{ mm}$		$E_Y = +12,6 \text{ mm}$		$E_Z = 5,72 \text{ mm}$	
M3	$X = -26,59 \text{ mm}$		$Y = 21,4 \text{ mm}$		$Z = 169,45 \text{ mm}$	
Error	$E_X = -3,99 \text{ mm}$		$E_Y = +0,4 \text{ mm}$		$E_Z = -0,85 \text{ mm}$	

La forma de tomar las medidas mostradas en las tablas anteriores fue la siguiente. Luego de realizar la configuración de los movimientos impuestos de cada motor, como se explicó anteriormente, y esperar que el modelo complete el movimiento se realizó una medición de centro a centro, como se ve a continuación:

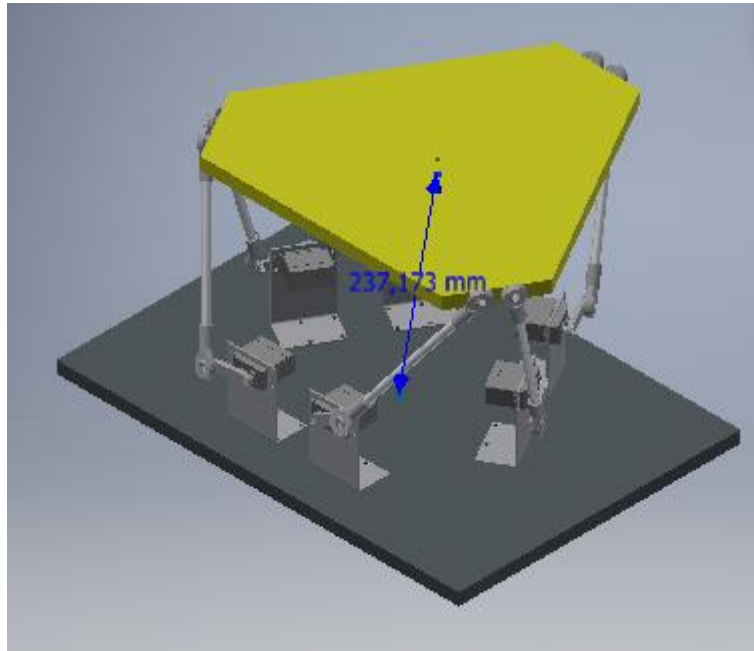


Fig. 59. Plataforma de Stewart

Como se puede observar en la figura, se traza una recta desde el centro de la base hasta el centro de la plataforma superior móvil, para así descomponer los valores correspondientes a los desplazamientos en X, Y, Z . Antes de realizar esta medición es necesario aclarar que el sistema de ejes coordenados, ubicado en el centro de la placa base, de Autodesk Inventor 2018 es el siguiente:

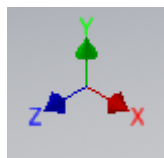


Fig. 60. Eje coordenado Autodesk Inventor

El cual, para los valores de variación de posición de la interfaz son:

$$Z_{Inventor} = X_{Plataforma}$$

$$X_{Inventor} = Y_{Plataforma}$$

$$Y_{Inventor} = Z_{Plataforma}$$

Teniendo en cuenta esto, las mediciones tomadas fueron las siguientes:

Resultados de medidas	
Distancia al centro	237,173 mm
∨ Delta	
Distancia X	21,411 mm
Distancia Y	234,703 mm
Distancia Z	26,593 mm

Fig. 61. Medidas prueba 3, superior M3

De las medidas entregadas, se advierte que no se considera el valor negativo (-) del desplazamiento en X . Para ver esto se debe observar la vista superior de la plataforma.

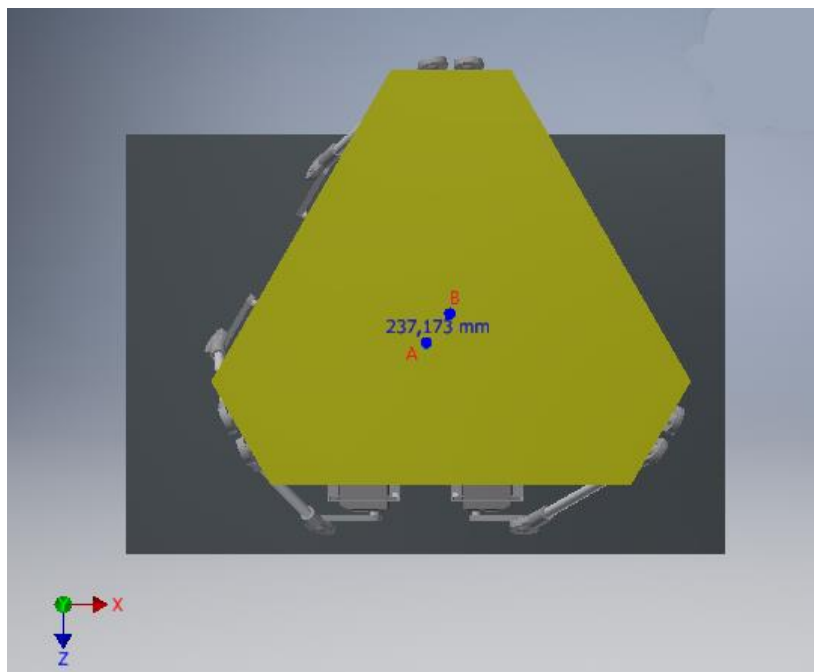


Fig. 62. Desplazamiento prueba 3, superior M3

Se observa claramente, tomando en cuenta que el punto inicial del desplazamiento es A y el punto final B , que la plataforma superior se desplaza en sentido contrario en el eje X ($Z_{Inventor}$).

Finalmente, de las simulaciones realizadas se puede concluir que el modelo que representa una mejor respuesta, basándonos en el error absoluto de cada una de ellas, es la plataforma superior M3 (Fig. 57 y Fig. 58) La cual tiene la conexión de las piernas a la plataforma superior a una distancia de 30 mm (véase Fig. 57). Esto se debe a que, como se observa en la figura 17, la idea original de la plataforma de Stewart es tener una plataforma superior móvil con forma de triángulo, lo cual en la práctica se hace complicado ya que, la conexión de cada uno de los 3 pares de piernas debe ser en el vértice de dicho triángulo lo cual es prácticamente imposible debido a las dimensiones de las juntas.

Por lo mencionado anteriormente es que la plataforma superior se modificó al truncar los vértices del triángulo equilátero, generando el hexágono semirregular con el cual se creó este modelo.



Capítulo 5: Conclusiones

En esta memoria de título se presentó un manipulador robótico paralelo, la plataforma de Stewart. Se analizó su geometría, su cinemática y dinámica, además de su impulsión mediante servomotores. Se realizó un modelo 3D de aquella, una interfaz gráfica regida por la cinemática y simulaciones para comprobar posibles variaciones en el modelo mecánico.

Analizando las ecuaciones de posición y las obtenidas de la cinemática inversa del manipulador, se puede concluir que el tipo de juntas utilizadas para conectar el sistema biela – manivela y completar la conexión con la plataforma superior, no tienen efecto alguno en la variación de posición y orientación de la plataforma superior (matemáticamente hablando).

Continuando con el análisis de la cinemática, los factores relevantes son las dimensiones de la placa base, la forma y dimensiones de la plataforma superior, la longitud de la biela y de la manivela.

Para que el usuario pueda modificar el código de la interfaz gráfica, según la manera o configuración que utilice de la plataforma, los valores mencionados anteriormente se encuentran en una ubicación de fácil acceso para realizar cambios de estos de forma rápida y sencilla.

Finalmente, el uso del software CAD (Autodesk Inventor 2018) no solo permite obtener una visión de cómo se verá la plataforma proyectada, sino también permite realizar pruebas de distintas configuraciones de esta. Aprovechando su fácil uso y acceso a análisis de entornos, entre los que se encuentra la utilizada simulación dinámica, existen otras que se pueden aplicar dependiendo de las necesidades del usuario (por ejemplo, el análisis de tensión y deformación de partes y piezas).

Trabajos Futuros

- Construir una plataforma.
- Realizar selección de controlador para los actuadores.
- Realizar conexión interfaz – controlador.

Referencias

- [1] D. Stewart, «A Platform with Six Degrees of Freedom,» *Proceedings of the Institution of Mechanical Engineers*, vol. 180, pp. 371 - 386, 1965.
- [2] «Merriam - Webster,» [En línea]. Available: <https://www.merriam-webster.com/dictionary/robot>.
- [3] «Real Academia Española,» [En línea]. Available: <http://dle.rae.es/srv/search?m=30&w=robot>.
- [4] [En línea]. Available: http://automata.cps.unizar.es/Historia/Webs/automatas_en_la_historia.htm.
- [5] J. A. Muñoz V., *Introducción a la Robótica*.
- [6] «Ciberneticzoo,» [En línea]. Available: <http://cyberneticzoo.com/man-amplifiers/1958-9-gehandyman-ralph-mosher-american/>.
- [7] «Vinssa,» [En línea]. Available: <https://vinssa.com/robot-scara>.
- [8] [En línea]. Available: <http://tecnologia-lulu.blogspot.com/2012/01/tablas-de-clasificacion-de-robots-segun.html>.
- [9] «Pcexpertos,» [En línea]. Available: <http://pcexpertos.com/2009/08/clasificacion-de-los-robots-segun-la-afri.html>.
- [10] J. M. Rico Martínez, *Las Diferentes Formas del Criterio de Kutzbach - Grübler*, Salamanca.
- [11] U. d. S. E. U. Politécnica, «Slideshare,» [En línea]. Available: <https://es.slideshare.net/borrecan/robotica-industrial-9142300>.
- [12] J. A. Gomez Portilla y J. V. Guacaneme González, *Diseño e Implementación de una Plataforma de Stewart*, Bogota.
- [13] C. C. Nguyen y F. J. Pooran, «Analysis of a closed - kinematic chain robot manipulator,» 1988.
- [14] «UKIRT,» [En línea]. Available: <http://www.ukirt.hawaii.edu/public/tele-descrip.html>.
- [15] F. Szufnarowski, «Stewart Platform with Fixed Rotary Actuators: a Low Cost Design Study».
- [16] K. Liu, J. M. Fitzgerald y F. L. Lewis, «Kinematic Analysis of a Stewart Platform Manipulator,» *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, vol. 40, n° 2, 1991.
- [17] J. Corchuelo, «Scrib,» 02 Nov 2016. [En línea]. Available: <https://es.scribd.com/document/329698629/Diferencias-Entre-Modelo-Cinematico-Directo-e-Inverso>.
- [18] J. A. Muñoz V., *Descripciones Espaciales y Transformaciones*.
- [19] Desconocido. [En línea]. Available: <https://cdn.instructables.com/ORIG/FFI/8ZXW/I55MMY14/FFI8ZXWI55MMY14.pdf>. [Último acceso: 31 08 2017].
- [20] «Grabcad community,» [En línea]. Available: <https://grabcad.com/library/tag/servo>.
- [21] SKF, *Rótulas y Cabezas de Articulación*.
- [22] «Python,» [En línea]. Available: <https://www.python.org/>.
- [23] «Eclipse fundation,» [En línea]. Available: <https://www.eclipse.org/>.
- [24] «Pololu robotics & electronics,» [En línea]. Available: <https://www.pololu.com/product/1350>.

- [25] A. S. Indrawanto, «Desing and Control of the Stewart Platform Robot,» *Third Asia International Conference on Modelling & Simulation*, 2009.
- [26] N. X. González Alvarez y E. P. Reinoso Mendoza, «Estudio, Diseño y Construcción de una Plataforma Robótica Didáctica tipo Stewart Aplicada al Estudio de Controladores Difusos,» Cuenca, 2011.
- [27] J. A. Muñoz V., *Modelo Cinemático*.



Anexo I – Código de la Interfaz

```
'''
```

```
Created on 10-04-2018
```

```
@author: Victor
```

```
'''
```

```
from Tkinter import *
import numpy as np
from math import *
from __builtin__ import float
from math import degrees
from numpy.core.multiarray import dtype
from cgitb import text
```

```
Interfaz = Tk()
Interfaz.geometry("1000x400")
Interfaz.title("Stewart")
```

```
#Variables globales iniciales
#Posicion y Orientacion inicial
```

```
x = 0
y = 0
z = 168.58
yaw = 0
pitch = 0
roll = 0
ii = jj = 0
aa = bb = cc = s = 0
running = True
```

```
#Matriz de Traslacion y Rotacion Inicial
```

```
T = np.array([[x],[y],[z]])
R = np.array([[1,0,0],[0,1,0],[0,0,1]])
```

```
#Largo piernas inicial
```

```
s1 = s2 = s3 = s4 = s5 = s6 = 1
l1 = l2 = l3 = l4 = l5 = l6 = np.array([[1],[1],[1]])
```

```
#L, M, N
```

```
L1 = L2 = L3 = L4 = L5 = L6 = 1
M1 = M2 = M3 = M4 = M5 = M6 = 1
N1 = N2 = N3 = N4 = N5 = N6 = 1
```

```
#Angulos
```

```
alpha1 = alpha2 = alpha3 = alpha4 = alpha5 = alpha6 = 1
```

```
#Ancho de pulso
```

```
W1 = W2 = W3 = W4 = W5 = W6 = 0
```



#Constantes

#Coordenadas plataforma base [X,Y,Z]

```
B1 = np.array([[149.25],[42.5],[0]])
B2 = np.array([[ -37.82],[150.5],[0]])
B3 = np.array([[ -111.43],[108],[0]])
B4 = np.array([[ -111.43],[ -108],[0]])
B5 = np.array([[ -37.82],[ -150.5],[0]])
B6 = np.array([[149.25],[ -42.5],[0]])
```

#Coordenadas plataforma superior [x,y,z]

```
t1 = np.array([[144.3376],[150],[0]])
t2 = np.array([[57.735],[200],[0]])
t3 = np.array([[ -202.0726],[50],[0]])
t4 = np.array([[ -202.0726],[ -50],[0]])
t5 = np.array([[57.735],[ -200],[0]])
t6 = np.array([[144.3376],[ -150],[0]])
```

#Dimensiones manivela y biela

```
bi = 200
mi = 45
alpha0 = radians(0.3505)
```

#Angulos de servo con respecto a X y datos del servomotor

```
Beta = [radians(15.89), radians(104.11), radians(135.9), radians(224.11), radians(255.9),
radians(344.11)]
W0 = 1500
r = 1600/(pi)
```

#Funciones

```
def Square():
    global aa, bb, ii, jj , running, s, cc
    s = speed.get()
    cc = 2

    if running and cc == 2:

        if bb == 0 and aa == 0:
            ii += 1
            X.set(ii)
            if ii == 30:
                aa = 1
        else:
            if bb == 0 and aa == 1:
                jj +=1
                Y.set(jj)
                if jj == 30:
                    bb = 1
            else:
                if bb == 1 and aa == 1:
                    ii -= 1
                    X.set(ii)
                    if ii == -30:
```

```

        aa = 0
    else:
        if bb == 1 and aa == 0 :
            jj -= 1
            Y.set(jj)
            if jj == -30:
                bb = 0

    Interfaz.after(s, Square)
def lineaX():
    global running, s, ii, aa, cc
    cc = 0
    s = speed.get()
    if running and cc == 0:
        if aa == 0:
            ii += 1
            X.set(ii)
            if ii == 40:
                aa = 1
        else:
            ii -=1
            X.set(ii)
            if ii == -40:
                aa = 0

    Interfaz.after(s, lineaX)
def lineaY():
    global running, s, ii, aa, cc
    cc = 1
    s = speed.get()
    if running and cc == 1:
        if aa == 0:
            ii += 1
            Y.set(ii)
            if ii == 40:
                aa = 1
        else:
            ii -=1
            Y.set(ii)
            if ii == -40:
                aa = 0

    Interfaz.after(s, lineaY)

def Continue():
    global running, cc, s
    running = True
    if cc == 0:
        Interfaz.after(s, lineaX)
    else:
        if cc == 1:
            Interfaz.after(s, lineaY)
        else:

```



```

        if cc ==2:
            Interfaz.after(s, Square)

def Stop():
    global running
    running = False

def Calcular(val):
    global x, y, z, pitch, yaw, roll, T, R, s1, s2, s3, s4, s5, s6, l1, l2,
    l3, l4, l5, l6, L1, L2, L3, L4, L5, L6, M1, M2, M3, M4, M5, M6
    global alpha1, alpha2, alpha3, alpha4, alpha5, alpha6, W1, W2, W3, W4, W5, W6
    global B1, B2, B3, B4, B5, B6, t1, t2, t3, t4, t5, t6, bi, mi, alpha0, Beta, W0, r

    try:
        x = var_X.get()
        y = var_Y.get()
        z = var_Z.get()

        yaw = radians(rot_X.get())
        pitch = radians(rot_Y.get())
        roll = radians(rot_Z.get())

        a = cos(pitch) * cos(roll)
        b = (sin(yaw) * sin(pitch) * cos(roll)) - (cos(yaw) * sin(roll))
        c = (sin(yaw) * sin(pitch)) + (cos(yaw) * sin(pitch) * cos(roll))
        e = cos(pitch) * sin(roll)
        f = (cos(yaw) * cos(roll)) + (sin(yaw) * sin(pitch) * sin(roll))
        g = (cos(yaw) * sin(pitch) * sin(roll)) - (sin(yaw) * cos(roll))
        i = -sin(pitch)
        j = sin(yaw) * cos(pitch)
        k = cos(yaw) * cos(pitch)

        R = np.array([[a,b,c],[e,f,g], [i,j,k]])
        T = np.array([[x],[y],[z]])

        l1 = np.subtract(np.add(T,np.dot(R,t1)),B1)
        l2 = np.subtract(np.add(T,np.dot(R,t2)),B2)
        l3 = np.subtract(np.add(T,np.dot(R,t3)),B3)
        l4 = np.subtract(np.add(T,np.dot(R,t4)),B4)
        l5 = np.subtract(np.add(T,np.dot(R,t5)),B5)
        l6 = np.subtract(np.add(T,np.dot(R,t6)),B6)

        s1 = np.linalg.norm(l1)
        s2 = np.linalg.norm(l2)
        s3 = np.linalg.norm(l3)
        s4 = np.linalg.norm(l4)
        s5 = np.linalg.norm(l5)
        s6 = np.linalg.norm(l6)

    #Calculo de angulo
        L1 = pow(s1,2) - pow(bi,2) + pow(mi,2)
        L2 = pow(s2,2) - pow(bi,2) + pow(mi,2)
        L3 = pow(s3,2) - pow(bi,2) + pow(mi,2)

```

```

L4 = pow(s4,2) - pow(bi,2) + pow(mi,2)
L5 = pow(s5,2) - pow(bi,2) + pow(mi,2)
L6 = pow(s6,2) - pow(bi,2) + pow(mi,2)

N1 = 2 * mi * ((cos(Beta[0])*l1[0]) + (sin(Beta[0])*l1[1]))
N2 = 2 * mi * ((cos(Beta[1])*l2[0]) + (sin(Beta[1])*l2[1]))
N3 = 2 * mi * ((cos(Beta[2])*l3[0]) + (sin(Beta[2])*l3[1]))
N4 = 2 * mi * ((cos(Beta[3])*l4[0]) + (sin(Beta[3])*l4[1]))
N5 = 2 * mi * ((cos(Beta[4])*l5[0]) + (sin(Beta[4])*l5[1]))
N6 = 2 * mi * ((cos(Beta[5])*l6[0]) + (sin(Beta[5])*l6[1]))

M1 = 2 * mi * l1[2]
M2 = 2 * mi * l2[2]
M3 = 2 * mi * l3[2]
M4 = 2 * mi * l4[2]
M5 = 2 * mi * l5[2]
M6 = 2 * mi * l6[2]

alpha1 = asin(L1/sqrt(pow(M1, 2) + pow(N1, 2))) - atan(N1/M1)
alpha2 = asin(L2/sqrt(pow(M2, 2) + pow(N2, 2))) - atan(N2/M2)
alpha3 = asin(L3/sqrt(pow(M3, 2) + pow(N3, 2))) - atan(N3/M3)
alpha4 = asin(L4/sqrt(pow(M4, 2) + pow(N4, 2))) - atan(N4/M4)
alpha5 = asin(L5/sqrt(pow(M5, 2) + pow(N5, 2))) - atan(N5/M5)
alpha6 = asin(L6/sqrt(pow(M6, 2) + pow(N6, 2))) - atan(N6/M6)

W1 = W0 + (alpha1 - alpha0)*r
W3 = W0 + (alpha3 - alpha0)*r
W5 = W0 + (alpha5 - alpha0)*r
W2 = W0 - (alpha2 - alpha0)*r
W4 = W0 - (alpha4 - alpha0)*r
W6 = W0 - (alpha6 - alpha0)*r

z1.set(round(s1,4))
z2.set(round(s2,4))
z3.set(round(s3,4))
z4.set(round(s4,4))
z5.set(round(s5,4))
z6.set(round(s6,4))

a1.set(round(degrees(alpha1),2))
a2.set(round(degrees(alpha2),2))
a3.set(round(degrees(alpha3),2))
a4.set(round(degrees(alpha4),2))
a5.set(round(degrees(alpha5),2))
a6.set(round(degrees(alpha6),2))

except ValueError:

z1.set("ERROR")
z2.set("ERROR")
z3.set("ERROR")
z4.set("ERROR")
z5.set("ERROR")
z6.set("ERROR")

```

```
a1.set("ERROR")
a2.set("ERROR")
a3.set("ERROR")
a4.set("ERROR")
a5.set("ERROR")
a6.set("ERROR")
```

```
def Home():
    global running, aa,bb,cc, ii,jj
    running = True
    aa = bb = cc = ii = jj = 0
    X.set(0)
    Y.set(0)
    Z.set(168.58)
    Rx.set(0)
    Ry.set(0)
    Rz.set(0)
```

```
Xpos = Label(Interfaz, text = "X pos", font=("Comic sans ms", 18))
Xpos.grid(row = 0, column = 0, padx = 10, pady = 10)
```

```
Ypos = Label(Interfaz, text = "Y pos", font = ("comic sans ms", 18))
Ypos.grid(row = 1, column = 0, padx = 10, pady = 10)
```

```
Zpos = Label(Interfaz, text = "Z pos", font = ("comic sans ms", 18))
Zpos.grid(row = 2, column = 0, padx = 10, pady = 10)
```

```
Xrot = Label(Interfaz, text = "X rot", font = ("comic sans ms", 18))
Xrot.grid(row = 3, column = 0, padx = 10, pady = 10)
```

```
Yrot = Label(Interfaz, text = "Y rot", font = ("comic sans ms", 18))
Yrot.grid(row = 4, column = 0, padx = 10, pady = 10)
```

```
Zrot = Label(Interfaz, text = "Z rot", font = ("comic sans ms", 18))
Zrot.grid(row = 5, column = 0, padx = 10, pady = 10)
```

```
var_X = DoubleVar()
var_Y = DoubleVar()
var_Z = DoubleVar()
rot_X = DoubleVar()
rot_Y = DoubleVar()
rot_Z = DoubleVar()
```

```
z1 = StringVar()
z2 = StringVar()
z3 = StringVar()
z4 = StringVar()
z5 = StringVar()
z6 = StringVar()
```

```
speed = IntVar()
```

```
a1 = StringVar()
```

```

a2 = StringVar()
a3 = StringVar()
a4 = StringVar()
a5 = StringVar()
a6 = StringVar()

X = Scale(Interfaz, from_ = -55, to = 55, resolution = 0.1, variable = var_X, command =
Calcular, orient = HORIZONTAL)
X.grid(row = 0, column = 1, padx = 10, pady = 10)
Y = Scale(Interfaz, from_ = -55, to = 55, resolution = 0.1, variable = var_Y, command =
Calcular, orient = HORIZONTAL)
Y.grid(row = 1, column = 1, padx = 10, pady = 10)
Z = Scale(Interfaz, from_ = 120, to = 215, resolution = 0.1, variable = var_Z, command
=Calcular, orient = HORIZONTAL)
Z.grid(row = 2, column = 1, padx = 10, pady = 10)
Z.set(168.58)

Rx = Scale(Interfaz, from_ = -10, to = 10, resolution = 0.1, variable = rot_X, command =
Calcular, orient = HORIZONTAL)
Rx.grid(row = 3, column = 1, padx = 10, pady = 10)
Ry = Scale(Interfaz, from_ = -10, to = 10, resolution = 0.1, variable = rot_Y, command =
Calcular, orient = HORIZONTAL)
Ry.grid(row = 4, column = 1, padx = 10, pady = 10)
Rz = Scale(Interfaz, from_ = -10, to = 10, resolution = 0.1, variable = rot_Z, command =
Calcular, orient = HORIZONTAL)
Rz.grid(row = 5, column = 1, padx = 10, pady = 10)

P1 = Label(Interfaz, text = "Pierna 1", font = ("Comic sans ms", 18)).grid(row = 0,
column = 2, padx = 10, pady = 10)
P2 = Label(Interfaz, text = "Pierna 2", font = ("Comic sans ms", 18)).grid(row = 1,
column = 2, padx = 10, pady = 10)
P3 = Label(Interfaz, text = "Pierna 3", font = ("Comic sans ms", 18)).grid(row = 2,
column = 2, padx = 10, pady = 10)
P4 = Label(Interfaz, text = "Pierna 4", font = ("Comic sans ms", 18)).grid(row = 3,
column = 2, padx = 10, pady = 10)
P5 = Label(Interfaz, text = "Pierna 5", font = ("Comic sans ms", 18)).grid(row = 4,
column = 2, padx = 10, pady = 10)
P6 = Label(Interfaz, text = "Pierna 6", font = ("Comic sans ms", 18)).grid(row = 5,
column = 2, padx = 10, pady = 10)

A1 = Label(Interfaz, text = "Alpha 1", font = ("Comic sans ms", 18)). grid(row = 0,
column = 4, padx = 10, pady = 10)
A2 = Label(Interfaz, text = "Alpha 2", font = ("Comic sans ms", 18)). grid(row = 1,
column = 4, padx = 10, pady = 10)
A3 = Label(Interfaz, text = "Alpha 3", font = ("Comic sans ms", 18)). grid(row = 2,
column = 4, padx = 10, pady = 10)
A4 = Label(Interfaz, text = "Alpha 4", font = ("Comic sans ms", 18)). grid(row = 3,
column = 4, padx = 10, pady = 10)
A5 = Label(Interfaz, text = "Alpha 5", font = ("Comic sans ms", 18)). grid(row = 4,
column = 4, padx = 10, pady = 10)
A6 = Label(Interfaz, text = "Alpha 6", font = ("Comic sans ms", 18)). grid(row = 5,
column = 4, padx = 10, pady = 10)

p1 = Label(Interfaz, textvariable = z1, font = ("Comic sans ms", 14)).grid(row = 0,
column = 3, padx = 10, pady = 10)

```



```

p2 = Label(Interfaz, textvariable = z2, font = ("Comic sans ms", 14)).grid(row = 1,
column = 3, padx = 10, pady = 10)
p3 = Label(Interfaz, textvariable = z3, font = ("Comic sans ms", 14)).grid(row = 2,
column = 3, padx = 10, pady = 10)
p4 = Label(Interfaz, textvariable = z4, font = ("Comic sans ms", 14)).grid(row = 3,
column = 3, padx = 10, pady = 10)
p5 = Label(Interfaz, textvariable = z5, font = ("Comic sans ms", 14)).grid(row = 4,
column = 3, padx = 10, pady = 10)
p6 = Label(Interfaz, textvariable = z6, font = ("Comic sans ms", 14)).grid(row = 5,
column = 3, padx = 10, pady = 10)

a11 = Label(Interfaz, textvariable = a1, font = ("Comic sans ms",14)). grid(row = 0,
column = 5, padx = 10, pady = 10)
a12 = Label(Interfaz, textvariable = a2, font = ("Comic sans ms",14)). grid(row = 1,
column = 5, padx = 10, pady = 10)
a13 = Label(Interfaz, textvariable = a3, font = ("Comic sans ms",14)). grid(row = 2,
column = 5, padx = 10, pady = 10)
a14 = Label(Interfaz, textvariable = a4, font = ("Comic sans ms",14)). grid(row = 3,
column = 5, padx = 10, pady = 10)
a15 = Label(Interfaz, textvariable = a5, font = ("Comic sans ms",14)). grid(row = 4,
column = 5, padx = 10, pady = 10)
a16 = Label(Interfaz, textvariable = a6, font = ("Comic sans ms",14)). grid(row = 5,
column = 5, padx = 10, pady = 10)

Label(Interfaz, text = "Velocidad", font = ("Comic sans ms", 14)).grid(row = 0, column =
6, padx = 10, pady = 10)
spe = Scale(Interfaz, variable = speed,from_ = 10, to = 1000, resolution = 10, orient =
HORIZONTAL).grid(row = 0, column = 7,padx = 10, pady = 10)
Mx = Button(Interfaz, text = "Linea X", command = lineaX).grid(row = 1, column = 6)
stop =Button(Interfaz, text = "Parar", command = Stop).grid(row = 2, column = 6)
Seguir = Button(Interfaz, text = "Continuar", command = Continue).grid(row = 2, column =
7)
Cuad = Button(Interfaz, text = "Cuadrado", command = Square).grid(row = 1, column = 8)

My = Button(Interfaz, text = "Linea Y", command = lineaY).grid(row = 1, column = 7)
Homa = Button(Interfaz, text = "Home", command = Home).grid(row = 2, column =8)
Interfaz.mainloop()

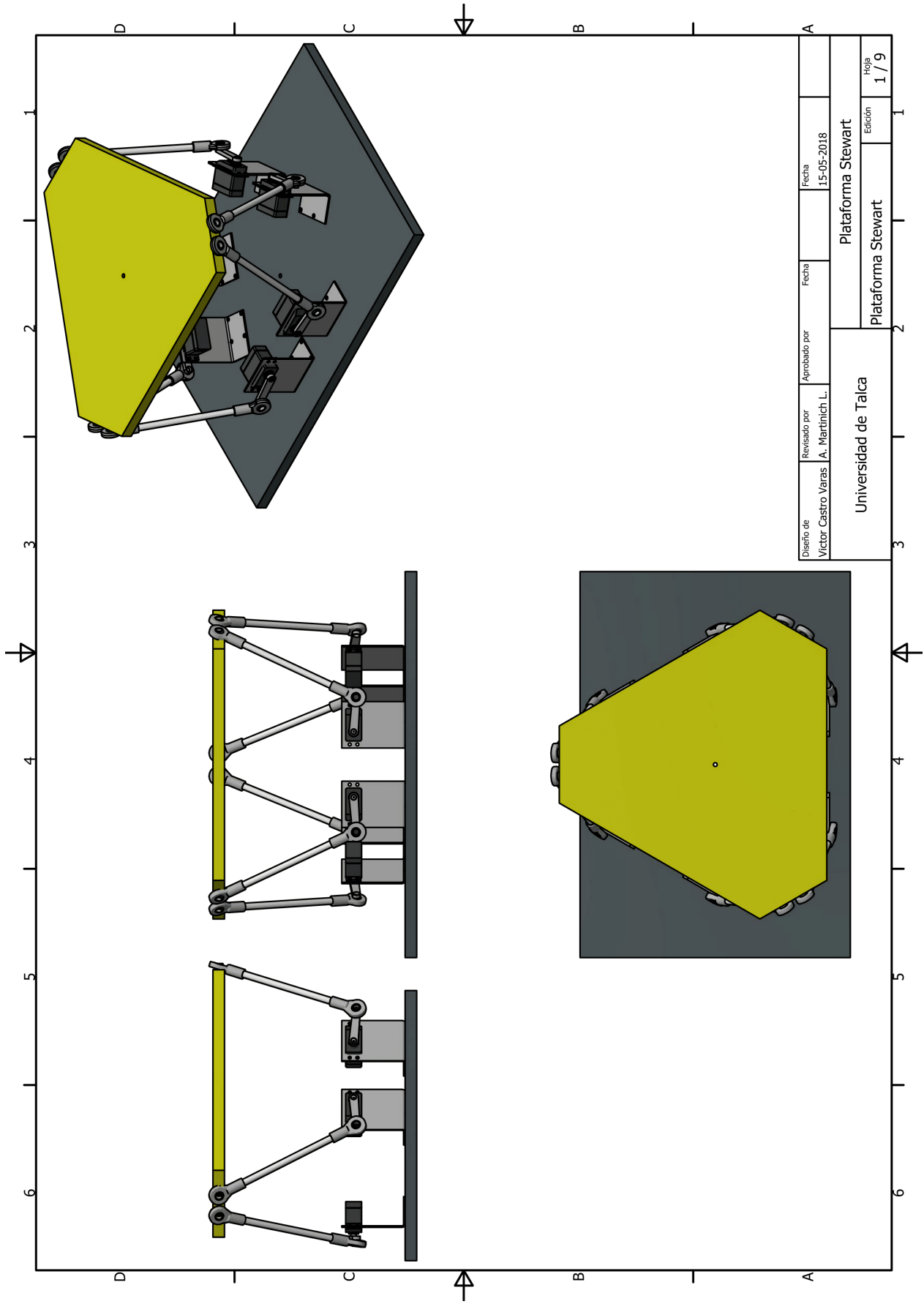
```

Anexo II – Planos Mecánicos

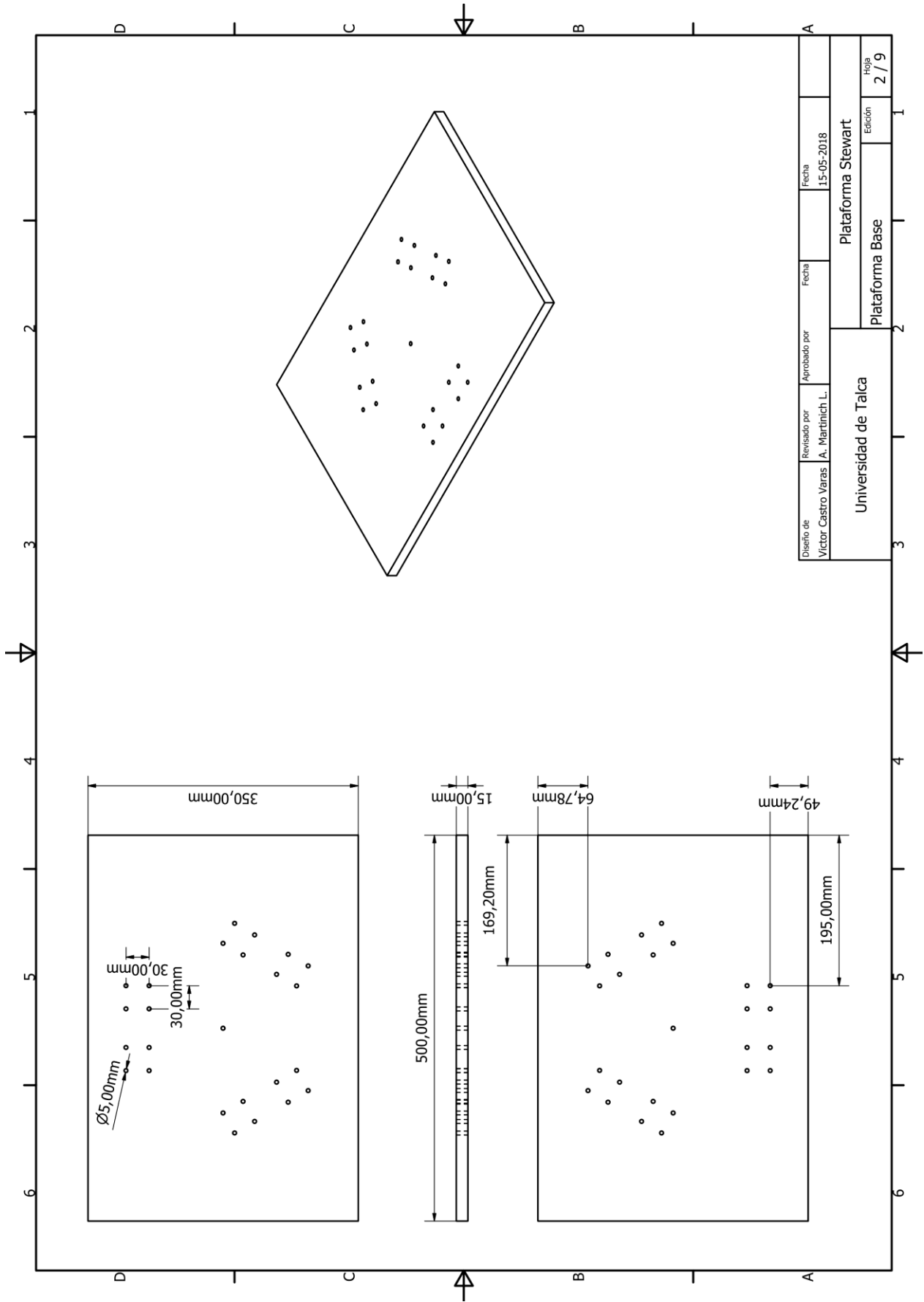
Nro. plano

1/9	Plataforma Stewart
2/9	Plataforma Base
3/9	Base Servo Par
4/9	Base Servo Impar
5/9	Manivela
6/9	Biela
7/9	Superior M1
8/9	Superior M2
9/9	Superior M3

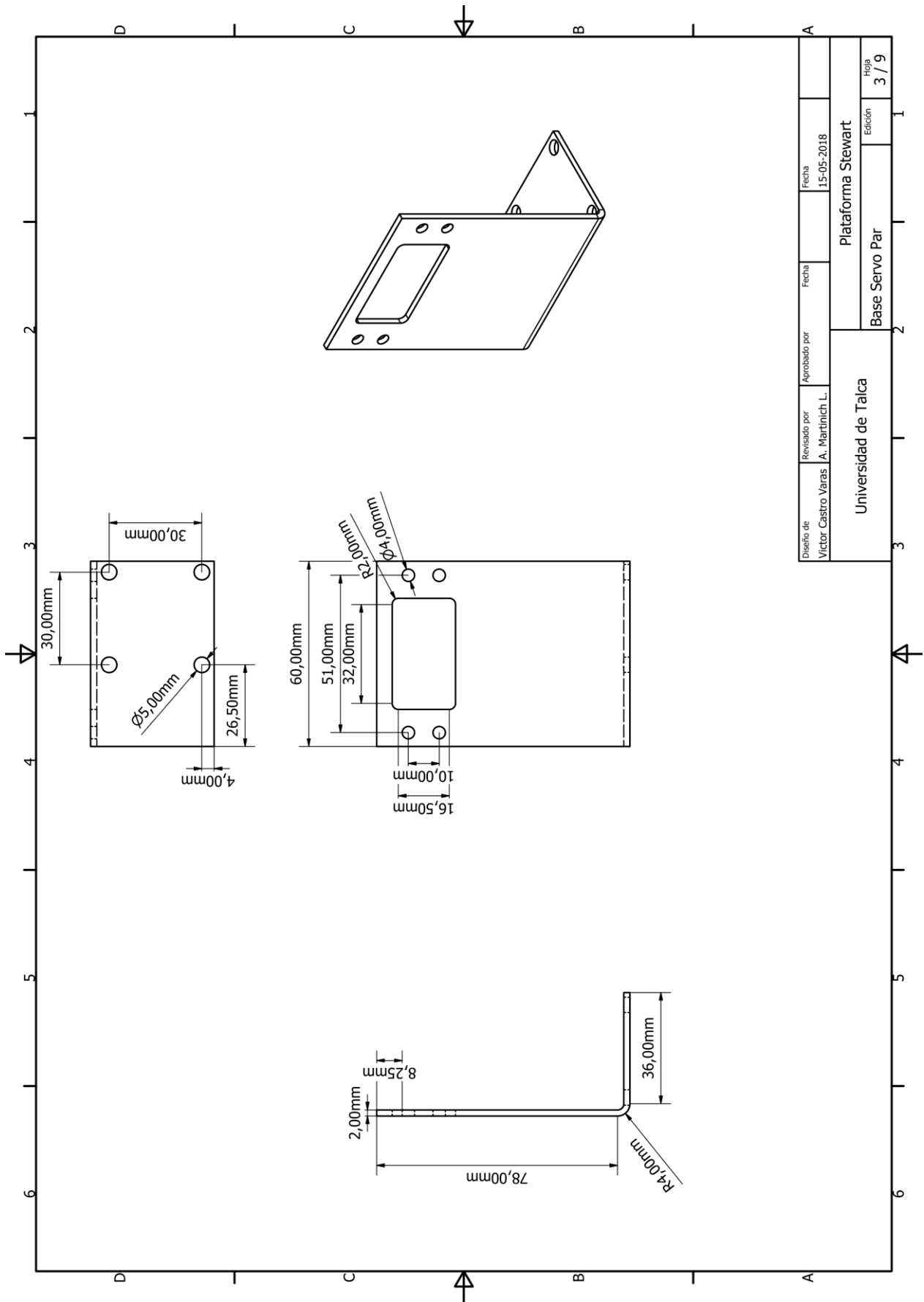




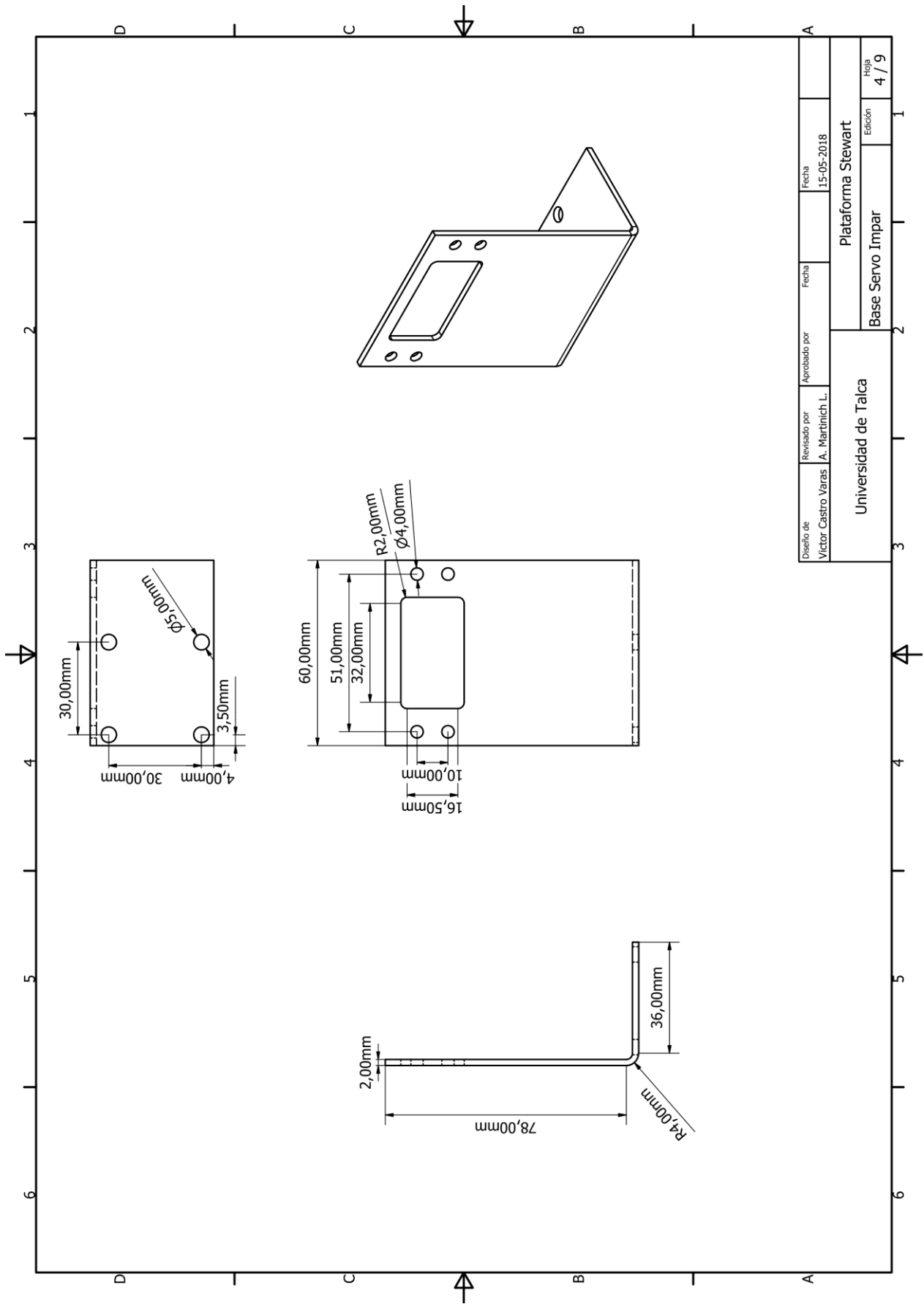
Diseño de Victor Castro Varas	Revisado por A. Martinich L.	Aprobado por	Fecha 15-05-2018	Fecha	Fecha
Universidad de Talca			Plataforma Stewart		
			Plataforma Stewart		
			Edición 1 / 9		
			Hoja 1 / 9		



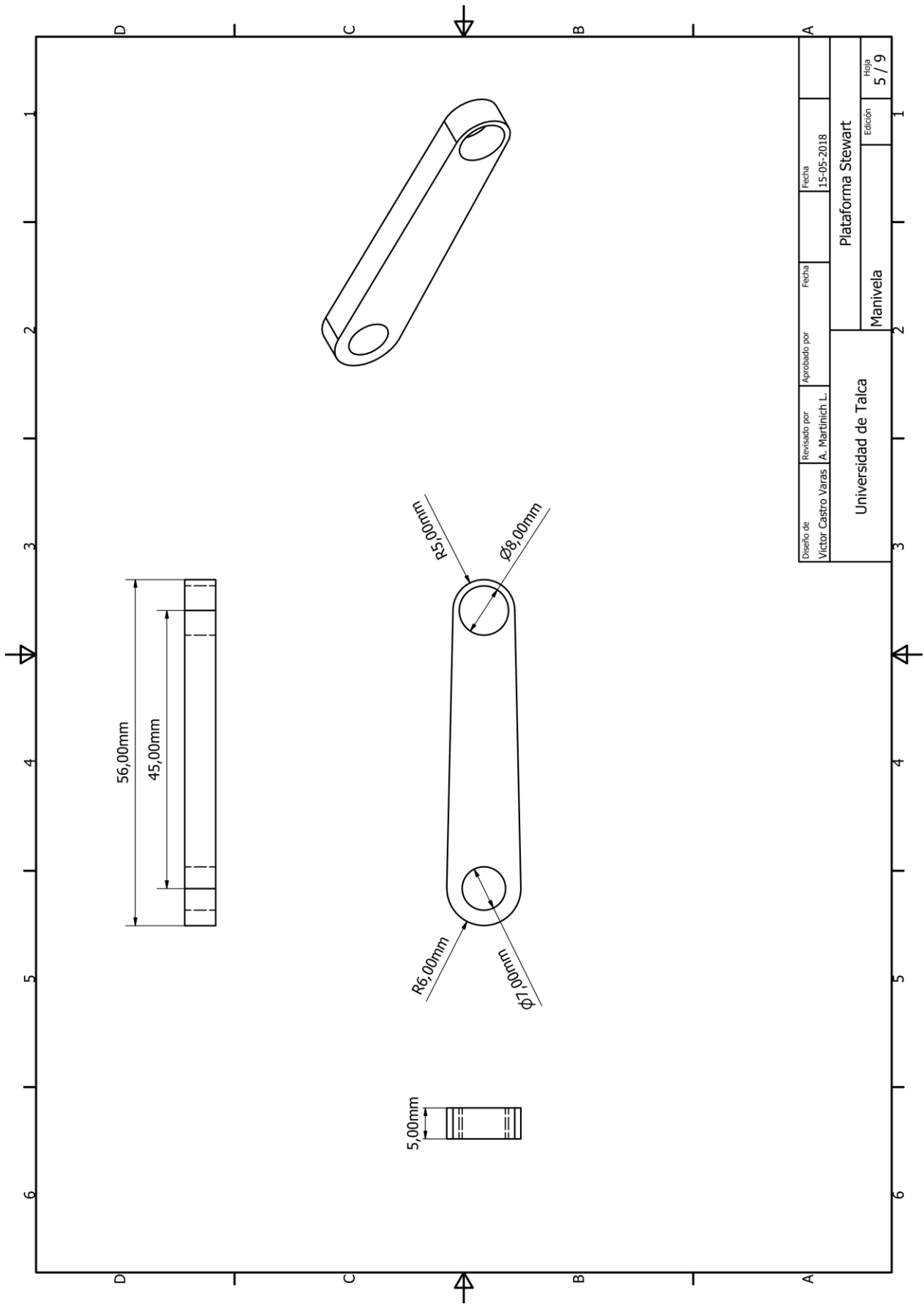
Diseño de Victor Castro Varas	Revisado por A. Martinić L.	Aprobado por	Fecha 15-05-2018	Fecha
Universidad de Talca			Plataforma Stewart	
			Plataforma Base	
				Hoja 2 / 9



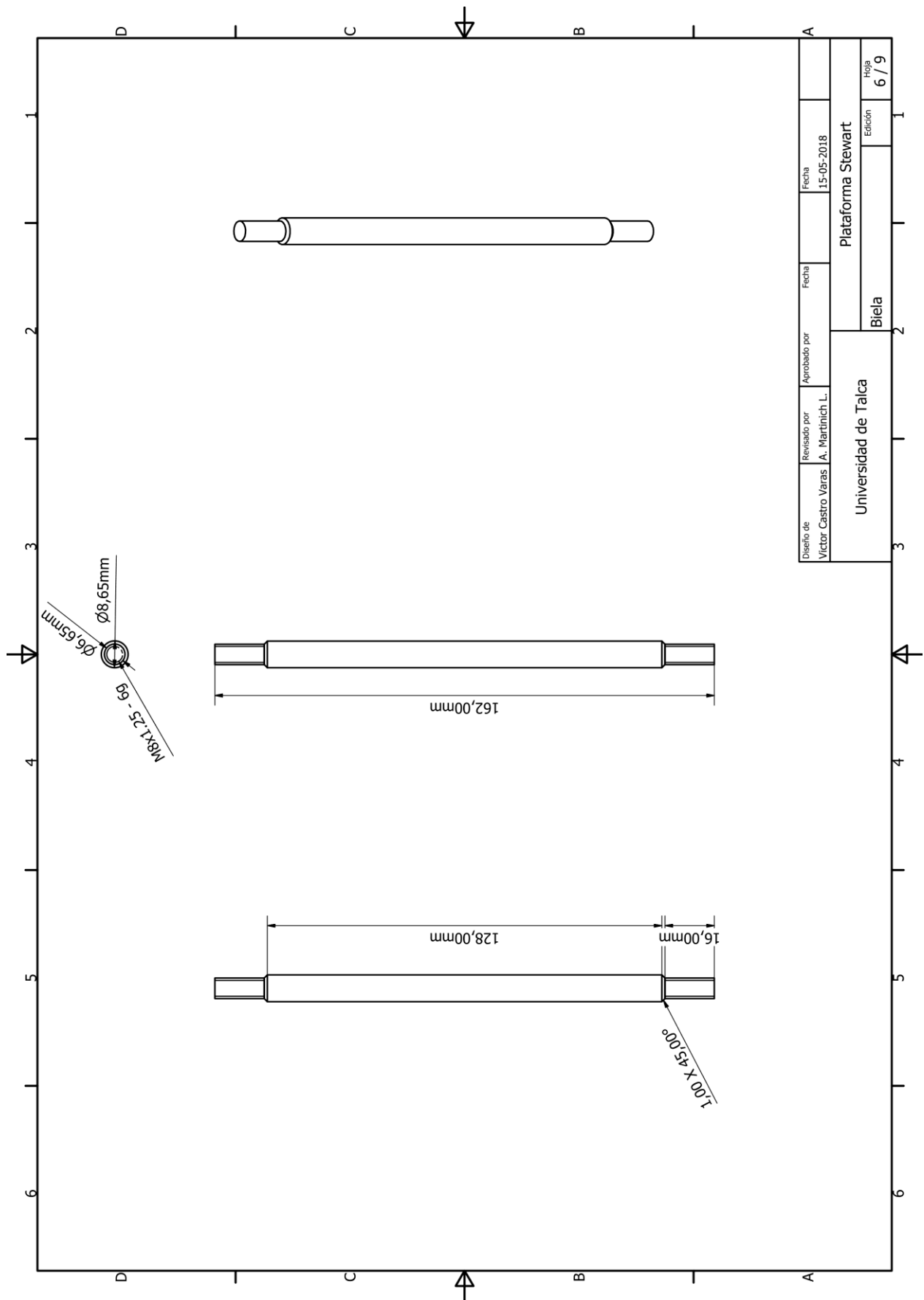
Diseño de	Revisado por	Aprobado por	Fecha
Víctor Castro Varas	A. Martínich L.		15-05-2018
Universidad de Talca		Plataforma Stewart	
		Base Servo Par	Edición
			3 / 9



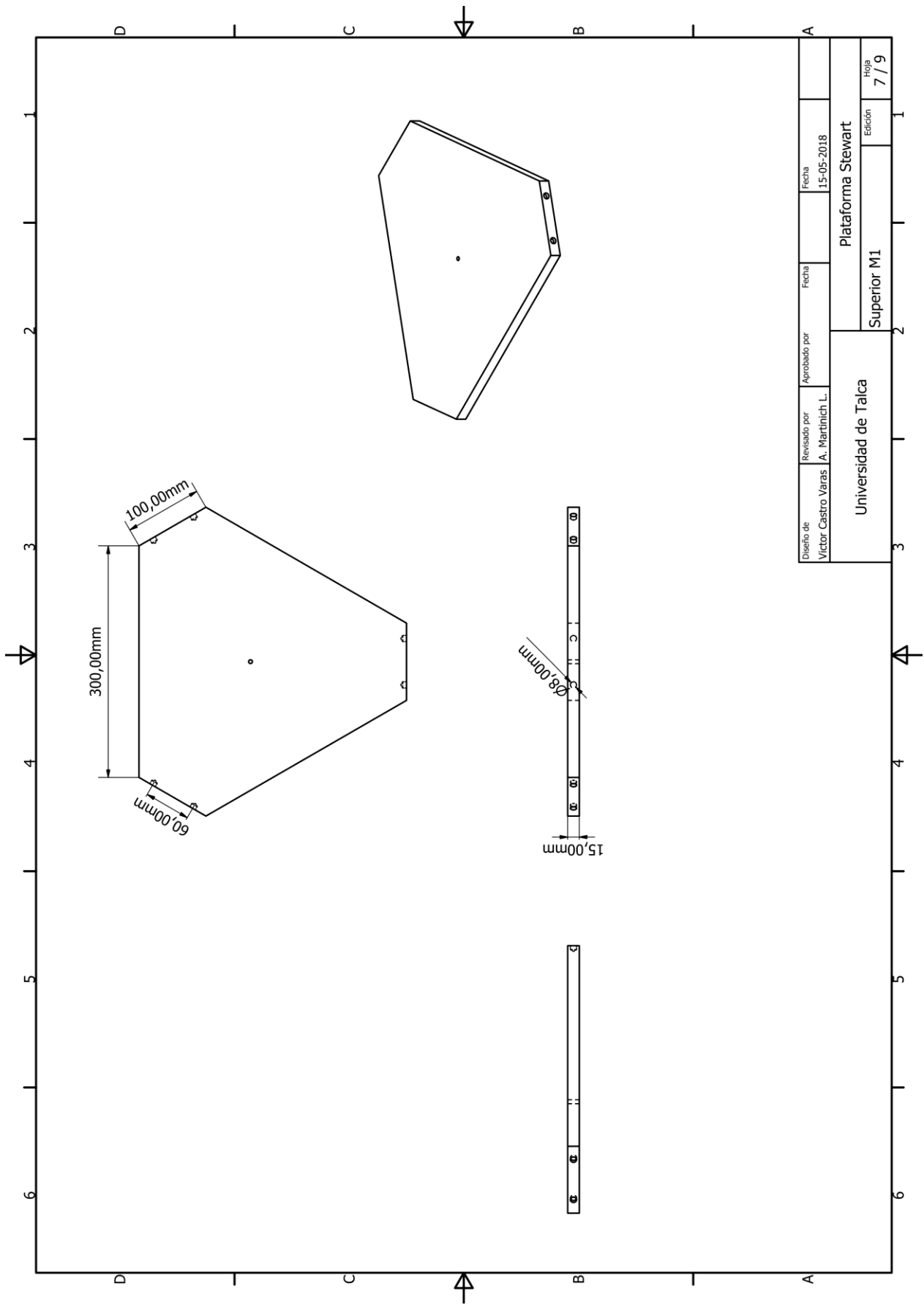
Diseño de Victor Castro Varas	Revisado por A. Martiniich L.	Aprobado por	Fecha 15-05-2018	Fecha
Universidad de Talca			Plataforma Stewart	
			Base Servo Impar	
			Edición 4 / 9	



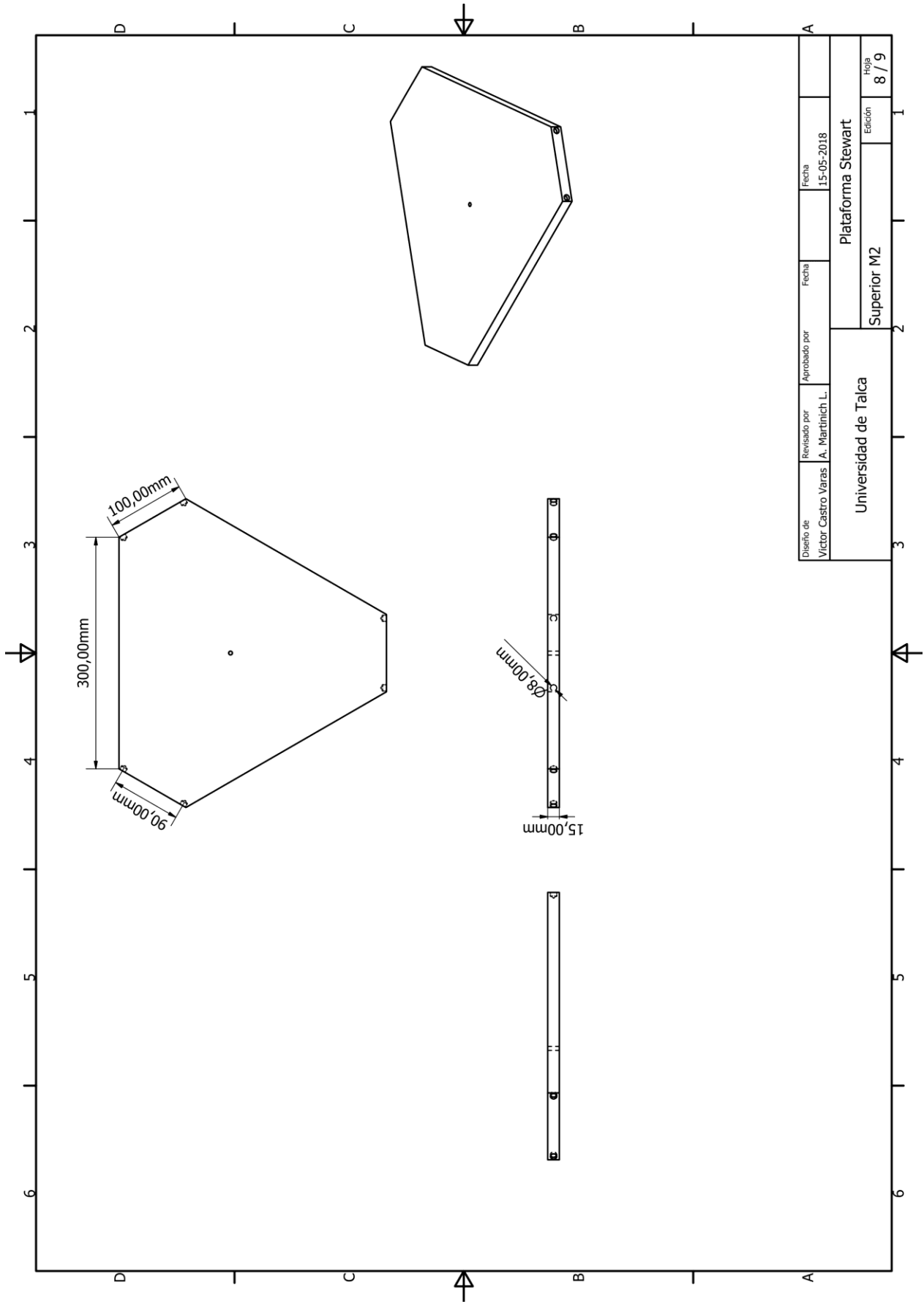
Diseño de Victor Castro Varas	Revisado por A. Martinich L.	Aprobado por	Fecha 15-05-2018	Fecha	Fecha
Universidad de Talca			Plataforma Stewart		
			Manivela		
			Edición		
			Hoja 5 / 9		

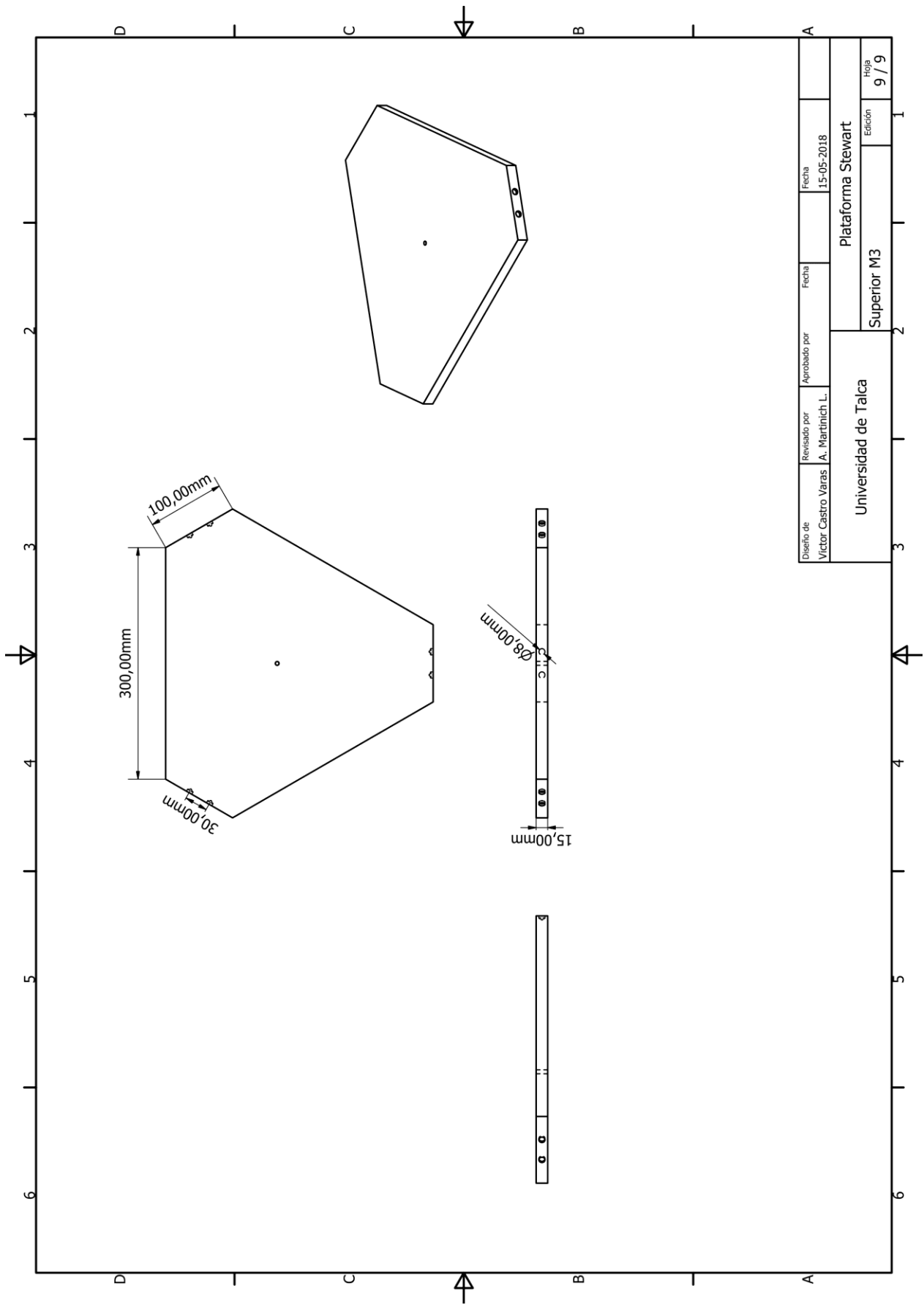


Diseño de Victor Castro Varas	Revisado por A. Martinić L.	Aprobado por	Fecha 15-05-2018	Fecha	Fecha
Universidad de Talca			Plataforma Stewart		
			Biela		
			Edición		
			Hoja 6 / 9		



Diseño de Victor Castro Varas	Revisado por A. Martnlich L.	Aprobado por	Fecha 15-05-2018
Universidad de Talca		Superior M1	Edición 7 / 9
Plataforma Stewart			





Diseño de Victor Castro Varas	Revisado por A. Martinich L.	Aprobado por	Fecha 15-05-2018
Universidad de Talca		Superior M3	Plataforma Stewart
			Hoja 9 / 9