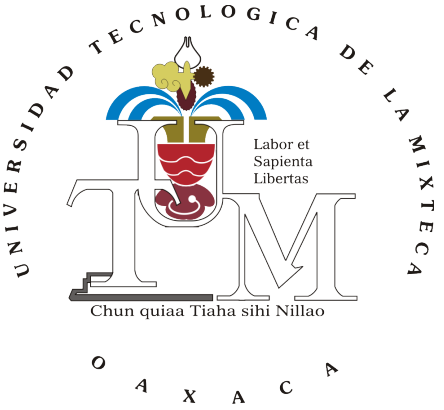


UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

CONSTRUCCIÓN Y ANÁLISIS CINEMÁTICO DE UN PROTOTIPO DE ROBOT PARALELO UPUR DE SEIS GRADOS DE LIBERTAD

TESIS:
PARA OBTENER EL TÍTULO DE
INGENIERO EN MECATRÓNICA

PRESENTA:

HUGO JAVIER CORTES RUIZ

DIRECTOR DE TESIS:

DR. MANUEL ARIAS MONTIEL

CO-DIRECTOR DE TESIS:

DR. RICARDO TAPIA HERRERA

HUAJUAPAN DE LEÓN, OAXACA, MÉXICO
MARZO DE 2019

Dedicatoria

A mis padres Guadalupe Ruiz y Erasto Cortes, con todo el cariño y amor, por mantenerse a mi lado y alcanzar juntos la realización de este gran logro.

A mi hermano Gustavo, por acompañarme en todos lo momentos durante el transcurso de la carrera, mi más profundo agradecimiento para él.

A todos mis amigos, con admiración y cariño por acompañarme en momentos de estrés y dificultades.

Agradecimientos

- Agradezco a toda mi familia por alentarme, apoyarme y mantenerse siempre junto a mí, ayudándome a terminar la carrera.
- Agradezco a mis directores de tesis, al Dr. Manuel Arias Montiel y al Dr. Ricardo Tapia Herrera por hacer posible la terminación de este proyecto de tesis y por toda la experiencia compartida.
- Agradezco a todos mis compañeros y amigos que durante mi estadía en la universidad se convirtieron en una nueva familia, forman parte importante del desarrollo de este proyecto.
- Finalmente, agradezco a la Universidad Tecnológica de la Mixteca por brindarme la oportunidad de estudiar la ingeniería en tan bellas instalaciones, bajo el asesoramiento de grandes profesores en el mejor de los ambientes.

Muchas gracias a todos.

Índice

Capítulo 1

Introducción

Los manipuladores robóticos en general comienzan a remplazar significativamente al hombre en tareas que resultan sumamente agotadoras o demasiado repetitivas y de alto riesgo para él. Esto mismo ha implicado un gran desarrollo en el tema de los manipuladores y en general en la robótica, teniendo increíbles progresos con máquinas que alcanzan grados de precisión nanométrica en la actualidad. Sin embargo, en la mayoría de las industrias un gran número de los robots que se encargan de realizar los distintos procesos de ensamble y manufactura son del tipo serial, es decir, son brazos antropomórficos o de alguna configuración parecida. Una de las principales desventajas de este tipo de robots es que la rigidez y exactitud que poseen se ven ampliamente afectadas por las fuerzas inerciales que actúan en ellos debido al gran tamaño y peso que poseen. Por lo que actualmente, una alternativa a este tipo de robots son los manipuladores paralelos, que al estar conformados por varias cadenas cinemáticas simples, son mucho más rígidos y pueden soportar cargas mucho mayores con un tamaño menor en comparación a los robots seriales, permitiendo además alcanzar exactitudes mucho mayores y velocidades realmente altas como es el caso de los robots DELTA encargados de paletizar y montar componentes.

Es por esto que en el presente proyecto de tesis se propone realizar el diseño y la construcción de un robot paralelo de seis grados de libertad con configuración U$\underline{\text{P}}$UR (que denota el tipo de juntas a utilizar en el robot: universal, prismática, universal y revoluta). Además, se desarrollará el modelo cinemático empleando la teoría de tornillos y se validará mediante simulaciones numéricas y mediciones experimentales. Para esto, será necesario implementar un sistema de instrumentación y de adquisición de datos que permita la lectura de las variables de posición y de orientación de la plataforma móvil (efector final) del robot.

1.1. Antecedentes

1.1.1. Plataforma de Stewart-Gough

La primera gran aportación a los robots paralelos surge del más celebre de los mecanismos paralelos de seis grados de libertad, la plataforma de pruebas universal para neumáticos propuesta por el Dr. Eric Gough en 1947. La idea de construcción del mecanismo paralelo surge de la necesidad de un dispositivo que pudiera deformar y aplicar fuerzas a los neumáticos en todas las direcciones y orientaciones dentro de un espacio de trabajo determinado y acotado, por lo que para resolver dicho problema, Eric Gough diseña y construye (1947) un mecanismo paralelo de seis grados de libertad compuesto de dos plataformas: una fija anclada al piso y la otra móvil conectada a la base a partir de una serie de actuadores de longitud variable cuyos extremos lo sujetan juntas universales en la parte inferior y juntas de bola en la parte superior (ver Fig. 1.1). Estableciendo de esta manera los principios de las estructuras de cadenas cinemáticas cerradas [1, 2, 3].

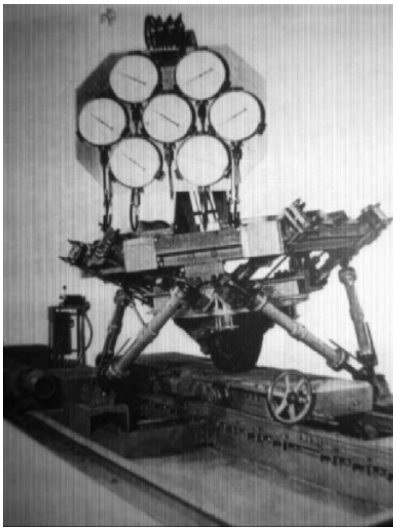
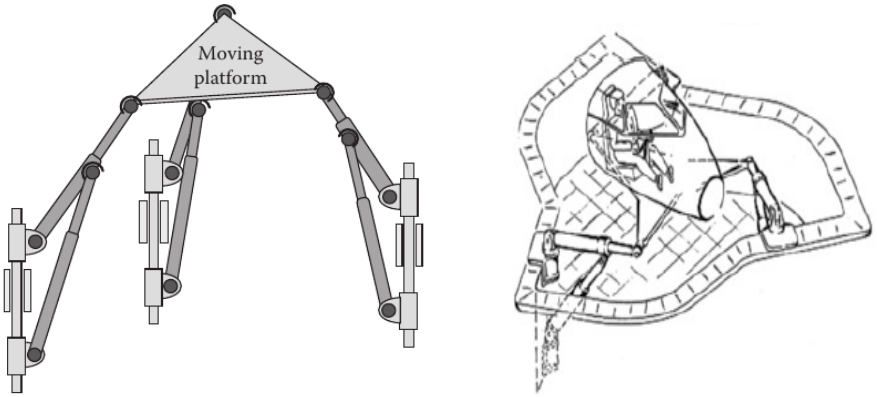


Figura 1.1: Plataforma diseñada por Gough en 1947 [2].

Las ventajas de la plataforma de Gough se hicieron notar de inmediato, la gran rigidez que poseía la estructura, además de la carga que podía ejercer sobre los neumáticos en comparación al peso de los elementos que la componían, era sumamente mayor en comparación a su contraparte serial. La máquina para probar neumáticos se terminó de construir en 1953 y se mantuvo funcional hasta el año 2000 en que se retiró de operación [2].
Posterior al diseño propuesto por Gough, en 1965 D. Stewart publicaba un artículo en el que presentaba un mecanismo paralelo cuya función sería la de un simulador de vuelo con lo que daba solución al problema propuesto por la industria aeronáutica desde 1960.

A diferencia de la plataforma de Gough, la propuesta por Stewart consta de solo tres extremidades con dos actuadores lineales en cada una como se puede observar en la Fig. 1.2 [1, 2, 3, 4].



(a) Diagrama simplificado del mecanismo.

(b) Diseño conceptual del simulador de vuelo.

Figura 1.2: Plataforma de Stewart para el simulador de vuelo [3].

Existe mucha confusión entre ambas plataformas presentadas anteriormente, en muchos de los casos la plataforma de Gough es referida como plataforma de Stewart, esto debido a que fue justamente Stewart quien introdujo el prototipo de Gough a la investigación académica, pues hasta antes de eso la plataforma era bastante conocida en la industria pero pasaba desapercibida en el ámbito académico. Para evitar confusiones ambos mecanismos son referidos como "Plataforma de Stewart-Gough" dando crédito a ambos autores.

1.1.2. Simulador de vuelo de Cappel

Un par de años antes que Stewart, en 1962 Klaus Cappel ingeniero del Instituto Franklin en Philadelphia diseñó un mecanismo similar al propuesto por Gough cuyo propósito es el de funcionar como un simulador de cabina de helicóptero, utilizando el mismo octágono con seis extremidades es capaz de simular los movimientos presentes durante el vuelo (ver Fig. 1.3) [1].

Cada uno de los autores antes mencionados realizaron grandes aportes a la robótica a partir de mecanismos paralelos, estableciendo los principios en cuanto a síntesis de mecanismos, análisis cinemático, geométrico y dimensional. Por ello, Stewart, Gough y Cappel son considerados los principales pioneros en esta área.

Figura 1.3: Simulador de vuelo de Helicóptero diseñado por Cappel en 1962 [1].

1.1.3. Robots DELTA

Los robots DELTA son un tipo de robot paralelo que consiste de tres extremidades conectadas a la base, los robots de este tipo son ampliamente utilizados en actividades de colocación y montaje de objetos a altas velocidades. Debido a la configuración de paralelogramos utilizada en los brazos o extremidades del robot, es que se asegura que la orientación del efector final no se vea modificada. Esta clase de robot fue diseñado por Reymond Clavel a principios de 1980 como parte de su tesis doctoral. Dada la rapidez del robot es que se expande con gran velocidad dentro de la industria, siendo así que en 1999 ABB presenta al mercado su propio robot delta, el "Flexpicker", mostrado en la Fig. 1.4 [3].

Figura 1.4: Robot DELTA Flexpicker de ABB [5].

1.1.4. Robots paralelos accionados por cables

Una de las desventajas de los mecanismos paralelos resulta ser el pequeño espacio de trabajo en el que se pueden mover, para resolver este inconveniente se sustituyen los actuadores lineales por cables y un controlador de longitud del mismo, de modo que las extremidades del robot serían los cables atados a él. Con esto, el espacio de trabajo se incrementa de manera significativa, sin embargo se presentan algunas complicaciones, como por ejemplo establecer un mecanismo que asegure que las cuerdas o cables siempre estarán sometidas a tensión dado que no pueden actuar en compresión, además de los puntos de singularidad que siempre existen en dichos mecanismos [3].
Ejemplo de este tipo de robot son las cámaras en los estadios que pueden moverse a partir de cables por todo el estadio, las llamadas Skycam o Spidercam como la que se muestra en la Fig. 1.5.

Figura 1.5: Ejemplo de robot paralelo accionado por cables (SPIDERCAM). [6]

1.1.5. Aplicaciones actuales de los robots paralelos

Actualmente los robots paralelos desempeñan un gran número de aplicaciones, por ejemplo dentro de la industria aeroespacial, se han utilizado robots paralelos para corregir errores en la alineación de espejos de telescopios, se desarrollan prototipos de estructuras reconfigurables utilizando mecanismos paralelos, además se utilizan en telescopios para orientar algunos de sus elementos y utilizar estructuras cada vez más ligeras.
En el campo de la vibraciones mecánicas se ha propuesto el uso de robots paralelos para el control y análisis de vibraciones, sin embargo aún no son aceptados del todo en la industria

a diferencia de los absorbedores que son ampliamente utilizados. De un modo más lento, los robots paralelos comienzan a incluirse como parte de equipo médico, son utilizados para realizar operaciones de suma precisión en cirugías ortopédicas, además de utilizarse en camillas de radioterapia para orientar y dirigir la radiación de modo más preciso en los pacientes. Finalmente, dentro de la industria de manufactura son ampliamente utilizados en máquinas herramientas de control numérico, para ensamble de productos y para el montaje de componentes, siendo los principales representantes los robots DELTA con la gran velocidad que pueden alcanzar [7].

1.2. Planteamiento del problema

Los mecanismos y manipuladores paralelos han sido parte del campo de la robótica por décadas. Con el paso del tiempo han comenzado a alcanzar cierto grado de madurez y una gran cantidad de arquitecturas son utilizadas en un sin fin de aplicaciones, a tal grado que empiezan a formar parte de un número bastante considerable de actividades industriales en la actualidad, las ventajas que ofrecen respecto de los robots seriales resultan benéficas para las empresas, la gran velocidad de respuesta, la fuerza que pueden ejercer o soportar sin necesidad de fuentes de potencia excesivamente grandes, son algunos de los factores que los colocan como favoritos dentro de las industrias, pudiendo alcanzar incluso mucho más potencial con el paso de los años [8, 9].

Sin embargo, dada su configuración y las múltiples cadenas cinemáticas que los conforman, los robots paralelos tienden a presentar un análisis cinemático más complicado que su contraparte serial en ciertos aspectos, por ejemplo para los robots seriales resulta sencillo obtener las relaciones de cinemática directa mientras que la solución de la cinemática inversa se torna un tanto complicada, lo contrario ocurre con los robots paralelos pues la solución de la cinemática inversa es simple, mientras que la cinemática directa es sumamente complicada, por lo que se han empleado distintos tipos de análisis tratando de simplificar dichas operaciones, una de las herramientas matemáticas que se ha introducido recientemente en el campo del análisis de robots paralelos es la *teoría de tornillos*, con la que se pretende representar el movimiento de los cuerpos rígidos a partir de un giro y un par alrededor de algún eje determinado.

Con base en lo anterior, se propone como parte del proyecto de tesis realizar el diseño y construcción de un prototipo de robot paralelo de seis grados de libertad, a partir del cual se podrá poner en práctica la *teoría de tornillos* para la resolución de la cinemática del mecanismo, de modo que se compare la versatilidad de dicha teoría con respecto al método tradicional.

También se plantea la validación de los modelos cinemáticos obtenidos mediante simulaciones numéricas y datos experimentales, para lo cual será necesario desarrollar el sistema de instrumentación y adquisición de datos que permita la medición de las variables cinemáticas de la plataforma móvil del robot. Además, el desarrollo y construcción del mecanismo permitirá un primer acercamiento a robots de este tipo de clase industrial permitiendo extender los conocimientos en cuanto a la movilidad y conformación de mecanismos de este tipo.

1.3. Justificación

Como se mencionó anteriormente, los robots paralelos actualmente comienzan a ganar terreno en la industria, debido a la precisión y velocidad que pueden alcanzar, además de ventajas en cuanto a relación peso-capacidad de carga y el fácil escalamiento de estas máquinas. Sin embargo en los ámbitos de la investigación y la academia resulta complicado adquirir robots paralelos por su elevado costo, con los que los estudiantes se familiaricen y practiquen la manipulación, programación y desarrollo de tareas con robots de este tipo.

Atendiendo este problema, se plantea la construcción de un robot paralelo de seis grados de libertad como parte fundamental del presente proyecto de tesis. De modo que el prototipo sea totalmente funcional y permita realizar todas las actividades anteriores, además de actividades posteriores como: probar distintas técnicas de control en el dispositivo e incluso la implementación de seguimiento de trayectorias y la aplicación de fuerza constante sobre algún objeto en la plataforma, por mencionar algunos ejemplos.

De la literatura consultada se puede observar que la solución de manera tradicional (a partir del álgebra vectorial convencional) de la cinemática de los robots paralelos resulta sumamente complicada debido a las no linealidades y a la cantidad de variables involucradas en las ecuaciones que describen el movimiento del robot, por lo que se pretende introducir el uso de la *teoría de tornillos* para simplificar dicho proceso. Además, se propone la validación de los modelos cinemáticos obtenidos mediante datos experimentales, tema que pocas veces se h abordado en trabajos similares y que resulta de suma importancia para el desarrollo posterior de modelos dinámicos y esquemas de control.

1.4. Objetivos

1.4.1. Objetivo general

Construir un robot paralelo con configuración U$\underline{P}$UR de seis grados de libertad para obtener y validar la cinemática directa e inversa del mismo aplicando *teoría de tornillos* y la solución a partir del álgebra vectorial convencional comparando ventajas y desventajas de ambos métodos.

1.4.2. Objetivos específicos

- Obtener el modelo CAD del robot para la simulación de la movilidad del mismo.
- Analizar la cinemática del prototipo de robot propuesto.
- Construir un prototipo del robot diseñado.
- Realizar el diseño de la etapa de potencia del robot e instrumentación del mismo para obtener variables de posición y orientación de la plataforma.
- Obtener y simular las soluciones de la cinemática directa e inversa con y sin la aplicación de la *teoría de tornillos*.
- Validar de manera experimental todos los resultados anteriores a través del prototipo construido.

1.5. Metodología

Para el desarrollo del proyecto de tesis se plantea utilizar una metodología de diseño iterativo bastante simple denominado Ciclo de Vida de Desarrollo de Sistemas (SDLC, System Development Life Cycle) [10, 11]. La metodología es ampliamente utilizada para el desarrollo de software, sin embargo no está específicamente diseñada para esto, pues abarca cualquier clase de proyectos. La metodología consta básicamente de seis etapas:

1. *Planificación*: Es una de las más importantes etapas del proceso de diseño, de esta etapa depende la factibilidad y buen desarrollo del proyecto. Consiste básicamente en definir qué es exactamente lo que se desea hacer, el problema que se resuelve con la implementación del proyecto, además de los objetivos que se alcanzarán.

2. *Análisis*: En esta etapa se definen los requerimientos con los que debe cumplir el producto o proyecto, además de analizar el funcionamiento esperado del producto.

3. *Diseño*: Esta etapa se encarga de proveer del diseño y la solución al problema a resolver. En otras palabras, se definen componentes, módulos o grupos de elementos que cumplen con cierta función dentro del producto, la forma de interacción con el usuario y la forma o modo de interacción de cada módulo.

4. *Construcción e implementación*: Esta etapa consiste de realizar el maquinado, en general la construcción del prototipo de acuerdo a los planos obtenidos en la etapa de diseño, además de comenzar con pruebas preliminares de solución al problema.

5. *Prueba e integración*: Consiste de poner a prueba las soluciones obtenidas al problema, bajo ciertas condiciones iniciales y observar si efectivamente la solución propuesta cumple con los requerimientos del sistema. Del mismo modo, se integran todos los distintos módulos al sistema.

6. *Mantenimiento*: En esta etapa se revisa que el prototipo a pesar de condiciones no establecidas se mantenga en operación de manera óptima, de no ser así tratar de reparar los errores.

Se puede observar que la metodología se adapta perfectamente al prototipo que se planea implementar, permitiendo probar distintas opciones de solución e iterar entre cada una de las etapas de modo que con cada falta de algún requerimiento se mejore ya sea desde la planeación o sobre la misma etapa en la que se encuentre el proceso, reduciendo tiempo y en consecuencia costos para el desarrollo del proyecto.

1.6. Estructura de la tesis

Este trabajo de tesis consta de siete capítulos en los cuales se describe el proceso de construcción, análisis y solución de la cinemática del mecanismo, así como el desarrollo del sistema de instrumentación del prototipo. En el Capítulo 1 se presentan una introducción al tema abordado, los antecedentes del tema, así como el planteamiento del problema, la justificación del mismo y los objetivos planteados para el desarrollo de la tesis. En el Capítulo 2 se presenta el marco teórico del documento, con las bases y conceptos fundamentales para el desarrollo de la tesis. En el Capítulo 3 se presentan los elementos que conforman el robot, así como la configuración de las extremidades, finalmente se muestra el ensamble y la manufactura del robot. El Capítulo 4 muestra el desarrollo del análisis cinemático directo e inverso de primer orden del robot. Mientras que lo que se refiere a la generación de señales, configuración de los controladores para los motores y demás cuestiones de instrumentación se abordan en el Capítulo 5. Los resultados analíticos, de simulación y experimentales se muestran en el Capítulo 6. Finalmente el Capitulo 7 presenta las conclusiones obtenidos con la realización de la tesis y los trabajos futuros que pueden desarrollarse. Además se anexan a modo de apéndices los programas implementados en MATLAB y Arduino®, así como dibujos técnicos de las piezas manufacturadas.

Capítulo 2

Marco teórico

2.1. Manipuladores paralelos

Los manipuladores paralelos son mecanismos que se encuentran conectados a la referencia o marco fijo (también denominado "Tierra") y la plataforma móvil al mismo tiempo (ver Fig. 2.1) [12]. Cuando los eslabones comparten solo un extremo, la cadena es abierta, por tanto se trata de una configuración serial, mientras que al compartir sus dos extremos o ser la cadena cerrada se considera una configuración paralela.
Una segunda definición de Manipulador Paralelo es la siguiente: *Un manipulador o mecanismo paralelo es un mecanismo con múltiples grados de libertad compuesto de una plataforma móvil y una base conectados entre sí al menos por dos cadenas cinemáticas en paralelo [13].* Dichas cadenas cinemáticas son denominadas extremidades, brazos o piernas y además, por lo regular, los actuadores se encuentran ubicados cerca de la base del robot, como se aprecia en la Fig. 2.1. Algunas de las características que distinguen a los manipuladores paralelos son los siguientes: [2, 14]

- Gran rigidez del mecanismo.
- Alta capacidad de carga.
- Gran precisión.
- Gran repetibilidad.
- Velocidad y aceleración más estables en el efector final.

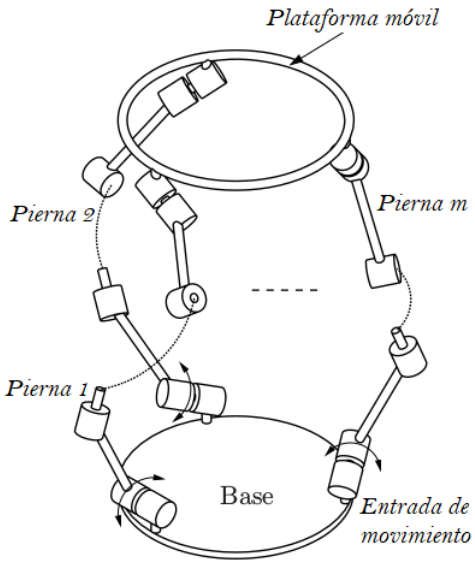


Figura 2.1: Ejemplo de manipulador paralelo [13].

Los manipuladores paralelos pueden clasificarse dentro de dos categorías principales [12]:

- **Manipulador Planar:** El que un manipulador sea planar significa que sus movimientos se limitan al plano sobre el que se encuentra la plataforma móvil, es decir, pueden moverse por ejemplo sobre sobre el plano XY y rotar sobre el eje Z.
- **Manipulador Espacial:** Los manipuladores espaciales poseen los seis grados de libertad de un cuerpo rígido, traslación a lo largo de los ejes X, Y y Z y rotación sobre cada uno de ellos.

2.2. Cadena cinemática

Las cadenas cinemáticas son pares cinemáticos o juntas unidas a eslabones para conformar uno o varios lazos. Existen dos tipos principales:

Abierta: También denominadas cadenas cinemáticas simples, son aquellas en las que los elementos miembros de cada una de las cadenas poseen a lo más dos grados de conexión, de lo que se puede concluir que efectivamente los manipuladores seriales cumplen con esta característica. Cabe aclarar que los grados de conexión se refieren al número de elementos atados o sujetos al eslabón por algún tipo de junta [2].

Cerrada: Las cadenas cinemáticas cerradas son aquellas en las que al menos un elemento que conforma la cadena consta de más de tres grados de conexión, como es el caso de la base móvil mostrada en la Fig. 2.2 que consta de seis puntos de conexión.

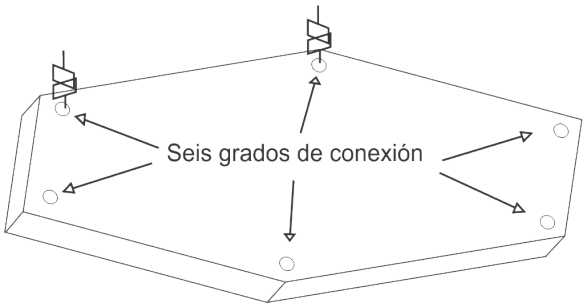


Figura 2.2: Base móvil ejemplificando los seis grados de conexión.

2.3. Representación de las extremidades (piernas) utilizadas en un robot paralelo.

Como se mencionó anteriormente, los mecanismos paralelos están compuestos de al menos dos cadenas cinemáticas simples conectadas en paralelo, mismas que proporcionan la movilidad a la plataforma superior y por ende al mecanismo en general. Por tanto, resulta fundamental establecer la notación y representación de cada una de las distintas estructuras y configuraciones de piernas.

2.3.1. Tipos de juntas en el mecanismo.

Para poder establecer y permitir movimiento relativo entre los eslabones que componen las piernas del robot, es necesario colocar juntas que impongan restricciones al mecanismo de modo que su movimiento no sea incontrolable, ya sea espacial o planar los tipos de junta más usadas en mecanismos y robots se muestran en la Fig. 2.3.

Como se puede observar, se tienen siete tipos de juntas diferentes, sin embargo existen dos principales de las que se derivan las demás, la junta revoluta (R) y la junta prismática (P). Por ejemplo, para la junta universal basta con colocar dos juntas revolutas cuyos ejes de rotación sean perpendiculares, mientras que para la junta esférica se puede obtener de tres juntas revolutas cuyos ejes de rotación sean ortogonales entre ellos. Además, la notación para referirse a cada una de las juntas se puede observar en la Tabla 2.1.

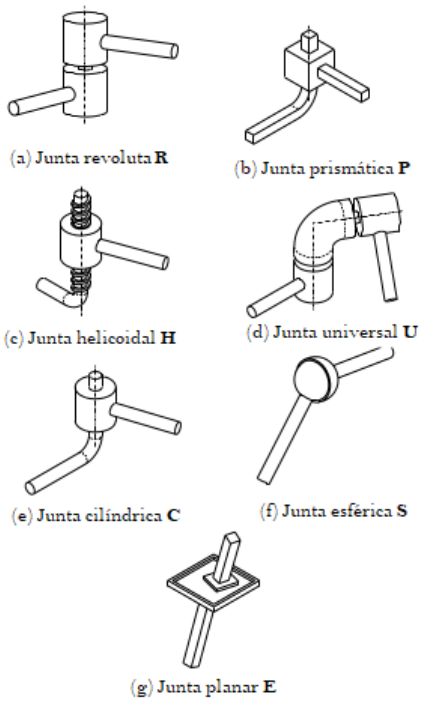


Figura 2.3: Tipos de juntas utilizadas en los mecanismos.[13]

2.3.2. Notación para las extremidades

Las piernas o extremidades del robot son representadas a partir de las juntas que componen cada una de las cadenas cinemáticas del mismo, comenzando desde la base fija hasta la plataforma móvil. Por ejemplo, considérese la Fig. 2.4 en la que se observa un mecanismo paralelo con tres piernas, cada una compuesta de cinco juntas revolutas, de las cuales algunas de sus ejes de rotación son paralelos. Por lo que las extremidades del mecanismo se pueden describir como: 3-$\acute{R}\acute{R}\grave{R}\grave{R}\grave{R}$. Lo que significa que el mecanismo está compuesto de tres extremidades iguales, de las cuales las primeras dos juntas revolutas empezando desde la base hacia la plataforma móvil, tienen sus ejes de rotación paralelos entre sí (esto se especifica con las marcas sobre las letras, marcas iguales significan ejes paralelos), mientras que las tres restantes cumplen que sus ejes de rotación también son paralelos entre ellos pero no respecto de las dos primeras. En caso de que alguna junta esté actuada, es decir, que sea una entrada de movimiento se especifica subrayando la letra que representa dicha junta.

Tabla 2.1: Notación para las juntas en el mecanismo [13].

C	Junta Cilíndrica
H	Junta Helicoidal
P	Junta Prismática
R	Junta Revoluta
$\acute{R}$	Juntas R con ejes paralelos pertenecientes a una misma extremidad
$\grave{R}$	Juntas R con ejes paralelos pertenecientes a una misma extremidad
$\dot{R}$	Juntas R con ejes que se intersecan y pertenecen a una misma extremidad
$\hat{R}$	Juntas R con ejes concurrentes pertenecientes a un mismo grupo de extremidades
$\check{R}$	Juntas R con ejes concurrentes pertenecientes a un mismo grupo de extremidades
$\ddot{R}$	Juntas R con ejes coaxiales que pasan a través de las intersecciones de los ejes de juntas $\dot{R}$ si existe alguna dentro de un mismo grupo de extremidades
$\tilde{R}$	Juntas R con ejes paralelos, que cumplen ser paralelos a los ejes de las juntas $\ddot{R}$ o a una linea que pasa a través de al menos dos intersecciones de los ejes $\dot{R}$ si existe alguno en el mismo grupo de extremidades
$\bar{R}$	Juntas R con ejes paralelos pertenecientes a un mismo grupo de extremidades
S	Junta esférica
U	Junta Universal
$()_A$	Sucesión de juntas tales que están colocadas de modo que todos los ejes de las juntas R son paralelas y la dirección de las juntas P (si existe alguna) no es perpendicular a los ejes de las juntas R
$()_E$	Sucesión de juntas pertenecientes a una misma extremidad tales que están colocadas de modo que todos los eslabones se mueven sobre planos paralelos
$()_E^{\mid}$	Cada sucesión $()_E$ que pertenece al mismo grupo de extremidades que cumplen que los planos asociados al movimiento son paralelos a una misma linea
$()_E^{\mid\mid}$	Cada sucesión $()_E$ que pertenece al mismo grupo de extremidades que cumplen que los planos asociados al movimiento son paralelos
$()_L$	Sucesión de juntas tales que al menos una junta R es coaxial o almenos una junta P es codireccional
$()_S$	Sucesión de juntas que están colocadas de modo que todos los eslabones se mueven sobre superficies esféricas concurrentes
$()_T$	Sucesión de juntas P pertenecientes a una misma extremidad acomodadas de modo que todos los eslabones se mueven sobre planos paralelos

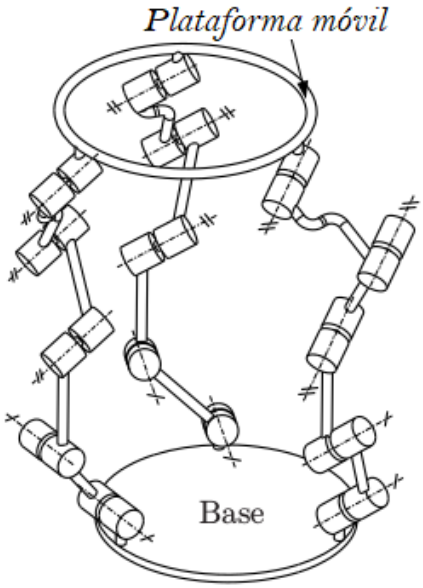


Figura 2.4: Notación para las extremidades en el mecanismo [13].

2.4. Espacio de trabajo

El espacio de trabajo de un manipulador ya sea paralelo o en serie, se define como el volumen total de puntos que puede alcanzar el robot [14]. El espacio de trabajo está restringido la mayoría de las veces por cuestiones del diseño mecánico de cada uno de los componentes que lo conforman, por ejemplo, los actuadores lineales en el caso del diseño que se plantea en este trabajo, tienen una carrera preestablecida que no se puede exceder, al igual que las juntas universales acotan su movimiento hasta un rango de grados. Existen dentro del espacio de trabajo dos subconjuntos.

Espacio de trabajo alcanzable: El espacio de trabajo alcanzable se refiere a todos los puntos que puede alcanzar el robot sin importar la orientación, pues muchos de ellos solo se satisfacen bajo ciertas condiciones de orientación [14].

Espacio de trabajo diestro: El espacio de trabajo diestro a diferencia del espacio de trabajo alcanzable contempla solo los puntos en los que el robot puede adoptar cualquier orientación (todo dentro del rango de movilidad del robot) [14].

2.5. Exactitud y repetibilidad

Dos parámetros de suma importancia dentro de los manipuladores ya sea seriales o paralelos, son la repetibilidad y la exactitud. La **repetibilidad** se refiere al error que presenta el robot al llegar al mismo punto desde posiciones diferentes o desde el mismo punto, es decir, con qué frecuencia repite la misma posición sin importar la pose o configuración de la que provenga. Se puede entender este concepto de manera gráfica como una esfera cuyo baricentro es el punto solicitado y el radio está dado por los errores promedio alrededor del punto, como se puede observar en la Fig. 2.5.
Mientras que la **exactitud** se refiere al error que existe entre la posición solicitada al robot y la que alcanza en el espacio de trabajo real, la mayoría de las veces estos pequeños errores se deben a error en los sistemas electrónicos como los encoders o debido a tolerancias mecánicas en los componentes del robot [15].

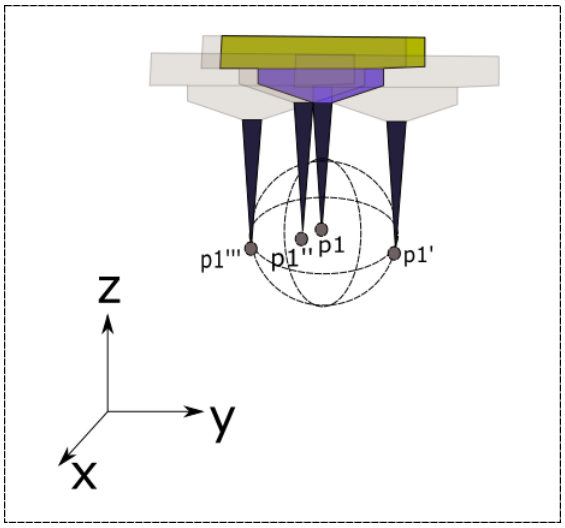


Figura 2.5: Representación gráfica de la repetibilidad.

2.6. Grados de libertad del robot o movilidad del mecanismo

Los grados de libertad del robot se refieren al número de entradas que son necesarias para definir completamente alguna configuración o pose del mecanismo, en el caso de los robots paralelos, la mayoría de las veces este número o cantidad de grados de libertad coincide con el número de actuadores.
Una segunda definición de la movilidad del mecanismo es: *La capacidad de un mecanismo de establecer un movimiento estable que depende de tres aspectos importantes, el primero de*

ellos la magnitud o cantidad de coordenadas independientes que se necesitan para definir una configuración del mecanismo, el segundo aspecto es el tipo o naturaleza del movimiento, esto se refiere a si se trata de un movimiento traslacional, rotacional, si se trata de un movimiento continuo o discreto, bajo ejes determinados o indeterminados, y finalmente el tercer aspecto es el tipo de movilidad, si este es instantáneo para cierta configuración solamente o de ciclo completo, cumpliendose para cualquier configuración [16].

Se pueden definir de manera general los grados de libertad de un mecanismo a partir de la siguiente interpretación, de acuerdo al caso que se esté analizando, un cuerpo rígido puede tener tres o seis grados de libertad dependiendo de si se trata de un movimiento planar o en el espacio, respectivamente. Además, todos los eslabones menos la base aportarán movimiento al mecanismo, así mismo las juntas con las que se conecten los eslabones entre sí les impondrán restricciones de movimiento. Por lo que se pueden obtener los grados de libertad de un mecanismo utilizando la siguiente fórmula (Fórmula de Kutzbach-Grübler) [3]:

$$F = \lambda(n-1) - \sum_{i=1}^{j} c_i \tag{2.1}$$

donde:

- F = grados de libertad del mecanismo.
- n = número de eslabones que conforman el mecanismo.
- λ = número de grados de libertad del espacio de trabajo (tres para mecanismo planares y seis para mecanismos espaciales).
- j = número de juntas en el mecanismo.
- c_i = grados de libertad restringidos por la junta i-ésima.

Sin embargo, la ecuación (2.1) es válida para mecanismos en los que no existe ningún tipo de grado de libertad pasivo. Dicha condición se refiere a un grado de libertad a partir del cual no se puede ejercer fuerza sobre esa dirección, se debe principalmente a grados de libertad redundantes sobre una misma dirección. Para aclarar esta idea, considérese un eslabón en cuyos extremos están conectadas dos juntas esféricas (ver Fig. 2.6), se puede notar que existe redundancia en la rotación a través del eje axial del eslabón, pues ambas juntas proporcionan el mismo grado de libertad, esto conduce a que este grado de libertad sea pasivo.

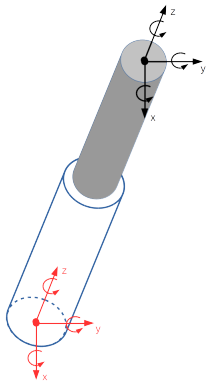


Figura 2.6: Ejemplo de grados de libertad pasivo.

Por tanto, teniendo eso en cuenta, se puede replantear la ecuación de Kutzbach-Grübler considerando ahora los grados de libertad pasivos. Se empieza por considerar la siguiente relación.

$$\lambda = c_i + f_i \tag{2.2}$$

donde:

- f_i = Grados de libertad permitidos por la junta.

Despejando c_i y sustituyendo en la ecuación (2.1), se obtiene la ecuación de Chebyshev-Kutzbach-Grübler y además, agregando el término relacionado a los grados de libertad pasivos [3].

$$F = \lambda(n - j - 1) + \sum_{i=1}^{j} f_i - f_p \tag{2.3}$$

donde:

- F = grados de libertad del mecanismo.
- n = número de eslabones en el mecanismo.
- j = juntas en el mecanismo.
- λ = grados de libertad del espacio de trabajo.
- f_i = número de grados de libertad permitidos por la junta i-ésima.
- f_p = número de grados de libertad pasivos.

2.7. Representación de la localización o pose del robot en el espacio

Para poder obtener y ubicar el robot en determinada posición en el espacio, se requiere definir la posición y orientación del robot con respecto a algún sistema de referencia. Se comienza por definir la posición, para ello se considera la Fig. 2.7, se empieza por definir un vector que ubique al cuerpo de interés con respecto de un sistema de coordenadas fijo o en movimiento relativo. Se define por ejemplo, un sistema de coordenadas fijo en el punto O_A y uno más colocado sobre el cuerpo en movimiento O_B. Además, se coloca un punto arbitrario P sobre la superficie del cuerpo [3, 17].

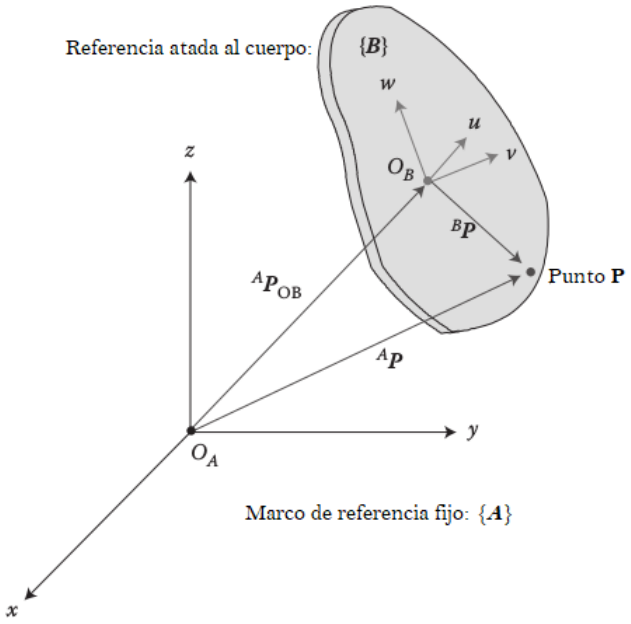


Figura 2.7: Posición de un cuerpo rígido en el espacio [3].

Por tanto, se pueden definir dos formas de ubicar el punto P del cuerpo rígido, el absoluto respecto del marco de referencia fijo O_A y el relativo al cuerpo mismo respecto del marco de referencia O_B. Por tanto, los vectores de posición que describen dichos puntos son ^AP y ^BP respectivamente. Donde el superíndice a la izquierda denota el marco de referencia desde el que es observado o medido el punto, definiendo ambos vectores como sigue:

$$^AP = \begin{Bmatrix} ^AP_x \\ ^AP_y \\ ^AP_z \end{Bmatrix} \quad ; \quad ^BP = \begin{Bmatrix} ^BP_x \\ ^BP_y \\ ^BP_z \end{Bmatrix} \tag{2.4}$$

De la ecuación (2.4) se observa que los vectores se definen a partir de los componentes X,Y,y Z del punto.
Dado que todos los puntos sobre el cuerpo poseen la misma orientación se puede definir dicha característica a partir de la orientación del sistema de referencia anclado al cuerpo O_B respecto del sistema de referencia fijo O_A, para esto se toma como referencia la Fig. 2.8 en la que se considera ahora que el cuerpo rígido se somete solamente a un cambio de orientación o rotación pura alrededor de sus ejes [3].

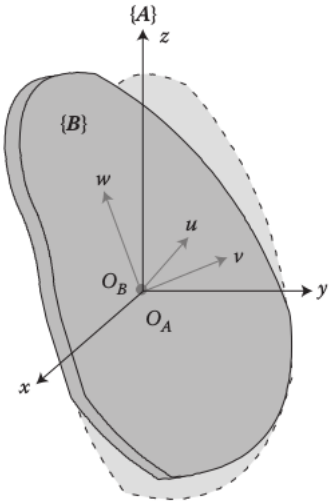


Figura 2.8: Cuerpo rígido sometido a rotación pura [3].

Se puede representar dicha rotación a partir de las proyecciones de los vectores unitarios del sistema de referencia O_B sobre el sistema de referencia O_A, obteniendo como resultado un total de nueve parámetros, definiendo así la matriz de rotación ${}^{A}R^{B}$ como se observa en la siguiente ecuación (los superíndices indican que es la matriz de rotación del marco de referencia B al A, es decir, a la derecha el marco de referencia origen y la izquierda el destino):

$$ {}^{A}R^{B} = [{}^{A}\hat{x}^{B}; {}^{A}\hat{y}^{B}; {}^{A}\hat{z}^{B}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.5} $$

$$ {}^{A}R^{B} = \begin{bmatrix} \hat{x}_B \cdot \hat{x}_A & \hat{y}_B \cdot \hat{x}_A & \hat{z}_B \cdot \hat{x}_A \\ \hat{x}_B \cdot \hat{y}_A & \hat{y}_B \cdot \hat{y}_A & \hat{z}_B \cdot \hat{y}_A \\ \hat{x}_B \cdot \hat{z}_A & \hat{y}_B \cdot \hat{z}_A & \hat{z}_B \cdot \hat{z}_A \end{bmatrix} \tag{2.6} $$

Como se observa en la ecuación (2.6), se obtienen cada uno de los componentes de la matriz de rotación a partir del producto punto de los vectores unitarios de cada uno de los sistemas de referencia, además es posible notar que al ser productos punto entre vectores unitarios el

resultado es el coseno del ángulo entre los vectores, por lo que la matriz también se denomina como de "cosenos directores".

De lo anterior se tienen las siguientes matrices para algunas rotaciones sobre un eje en específico, las rotaciones se ejemplifican en la Fig. 2.9.

- Rotación θ grados alrededor del eje x:

$$R_{\theta,x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{2.7}$$

- Rotación ϕ grados alrededor del eje y:

$$R_{\phi,y} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \tag{2.8}$$

- Rotación γ grados alrededor del eje z:

$$R_{\gamma,z} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

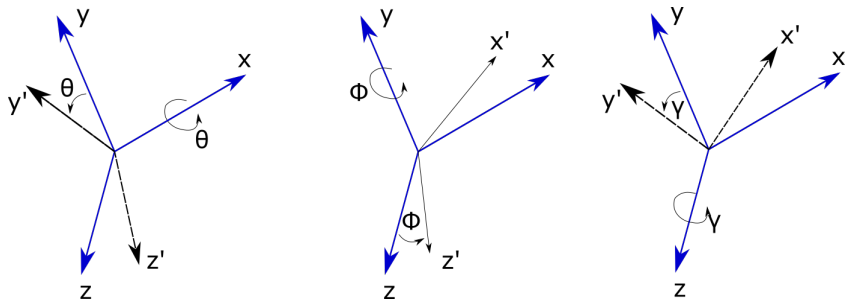


Figura 2.9: Representación de los giros en cada uno de los ejes.

2.7.1. Transformaciones rotacionales

Considere la Fig. 2.10 en la que se observan dos sistemas coordenados rotados entre sí. Si la matriz que define dicha rotación entre los sistemas coordenados es denominada R entonces las coordenadas del punto P pueden transformarse de un sistema coordenado a otro con suma facilidad. Por ejemplo, sea $P = [{}^mP_x \ {}^mP_y \ {}^mP_z]^T$ un punto visto desde el marco de referencia

$OX_mY_mZ_m$ en color rojo y la matriz R la matriz que va del marco de referencia $OX_mY_mZ_m$ al $OX_fY_fZ_f$, se define el punto P visto desde el marco de referencia $OX_fY_fZ_f$ en color gris como sigue [14]:

$$ {}^{f}P = R\ {}^{m}P = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} {}^{m}P_x \\ {}^{m}P_y \\ {}^{m}P_z \end{bmatrix} \tag{2.10} $$

De manera similar ocurre para pasar ahora del marco de referencia $OX_fY_fZ_f$ al $OX_mY_mZ_m$, nuevamente se considera que la matriz R es la matriz de rotación entre el sistema $OX_mY_mZ_m$ al $OX_fY_fZ_f$, por lo que para obtener la transformación en sentido opuesto basta con obtener la inversa de la matriz R, sin embargo, dado que las matrices de rotación son ortogonales se cumple que:

$$ R^{-1} = R^{T} \tag{2.11} $$

Por lo que dado el punto ${}^{f}P = [{}^{f}P_x\ {}^{f}P_y\ {}^{f}P_z]^T$ visto desde el marco de referencia en color gris se define el punto ${}^{m}P$ visto desde el marco $OX_mY_mZ_m$ como sigue:

$$ {}^{m}P = R^{-1}\ {}^{m}P = \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} \begin{bmatrix} {}^{f}P_x \\ {}^{f}P_y \\ {}^{f}P_z \end{bmatrix} \tag{2.12} $$

De la ecuación (2.12) se puede observar que la matriz inversa utilizada se obtiene de transponer la matriz utilizada en la ecuación (2.10).

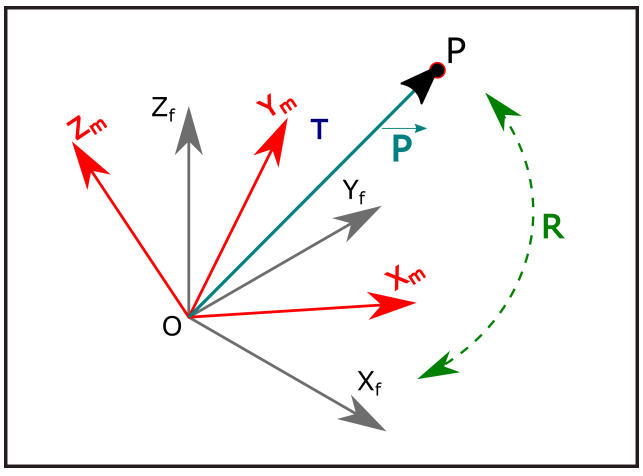


Figura 2.10: Transformación rotacional entre sistemas coordenados.

2.7.2. Composición de rotaciones

La composición de rotaciones se refiere a múltiples rotaciones de un sistema coordenado respecto a otro sobre algún eje en específico, de las que se destacan dos opciones, realizar rotaciones respecto del sistema de referencia actual tomando en cuenta la última rotación realizada (rotación respecto al eje actual), o realizar la rotación en base al sistema de referencia original sin modificar por las rotaciones hechas con anterioridad (rotación respecto a eje fijo) [14].

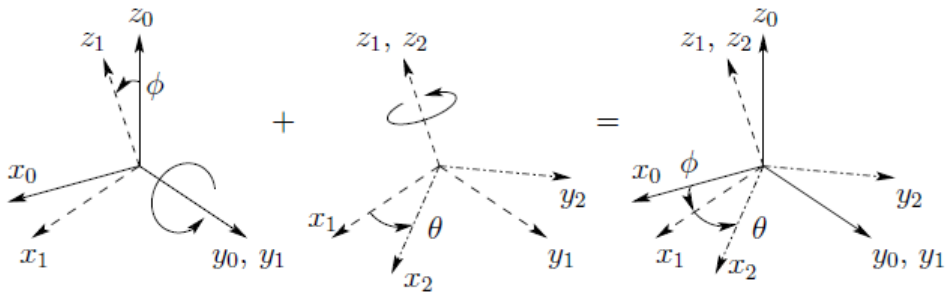


Figura 2.11: Composición de rotaciones respecto del eje actual [14].

Para ejemplificar dichas rotaciones considere las Figs. 2.11 y 2.12 en las que se representan rotaciones sobre eje fijo y actual, respectivamente. Se puede observar en la Fig. 2.11 una rotación respecto al eje y_0 y posteriormente una rotación sobre el eje z_1 obtenido después de realizar la primer rotación, por lo que se puede concluir que es una rotación respecto de los ejes actuales. Mientras que en la Fig. 2.12 se observa una rotación respecto del eje y_0 y posteriormente una rotación respecto del eje z_0 ambos son los ejes originales del sistema coordenado por lo que se trata de una rotación respecto de ejes fijos.

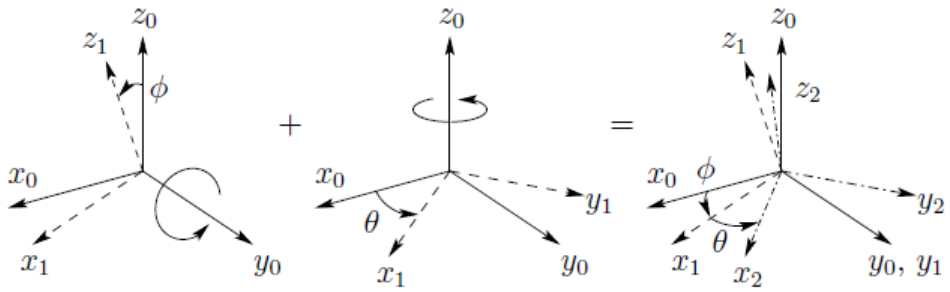


Figura 2.12: Composición de rotaciones respecto del eje fijo [14].

Una vez definidas las diferencias entre rotar el sistema coordenado respecto del eje fijo o actual se puede obtener la matriz que represente ambas rotaciones de la siguiente manera. Para rotaciones respecto de ejes fijos, se tiene que multiplicar las rotaciones individuales por la izquierda conforme se van realizando como se observa en la ecuación (2.13), por lo que si primero se realiza una rotación sobre el eje y y después una rotación respecto del eje z, se colocará la matriz de la primer rotación al centro y todas las demás a la izquierda de ésta, encontrando al extremo izquierdo la última rotación realizada:

$$R_{y_0 z_0} = R_{z_0} R_{y_0} \tag{2.13}$$

Al contrario de las rotaciones respecto de ejes fijos, en las rotaciones con respecto de ejes actuales las matrices de rotación individuales se van colocando a la derecha de la primera, tomando esto en cuenta, la última rotación realizada se encuentra en el extremo derecho, como se observa en la ecuación (2.14).

$$R_{y_0 z_1} = R_{y_0} R_{z_1} \tag{2.14}$$

2.7.3. Traslación y rotación entre sistemas coordenados

Una vez definidas la posición y orientación del cuerpo rígido, se procede a representar la orientación y posición (lo que se denominará *pose* del mecanismo) con ayuda de la matriz de rotación y el vector de posición entre sistemas coordenados. Para ello se hará uso de la Fig. 4.2, en la que se observan dos sistemas coordenados, el primero de ellos en color negro que se definirá como fijo a la referencia, mientras el segundo en color rojo se anclará al objeto en movimiento.

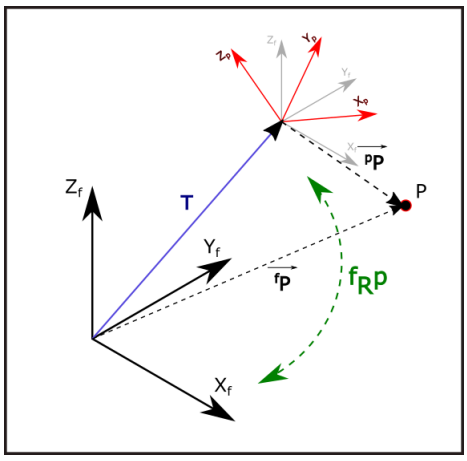


Figura 2.13: Vectores y elementos importantes para la transformación entre sistemas.

Por tanto, si se define una matriz de rotación entre el sistema coordenado en movimiento y el fijo denominada ${}^{f}R^{p}$ y además el vector que ubica el origen es el vector T se puede tener una representación del punto P visto desde el sistema de referencia fija como sigue:

$$ {}^{f}\vec{P} = T + {}^{f}R^{p}{}^{p}\vec{P} \tag{2.15} $$

De la ecuación (2.15) se puede observar que en primer lugar se obtiene la transformación del punto P visto desde el marco de referencia móvil al marco de referencia fijo a partir de la multiplicación con la matriz de rotación ${}^{f}R^{p}$ entre sistemas y después se suma el vector de posición que ubica el origen (T), obteniendo el vector ${}^{f}P$.

2.8. Representación por la teoría de tornillos

Una forma alternativa de representar la posición de un cuerpo rígido en el espacio es a través de la teoría de tornillos. Básicamente se refiere a que la configuración o pose de un cuerpo se puede establecer a partir de una rotación alrededor del que se denominará eje del tornillo y un desplazamiento sobre el mismo. Esta forma de representar la posición y orientación del cuerpo resulta ser una herramienta sumamente poderosa, además de denotar la posición, obtiene la velocidad lineal y angular del cuerpo así como las fuerzas experimentadas por el mismo de manera sumamente simplificada. [3, 18]
La teoría de tornillos fue establecida en 1830 por Chasles, quien fue el primero en referirse al concepto de "Giro" de cuerpos rígidos, misma idea que desarrollaría después Poinsot en 1848. Posterior a ellos, Plücker propondría su expresión para los llamados "Tornillos". Sin embargo, no es hasta 1875 cuando en su libro "Screw Theory", Ball establece una forma sistémica de análisis utilizando dicha teoría para analizar la cinemática y dinámica de un cuerpo rígido bajo restricciones realmente complejas. Así continuaron las investigaciones y desarrollo alrededor de la teoría, Hunt y Phillips presentaron la representación matemática y geométrica de los tornillos en 1964, haciendo resurgir un tanto la teoría, Hunt presenta en 1976 un artículo de suma importancia en el campo titulado "Screw Systems in Spatial Kinematics", así continuamente se siguen desarrollando nuevas aplicaciones de la teoría. [3, 18, 19, 20]

2.8.1. Giros y momentos

Existen dos conceptos fundamentales que sustentan la base de la teoría de tornillos, el primer elemento fundamental es el Giro o "Twist" y el segundo el momento alrededor del eje de tornillo o "Wrench". Estos dos conceptos son conocidos como Dualidad en la teoría. Por tanto, se puede crear una primer idea de lo que representa matemáticamente a un "Tornillo".

En realidad el Tornillo puede entenderse como un par de vectores concatenados cuyas partes son las siguientes [1, 16]:

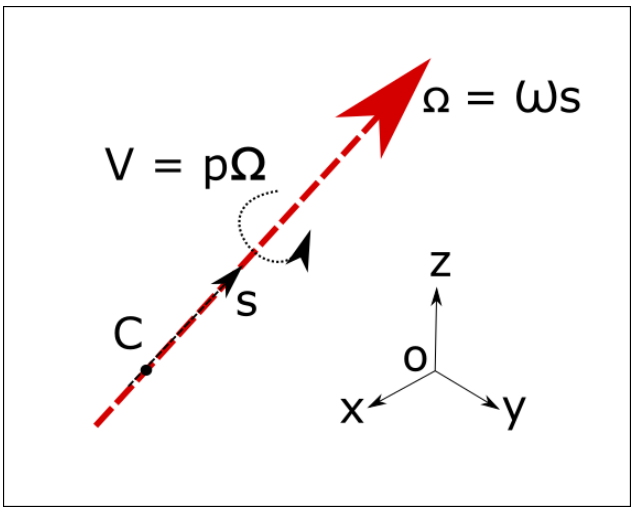


Figura 2.14: Esquema representativo de un tornillo y los elementos que lo componen.

- **Parte Primaria:** Consiste de un vector unitario a lo largo del eje de giro del tornillo, asociado a la velocidad angular del cuerpo.
- **Parte Dual:** Consiste de un segundo vector que representa el momento producido por la parte primaria alrededor de un punto de referencia fijo, denominado Polo de Referencia, de igual manera se puede asociar al componente de velocidad lineal del cuerpo.

De la Fig. 2.14 se puede observar de manera un tanto más representativa los elementos que conforman a un tornillo. Básicamente el tornillo puede expresarse como: $\$ = \{\Omega|\mathbf{V}\}$, que no es más que un par de vectores concatenados, que representan la velocidad lineal y angular del cuerpo. Se puede extender la definición del vector de tornillo como sigue:

$$\$ = \{\Omega = \omega s|\mathbf{V} = c \times \omega s + vs\} \tag{2.16}$$

La ecuación (2.16) se puede entender rápidamente apoyándose en la Fig. 2.14 donde s es un vector unitario a lo largo del eje de tornillo, c es un punto sobre el eje de tornillo, ω es la magnitud de la velocidad angular del cuerpo, p es el cabeceo o "pitch" que básicamente es la relación entre velocidad angular y lineal: $p = V/\Omega$.

2.8.2. Estados de velocidad del cuerpo rígido

El estado de velocidad de un cuerpo rígido ($\mathbf{V}$) está definido por el conjunto de velocidad angular y lineal de un punto en un cuerpo rígido medido desde otro cuerpo o marco de

referencia. El objetivo del estado de velocidad es proporcionar la suficiente información para calcular la velocidad de cualquier otro punto en el cuerpo rígido.
Para casos en donde el movimiento se limita a tres grados de libertad, el vector consta de tres dimensiones, mientras que para el caso espacial el vector consta de seis dimensiones. Básicamente el vector está conformado por la concatenación del componente de velocidad angular denominado *parte primaria* P($\mathbf{V}$) y el componente de velocidad lineal denominado *parte dual* D($\mathbf{V}$) [20]. La ecuación que representa al estado de velocidad es la siguiente:

$$ {}^{A}\mathbf{V}_{o}^{B} = \left\{ \begin{array}{c} {}^{A}\omega^{B} \\ {}^{A}v_{o}^{B} \end{array} \right\} \tag{2.17} $$

Donde ${}^{A}\mathbf{V}_{o}^{B}$ es el estado de velocidad del cuerpo B respecto del A donde el polo de referencia es el punto O, mientras que ${}^{A}\omega^{B}$ es la velocidad angular del cuerpo rígido B visto desde A, que no depende del polo de referencia y ${}^{A}v_{o}^{B}$ es la velocidad lineal del punto O en B visto desde A.

2.8.3. Coordenadas de Plücker para líneas

Las coordenadas de Plücker se refieren básicamente a asignar seis coordenadas homogéneas a un espacio proyectivo tridimensional. En otras palabras, se puede entender como definir un tornillo por cada línea en el espacio, por lo que las seis coordenadas homogéneas corresponden a los elementos del par de vectores concatenados que conforman el tornillo. De igual manera se compone de parte primaria y dual [20].
Por tanto, se puede definir un vector unitario en dirección de la línea para representar la *parte primaria* (${}^{A}s^{B}$) de modo que se cumpla que:

$$ {}^{A}\omega^{B} =_{A} \omega_{B} \, {}^{A}s^{B} \tag{2.18} $$

donde ${}_{A}\omega_{B}$ es la magnitud de la velocidad angular del cuerpo B respecto de A.

Mientras que la *parte dual* o momento producido por la linea alrededor del polo de referencia está dado por:

$$ {}^{A}s_{O}^{B} = h^{A}s^{B} +^{A} s^{B} \times r_{O/P} \tag{2.19} $$

Donde h es el paso del tornillo, que representa la relación entre la velocidad lineal y angular, se puede entender al hacer la analogía con el movimiento lineal que realiza un tornillo

al desplazarse cierto ángulo, dichos movimientos se relacionan por el paso. De igual manera ocurre con las velocidades, para cierta cantidad de velocidad angular, la cantidad de velocidad lineal que experimenta el cuerpo se relaciona con el paso (h). Mientras que $r_{O/P}$ es el vector de posición del punto P que ubica la línea respecto del polo de referencia.

El paso está definido de la siguiente manera:

$$h = \frac{{}^{A}v_{O}^{B} \cdot {}^{A}\omega^{B}}{{}_{A}\omega_{B}^{2}} \tag{2.20}$$

Finalmente, se puede definir un tornillo normalizado (solo la parte primaria está normalizada) de la siguiente manera:

$${}^{A}\$^{B} = \begin{bmatrix} {}^{A}s^{B} \\ {}^{A}s_{O}^{B} \end{bmatrix} \tag{2.21}$$

Y además se cumple que el estado de velocidad y el tornillo unitario están relacionados a partir de la velocidad angular del cuerpo, obteniendo la siguiente igualdad.

$${}^{A}\mathbf{V}^{B} = {}_{A}\omega_{B}\ {}^{A}\$^{B} \tag{2.22}$$

2.8.4. Operaciones básicas del álgebra de Lie

Para poder operar con los tornillos dado que su representación es análoga a la forma que tiene el estado de velocidad, se hace uso de algunas operaciones definidas en el álgebra de Lie [20]. Se consideran tres estados de velocidad definidos como sigue:

$$\mathbf{V}_1 = (\omega_1, V_{O_1})\ ;\ \mathbf{V}_2 = (\omega_2, V_{O_2})\ ;\ \mathbf{V}_3 = (\omega_3, V_{O_3}) \tag{2.23}$$

Por tanto, la **suma** se define de la siguiente manera:

$$\mathbf{V}_1 + \mathbf{V}_2 = (\omega_1, V_{O_1}) + (\omega_2, V_{O_2}) = (\omega_1 + \omega_2, V_{O_1} + V_{O_2}) \tag{2.24}$$

De igual manera se define el **producto por un escalar** a continuación:

$$\lambda\mathbf{V}_1 = \lambda(\omega_1, V_{O_1}) = (\lambda\omega_1, \lambda V_{O_1}) \tag{2.25}$$

Finalmente se define el **producto de Lie** de la siguiente manera, mostrando además algunas de sus propiedades que podrán resultar útiles para el análisis cinemático del mecanismo a partir de tornillos.

$$[\mathbf{V}_1 \ \ \mathbf{V}_2] = [(\omega_1, V_{O_1}) \ \ (\omega_2, V_{O_2})] = (\omega_1 \times \omega_2, \omega_1 \times V_{O_2} - \omega_2 \times V_{O_1}) \tag{2.26}$$

Algunas de las propiedades son:

1. **Potencia nula:**

$$[\mathbf{V}_1 \ \ \mathbf{V}_1] = 0 \tag{2.27}$$

2. **Distributivo:**

$$\begin{array}{rcl} [\mathbf{V}_1 \ \ \lambda_2\mathbf{V}_2 + \lambda_3\mathbf{V}_3] & = & \lambda_2[\mathbf{V}_1 \ \ \mathbf{V}_2] + \lambda_3[\mathbf{V}_1 \ \ \mathbf{V}_3] \\ \text{ó} \quad [\lambda_1\mathbf{V}_1 + \lambda_2\mathbf{V}_2 \ \ \mathbf{V}_3] & = & \lambda_1[\mathbf{V}_1 \ \ \mathbf{V}_3] + \lambda_2[\mathbf{V}_2 \ \ \mathbf{V}_3] \end{array} \tag{2.28}$$

3. **Identidad de Jacobi:**

$$[\mathbf{V}_1 \ \ [\mathbf{V}_2 \ \ \mathbf{V}_3]] + [\mathbf{V}_3 \ \ [\mathbf{V}_1 \ \ \mathbf{V}_2]] + [\mathbf{V}_2 \ \ [\mathbf{V}_3 \ \ \mathbf{V}_1]] = 0 \tag{2.29}$$

4. **No es conmutativo:**

$$[\mathbf{V}_1\mathbf{V}_2] \neq [\mathbf{V}_2\mathbf{V}_1] \tag{2.30}$$

de hecho se cumple que:

$$[\mathbf{V}_1\mathbf{V}_2] = -[\mathbf{V}_2\mathbf{V}_1] \tag{2.31}$$

2.8.5. Formas de Killing y Klein

Una operación análoga de cierto modo al producto escalar en el álgebra vectorial convencional, son las formas de Killing y Klein en el algebra de Lie, a partir de estas operaciones se pasa del espacio e(3) al de los reales ($\mathbb{R}$) [20].
Considere dos elemento del álgebra de Lie definidos como sigue:

$$\mathbf{V}_1 = (\omega_1, v_{O_1}) \ ; \ \mathbf{V}_2 = (\omega_2, v_{O_2}) \tag{2.32}$$

Considerando los elementos anteriores, la **forma de Killing** se define como:

$$\begin{aligned} < *; * > &= e(3) \times e(3) \to \mathbb{R} \\ \langle \mathbf{V}_1; \mathbf{V}_2 \rangle &= \langle (\omega_1; v_{O_1}); (\omega_1; v_{O_2}) \rangle \\ &= \omega_1 \cdot \omega_2 \end{aligned} \tag{2.33}$$

Donde el punto en la expresión final entre ω_1 y ω_2 denota el producto escalar en el álgebra vectorial convencional. Por lo que, de manera general se puede definir la forma de Killing como el producto escalar entre las partes primarias de los elementos.

Mientras que la **forma de Klein** está definida de la siguiente manera:

$$\begin{aligned} \{*; *\} &= e(3) \times e(3) \to \mathbb{R} \\ \{\mathbf{V_1}; \mathbf{V_2}\} &= \{(\omega_1; v_{O_1}); (\omega_2; v_{O_2})\} \\ &= \omega_1 \cdot v_{O_2} + \omega_2 \cdot v_{O_1} \end{aligned} \tag{2.34}$$

Por último, un concepto de igual importancia que las operaciones definidas anteriormente, es el operador de polaridad, debido a que será de utilidad para relacionar entradas y salidas en ecuaciones de cinemática de velocidad y aceleración. Su función radica en poder expresar la forma de Klein de manera matricial. Una forma de definir al **operador de polaridad** ($\mathbf{\Delta}$) es la siguiente:

$$\mathbf{\Delta} = \begin{bmatrix} 0 & I_3 \\ I_3 & 0 \end{bmatrix} \tag{2.35}$$

donde I_3 representa a la matriz identidad de 3×3 y los 0 una matriz de ceros de 3×3. La anterior es una de muchas formas de representar el operador de polaridad, pues se puede definir de maneras alternas siempre y cuando cumpla que $\mathbf{\Delta\Delta} = I_6$.

2.8.6. Tornillos de las juntas básicas

Para las juntas rotacional y traslacional la definición de los tornillos se simplifica de manera significativa, para el caso traslacional se tiene que la parte primaria es igual a cero y la parte dual está dada por el vector unitario en dirección del movimiento, como se observa en la ecuación (2.36) [20].

$$\$_t = \begin{bmatrix} 0 \\ s \end{bmatrix} \tag{2.36}$$

Mientras que para la junta rotacional la parte primaria es el vector unitario en dirección del tornillo y la parte dual se obtiene del producto cruz entre la parte primaria y vector de posición al polo de referencia, como se establece en la ecuación (2.37).

$$\$_r = \begin{bmatrix} s \\ s \times r_{O/P} \end{bmatrix} \tag{2.37}$$

2.8.7. Reciprocidad entre tornillos

Se dice que dos tornillos son recíprocos si se cumple que la forma de Klein entre ellos es igual a cero, es decir: $\$_1$ y $\$_2$ son recíprocos sí [20]:

$$\{\$_1; \$_2\} = 0 \tag{2.38}$$

Existen tres condiciones básicas para que tornillos que representan movimientos puramente rotacionales o traslacionales sean recíprocos entre ellos:

1. Dos tornillos con movimiento rotacional puro, es decir, con paso ($h = 0$) son recíprocos si estos son cooplanares.
2. Dos tornillos con movimiento puramente traslacional son siempre recíprocos entre ellos.
3. Un tornillo traslacional y uno rotacional son recíprocos si son perpendiculares entre sí.

El significado físico que tiene el hecho de que dos tornillos sean recíprocos es que no aportan trabajo al movimiento del cuerpo o del sistema en general. Considere dos tornillos, uno que represente el movimiento puramente rotacional de un cuerpo y uno más que represente la fuerza aplicada al mismo, por lo que se definen como sigue:

$$\omega\$_1 = \omega(s_1, s_{01}) = (\omega, v_0) \ ; \ f\$_2 = f(s_2, s_{02}) = (f, C_0) \tag{2.39}$$

Por lo que el trabajo realizado por dicha fuerza está dado por:

$$\{\omega\$_1; f\$_2\} = \omega_1 \cdot C_0 + f \cdot v_0 \tag{2.40}$$

que básicamente es el producto de Klein entre los tornillos de fuerza y velocidad, por tanto, el hecho de igualar la ecuación (2.40) a cero implica que el producto de Klein es cero (definición de tornillos recíprocos), y por ende la fuerza no realiza trabajo sobre el movimiento del cuerpo. Una interpretación física alterna de los tornillos recíprocos se relaciona con las restricciones impuestas al mecanismo, considere por ejemplo una corredera que solo se desplaza a lo largo del eje X (ver Fig. 2.15).

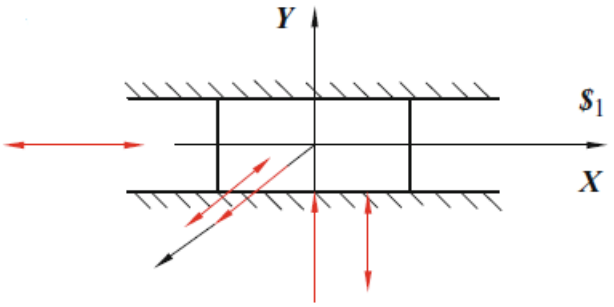


Figura 2.15: Restricciones asociadas a los tornillos recíprocos [16].

Se define un tornillo puramente traslacional en esa dirección como sigue:

$$\$ = (\ 0 \quad 0 \quad 0 \ ; \ 1 \quad 0 \quad 0\ ) \tag{2.41}$$

Tomando en cuenta que los tornillos recíprocos a la ecuación (2.41) representarán las restricciones de movimiento impuestos sobre la corredera, se puede esperar que existan cinco tornillos recíprocos, dos de ellos en las direcciones de los dos ejes restantes Y y Z, además de las restricciones de giros en los tres ejes, por lo que estos estarían definidos como sigue:

$$\$_1 = (\ 0 \quad 0 \quad 0 \ ; \ 0 \quad 1 \quad 0\ ) \tag{2.42}$$

$$\$_2 = (\ 0 \quad 0 \quad 0 \ ; \ 0 \quad 0 \quad 1\ ) \tag{2.43}$$

$$\$_3 = (\ 1 \quad 0 \quad 0 \ ; \ 0 \quad 0 \quad 0\ ) \tag{2.44}$$

$$\$_4 = (\ 0 \quad 1 \quad 0 \ ; \ 0 \quad 0 \quad 0\ ) \tag{2.45}$$

$$\$_5 = (\ 0 \quad 0 \quad 1 \ ; \ 0 \quad 0 \quad 0\ ) \tag{2.46}$$

Cumpliendo con los requisitos para ser recíprocos entre sí, los dos primeros ($\$_1, \$_2$) son traslacionales por lo que son recíprocos con el traslacional a lo largo del eje X, mientras que los tres restantes son rotacionales y perpendiculares al traslacional en la dirección X y por tanto son recíprocos.

2.9. Cinemática del mecanismo

La mayoría de los robots son utilizados como manipuladores, es decir, resulta fundamental el hecho de conocer la posición del efector final o plataforma móvil en el caso de los robots paralelos. Establecer y conocer el movimiento de los cuerpos sin importar la causa que lo produce, es justamente de lo que se encarga la cinemática. Por lo que los modelos matemáticos se enfocan en describir la posición y orientación del dispositivo de modo que se defina lo que se conoce como configuración o pose del mecanismo.[4, 21]

Cinemática directa: Básicamente la cinemática directa se refiere a obtener la posición y orientación del efector final de un robot dado los valores de las juntas actuadas, los grados de inclinación de una junta revoluta por ejemplo o la longitud del eslabón para una junta prismática. Dependiendo del tipo de mecanismo implementado para el robot la cinemática directa puede resultar o no sencilla. Para los mecanismoS seriales la cinemática directa resulta bastante simple, mientras que para los mecanismos paralelos se complica un poco más, de la literatura se conoce que existen incluso soluciones para la cinemática directa de mecanismos paralelos que no son cerradas, de las que solo se puede aproximar una solución de manera iterativa de todas las posibles [22, 23].

Cinemática inversa: La cinemática inversa consiste en que dados ciertos puntos en el espacio de trabajo del robot se puedan obtener valores de las juntas actuadas que permitan al robot alcanzar dicho punto. Al contrario de lo que sucede con la cinemática directa, en este caso resulta mucho más complicado el cálculo para robots seriales que para robots paralelos, pues en muchos de los casos existen múltiples soluciones para el caso serial mientras que para los mecanismos paralelos la solución es prácticamente trivial [23].

2.10. Unidades de medición inercial (IMU)

Las unidades de medición inercial son dispositivos que incluyen varios sensores integrados a un mismo marco inercial, entre los dispositivos más populares que se incluyen en las IMU se encuentran giroscopios, acelerómetros y magnetómetros. Permitiendo con eso, obtener una unidad de medición inercial de nueve grados de libertad, pues cada uno de los dispositivos realiza mediciones en los tres ejes coordenados (X,Y y Z). Con la utilización de la IMU en este proyecto de tesis se pretende obtener y medir las variables de orientación y velocidad angular de la plataforma móvil para comparar con los resultados analíticos.

2.10.1. Acelerómetros

Los acelerómetros son dispositivos capaces de medir la aceleración a la que están sometidos. Dicha aceleración es la variación de la velocidad por unidad de tiempo, medido en $[m/s^2]$ o $[G]$, donde $G = 9.81m/s^2$, que corresponde con la gravedad de la tierra. La mayoría de estos dispositivos funcionan con tecnologías MEMS, contienen placas capacitivas unas fijas y otras móviles, debido al movimiento relativo entre estas placas es que se obtiene la acelaración de dicho cuerpo en movimiento [24].
Los acelerómetros pueden ser analógicos o digitales. Los acelerómetros analógicos muestran un voltaje proporcional a la aceleración que experimenta el sensor, normalmente entre el rango de voltaje de alimentación y tierra. Mientras que los acelerómetros digitales obtienen resultados con mucho menor ruido a partir de protocolos de comunicación I2C o SPI, por lo que los datos se leen de forma binaria [24].
El acelerómetro incluido en la IMU nos permitirá descomponer el vector de gravedad normal a la superficie de la tierra que siempre se lee por el sensor en sus componentes sobre el plano sobre el que se encuentra la plataforma móvil para así obtener la inclinación de la misma.

2.10.2. Magnetómetros

Los magnetómetros son utilizados para medir o cuantificar la intensidad magnética. La unidad de medida de la intensidad del campo magnético es Tesla (T) y difiere de acuerdo a la región del globo terráqueo [25]. Existen dos tipos de magnetómetros, los escalares y los vectoriales. Los magnetómetros escalaras miden únicamente la intensidad total del campo magnético al que están sometidos, mientras los de tipo vectorial tienen la capacidad de medir la intensidad del campo magnético en determinadas direcciones. Dicho sensor nos permitirá obtener la orientación alrededor del eje Z funcionando como brújula digital midiendo la variación respecto del polo norte magnético.

2.10.3. Giroscopios

Los giroscopios son dispositivos capaces de medir la velocidad angular de un cuerpo en cada momento, es decir, la variación del desplazamiento angular por unidad de tiempo. Los giroscopios pueden clasificarse en dos categorías de acuerdo a la forma en que obtienen las medidas.
Pueden ser **giroscopios mecánicos** cuyo principio de funcionamiento se basa en la conservación del momento angular. Mientras que los **giroscopios electrónicos** basan su funcionamiento en tecnología MEMS, estos dispositivos contienen pequeñas masas unidas directamente al chip de silicio sensibles a las vibraciones y basados en el principio de Coriolis obtienen la

velocidad angular del cuerpo [24]. Dicho sensor nos permitirá obtener de manera directa las mediciones de velocidad angular respecto de los tres ejes (X, Y y Z).

2.11. Servo motores

Un servomotor es básicamente un motor con algunas características especiales, entre las que se encuentran por ejemplo, el sistema de retroalimentación que le permite saber al circuito o módulo de control del motor la posición en la que se encuentra el eje del motor y éste pueda tomar acciones de correción en caso de que la posición no fuera la adecuada. El servomotor es un tipo de motor en el que se tiene control de la posición del eje y de la velocidad con la que gira el mismo, todo esto indicado a partir de señales eléctricas. Las principales formas de control del servomotor son: a partir de una señal PWM para variar la posición de acuerdo al ancho de pulso y frecuencia o a partir de secuencias de pulsos a distintas frecuencias sin variar el ciclo de trabajo de la señal [26]. Para el desarrollo del presente trabajo resultan fundamentales al ser las juntas prismáticas encargadas de la entrada de movimiento en el mecanismo. Se utilizan servomotores para considerar como válidas todas las posiciones o longitudes de extensión deseadas y enfocarse en el análisis cinemático.

2.12. Modulación por ancho de pulso (PWM)

La modulación por ancho de pulso se basa en la comparación de una señal de referencia modular y una señal portadora de forma triangular o diente de sierra (ver Fig. 2.16), la comparación entre dichas señales generará como resultado un tren de pulsos con ancho específico. La variación de la señal de referencia da como resultado distintos anchos de pulsos de los que se pueden obtener múltiples aplicaciones, desde conmutación de puentes inversores hasta control de posición de servomotores [27]. Para el caso del presente trabajo será de utilidad para controlar la extensión y retracción de los servomotores, indicando dicha cantidad de manera proporcional al ciclo de trabajo de la señal PWM.

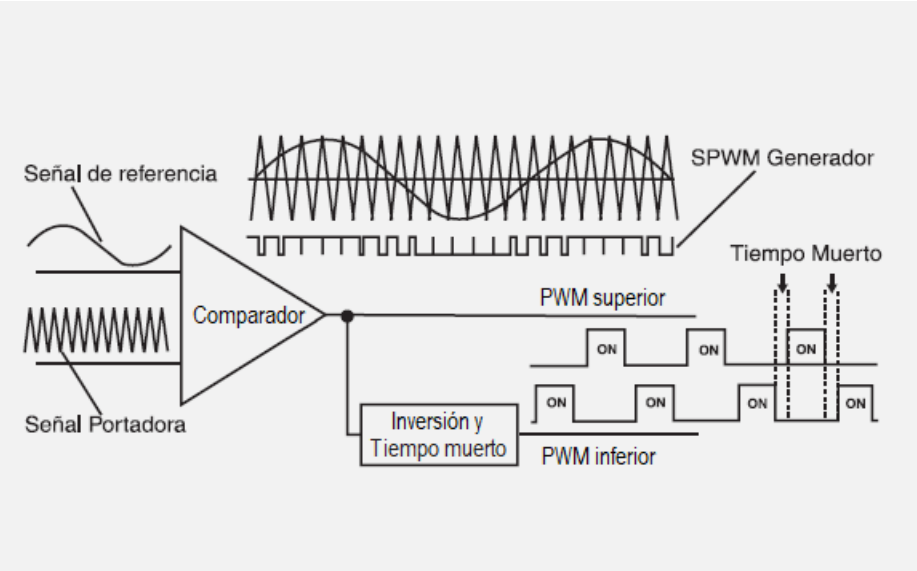


Figura 2.16: Circuito generador de PWM.

Capítulo 3

Diseño, manufactura y ensamble del robot

De la bibliografía consultada [1, 2, 20], se ha observado que los robots paralelos cumplen con una restricción en particular que les permite simplificar el análisis cinemático, dicha restricción se refiere a que la plataforma móvil debe ser una copia escalada de la base. Además, de todas las configuraciones revisadas, se opta por una configuración parecida a la plataforma de "Gough" dada la versatilidad y movilidad que presenta dicha arreglo en las extremidades.

De la Tabla 3.1 se puede observar que el tipo de robot que ofrece la mayor cantidad de aplicaciones y el máximo número de grados de libertad es justamente la plataforma de Gough, por lo que una vez seleccionado el tipo de robot, hace falta definir la configuración en cada una de las extremidades y la forma que tendrán cada una de las plataformas.
Por tanto, las características generales con las que debe cumplir la plataforma son las siguientes:

- Dos plataformas, una escalada respecto de la otra, de preferencia hexagonales para facilitar el análisis cinemático.
- Seis extremidades de longitud variable.
- Seis grados de libertad.
- De aplicación múltiple, permitiendo tareas posteriores como control de trayectorias, análisis dinámico, entre otras.

Tabla 3.1: Comparación de características de los robots presentados en el marco teórico.

Nombre de la plataforma	Grados de libertad	Número de extremidades del mecanismo	Aplicaciones comunes
Gough	6	6-UPS	·Prueba de neumáticos ·Alineación de objetos ·Estructuras reconfigurables ·Simuladores de vuelo ·Simuladores de conducción ·Soporte para camillas ·Excitadores de vibración
Stewart	6	3-RPPS	·Simulador de vuelo ·Simulador de conducción ·Alineación de objetos
DELTA	4	6-RRR	·Paletizar componentes ·Impresión 3D ·Máquinas herramientas
Robots accionados por cables	5	2 ó 3 Cables	·Captura de video ·Transporte de objetos ·Paletizar objetos

3.1. Opciones de conformación de las extremidades del robot

Se presentan tres alternativas para la conformación de las extremidades, tratando de reducir lo más posible el número de elementos que conformarían el mecanismo, sin afectar los grados de libertad, procurando que en cada una de ellas se incluya una junta prismática, misma que será la entrada de movimiento que definirá la longitud de las extremidades.

3.1.1. Configuración 1. 6-U<u>P</u>U

Para la primer configuración se proponen seis extremidades compuestas cada una de una junta universal conectada a la base, seguida de una junta prismática y finalmente una junta universal conectada a la plataforma móvil. La junta prismática será justamente el actuador lineal.

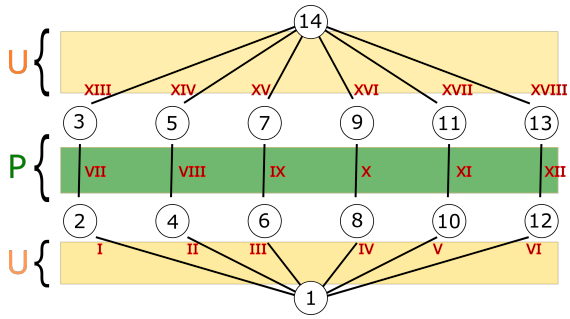


Figura 3.1: Grafo asociado a la configuración uno.

Para entender y realizar el análisis de los grados de libertad bajo esta configuración, se muestra en la Fig. 3.1 el grafo correspondiente, en el que se observan los elementos que conforman el mecanismo encerrados en círculos, además de las juntas denotadas por números romanos. En general los grafos se describen de la siguiente manera:

- En color azul se resaltan las líneas correspondientes a juntas revolutas.
- En color verde se resaltan las líneas correspondientes a juntas prismáticas.
- En color beige se resaltan las líneas correspondientes a las juntas universales.
- Los nodos en los grafos representan los eslabones o cuerpos de los que se conforma el mecanismo, siendo siempre el "1" la base fija y el número mayor corresponde a la "plataforma móvil".
- Las líneas que conectan a los nodos corresponden a las juntas entre cada uno de los cuerpos.

De la Fig. 3.1 se puede observar que el mecanismo estaría compuesto de 14 elementos ($n = 14$), además existen 18 juntas que conectan cada uno de los cuerpos ($j = 18$), de igual manera se puede notar que las juntas son de dos tipos, universales y prismáticas, permitiendo dos y un grado de libertad respectivamente. Mientras que no existen grados de libertad pasivos ($f_p = 0$). Por tanto, utilizando la ecuación (2.3) se tiene lo siguiente:

$$F = \lambda(n - j - 1) + \sum_{i=1}^{j} f_i - f_p = 6(14 - 18 - 1) + 12(2) + 6(1) - 0 = 0 \text{ GDL} \tag{3.1}$$

De la ecuación (3.1) se tiene que el mecanismo bajo la configuración número uno no tendría ningún grado de libertad. Es decir, quedaría totalmente definido una vez que se ensamble

completamente el robot. Se puede notar de igual manera que de la sumatoria de grados de libertad permitidos por la junta i-ésima se obtuvieron los productos 12×2 y 6×1 que se refieren a las doce juntas universales que permiten dos grados de libertad y las seis juntas prismáticas que permiten solo un grado.

3.1.2. Configuración 2. 6-RUPUR

Como segunda propuesta se tiene la siguiente configuración, que varía respecto de la configuración uno por el hecho de que se agregan en los extremos de las extremidades dos juntas revolutas, cuyos ejes de rotación se proponen normal al plano que contiene a los ejes de rotación de las juntas universales que los preceden. Tratando de emular juntas esféricas en los extremos de las extremidades.

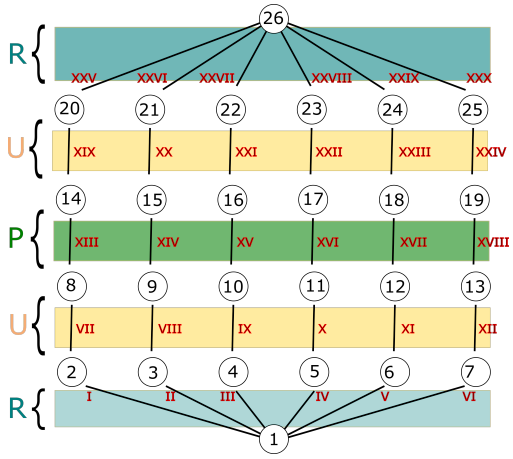


Figura 3.2: Grafo asociado a la configuración 2.

De igual manera que para el caso anterior, en la Fig. 3.2 se muestra el grafo asociado a la configuración dos, del que se obtienen los siguientes datos: $n = 26$, $j = 30$, $f_p = 6$, además de considerar que las juntas revolutas y prismáticas permiten un grado de libertad, mientras que las juntas universales permiten dos grados de libertad. Por lo que haciendo uso de la ecuación (2.3), se tiene el siguiente resultado:

$$F = 6(26 - 30 - 1) + 18(1) + 12(2) - 6 = 6 \text{ GDL} \tag{3.2}$$

De la ecuación (3.2) se obtiene que los grados de libertad del mecanismo de la configuración dos son seis, obteniendo con esto los grados que se desean en el robot, sin embargo se puede notar de igual manera que existen seis grados de libertad pasivos debido a las juntas esféricas virtuales cuyo eje de rotación colineal al eje de actuación de la junta prismática provoca grados de libertad redundantes sobre esa dirección.

3.1.3. Configuración 3. 6-UPUR

Para la configuración tres se propone casi la misma configuración que en la mostrada anteriormente, eliminando únicamente la junta revoluta anclada a la parte inferior de la extremidad, con lo que se pretende eliminar los grados de libertad pasivos. Por tanto, la extremidad estaría compuesta de una junta universal, seguida de una junta prismática, seguida de otra junta universal para finalizar con una junta revoluta.

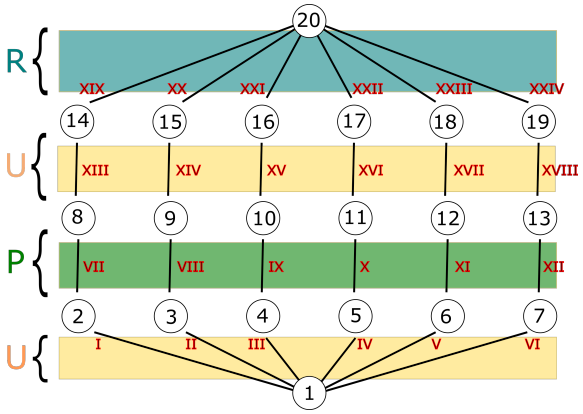


Figura 3.3: Grafo asociado a la configuración 3.

En la Fig. 3.3 se observa el grafo asociado a las configuración tres, obteniendo los siguientes datos, $n = 20$, $j = 24$, $f_p = 0$, considerando de igual manera que para las juntas prismáticas y revolutas se tiene permitido un grado de libertad y dos para las juntas universales. Nuevamente, de la ecuación (2.3) se tiene el siguiente resultado:

$$F = 6(20 - 24 - 1) + 12(2) + 12(1) = 6 \text{ GDL} \tag{3.3}$$

Una vez analizados los grados de libertad para cada una de las configuraciones, se opta por utilizar la configuración tres dado que permite los seis grados de libertad evitando los grados de libertad pasivos, además con esto se logra acercarse lo más posible al diseño original propuesto por Gough al emular la junta esférica en el extremo superior de la extremidad con el arreglo entre la junta universal y revoluta propuesta en la configuración tres, al mismo tiempo que se requieren menos componentes que en la configuración dos.

3.2. Diseño de las plataformas del robot

Para las plataformas fija y móvil se propone que sean hexagonales, simplemente escaladas una de la otra por un factor $r = 1.5$, de modo que las dimensiones de las plataformas están relacionadas de la siguiente manera:

$$\text{Dim}_{\text{base}} = r \cdot \text{Dim}_{\text{pmovil}} \tag{3.4}$$

Por lo que una vez definidas las dimensiones, ya sea de la plataforma móvil o la plataforma fija (base), la plataforma restante queda automáticamente acotada.
Las dimensiones de los hexágonos propiamente no son de gran importancia dado que para el análisis cinemático lo que realmente importará será la ubicación de los puntos de conexión de las extremidades (barrenos). Sin embargo, en la Fig. 3.4 se muestran las dimensiones propuestas para la base, siendo estas 450$[mm]$ para la distancia entre vértices y 225$[mm]$ para la dimensión de las caras.

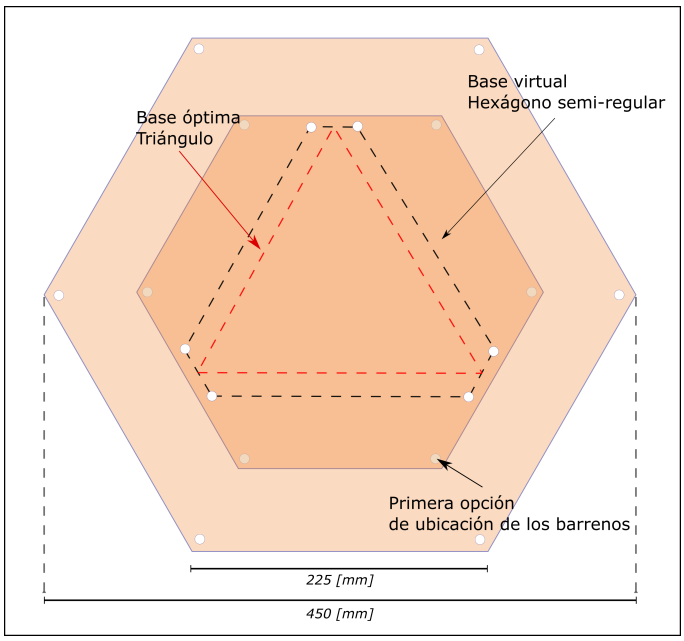


Figura 3.4: Dimensiones de las plataformas y ubicación de los barrenos.

Como primera opción se tenía planeado colocar los barrenos en ambas plataformas en los vértices del hexágono pues en la mayoría de los ejemplos teóricos consultados de la bibliografía se hace de esta manera [1, 3]. Sin embargo, al momento de probar dicha configuración a partir de simulaciones en SolidWorks, se obtuvo que ésta es inestable pues presenta un gran número de puntos de singularidad en los que el mecanismo se vuelve incontrolable debido a los grados de libertad que gana en estas poses.
Debido a este inconveniente, se procedió a modificar la ubicación de los barrenos de modo que la plataforma quedará soportada por una especie de triángulos como se observa en la Fig. 3.5. De forma ideal, para lograr la mayor estabilidad del robot y evitar singularidades, los actuadores deberían conectarse por pares a un mismo punto en la plataforma móvil formando una plataforma móvil triangular como se observa en las Figs. 3.4 y 3.5.

Dicha configuración es referida en la literatura como configuración 6-3 de la plataforma de Stewart-Gough, haciendo referencia a los seis puntos de conexión en la base y tres puntos de conexión en la plataforma móvil, entre las características que presenta se encuentra una mayor estabilidad respecto de la configuración 6-6 (seis conexiones en la base y seis en la plataforma móvil) [1, 2, 16].
Sin embargo, el hecho de conectar dos actuadores a un mismo punto implicaría buscar o diseñar una junta que permita tal configuración de modo que no se afecte la movilidad del mecanismo. Por lo que para evitar este problema, se propone colocar dos barrenos lo más cercanos posible formando un hexágono semiregular (ver Fig. 3.4) mejorando la estabilidad sin incluir componentes extra a las extremidades.

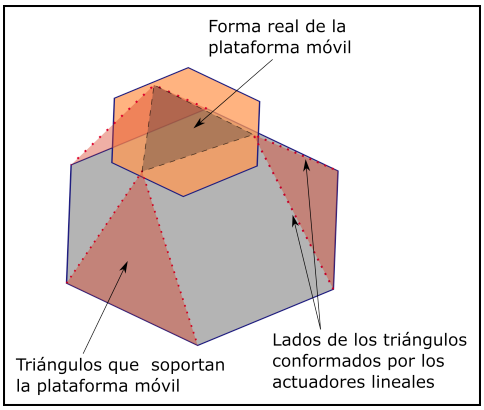


Figura 3.5: Triángulos ideales para soportar la plataforma.

Como se mencionó anteriormente, las dimensiones de los hexágonos no son de interés para el análisis cinemático, en cambio la ubicación de los barrenos y el espesor de las plataformas, además de la ubicación del marco de referencia en cada una de ellas es de suma importancia, pues brindarán la ubicación espacial de cada uno de los puntos.

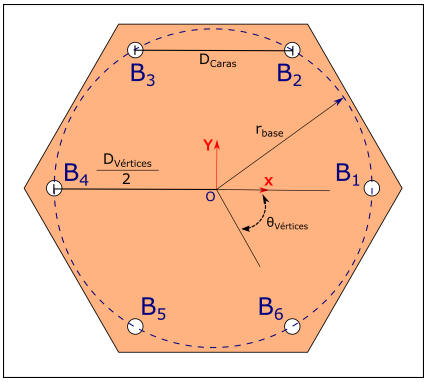


Figura 3.6: Nomenclatura y dimensiones de la base.

En la Fig. 3.6 se pueden observar las etiquetas utilizadas para denotar cada uno de los barrenos o puntos de conexión (B_i), además del radio del círculo (r_{base}) sobre el que se encuentran ubicados los puntos de conexión, de modo que se pueda obtener su ubicación bajo la siguiente fórmula:

$$[B_{X_i}, B_{Y_i}] = [r_{base}\cos(\theta_{vertices}\cdot(i-1)), r_{base}\sin(\theta_{vertices}\cdot(i-1))] \text{ para } i = 1, ..., 6 \tag{3.5}$$

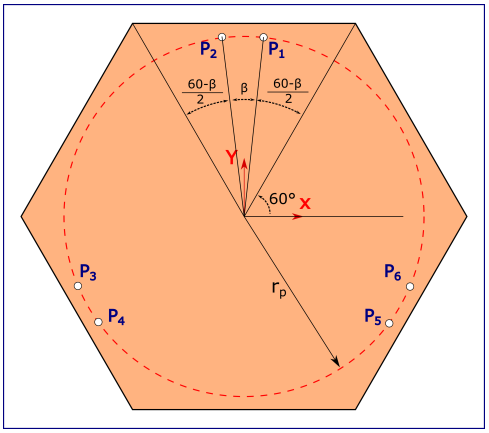


Figura 3.7: Nomenclatura y dimensiones de la plataforma.

De manera similar, en la Fig. 3.7 se puede observar la nomenclatura y algunos ángulos importantes para la ubicación de cada uno de los puntos de conexión (P_i), el ángulo de separación entre puntos de conexión (β) y r_p el radio del círculo sobre el que se encuentran inscritos los puntos de conexión, de modo que la ubicación de estos se obtenga bajo la siguiente fórmula:

$$[P_{X_i}, P_{Y_i}] = [r_p\cos(60i - (-1)^i(\gamma)), r_p\sin(60i - (-1)^i(\gamma))] \text{ para } i = 1, ..., 6 \tag{3.6}$$

donde:

$$\gamma = \frac{60-\beta}{2} \tag{3.7}$$

Una vez obtenidas las ecuaciones para ubicar en el plano XY los puntos de conexión tanto en la plataforma móvil como en la base, falta determinar su ubicación respecto del eje Z, por lo que haciendo uso de la Fig. 3.8 se observa que para todos los puntos P_i el componente $P_{Z_i} = -(z_u + t)$, mientras que para los puntos B_i en la base se tiene $B_{Z_i} = z_u$. Donde z_u denota la distancia al centro de las juntas universales y t el espesor de las plataformas hexagonales.

En referencia a la Fig. 3.8 se puede observar una línea verde que une el centro de las juntas universales, dicha línea se considerará como la longitud de las extremidades, por lo que, como se mostrará más adelante, estará constituida de media junta universal, el actuador lineal y la media junta universal atada al otro extremo de la pierna del robot.

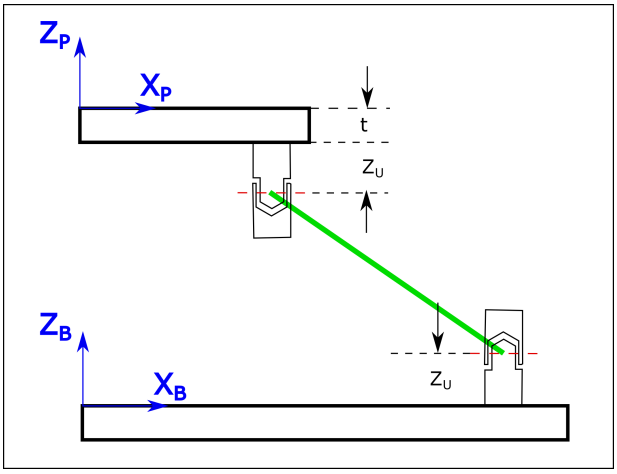


Figura 3.8: Desfase en las coordenadas en z de los puntos de conexión en las plataformas.

Una vez establecido lo anterior, es posible obtener la ubicación de los puntos B_i colocados en la base fija, así como los puntos P_i colocados en la plataforma móvil, vistos desde el marco de referencia atado a la plataforma móvil, obteniendo las coordenadas mostradas en la Tabla 3.2.

Tabla 3.2: Ubicación de los puntos P_i y B_i en [mm].

B_1	[200,0,21.5]	P_1	[14.01,114.14,-27.85]
B_2	[100,173.2,21.5]	P_2	[-14.01,114.14,-27.85]
B_3	[-100,173.2,21.5]	P_3	[-105.85,-44.93,-27.85]
B_4	[-200,0,21.5]	P_4	[-91.84,-69.20,-27.85]
B_5	[-100,-173.2,21.5]	P_5	[91.84,-69.20,-27.85]
B_6	[100,-173.2,21.5]	P_6	[105.85,-44.93,-27.85]

3.3. Elementos mecánicos y eléctricos que conforman las extremidades

A continuación se describen los componentes que constituyen cada una de las extremidades como son las juntas universales, el actuador lineal, así como las juntas revolutas, además

de un par de elementos que funcionan como conectores entre las juntas universales con la base del actuador lineal y el vástago del mismo.

3.3.1. Juntas universales de 20mm

Para las juntas universales, se optó por adquirir elementos que fueran comerciales y de fácil acceso, procurando que el costo fuese el menor posible, pues de presentarse algún desperfecto o daño en los componentes estos se puedan sustituir fácilmente. Las características de las juntas son las siguientes: aleación de acero, diámetro interior (D_i) igual a $10[mm]$, diámetro exterior (D_e) igual a $20[mm]$, distancia al centro de la junta universal marcada en la Fig. 3.9 como Z_u igual a $21.5[mm]$.

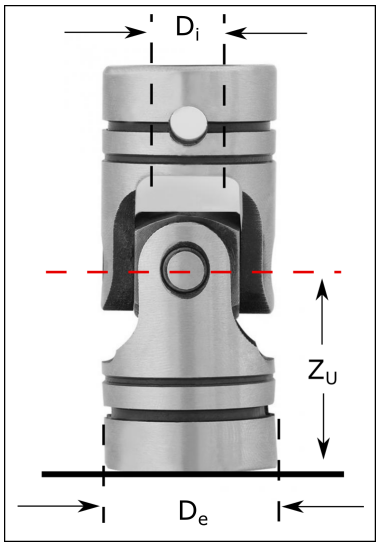


Figura 3.9: Medidas importantes en la junta universal.

3.3.2. Actuador Lineal L-16R

Para el caso de los actuadores lineales se optó por servomotores lineales de la marca Actuonix, cuyo modelo es **L16-R-140-35-6** dada las características que ofrecen, como son: una carrera de $140[mm]$, carga dinámica de $50[N]$, además de ser bastante compactos [28]. Otra de las ventajas que ofrecen este tipo de actuadores es que al tratarse de servomotores, tienen implementado el control de posición que asegura alcanzar la posición deseada, con cierto grado de error, sin embargo, con esto la tarea se reduce a resolver la cinemática considerando todas las posiciones como correctas en cuanto a longitud del vástago se refiere.

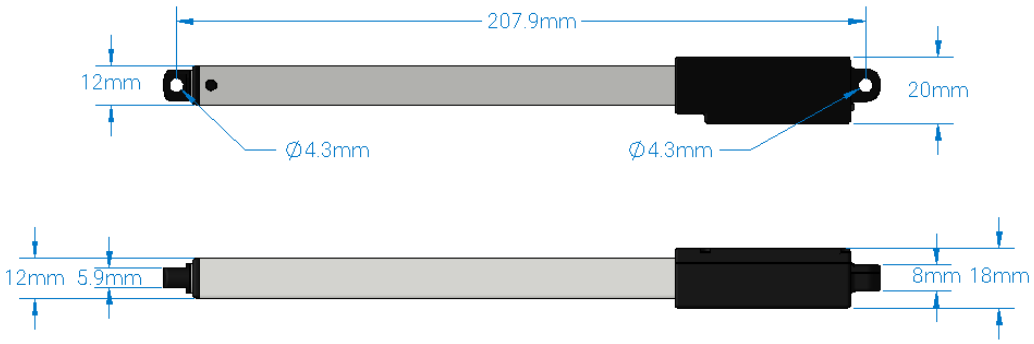


Figura 3.10: Actuador L-16R con el vástago totalmente retraído.

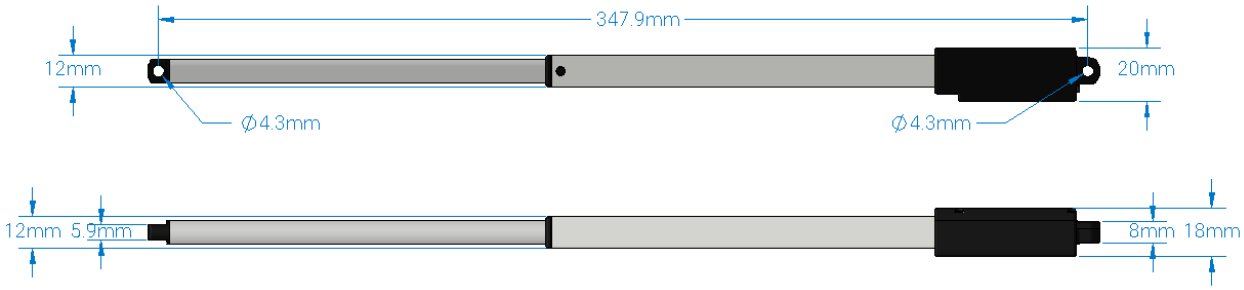


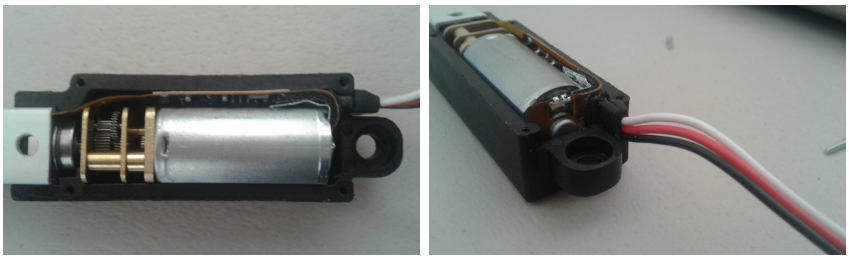
Figura 3.11: Actuador L-16R con el vástago totalmente extendido.

En las Figs. 3.10 y 3.11 se observan las dimensiones del actuador en completa retracción y en total extensión, dimensiones que serán de utilidad para el análisis cinemático posterior. Además, en la Tabla 3.3 se muestran todas las características del servomotor, de las que destacan un par de elementos, el primero de ellos el tipo de motor, pues resultará fundamental conocer sus características para trabajos futuros de control, y el segundo elemento que resulta de mucho interés es la forma de retroalimentación que tiene el motor respecto de la extensión del vástago, de las características se puede concluir que es un potenciómetro que varía de manera lineal con respecto a la longitud en pulgadas como se ve en la Tabla 3.3.

Sin embargo, dado que los actuadores son servomotores, la señal de retroalimentación está incluida dentro del pequeño circuito de control en la carcasa del motor como se puede observar en la Fig. 3.12, por lo que una de las tareas futuras a realizar para aplicar control, será justamente aislar dicha señal y las terminales del motor obteniendo mayor control del actuador.

Tabla 3.3: Características mecánicas y electrónicas del servomotor LR-16.

Rango de voltaje de operación	4.5-7.5 [V]
Peso	103.47 [g]
Amplitud de la señal de control	3-5 [V]
Temperatura de operación	-10°C∼+50°C
Velocidad (Sin carga)	32 [mm/s]
Corriente de operación (Sin carga)	650 [mA]
Empuje dinámico	50 [N]
Carga estática	200 [N]
Tipo de motor	Motor de CD de escobillas
Relación de retroalimentación	16 [kΩ/inch]
Relación de engranajes	32:1
Longitud del cable de conexión	300 [mm]
Calibre del cable de conexión	26 AWG
Carrera	140 [mm]



(a) Vista superior de la carcasa (b) Vista posterior de la carcasa

Figura 3.12: Interior de la carcasa de los servomotores.

3.3.3. Rodamientos 607-Z

Para la junta revoluta colocada dentro de la plataforma móvil se optó por utilizar un rodamiento 607-Z de la marca *KDYD bearings*, dado que los requerimientos de carga son mínimos y las dimensiones del rodamiento se adecúan perfectamente con las plataformas diseñadas, es que se optó por utilizarlos. Algunas de las características más importantes se muestran en la Tabla 3.4 y en la Fig. 3.13 [29].

Tabla 3.4: Características mecánicas del rodamiento de bolas 607-Z.

Diámetro interior (d) (ver Fig. 3.13)	7 [mm]
Diámetro exterior (D) (ver Fig. 3.13)	19 [mm]
Espesor (B) (ver Fig. 3.13)	6.75 [mm]
Capacidad de carga dinámica básica (C)	2.34 [kN]
Capacidad de carga estática básica(C_o)	0.95 [kN]
Carga límite de fatiga (P_u)	0.04 [kN]
Velocidad de referencia	85000 [RPM]
Velocidad límite	53000 [RPM]
Factor de cálculo k_r	0.025 [N]
Factor de cálculo f_0	13 [N]
Masa del rodamiento	0.0079 [kg]

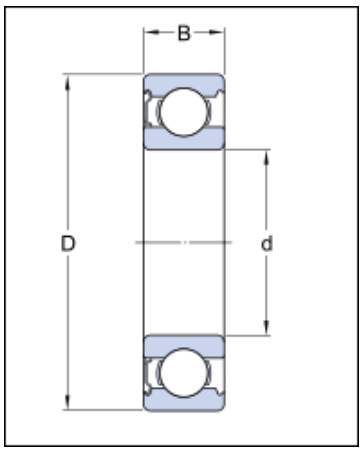


Figura 3.13: Dimensiones del rodamiento 607-Z.

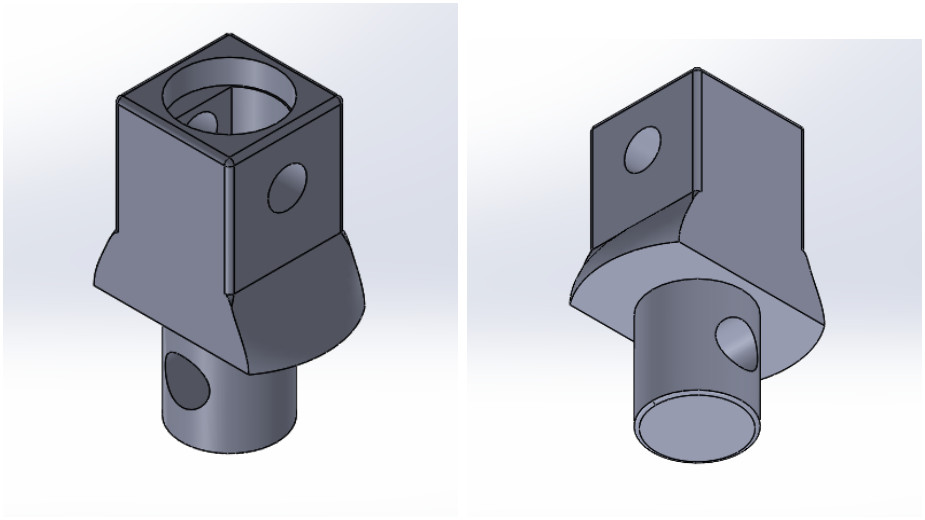
3.3.4. Elementos de acoplamiento

Dada la geometría de la base del actuador y la geometría de la punta del vástago, es necesario diseñar un par de elementos que permitan la conexión entre ellos y las juntas universales que poseen una cavidad cilíndrica de 10[mm] de diámetro, por lo que se proponen los siguientes elementos diseñados, considerando el modelo CAD del actuador lineal, en particular del extremo del vástago y la superficie inferior de la base del actuador, se obtuvieron moldes con las formas correspondientes a la punta del vástago y la base del actuador, que son básicamente los cubos con cavidades que se observan en las Figs. 3.14 y 3.15. Posteriormente, se procedió a agregar el cilindro de diez milímetros que permite la conexión con la junta revoluta además de un pequeño soporte que cubre hasta el diámetro exterior de la junta, mismo que

se observa en la vista inferior de las Figs. 3.14 y 3.15 (se pueden consultar los dibujos técnicos de los elementos de soporte en el Apéndice D).

3.4. Ensamble del mecanismo

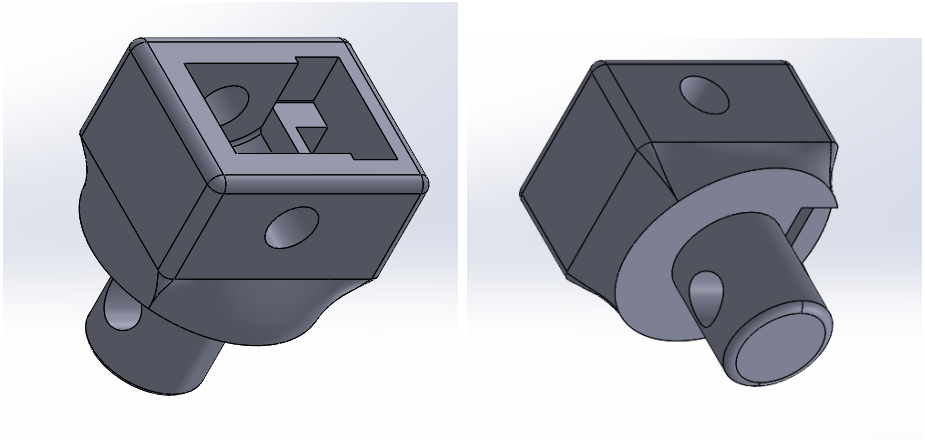
Una vez descritos todos los componentes en las extremidades, se muestra a continuación la forma en que se ensambla cada una de las extremidades del robot, así como la disposición de las extremidades alrededor de la plataforma fija.



(a) Vista superior del elemento de acoplamiento I

(b) Vista inferior del elemento de acoplamiento I

Figura 3.14: Elemento de acoplamiento I.



(a) Vista superior del elemento de acoplamiento II

(b) Vista inferior del elemento de acoplamiento II

Figura 3.15: Elemento de acoplamiento II.

Se comienza por describir el ensamble de la parte inferior de las extremidades tomando como referencia la Fig. 3.16. Se observa que el ensamble consta de cinco elementos sin contar el actuador. La forma de ensamble se describe a continuación:

1. Se coloca el elemento de acoplamiento sobre la base del actuador.
2. Se sujeta a partir del tornillo y la tuerca métrica M4x0.7.
3. Se coloca sobre el elemento de acoplamiento la junta universal.
4. Se sujeta la junta a universal al elemento de acoplamiento a partir del "pasador U".

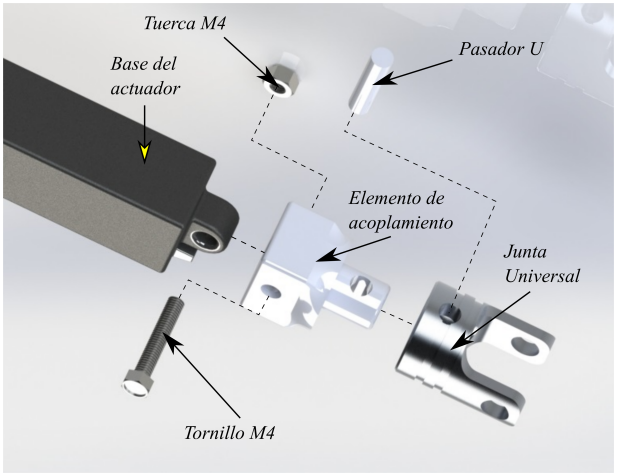


Figura 3.16: Dibujo explosionado de los elementos en la parte inferior de las extremidades.

A continuación se describe el ensamble de los elementos en la parte superior de la extremidad. De la Fig. 3.17 se observa que de igual manera que para el caso anterior se utilizan cinco elementos para ensamblar la parte superior de la extremidad. El procedimiento se describe a continuación:

1. Se coloca sobre el vástago el elemento de acoplamiento.
2. Se sujeta el elemento de acoplamiento al vástago del actuador a partir del tornillo y tuerca métricas M4x0.7.
3. Se coloca sobre el elemento de acoplamiento la junta universal.
4. Se sujeta la junta al elemento de acoplamiento a partir del "pasador U".

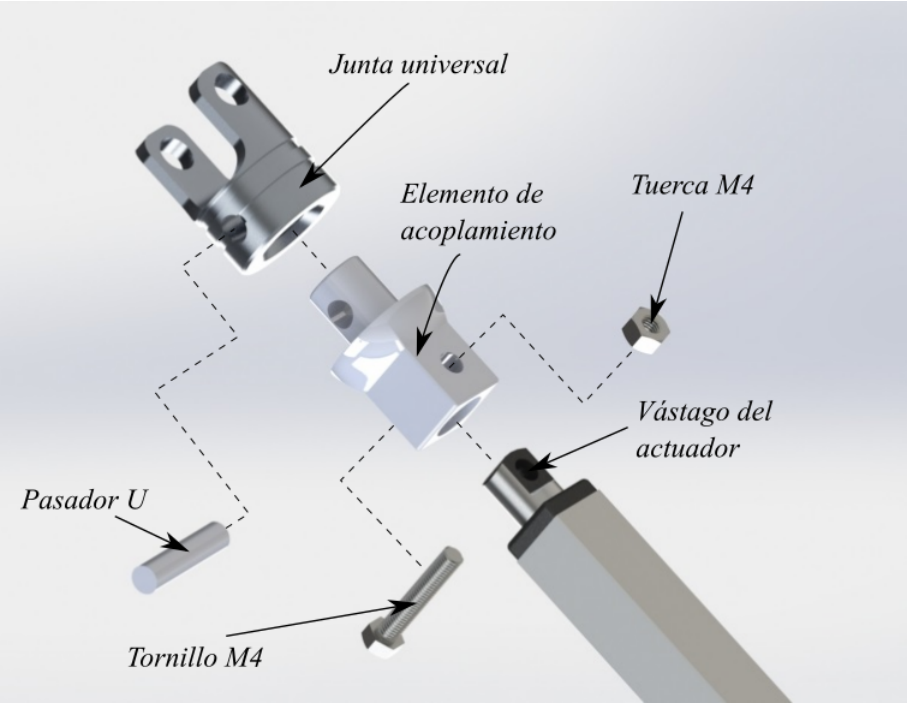


Figura 3.17: Dibujo explosionado de los elementos en la parte superior de las extremidades.

Una vez ensamblada la extremidad se observa como en la Fig. 3.18.

Figura 3.18: Extremidad ensamblada.

Por último, se muestra el ensamble de la plataforma con la junta universal y la colocación del rodamiento en la plataforma. Como se observa en la Fig. 3.19, el ensamble consta de dos elementos, un pasador entre las juntas universales y los rodamientos y posteriormente éstos se introducen a presión en los barrenos sobre la plataforma móvil. De igual manera que en el ensamble de las extremidades, el pasador se sujeta a la junta universal con ayuda de un pasador transversal a éste.

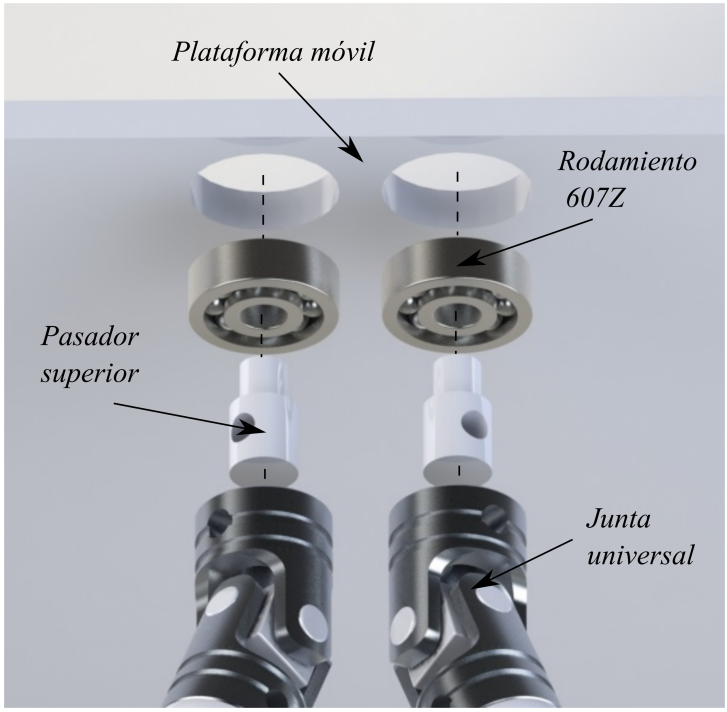


Figura 3.19: Ensamble de las extremidades con la plataforma móvil.

Finalmente el robot ensamblado se observa en la Fig. 3.20.

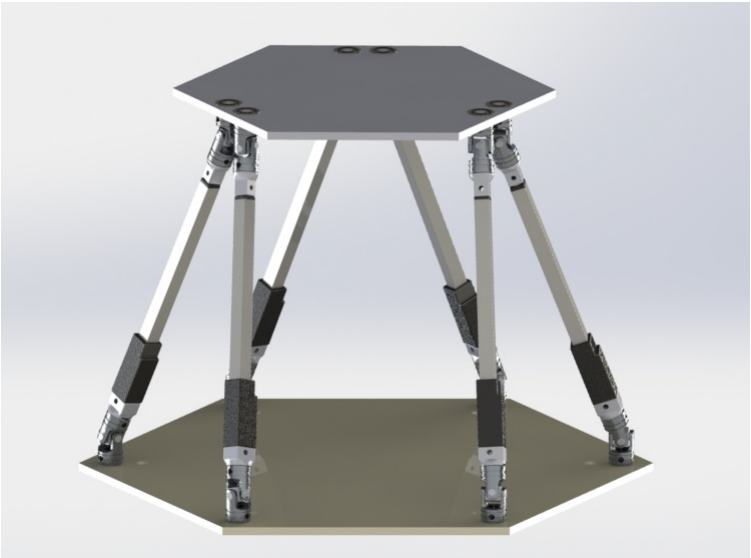


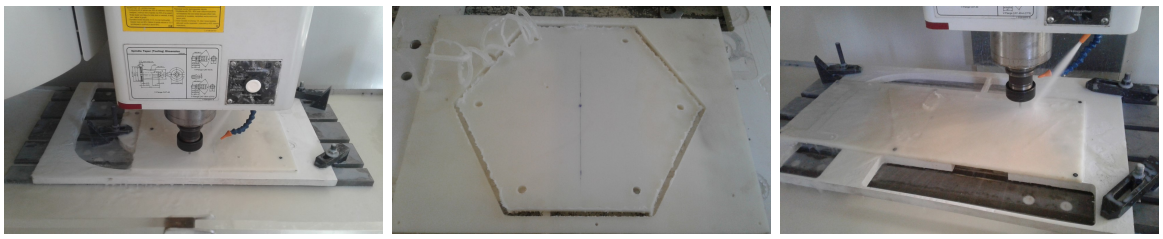
Figura 3.20: Robot paralelo 6-UPUR ensamblado.

3.5. Manufactura del robot

De lo descrito anteriormente, se puede observar que de las partes que componen al robot se deben maquinar las bases móvil y fija, además de los soportes para las bases y el extremo de los actuadores, al igual que los pasadores para sujetar las juntas universales a las plataformas. Por lo que a continuación se describe dicho proceso comenzando por las plataformas.

3.5.1. Manufactura de las plataformas móvil y fija

Ambas plataformas son de forma hexagonal regular variando únicamente en la ubicación de los barrenos o puntos de conexión con las extremidades. Debido a que el robot construido no está diseñado para una tarea en particular no se cuenta con requerimientos de fuerza o esfuerzos que debe soportar o restricciones de peso, entre otras características que permitan realizar un proceso de selección de materiales más detallado, por lo que tratando de reducir lo más posible el peso del prototipo sin afectar su resistencia es que se elige Nylamid como materia prima, además el material resulta ser de fácil y rápido maquinado por lo que es una buena elección para fabricar las plataformas. Se procedió al maquinado con ayuda de la fresadora CNC del taller de manufactura avanzada de la universidad. Dado que la geometría no es complicada, se utilizó el modo semiautomático de la máquina y se programó desde ella misma la ruta a seguir para el maquinado.



(a) Maquinado de la plataforma móvil.

(b) Plataforma móvil terminada de maquinar.

(c) Maquinado de media plataforma fija y elementos de soporte.

Figura 3.21: Maquinado de las plataformas móvil y fija del robot.

En la Fig. 3.21 se puede observar el maquinado de las plataformas móvil y fija, además de la plataforma móvil terminada de maquinar. Para la plataforma fija se optó por separar en dos partes la pieza, debido a que el espacio de trabajo de la máquina es solo de 70×30 [cm], mientras que la plataforma mide 45 [cm] entre vértices, por lo que como se puede observar en la Fig. 3.21 c), se tuvo que separar en dos operaciones la fabricación de dicha pieza.

3.5.2. Manufactura de los elementos de soporte para los actuadores

Para el caso de los elementos de soporte para los actuadores se utilizó una impresora 3D bajo el principio de modelado por deposición fundida, por lo que el proceso resultó bastante simple, una vez obtenido el modelo CAD en Solidworks™simplemente se exportó al software MPrint™utilizado por la impresora y se comenzó con la impresión de las piezas.

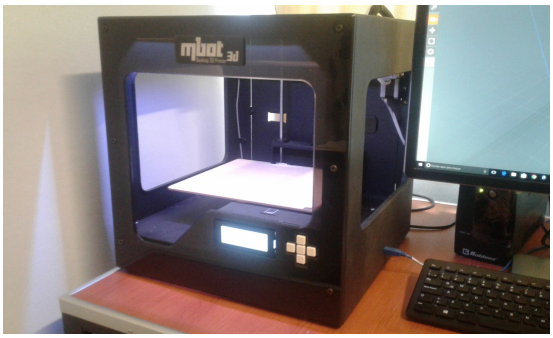


Figura 3.22: Impresora utilizada para maquinar las piezas de soporte.

En la Fig. 3.22 se puede observar el modelo de impresora utilizado para el maquinado de las piezas de soporte, así como en la Fig. 3.23 se observan las piezas terminadas y listas para el ensamble.

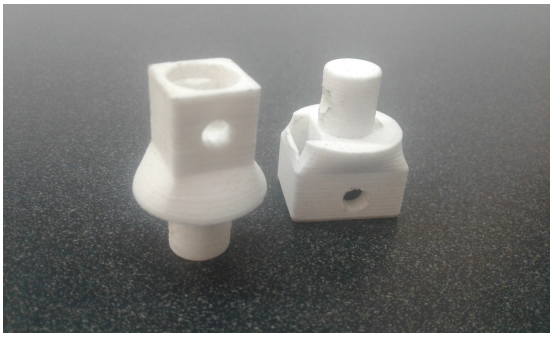


Figura 3.23: Piezas de soporte terminadas.

3.5.3. Manufactura de los pasadores para las juntas universales

Para sujetar las juntas universales a ambas plataformas, fue necesario diseñar un par de pasadores que se adecuarán al ensamble, para sujetar las juntas universales a la base móvil, los pasadores deben tener en uno de los extremos un barreno para que por él se coloque otro pasador transversal y se sujete con la junta universal, mientras que para la base, debe tener una pequeña ranura para colocar seguros tipo E.
Mientras que los pasadores para sujetar la plataforma móvil con las juntas universales solo

deben tener un único barreno en uno de los extremos y una reducción de diámetro para ser introducidos a presión sobre los rodamientos como se observa en la Fig. 3.19.
Como material para maquinar los pasadores se optó por acero A36, particularmente redondo de 3/8 [in], debido a la fácil maquinabilidad del material y a que los requerimientos de resistencia son mínimos. Para el maquinado se utilizaron los tornos pequeños del taller de manufactura avanzada.

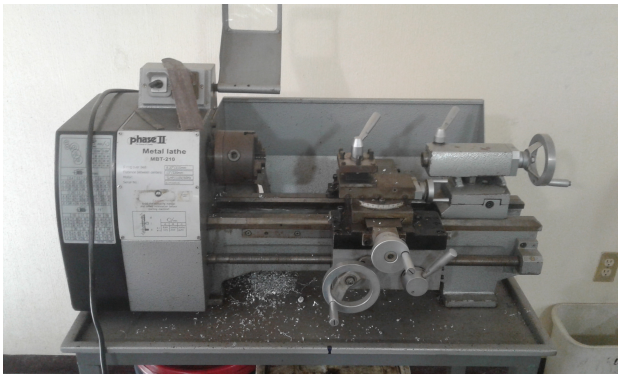


Figura 3.24: Mini torno utilizado para la manufactura de los pasadores.

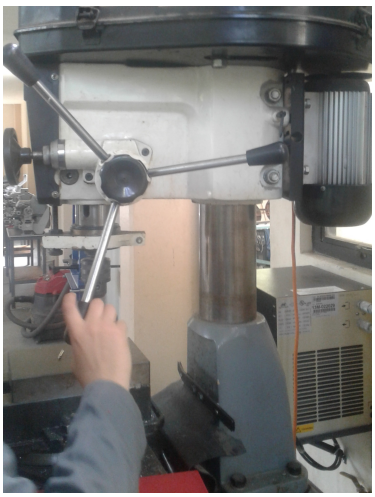


Figura 3.25: Taladro vertical utilizado para realizar los barrenos.

De las Figs. 3.24, 3.25 y 3.26 se observan las máquinas-herramientas utilizadas para la manufactura de los pasadores y los pasadores terminados.

Figura 3.26: Pasador terminado y colocado en la junta universal.

El procedimiento consistió de un pequeño desbaste con el torno y la disminución del diámetro para el caso de los pasadores para la plataforma móvil y posteriormente se realizaron los barrenos con ayuda del taladro vertical.

Finalmente el prototipo terminado y ensamblado se puede observar en la Fig. 3.27.

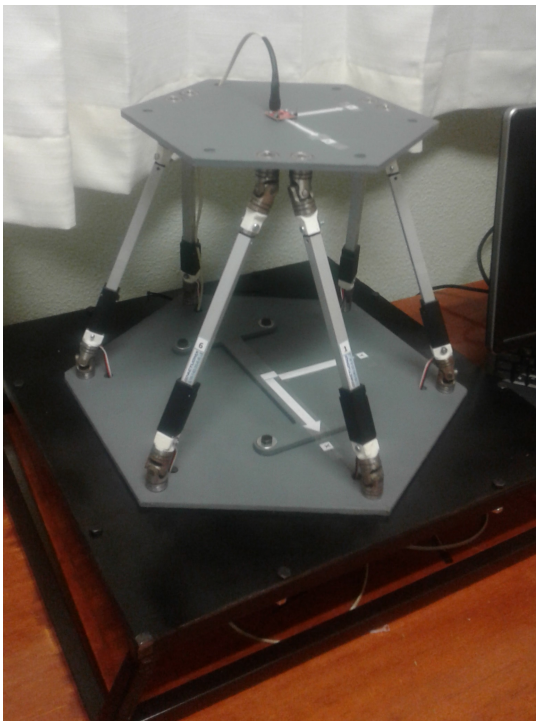


Figura 3.27: Prototipo de robot paralelo 6-U$\underline{P}$UR terminado y ensamblado.

Capítulo 4

Análisis cinemático del robot

A continuación se muestra el análisis de la cinemática directa e inversa del prototipo de robot paralelo de seis grados de libertad propuesto en esta tesis, bajo la configuración 6-U$\underline{P}$UR. Se hará uso de la notación planteada y descrita en el capítulo anterior para cada uno de los puntos de conexión tanto en la base como en la plataforma móvil, así como las dimensiones de las juntas universales, en particular la distancia que ubica el centro de los ejes de rotación de las mismas respecto de la base.

4.1. Análisis cinemático directo

Para poder realizar el análisis cinemático directo como se ha mencionado, se refiere al hecho de encontrar la pose del robot conociendo los valores de todas las entradas de movimiento, lo que en este caso está representado por las longitudes de las extremidades, medidas desde la mitad de las juntas universales como se mencionó en la descripción de la Fig. 3.8, es necesario obtener cada una de las ecuaciones de cerradura de las extremidades, por lo que se procederá a obtener la ecuación de cerradura para la extremidad uno haciendo uso de los vectores representados en la Fig. 4.1.
Posteriormente, se generalizará la ecuación obtenida de modo que se pueda aplicar dicha solución a cualquier clase de configuración conociendo únicamente los puntos de conexión que están atados a cada una de las extremidades.

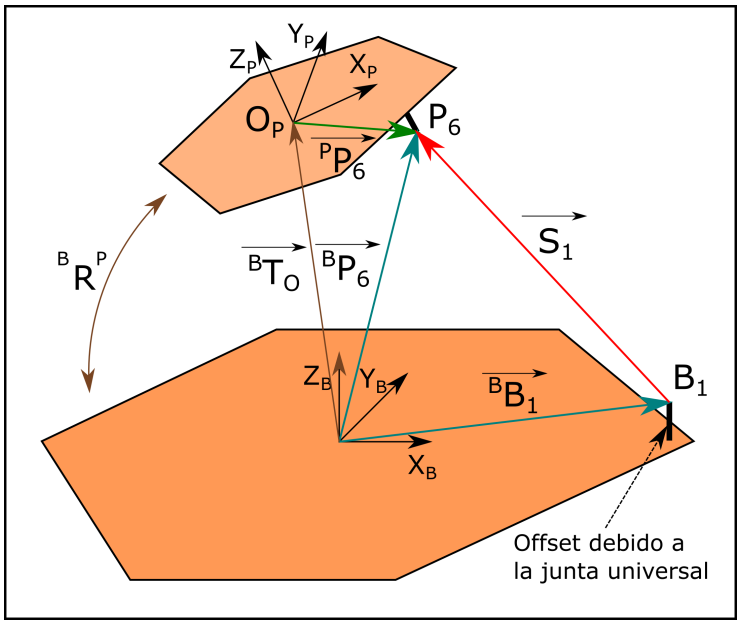


Figura 4.1: Vectores útiles para encontrar las ecuaciones de cerradura.

En la Fig. 4.1 se pueden observar varios vectores ubicando los puntos de conexión, todos ellos se describen a continuación:

- ${}^{B}\vec{P}_6$: Es el vector (pintado en azul) que apunta o ubica el punto P_6 visto desde el marco de referencia anclado a la base.
- ${}^{P}\vec{P}_6$: Es el vector (pintado en verde) que apunta al punto P_6, sin embargo a diferencia del anterior, está visto desde el marco de referencia anclado a la plataforma móvil.
- ${}^{B}\vec{B}_1$: Es el vector (pintado en azul) que ubica el punto de conexión B_1 visto desde el marco de referencia anclado a la base.
- ${}^{B}T_O$: Es el vector (pintado en cafe) que ubica el origen del marco de referencia anclado a la plataforma móvil visto desde el marco de referencia anclado a la base.
- ${}^{B}R^{P}$: Es la matriz de rotación que permite transformar las coordenadas del marco de referencia móvil al fijo definida como la representación de la orientación de un cuerpo a partir de los ángulos de Euler utilizada en SolidWorks, que consiste en una rotación α grados sobre el eje Z actual, después β grados sobre el eje X actual y por último γ grados sobre el eje Z actual (ver Fig. 4.2), esto con el fin de que los resultados obtenidos de manera analítica se puedan corroborar con dicho software.
- $\vec{S}_1$: Es el vector (pintado en rojo) colineal a la extremidad uno que representa al actuador, cuya magnitud es la longitud del mismo.

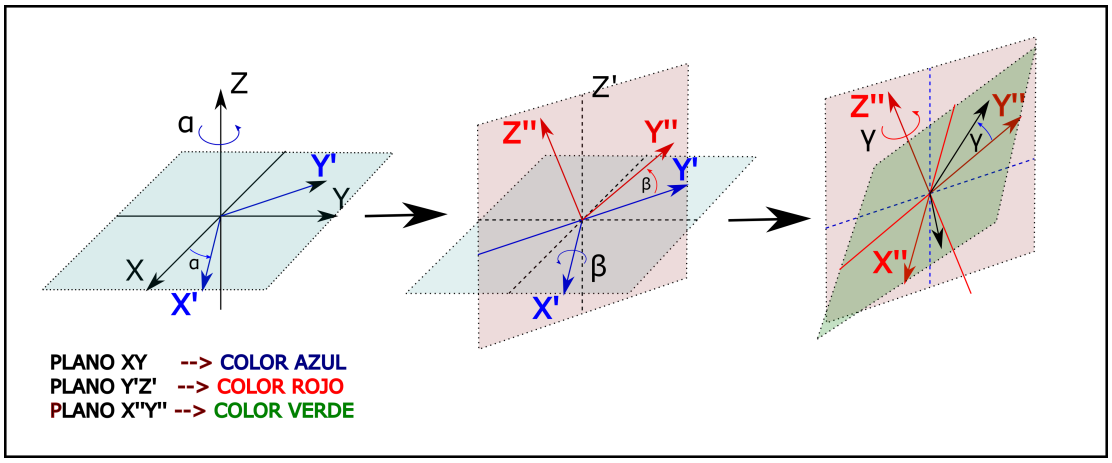


Figura 4.2: Sucesión de rotaciones para obtener la matriz de rotación ${}^{B}R^{P}$.

Nuevamente, de la Fig. 4.1 se puede observar que la ecuación de cerradura para el caso de la extremidad uno se puede obtener de la siguiente manera:

$$ {}^{B}\vec{P}_6 - {}^{B}\vec{B}_1 - \vec{S}_1 = 0 \tag{4.1} $$

Sin embargo, el vector ${}^{B}\vec{P}_6$ no es conocido dado que solo se ubicó el punto P_6 respecto del marco de referencia móvil en el capítulo anterior. Por tanto, para obtener el vector ${}^{B}\vec{P}_6$ se hace uso de la matriz de rotación ${}^{B}R^{P}$ y el vector de posición ${}^{B}\vec{T}_o$ para transformar las coordenadas al marco de referencia fijo, de modo que el vector ${}^{B}\vec{P}_6$ se define como sigue:

$$ {}^{B}\vec{P}_6 = {}^{B}\vec{T}_o + {}^{B}R^{P}\,{}^{P}\vec{P}_6 \tag{4.2} $$

Finalmente, sustituyendo la ecuación (4.2) en la ecuación (4.1) se obtiene la ecuación (4.3), expresada en términos de las longitud de la extremidad S_1, los puntos ${}^{P}P_6$ y ${}^{B}B_1$ definidos por la geometría de las plataformas y las variables a determinar ${}^{B}T_O$ y ${}^{B}R^{P}$ que describen la posición y orientación del robot respectivamente, definiendo por tanto la pose del mismo.

$$ {}^{B}\vec{T}_o + {}^{B}R^{P}\,{}^{P}\vec{P}_6 - {}^{B}\vec{B}_1 - \vec{S}_1 = 0 \tag{4.3} $$

Además, de la ecuación (4.3) se define una **ecuación general de cerradura** para cualquier extremidad, considerando solamente el par de puntos que están atados por dicha extremidad, expresada de la siguiente manera:

$$ {}^{B}\vec{T}_o + {}^{B}R^{P}\,{}^{P}\vec{P}_i - {}^{B}\vec{B}_i - \vec{S}_i = 0 \tag{4.4} $$

donde:

- $\vec{P}_i$: Es el punto de conexión i-ésimo en la plataforma móvil atado a la extremidad i-ésima.
- $\vec{B}_i$: Es el punto de conexión i-ésimo en la base fija atado a la extremidad i-ésima.
- $\vec{S}_i$: Es el vector a lo largo de la extremidad i-ésima.

Sin embargo, como se puede notar en la Fig. 4.3 los subíndices i, no coinciden para todos los puntos, por ejemplo para la extremidad uno (S_1) los puntos de conexión atados son B_1 y P_6 y no B_1 y P_1 como se esperaría, esto debido a la notación planteada en la Fig. 3.7 en el Capítulo 3. Por tanto, en la Tabla 4.1 se muestran los pares de puntos atados a cada una de las extremidades para evitar confusiones.

Tabla 4.1: Pares de puntos conectados a cada extremidad.

S_1	$B_1 \longrightarrow P_6$
S_2	$B_2 \longrightarrow P_1$
S_3	$B_3 \longrightarrow P_2$
S_4	$B_4 \longrightarrow P_3$
S_5	$B_5 \longrightarrow P_4$
S_6	$B_6 \longrightarrow P_5$

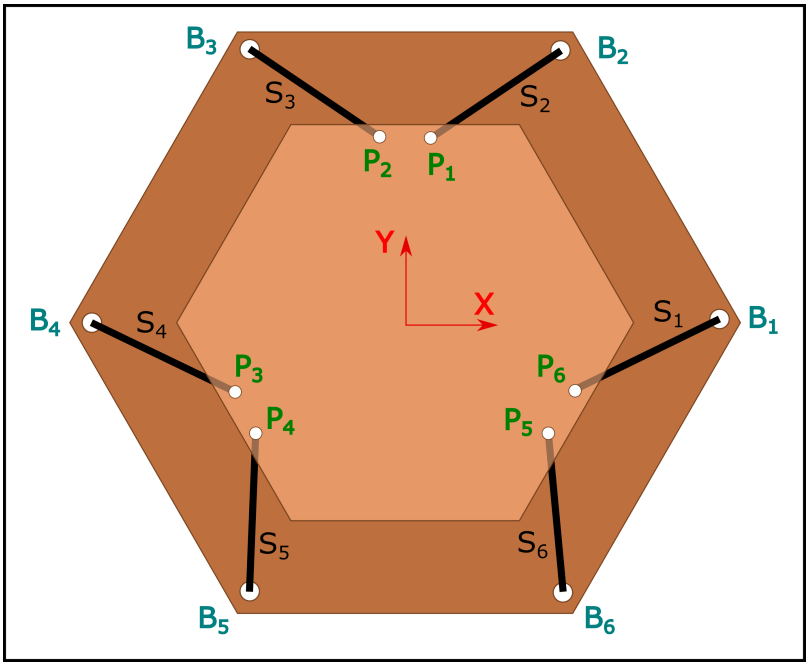


Figura 4.3: Forma de conexión de las extremidades para el robot propuesto en la tesis.

Una vez obtenida la ecuación general de cerradura, hay tomar en cuenta algunas consideraciones para resolver el sistema que se formará con las seis ecuaciones correspondientes a cada una de las extremidades.
La primer consideración es que no se conoce como tal el vector S_i, lo único que se tendrá como entradas del sistema serán sus respectivas longitudes, la magnitud de dicho vector. Por lo que se obtiene una expresión para la magnitud de S_i como sigue:

$$|S_i| = \sqrt{S_i \cdot S_i} \tag{4.5}$$

Además, se tiene una expresión alterna para S_i de la ecuación (4.4) al pasar dicha variable (S_i) al otro lado de la igualdad, una vez hecho esto se sustituye en la ecuación (4.5) obteniéndose una expresión en términos de la magnitud de S_i.

$$|S_i| = \sqrt{\left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_i - {}^B\vec{B}_i\right) \cdot \left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_i - {}^B\vec{B}_i\right)} \tag{4.6}$$

Reescribiendo la ecuación anterior:

$$\sqrt{\left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_i - {}^B\vec{B}_i\right) \cdot \left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_i - {}^B\vec{B}_i\right)} - |S_i| = 0 \tag{4.7}$$

Finalmente el sistema de ecuaciones a resolver se presenta a continuación:

$$\begin{aligned}
\sqrt{\left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_6 - {}^B\vec{B}_1\right) \cdot \left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_6 - {}^B\vec{B}_1\right)} - |S_1| &= 0 \\
\sqrt{\left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_1 - {}^B\vec{B}_2\right) \cdot \left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_1 - {}^B\vec{B}_2\right)} - |S_2| &= 0 \\
\sqrt{\left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_2 - {}^B\vec{B}_3\right) \cdot \left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_2 - {}^B\vec{B}_3\right)} - |S_3| &= 0 \\
\sqrt{\left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_3 - {}^B\vec{B}_4\right) \cdot \left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_3 - {}^B\vec{B}_4\right)} - |S_4| &= 0 \\
\sqrt{\left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_4 - {}^B\vec{B}_5\right) \cdot \left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_4 - {}^B\vec{B}_5\right)} - |S_5| &= 0 \\
\sqrt{\left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_5 - {}^B\vec{B}_6\right) \cdot \left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_5 - {}^B\vec{B}_6\right)} - |S_6| &= 0
\end{aligned} \tag{4.8}$$

Donde las variables a encontrar son los tres componentes de ${}^BT_O = \left[{}^BT_{O_X}, {}^BT_{O_Y}, {}^BT_{O_Z}\right]^T$ y la matriz de rotación ${}^BR^P$ compuesta por los ángulos α, β y γ, de lo que se concluye que el sistema de ecuaciones a resolver es un sistema no lineal determinado.

4.2. Solución de las ecuaciones de cinemática directa a partir de la función *fsolve* de MATLAB

Dada la complejidad para resolver el sistema de ecuaciones (4.8), se propone utilizar un método numérico de modo que se simplifique esta tarea, el método numérico a utilizar es el implementado en MATLAB a partir de la función *fsolve* bajo el algoritmo de Levenberg-Marquardt [30]. Aprovechando que la función ya está implementada, solamente restará desarrollar el sistema de ecuaciones mostrado en la ecuación (4.8) de un modo más explícito para poder ser programado en MATLAB.

Se comienza por expresar la ecuación general (4.4) de modo que se observen los elementos de cada una de las matrices y vectores como se observa en la ecuación (4.9).

$$\underbrace{\begin{bmatrix} {}^{B}T_{O_X} \\ {}^{B}T_{O_Y} \\ {}^{B}T_{O_Z} \end{bmatrix}}_{{}^{B}T_{O}} + \underbrace{\overbrace{\begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{R_{\alpha,Z}} \overbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & c\beta & -s\beta \\ 0 & s\beta & c\beta \end{bmatrix}}^{R_{\beta,X}} \overbrace{\begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{R_{\gamma,Z}}}_{{}^{B}R^{P}} \underbrace{\begin{bmatrix} {}^{P}P_{i_X} \\ {}^{P}P_{i_Y} \\ {}^{P}P_{i_Z} \end{bmatrix}}_{{}^{P}P_{i}} - \underbrace{\begin{bmatrix} {}^{B}B_{i_X} \\ {}^{B}B_{i_Y} \\ {}^{B}B_{i_Z} \end{bmatrix}}_{{}^{B}B_{i}} = S_i \tag{4.9}$$

Se obtiene el producto de las tres matrices correspondientes a los giros en los ángulos de Euler de la matriz ${}^{B}R^{P}$ obteniendo como resultado la ecuación (4.10):

$${}^{B}R^{P} = \begin{bmatrix} c\alpha c\gamma - s\alpha c\beta s\gamma & -c\alpha s\gamma - s\alpha c\beta c\gamma & s\alpha s\beta \\ s\alpha c\beta + c\alpha c\beta s\gamma & -s\alpha s\gamma + c\alpha c\beta c\gamma & -c\alpha s\beta \\ s\beta s\gamma & s\beta c\gamma & c\beta \end{bmatrix} \tag{4.10}$$

Una vez obtenida la matriz ${}^{B}R^{P}$ es posible obtener el vector ${}^{B}R^{P}{}^{P}P_i$ como sigue:

$${}^{B}R^{P}{}^{P}P_i = \begin{bmatrix} (c\alpha c\gamma - s\alpha c\beta s\gamma){}^{P}P_{i_X} - (c\alpha s\gamma + s\alpha c\beta c\gamma){}^{P}P_{i_Y} + (s\alpha s\beta){}^{P}P_{i_Z} \\ (s\alpha c\beta + c\alpha c\beta s\gamma){}^{P}P_{i_X} - (s\alpha s\gamma - c\alpha c\beta c\gamma){}^{P}P_{i_Y} - (c\alpha s\beta){}^{P}P_{i_Z} \\ (s\beta s\gamma){}^{P}P_{i_X} + (s\beta c\gamma){}^{P}P_{i_Y} + c\beta{}^{P}P_{i_Z} \end{bmatrix} \tag{4.11}$$

Se obtienen ecuaciones para cada uno de los tres componentes del vector S_i, como se observa a continuación:

$$S_{i_X} = {}^{B}T_{O_X} + (c\alpha c\gamma - s\alpha c\beta s\gamma){}^{P}P_{i_X} - (c\alpha s\gamma + s\alpha c\beta c\gamma){}^{P}P_{i_Y} + (s\alpha s\beta){}^{P}P_{i_Z} - {}^{B}B_{i_X} \quad (4.12)$$

$$S_{i_Y} = {}^{B}T_{O_Y} + (s\alpha c\beta + c\alpha c\beta s\gamma){}^{P}P_{i_X} - (s\alpha s\gamma - c\alpha c\beta c\gamma){}^{P}P_{i_Y} - (c\alpha s\beta){}^{P}P_{i_Z} - {}^{B}B_{i_Y} \quad (4.13)$$

$$S_{i_Z} = {}^{B}T_{O_Z} + (s\beta s\gamma){}^{P}P_{i_X} + (s\beta c\gamma){}^{P}P_{i_Y} + c\beta{}^{P}P_{i_Z} - {}^{B}B_{i_Z} \quad (4.14)$$

Retomando la ecuación (4.5) se obtiene una expresión alterna de ésta como sigue:

$$|S_i| = \sqrt{S_i \cdot S_i} = \sqrt{(S_{i_X} \cdot S_{i_X}) + (S_{i_Y} \cdot S_{i_Y}) + (S_{i_Z} \cdot S_{i_Z})} \quad (4.15)$$

Finalmente la función a programar será:

$$\sqrt{(S_{i_X} \cdot S_{i_X}) + (S_{i_Y} \cdot S_{i_Y}) + (S_{i_Z} \cdot S_{i_Z})} - |S_i| = 0 \quad (4.16)$$

Donde los términos S_{i_X}, S_{i_Y} y S_{i_Z} son los mostrados en las ecuaciones (4.12), (4.13) y (4.14) respectivamente.

4.2.1. Función *fsolve* de MATLAB

La función *fsolve* de MATLAB permite resolver ecuaciones y sistemas de ecuaciones no lineales de la forma $F(x) = 0$ donde x puede ser un vector o una matriz. Dicho de otro modo, la función obtiene las raíces del sistema de ecuaciones no lineales a resolver.

La sintaxis de la función es bastante simple.

```
1  % Sintaxis de la funcion fsolve
2
3        x = fsolve(fun,x0,options);
```

Donde x es el vector con los valores de las raíces y los tres parámetros que recibe la función son: "fun" que se refiere a la función que contiene el sistema de ecuaciones a resolver, "x_0" que representa al vector inicial para comenzar la iteración y finalmente "options" cuyo propósito es definir la forma en que se configura el algoritmo, definiendo el número máximo de iteraciones, el número máximo de evaluaciones de la función, el error de paro del algoritmo, entre otros parámetros.

- **Características de la función "fun" :**

  Para que la función *fsolve* funcione de manera adecuada, la función de entrada "fun" debe estructurarse de la siguiente manera. Considere por ejemplo, el siguiente sistema de ecuaciones:

$$\begin{array}{rcrcl} x & + & \sin(y) & = & 0 \\ y^2 & - & 2x & = & 0 \end{array} \tag{4.17}$$

  El primer paso a realizar es escribir el sistema de modo que todas las ecuaciones estén igualadas con cero. Una vez hecho esto, se procede a crear la función.

```
1  % Estructura de la funcion -fun-
2  % Donde x(1) --> x
3  % Donde x(2) --> y
4
5          function F = fun(x)
6
7                  F = [
8                  x(1), sin(x(2));
9                  x(2)^2, -2*x(1)
10                 ];
11
12         end
```

  Como se observa en el código anterior, se redefinen las variables de modo que formen parte de un solo vector, por tanto x se sustituirá por $x(1)$, mientras que y se sustituye por $x(2)$. Una vez hecho esto, se procede a estructurar la función a modo de matriz, donde las filas de la matriz estarán formadas por cada una de las ecuaciones y las columnas separarán a cada una de las variables. Se puede notar de igual manera que esta matriz se debe asignar a una variable F que será la operada por el algoritmo de la función *fsolve*.

- **Características del vector "x_0" :**

  El vector x_0 es bastante simple de definir debido a que la única condición con la que debe de cumplir es estar compuesto de tantos elementos como variables existan en el sistema de ecuaciones. Considerando el sistema de ecuaciones de ejemplo, se definirá x_0 como sigue (colocando los valores 1 y 2 de manera aleatoría):

```
1  % Vector con los valores iniciales para la iteracion
2
```

```
x0 = [1  2];
```

- **Características de la variable "options" :**

  La variable "options" como se mencionó anteriormente, almacena los datos de configuración del algoritmo implementado por la función *fsolve*. Para poder configurar el algoritmo a utilizar se hace uso de la función *optimset*.

```
% Configuracion del algoritmo a utilizar
% Se puede escoger entre:
% 'trust-region-dogleg'
% 'trust-region'
% 'levenberg-marquardt'

      options = optimset('Algorithm','levenberg-marquardt');
```

  De esta manera se tiene completa la configuración y definición de todos los parámetros para utilizar la función *fsolve* para resolver el sistema de ecuaciones (4.16). (El código implementado para la solución del sistema de ecuaciones se puede consultar en el Apéndice A.)

4.3. Análisis cinemático inverso

Para el análisis cinemático inverso la tarea resulta un tanto más sencilla, el problema radica en encontrar las longitudes de las extremidades siendo conocidas la posición (${}^{B}T_{O}$) y la orientación de la plataforma (${}^{B}R^{P}$).

Se procede de forma similar que con el análisis de cinemática directa, se obtienen las ecuaciones de cerradura, por lo que nuevamente a partir de la Fig. 4.1, se obtiene la ecuación general (4.4) repetida en la siguiente ecuación.

$$ {}^{B}\vec{T}_{o} + {}^{B}R^{P}\,{}^{P}\vec{P}_{i} - {}^{B}\vec{B}_{i} - \vec{S}_{i} = 0 \tag{4.18} $$

A diferencia del caso anterior, ahora la variable desconocida es S_i mientras que ${}^{B}T_{O}$, ${}^{B}R^{P}$ son conocidas, por lo que la solución de la cinemática inversa se reduce a despejar S_i y sustituir todos los valores conocidos y realizar las multiplicaciones de vectores y matrices. Por lo que se obtiene la siguiente ecuación:

$$ \vec{S}_{i} = {}^{B}\vec{T}_{o} + {}^{B}R^{P}\,{}^{P}\vec{P}_{i} - {}^{B}\vec{B}_{i} \tag{4.19} $$

Sin embargo, el valor útil para el presente caso es la magnitud del vector S_i que está definida como: $|S_i| = \sqrt{S_i \cdot S_i}$ por lo que se obtiene la siguiente expresión sustituyendo simplemente el valor de S_i obtenido en la ecuación (4.19):

$$|S_i| = \sqrt{\left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_i - {}^B\vec{B}_i\right) \cdot \left({}^B\vec{T}_o + {}^B R^{PP}\vec{P}_i - {}^B\vec{B}_i\right)} \tag{4.20}$$

A partir de este punto, se repite todo el procedimiento realizado en el análisis de cinemática directa al extender por componentes cada uno de los vectores involucrados en la ecuación, de manera que se pueda programar en MATLAB. Por tanto, únicamente se escribe en la ecuación (4.21) el resultado a programar.

$$|S_i| = \sqrt{(S_{i_X} \cdot S_{i_X}) + (S_{i_Y} \cdot S_{i_Y}) + (S_{i_Z} \cdot S_{i_Z})} \tag{4.21}$$

donde los términos S_{i_X}, S_{i_Y} y S_{i_Z} son los mostrados en las ecuaciones (4.12), (4.13) y (4.14) respectivamente. Ecuaciones expresadas en términos de las variables de posición del robot (${}^BT_O$) y los ángulos que definirán la orientación de la plataforma (α, β y γ), por lo que una vez programadas las series de ecuaciones, basta con sustituir dichos valores para obtener las longitudes (el código implementado se puede consultar en el Apéndice A).

4.4. Análisis cinemático de velocidad utilizando *Teoría de Tornillos*

Para realizar el análisis cinemático de velocidad, las ecuaciones de cerradura (4.18) tienen que ser diferenciadas con respecto al tiempo. En la literatura consultada se ha reportado que este procedimiento conlleva a una serie de dificultades para obtener una representación de entrada-salida, principalmente debido a las variables relacionadas con las juntas pasivas [1, 2, 3, 13, 18]. La teoría de tornillos representa una metodología alternativa para superar las dificultades mencionadas.

Para llevar a cabo el análisis cinemático de velocidad del robot es necesario definir tornillos para cada una de las juntas que componen el mecanismo, por tanto, de la configuración establecida para el robot 6-UPUR (ver Fig. 3.3), se determina que para cada una de las extremidades se definen seis tornillos, dos para cada junta universal, estos serán cooplanares y normales entre sí, un tornillo más para la junta prismática, en dirección del actuador lineal (dicho tornillo será el de más importancia pues está relacionado directamente con la entrada de movimiento) y finalmente uno más para la junta revoluta en la plataforma móvil justo después de la universal (ver Fig. 4.4).

El método es a grandes rasgos simple, el objetivo es definir los tornillos de modo que se ob-

tenga alguna línea o tornillo que sea recíproco de todos los tornillos con variables pasivas de modo que al realizar las formas de Klein éstas se anulen, dado que dos tornillos son recíprocos si y sólo si la forma de Klein entre ellos es igual con cero, es decir:

$$\{\$_1, \$_2\} = \{(\omega_1, v_1), (\omega_2, v_2)\} = \omega_1 \cdot v_2 + \omega_2 \cdot v_1 = 0 \tag{4.22}$$

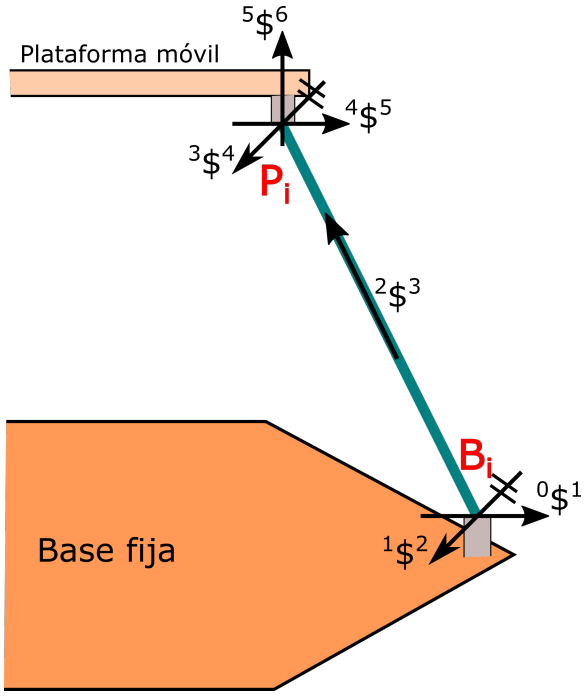


Figura 4.4: Diagrama general con los seis tornillos a utilizar para cada extremidad.

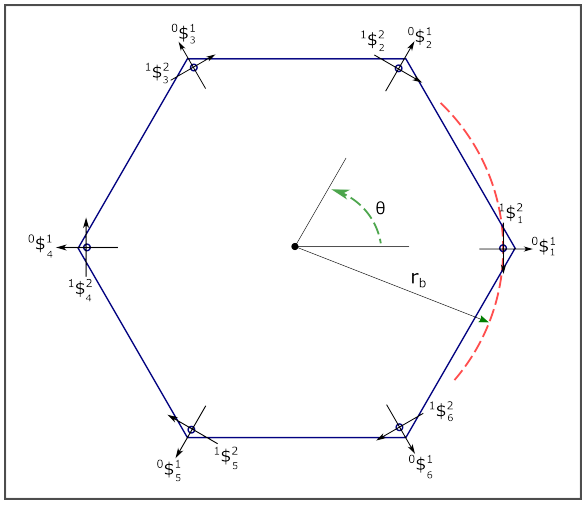


Figura 4.5: Definición de los tornillos para las juntas universales en la base.

Se comienza por definir los tornillos en las juntas universales en la base del robot. Para ello se hace uso de la Fíg. 4.5, se observa que el primer tornillo se define de manera radial variando su dirección de acuerdo al ángulo θ, por lo que el momento alrededor de cada uno de los eje es igual a cero, y la expresión matemática que describe a los tornillos en la parte más baja de cada una de las extremidades es la siguiente (para cada tornillo se presenta en la parte superior la definición de la parte primaria y a continuación el tornillo propiamente):

$$\begin{aligned} {}^{0}s_i^1 &= \begin{bmatrix} \cos(\theta \cdot (i-1)) \\ \sin(\theta \cdot (i-1)) \\ 0 \end{bmatrix} \\ {}^{0}\$_i^1 &= \begin{bmatrix} {}^{0}s_i^1 \\ {}^{0}s_i^1 \times -B_i; \end{bmatrix} \end{aligned} \tag{4.23}$$

De la ecuación (4.23) el momento alrededor del polo de referencia se obtiene del producto cruz con el vector de posición B_i definido así en el marco teórico para tornillos correspondientes a juntas revolutas (esto debido a que la junta universal se descompone en dos juntas revolutas).

A continuación, se define el tornillo en la dirección del actuador denominado ${}^{2}\$^{3}$ (ver Fig. 4.4), para obtener la dirección y por tanto la parte dual del tornillo se realiza la resta de los vectores P_i y B_i (puntos vistos desde el marco de referencia fijo a la base) de modo que se obtenga la dirección del actuador y se divide entre el módulo para volver unitario al vector, como se observa en la ecuación (4.24). Sin olvidar que los subíndices en los puntos P_i no coinciden con los de B_i (revisar puntos de conexión entre extremidades Fig. 4.3).

$$\begin{aligned} {}^{2}s_i^3 &= \tfrac{P_i - B_i}{|P_i - B_i|} \\ {}^{2}\$_i^3 &= \begin{bmatrix} \mathbf{0} \\ {}^{2}s_i^3 \end{bmatrix} \end{aligned} \tag{4.24}$$

Como se observa en la ecuación (4.24) la parte dual coincide con la dirección del tornillo, esto porque se trata de un tornillo para la junta prismática.
Una vez obtenidos los tornillos ${}^{2}\$_i^3$ y ${}^{0}\$_i^1$ se obtiene la parte primaria del tornillo ${}^{1}\$_i^2$ que corresponde a la junta revoluta en la parte superior de la junta universal, a partir del producto cruz entre ${}^{2}\$_i^3$ y ${}^{0}\$_i^1$ como se observa en la ecuación (4.25).

$$
\begin{aligned}
{}^{1}s_i^2 &= {}^{0}s_i^1 \times {}^{2}s_i^3 \\
{}^{1}\$_i^2 &= \begin{bmatrix} {}^{1}s_i^2 \\ {}^{1}s_i^2 \times -B_i \end{bmatrix}
\end{aligned}
\tag{4.25}
$$

Retomando la Fig. 4.4 se observa que los tornillos ${}^{1}\$_i^2$ y ${}^{3}\$_i^4$ son paralelos por lo que la parte primaria de ambos es la misma, variando únicamente el vector de posición para obtener el momento respecto del polo de referencia.

$$
\begin{aligned}
{}^{3}s_i^4 &= {}^{0}s_i^1 \times {}^{2}s_i^3 \\
{}^{3}\$_i^4 &= \begin{bmatrix} {}^{3}s_i^4 \\ {}^{3}s_i^4 \times -P_i \end{bmatrix}
\end{aligned}
\tag{4.26}
$$

A continuación se obtiene el tornillo correspondiente a la junta revoluta colocada sobre la plataforma móvil, se observa de la Fig. 4.4 que el tornillo siempre es normal a la superficie de la plataforma móvil, por lo que para obtener la dirección del mismo, se utilizan los vectores que se observan en la Fig. 4.6, bajo el siguiente procedimiento:

- Para obtener la dirección del tornillo ${}^{5}\$^{6}$ es necesario ubicar el punto de conexión ${}^{M}P_i$ y el consecutivo siguiendo el sentido contrario a las manecillas del reloj (${}^{M}P_{i+1}$). Por tanto, realizando el producto cruz de los vectores ${}^{M}P_i$ y (${}^{M}P_{i+1} - {}^{M}P_i$) se obtiene la parte primaria del tornillo en cuestión y para normalizarlo se divide entre su magnitud.
- Se debe notar que dichos vectores se encuentran medidos con respecto del marco de referencia atado a la base móvil, por lo que como se observa en la ecuación (4.27), para obtener la dirección respecto del marco de referencia fijo a la base, hace falta multiplicar por la matriz de rotación ${}^{B}R^{P}$.
- De igual manera que con los demás tornillos, para las juntas revolutas la parte dual se obtiene al realizar el producto cruz con el vector de posición P_i.

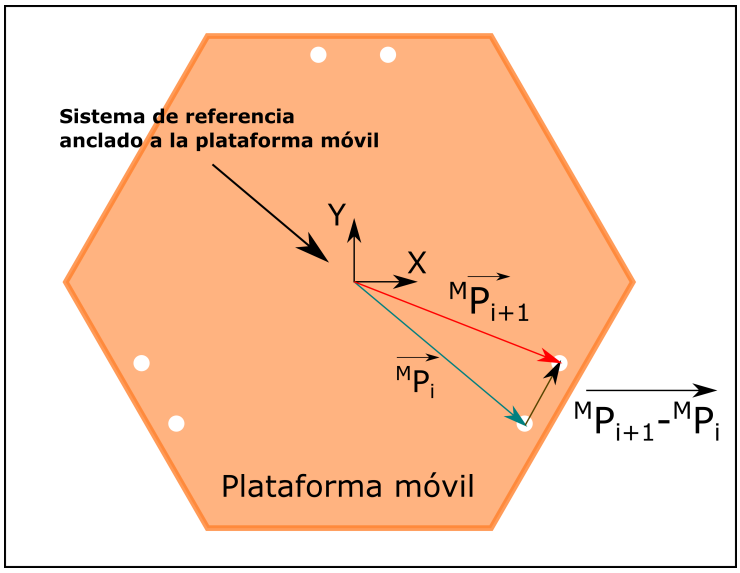


Figura 4.6: Vectores para obtener el tornillo ${}^{5}s_i^6$.

$$
\begin{aligned}
{}^{M5}s_i^6 &= \frac{{}^{M}P_i \times ({}^{M}P_{i+1} - {}^{M}P_i)}{|{}^{M}P_i \times ({}^{M}P_{i+1} - {}^{M}P_i)|} \\
{}^{5}s_i^6 &= {}^{B}R^{P}\ {}^{M5}s_i^6 \\
{}^{5}\$_i^6 &= \begin{bmatrix} {}^{5}s_i^6 \\ {}^{5}s_i^6 \times -P_i \end{bmatrix}
\end{aligned}
\tag{4.27}
$$

Finalmente, se obtiene la parte primaria del último de los tornillos simplemente del producto cruz entre las partes primarias de los tornillos ${}^{5}s^6$ y ${}^{3}s^4$, mientras que la parte dual surge del producto cruz con el vector de posición P_i como se observa en la ecuación (4.28).

$$
\begin{aligned}
{}^{4}s_i^5 &= {}^{5}s^6 \times {}^{3}s^4 \\
{}^{4}\$_i^5 &= \begin{bmatrix} {}^{4}s_i^5 \\ {}^{4}s_i^5 \times -P_i \end{bmatrix}
\end{aligned}
\tag{4.28}
$$

Como se mencionó anteriormente, el objetivo una vez expresados todos los tornillos, es hallar un tornillo recíproco de modo que al realizar las formas de Klein con el estado de velocidad se eliminen las variables pasivas. Revisando los tornillos planteados anteriormente se nota que ninguno cumple con la definición de reciprocidad, por lo que en la ecuación (4.29) se plantea una línea de Plücker que es recíproca a todos los tornillos anteriores con excepción del ${}^{2}\$_i^3$ permitiendo mantener solo las velocidades correspondientes a los actuadores, como se muestra a continuación.

$$
\begin{aligned}
{}^{2}\$^{3}_{r_i} &= \tfrac{P_i - B_i}{|P_i - B_i|} \\
{}^{2}\$^{3}_{r_i} &= \begin{bmatrix} {}^{2}\$^{3}_{r_i} \\ {}^{2}\$^{3}_{r_i} \times -P_i \end{bmatrix}
\end{aligned}
\tag{4.29}
$$

Se observa de la ecuación (4.29) que la línea de Plücker se interpreta como una junta revoluta en dirección del actuador que intersecta a todos los demás tornillos por lo que se concluye que todas los tornillos pertenecientes a juntas revolutas son coplanares a la línea de Plücker y por tanto recíprocas.

4.4.1. Estado de velocidad del robot

El estado de velocidad del robot se define como sigue:

$$
{}^{B}\mathbf{V}^{P}_{o} = {}_{0}\omega^{i}_{1}\ {}^{0}\$^{1}_{i} + {}_{1}\omega^{i}_{2}\ {}^{1}\$^{2}_{i} + {}_{2}\omega^{i}_{3}\ {}^{2}\$^{3}_{i} + {}_{3}\omega^{i}_{4}\ {}^{3}\$^{4}_{i} + {}_{4}\omega^{i}_{5}\ {}^{4}\$^{5}_{i} + {}_{5}\omega^{i}_{6}\ {}^{5}\$^{6}_{i} \tag{4.30}
$$

donde ${}^{B}\mathbf{V}^{P}_{o}$, es el estado de velocidad del punto O (origen del marco de referencia atado a la plataforma móvil) en el cuerpo P (plataforma móvil) respecto del cuerpo B (base fija) y las velocidades ${}_{0}\omega^{i}_{1}$, ${}_{1}\omega^{i}_{2}$, ${}_{3}\omega^{i}_{4}$, ${}_{4}\omega^{i}_{5}$ y ${}_{5}\omega^{i}_{6}$ corresponden a las velocidades angulares de las juntas universales y revoluta, relacionadas a sus respectivos tornillos. Solamente la velocidad ${}_{2}\omega^{i}_{3} = \dot{S}_i$ es la velocidad del actuador i-ésimo que resulta de interés para el análisis.

Aplicando las formas de Klein a ambos lados de la ecuación (4.30) se obtiene la ecuación (4.31), de la que se observan todos excepto el tornillo perteneciente a la junta prismática tachados, debido a que la forma de Klein entre ellos es cero y uno para la junta prismática, obteniendo como resultado final la ecuación (4.32).

$$
\begin{aligned}
\left\{ {}^{B}\mathbf{V}^{P}_{o}; {}^{2}\$^{3}_{r_i} \right\} = {} & \cancel{{}_{0}\omega^{i}_{1} \left\{ {}^{0}\$^{1}_{i}; {}^{2}\$^{3}_{r_i} \right\}} + \cancel{{}_{1}\omega^{i}_{2} \left\{ {}^{1}\$^{2}_{i}; {}^{2}\$^{3}_{r_i} \right\}} + {}_{2}\omega^{i}_{3} \left\{ {}^{2}\$^{3}_{i}; {}^{2}\$^{3}_{r_i} \right\} \\
& + \cancel{{}_{3}\omega^{i}_{4} \left\{ {}^{3}\$^{4}_{i}; {}^{2}\$^{3}_{r_i} \right\}} + \cancel{{}_{4}\omega^{i}_{5} \left\{ {}^{4}\$^{5}_{i}; {}^{2}\$^{3}_{r_i} \right\}} + \cancel{{}_{5}\omega^{i}_{6} \left\{ {}^{5}\$^{6}_{i}; {}^{2}\$^{3}_{r_i} \right\}}
\end{aligned}
\tag{4.31}
$$

$$
\left\{ {}^{B}\mathbf{V}^{P}_{o}; {}^{2}\$^{3}_{r_i} \right\} = {}_{2}\omega^{i}_{3} = \dot{S}_i \tag{4.32}
$$

finalmente, al considerar las seis extremidades se obtiene la siguiente expresión que relaciona entradas y salidas de manera matricial agrupando las ecuaciones para todas las extremidades:

$$
\mathbf{J}\boldsymbol{\Delta}\,{}^{B}\mathbf{V}^{P}_{o} = \dot{\mathbf{S}} \tag{4.33}
$$

Donde **J** es la matriz Jacobiana (o Jacobiano) cuyas columnas están constituidas por las líneas de Plücker con las que se realizaron las formas de Klein, como se observa a continuación:

$$\mathbf{J} = \begin{bmatrix} {}^{2}\$^{3}_{r_1} \\ {}^{2}\$^{3}_{r_2} \\ {}^{2}\$^{3}_{r_3} \\ {}^{2}\$^{3}_{r_4} \\ {}^{2}\$^{3}_{r_5} \\ {}^{2}\$^{3}_{r_6} \end{bmatrix} \tag{4.34}$$

Δ es el operador de polaridad y $\dot{\mathbf{S}}$ es un vector columna cuyas componentes son las velocidades de los actuadores, como se define a continuación:

$$\dot{\mathbf{S}} = \begin{bmatrix} \dot{S}_1 \\ \dot{S}_2 \\ \dot{S}_3 \\ \dot{S}_4 \\ \dot{S}_5 \\ \dot{S}_6 \end{bmatrix} \tag{4.35}$$

Finalmente, ${}^{B}\mathbf{V}_{o}^{P}$ es el estado de velocidad del robot cuyas componentes son las siguientes:

$${}^{B}\mathbf{V}_{o}^{P} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ v_x \\ v_y \\ v_z \end{bmatrix} \tag{4.36}$$

Por lo que, como se puede notar en el proceso de obtención del estado de velocidad del robot, se calculan los tornillos para cada condición y posición en particular. La ecuación (4.33) muestra la relación entre el estado de velocidad del robot (${}^{B}\mathbf{V}_{o}^{P}$) y el vector de velocidades de los actuadores ($\dot{S}$), dicha relación permite obtener el estado de velocidad siendo conocidas las velocidades de los actuadores o viceversa, en otras palabras se resuelve la cinemática directa e inversa de velocidad del mecanismo.

Capítulo 5

Instrumentación

En este capítulo se presentan los dispositivos electrónicos encargados de accionar los actuadores del robot, así como los sensores que obtienen las variables de posición y orientación de la plataforma móvil, haciendo énfasis en los requerimientos de las señales de control y los circuitos implementados para la comunicación entre todos los módulos y el correcto funcionamiento del robot en general.

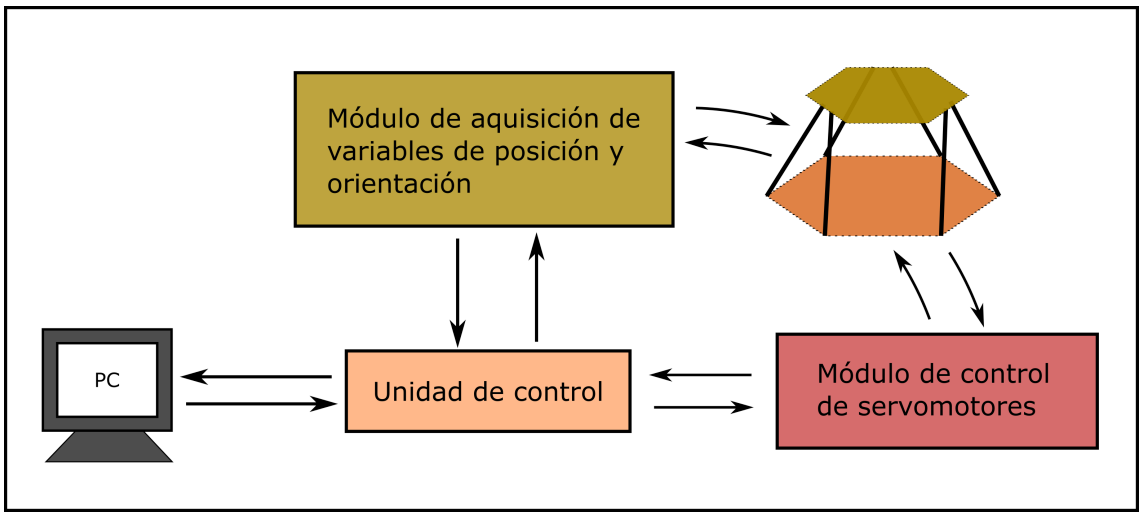


Figura 5.1: Estructura general de todos los componentes que conforman el sistema.

Como se observa en la Fig. 5.1, el sistema está estructurado en cuatro módulos principales, todos ellos en constante comunicación, atendiendo las indicaciones recibidas a través de la PC que fungirá como interfaz para el usuario.

- **PC:** Se encarga de procesar los datos de modo que envíe instrucciones al robot y reciba información respecto al estado del mismo.

- **Unidad de control:** Coordina la adquisición de datos y el control de los servomotores de modo que recibe las señales de la PC y se encarga de interpretarlas y dirigirlas al módulo correspondiente.
- **Módulo de control de servomotores:** Genera las señales adecuadas para la extensión y retracción de los servomotores.
- **Módulo de adquisición de datos:** Está constituido por el sensor de posición y orientación, básicamente es una unidad de medición inercial (IMU, por sus siglas en inglés).

5.1. Controlador pololu micromaestro 6

Cada uno de los servomotores requiere de una señal de modulación de ancho de pulso (PWM, por sus siglas en inglés) que varíe el ancho de pulso entre 1000 [μs] y 2000 [μs] pues estos límites representan o se interpretan por el servomotor como totalmente retraído (1000 [μs]) y totalmente extendido (2000 [μs]) como se observa en la Fig. 5.2 en la que además, se especifica el rango de frecuencias en las que opera el servomotor de manera adecuada.

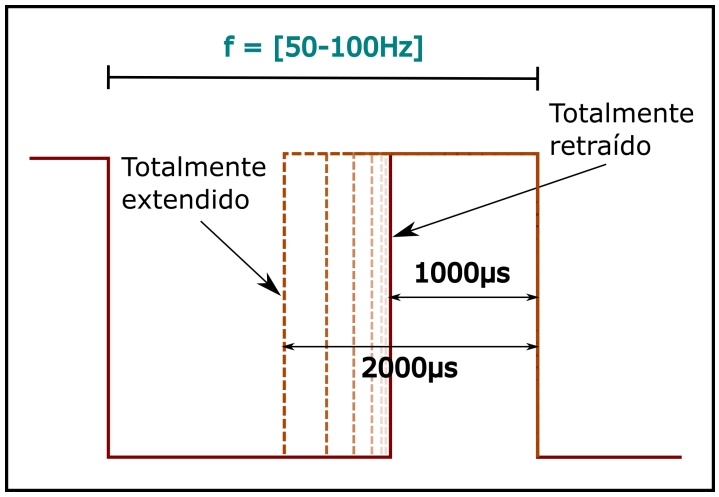


Figura 5.2: Características de las señal de control de los servomotores.

Por tanto, debido a que para el prototipo se requieren de seis extremidades, también se utilizan seis servomotores, se necesitan por igual seis señales de control para configurar las longitudes de las extremidades del robot. Esto genera la necesidad de obtener esas señales de alguna manera.
Una primer opción consistió en utilizar microcontroladores para generar las señales, sin embargo la mayoría de los microcontroladores permiten a lo más, tres señales PWM simultáneas por lo que se requerirían dos dispositivos, además la resolución que se podría alcanzar sería

bastante deficiente.
La segunda opción es adquirir un dispositivo externo que se encargue de la generación de las señales y que se pueda controlar con un microntrolador de modo que no se agreguen más elementos al sistema y se respete el modelo de la Fig. 5.1.
Dados los requerimientos en las señales de control, el controlador Micromaestro 6 de Pololu resulta el elemento perfecto para el control de los servomotores. Este controlador presenta las siguientes características que como se nota, se ajustan perfecto a las demandas del sistema [31].

Tabla 5.1: Características del controlador pololu.

Canales	6
Entradas analógicas	6
Entradas digitales	0
Ancho	2.16 [cm]
Largo	3.05 [cm]
Peso	3 [g]
Frecuencia	33-100 [Hz]
Ancho de pulso	64-3280 [μs]
Tamaño del script	1 [kB]

Como se observa de la Tabla 5.1 el controlador tiene justamente seis canales para los servomotores, además el rango de frecuencias y de ancho de pulso se ajusta a las características solicitadas por el servomotor. Por lo que una vez elegido el controlador, se procede a configurarlo de modo que funcione de manera óptima con los actuadores.

5.1.1. Configuración del controlador micromaestro 6:

El fabricante del controlador proporciona una aplicación bastante simple con la que se puede probar el correcto funcionamiento del mismo, la interfaz de dicha aplicación se observa en la Fig. 5.3 en la que se activa cada uno de los servomotores habilitando las casillas encerradas en el recuadro azul, mientras que la posición del servomotor se modifica en las casillas encerradas por el recuadro negro o deslizando las barras horizontales y se puede corroborar su posición a partir de las casillas encerradas en rojo.
Esta aplicación es útil pues permite configurar las características de los canales, además de la velocidad de comunicación serial con el módulo de control de una manera gráfica y bastante simple.

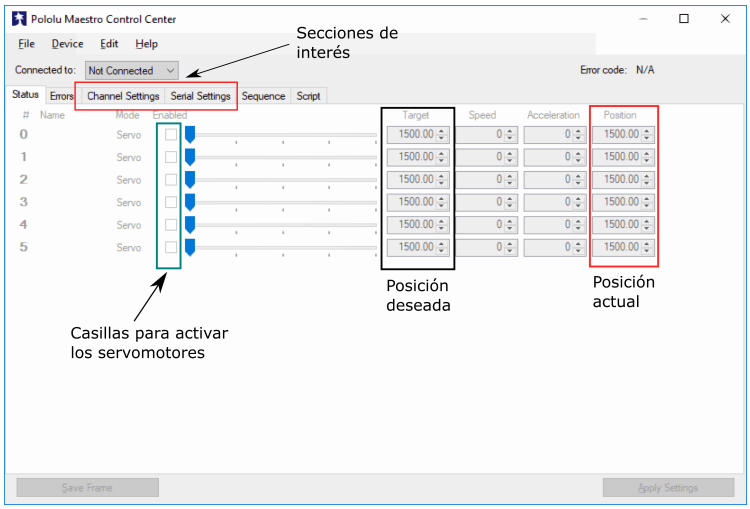


Figura 5.3: Interfaz de la aplicación de pruebas de Pololu.

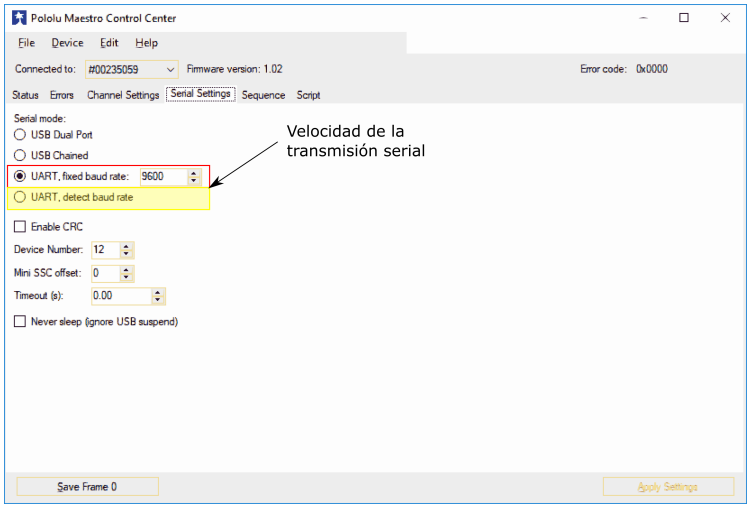


Figura 5.4: Configuración de la velocidad de transmisión.

Para la correcta comunicación, lo primero que se debe configurar es la velocidad de transmisión serial en la sección “Serial Settings”, dicha característica dependerá del rango de velocidades que soporte la unidad de control, en este caso 9600 bits por segundo. Como se observa en la Fig. 5.4, este valor se modifica en la casilla encerrada con rojo simplemente colocando el valor de transmisión en la casilla. Existe una segunda opción para casos en los que no se conoce la velocidad de transmisión de la unidad de control, se activa la casilla sombreada con amarillo con lo que el controlador detectará de manera automática la velocidad de transmisión, sin embargo no se garantiza el correcto funcionamiento del sistema bajo esta configuración.

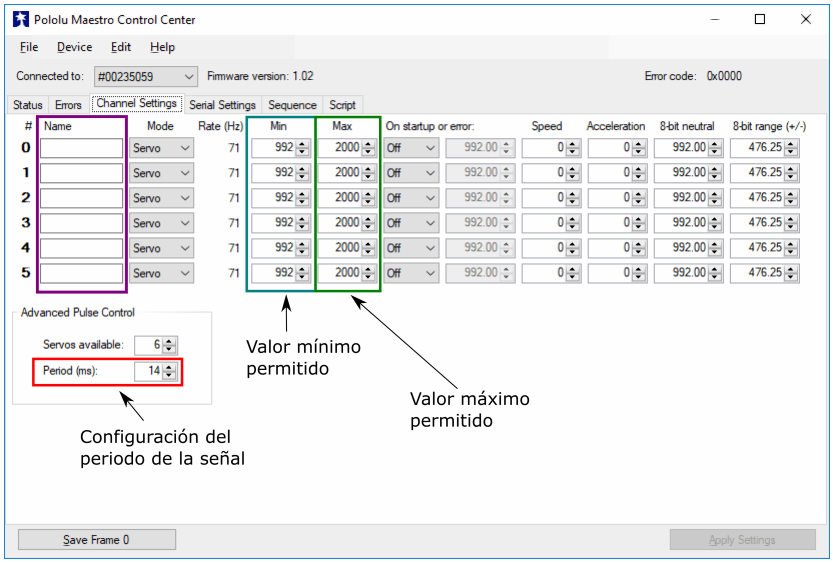


Figura 5.5: Configuración de las características de los canales.

Para adecuar las señales PWM del controlador se hace uso de la sección "channel settings". Como se puede observar en la Fig. 5.5, en esta sección se modificarán el periodo de la señal, el valor mínimo y máximo permisibles para la aplicación, e incluso se podrá agregar alguna etiqueta para distinguir a cada una de las señales para los servomotores. El periodo de la señal se definirá como 14 $[ms]$ de modo que la frecuencia sea de 71 [Hz] aproximadamente tratando de optar por un valor intermedio del rango de frecuencias permitido por los servomotores. De igual manera los valores mínimo y máximo se establecerán en 992 $[\mu s]$ y 2000 $[\mu s]$ respectivamente, acatando las características de los servomotores. En cuanto a las etiquetas no se consideran necesarias por lo que se dejan en blanco.

5.1.2. Dimensiones y pines de conexión

Una vez configurado el controlador, es necesario realizar las conexiones pertinentes para su correcto funcionamiento.

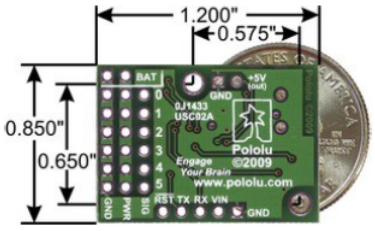


Figura 5.6: Dimensiones del controlador micromaestro 6.

En lo que respecta a las conexiones, en principio se usará solamente la conexión USB para configurar la comunicación y las características de los canales del controlador (procedimiento realizado anteriormente), una vez configurado, se conectarán en las dos terminales de alimentación para los servomotores una fuente de alimentación de 5[V] y en cada uno de los tres pines consecutivos se conectarán cada uno de los servomotores. De igual manera se conectarán las terminales **GND** y **VIN** a la misma fuente de 5[V] que alimenta los servomotores, mientras que las terminales **TX** y **RX** se conectarán a las terminales correspondientes de la unidad de control (ver Fig. 5.7).

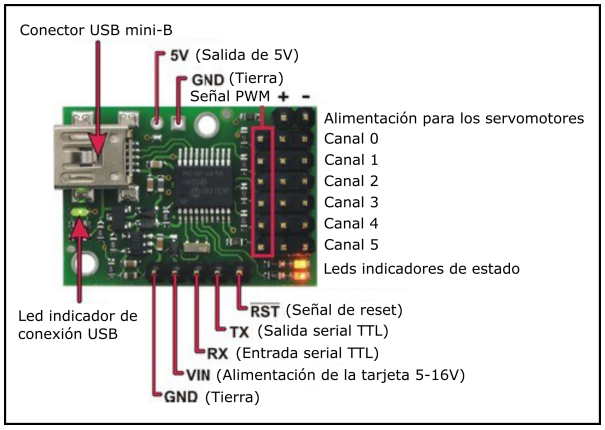


Figura 5.7: Pines de conexión del controlador micromaestro 6.

Una vez hechas las conexiones pertinentes y la configuración del controlador a partir de la aplicación proporcionada por el fabricante, simplemente falta integrar a la unidad de control el programa encargado del control y comunicación con el controlador (ver Sección 5.3).

5.2. Instrumentación del sensor LSM9DS1

El sensor LSM9DS1 es una unidad de medición inercial (IMU) de nueve grados de libertad, lo que significa que dentro de los dispositivos que se integran en la tarjeta se encuentran [32]:

Tabla 5.2: Resolución y elementos de la IMU.

	Resolución
Un giroscopio de tres ejes	± 245, 500 y 2000 [grados/s]
Un acelerómetro de tres ejes	± 2, 4, 8 o 16 [g]
Un magnetómetro de tres ejes	± 4, 8, 12 o 16 [gauss]

El conjunto de los tres elementos anteriores en un solo circuito integrado (ver Tabla 5.2) ofrece grandes ventajas, como el hecho de no afectar de manera significativa la masa e inercia de la plataforma con múltiples o muy pesados sensores, lo que permitirá obtener la orientación y posición de la plataforma móvil del robot una vez fijada la placa en la plataforma móvil, como se observa en la Fig. 5.9.

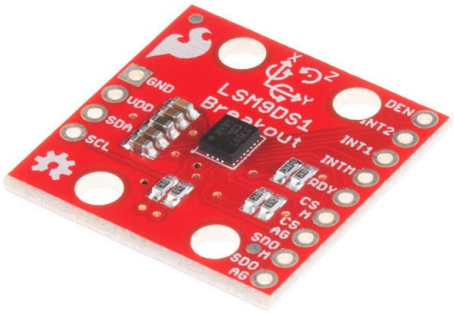


Figura 5.8: Unidad de medición inercial LSM9DS1.

Se aprovechará además, la versatilidad en comunicación que ofrece el dispositivo al soportar el protocolo I2C de modo que los únicos cuatro pines a utilizar del sensor serán los sombreados en amarillo en la Fig. 5.9. Conectando la términal **GND** a la tierra de la unidad de control, mientras que la terminal **VDD** se deberá conectar a una fuente de alimentación que no sobrepase los 3.6[V], para finalmente conectar las terminales **SDA** y **SCL** a los respectivos pines en la unidad de control de modo que se comuniquen a partir del protocolo I2C.

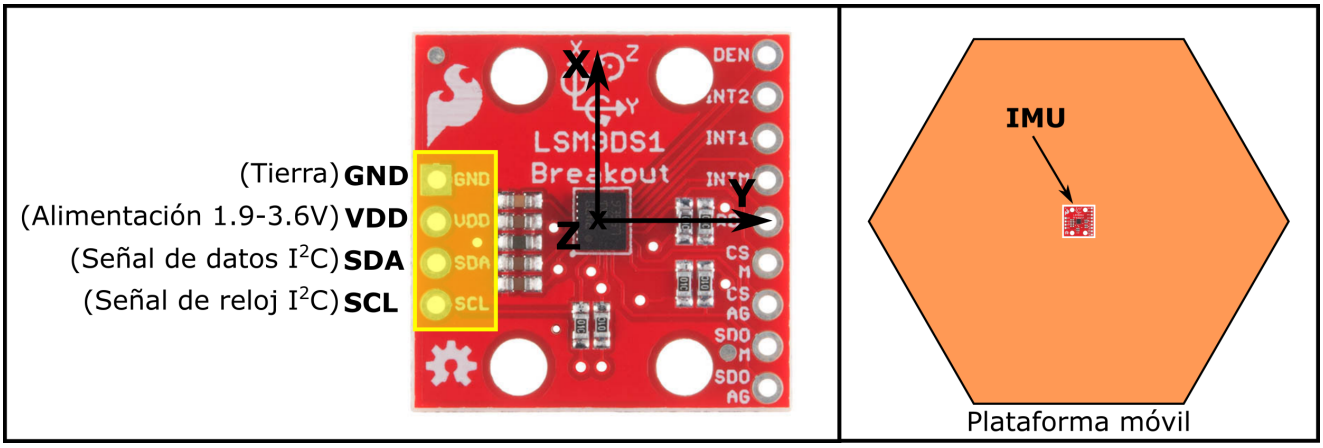


Figura 5.9: Terminales de conexión de la IMU.

Además de lo anterior, de la Fig. 5.9 se observa que el sistema coordenado o marco de referencia a partir del cual se realizan las mediciones se encuentra ubicado justamente en el centro de la placa, para ser más precisos en el centro del circuito integrado al centro de la placa.
Por tanto, el eje "x positivo" es hacia arriba, mientras que el eje "y positivo" es hacia la derecha. Por lo que respetando el hecho de que se trata de un marco de referencia derecho, el eje "z positivo" apunta hacia dentro de la hoja. Teniendo eso en cuenta, la velocidad angular y aceleración en cada uno de los ejes se considerará positivo en el mismo sentido que el marco de referencia.

5.3. Integración de los módulos a través de una tarjeta ArduinoNANO

Para integrar los dos módulos anteriores, se requiere de una unidad de control, como se observa en el diagrama general de la Fig. 5.1, por lo que se opta por una de las tarjetas de desarrollo de Arduino para fungir como unidad de control, esto debido a que para ambos dispositivos, el sensor y el controlador existen librerías que permiten operarlos de manera mucho más sencilla en comparación a utilizar un microcontrolador como unidad de control y acceder a cada uno de los registros de control de los módulos de servomotores y de la IMU, sin mencionar que se debía programar y establecer la comunicación de una manera mucho más complicada.

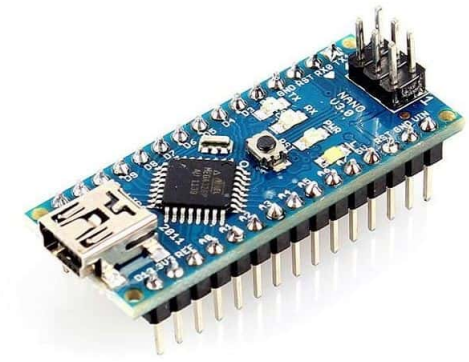


Figura 5.10: Tarjeta ArduinoNANO.

De las múltiples tarjetas que ofrece Arduino se selecciona la versión NANO (ver Fig. 5.10) debido al tamaño compacto de la tarjeta y las características que presenta, de las cuales se presentan las principales a continuación [33, 34]:

Tabla 5.3: Características de la tarjeta ArduinoNANO.

Microcontrolador	ATmega328
Voltaje de operación	5 [V]
Voltaje de entrada (recomendado)	7-12 [V]
Voltaje de entrada (límites)	6-20 [V]
Pines digitales de entrada y salida	14 (de los que 6 son PWM)
Pines con entrada analógica	8
Corriente (CD) permisible por pin	40 [mA]
Memoria Flash	30 [kB]
SRAM	2 [kB]
EEPROM	1 [kB]
Frecuencia de operación	16 [MHz]
Dimensiones	18.54 [mm] × 43.18 [mm]

Además de las características mencionadas en la Tabla 5.3, la tarjeta posee los protocolos de comunicación USART, I2C y SPI, que se refieren a la comunicación serial, el protocolo de comunicación por dos líneas I2C y el puerto para una interfaz serial (SPI) respectivamente. De las cuales se usarán la USART para comunicar con el driver de los motores y el protocolo I2C para comunicar con la IMU, de modo que se pueda acceder a ambos dispositivos sin desconectarse del otro.

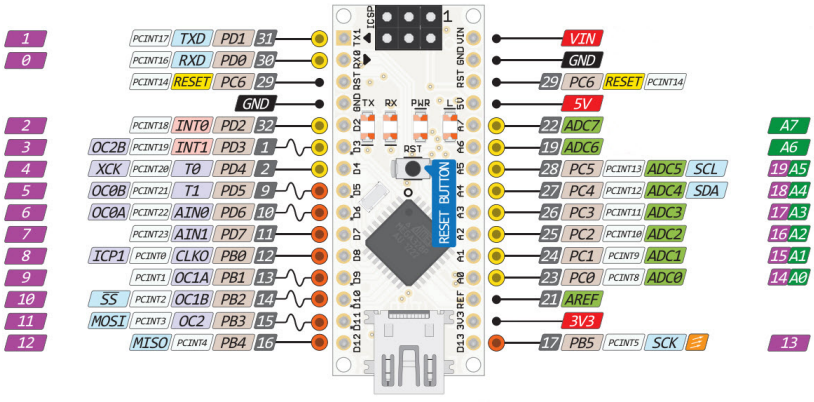


Figura 5.11: Etiquetas y funciones de cada uno de los pines del ArduinoNANO.

Por tanto, se deben ubicar los pines de la tarjeta que se requieren para cada uno de los protocolos, haciendo uso de la Fig. 5.11 se observan las funciones y etiquetas de cada uno de los pines. Para el protocolo serial se definen los pines 10 y 11 (coloreadas en morado) como las señales **RX** y **TX** respectivamente, mientras que para el protocolo I2C se utilizan los pines definidos por las etiquetas **SDA** y **SCL** (coloreadas en azul).

Algunos otros pines útiles son los etiquetados con **GND** (coloreada en negro) y **3V3** (coloreada en rojo) para conectar la referencia de todos los circuitos y la alimentación de la IMU respectivamente. Mientras que para comunicar con la computadora se utilizará el puerto USB mini-B, además la PC fungirá como fuente de alimentación de la tarjeta ArduinoNANO.

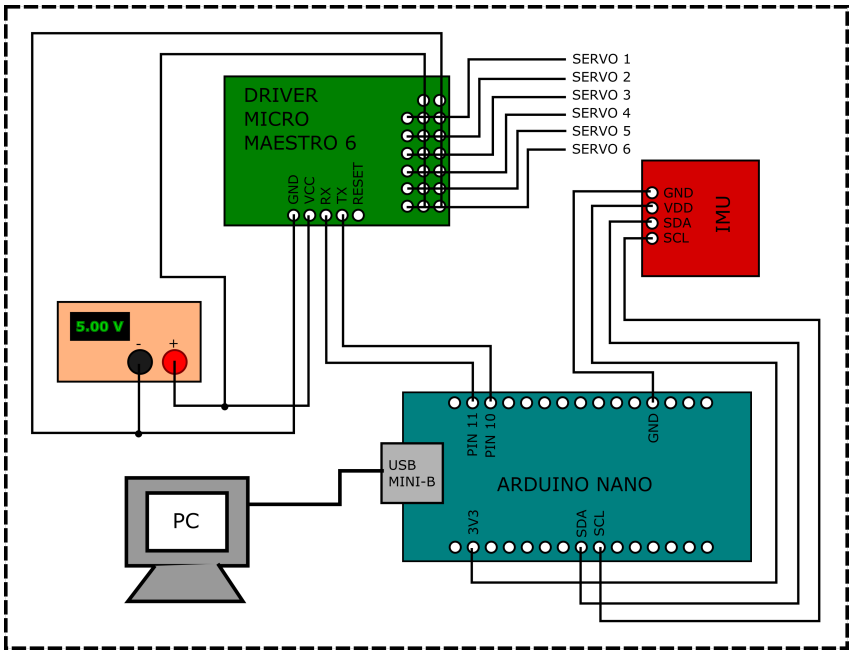


Figura 5.12: Conexiones entre cada uno de los dispositivos.

Una vez establecidos los pines a utilizar en cada una de las tarjetas, el último paso para terminar la instrumentación del sistema es la conexión de todos los módulos, que como se puede observar en la Fig. 5.12 el módulo de control consiste del arduinoNANO, mientras que el módulo de control de los servomotores por el driver y finalmente el módulo que adquisición de datos por la IMU.

Para poder establecer la comunicación entre la PC y la tarjeta Arduino y que ésta a su vez se comunique con los dos dispositivos externos, se implementó un programa a partir de la IDE de Arduino que se puede consultar en el Apéndice B.

5.3.1. Interfaz de operación

Dado que uno de los propósitos del robot será el acercamiento de los estudiantes a plataformas paralelas, se desarrolló una interfaz para controlar y accionar los movimientos del robot de modo que se pueda operar de una manera más sencilla e intuitiva.
La interfaz consta de siete paneles principales, como se muestra en la Fig. 5.13, descritos en la lista a continuación:

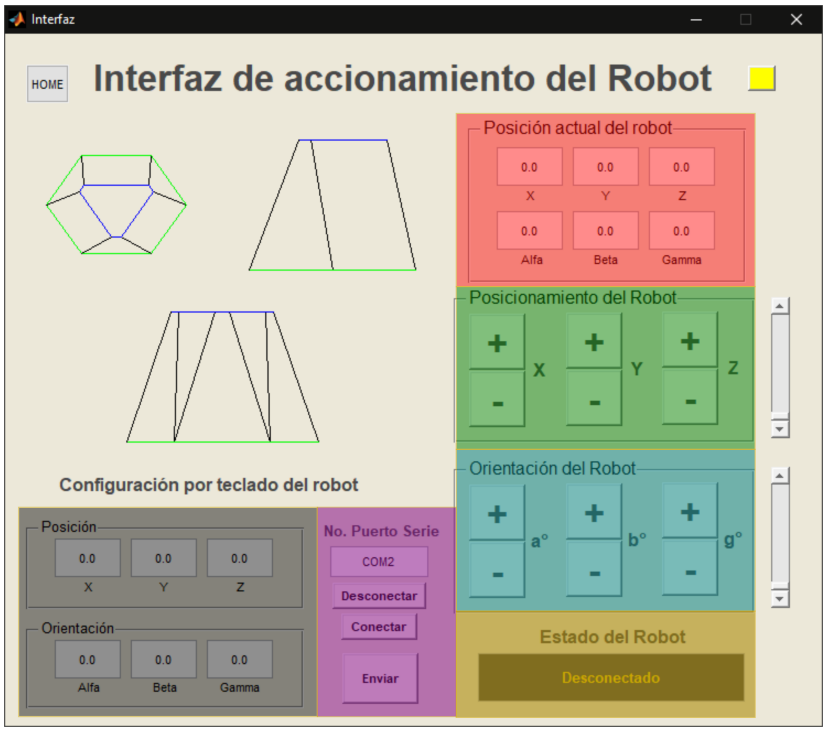


Figura 5.13: Interfaz diseñada para operar el robot.

- **Posición actual del robot:** Describe la posición y orientación actual del robot, dichos datos son leídos desde la IMU y desplegadas en la interfaz con cada envío de datos. Se despliegan las coordenadas en x, y y z y los angulos correspondientes a "yaw", "pitch" y "roll" (coloreada de rojo).

- **Posicionamiento del robot (x, y, z):** Permite modificar la posición en x, y y z del robot a partir de los botones "más" y "menos", incrementando y decrementando la posición actual respectivamente. Además, con la barra de desplazamiento a la derecha se aumenta el paso de los incrementos, en lugar de incrementar de uno en uno, generar incrementos de diez en diez incrementando así sucesivamente entre más arriba se ubique la barra de desplazamiento (coloreado de verde).

- **Orientación del robot (α, β, γ):** Permite modificar los ángulos α, β y γ correspondientes a los ángulos de Euler, de igual manera que la posición se incrementa y decrementa con los botones "más" y "menos" respectivamente y de igual manera aumentar el paso de incrementos al mover la barra de desplazamiento a la derecha (coloreado de azul).

- **Estado del robot:** Indica el estado en el que se encuentra el robot, es decir, si se encuentra conectado a la PC o no, si la última posición es alcanzable o no, todo se muestra en el recuadro en negro (coloreado de amarillo).

- **Conexión del robot al puerto serie:** Permite conectar el robot a la PC a partir del puerto serie asignado al arduino NANO, se recomienda revisar el número en el administrador de dispositivos y colocarlo en la casilla en lugar de "COM2". Finalmente, el tercer botón con la etiqueta "Enviar" envía los valores capturados en las casillas a la izquierda de modo que el robot alcanza dicha posición si se encuentra dentro del espacio de trabajo (coloreado de morado).

- **Posición y orientación ingresada por teclado:** Permite ingresar alguna posición precisa a la que se desea llevar la plataforma del robot, las tres casillas superiores corresponden a la posición y las tres inferiores a la orientación, nuevamente los ángulos se refieren a los ángulos de Euler utilizados durante todo el trabajo. Finalmente, con el botón enviar descrito en el punto anterior, se envían los valores y el robot se moverá o no dependiendo de si la posición es alcanzable o no (coloreado de negro).
  NOTA: Se recomienda tener cuidado con posiciones muy alejadas unas de otras pues debido a que no se tiene control de la velocidad de accionamiento de los actuadores puede llegar a encontrarse en puntos singulares y desestabilizar el movimiento del robot.

- **Vista preliminar de la pose del robot:** Obtiene de manera gráfica una pose estimada del robot en el espacio de trabajo (se representa en la gráfica en la esquina superior izquierda).

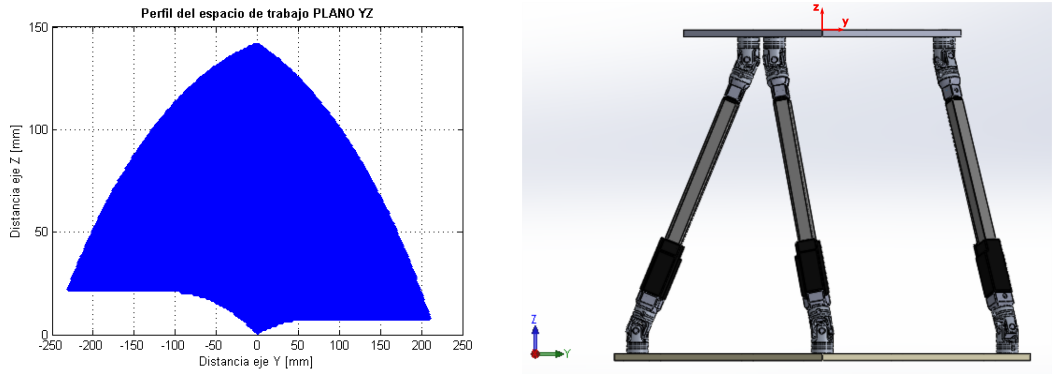
Capítulo 6

Resultados

A continuación se presentan los resultados obtenidos a partir de las pruebas realizadas en el prototipo construido, desde el espacio de trabajo en el que puede operar el robot, hasta la comparación de los resultados de cinemática directa e inversa.

6.1. Determinación del espacio de trabajo del robot

Para determinar el espacio de trabajo del robot, se implementaron un par de programas en MATLAB para iterar entre todas las posiciones alcanzables del robot y obtener los perfiles límite para valores constantes de Z. De las Figs. 6.1, 6.2, 6.3 y 6.4 se observa que el espacio de trabajo alcanzable tiene una forma más o menos cónica, todas las gráficas representan el punto central de la plataforma móvil y están referenciadas a un sistema coordenado fijo justo en el centro de la plataforma móvil coloreada de rojo como se observa en todas las figuras a la izquierda, que corresponde a la posición de la plataforma cuando los actuadores están totalmente retraidos.

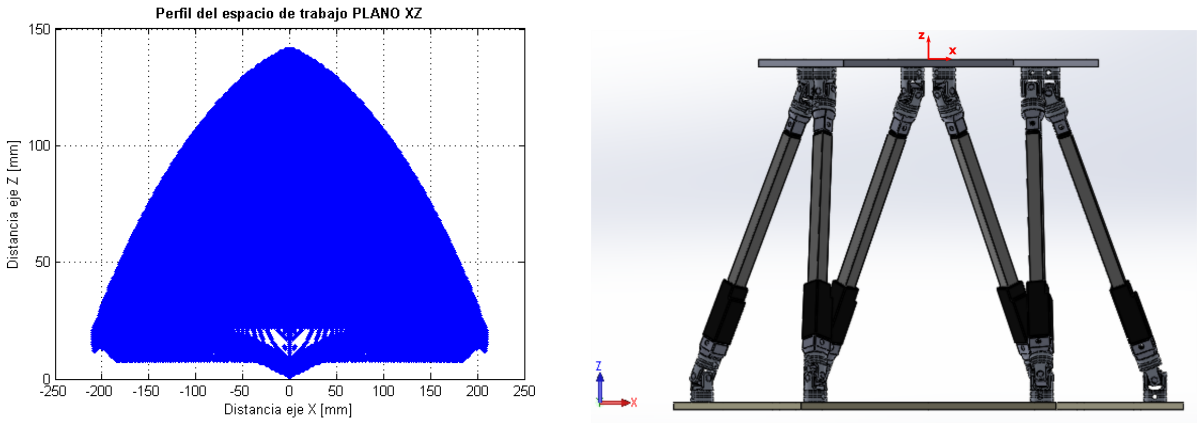


(a) Vista de perfil del espacio de trabajo del robot (PLANO YZ).

(b) Perfil del robot en el plano YZ

Figura 6.1: Espacio de trabajo del robot en el plano YZ.

De la Fig. 6.1 a) se observa el perfil del espacio de trabajo visto desde el plano YZ, notando de inmediato que el perfil no es simétrico respecto del eje Z, esto debido a la configuración y colocación de las extremidades del robot. Como se observa en la Fig. 6.1 b), la extremidad del extremo izquierdo está más inclinada que la de la derecha, por lo que cuando el robot intente desplazarse hacia la izquierda tendrá que levantarse mucho más que del lado izquierdo compensando la longitud mínima que posee el actuador lineal. Sin embargo, se conserva la característica de conicidad conforme la altura aumenta.



(a) Vista de perfil del espacio de trabajo del robot (PLANO XZ).

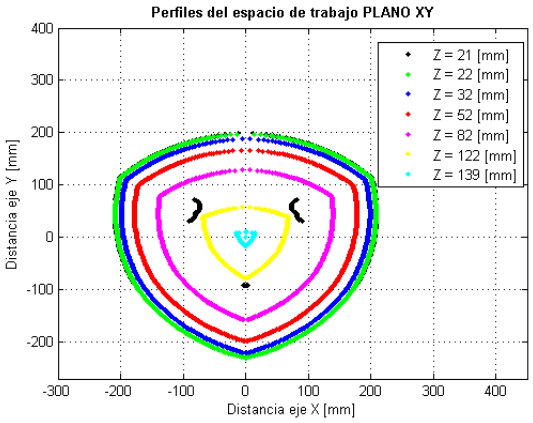
(b) Perfil del robot en el plano XZ

Figura 6.2: Espacio de trabajo del robot en el plano XZ.

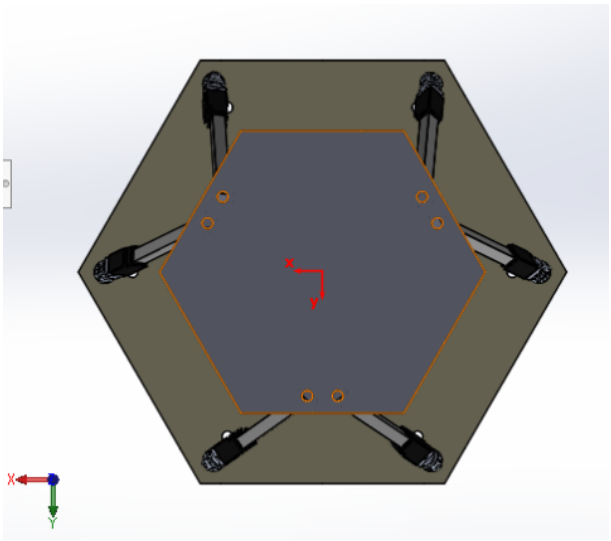
Mientras que del perfil del espacio de trabajo mostrado en la Fig. 6.2 a) se puede observar que es simétrico respecto del eje Z, presentando de igual manera la forma cónica particular del robot conforme aumenta la altura. La simetría en el perfil del espacio de trabajo se debe

a que el arreglo de extremidades es de igual manera simétrico en ese plano, por lo que los desplazamientos alcanzables son los mismos en ambos lados como se muestra en la Fig. 6.2 b).

En la Fig. 6.3 se muestran los perfiles del espacio de trabajo y del robot sobre el plano XY, sin embargo a diferencia de los perfiles anteriores, solo se muestran algunos para distintos valores de Z constantes de modo que pueda apreciarse la reducción del área de movimiento conforme se incrementa la altura.



(a) Vista de perfil del espacio de trabajo del robot (PLANO XY).



(b) Perfil del robot en el plano XY

Figura 6.3: Espacio de trabajo del robot en el plano XY.

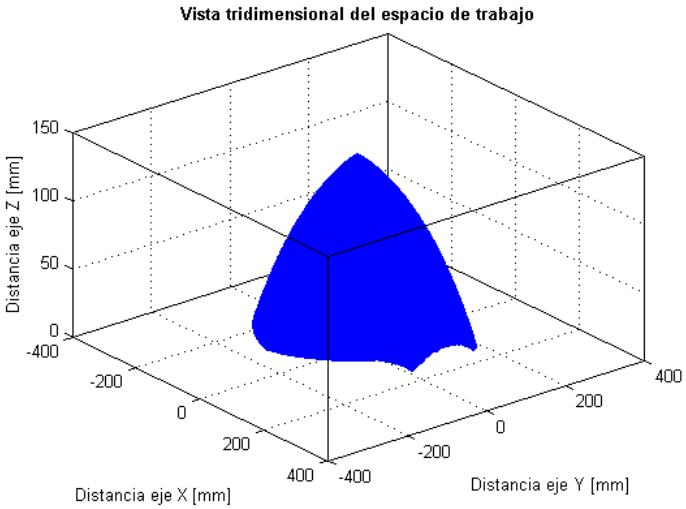


Figura 6.4: Vista tridimensional del espacio de trabajo.

Finalmente, en la Fig. 6.4 se puede observar una vista tridimensional del espacio de trabajo del robot. De la que nuevamente se nota la forma cónica del espacio de trabajo, esto debido a que en el punto más alto las extremidades están totalmente extendidas impidiendo el movimiento sobre el plano XY.

6.2. Validación del análisis cinemático

Para la validación del análisis cinemático de posición realizado, se propone el método cíclico mostrado en la Fig. 6.5 que consiste básicamente en proponer una posición deseada para la plataforma móvil del robot, resolver la cinemática inversa de acuerdo al procedimiento establecido en el Capítulo 4 para obtener las longitudes requeridas de los actuadores y, posteriormente estas longitudes usarlas como entrada para la cinemática directa, con lo que se obtiene la pose real del manipulador. La validez de los resultados se obtiene mediante la comparación de la posición deseada y la posición real alcanzada por el manipulador.

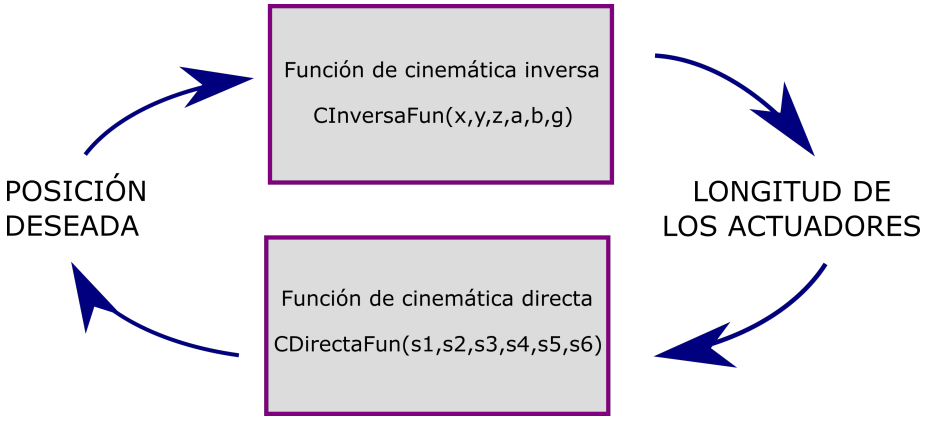


Figura 6.5: Diagrama del proceso para validar el análisis cinemático.

6.2.1. Validación de puntos en el espacio

Las funciones de cinemática inversa y directa son las obtenidas del análisis realizado en el Capítulo 4, además para más detalle de las funciones se puede consultar el código anexado en el Apéndice A. De manera general, la función de cinemática inversa *CinversaFun* recibe como entrada los valores de posición y orientación deseados para la plataforma, arrojando como salida un vector con seis elementos correspondientes a las longitudes de los actuadores en cada extremidad, comenzado por la extremidad marcada como S_1 y así sucesivamente.
Mientras que la función de cinemática directa, recibe como entrada las longitudes de los actuadores como un vector de seis elementos (esto para trabajar acorde a la salida de la función

de cinemática inversa) y como salida arroja la posición y orientación de la plataforma que se alcanza bajo las longitudes dadas. Por tanto, una vez realizado todo el ciclo, el resultado de la cinemática directa debe coincidir o ser bastante cercano a la posición y orientación deseadas en el principio. A continuación se muestran en la Tabla 6.1, diez puntos seleccionados de manera aleatoria con los que se verificó el correcto funcionamiento de las soluciones cinemáticas.

Tabla 6.1: Resultados obtenidos de la validación cinemática.

Pose deseada [mm,grados]	**Resultado cinemática directa [mm,grados]**	**Porcentaje de error**
[10,10,50,0,0,0]	[10.0,9.9,49.9,-0.1,-0.0,0.1]	0.0 %
[10,10,50,10,3,20]	[10.3,13.5,50.6,15.4,3.5,14.3]	14.0 %
[0,0,0,0,0,0]	[-0.0,-0.0,-0.1,-0.0,-0.1]	0.0 %
[1,-10,63,-20,0,0]	[1.0,-10.0,62.9,-20.0,-0.0,0.0]	0.0 %
[-10,-13,87,-12,-4,-12]	[-10.1,-10.8,87.5,-9.2,-3.5,-14.5]	4.5 %
[-4,1,43,19,5,10]	[-3.8,2.4,43.3,20.0,5.3,8.8]	24.0 %
[2,4,10,1,1,1]	[2.0,3.9,10.0,1.0,1.0,1.0]	0.0 %
[18,-12,57,13,10,-9]	[17.8,-17.0,55.8,14.0,8.9,-10.0]	8.0 %
[-6,2,80,-4,10,10]	[-6.0,1.9,80.0,-4.0,10.0,10.0]	0.0 %
[10,-10,80,0,-20,0]	[10.0,-10.0,80.0,-0.0,-20.0,0.0]	0.0 %

De la Tabla 6.1 se observa que al utilizar la función *fsolve* de MATLAB, la solución a las ecuaciones de cinemática directa arrojan resultados sumamente cercanos a los deseados, observando en la mayoría de los puntos un error promedio de 0.0 %, por lo que se puede concluir que la solución de cinemática directa es correcta.

Para el cálculo del error, se procedió de la siguiente manera:

1. Se calculó el error relativo entre la coordenada deseada y la obtenida a partir de la solución de la cinemática directa bajo la siguiente fórmula:

$$\epsilon = \frac{X_{deseada} - X_{obtenida}}{X_{deseada}} * 100 \tag{6.1}$$

2. Una vez obtenidos los errores para cada uno de las coordenadas, se procedió a obtener un promedio de cada una de las posiciones deseadas. Por ejemplo, se obtuvo el error de ϵ_x,ϵ_y,ϵ_z,ϵ_α,ϵ_β,ϵ_γ que corresponden a los errores para la coordenada X, la coordenada Y, la coordenada Z y los ángulo α, β y γ respectivamente. Por lo que el porcentaje de error mostrado en la tabla obedece la siguiente ecuación:

$$\text{Porcentaje de error} = \frac{\epsilon_x + \epsilon_y + \epsilon_z + \epsilon_\alpha + \epsilon_\beta + \epsilon_\gamma}{6} \tag{6.2}$$

No obstante de los resultados obtenidos, se observa de igual manera que para cuatro de los diez puntos el valor del error es diferente de cero y para dos de los casos de un valor significativamente alto, esto debido a que el algoritmo para resolver de manera numérica las ecuaciones no converge a una solución correcta debido a las condiciones iniciales del método, ya que para todos los puntos las condiciones iniciales permanecen fijas, sin embargo, la pose obtenida como resultado no es del todo errónea pero consta de coordenadas que si varían de la posición solicitada hasta en 3 [mm], pero solo es una o a lo más dos coordenadas de las seis que constituyen la pose, por lo que de manera general se concluye que la cinemática es correcta para la mayoría de los puntos.

6.2.2. Validación de trayectorias en el espacio

Una segunda forma en la que se valida el análisis cinemático de posición, es a partir del uso de trayectorias, comparando los recorridos deseados y los alcanzados a partir de la solución de la cinemática directa, aportando más evidencia del correcto funcionamiento de las soluciones de cinemáticas directa e inversa para la mayoría de poses. En principio, se comienza con desplazamientos bajo una orientación constante, variando parámetros de posición únicamente.

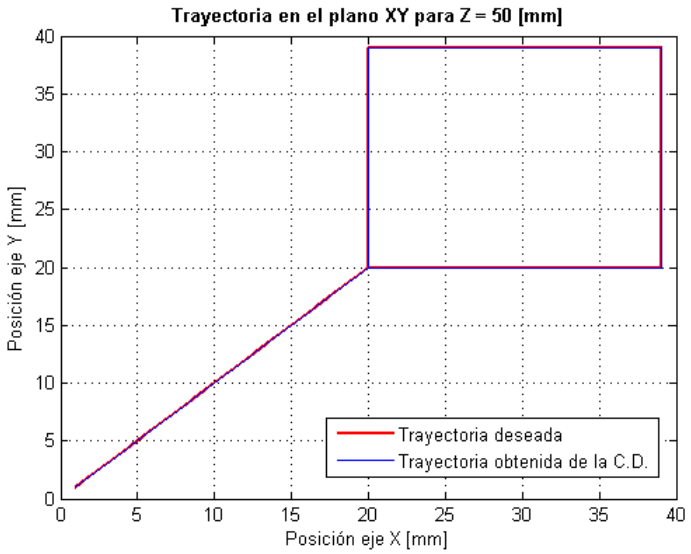


Figura 6.6: Trayectoria realizada en el plano XY con Z=50[mm].

Continuando con el método de validación mostrado en la Fig. 6.5, se comienza por una trayectoria bastante simple sobre el plano XY, es decir se varió las posiciones de XY y se mantuvo constante Z = 50 [mm], obteniendo el resultado mostrado en la Fig. 6.6, de la que se observa que las trayectorias concuerdan e incluso se superponen por lo que la solución es correcta.

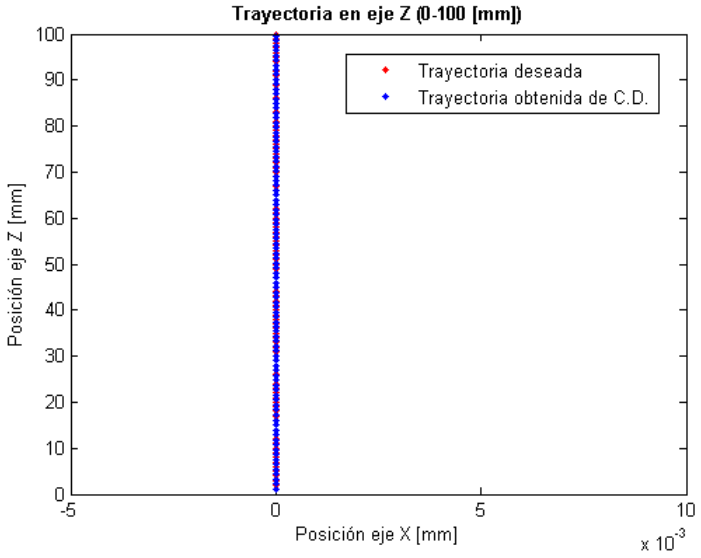


Figura 6.7: Trayectoria realizada en el plano XZ con Y = 0 [mm] y X = 0 [mm].

De la Fig. 6.7 de igual manera se concluye rápidamente que el análisis cinemático es correcto pues ambas trayectoria se superponen. Así mismo, ocurre con una trayectoria sinusoidal para el eje Z, mientras se mantienen fijos X y Y iguales a cero como se observa en la Fig. 6.8.

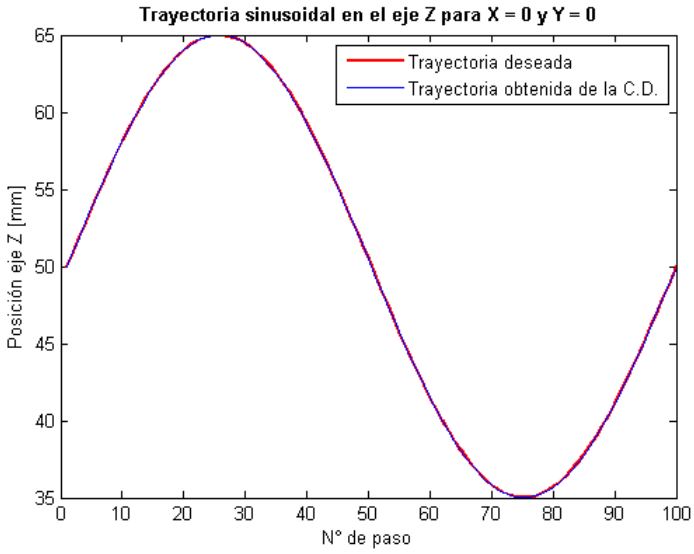
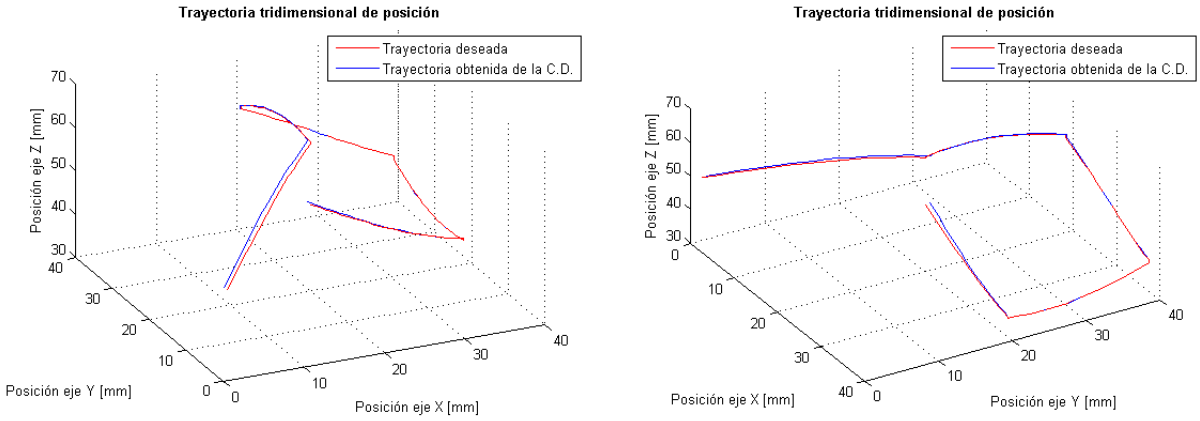


Figura 6.8: Trayectoria senoidal para el eje Z bajo Y = 0 [mm] y X = 0 [mm].

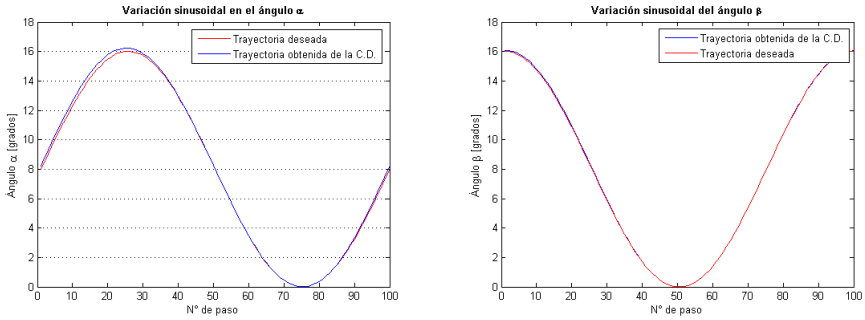
Finalmente, para corroborar de manera más certera las soluciones a la cinemática inversa y directa, se proponen ahora trayectorias tridimensionales con la orientación fija en $\alpha = 0$, $\beta = 0$ y $\gamma = 0$ como se observa en la Fig. 6.9.



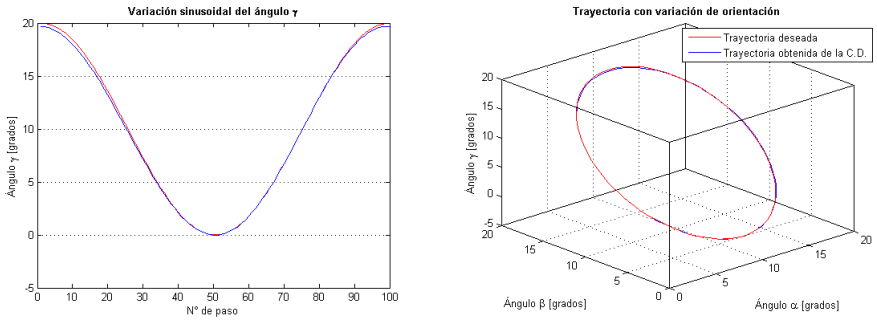
(a) Trayectoria tridimensional (VISTA 1). (b) Trayectoria tridimensional (VISTA 2).

Figura 6.9: Trayectoria tridimensional para una orientación fija

Una última prueba consistió en variar la orientación con perfiles sinusoidales de modo que la trayectoria tridimensional seguida es el círculo que se observa en la Fig. 6.10, corroborando en ambos casos que las soluciones se sobreponen o se aproximan con un error mínimo, por lo que las soluciones son correctas.



(a) Trayectoria seguida por el ángulo α. (b) Trayectoria seguida por el ángulo β.



(c) Trayectoria seguida por el ángulo γ. (d) Vista tridimensional de la trayectoria.

Figura 6.10: Trayectoria tridimensional con orientación variable.

6.2.3. Validación de la cinemática de velocidad del robot a partir de la teoría de tornillos

Para validar los resultados obtenidos de la solución de cinemática de velocidad del robot se propone ahora comparar el estado de velocidad obtenido de la solución analítica en MATLAB con lo obtenido del software SolidWorks. De modo que, se construye el siguiente modelo en Simulink:

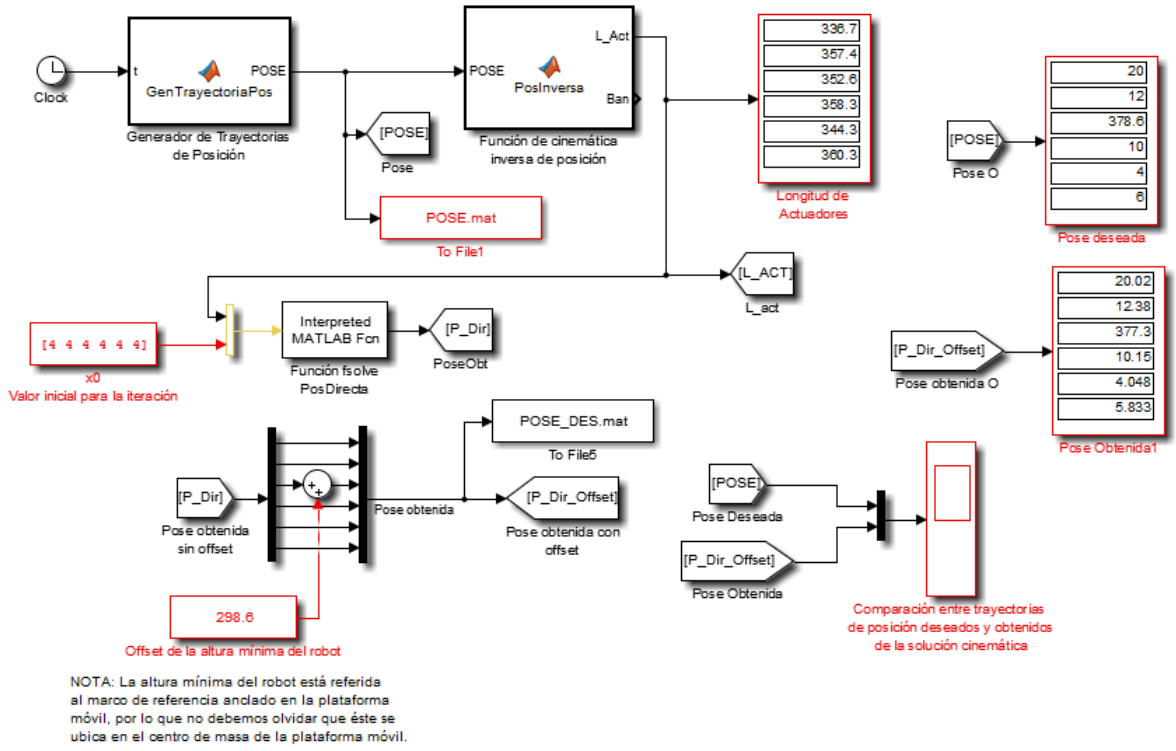


Figura 6.11: Diagrama de bloques para validar la cinemática de posición.

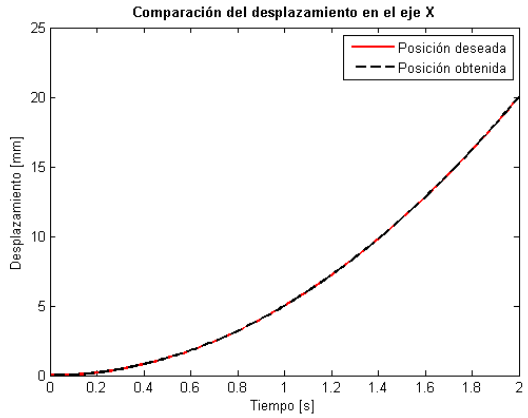
En la Fig. 6.11, se observa el modelo de bloques construido para verificar y validar el análisis cinemático de posición. El funcionamiento es bastante simple, el bloque generador de trayectorias recibe como entrada un vector representativo del tiempo y obtiene posición y orientación del robot para cada instante de tiempo. Para este caso se programaron las siguientes ecuaciones para cada una de las variables:

$$\begin{aligned} X &= 5t^2 \\ Y &= 3t^2 \\ Z &= 20t^2 + 298.6 \\ A &= 2.5t^2 \\ B &= t^2 \\ G &= 1.5t^2 \end{aligned} \tag{6.3}$$

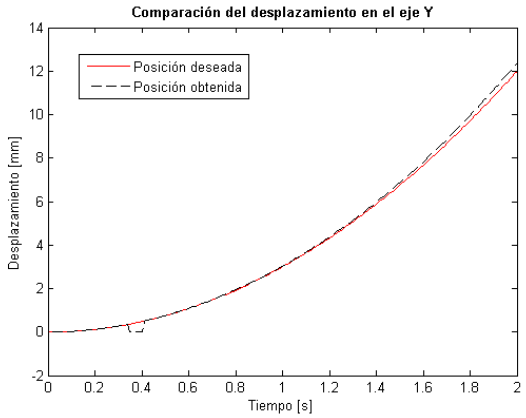
Una vez hecho esto, la salida del bloque de trayectorias genera un vector con la pose del mecanismo que sirve como entrada del bloque que resuelve la cinemática inversa de posición, dentro del bloque se implementa el mismo código que se muestra en el Apéndice A. La salida de dicho bloque se introduce al módulo que implementa la función *fsolve*, además de un vector de condición inicial para la primer iteración. Finalmente, ambos resultados se comparan obteniendo las gráficas mostradas en la Fig. 6.12.

De manera general las seis gráficas coinciden, aunque para casos particulares como son en las Figs. 6.12.d y 6.12.f existen regiones en donde la posición obtenida se aleja de la deseada y se generan picos, esto debido a que el algoritmo numérico para resolver la cinemática directa no encuentra una solución para tales puntos, sin embargo en el resto de la trayectoria los resultados coinciden al grado de traslaparse por lo que se concluye que la solución de cinemática de posición es correcta.

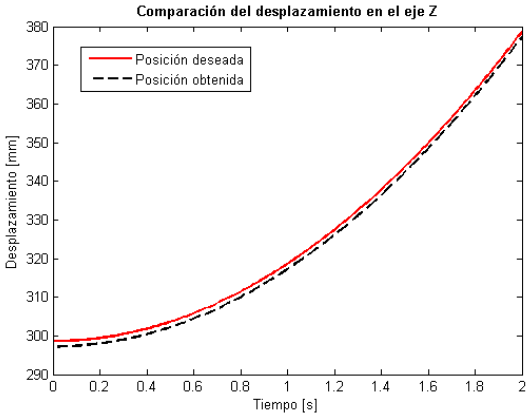
Una vez obtenidos los desplazamientos requeridos en los actuadores para alcanzar dicha trayectoria con ayuda de la cinemática directa, se derivan dichos desplazamientos y obtener la velocidad de los mismos. Por lo que haciendo uso de los datos obtenidos del bloque de cinemática inversa de posición, como se observa en la Fig. 6.13, a la entrada del bloque de cinemática directa de velocidad *VelDirecta* se derivan y se introducen como velocidades dadas para los actuadores, además, el bloque recibe la pose como entrada para actualizar la dirección de los tornillos en las extremidades con el paso de cada iteración.



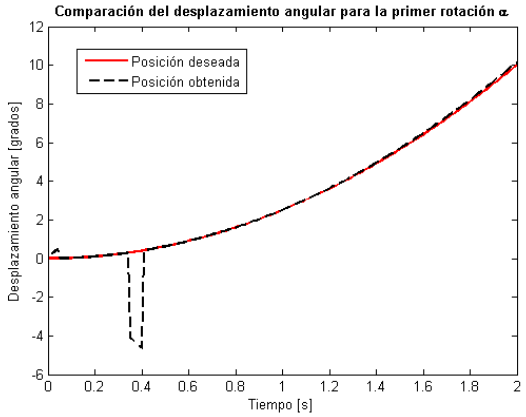
(a) Trayectoria seguida en la dirección del eje X.



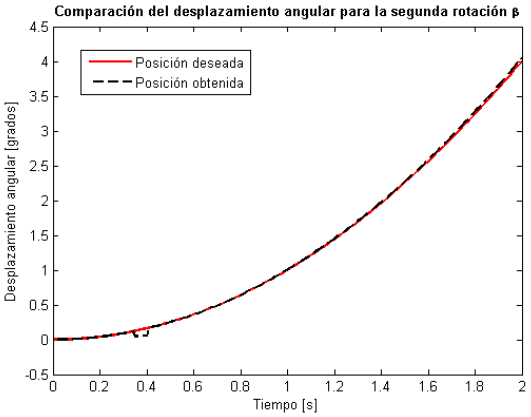
(b) Trayectoria seguida en la dirección del eje Y.



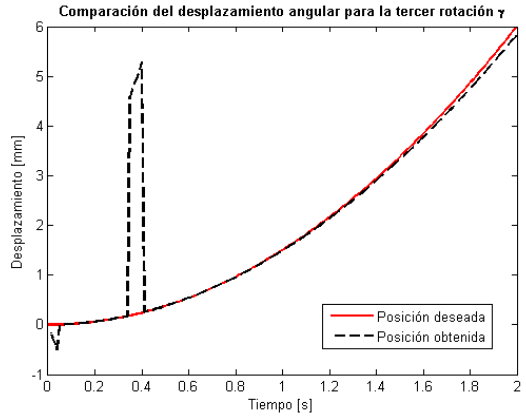
(c) Trayectoria seguida en la dirección del eje Z.



(d) Trayectoria seguida por el ángulo α.



(e) Trayectoria seguida por el ángulo β.



(f) Trayectoria seguida por el ángulo γ.

Figura 6.12: Validacion de cinemática de posición.

Continuando con el modelo a bloques observado en la Fig. 6.13, se tiene como salida del bloque de cinemática directa de velocidad el estado de velocidad del robot, mismo que se introduce dentro del bloque de cinemática inversa de velocidad obteniendo las velocidades de los actuadores nuevamente. Esto con la finalidad de comparar ambos resultados, las velocidades de los actuadores deseados con los obtenidos con ayuda del análisis cinemático.

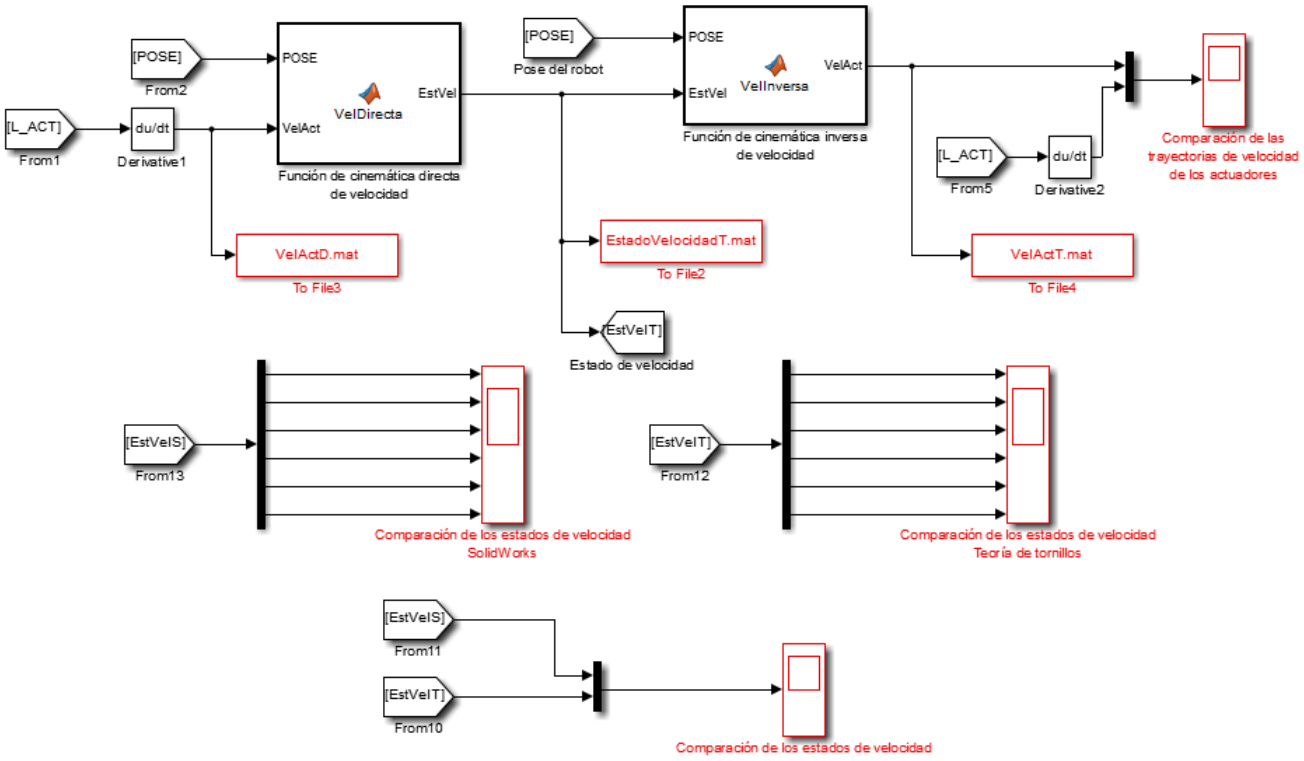


Figura 6.13: Diagrama de bloques para validar la cinemática de velocidad.

De igual manera, se observa en las Figs. 6.13 y 6.14 que se comparan los estados de velocidad obtenidos con la teoría de tornillos y los obtenidos al derivar la pose del robot respecto del tiempo.

Sin embargo, dado que el estado de velocidad arroja velocidades angulares vistos desde el marco de referencia fijo, es necesario encontrar una relación entre las rotaciones de Euler y las obtenidas de la solución analítica, dicha relación es justamente la que se programa dentro del bloque denominado *Transformación* que se muestra en la Fig. 6.14. La comparación de resultados se realiza en la Fig. 6.16.

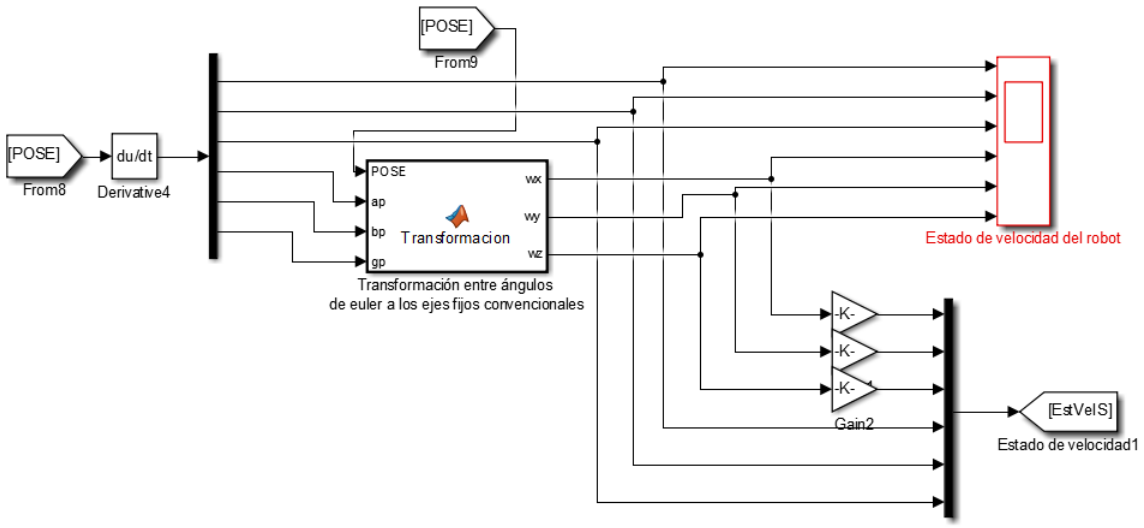


Figura 6.14: Diagrama para obtener el estado de velocidad a partir de las derivadas de la pose.

Finalmente, en la Fig. 6.15 se muestran los valores de extensión del vástago de los actuadores, valores que se introducen y envían al prototipo experimental a partir de la comunicación serial. Los programas implementados en cada uno de los bloques de funciones mostrados anteriormente se pueden consultar en el Apéndice C.

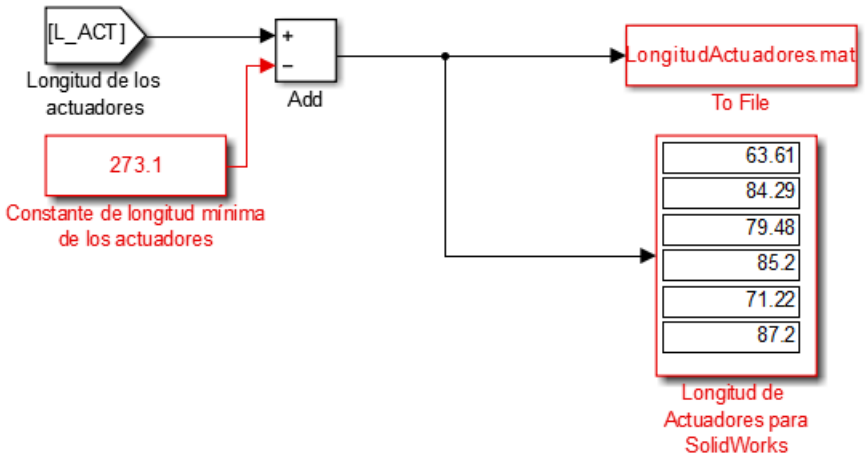
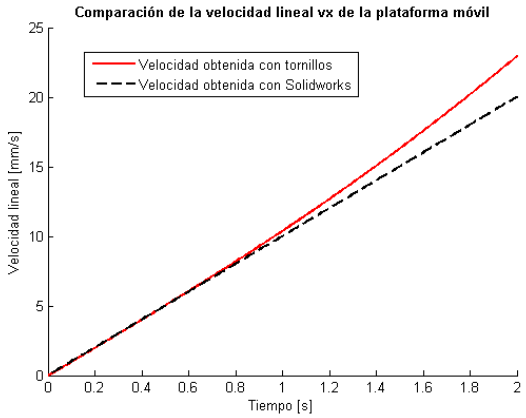
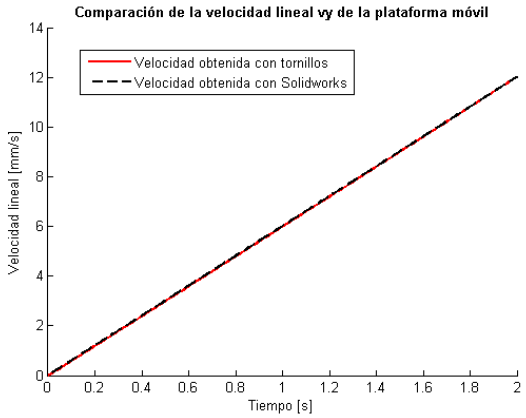


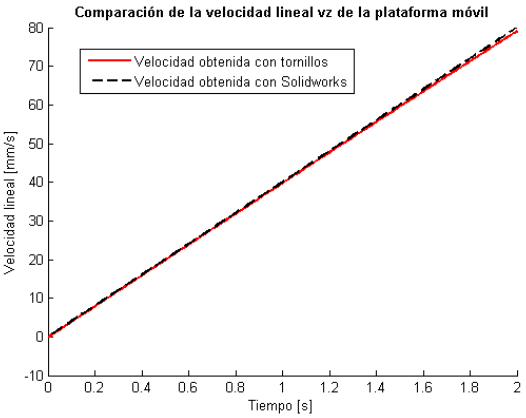
Figura 6.15: Diagrama para obtener la longitud del vástago de los actuadores.



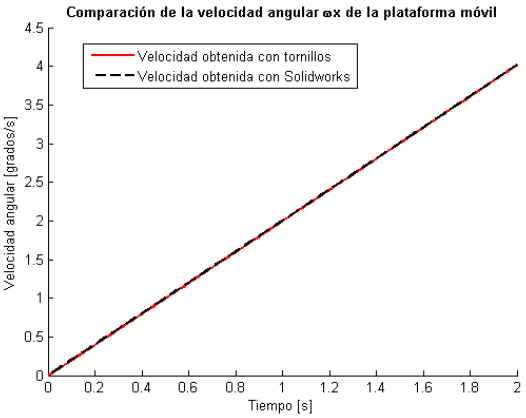
(a) Trayectoria de velocidad seguida en la dirección del eje X.



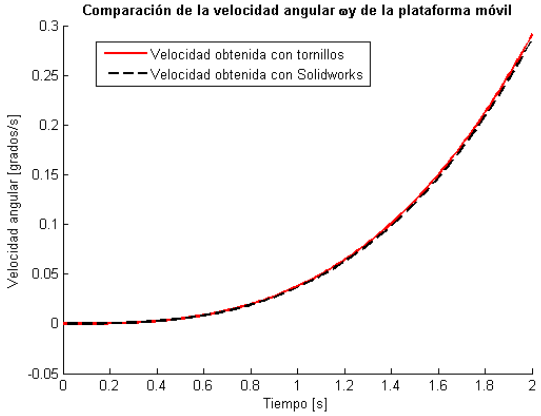
(b) Trayectoria de velocidad seguida en la dirección del eje Y.



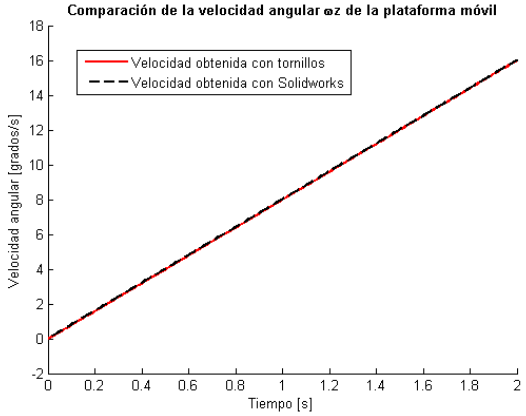
(c) Trayectoria de velocidad seguida en la dirección del eje Z.



(d) Trayectoria de velocidad angular ωx.



(e) Trayectoria de velocidad angular ωy.



(f) Trayectoria de velocidad angular ωz.

Figura 6.16: Validacion de cinemática de velocidad.

6.3. Validación experimental de los resultados cinemáticos

Una vez corroborados los resultados cinemáticos de posición y velocidad de manera analítica y en simulación, se presenta como última forma de validación, los resultados experimentales obtenidos del prototipo construido.

6.3.1. Validación cinemática de la orientación del robot

En primer lugar, se validan los grados de libertad correspondientes a la orientación de la plataforma, solicitando determinadas configuraciones de manera puntual y posteriormente a partir de trayectorias.

Para validar configuraciones de orientación de manera puntual, se utilizó la interfaz para accionar al robot, comparando los ángulos medidos por el sensor y la orientación deseada. Como se observa en la Tabla 6.2, el máximo error obtenido es de 5 [grados] en el caso más crítico de todas las orientaciones deseadas (sólo se muestran los valores correspondientes a las variables involucradas con la orientación: α, β y γ [1]).

Tabla 6.2: Resultados obtenidos de la validación cinemática de forma experimental.

Pose deseada [grados]	**Medición experimental [grados]**	**Error máximo [grados]**
[20,20,0]	[17,19,2]	3
[-20,-20,0]	[-20,-21,0]	1
[0,-20,0]	[-1,-20,0]	1
[20,-22,-27]	[15,-21,-27]	5

Observando estos mismos resultados de manera individual en las Figs. 6.17, 6.18, 6.19 y 6.20. En la parte inferior de la interfaz se introdujo la orientación deseada y en la parte superior se puede observar la orientación alcanzada medida con el sensor (IMU).

Una vez validada de manera puntual la orientación, se utilizan trayectorias para cada una de las rotaciones alrededor de los ejes, comenzando por eje X, al que se le programa una trayectoria sinusoidal de la forma $20\sin(2\pi t)$, donde t varía de 0 a 18 segundos, dicho resultado se comparará con la trayectoria deseada y con lo obtenido de la solución de la cinemática directa.

[1] Las rotaciones se modificaron para simplificar la medición de la velocidad, por lo que estos ángulos corresponden a rotaciones sobre el eje X, Y y Z respectivamente

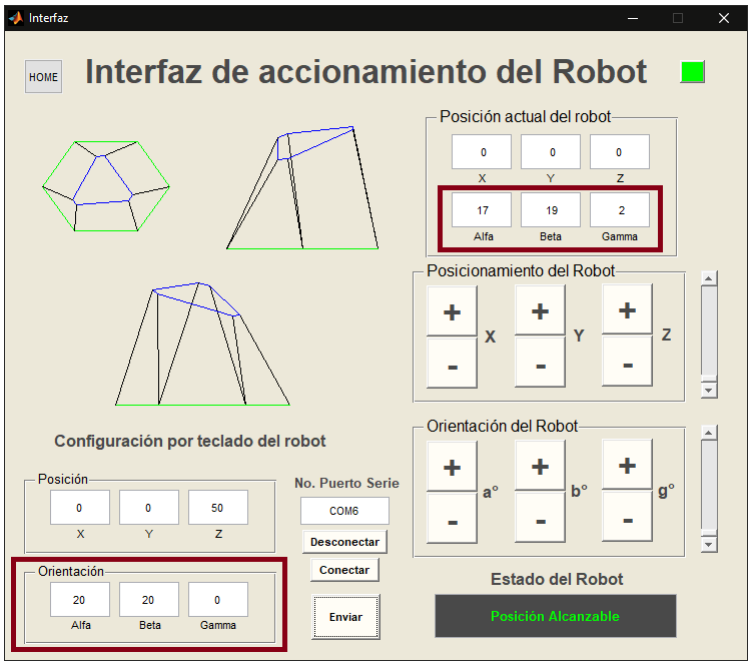


Figura 6.17: Validación puntual de orientación a partir de la interfaz de operación del robot (caso I).

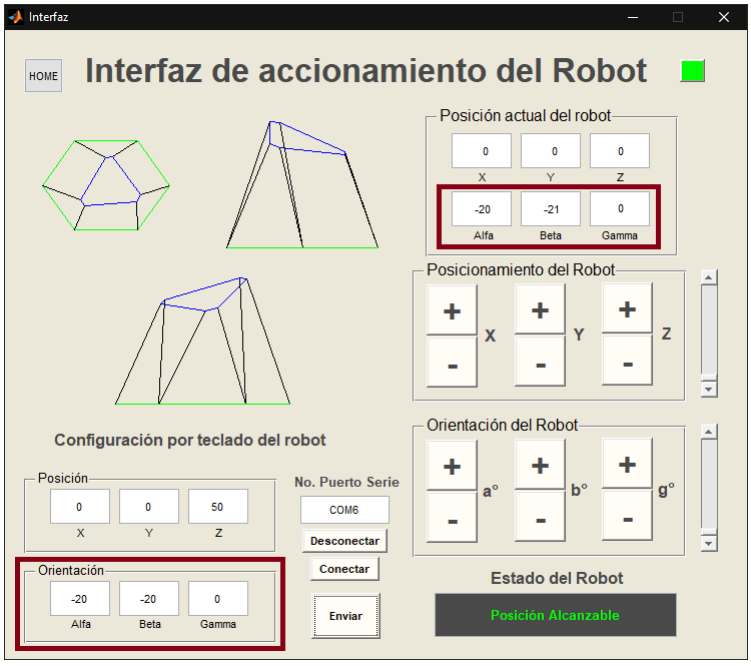


Figura 6.18: Validación puntual de orientación a partir de la interfaz de operación del robot (caso II).

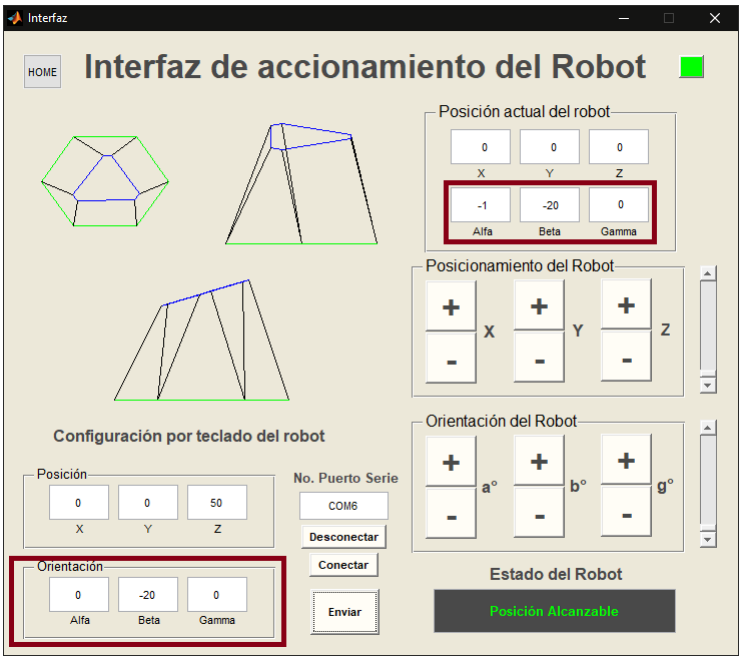


Figura 6.19: Validación puntual de orientación a partir de la interfaz de operación del robot (caso III).

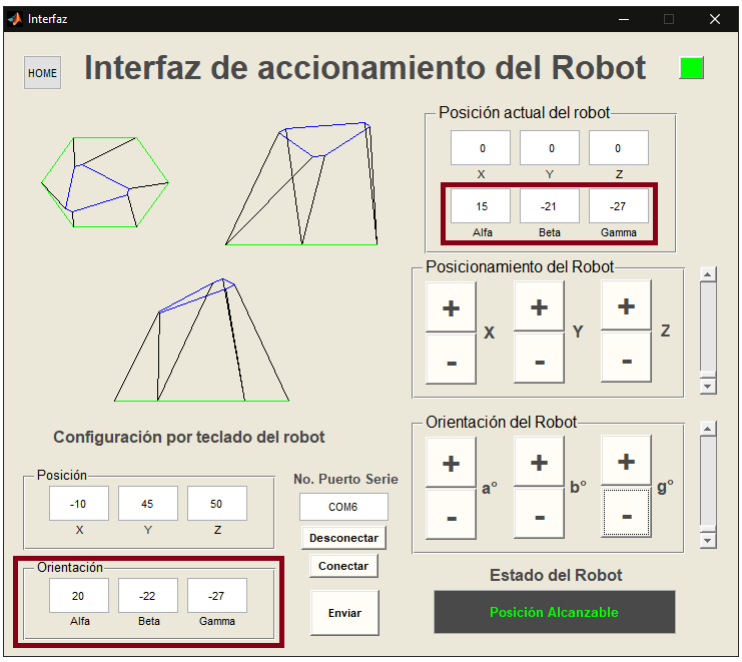


Figura 6.20: Validación puntual de orientación a partir de la interfaz de operación del robot (caso IV).

En la Fig. 6.21 se observan cuatro trayectorias, la primera de ellas en color azul se refiere a la trayectoria deseada, se muestra también en color verde y con una línea punteada la solución obtenida de la cinemática directa, dichos resultados se superponen, de igual manera se muestran los resultados obtenidos de forma experimental con ayuda de la plataforma construida, la medición con la letra (F) es la señal filtrada del sensor, observando que la trayectoria se aproxima de muy buena manera a la trayectoria deseada.

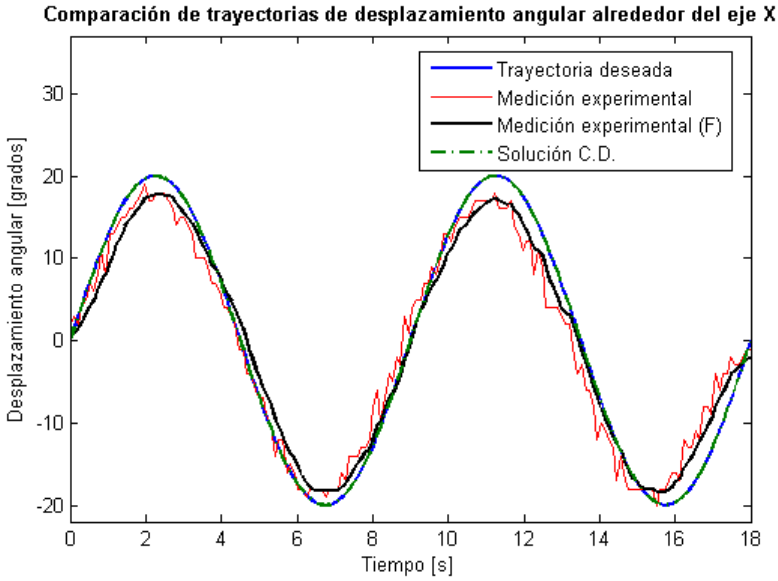


Figura 6.21: Resultados para una trayectoria senoidal alrededor del eje X.

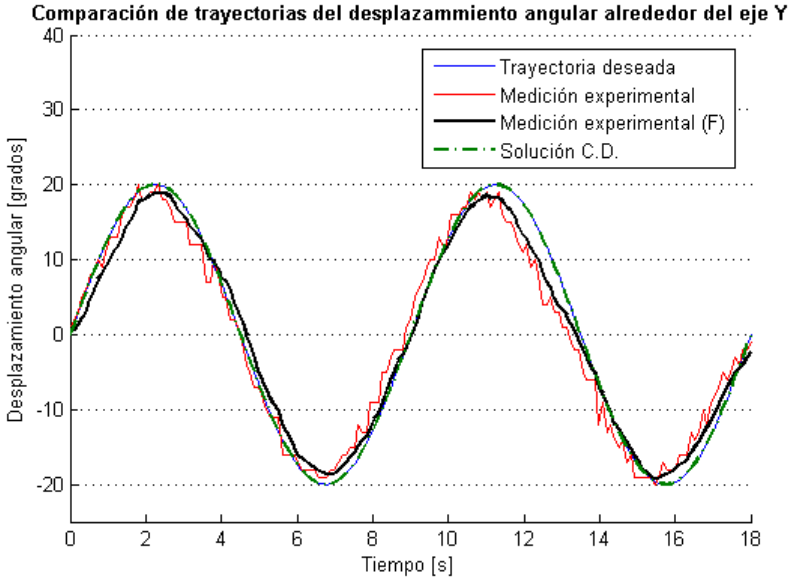


Figura 6.22: Resultados para una trayectoria senoidal alrededor del eje Y.

Continuando con la validación de la cinemática, ahora con el eje Y bajo la trayectoria $20\sin(2\pi t)$, obteniendo lo resultados que se observan en la Fig. 6.22. De igual manera se observan cuatro trazos correspondientes a la orientación deseada, la trayectoria obtenida de la cinemática directa y finalmente las dos obtenidas de las lecturas del sensor en el prototipo. Del mismo modo que para el caso anterior, las trayectorias son bastante aproximadas de la deseada.

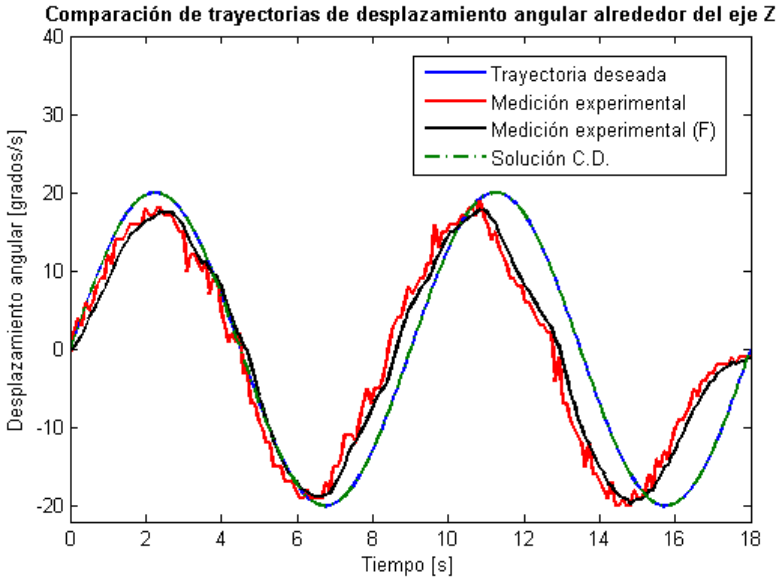


Figura 6.23: Resultados para una trayectoria senoidal alrededor del eje Z.

Finalmente se valida la rotación respecto del eje Z, considerando la misma trayectoria que para los casos anteriores se obtienen los resultados mostrados en la Fig. 6.23 de la que se observa que existe un pequeño desfase de la trayectoria deseada después del primer periodo, sin embargo, la forma sinusoidal y las magnitudes del desplazamiento se conservan.

Continuando con las variables de orientación, se validan a continuación trayectorias de velocidad correspondientes a los desplazamientos angulares mostrados anteriormente, comparando lo obtenido de manera experimental con los resultados arrojados de la cinemática utilizando la teoría de tornillos.

En la Fig. 6.24 se pueden observar los resultados obtenidos para la velocidad alrededor del eje X, se muestra la velocidad obtenida de la solución a partir de la teoría de tornillos en color azul y las mediciones del sensor en colores rojo y negro representando la señal filtrada y sin filtro respectivamente.

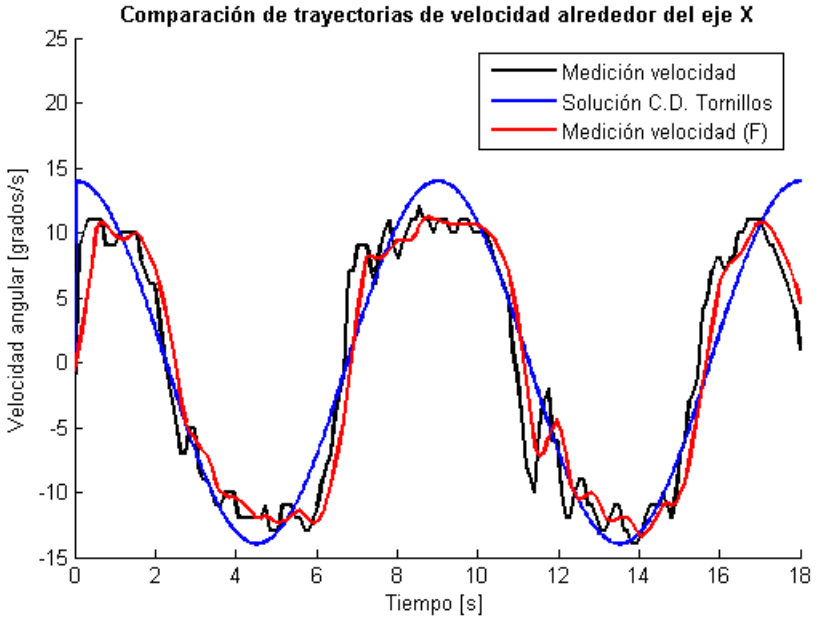


Figura 6.24: Resultados para una trayectoria de velocidad senoidal alrededor del eje X.

En la Fig. 6.25 se muestra la trayectoria de velocidad alrededor del eje Y, apreciando que la trayectoria es bastante parecida, sin embargo, la magnitud de la velocidad leída de manera experimental es menor que la calculada de manera teórica, de igual manera se observa un pequeño desfase entre las señales.

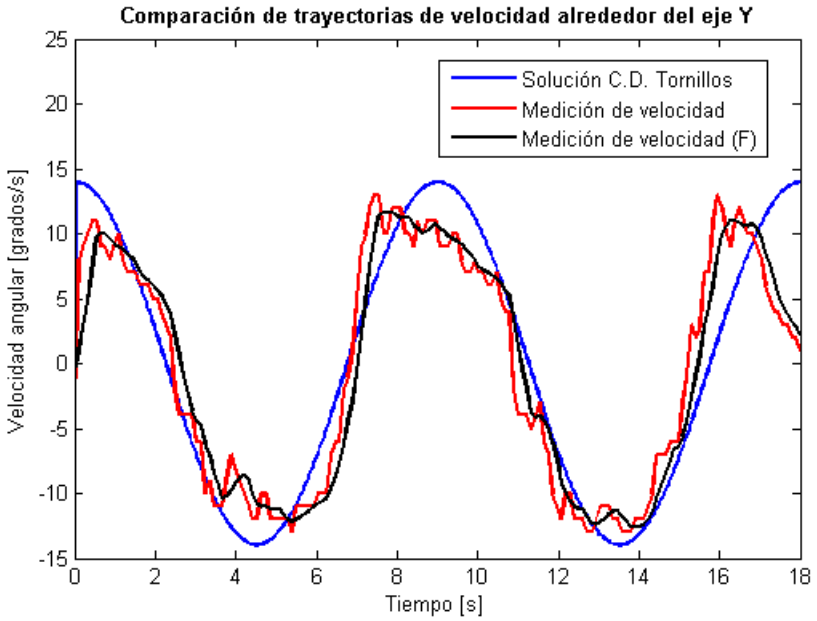


Figura 6.25: Resultados para una trayectoria de velocidad senoidal alrededor del eje Y.

Finalmente, se observan en la Fig. 6.26 los resultados obtenidos para la trayectoria de velocidad alrededor del eje Z de las que se puede observar un mejor acoplamiento de la solución teórica, sin embargo, al finalizar el movimiento la trayectoria de velocidad leída del sensor se desfasa significativamente de lo deseado.
Con los resultados mostrados anteriormente se culmina con la validación de la cinemática de posición y velocidad de los tres primeros grados de libertad de la plataforma correspondientes a las rotaciones y velocidades angulares alrededor de los tres ejes coordenados, llegando a la conclusión de que lo obtenido de manera analítica se cumple en lo experimental, si bien es cierto que existe un porcentaje de error, esto es debido a que en realidad el prototipo opera en lazo abierto, es decir, no se realiza o toma ninguna acción de control para asegurar la extensión exacta de los actuadores.

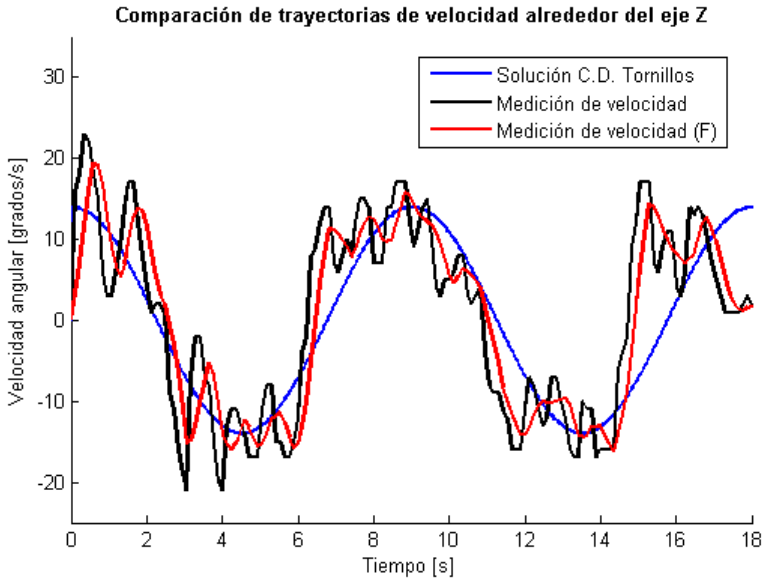


Figura 6.26: Resultados para una trayectoria de velocidad senoidal alrededor del eje Z.

6.3.2. Validación cinemática de la posición del robot

Continuando con la validación cinemática de los tres grados de libertad restantes, se comienza con la coordenada X de la que nuevamente se programa una trayectoria sinusoidal con la forma $30\sin(2\pi t)$ [mm]. Obteniendo resultados muy satisfactorios, como se muestra en la Fig. 6.27, las tres curvas de trayectorias están muy próximas una de la otra, con excepción del primer pico de la función toda la demás trayectoria medida de forma experimental se aproxima bastante a la deseada.

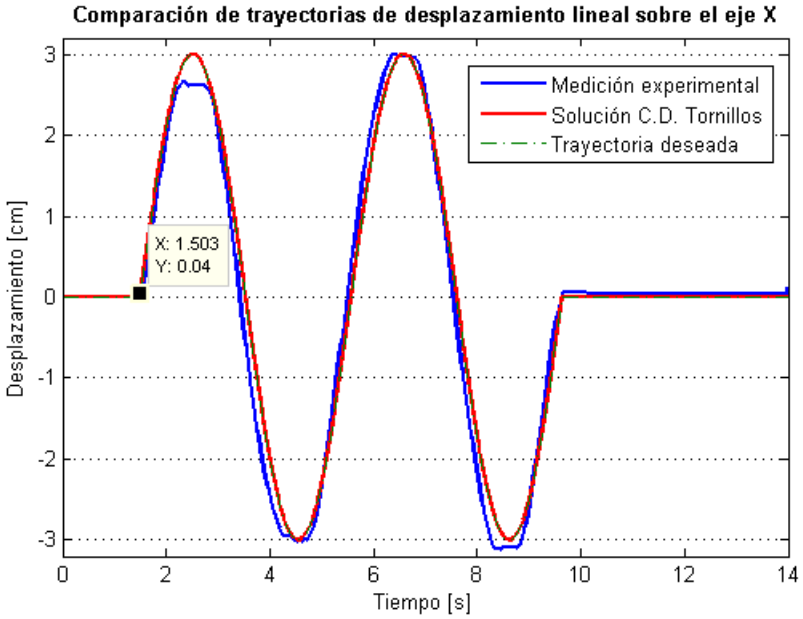


Figura 6.27: Resultados para una trayectoria de desplazamiento lineal sobre el eje X.

En la Fig. 6.28, se observan los resultados obtenidos para desplazamientos sobre el eje Y bajo la misma trayectoria sinusoidal descrita en el caso anterior. Al igual que para el caso anterior, las curvas son sumamente parecidas por lo que la aproximación a la trayectoria deseada es muy satisfactoria.

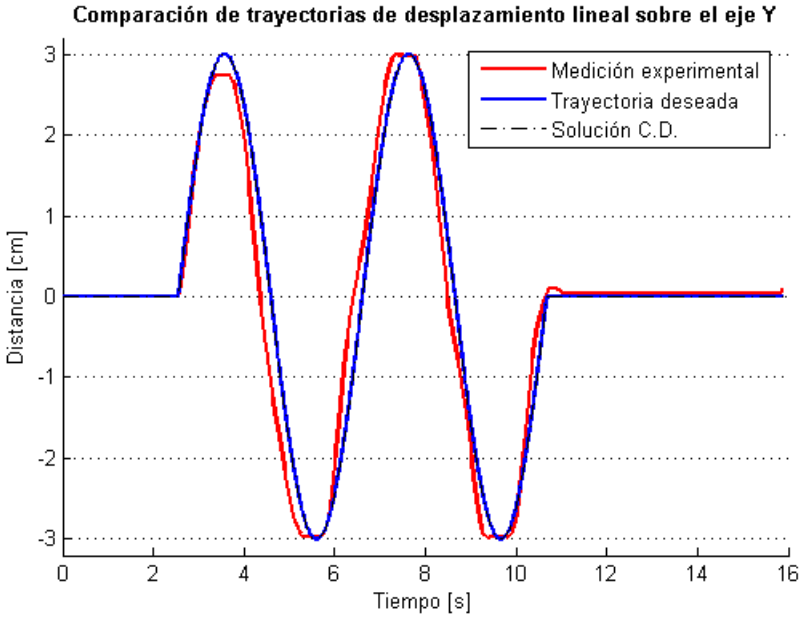


Figura 6.28: Resultados para una trayectoria de desplazamiento lineal sobre el eje Y.

Finalmente, para la última de las coordenadas se tienen los resultados observados en la Fig. 6.29, de la que se concluye que nuevamente las trayectorias son similares, sin embargo, no tan próximas entre ellas, en particular para el segundo periodo de la trayectoria existe un desfase notorio entre las curvas, además, de un pequeño escalón al final, esto debido a un pequeño retardo en los servomotores al alcanzar la posición final.

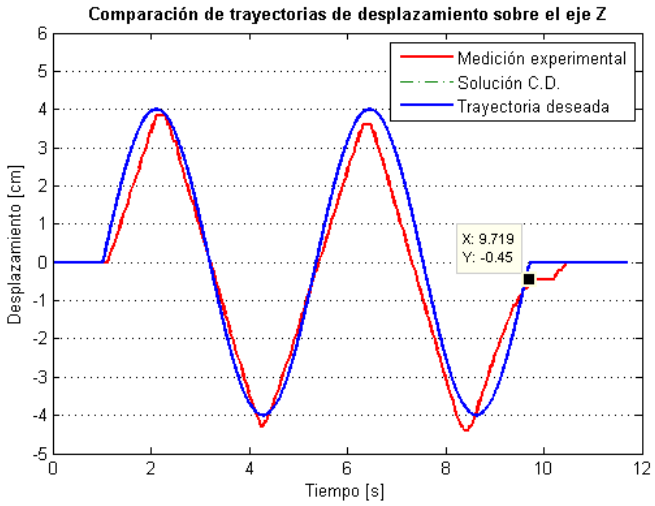


Figura 6.29: Resultados para una trayectoria de desplazamiento lineal sobre el eje Z.

Para finalizar la validación cinemática referente a los grados de libertad correspondientes al desplazamiento lineal de las plataforma, se presentan los resultados para las trayectorias de velocidad de cada una de las coordenadas.

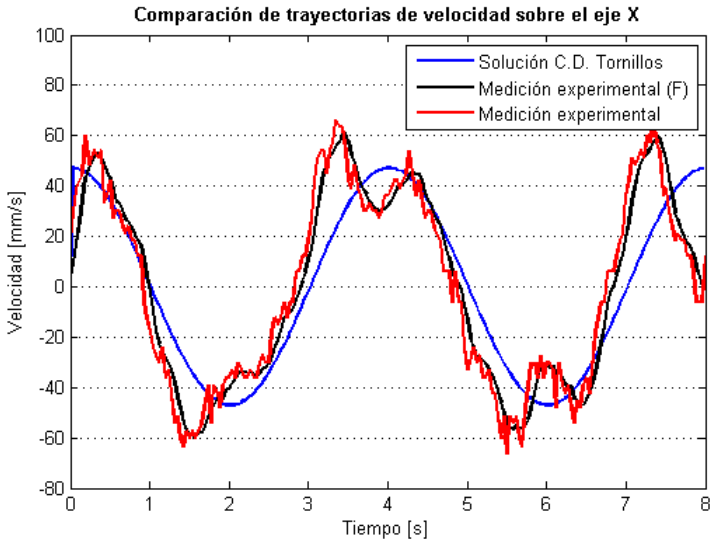


Figura 6.30: Resultados para una trayectoria de velocidad lineal sobre el eje X.

Se comienza por la coordenada X, cuyos resultados se muestran en la Fig. 6.30 de la que se observa que la trayectoria medida con el sensor se asemeja en magnitudes pico de la señal obtenida como solución de la cinemática directa de velocidad, sin embargo, la forma de la curva difiere un tanto, en especial en la parte central donde aparecen dos picos, pero de manera general se aproxima a la trayectoria cosenoidal.

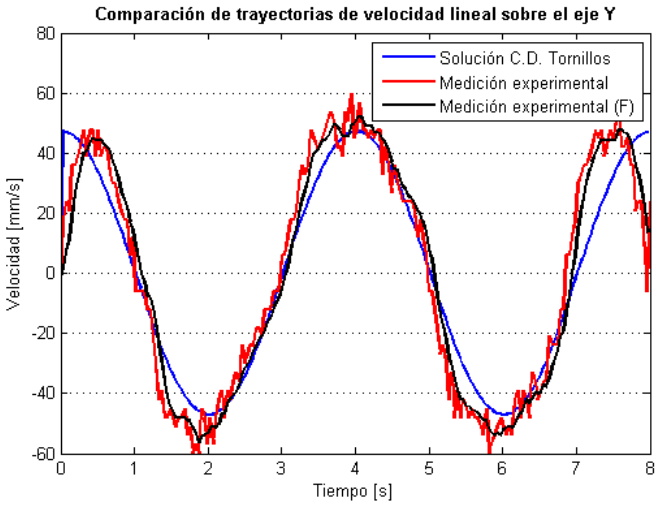


Figura 6.31: Resultados para una trayectoria de velocidad lineal sobre el eje Y.

En la Fig. 6.31 se observan los resultados de la trayectoria de velocidad sobre el eje Y, para este caso las trayectorias se aproximan entre sí de mucho mejor manera con excepción de los bordes al inicio y final de las trayectorias en donde la señal medida se desfasa un poco de la solución de cinemática directa obtenida con la teoría de tornillos.

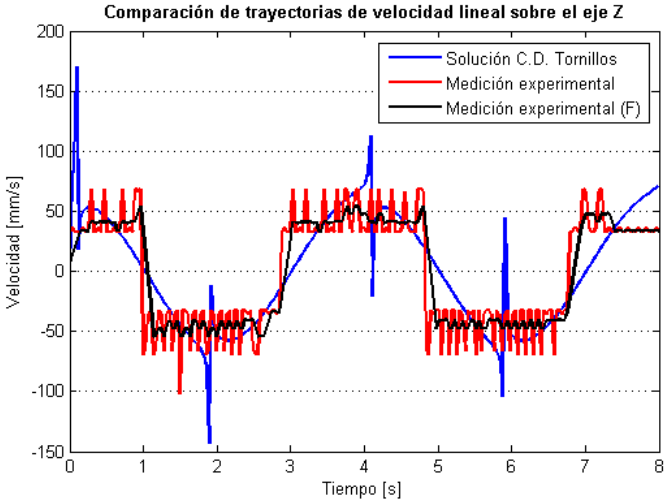


Figura 6.32: Resultados para una trayectoria de velocidad lineal sobre el eje Z.

Finalmente, en la Fig. 6.32 se observan los resultados de velocidad sobre el eje Z, obteniendo algunos picos en la solución de cinemática directa debido a la solución numérica de las ecuaciones, además, las señales medidas de manera experimental concuerdan en magnitudes pico de la señal, sin embargo, posee una forma más cuadrada que cosenoidal aunque se mantiene la frecuencia de la señal.
Se puede concluir que para los grados de libertad correspondientes a la posición, los errores entre las trayectorias deseadas y las obtenidas de manera experimental son mínimos, en especial para el caso del desplazamiento lineal donde las curvas casi se traslapan. Mientras que para el caso de velocidad no es tan cercana la aproximación a la solución analítica, sin embargo, se sigue la trayectoria de manera satisfactoria, recordando nuevamente que el prototipo opera en lazo abierto.

Capítulo 7

Conclusiones y trabajos futuros

Con el desarrollo de esta tesis se cumplió con el objetivo de validar la cinemática del robot paralelo de seis grados de libertad de configuración 6-U$\underline{P}$UR, de igual manera se realizó el análisis cinemático de velocidad mediante la *Teoría de tornillos* para corroborar la reducción de la complejidad del análisis bajo esta técnica. Así mismo, la construcción de un prototipo de robot paralelo de seis grados de libertad permitió la validación de los resultados analíticos obtenidos.

Para llevar a cabo la validación del análisis cinemático de posición, fue necesario obtener las ecuaciones de cinemática directa e inversa del robot paralelo para verificar de manera cíclica ambos resultados utilizando la salida de una función como entrada de la otra, observando que para trayectorias y de forma puntual en el espacio, las soluciones convergen hacia el mismo valor deseado, validando de manera numérica las soluciones obtenidas para la cinemática de posición del robot.

De igual manera, se procedió con el análisis cinemático de velocidad para mostrar las ventajas de la utilización de la *Teoría de tornillos* siendo un método mucho más rápido y eficiente para llegar a la matriz Jacobiana del mecanismo y por tanto a la ecuación de velocidad con el enfoque de entradas-salidas. Sin embargo, existe cierto grado de complejidad debido a la configuración propuesta en el robot, en la que se sustituye la junta esférica atada a la plataforma móvil por un arreglo de junta universal y revoluta, emulando los grados de libertad ofrecidos por la junta esférica. Por lo que se concluye que en efecto, el método de análisis a partir de la *Teoría de tornillos* si es más fácil que derivar y manipular algebraicamente las ecuaciones de posición, sin embargo la definición de los tornillos como tal, puede resultar un tanto compleja para quienes empiezan a adentrarse en la técnica, siendo más laborioso y extenso el método convencional pero hasta cierto punto mucho menos confuso en cuanto a la definición de vectores se refiere.

Otra de las ventajas de la teoría de tornillos, es que se pueden eliminar de forma metódica las

variables correspondientes a las juntas pasivas para obtener modelos de velocidad de entrada-salida, los cuales pueden ser extendidos para los análisis de aceleración y de fuerzas.
Como segunda forma de validación de la cinemática de posición, se construyó un prototipo de robot paralelo de seis grado de libertad bajo la configuración 6-U$\underline{P}$UR, además de instrumentarlo de modo que se pudieran medir las variables de posición y orientación del mismo. Obteniendo de igual manera resultados sumamente satisfactorios en los que el error a los puntos deseados es a lo más de un par de milímetros para los casos más críticos. Respecto de la elección del modelo y configuración de las extremidades del robot paralelo se llegó a tal decisión tratando de generalizar lo más posible el análisis y la movilidad que pudiera desarrollar el robot, teniendo en cuenta que dado que se ofrece como una herramienta didáctica de aplicación general, debe presentar el mayor número posible de grados de libertad, y permitir realizar actividades posteriores como control de los actuadores y por ende de trayectorias en posición, velocidad e incluso aceleración.
Finalmente para validar los resultados analíticos obtenidos de la cinemática de velocidad se recurre a simulaciones a partir del software SolidWorks para corroborar que el estado de velocidad del robot concuerde con los resultados del software, obteniendo en todos los casos resultados sumamente satisfactorios pues concuerdan de una manera bastante buena, llegando a traslaparse las trayectorias de velocidad, con lo que se concluye que la solución de cinemática de velocidad es correcta.

7.1. Trabajos futuros:

A partir de la experiencia obtenida con el desarrollo de este trabajo, se proponen los siguientes trabajos futuros para mejorar algunos aspectos del mismo y extender los resultados obtenidos.

- Aislar las terminales de alimentación de los servomotores y la señal de retroalimentación del potenciómetro para desarrollar placas de control para los motores que proporcionan el movimiento de las extremidades del robot.
- Extender el análisis cinemático por teoría de tornillos a aceleración.
- Obtener el modelo dinámico del robot para simular su comportamiento y aplicar distintas leyes de control al modelo.
- Realizar control de velocidad y aceleración de la plataforma para distintos tipos de trayectorias.
- Obtener los puntos de singularidad del robot y por tanto definir de manera mucho más precisa el área de trabajo del robot, además de determinar el espacio de trabajo diestro del mismo.

Bibliografía

[1] D. Zhang, *Parallel Robotic Machine Tools*. Springer, first ed., 2010.

[2] J. P. Merlet, *Parallel Robots*. Springer, second ed., 2006.

[3] H. D. Taghirad, *Parallel Robots Mechanics and Control*. CRC Press, first ed., 2013.

[4] D. Stewart, "A platform with six degrees of freedom," *Institution of Mechanical Engineers Proceedings*, vol. 180 No. 1, pp. 371–386, 1965.

[5] "IRB 360 FlexPicker." url: https://new.abb.com/products/robotics/es/robots-industriales/irb-360, 2018.

[6] "Spidercam technical info." url: https://www.spidercam.tv/technical-info/, (2018, Septiembre 10), 2018.

[7] N. G. Hockstein, C. G. Gourin, R. A. Faust, and D. J. Terris, "A history of robots: from science fiction to surgical robotics," *Journal of Robotic Surgery*, vol. 1, No. 2, pp. 113–118, 2007.

[8] M. Shafiee-Ashtiani, A. Yousefi-Koma, S. Iravanimanesh, and A. S. Bashardoust, "Kinematic analysis of a 3-UPU parallel robot using the ostrowski-homotopy continuation," *Center of Advanced Systems and Technologies*, 09 2016.

[9] C. Gosselin, V. Parenti-Castelli, and F. Pierrot, "Fundamental issues and new trends in parallel mechanisms and manipulators," *Meccanica*, vol. 46 No.1, pp. 1–1, 2011.

[10] W. Wallace, *Gestión de proyectos*. Gran Bretaña:: Edinburgh Business School, primera edición ed., 2002.

[11] N. Cross, *Engineering design Methods*. New York:: John Wiley and Sons Inc., third ed., 2005.

[12] S. Kücük, ed., *Serial and Parallel Robot Manipulators-Kinematics, Dynamics, Control and Optimization*. InTech, first ed., 2012.

[13] X. Kong and C. Gosselin, eds., *Type Synthesis of Parallel Mechanisms*. Springer, first ed., 2007.

[14] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. John Wiley and Sons Inc., first ed., 2005.

[15] J. D. Barnfather, M. J. Goodfellow, and T. Abram, "Positional capability of a hexapod robot for machining applications," *The International Journal of Advanced Manufacturing Technology*, vol. 89 No. 1-4, pp. 1103–1111, 2017.

[16] Z. Huang, Q. Li, and H. Ding, *Theory of Parallel Mechanisms*. Springer, first ed., 2013.

[17] S. Stramigioli and H. Bruyninckx, *Geometry and Screw Theory for Robotics*. IEEE Robotics and Automation Society, primera ed., 2001.

[18] Z. Huang, Q. Li, and H. Ding, *Basics of Screw Theory*. In: Theory of Parallel Mechanisms. Mechanisms and Machine Science, vol 6. Springer, Dordrecht, 2013.

[19] H. J. Su, D. V. Dorozhkin, and J. M. Vance, "A screw theory approach for the conceptual design of flexible joints for compilant mechanisms," tech. rep., Iowa State University, 2009.

[20] J. G. Alvarado, *Kinematic analysis of parallel manipulators by algebraic Screw theory*. Springer, first ed., 2016.

[21] G. J. Hamlin and A. C. Sanderson, eds., *Tetrobot a Modular Appproach to Reconfigurable Parallel Robotics*. Springer, first ed., 2007.

[22] K. M. Lynch and F. C. Park, *Modern Robotics Mechanics, Planning and Control*. Springer, preprint ed., 2017.

[23] M. Ben-Ari and F. Mondada, *Elements of Robotics*. Springer, first ed., 2018.

[24] M. J. P. Márquez, "Estudio de la aceleración y posición en sistemas inerciales por medio de arduino," Universidad politécnica de valencia, 2016.

[25] "¿qué es la geofísica?." url: https://www.gob.mx/cms/uploads/attachment/file/157796/Que-es-la-Geofisica.pdf, (2018, Septiembre 20), 2018.

[26] M. Automatización, "Srvomotores: control, precisión y velocidad," *AADECA REVISTA*, vol. 1, Edición 4, pp. 22–23, 2017.

[27] J. P. Contreras, "Modulación por ancho de pulso (pwm) y modulación vectorial (svm). una introducción a las técnicas de modulación.," *El hombre y la máquina*, vol. 1, No. 25, pp. 70–83, 2005.

[28] "L16r miniature linear servos for rc and arduino." url: https://www.actuonix.com/L16-R-Miniature-Linear-Servo-For-RC-p/l16-r.htm, (2018, Septiembre 10), 2018.

[29] "Rodamientos rígidos de bolas." url: https://www.skf.com/pe/products/bearings-units-housings/ball-bearings/deep-groove-ball-bearings/deep-groove-ball-bearings/index.html?designation=607-Z, (2018, Septiembre 11), 2018.

[30] "fsolve mathworks documentation." url: https://www.mathworks.com/help/optim/ug/fsolve.html, (2018, Septiembre 11), 2018.

[31] "Micro maestro 6-channel usb servo controller." url: https://www.pololu.com/product/1350/resources, (2018, Septiembre 01), 2018.

[32] "Sparkfun 9dof imu breakout - lsm9ds1." url: https://www.sparkfun.com/products/13284, (2018, Septiembre 02), 2018.

[33] "Manual de usuario Arduino NANO." url: https://www.arduino.cc/en/uploads/Main/ArduinoNanoMa (2018, Septiembre 02), 2018.

[34] Óscar Torrente Artero, *ARDUINO curso práctico de formación.* Alfaomega, first ed., 2013.

Apéndice A

Programas implementados en MATLAB para el cálculo cinemático.

A continuación se presentan los tres programas utilizados para el cálculo de cinemática directa e inversa del robot, comenzando por la función de cinemática directa mostrada a continuación (Dicha función recibe como parámetros un vector **x** que representa las variables de posición y orientación y devuelve justamente **F** un vector columna con la evaluación de cada función):

```
function F = CinematicaDirecta(x)
```

Figura A.1: Encabezado de la función de MATLAB.

Como se mencionó en el Capítulo 4, para la resolución de la cinemática directa se optó por utilizar un método numérico, dicho método es el implementado por la función *fsolve* de MATLAB configurado para usar el algoritmo de *Levenberg-Marquardt*, por tanto en las Figs. A.1, A.2, A.3, A.4 y A.5 se muestra la función **CinematicaDirecta** que se requiere como entrada de la función *fsolve*.

Consideraciones del robot:

```
% De las restricciones geométricas que presenta el robot estan las
% siguientes:
% Longitud mínima de los actuadores S_min = 274 [mm]
% Altura mínima de la plataforma Tz_min = 302.7158 [mm]

Tz_offset = 302.7158;
```

Figura A.2: Consideraciones geométricas de la plataforma.

Vector de longitudes de eslabones:

```
% Se supone conocido el vector
% NOTA: Sumar a los resultados obtenidos de la función de cinemática
% inversa la cantidad de 274 [mm] pues solo entrega la extención del
% actuador.

S = [274+5 274+27 274+19 274+16 274+6 274+19];
```

Figura A.3: Vector de entradas de movimiento.

Las Figs. A.2 y A.3 son líneas de código que aportan restricciones como la altura mínima, y las entradas de movimiento que son las longitudes de los eslabones, sin embargo si se utiliza la función de cinemática inversa para obtener dichos valores se le debe sumar 274 pues el resultado de la función es solo la longitud de extensión del vástago.

Ubicación de los vértices y puntos de conexión de la plataforma con las extremidades:

```
AX = [105.8581 14.0150 -14.0150 -105.8581 -91.8431 91.8431];
AY = [-44.9341 114.1428 114.1428 -44.9341 -69.2087 -69.2087];
AZ = [-27.85 -27.85 -27.85 -27.85 -27.85 -27.85];
BX = [200 100 -100 -200 -100 100];
BY = [0 173.2051 173.2051 0 -173.2051 -173.2051];
BZ = [21.5 21.5 21.5 21.5 21.5 21.5];
```

Figura A.4: Ubicación de los puntos de conexión.

Sistema de ecuaciones programada en MATLAB

```
% NOTA: Las filas denotan cada una de las ecuaciones igualadas a cero.
% y las variables fueron sustituidas por el vector X

F = [((x(1)+((cosd(x(4))*cosd(x(6))-sind(x(4))*cosd(x(5))*sind(x(6)))*AX(1))
    -((cosd(x(4))*sind(x(6))+sind(x(4))*cosd(x(5))*cosd(x(6)))*AY(1))
    +((sind(x(4))*sind(x(5)))*AZ(1))-BX(1))*(x(1)+((cosd(x(4))*cosd(x(6))
    -sind(x(4))*cosd(x(5))*sind(x(6)))*AX(1))-((cosd(x(4))*sind(x(6))
    +sind(x(4))*cosd(x(5))*cosd(x(6)))*AY(1))+((sind(x(4))*sind(x(5)))*AZ(1))
    -BX(1)))+((x(2)+((sind(x(4))*cosd(x(5))+cosd(x(4))*cosd(x(5))*sind(x(6)))*AX(1))
    -((sind(x(4))*sind(x(5))-cosd(x(4))*cosd(x(5))*cosd(x(6)))*AY(1))
    -((cosd(x(4))*sind(x(5)))*AZ(1))-BY(1))*(x(2)+((sind(x(4))*cosd(x(5))
    +cosd(x(4))*cosd(x(5))*sind(x(6)))*AX(1))-((sind(x(4))*sind(x(5))
    -cosd(x(4))*cosd(x(5))*cosd(x(6)))*AY(1))-((cosd(x(4))*sind(x(5)))*AZ(1))
    -BY(1)))+((x(3)+Tz_offset+((sind(x(5))*sind(x(6)))*AX(1))+((sind(x(5))*cosd(x(6)))*AY(1))
    +(cosd(x(5))*AZ(1))-BZ(1))*(x(3)+Tz_offset+((sind(x(5))*sind(x(6)))*AX(1))
    +((sind(x(5))*cosd(x(6)))*AY(1))+(cosd(x(5))*AZ(1))-BZ(1)))-((S(1))*(S(1)));

    ((x(1)+((cosd(x(4))*cosd(x(6))-sind(x(4))*cosd(x(5))*sind(x(6)))*AX(2))
    -((cosd(x(4))*sind(x(6))+sind(x(4))*cosd(x(5))*cosd(x(6)))*AY(2))
    +((sind(x(4))*sind(x(5)))*AZ(2))-BX(2))*(x(1)+((cosd(x(4))*cosd(x(6))
    -sind(x(4))*cosd(x(5))*sind(x(6)))*AX(2))-((cosd(x(4))*sind(x(6))
    +sind(x(4))*cosd(x(5))*cosd(x(6)))*AY(2))+((sind(x(4))*sind(x(5)))*AZ(2))
    -BX(2)))+((x(2)+((sind(x(4))*cosd(x(5))+cosd(x(4))*cosd(x(5))*sind(x(6)))*AX(2))
    -((sind(x(4))*sind(x(5))-cosd(x(4))*cosd(x(5))*cosd(x(6)))*AY(2))
    -((cosd(x(4))*sind(x(5)))*AZ(2))-BY(2))*(x(2)+((sind(x(4))*cosd(x(5))
    +cosd(x(4))*cosd(x(5))*sind(x(6)))*AX(2))-((sind(x(4))*sind(x(5))
    -cosd(x(4))*cosd(x(5))*cosd(x(6)))*AY(2))-((cosd(x(4))*sind(x(5)))*AZ(2))
    -BY(2)))+((x(3)+Tz_offset+((sind(x(5))*sind(x(6)))*AX(2))+((sind(x(5))*cosd(x(6)))*AY(2))
    +(cosd(x(5))*AZ(2))-BZ(2))*(x(3)+Tz_offset+((sind(x(5))*sind(x(6)))*AX(2))
    +((sind(x(5))*cosd(x(6)))*AY(2))+(cosd(x(5))*AZ(2))-BZ(2)))-((S(2))*(S(2)));

    ((x(1)+((cosd(x(4))*cosd(x(6))-sind(x(4))*cosd(x(5))*sind(x(6)))*AX(3)
    )-((cosd(x(4))*sind(x(6))+sind(x(4))*cosd(x(5))*cosd(x(6)))*AY(3))
    +((sind(x(4))*sind(x(5)))*AZ(3))-BX(3))*(x(1)+((cosd(x(4))*cosd(x(6))
    -sind(x(4))*cosd(x(5))*sind(x(6)))*AX(3))-((cosd(x(4))*sind(x(6))
    +sind(x(4))*cosd(x(5))*cosd(x(6)))*AY(3))+((sind(x(4))*sind(x(5)))*AZ(3))
    -BX(3)))+((x(2)+((sind(x(4))*cosd(x(5))+cosd(x(4))*cosd(x(5))*sind(x(6)))*AX(3))
    -((sind(x(4))*sind(x(5))-cosd(x(4))*cosd(x(5))*cosd(x(6)))*AY(3))
    -((cosd(x(4))*sind(x(5)))*AZ(3))-BY(3))*(x(2)+((sind(x(4))*cosd(x(5))
    +cosd(x(4))*cosd(x(5))*sind(x(6)))*AX(3))-((sind(x(4))*sind(x(5))
    -cosd(x(4))*cosd(x(5))*cosd(x(6)))*AY(3))-((cosd(x(4))*sind(x(5)))*AZ(3))
    -BY(3)))+((x(3)+Tz_offset+((sind(x(5))*sind(x(6)))*AX(3))+((sind(x(5))*cosd(x(6)))*AY(3))
    +(cosd(x(5))*AZ(3))-BZ(3))*(x(3)+Tz_offset+((sind(x(5))*sind(x(6)))*AX(3))
    +((sind(x(5))*cosd(x(6)))*AY(3))+(cosd(x(5))*AZ(3))-BZ(3)))-((S(3))*(S(3)));

    ((x(1)+((cosd(x(4))*cosd(x(6))-sind(x(4))*cosd(x(5))*sind(x(6)))*AX(4))
    -((cosd(x(4))*sind(x(6))+sind(x(4))*cosd(x(5))*cosd(x(6)))*AY(4))
    +((sind(x(4))*sind(x(5)))*AZ(4))-BX(4))*(x(1)+((cosd(x(4))*cosd(x(6))
    -sind(x(4))*cosd(x(5))*sind(x(6)))*AX(4))-((cosd(x(4))*sind(x(6))
    +sind(x(4))*cosd(x(5))*cosd(x(6)))*AY(4))+((sind(x(4))*sind(x(5)))*AZ(4))
    -BX(4)))+((x(2)+((sind(x(4))*cosd(x(5))+cosd(x(4))*cosd(x(5))*sind(x(6)))*AX(4))
    -((sind(x(4))*sind(x(5))-cosd(x(4))*cosd(x(5))*cosd(x(6)))*AY(4))
    -((cosd(x(4))*sind(x(5)))*AZ(4))-BY(4))*(x(2)+((sind(x(4))*cosd(x(5))
    +cosd(x(4))*cosd(x(5))*sind(x(6)))*AX(4))-((sind(x(4))*sind(x(5))
    -cosd(x(4))*cosd(x(5))*cosd(x(6)))*AY(4))-((cosd(x(4))*sind(x(5)))*AZ(4))
    -BY(4)))+((x(3)+Tz_offset+((sind(x(5))*sind(x(6)))*AX(4))+((sind(x(5))*cosd(x(6)))*AY(4))
    +(cosd(x(5))*AZ(4))-BZ(4))*(x(3)+Tz_offset+((sind(x(5))*sind(x(6)))*AX(4))
    +((sind(x(5))*cosd(x(6)))*AY(4))+(cosd(x(5))*AZ(4))-BZ(4)))-((S(4))*(S(4)));

    ((x(1)+((cosd(x(4))*cosd(x(6))-sind(x(4))*cosd(x(5))*sind(x(6)))*AX(5))
    -((cosd(x(4))*sind(x(6))+sind(x(4))*cosd(x(5))*cosd(x(6)))*AY(5))
    +((sind(x(4))*sind(x(5)))*AZ(5))-BX(5))*(x(1)+((cosd(x(4))*cosd(x(6))
    -sind(x(4))*cosd(x(5))*sind(x(6)))*AX(5))-((cosd(x(4))*sind(x(6))
    +sind(x(4))*cosd(x(5))*cosd(x(6)))*AY(5))+((sind(x(4))*sind(x(5)))*AZ(5))
    -BX(5)))+((x(2)+((sind(x(4))*cosd(x(5))+cosd(x(4))*cosd(x(5))*sind(x(6)))*AX(5))
    -((sind(x(4))*sind(x(5))-cosd(x(4))*cosd(x(5))*cosd(x(6)))*AY(5))
    -((cosd(x(4))*sind(x(5)))*AZ(5))-BY(5))*(x(2)+((sind(x(4))*cosd(x(5))
    +cosd(x(4))*cosd(x(5))*sind(x(6)))*AX(5))-((sind(x(4))*sind(x(5))
    -cosd(x(4))*cosd(x(5))*cosd(x(6)))*AY(5))-((cosd(x(4))*sind(x(5)))*AZ(5))
    -BY(5)))+((x(3)+Tz_offset+((sind(x(5))*sind(x(6)))*AX(5))+((sind(x(5))*cosd(x(6)))*AY(5))
    +(cosd(x(5))*AZ(5))-BZ(5))*(x(3)+Tz_offset+((sind(x(5))*sind(x(6)))*AX(5))
    +((sind(x(5))*cosd(x(6)))*AY(5))+(cosd(x(5))*AZ(5))-BZ(5)))-((S(5))*(S(5)));

    ((x(1)+((cosd(x(4))*cosd(x(6))-sind(x(4))*cosd(x(5))*sind(x(6)))*AX(6))
    -((cosd(x(4))*sind(x(6))+sind(x(4))*cosd(x(5))*cosd(x(6)))*AY(6))
    +((sind(x(4))*sind(x(5)))*AZ(6))-BX(6))*(x(1)+((cosd(x(4))*cosd(x(6))
    -sind(x(4))*cosd(x(5))*sind(x(6)))*AX(6))-((cosd(x(4))*sind(x(6))
    +sind(x(4))*cosd(x(5))*cosd(x(6)))*AY(6))+((sind(x(4))*sind(x(5)))*AZ(6))
    -BX(6)))+((x(2)+((sind(x(4))*cosd(x(5))+cosd(x(4))*cosd(x(5))*sind(x(6)))*AX(6))
    -((sind(x(4))*sind(x(5))-cosd(x(4))*cosd(x(5))*cosd(x(6)))*AY(6))
    -((cosd(x(4))*sind(x(5)))*AZ(6))-BY(6))*(x(2)+((sind(x(4))*cosd(x(5))
    +cosd(x(4))*cosd(x(5))*sind(x(6)))*AX(6))-((sind(x(4))*sind(x(5))
    -cosd(x(4))*cosd(x(5))*cosd(x(6)))*AY(6))-((cosd(x(4))*sind(x(5)))*AZ(6))
    -BY(6)))+((x(3)+Tz_offset+((sind(x(5))*sind(x(6)))*AX(6))+((sind(x(5))*cosd(x(6)))*AY(6))
    +(cosd(x(5))*AZ(6))-BZ(6))*(x(3)+Tz_offset+((sind(x(5))*sind(x(6)))*AX(6))
    +((sind(x(5))*cosd(x(6)))*AY(6))+(cosd(x(5))*AZ(6))-BZ(6)))-((S(6))*(S(6)))];

end
```

Figura A.5: Sistema de ecuaciones programadas en MATLAB.

En lo que respecta a las Figs. A.4 y A.5, muestran la ubicación de los puntos de conexión de las extremidades y el sistema de ecuaciones descrito en MATLAB respectivamente. Del sistema de ecuaciones (Fig. A.5) cabe aclarar que cada uno de los que parecen párrafos deben ser una sola línea pues representan a una de las seis ecuaciones que componen el sistema, se muestran así para que no salgan del borde de la hoja, sin embargo, la función **F** en realidad solo consta de seis líneas, donde cada línea es uno de estos párrafos.
Fuera de eso todo el código funciona tal cual se muestra, sin olvidar que todo el código de las Figs. A.1, A.2, A.3, A.4 y A.5 deben estar escritas en un solo archivo de MATLAB y que las variables X, Y, Z, α, β y γ están contenidas dentro del vector x como se observa en la Tabla A.1.

Tabla A.1: Forma en que se sustituyeron las variables en MATLAB.

x	x(1)
y	x(2)
z	x(3)
α	x(4)
β	x(5)
γ	x(6)

– –

A continuación, en la Figs. A.6, A.7, A.8, A.9, A.10 y A.11 se muestra las líneas de código que resuelven el cálculo de la cinemática inversa. La función se llama **CInversaFun** y recibe como parámetros las seis variables de la pose del robot, es decir, tres posiciones en "x", "y" y "z" y los tres ángulos de Euler a (α), b (β) y g (γ), además de un vector con las longitudes de la última posición del actuador ([0 0 0 0 0 0] para la primer ejecución). Y devuelve dos parámetros un vector con la longitud de cada uno de los vástagos comenzando por el de la extremidad S_1 y una cadena con la leyenda "Longitud dentro de rango" o "Longitud fuera de rango" dependiendo de si se pudo o no alcanzar dicha posición.

```
function [Estado_actuadores,L_Act] = CInversaFun(x,y,z,a,b,g,L)
```

Programa que se encarga de calcular las longitudes de cada uno de los actuadores del robot.

CORTES RUIZ HUGO JAVIER

Figura A.6: Encabezado de la función de cinemática inversa.

NOTA: No olvidar que todas las Figs. A.6, A.7, A.8, A.9, A.10 y A.11 deben estar contenidas en un solo archivo de MATLAB, se muestran separadas para aclarar de mejor manera el código.

Constantes y variables geométricas del robot:

Unidades: Grados, mm, s;

```
Dim_Excedida = 0;   %Bandera para verificar longitud de los eslabones (1-> Fuera de rango)
S_min = 274;        %Longitud mínima del brazo
S_max = 408;        %Longitud máxima del brazo
AngB = 60;          %Separación entre los vértices de la base 60° Hexágono Regular
AngPP = 14;         %Separación entre vertices contiguos
r_b = 200;          %Radio sobre el que se encuentran los vértices de la base
z_h = 21.5;         %Altura al centro de la junta universal
r_p = 115;          %Radio sobre el que se encuentran los vértices de la plataforma
t = 6.35;           %Espesor de la plataforma
alfa = a;           %ALFA Giro sobre el eje Z actual (primer rotación)(Se asigna la variable de entrada "a")
beta = b;           %BETA Giro sobre el eje X actual (segunda rotación)(Se asigna la variable de entrada "b")
gamma = g;          %GAMMA Giro sobre el eje Z actual (tercer rotación)(Se asigna la variable de entrada "g")
Tx = x;             %Coordenada X de la plataforma (Se asigna la variable de entrada "x")
Ty = y;             %Coordenada Y de la plataforma (Se asigna la variable de entrada "y")
Tz = z+303;         %Coordenada Z de la plataforma (Se asigna la variable de entrada "z")
L_Act = L;          %Vector con las longitudes de los actuadores de la última posición consultada
```

Figura A.7: Constantes y variables geométricas del robot.

En las Figs. A.6 y A.7 se muestran el encabezado y los parámetros geométricos del robot, se definen las dimensiones de las plataformas y algunos otras medidas de interés. Mientras en las Figs. A.8 y A.9 se observan el cálculo de la ubicación de cada uno de los puntos de conexión y la obtención de los vectores de posición y la matriz de rotación de acuerdo a los parámetro de entrada.

Vectores con la descripción geométrica del robot

```
AngV = (60-AngPP)/2; %Ángulo gamma ver ecuación 17 del documento de tesis
Teta_b = AngB*[0 1 2 3 4 5]; %Vector de ángulos entre los vértices de la  base
Teta_p = 60*[6 1 2 3 4 5]+[-AngV AngV -AngV AngV -AngV AngV];%Vector de ángulos entre los vértices de la plataforma

B_x = r_b*cosd(Teta_b);     %Componentes X de cada uno de los vectores que apuntan a los vértices de la base
B_y = r_b*sind(Teta_b);     %Componentes Y de cada uno de los vectores que apuntan a los vértices de la base
B_z = z_h*[1 1 1 1 1 1];    %Componentes Z de cada uno de los vectores que apuntan a los vértices de la base
B = [B_x ; B_y ; B_z];      %Matriz cuyos vectores columna son cada uno de los vectores a los vértices Bi

P_x = r_p*cosd(Teta_p);         %Componentes X de cada uno de los vectores que apuntan a los vértices de la plataforma
P_y = r_p*sind(Teta_p);         %Componentes Y de cada uno de los vectores que apuntan a los vértices de la plataforma
P_z = (-z_h)*[1 1 1 1 1 1]-t;   %Componentes Z de cada uno de los vectores que apuntan a los vértices de la plataforma
P = [P_x ; P_y ; P_z];          %Matriz cuyos vectores columna son cada uno de los vectores a los vértices Pi
```

Figura A.8: Descripción geométrica del robot en MATLAB.

Vectores que describen la posición deseada de la plataforma del robot

```
T = [Tx Ty Tz];     %Vector con la posición de la plataforma

%NOTA: MR(grados,eje) es una función que devuelve la matriz de rotación
%correspondiente al "eje" elegido rotado la cantidad "grados" (Revisar la función)

Rz1 = MR(alfa,3);   %Obtención de la matriz de rotación correspondiente al primer giro alrededor del eje Z
Rx1 = MR(beta,1);   %Obtención de la matriz de rotación correspondiente al segundo giro alrededor del eje X
Rz2 = MR(gamma,3);  %Obtención de la matriz de rotación correspondiente al tercer giro alrededor del eje Z

R_eulerSolid = Rz1*Rx1*Rz2;     %Obtención de la matriz general de rotación (multiplicación por la derecha -> eje actual)
                                %Convención utilizada por SolidWorks para los ángulos de Euler
```

Figura A.9: Cálculo de la posición deseada del robot.

Finalmente, en las Figs. A.10 y A.11, se observa el cálculo de las longitudes de cada una de las extremidades formadas como las columnas de la matriz S para posteriormente obtener la longitud de extensión de los actuadores de acuerdo a si la posición es alcanzable o no.

Cálculo de las dimensiones de los brazos

```
% Componente X del vector colineal al actuador
S_x = Tx+(R_eulerSolid(1,1)*P_x+R_eulerSolid(1,2)*P_y+R_eulerSolid(1,3)*P_z)-B_x;
% Componente Y del vector colineal al actuador
S_y = Ty+(R_eulerSolid(2,1)*P_x+R_eulerSolid(2,2)*P_y+R_eulerSolid(2,3)*P_z)-B_y;
% Componente Z del vector colineal al actuador
S_z = Tz+(R_eulerSolid(3,1)*P_x+R_eulerSolid(3,2)*P_y+R_eulerSolid(3,3)*P_z)-B_z;

% Matriz con los vectores S_i formados en cada columna de la matriz
S = [S_x ; S_y ; S_z];

% Magnitud o longitud del brazo
MagS = sqrt(dot(S,S));
```

Figura A.10: Cálculo de las longitudes de los brazos.

Cálculo de la cantidad de extención del actuador en [mm]

```
% Ciclo para verificar que la dimensión del brazo no exceda la capacidad de extención del actuador
for i = 1:6
    if MagS(i)<S_min || MagS(i)>S_max
        Dim_Excedida = 1;   %Bandera que indica error de dimensión (1->Error)
    end
end

% Cálculo de la dimensión del actuador
if Dim_Excedida == 0
   Estado_actuadores = 'Longitud dentro de rango';
   Mag_Act = round(MagS-S_min);     %Longitud requerida del actuador
   L_Act = cast(Mag_Act,'uint8');   %Conversión a datos de ocho bits para envío
else
   Estado_actuadores = 'Longitud fuera de rango';
end
```

```
end
```

Figura A.11: Cálculo de la extensión del actuador.

– –

Finalmente, el último programa implementado es el encargado de generar las matrices de rotación, la función se llama **MR** y recibe dos parámetros como entradas el eje y los grados que debe girar respecto de ese eje, arrojando como resultado la matriz de rotación correspondiente, esto con el fin de evitar el desarrollo manual que aunque se realizó se pueden cometer errores en las multiplicaciones de matrices, además que podemos adecuarnos a cualquier representación de orientación más allá de la de Euler para trabajos futuros.

```
function [ R ] = MR( Grados , Eje )
```

Función que genera matrices de rotación dados los grados y el eje sobre el que actua la rotación

HUGO CORTES RUIZ

Figura A.12: Encabezado de la función de matrices de rotación.

Programación de los casos

Unidades: grados,mm,s

```
% La variable de entrada "eje" recibe un entero entre [1,2,3] de lo que depende el eje de giro.
% La variable de entrada "grados" especifica justamente eso los grados a girar en [grados].
% Para Eje = 1 -> Rotación en X
% Para Eje = 2 -> Rotación en Y
% Para Eje = 3 -> Rotación en Z

switch Eje
    case 1
        R = [1 0 0;0 cosd(Grados) -sind(Grados); 0 sind(Grados) cosd(Grados)];
    case 2
        R = [cosd(Grados) 0 sind(Grados); 0 1 0; -sind(Grados) 0 cosd(Grados)];
    case 3
        R = [cosd(Grados) -sind(Grados) 0; sind(Grados) cosd(Grados) 0; 0 0 1];
end
```

```
end
```

Figura A.13: Casos implementados para cada una de las rotaciones.

Se observa en el encabezado de la función en la Fig. A.12 y el desarrollo de los casos en la Fig. A.13, el funcionamiento es bastante simple de acuerdo al número se escoge la forma de la matriz y se sustituye el valor del ángulo.

Apéndice B

Programa implementado en Arduino para comunicar todos los dispositivos.

A continuación se presenta el programa implementado en Arduino para conectar y comunicar la plataforma con la PC, en particular con MATLAB para poder operar al robot desde la interfaz diseñada.

```
//---------------------------------------
//         Librerias utilizadas:
//---------------------------------------
#include <PololuMaestro.h>
#include <EEPROM.h>
#include <Wire.h>
#include <SPI.h>
#include <SparkFunLSM9DS1.h>

//-----------------------------------------------------
//          Objeto para acceder a la IMU
//-----------------------------------------------------
LSM9DS1 imu;

//-----------------------------------------------------
//            Configuración de la IMU
//-----------------------------------------------------
void Configuracion()
{
  // Forma de comunicación:
  imu.settings.device.commInterface = IMU_MODE_I2C;

  // Dirección de los registros para el magnetómetro:
  imu.settings.device.mAddress = 0x1E; // 0x1E para comunicación I2C
  // Dirección de los registros para el acelerómetro y el giroscopio:
  imu.settings.device.agAddress = 0x6B; // 0x6B para comunicación I2C
}

//-----------------------------------------------------
```

Figura B.1: Librerías implementadas y configuración de la IMU.

En la Fig. B.1 se puede observar el inicio del programa, en el que se declaran las librerías a utilizar además de la configuración de la dirección de la unidad de medición inercial (IMU).

```
//---------------------------------------------------
//          Configuración del Girsocopio
//---------------------------------------------------

void GiroConfig()
{
  // Habilitación del giroscopio:
  imu.settings.gyro.enabled = true;
  // Escala del giroscopio entre: +|- 245, 500, or 2000 [grados/s]
  imu.settings.gyro.scale = 245;
  // Taza de muestreo del giroscopio:
  // 1 = 14.9    4 = 238
  // 2 = 59.5    5 = 476
  // 3 = 119     6 = 952
  imu.settings.gyro.sampleRate = 1; // 59.5Hz Taza de salida de datos
  // Frecuencia de corte del filtro pasa bajas:
  // Para tazas de muestreo menores a 238 no importa dicha frecuencia.
  imu.settings.gyro.bandwidth = 1;
  // Desactivación del modo de bajo consumo:
  imu.settings.gyro.lowPowerEnable = false;
  // Desactivación del filtro pasa altas (HPF)
  imu.settings.gyro.HPFEnable = false;
  // Frecuencia de corte del fltro pasa altas (ver sección 7.14)
  imu.settings.gyro.HPFCutoff = 1;
  // Cambio de dirección posistiva de los ejes:
  imu.settings.gyro.flipX = false;
  imu.settings.gyro.flipY = false;
  imu.settings.gyro.flipZ = false;
}
```

Figura B.2: Configuración del giroscopio.

```
//---------------------------------------------------
//          Configuración del Acelerómetro
//---------------------------------------------------

void AcelConfig()
{
  // Habilitación del acelerómetro:
  imu.settings.accel.enabled = true;
  // Habilitación de cada uno de los ejes del acelerómetro:
  imu.settings.accel.enableX = true;
  imu.settings.accel.enableY = true;
  imu.settings.accel.enableZ = true;
  // Configuración de la escala del acelerómetro +|-2, 4, 8, or 16 [g]
  imu.settings.accel.scale = 2;
  // Frecuencia de muestreo del acelerómetro (solo es independiente cuando no
  // esta activado el giroscopio)
  // 1 = 10 Hz     4 = 238 Hz
  // 2 = 50 Hz     5 = 476 Hz
  // 3 = 119 Hz    6 = 952 Hz
  imu.settings.accel.sampleRate = 1;
  // Ancho de banda para el filtro pasa bajas
  // 0 = 408 Hz    2 = 105 Hz
  // 1 = 211 Hz    3 = 50 Hz
  imu.settings.accel.bandwidth = 0;
  // Habilitación del modo de alta resolucion
  imu.settings.accel.highResEnable = true;
  // Establecimiento de la frecuencia de corte para el modo de alta resolucion
  // 0 = ODR/50    2 = ODR/9
  // 1 = ODR/100   3 = ODR/400
  imu.settings.accel.highResBandwidth = 0;
}
```

Figura B.3: Configuración del acelerómetro.

```
//---------------------------------------------------
//          Configuración del Magnetómetro
//---------------------------------------------------
void MagConfig()
{
  // Habilitacion del magnetometro
  imu.settings.mag.enabled = true; // Habilitacion del magnetometro
  // Establecimiento de la escala de medicion del magnetometro
  imu.settings.mag.scale = 12; // Se configura a 12 [Gauss]
  // Taza de muestreo del magnetometro
  // 0 = 0.625 Hz  4 = 10 Hz
  // 1 = 1.25 Hz   5 = 20 Hz
  // 2 = 2.5 Hz    6 = 40 Hz
  // 3 = 5 Hz      7 = 80 Hz
  imu.settings.mag.sampleRate = 5; // Se configura la taza de muestreo a 20Hz
  // Habilitacion de la compensación por temperatura
  imu.settings.mag.tempCompensationEnable = false;
  // Desempeño de la medicion sobre los ejes X y Y
  // 0 = Modo de bajo consumo      2 = Alto desempeño
  // 1 = Desempeño medio  3 = Maximo desempeño (superior al alto)
  imu.settings.mag.XYPerformance = 3;
  // Desempeño de la medición sobre el eje Z
  imu.settings.mag.ZPerformance = 3;
  // Habilitacion del modo de bajo consumo
  imu.settings.mag.lowPowerEnable = false;
  // Establecimiento del modo de operación del magnetometro
  // 0 = conversion continua
  // 1 = una sola conversion
  // 2 = modo de espera
  imu.settings.mag.operatingMode = 0;
}
```

Figura B.4: Configuración del magnetómetro.

En las Figs. B.2, B.3 y B.4 se observa la configuración del giroscopio, acelerómetro y magnetómetro respectivamente, se establecen velocidades de muestreo de datos, modos de operación así como activación de algunos filtros digitales.

```
void setupTemperature()
{
  // Se deshabilita el sensor de temperatura
  imu.settings.temp.enabled = false;
}

uint16_t initLSM9DS1()
{
  Configuracion(); // Establece la configuración del dispositivo
  GiroConfig(); // Configura el giroscopio
  AcelConfig(); // Configura el acelerometro
  MagConfig(); // Configura el magnetometro
  setupTemperature(); // Configura el sensor de temperatura

  return imu.begin();
}

//---------------------------------------------------------------
//      Variación del campo magnético dependiendo la región
//---------------------------------------------------------------
#define DECLINATION 4

//---------------------------------------------------
//     Selección de pines para transmisión serial
//---------------------------------------------------
#ifdef SERIAL_PORT_HARDWARE_OPEN
  #define maestroSerial SERIAL_PORT_HARDWARE_OPEN
#else
  #include <SoftwareSerial.h>
  SoftwareSerial maestroSerial(10, 11); //10RX-11TX
#endif
```

Figura B.5: Configuración del sensor de temperatura y de los pines de la comunicación serial.

En la Fig. B.5 se observa la configuración del sensor de temperatura, además de la selección de los pines para la transmisión serial con el controlador de los servomotores.

```
//-------------------------------------------------------
//  Creación del objeto maestro para controlar los servos
//-------------------------------------------------------
MicroMaestro maestro(maestroSerial);

//-----------------------------------------
//            Variables Globales
//-----------------------------------------
byte EntradaSerial = 0;
int Bandera  = 0;
int Direccion = 0;
byte L_act[6] = {0,0,0,0,0,0};
double ValServoD[6] = {0,0,0,0,0,0};
int ValServo[6] = {0,0,0,0,0,0};
double GGX = 0,GGY = 0,GGZ = 0,AAX = 0,AAY = 0,AAZ = 0,MMX = 0,MMY = 0,MMZ = 0;
int PreMovimiento = 0;
int Movimiento = 0;
int s = 0;
float DX = 0,DY = 0,DZ = 0;
int IDX = 0,IDY = 0,IDZ = 0;
byte EDX = 0,EDY = 0,EDZ = 0;
float VX = 0,VY = 0,VZ = 0;
float VXANT = 0,VYANT = 0,VZANT = 0;
float AXANT = 0,AYANT = 0,AZANT = 0;
float tiempo = 0;
float tant = 0;
float t = 0;
float roll = 0,heading = 0, pitch = 0;
int Iroll = 0,Iheading = 0, Ipitch = 0;
byte Eroll = 0,Eheading = 0, Epitch = 0;
```

Figura B.6: Variables globales utilizadas y creación del objeto para manipular el controlador de los motores.

En la Fig. B.6 se puede observar la declaración de todas las variables utilizadas en el programa y la creación del objeto para manipular y tener acceso al controlador de los servomotores.

```
//-----------------------------------------------
//   Configuración de la comunicación serial
//-----------------------------------------------
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  maestroSerial.begin(9600);
  Serial.begin(9600);
  while (!Serial) {
    ;
  }

  //-------------------------------------------------------
  //               Inicialización de la IMU
  //-------------------------------------------------------
  uint16_t status = initLSM9DS1();

}
```

Figura B.7: Configuración de la comunicación serial e inicialización de la IMU.

Continuando con la configuración de los dispositivos en la Fig. B.7 se observa el establecimiento de la velocidad de transmisión serial, así como la inicialización de la IMU.

```
//-------------------------------------------------------
//              Cuerpo principal del programa
//-------------------------------------------------------
void loop()
{
     if (Serial.available()) {
        EntradaSerial = Serial.read();
        delay(20);
      if(EntradaSerial==200){
        for(int i = 7;i<13;i++){
          L_act[i] = EEPROM.read(i);
          Serial.write(L_act[i]);
        }
      }
      //Lectura de las extenciones de los vástagos
      if(EntradaSerial>=0&&EntradaSerial<=140){
          L_act[0] = EntradaSerial;
          EEPROM.write(0, L_act[0]);
          for(int i=1;i<6;i++){
            L_act[i] = Serial.read();
            EEPROM.write(i, L_act[i]);
            delay(20);
          }
          for(int i=0;i<6;i++){
            L_act[i] = EEPROM.read(i);
            ValServoD[i] = (L_act[i]*30)+4000;
            ValServo[i] = ValServoD[i];
          }
          maestro.setTarget(0, ValServo[0]);
          maestro.setTarget(1, ValServo[1]);
          maestro.setTarget(2, ValServo[2]);
          maestro.setTarget(3, ValServo[3]);
          maestro.setTarget(4, ValServo[4]);
          maestro.setTarget(5, ValServo[5]);
          for(int i = 7;i<13;i++){
            L_act[i] = EEPROM.read(i);
            Serial.write(L_act[i]);
          }
      }
    }
  //-------------------------------------------------
  //        Actualización de los datos del sensor
  //-------------------------------------------------

  if ( imu.gyroAvailable() )
  {
    imu.readGyro();
  }
  if ( imu.accelAvailable() )
  {
    imu.readAccel();
  }
  if ( imu.magAvailable() )
  {
    imu.readMag();
  }
```

Figura B.8: Recepción de la longitud de extensión de los actuadores y lectura de la IMU.

En la Fig. B.8 se observa el cuerpo principal del programa, el funcionamiento descrito de manera general es el siguiente. Como primer paso se analiza si el dato recibido se encuentra dentro del rango de extensión de los actuadores o corresponde solo a una indicación de lectura de datos. Si se trata del primer caso y se encuentra dentro de la longitud de extensión de los actuadores, los valores se almacenan en la memoria *eeprom* y posteriormente son enviados al controlador para que extienda o retraiga los actuadores. Para el segundo caso, si se trata del dato 200, solo se envían los valores de la extensión actual de los actuadores del robot.

```
//-------------------------------------------------------------
//  Almacenamiento de los datos de posición y orientación
//-------------------------------------------------------------

    pitch = printroll(imu.ay,imu.az);
    roll = printpitch(imu.ax,imu.ay,imu.az);
    heading = printyaw(imu.mx,imu.my);

    DX = imu.ax*10;
    DY = imu.ay*10;
    DZ = imu.az*10;
    IDX = int(DX);
    EDX = byte(IDX);
    IDY = int(DY);
    EDY = byte(IDY);
    IDZ = int(DZ);
    EDZ = byte(IDZ);
    Ipitch = int(pitch);
    Iroll = int(roll);
    Iheading = int(heading);
    Epitch = byte(Ipitch);
    Eroll = byte(Iroll);
    Eheading = byte(Iheading);
```

Figura B.9: Lectura y conversión de los datos de posición y orientación.

Finalmente, en las Figs. B.9 y B.10 se observan la conversión de tipos de datos y el cálculo de los ángulos de rotación alrededor de cada uno de los ejes. Para la rotación sobre el eje X y Y simplemente se descompone el vector de gravedad normal a la superficie de la tierra y con sus componentes se obtiene el grado de inclinación de la IMU, mientras que con el magnetómetro se genera una especie de brújula y medir la rotación sobre el eje Z.

```
//Calculo de los ángulos de rotacion sobre los ejes

float printroll( float ay, float az)
{
  float roll = atan2(ay, az);
  roll  *= 180.0 / PI;
  return roll;
}
float printpitch(float ax, float ay, float az)
{
  float pitch = atan2(-ax, sqrt(ay * ay + az * az));
  pitch *= 180.0 / PI;
  return pitch;
}
float printyaw( float mx, float my)
{
  float heading;
  if (my == 0)
    heading = (mx < 0) ? PI : 0;
  else
    heading = atan2(my, mx);

  heading -= DECLINATION * PI / 180;
  if (heading > PI) heading -= (2 * PI);
  else if (heading < -PI) heading += (2 * PI);
  heading *= (180.0 / PI)-190;
  return heading;
}
```

Figura B.10: Cálculo de la rotación de la plataforma sobre cada uno de los ejes.

Apéndice C

Funciones implementadas para el modelo de bloques en Simulink.

Bloque de generación de trayectorias

Para empezar con la simulación y obtención del estado de velocidad del robot se requiere de una trayectoria establecida para el robot, por lo que dentro del la función denominada generador de trayectorias se implementa el programa que se muestra en la Fig. C.1.

```
function POSE  = GenTrayectoriaPos(t)
```

Función para generar la trayectoria del robot

TRAYECTORIA DE POSICIÓN Nota: la altura mínima del robot debe ser 298.6 [mm] Sin olvidar que esta distancia se mide desde el marco de referencia fijo a la base al marco de referencia atado al centro de masa de la plataforma, por lo que el marco se encuentra inmerso en medio de la plataforma esto con el fin de hacer coincidir los resultados con los obtenidos en solidworks.

Funciones aleatorias para la pose del robot:

```
%Posición del robot:
    X = (t^2)*5;
    Y = (t^2)*3;
    Z = 298.6+((t^2)*20);

%Orientación del robot:
    A = (t^2)*2.5;
    B = (t^2);
    G = (t^2)*1.5;

    POSE = [X Y Z A B G];
```

```
end
```

Figura C.1: Programa implementado en el bloque para generar trayectorias.

Se puede observar de la Fig. C.1 que simplemente se programan funciones lineales para la posición y orientación del robot en función del tiempo y se envían como vector a la salida de la función.

Panel de configuración de la función *fsolve*

De la Fig. C.2 se puede observar la ventana de configuración del bloque *Function Block*, dicho bloque permite utilizar la función *fsolve* que se encarga de resolver la cinemática directa, por lo que es necesaria para la verificación de la resultados de posición, dentro de la casilla **MATLAB function** se declara la función a utilizar y los parámetros o variables que recibe, en este caso se utilizó la función *fsolve*, donde el $@X$ define la incógnita y los dos parámetros siguientes $u(1:6)$ y $u(7:12)$ definen las longitudes de los eslabones como primer entrada y el vector con la condición inicial para comenzar a iterar como segunda entrada. Los demás campos se dejan con los valores por defecto.

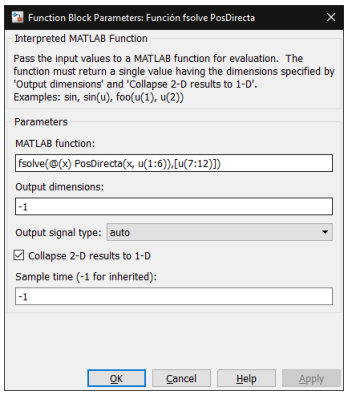


Figura C.2: Ventana de configuración del bloque de funciones “Function Block”.

Bloque de solución de la cinemática directa de velocidad

```
function EstVel  = VelDirecta(POSE,VelAct) % Unidades [grados/s] y [mm/s]
```

Análisis de velocidad del robot paralelo utilizando teoría de tornillos

CORTES RUIZ HUGO JAVIER Función que recibe como entrada la pose [POSE] del robot para actualizar las direcciones de los tornillos y la velocidad de los actuadores [VelAct] del mismo para obtener como resultado el estado de velocidad del robot. NOTA: La velocidad angular está dada en [rad/s] y la lineal en [mm/s]

Constantes y variables geométricas del robot:

Unidades: Grados, mm, s;

```
AngB = 60;          %Separación entre los vértices de la base 60° Hexágono Regular
AngPP = 14;         %Separación entre vertices contiguos
r_b = 200;          %Radio sobre el que se encuentran los vértices de la base
z_h = 21.5;         %Altura al centro de la junta universal
r_p = 115;          %Radio sobre el que se encuentran los vértices de la plataforma
t = 6.35;           %Espesor de la plataforma
alfa = POSE(4);           %PSI Giro sobre el eje Z actual (primer rotación)
beta = POSE(5);           %THETA Giro sobre el eje X actual (segunda rotación)
gamma = POSE(6);          %PHI Giro sobre el eje Z actual (tercer rotación)
Tx = POSE(1);             %Coordenada X de la plataforma
Ty = POSE(2);             %Coordenada Y de la plataforma
Tz = POSE(3);             %Coordenada Z de la plataforma
O_x = 0;                  %Offset del polo de referencia en la coordenada X
O_y = 0;                  %Offset del polo de referencia en la coordenada Y
O_z = 298.6;              %Offset del polo de referencia en la coordenada Z
```

Figura C.3: Programa implementado en el bloque *VelDirecta* I.

En lo que respecta a la Fig. C.3 se refiere a la primer parte del programa implementado dentro del bloque denominado *VelDirecta*, se puede observar la declaración de las constantes geométricas del robot y la asignación de las variables de entrada de la función.
A continuación en las Figs. C.5 se puede observar la definición de los puntos en la plataforma móvil (P) y en la base fija (B) vistos desde sus respectivos marcos de referencia, además de un par de vectores de posición T y $T2$ con los que se ubica el origen del marco de referencia móvil (T) y el mismo origen pero corrigiendo la distancia al polo de referencia ($T2$).

Vectores con la descripción geométrica del robot

```
AngV = (60-AngPP)/2; %Separación
Teta_b = AngB*[0 1 2 3 4 5]; %Vector de ángulos entre los vértices de la  base
Teta_p = 60*[6 1 2 3 4 5]+[-AngV AngV -AngV AngV -AngV AngV];%Vector de ángulos entre los vértices de la plataforma
```

Definición de los vectores a cada vértice en la base visto desde el marco de referencia en la base

```
B_x = r_b*cosd(Teta_b);     %Componentes X de cada uno de los vectores que apuntan a los vértices de la base
B_y = r_b*sind(Teta_b);     %Componentes Y de cada uno de los vectores que apuntan a los vértices de la base

% NOTA: En caso de que los resultados no concuerden con solidWorks recordar
% que el sistema de referencia en solid está anclado a la cara inferior de
% la base fija y no sobre la superior como se plantea en el análisis
% cinemático. No debería haber problema si se utiliza un marco de
% referencia distinto para las mediciones pero no esta de más mencionarlo.

B_z = z_h*[1 1 1 1 1 1];    %Componentes Z de cada uno de los vectores que apuntan a los vértices de la base
B = [B_x ; B_y ; B_z];      %Matriz cuyos vectores columna son cada uno de los vectores a los vértices Bi
```

Figura C.4: Programa implementado en el bloque *VelDirecta* II.

Definición de los vectores a cada vértice en la plataforma visto desde el marco de referencia anclado a la plataforma

```
P_x = r_p*cosd(Teta_p);             %Componentes X de cada uno de los vectores que apuntan a los vértices de la plataforma
P_y = r_p*sind(Teta_p);             %Componentes Y de cada uno de los vectores que apuntan a los vértices de la plataforma
P_z = (-z_h)*[1 1 1 1 1 1]-(t/2);   %Componentes Z de cada uno de los vectores que apuntan a los vértices de la plataforma


P = [P_x ; P_y ; P_z];          %Matriz cuyos vectores columna son cada uno de los vectores a los vértices Pi
```

Vectores que describen la posición deseada de la plataforma del robot

```
T = [Tx Tx Tx Tx Tx Tx;Ty Ty Ty Ty Ty Ty;Tz Tz Tz Tz Tz Tz];     %Matriz con la posición de la plataforma

% Se define una segunda matriz de posición ubicando el polo de referencia menos el desfasamiento de las coordenadas:
T2 = [Tx-O_x Tx-O_x Tx-O_x Tx-O_x Tx-O_x Tx-O_x;Ty-O_y Ty-O_y Ty-O_y Ty-O_y Ty-O_y Ty-O_y;Tz-O_z Tz-O_z Tz-O_z Tz-O_z Tz-O_z Tz-O_z];

% Se define la matriz de esta manera dada la composición de la matriz P con
% los puntos de los vértices de las plataformas, al multiplicar con la
% matriz de rotación más adelante el resultado sera una matriz igual de 3x6
% por lo que el vector de posición T más bien debe ser la siguiente matriz:

%
%       | TX TX TX TX TX TX |
%   T = | TY TY TY TY TY TY |
%       | TZ TZ TZ TZ TZ TZ |
%
```

Figura C.5: Programa implementado en el bloque *VelDirecta* III.

Mientras que en la Fig. C.6 se puede observar ahora la definición de la matriz de rotación utilizada para representar la orientación del robot, además de los vectores PB y $PB2$ que representan los puntos o vértices en la plataforma móvil respecto del marco de referencia fijo. Lo que se realizó fue una transformación al multiplicar por la matriz de rotación y se le sumó el vector de posición al marco de referencia móvil, utilizando los vectores T y $T2$ para los vectores B y $B2$ respectivamente.

Definición de la matriz de rotación de acuerdo a la pose del robot

```
%NOTA: MR(angulo,eje) es una función que devuelve la matriz de rotación
% correspondiente al "eje" elegido rotado la cantidad "angulo" (Revisar la
% función) (ángulo --> [grados])

Rz1 = MRot(alfa,3);   %Obtención de la matriz de rotación correspondiente al primer giro alrededor del eje Z
Rx1 = MRot(beta,1);   %Obtención de la matriz de rotación correspondiente al segundo giro alrededor del eje X
Rz2 = MRot(gamma,3);  %Obtención de la matriz de rotación correspondiente al tercer giro alrededor del eje Z

R_eulerSolid = Rz1*Rx1*Rz2;
```

Definición de los vectores a los vértices en las plataformas vistos desde el marco de referencia en la base

```
PB = T+R_eulerSolid*P;
PB2 = T2+R_eulerSolid*P;
```

Transpuesta de las matrices con las ubicaciones de los vértices para operar con ellos

```
BASE = B';       %Matriz de posiciones de los vértices de la base.
PLAT = PB';      %Matriz de posiciones de los vértices de la plataforma móvil.
PLAT2 = PB2';    %Matriz de posiciones de los vértices de la plataforma móvil ubicando el polo de referencia.
```

Figura C.6: Programa implementado en el bloque *VelDirecta* IV.

Definición de los tornillos en cada extremidad

```
T7 = zeros(6,6);
for i = 1:6
    % Tornillo 3 -> 2$3  ---------------------------------------------------
    t3 = (PLAT(i,:)-BASE(i,:));
    ut3 = t3/sqrt(dot(t3,t3));  %PARTE PRIMARIA DE LA LINEA DE PLÜCKER
    %Linea de plücker recíproca
    t7 =  ut3;
    T7(i,:) = [t7 cross(t7,-PLAT2(i,:))];
end
```

Definimos el jacobiano

```
J = T7; %El jacobiano es la misma matriz T7 con las lineas de plücker en cada fila
```

Definimos el operador de polaridad: \Delta

```
I3X3 = [1 0 0;0 1 0;0 0 1];
CERO = [0 0 0; 0 0 0; 0 0 0];
Delta = [CERO I3X3;I3X3 CERO];
```

Figura C.7: Programa implementado en el bloque *VelDirecta* V.

De la Fig. C.7 se puede observar la parte del programa correspondiente a la obtención de las líneas de Plücker. Se obtiene la parte primaria como la resta de los vectores $PLAT$ y $BASE$, mientras la parte dual se obtiene del producto cruz con el vector de posición al polo de referencia con la corrección de offset (producto cruz con el vector $PLAT2$). Además, se define el jacobiano como el conjunto de las líneas de Plücker de todas las extremidades en cada una de las filas. Finalmente, se define el operador de polaridad a partir de matrices identidad y de matrices de ceros de tamaño tres por tres.

Matriz A = J*Delta

```
A = J*Delta;
Ainv = inv(A);
```

Definición del estado de velocidad del robot:

```
SP = VelAct';
```

Obtención de la velocidad del los actuadores:

```
Est = Ainv*SP;
EstVel = Est';
```

Figura C.8: Programa implementado en el bloque *VelDirecta* VI.

En la Fig. C.8 se observa la definición de los últimos vectores y matrices para llegar a la ecuación de entrada-salida de la que para el caso de cinemática directa de velocidad se despeja el estado de velocidad del robot (revisar ecuación (4.33)).

Bloque de solución de la cinemática inversa de velocidad

Para el caso de la cinemática inversa de velocidad se tiene de igual manera un bloque en el que se implementa dicha función, la cabecera del programa se puede observar en la Fig. C.9 de la que varían únicamente uno de los parámetros de entrada. A diferencia del anterior ahora se recibe el estado de velocidad del robot y se obtiene la velocidad de los actuadores.

```
function VelAct  = VelInversa(POSE,EstVel) % Unidades [rad/s] y [mm/s]
```

Análisis de velocidad del robot paralelo utilizando teoría de tornillos

CORTES RUIZ HUGO JAVIER Función que recibe como entrada la pose [POSE] del robot para actualizar las direcciones de los tornillos y el estado de velocidad [EstVel] del mismo para obtener como resultado las velocidades de los actuadores. NOTA: La orientación esta dada por los ángulos de euler

Figura C.9: Programa implementado en el bloque *VelInversa* I.

Se omiten las demás partes del programa hasta la definición de la matriz A y el estado de velocidad (Fig. C.10) debido a que el código es el mismo que para el caso anterior de cinemática directa. La diferencia radica justamente en el despeje ahora de las velocidades de los actuadores en lugar del estado de velocidad.

Matriz A = J*Delta

```
A = J*Delta;
```

Definición del estado de velocidad del robot:

```
VOP = EstVel';
```

Obtención de la velocidad del los actuadores:

```
VelA = A*VOP;
VelAct = VelA';
```

Figura C.10: Programa implementado en el bloque *VelInversa* II.

Bloque de transformación de velocidades angulares

Por último, se muestra en la Fig. C.11 el programa implementado en el bloque transformación de velocidades, consiste básicamente de tres ecuaciones que relacionan las derivadas de los ángulos de Euler con las velocidades angulares en los ejes X, Y y Z. El bloque recibe como entrada la pose del mecanismo y las derivadas de los ángulos de Euler.

```
function [wx,wy,wz] =  Transformacion(POSE,ap,bp,gp)
```

Transformación entre formas de representación de la rotación del robot:

Función que transforma las velocidades ángulares vistas como la representación de los ángulos de euler a la forma convencional sobre los ejes X, Y y Z. Recibe como entrada la pose actual del robot y las derivadas respecto del tiempo de las rotaciones representadas a partir de los ángulos de euler. a, b y g son alfa, beta y gamma respectivamente. ap = da/dt; bp = db/dt; gp = dg/dt;

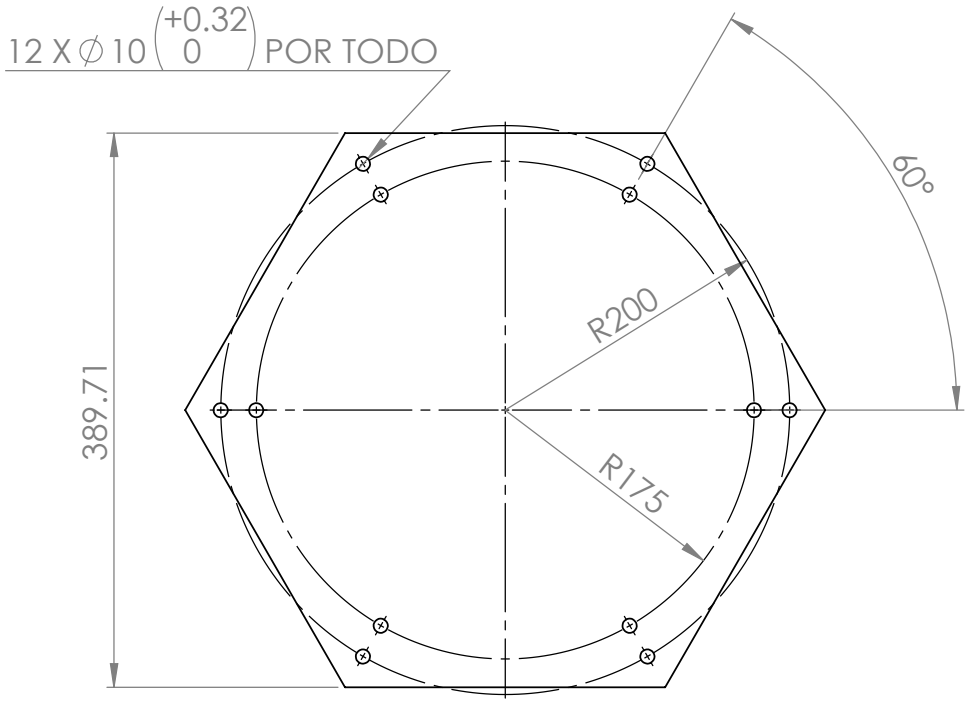
```
% Velocidad angular sobre el eje X
wx = (bp*cosd(POSE(4)))+(gp*sind(POSE(5))*sind(POSE(4)));
% Velocidad angular sobre el eje Y
wy = -(((-bp)*sind(POSE(4)))+(gp*sind(POSE(5))*cosd(POSE(4))));
% Velocidad angular sobre el eje Z
wz = ap+(gp*cosd(POSE(5)));
```

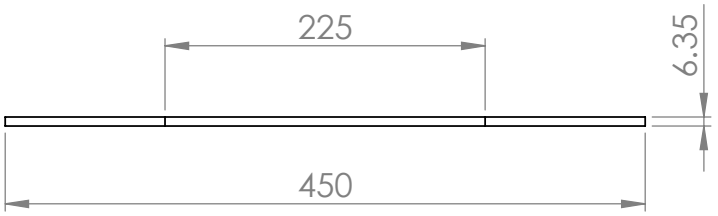
Figura C.11: Programa implementado en el bloque *VelInversa* III.

Apéndice D

Dibujos técnicos

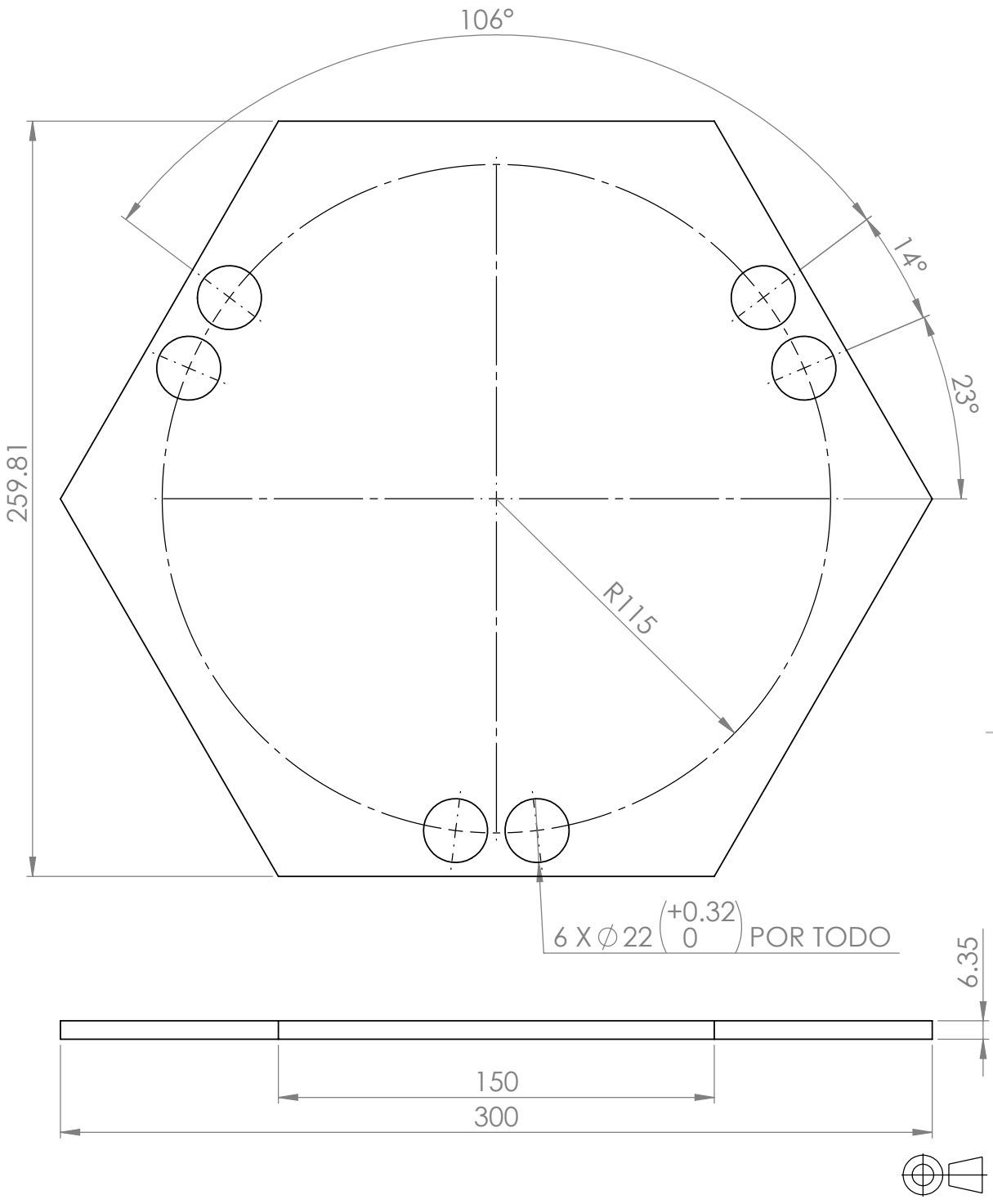
Plataforma fija



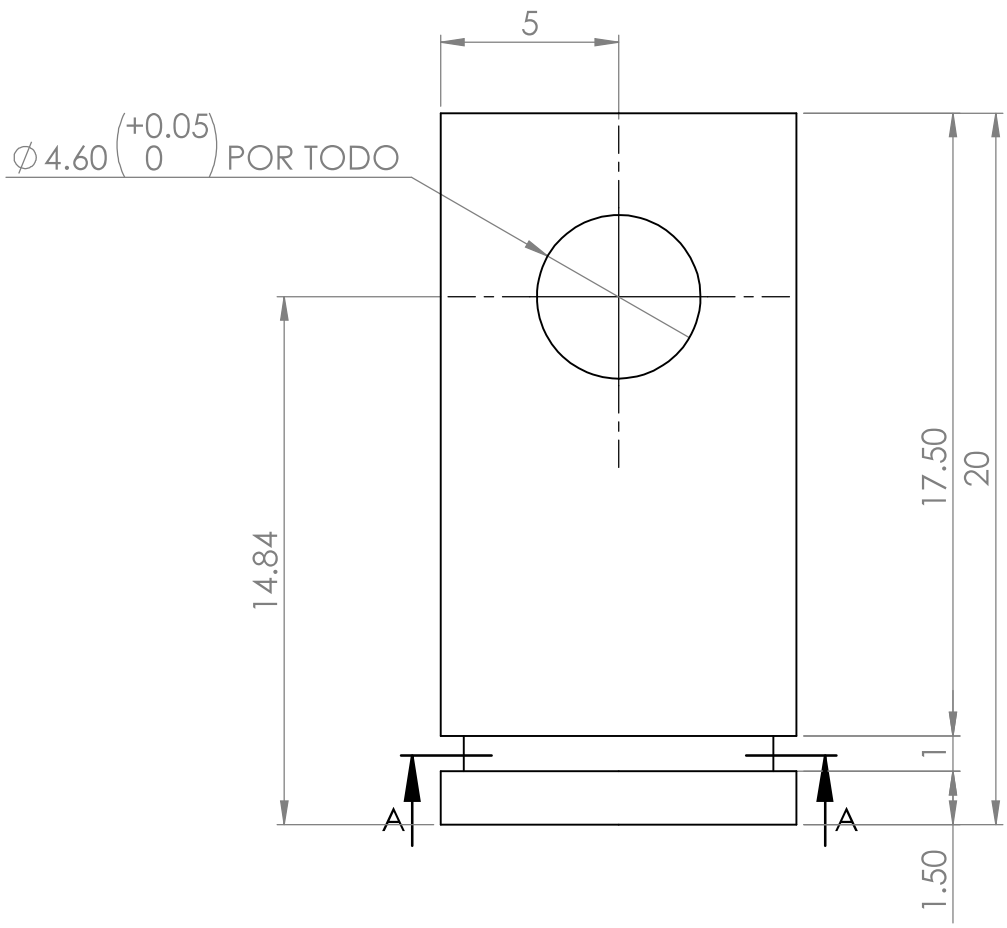


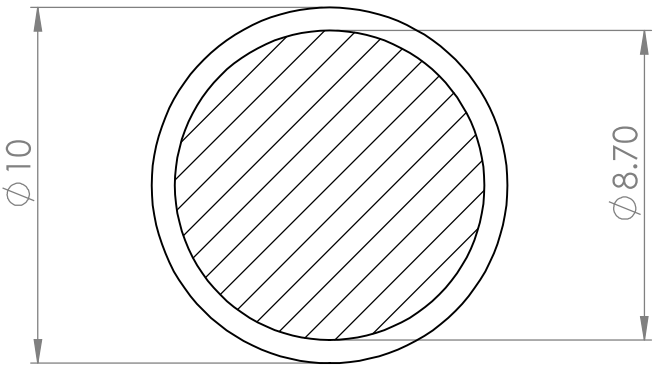
SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM	ACABADO: FRESADO		REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN	
				Universidad Tecnológica de la Mixteca		
	NOMBRE	FIRMA	FECHA	TÍTULO: Plataforma fija		
DIBUJ.	HUGO CORTES					
VERIF.	DR. MANUEL ARIAS					
APROB.	DR. MANUEL ARIAS					
FABR.	HUGO CORTES					
CALID.				MATERIAL: NYLAMID M	N.º DE DIBUJO 01	A4
				PESO:	ESCALA:1:5	HOJA 1 DE 1

PLATAFORMA MÓVIL



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM	ACABADO: FRESADO		REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN	
				Universidad Tecnológica de la Mixteca		
	NOMBRE	FIRMA	FECHA	TÍTULO: Plataforma móvil		
DIBUJ.	HUGO CORTES					
VERIF.	DR. MANUEL ARIAS					
APROB.	DR. MANUEL ARIAS					
FABR.	HUGO CORTES					
CALID.				MATERIAL: NYLAMID M	N.º DE DIBUJO 02	A4
				PESO:	ESCALA:1:2	HOJA 1 DE 1





SECCIÓN A-A
ESCALA 5 : 1

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM	ACABADO: TORNEADO, FRESADO	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA / REVISIÓN

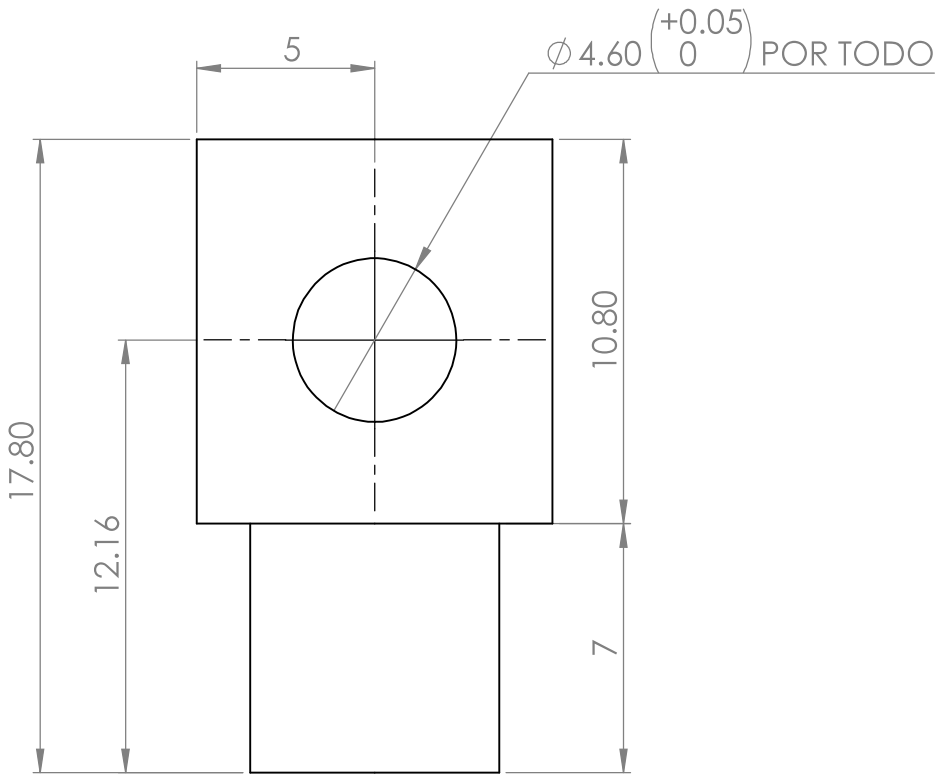
Universidad Tecnológica de la Mixteca

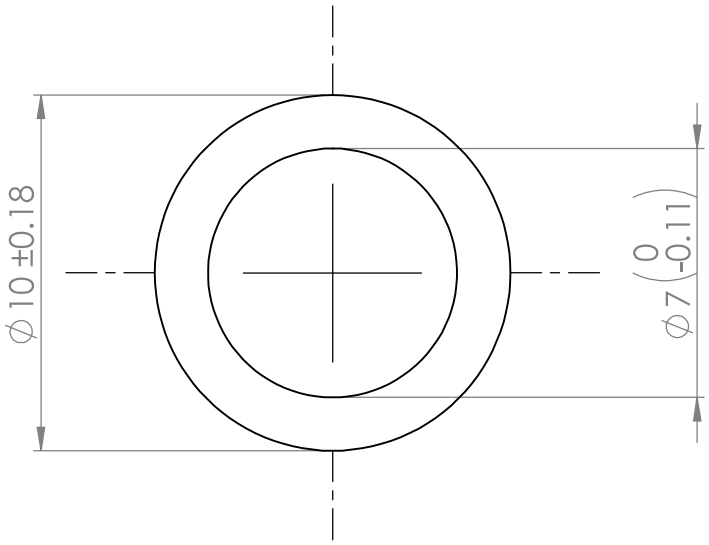
	NOMBRE	FIRMA	FECHA
DIBUJ.	HUGO CORTES		
VERIF.	DR. MANUEL ARIAS		
APROB.	DR. MANUEL ARIAS		
FABR.	HUGO CORTES		
CALID.			

TÍTULO:

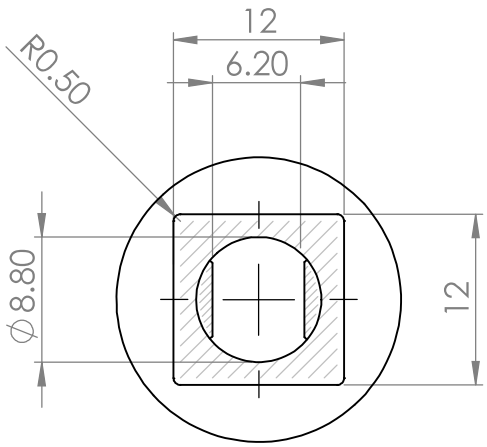
Pasador Base

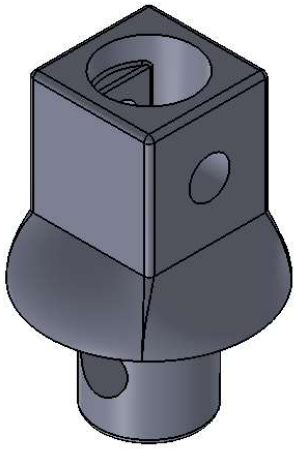
MATERIAL: ASTM A36	N.º DE DIBUJO 03	A4	
PESO:	ESCALA:5:1	HOJA 1 DE 1	

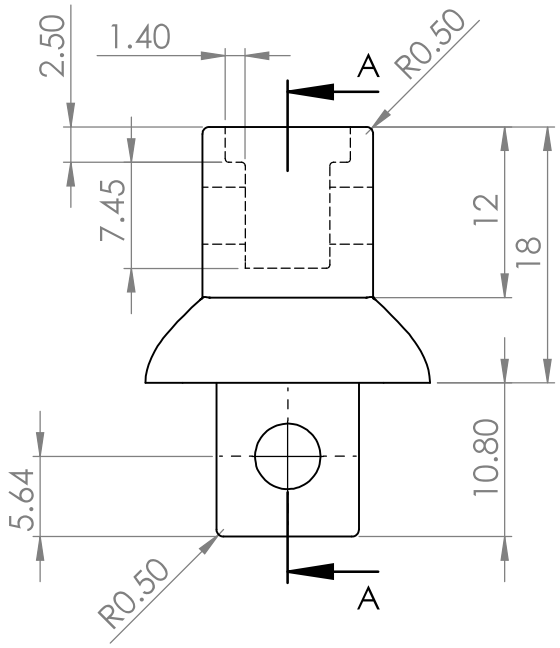


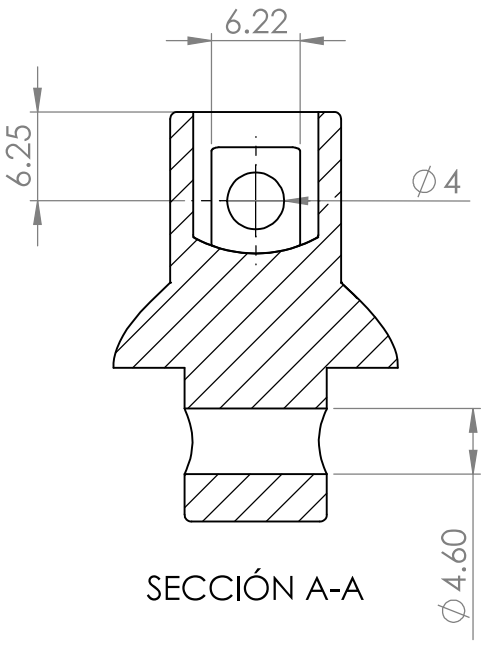


SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM	ACABADO: TORNEADO, FRESADO		REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN	
				Universidad Tecnológica de la Mixteca		
	NOMBRE	FIRMA	FECHA	TÍTULO: Pasador Junta Universal		
DIBUJ.	HUGO CORTES					
VERIF.	DR. MANUEL ARIAS					
APROB.	DR. MANUEL ARIAS					
FABR.	HUGO CORTES					
CALID.				MATERIAL: ASTM A36	N.º DE DIBUJO 04	A4
				PESO:	ESCALA:2:1	HOJA 1 DE 1

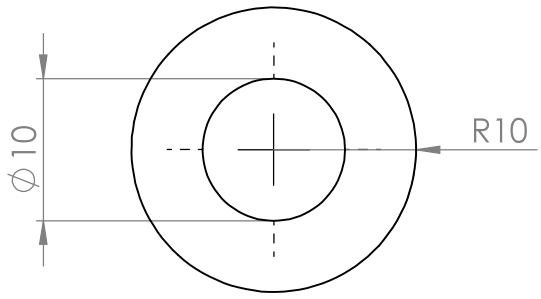








SECCIÓN A-A



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM TOLERANCIAS: ± 0.1 mm	ACABADO: IMPRESIÓN 3D (modelado por deposición fundida)	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA / REVISIÓN

Universidad Tecnológica de la Mixteca

	NOMBRE	FIRMA	FECHA
DIBUJ.	HUGO CORTES		
VERIF.	DR.MANUEL ARIAS		
APROB.	DR.MANUEL ARIAS		
FABR.	HUGO CORTES		
CALID.			

TÍTULO:

Soporte vástago del actuador

MATERIAL: ABS

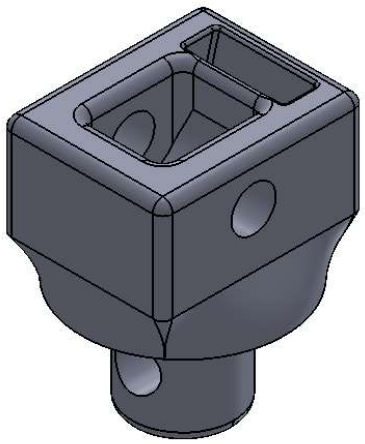
N.º DE DIBUJO: 05

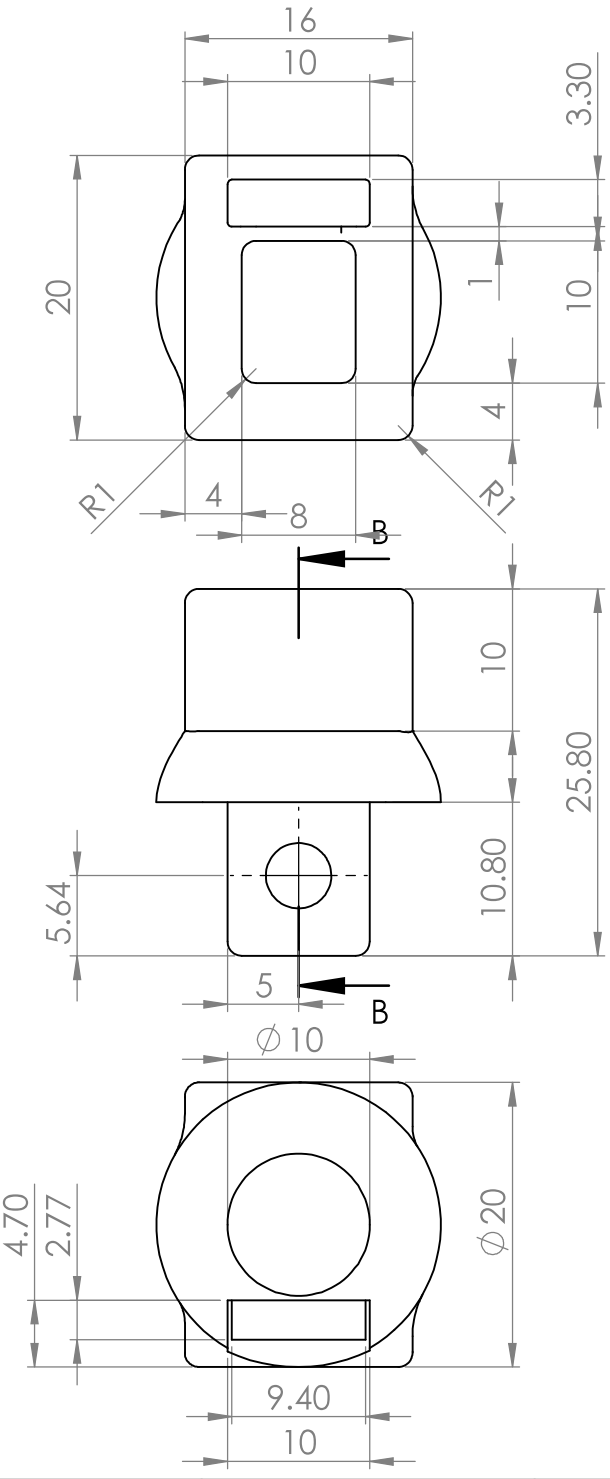
A4

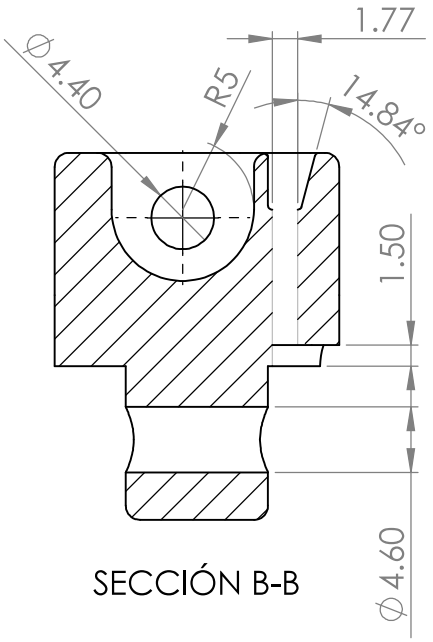
PESO:

ESCALA:2:1

HOJA 1 DE 1







SECCIÓN B-B

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM TOLERANCIAS: ± 0.1 mm	ACABADO: IMPRESIÓN 3D (modelado por deposición fundida)	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN			
			Universidad Tecnológica de la Mixteca				
	NOMBRE	FIRMA	FECHA				TÍTULO:
DIBUJ.	HUGO CORTES						Soporte base del actuador
VERIF.	DR. MANUEL ARIAS						
APROB.	DR. MANUEL ARIAS						
FABR.	HUGO CORTES						
CALID.				MATERIAL: ABS		N.º DE DIBUJO 06	A4
				PESO:		ESCALA:2:1	HOJA 1 DE 1