

Trabajo de Fin de Grado

**Grado en Ingeniería en Tecnologías Industriales**

# **Análisis cinemático y dinámico de la plataforma Hexaglide**

**MEMORIA**

**Autor:** Rita Roca Taxonera  
**Director:** Federico Thomas  
**Codirector:** Lluís Ros  
**Convocatoria:** Septiembre 2016



Escuela Técnica Superior  
De Ingeniería Industrial de Barcelona



## Resumen

El trabajo que se presenta a continuación consiste en el análisis y diseño de la plataforma Hexaglide. Dicha plataforma es un robot paralelo de seis grados de libertad cuyo fin es controlar la posición de una bola situada sobre sí misma. La plataforma se une mediante seis barras de longitud constante a un conjunto de seis guías. Por estas guías se desplazan los actuadores lineales que mueven la plataforma. Por un lado, se han utilizado juntas cardan para unir las barras a la plataforma y, por otro, juntas esféricas para unir éstas a los actuadores lineales.

El objetivo de este trabajo es realizar un estudio previo de la plataforma Hexaglide cinemático y dinámico para posteriormente poder realizar el control del robot.

En primer lugar, se han resuelto el problema cinemático inverso y el directo, que relacionan las posiciones de los actuadores en sus respectivas guías con la posición y orientación de la plataforma y viceversa. Asimismo se ha obtenido el Jacobiano Cinemático Inverso, que relaciona las velocidades angulares y lineales de la plataforma móvil con las velocidades de los actuadores.

También se ha realizado un estudio del espacio de trabajo del robot y de las posibles singularidades que pueden afectar y que deben ser evitadas para poder asegurar en todo momento el correcto control de la plataforma. Para estudiar las posibles singularidades se ha partido del Jacobiano Cinemático inverso.

Finalmente, se ha realizado el estudio dinámico de la plataforma, resolviendo la dinámica directa y la inversa, que relacionan las fuerzas y momentos de los actuadores con el movimiento de la plataforma y viceversa.

Los distintos aspectos de estudio comentados se han introducido como funciones de un programa realizado con *Matlab*<sup>®</sup>, para facilitar su utilización. También se ha utilizado para el presente proyecto, la librería de *Matlab*<sup>®</sup> llamada "*Robotics Toolbox*".

Este trabajo ha sido realizado en el laboratorio de robots paralelos del "Institut de Robòtica i Informàtica Industrial" situado en el edificio de la Facultad de Matemáticas y Estadística de la UPC.



## Sumario

<b>RESUMEN</b>	<b>1</b>
<b>SUMARIO</b>	<b>3</b>
<b>1. INTRODUCCIÓN</b>	<b>5</b>
1.1. Objetivos del proyecto .....	5
1.2. Alcance del proyecto .....	6
<b>2. ESTADO DEL ARTE</b>	<b>7</b>
<b>3. PLATAFORMA HEXAGLIDE</b>	<b>9</b>
3.1. Representación de la posición y la orientación.....	9
3.2. Esquema de la plataforma Hexaglide .....	11
<b>4. CINEMÁTICA</b>	<b>17</b>
4.1. Cinemática inversa .....	17
4.2. Cinemática directa.....	19
<b>5. JACOBIANO</b>	<b>23</b>
5.1. Jacobiano cinemático inverso .....	23
<b>6. ESPACIO DE TRABAJO</b>	<b>27</b>
<b>7. SINGULARIDADES</b>	<b>29</b>
7.1. Singularidades inversas .....	29
7.1.1. Límites articulares .....	30
7.1.2. Aplicación a la plataforma de estudio.....	31
7.2. Singularidades directas .....	31
7.2.1. Aplicación a la plataforma de estudio.....	32
<b>8. DINÁMICA</b>	<b>34</b>
8.1. El problema dinámico directo.....	35
8.1.1. Translación de la plataforma .....	37
8.1.2. Orientación de la plataforma .....	39
8.2. El problema dinámico inverso.....	42
<b>9. IMPACTO ECONÓMICO Y AMBIENTAL</b>	<b>44</b>
<b>CONCLUSIONES</b>	<b>47</b>

---

<b>AGRADECIMIENTOS</b>	<b>49</b>
<b>BIBLIOGRAFÍA</b>	<b>50</b>
Referencias bibliográficas .....	50

# 1. Introducción

Los robots paralelos son cada vez más utilizados, especialmente cuando se necesita poder transportar grandes cargas, o realizar operaciones de alta precisión con mucha rapidez. El presente trabajo se enmarca en este contexto, y viene motivado por un trabajo fin de grado previo en el que se desarrolla el control de la posición y el balanceo de una bola lanzada sobre un plano, utilizando para ello un robot paralelo hexápodo de tipo Gough-Stewart [1]. Ahora, tras conseguirse con éxito este control, se desea poderlo efectuar en un nuevo robot, de tipo Hexaglide, que resulta más ventajoso por su mayor espacio de trabajo. La intención de este proyecto es proporcionar las herramientas de base necesarias para, en un futuro próximo, poder efectuar este control con suficiente facilidad.

## 1.1. Objetivos del proyecto

Específicamente, el objetivo de este proyecto consiste en realizar un análisis cinemático y dinámico lo más completo posible de la plataforma Hexaglide. Ello requiere resolver los problemas cinemático directo e inverso, estudiar todas las posibles singularidades del robot, determinar su espacio de trabajo, y resolver los problemas dinámico directo e inverso. De esta forma, se obtienen los algoritmos necesarios para

- Simular el movimiento del robot ante determinadas variaciones de sus grados de libertad actuados.
- Determinar la región de *poses* realmente alcanzables dentro del espacio de tareas.
- Anticipar si habrá configuraciones donde se pierda el control (singularidades directas) o aquellas en las que haya una disminución de la destreza del efector (singularidades inversas).
- Simular el movimiento que resulta de aplicar ciertos perfiles de fuerza en los actuadores.
- Determinar las fuerzas de actuación que se requieren para el seguimiento de una trayectoria deseada.
- Dimensionar los actuadores de acuerdo con las tareas que se pretendan efectuar.

## 1.2. Alcance del proyecto

El proyecto cubre todos los aspectos relativos a la modelización geométrica del robot Hexaglide, y desarrolla soluciones algorítmicas para todos sus problemas cinemáticos y dinámicos, ya sean directos o inversos. También abarca el estudio de la zona de trabajo del robot: la determinación del espacio de trabajo disponible –es decir, las posiciones y orientaciones de la plataforma que se pueden alcanzar – así como la detección de posibles singularidades en su interior. El objetivo, en este punto, es determinar una zona segura de trabajo para la cual la plataforma quede suficientemente alejada de cualquier tipo de singularidad.

Para facilitar el estudio del robot, los algoritmos asociados se han implementado en *Matlab*<sup>®</sup>, automatizando todos los cálculos numéricos y proporcionando representaciones gráficas que favorezcan la comprensión de los resultados.

## 2. Estado del arte

Si clasificamos los robots en función de su topología, encontramos dos tipos distintos: los robots serie i los robots paralelos. En la actualidad los robots serie son los más habituales y conocidos.

El robot serie consiste en una cadena cinemática abierta cuyos actuadores están conectados, como su nombre indica, en serie. En el caso del robot paralelo, los actuadores están anclados directamente a la base del robot, creando una cadena cinemática cerrada.



Fig. 2.1. a) Robot serie b) robot paralelo

La estructura del robot paralelo posee varias ventajas. La principal ventaja respecto al robot serie, es que puede manejar cargas superiores y posee alta rigidez, ya que el peso está repartido equitativamente entre las distintas cadenas cinemáticas [2].

Al manipular carga, los actuadores de los robots serie son sometidos principalmente a esfuerzos de flexión, mientras que en el robot paralelo, los esfuerzos son principalmente de tracción y compresión, permitiéndole manejar cargas superiores. Este mismo hecho es el que favorece que el robot paralelo posea mayor precisión que el robot serie y que pueda alcanzar mayores velocidades y aceleraciones.

Cabe también destacar que, el robot paralelo, al tener los motores situados en partes fijas de la estructura, reduce las inercias producidas en los movimientos, a diferencia de los robots serie, donde los motores están situados en partes móviles, añadiendo además una carga adicional que debe ser soportada por el robot.



Por otro lado, el robot paralelo cuenta con un conjunto de desventajas respecto al robot serie. En primer lugar, cuenta con un espacio de trabajo más reducido respecto al que puede llegar a tener un robot serie. También posee una cinemática más compleja de calcular que el robot serie y por último, el manejo de las singularidades es poco extrapolable, ya que depende mucho de cada estructura en concreto.

El uso y estudio de los robots paralelos ha crecido en los últimos años considerablemente, ya que sus ventajas, lo hacen especialmente atractivo para realizar operaciones en las que se requiera precisión y rapidez.

Dentro de los robots paralelos, predominan especialmente los Hexápodos, que poseen seis grados de libertad para la plataforma móvil: tres translaciones y tres rotaciones. En la presente memoria se estudiará uno de ellos: la plataforma Hexaglide.

### 3. Plataforma Hexaglide

La plataforma Hexaglide es un tipo de robot paralelo de seis grados de libertad. Ésta está formada por un cuerpo rígido (la plataforma móvil) conectado a una base mediante seis cadenas cinemáticas independientes, cada una de ellas con motor propio. Estas cadenas cinemáticas consisten en una barra cuyos extremos se hallan conectados a través de articulaciones esféricas a un raíl, por un lado, y a la plataforma móvil, por el otro.

En la plataforma Hexaglide (Fig. 3.1.) en la que se basa este trabajo, los raíles son verticales y paralelos dos a dos.

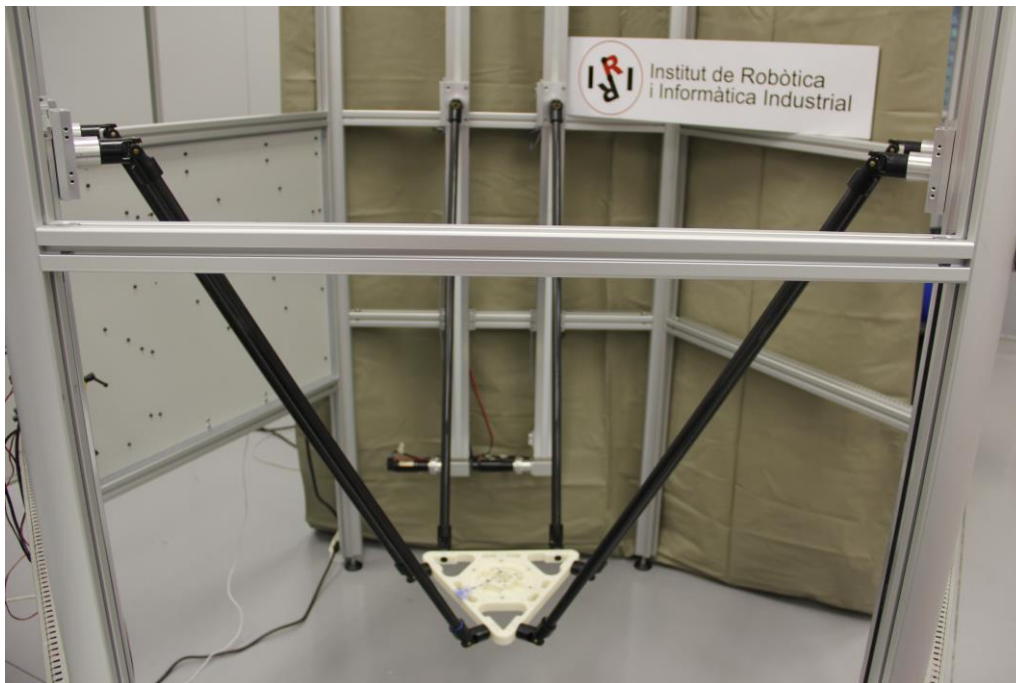


Fig. 3.1. Plataforma Hexaglide del “Institut de Robòtica i Informàtica Industrial”

#### 3.1. Representación de la posición y la orientación

El mínimo número de coordenadas para describir la *pose*  $(x, y, z, \theta_x, \theta_y, \theta_z)$  de un cuerpo en un espacio euclídeo es de seis. Se define una referencia  $i$ , que consiste en un origen  $O_i$  y tres ejes de coordenadas ortogonales  $(x_i, y_i, z_i)$  fijos en un determinado cuerpo. Si expresamos la posición de

dicho cuerpo relativa a una referencia  $j$ , tenemos:

$${}^jO_i = \begin{pmatrix} {}^jO_i^x \\ {}^jO_i^y \\ {}^jO_i^z \end{pmatrix} \quad (3.1)$$

Asimismo, podemos representar la orientación relativa de un cuerpo a través de la matriz de rotación:

$${}^jR_i = \begin{pmatrix} \hat{x}_i \cdot \hat{x}_j & \hat{y}_i \cdot \hat{x}_j & \hat{z}_i \cdot \hat{x}_j \\ \hat{x}_i \cdot \hat{y}_j & \hat{y}_i \cdot \hat{y}_j & \hat{z}_i \cdot \hat{y}_j \\ \hat{x}_i \cdot \hat{z}_j & \hat{y}_i \cdot \hat{z}_j & \hat{z}_i \cdot \hat{z}_j \end{pmatrix} \quad (3.2)$$

Las matrices ortonormales de rotación de un ángulo  $\theta$  alrededor de los ejes  $x,y,z$  son respectivamente:

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \quad (3.3)$$

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \quad (3.4)$$

$$R_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.5)$$

A partir de estas matrices se puede representar cualquier orientación usando el sistema de tres ángulos independientes que se desee, como por ejemplo:

- Ángulos de Euler:  $R = R_z(\phi)R_y(\theta)R_z(\psi)$
- Roll-pitch-yaw:  $R = R_x(\theta_r)R_y(\theta_p)R_z(\theta_y)$

Para poder representar una *pose* se utiliza una matriz de 4x4 llamada *transformada en coordenadas homogéneas*  ${}^jT_i$ :

$${}^j P = \begin{pmatrix} {}^j R_i & t \\ 0_{1 \times 3} & 1 \end{pmatrix} {}^i P = {}^j T_i \cdot {}^i P \quad (3.6)$$

Donde  $t$  es el vector de translación entre los orígenes de coordenadas.

En la librería *Robotics Toolbox* [3] de *Matlab*<sup>®</sup> existen funciones para facilitar estas transformaciones y otros muchas herramientas útiles para el estudio de robots.

### 3.2. Esquema de la plataforma Hexaglide

Para la construcción del modelo geométrico del robot mediante el programa *Matlab*<sup>®</sup>, se ha dividido el mismo en tres sistemas distintos: la estructura con base hexagonal que sostiene el robot (base), las guías por las que se desplazan las barras y la plataforma.

Para la creación del sistema base (prisma con base hexagonal), se ha tomado como origen el punto situado a altura cero en el centro de la base hexagonal. Para representar cada una de las aristas laterales del prisma, se realiza la rotación de los ejes de coordenadas del origen y una traslación del sistema de coordenadas igual al radio de la circunferencia  $r_b$  en la que el hexágono está circunscrito.

$$TB_i = \begin{bmatrix} R_z \left( \frac{2\pi \cdot i}{6} \right) & (0, -r_b, 0)^T \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (3.7)$$

Para la creación del sistema de guías, se ha realizado un procedimiento similar: se calcula el ángulo de giro y la translación necesaria. También se realiza una translación a lo largo del eje 'z', ya que las guías están elevadas  $z_g = 300mm$  respecto al suelo:

Para  $i =$  número impar:

$$TG_i = TB_i \cdot \begin{bmatrix} R_z \left( \frac{-\pi}{6} \right) & (-x_g, 0, z_g)^T \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (3.8)$$

Para  $i =$  número par:

$$TG_i = TB_i \cdot \begin{bmatrix} R_z\left(\frac{\pi}{6}\right) & (-x_g, 0, z_g)^T \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (3.9)$$

Donde  $x_g = 354,7\text{mm}$ .

El sistema plataforma consta de dos partes: la plataforma y unas barras cortas perpendiculares situadas a los lados de la plataforma que sirven de unión con las barras que mueven la plataforma. Para la construcción de la plataforma, se han calculado los ángulos y translaciones necesarios para obtener los puntos de unión con las barras.

Para ello, ha sido necesario definir las variables que se muestran a continuación y que se pueden ver representadas en las figuras (3.2), (3.3) y (3.4):

- $h_1 = \frac{2}{3} \sqrt{1707552} \text{ (mm)}$
- $\alpha_1 = \tan^{-1}\left(\frac{60\sin(5\pi/6)}{h_1 + 60\cos(5\pi/6)}\right) \text{ (rad)}$
- $k = \sqrt{5391.8289} \text{ (mm)}$
- $\beta = \sin^{-1}\left(\frac{\sin(\pi/2)42.33}{k}\right) \text{ (rad)}$
- $\alpha_2 = \tan^{-1}\left(\frac{k \sin((5\pi/6) - \beta)}{h_1 + k \cos((5\pi/6) - \beta)}\right) \text{ (rad)}$
- $s_1 = \frac{60\sin(\pi/6)}{\sin \alpha_1} \text{ (mm)}$
- $s_2 = \frac{k \sin((\pi/6) + \beta)}{\sin \alpha_2} \text{ (mm)}$

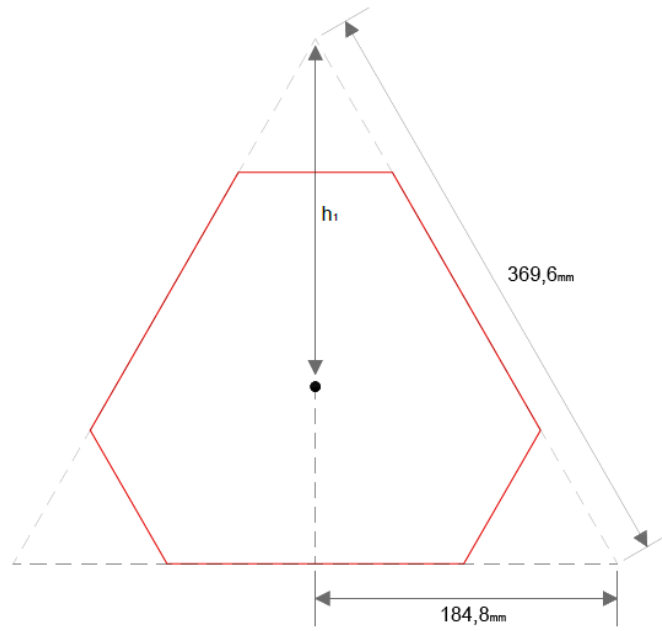


Fig. 3.2. Representación de las variables (1)

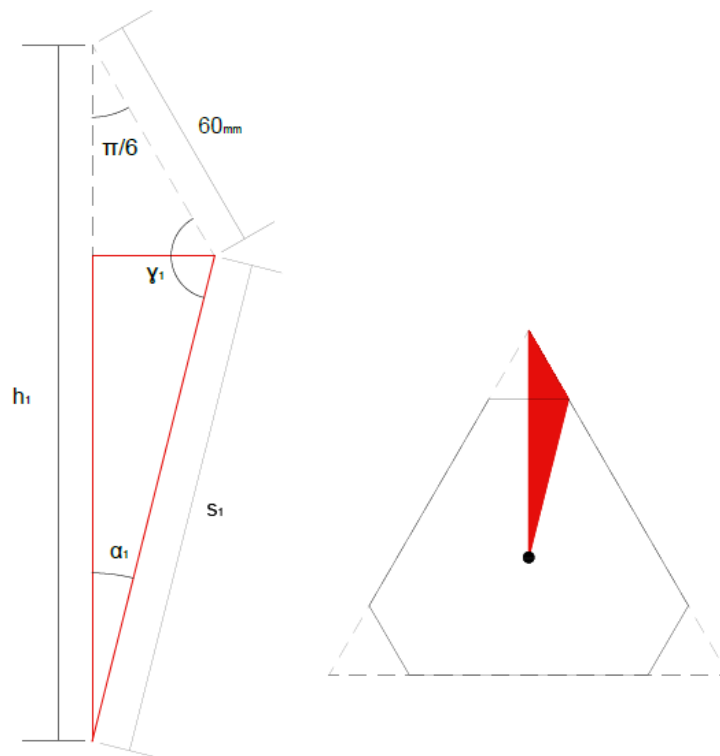


Fig. 3.3. Representación de las variables (2)

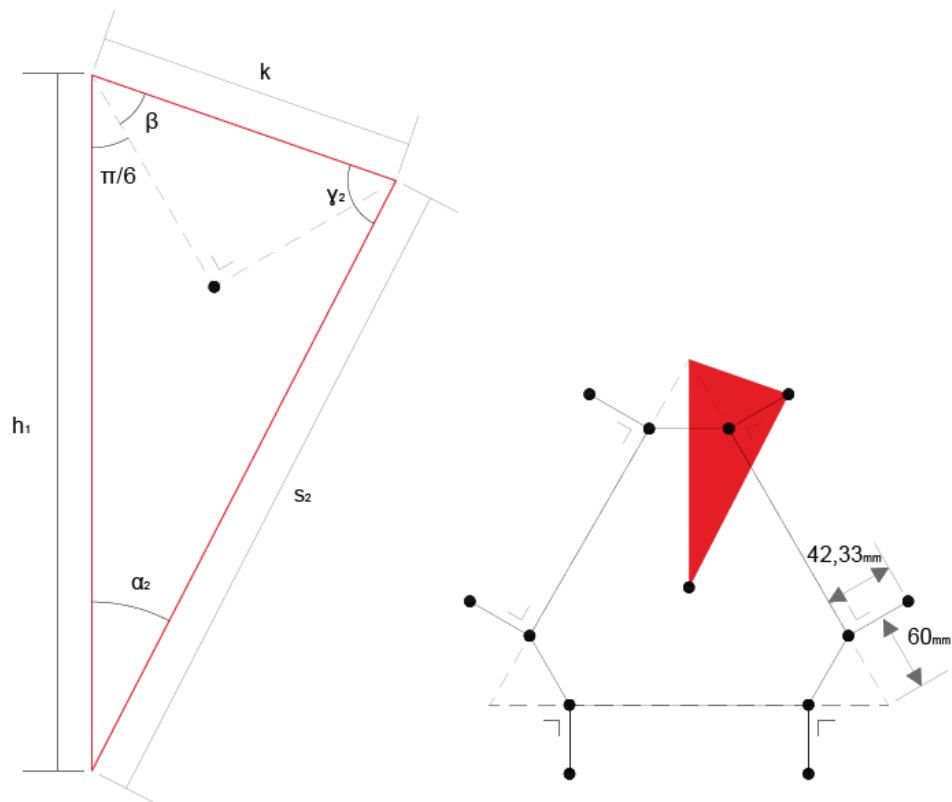


Fig. 3.4. Representación de las variables (3)

Las transformaciones aplicadas para obtener los distintos puntos de la plataforma son:

Para  $i =$  número impar:

$$TP1_i = TB_i \cdot \begin{bmatrix} R_z \left( \frac{2\pi(i-1)}{6} - \alpha_1 + \frac{\pi}{2} \right) & (0, -s_1, 0)^T \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (3.10)$$

$$TP2_i = TB_i \cdot \begin{bmatrix} R_z \left( \frac{2\pi(i-1)}{6} - \alpha_2 + \frac{\pi}{2} \right) & (0, -s_2, 0)^T \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (3.11)$$

Para  $i =$  número par:

$$TP1_i = TB_i \cdot \begin{bmatrix} R_z \left( \frac{2\pi(i-1)}{6} + \alpha_1 + \frac{\pi}{2} \right) & (0, -s_1, 0)^T \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (3.12)$$

$$TP2_i = TB_i \cdot \begin{bmatrix} R_z \left( \frac{2\pi(i-1)}{6} + \alpha_2 + \frac{\pi}{2} \right) & (0, -s_2, 0)^T \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (3.13)$$

Donde TP1 se refiere a la transformación para obtener los seis puntos de unión de la plataforma, y TP2 se refiere a la transformación para obtener los seis extremos de las barras cortas unidas a la plataforma.

El conjunto de ecuaciones arriba planteado ha sido introducido en el programa realizado con *Matlab*®. Para ello, han sido creadas las siguientes funciones: **MainHexaglide.m**, que actúa como programa principal desde el cual se llama al resto de funciones (Figura 3.5) y en el cual se inicializan las variables globales y se realizan una serie de cálculos necesarios para el resto de funciones

```

INDEX
1.Draw Hexaglide using inverse kinematics
2.Forward Kinematics: calculate the platform position knowing the position of the actuators
3.Video Hexaglide
4.Jacobian Kinematic Matrix
5.Workspace for a fixed orientation
6.Forward Dynamics: calculate the movement of the platform knowing the forces of the actuators
7.Forward Dynamics: path of a point of the platform
8.Inverse Dynamics: calculate forces of the actuators knowing the movement of the platform
fx Choose an option: |

```

Fig. 3.5. Respuesta de la función **MainHexaglide.m** al ser llamada

y **DrawHexaglide.m**, que sirve para representar la plataforma Hexaglide dados los siguientes parámetros: TB, TG, TP1, TP2 y la matriz de transformación de la plataforma (TT). Estos últimos parámetros están definidos en **MainHexaglide.m**, ya que para la mayoría de las funciones son necesarios.

A continuación se puede observar una imagen del resultado final con la plataforma en la posición de referencia:



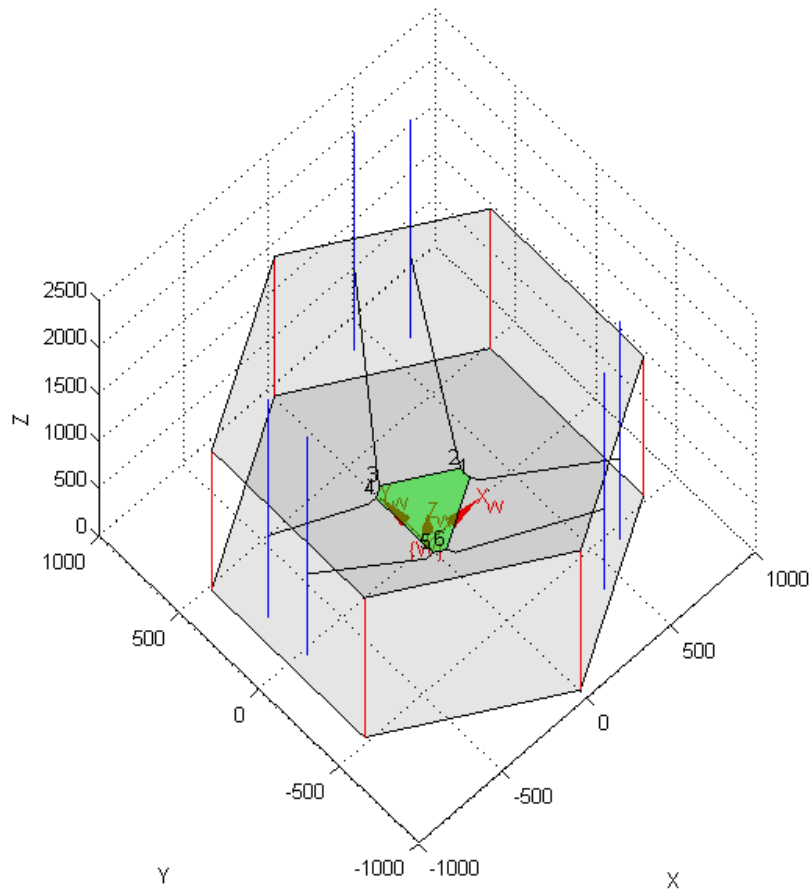


Fig. 3.6. Representación de la plataforma Hexaglride con el programa *Matlab*<sup>®</sup>

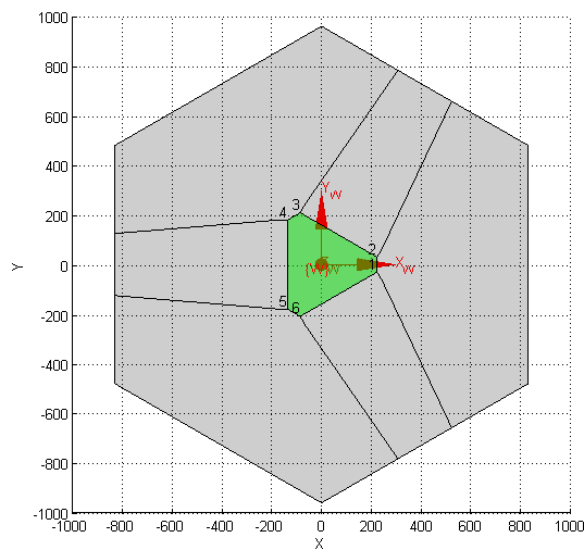


Fig. 3.7. Representación en planta de la plataforma Hexaglride con el programa *Matlab*<sup>®</sup>

## 4. Cinemática

La cinemática es la rama de la física que estudia los movimientos de cuerpos sin tener en cuenta las fuerzas ni momentos involucrados. En el campo de la robótica se parte de la idealización de los robots como sistemas de cuerpos rígidos conectados por articulaciones. Existen dos tipos de estudio de la cinemática de un robot, en función de cuáles son los datos conocidos y cuáles son las incógnitas de las ecuaciones. Estos dos tipos son la cinemática inversa y la cinemática directa, que se definen a continuación.

### 4.1. Cinemática inversa

La cinemática inversa es la técnica que permite determinar el movimiento de una cadena de articulaciones para lograr que un actuador final se ubique en una posición concreta. El cálculo de la cinemática inversa se efectúa resolviendo una serie de ecuaciones cuya solución normalmente no es única.

En el caso del robot paralelo Hexaglide, la cinemática inversa consiste en, dada la posición de la plataforma que se desea, conocer la posición del extremo de cada una de las barras en sus respectivos raíles.

Para calcular la cinemática inversa, se ha estudiado una sola barra, ya que la cinemática inversa de una de las barras es extrapolable a las otras cinco.

El sistema de estudio, por tanto, queda reducido a tres elementos: un punto de la plataforma, un raíl y una barra (por un lado unida a la plataforma y por el otro, unida al raíl). Si se toma la unión de la barra con la plataforma, el movimiento que puede realizar la barra queda restringido a una esfera cuyo radio es la longitud de la misma barra. Tomando el raíl como una recta, se producen dos intersecciones con la esfera. Estas dos intersecciones son las dos posibles soluciones para el problema planteado.

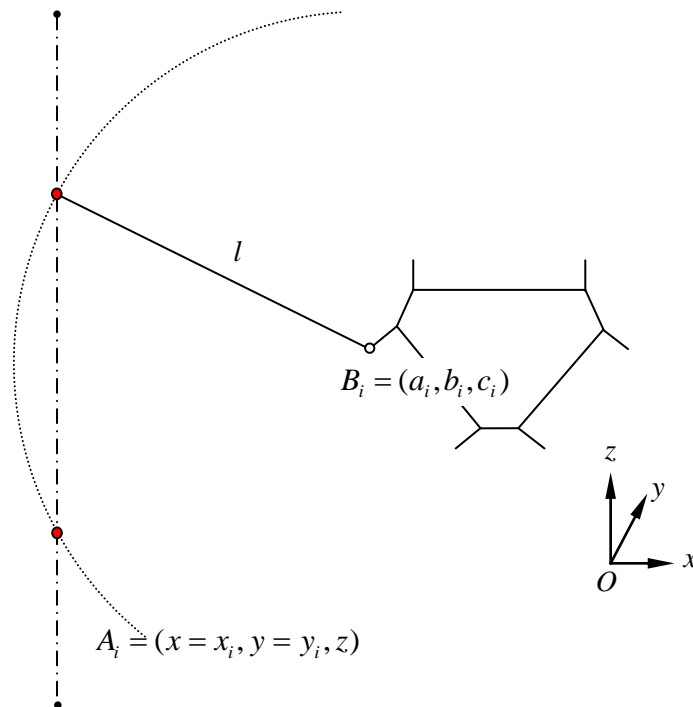


Fig. 4.1. Esquema de la cinemática inversa

Tomando como sistema de ecuaciones la ecuación de la esfera y la de la recta,

$$(x - a_i)^2 + (y - b_i)^2 + (z - c_i)^2 = l^2 \tag{4.1}$$

$$x = x_i; \quad y = y_i \tag{4.2}$$

Se llega a las dos posibles soluciones:

$$z = c_i \pm \sqrt{l^2 - (x_i - a_i)^2 - (y_i - b_i)^2} \tag{4.3}$$

De estas dos soluciones, la primera es la única viable, ya que la segunda solución implicaría que la plataforma estuviera por encima de las barras, caso al que nunca se llegará. Por tanto, la cinemática inversa de la plataforma Hexaglidle de estudio se resume en la siguiente ecuación:

$$z = c_i + \sqrt{l^2 - (x_i - a_i)^2 - (y_i - b_i)^2} \quad (4.4)$$

Una vez encontrada la ecuación se ha integrado en el programa *Matlab* para poder obtener, dada una transformación cualquiera de la plataforma, la posición de cada una de las barras en su respectiva guía [4]. La función que realiza el cálculo se encuentra en el archivo *InverseKinematics.m*.

## 4.2. Cinemática directa

El problema de la cinemática directa consiste en encontrar la posición final de la plataforma (actuador final) en función de las posiciones de cada una de las barras [5].

Mientras que la cinemática inversa de los robots paralelos es bastante intuitiva, la cinemática directa posee mayor complicación. Al contrario que en los robots serie, donde simplemente consiste en el producto de las transformadas que hay desde una referencia en la que se quiere expresar la posición del actuador final. Por ejemplo, las relaciones para una transformada de la referencia 0 a la 3 sería:

$${}^0T_3 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \quad (4.5)$$



**Fig. 4.2.** Distintas soluciones al problema cinemático directo

Se puede observar por tanto, que la solución de la cinemática directa para un robot serie posee solución única, mientras que en el caso del robot paralelo, ésta tiene múltiples soluciones, ya que consiste en la solución de un sistema de ecuaciones no lineal.

En el caso de una plataforma genérica la cinemática directa puede llegar a tener 40 soluciones distintas.

Para encontrar la solución al problema más cercana a una *pose* dada, se ha resuelto la cinemática directa de la plataforma realizando una aproximación numérica del tipo Newton-Raphson.

La variable de entrada es el vector  $z_b$ . Éste contiene la altura a la que están situados cada uno de los actuadores de las barras,

$$z_b = [z_{b1}, z_{b2}, z_{b3}, z_{b4}, z_{b5}, z_{b6}] \quad (4.6)$$

Durante el cálculo de la solución es necesario utilizar la matriz Jacobiana Analítica[6]. Dicha matriz relaciona la variación de la *pose* de la plataforma (posición y orientación) con el vector de velocidades de los actuadores  $\dot{z}_b$  como se muestra a continuación:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{pmatrix} = J_a \begin{pmatrix} \dot{z}_{b1} \\ \dot{z}_{b2} \\ \dot{z}_{b3} \\ \dot{z}_{b4} \\ \dot{z}_{b5} \\ \dot{z}_{b6} \end{pmatrix} \quad (4.7)$$

Si escribimos la ecuación de cada componente del vector, tenemos:

$$\begin{aligned} \dot{x} &= \sum_{i=1}^6 \frac{\partial f_x}{\partial z_{bi}} \dot{z}_{bi} & \dot{y} &= \sum_{i=1}^6 \frac{\partial f_y}{\partial z_{bi}} \dot{z}_{bi} & \dot{z} &= \sum_{i=1}^6 \frac{\partial f_z}{\partial z_{bi}} \dot{z}_{bi} \\ \dot{\theta}_x &= \sum_{i=1}^6 \frac{\partial f_{\theta_x}}{\partial z_{bi}} \dot{z}_{bi} & \dot{\theta}_y &= \sum_{i=1}^6 \frac{\partial f_{\theta_y}}{\partial z_{bi}} \dot{z}_{bi} & \dot{\theta}_z &= \sum_{i=1}^6 \frac{\partial f_{\theta_z}}{\partial z_{bi}} \dot{z}_{bi} \end{aligned} \quad (4.8)$$

Para la resolución del cálculo, se aproximan por diferencias finitas las derivadas de la ecuación anterior, donde  $\Delta f_j = z_{bi} - z_{bi+1}$  es la variación de  $z_b$  debida a una transformación de una variación de la posición mínima respecto a la coordenada  $j$ . Por ejemplo: para  $\Delta f_x$  dicha transformación diferencial sería únicamente una translación en  $\delta_x$  definida por la siguiente matriz de coordenadas homogéneas

$$\begin{bmatrix} 0_{1 \times 3} & (\delta_x, 0, 0)^T \\ 0_{1 \times 3} & 1 \end{bmatrix}.$$

Para iniciar el proceso iterativo, se debe tomar una *pose* inicial para la plataforma móvil. En el caso presente se ha tomado como *pose* inicial la de referencia.

Se repiten los cálculos que se describen a continuación hasta que el error se sitúa por debajo de un cierto umbral:

1. Cálculo del vector  $z_{ini}$  para la transformación inicial mediante la cinemática inversa (de la cual tenemos solución única)

2. Cálculo del error cometido en la aproximación:  $error = \|z_{ini} - z_{final}\|$
3. Cálculo del Jacobiano Analítico.
4. Cálculo del incremento de la *pose* de la plataforma móvil. Se escala el incremento obtenido por un factor  $escala = 1.5$  para reducir la velocidad de convergencia ya que si el punto de partida está muy lejos de la solución, al corregir el valor inicial se podría dar un salto en la posición de la plataforma que la sacara de su espacio de trabajo, llevándola a un espacio donde la cinemática inversa tiene componente imaginario. En la siguiente ecuación (4.9) se observa el cálculo de los incrementos de la *pose* de la plataforma:

$$\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta \theta_x \\ \Delta \theta_y \\ \Delta \theta_z \end{pmatrix} = \frac{J_a(z_{ini} - z_{final})^T}{escala} \quad (4.9)$$

5. Se recalcula el valor de  $z_{ini}$  aplicando a la plataforma la transformación de los incrementos encontrados.

En el siguiente gráfico se muestra la variación del error a lo largo de las distintas iteraciones.

Como se puede comprobar, dado un vector  $z_{final}$  cualquiera (en este caso en concreto se ha tomado  $z_{final} = [1410 \ 1540.9 \ 1510.3 \ 1019.1 \ 931 \ 1297.9]$ (mm)) se encuentra rápidamente el vector *pose* final de la plataforma móvil (el error alcanza un valor prácticamente nulo en quinta iteración):

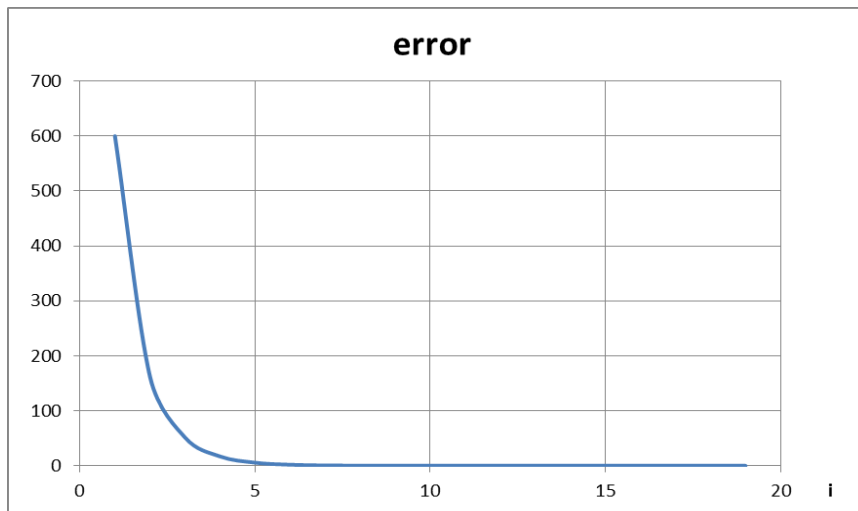


Fig. 4.2. Gráfico de la evolución del error a lo largo de las iteraciones

## 5. Jacobiano

El Jacobiano de un robot puede referirse bien

- Al “Jacobiano Cinemático”, o la matriz que relaciona las velocidades angulares y lineales de la plataforma móvil con las velocidades de los actuadores, o bien
- Al “Jacobiano Analítico” descrito en el apartado anterior, que es la matriz que relaciona las variaciones en los ángulos utilizados para representar la orientación de la plataforma móvil y su velocidad lineal, con las velocidades de los actuadores.

Así pues, el Jacobiano Cinemático no es una matriz Jacobiana en el sentido estricto, ya que no existe una representación de la orientación de cuyas variaciones obtengamos las velocidades angulares.

El Jacobiano Cinemático se utiliza para el control por velocidad del manipulador, mientras que el Jacobiano Analítico se utiliza habitualmente en métodos numéricos iterativos para resolver la cinemática directa.

### 5.1. Jacobiano cinemático inverso

A continuación se realiza el cálculo del Jacobiano Cinemático Inverso, cuyo fin es obtener la relación entre las velocidades generalizadas del robot ( $\dot{q}$ ) y las velocidades articulares ( $\dot{\Lambda}$ ).

El vector  $\dot{q}$  es un vector de seis componentes, las tres primeras son la velocidad lineal del origen del sistema de referencia de la plataforma y las otras tres son la velocidad angular de la plataforma:

$$\dot{q} = [\dot{x}_c, \dot{y}_c, \dot{z}_c, \omega_x, \omega_y, \omega_z]^T.$$

La ecuación que relaciona estas dos velocidades es la siguiente:

$$\dot{\Lambda} = J^{-1} \cdot \dot{q} \quad (5.1)$$



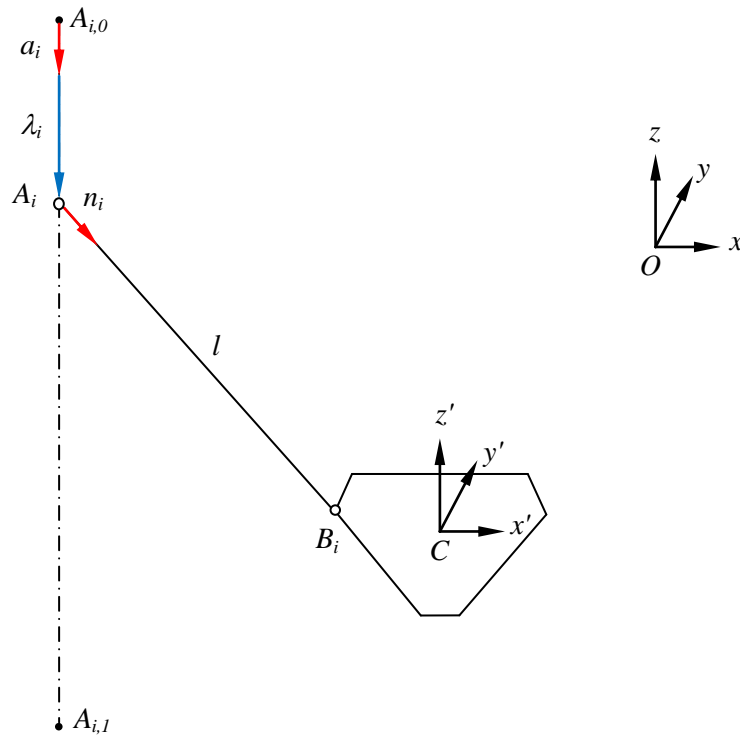


Fig. 5.1. Esquema con variables de la plataforma Hexaglide

Para poder realizar el cálculo del Jacobiano Cinemático Inverso, se ha partido de la cinemática inversa. Para ello, se han reescrito las ecuaciones partiendo del planteamiento realizado por I. A. Bonev en su tesis [7].

A partir de la notación de la figura, se deduce que:

$$A_{i,0}B_i = OC + CB_i - OA_{i,0} = OC + RCB_i' - OA_{i,0} \quad (5.2)$$

$$l \cdot n_i = A_{i,0}B_i - \lambda_i a_i \quad (5.3)$$

Elevando al cuadrado los dos extremos de la ecuación se obtiene:

$$l^2 = \|A_{i,0}B_i\|^2 + \lambda^2 - 2\lambda_i \cdot (a_i^T \cdot A_{i,0}B_i) \quad (5.4)$$

Se deriva la ecuación (5.4) respecto al tiempo:

$$\dot{\lambda}_i \cdot \lambda_i - \dot{\lambda}_i \cdot a_i^T \cdot A_{i,0}B_i - \dot{\lambda}_i \cdot a_i^T (V_C + \dot{R}CB_i') + A_{i,0}B_i^T (V_C + \dot{R}CB_i') = 0 \quad (5.5)$$

Sabiendo que  $CB_i' = R^T CB_i$  y recomponiendo la ecuación anterior:

$$\dot{\lambda}_i \cdot a_i^T \cdot (\lambda_i \cdot a_i - A_{i,0}B_i) - (\lambda_i \cdot a_i^T - A_{i,0}B_i^T)(V_C + \dot{R}R^T CB_i) = 0 \quad (5.6)$$

Utilizando la ecuación (5.3) y que  $\tilde{\omega} = \dot{R}R^T$ , donde  $\tilde{\omega}$  es la matriz antisimétrica de la velocidad

angular  $\tilde{\omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$ , se obtiene:

$$-(a_i^T \cdot A_i B_i) \dot{\lambda}_i + A_i B_i^T (V_C + \tilde{\omega} CB_i) = 0 \quad (5.7)$$

Utilizando la propiedad de la matriz antisimétrica ( $\tilde{\omega} CB_i = \omega \times CB_i$ ) y la anticonmutatividad del producto vectorial ( $\omega \times CB_i = B_i C \times \omega$ ), se reescribe la ecuación anterior:

$$-(a_i^T \cdot A_i B_i) \dot{\lambda}_i + A_i B_i^T (V_C + B_i C \times \omega) = 0 \quad (5.8)$$

La misma ecuación en forma vectorial:

$$-(a_i^T \cdot A_i B_i) \cdot \dot{\lambda}_i + [A_i B_i^T \mid (A_i B_i \times B_i C)^T] \dot{q} = 0 \quad (5.9)$$

Finalmente, dado que  $A_i B_i = l \cdot n_i$  y dividiendo ambos extremos de la ecuación por  $l$ :

$$-\begin{bmatrix} a_1^T n_1 & 0 & \cdots & 0 \\ 0 & a_2^T n_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_6^T n_6 \end{bmatrix} \dot{\Lambda} + \begin{bmatrix} n_1^T & (n_1 \times B_1 C)^T \\ n_2^T & (n_2 \times B_2 C)^T \\ \vdots & \vdots \\ n_6^T & (n_6 \times B_6 C)^T \end{bmatrix} \dot{q} = 0 \quad (5.10)$$

$$\Phi_\Lambda \dot{\Lambda} + \Phi_q \dot{q} = 0 \quad (5.11)$$

Al aislar el vector de velocidades articulares, se obtiene el Jacobiano Cinemático Inverso:

$$\dot{\Lambda} = -\Phi_{\Lambda}^{-1} \Phi_q \dot{q} = J^{-1} \dot{q} \quad (5.12)$$

$$J^{-1} = \begin{bmatrix} (a_1^T n_1) n_1^T, & (a_1^T n_1) (n_1 \times B_1 C)^T \\ (a_2^T n_2) n_2^T, & (a_2^T n_2) (n_2 \times B_2 C)^T \\ \vdots & \vdots \\ (a_6^T n_6) n_6^T, & (a_6^T n_6) (n_6 \times B_6 C)^T \end{bmatrix}^{-1} \quad (5.13)$$

Dicha matriz se ha implementado en la función **JacobianMatrix.m** generada con *Matlab*®. En la siguiente figura se observa el Jacobiano Cinemático Inverso, obtenido una vez el usuario ha introducido la posición y orientación de la plataforma deseadas:

```
Enter the platform displacement in mm as follows [dispx, dispy, dispz]: [200, 100, 400]
Enter the platform x rotation in radians: pi/6
Enter the platform y rotation in radians: 0
Enter the platform z rotation in radians: 0

Jac =

    -0.1107    0.9838   -1.0000    90.5164   248.3758   234.3241
    -0.0911   -0.5784   -1.0000   -38.4510   241.6493  -136.2622
    -0.1983   -0.5418   -1.0000  -145.9492  -88.9346    77.1289
     1.6683    0.2547   -1.0000  -177.3944  -31.1172  -303.8644
     1.6312    0.1345   -1.0000   166.6647  -325.7180   228.0381
    -0.2377    0.9258   -1.0000   325.9025  -35.4751  -110.2951
```

Fig. 5.2. Respuesta de la función **JacobianMatrix.m** para un caso concreto

## 6. Espacio de trabajo

En el diseño de los robots paralelos se da especial importancia al espacio de trabajo, ya que este es menor en relación al espacio de trabajo de los robots serie y además, queda restringido por su propia estructura.

Para obtener el espacio de trabajo de la plataforma Hexaglide se ha desarrollado la función **Workspace.m** en la cual se calcula el espacio de trabajo para una orientación constante determinada. Se define dicho espacio de trabajo como una región tridimensional que puede ser alcanzada por la plataforma dada una constante orientación de ésta.

Para obtener todas las posibles translaciones del robot se ha recorrido mediante un proceso iterativo en las tres direcciones una región prismática del espacio que contiene la estructura del robot.

Para cada translación obtenida se ha resuelto el problema cinemático inverso, comprobando que la posición planteada realmente puede ser alcanzada por la plataforma de estudio.

Las condiciones con las que se ha comprobado la viabilidad de cada posición se describen a continuación:

1. Que todos los puntos de la plataforma estén contenidos en el interior de la estructura base.
2. Que la posición alcanzada por los distintos actuadores no supere la altura de las guías permitida:  $z_i \leq h_g + 300$ .
3. Que todos los puntos de la plataforma, para la transformación aplicada, siempre estén por encima del suelo.

A continuación se muestran dos imágenes del espacio de trabajo dadas dos orientaciones distintas de la plataforma:

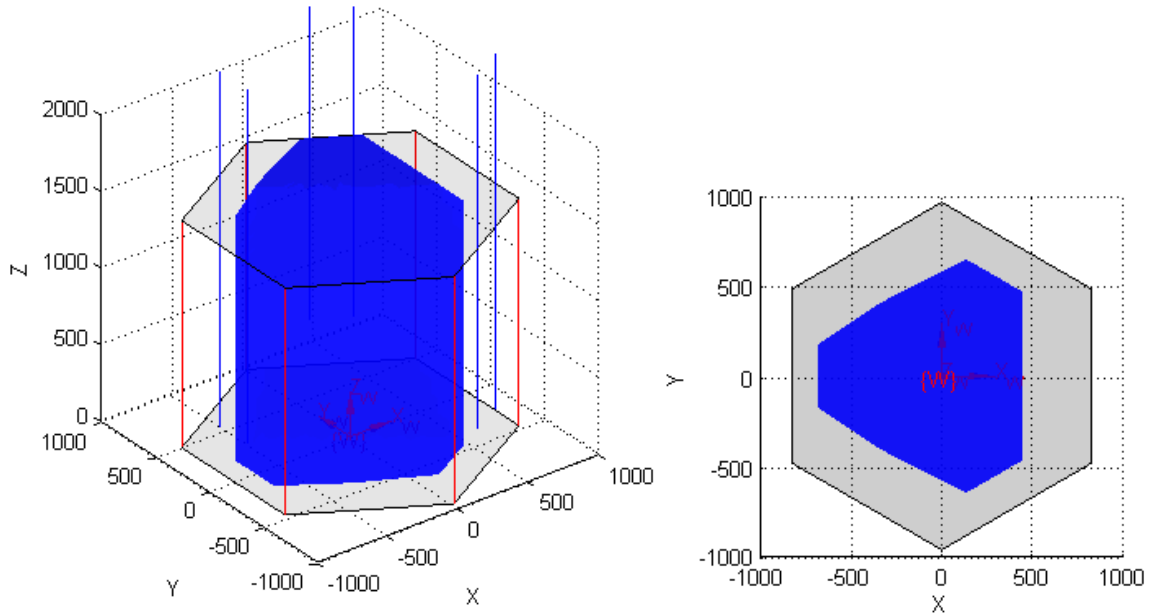


Fig. 6.1. Representación del espacio de trabajo para una orientación  $(0,0,0)$  (rad).

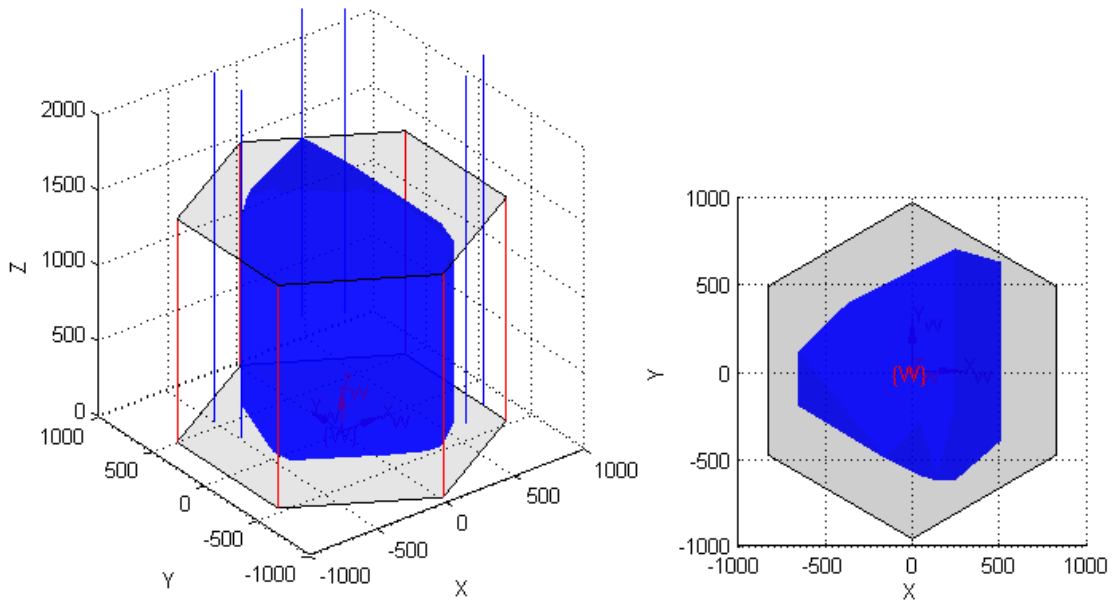


Fig. 6.2. Representación del espacio de trabajo para una orientación  $\left(\frac{\pi}{6}, \frac{\pi}{6}, 0\right)$  (rad).

## 7. Singularidades

Se dice que un robot paralelo está en configuración singular cuando cualquiera de las dos matrices ( $\Phi_\Lambda$  o  $\Phi_q$ ) es singular. Dichas matrices proporcionan una relación entre las velocidades de entrada y las de salida que queda descrita por la ecuación de velocidad (5.4), donde:

$$\Phi_\Lambda = \frac{\partial \Phi}{\partial \Lambda}, \quad \Phi_q = \frac{\partial \Phi}{\partial q} \quad (7.1)$$

$\Phi_\Lambda$  y  $\Phi_q$  son Jacobianos Cinemáticos, ambos dependientes de  $\dot{\Lambda}$  (velocidades de los actuadores) y  $\dot{q}$  (velocidad de la plataforma). Si una de dichas matrices es singular, esto deriva en una deficiencia de rango y, por tanto, se pierden grados de libertad. Se pueden distinguir dos tipos principales de singularidades: singularidades inversas y singularidades directas.

### 7.1. Singularidades inversas

El primer caso de singularidades aparece al resolver el problema cinemático inverso. La singularidad inversa aparece cuando la matriz  $\Phi_\Lambda$  es singular.

Si dadas las velocidades de la plataforma ( $\dot{q}$ ), se quieren obtener las velocidades de los actuadores ( $\dot{\Lambda}$ ), se puede reescribir la ecuación de velocidades (5.4) de la siguiente forma:

$$\dot{\Lambda} = -\Phi_\Lambda^{-1} \Phi_q \dot{q} \quad (7.2)$$

En el caso de que la matriz  $\Phi_\Lambda$  sea no singular, no hay ningún problema para obtener las velocidades de los actuadores. No obstante, si la matriz  $\Phi_\Lambda$  es singular, la ecuación (7.2) no se satisface ya que una matriz singular no es invertible y, por tanto, no se pueden aislar las incógnitas ( $\dot{\Lambda}$ ). Queda por tanto el problema irresoluble o indeterminado, como se demuestra a continuación.

Se reescribe la ecuación (5.4) de la siguiente forma:

$$\Phi_\Lambda \dot{\Lambda} = -\Phi_q \dot{q} \quad (7.3)$$

Dado que, en el caso presente, se parte de la hipótesis de que el vector ( $\dot{q}$ ) es conocido, se puede reescribir la ecuación anterior tomando el bloque de la derecha como un vector ( $a$ ) cuyos componentes son constantes y conocidos:

$$\Phi_{\Lambda} \dot{\Lambda} = a \quad (7.4)$$

Dado que la matriz  $\Phi_{\Lambda}$  es singular, su imagen posee dimensión inferior a seis. Por tanto, el vector  $a$ , para que exista solución debe estar contenido sobre dicha imagen y, en este caso, se darían infinitas soluciones.

Por tanto, la solución del sistema queda reducida a un subespacio de  $a$  y no a su totalidad. En definitiva, el sistema es compatible indeterminado ya que para un valor de  $a$  existen infinitas soluciones, es decir, infinitas posibles velocidades de los actuadores ( $\dot{\Lambda}$ ).

Partiendo del razonamiento anterior, se observa que, para una velocidad nula de la plataforma ( $\dot{q} = 0$ ), existen distintas soluciones del sistema en las que  $\dot{\Lambda}$  es distinto de cero. Por tanto, habrá velocidades de los actuadores para los que la plataforma se mantendrá inmóvil. Esto quiere decir que se da una pérdida de destreza en el sistema, ya que se pierden grados de libertad. Si se reescribe la ecuación (5.4) aislando las velocidades de la plataforma,

$$\dot{q} = -\Phi_q^{-1} \Phi_{\Lambda} \dot{\Lambda} = M \cdot \dot{\Lambda} \quad (7.5)$$

Dado que  $\Phi_{\Lambda}$  es singular, la matriz  $M$  también es singular, como se demuestra en la siguiente ecuación:

$$\det(M) = \det(-\Phi_q^{-1} \Phi_{\Lambda}) \det(-\Phi_q^{-1}) \det(\Phi_{\Lambda}) = \det(-\Phi_q^{-1}) \cdot 0 = 0 \quad (7.6)$$

Si la matriz  $M$  también es singular, se observa que no es posible alcanzar todas las velocidades de la plataforma ( $\dot{q}$ ) sino solo un subespacio.

### 7.1.1. Límites articulares

Aparte de la singularidad comentada, los límites articulares habitualmente también pueden verse como una singularidad inversa.

Un actuador no puede llegar a su límite articular con velocidad distinta de cero ya que se empotraría contra el mismo, por tanto, una de las velocidades articulares debe ser  $\dot{\lambda}_i = 0$ .

Si se parte de la ecuación anterior (8.6), se puede observar que dado que uno de los elementos del vector  $\dot{\Lambda}$  es igual a cero, el espacio de velocidades posibles de la plataforma ( $\dot{q}$ ) tendrá como máximo dimensión cinco. Es decir, como en el caso anterior, se produce una pérdida de destreza y, por tanto, también se pueden considerar los límites articulares como singularidades inversas.

### 7.1.2. Aplicación a la plataforma de estudio

Para el caso de estudio, conocemos que:

$$\Phi_{\Lambda} = - \begin{bmatrix} a_1^T n_1 & 0 & \cdots & 0 \\ 0 & a_2^T n_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_6^T n_6 \end{bmatrix} \quad (7.7)$$

Para que dicha matriz sea singular (determinante nulo), la única opción posible es que una de las barras sea perpendicular a su respectiva guía (el producto escalar  $a_i^T n_i$  sería nulo).

Dado que dicho caso no se puede dar en el robot de estudio por limitaciones en su espacio de trabajo, se puede asegurar que no se dará ninguna singularidad inversa.

## 7.2. Singularidades directas

El segundo caso de singularidades aparece en la resolución del problema cinemático directo. La singularidad directa aparece cuando la matriz  $\Phi_q$  es singular.

Si, dadas las velocidades de los actuadores ( $\dot{\Lambda}$ ), se quieren obtener las velocidades la plataforma ( $\dot{q}$ ), se puede reescribir la ecuación de velocidades (7.4) de la siguiente forma:

$$\dot{q} = -\Phi_q^{-1} \Phi_{\Lambda} \dot{\Lambda} \quad (7.8)$$

En el caso de que la matriz  $\Phi_q$  sea no singular, como en el primer caso, no hay ningún problema para



obtener las velocidades de la plataforma. No obstante, si la matriz  $\Phi_q$  es singular, la ecuación (7.8) no se cumple ya que una matriz singular no es invertible y, por tanto, no se pueden aislar las incógnitas ( $\dot{q}$ ). Queda por tanto el problema irresoluble o indeterminado, como se ha demostrado en el primer caso.

### 7.2.1. Aplicación a la plataforma de estudio

Para el caso de estudio, conocemos que:

$$\Phi_q = \begin{bmatrix} n_1^T & (n_1 \times B_1 C)^T \\ n_2^T & (n_2 \times B_2 C)^T \\ \vdots & \vdots \\ n_6^T & (n_6 \times B_6 C)^T \end{bmatrix} \quad (7.9)$$

La plataforma, caerá en una configuración singular cuando cualquiera de las direcciones de las barras de la plataforma pase a ser linealmente dependiente con las otras.

Se dice que una singularidad es de *tipo n* si la dirección de la barra es dependiente de la de *n* o más barras[8]. En las siguientes figuras se pueden ver los distintos tipos de singularidades directas:

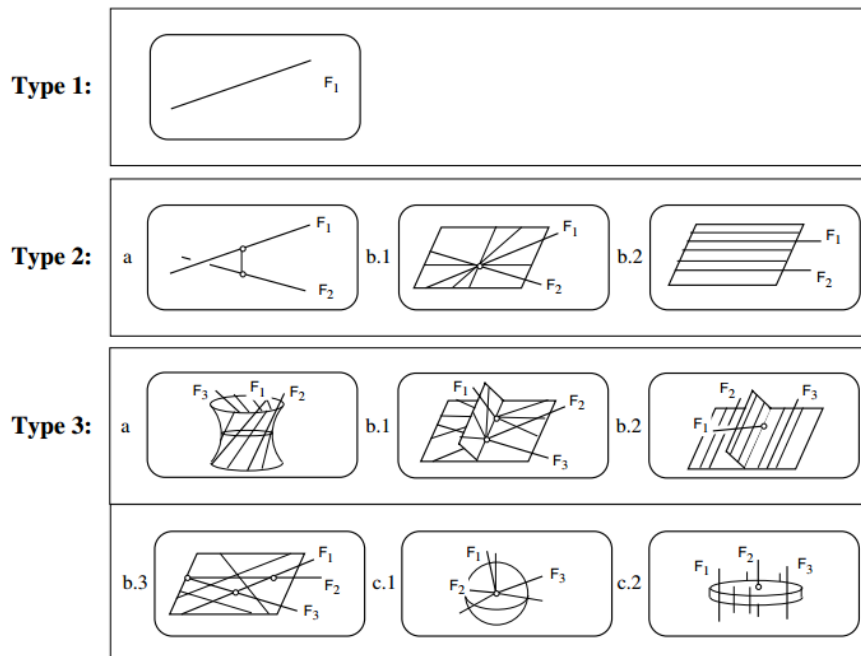


Fig. 7.1. Tipos de singularidades directas (1)

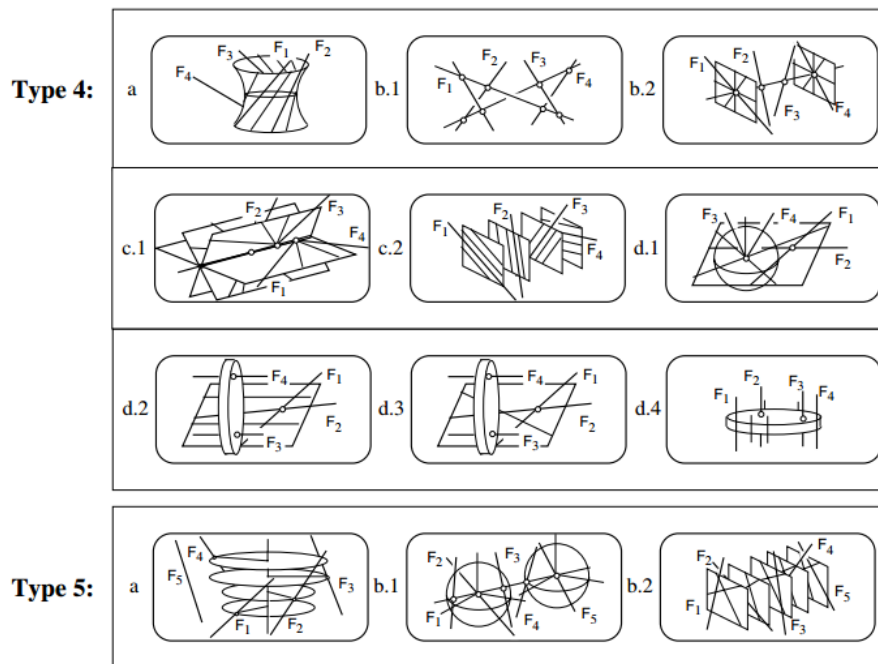


Fig. 7.2. Tipos de singularidades directas (2)

Si la orientación de la plataforma se limita a  $\pm \frac{\pi}{6} rad$  en torno a un eje arbitrario que pase por el origen, se ha observado que el robot no alcanza ninguna singularidad.

## 8. Dinámica

Para el estudio de la dinámica se ha partido de las ecuaciones Newton – Euler, que describen el comportamiento de un sólido rígido sujeto a la acción de fuerzas exteriormente aplicadas [9]. Estas ecuaciones resultan de la aplicación del teorema de la cantidad de movimiento y del teorema del momento cinético al sólido, y se pueden escribir así:

$$F + mg = m\gamma \quad (8.1)$$

$$M = I\dot{\omega} + \omega \times I\omega \quad (8.2)$$

Donde:

$F$  : Resultante de las fuerzas externas que actúan sobre el sólido

$m$  : Masa del sólido

$\gamma$  : Aceleración total del centro de masas C del sólido

$g$  : Aceleración de la gravedad

$M$  : Momento resultante con respecto a C de las fuerzas externas que actúan sobre el sólido

$I$  : Tensor de inercia del sólido referido a C

$\dot{\omega}$  : Aceleración angular del sólido

$\omega$  : Velocidad angular del sólido

La obtención de un modelo dinámico detallado del robot Hexaglide requiere un proceso minucioso de descomposición del robot en sus distintos sólidos rígidos, identificando las fuerzas de interacción en todos los enlaces, y aplicando las anteriores ecuaciones de Newton-Euler a cada uno de los sólidos individualmente. Ello da lugar a un sistema de ecuaciones diferenciales ordinarias con muchas ecuaciones y variables que resulta costoso de manipular. Para evitarlo, en nuestro caso introduciremos las siguientes simplificaciones, que resultan plausibles en una primera aproximación al problema:

- La masa de cada barra, debido a su fabricación en fibra de carbono, puede despreciarse frente a la masa de la plataforma móvil.
- Los cursores deslizantes en las articulaciones actuadas son también de masa despreciable frente a la de la plataforma móvil, al ser pequeños y fabricados en aluminio.

- Las fuerzas de fricción en las articulaciones se desprecian, al considerarse éstas tienen un buen acabado superficial y están bien lubricadas.

Por otro lado, se realiza el cálculo del tensor de inercia de la plataforma considerándola como una placa triangular plana, dado que el grosor de la plataforma es despreciable respecto al lado del triángulo.

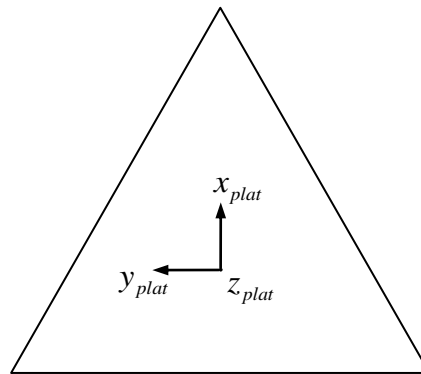


Fig. 8.1. Sistema de referencia de la plataforma

Con esta simplificación, el tensor de inercia de la plataforma referido a su centro de masas y escrito en la base vectorial indicada en la figura (8.1) es:

$$I_{plat} = \begin{bmatrix} I_{plat\_x'} & 0 & 0 \\ 0 & I_{plat\_y'} & 0 \\ 0 & 0 & I_{plat\_z'} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{96} b^4 m_{plat} & 0 & 0 \\ 0 & \frac{\sqrt{3}}{96} b^4 m_{plat} & 0 \\ 0 & 0 & \frac{\sqrt{3}}{48} b^4 m_{plat} \end{bmatrix} \quad (8.3)$$

Tanto para la dinámica directa como para la inversa, se ha partido del estudio de la plataforma como un sólido rígido en el espacio, sobre el que actúan la fuerza de la gravedad y una fuerza y momento resultantes (estos son el sumatorio de las fuerzas y momentos que realiza cada barra sobre la plataforma).

### 8.1. El problema dinámico directo

La dinámica directa consiste en, dadas las fuerzas y momentos que actúan sobre el robot en función

del tiempo, calcular el movimiento resultante de éste.

Dado que se han considerado las barras con masa despreciable, y al estar estas barras unidas a la base y la plataforma mediante una junta cardan y una rótula esférica, respectivamente, la fuerza que cada una de las barras ejerce sobre la plataforma tiene en la misma dirección que la barra. Por tanto, la ecuación que relaciona la fuerza  $\tau_i$  en el  $i$ -ésimo actuador con la fuerza  $f_{s_i}$  aplicada en el punto de unión de la barra  $i$  con la plataforma, como se observa en la figura (8.2), es la siguiente:

$$f_{s_i} = \tau_i \cdot (a_i^T \cdot n_i) \cdot n_i \tag{8.4}$$

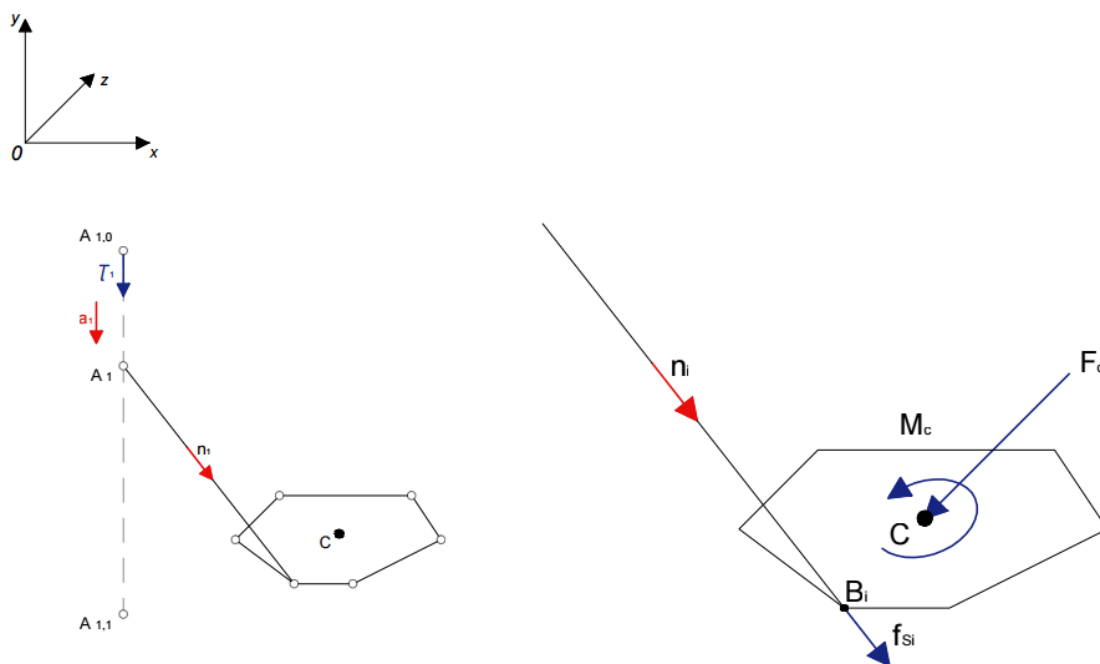


Fig. 8.2. Esquema de fuerzas y parámetros para la resolución de la dinámica

Aplicando las condiciones de equilibrio, a partir de la ecuación anterior se puede calcular el tursor de fuerzas resultante ejercido por las seis barras sobre la plataforma, reducido a su centro de masas C. Si  $F_C$  y  $M_C$  denotan los vectores de fuerza y de momento de este tursor, se obtiene:

$$F_C = \sum_{i=1}^6 f_{s_i} = \sum_{i=1}^6 \tau_i \cdot (a_i^T \cdot n_i) \cdot n_i \tag{8.5}$$

$$M_C = \sum_{i=1}^6 m_{s_i} = \sum_{i=1}^6 \tau_i \cdot (a_i^T \cdot n_i) \cdot (n_i \times B_i C) \quad (8.6)$$

Ahora, podemos particularizar las ecuaciones (8.1) i (8.2) para la plataforma del robot, considerándola como un sólido rígido sobre el que actúa este torsor, sujeto a una velocidad angular  $\omega$ , y a una aceleración lineal  $\gamma_C$  en su centro de masas. Aislando  $\gamma_C$  y  $\dot{\omega}$  en el miembro izquierdo, estas ecuaciones quedan así

$$\gamma_C = 1/m_{plat} (F_C + m_{plat} g) \quad (8.7)$$

$$\dot{\omega} = I_{plat}^{-1} (M_C + (I_{plat} \omega) \times \omega) \quad (8.8)$$

Donde  $m_{plat}$  y  $I_{plat}$  son la masa de la plataforma y su tensor de inercia referido al centro de masas.

A continuación, se divide el estudio del problema dinámico directo en dos partes: por un lado translación de la plataforma y, por otro, orientación de la plataforma.

Los desarrollos correspondientes se han implementado en las funciones **Video\_Forward\_Dynamics.m** y **Video\_Path\_ForwardDynamics.m** que, entrados los parámetros conocidos y la fuerza en los actuadores, reproducen el movimiento de la plataforma resultante.

### 8.1.1. Translación de la plataforma

Dadas las fuerzas de los actuadores, para obtener la aceleración, velocidad y posición del centro de masas de la plataforma, es necesario integrar la ecuación (8.7). Para ello, se ha utilizado el método de Runge-Kutta[10], [11]. Este método es muy utilizado por su rapidez y exactitud para resolver numéricamente problemas de valor inicial. Entre los algoritmos de RK destaca por su mayor precisión el de cuarto orden "RK4", que es el que se usará en el presente trabajo para realizar el cálculo de la dinámica.

Para cada una de las barras, se considera un perfil de fuerzas de los actuadores dependiente del tiempo  $\tau_i = f(t)$ . Mediante la ecuación (8.5) se calcula la fuerza resultante reducida al centro de masas de la plataforma:

$$F_C(t) = \sum_{i=1}^6 f_{s_i}(t) \quad (8.9)$$

Una vez se tiene la fuerza externa resultante sobre el centro de masas expresada en función del tiempo se procede a solucionar el problema dinámico directo para la translación de la plataforma.

Teniendo en cuenta que la ecuación (8.7) únicamente depende del tiempo  $\gamma = f_1(t)$ , se aplica el método de Runge-Kutta para la integración de la aceleración lineal del centro de masas de la plataforma. En este caso, de hecho, el método puede ver como equivalente al método de integración mediante la regla de Simpson. En cada iteración, el método calcula la velocidad en el instante de tiempo  $i+1$  utilizando esta expresión

$$v_{i+1} = v_i + \frac{1}{6}(k_{1v} + 2k_{2v} + 2k_{3v} + k_{4v}) \quad (8.10)$$

donde:

$$k_{1v} = dt \cdot f_1(t_i) \quad (8.11)$$

$$k_{2v} = dt \cdot f_1\left(t_i + \frac{dt}{2}\right) \quad (8.12)$$

$$k_{3v} = k_{2v} \quad (8.13)$$

$$k_{4v} = dt \cdot f_1(t_i + dt) \quad (8.14)$$

Como condición inicial para la integración de la aceleración, se ha considerado que la plataforma parte del reposo, y que por tanto la velocidad inicial del centro de masas es nula:  $v(t=0) = v_0 = (0,0,0)mm/s$ .

Sabiendo que la velocidad también es una función dependiente únicamente del tiempo  $v = f_2(t)$  se realiza una segunda integración mediante RK4 para obtener la posición de la plataforma. Igual que en el caso anterior, se toma una posición inicial del centro de masas para poder empezar el proceso iterativo  $p(t=0) = p_0 = (0,0,400)mm$ . El cálculo iterativo viene dado por

$$p_{i+1} = p_i + \frac{1}{6}(k_{1p} + 2k_{2p} + 2k_{3p} + k_{4p}) \quad (8.15)$$

donde  $k_{1p}$ ,  $k_{2p}$ ,  $k_{3p}$  y  $k_{4p}$  tienen la misma forma que en la integración de la aceleración pero

variando la función a integrar, que ahora será  $f_2(t)$ .

La función  $v = f_2(t)$  -contrariamente al caso anterior- no se conoce, ya que después de realizar la primera integración se obtienen las velocidades para cada instante de tiempo, pero no la expresión que define la velocidad en función del tiempo.

Por tanto, para poder hallar  $k_{2p} = k_{3p} = dt \cdot f_2\left(t_i + \frac{dt}{2}\right)$ , para las que el valor de la velocidad en el instante indicado  $\left(t_i + \frac{dt}{2}\right)$  es desconocido, se realiza una interpolación lineal entre  $k_{1p} = dt \cdot f_2(t_i)$  y  $k_{4p} = dt \cdot f_2(t_i + dt)$ , cuyos valores sí son conocidos.

$$k_{2p} = k_{3p} = k_{1p} + (k_{4p} - k_{1p}) \frac{t_i + \frac{dt}{2} - t_i}{t_i + dt - t_i} \quad (8.16)$$

$$k_{2p} = k_{3p} = k_{1p} + (k_{4p} - k_{1p}) 0.5$$

Una vez realizado este último cálculo, se llega al objetivo de este apartado: conocer la aceleración, velocidad y posición del centro de masas de la plataforma dadas las fuerzas en los actuadores (véase el fichero **Translation.m**).

### 8.1.2. Orientación de la plataforma

Para el cálculo de la orientación de la plataforma se procede de un modo similar al explicado en el anterior subapartado.

Dada la ecuación de la aceleración angular (8.8)  $\dot{\omega}(t) = f(\omega(t), t)$  expresada en el sistema de referencia de la plataforma y, conociendo la velocidad angular inicial del centro de masas  $\omega(t_o) = \omega_0$ , se aplica el método Runge-Kutta para obtener la velocidad angular en dicho sistema de referencia:

$$\omega_{i+1}^{plat} = \omega_i^{plat} + \frac{1}{6} (k_{1\omega} + 2k_{2\omega} + 2k_{3\omega} + k_{4\omega}) \quad (8.17)$$

Donde  $dt$  es el intervalo de tiempo utilizado, y:



$$k_{1\omega} = dt \cdot f(\omega_i, t_i) \quad (8.18)$$

$$k_{2\omega} = dt \cdot f\left(\omega_i + \frac{k_{1\omega}}{2}, t_i + \frac{dt}{2}\right) \quad (8.19)$$

$$k_{3\omega} = dt \cdot f\left(\omega_i + \frac{k_{2\omega}}{2}, t_i + \frac{dt}{2}\right) \quad (8.20)$$

$$k_{4\omega} = dt \cdot f(\omega_i + k_{3\omega}, t_i + dt) \quad (8.21)$$

En segundo lugar, una vez hallada la velocidad angular de la plataforma, se realiza una segunda integración para encontrar el valor de la orientación. Se realiza dicho cálculo haciendo uso de cuaterniones para representar la orientación de la plataforma [12].

Se realiza dicho cálculo haciendo uso de cuaterniones para representar la orientación de la plataforma.

Los cuaterniones, desarrollados por el físico W. R. Hamilton, son una extensión de los números complejos y se expresan como un escalar más un vector:

$$Q = s + v = s + v_1 i + v_2 j + v_3 k \quad (8.22)$$

$$Q = s \langle v_1, v_2, v_3 \rangle \quad (8.23)$$

Una de las utilidades de los cuaterniones unitarios ( $s^2 + v_1^2 + v_2^2 + v_3^2 = 1$ ) es que pueden servir para representar rotaciones. Esto es así ya que cualquier rotación en el espacio puede ser expresada como la rotación de un ángulo  $\theta$  alrededor de un vector unitario  $\hat{n}$ . Estas dos componentes que describen una rotación se relacionan directamente con las componentes del cuaternión mediante las siguientes fórmulas:

$$s = \cos \frac{\theta}{2}, \quad v = \left( \sin \frac{\theta}{2} \right) \hat{n} \quad (8.24)$$

La relación entre el cuaternión y la matriz de rotación queda definida por la siguiente ecuación:

$$R_q = \begin{bmatrix} 1 - 2v_2v_2 - 2v_3v_3 & 2v_1v_2 - 2sv_3 & 2v_1v_3 + 2sv_2 \\ 2v_1v_2 + 2sv_3 & 1 - 2v_1v_1 - 2v_3v_3 & 2v_2v_3 - 2sv_1 \\ 2v_1v_3 - 2sv_2 & 2v_2v_3 + 2sv_1 & 1 - 2v_1v_1 - 2v_2v_2 \end{bmatrix} \quad (8.25)$$

Un cuerpo que gira en un espacio tridimensional, posee cierta velocidad angular  $\omega = (\omega_1, \omega_2, \omega_3)$ . Dicho vector define la dirección del eje instantáneo de rotación (eje alrededor del cual gira el cuerpo en un instante de tiempo determinado). Por lo general, dicho eje varía con el tiempo. Existe una relación entre la derivada de la matriz de rotación de un cuerpo y la velocidad angular del mismo:

$$\dot{R}(t) = S(\omega) \cdot R(t) \quad (8.26)$$

Donde  $S(\omega)$  es la matriz antisimétrica de la velocidad angular. En el caso de operar con cuaterniones, dicha ecuación se reformula de la siguiente forma:

$$\dot{Q}(t) = [0 \quad \omega] \cdot Q(t) \quad (8.27)$$

Donde  $[0 \quad \omega]$  es interpretado como un cuaternión en el que su componente escalar es nula y la parte vectorial queda definida por la velocidad angular ( $s = 0$  y  $v = \omega$ ).

Para integrar la ecuación (8.27)  $\dot{Q}(t) = f(\omega(t), Q(t))$  se toma la velocidad angular  $\omega$  en el sistema de referencia fijo ( $\omega = R_q \cdot \omega^{plat}$ ).

Finalmente, se aplica el método de Runge-Kutta 4 general para el caso de la Eq. (8.27) utilizando la siguiente expresión:

$$Q_{i+1} = Q_i + \frac{1}{6} (k_{1q} + 2k_{2q} + 2k_{3q} + k_{4q}) \quad (8.28)$$

Donde  $dt$  es el intervalo de tiempo utilizado, y:

$$k_{1q} = dt \cdot f(\omega_i, Q_i) \quad (8.29)$$

$$k_{2q} = dt \cdot f\left(\omega_i + \frac{k_{1\omega}}{2}, Q_i + \frac{k_{1q}}{2}\right) \quad (8.30)$$

$$k_{3q} = dt \cdot f\left(\omega_i + \frac{k_{2\omega}}{2}, Q_i + \frac{k_{2q}}{2}\right) \quad (8.31)$$

$$k_{4q} = dt \cdot f(\omega_i + k_{3\omega}, Q_i + k_{3q}) \quad (8.32)$$

Después de cada iteración, el cuaternión puede gradualmente dejar de ser unitario debido al error

numérico. Es importante por tanto, renormalizar el cuaternión de forma que apunte a la misma dirección pero teniendo magnitud unitaria:

$$Q_{i+1} = \frac{Q_{i+1}}{\|Q_{i+1}\|} \quad (8.33)$$

Para el programa con *Matlab*<sup>®</sup> se ha generado la siguiente función **Orientation.m**.

## 8.2. El problema dinámico inverso

El problema dinámico inverso consiste en, dado un movimiento conocido para la plataforma, realizar el cálculo de las fuerzas y momentos necesarios en los actuadores.

Partiendo de la simplificación de considerar la plataforma como un sólido rígido en el espacio sobre el que actúa un torsor de fuerza  $F_C$  y momento  $M_C$ , y dado que se conoce el movimiento de la plataforma, se reescriben las ecuaciones (9.1) y (9.2) de la siguiente forma:

$$F_C = m_{plat}(\gamma_C - g) \quad (8.34)$$

$$M_C = I_{plat}\dot{\omega} + \omega \times I_{plat}\omega \quad (8.35)$$

Dado que se conoce el movimiento de la plataforma,  $\gamma_C$ ,  $\omega$ , y  $\dot{\omega}$  son conocidas a lo largo del tiempo, y la obtención de  $F_C$  y  $M_C$  es directa mediante (8.34) y (8.35). Una vez encontrados  $F_C$  y  $M_C$  se calculan las fuerzas de los actuadores como se muestra a continuación.

Si se aplican las condiciones de equilibrio:

$$F_C = \sum_{i=1}^6 f_{s_i} = \sum_{i=1}^6 \tau_i (a_i^T n_i) n_i \quad (8.36)$$

$$M_C = \sum_{i=1}^6 m_{s_i} = \sum_{i=1}^6 \tau_i (a_i^T n_i) (n_i \times B_i C) \quad (8.37)$$

Reescribiendo las ecuaciones anteriores en forma matricial y substituyendo por el Jacobiano Cinemático Inverso calculado anteriormente en la ecuación (5.13), se obtiene:

$$\sigma_C = J^{-T} \tau \quad (8.38)$$

donde  $\sigma_C = \begin{bmatrix} F_C \\ M_C \end{bmatrix}$  [13]. Por tanto, de la ecuación anterior se obtiene el valor deseado para las fuerzas

ejercidas por los actuadores:

$$\tau = J^T \sigma_C \quad (8.39)$$

Se ha creado una función en el programa *Matlab*<sup>®</sup> que dado un fichero con una trayectoria de *poses* de la plataforma en función del tiempo, calcula la fuerza necesaria en los actuadores para llevarla a cabo (véase el fichero *InverseDynamics.m*).

## 9. Impacto económico y ambiental

Los costes de realizar este proyecto han sido de dos tipos: costes materiales i costes de consumo energético y de conexión a internet.

Los costes debido a material que se derivan de la realización del proyecto son del ordenador con el que se ha trabajado y la licencia del programa *Matlab*<sup>®</sup>. Como el ordenador tiene más de cuatro años, se considera totalmente amortizado. La licencia de Matlab para estudiantes tiene un coste de 35€<sup>1</sup>, por tanto, los costes materiales se reducen a la siguiente tabla:

Concepto	Coste (€)
Ordenador	0
Licencia Matlab	35
<b>TOTAL</b>	<b>35</b>

Tabla 9.1: Costes debidos a material

Los costes derivados del gasto energético del proyecto, quedan reducidos al consumo eléctrico para la luz i para el uso del ordenador. La conexión a internet ha sido necesaria para la búsqueda de artículos científicos, fórmulas y antecedentes del proyecto y, por tanto, el uso que se ha hecho de internet ha sido constante mientras se realizaba el proyecto.

Concepto	Coste (€/mes)	Meses	Porcentaje utilización respecto total	Coste Total (€)
Consumo eléctrico	75,00	7	1%	5,25
Internet	34,90	7	10%	24,43
<b>TOTAL</b>				<b>29,68</b>

Tabla 9.2: Costes debidos al gasto energético

<sup>1</sup> <https://es.mathworks.com/pricing-licensing/index.html?intendeduse=student>

En resumen, los gastos totales aproximados se encuentran en la tabla (9.3). Como se puede ver, ninguno de los costes es significativo al tratarse únicamente de creación de software.

Recurso	Coste (€)
Material	35
Energético	29,68
<b>COSTE TOTAL</b>	<b>64,68</b>

Tabla 9.3: Costes Totales

Respecto al impacto ambiental del proyecto, como el trabajo ha consistido en desarrollo de software, no se han generado residuos de ningún tipo. El impacto ambiental se reduce a las emisiones de CO<sub>2</sub> derivadas del consumo eléctrico. Se puede decir que el impacto sobre el medio ambiente es muy bajo.



## Conclusiones

En el presente proyecto se ha conseguido alcanzar el objetivo de realizar el estudio de un robot paralelo de seis grados de libertad. Se han calculado tanto la cinemática inversa como la directa, proporcionando así las dos perspectivas. Haciendo uso de la cinemática inversa, se han podido concluir las posiciones de la plataforma permitidas por el diseño del robot. Asimismo, la cinemática inversa ha servido para poder realizar el estudio del espacio de trabajo, elemento importante a la hora de trabajar con robots paralelos.

Se ha concluido del análisis de las singularidades, que varias de ellas no pueden darse nunca en la plataforma Hexaglide. Dicho análisis facilita también el futuro control de la plataforma. Una mejora consistiría en realizar dicho estudio más profundamente, en vez de restringir los movimientos del robot a un área en la que se ha podido comprobar que estaba alejada de las singularidades.

Por otro lado, se ha resuelto también la dinámica directa e inversa del robot, facilitando así conocer mejor el comportamiento del robot. Esta sección es interesante, ya que tiene diversas aplicaciones. Se puede, por un lado, realizar un estudio de las fuerzas necesarias para producir un movimiento deseado. Por otro lado, se puede conocer el comportamiento de la plataforma ante determinadas fuerzas de los actuadores. Además, el estudio dinámico permite saber la fuerza y momento máximos que puede soportar el robot.

Al realizar dichos cálculos, se han adoptado una serie de simplificaciones para facilitar la primera aproximación con el robot. Otro campo de mejora sería realizar un estudio más complejo de la dinámica, como por ejemplo no considerar despreciable la masa de las barras y por tanto, calcular la fuerza que se deriva de ellas y que actúa sobre la plataforma.

Es interesante haber realizado un estudio completo de un robot paralelo, ya que éstos son menos habituales que los robots serie. Como se ha visto a lo largo del trabajo, los robots paralelos presentan una serie de ventajas respecto a los robots serie que deben ser tenidas en consideración, ya que dependiendo de los requerimientos, un robot paralelo puede ser mejor solución que un robot serie.

Finalmente, haciendo uso del programa *Matlab*<sup>®</sup> se ha implementado un programa para facilitar el estudio del robot, buscando hacerlo intuitivo para facilitar trabajos posteriores que se quieran realizar, como por ejemplo, el control del robot. Se han añadido todos los campos comentados anteriormente,



de forma que se pueda acceder con agilidad a cualquiera de las funciones y así poder extraer las conclusiones necesarias.

## Agradecimientos

Quiero agradecer a Federico Thomas y a Lluís Ros, el haberme ofrecido la posibilidad de realizar el trabajo de fin de Grado en el Instituto de Robótica e Informática Industrial. Agradecerles también el apoyo y la atención que me han prestado a lo largo del proyecto, sin el cual nada de esto habría sido posible.

Agradecer también a Patrick Grosch el tiempo que ha dedicado a resolver mis dudas.

Finalmente, quiero agradecer Miriam Febrer y Francesc Roca su dedicación y ayuda. Y también a todas esas otras personas que me han apoyado a lo largo del proyecto.

## Bibliografía

### Referencias bibliográficas

- [1] Albert Bonet Muñoz. "Control de la posició i balanceig d'una bola sobre un pla emprant una plataforma del tipus Stewart-Gough". Treball fi de grau. ETSEIB, UPC, 2015.
- [2] A. Vivas, "Robótica paralela: aplicaciones industriales, modelado y control."
- [3] P. I. Corke, "A robotics toolbox for MATLAB," IEEE Robot. Autom. Mag., 1996.
- [4] B. D. Harper, "Solving Dynamics Problems in MATLAB."
- [5] J. Fu, F. Gao, Y. Pan, and H. Du, "Forward kinematics solutions of a special six-degree-of-freedom parallel manipulator with three limbs," Adv. Mech. Eng., vol. 7, no. 5, pp. 1–11, 2015.
- [6] I. Ansoategui, A. Lamikiz, and J. Aguirrebeitia, "Estimación de errores de posición y velocidad en centros de mecanizado de cinco ejes."
- [7] I. A. Bonev, "Analysis and design of 6-DOF 6-PRRS parallel manipulators," 1998.
- [8] T. Jacobian, "Platform Manipulators," 2011.
- [9] A. Codourey, M. Honegger, E. Burdet "A body-oriented method for dynamic modeling and adaptative control of fully parallel robots."
- [10] A. O. Anidu, S. A. Arekete, A. O. Adedayo, and A. O. Adekoya, "Dynamic Computation of Runge-Kutta's Fourth-Order Algorithm for First and Second Order Ordinary Differential Equation Using Java."
- [11] A. W. Sandvik, "Numerical Solutions of Classical Equations of Motion," PY Comput. Phys., vol. 502, 2015.
- [12] V. Ganapathi, "Simulation of Rigid Body Dynamics in Matlab," 2005.
- [13] A. M. Lopes, "A Fast Dynamic Model for a Hexa-Type Parallel Manipulator," no. x, 2006.