

Trabajo Fin de Máster  
Máster en Ingeniería Aeronáutica

Análisis del desacoplo dinámico de robots aéreos

Autor: Rafael Martínez Vázquez

Tutor: José Ángel Acosta Rodríguez

Dep. de Ingeniería de Sistemas y Automática  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2016





Trabajo Fin de Máster  
Máster en Ingeniería Aeronáutica

# **Análisis del desacoplo dinámico de robots aéreos**

Autor:

Rafael Martínez Vázquez

Tutor:

José Ángel Acosta Rodríguez

Profesor Titular

Dep. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2016



Trabajo Fin de Máster: Análisis del desacoplo dinámico de robots aéreos

Autor: Rafael Martínez Vázquez

Tutor: José Ángel Acosta Rodríguez

El tribunal nombrado para juzgar el Trabajo arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocal/es:

Secretario:

Acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:



# Índice general

Notación . . . . .	III
<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del proyecto . . . . .	1
1.2. Motivaciones . . . . .	2
1.3. Objetivos principales . . . . .	4
<b>2. Construcción de los modelos cinemático y dinámico del sistema <i>quadrotor-manipulador</i></b>	<b>7</b>
2.1. Cinemática . . . . .	8
2.1.1. Cálculo de posición, orientación y velocidad del efector final . . . . .	9
2.2. Dinámica . . . . .	12
2.2.1. Matriz de inercia . . . . .	12
2.2.2. Matriz de Coriolis . . . . .	15
2.2.3. Vector fuerza gravitatoria . . . . .	16
2.2.4. Acción de control . . . . .	16
2.2.5. Vector de perturbaciones externas . . . . .	17
2.3. Modelo numérico del sistema <i>quadrotor-manipulador</i> con 4 enlaces y efector final	17
2.3.1. Cinemática . . . . .	17
2.3.2. Dinámica . . . . .	19
2.4. Conclusiones alcanzadas y aspectos a tener en cuenta tras la simulación del modelo . . . . .	22
2.4.1. Excitación de $\tau_{\phi_y}$ . . . . .	23
2.4.2. Excitación de $\tau_{\phi_p}$ . . . . .	24
2.4.3. Excitación de $\tau_{\phi_r}$ . . . . .	25
2.4.4. Excitación de $\tau_{\alpha_1}$ . . . . .	27
2.4.5. Excitación de $\tau_{\beta_1}$ . . . . .	29
2.4.6. Excitación de $\tau_{\beta_2}$ . . . . .	30
<b>3. Descomposición de la dinámica del sistema <i>quadrotor-manipulador</i> en dos dinámicas desacopladas</b>	<b>33</b>
3.1. Simulación del modelo con la nueva estructura desacoplada . . . . .	37
3.1.1. Excitación de $\tau_L$ . . . . .	38
3.1.2. Excitación de $\tau_E$ . . . . .	41
<b>4. Modelo Simulink del sistema <i>quadrotor-manipulador</i></b>	<b>47</b>

<b>5. Modelo Simulink del sistema <i>quadrotor</i>-manipulador con dinámicas desacopladas</b>	<b>51</b>
<b>A. Estructura modelo Simulink</b>	<b>55</b>
<b>B. Estructura modelo Simulink con dinámicas desacopladas</b>	<b>57</b>
<b>C. Código de las distintas funciones usadas en el modelo</b>	<b>59</b>
C.1. Función "jacobianos" . . . . .	59
C.2. Función "INERCIA" . . . . .	61
C.3. Función "CORIOLIS" . . . . .	62
C.4. Función "posicionpuntos" . . . . .	65
C.5. Función "graficas" . . . . .	66
C.6. Función "centromasasinicial" . . . . .	70
<b>Bibliografía</b>	<b>73</b>



## Notación

Símbolo	Unidad	Nombre
$q$	$vble$	Vector de estado de las variables del problema
$p$	$m$	Vector de coordenadas espaciales del <i>quadrotor</i>
$x$	$m$	Coordenada espacial del <i>quadrotor</i> en el eje “North” inercial
$y$	$m$	Coordenada espacial del <i>quadrotor</i> en el eje “East” inercial
$z$	$m$	Coordenada espacial del <i>quadrotor</i> en el eje “Down” inercial
$\phi$	$rad$	Vector de ángulos de Euler de la actitud del <i>quadrotor</i>
$\phi_r$	$rad$	Ángulo de alabeo del <i>quadrotor</i>
$\phi_p$	$rad$	Ángulo de cabeceo del <i>quadrotor</i>
$\phi_y$	$rad$	Ángulo de guiñada del <i>quadrotor</i>
$\theta$	$rad$	Vector de ángulos de las articulaciones del brazo robótico
$\alpha_i$	$rad$	Ángulo que gira el sistema de coordenadas del elemento $i$ en torno al eje “Down” del sistema de coordenadas del elemento $i - 1$ (equivalente a el ángulo $\phi$ de las coordenadas esféricas)
$\beta_i$	$rad$	Ángulo que gira el sistema de coordenadas del elemento $i$ en torno al eje “East” del sistema de coordenadas intermedio resultante del giro previo del ángulo $\alpha_i$ (equivalente a el ángulo $\theta$ de las coordenadas esféricas)
$J_{v_0}$	$vble$	Jacobiano de traslación del <i>quadrotor</i>
$J_{w_0}$	$vble$	Jacobiano de rotación del <i>quadrotor</i>
$J_{v_j}$	$vble$	Jacobiano de traslación del elemento $j$
$J_{w_j}$	$vble$	Jacobiano de rotación del elemento $j$
$v_0$	$m/s$	Vector velocidad de traslación del CG del <i>quadrotor</i>
$w_0$	$rad/s$	Vector velocidad angular del CG del <i>quadrotor</i>
$v_j$	$m/s$	Vector velocidad de traslación del CG del elemento $j$
$w_j$	$rad/s$	Vector velocidad angular del CG del elemento $j$
$M$	$vble$	Matriz de inercia del sistema <i>quadrotor</i> -manipulador
$m_0$	$Kg$	Masa del <i>quadrotor</i>
$m_j$	$Kg$	Masa del elemento $j$
$I_0$	$Kg \cdot m^2$	Momento de inercia del <i>quadrotor</i> respecto de su centro de masa expresado en ejes cuerpo del <i>quadrotor</i>
$I_j$	$Kg \cdot m^2$	Momento de inercia del elemento $j$ respecto de su centro de masa expresado en el sistema de referencia del elemento $j$
$R_j$	-	Matriz de rotación para pasar del sistema de coordenadas del elemento $j$ al sistema inercial



# Capítulo 1

## Introducción

A lo largo de este capítulo se va a realizar una descripción del contenido del trabajo, así como las motivaciones que originan el mismo. Por último se describen los objetivos principales que se quieren alcanzar.

### 1.1. Descripción del proyecto

En este trabajo lo que se pretende es descomponer la dinámica de Lagrange de un sistema *quadrotor*-manipulador, consistente en 6 grados de libertad asociados al *quadrotor* (posición de su centro de gravedad y actitud mediante los ángulos de Euler del mismo) y  $2m + 2$  grados de libertad asociados al manipulador (correspondientes a los ángulos que forman entre si los distintos  $m$  elementos del manipulador y el efector final), en dos sistemas completamente desacoplados:

- Por un lado tendríamos la dinámica de la traslación del centro de masas en  $E(3)$  [1], que es el punto de masas del sistema sobre el que únicamente afectan a su dinámica la fuerza de empuje de los rotores asociada a ejes cuerpo del *quadrotor* y el efecto de la gravedad.
- Por otro lado estaría la dinámica de rotación interna con  $(3 + 2m + 2)$  grados de libertad, asociados a los ángulos de Euler del *quadrotor* y a la configuración de ángulos del manipulador con efector final, dicha dinámica tiene la forma de la dinámica estándar de Lagrange de los manipuladores robóticos pero sin existir efecto de la gravedad.

Esto simplificará mucho el control del sistema y permitirá nuevas leyes de control novedosas para el seguimiento de la trayectoria del efector final, las cuales nos permitirán asignar diferentes funciones al control de la dinámica del centro de masas (como un mayor dominio durante tareas de transporte) y a la dinámica de rotación interna (como un mayor dominio para el control preciso del efector final).

Para poder llevar todo esto a cabo va a ser necesario en primer lugar realizar un modelo cinemático y dinámico de nuestro sistema *quadrotor*-manipulador, al cual le realizaremos el cambio de variables descrito en [2].

## 1.2. Motivaciones

Los *quadrotores* han sido investigados extensamente durante los últimos años debido a la rapidez de sus actuaciones, facilidad de control y asequibilidad; con grandes y rápidos avances en sus sensores, actuadores, materiales y sistemas embebidos.

Los *quadrotores* son el medio para extender futuramente las habilidades de los robots móviles terrestres a el espacio tridimensional, algunas de las posibles aplicaciones podrían ser: inspección remota del terreno, fotografía aérea, vigilancia y reconocimiento, etc.



Figura 1.1: Dron-ambulancia:transporta un desfibrilador, vuela a 100 km/h y puede llegar al lugar de los hechos en un minuto [3].

Para hacer que un *quadrotor* sea realmente útil y versátil es deseable poder dotar a este de la capacidad de manipular e interactuar con el entorno. Algunas de las posibles aplicaciones que se podrían dar si esto se hace posible son:

- Tareas de transporte de cargas suspendidas con cables usando uno o varios *quadrotores*.
- Agarrar cargas usando una pinza acoplada al *quadrotor*.
- Análisis de estabilidad para prevenir inestabilidades debidas a cargas de inercia.
- Manipulación aérea con una herramienta rígidamente unida al *quadrotor* y análisis de su dinámica interna.



Figura 1.2: Dron capaz de realizar un seguimiento continuo de cultivos y detectar problemas que a vista de suelo pasan desapercibidos como zonas sin sembrar, con falta de crecimiento por diferencias en el suelo,etc [4].



Figura 1.3: Fotografía tomada por un dron tras el terremoto en Nepal. Estas fotos serán procesadas para crear un modelo 3D y así tener más información de la destrucción producida [5].



### 1.3 Objetivos principales

---

Las aplicaciones de los drones se están multiplicando y hoy invaden la agricultura o la arquitectura, la topografía, el control del tráfico urbano o la seguridad ciudadana, la investigación de la fauna o seguimiento de la evolución de los recursos naturales, la cartografía de zonas tropicales o el transporte de medicinas a zonas de difícil acceso; incluso pueden resultar útiles en operaciones de rescate después de catástrofes naturales.

Es por esto que es de gran importancia investigar en nuevas formas de control de estos dispositivos, que faciliten su operación para poder realizar toda esta gran variedad de tareas de la forma más precisa y efectiva posible.

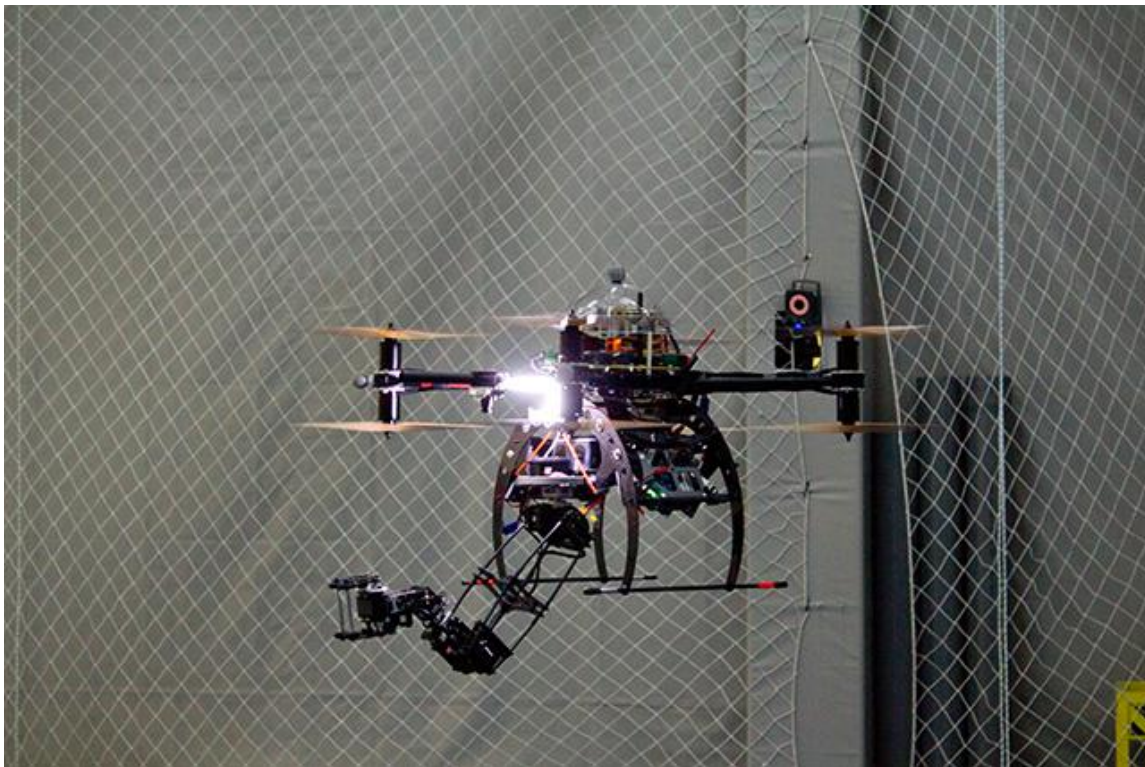


Figura 1.4: Robots voladores dotados de brazos manipuladores con varias articulaciones capaces de colaborar para agarrar, transportar y depositar piezas de manera segura y eficaz diseñados en el proyecto europeo ARCAS (Aerial Robotics Cooperative Assembly System) [6].

### 1.3. Objetivos principales

El objetivo principal de este proyecto es ilustrar y estudiar una nueva forma de control de la dinámica de Lagrange de los sistemas *quadrotor*-manipulador descomponiendo esta última en dos dinámicas desacopladas como son la del centro de masas y la de rotación interna debida a la rotación del *quadrotor* y a la configuración del manipulador.

Para ello se va a analizar el problema tridimensional para el caso de un *quadrotor* con sus tres ángulos de Euler para determinar su actitud (6 variables: 3 coordenadas espaciales y 3 ángulos); y que lleva acoplado un manipulador de 4 enlaces más efector final donde todos

ellos tienen movimiento de guiñada y cabeceo (10 variables más asociadas al manipulador, 2 ángulos en cada articulación).

Los pasos a seguir serán:

- Cálculo de los modelos cinemático y dinámico de nuestro sistema *quadrotor*-manipulador.
- Descomposición de la dinámica del sistema *quadrotor*-manipulador en las dinámicas desacopladas de dos sistemas distintos.





## Capítulo 2

# Construcción de los modelos cinemático y dinámico del sistema *quadrotor*-manipulador

Se considera un sistema *quadrotor* con un brazo robótico de  $m$  enlaces y efector final,  $2m + 2$  grados de libertad, como el mostrado en la Figura 2.1. Los rotores del *quadrotor* son no coaxiales. En la Figura 2.2 se puede observar como se han definido los ángulos  $\alpha$  y  $\beta$  de cada articulación, respectivamente movimientos de guiñada y cabeceo, similares a los ángulos  $\phi$  y  $\theta$  usados en coordenadas esféricas.

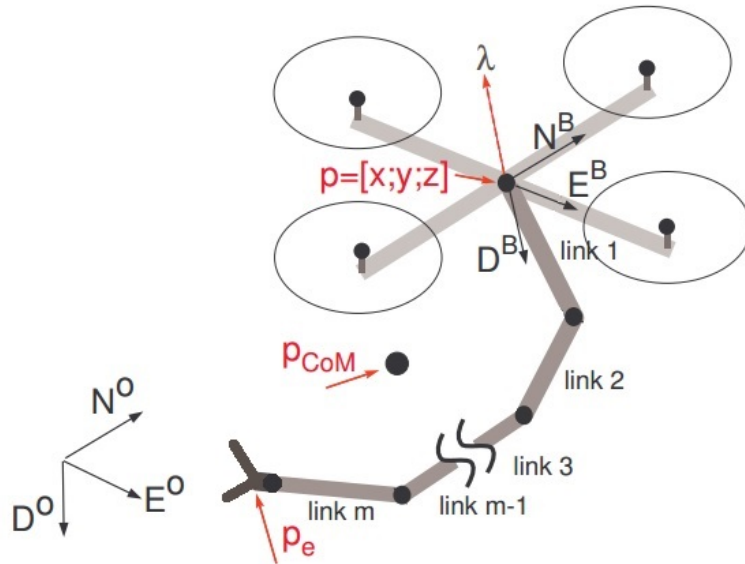


Figura 2.1: *Quadrotor* con brazo robótico de  $2m + 2$  grados de libertad:  $p = [x; y; z]$  es la posición del centro de la plataforma del *quadrotor*,  $p_e = [x_e; y_e; z_e]$  es la posición del efector final,  $\lambda$  es la fuerza propulsiva y  $p_{CoM}$  es la posición del centro de gravedad total del sistema *quadrotor*-manipulador.

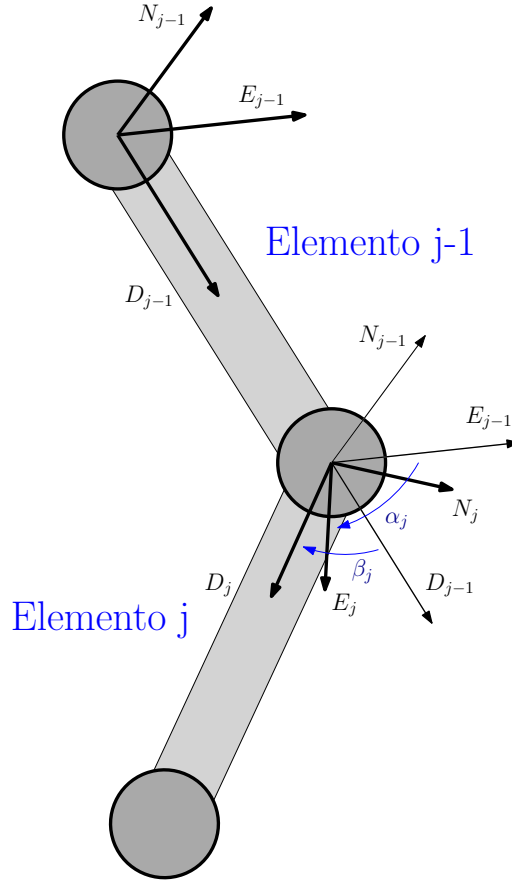


Figura 2.2: Variación de la orientación del sistema de referencia del elemento  $j$  con respecto a su elemento predecesor  $j - 1$  dada por los ángulos  $\alpha_j$  (giro en torno a  $D_{j-1}$ ) y  $\beta_j$  (giro en torno a  $E_{j-1}$  resultante del giro previo). Destacar que los elementos están orientados según la dirección  $D$  de su sistema de referencia y que en este caso el ángulo  $\beta_j$  representado es negativo.

## 2.1. Cinemática

La configuración del sistema *quadrotor*-manipulador puede ser definida por:

$$q = [p; \phi; \theta] \in \mathfrak{R}^n, \quad n = 6 + 2m + 2 \quad (2.1)$$

donde  $p = [x; y; z] \in \mathfrak{R}^3$  es la posición del centro de gravedad del *quadrotor* en el sistema de referencia inercial  $(N^0, E^0, D^0)$ ,  $\phi = [\phi_r; \phi_p; \phi_y] \in \mathfrak{R}^3$  son los ángulos de alabeo, cabeceo y guiñada del *quadrotor*; y  $\theta = [\alpha_1; \beta_1; \dots; \alpha_m; \beta_m; \alpha_E; \beta_E] \in \mathfrak{R}^{2m+2}$  son los ángulos de las articulaciones del brazo robótico.

Pasamos a calcular la posición, orientación y velocidad del efector final a partir de las distintas variables usando matrices de cambio de sistema de referencia [7].

### 2.1.1. Cálculo de posición, orientación y velocidad del efector final

#### Posición y orientación

Lo primero que vamos a calcular es la matriz de rotación del sistema de referencia ejes cuerpo del *quadrotor* al sistema de referencia inercial. Esto se consigue a partir de 3 matrices de giro básicas, cada una de las cuales girando sobre un eje adecuado uno de los respectivos ángulos de Euler:

##### 1. Guiñada

$$\begin{pmatrix} N_0 \\ E_0 \\ D_0 \end{pmatrix} = \begin{pmatrix} \cos(\phi_y) & -\text{sen}(\phi_y) & 0 \\ \text{sen}(\phi_y) & \cos(\phi_y) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} N_{0'} \\ E_{0'} \\ D_{0'} \end{pmatrix} \Rightarrow R_{0'}^0 = \begin{pmatrix} \cos(\phi_y) & -\text{sen}(\phi_y) & 0 \\ \text{sen}(\phi_y) & \cos(\phi_y) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.2)$$

##### 2. Cabeceo

$$\begin{pmatrix} N_{0'} \\ E_{0'} \\ D_{0'} \end{pmatrix} = \begin{pmatrix} \cos(\phi_p) & 0 & \text{sen}(\phi_p) \\ 0 & 1 & 0 \\ -\text{sen}(\phi_p) & 0 & \cos(\phi_p) \end{pmatrix} \begin{pmatrix} N_{0''} \\ E_{0''} \\ D_{0''} \end{pmatrix} \Rightarrow R_{0''}^{0'} = \begin{pmatrix} \cos(\phi_p) & 0 & \text{sen}(\phi_p) \\ 0 & 1 & 0 \\ -\text{sen}(\phi_p) & 0 & \cos(\phi_p) \end{pmatrix} \quad (2.3)$$

##### 3. Alabeo

$$\begin{pmatrix} N_{0''} \\ E_{0''} \\ D_{0''} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_r) & -\text{sen}(\phi_r) \\ 0 & \text{sen}(\phi_r) & \cos(\phi_r) \end{pmatrix} \begin{pmatrix} N_B \\ E_B \\ D_B \end{pmatrix} \Rightarrow R_B^{0''} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_r) & -\text{sen}(\phi_r) \\ 0 & \text{sen}(\phi_r) & \cos(\phi_r) \end{pmatrix} \quad (2.4)$$

Siendo la matriz definitiva de rotación de ejes cuerpo del *quadrotor* a los ejes del sistema inercial:

$$R_B^0 = R_{0'}^0 \cdot R_{0''}^{0'} \cdot R_B^{0''} = \begin{pmatrix} c_p c_y & c_y s_p s_r - c_r s_y & s_r s_y + c_r c_y s_p \\ c_p s_y & c_r c_y + s_p s_r s_y & c_r s_p s_y - c_y s_r \\ -s_p & c_p s_r & c_p c_r \end{pmatrix} \quad (2.5)$$

donde se ha hecho un cambio en la notación para facilitar y acortar la expresión, siendo  $c = \cos$ ,  $s = \text{sen}$  y los subíndices  $y = \phi_y$ ,  $p = \phi_p$  y  $r = \phi_r$ .

Por último, para calcular la matriz de cambio de sistema de referencia es necesario añadir una traslación que se corresponde con el diferente origen de ambos sistemas:

$$T_B^0 = \begin{pmatrix} c_p c_y & c_y s_p s_r - c_r s_y & s_r s_y + c_r c_y s_p & x \\ c_p s_y & c_r c_y + s_p s_r s_y & c_r s_p s_y - c_y s_r & y \\ -s_p & c_p s_r & c_p c_r & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.6)$$

Pasamos a calcular ahora la matriz de cambio de sistema de referencia de un elemento al sistema de referencia del elemento anterior. Para ello se actúa como se ha hecho anteriormente haciendo una composición de las matrices elementales de rotación de cada uno de los ángulos de la articulación:

1. Giro de ángulo  $\alpha_j$  en torno al eje D del sistema de referencia del elemento  $j - 1$

$$\begin{pmatrix} N_{j-1} \\ E_{j-1} \\ D_{j-1} \end{pmatrix} = \begin{pmatrix} \cos(\alpha_j) & -\text{sen}(\alpha_j) & 0 \\ \text{sen}(\alpha_j) & \cos(\alpha_j) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} N_{j-1'} \\ E_{j-1'} \\ D_{j-1'} \end{pmatrix} \Rightarrow R_{j-1'}^{j-1} = \begin{pmatrix} \cos(\alpha_j) & -\text{sen}(\alpha_j) & 0 \\ \text{sen}(\alpha_j) & \cos(\alpha_j) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.7)$$

2. Giro de ángulo  $\beta_j$  en torno al eje E del sistema de referencia intermedio  $j - 1'$

$$\begin{pmatrix} N_{j-1'} \\ E_{j-1'} \\ D_{j-1'} \end{pmatrix} = \begin{pmatrix} \cos(\beta_j) & 0 & \text{sen}(\beta_j) \\ 0 & 1 & 0 \\ -\text{sen}(\beta_j) & 0 & \cos(\beta_j) \end{pmatrix} \begin{pmatrix} N_j \\ E_j \\ D_j \end{pmatrix} \Rightarrow R_j^{j-1'} = \begin{pmatrix} \cos(\beta_j) & 0 & \text{sen}(\beta_j) \\ 0 & 1 & 0 \\ -\text{sen}(\beta_j) & 0 & \cos(\beta_j) \end{pmatrix} \quad (2.8)$$

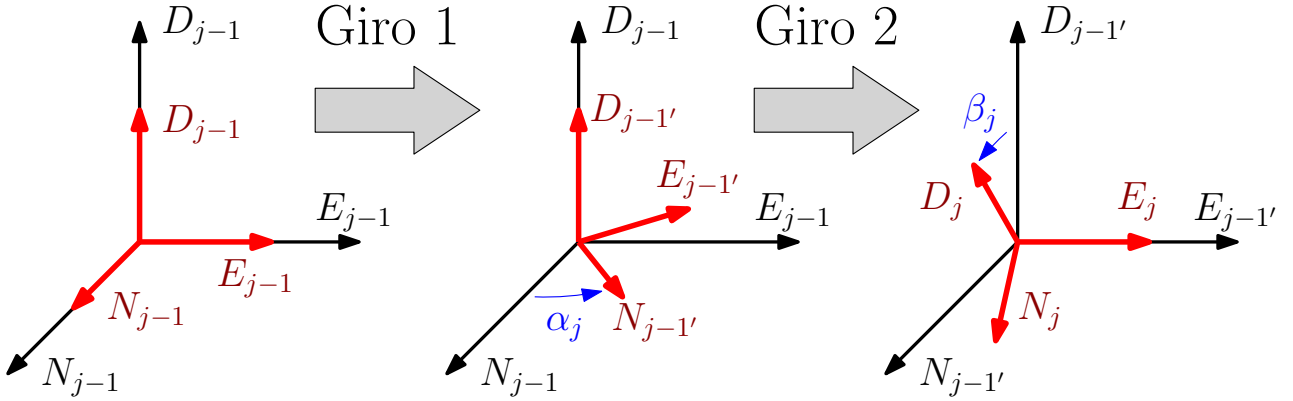


Figura 2.3: Secuencia de giros que definen la variación en la orientación del sistema de referencia del elemento  $j$  con respecto a su elemento predecesor  $j - 1$  dada por los ángulos  $\alpha_j$  y  $\beta_j$ . Destacar que los elementos están orientados según la dirección D de su sistema de referencia y que en la realidad esta va hacia abajo pero se ha invertido en la imagen por ser más sencillo de representar.

La matriz de rotación tiene por tanto la siguiente forma:

$$R_j^{j-1} = R_{j-1'}^{j-1} \cdot R_j^{j-1'} = \begin{pmatrix} \cos(\alpha_j)\cos(\beta_j) & -\text{sen}(\alpha_j) & \cos(\alpha_j)\text{sen}(\beta_j) \\ \text{sen}(\alpha_j)\cos(\beta_j) & \cos(\alpha_j) & \text{sen}(\alpha_j)\text{sen}(\beta_j) \\ -\text{sen}(\beta_j) & 0 & \cos(\beta_j) \end{pmatrix} \quad (2.9)$$

Por último, para calcular la matriz de cambio de sistema de referencia es necesario añadir una traslación que se corresponde con el diferente origen de ambos sistemas. Pasamos a calcular la expresión de dicha traslación sabiendo que  $a_j$  es la longitud del elemento  $j$  y este tiene la dirección del eje  $D_j$ :

$$R_j^{j-1} \cdot \begin{pmatrix} 0 \\ 0 \\ a_j \end{pmatrix} = \begin{pmatrix} a_j \cos(\alpha_j) \text{sen}(\beta_j) \\ a_j \text{sen}(\alpha_j) \text{sen}(\beta_j) \\ a_j \cos(\beta_j) \end{pmatrix} \quad (2.10)$$

resultando como era de esperar la expresión típica de las coordenadas esféricas. Añadiendo la traslación se obtiene finalmente:

$$T_j^{j-1} = \begin{pmatrix} \cos(\alpha_j)\cos(\beta_j) & -\text{sen}(\alpha_j) & \cos(\alpha_j)\text{sen}(\beta_j) & a_j\cos(\alpha_j)\text{sen}(\beta_j) \\ \text{sen}(\alpha_j)\cos(\beta_j) & \cos(\alpha_j) & \text{sen}(\alpha_j)\text{sen}(\beta_j) & a_j\text{sen}(\alpha_j)\text{sen}(\beta_j) \\ -\text{sen}(\beta_j) & 0 & \cos(\beta_j) & a_j\cos(\beta_j) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.11)$$

que es la matriz de cambio de sistema de referencia de un elemento al sistema de referencia del elemento anterior.

Para el caso del efector final tenemos que este tiene un sistema de referencia asociado pero con mismo origen que el sistema de referencia del último elemento, por lo que lo único que varía es la dirección de los ejes, obteniéndose:

$$T_E^m = \begin{pmatrix} \cos(\alpha_E)\cos(\beta_E) & -\text{sen}(\alpha_E) & \cos(\alpha_E)\text{sen}(\beta_E) & 0 \\ \text{sen}(\alpha_E)\cos(\beta_E) & \cos(\alpha_E) & \text{sen}(\alpha_E)\text{sen}(\beta_E) & 0 \\ -\text{sen}(\beta_E) & 0 & \cos(\beta_E) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.12)$$

Una vez que ya tenemos todas estas matrices es posible calcular la posición y orientación del efector final simplemente haciendo una composición de cambios de sistema de referencia desde el sistema de referencia del efector final hasta el sistema de referencia inercial:

$$T_E^0 = T_B^0 \cdot T_1^B \cdot T_2^1 \cdot T_3^2 \cdot \dots \cdot T_m^{m-1} \cdot T_E^m \quad (2.13)$$

Dando como resultado una matriz con la siguiente forma:

$$T_E^0 = \left( \begin{array}{c|c} R_E^0 & \vec{X}^E \\ \hline \emptyset_{1 \times 3} & 1 \end{array} \right) \quad (2.14)$$

donde  $R_E^0$  es la matriz de rotación 3x3 de los ejes del efector final a los ejes del sistema inercial y por lo tanto nos da información de la orientación del efector final; y  $\vec{X}^E$  es el vector 3x1 con las coordenadas de la posición del efector final en el sistema inercial.

## Velocidad

La velocidad del efector final en función de la variación de las distintas variables del problema va a ser calculada a partir del jacobiano de traslación del efector final  $J_{v_E}$ :

$$v^E = J_{v_E} \cdot \dot{q} \quad (2.15)$$

$$\dot{q} = \begin{pmatrix} \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial t} \\ \frac{\partial z}{\partial t} \\ \frac{\partial \phi_y}{\partial t} \\ \frac{\partial \phi_p}{\partial t} \\ \frac{\partial \phi_r}{\partial t} \\ \frac{\partial \alpha_1}{\partial t} \\ \frac{\partial \beta_1}{\partial t} \\ \vdots \\ \frac{\partial \alpha_m}{\partial t} \\ \frac{\partial \beta_m}{\partial t} \\ \frac{\partial \alpha_E}{\partial t} \\ \frac{\partial \beta_E}{\partial t} \end{pmatrix} \quad (2.16)$$

donde dicha matriz jacobiana tiene la siguiente expresión:

$$J_{v_E} = \begin{pmatrix} \frac{\partial x^E}{\partial x} & \frac{\partial x^E}{\partial y} & \frac{\partial x^E}{\partial z} & \frac{\partial x^E}{\partial \phi_y} & \frac{\partial x^E}{\partial \phi_p} & \frac{\partial x^E}{\partial \phi_r} & \frac{\partial x^E}{\partial \alpha_1} & \frac{\partial x^E}{\partial \beta_1} & \cdots & \frac{\partial x^E}{\partial \alpha_m} & \frac{\partial x^E}{\partial \beta_m} & \frac{\partial x^E}{\partial \alpha_E} & \frac{\partial x^E}{\partial \beta_E} \\ \frac{\partial y^E}{\partial x} & \frac{\partial y^E}{\partial y} & \frac{\partial y^E}{\partial z} & \frac{\partial y^E}{\partial \phi_y} & \frac{\partial y^E}{\partial \phi_p} & \frac{\partial y^E}{\partial \phi_r} & \frac{\partial y^E}{\partial \alpha_1} & \frac{\partial y^E}{\partial \beta_1} & \cdots & \frac{\partial y^E}{\partial \alpha_m} & \frac{\partial y^E}{\partial \beta_m} & \frac{\partial y^E}{\partial \alpha_E} & \frac{\partial y^E}{\partial \beta_E} \\ \frac{\partial z^E}{\partial x} & \frac{\partial z^E}{\partial y} & \frac{\partial z^E}{\partial z} & \frac{\partial z^E}{\partial \phi_y} & \frac{\partial z^E}{\partial \phi_p} & \frac{\partial z^E}{\partial \phi_r} & \frac{\partial z^E}{\partial \alpha_1} & \frac{\partial z^E}{\partial \beta_1} & \cdots & \frac{\partial z^E}{\partial \alpha_m} & \frac{\partial z^E}{\partial \beta_m} & \frac{\partial z^E}{\partial \alpha_E} & \frac{\partial z^E}{\partial \beta_E} \end{pmatrix} \quad (2.17)$$

siendo  $x^E$ ,  $y^E$  y  $z^E$  las componentes del vector  $\vec{X}^E$ .

## 2.2. Dinámica

A lo largo de esta sección lo que se pretende es obtener las matrices que caracterizan la dinámica de Lagrange del sistema completo *quadrotor*-manipulador dada por la siguiente ecuación:

$$M(r)\ddot{q} + C(r, \dot{r})\dot{q} + g(q) = \tau + f \quad (2.18)$$

donde  $r = [\phi; \theta] \in \mathfrak{R}^{3+2m+2}$  representa la rotación del *quadrotor* y los ángulos de las articulaciones del brazo robótico,  $M(r) \in \mathfrak{R}^{n \times n}$  es la matriz de inercia,  $C(r, \dot{r}) \in \mathfrak{R}^{n \times n}$  es la matriz de Coriolis,  $g(q) \in \mathfrak{R}^n$  es la fuerza gravitatoria,  $\tau \in \mathfrak{R}^n$  es la acción de control y  $f \in \mathfrak{R}^n$  es la perturbación externa.

### 2.2.1. Matriz de inercia

A continuación se va a explicar el procedimiento para calcular la matriz de inercia de nuestro sistema *quadrotor*-manipulador, para ello tenemos que comenzar definiendo los jacobianos de traslación y rotación,  $J_{v_j}(r) \in \mathfrak{R}^{3 \times n}$  y  $J_{w_j}(r) \in \mathfrak{R}^{3 \times n}$ , esenciales en el cálculo de esta:

$$v_j = J_{v_j}(r)\dot{q} \quad (2.19)$$

$$w_j = J_{w_j}(r)\dot{q} \quad (2.20)$$

para  $j = 0, 1, \dots, m$ , donde  $v_j, w_j \in \mathfrak{R}^3$  son respectivamente la velocidad de traslación del centro de gravedad y la velocidad angular del  $j$ -ésimo enlace, con  $j = 0$  representando la

plataforma del *quadrotor* y  $j = 1, \dots, m$  cada enlace del brazo robótico.

Para calcular estos jacobianos lo primero que necesitamos es calcular la posición del centro de gravedad de cada enlace y su ángulo girado respecto del sistema de referencia inercial y sus ejes, esto se hace de la siguiente forma:

$$T_{jG}^0 = T_B^0 \cdot T_1^B \cdot \dots \cdot T_{j-1}^{j-2} \cdot \begin{pmatrix} \cos(\alpha_j)\cos(\beta_j) & -\text{sen}(\alpha_j) & \cos(\alpha_j)\text{sen}(\beta_j) & k_j a_j \cos(\alpha_j)\text{sen}(\beta_j) \\ \text{sen}(\alpha_j)\cos(\beta_j) & \cos(\alpha_j) & \text{sen}(\alpha_j)\text{sen}(\beta_j) & k_j a_j \text{sen}(\alpha_j)\text{sen}(\beta_j) \\ -\text{sen}(\beta_j) & 0 & \cos(\beta_j) & k_j a_j \cos(\beta_j) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.21)$$

donde  $k_j$  es una constante adimensional que indica la posición del centro de gravedad del elemento a lo largo de este.

$$T_{jG}^0 = \left( \begin{array}{c|c} R_j^0 & X^{jG} \\ \hline \emptyset_{1 \times 3} & 1 \end{array} \right) \quad (2.22)$$

$X^{jG}$  es el vector 3x1 con las coordenadas de la posición del centro de gravedad del elemento  $j$  en el sistema inercial. Para el caso de  $j = 0$  tenemos que el centro de gravedad del *quadrotor* es:

$$X^{0G} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (2.23)$$

Para los ángulos lo hacemos de la siguiente forma:

$$\theta^{0G} = \begin{pmatrix} 0 \\ 0 \\ \phi_y \end{pmatrix} + R_{0'}^0 \cdot \begin{pmatrix} 0 \\ \phi_p \\ 0 \end{pmatrix} + R_{0'}^0 \cdot R_{0''}^{0'} \cdot \begin{pmatrix} \phi_r \\ 0 \\ 0 \end{pmatrix} \quad (2.24)$$

$$\theta^{1G} = \theta^{0G} + R_B^0 \cdot \begin{pmatrix} 0 \\ 0 \\ \alpha_1 \end{pmatrix} + R_B^0 \cdot R_{B'}^B \cdot \begin{pmatrix} 0 \\ \beta_1 \\ 0 \end{pmatrix} \quad (2.25)$$

$$\theta^{2G} = \theta^{1G} + R_1^0 \cdot \begin{pmatrix} 0 \\ 0 \\ \alpha_2 \end{pmatrix} + R_1^0 \cdot R_{1'}^1 \cdot \begin{pmatrix} 0 \\ \beta_2 \\ 0 \end{pmatrix} \quad (2.26)$$

$$\theta^{jG} = \theta^{j-1G} + R_{j-1}^0 \cdot \begin{pmatrix} 0 \\ 0 \\ \alpha_j \end{pmatrix} + R_{j-1}^0 \cdot R_{j-1'}^{j-1} \cdot \begin{pmatrix} 0 \\ \beta_j \\ 0 \end{pmatrix} \quad (2.27)$$

donde  $\theta^{jG}$  es el vector con las componentes del ángulo girado en torno a los ejes del sistema inercial del centro de gravedad del elemento  $j$ -ésimo.

Una vez que hemos calculado tanto la posición como el ángulo girado de cada uno de ellos calculamos los jacobianos de la siguiente forma:

$$J_{v_j} = \begin{pmatrix} \frac{\partial x^{jG}}{\partial x} & \frac{\partial x^{jG}}{\partial y} & \frac{\partial x^{jG}}{\partial z} & \frac{\partial x^{jG}}{\partial \phi_y} & \frac{\partial x^{jG}}{\partial \phi_p} & \frac{\partial x^{jG}}{\partial \phi_r} & \frac{\partial x^{jG}}{\partial \alpha_1} & \frac{\partial x^{jG}}{\partial \beta_1} & \dots & \frac{\partial x^{jG}}{\partial \alpha_m} & \frac{\partial x^{jG}}{\partial \beta_m} & \frac{\partial x^{jG}}{\partial \alpha_E} & \frac{\partial x^{jG}}{\partial \beta_E} \\ \frac{\partial y^{jG}}{\partial x} & \frac{\partial y^{jG}}{\partial y} & \frac{\partial y^{jG}}{\partial z} & \frac{\partial y^{jG}}{\partial \phi_y} & \frac{\partial y^{jG}}{\partial \phi_p} & \frac{\partial y^{jG}}{\partial \phi_r} & \frac{\partial y^{jG}}{\partial \alpha_1} & \frac{\partial y^{jG}}{\partial \beta_1} & \dots & \frac{\partial y^{jG}}{\partial \alpha_m} & \frac{\partial y^{jG}}{\partial \beta_m} & \frac{\partial y^{jG}}{\partial \alpha_E} & \frac{\partial y^{jG}}{\partial \beta_E} \\ \frac{\partial z^{jG}}{\partial x} & \frac{\partial z^{jG}}{\partial y} & \frac{\partial z^{jG}}{\partial z} & \frac{\partial z^{jG}}{\partial \phi_y} & \frac{\partial z^{jG}}{\partial \phi_p} & \frac{\partial z^{jG}}{\partial \phi_r} & \frac{\partial z^{jG}}{\partial \alpha_1} & \frac{\partial z^{jG}}{\partial \beta_1} & \dots & \frac{\partial z^{jG}}{\partial \alpha_m} & \frac{\partial z^{jG}}{\partial \beta_m} & \frac{\partial z^{jG}}{\partial \alpha_E} & \frac{\partial z^{jG}}{\partial \beta_E} \end{pmatrix} \quad (2.28)$$

$$J_{w_j} = \begin{pmatrix} \frac{\partial \theta_x^{jG}}{\partial x} & \frac{\partial \theta_x^{jG}}{\partial y} & \frac{\partial \theta_x^{jG}}{\partial z} & \frac{\partial \theta_x^{jG}}{\partial \phi_y} & \frac{\partial \theta_x^{jG}}{\partial \phi_p} & \frac{\partial \theta_x^{jG}}{\partial \phi_r} & \frac{\partial \theta_x^{jG}}{\partial \alpha_1} & \frac{\partial \theta_x^{jG}}{\partial \beta_1} & \cdots & \frac{\partial \theta_x^{jG}}{\partial \alpha_m} & \frac{\partial \theta_x^{jG}}{\partial \beta_m} & \frac{\partial \theta_x^{jG}}{\partial \alpha_E} & \frac{\partial \theta_x^{jG}}{\partial \beta_E} \\ \frac{\partial \theta_y^{jG}}{\partial x} & \frac{\partial \theta_y^{jG}}{\partial y} & \frac{\partial \theta_y^{jG}}{\partial z} & \frac{\partial \theta_y^{jG}}{\partial \phi_y} & \frac{\partial \theta_y^{jG}}{\partial \phi_p} & \frac{\partial \theta_y^{jG}}{\partial \phi_r} & \frac{\partial \theta_y^{jG}}{\partial \alpha_1} & \frac{\partial \theta_y^{jG}}{\partial \beta_1} & \cdots & \frac{\partial \theta_y^{jG}}{\partial \alpha_m} & \frac{\partial \theta_y^{jG}}{\partial \beta_m} & \frac{\partial \theta_y^{jG}}{\partial \alpha_E} & \frac{\partial \theta_y^{jG}}{\partial \beta_E} \\ \frac{\partial \theta_z^{jG}}{\partial x} & \frac{\partial \theta_z^{jG}}{\partial y} & \frac{\partial \theta_z^{jG}}{\partial z} & \frac{\partial \theta_z^{jG}}{\partial \phi_y} & \frac{\partial \theta_z^{jG}}{\partial \phi_p} & \frac{\partial \theta_z^{jG}}{\partial \phi_r} & \frac{\partial \theta_z^{jG}}{\partial \alpha_1} & \frac{\partial \theta_z^{jG}}{\partial \beta_1} & \cdots & \frac{\partial \theta_z^{jG}}{\partial \alpha_m} & \frac{\partial \theta_z^{jG}}{\partial \beta_m} & \frac{\partial \theta_z^{jG}}{\partial \alpha_E} & \frac{\partial \theta_z^{jG}}{\partial \beta_E} \end{pmatrix} \quad (2.29)$$

siendo  $x^{jG}$ ,  $y^{jG}$  y  $z^{jG}$  las componentes del vector  $X^{\vec{j}G}$ ; y  $\theta_x^{jG}$ ,  $\theta_y^{jG}$  y  $\theta_z^{jG}$  las componentes del vector  $\theta^{\vec{j}G}$ .

Ambas matrices tienen las siguientes estructuras:

$$J_{v_j}(r) = \begin{pmatrix} I_3 & J_{v_j}^{r_1} & \cdots & J_{v_j}^{r_{3+2m+2}} \end{pmatrix} \quad (2.30)$$

$$J_{w_j}(r) = \begin{pmatrix} \emptyset_3 & J_{w_j}^{r_1} & \cdots & J_{w_j}^{r_{3+2m+2}} \end{pmatrix} \quad (2.31)$$

donde  $\emptyset_3 \in \mathfrak{R}^{3 \times 3}$  es la matriz donde todos sus elementos son nulos. Destacar que los jacobianos  $J_{v_j}$  y  $J_{w_j}$  son función únicamente de  $r$  y no de  $p = [x; y; z]$ .

Haciendo uso de (2.19) y (2.20) podemos expresar la energía cinética del sistema *quadrotor*-manipulador como:

$$\kappa = \frac{1}{2} \dot{q}^T M(r) \dot{q} \quad (2.32)$$

con la matriz de inercia  $M(r) \in \mathfrak{R}^{n \times n}$  calculada con la siguiente expresión:

$$M(r) = \sum_{j=0}^{m+1} \left[ m_j J_{v_j}^T J_{v_j} + J_{w_j}^T R_j I_j R_j^T J_{w_j} \right] \quad (2.33)$$

donde  $m_j$  e  $I_j$  son la masa y el momento de inercia del  $j$ -ésimo enlace respecto a su centro de gravedad expresado en los ejes cuerpo del elemento;  $R_j$  es la matriz de rotación de los ejes cuerpo del elemento  $j$ -ésimo a los ejes del sistema de referencia inercial. Destacar que el efector final se considera como un elemento más de subíndice  $m+1$  a la hora de realizar los cálculos.

Se puede demostrar que dicha matriz de inercia  $M(r)$  tiene la siguiente estructura:

$$M(r) = \begin{pmatrix} M_p & M_{pr}(r) \\ M_{pr}^T(r) & M_r(r) \end{pmatrix} \quad (2.34)$$

donde  $M_p = m_L I_3 \in \mathfrak{R}^{3 \times 3}$  con  $m_L = \sum_{j=0}^m m_j$  siendo la masa total del sistema *quadrotor*-manipulador. Para ello vamos a introducir en la ecuación (2.33) las formas que tienen los jacobianos de traslación (2.30) y rotación (2.31):

$$\begin{aligned} M(r) = \sum_{j=0}^{m+1} & \left[ m_j \begin{pmatrix} I_3 \\ J_{v_j}^{r_1 T} \\ \vdots \\ J_{v_j}^{r_{3+2m+2} T} \end{pmatrix} \begin{pmatrix} I_3 & J_{v_j}^{r_1} & \cdots & J_{v_j}^{r_{3+2m+2}} \end{pmatrix} + \right. \\ & \left. + \begin{pmatrix} \emptyset_3 \\ J_{w_j}^{r_1 T} \\ \vdots \\ J_{w_j}^{r_{3+2m+2} T} \end{pmatrix} R_j I_j R_j^T \begin{pmatrix} \emptyset_3 & J_{w_j}^{r_1} & \cdots & J_{w_j}^{r_{3+2m+2}} \end{pmatrix} \right] \end{aligned} \quad (2.35)$$



donde operando resulta:

$$\begin{aligned}
 M(r) = \sum_{j=0}^{m+1} & \left[ m_j \left( \begin{array}{c|c} I_3 & I_3 \cdot \begin{bmatrix} J_{v_j}^{r_1} & \dots & J_{v_j}^{r_{3+2m+2}} \end{bmatrix} \\ \hline \begin{bmatrix} J_{v_j}^{r_1 T} \\ \vdots \\ J_{v_j}^{r_{3+2m+2} T} \end{bmatrix} \cdot I_3 & \begin{bmatrix} J_{v_j}^{r_1 T} \\ \vdots \\ J_{v_j}^{r_{3+2m+2} T} \end{bmatrix} \cdot \begin{bmatrix} J_{v_j}^{r_1} & \dots & J_{v_j}^{r_{3+2m+2}} \end{bmatrix} \end{array} \right) + \\
 & + \left( \begin{array}{c|c} \emptyset_3 & \emptyset_{3 \times 13} \\ \hline \emptyset_{13 \times 3} & \begin{bmatrix} J_{w_j}^{r_1 T} \\ \vdots \\ J_{w_j}^{r_{3+2m+2} T} \end{bmatrix} R_j I_j R_j^T \begin{bmatrix} J_{w_j}^{r_1} & \dots & J_{w_j}^{r_{3+2m+2}} \end{bmatrix} \end{array} \right) \Bigg] \quad (2.36)
 \end{aligned}$$

Con lo que concluimos que las expresiones de las submatrices que aparecen en (2.34) son:

$$M_p = \sum_{j=0}^{m+1} [m_j I_3] \quad (2.37)$$

$$M_{pr}(r) = \sum_{j=0}^{m+1} \left[ m_j \cdot \begin{pmatrix} J_{v_j}^{r_1} & \dots & J_{v_j}^{r_{3+2m+2}} \end{pmatrix} \right] \quad (2.38)$$

$$\begin{aligned}
 M_r(r) = \sum_{j=0}^{m+1} & \left[ m_j \begin{pmatrix} J_{v_j}^{r_1 T} \\ \vdots \\ J_{v_j}^{r_{3+2m+2} T} \end{pmatrix} \begin{pmatrix} J_{v_j}^{r_1} & \dots & J_{v_j}^{r_{3+2m+2}} \end{pmatrix} + \\
 & + \begin{pmatrix} J_{w_j}^{r_1 T} \\ \vdots \\ J_{w_j}^{r_{3+2m+2} T} \end{pmatrix} R_j I_j R_j^T \begin{pmatrix} J_{w_j}^{r_1} & \dots & J_{w_j}^{r_{3+2m+2}} \end{pmatrix} \right] \quad (2.39)
 \end{aligned}$$

### 2.2.2. Matriz de Coriolis

Una vez se tiene la matriz de inercias se obtiene la matriz de términos de Coriolis mediante los símbolos de Christoffel de esta:

$$C(q, \dot{q}) \equiv C_{ij} = \sum_{k=1}^n c_{ijk} \dot{q}_k \quad (2.40)$$

siendo

$$c_{ijk} = \frac{1}{2} (M_{ij,k} + M_{ik,j} - M_{jk,i}) = \frac{1}{2} \left( \frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{jk}}{\partial q_i} \right) \quad (2.41)$$

Se calculan por tanto las derivadas no nulas de los términos de la matriz de inercias con respecto a cada uno de los grados de libertad para posteriormente calcular los términos de la matriz de Coriolis.

### 2.2.3. Vector fuerza gravitatoria

El vector fuerza gravitatoria tiene la siguiente expresión :

$$g(q) = -g \begin{pmatrix} 0 \\ 0 \\ m_L \\ c_{\phi_y}^z \\ c_{\phi_p}^z \\ c_{\phi_r}^z \\ c_{\alpha_1}^z \\ c_{\beta_1}^z \\ \vdots \\ c_{\alpha_m}^z \\ c_{\beta_m}^z \end{pmatrix} \quad (2.42)$$

donde  $m_L$  es la masa total del sistema y el resto de términos se obtienen de parte de la matriz de inercia [2]:

$$M_{pr}(r) = \begin{pmatrix} c_{\phi_y}^x & c_{\phi_p}^x & c_{\phi_r}^x & c_{\alpha_1}^x & c_{\beta_1}^x & \dots & c_{\alpha_m}^x & c_{\beta_m}^x \\ c_{\phi_y}^y & c_{\phi_p}^y & c_{\phi_r}^y & c_{\alpha_1}^y & c_{\beta_1}^y & \dots & c_{\alpha_m}^y & c_{\beta_m}^y \\ c_{\phi_y}^z & c_{\phi_p}^z & c_{\phi_r}^z & c_{\alpha_1}^z & c_{\beta_1}^z & \dots & c_{\alpha_m}^z & c_{\beta_m}^z \end{pmatrix} \quad (2.43)$$

### 2.2.4. Acción de control

Podemos escribir la acción de control del sistema *quadrotor*-manipulador como:

$$\tau = \begin{pmatrix} -\lambda R_0 \vec{e}_3 \\ \tau_{\phi_y} \\ \tau_{\phi_p} \\ \tau_{\phi_r} \\ \tau_{\alpha_1} \\ \tau_{\beta_1} \\ \vdots \\ \tau_{\alpha_m} \\ \tau_{\beta_m} \end{pmatrix} \in \mathfrak{R}^n \quad (2.44)$$

donde  $\lambda \in \mathfrak{R}$  es la fuerza propulsiva,  $R_0(\phi_r, \phi_p, \phi_y)$  es la matriz de rotación de ejes cuerpo del *quadrotor* a ejes inerciales,  $\vec{e}_3 = [0; 0; 1]$  es el vector unidad en la dirección D del sistema de referencia ejes cuerpo del *quadrotor*; y  $\tau_{\phi_y}$ ,  $\tau_{\phi_p}$ ,  $\tau_{\phi_r}$ ,  $\tau_{\alpha_i}$  y  $\tau_{\beta_i} \in \mathfrak{R}$  son respectivamente los momentos de guiñada, cabeceo, alabeo y los momentos de giro de cada uno de los ángulos de las articulaciones.

### 2.2.5. Vector de perturbaciones externas

El vector de perturbaciones externas tiene la siguiente estructura:

$$f = \begin{pmatrix} f_x \\ f_y \\ f_z \\ f_{\phi_y} \\ f_{\phi_p} \\ f_{\phi_r} \\ f_{\alpha_1} \\ f_{\beta_1} \\ \vdots \\ f_{\alpha_m} \\ f_{\beta_m} \end{pmatrix} \in \mathfrak{R}^n \quad (2.45)$$

Su misión es contabilizar todas las posibles perturbaciones existentes en el sistema que se dan en la vida real tales como el viento o posibles funcionamientos deficientes de las articulaciones.

## 2.3. Modelo numérico del sistema *quadrotor*-manipulador con 4 enlaces y efector final

Se ha resuelto numéricamente el problema planteado previamente para el caso de un manipulador con 4 enlaces, es decir,  $m = 4$ . Pasamos a ver como se ha realizado dicho proceso.

### 2.3.1. Cinemática

La configuración del sistema *quadrotor*-manipulador puede ser definida por:

$$q = [p; \phi; \theta] \in \mathfrak{R}^n, \quad n = 16 \quad (2.46)$$

donde  $p = [x; y; z] \in \mathfrak{R}^3$  es la posición del centro de gravedad del *quadrotor* en el sistema de referencia inercial  $(N^0, E^0, D^0)$ ,  $\phi = [\phi_r; \phi_p; \phi_y] \in \mathfrak{R}^3$  son los ángulos de alabeo, cabeceo y guiñada del *quadrotor*; y  $\theta = [\alpha_1; \beta_1; \alpha_2; \beta_2; \alpha_3; \beta_3; \alpha_4; \beta_4; \alpha_E; \beta_E;] \in \mathfrak{R}^{10}$  son los ángulos de las articulaciones del brazo robótico y efector final.

Pasamos a calcular la posición, orientación y velocidad del efector final a partir de las distintas variables usando matrices de cambio de sistema de referencia.

### Cálculo de posición, orientación y velocidad del efector final

Lo primero que vamos a calcular es la matriz de rotación del sistema de referencia ejes cuerpo del *quadrotor* al sistema de referencia inercial, para ello usamos la expresión (2.6) obteniendo:

$$T_B^0 = \begin{pmatrix} c_p c_y & c_y s_p s_r - c_r s_y & s_r s_y + c_r c_y s_p & x \\ c_p s_y & c_r c_y + s_p s_r s_y & c_r s_p s_y - c_y s_r & y \\ -s_p & c_p s_r & c_p c_r & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.47)$$

Tras esto pasamos a calcular la matriz de cambio de sistema de referencia ejes cuerpo del *quadrotor* a el sistema de referencia del primero de los enlaces para posteriormente ir calculando cada una de las matrices de cambio entre los distintos enlaces, para ello vamos a hacer uso de la expresión (2.11) obteniendo los siguientes resultados:

$$T_1^B = \begin{pmatrix} \cos(\alpha_1)\cos(\beta_1) & -\text{sen}(\alpha_1) & \cos(\alpha_1)\text{sen}(\beta_1) & a_1\cos(\alpha_1)\text{sen}(\beta_1) \\ \text{sen}(\alpha_1)\cos(\beta_1) & \cos(\alpha_1) & \text{sen}(\alpha_1)\text{sen}(\beta_1) & a_1\text{sen}(\alpha_1)\text{sen}(\beta_1) \\ -\text{sen}(\beta_1) & 0 & \cos(\beta_1) & a_1\cos(\beta_1) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.48)$$

$$T_2^1 = \begin{pmatrix} \cos(\alpha_2)\cos(\beta_2) & -\text{sen}(\alpha_2) & \cos(\alpha_2)\text{sen}(\beta_2) & a_2\cos(\alpha_2)\text{sen}(\beta_2) \\ \text{sen}(\alpha_2)\cos(\beta_2) & \cos(\alpha_2) & \text{sen}(\alpha_2)\text{sen}(\beta_2) & a_2\text{sen}(\alpha_2)\text{sen}(\beta_2) \\ -\text{sen}(\beta_2) & 0 & \cos(\beta_2) & a_2\cos(\beta_2) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.49)$$

$$T_3^2 = \begin{pmatrix} \cos(\alpha_3)\cos(\beta_3) & -\text{sen}(\alpha_3) & \cos(\alpha_3)\text{sen}(\beta_3) & a_3\cos(\alpha_3)\text{sen}(\beta_3) \\ \text{sen}(\alpha_3)\cos(\beta_3) & \cos(\alpha_3) & \text{sen}(\alpha_3)\text{sen}(\beta_3) & a_3\text{sen}(\alpha_3)\text{sen}(\beta_3) \\ -\text{sen}(\beta_3) & 0 & \cos(\beta_3) & a_3\cos(\beta_3) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.50)$$

$$T_4^3 = \begin{pmatrix} \cos(\alpha_4)\cos(\beta_4) & -\text{sen}(\alpha_4) & \cos(\alpha_4)\text{sen}(\beta_4) & a_4\cos(\alpha_4)\text{sen}(\beta_4) \\ \text{sen}(\alpha_4)\cos(\beta_4) & \cos(\alpha_4) & \text{sen}(\alpha_4)\text{sen}(\beta_4) & a_4\text{sen}(\alpha_4)\text{sen}(\beta_4) \\ -\text{sen}(\beta_4) & 0 & \cos(\beta_4) & a_4\cos(\beta_4) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.51)$$

Para el caso del efector final tenemos que este tiene un sistema de referencia asociado pero con mismo origen que el sistema de referencia del último elemento, por lo que lo único que varía es la dirección de los ejes, obteniéndose de la expresión (2.12):

$$T_E^4 = \begin{pmatrix} \cos(\alpha_E)\cos(\beta_E) & -\text{sen}(\alpha_E) & \cos(\alpha_E)\text{sen}(\beta_E) & 0 \\ \text{sen}(\alpha_E)\cos(\beta_E) & \cos(\alpha_E) & \text{sen}(\alpha_E)\text{sen}(\beta_E) & 0 \\ -\text{sen}(\beta_E) & 0 & \cos(\beta_E) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.52)$$

Una vez que ya tenemos todas estas matrices es posible calcular la posición y orientación del efector final simplemente haciendo una composición de cambios de sistema de referencia desde el sistema de referencia del efector final hasta el sistema de referencia inercial:

$$T_E^0 = T_B^0 \cdot T_1^B \cdot T_2^1 \cdot T_3^2 \cdot T_4^3 \cdot T_E^4 \quad (2.53)$$

La velocidad del efector final en función de la variación de las distintas variables del problema va a ser calculada a partir del jacobiano de traslación del efector final  $J_{v_E}$ , el cual se calcula a partir de (2.17).

### 2.3.2. Dinámica

#### Matriz de inercias

Se calculan la posición y ángulo girado de los CG de cada uno de los 4 enlaces empleando las ecuaciones (2.21)-(2.27), posteriormente se calculan los jacobianos de traslación y rotación usando las expresiones (2.28) y (2.29). Para esto se ha planteado primeramente en Matlab el problema de forma simbólica, hallando las expresiones que tienen estos jacobianos y generando una función con el comando "matlabFunction" que usaremos en nuestro programa numérico para el cálculo de estos.

Una vez que conocemos los jacobianos se calcula la matriz de inercia con la expresión (2.33).

#### Matriz de Coriolis

Para el cálculo de la matriz de Coriolis a partir de las ecuaciones (2.40) y (2.41) necesitamos hallar previamente las derivadas parciales de cada uno de los términos de la matriz de inercia con respecto a las variables de estado.

Debido al tamaño del problema ha sido imposible generar una función para el cálculo de la matriz de Coriolis de la misma forma que se ha hecho para calcular los jacobianos, por lo que ha tenido que hacerse el cálculo de forma completamente numérica. Para ello hemos elegido un valor  $\delta$  muy pequeño ( $\delta = 10^{-8}$ ) y hemos calculado la matriz de inercia  $M$  dejando todos los valores de estado constantes excepto aquel con respecto al cual vamos a calcular la derivada parcial, que lo vamos a incrementar y disminuir en un valor igual a  $\delta$ , resultando las derivadas parciales:

$$\frac{\partial M}{\partial x} = \frac{M(x + \delta, y, z, \phi_y, \dots, \alpha_E, \beta_E) - M(x - \delta, y, z, \phi_y, \dots, \alpha_E, \beta_E)}{2\delta} \quad (2.54)$$

$$\frac{\partial M}{\partial y} = \frac{M(x, y + \delta, z, \phi_y, \dots, \alpha_E, \beta_E) - M(x, y - \delta, z, \phi_y, \dots, \alpha_E, \beta_E)}{2\delta} \quad (2.55)$$

$$\frac{\partial M}{\partial z} = \frac{M(x, y, z + \delta, \phi_y, \dots, \alpha_E, \beta_E) - M(x, y, z - \delta, \phi_y, \dots, \alpha_E, \beta_E)}{2\delta} \quad (2.56)$$

$$\frac{\partial M}{\partial \phi_y} = \frac{M(x, y, z, \phi_y + \delta, \dots, \alpha_E, \beta_E) - M(x, y, z, \phi_y - \delta, \dots, \alpha_E, \beta_E)}{2\delta} \quad (2.57)$$

$$\frac{\partial M}{\partial \phi_p} = \frac{M(x, y, z, \phi_y, \phi_p + \delta, \dots, \alpha_E, \beta_E) - M(x, y, z, \phi_y, \phi_p - \delta, \dots, \alpha_E, \beta_E)}{2\delta} \quad (2.58)$$

$$\frac{\partial M}{\partial \phi_r} = \frac{M(x, y, z, \dots, \phi_r + \delta, \dots, \alpha_E, \beta_E) - M(x, y, z, \dots, \phi_r - \delta, \dots, \alpha_E, \beta_E)}{2\delta} \quad (2.59)$$

$$\frac{\partial M}{\partial \alpha_1} = \frac{M(x, y, z, \dots, \alpha_1 + \delta, \dots, \alpha_E, \beta_E) - M(x, y, z, \dots, \alpha_1 - \delta, \dots, \alpha_E, \beta_E)}{2\delta} \quad (2.60)$$

$$\frac{\partial M}{\partial \beta_1} = \frac{M(x, y, z, \dots, \beta_1 + \delta, \dots, \alpha_E, \beta_E) - M(x, y, z, \dots, \beta_1 - \delta, \dots, \alpha_E, \beta_E)}{2\delta} \quad (2.61)$$

$$\frac{\partial M}{\partial \alpha_2} = \frac{M(x, y, z, \dots, \alpha_2 + \delta, \dots, \alpha_E, \beta_E) - M(x, y, z, \dots, \alpha_2 - \delta, \dots, \alpha_E, \beta_E)}{2\delta} \quad (2.62)$$

$$\frac{\partial M}{\partial \beta_2} = \frac{M(x, y, z, \dots, \beta_2 + \delta, \dots, \alpha_E, \beta_E) - M(x, y, z, \dots, \beta_2 - \delta, \dots, \alpha_E, \beta_E)}{2\delta} \quad (2.63)$$

$$\frac{\partial M}{\partial \alpha_3} = \frac{M(x, y, z, \dots, \alpha_3 + \delta, \dots, \alpha_E, \beta_E) - M(x, y, z, \dots, \alpha_3 - \delta, \dots, \alpha_E, \beta_E)}{2\delta} \quad (2.64)$$

$$\frac{\partial M}{\partial \beta_3} = \frac{M(x, y, z, \dots, \beta_3 + \delta, \dots, \alpha_E, \beta_E) - M(x, y, z, \dots, \beta_3 - \delta, \dots, \alpha_E, \beta_E)}{2\delta} \quad (2.65)$$

$$\frac{\partial M}{\partial \alpha_4} = \frac{M(x, y, z, \dots, \alpha_4 + \delta, \beta_4, \alpha_E, \beta_E) - M(x, y, z, \dots, \alpha_4 - \delta, \beta_4, \alpha_E, \beta_E)}{2\delta} \quad (2.66)$$

$$\frac{\partial M}{\partial \beta_4} = \frac{M(x, y, z, \dots, \alpha_4, \beta_4 + \delta, \alpha_E, \beta_E) - M(x, y, z, \dots, \alpha_4, \beta_4 - \delta, \alpha_E, \beta_E)}{2\delta} \quad (2.67)$$

$$\frac{\partial M}{\partial \alpha_E} = \frac{M(x, y, z, \dots, \alpha_E + \delta, \beta_E) - M(x, y, z, \dots, \alpha_E - \delta, \beta_E)}{2\delta} \quad (2.68)$$

$$\frac{\partial M}{\partial \beta_E} = \frac{M(x, y, z, \dots, \alpha_E, \beta_E + \delta) - M(x, y, z, \dots, \alpha_E, \beta_E - \delta)}{2\delta} \quad (2.69)$$

### Vector fuerza gravitatoria

El cálculo de las fuerzas gravitatorias se hace a partir de (2.42) donde se ha escogido como valor de la aceleración de la gravedad  $g=9.80665 \frac{m}{s^2}$ .

### Acción de control

En el caso de la acción de control tenemos que destacar que vamos a ser capaz de controlar 14 parámetros en el sistema *quadrotor*-manipulador que son:

- El empuje de cada uno de los 4 rotores del *quadrotor*.
- Los momentos en cada una de las articulaciones que provocan variaciones en los ángulos  $\alpha_j$  y  $\beta_j$ .

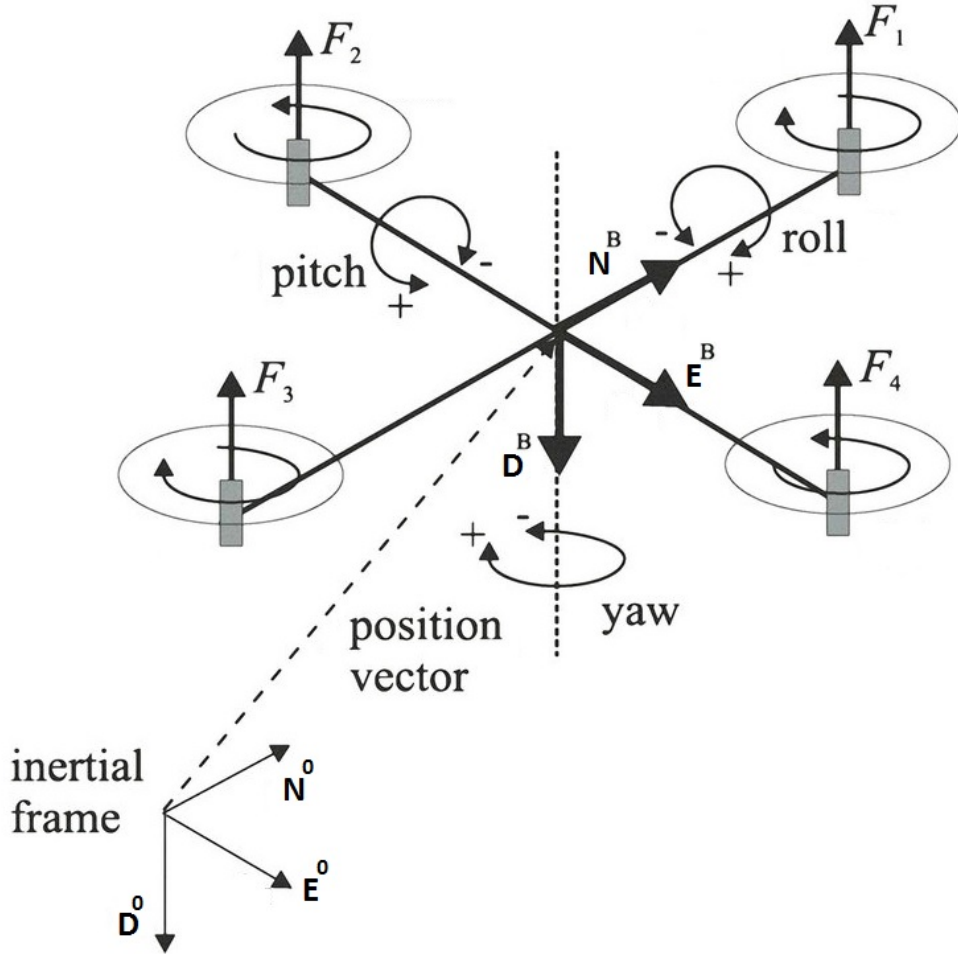


Figura 2.4: Esquema de la definición de cada rotor y su sentido de giro, considerando el rotor número 1 aquel al que apunta el eje  $N^B$  y continuando con la enumeración en sentido antihorario visto el *quadrotor* desde arriba. Observar que los vecinos de cada rotor giran en sentido opuesto a este.

Los momentos en cada una de las articulaciones son directamente componentes del vector  $\tau$  dado por la expresión (2.44) pero falta calcular los valores de  $\lambda$ ,  $\tau_{\phi_y}$ ,  $\tau_{\phi_p}$  y  $\tau_{\phi_r}$  para completar dicho vector, esto lo hacemos a partir del valor del empuje en cada rotor de la siguiente forma:

$$\lambda = F_1 + F_2 + F_3 + F_4 \quad (2.70)$$

$$\tau_{\phi_y} = K (F_2 + F_4 - F_1 - F_3) \quad (2.71)$$

$$\tau_{\phi_p} = l_b (F_1 - F_3) \quad (2.72)$$

$$\tau_{\phi_r} = l_b (F_2 - F_4) \quad (2.73)$$

donde  $K$  es una constante que se tiene que medir de forma experimental y  $l_b$  es el brazo del *quadrotor* (distancia del centro de gravedad del *quadrotor* a cada rotor). Para las simulaciones que se han realizado se ha supuesto un valor de  $K = 1 \cdot 10^{-3}$  y de  $l_b = 0.4$  [m].

### Vector de perturbaciones externas

El vector de perturbaciones externas en este caso sigue teniendo la estructura de (2.45).

## 2.4. Conclusiones alcanzadas y aspectos a tener en cuenta tras la simulación del modelo

Se han realizado varias simulaciones partiendo de la posición de equilibrio con *quadrotor* nivelado con empuje total igual al peso y brazo extendido verticalmente, a partir de esta se ha ido dando valores a cada componente del vector  $\tau$  por separado para así poder ver como influyen cada una de estas en la evolución del estado del sistema *quadrotor*-manipulador.

Los datos de masas, inercias y longitudes de elementos que se han usado para esta simulación son los siguientes:

	Masa [Kg]	Matriz de Inercia [Kg · m <sup>2</sup> ]	Longitud elemento [m]
<i>Quadrotor</i>	5	$4 \cdot 10^{-2} \begin{pmatrix} 0.3 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	-
Primer elemento	0.4	$4 \cdot 10^{-4} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.3 \end{pmatrix}$	0.3
Segundo elemento	0.35	$3 \cdot 10^{-4} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.3 \end{pmatrix}$	0.25
Tercer elemento	0.3	$2 \cdot 10^{-4} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.3 \end{pmatrix}$	0.2
Cuarto elemento	0.25	$2 \cdot 10^{-4} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.3 \end{pmatrix}$	0.15
Efecto final	0.2	$1 \cdot 10^{-5} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{pmatrix}$	0.1

Se ha considerado que el centro de gravedad de cada elemento está situado en el centro de este como primera aproximación, es decir, el vector  $k=[0.5, 0.5, 0.5, 0.5, 0.5]$ . Si quisiéramos suponer que estos últimos se encuentran en otro sitio a lo largo del elemento bastará con cambiar los valores de las componentes del vector  $k$ .

Los valores de tracción dados por cada uno de los rotores son:

	Tracción [N]
Rotor 1	15.9358
Rotor 2	15.9358
Rotor 3	15.9358
Rotor 4	15.9358



que son aquellos cuya suma es igual al peso total y no provocan guiñada, cabeceo o alabeo al ser todos idénticos.

El estado inicial elegido es:

$$q_0 = [0, 0, -3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$\dot{q}_0 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

que hace referencia a que el *quadrotor* se encuentra en reposo a 3 metros de altura con todos los elementos estirados y alineados con la vertical, además todos los sistemas de coordenadas asociados a cada elemento tienen la misma orientación al ser todos los ángulos  $\alpha_j$  nulos.

### 2.4.1. Excitación de $\tau_{\phi_y}$

Primero hemos querido comprobar como evolucionaría el modelo al hacer  $\tau_{\phi_y} \neq 0$ , para ello hemos aumentado en un newton la fuerza aportada por los rotores 1 y 3 a la vez que hemos disminuido esa misma cantidad en los rotores 2 y 4, obteniéndose una guiñada resultante. A continuación se presentan las gráficas con la evolución de las variables de estado que varían a lo largo de los 5 segundos de simulación:

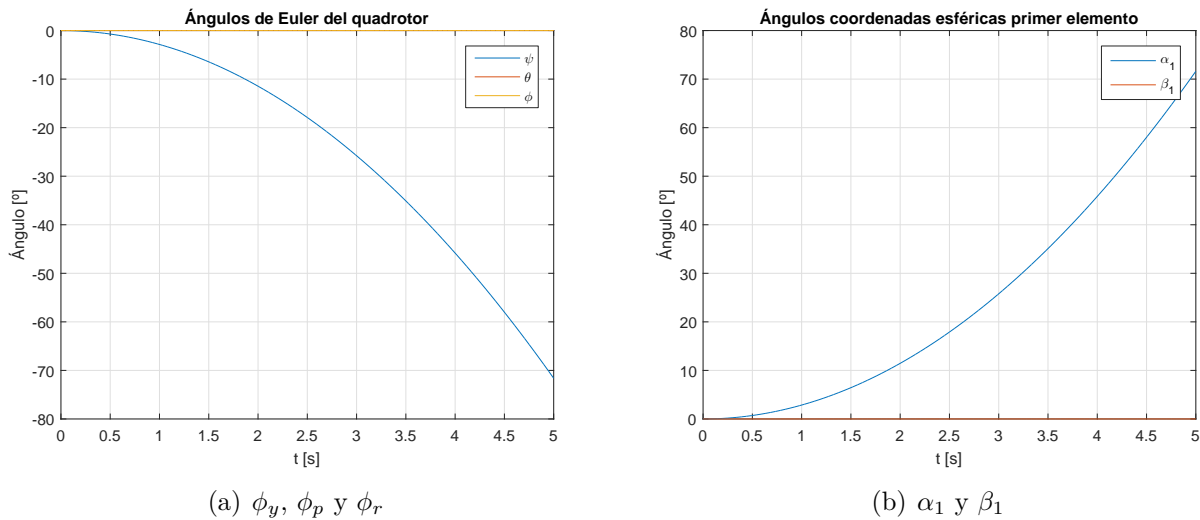
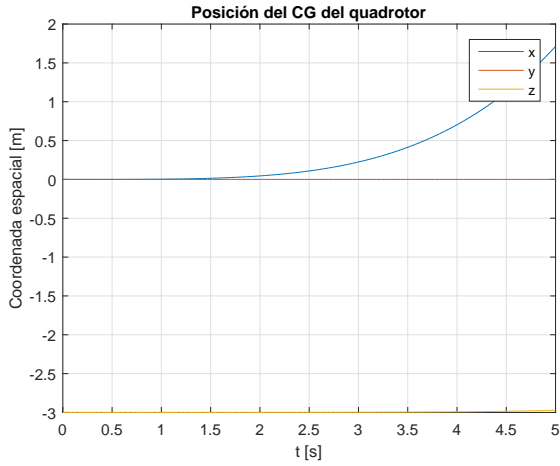


Figura 2.5: Evolución de las variables de estado de interés ante una excitación en guiñada  $\tau_{\phi_y} \neq 0$ .

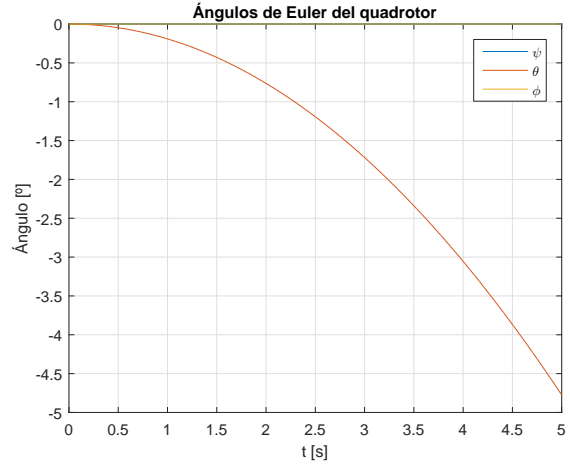
En el caso de provocar una excitación en guiñada tenemos que las únicas variables que varían son el ángulo de guiñada  $\phi_y$  y el ángulo  $\alpha$  del primer elemento  $\alpha_1$ . Vemos que estos varían de la misma forma y en la misma magnitud pero con signos opuestos, esto se traduce en que al ser  $\tau_{\alpha_1}=0$  el primer elemento permite la guiñada del *quadrotor* sin ofrecer resistencia alguna, quedándose el primer elemento exactamente en la misma posición en la que se encontraba al comienzo de la simulación. Para simular un bloqueo de dicha articulación para que el giro de guiñada del *quadrotor* se transmita al primer elemento habría que calcular el  $\tau_{\alpha_1}$  aplicable en cada momento, que equivale al momento que ejerce la articulación al encontrarse bloqueada para evitar el giro relativo con el *quadrotor*.

### 2.4.2. Excitación de $\tau_{\phi_p}$

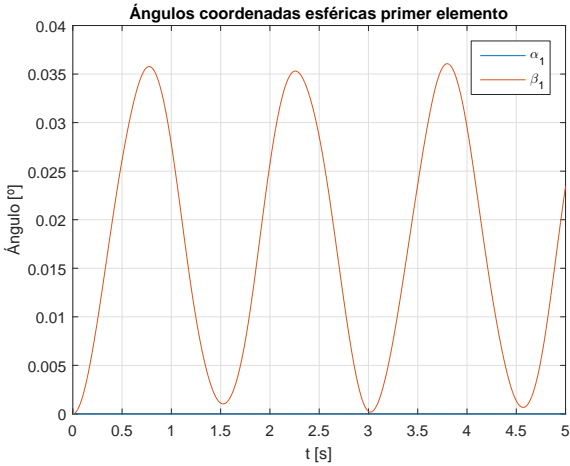
Vamos a ver ahora como evolucionaría el modelo al hacer  $\tau_{\phi_p} \neq 0$ , para ello hemos disminuido en  $1 \cdot 10^{-4}$  la fuerza aportada por el rotor 1 a la vez que hemos aumentado esa misma cantidad en el rotor 3, obteniéndose un cabeceo resultante. A continuación se presentan las gráficas con la evolución de las variables de estado que varían a lo largo de los 5 segundos de simulación:



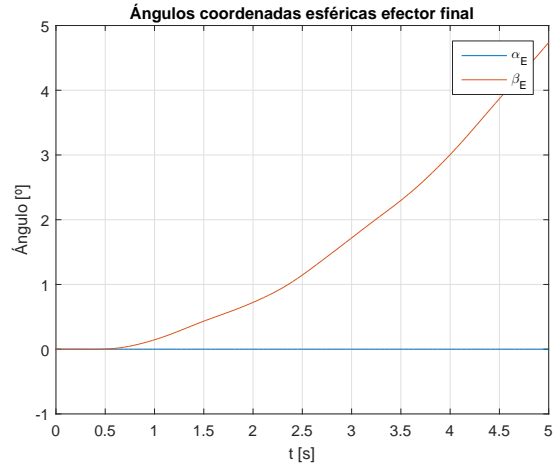
(a)  $x, y$  y  $z$



(b)  $\phi_y, \phi_p$  y  $\phi_r$



(c)  $\alpha_1$  y  $\beta_1$



(d)  $\alpha_E$  y  $\beta_E$

Figura 2.6: Evolución de las variables de estado de interés ante una excitación en cabeceo  $\tau_{\phi_p} \neq 0$ .

En este caso podemos ver que como consecuencia de la excitación en cabeceo el *quadrotor* se inclina hacia delante y se produce un desplazamiento en la dirección  $x$  mientras que se reduce la altitud mínimamente como consecuencia de que la sustentación total ya no está alineada con la vertical y no es capaz de contrarrestar totalmente al peso.

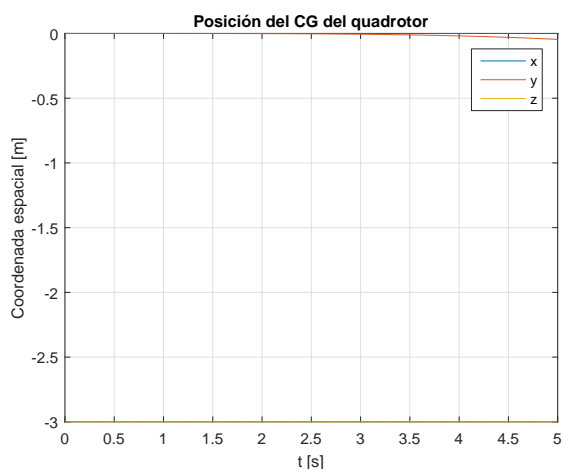
Si observamos la gráfica relativa a los ángulos del primer elemento vemos que en este caso

al contrario que ocurría con la guiñada no se conserva la posición de este a lo largo de la simulación, sino que se produce una oscilación del ángulo  $\beta_1$  en torno a un valor cercano a cero pero diferente de este. Esto se debe al efecto de las fuerzas de inercia debido al movimiento del *quadrotor* pero no se amortigua dicha oscilación ya que tenemos que  $\tau_{\beta_1}=0$ , lo que significa que la articulación no está ejerciendo rozamiento alguno.

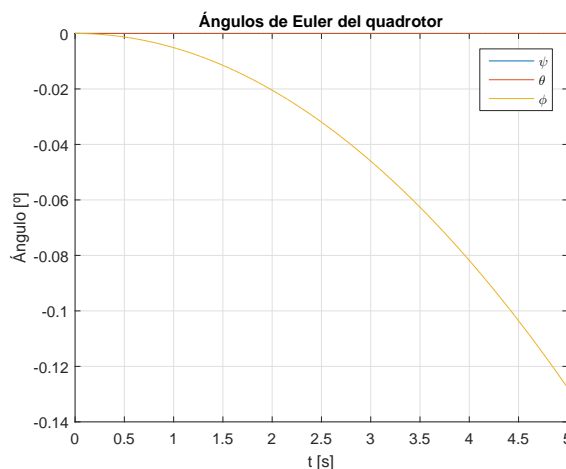
En cuanto al efector final tenemos que, al haberlo considerado como un objeto con masa e inercia pero sin dimensiones efectivas para simplificar el problema, su centro de gravedad está situado en el mismo punto que el eje de rotación y el momento debido a la inercia que hemos comentado que aparecía en el elemento (debido a que el centro de gravedad estaba separado una cierta distancia del eje de rotación) no aparece en este caso, con lo que el efector final tiende a permanecer en su posición inicial que era vertical hacia abajo.

### 2.4.3. Excitación de $\tau_{\phi_r}$

Analicemos ahora como evolucionaría el modelo al hacer  $\tau_{\phi_r} \neq 0$ , para ello hemos disminuido en  $1 \cdot 10^{-4}$  la fuerza aportada por el rotor 2 a la vez que hemos aumentado esa misma cantidad en el rotor 4, obteniéndose un alabeo resultante. A continuación se presentan las gráficas con la evolución de las mismas variables de estado que se han comentado en el caso del cabeceo a lo largo de los 5 segundos de simulación:



(a)  $x$ ,  $y$  y  $z$



(b)  $\phi_y$ ,  $\phi_p$  y  $\phi_r$

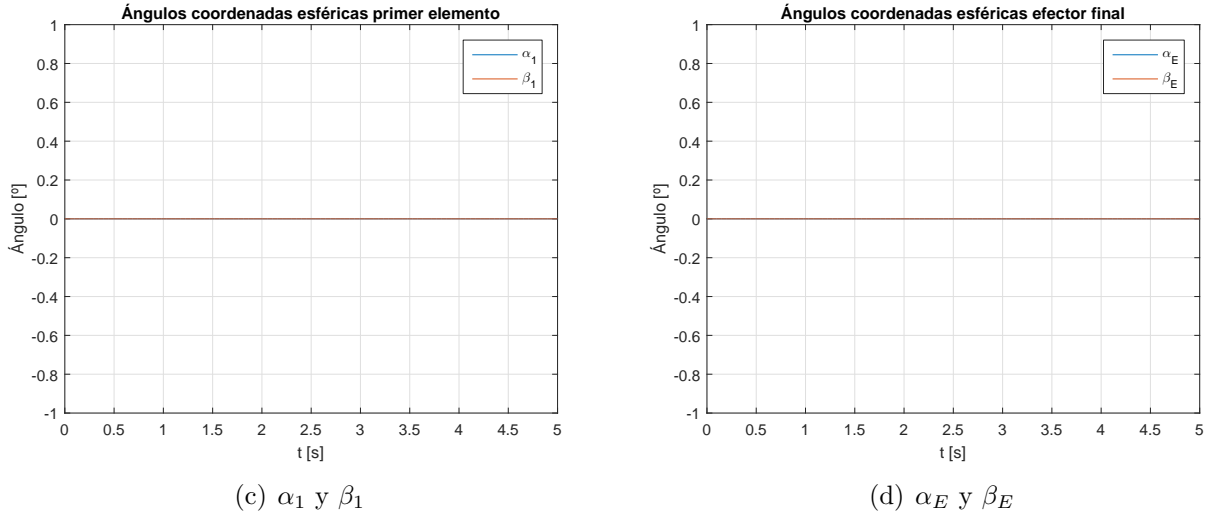


Figura 2.7: Evolución de algunas de las variables de estado ante una excitación en alabeo  $\tau_{\phi_r} \neq 0$ .

Al contrario de como era intuitivo esperar vemos que las tendencias en las gráficas de cabeceo y alabeo no son las mismas pese a que la perturbación aplicada en cada uno de los casos es idéntica.

Si observamos como se ha planteado el modelo podemos llegar a la conclusión rápidamente de que esto último se debe a que, con la condición inicial impuesta, las rotulas de los elementos que proporcionan giro de ángulo  $\beta_j$  están colocadas perpendiculares a la dirección del movimiento en el caso del alabeo, cosa que no ocurría con el cabeceo. El brazo robótico se ha mantenido estirado y perpendicular a la plataforma del *quadrotor* durante toda la simulación debido a que no se ha producido giro alguno en ninguna de sus articulaciones, lo que ha provocado que este se desplazara menos en los 5 segundos ya que el peso del brazo ofrecía resistencia al aumento del ángulo de alabeo.

Para comprobar esto último vamos a volver a ejecutar la simulación pero con la nueva condición inicial:

$$q_0 = [0, 0, -3, 0, 0, 0, -90 \cdot \pi/180, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$\dot{q}_0 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

lo que se traduce en un giro en guiñada del primer elemento de  $90^\circ$ , haciendo que ahora las rotulas de los elementos se encuentren en la posición adecuada, obteniéndose los siguientes resultados:

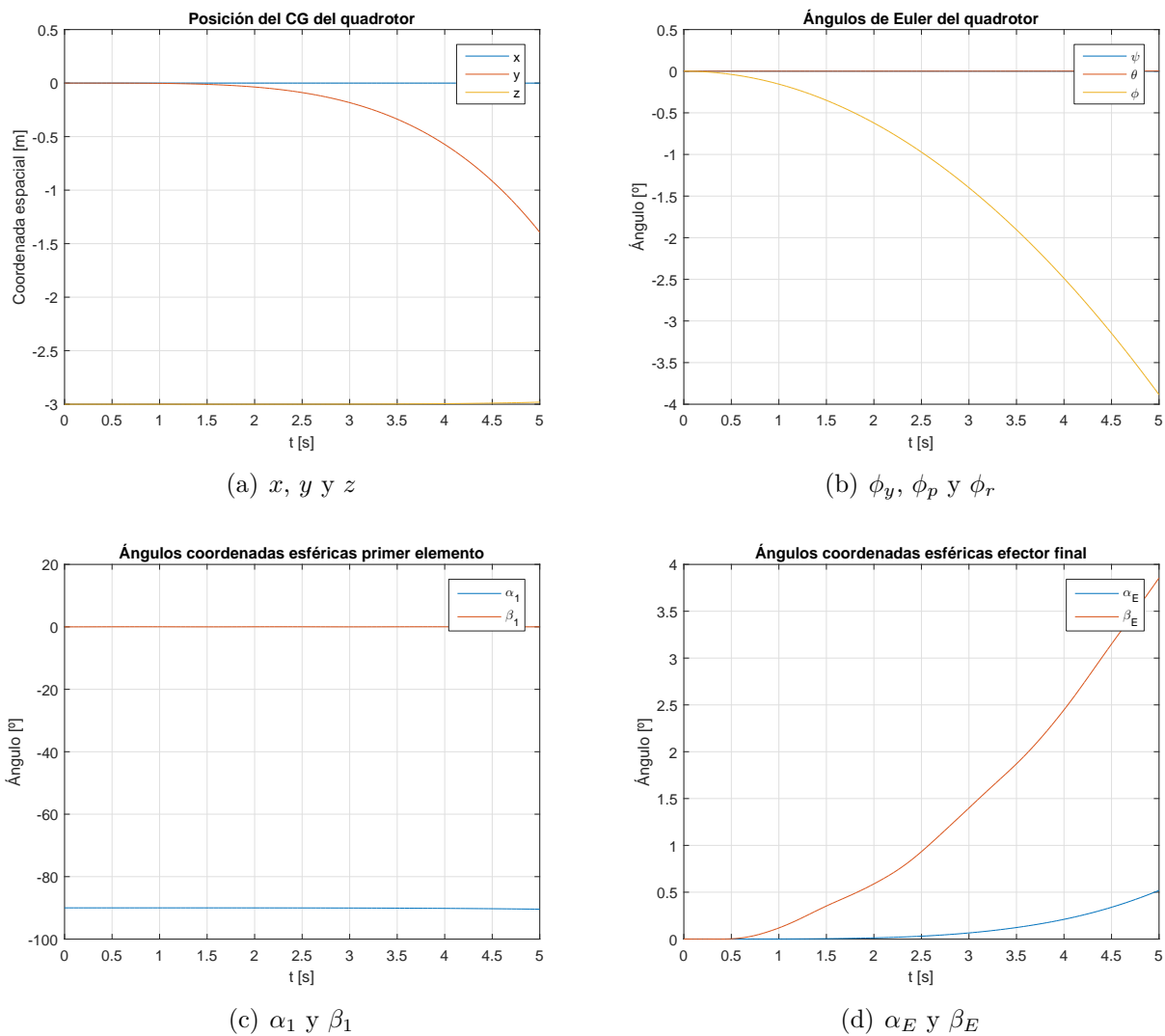


Figura 2.8: Evolución de algunas de las variables de estado ante una excitación en alabeo  $\tau_{\phi_r} \neq 0$  para la nueva condición inicial.

Puede verse que ahora los resultados obtenidos sí que se parecen mucho a los obtenidos en el caso del cabeceo en términos de desplazamiento total y ángulo girado por el *quadrotor*. Comentar que vuelve a producirse la oscilación del ángulo  $\beta_1$  pero esta no es apreciable por el tamaño de los ejes, ya que la amplitud de esta oscilación es muy pequeña.

#### 2.4.4. Excitación de $\tau_{\alpha_1}$

Estudiemos como evolucionaría el modelo al hacer  $\tau_{\alpha_1} \neq 0$ , se ha escogido para ello un valor  $\tau_{\alpha_1} = 1 \cdot 10^{-4}$ . A continuación se presentan las gráficas con la evolución de las variables de estado que varían a lo largo de los 5 segundos de simulación:

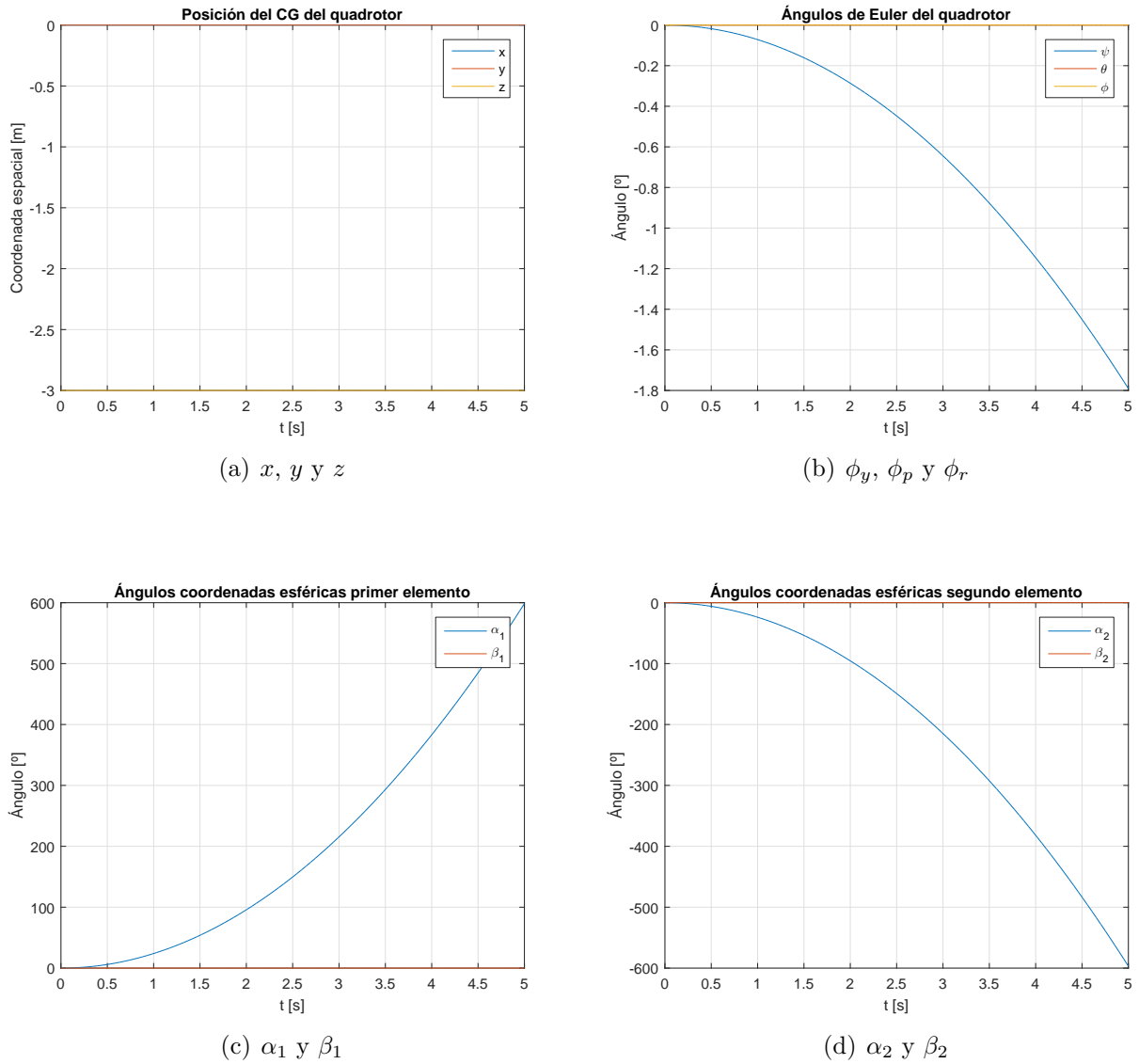


Figura 2.9: Evolución de las variables de estado de interés ante una excitación  $\tau_{\alpha_1} \neq 0$ .

Podemos apreciar que no se produce un cambio en la posición del *quadrotor* y que lo único en lo que afecta esta excitación es en el giro del ángulo  $\alpha_1$  del elemento en cuestión y un giro del elemento inmediatamente anterior, que en este caso al tratarse del *quadrotor* se trata de giro de guiñada  $\phi_y$ . El siguiente elemento vemos como ha permanecido en su posición inicial al verse una evolución exactamente igual pero opuesta.

Si se tratara de una excitación por ejemplo del ángulo  $\alpha_3$  mediante  $\tau_{\alpha_3} \neq 0$  tendríamos que se produciría una variación de  $\alpha_3$  y también del ángulo del elemento inmediatamente anterior  $\alpha_2$  mientras que el resto del conjunto permanece en su posición inicial. Debido a esto no es necesario representar las excitaciones del resto de elementos ya que con estudiar el primero se entiende como se comportarían las excitaciones de los ángulos  $\alpha_j$  en el resto de elementos.

Comentar que si la condición inicial fuese diferente o simplemente el brazo robótico no se encontrara en posición vertical y estuviese orientado en una dirección las variaciones en las

variables de estado serían completamente diferentes. Esto es debido a que, en el caso de brazo colocado en posición vertical, el centro de gravedad de dicho brazo no modifica su posición al variar el ángulo  $\alpha_1$ , pero si este punto ya no se encuentra en la recta vertical que pasa por el centro de gravedad del *quadrotor* sí que se producirían cambios, incluso en la dirección del movimiento global del sistema.

### 2.4.5. Excitación de $\tau_{\beta_1}$

Estudiemos como evolucionaría el modelo al hacer  $\tau_{\beta_1} \neq 0$ , se ha escogido para ello un valor  $\tau_{\beta_1} = 1 \cdot 10^{-4}$ . A continuación se presentan las gráficas con la evolución de las variables de estado que varían a lo largo de los 5 segundos de simulación:

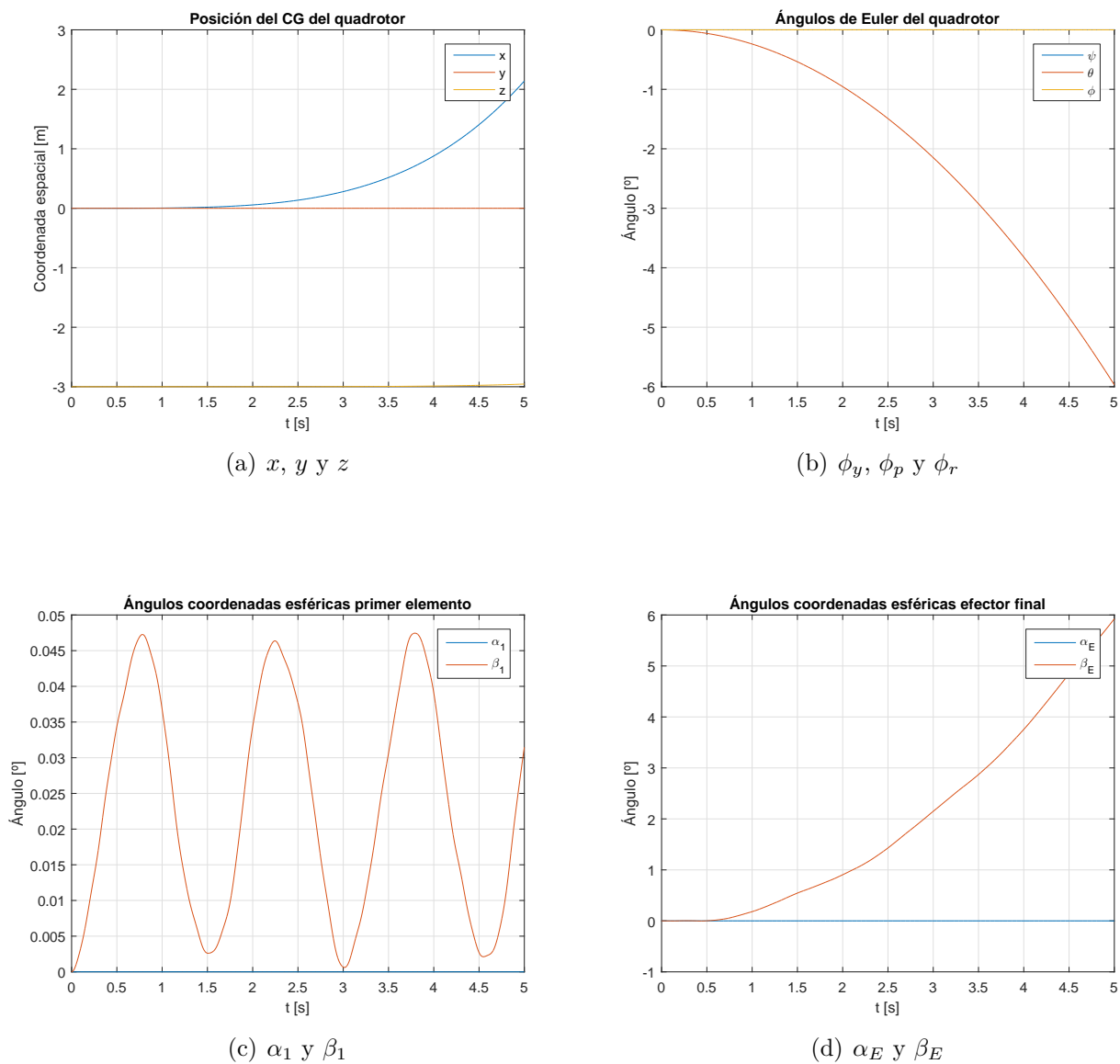


Figura 2.10: Evolución de las variables de estado de interés ante una excitación  $\tau_{\beta_1} \neq 0$ .

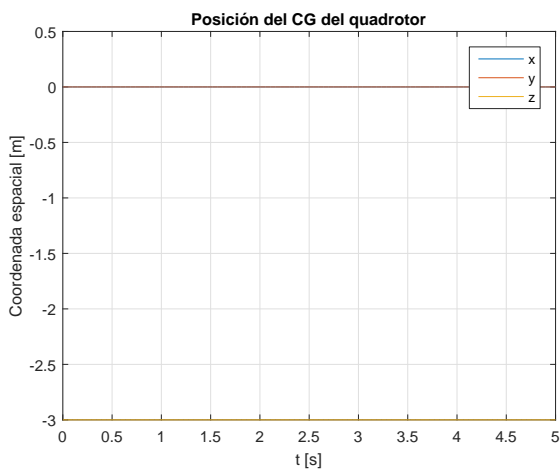
En este caso podemos ver que como consecuencia de la excitación se desalinea la posición del centro de gravedad del brazo con la vertical que pasa por el centro de gravedad del *quadrotor* y este se inclina hacia delante (cabeceo negativo), lo que produce un desplazamiento en la dirección  $x$  mientras que se reduce la altitud mínimamente como consecuencia de que la sustentación total ya no está alineada con la vertical y no es capaz de contrarrestar totalmente al peso. Además esto también es debido en gran medida a que el momento que ejercemos en la articulación para girar el brazo se transmite al *quadrotor* y produce el giro en cabeceo de este, más adelante veremos como este doble efecto debido a la tercera ley de Newton se produce además en el resto de articulaciones pero con resultados diferentes.

Si observamos la gráfica relativa a los ángulos del primer elemento vemos que el ángulo  $\beta_1$  comienza a aumentar debido a la excitación y posteriormente se produce una oscilación en torno a un cierto valor. Esto se debe al efecto de las fuerzas de inercia debido al movimiento del *quadrotor* que hace que el brazo se desplace en la dirección opuesta al movimiento.

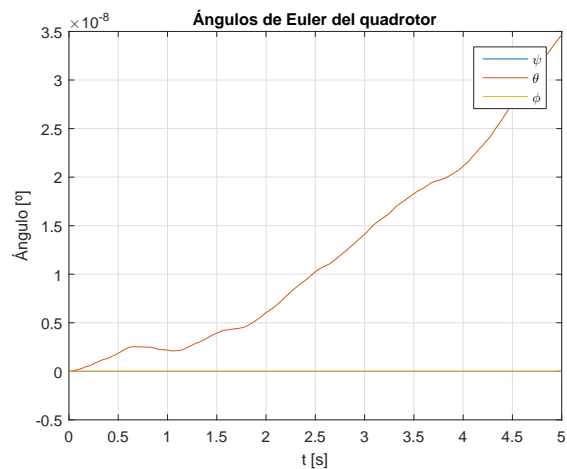
En cuanto al efector final tenemos que, al haberlo considerado como un objeto con masa e inercia pero sin dimensiones efectivas para simplificar el problema, su centro de gravedad está situado en el mismo punto que el eje de rotación y el momento debido a la inercia que hemos comentado que aparecía en el elemento (debido a que el centro de gravedad estaba separado una cierta distancia del eje de rotación) no aparece en este caso, con lo que el efector final tiende a permanecer en su posición inicial que era vertical hacia abajo.

### 2.4.6. Excitación de $\tau_{\beta_2}$

Estudiemos como evolucionaría el modelo al hacer  $\tau_{\beta_2} \neq 0$ , se ha escogido para ello un valor  $\tau_{\beta_2} = 1 \cdot 10^{-2} [N \cdot m]$ . A continuación se presentan las gráficas con la evolución de las variables de estado que varían a lo largo de los 5 segundos de simulación:



(a)  $x, y$  y  $z$



(b)  $\phi_y, \phi_p$  y  $\phi_r$



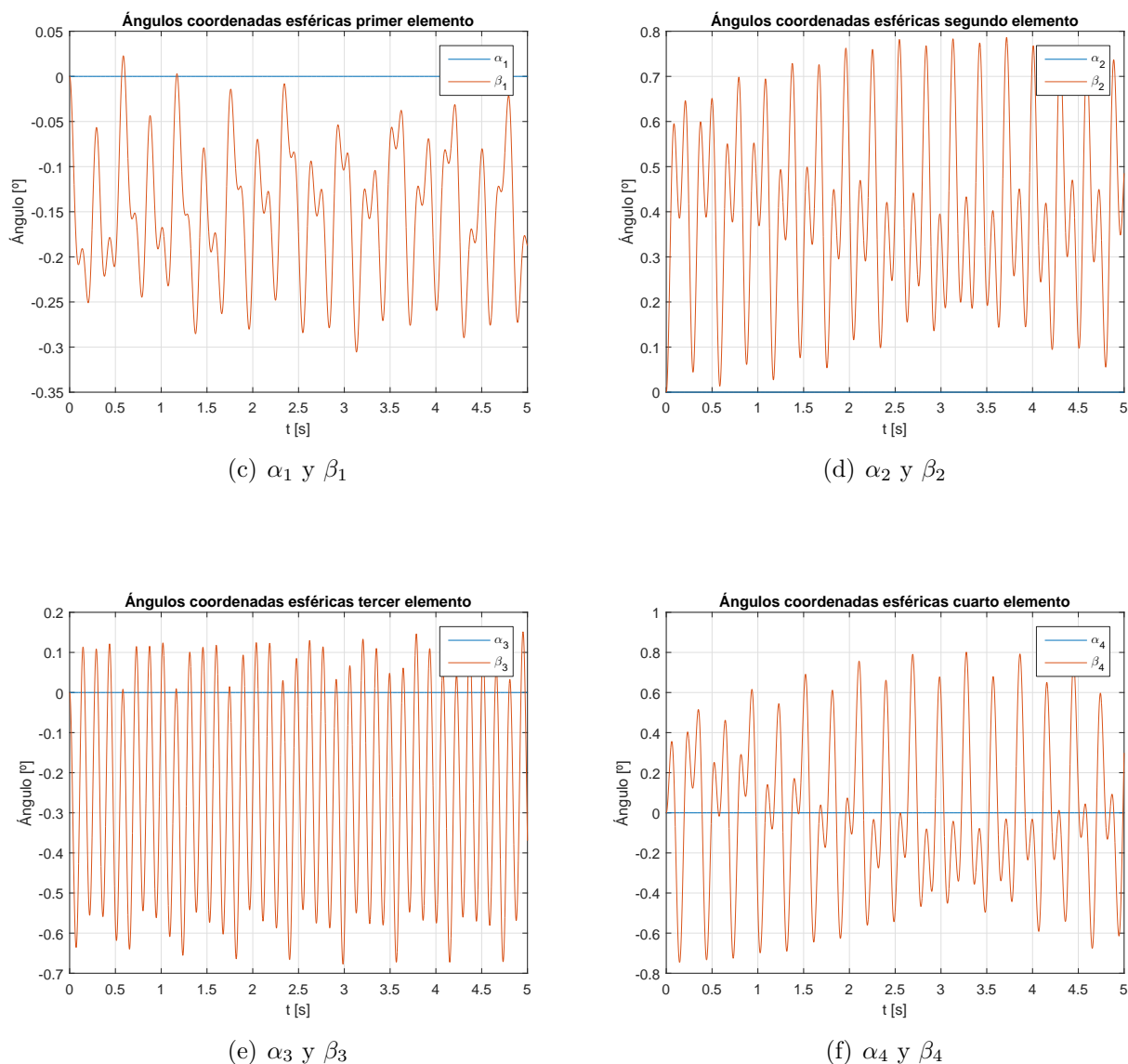


Figura 2.11: Evolución de las variables de estado de interés ante una excitación  $\tau_{\beta_2} \neq 0$ .

En este caso podemos ver que no se ha producido desplazamiento del sistema *quadrotor*-manipulador en ninguna de las direcciones ni en el plano vertical, además se ha mantenido también la posición horizontal del *quadrotor* con ángulos de Euler nulos.

Si observamos las gráficas relativas a los distintos enlaces del brazo podemos comprobar que lo que ha ocurrido es que al hacer  $\tau_{\beta_2} = 1 \cdot 10^{-2} [N \cdot m]$  ha comenzado a aumentar el ángulo  $\beta_2$  mientras que en el primer elemento se ha conseguido el efecto opuesto en  $\beta_1$ , disminuyendo este. Si seguimos viendo las tendencias en los elementos posteriores vemos como la tendencia sigue siendo la misma y se van orientando cada uno hacia un lado de forma alternativa.

Esto hace que se equilibren los pesos a un lado y otro del plano  $E^B D^B$  consiguiendo que el centro de gravedad del manipulador se mantenga prácticamente en este plano y no se produzca perturbación alguna en el *quadrotor* debida a un cambio en el centro de masas global.



# Capítulo 3

## Descomposición de la dinámica del sistema *quadrotor*-manipulador en dos dinámicas desacopladas

Como hemos comentado previamente en este trabajo lo que se pretende es descomponer la dinámica de Lagrange de un sistema *quadrotor*-manipulador, consistente en 6 grados de libertad asociados al *quadrotor* (posición de su centro de gravedad y actitud mediante los ángulos de Euler del mismo) y  $2m + 2$  grados de libertad asociados al manipulador (correspondientes a los ángulos que forman entre sí los distintos  $m$  elementos del manipulador y el efector final), en dos sistemas completamente desacoplados:

- Por un lado tendríamos la dinámica de la traslación del centro de masas en  $E(3)$  [1], que es el punto de masas del sistema sobre el que únicamente afectan a su dinámica la fuerza de empuje de los rotores asociada a ejes cuerpo del *quadrotor* y el efecto de la gravedad.
- Por otro lado estaría la dinámica de rotación interna con  $(3 + 2m + 2)$  grados de libertad, asociados a los ángulos de Euler del *quadrotor* y a la configuración de ángulos del manipulador con efector final, dicha dinámica tiene la forma de la dinámica estándar de Lagrange de los manipuladores robóticos pero sin existir efecto de la gravedad.

Para revelar los fundamentos que yacen en esta nueva estructura de la dinámica del sistema *quadrotor*-manipulador vamos a aplicar la descomposición pasiva [8], [9]. Para ello vamos a definir primero el mapa de coordinación  $h(q) = r$  que son los ángulos de Euler del *quadrotor* y los ángulos de las uniones del brazo robótico. El espacio tangente del sistema *quadrotor*-manipulador, que en geometría diferencial es el conjunto asociado a cada punto de una variedad diferenciable formado por todos los vectores tangentes a dicho punto [10], puede ser dividido en:

$$\Delta^\top = \{\dot{q} \in \mathbb{R}^n \mid \mathcal{L}_{\dot{q}}h(q) = \mathcal{L}_{\dot{q}}r = 0\} = \text{null}(\partial r / \partial q) \quad (3.1)$$

$$\Delta^\perp = \{v \in \mathbb{R}^n \mid v^T M(q)\xi = 0, \forall \xi \in \Delta^\top\} \quad (3.2)$$

donde  $\mathcal{L}_{\dot{q}}h(q)$  es la derivada de Lie [11] de  $h(q)$  a lo largo de  $\dot{q}$ . Esto implica que el espacio tangente del sistema *quadrotor*-manipulador se expresa como:

$$T_q\mathcal{M} = \Delta^\top \oplus \Delta^\perp \quad (3.3)$$

donde:

- $\Delta^\top$  se llama distribución tangencial (por ejemplo, paralela al conjunto de  $h(q)$ ) y la dinámica del sistema *quadrotor*-manipulador proyectada sobre esta distribución se llama dinámica bloqueada del sistema.
- $\Delta^\perp$  se llama distribución normal (por ejemplo, el complemento ortogonal de  $\Delta^\top$  con respecto a la matriz de inercia  $M(r)$ ) y la dinámica del sistema *quadrotor*-manipulador proyectada sobre esta distribución se llama dinámica de la forma del sistema.

Al ser  $\Delta^\top$  la distribución que es paralela a  $h(q) = r$  cualquier cambio del sistema *quadrotor*-manipulador en esta distribución  $\Delta^\top$  tridimensional no debería variar el valor de  $r$ . Esto significa que el cambio de cualquier componente del sistema *quadrotor*-manipulador en  $\Delta^\top$  debería abarcar todos los posibles movimientos como una variación en  $\dot{p}$ .

Por otra parte la distribución  $\Delta^\perp$   $(3+2m+2)$ -dimensional es el complemento ortogonal de  $\Delta^\top$  con respecto a  $M(r)$ , lo que significa que cualquier cambio en  $\Delta^\perp$  debe contener tanto variaciones en  $\dot{r}$  como en  $\dot{p}$ .

Esto significa que, por tanto, podemos escribir la variación del vector de estado del sistema *quadrotor*-manipulador como:

$$\dot{q} = \begin{bmatrix} \Delta_\top & \Delta_\perp \end{bmatrix} \begin{pmatrix} \dot{p}_L \\ \dot{r} \end{pmatrix} = \begin{bmatrix} I_3 & S_E(r) \\ \emptyset & I_{n-3} \end{bmatrix} \begin{pmatrix} \dot{p}_L \\ \dot{r} \end{pmatrix} = S(r)\nu \quad (3.4)$$

donde  $\Delta_\top = [I_3; 0] \in \mathfrak{R}^{n \times 3}$  y  $\Delta_\perp = [S_E(r); I_{n-3}] \in \mathfrak{R}^{n \times (n-3)}$  son matrices que identifican respectivamente a  $\Delta^\top$  y  $\Delta^\perp$ , y  $\dot{p}_L \in \mathfrak{R}^3$  resulta ser la velocidad del centro de masas del sistema *quadrotor*-manipulador. Nótese la diferencia entre la posición del centro de masas del sistema *quadrotor*-manipulador  $p_L$  y el centro de masas de la plataforma del *quadrotor*  $p$ .

De la descomposición pasiva (3.3) tenemos que las distribuciones tangencial y normal son ortogonales entre si con respecto a  $M(q)$  por lo que:

$$\begin{bmatrix} I_3 & \emptyset \end{bmatrix} M(r) \begin{bmatrix} S_E(r) \\ I_{n-3} \end{bmatrix} = M_p S_E(r) + M_{pr} = 0 \quad (3.5)$$

ya al ser  $M_p = m_L I_3$  tenemos la expresión simplificada:

$$S_E(r) = -\frac{M_{pr}}{m_L} \quad (3.6)$$

Ahora que ya conocemos la expresión de la matriz  $S_E(r)$  podemos relacionar las antiguas variables de estado con las nuevas y realizar el cambio de variable para resolver nuestros dos nuevos sistemas desacoplados.

Aplicando la descomposición pasiva en (3.4) transformamos la dinámica del sistema *quadrotor*-manipulador en:

$$m_L \ddot{p}_L + g_L = \tau_L \quad (3.7)$$

$$M_E(r) \ddot{r} + C_E(r, \dot{r}) \dot{r} = \tau_E \quad (3.8)$$

donde:

1. El sistema bloqueado (3.7) describe la dinámca del centro de masas del sistema *quadrotor*-manipulador con  $\dot{p}_L = \dot{p}_{CoM}$ ,  $m_L = \sum_{j=0}^{m+1} m_j$ ,  $g_L = -[0; 0; m_L g] \in \mathfrak{R}^3$  y  $\tau_L = \lambda R_0(\phi)$ , donde  $p_{CoM}$  es la posición del centro de masas del sistema *quadrotor*-manipulador completo.
2. El sistema de la forma (3.8) describe la dinámca de rotación interna de  $r = [\phi; \theta]$  del sistema *quadrotor*-manipulador con actuación total  $\tau_E \in \mathfrak{R}^{3+2m+2}$ , donde  $M_E \in \mathfrak{R}^{(3+2m+2) \times (3+2m+2)}$  es la matriz de inercia simétrica positiva con  $\dot{M}_E - 2C_E$  siendo anti-simétrica.

La clave para la identificación y cálculo de cada uno de los términos que aparecen en las ecuaciones diferenciales (3.7) y (3.8) está en la matriz  $S(r)$  definida en (3.4), ya que tenemos que si reescribimos la dinámca usando  $\dot{q} = S\nu$  (y  $\ddot{q} = S\dot{\nu} + \dot{S}\nu$ ) se llega a las siguientes expresiones:

$$\begin{bmatrix} m_L I_3 & \emptyset \\ \emptyset & M_E \end{bmatrix} = S^T M(r) S \quad (3.9)$$

$$\begin{bmatrix} C_L & C_{LE} \\ C_{EL} & C_E \end{bmatrix} = S^T [M(r)\dot{S} + C(r, \dot{r})S] \quad (3.10)$$

con  $\dot{M}_L - 2C_L$  y  $\dot{M}_E - 2C_E$  ambas antisimétricas, y  $C_{EL} = -C_{LE}^T$ .

Se va a realizar a continuación el desarrollo que nos va a permitir obtener las expresiones de los términos de la ecuación diferencial de la dinámca de rotación interna  $M_E$  y  $C_E$  a partir de (3.9) y (3.10):

$$\begin{aligned} \begin{bmatrix} m_L I_3 & \emptyset \\ \emptyset & M_E \end{bmatrix} &= S^T M(r) S = \begin{bmatrix} I_3 & \emptyset \\ S_E^T & I_{n-3} \end{bmatrix} \cdot \begin{bmatrix} M_p & M_{pr} \\ M_{pr}^T & M_r \end{bmatrix} \cdot \begin{bmatrix} I_3 & S_E \\ \emptyset & I_{n-3} \end{bmatrix} = \\ &= \begin{bmatrix} M_p & M_p S_E + M_{pr} \\ S_E^T M_p + M_{pr}^T & S_E^T M_p S_E + M_{pr}^T S_E + S_E^T M_{pr} + M_r \end{bmatrix} \\ \begin{bmatrix} C_L & C_{LE} \\ C_{EL} & C_E \end{bmatrix} &= S^T [M(r)\dot{S} + C(r, \dot{r})S] = \\ &= \begin{bmatrix} I_3 & \emptyset \\ S_E^T & I_{n-3} \end{bmatrix} \cdot \left( \begin{bmatrix} M_p & M_{pr} \\ M_{pr}^T & M_r \end{bmatrix} \cdot \begin{bmatrix} \emptyset_3 & \dot{S}_E \\ \emptyset & \emptyset_{n-3} \end{bmatrix} + \begin{bmatrix} C_p & C_{pr} \\ C_{pr}^T & C_r \end{bmatrix} \cdot \begin{bmatrix} I_3 & S_E \\ \emptyset & I_{n-3} \end{bmatrix} \right) = \\ &= \begin{bmatrix} C_p & M_p \dot{S}_E + C_p S_E + C_{pr} \\ S_E^T C_p + C_{pr}^T & S_E^T \cdot (M_p \dot{S}_E + C_p S_E + C_{pr}) + M_{pr}^T \dot{S}_E + C_{pr}^T S_E + C_r \end{bmatrix} \end{aligned}$$

Con lo que se concluye que:

$$M_E = S_E^T M_p S_E + M_{pr}^T S_E + S_E^T M_{pr} + M_r \quad (3.11)$$

$$C_E = S_E^T \cdot (M_p \dot{S}_E + C_p S_E + C_{pr}) + M_{pr}^T \dot{S}_E + C_{pr}^T S_E + C_r \quad (3.12)$$

En el caso de la nueva señal de control podemos ver que las relaciones existentes con la del modelo original son:

$$\begin{bmatrix} \tau_L \\ \tau_E \end{bmatrix} = S^T(r)\tau = \begin{bmatrix} I_3 & \emptyset \\ S_E^T & I_{n-3} \end{bmatrix} \cdot \begin{bmatrix} -\lambda R_0 \vec{e}_3 \\ \tau_i \end{bmatrix} = \begin{bmatrix} -\lambda R_0 \vec{e}_3 \\ -\lambda S_E^T R_0 \vec{e}_3 + \tau_i \end{bmatrix} \quad (3.13)$$

donde  $\tau_i$  hace referencia a las componentes del vector de control  $\tau$  que afectan a los grados de libertad de  $r$ . Se observa que en el caso de  $\tau_E$  aparece un nuevo termino  $-\lambda S_E^T R_0 \vec{e}_3$ , el cual es igual a la acción de control necesaria para contrarrestar el efecto de la gravedad en los grados de libertad de  $r$ . Esto último es lo que hace desaparecer el termino de la gravedad en la ecuación que describe la dinámica de rotación interna, ya que intrínsecamente se esta eliminando el efecto que produce esta siempre.

A continuación vamos a ver la relación que existe entre la acción gravitatoria de este modelo con la del modelo original:

$$\begin{bmatrix} g_L \\ g_E \end{bmatrix} = S^T(r)g(q) \quad (3.14)$$

donde  $g_L = \begin{bmatrix} I_3 & \emptyset_{n-3} \end{bmatrix} g(q) = -[0; 0; m_L g]$  y  $g_E = \begin{bmatrix} S_E^T(r) & I_{n-3} \end{bmatrix} g(q) = 0$ .

Esto último se demuestra haciendo uso de la expresión (3.6) y desarrollando matemáticamente de la siguiente forma:

$$M_{pr}(r) = \begin{pmatrix} M_{pr_{1x}}(r) \\ M_{pr_{2x}}(r) \\ M_{pr_{3x}}(r) \end{pmatrix} \quad (3.15)$$

$$g(q) = -g \cdot \begin{pmatrix} 0 \\ 0 \\ m_L \\ \hline M_{pr_{3x}}^T(r) \end{pmatrix} \quad (3.16)$$

$$\begin{aligned} g_E &= \begin{pmatrix} S_E^T(r) & I_{n-3} \end{pmatrix} g(q) = g \cdot \left( -\frac{M_{pr}^T}{m_L} \mid I_{n-3} \right) \cdot \begin{pmatrix} 0 \\ 0 \\ -m_L \\ \hline -M_{pr_{3x}}^T(r) \end{pmatrix} = \\ &= g \cdot \left( M_{pr_{3x}}^T - M_{pr_{3x}}^T \right) = 0 \end{aligned}$$

Con las expresiones (3.9), (3.10), (3.13) y (3.14) podemos calcular los términos  $M_E$ ,  $C_E$ ,  $\tau_L$ ,  $\tau_E$  y  $g_L$  de las ecuaciones diferenciales (3.7) y (3.8) y así resolver estas últimas para una cierta condición inicial.

La nueva estructura aquí planteada de la dinámica del sistema *quadrotor*-manipulador es bastante interesante ya que presenta las siguientes propiedades:

- Es una combinación de la dinámica de la traslación del centro de masas  $p_L$  y de la dinámica de rotación interna  $r$ .

- La dinámica de rotación interna  $r$  es similar a la dinámica estándar de Lagrange de un brazo robótico con enlaces en serie y actuación total.
- El efecto de la gravedad se encuentra presente únicamente en la dinámica de la traslación del centro de masas  $p_L$  y desaparece en la dinámica de rotación interna  $r$ .
- La dinámica de la traslación del centro de masas  $p_L$  y la dinámica de rotación interna  $r$  están completamente desacopladas la una de la otra, no existe acoplamiento en aceleraciones de inercia ni acoplamiento debido a términos de Coriolis.

Esta nueva estructura es universalmente aplicable a muchos sistemas prácticos de vehículos con manipulador, incluyendo satélites con brazo robótico o vehículos submarinos operados remotamente con manipulador robótico.

El desacoplamiento en esta dinámica es una propiedad inherente de la dinámica del sistema, garantizado sin necesidad de una acción de control para que este se produzca. Además esta descomposición facilitará el diseño del control, ya que podemos diseñar leyes de control para el movimiento del centro de masas  $p_L$  y para la variación en la rotación interna  $r$ , de forma separada e individual.

Por último vamos a hacer una recopilación de como han cambiado los términos de las ecuaciones de la dinámica de un modelo a otro:

	Modelo original	Modelo con dinámicas desacopladas
Matriz de inercia	$M = \begin{pmatrix} M_p & M_{pr} \\ M_{pr} & M_r \end{pmatrix}$	$M_E = S_E^T M_P S_E + M_{pr}^T S_E + S_E^T M_{pr} + M_r$
Matriz de Coriolis	$C = \begin{pmatrix} C_p & C_{pr} \\ C_{pr} & C_r \end{pmatrix}$	$C_E = S_E^T \cdot (M_P \dot{S}_E + C_p S_E + C_{pr}) + \dots + M_{pr}^T \dot{S}_E + C_{pr}^T S_E + C_r$
Gravedad	$g(q) = -g \cdot \begin{pmatrix} 0 \\ 0 \\ m_L \\ \frac{M_{pr3x}^T(r)}{M_{pr3x}^T(r)} \end{pmatrix}$	$g_L = - \begin{pmatrix} 0 \\ 0 \\ m_L g \end{pmatrix}$ $g_E = \emptyset$
Acción de control	$\tau = \begin{pmatrix} -\lambda R_0 \vec{e}_3 \\ \tau_i \end{pmatrix}$	$\tau_L = -\lambda R_0 \vec{e}_3$ $\tau_E = -\lambda S_E^T R_0 \vec{e}_3 + \tau_i$

### 3.1. Simulación del modelo con la nueva estructura desacoplada

Se han realizado varias simulaciones partiendo de distintas condiciones iniciales con la finalidad de ilustrar las diferencias existentes con respecto al modelo anterior.

Los datos de masas, inercias y longitudes de elementos que se han usado para estas simulaciones son los siguientes:

	Masa [Kg]	Matriz de Inercia [Kg · m <sup>2</sup> ]	Longitud elemento [m]
<i>Quadrotor</i>	5	$4 \cdot 10^{-2} \begin{pmatrix} 0.3 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	-
Primer elemento	0.4	$4 \cdot 10^{-4} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.3 \end{pmatrix}$	0.3
Segundo elemento	0.35	$3 \cdot 10^{-4} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.3 \end{pmatrix}$	0.25
Tercer elemento	0.3	$2 \cdot 10^{-4} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.3 \end{pmatrix}$	0.2
Cuarto elemento	0.25	$2 \cdot 10^{-4} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.3 \end{pmatrix}$	0.15
Efecto final	0.2	$1 \cdot 10^{-5} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{pmatrix}$	0.1

Se ha considerado que el centro de gravedad de cada elemento está situado en el centro de este como primera aproximación, es decir, el vector  $k=[0.5, 0.5, 0.5, 0.5, 0.5]$ . Si quisiéramos suponer que estos últimos se encuentran en otro sitio a lo largo del elemento bastará con cambiar los valores de las componentes del vector  $k$ .

### 3.1.1. Excitación de $\tau_L$

En esta simulación se pretende demostrar que al excitar solamente  $\tau_L$  permaneciendo  $\tau_E = 0$  no se van a producir variaciones en la rotación interna del sistema, es decir,  $r$  permanece constante.

Los valores de tracción dados por cada uno de los rotores en la simulación son:

	Tracción [N]
Rotor 1	16
Rotor 2	16
Rotor 3	16
Rotor 4	16

que son aquellos cuya suma es ligeramente superior al peso total y no provocan guiñada, cabeceo o alabeo al ser todos idénticos.



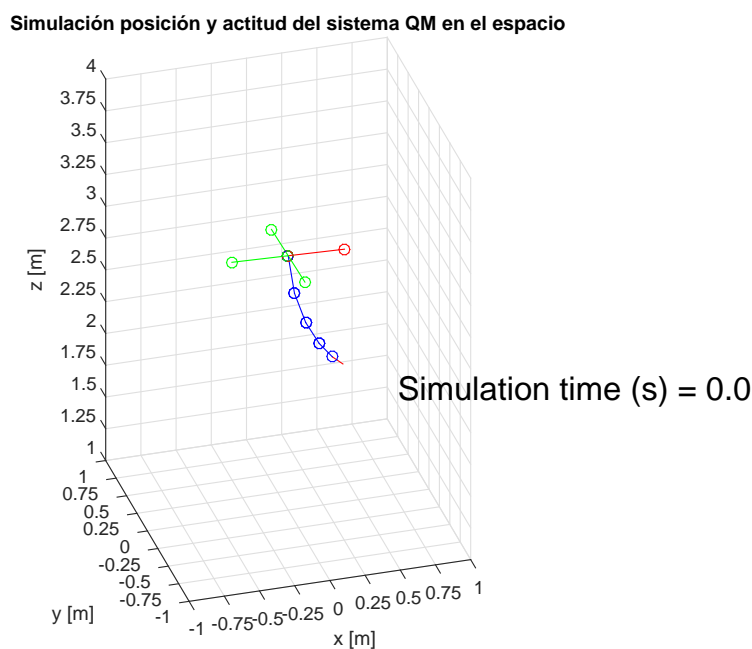


Figura 3.1: Estado inicial del que parte el sistema *quadrotor*-manipulador.

El estado inicial elegido es:

$$q_0 = [0, 0, -3, 0, -2\pi/180, 2\pi/180, 0, 10\pi/180, 0, 10\pi/180, 0, 10\pi/180, 0, 10\pi/180, 0, 10\pi/180]$$

$$\dot{q}_0 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

que hace referencia a que el *quadrotor* se encuentra en reposo a 3 metros de altura con una cierta inclinación de su plataforma de partida y con los elementos del brazo formando cada uno un cierto ángulo con respecto al anterior y orientados hacia delante.

Dicho estado inicial es equivalente a:

$$p_{L_0} = [0.0326, -0.0040, -2.8846]$$

$$p\dot{L}_0 = [0, 0, 0]$$

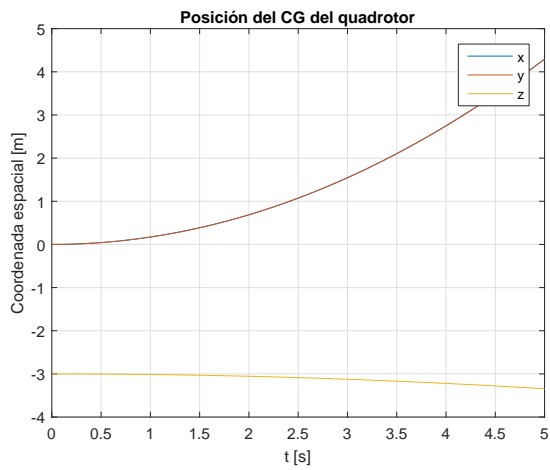
$$r_0 = [0, -2\pi/180, 2\pi/180, 0, 10\pi/180, 0, 10\pi/180, 0, 10\pi/180, 0, 10\pi/180, 0, 10\pi/180]$$

$$r\dot{0} = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

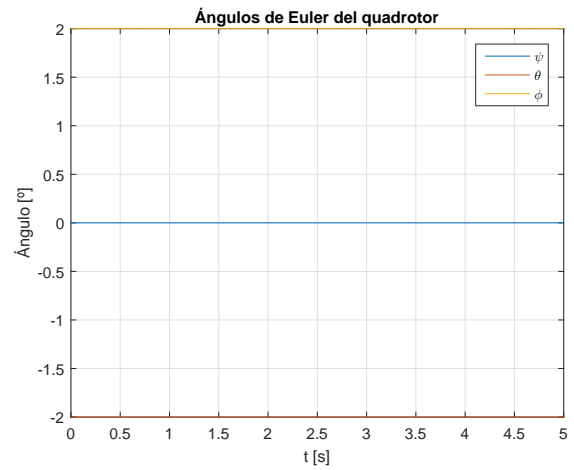
donde se están usando las 2 nuevas variables de estado de nuestro sistema desacoplado.

A continuación se presentan las gráficas con la evolución de las variables de estado que varían a lo largo de los 5 segundos de simulación:

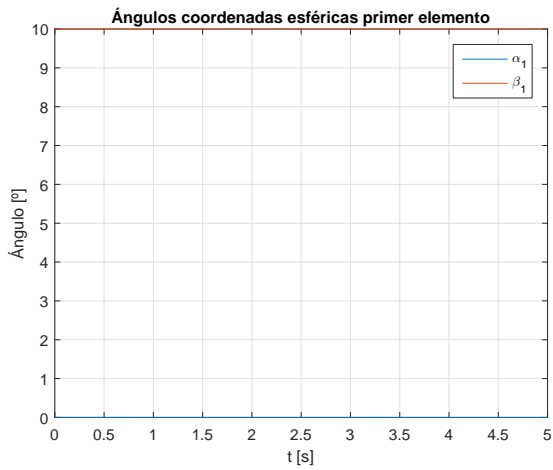
### 3.1 Simulación del modelo con la nueva estructura desacoplada



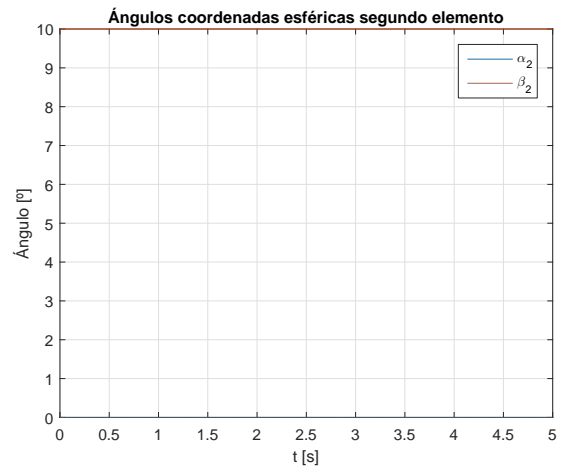
(a)  $x$ ,  $y$  y  $z$



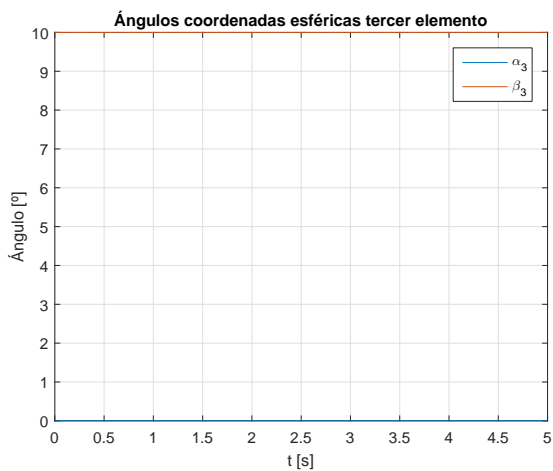
(b)  $\phi_y$ ,  $\phi_p$  y  $\phi_r$



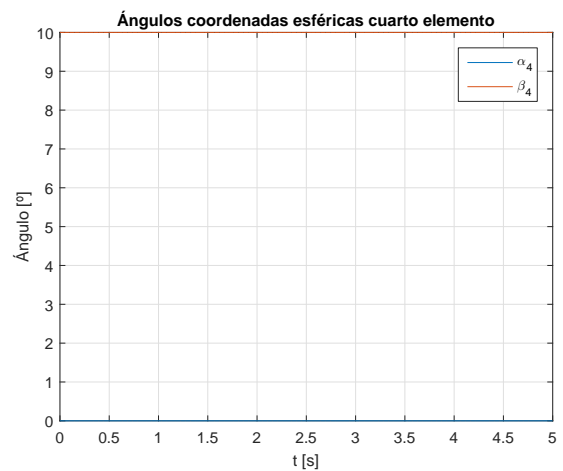
(c)  $\alpha_1$  y  $\beta_1$



(d)  $\alpha_2$  y  $\beta_2$



(e)  $\alpha_3$  y  $\beta_3$



(f)  $\alpha_4$  y  $\beta_4$

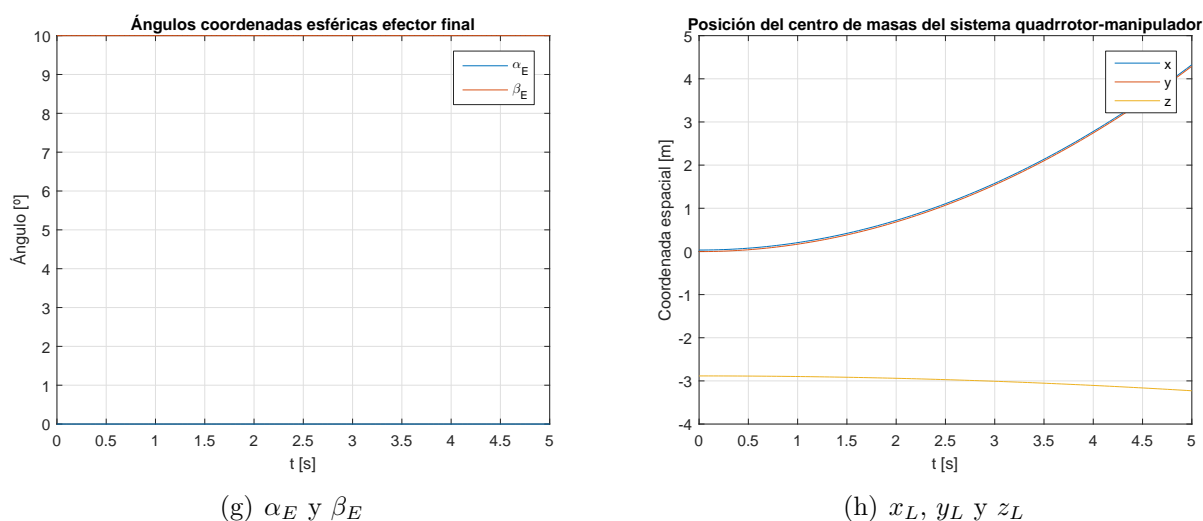


Figura 3.2: Evolución de las variables de estado de interés ante una excitación de  $\tau_L$  con  $\tau_E = 0$ .

Puede observarse que, tal y como se ha comentado previamente, las componentes de  $r$  no varían su valor a lo largo de la simulación ante una excitación en  $\tau_L$  y permanecen constantes y con un valor igual al inicial. Lo único que varía es la posición del *quadrotor* y la del centro de masas del sistema *quadrotor*-manipulador.

También es importante destacar como es cierto que ha desaparecido completamente el efecto de la gravedad en la dinámica de rotación interna  $r$ , ya que no ha sido necesario aplicar valor alguno en  $\tau_E$  de momento en las articulaciones del brazo para que estos permanecieran en su posición inicial y no tendieran a caer y ponerse en posición vertical hacia abajo por efecto de la gravedad.

Otra conclusión importante es que vemos que desaparecen también el acoplamiento entre aceleraciones inerciales y de Coriolis, ya que el sistema *quadrotor*-manipulador se encuentra en movimiento con una cierta aceleración y podemos ver que esto no influye para nada en el valor de las componentes de  $r$  que permanecen constantes a lo largo de toda la simulación, cosa que no ocurría con nuestro primer modelo.

### 3.1.2. Excitación de $\tau_E$

Vamos a ver como varían los parámetros de interés al hacer  $\tau_E \neq 0$  y sobre todo vamos a analizar que ha cambiado en comparación con el modelo anterior. Se ha escogido un valor de  $\tau_E = [0, 0, 0, 0, 10^{-4}, 0, 0, 0, 0, 0, 0, 0] [N \cdot m]$  que es una excitación en el ángulo  $\beta$  del primer elemento del brazo.

Los valores de tracción dados por cada uno de los rotores en la simulación son:

	Tracción [N]
Rotor 1	15.9358
Rotor 2	15.9358
Rotor 3	15.9358
Rotor 4	15.9358

que son aquellos cuya suma es igual al peso total y no provocan guiñada, cabeceo o alabeo al ser todos idénticos.

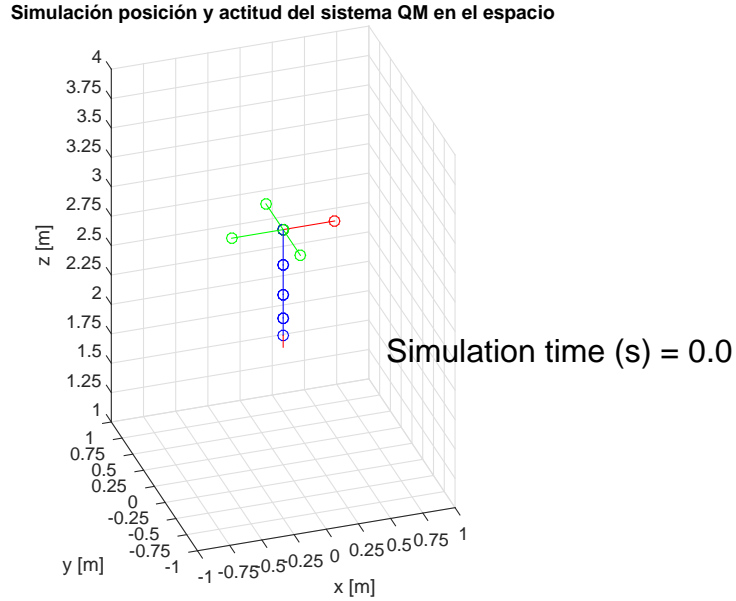


Figura 3.3: Estado inicial del que parte el sistema *quadrotor*-manipulador.

El estado inicial elegido es:

$$q_0 = [0, 0, -3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$\dot{q}_0 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

que hace referencia a que el *quadrotor* se encuentra en reposo a 3 metros de altura con su plataforma en posición horizontal y con los elementos del brazo todos alineados en posición vertical.

Dicho estado inicial es equivalente a:

$$p_{L_0} = [0, 0, -2.8785]$$

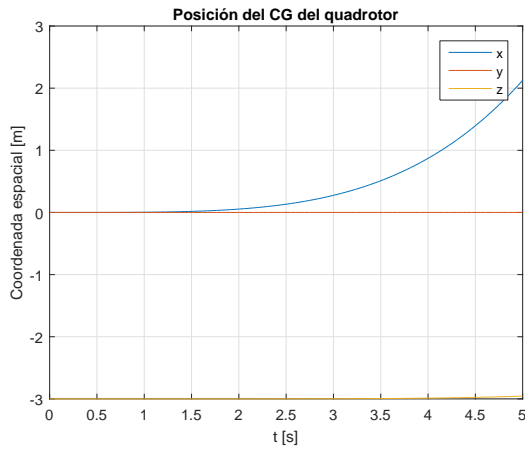
$$p_{\dot{L}_0} = [0, 0, 0]$$

$$r_0 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

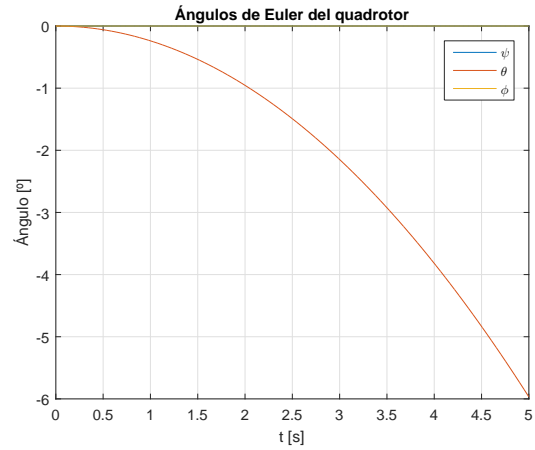
$$\dot{r}_0 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

donde se están usando las 2 nuevas variables de estado de nuestro sistema desacoplado.

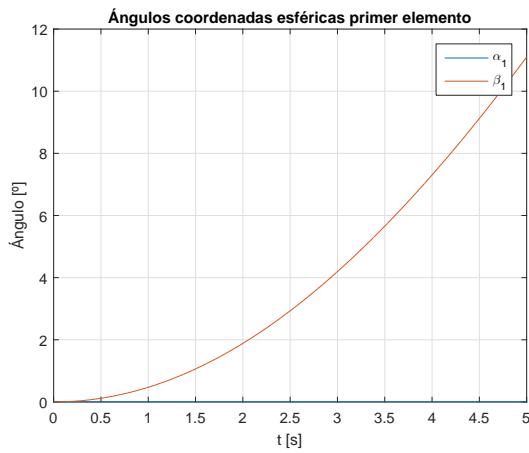
A continuación se presentan las gráficas con la evolución de las variables de estado que varían a lo largo de los 5 segundos de simulación:



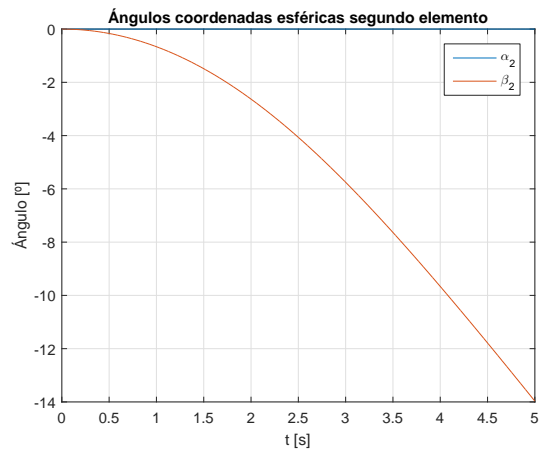
(a)  $x$ ,  $y$  y  $z$



(b)  $\phi_y$ ,  $\phi_p$  y  $\phi_r$

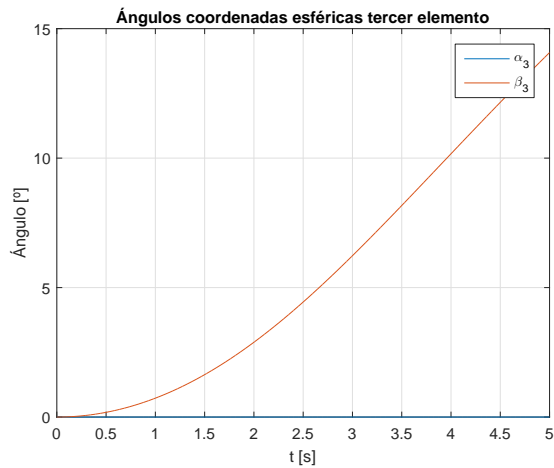


(c)  $\alpha_1$  y  $\beta_1$

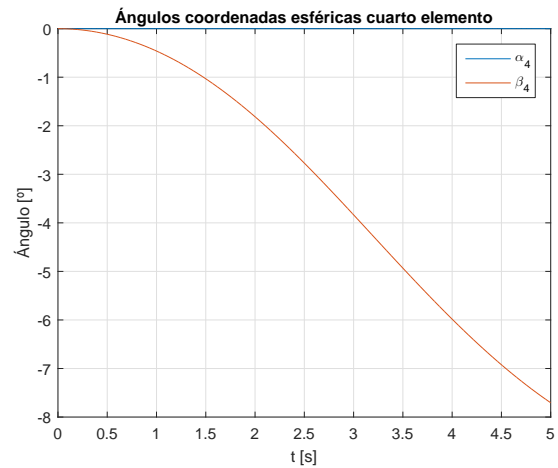


(d)  $\alpha_2$  y  $\beta_2$

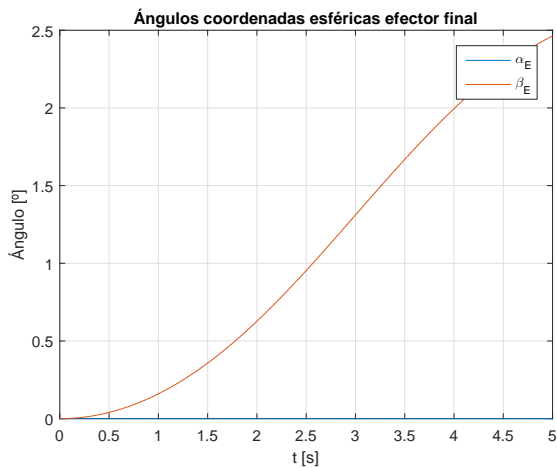
### 3.1 Simulación del modelo con la nueva estructura desacoplada



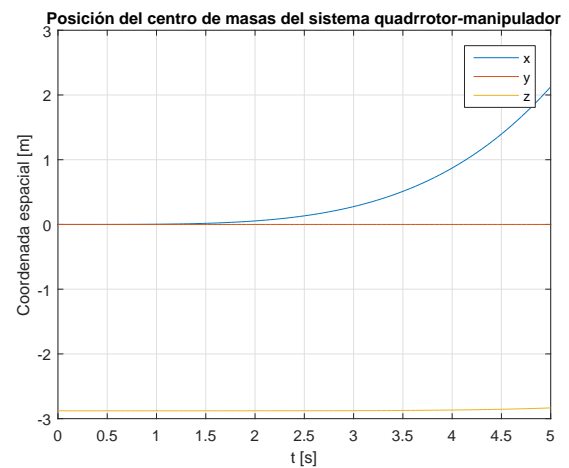
(e)  $\alpha_3$  y  $\beta_3$



(f)  $\alpha_4$  y  $\beta_4$



(g)  $\alpha_E$  y  $\beta_E$



(h)  $x_L$ ,  $y_L$  y  $z_L$

Figura 3.4: Evolución de las variables de estado de interés ante una excitación en la quinta componente de  $\tau_E$ .

Lo primero que destaca a la vista a la hora de observar las gráficas es el hecho de que a pesar de que la excitación se ha llevado a cabo en  $\tau_E$  se han producido variaciones en la posición del centro de masas  $p_L$ , lo cual es un poco contradictorio ya que en [2] se decía que se conseguía un desacople de ambas dinámicas al usar esta nueva descomposición del problema. La razón por la que ocurre esto va a ser fácilmente explicable si representamos el vector  $\tau_L$  a lo largo del tiempo de simulación:

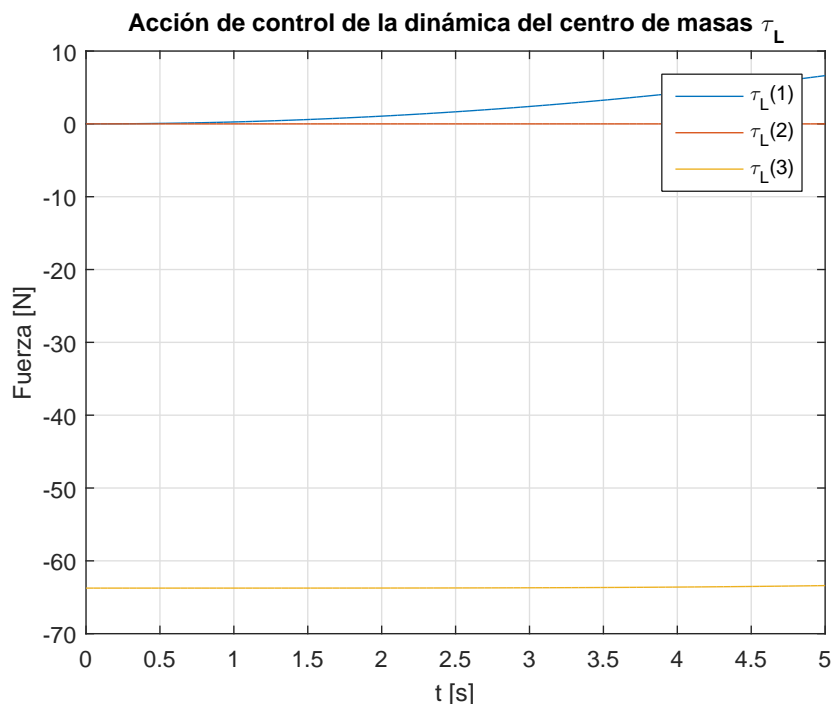


Figura 3.5: Evolución del vector  $\tau_L$  a lo largo de la simulación.

Podemos comprobar que en el instante inicial hemos escogido un valor de  $=[0,0,-63.7432]$  [N], que es aquel que simplemente contrarresta el peso del sistema *quadrotor*-manipulador y hace que este permanezca en reposo. Al avanzar la simulación e ir cambiando  $r$  tenemos que nuestro vector  $\tau_L$  también cambia hasta alcanzar un valor final  $\tau_L=[19.5971,0,-60.6560]$  [N], esto es debido a que  $\tau_L$  no puede tomar cualquier valor, ya que es la fuerza proporcionada por los rotores y el vector tiene que tener la dirección de la fuerza proporcionada por estos, por lo que tiene la siguiente expresión:

$$\tau_L = \lambda R_0(\phi) \quad (3.17)$$

donde  $R_0$  es la matriz de cambio de sistema de coordenadas de ejes cuerpo del *quadrotor* a ejes del sistema de referencia inercial, la cual varía con los ángulos de Euler del *quadrotor* que son las 3 primeras componentes del vector  $r$ .

Por tanto podemos concluir que el desacople no es total, ya que  $r$  está en el vector  $\tau_L$ , pero esto es completamente normal ya que este vector solo puede tomar valores con la dirección del eje  $D^B$  en sentido negativo y por tanto no va a ser posible realizar nunca un desacople total.

Otro detalle importante que se puede ver en las gráficas de la simulación es que vemos que desaparecen las oscilaciones que aparecían en el primer modelo debido al efecto de la gravedad y al acoplamiento entre aceleraciones inerciales y de Coriolis en la rotación interna del sistema, por lo que el control de la posición de los elementos es mucho más sencilla que en el modelo global.





# Capítulo 4

## Modelo Simulink del sistema *quadrotor*-manipulador

En este apartado vamos a presentar y describir la estructura del modelo del sistema *quadrotor*-manipulador realizado en Simulink.

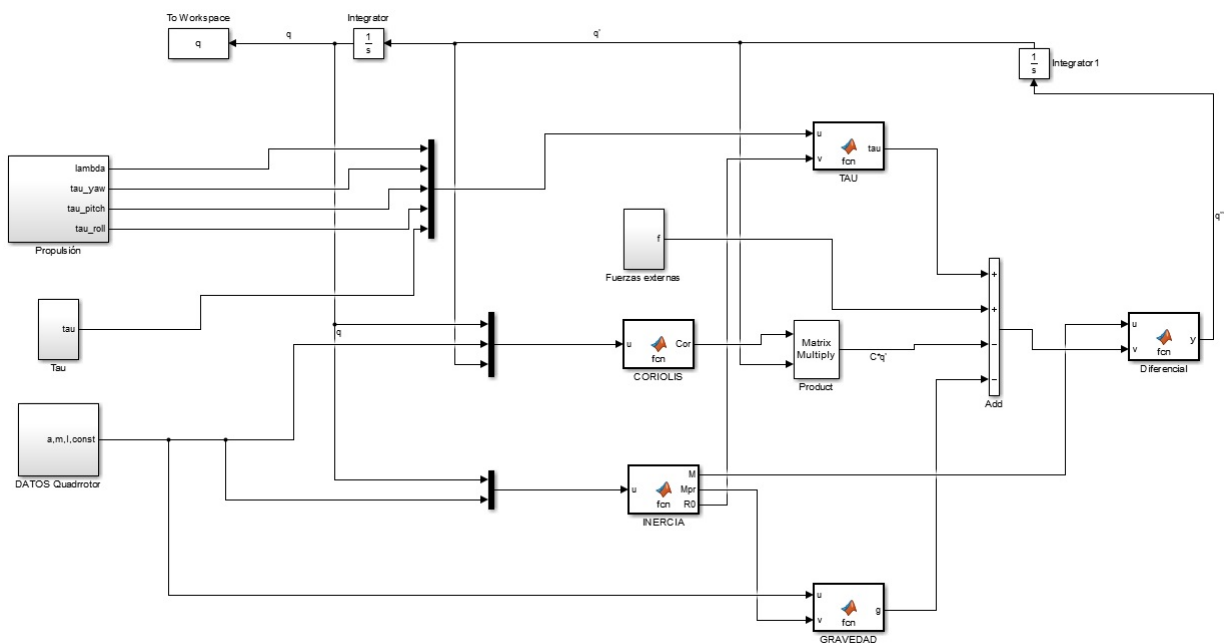


Figura 4.1: Estructura del modelo Simulink del sistema *quadrotor*-manipulador (para poder verla mejor se incluirá también como apéndice).

Vamos a ir recorriendo uno a uno los bloques de la estructura de izquierda a derecha, ya que es de esta forma como se ha planteado la estructura del modelo, encontrándose a la izquierda del todo los datos necesarios para la simulación y ejecutándose los cálculos a medida que nos desplazamos hacia la derecha.

1. **Propulsión:** En este bloque se han dado valores de tracción a cada uno de los rotores

del *quadrotor* y a partir de estos se han calculado valores de interés como son  $\lambda$ ,  $\tau_y$ ,  $\tau_p$  y  $\tau_r$ .

2. **Tau:** Dentro de este bloque se han dado valores al resto de componentes del vector acción de control  $\tau$ , estas componentes son  $\tau_{\alpha_1}$ ,  $\tau_{\beta_1}$ ,  $\tau_{\alpha_2}$ ,  $\tau_{\beta_2}$ ,  $\tau_{\alpha_3}$ ,  $\tau_{\beta_3}$ ,  $\tau_{\alpha_4}$ ,  $\tau_{\beta_4}$ ,  $\tau_{\alpha_E}$  y  $\tau_{\beta_E}$ .
3. **DATOS *Quadrotor*:** Los datos que se introducen en este bloque son las longitudes de los elementos del brazo robótico  $a$ , las masas de *quadrotor* y elementos del brazo robótico  $m$ , el momento de inercia máximo en *quadrotor* y elementos del brazo robótico  $I$  y la posición adimensional del centro de gravedad de cada elemento a lo largo de este  $k$ .
4. **INERCIA:** Esta función recibe como entradas los valores de las componentes del vector de estado  $q$  y los valores de los datos introducidos en "**DATOS *Quadrotor***", a partir de estos realiza el cálculo de la matriz de inercia del sistema con ayuda de una función llamada "jacobianos" que hemos generado usando la programación simbólica de Matlab y nos calcula todos los jacobianos de traslación y rotación del problema (más información en los apéndices). Como salidas de este bloque tenemos la matriz de inercias total  $M$ , la submatriz  $M_{pr}$  que nos va a ser necesaria para el cálculo del vector de fuerzas gravitatorias  $g$  y el conjunto de matrices de rotación de los distintos sistemas de referencia en *quadrotor* y elementos a el sistema de referencia inercial  $R_B^0$ ,  $R_1^0$ ,  $R_2^0$ ,  $R_3^0$ ,  $R_4^0$  y  $R_E^0$ .
5. **CORIOLIS:** Esta función recibe como entradas los valores de las componentes del vector de estado  $q$ , los valores de los datos introducidos en "**DATOS *Quadrotor***" y las derivadas de las componentes del vector de estados  $\dot{q}$ , a partir de estos realiza el cálculo de la matriz de Coriolis mediante una diferenciación numérica centrada de la matriz de inercia con respecto a cada una de las componentes de  $q$  para calcular las derivadas parciales que nos serán necesarias para hallar los símbolos de Christoffel. La única salida de este bloque es la matriz de Coriolis  $C$ .
6. **TAU:** En dicho bloque lo que se hace es agrupar todo el vector acción de control de la forma:

$$\tau = \begin{pmatrix} -\lambda R_0 \vec{e}_3 \\ \tau_{\phi_y} \\ \tau_{\phi_p} \\ \tau_{\phi_r} \\ \tau_{\alpha_1} \\ \tau_{\beta_1} \\ \vdots \\ \tau_{\alpha_m} \\ \tau_{\beta_m} \end{pmatrix} \in \mathfrak{R}^n \quad (4.1)$$

donde  $\lambda \in \mathfrak{R}$  es la fuerza propulsiva,  $R_0(\phi_r, \phi_p, \phi_y)$  es la matriz de rotación de ejes cuerpo del *quadrotor* a ejes inerciales,  $\vec{e}_3 = [0; 0; 1]$  es el vector unidad en la dirección D del sistema de referencia ejes cuerpo del *quadrotor*; y  $\tau_{\phi_y}$ ,  $\tau_{\phi_p}$ ,  $\tau_{\phi_r}$ ,  $\tau_{\alpha_i}$  y  $\tau_{\beta_i} \in \mathfrak{R}$  son respectivamente los momentos de guiñada, cabeceo, alabeo y los momentos de giro de cada uno de los ángulos de las articulaciones.

- 
7. **GRAVEDAD:** En este bloque se va a calcular el vector fuerza gravitatoria, el cual tiene la siguiente expresión :

$$g(q) = -g \begin{pmatrix} 0 \\ 0 \\ m_L \\ c_{\phi_y}^z \\ c_{\phi_p}^z \\ c_{\phi_r}^z \\ c_{\alpha_1}^z \\ c_{\beta_1}^z \\ \vdots \\ c_{\alpha_m}^z \\ c_{\beta_m}^z \end{pmatrix} \quad (4.2)$$

donde  $m_L$  es la masa total del sistema y el resto de términos se obtienen de parte de la matriz de inercia:

$$M_{pr}(r) = \begin{pmatrix} c_{\phi_y}^x & c_{\phi_p}^x & c_{\phi_r}^x & c_{\alpha_1}^x & c_{\beta_1}^x & \dots & c_{\alpha_m}^x & c_{\beta_m}^x \\ c_{\phi_y}^y & c_{\phi_p}^y & c_{\phi_r}^y & c_{\alpha_1}^y & c_{\beta_1}^y & \dots & c_{\alpha_m}^y & c_{\beta_m}^y \\ c_{\phi_y}^z & c_{\phi_p}^z & c_{\phi_r}^z & c_{\alpha_1}^z & c_{\beta_1}^z & \dots & c_{\alpha_m}^z & c_{\beta_m}^z \end{pmatrix} \quad (4.3)$$

8. **Diferencial:** En este bloque lo que se plantea es la resolución de la ecuación diferencial de la dinámica del sistema:

$$M(r)\ddot{q} + C(r, \dot{r})\dot{q} + g(q) = \tau + f \quad (4.4)$$

teniendo como salida de dicho bloque la derivada segunda del vector de estado  $\ddot{q}$ , la cual pasará por dos bloques integradores con sus correspondientes condiciones iniciales para cerrar la estructura del modelo.

Para la representación gráfica de los resultados de la simulación lo único que es necesario hacer es ejecutar la función "graficas" en la ventana de comandos de Matlab tras haber ejecutado la simulación en Simulink.



# Capítulo 5

## Modelo Simulink del sistema *quadrotor*-manipulador con dinámicas desacopladas

En este apartado vamos a presentar y describir la estructura del modelo del sistema *quadrotor*-manipulador con dinámicas desacopladas realizado en Simulink.

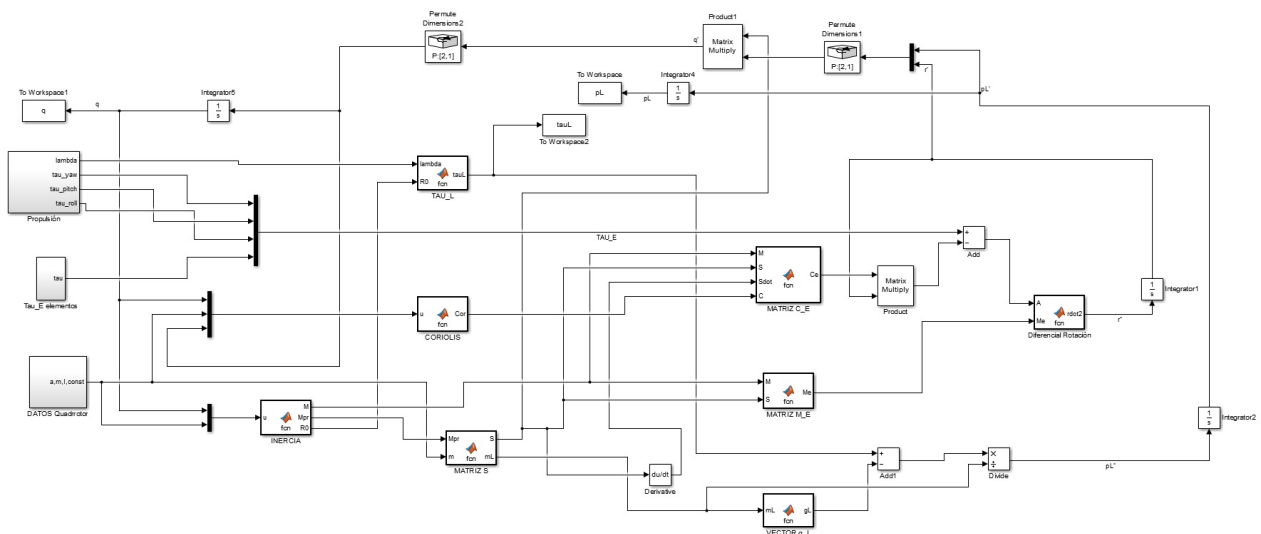


Figura 5.1: Estructura del modelo Simulink del sistema *quadrotor*-manipulador con dinámicas desacopladas (para poder verla mejor se incluirá también como apéndice).

Vamos a ir recorriendo uno a uno los bloques de la estructura de izquierda a derecha, ya que es de esta forma como se ha planteado la estructura del modelo, encontrándose a la izquierda del todo los datos necesarios para la simulación y ejecutándose los cálculos a medida que nos desplazamos hacia la derecha.

1. **Propulsión:** En este bloque se han dado valores de tracción a cada uno de los rotores del *quadrotor* y a partir de estos se han calculado valores de interés como son  $\lambda$ ,  $\tau_y$ ,  $\tau_p$  y  $\tau_r$ .

- 
2. **Tau\_E elementos:** Dentro de este bloque se han dado valores al resto de las componentes del vector acción de control asociado a la rotación interna del sistema  $\tau_E$ , estas componentes son  $\tau_{\alpha_1}, \tau_{\beta_1}, \tau_{\alpha_2}, \tau_{\beta_2}, \tau_{\alpha_3}, \tau_{\beta_3}, \tau_{\alpha_4}, \tau_{\beta_4}, \tau_{\alpha_E}$  y  $\tau_{\beta_E}$ .
  3. **DATOS Quadrotor:** Los datos que se introducen en este bloque son las longitudes de los elementos del brazo robótico  $a$ , las masas de *quadrotor* y elementos del brazo robótico  $m$ , el momento de inercia máximo en *quadrotor* y elementos del brazo robótico  $I$  y la posición adimensional del centro de gravedad de cada elemento a lo largo de este  $k$ .
  4. **INERCIA:** Esta función recibe como entradas los valores de las componentes del vector de estado  $q$  y los valores de los datos introducidos en "DATOS *Quadrotor*", a partir de estos realiza el cálculo de la matriz de inercia del sistema con ayuda de una función llamada "jacobianos" que hemos generado usando la programación simbólica de Matlab y nos calcula todos los jacobianos de traslación y rotación del problema (más información en los apéndices). Como salidas de este bloque tenemos la matriz de inercias total  $M$ , la submatriz  $M_{pr}$  que nos va a ser necesaria para el cálculo del vector de fuerzas gravitatorias  $g$  y el conjunto de matrices de rotación de los distintos sistemas de referencia en *quadrotor* y elementos a el sistema de referencia inercial  $R_B^0, R_1^0, R_2^0, R_3^0, R_4^0$  y  $R_E^0$ .
  5. **CORIOLIS:** Esta función recibe como entradas los valores de las componentes del vector de estado  $q$ , los valores de los datos introducidos en "DATOS *Quadrotor*" y las derivadas de las componentes del vector de estados  $\dot{q}$ , a partir de estos realiza el cálculo de la matriz de Coriolis mediante una diferenciación numérica centrada de la matriz de inercia con respecto a cada una de las componentes de  $q$  para calcular las derivadas parciales que nos serán necesarias para hallar los símbolos de Christoffel. La única salida de este bloque es la matriz de Coriolis  $C$ .
  6. **TAU\_L:** En dicho bloque lo que se hace es agrupar todo el vector acción de control asociado a la dinámica del centro de masas de la forma:

$$\tau_L = -\lambda R_0 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \in \mathfrak{R}^3 \quad (5.1)$$

donde  $\lambda \in \mathfrak{R}$  es la fuerza propulsiva y  $R_0(\phi_r, \phi_p, \phi_y)$  es la matriz de rotación de ejes cuerpo del *quadrotor* a ejes inerciales.

7. **MATRIZ\_S:** Esta función recibe como entradas los valores de los datos introducidos en "DATOS *Quadrotor*" y la submatriz  $M_{pr}$  de la matriz de inercia, con estos datos se calcula la matriz  $S_E$  de la siguiente forma:

$$S_E(r) = -\frac{M_{pr}}{m_L} \quad (5.2)$$

Una vez obtenida esta matriz se monta la matriz  $S$  que será la salida del bloque junto con la masa total  $m_L$ :

$$S(r) = \begin{pmatrix} I_3 & S_E(r) \\ \emptyset & I_{n-3} \end{pmatrix} \quad (5.3)$$

- 
8. **MATRIZ C\_E:** Esta función recibe como entradas los valores de la matriz de inercia  $M$ , la matriz  $S$  y su derivada  $\dot{S}$ ; y la matriz de Coriolis  $C$ , con ellos se calcula la matriz  $C_E$  necesaria para resolver la ecuación diferencial de la rotación interna del sistema de la siguiente forma:

$$\begin{bmatrix} C_L & C_{LE} \\ C_{EL} & C_E \end{bmatrix} = S^T [M(r)\dot{S} + C(r, \dot{r})S] \quad (5.4)$$

9. **MATRIZ M\_E:** Esta función recibe como entradas los valores de la matriz de inercia  $M$  y la matriz  $S$ , con ellos se calcula la matriz  $M_E$  necesaria para resolver la ecuación diferencial de la rotación interna del sistema de la siguiente forma:

$$\begin{bmatrix} m_L I_3 & \emptyset \\ \emptyset & M_E \end{bmatrix} = S^T M(r) S \quad (5.5)$$

10. **VECTOR g\_L:** En este bloque se va a calcular el vector fuerza gravitatoria, el cual solo afecta a la dinámica del centro de masas del sistema y tiene la siguiente expresión :

$$g_L = -g \begin{pmatrix} 0 \\ 0 \\ m_L \end{pmatrix} \quad (5.6)$$

11. **Diferencial Rotación:** En este bloque se resuelve la ecuación diferencial de la dinámica de rotación interna del sistema *quadrotor*-manipulador:

$$M_E(r)\ddot{r} + C_E(r, \dot{r})\dot{r} = \tau_E \quad (5.7)$$

teniendo como salida de dicho bloque la derivada segunda del vector de rotación interna  $\ddot{r}$ , la cual pasará por un bloque integrador con sus correspondientes condiciones iniciales y posteriormente se formará la derivada del vector de estado  $\dot{q}$  a partir de ambas dinámicas mediante la expresión:

$$\dot{q} = \begin{bmatrix} \Delta_{\top} & \Delta_{\perp} \end{bmatrix} \begin{pmatrix} \dot{p}_L \\ \dot{r} \end{pmatrix} = \begin{bmatrix} I_3 & S_E(r) \\ \emptyset & I_{n-3} \end{bmatrix} \begin{pmatrix} \dot{p}_L \\ \dot{r} \end{pmatrix} \quad (5.8)$$

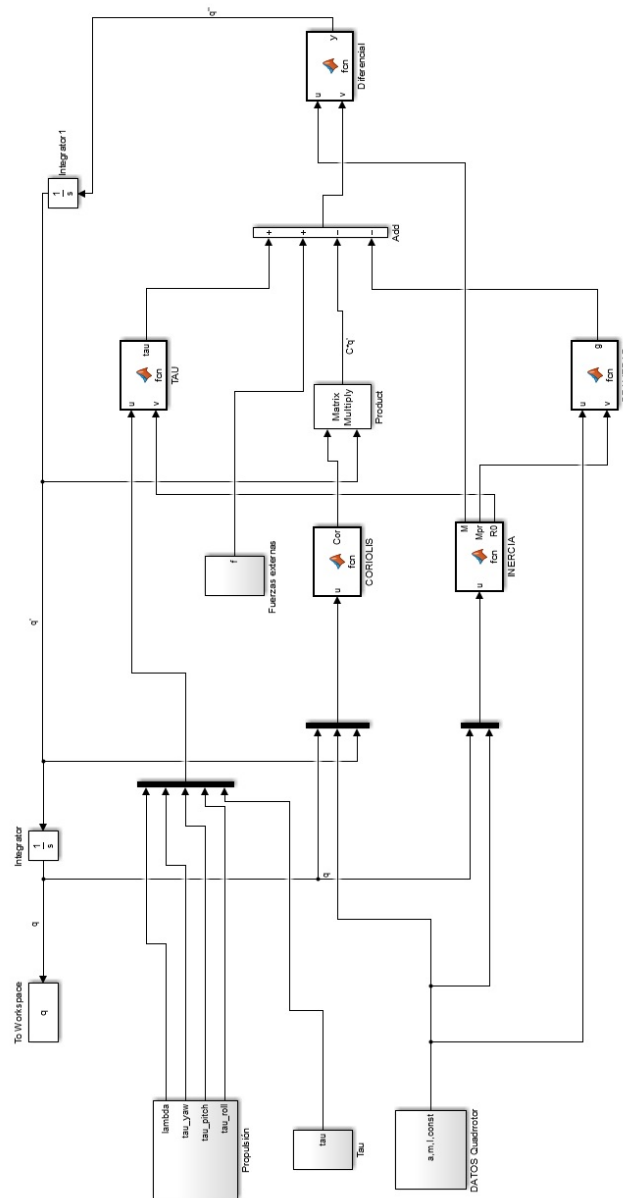
Para la representación gráfica de los resultados de la simulación lo único que es necesario hacer es ejecutar la función "graficas" en la ventana de comandos de Matlab tras haber ejecutado la simulación en Simulink.





# Apéndice A

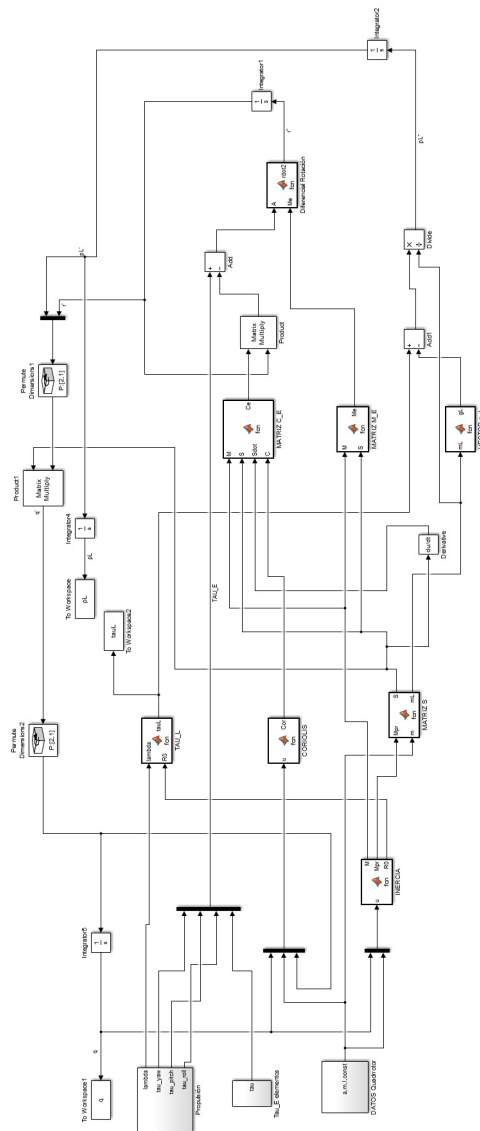
## Estructura modelo Simulink





# Apéndice B

## Estructura modelo Simulink con dinámicas desacopladas





# Apéndice C

## Código de las distintas funciones usadas en el modelo

### C.1. Función "jacobianos"

Para generar dicha función hemos usado el siguiente código con variables simbólicas que lo que hace es calcular las expresiones que tienen los jacobianos en función de los distintos parámetros y generar una función optimizada para su posterior cálculo numérico:

```
1 function generador_expresiones
2 clear all; close all; clc;
3 syms xs ys zs yaws pitches rolls alphals betalS alpha2s beta2s alpha3s ...
    beta3s alpha4s beta4s alphaEs betaEs a1 a2 a3 a4 m0 m1 m2 m3 m4 I0 ...
    I1 I2 I3 I4 const1 const2 const3 const4
4 q=[xs ys zs yaws pitches rolls alphals betalS alpha2s beta2s alpha3s ...
    beta3s alpha4s beta4s alphaEs betaEs];
5 %% Cálculo de las matrices de rotación %%
6 A=[cos(q(4)) -sin(q(4)) 0;sin(q(4)) cos(q(4)) 0;0 0 1];
7 B=[cos(q(5)) 0 sin(q(5)); 0 1 0;-sin(q(5)) 0 cos(q(5))];
8 C=[1 0 0;0 cos(q(6)) -sin(q(6));0 sin(q(6)) cos(q(6))];
9 RB0=A*B*C;
10 TB0=[RB0(1,:) q(1);RB0(2,:) q(2);RB0(3,:) q(3);0 0 0 1];
11 D1=[cos(q(7)) -sin(q(7)) 0;sin(q(7)) cos(q(7)) 0;0 0 1];
12 E1=[cos(q(8)) 0 sin(q(8));0 1 0;-sin(q(8)) 0 cos(q(8))];
13 R1B=D1*E1;
14 T1B=[R1B(1,:) a1*cos(q(7))*sin(q(8));R1B(2,:) ...
    a1*sin(q(7))*sin(q(8));R1B(3,:) a1*cos(q(8));0 0 0 1];
15 D2=[cos(q(9)) -sin(q(9)) 0;sin(q(9)) cos(q(9)) 0;0 0 1];
16 E2=[cos(q(10)) 0 sin(q(10));0 1 0;-sin(q(10)) 0 cos(q(10))];
17 R21=D2*E2;
18 T21=[R21(1,:) a2*cos(q(9))*sin(q(10));R21(2,:) ...
    a2*sin(q(9))*sin(q(10));R21(3,:) a2*cos(q(10));0 0 0 1];
19 D3=[cos(q(11)) -sin(q(11)) 0;sin(q(11)) cos(q(11)) 0;0 0 1];
20 E3=[cos(q(12)) 0 sin(q(12));0 1 0;-sin(q(12)) 0 cos(q(12))];
21 R32=D3*E3;
22 T32=[R32(1,:) a3*cos(q(11))*sin(q(12));R32(2,:) ...
    a3*sin(q(11))*sin(q(12));R32(3,:) a3*cos(q(12));0 0 0 1];
23 D4=[cos(q(13)) -sin(q(13)) 0;sin(q(13)) cos(q(13)) 0;0 0 1];
24 E4=[cos(q(14)) 0 sin(q(14));0 1 0;-sin(q(14)) 0 cos(q(14))];
```

## C.1 Función "jacobianos"

```
25 R43=D4*E4;
26 T43=[R43(1,:) a4*cos(q(13))*sin(q(14));R43(2,:) ...
      a4*sin(q(13))*sin(q(14));R43(3,:) a4*cos(q(14));0 0 0 1];
27 DE=[cos(q(15)) -sin(q(15)) 0;sin(q(15)) cos(q(15)) 0;0 0 1];
28 EE=[cos(q(16)) 0 sin(q(16));0 1 0;-sin(q(16)) 0 cos(q(16))];
29 RE4=DE*EE;
30 TE4=[RE4(1,:) 0;RE4(2,:) 0;RE4(3,:) 0;0 0 0 1];
31 TE0=TB0*T1B*T21*T32*T43*TE4; %Matriz de transformación del sistema de ...
      coordenadas inercial al del efector final.
32 Xe=TE0(1:3,4); %Posición del efector final.
33 %% Cálculo de los Jacobianos %%
34 % Tenemos que calcular primero la posición de los centros de gravedad de
35 % los distintos elementos y ángulo.
36 X0G=[q(1),q(2),q(3)]';
37 T10G=TB0*[T1B(1:4,1:3),[T1B(1:3,4)*const1;1]];
38 X1G=T10G(1:3,4);
39 T20G=TB0*T1B*[T21(1:4,1:3),[T21(1:3,4)*const2;1]];
40 X2G=T20G(1:3,4);
41 T30G=TB0*T1B*T21*[T32(1:4,1:3),[T32(1:3,4)*const3;1]];
42 X3G=T30G(1:3,4);
43 T40G=TB0*T1B*T21*T32*[T43(1:4,1:3),[T43(1:3,4)*const4;1]];
44 X4G=T40G(1:3,4);
45 ang0G=[0,0,q(4)]'+A*[0,q(5),0]'+A*B*[q(6),0,0]';
46 ang1G=ang0G+RB0*[0,0,q(7)]'+RB0*D1*[0,q(8),0]';
47 ang2G=ang1G+RB0*R1B*[0,0,q(9)]'+RB0*R1B*D2*[0,q(10),0]';
48 ang3G=ang2G+RB0*R1B*R21*[0,0,q(11)]'+RB0*R1B*R21*D3*[0,q(12),0]';
49 ang4G=ang3G+RB0*R1B*R21*R32*[0,0,q(13)]'+RB0*R1B*R21*R32*D4*[0,q(14),0]';
50 ange=ang4+RB0*R1B*R21*R32*R43*[0,0,q(15)]'+RB0*R1B*R21*R32*R43*DE*[0 ...
      q(16) 0]';
51 %Pasamos a calcular los jacobianos de velocidad de traslación y velocidad
52 %de rotación [Ecuación (2) del paper].
53 Jv(:, :, 1)=jacobian(X0G,q);
54 Jw(:, :, 1)=jacobian(ang0G,q);
55 Jv(:, :, 2)=jacobian(X1G,q);
56 Jw(:, :, 2)=jacobian(ang1G,q);
57 Jv(:, :, 3)=jacobian(X2G,q);
58 Jw(:, :, 3)=jacobian(ang2G,q);
59 Jv(:, :, 4)=jacobian(X3G,q);
60 Jw(:, :, 4)=jacobian(ang3G,q);
61 Jv(:, :, 5)=jacobian(X4G,q);
62 Jw(:, :, 5)=jacobian(ang4G,q);
63 Jv(:, :, 6)=jacobian(Xe,q);
64 Jw(:, :, 6)=jacobian(ange,q);
65 %
66 f1 = matlabFunction(Jv,Jw,'File','jacobianos','Vars',[q a1 a2 a3 a4 ...
      const1 const2 const3 const4]);
67 end
```

No se va a poder incluir el código de la función generada en el documento debido a su enorme tamaño, muestra de la complejidad del problema que se está tratando, pero se va a facilitar el archivo "jacobianos.m" para que el lector pueda ver la estructura que sigue dicha función generada.

## C.2. Función "INERCIA"

```

1 function [M,Mpr,R0] = fcn(u)
2 %#codegen
3 xs=u(1); % Coordenada x del CG del quadrrrotor [m]
4 ys=u(2); % Coordenada y del CG del quadrrrotor [m]
5 zs=u(3); % Coordenada z del CG del quadrrrotor [m]
6 yaws=u(4); % Ángulo de guiñada del quadrrrotor [rad]
7 pitches=u(5); % Ángulo de cabeceo del quadrrrotor [rad]
8 rolls=u(6); % Ángulo de alabeo del quadrrrotor [rad]
9 alphas=[u(7) u(9) u(11) u(13) u(15)]; % Vector de ángulos alpha de cada ...
    elemento del brazo robótico [rad]
10 betas=[u(8) u(10) u(12) u(14) u(16)]; % Vector de ángulos beta de cada ...
    elemento del brazo robótico [rad]
11 a=[u(17) u(18) u(19) u(20) u(21)]; % Longitud de los elementos del brazo ...
    robótico [m]
12 m=[u(22) u(23) u(24) u(25) u(26) u(27)]; % Masa del quadrrrotor y cada ...
    elemento del brazo robótico [Kg]
13 I=zeros(3,3,6);
14 I(:,:,1)=u(28)*[0.3 0 0;0 0.3 0;0 0 1]; % Matriz de inercia del ...
    quadrrrotor [Kg*m^2]
15 I(:,:,2)=u(29)*[1 0 0;0 1 0;0 0 0.3]; % Matriz de inercia del primer ...
    elemento [Kg*m^2]
16 I(:,:,3)=u(30)*[1 0 0;0 1 0;0 0 0.3]; % Matriz de inercia del segundo ...
    elemento [Kg*m^2]
17 I(:,:,4)=u(31)*[1 0 0;0 1 0;0 0 0.3]; % Matriz de inercia del tercer ...
    elemento [Kg*m^2]
18 I(:,:,5)=u(32)*[1 0 0;0 1 0;0 0 0.3]; % Matriz de inercia del cuarto ...
    elemento [Kg*m^2]
19 I(:,:,6)=u(33)*[1 0 0;0 1 0;0 0 0.5]; % Matriz de inercia del efector ...
    final [Kg*m^2]
20 const=[u(34) u(35) u(36) u(37) u(38)]; % Posición adimensional del CG de ...
    cada elemento a lo largo de este [-]
21 %% Cálculo posición del efector final y actitud %%
22 A1=[cos(yaws) -sin(yaws) 0;sin(yaws) cos(yaws) 0;0 0 1]; %Matriz R_{0'}^{0}
23 A2=[cos(pitches) 0 sin(pitches); 0 1 0;-sin(pitches) 0 cos(pitches)]; %Matriz ...
    R_{0''}^{0'}
24 A3=[1 0 0;0 cos(rolls) -sin(rolls);0 sin(rolls) cos(rolls)]; %Matriz ...
    R_{B}^{0''}
25 R=zeros(3,3,6);
26 T=zeros(4,4,6);
27 R(:,:,1)=A1*A2*A3; %Matriz R_{B}^{0}
28 T(:,:,1)=[R(1,:,1) xs;R(2,:,1) ys;R(3,:,1) zs;0 0 0 1]; %Matriz T_{B}^{0}
29 D=zeros(3,3,6);
30 E=zeros(3,3,6);
31 % Se va a calcular a continuación las matrices R_{j-1'}^{j-1} (D) y
32 % R_{j}^{j-1'} (E) de cada elemento para hallar las matrices ...
    R_{j}^{j-1'} y
33 % T_{j}^{j-1'}.
34 for k=2:6
35 D(:,:,k)=[cos(alphas(k-1)) -sin(alphas(k-1)) 0;sin(alphas(k-1)) ...
    cos(alphas(k-1)) 0;0 0 1];
36 E(:,:,k)=[cos(betas(k-1)) 0 sin(betas(k-1));0 1 0;-sin(betas(k-1)) 0 ...
    cos(betas(k-1))];

```

### C.3 Función "CORIOLIS"

```
37 R(:, :, k) = D(:, :, k) * E(:, :, k);
38 T(:, :, k) = [R(1, :, k) a(k-1) * cos(alphas(k-1)) * sin(betas(k-1)); R(2, :, k) ...
    a(k-1) * sin(alphas(k-1)) * sin(betas(k-1)); R(3, :, k) ...
    a(k-1) * cos(betas(k-1)); 0 0 0 1];
39 end
40 T0 = zeros(4, 4, 6);
41 R0 = zeros(3, 3, 6);
42 T0(:, :, 1) = T(:, :, 1); %Matriz de cambio de sistema de referencia ejes ...
    cuerpo al inercial
43 R0(:, :, 1) = R(:, :, 1); %Matriz de rotación de los ejes cuerpo a los inerciales.
44 for k = 2:6
45     T0(:, :, k) = T0(:, :, k-1) * T(:, :, k); %Matriz de cambio de los sistemas de ...
        ref de los elementos al inercial.
46     R0(:, :, k) = R0(:, :, k-1) * R(:, :, k); %Matriz de rotación de los ejes del ...
        sistema de ref de los elementos a los inerciales.
47 end
48 %% Cálculo de los Jacobianos %%
49 %Pasamos a calcular los jacobianos de velocidad de traslación y velocidad
50 [Jv, Jw] = jacobianos(xs, ys, zs, yaws, pitches, rolls, alphas(1), ...
51 betas(1), alphas(2), betas(2), alphas(3), betas(3), alphas(4), ...
52 betas(4), alphas(5), betas(5), a(1), a(2), a(3), a(4), const(1), ...
53 const(2), const(3), const(4));
54 %% Matriz de Inercias Total %%
55 Mp = zeros(3);
56 Mpr = zeros(3, 13);
57 Mr = zeros(13);
58 M = zeros(16);
59 for k = 1:6
60     aux1 = m(k) * eye(3);
61     Mp = Mp + aux1;
62     aux2 = m(k) * Jv(:, 4:16, k);
63     Mpr = Mpr + aux2;
64     aux3 = m(k) * (Jv(:, 4:16, k)') * Jv(:, 4:16, k) ...
65     + Jw(:, 4:16, k)' * R0(:, :, k) * I(:, :, k) * R0(:, :, k)' * Jw(:, 4:16, k);
66     Mr = Mr + aux3;
67 end
68 M(1:3, 1:3) = Mp;
69 M(1:3, 4:16) = Mpr;
70 M(4:16, 1:3) = Mpr';
71 M(4:16, 4:16) = Mr;
72 end
```

### C.3. Función "CORIOLIS"

```
1 function Cor = fcn(u)
2     %#codegen
3     xs = u(1); % Coordenada x del CG del quadrrrotor [m]
4     ys = u(2); % Coordenada y del CG del quadrrrotor [m]
5     zs = u(3); % Coordenada z del CG del quadrrrotor [m]
6     yaws = u(4); % Ángulo de guiñada del quadrrrotor [rad]
7     pitches = u(5); % Ángulo de cabeceo del quadrrrotor [rad]
8     rolls = u(6); % Ángulo de alabeo del quadrrrotor [rad]
```



```

9  alphas=[u(7) u(9) u(11) u(13) u(15)]; % Vector de ángulos alpha de cada ...
    elemento del brazo robótico [rad]
10 betas=[u(8) u(10) u(12) u(14) u(16)]; % Vector de ángulos beta de cada ...
    elemento del brazo robótico [rad]
11 a=[u(17) u(18) u(19) u(20) u(21)]; % Longitud de los elementos del brazo ...
    robótico [m]
12 m=[u(22) u(23) u(24) u(25) u(26) u(27)]; % Masa del quadrrrotor y cada ...
    elemento del brazo robótico [Kg]
13 I=zeros(3,3,6);
14 I(:,:,1)=u(28)*[0.3 0 0;0 0.3 0;0 0 1]; % Matriz de inercia del ...
    quadrrrotor [Kg*m^{2}]
15 I(:,:,2)=u(29)*[1 0 0;0 1 0;0 0 0.3]; % Matriz de inercia del primer ...
    elemento [Kg*m^{2}]
16 I(:,:,3)=u(30)*[1 0 0;0 1 0;0 0 0.3]; % Matriz de inercia del segundo ...
    elemento [Kg*m^{2}]
17 I(:,:,4)=u(31)*[1 0 0;0 1 0;0 0 0.3]; % Matriz de inercia del tercer ...
    elemento [Kg*m^{2}]
18 I(:,:,5)=u(32)*[1 0 0;0 1 0;0 0 0.3]; % Matriz de inercia del cuarto ...
    elemento [Kg*m^{2}]
19 I(:,:,6)=u(33)*[1 0 0;0 1 0;0 0 0.5]; % Matriz de inercia del efector ...
    final [Kg*m^{2}]
20 xsdot=u(39); % Velocidad en x del quadrrrotor [m/s]
21 ysdot=u(40); % Velocidad en y del quadrrrotor [m/s]
22 zsdot=u(41); % Velocidad en z del quadrrrotor [m/s]
23 yawssdot=u(42); % Velocidad angular en guiñada del quadrrrotor [rad/s]
24 pitchssdot=u(43); % Velocidad angular en cabeceo del quadrrrotor [rad/s]
25 rollssdot=u(44); % Velocidad angular en alabeo del quadrrrotor [rad/s]
26 alphasdot=[u(45) u(47) u(49) u(51) u(53)]; % Velocidad angular en alpha ...
    de cada elemento del brazo robótico [rad/s]
27 betassdot=[u(46) u(48) u(50) u(52) u(54)]; % Velocidad angular en beta de ...
    cada elemento del brazo robótico [rad/s]
28 const=[u(34) u(35) u(36) u(37) u(38)]; % Posición adimensional del CG de ...
    cada elemento a lo largo de este [-]
29 Δ=zeros(16,16,2);
30 Minercia=zeros(16,16,2);
31 dM_dq=zeros(16,16,16);
32 c=zeros(16,16,16);
33 Cor=zeros(16);
34 %Para realizar las derivadas parciales de forma numérica evaluamos la
35 %componente respecto a la cual se va a derivar para un valor superior y uno
36 %inferior pero ambos difiriendo muy muy poco, sumando y restando Δ al
37 %valor en que se evalua.
38 Δ(:,:,1)=-1e-8*eye(16);
39 Δ(:,:,2)=1e-8*eye(16);
40 for j=1:16
41     for i=1:2
42 A=[cos(yaws+Δ(4,j,i)) -sin(yaws+Δ(4,j,i)) 0;sin(yaws+Δ(4,j,i)) ...
        cos(yaws+Δ(4,j,i)) 0;0 0 1]; %Matriz R_{0'}^{0}
43 B=[cos(pitchs+Δ(5,j,i)) 0 sin(pitchs+Δ(5,j,i)); 0 1 ...
        0;-sin(pitchs+Δ(5,j,i)) 0 cos(pitchs+Δ(5,j,i))]; %Matriz R_{0''}^{0'}
44 C=[1 0 0;0 cos(rollss+Δ(6,j,i)) -sin(rollss+Δ(6,j,i));0 ...
        sin(rollss+Δ(6,j,i)) cos(rollss+Δ(6,j,i))]; %Matriz R_{B}^{0''}
45 R=zeros(3,3,6);
46 T=zeros(4,4,6);
47 R(:,:,1)=A*B*C; %Matriz R_{B}^{0}

```

### C.3 Función "CORIOLIS"

```

48 T(:, :, 1)=[R(1, :, 1) xs+Δ(1, j, i);R(2, :, 1) ys+Δ(2, j, i);R(3, :, 1) ...
      zs+Δ(3, j, i);0 0 0 1]; %Matriz T_{B}^{0}
49 D=zeros(3, 3, 6);
50 E=zeros(3, 3, 6);
51 % Se va a calcular a continuación las matrices R_{j-1}^{j-1} (D) y
52 % R_{j}^{j-1} (E) de cada elemento para hallar las matrices ...
      R_{j}^{j-1} y
53 % T_{j}^{j-1}.
54 for k=2:6
55 D(:, :, k)=[cos(alphas(k-1)+Δ(2*k+3, j, i)) -sin(alphas(k-1)+Δ(2*k+3, j, i)) ...
      0;sin(alphas(k-1)+Δ(2*k+3, j, i)) cos(alphas(k-1)+Δ(2*k+3, j, i)) 0;0 0 1];
56 E(:, :, k)=[cos(betas(k-1)+Δ(2*k+4, j, i)) 0 sin(betas(k-1)+Δ(2*k+4, j, i));0 ...
      1 0;-sin(betas(k-1)+Δ(2*k+4, j, i)) 0 cos(betas(k-1)+Δ(2*k+4, j, i))];
57 R(:, :, k)=D(:, :, k)*E(:, :, k);
58 T(:, :, k)=[R(1, :, k) a(k-1)*cos(alphas(k-1)+Δ(2*k+3, j, i))*sin(betas(k-1)...
59 +Δ(2*k+4, j, i));R(2, :, k) a(k-1)*sin(alphas(k-1)+Δ(2*k+3, j, i))...
60 *sin(betas(k-1)+Δ(2*k+4, j, i));R(3, :, k) ...
      a(k-1)*cos(betas(k-1)+Δ(2*k+4, j, i));0 0 0 1];
61 end
62 T0=zeros(4, 4, 6);
63 R0=zeros(3, 3, 6);
64 T0(:, :, 1)=T(:, :, 1); %Matriz de cambio de sistema de referencia ejes ...
      cuerpo al inercial
65 R0(:, :, 1)=R(:, :, 1); %Matriz de rotación de los ejes cuerpo a los inerciales.
66 for k=2:6
67     T0(:, :, k)=T0(:, :, k-1)*T(:, :, k); %Matriz de cambio de los sistemas de ...
      ref de los elementos al inercial.
68     R0(:, :, k)=R0(:, :, k-1)*R(:, :, k); %Matriz de rotación de los ejes del ...
      sistema de ref de los elementos a los inerciales.
69 end
70 %% Cálculo de los jacobianos %%
71 [Jv, Jw] = jacobianos(xs+Δ(1, j, i), ys+Δ(2, j, i), zs+Δ(3, j, i), yaws+Δ(4, j, i), ...
72 pitches+Δ(5, j, i), rolls+Δ(6, j, i), alphas(1)+Δ(7, j, i), betas(1)+Δ(8, j, i), ...
73 alphas(2)+Δ(9, j, i), betas(2)+Δ(10, j, i), alphas(3)+Δ(11, j, i), ...
74 betas(3)+Δ(12, j, i), alphas(4)+Δ(13, j, i), betas(4)+Δ(14, j, i), ...
75 alphas(5)+Δ(15, j, i), betas(5)+Δ(16, j, i), a(1), a(2), a(3), a(4), const(1), ...
76 const(2), const(3), const(4));
77 %% Matriz de Inercias Total %%
78 Mp=zeros(3);
79 Mpr=zeros(3, 13);
80 Mr=zeros(13);
81 M=zeros(16);
82 for k=1:6
83     aux1=m(k)*eye(3);
84     Mp=Mp+aux1;
85     aux2=m(k)*Jv(:, 4:16, k);
86     Mpr=Mpr+aux2;
87     aux3=m(k)*(Jv(:, 4:16, k)')*Jv(:, 4:16, k)...
88     +Jw(:, 4:16, k)')*R0(:, :, k)*I(:, :, k)*R0(:, :, k)')*Jw(:, 4:16, k);
89     Mr=Mr+aux3;
90 end
91 M(1:3, 1:3)=Mp;
92 M(1:3, 4:16)=Mpr;
93 M(4:16, 1:3)=Mpr';
94 M(4:16, 4:16)=Mr;
95 Minercia(:, :, i)=M;

```

```

96     end
97     %% Cálculo de las derivadas parciales %%
98     %%Derivada de M respecto a q(j)
99     dM_dq(:, :, j)=(Minercia(:, :, 2)-Minercia(:, :, 1))/(Δ(1, 1, 2)-Δ(1, 1, 1));
100 end
101 %% Cálculo de la matriz de Coriolis %%
102 for i=1:16
103     for j=1:16
104         for k=1:16
105             %%Simbolos de Christoffel de la matriz de Coriolis
106             c(i, j, k)=0.5*(dM_dq(i, j, k)+dM_dq(i, k, j)-dM_dq(j, k, i));
107         end
108     end
109 end
110 for i=1:16
111     for j=1:16
112         %%Matriz de Coriolis
113 Cor(i, j)=c(i, j, 1)*xsdot+c(i, j, 2)*ysdot+c(i, j, 3)*zsdot+c(i, j, 4)*yawsdot...
114 +c(i, j, 5)*pitchsdot+c(i, j, 6)*rollsdot+c(i, j, 7)*alphasdot(1)...
115 +c(i, j, 8)*betasdot(1)+c(i, j, 9)*alphasdot(2)+c(i, j, 10)*betasdot(2)...
116 +c(i, j, 11)*alphasdot(3)+c(i, j, 12)*betasdot(3)+c(i, j, 13)*alphasdot(4)...
117 +c(i, j, 14)*betasdot(4)+c(i, j, 15)*alphasdot(5)+c(i, j, 16)*betasdot(5);
118     end
119 end

```

## C.4. Función "posicionpuntos"

Devuelve la posición de los 4 rotores del *quadrotor* y de los extremos de cada uno de los elementos para poder realizar una representación de la simulación.

```

1 function [X0,X1,X2,X3,X4,Xe,XN,XS,XE,XO]=posicionpuntos(xs,ys,...
2 zs,yaws,pitches,rolls,alpha1s,beta1s,alpha2s,beta2s,alpha3s,...
3 beta3s,alpha4s,beta4s,alphaEs,betaEs,a1,a2,a3,a4,a5,lt)
4 % Devuelve la posición de los 4 rotores del quadrotor y de los extremos de
5 % cada uno de los elementos para poder realizar una representación de la
6 % simulación.
7 %% Cálculo posición del efector final y actitud %%
8 A1=[cos(yaws) -sin(yaws) 0;sin(yaws) cos(yaws) 0;0 0 1];
9 A2=[cos(pitches) 0 sin(pitches); 0 1 0;-sin(pitches) 0 cos(pitches)];
10 A3=[1 0 0;0 cos(rolls) -sin(rolls);0 sin(rolls) cos(rolls)];
11 RB0=A1*A2*A3;
12 TB0=[RB0(1, :) xs;RB0(2, :) ys;RB0(3, :) zs;0 0 0 1];
13 D1=[cos(alpha1s) -sin(alpha1s) 0;sin(alpha1s) cos(alpha1s) 0;0 0 1];
14 E1=[cos(beta1s) 0 sin(beta1s);0 1 0;-sin(beta1s) 0 cos(beta1s)];
15 R1B=D1*E1;
16 T1B=[R1B(1, :) a1*cos(alpha1s)*sin(beta1s);R1B(2, :) ...
17     a1*sin(alpha1s)*sin(beta1s);R1B(3, :) a1*cos(beta1s);0 0 0 1];
18 D2=[cos(alpha2s) -sin(alpha2s) 0;sin(alpha2s) cos(alpha2s) 0;0 0 1];
19 E2=[cos(beta2s) 0 sin(beta2s);0 1 0;-sin(beta2s) 0 cos(beta2s)];
20 R21=D2*E2;
21 T21=[R21(1, :) a2*cos(alpha2s)*sin(beta2s);R21(2, :) ...
22     a2*sin(alpha2s)*sin(beta2s);R21(3, :) a2*cos(beta2s);0 0 0 1];
23 D3=[cos(alpha3s) -sin(alpha3s) 0;sin(alpha3s) cos(alpha3s) 0;0 0 1];

```

```

22 E3=[cos(beta3s) 0 sin(beta3s);0 1 0;-sin(beta3s) 0 cos(beta3s)];
23 R32=D3*E3;
24 T32=[R32(1,:) a3*cos(alpha3s)*sin(beta3s);R32(2,:) ...
      a3*sin(alpha3s)*sin(beta3s);R32(3,:) a3*cos(beta3s);0 0 0 1];
25 D4=[cos(alpha4s) -sin(alpha4s) 0;sin(alpha4s) cos(alpha4s) 0;0 0 1];
26 E4=[cos(beta4s) 0 sin(beta4s);0 1 0;-sin(beta4s) 0 cos(beta4s)];
27 R43=D4*E4;
28 T43=[R43(1,:) a4*cos(alpha4s)*sin(beta4s);R43(2,:) ...
      a4*sin(alpha4s)*sin(beta4s);R43(3,:) a4*cos(beta4s);0 0 0 1];
29 DE=[cos(alphaEs) -sin(alphaEs) 0;sin(alphaEs) cos(alphaEs) 0;0 0 1];
30 EE=[cos(betaEs) 0 sin(betaEs);0 1 0;-sin(betaEs) 0 cos(betaEs)];
31 RE4=DE*EE;
32 TE4=[RE4(1,:) a5*cos(alphaEs)*sin(betaEs);RE4(2,:) ...
      a5*sin(alphaEs)*sin(betaEs);RE4(3,:) a5*cos(betaEs);0 0 0 1];
33 TE0=TB0*T1B*T21*T32*T43*TE4; %Matriz de transformación del sistema de ...
      coordenadas inercial al del efector final.
34 Xe=TE0(1:3,4); %Posición del efector final.
35 %% Cálculo de los Jacobianos %%
36 % Tenemos que calcular primero la posición de los centros de gravedad de
37 % los distintos elementos y ángulo. (Centros de gravedad en el centro de ...
      cada elemento.)
38 X0=[xs,ys,zs]';
39 T10G=TB0*T1B;
40 X1=T10G(1:3,4);
41 T20G=TB0*T1B*T21;
42 X2=T20G(1:3,4);
43 T30G=TB0*T1B*T21*T32;
44 X3=T30G(1:3,4);
45 T40G=TB0*T1B*T21*T32*T43;
46 X4=T40G(1:3,4);
47 XN=X0+RB0*[lt 0 0]';
48 XS=X0+RB0*[-lt 0 0]';
49 XE=X0+RB0*[0 lt 0]';
50 XO=X0+RB0*[0 -lt 0]';
51 end

```

## C.5. Función "graficas"

Esta función es la que vamos a ejecutar tras finalizar la simulación del modelo en Simulink para ver una animación del sistema *quadrotor*-manipulador moviéndose en el espacio tridimensional y para obtener las gráficas de la evolución de los valores de las distintas variables de estado del problema a lo largo del tiempo.

```

1 function graficas(q,a,lt)
2 close all;clc;
3 t=q.Time;% Vector de tiempos de simulación [s]
4 Q=q.Data;% Componentes del vector de estado en cada instante de tiempo
5 n=length(Q(:,1));
6 a=a.Data(1,:);% Longitudes de los elementos del brazo robótico [m]
7 lt=lt.Data(1);% Distancia del CG del quadrotor a los rotores [m]
8 M=20;% Parámetro que acorta la duración de la representación de la ...
      simulación.

```

```

9 NN=floor(n/M);
10
11 % Gráfica animada de la simulación
12 for lll=1:NN
13     k=lll*M;
14     [X0(:,k),X1(:,k),X2(:,k),X3(:,k),X4(:,k),Xe(:,k),XN(:,k),XS(:,k),XE(:,k),...
15     XO(:,k)]=posicionpuntos(Q(k,1),Q(k,2),Q(k,3),Q(k,4),Q(k,5),Q(k,6),Q(k,7),...
16     Q(k,8),Q(k,9),Q(k,10),Q(k,11),Q(k,12),Q(k,13),Q(k,14),Q(k,15),Q(k,16),...
17     a(1),a(2),a(3),a(4),a(5),lt);
18     figure(8)
19     plot3([X0(1,k) XN(1,k)], [X0(2,k) XN(2,k)],-[X0(3,k) XN(3,k)], 'r-o') % El ...
20         rojo es el eje norte cuerpo
21     axis([-2 2 -2 2 0 4])
22     grid on
23     hold on
24     title('Simulación posición y actitud del sistema QM en el espacio')
25     xlabel('x [m]')
26     ylabel('y [m]')
27     zlabel('z [m]')
28     plot3([X0(1,k) XS(1,k)], [X0(2,k) XS(2,k)],-[X0(3,k) XS(3,k)], 'g-o')
29     plot3([X0(1,k) XE(1,k)], [X0(2,k) XE(2,k)],-[X0(3,k) XE(3,k)], 'g-o')
30     plot3([X0(1,k) XO(1,k)], [X0(2,k) XO(2,k)],-[X0(3,k) XO(3,k)], 'g-o')
31     plot3([X0(1,k) X1(1,k)], [X0(2,k) X1(2,k)],-[X0(3,k) X1(3,k)], 'b-o')
32     plot3([X1(1,k) X2(1,k)], [X1(2,k) X2(2,k)],-[X1(3,k) X2(3,k)], 'b-o')
33     plot3([X2(1,k) X3(1,k)], [X2(2,k) X3(2,k)],-[X2(3,k) X3(3,k)], 'b-o')
34     plot3([X3(1,k) X4(1,k)], [X3(2,k) X4(2,k)],-[X3(3,k) X4(3,k)], 'b-o')
35     plot3([X4(1,k) Xe(1,k)], [X4(2,k) Xe(2,k)],-[X4(3,k) Xe(3,k)], 'r-')
36     hold off
37     daspect([1 1 1]);
38     axis([]);
39     ax = gca;
40     ax.XTick = linspace(-5,5,41);
41     ax.YTick = linspace(-5,5,41);
42     ax.ZTick = linspace(-5,5,41);
43     text(X0(1,k)+0.6,X0(2,k)-0.6,-X0(3,k)-0.8,sprintf('Simulation time (s) = ...
44         %.1f',round(t(k),1)), 'FontSize',18);
45     drawnow;
46     end
47     % Gráficas de la evolución de las variables de estado
48     figure(1)
49     plot(t,Q(:,1:3))
50     grid on
51     title('Posición del CG del quadrotor')
52     xlabel('t [s]')
53     ylabel('Coordenada espacial [m]')
54     legend('x','y','z')
55     figure(2)
56     plot(t,180*Q(:,4:6)./pi)
57     grid on
58     title('Ángulos de Euler del quadrotor')
59     xlabel('t [s]')
60     ylabel('Ángulo [°]')
61     legend('\psi','\theta','\phi')
62     figure(3)
63     plot(t,180*Q(:,7:8)./pi)
64     grid on

```

```

63 title('Ángulos coordenadas esféricas primer elemento')
64 xlabel('t [s]')
65 ylabel('Ángulo [°]')
66 legend('\alpha_{1}', '\beta_{1}')
67 figure(4)
68 plot(t, 180*Q(:, 9:10) ./ pi)
69 grid on
70 title('Ángulos coordenadas esféricas segundo elemento')
71 xlabel('t [s]')
72 ylabel('Ángulo [°]')
73 legend('\alpha_{2}', '\beta_{2}')
74 figure(5)
75 plot(t, 180*Q(:, 11:12) ./ pi)
76 grid on
77 title('Ángulos coordenadas esféricas tercer elemento')
78 xlabel('t [s]')
79 ylabel('Ángulo [°]')
80 legend('\alpha_{3}', '\beta_{3}')
81 figure(6)
82 plot(t, 180*Q(:, 13:14) ./ pi)
83 grid on
84 title('Ángulos coordenadas esféricas cuarto elemento')
85 xlabel('t [s]')
86 ylabel('Ángulo [°]')
87 legend('\alpha_{4}', '\beta_{4}')
88 figure(7)
89 plot(t, 180*Q(:, 15:16) ./ pi)
90 grid on
91 title('Ángulos coordenadas esféricas efector final')
92 xlabel('t [s]')
93 ylabel('Ángulo [°]')
94 legend('\alpha_{E}', '\beta_{E}')
95 end

```

Para el caso del modelo desacoplado tenemos que el código de esta función es ligeramente diferente, por lo que tenemos que recordar ejecutar el archivo adecuado para cada simulación que es aquel que se encuentra en el mismo directorio:

```

1 function graficas(q, a, lt, pL, tauL)
2 close all; clc;
3 t=q.Time; % Vector de tiempos de simulación [s]
4 n=length(t);
5 a=a.Data(1, :); % Longitudes de los elementos del brazo robótico [m]
6 lt=lt.Data(1); % Distancia del CG del cuadrrotor a los rotores [m]
7 for i=1:16
8     for j=1:n
9         Q(j, i)=q.Data(1, i, j); % Componentes del vector de estado en cada instante ...
            de tiempo
10        end
11    end
12    for i=1:3
13        for j=1:n
14            PL(j, i)=pL.Data(1, i, j);
15        end
16    end

```

```

17 M=20;% Parámetro que acorta la duración de la representación de la ...
    simulación.
18 NN=floor(n/M);
19
20 % Gráfica animada de la simulación
21 for lll=1:NN
22     k=lll*M;
23     [X0(:,k),X1(:,k),X2(:,k),X3(:,k),X4(:,k),Xe(:,k),XN(:,k),XS(:,k),...
24     XE(:,k),XO(:,k)]=posicionpuntos(Q(k,1),Q(k,2),Q(k,3),Q(k,4),Q(k,5),...
25     Q(k,6),Q(k,7),Q(k,8),Q(k,9),Q(k,10),Q(k,11),Q(k,12),Q(k,13),Q(k,14),...
26     Q(k,15),Q(k,16),a(1),a(2),a(3),a(4),a(5),lt);
27     figure(8)
28     plot3([X0(1,k) XN(1,k)], [X0(2,k) XN(2,k)], -[X0(3,k) XN(3,k)], 'r-o') % El ...
        rojo es el eje norte cuerpo
29     axis([-2 2 -2 2 0 4])
30     grid on
31     hold on
32     title('Simulación posición y actitud del sistema QM en el espacio')
33     xlabel('x [m]')
34     ylabel('y [m]')
35     zlabel('z [m]')
36     plot3([X0(1,k) XS(1,k)], [X0(2,k) XS(2,k)], -[X0(3,k) XS(3,k)], 'g-o')
37     plot3([X0(1,k) XE(1,k)], [X0(2,k) XE(2,k)], -[X0(3,k) XE(3,k)], 'g-o')
38     plot3([X0(1,k) XO(1,k)], [X0(2,k) XO(2,k)], -[X0(3,k) XO(3,k)], 'g-o')
39     plot3([X0(1,k) X1(1,k)], [X0(2,k) X1(2,k)], -[X0(3,k) X1(3,k)], 'b-o')
40     plot3([X1(1,k) X2(1,k)], [X1(2,k) X2(2,k)], -[X1(3,k) X2(3,k)], 'b-o')
41     plot3([X2(1,k) X3(1,k)], [X2(2,k) X3(2,k)], -[X2(3,k) X3(3,k)], 'b-o')
42     plot3([X3(1,k) X4(1,k)], [X3(2,k) X4(2,k)], -[X3(3,k) X4(3,k)], 'b-o')
43     plot3([X4(1,k) Xe(1,k)], [X4(2,k) Xe(2,k)], -[X4(3,k) Xe(3,k)], 'r-')
44     hold off
45     daspect([1 1 1]);
46     axis([]);
47     ax = gca;
48     ax.XTick = linspace(-5,5,41);
49     ax.YTick = linspace(-5,5,41);
50     ax.ZTick = linspace(-5,5,41);
51     text(X0(1,k)+0.6,X0(2,k)-0.6,-X0(3,k)-0.8,sprintf('Simulation time (s) = ...
        %.1f',round(t(k),1)), 'FontSize',18);
52     drawnow;
53     end
54 % Gráficas de la evolución de las variables de estado
55     figure(1)
56     plot(t,Q(:,1:3))
57     grid on
58     title('Posición del CG del quadrotor')
59     xlabel('t [s]')
60     ylabel('Coordenada espacial [m]')
61     legend('x','y','z')
62     figure(2)
63     plot(t,180*Q(:,4:6)./pi)
64     grid on
65     title('Ángulos de Euler del quadrotor')
66     xlabel('t [s]')
67     ylabel('Ángulo [°]')
68     legend('\psi','\theta','\phi')
69     figure(3)

```

```
70 plot(t,180*Q(:,7:8)./pi)
71 grid on
72 title('Ángulos coordenadas esféricas primer elemento')
73 xlabel('t [s]')
74 ylabel('Ángulo [°]')
75 legend('\alpha_{1}','\beta_{1}')
76 figure(4)
77 plot(t,180*Q(:,9:10)./pi)
78 grid on
79 title('Ángulos coordenadas esféricas segundo elemento')
80 xlabel('t [s]')
81 ylabel('Ángulo [°]')
82 legend('\alpha_{2}','\beta_{2}')
83 figure(5)
84 plot(t,180*Q(:,11:12)./pi)
85 grid on
86 title('Ángulos coordenadas esféricas tercer elemento')
87 xlabel('t [s]')
88 ylabel('Ángulo [°]')
89 legend('\alpha_{3}','\beta_{3}')
90 figure(6)
91 plot(t,180*Q(:,13:14)./pi)
92 grid on
93 title('Ángulos coordenadas esféricas cuarto elemento')
94 xlabel('t [s]')
95 ylabel('Ángulo [°]')
96 legend('\alpha_{4}','\beta_{4}')
97 figure(7)
98 plot(t,180*Q(:,15:16)./pi)
99 grid on
100 title('Ángulos coordenadas esféricas efector final')
101 xlabel('t [s]')
102 ylabel('Ángulo [°]')
103 legend('\alpha_{E}','\beta_{E}')
104 figure(9)
105 plot(t,PL)
106 grid on
107 title('Posición del centro de masas del sistema quadrrotor-manipulador')
108 xlabel('t [s]')
109 ylabel('Coordenada espacial [m]')
110 legend('x','y','z')
111 figure(10)
112 plot(t,tauL.Data)
113 grid on
114 title('Acción de control de la dinámica del centro de masas \tau_{L}')
115 xlabel('t [s]')
116 ylabel('Fuerza [N]')
117 legend('\tau_{L}(1)','\tau_{L}(2)','\tau_{L}(3)')
118 end
```

## C.6. Función "centromasasinicial"

Esta función nos calcula para una condición inicial  $q_0$  el vector de posición del centro de masas del sistema inicial  $p_{L_0}$ , lo cual es útil cuando queremos hacer una simulación del modelo



de dinámicas desacopladas y queremos poner la condición inicial en el integrador de  $p_L$ .

```

1 function [xCG,yCG,zCG]=centromasasinicial(q0,a,const,m)
2 q=q0(1:16); % Vector de estado inicial
3 %% Cálculo de las matrices de rotación %%
4 A=[cos(q(4)) -sin(q(4)) 0;sin(q(4)) cos(q(4)) 0;0 0 1];
5 B=[cos(q(5)) 0 sin(q(5)); 0 1 0;-sin(q(5)) 0 cos(q(5))];
6 C=[1 0 0;0 cos(q(6)) -sin(q(6));0 sin(q(6)) cos(q(6))];
7 RB0=A*B*C;
8 TB0=[RB0(1,:) q(1);RB0(2,:) q(2);RB0(3,:) q(3);0 0 0 1];
9 D1=[cos(q(7)) -sin(q(7)) 0;sin(q(7)) cos(q(7)) 0;0 0 1];
10 E1=[cos(q(8)) 0 sin(q(8));0 1 0;-sin(q(8)) 0 cos(q(8))];
11 R1B=D1*E1;
12 T1B=[R1B(1,:) a(1)*cos(q(7))*sin(q(8));R1B(2,:) ...
    a(1)*sin(q(7))*sin(q(8));R1B(3,:) a(1)*cos(q(8));0 0 0 1];
13 D2=[cos(q(9)) -sin(q(9)) 0;sin(q(9)) cos(q(9)) 0;0 0 1];
14 E2=[cos(q(10)) 0 sin(q(10));0 1 0;-sin(q(10)) 0 cos(q(10))];
15 R21=D2*E2;
16 T21=[R21(1,:) a(2)*cos(q(9))*sin(q(10));R21(2,:) ...
    a(2)*sin(q(9))*sin(q(10));R21(3,:) a(2)*cos(q(10));0 0 0 1];
17 D3=[cos(q(11)) -sin(q(11)) 0;sin(q(11)) cos(q(11)) 0;0 0 1];
18 E3=[cos(q(12)) 0 sin(q(12));0 1 0;-sin(q(12)) 0 cos(q(12))];
19 R32=D3*E3;
20 T32=[R32(1,:) a(3)*cos(q(11))*sin(q(12));R32(2,:) ...
    a(3)*sin(q(11))*sin(q(12));R32(3,:) a(3)*cos(q(12));0 0 0 1];
21 D4=[cos(q(13)) -sin(q(13)) 0;sin(q(13)) cos(q(13)) 0;0 0 1];
22 E4=[cos(q(14)) 0 sin(q(14));0 1 0;-sin(q(14)) 0 cos(q(14))];
23 R43=D4*E4;
24 T43=[R43(1,:) a(4)*cos(q(13))*sin(q(14));R43(2,:) ...
    a(4)*sin(q(13))*sin(q(14));R43(3,:) a(4)*cos(q(14));0 0 0 1];
25 DE=[cos(q(15)) -sin(q(15)) 0;sin(q(15)) cos(q(15)) 0;0 0 1];
26 EE=[cos(q(16)) 0 sin(q(16));0 1 0;-sin(q(16)) 0 cos(q(16))];
27 RE4=DE*EE;
28 TE4=[RE4(1,:) 0;RE4(2,:) 0;RE4(3,:) 0;0 0 0 1];
29 TE0=TB0*T1B*T21*T32*T43*TE4; % Matriz de transformación del sistema de ...
    coordenadas inercial al del efector final.
30 Xe=TE0(1:3,4); % Posición del efector final.
31 % Calculamos la posición de los centros de gravedad de
32 % los distintos elementos.
33 X0G=[q(1),q(2),q(3)]';
34 T10G=TB0*[T1B(1:4,1:3), [T1B(1:3,4)*const(1);1]];
35 X1G=T10G(1:3,4);
36 T20G=TB0*T1B*[T21(1:4,1:3), [T21(1:3,4)*const(2);1]];
37 X2G=T20G(1:3,4);
38 T30G=TB0*T1B*T21*[T32(1:4,1:3), [T32(1:3,4)*const(3);1]];
39 X3G=T30G(1:3,4);
40 T40G=TB0*T1B*T21*T32*[T43(1:4,1:3), [T43(1:3,4)*const(4);1]];
41 X4G=T40G(1:3,4);
42 Xe;
43 % Cálculo del vector de posición del CG del sistema
44 posicionCG=(X0G*m(1)+X1G*m(2)+X2G*m(3)+X3G*m(4)+X4G*m(5)+Xe*m(6))/sum(m);
45 xCG=posicionCG(1); % Coordenada X del CG [m]
46 yCG=posicionCG(2); % Coordenada Y del CG [m]
47 zCG=posicionCG(3); % Coordenada Z del CG [m]
48 end

```



# Bibliografía

- [1] Wikipedia: [https://en.wikipedia.org/wiki/Euclidean\\_group](https://en.wikipedia.org/wiki/Euclidean_group)
- [2] Hyunsoo Yang and Dongjun Lee: *Dynamics and Control of Quadroter with Robotic Manipulator, 2014 IEEE International Conference on Robotics & Automation (ICRA)*
- [3] FayerWayer: <https://www.fayerwayer.com/2014/11/fabrican-un-dron-ambulancia-para-ayudar-en-situaciones-de-emergencia/>
- [4] Droning: <http://www.droning.es/servicios-rpa-droning/>
- [5] Robotics Trends: [http://www.roboticstrends.com/article/nepal\\_earthquake\\_drones\\_helping\\_recovery\\_efforts](http://www.roboticstrends.com/article/nepal_earthquake_drones_helping_recovery_efforts)
- [6] Euroxpress: <http://www.euroxpress.es/noticias/arcas-robots-voladores-para-llegar-a-donde-no-llegan-los-humanos>
- [7] Siciliano.B, Sciavicco.L, Villani.L y Oriolo.G: *Robotics: Modelling, Planning and Control*
- [8] D. J. Lee: *Passive decomposition and control of nonholonomic mechanical systems. IEEE Transactions on Robotics, 26(6):978-992, 2010*
- [9] D. J. Lee: *Passive decomposition of mechanical systems with coordination requirements. IEEE Transactions on Automatic Control, 58(1):230-235, 2013*
- [10] Wikipedia: [https://en.wikipedia.org/wiki/Tangent\\_space](https://en.wikipedia.org/wiki/Tangent_space)
- [11] Wikipedia: [https://en.wikipedia.org/wiki/Lie\\_derivative](https://en.wikipedia.org/wiki/Lie_derivative)