



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

MASTER EN INGENIERÍA INDUSTRIAL
ESCUELA DE INGENIERÍAS INDUSTRIALES
UNIVERSIDAD DE VALLADOLID

TRABAJO FIN DE MÁSTER

MODELO DINÁMICO Y SIMULACIÓN DEL ROBOT
MANIPULADOR IRB 120

Autor: D. Pablo Abia Morán
Tutor: D. Juan Carlos Fraile Marinero

Valladolid, septiembre de 2020

Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

**MODELO DINÁMICO Y SIMULACIÓN DEL
ROBOT MANIPULADOR IRB 120**

Trabajo Fin de Máster

Autor:

D. Pablo Abia Morán

Tutor:

D. Juan Carlos Fraile Marinero

Septiembre 2020

Resumen

Los robots manipuladores son un tipo de robot muy común caracterizado por tener una base estática sobre la que se conectan una serie de eslabones encadenados mediante articulaciones que le otorgan libertad de movimiento. Los manipuladores más comunes en la industria son los de seis grados de libertad.

El modelo dinámico de un sistema como el de un robot manipulador consiste en el conjunto de ecuaciones, basadas en las leyes de la mecánica clásica, que relacionan los parámetros cinemáticos y sus derivadas con las fuerzas y pares aplicados sobre él con el fin de describir su movimiento. En el caso de sistemas dinámicos complejos como el del presente estudio, compuesto por un mecanismo de múltiples grados de libertad, dos métodos de uso general facilitan el proceso de obtención de este modelo: se trata de los métodos de Euler-Lagrange y el de Newton-Euler.

Este trabajo se ha centrado en la aplicación del método recursivo de Newton-Euler al robot manipulador de la marca ABB: IRB 120. Sus parámetros dinámicos han sido estimados y verificados a través de múltiples simulaciones ejecutadas mediante métodos de control en lazo abierto y lazo cerrado. Efectivamente, cuando se trata de diseñar un sistema de control automático para un robot cualquiera, disponer de un modelo dinámico preciso se advierte fundamental para simular su movimiento real de modo fiel y repetible. Los resultados de las simulaciones han mostrado que el método de control en lazo abierto es inestable, mientras que los métodos de control en lazo cerrado —controlador PD y PID— consiguen respuestas asintóticamente estables.

Una línea de continuación de este trabajo podría ser la optimización del modelo dinámico mediante el análisis comparativo de las respuestas reales del robot IRB 120 y los resultados de las simulaciones, así como la aplicación de otras estrategias de control más complejas para mejorar la respuesta del sistema.

Abstract

Robot manipulators are a very common type of robot characterized by having a static base on which a series of links are connected by joints that let the whole mechanism have some many degrees of freedom. The dynamic model of a robot manipulator consists of a set of equations that relate its kinematic parameters and its derivatives with the forces and torques applied to it. This work has focused on the application of the Newton-Euler recursive method to the manipulator robot of the ABB brand: IRB 120. The results of the simulations have shown that the open-loop control method is unstable, while the closed-loop control methods achieve asymptotically stable responses.

Agradecimientos

En primer lugar, mi sincero agradecimiento a Don Juan Carlos Fraile Marinero, tutor académico de mi trabajo de fin de máster, por su atenta dedicación y su plena accesibilidad para atender mis dudas y ayudarme a avanzar en el estudio. También por todos los conocimientos que de él he aprendido durante la realización de mi trabajo.

En segundo lugar, quisiera mostrarle mi profunda gratitud a Doña Blanca Giménez Olavarria, por haberme dado la oportunidad de participar en el programa de doble titulación con la ENSAM y haber ejercido como mi tutora de principio a fin. Al término de esta etapa académica, mi valoración es tan positiva que no puedo expresarla totalmente en estas líneas. Sin lugar a dudas, me ha hecho mejor ingeniero y mejor persona.

Gracias también a la Universidad de Valladolid por todos los recursos que ha puesto a mi disposición para hacerme crecer en todos los ámbitos de mi persona. También por todas las amistades que me ha otorgado durante mis años de estudiante. Me siento muy orgulloso de egresar de la institución octocentenaria que representa y portaré toda mi vida este título con honor.

Por último, eterna gratitud a mi padre y a mi madre por haberme dado todo lo que soy y lo que tengo.

Índice general

Resumen	III
Abstract	v
Agradecimientos	vii
Índice de figuras	xii
1. Introducción	1
1.1. Contexto	1
1.2. Descripción del robot manipulador IRB 120	4
1.3. Objetivos del trabajo	5
1.4. Estado del arte	5
1.5. Herramientas informáticas utilizadas	6
2. Fundamentos teóricos	9
2.1. Posición y orientación del robot	9
2.1.1. Matrices de rotación	10
2.1.2. Matrices de transformación homogénea	11
2.1.3. Matrices anti-simétricas, velocidad angular y aceleración	11
2.2. Definición de la cadena cinemática del IRB120	13
2.3. Cálculo de la cinemática directa	14
2.4. Dinámica: Métodos de Newton-Euler y Euler-Lagrange	14
3. Caracterización del robot IRB 120	21
3.1. Datos técnicos del robot IRB 120	21
3.2. Parámetros de Denavit-Hartenberg del robot IRB 120	23
3.3. Parámetros cinemáticos del robot IRB 120	26
3.4. Parámetros dinámicos del IRB 120	28
3.5. El modelo dinámico del robot IRB 120	30

3.5.1. Recursión progresiva:	30
3.5.2. Recursión regresiva:	32
4. Simulaciones y resultados	35
4.1. Estructura del modelo y de las simulaciones en Simulink	36
4.2. Simulación en lazo abierto	38
4.3. Control proporcional con realimentación de velocidad	39
4.4. Simulación en lazo cerrado con control PD	41
4.5. Simulación en lazo cerrado con control PID	43
5. Conclusión del trabajo	51
5.1. Contribuciones	51
5.2. Posibles líneas de trabajo a futuro	52
A. Ficha técnica del robot IRB 120	53
B. Programa MAPLE del modelo dinámico del IRB120	57
Bibliografía	73

Índice de figuras

1.1. El robot IRb6, primer robot de la compañía ASEA	1
1.2. Robot "Leonardo da Vinci", ejemplo de cirugía robótica.	2
1.3. Robot IRB 120 de la marca ABB.	4
2.1. Cadena cinemática del IRB 120 y nomenclatura.	14
2.2. Ejemplo de robot montado sobre un vehículo de guiado automático.	17
2.3. Fuerzas y pares ejercidos sobre el eslabón i.	18
3.1. Esquema del robot manipulador IRB 120.	22
3.2. Cotas descriptivas del brazo robótico IRB 120.	23
3.3. Parámetros de Denavit-Hartenberg para una unión articulada.	25
3.4. Cadena cinemática del IRB 120 según el convenio de Denavit-Hartenberg.	26
4.1. Representación simple de las entradas y salidas de un controlador de posición en lazo cerrado.	36
4.2. Diagrama de bloques de la dinámica directa.	37
4.3. Diagrama de bloques de la dinámica inversa.	37
4.4. Esquema del modelo en Simulink con control en lazo abierto.	38
4.5. Resultados de la simulación en lazo abierto: posición y velocidad angulares.	39
4.6. Esquema del modelo en Simulink con control proporcional y realimentación de la velocidad.	40
4.7. Simulación 2: evolución de $\mathbf{e}(t)$ en LC con control proporcional y realimentación de la velocidad.	41
4.8. Simulación 2: evolución de \mathbf{q} y $\dot{\mathbf{q}}$ en LC con control proporcional y realimentación de la velocidad.	42
4.9. Simulación 2: evolución de $\boldsymbol{\tau}$ en LC con control proporcional y realimentación de la velocidad.	43
4.10. Esquema del modelo en Simulink con control PD en lazo cerrado.	44
4.11. Simulación 3: evolución de la señal de error de posición con un controlador PD.	45

4.12. Simulación 3: evolución de los pares en las articulaciones con un controlador PD.	45
4.13. Esquema del modelo de Simulink con control PID en lazo cerrado.	46
4.14. Simulación 4: evolución de los pares con el controlador PID.	47
4.15. Simulación 4: evolución de la posición y la velocidad con el controlador PID.	48
4.16. Simulación 4: evolución del error $e(t)$ con el controlador PID.	48
4.17. Simulación 5: evolución de los pares con el controlador PID.	49
4.18. Simulación 5: evolución de la posición y la velocidad con el controlador PID.	49
4.19. Simulación 5: evolución del error $e(t)$ con el controlador PID.	50

Capítulo 1

Introducción

Este capítulo pretende aportar al lector una visión general del trabajo desarrollado; definiendo en un primer lugar el contexto y los objetivos que lo han motivado, y posteriormente presentando la estructura y el contenido de la memoria.

1.1. Contexto

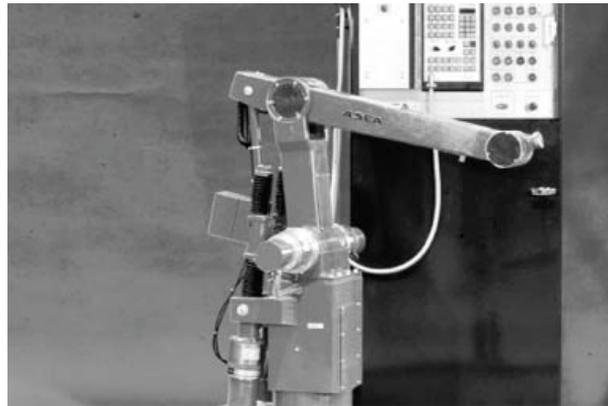


Figura 1.1: El robot IRb6, primer robot de la compañía ASEA

No es posible describir el robot manipulador objeto de este estudio sin antes introducir brevemente el campo de la robótica. El primer uso de la palabra *robot* se remonta a 1921, cuando fue empleada por el escritor y dramaturgo checo Karel Čapek en su obra de teatro “*Robots universales Rossum*” para describir unos “seres humanos artificiales” construidos con el objetivo de aligerar la carga de trabajo del resto de personas. El término procede de la palabra *robota*, que en checo significa *esclavo*. Sin embargo, los progenitores más directos de lo que hoy conocemos como robot no se denominaban como tal. Estos primeros aparatos, llamados “*manipuladores teleoperados*” se desarrollaron en la década de 1940 con el objetivo de manejar productos radioactivos a distancia sin exponer al operario. A partir de la década siguiente, se presentaron las primeras patentes de dispositivos robóticos en Reino Unido y Estados Unidos; y ya a principios de los años sesenta se instaló en la fábrica de General Motors de Trenton (EEUU) la primera máquina definida como robot *Unimate*, de la compañía norteamericana *Unimation (Universal Automation)*.

En Europa, la primera compañía que desarrolló un robot manipulador con accionamiento totalmente eléctrico fue ASEA, de origen sueco y sede en Estocolmo. Su robot se

llamó *IRb6* (Figura 1.1). Más adelante, en 1988, ASEA se unió a la empresa suiza Brown Boveri para formar la actual ABB, con sede en Zurich.

Hoy en día, la robótica se puede definir como la técnica dedicada al diseño y empleo de aparatos programables que, en sustitución de personas, realizan operaciones o trabajos, por lo general en instalaciones industriales y bajo control informático. No obstante, esta definición resulta muy genérica dada la gran variedad de sistemas que actualmente pueden ser definidos como robot. Por tanto, con el fin de especificar el tipo de robot del que se habla, se acostumbra a añadir un adjetivo a este término (robots manipuladores, robots humanoides, robots domésticos...). En concreto, los robots manipuladores son un tipo de robot definido por la Organización Internacional de Estándares (ISO), en su norma *ISO 8373* - equivalente a la *UNE EN ISO 8373:1998 "Robots Manipuladores Industriales. Vocabulario"* - como:

Robot manipulador industrial (ISO): Manipulador de 3 o más ejes, con control automático, reprogramable, multiaplicación, móvil o no, destinado a ser utilizado en aplicaciones de automatización industrial. Incluye al manipulador (sistema mecánico y accionadores) y al sistema de control (software y hardware de control y potencia).

Su uso está muy extendido en la industria; principalmente en el sector de la automoción, donde dadas las altas cadencias de fabricación, resulta muy rentable utilizar estos aparatos para la ejecución de tareas repetitivas. Algunas de las ventajas del uso de estos robots son su precisión, la alta repetibilidad de sus operaciones y la velocidad de ejecución de sus movimientos. Estas virtudes permiten mejorar la calidad de los procesos y reducir los tiempos y los costes de producción. Una de sus aplicaciones principales en esta industria es la soldadura de las carrocerías de los vehículos; proceso en el cual podemos encontrar hileras de estos robots trabajando coordinadamente.

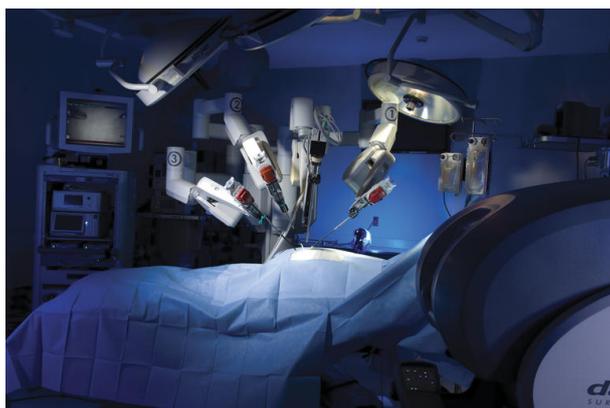


Figura 1.2: Robot "Leonardo da Vinci", ejemplo de cirugía robótica.

Pero las aplicaciones basadas en robots manipuladores no se encuentran únicamente en el sector industrial, sino que también tienen importancia en otros campos como el de la medicina, donde ya se utilizan en cirugía o en la rehabilitación física de enfermos. En cirugía son muy útiles para intervenir en operaciones sobre partes del cuerpo de difícil acceso para el cirujano que además requieren una elevada precisión (ver Figura 1.2).

Por último, antes de presentar en el siguiente epígrafe el robot IRB120 objeto de estudio, se enumeran las partes de la estructura mecánica de un robot manipulador genérico descritas en la norma *ISO 8373*, de modo que el lector pueda seguir la explicación cuando se haga referencia a unas u otras partes del robot:

1. **Accionador:** órgano de potencia capaz de generar un movimiento en el robot. Por lo general se trata de motores eléctricos que transforman la energía eléctrica en movimiento del robot.
2. **Brazo:** ejes principales del robot. Conjunto interconectado de eslabones y de articulaciones motorizadas, que forma una cadena que posiciona la muñeca.
3. **Muñeca:** ejes secundarios. Conjunto de eslabones y articulaciones motorizadas entre el brazo y el terminal que soporta, posiciona y orienta este terminal.
4. **Eslabón:** Cuerpo rígido que mantiene unidas las articulaciones.
5. **Articulaciones:** pueden ser de cuatro tipos según los grados de libertad que otorguen a los eslabones que conectan:
 - a) **Articulación prismática. Colisa:** unión entre dos eslabones que permite a uno de ellos tener un movimiento lineal en relación con el otro.
 - b) **Articulación rotativa:** articulación giratoria/rotativa. Unión entre dos eslabones que permite a uno de ellos tener un movimiento giratorio alrededor del eje del otro.
 - c) **Articulación cilíndrica:** unión entre dos eslabones que permite a uno de ellos tener un movimiento lineal o de rotación respecto al otro, según un eje de rotación asociado a la traslación.
 - d) **Articulación esférica:** unión entre dos eslabones que permite a uno de ellos un movimiento relativo respecto del otro alrededor de un punto fijo, según tres grados de libertad.
6. **Base:** plataforma o estructura asociada al extremo inicial del primer eslabón de la estructura articulada.
7. **Interfaz mecánica:** Superficie de montaje en el extremo de la estructura articulada del robot sobre la que se monta la herramienta.
8. **Terminal o herramienta:** Dispositivo específicamente concebido para fijarse a la interfaz mecánica del robot que permite al robot realizar un trabajo determinado.
9. **Eje:** Dirección utilizada para indicar el movimiento del robot, de forma lineal o angular.
10. **Grado de libertad (GDL):** una de las variables (de un máximo de seis) necesarias para definir los movimientos de un cuerpo en el espacio.

11. **Pose:** Combinación de posición y orientación del extremo del robot.
12. **Centro de herramienta (CDH) o Tool Center Point (TCP) en inglés:** punto definido para una aplicación dada en relación con el sistema de coordenadas de la herramienta.

1.2. Descripción del robot manipulador IRB 120

El robot manipulador IRB 120 es el robot industrial multiusos más pequeño de la firma ABB y en él se ha basado este trabajo de fin de máster (Figura 1.3). Su mecanismo cuenta con 6 grados de libertad —otorgados por sus 6 articulaciones rotativas accionadas mediante motores de corriente alterna sin mantenimiento. Pesa 25 kilogramos y es capaz de portar cargas de hasta 3 kilogramos (4 kilogramos con la muñeca en posición vertical). En este robot, el brazo está compuesto por los eslabones 1, 2 y 3; que sirven para posicionar la herramienta. La orientación de ésta se logra mediante los eslabones 4, 5 y 6; que componen la muñeca. La base permite fijar el manipulador en posición vertical u horizontal y en ella se localizan las conexiones para la alimentación de los motores eléctricos, las tomas de aire comprimido y las comunicaciones con el ordenador de control.



Figura 1.3: Robot IRB 120 de la marca ABB.

El robot IRB 120 resulta ideal para ejecutar tareas rápidas de recogida y desplazamiento de objetos; también para tareas de montaje y embalaje de objetos poco voluminosos. Su sistema de control avanzado le otorga una repetibilidad de pose con fluctuaciones máximas de 0,01 mm, una de las más altas del mercado. ABB presentó en 2009 el "*Fanta Can Challenge*", del que se pueden ver vídeos en su página web, y que le sirvió para presumir de la alta velocidad con la que sus modelos son capaces de ejecutar los movimientos respetando una gran exactitud y repetibilidad.

1.3. Objetivos del trabajo

El objetivo principal de este trabajo de fin de máster es la obtención del modelo dinámico del robot manipulador IRB 120 a partir del método recursivo de Newton-Euler, cuyo desarrollo matemático se presenta en capítulos posteriores. Como ya se ha explicado, el modelo dinámico de cualquier robot es un elemento imprescindible en el diseño de su sistema de control, puesto que representa las ecuaciones que relacionan sus parámetros cinemáticos (posición, velocidad y aceleración) con los pares y las fuerzas ejercidos por los actuadores para definir su movimiento.

En el trabajo se hace hincapié en las divergencias entre el modelo dinámico obtenido y el que sería un modelo dinámico perfecto, de modo que el lector entienda cuáles son las limitaciones de este estudio y qué dimensión podría alcanzar la obtención de un dinámico preciso susceptible de poder implementarse en un controlador real.

Por último, con el modelo dinámico calculado se han llevado a cabo varias simulaciones, añadiéndole controladores simples en lazo abierto y en lazo cerrado. Estas simulaciones han permitido analizar la estabilidad de unos y otros.

1.4. Estado del arte

El estudio de la cinemática y dinámica, así como de las estrategias de control automático de los robots manipuladores, es tan antiguo como la propia tecnología. Desde el principio, muchos trabajos se han orientado a optimizar los métodos computacionales que permiten el control automático de estos sistemas, y se han llevado a cabo numerosos estudios sobre este tema.

En este trabajo, para el cálculo del modelo dinámico del robot IRB 120 se han tomado como referencia dos estudios similares llevados a cabo con otros modelos de manipuladores de ABB: [12] y [19].

El método de Denavit-Hartenberg, que se presenta posteriormente y permite construir el modelo cinemático directo y obtener las matrices de rotación o de transformación homogénea del sistema, ha sido extraído de [5], donde se explica muy detalladamente el algoritmo que permite establecer los sistemas de coordenadas de cada articulación de tal modo que se pueda representar la cinemática del robot mediante 6 variables.

Las distintas estrategias de control que se han empleado en este trabajo son convencionales y muy conocidas en el campo del control automático. En el libro [15] y en el [21] se hace un estudio teórico muy detallado de cada una de ellas.

Por último, varios trabajos anteriores ya se han enfocado en la obtención del modelo dinámico del robot IRB 120, como son [4] y [17]. De la primera de estas fuentes bibliográficas se han tomado los parámetros dinámicos tales como las masas de los eslabones, los centros de masas y las matrices de inercias.

La documentación técnica del robot manipulador IRB 120 puesta a disposición del público por ABB [1] y [2] también han servido para obtener las dimensiones reales de los eslabones.

1.5. Herramientas informáticas utilizadas

En el desarrollo del estudio se han utilizado las siguientes herramientas informáticas:

Maple 2018

Maple es un programa informático orientado a la resolución de cálculos matemáticos simbólicos y algebraicos. Fue desarrollado en la Universidad de Waterloo, en Ontario (Canadá) en 1981. A partir de 1988 el programa fue mejorado y la compañía canadiense *Maplesoft*, con sede en la misma ciudad, comenzó a comercializarlo.

Este programa ha sido utilizado en este trabajo para derivar las ecuaciones de la dinámica del manipulador. Una librería específica de *Maple* ha permitido obtener el código informático equivalente en otros lenguajes de programación. Esto ha simplificado mucho el proceso de construcción del modelo en *Matlab-Simulink*.

Matlab R2015b/Simulink

Matlab (abreviatura de "Matrix Laboratory") es un programa de cálculo numérico computarizado con un entorno de desarrollo integrado (IDE) y un lenguaje de programación propio (lenguaje M). *Matlab* está optimizado para la manipulación matemática de matrices y vectores, siendo más rápido que otros lenguajes de programación convencionales en este tipo de operaciones.

El paquete MATLAB incluye la herramienta de simulación *Simulink*, que será utilizada en este trabajo para la creación del modelo dinámico y la implementación de estrategias de control en lazo abierto y en lazo cerrado.

RoKiSim

RoKiSim es una herramienta informática educativa y gratuita que permite la simulación en tres dimensiones (3D) de diferentes robots manipuladores de seis ejes. Fue desarrollado en el *Control and Robotics Lab* de la "*École de technologie Supérieure de Montreal*" (Canadá). Este programa cuenta en su librería de robots manipuladores con el IRB 120 de ABB, y permite observar la posición de los ejes coordinados de cada una de sus articulaciones (de acuerdo con el convenio de Denavit-Hartenberg). Las orientaciones de las articulaciones pueden representarse en varios convenios de los ángulos de Euler o en cuaternios.

Este programa se ha utilizado para analizar diferentes trayectorias que después se han tratado de reproducir en las simulaciones del modelo dinámico de este trabajo.

Inkscape

Inkscape es un programa de edición de gráficos vectoriales de código abierto (GPL - General Public License) que utiliza el formato Scalable Vector Graphics (SVG). Se ha empleado para el diseño de los gráficos utilizados en la memoria del trabajo.

MiKTeX/L^AT_EX

MiKTeX es una implementación de software libre del sistema T_EX para la producción profesional de documentos, especialmente científicos, que se basa en marcadores de texto. El sistema T_EX fue creado por Donald Knuth en 1978, y L^AT_EX es una extensión debida a Leslie Lamport que añade funcionalidades que facilitan la estructuración de los documentos.

Capítulo 2

Fundamentos teóricos

Es conveniente comenzar introduciendo de forma breve la nomenclatura utilizada y los conceptos matemáticos y físicos que permitan al lector entender adecuadamente lo expuesto en capítulos posteriores.

En la Sección 2.1 se enumeran distintas formas de representar la posición y la orientación del extremo del robot —y de cada uno de sus eslabones— haciendo hincapié en las matrices de transformación homogénea. Éstas representan una de las herramientas matemáticas más utilizadas en robótica para este fin y es a la que se ha recurrido en este trabajo. La teoría está basada en el contenido de [8], [9] y [22].

Seguidamente, en la Sección 2.2 se enumeran los elementos que componen una cadena cinemática y se presenta la nomenclatura empleada para la identificación de cada uno de ellos. El contenido de esta sección está basado en [12].

Una vez presentados los conceptos anteriores, se presenta en la Sección 2.3 la cinemática directa, que permite obtener la pose del robot a partir de las coordenadas articulares. Para el robot ABB IRB 120 es posible resolver las ecuaciones no lineales en las variables articulares que resultan cuando se fijan los valores de la pose del robot: se resuelve así la llamada cinemática inversa del robot. Esta circunstancia afortunada se deriva esencialmente del diseño del robot, al presentar éste tres ejes de rotación consecutivos que concurren en un punto. Los desarrollos teóricos completos se pueden consultar en [5].

Por último, en la Sección 2.4 se exponen los dos métodos más utilizados generalmente para la derivación de las ecuaciones que rigen la dinámica de un robot manipulador: el método de Newton-Euler y el de Euler-Lagrange. Se describe brevemente cada uno de ellos, haciendo hincapié en sus ventajas y desventajas, y se explica por qué se ha utilizado el primero para el desarrollo del modelo del IRB 120 en este trabajo. Para más detalles, se puede consultar la siguiente literatura, en la que está basada este contenido: [5], [12] y [22].

2.1. Posición y orientación del robot

La posición y orientación de un sólido rígido cualquiera se describe mediante la posición y orientación del sistema de coordenadas solidario a su movimiento. A esto lo denominamos la pose. Un vector que represente un punto en el espacio puede ser expresado en diferentes sistemas coordenados mediante el uso de poses relativas.

En matemáticas existen múltiples formas de representar tanto la posición como la

orientación de un sistema de coordenadas: las matrices de rotación ortonormales, los ángulos de Euler, los cuaternios, las matrices de transformación homogénea. Todas ellas consisten en representaciones matriciales; sin embargo, las matrices de transformación homogénea son las únicas que permiten la representación simultánea de la posición y la orientación y la ventaja de que la composición de transformaciones sucesivas se reduce al producto de matrices.

2.1.1. Matrices de rotación

Una matriz de rotación de dimensiones $n \times n$ describe la orientación de un sistema de coordenadas respecto a otro en el espacio euclídeo de n dimensiones. Para expresar las coordenadas de un vector en S_1 respecto a un sistema de coordenadas S_0 en tres dimensiones, se recurre a una matriz 3×3 de rotación:

$$R_1^0 = [\mathbf{x}_1^0 \quad \mathbf{y}_1^0 \quad \mathbf{z}_1^0]$$

en la que $\mathbf{x}_1^0, \mathbf{y}_1^0, \mathbf{z}_1^0$ son los vectores unitarios de los ejes de S_1 expresados respecto del sistema de referencia S_0 , y $R_1^0 \in SO(3) \subset \mathbb{R}^{3 \times 3}$.

Las matrices de rotación tienen, por tanto, las propiedades de una matriz ortonomal:

- Los vectores columna son ortogonales entre sí.
- Los vectores columna tienen magnitud unitaria.
- La inversa de una matriz de rotación es igual a su traspuesta: $R^T = R^{-1}$.
- El determinante de la matriz de rotación es igual a 1: $\det(R) = 1$.

Las tres matrices de rotación para un giro θ en torno a cada uno de los ejes $x-, y-, z-$ son:

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

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

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

Según el *teorema de rotación de Euler*, cualquier rotación de un sólido tridimensional puede representarse como una secuencia de tres rotaciones en torno a ejes de coordenadas diferentes. Las matrices de rotación ortonomal en tres dimensiones poseen nueve elementos, sin embargo no son independientes: como ya hemos visto, los vectores columna tienen magnitud unitaria, lo que añade tres restricciones al sistema. Por otra parte, los vectores columna son ortogonales entre sí, lo que añade otras tres restricciones hasta tener efectivamente 3 valores independientes.

Es importante tener en cuenta que las matrices de rotación no poseen la propiedad conmutativa; es decir: el orden en que se concatenan es muy importante.

Si se considera un punto \mathbf{p} y tres sistemas de referencia distintos S_0 , S_1 y S_2 , existen tres posibles representaciones de este punto \mathbf{p} : \mathbf{p}^0 , \mathbf{p}^1 y \mathbf{p}^2 . Las relaciones entre unas y otras representaciones son:

$$\mathbf{p}^0 = R_1^0 \mathbf{p}^1 \quad (2.1)$$

$$\mathbf{p}^0 = R_2^0 \mathbf{p}^2 \quad (2.2)$$

$$\mathbf{p}^1 = R_2^1 \mathbf{p}^2 \quad (2.3)$$

Por último, sustituyendo (2.3) en (2.1) y comparando el resultado con (2.2), se obtiene la siguiente identidad:

$$R_2^0 = R_1^0 R_2^1$$

que representa la regla de composición de rotaciones y establece que para expresar las coordenadas de \mathbf{p} en el sistema de coordenadas de S_2 respecto al sistema de coordenadas S_0 , se debe primero transformar \mathbf{p}^2 a \mathbf{p}^1 —en el sistema de referencia S_1 — y a continuación transformar \mathbf{p}^1 a \mathbf{p}^0 . La expresión generalizada de esta regla es:

$$R_n^0 = R_1^0 R_2^1 \dots R_n^{n-1}$$

2.1.2. Matrices de transformación homogénea

Como ya se introdujo, las matrices de transformación homogénea tienen la propiedad de expresar una rotación y una traslación simultáneamente. De aquí en adelante, las matrices de transformación homogénea serán denotadas por la letra A . Su estructura se compone de cuatro submatrices:

$$A = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{p}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ 0 & 1 \end{bmatrix}$$

Esta matriz permite representar la orientación y posición de un sistema de referencia S_1 resultado de rotar y trasladar el sistema original S_0 según $R_{3 \times 3}$ y $\mathbf{p}_{3 \times 1}$ respectivamente. También serviría para conocer las coordenadas (r_x^0, r_y^0, r_z^0) del vector \mathbf{r} en el sistema S_0 a partir de sus coordenadas en el sistema S_1 :

$$\begin{bmatrix} r_x^0 \\ r_y^0 \\ r_z^0 \\ 1 \end{bmatrix} = A_1^0 \begin{bmatrix} r_x^1 \\ r_y^1 \\ r_z^1 \\ 1 \end{bmatrix}$$

O para representar las nuevas coordenadas de un vector que ha experimentado un giro y una traslación en un sistema de referencia S_0 :

$$\begin{bmatrix} r_{x'} \\ r_{y'} \\ r_{z'} \\ 1 \end{bmatrix} = A_1^0 \begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix}$$

2.1.3. Matrices anti-simétricas, velocidad angular y aceleración

Las matrices de rotación resultan muy útiles para calcular por ordenador la velocidad y la aceleración relativas entre sistemas de coordenadas. Tales operaciones implican

derivadas temporales de los vectores de posición. Mediante el concepto de matriz anti-simétrica, se introducen algunas propiedades matemáticas que logran simplificar mucho los cálculos necesarios para obtener estas derivadas.

En primer lugar, una matriz es anti-simétrica sí y sólo sí satisface:

$$S^T + S = 0$$

Denotamos al conjunto de matrices 3×3 anti-simétricas como $SS(3)$. Siendo la matriz $S \in SS(3)$ de componentes s_{ij} (para $i, j = 1, 2, 3$), obtenemos:

$$s_{ij} + s_{ji} = 0$$

Ello implica que $s_{ii} = 0$, es decir, que los términos de la diagonal de una matriz anti-simétrica son 0. Por tanto, la estructura de una matriz anti-simétrica resulta:

$$S = \begin{bmatrix} 0 & -s_1 & s_2 \\ s_1 & 0 & -s_3 \\ -s_2 & s_3 & 0 \end{bmatrix}$$

Definimos la matriz anti-simétrica asociada al vector $\mathbf{a} = (a_x, a_y, a_z)$ como:

$$S(\mathbf{a}) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

Se presentan a continuación cuatro propiedades muy útiles e importantes de las matrices anti-simétricas:

1. Para dos vectores cualesquiera \mathbf{a} y \mathbf{b} pertenecientes a \mathbb{R}^3 y dos escalares α y β :

$$S(\alpha \mathbf{a} + \beta \mathbf{b}) = \alpha S(\mathbf{a}) + \beta S(\mathbf{b})$$

2. Y para dos vectores \mathbf{a} y \mathbf{p} pertenecientes a \mathbb{R}^3 :

$$S(\mathbf{a}) \mathbf{p} = \mathbf{a} \times \mathbf{p}$$

3. Considerando R como una matriz ortogonal perteneciente a $\mathbb{R}^{3 \times 3}$ y $\mathbf{a} \in \mathbb{R}^3$, se tiene que:

$$R S(\mathbf{a}) R^T = S(R \mathbf{a})$$

4. Y por último, si los elementos de una matriz ortogonal $R(t)$ dependen de t , entonces:

$$\dot{R}(t) = S(\boldsymbol{\omega}(t)) R(t) \tag{2.4}$$

El vector $\boldsymbol{\omega}(t)$ así definido es el vector velocidad angular instantáneo asociado a la rotación $R(t)$.

Suponiendo un vector \mathbf{p} fijo a un sistema de coordenadas S_1 que rota respecto a S_0 , se pueden obtener sus coordenadas respecto de éste último mediante la expresión $\mathbf{p}^0 = R_1^0(t) \mathbf{p}^1$. De ésto y de la identidad (2.4), se deriva la expresión para la velocidad con respecto al sistema de referencia S_0 :

$$\dot{\mathbf{p}}^0 = \boldsymbol{\omega}^0 \times \mathbf{p}^0.$$

Si ahora el sistema de referencia S_1 se mueve de formas más general, rotando con respecto a S_0 y con el origen O_1 de S_1 desplazándose linealmente, de modo que

$$\mathbf{p}^0 = R_1^0(t)\mathbf{p}^1 + \mathbf{d}(t)$$

entonces

$$\dot{\mathbf{p}}^0 = \boldsymbol{\omega} \times \mathbf{r} + \mathbf{v}$$

donde $\mathbf{r} = R_1^0(t)\mathbf{p}^1$ es el vector de O_1 a \mathbf{p} expresado en el sistema de referencia S_0 y $\mathbf{v} = \dot{\mathbf{d}}(t)$ es el vector de velocidad lineal del origen O_1 . Si además el vector \mathbf{p}^1 se mueve con respecto al sistema de referencia S_1 debemos añadir en la expresión anterior el término $R_1^0(t)\dot{\mathbf{p}}^1$, que es la velocidad de cambio de \mathbf{p}^1 expresada en el sistema de referencia S_0 . Una nueva derivación permite obtener la expresión de la aceleración relativa:

$$\ddot{\mathbf{p}}^0 = \dot{\boldsymbol{\omega}} \times \mathbf{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) + 2\boldsymbol{\omega} \times R_1^0(t)\dot{\mathbf{p}}^1 + \mathbf{a}$$

Ahora, $\mathbf{a} = \ddot{\mathbf{d}}$ es la aceleración lineal de O_1 ; la expresión $\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})$ se denomina aceleración centrípeta; y el término $\dot{\boldsymbol{\omega}} \times \mathbf{r}$ representa la aceleración tangencial. Por último, la aceleración de Coriolis viene representada por el término $2\boldsymbol{\omega} \times R_1^0(t)\dot{\mathbf{p}}^1$.

2.2. Definición de la cadena cinemática del IRB120

Como ya se ha explicado en secciones anteriores, los robots manipuladores están compuestos de una cadena de eslabones unidos entre sí mediante articulaciones y conectados a una base fija por uno de sus extremos. Las articulaciones pueden ser de varios tipos (rótulas, articulaciones rotativas, prismáticas, ...). En el caso del robot IRB 120 de ABB, se tiene seis articulaciones rotativas. Este tipo de articulaciones aportan un grado de libertad, y por tanto su movimiento es representado mediante una sola variable, que en nuestro trabajo identificaremos como $q_i, i = 1, 2, \dots, 6$. También utilizaremos la notación vectorial $\mathbf{q} = [q_1, \dots, q_6]^T$ para denotar el conjunto de estas variables o coordenadas generalizadas.

Cada sistema de referencia S_i es solidario al eslabón i . Los eslabones, las articulaciones y los sistemas de referencia utilizados se han identificado de la siguiente manera:

- Los eslabones se han numerado del 0 al 6, considerando la base como el eslabón 0.
- Las articulaciones se han numerado del 1 al 6, situándose la articulación i -ésima entre los eslabones $i - 1$ e i .
- La articulación i hace rotar al eslabón i . La base, no obstante, no tiene movimiento.
- Los sistemas de referencia se numeran de 0 a 6, siendo el sistema S_i solidario al eslabón i (frecuentemente con origen en la articulación $i + 1$).
- Los sistemas de referencia se fijan de modo que el eje z_i del sistema S_i se corresponda con el eje de rotación de la articulación $i + 1$.
- La variable q_i está asociada al eslabón i -ésimo, y cuando es rotatoria representa un giro con eje de rotación z_{i-1} .

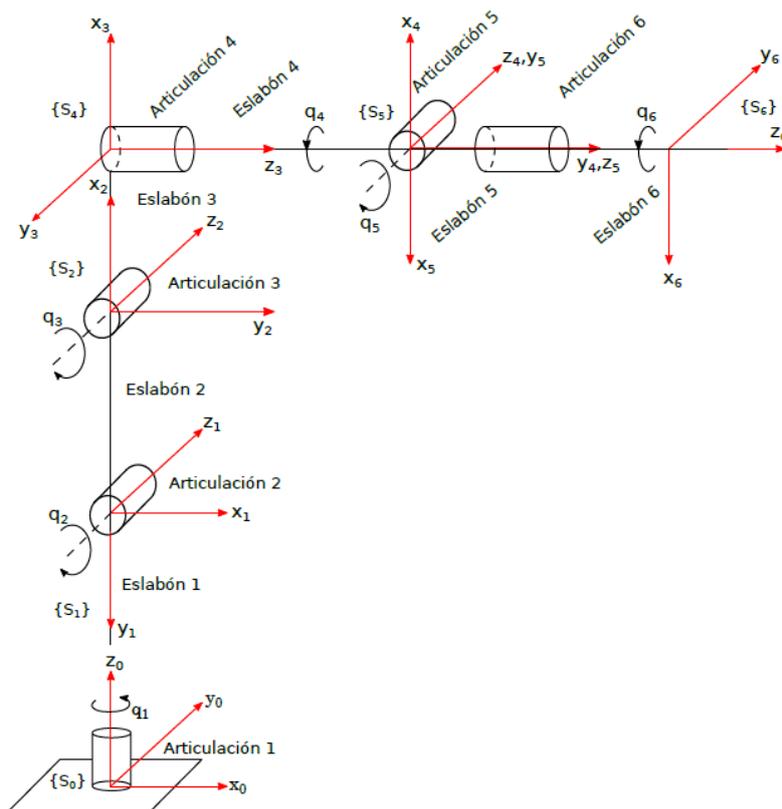


Figura 2.1: Cadena cinemática del IRB 120 y nomenclatura.

2.3. Cálculo de la cinemática directa

La cinemática directa permite obtener la posición y orientación del extremo del robot respecto al sistema de referencia de la base en función de los variables articulares q_1, q_2, q_3, q_4, q_5 y q_6 . Una vez obtenidas las matrices de transformación homogénea que relacionan entre sí los sistemas de referencia de cada eslabón, es trivial el cálculo de la cinemática directa: encadenando productos de estas matrices, se obtiene la matriz de transformación homogénea T_6^0 que representa la posición y la orientación del extremo del robot respecto al sistema de referencia absoluto

$$T_6^0 = A_1^0 A_2^1 A_3^2 \dots A_6^5$$

Las ecuaciones que caracterizan la pose del extremo del robot IRB 120 en términos de las coordenadas generalizadas q_1, q_2, \dots, q_6 se presentan en el siguiente capítulo una vez introduzcamos la forma explícita de las matrices de transformación homogénea A_i^{i-1} , $i = 1, 2, \dots, 6$, entre los sistemas solidarios de cada eslabón.

2.4. Dinámica: Métodos de Newton-Euler y Euler-Lagrange

Como ya se ha explicado en la introducción, la dinámica describe las relaciones existentes entre el movimiento de un mecanismo y los pares y fuerzas asociados a él. En el caso de un robot manipulador, la dinámica permite calcular los pares ejercidos por los motores sobre las articulaciones de modo que el robot desarrolle una trayectoria previamente calculada.

La obtención del modelo dinámico de un robot manipulador es de gran utilidad por varios motivos [5] :

- Para simular el movimiento del robot
- Para dimensionar el tamaño de los motores que actúan sobre cada articulación
- Para desarrollar un sistema de control ajustado
- Para analizar las sollicitaciones mecánicas ejercidas sobre la estructura del robot bajo distintas condiciones

En mecanismos simples de dos o tres grados de libertad, resulta relativamente sencillo calcular las expresiones de la dinámica e integrarlas para obtener su posición y orientación; sin embargo, a medida que aumenta el número de grados de libertad de un sistema, éstas se vuelven muy complejas y es necesario recurrir a otros métodos para calcularlas.

No obstante, el modelo dinámico completo de un robot no implica únicamente la mecánica de sus elementos, sino también las inercias y los rozamientos propios a los motores, a la transmisión, los valores de saturación de la electrónica de potencia, etc. El estudio de todas estas variables excede el objetivo de este trabajo, y por tanto en él sólo se han tenido en cuenta los factores mecánicos propios a la estructura del manipulador, de modo que se ha simplificado mucho el modelo dinámico calculado.

En el cálculo dinámico existen principalmente dos planteamientos distintos: el primero de ellos se basa en el concepto de equilibrio de fuerzas definido en la segunda ley de Newton y su equivalente para movimientos rotativos —la ley de Euler— que define el sumatorio de momentos como la derivada temporal del momento angular —producto de la matriz de inercias por la velocidad angular del cuerpo— [5], siempre refiriendo las magnitudes vectoriales a un sistema inercial:

$$\sum \mathbf{F} = \frac{d}{dt}(m\mathbf{v}) \quad (2.5)$$

y

$$\sum \mathbf{T} = \frac{d}{dt}(I\boldsymbol{\omega}) = I\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (I\boldsymbol{\omega}) \quad (2.6)$$

donde m es la masa del elemento, que se supone concentrada en un punto llamado centro de masas, \mathbf{v} es la velocidad lineal de este punto, I representa la matriz de inercias respecto al centro de masas, $\boldsymbol{\omega}$ es la velocidad angular y $\dot{\boldsymbol{\omega}}$ la aceleración angular del elemento. Conocida la expresión del par \mathbf{T} , se podría obtener la expresión de $\theta(t)$ y de sus derivadas $\dot{\theta}(t)$ y $\ddot{\theta}(t)$.

El segundo planteamiento posible utiliza la formulación Lagrangiana, basada en cálculos energéticos:

$$L(\mathbf{q}, \dot{\mathbf{q}}) = K(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q}),$$

donde $K = K(\mathbf{q}, \dot{\mathbf{q}})$ y $U = U(\mathbf{q})$ denotan respectivamente la energía cinética y la energía potencial del sistema formado por los eslabones del robot, expresadas en términos de las coordenadas articulares (coordenadas generalizadas). La función L recibe el nombre de Lagrangiana del sistema y las ecuaciones de la dinámica vienen dadas por las ecuaciones de Euler-Lagrange

$$\tau_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i}, \quad i = 1, \dots, n,$$

siendo τ_i el par aplicado en la coordenada generalizada q_i .

Mediante ambos planteamientos se alcanzaría la misma expresión matemática para el modelo dinámico de un robot [5] :

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (2.7)$$

donde

- \mathbf{q} es el vector de coordenadas articulares, de n componentes
- $\boldsymbol{\tau}$ es el vector ($n \times 1$) de fuerzas y pares aplicados en cada articulación
- $M(\mathbf{q})$ es la matriz de inercias, cuya dimensión es de $n \times n$
- $C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ es un vector ($n \times 1$) de fuerzas de Coriolis y fuerzas centrípetas, dependiente de \mathbf{q} y $\dot{\mathbf{q}}$
- $\mathbf{g}(\mathbf{q})$ es el vector ($n \times 1$) de fuerzas de gravedad, dependiente de \mathbf{q}
- n es el número de grados de libertad del robot.

La matriz de inercia $M(\mathbf{q})$ está estrechamente relacionada con la energética cinética del robot:

$$K(\mathbf{q}) = \frac{1}{2}\dot{\mathbf{q}}^T M(\mathbf{q})\dot{\mathbf{q}}$$

y es fundamental en la dinámica del robot y en el diseño de controles. Su propiedad más importante es que es una matriz simétrica definida positiva. La matriz de fuerzas centrípetas y de Coriolis $C(\mathbf{q}, \dot{\mathbf{q}})$ puede no ser única pero el vector $C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ lo es. La matriz $C(\mathbf{q}, \dot{\mathbf{q}})$ es lineal en su segundo argumento (en particular, $C(\mathbf{q}, \mathbf{0}) = 0$, para todos los vectores \mathbf{q}), y se relaciona con la matriz de inercia mediante:

$$\frac{d}{dt}M(\mathbf{q}) = C(\mathbf{q}, \dot{\mathbf{q}}) + C(\mathbf{q}, \dot{\mathbf{q}})^T.$$

Ambas formulaciones, método de Newton-Euler y método de Euler-Lagrange, admiten un procedimiento recursivo para el cálculo del modelo dinámico de un robot (2.7) que permite obtener éste con un coste computacional que varía linealmente con n , el número de grados de libertad del robot [13]. Cada método tiene ventajas e inconvenientes frente al otro: el método de Euler-Lagrange es más sistemático, pero para cadenas cinemáticas abiertas como el robot IRB 120 el método recursivo de Newton-Euler es el algoritmo más importante para el cálculo de la dinámica inversa del robot (obtención del vector de fuerzas generalizadas a partir de \mathbf{q} , $\dot{\mathbf{q}}$ y $\ddot{\mathbf{q}}$).

El método de Newton-Euler

El método de Newton-Euler está basado principalmente en tres leyes de la mecánica clásica [12], tal y como ya se explicó inicialmente en esta sección:

1. Toda acción tiene una reacción igual y de sentido contrario. Si el eslabón 6 ejerce un par $\boldsymbol{\tau}$ y una fuerza \mathbf{f} sobre el eslabón 5, entonces el eslabón 5 ejerce un par $-\boldsymbol{\tau}$ y una fuerza $-\mathbf{f}$.

2. La derivada del momento lineal aplicado al centro de masas es igual a la suma de fuerzas aplicadas sobre el eslabón.
3. La derivada del momento angular de un eslabón es igual a la resultante de todos los pares aplicados sobre dicho eslabón.

Se trata de un método recursivo en el que se parte de unas condiciones iniciales conocidas, como son las velocidades y aceleraciones lineales y angulares del eslabón 0 o base del robot manipulador, para recorrer cada eslabón desde 0 hasta n resolviendo las ecuaciones cinemáticas correspondientes para las velocidades y aceleraciones. Posteriormente, mediante los valores de aceleraciones y velocidades calculados en un primer momento y una serie de condiciones terminales definidas en el extremo del eslabón n —relativas a las fuerzas y pares aplicados en dicho extremo— se recorren los eslabones desde n hasta 0 calculando esta vez las expresiones de fuerzas y pares asociados a cada uno.

Los valores de las variables iniciales son generalmente:

$$\begin{aligned}\mathbf{v}_0 &= [0 \ 0 \ 0]^T \\ \mathbf{a}_0 &= [0 \ 0 \ 0]^T \\ \boldsymbol{\omega}_0 &= [0 \ 0 \ 0]^T \\ \dot{\boldsymbol{\omega}}_0 &= [0 \ 0 \ 0]^T\end{aligned}$$

Estas condiciones podrían ser distintas si el robot manipulador estuviera montado sobre una base móvil, como ocurre en el caso de la Figura 2.2, y por tanto tuviera una velocidad distinta de 0 o una aceleración lineal no nula. Por ejemplo, fijar una aceleración $\mathbf{a}_0 = [0, 0, -g]^T$ igual a la aceleración de la gravedad (con magnitud dada por la variable g de valor 9.8 m/s^2 y con sentido negativo en el eje z), equivale a incluir el efecto de las fuerzas gravitatoria de cada vínculo en el término masa por aceleración.



Figura 2.2: Ejemplo de robot montado sobre un vehículo de guiado automático.

Los valores terminales pueden ser muy variables según el tipo de simulación que se desee realizar, pero de manera general se considera que no hay ningún par ni fuerza aplicada en el extremo del robot manipulador en el momento de realizar las simulaciones:

$$\mathbf{f}^{ext} = \boldsymbol{\tau}^{ext} = 0$$

Para que el lector entienda mejor la nomenclatura utilizada en las ecuaciones, se ha hecho un esquema (Figura 2.3) con una representación simple de un eslabón cualquiera i , y

sobre él se han definido las fuerzas y pares que actúan. A continuación se presenta también un glosario de los términos empleados en el método, con todos los vectores expresados en términos del sistema de referencia S_i :

- $\mathbf{r}_{i-1,ci}$ = vector que une el origen del sistema de referencia S_{i-1} al centro de masas del eslabón i .
- $\mathbf{r}_{i-1,i}$ = vector que une el origen del sistema S_{i-1} al origen de S_i .
- $\mathbf{r}_{i,ci}$ = vector que une el origen de S_i con el centro de masas del eslabón i .
- \mathbf{f}_i = fuerza ejercida por el eslabón $i - 1$ sobre el eslabón i .
- $\boldsymbol{\tau}_i$ = par ejercido por el eslabón $i - 1$ sobre el eslabón i .
- m_i = masa del eslabón i .
- I_i = matriz de inercia del eslabón i -ésimo con respecto a un sistema de referencia paralelo al S_i centrado en el centro de masas
- $\mathbf{a}_{c,i}$ = aceleración absoluta del centro de masas del eslabón i .
- $\mathbf{a}_{e,i}$ = aceleración del final del eslabón i , coincidente con el origen de S_{i+1} .
- \mathbf{g}_i = aceleración de la gravedad expresada en el sistema S_i .
- $\boldsymbol{\omega}_i$ = velocidad angular del sistema de referencia S_i .
- $\boldsymbol{\alpha}_i$ = aceleración angular de S_i .
- \mathbf{z}_i = eje de rotación del sistema coordenado S_i respecto de S_0 .
- R_{i+1}^i = matriz de rotación de S_i respecto de S_{i+1} .

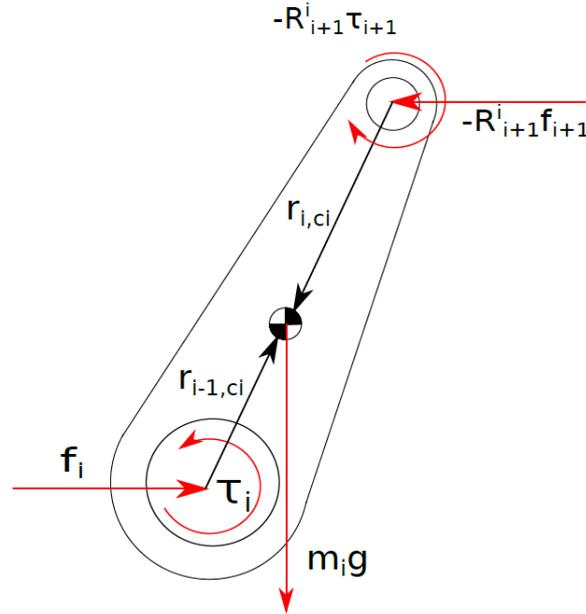


Figura 2.3: Fuerzas y pares ejercidos sobre el eslabón i .

Según la ley de acción-reacción, la fuerza \mathbf{f}_i es la ejercida por el eslabón $i - 1$ sobre el eslabón i , y $-\mathbf{f}_{i+1}$ es la fuerza ejercida por el eslabón $i + 1$. Para poder manipular ambos vectores, es necesario expresarlos en el mismo sistema de referencia; sin embargo, \mathbf{f}_{i+1} está expresada según el sistema de referencia $i + 1$, y \mathbf{f}_i en el sistema de referencia i . Por tanto, se debe multiplicar el primero por la matriz de rotación R_{i+1}^i para expresarla en el sistema de coordenadas adecuado. Una vez que todos los vectores están expresados respecto del mismo sistema de referencia, se puede hacer el balance de fuerzas en el eslabón i según la

expresión (2.5), tal como sigue:

$$\begin{aligned}\sum \mathbf{F}_i &= m_i \mathbf{a}_{i,ci} \\ \mathbf{f}_i - R_{i+1}^i \mathbf{f}_{i+1} + m_i \mathbf{g}_i &= m_i \mathbf{a}_{c,i} \\ \mathbf{f}_i &= R_{i+1}^i \mathbf{f}_{i+1} + m_i \mathbf{a}_{c,i} - m_i \mathbf{g}_i\end{aligned}\quad (2.8)$$

Como ya advertimos, el término gravitatorio $-m_i \mathbf{g}_i$ puede eliminarse de la ecuación (2.8) si se impone a la base una aceleración $\mathbf{a}_0 = [0, 0, -g]^T$.

De idéntica forma, se puede hacer el balance de pares según la expresión (2.6) tal como se muestra a continuación:

$$\begin{aligned}\sum \boldsymbol{\tau}_i &= \boldsymbol{\omega}_i \times (I_i \boldsymbol{\omega}_i) + I_i \dot{\boldsymbol{\omega}}_i \\ \boldsymbol{\tau}_i - R_{i+1}^i \boldsymbol{\tau}_{i+1} + \mathbf{f}_i \times \mathbf{r}_{i-1,ci} - (R_{i+1}^i \mathbf{f}_{i+1}) \times \mathbf{r}_{i,ci} &= \boldsymbol{\omega}_i \times (I_i \boldsymbol{\omega}_i) + I_i \boldsymbol{\alpha}_i \\ \boldsymbol{\tau}_i &= R_{i+1}^i \boldsymbol{\tau}_{i+1} - \mathbf{f}_i \times \mathbf{r}_{i-1,ci} + (R_{i+1}^i \mathbf{f}_{i+1}) \times \mathbf{r}_{i,ci} + \boldsymbol{\omega}_i \times (I_i \boldsymbol{\omega}_i) + I_i \boldsymbol{\alpha}_i\end{aligned}\quad (2.9)$$

Resolviendo la ecuación (2.9) a partir de las condiciones terminales —correspondientes a los valores de pares y fuerzas externos ejercidos sobre el extremo del robot manipulador— y sustituyendo en ella la expresión de la fuerza de la ecuación (2.8), se obtendría el resultado del par en cada articulación. No obstante, estas ecuaciones deben expresarse en función de las coordenadas articulares y sus derivadas q_i , \dot{q}_i y \ddot{q}_i . Para ello, el método de Newton-Euler recurre previamente a un cálculo recursivo en orden creciente de i , partiendo de las condiciones iniciales ya indicadas anteriormente, y relacionando $\boldsymbol{\omega}_i$, $\dot{\boldsymbol{\omega}}_i$ y $\mathbf{a}_{c,i}$ con q_i , \dot{q}_i y \ddot{q}_i .

En primer lugar, la expresión de la velocidad angular $\boldsymbol{\omega}_i$ en función del sistema de referencia inercial o absoluto —representada por un (0) en el superíndice— es:

$$\boldsymbol{\omega}_i^{(0)} = \boldsymbol{\omega}_{i-1}^{(0)} + \mathbf{z}_{i-1} \dot{q}_i$$

Y esta misma expresión, según el sistema de referencia fijo al eslabón i , resultaría:

$$\boldsymbol{\omega}_i = (R_i^{i-1})^T \boldsymbol{\omega}_{i-1} + \mathbf{b}_i \dot{q}_i \quad (2.10)$$

donde \mathbf{b}_i es igual a:

$$\mathbf{b}_i = (R_i^0)^T R_{i-1}^0 \mathbf{z}_0 \quad (2.11)$$

y es el vector que representa la dirección del eje de rotación de la articulación i expresado en el sistema de referencia S_i .

Respecto a la aceleración angular $\boldsymbol{\alpha}_i$, derivada de la velocidad angular del eslabón i pero expresada en el sistema de coordenadas solidario con ese eslabón, es:

$$\boldsymbol{\alpha}_i = (R_i^0)^T \dot{\boldsymbol{\omega}}_i^{(0)}$$

Es importante que el lector entienda que $\boldsymbol{\alpha}_i \neq \dot{\boldsymbol{\omega}}_i$. La derivada de la expresión (2.10) es:

$$\dot{\boldsymbol{\omega}}_i^{(0)} = \dot{\boldsymbol{\omega}}_{i-1}^{(0)} + \mathbf{z}_{i-1} \ddot{q}_i + \boldsymbol{\omega}_i^{(0)} \times \mathbf{z}_{i-1} \dot{q}_i$$

y esta expresada en el sistema de referencia S_i es:

$$\boldsymbol{\alpha}_i = (R_i^{i-1})^T \boldsymbol{\alpha}_{i-1} + \mathbf{b}_i \ddot{q}_i + \boldsymbol{\omega}_i \times \mathbf{b}_i \dot{q}_i \quad (2.12)$$

Por último, la aceleración lineal del centro de masas del eslabón i se obtiene derivando la ecuación de su velocidad lineal, cuya expresión es:

$$\mathbf{v}_{c,i}^{(0)} = \mathbf{v}_{e,i-1}^{(0)} + \boldsymbol{\omega}_i^{(0)} \times \mathbf{r}_{i-1,ci}^{(0)} \quad (2.13)$$

y por tanto, teniendo en cuenta que $\mathbf{r}_{i-1,ci}^{(0)}$ es constante, la expresión de la aceleración del centro de masas del eslabón i respecto del sistema absoluto o inercial S_0 resulta:

$$\mathbf{a}_{c,i}^{(0)} = \mathbf{a}_{e,i-1}^{(0)} \times \mathbf{r}_{i-1,ci}^{(0)} + \boldsymbol{\omega}_i^{(0)} \times (\boldsymbol{\omega}_i^{(0)} \times \mathbf{r}_{i-1,ci}^{(0)})$$

Esta aceleración, expresada en el sistema de coordenadas solidario al eslabón i mediante el uso de las matrices de rotación y sus propiedades (ver sección 2.1.1), resulta:

$$\mathbf{a}_{c,i} = (R_i^{i-1})^T \mathbf{a}_{e,i-1} + \dot{\boldsymbol{\omega}}_i \times \mathbf{r}_{i-1,ci} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_{i-1,ci}) \quad (2.14)$$

Y por último, se puede expresar la aceleración del extremo del eslabón - coincidente con el origen de coordenadas del sistema S_{i+1} - sustituyendo $\mathbf{r}_{i-1,ci}$ por $\mathbf{r}_{i-1,i}$ en la ecuación anterior, de modo que se obtiene:

$$\mathbf{a}_{e,i} = (R_i^{i-1})^T \mathbf{a}_{e,i-1} + \dot{\boldsymbol{\omega}}_i \times \mathbf{r}_{i-1,i} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_{i-1,i}) \quad (2.15)$$

El método de Newton-Euler consistiría, por tanto, en una recursión progresiva de 0 a n y una regresiva de n a 0. En resumen:

1. **Recurrencia progresiva:** que parte de las condiciones iniciales ya definidas anteriormente:

$$\mathbf{a}_0 = \mathbf{v}_0 = \boldsymbol{\omega}_0 = \dot{\boldsymbol{\omega}}_0 = [0 \ 0 \ 0]^T$$

y recorre los eslabones desde $i = 1$ hasta n calculando en cada paso las expresiones (2.10), (2.12), (2.15) y (2.14) según ese orden.

2. **Recurrencia regresiva:** que, partiendo de las condiciones

$$\boldsymbol{\tau}_{n+1} = \mathbf{f}_{n+1} = 0$$

y calculando desde $i = n$ hasta $i = 0$, evalúa las expresiones (2.9) y (2.8).

Este procedimiento es susceptible de ser programado en un sistema de cálculo simbólico para obtener fórmulas para las componentes del vector de fuerzas y momentos $\boldsymbol{\tau}$ en términos de \mathbf{q} , $\dot{\mathbf{q}}$ y $\ddot{\mathbf{q}}$. El apéndice B muestra una implementación en el programa Maple para la realización de estos cálculos.

Capítulo 3

Caracterización del robot IRB 120

En el presente capítulo se presentan los parámetros cinemáticos y dinámicos necesarios para establecer el modelo del robot IRB 120 y se explica cómo se han obtenido.

En la sección 3.1 se presentan los datos técnicos del modelo IRB 120 que ABB ha hecho de dominio público en la documentación técnica del equipo. ABB ha generado, para este robot manipulador, igual que para el resto de los de su gama, la documentación técnica suficiente para facilitar al usuario la elección del más adecuado según el tipo de aplicación que desee; así como instalar, manipular y conservar correctamente el sistema. Sin embargo, como desarrollador y fabricante, ABB evita hacer pública cualquier información que desvele parámetros confidenciales o que facilite la reproducción de su modelo por terceros. Por esta razón, para caracterizar el robot IRB 120 se ha debido recurrir a las estimaciones disponibles de algunos de sus parámetros.

En la sección 3.2, se detalla el cálculo de los parámetros cinemáticos del robot a partir de los valores obtenidos mediante el algoritmo de Denavit-Hartenberg. Este algoritmo, introducido por Jacques Denavit y Richard S. Hartenberg en 1955, permite reducir a 4 los parámetros necesarios para asociar sistemas de referencia no inerciales a los eslabones de las cadenas cinemáticas o robots manipuladores. A partir de estos parámetros se obtienen las matrices de rotación de la cadena cinemática.

En la sección 3.3 se detalla el cálculo de las matrices de transformación homogénea del manipulador IRB 120 así como el resto de parámetros cinemáticos necesarios para establecer el modelo del robot.

A continuación, en la sección 3.4 se explica cómo se han obtenido los valores de los parámetros dinámicos fundamentales en la descripción del robot manipulador IRB120. Estos son: las masas, los centros de masas y las matrices de inercia de cada eslabón respecto de los sistemas de referencia solidarios con origen trasladado a los centros de masas.

Por último, en la sección 3.5 se explica cómo ha sido el desarrollo del modelo dinámico del robot manipulador IRB 120 a partir de los parámetros obtenidos.

3.1. Datos técnicos del robot IRB 120

Las referencias bibliográficas [1] y [2] corresponden a la documentación técnica que ABB pone a disposición del usuario del robot manipulador IRB 120 para su correcto uso y mantenimiento. Como ya se presentó en la sección 1.2, se trata de un robot de 6 grados

de libertad cuyas características técnicas principales son:

Robot manipulador	Capacidad de porte (kg)	Alcance (m)	Peso (kg)
IRB 120	3 kg	0.58 m	25 kg

Cuadro 3.1: Datos generales del robot IRB 120

La Figura 3.1 representa un esquema en tres dimensiones del IRB 120 en el que se aprecian los ejes de rotación de cada una de sus articulaciones. Los rangos de movimiento de cada uno de ellos, medidos en grados de ángulo, se han compilado en el Cuadro 3.2.

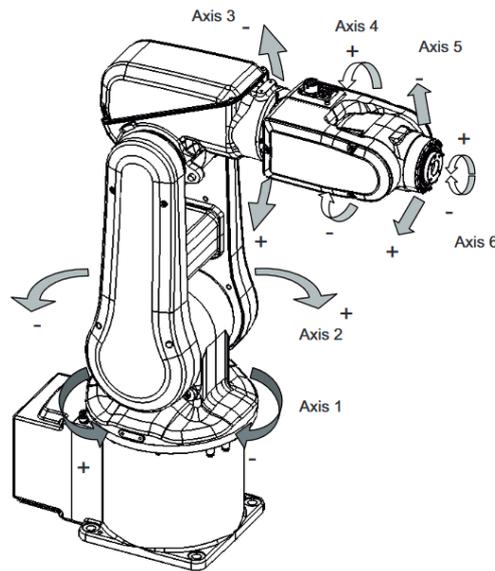


Figura 3.1: Esquema del robot manipulador IRB 120.

Ubicación del movimiento	Tipo de movimiento	Rango de movimiento
Eje 1	Movimiento de rotación	+165° a -165°
Eje 2	Movimiento del brazo	+110° a -110°
Eje 3	Movimiento del brazo	+70° a -110°
Eje 4	Movimiento de la muñeca	+160° a -160°
Eje 5	Movimiento de doblado	+120° a -120°
Eje 6	Movimiento de giro	+400° a -400°

Cuadro 3.2: Rango de movimiento de cada eje del robot manipulador.

Por otra parte, los ensayos de rendimiento del IRB 120 realizados por el fabricante según la norma *ISO 9283* han aportado los siguientes valores de repetibilidad y exactitud:

- Repetibilidad de pose (RP) en mm: 0.01.
- Exactitud de pose (AP) en mm: 0.02.
- Repetibilidad de trayectoria lineal (RT) en mm: 0.07 - 0.16.
- Exactitud de trayectoria lineal (AT) en mm: 0.21-0.38.

Para lograr tales valores, ha sido imprescindible el desarrollo de un modelo dinámico muy preciso que el controlador del manipulador utiliza para regular las respuestas de los actuadores. En este modelo dinámico, uno de los desafíos más grandes de ABB consiste en identificar todos los parámetros dinámicos del mecanismo complejo del robot manipulador. Generalmente, este proceso conlleva muchos meses de trabajo y numerosos ensayos. De hecho, el modelo dinámico implementado en el controlador comercial de ABB es completamente confidencial. Por último, las cotas que se incluyen en los planos

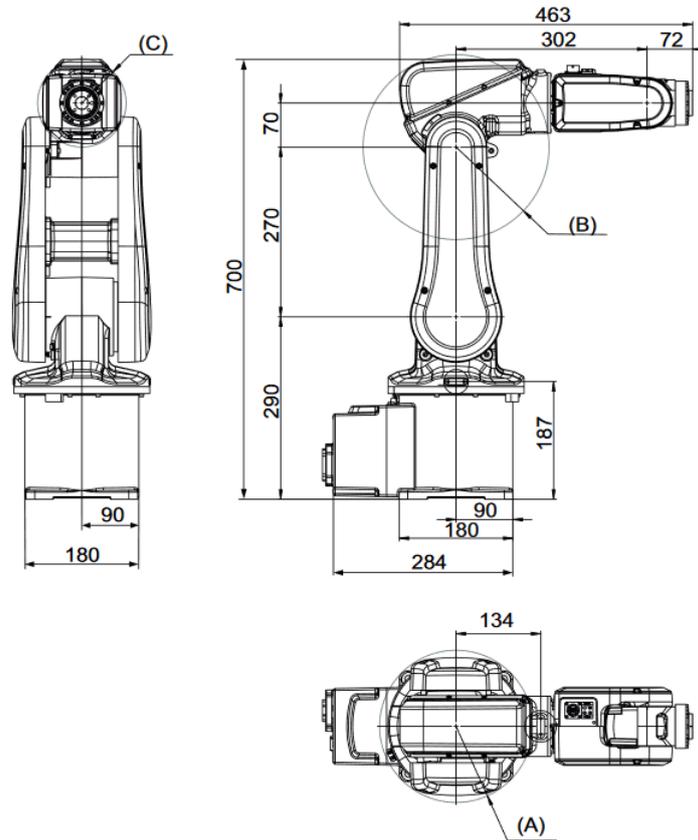


Figura 3.2: Cotas descriptivas del brazo robótico IRB 120.

de la documentación técnica del producto han facilitado la obtención de los parámetros de Denavit-Hartenberg para el robot manipulador IRB 120. Este proceso se explica en la sección siguiente.

3.2. Parámetros de Denavit-Hartenberg del robot IRB 120

El método más habitual en robótica para establecer los sistemas de referencia ligados a cada elemento es el de Denavit-Hartenberg. En esta sección se hace referencia al Capítulo 4 del libro *Fundamentos de robótica* de Antonio Barrientos [5], donde se explica en detalle el algoritmo de Denavit-Hartenberg para establecer los sistemas de coordenadas de cualquier cadena cinemática.

Jacques Denavit y Richard S. Hartenberg formularon en 1955 un método matricial que define la posición y orientación que debe tomar cada sistema de referencia S_i asociado a cada eslabón i para poder sistematizar el cálculo de las ecuaciones cinemáticas de una

cadena completa. Según este método, sólo son necesarias cuatro transformaciones simples para pasar de un sistema de referencia i al siguiente. Estas transformaciones dependen únicamente de las características geométricas de cada eslabón (ver Figura 3.2).

Las cuatro transformaciones básicas que deben sucederse consisten en una serie de rotaciones y traslaciones que permiten relacionar el sistema de referencia S_{i-1} en el sistema de referencia S_i . Estas son:

1. Rotación alrededor del eje \mathbf{z}_{i-1} un ángulo θ_i .
2. Traslación a lo largo del eje \mathbf{z}_{i-1} una distancia d_i .
3. Traslación a lo largo del eje \mathbf{x}_i una distancia a_i .
4. Rotación alrededor del eje \mathbf{x}_i un ángulo α_i .

El producto de estas cuatro transformaciones matriciales, que dado que no es conmutativo debe hacerse en el orden indicado, permiten obtener las matrices de transformación que relacionan los sistemas de referencia de cada eslabón con el siguiente:

$$\begin{aligned}
 A_i^{i-1} &= Rot_z(\theta_i) T(0, 0, d_i) T(a_i, 0, 0) Rot_x(\alpha_i) \\
 A_i^{i-1} &= \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}
 \end{aligned}$$

Aquí θ_i , d_i , a_i , α_i son los parámetros de Denavit-Hartenberg del eslabón i . Los senos y cosenos han sido simplificados mediante S y C .

Los parámetros de D-H se obtienen de la siguiente forma (ver Figura 3.3):

- θ_i Se trata del ángulo entre los ejes \mathbf{x}_{i-1} y \mathbf{x}_i según un plano perpendicular al eje \mathbf{z}_{i-1} . Este parámetro es variable en articulaciones giratorias.
- d_i Es la distancia medida a lo largo del eje \mathbf{z}_{i-1} desde el origen del sistema de coordenadas $(i-1)$ -ésimo hasta la intersección del eje \mathbf{z}_{i-1} con la dirección del eje \mathbf{x}_i . Es un parámetro variable en articulaciones prismáticas.
- a_i Es la distancia medida a lo largo del eje \mathbf{x}_i entre la intersección del eje \mathbf{z}_{i-1} con el eje \mathbf{x}_i hasta el origen del sistema i -ésimo para articulaciones giratorias como las del IRB 120.
- α_i Se mide como el ángulo entre los ejes \mathbf{z}_{i-1} y \mathbf{z}_i , según el plano perpendicular al eje \mathbf{x}_i y utilizando la regla de la mano derecha.

Cuando se quieren calcular relaciones entre sistemas de referencia no consecutivos, basta con hacer el producto de las sucesivas matrices de transformación A_i^{i-1} entre ambos sistemas de referencia. Así, la matriz de transformación entre el sistema de referencia S_i y el sistema de referencia S_j , con $j > i$, viene dado por

$$A_j^i = A_{i+1}^i A_{i+2}^{i+1} \cdots A_j^{j-1}.$$

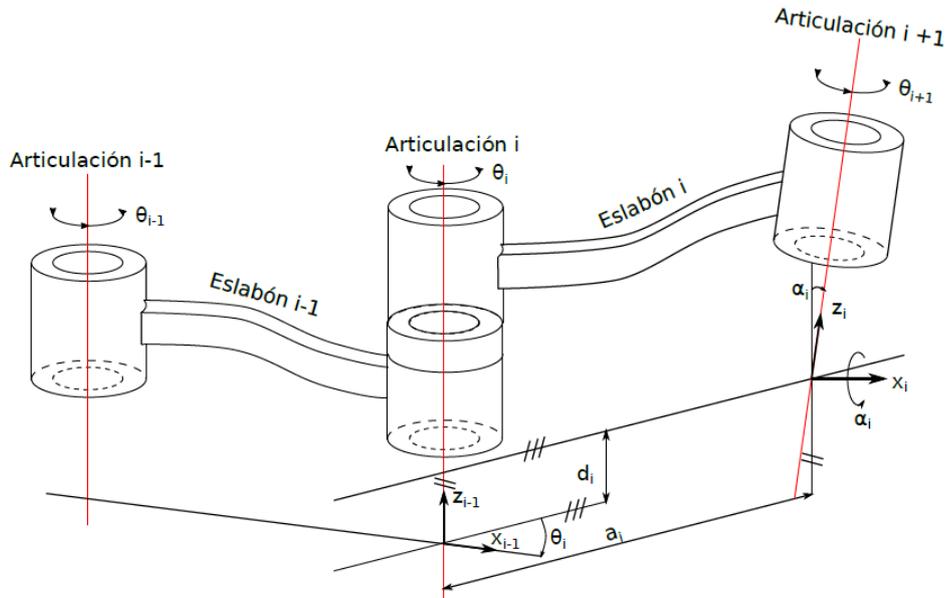


Figura 3.3: Parámetros de Denavit-Hartenberg para una unión articulada.

Para obtener los parámetros de Denavit-Hartenberg, previamente se debe establecer la posición y orientación de los sistemas de referencia $\{S_i\}$ de cada eslabón según el convenio definido por este método. En este trabajo, se ha seguido el algoritmo propuesto por Antonio Barrientos en [5]. Uno de los criterios principales de este convenio es establecer el eje z_i en la misma dirección que el eje de rotación de la articulación $i + 1$, situando posteriormente x_i en la línea normal común a z_i y z_{i-1} . Por último, la orientación del eje y_i debe conformar un sistema dextrógiro junto a x_i y z_i .

El conjunto de sistemas de referencia utilizado en este trabajo para el robot IRB 120 se muestra en la Figura 3.4. A partir de este esquema se han establecido los parámetros de Denavit-Hartenberg de la Tabla 3.3, cuyas unidades son radianes para los ángulos θ y α , y metros para las longitudes d y a . Los desplazamientos de $-\pi/2$ y π de los ángulos θ_2 y θ_5 tienen por objeto cambiar el origen de medida de los ángulos de modo que la posición cero de las variables θ_i , $i = 1, \dots, 6$, correspondan con las posiciones naturales de los brazos articulados del robot a partir de las cuales se mide el rango de variación de los movimientos.

Articulación	θ (rad)	d (m)	a (m)	α (rad)
1	θ_1	0.290	0	$-\pi/2$
2	$\theta_2 - \pi/2$	0	0.270	0
3	θ_3	0	0.070	$-\pi/2$
4	θ_4	0.302	0	$\pi/2$
5	$\theta_5 + \pi$	0	0	$\pi/2$
6	θ_6	0.072	0	0

Cuadro 3.3: Valores de los parámetros de D-H para el IRB 120

Las articulaciones del IRB 120 se han representado mediante cilindros para identificarlas como rotativas. Generalmente, se reservan los prismas para representar articulaciones prismáticas, pero en el caso del IRB 120 ninguna de ellas es de este tipo, como ya se ha explicado previamente en la sección 2.2.

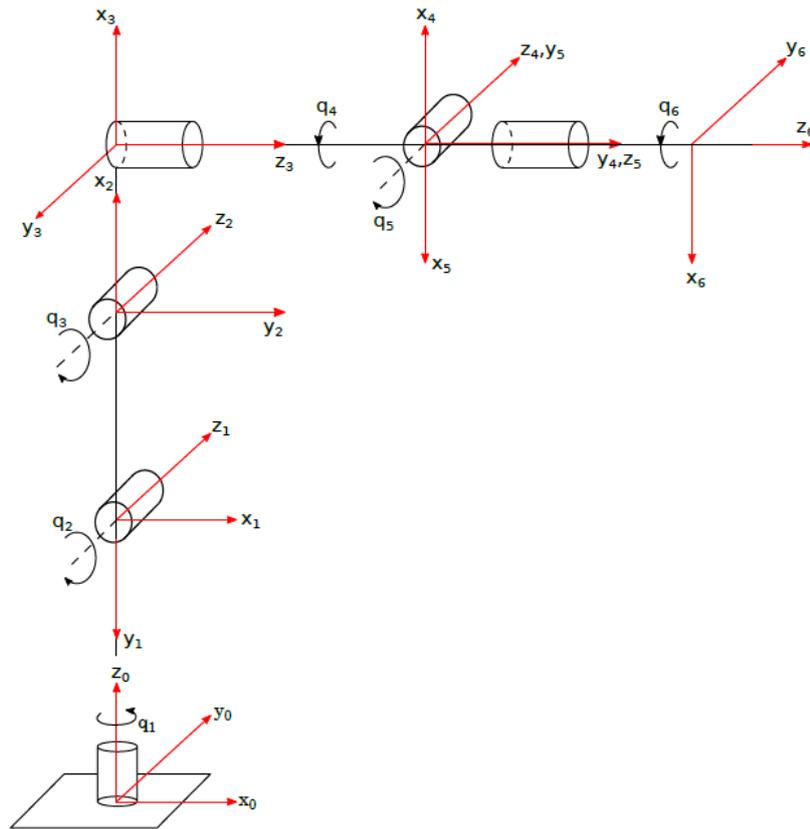


Figura 3.4: Cadena cinemática del IRB 120 según el convenio de Denavit-Hartenberg.

3.3. Parámetros cinemáticos del robot IRB 120

Las matrices de transformación homogénea A_i^{i-1} que se obtienen mediante la ecuación (3.1) para el robot IRB 120, se representan mediante la letra A , con un superíndice que representa el sistema de referencia de la articulación anterior y un subíndice que representa la articulación siguiente. Estas matrices son:

$$A_1^0 = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & \cos(\theta_1) & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad d_1 = 0.290$$

$$A_2^1 = \begin{bmatrix} \sin(\theta_2) & \cos(\theta_2) & 0 & a_2 \sin(\theta_2) \\ -\cos(\theta_2) & \sin(\theta_2) & 0 & -a_2 \cos(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad a_2 = 0.270$$

$$A_3^2 = \begin{bmatrix} \cos(\theta_3) & 0 & -\sin(\theta_3) & a_3 \cos(\theta_3) \\ \sin(\theta_3) & 0 & \cos(\theta_3) & a_3 \sin(\theta_3) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad a_3 = 0.070$$

$$A_4^3 = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad d_4 = 0.302$$

$$A_5^4 = \begin{bmatrix} -\cos(\theta_5) & 0 & \sin(\theta_5) & 0 \\ -\sin(\theta_5) & 0 & \cos(\theta_5) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6^5 = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad d_6 = 0.072$$

Todas ellas se pueden descomponer, como se ha repasado en la Sección 2.1, en una matriz de rotación $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ y un vector de traslación $\mathbf{d} = [d_x, d_y, d_z]^T \in \mathbb{R}^3$. La cinemática directa del robot IRB-120 queda completamente determinada por el producto matricial

$$T_6^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 A_6^5 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

El vector $\mathbf{p} = [p_x, p_y, p_z]^T$ es el vector de posición de la herramienta con respecto al sistema de referencia de la base, y la matriz de rotación $\mathbf{R} = (r_{ij})$ define la orientación de dicha herramienta.

Las fórmulas que dan los elementos de la matriz de rotación \mathbf{R} , y del vector de traslación \mathbf{p} en términos de los ángulos de rotación y de los parámetros de Denavit-Hartenberg se relacionan, por completitud, a continuación:

$$r_{11} = c_6 [c_5 (-c_1 s_2 c_3 c_4 - c_1 c_2 s_3 c_4 - s_1 s_4) + s_5 (c_1 s_2 s_3 - c_1 c_2 c_3)] \\ + s_6 (-c_1 s_2 c_3 s_4 - c_1 c_2 s_3 s_4 + s_1 c_4)$$

$$r_{21} = c_6 [c_5 (-s_1 s_2 c_3 c_4 - s_1 c_2 s_3 c_4 + c_1 s_4) - s_5 (-s_1 s_2 s_3 + s_1 c_2 c_3)] \\ - s_6 (s_1 s_2 c_3 s_4 + s_1 c_2 s_3 s_4 + c_1 c_4)$$

$$r_{31} = -c_6 [c_5 (c_2 c_3 c_4 - s_2 s_3 c_4) - s_5 (c_2 s_3 + s_2 c_3)] \\ + s_6 (-c_2 c_3 s_4 + s_2 s_3 s_4)$$

$$r_{12} = -s_6 [-c_5 (c_1 s_2 c_3 c_4 + c_1 c_2 s_3 c_4 + s_1 s_4) - s_5 (-c_1 s_2 s_3 + c_1 c_2 c_3)] \\ - c_6 (c_1 s_2 c_3 s_4 + c_1 c_2 s_3 s_4 - s_1 c_4)$$

$$r_{22} = -s_6 [-c_5 (s_1 s_2 c_3 c_4 + s_1 c_2 s_3 c_4 + c_1 s_4) - s_5 (-s_1 s_2 s_3 + s_1 c_2 c_3)] \\ - c_6 (s_1 s_2 c_3 s_4 + s_1 c_2 s_3 s_4 + c_1 c_4)$$

$$r_{32} = -s_6 [c_5 (-c_2 c_3 c_4 + s_2 s_3 c_4) + s_5 (c_2 s_3 + s_2 c_3)] - c_6 (c_2 c_3 s_4 - s_2 s_3 s_4)$$

$$r_{13} = s_5 (c_1 s_2 c_3 c_4 + c_1 c_2 s_3 c_4 + s_1 s_4) + c_5 (-c_1 s_2 s_3 + c_1 c_2 c_3)$$

$$r_{23} = s_5 (s_1 s_2 c_3 c_4 s_1 c_2 s_3 c_4 - c_1 s_4) + c_5 (-s_1 s_2 s_3 + s_1 c_2 c_3)$$

$$r_{33} = s_5(c_2c_3c_4 - s_2s_3c_4) - c_5(c_2s_3 + s_2c_3)$$

$$p_x = d_6s_5(c_1s_2c_3c_4 + c_1c_2s_3c_4 + s_1s_4) + d_6c_5(-c_1s_2s_3 + c_1c_2c_3) \\ - d_4(c_1s_2s_3 - c_1c_2c_3) + a_3(c_1s_2c_3 + c_1c_2s_3) + a_2c_1s_2$$

$$p_y = d_6s_5(s_1s_2c_3c_4 + s_1c_2s_3c_4 - c_1s_4) + d_6c_5(-s_1s_2s_3 + s_1c_2c_3) \\ + d_4(-s_1s_2s_3 + s_1c_2c_3) + a_3(s_1s_2c_3 + s_1c_2s_3) + a_2s_1s_2$$

$$p_z = d_6s_5(c_2c_3c_4 - s_2s_3c_4) - d_6c_5(c_2s_3 + s_2c_3) - d_4(c_2s_3 + s_2c_3) \\ + a_3(c_2c_3 - s_2s_3) + a_2s_2 + d_1$$

donde c_i y s_i denotan respectivamente el coseno y seno de θ_i , $i = 1, 2, 3, 4, 5, 6$, y los coeficientes d_1, a_2, a_3, d_4, d_6 son los parámetros de Denavit-Hartenberg definidos en el cuadro 3.3.

3.4. Parámetros dinámicos del IRB 120

Los valores de todos estos parámetros han sido obtenidos de dos fuentes bibliográficas. Por un lado, los parámetros geométricos se han extraído directamente de los planos aportados por ABB (ver Figura 3.2) en la ficha técnica del producto [2]; por otra parte, las masas, los centros de masas y las matrices de inercia respecto de los centros de masas de cada uno de los eslabones del robot IRB 120 han sido obtenidos a partir de [4]. En esta publicación, el autor explica que se ha basado en un modelo en tres dimensiones del robot IRB 120 para estimar todos estos valores utilizando el programa SolidWorks. En la bibliografía consultada es común utilizar los modelos CAD del IRB 120 que facilita la marca en su página web para calcular, mediante programas de "Dibujo Asistido por Ordenador" como el mencionado, los parámetros dinámicos de cada uno de los elementos del robot en función de la geometría y el material de los mismos en sistemas predefinidos de coordenadas.

A continuación se presentan uno a uno estos parámetros y sus valores. En primer lugar, el Cuadro 3.4 recopila las masas de cada eslabón [4] en unidades del Sistema Internacional:

Eslabón	1	2	3	4	5	6
Masa (kg)	3.36	6.80	6.22	2.00	1.30	0.11

Cuadro 3.4: Masas de cada eslabón del IRB 120.

Es importante indicar que en el modelo en tres dimensiones del IRB 120 la geometría no es idéntica a la del robot real, y además se asume que las piezas están totalmente huecas y que el conjunto tiene una distribución uniforme de la densidad. Todas estas simplificaciones alejarán en cierto grado el modelo dinámico obtenido de la respuesta dinámica del robot real.

Vectores del centro de masas según el sistema de referencia S_i :

En primer lugar se determinan los vectores que unen el origen del sistema de referencia

S_{i-1} con el centro de masas del eslabón i en coordenadas del sistema S_i (ver Figura 2.3):

$$\begin{aligned}\mathbf{r}_{0,c1} &= [0 \quad -0.240 \quad 0]^T \quad \text{m} \\ \mathbf{r}_{1,c2} &= [0.146 \quad 0 \quad 0]^T \quad \text{m} \\ \mathbf{r}_{2,c3} &= [0.024 \quad 0 \quad 0.058]^T \quad \text{m} \\ \mathbf{r}_{3,c4} &= [0 \quad 0.192 \quad 0]^T \quad \text{m} \\ \mathbf{r}_{4,c5} &= [0 \quad 0 \quad 0]^T \quad \text{m} \\ \mathbf{r}_{5,c6} &= [0 \quad 0 \quad 0.063]^T \quad \text{m}\end{aligned}$$

A continuación los vectores que unen el origen del sistema de referencia S_i con el centro de masas del eslabón i , que resultan de restar a los anteriores los vectores $\mathbf{r}_{i,i+1}$ que unen los orígenes de los sistemas de referencia S_i y S_{i+1} y que se corresponden con los vectores de traslación de las matrices de transformación homogénea A_i^{i-1} :

$$\begin{aligned}\mathbf{r}_{1,c1} &= [0 \quad 0.05 \quad 0]^T \quad \text{m} \\ \mathbf{r}_{2,c2} &= [-0.124 \quad 0 \quad 0]^T \quad \text{m} \\ \mathbf{r}_{3,c3} &= [-0.046 \quad 0 \quad 0.058]^T \quad \text{m} \\ \mathbf{r}_{4,c4} &= [0 \quad -0.11 \quad 0]^T \quad \text{m} \\ \mathbf{r}_{5,c5} &= [0 \quad 0 \quad 0]^T \quad \text{m} \\ \mathbf{r}_{6,c6} &= [0 \quad 0 \quad -0.0089]^T \quad \text{m}\end{aligned}$$

Matrices de inercia respecto del sistema de referencia de cada eslabón en el centro de masas:

$$\begin{aligned}I_1 &= \begin{bmatrix} 0.012 & 0 & 0 \\ 0 & 0.010 & 0 \\ 0 & 0 & 0.011 \end{bmatrix} \quad \text{kg m}^2 \\ I_2 &= \begin{bmatrix} 0.064 & 0 & 0 \\ 0 & 0.012 & 0 \\ 0 & 0 & 0.064 \end{bmatrix} \quad \text{kg m}^2 \\ I_3 &= \begin{bmatrix} 0.35 & 0 & 0.23 \\ 0 & 0.37 & 0 \\ 0.23 & 0 & 0.18 \end{bmatrix} \quad \text{kg m}^2 \\ I_4 &= \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.006 & 0 \\ 0 & 0 & 0.008 \end{bmatrix} \quad \text{kg m}^2 \\ I_5 &= \begin{bmatrix} 0.0023 & 0 & 0 \\ 0 & 0.0023 & 0 \\ 0 & 0 & 0.015 \end{bmatrix} \quad \text{kg m}^2 \\ I_6 &= \begin{bmatrix} 0.0004 & 0 & 0 \\ 0 & 0.0004 & 0 \\ 0 & 0 & 0.0008 \end{bmatrix} \quad \text{kg m}^2\end{aligned}$$

3.5. El modelo dinámico del robot IRB 120

De acuerdo con el método de Newton-Euler presentado en el epígrafe 2.4, el modelo dinámico del robot manipulador IRB 120 nos permite calcular los pares aplicados en cada eslabón y en cada articulación. Para desarrollar este método, se ha interpretado el robot manipulador como la cadena cinemática presentada en el epígrafe 2.2. En este apartado se presenta, inicialmente, la recursión progresiva del método y posteriormente la recursión regresiva. En ambos casos, las expresiones matriciales que se obtienen son tan complejas y tan largas que no se presentan en este texto. El lector encontrará en el Apéndice B el desarrollo del modelo dinámico completo en el software de manipulación simbólica Maple.

3.5.1. Recursión progresiva:

La recursión progresiva permite el cálculo de la cinemática lineal y angular de cada eslabón desde $i = 1$ hasta n . Una etapa previa a la ejecución de la recursión progresiva es el cálculo de los vectores unitarios \mathbf{b}_i que representan los ejes de rotación de cada articulación en coordenadas del sistema S_i . Para ello, una vez más se recurre a las matrices de rotación partiendo del vector $\mathbf{z}_0 = [0 \ 0 \ 1]^T$ y aplicando la ecuación 2.11:

$$\begin{aligned}\mathbf{b}_1 &= (R_0^1)^T \mathbf{z}_0 = [0 \ -1 \ 0]^T \quad \text{m} \\ \mathbf{b}_2 &= (R_0^2)^T R_0^1 \mathbf{z}_0 = [0 \ 0 \ 1]^T \quad \text{m} \\ \mathbf{b}_3 &= (R_0^3)^T R_0^2 \mathbf{z}_0 = [0 \ -1 \ 0]^T \quad \text{m} \\ \mathbf{b}_4 &= (R_0^4)^T R_0^3 \mathbf{z}_0 = [0 \ 1 \ 0]^T \quad \text{m} \\ \mathbf{b}_5 &= (R_0^5)^T R_0^4 \mathbf{z}_0 = [0 \ 1 \ 0]^T \quad \text{m} \\ \mathbf{b}_6 &= (R_0^6)^T R_0^5 \mathbf{z}_0 = [0 \ 0 \ 1]^T \quad \text{m}\end{aligned}$$

A continuación, se explicita eslabón a eslabón cada una de las fórmulas recursivas:

Eslabón 1:

Partiendo de las condiciones iniciales

$$\mathbf{v}_{c,0} = \boldsymbol{\omega}_0 = \dot{\boldsymbol{\omega}}_0 = [0 \ 0 \ 0]^T$$

y

$$\mathbf{a}_{c,0} = \mathbf{a}_{e,0} = [0 \ 0 \ 0]^T$$

se evalúan las expresiones (2.10) y (2.12) para las velocidades y aceleraciones angulares, que resultan:

$$\boldsymbol{\omega}_1 = \mathbf{b}_1 \dot{q}_1$$

$$\boldsymbol{\alpha}_1 = \mathbf{b}_1 \ddot{q}_1 + \boldsymbol{\omega}_1 \times \mathbf{b}_1 \dot{q}_1$$

y las expresiones (2.14) y (2.15) para las aceleraciones lineales, que en el eslabón 1 tienen la siguiente forma:

$$\mathbf{a}_{e,1} = \dot{\boldsymbol{\omega}}_1 \times \mathbf{r}_{0,1} + \boldsymbol{\omega}_1 \times (\boldsymbol{\omega}_1 \times \mathbf{r}_{0,1}) + (R_1^0)^T \mathbf{a}_{e,0}$$

$$\mathbf{a}_{c,1} = \dot{\boldsymbol{\omega}}_1 \times \mathbf{r}_{0,c1} + \boldsymbol{\omega}_1 \times (\boldsymbol{\omega}_1 \times \mathbf{r}_{0,c1}) + (R_1^0)^T \mathbf{a}_{e,0}$$

Eslabón 2:

Sustituyendo las mismas expresiones que en el eslabón 1, se obtiene para el eslabón 2:

$$\begin{aligned}\boldsymbol{\omega}_2 &= (R_2^1)^T \boldsymbol{\omega}_1 + \mathbf{b}_2 \dot{q}_2 \\ \boldsymbol{\alpha}_2 &= (R_2^1)^T \boldsymbol{\alpha}_1 + \mathbf{b}_2 \ddot{q}_2 + \boldsymbol{\omega}_2 \times (\mathbf{b}_2 \dot{q}_2) \\ \mathbf{a}_{e,2} &= \dot{\boldsymbol{\omega}}_2 \times \mathbf{r}_{1,2} + \boldsymbol{\omega}_2 \times (\boldsymbol{\omega}_2 \times \mathbf{r}_{1,2}) + (R_2^1)^T \mathbf{a}_{e,1} \\ \mathbf{a}_{c,2} &= \dot{\boldsymbol{\omega}}_2 \times \mathbf{r}_{1,c2} + \boldsymbol{\omega}_2 \times (\boldsymbol{\omega}_2 \times \mathbf{r}_{1,c2}) + (R_2^1)^T \mathbf{a}_{e,1}\end{aligned}$$

Eslabón 3:

Procediendo de la misma forma, el eslabón 3 resulta:

$$\begin{aligned}\boldsymbol{\omega}_3 &= (R_3^2)^T \boldsymbol{\omega}_2 + \mathbf{b}_3 \dot{q}_3 \\ \boldsymbol{\alpha}_3 &= (R_3^2)^T \boldsymbol{\alpha}_2 + \mathbf{b}_3 \ddot{q}_3 + \boldsymbol{\omega}_3 \times (\mathbf{b}_3 \dot{q}_3) \\ \mathbf{a}_{e,3} &= \dot{\boldsymbol{\omega}}_3 \times \mathbf{r}_{2,3} + \boldsymbol{\omega}_3 \times (\boldsymbol{\omega}_3 \times \mathbf{r}_{2,3}) + (R_3^2)^T \mathbf{a}_{e,2} \\ \mathbf{a}_{c,3} &= \dot{\boldsymbol{\omega}}_3 \times \mathbf{r}_{2,c3} + \boldsymbol{\omega}_3 \times (\boldsymbol{\omega}_3 \times \mathbf{r}_{2,c3}) + (R_3^2)^T \mathbf{a}_{e,2}\end{aligned}$$

Eslabón 4:

Para el cuarto eslabón:

$$\begin{aligned}\boldsymbol{\omega}_4 &= (R_4^3)^T \boldsymbol{\omega}_3 + \mathbf{b}_4 \dot{q}_4 \\ \boldsymbol{\alpha}_4 &= (R_4^3)^T \boldsymbol{\alpha}_3 + \mathbf{b}_4 \ddot{q}_4 + \boldsymbol{\omega}_4 \times (\mathbf{b}_4 \dot{q}_4) \\ \mathbf{a}_{e,4} &= \dot{\boldsymbol{\omega}}_4 \times \mathbf{r}_{3,4} + \boldsymbol{\omega}_4 \times (\boldsymbol{\omega}_4 \times \mathbf{r}_{3,4}) + (R_4^3)^T \mathbf{a}_{e,3} \\ \mathbf{a}_{c,4} &= \dot{\boldsymbol{\omega}}_4 \times \mathbf{r}_{3,c4} + \boldsymbol{\omega}_4 \times (\boldsymbol{\omega}_4 \times \mathbf{r}_{3,c4}) + (R_4^3)^T \mathbf{a}_{e,3}\end{aligned}$$

Eslabón 5:

En el quinto eslabón, las expresiones de las aceleraciones lineales se simplifican puesto que los vectores $\mathbf{r}_{4,c5}$ y $\mathbf{r}_{5,c5}$ son nulos:

$$\begin{aligned}\boldsymbol{\omega}_5 &= (R_5^4)^T \boldsymbol{\omega}_4 + \mathbf{b}_5 \dot{q}_5 \\ \boldsymbol{\alpha}_5 &= (R_5^4)^T \boldsymbol{\alpha}_4 + \mathbf{b}_5 \ddot{q}_5 + \boldsymbol{\omega}_5 \times (\mathbf{b}_5 \dot{q}_5) \\ \mathbf{a}_{e,5} &= (R_5^4)^T \mathbf{a}_{e,4} \\ \mathbf{a}_{c,5} &= (R_5^4)^T \mathbf{a}_{e,4}\end{aligned}$$

Eslabón 6:

Por último, en el eslabón 6 no es necesario evaluar la aceleración lineal de su extremo $\mathbf{a}_{e,6}$, puesto que esta se utiliza únicamente para obtener $\mathbf{a}_{e,i+1}$, y en este caso ya no hay un séptimo eslabón.

$$\begin{aligned}\boldsymbol{\omega}_6 &= (R_6^5)^T \boldsymbol{\omega}_5 + \mathbf{b}_6 \dot{q}_6 \\ \boldsymbol{\alpha}_6 &= (R_6^5)^T \boldsymbol{\alpha}_5 + \mathbf{b}_6 \ddot{q}_6 + \boldsymbol{\omega}_6 \times (\mathbf{b}_6 \dot{q}_6) \\ \mathbf{a}_{e,6} &= \dot{\boldsymbol{\omega}}_6 \times \mathbf{r}_{5,c6} + \boldsymbol{\omega}_6 \times (\boldsymbol{\omega}_6 \times \mathbf{r}_{5,c6}) + (R_6^5)^T \mathbf{a}_{e,5}\end{aligned}$$

3.5.2. Recursión regresiva:

La recursión regresiva permite obtener las fuerzas y los pares ejercidos sobre cada eslabón y, de este modo, conseguimos los pares de las articulaciones, que son los que deben ejercer los motores para generar el movimiento deseado del manipulador.

Las condiciones terminales, correspondientes al par externo y la fuerza externa aplicados sobre el eslabón 6, tomarán valor nulo si no hay ninguna herramienta montada en el extremo o si el robot manipulador no está ejerciendo con su movimiento una oposición a un elemento externo. Pero también podrán tomar valor no nulo si se deben tener en cuenta dentro de la simulación cualquiera de estos factores. En este desarrollo, consideramos

$$\boldsymbol{\tau}^{ext} = \mathbf{f}^{ext} = 0$$

Eslabón 6:

Partiendo de las condiciones de par y fuerza externos nulos, las expresiones de \mathbf{f}_6 y $\boldsymbol{\tau}_6$, que ya fueron discutidas en la sección 2.4 y se corresponden con (2.8) y (2.9), son:

$$\begin{aligned}\mathbf{g}_6 &= (R_6^0)^T \mathbf{g}_0 \\ \mathbf{f}_6 &= m_6 \mathbf{a}_{c,6} - m_6 \mathbf{g}_6 \\ \boldsymbol{\tau}_6 &= -\mathbf{f}_6 \times \mathbf{r}_{5,c6} + I_6 \boldsymbol{\alpha}_6 + \boldsymbol{\omega}_6 \times (I_6 \boldsymbol{\omega}_6)\end{aligned}$$

donde U representa la matriz identidad de 3×3 que equivaldría a R_7^6 ; y en $\mathbf{a}_{c,6}$ se cuenta con la acción de la gravedad.

Eslabón 5:

En el caso del eslabón 5, se trata únicamente de sustituir las mismas expresiones:

$$\begin{aligned}\mathbf{g}_5 &= (R_5^0)^T \mathbf{g}_0 \\ \mathbf{f}_5 &= m_5 \mathbf{a}_{c,5} - m_5 \mathbf{g}_5 + R_6^5 \mathbf{f}_6 \\ \boldsymbol{\tau}_5 &= -\mathbf{f}_5 \times \mathbf{r}_{4,c5} + I_5 \boldsymbol{\alpha}_5 + \boldsymbol{\omega}_5 \times (I_5 \boldsymbol{\omega}_5) + R_6^5 \boldsymbol{\tau}_6\end{aligned}$$

Eslabón 4:

Para el eslabón 4, se aplica:

$$\begin{aligned}\mathbf{g}_4 &= (R_4^0)^T \mathbf{g}_0 \\ \mathbf{f}_4 &= m_4 \mathbf{a}_{c,4} - m_4 \mathbf{g}_4 + R_5^4 \mathbf{f}_5 \\ \boldsymbol{\tau}_4 &= -\mathbf{f}_4 \times \mathbf{r}_{3,c4} + I_4 \boldsymbol{\alpha}_4 + \boldsymbol{\omega}_4 \times (I_4 \boldsymbol{\omega}_4) + R_5^4 \boldsymbol{\tau}_5 + (R_5^4 \mathbf{f}_5) \times \mathbf{r}_{4,c4}\end{aligned}$$

Eslabón 3:

Para el eslabón 3:

$$\begin{aligned}\mathbf{g}_3 &= (R_3^0)^T \mathbf{g}_0 \\ \mathbf{f}_3 &= m_3 \mathbf{a}_{c,3} - m_3 \mathbf{g}_3 + R_4^3 \mathbf{f}_4 \\ \boldsymbol{\tau}_3 &= -\mathbf{f}_3 \times \mathbf{r}_{2,c3} + I_3 \boldsymbol{\alpha}_3 + \boldsymbol{\omega}_3 \times (I_3 \boldsymbol{\omega}_3) + R_4^3 \boldsymbol{\tau}_4 + (R_4^3 \mathbf{f}_4) \times \mathbf{r}_{3,c3}\end{aligned}$$

Eslabón 2:

Para el eslabón 2:

$$\begin{aligned}\mathbf{g}_2 &= (R_2^0)^T \mathbf{g}_0 \\ \mathbf{f}_2 &= m_2 \mathbf{a}_{c,2} - m_2 \mathbf{g}_2 + R_3^2 \mathbf{f}_3 \\ \boldsymbol{\tau}_2 &= -\mathbf{f}_2 \times \mathbf{r}_{1,c2} + I_2 \boldsymbol{\alpha}_2 + \boldsymbol{\omega}_2 \times (I_2 \boldsymbol{\omega}_2) + R_3^2 \boldsymbol{\tau}_3 + (R_3^2 \mathbf{f}_3) \times \mathbf{r}_{2,c2}\end{aligned}$$

Eslabón 1:

Por último, el eslabón 1 será:

$$\begin{aligned}\mathbf{g}_1 &= (R_1^0)^T \mathbf{g}_0 \\ \mathbf{f}_1 &= m_1 \mathbf{a}_{c,1} - m_1 \mathbf{g}_1 + R_2^1 \mathbf{f}_2 \\ \boldsymbol{\tau}_1 &= -\mathbf{f}_1 \times \mathbf{r}_{0,c1} + I_1 \boldsymbol{\alpha}_1 + \boldsymbol{\omega}_1 \times (I_1 \boldsymbol{\omega}_1) + R_2^1 \boldsymbol{\tau}_2 + (R_2^1 \mathbf{f}_2) \times \mathbf{r}_{1,c2}\end{aligned}$$

Capítulo 4

Simulaciones y resultados

En este capítulo se explica la estructura del modelo elaborado en el entorno de Matlab Simulink del robot manipulador IRB 120. En base a ese modelo, se han realizado diversas simulaciones con control en lazo abierto y en lazo cerrado con el objetivo de probar su validez. Todo el modelo está basado en el dominio articular según la ecuación ya analizada en capítulos anteriores:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (4.1)$$

Las diferentes estrategias de control que se han utilizado en este modelo permiten controlar la posición (*set-point controllers*) —estableciendo una posición inicial y una posición final— ajustando las variables articulares $\mathbf{q}(t)$ y sus derivadas primera y segunda para alcanzarla. Se ha considerado que el robot manipulador está provisto de motores ideales, cuya dinámica es despreciable. Tampoco se han tenido en cuenta posibles efectos viscosos. En términos formales y siguiendo el argumento del capítulo 8 de [15], el objetivo del control de la posición consiste en encontrar un par $\boldsymbol{\tau}$ tal que:

$$\lim_{t \rightarrow \infty} \mathbf{q}(t) = \mathbf{q}_d$$

donde $\mathbf{q}_d \in \mathbb{R}^n$ es un vector constante que representa la posición deseada de las articulaciones. La forma de evaluar si un controlador consigue el objetivo es estudiar la estabilidad asintótica en el origen del sistema en lazo cerrado según Lyapunov. Para ello, se considera que el objetivo del control de la posición es:

$$\lim_{t \rightarrow \infty} \tilde{\mathbf{q}}(t) = 0$$

donde $\tilde{\mathbf{q}} \in \mathbb{R}^n$ es un vector que representa el error de posición de cada articulación. Este error se define como:

$$\tilde{\mathbf{q}}(t) := \mathbf{q}_d - \mathbf{q}(t)$$

En primer lugar, en la sección 4.1 se presentan los parámetros bajo los que se ha construido el modelo y se han ejecutado todas las simulaciones.

En la sección 4.2 se explica el modelo en lazo abierto con control por posición deseada y se presentan los resultados de dos simulaciones distintas. Como se observará en esta sección, el lazo abierto no permite el control estable de la respuesta del sistema, que sigue trayectorias erráticas sin converger al valor deseado.

En la sección 4.3 se analizan los resultados de un modelo basado en un controlador proporcional con realimentación de la velocidad.

En la sección 4.4 se presenta un modelo en lazo cerrado con un controlador Proporcional-Derivativo basado en la posición deseada. En este caso sí se consigue una salida estable, como se verá en los resultados de las simulaciones ejecutadas con este modelo. Sin embargo, siguiendo lo expuesto en el capítulo 7 de [15], un control PD no garantiza alcanzar el objetivo de posición cuando el modelo dinámico contempla la acción de la gravedad sobre los eslabones, a no ser que la posición final o deseada sea tal que $\mathbf{g}(\mathbf{q}_d) = 0$.

Por último, se ha probado a implementar un controlador PID al robot manipulador IRB 120. En la sección 4.5 se analizan los resultados de las simulaciones llevadas a cabo con este tipo de control. El ajuste de los parámetros de una PID no es trivial como en el caso de un controlador PD —donde basta con que las matrices K_p y K_d sean definidas positivas— pues depende directamente de la matriz de inercia y la matriz de gravedad del sistema. Un método para ajustar los parámetros es presentado en la literatura (capítulo 9 de [15]).

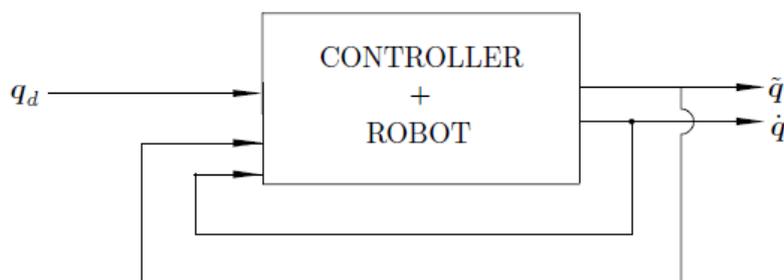


Figura 4.1: Representación simple de las entradas y salidas de un controlador de posición en lazo cerrado.

4.1. Estructura del modelo y de las simulaciones en Simulink

La dinámica del robot IRB 120 se encapsula en los modelos Simulink mediante dos bloques `Dinamica_inversa` y `Dinamica_directa`, con puertas de entrada y salida múltiples, que internamente incluyen funciones Matlab que evalúan los diferentes términos de la ecuación (4.1).

Las puertas de entrada del bloque `Dinamica_directa` corresponden a los valores iniciales $\mathbf{q}(0)$ y $\dot{\mathbf{q}}(0)$ y al vector de pares en las articulaciones $\boldsymbol{\tau}$. Internamente el bloque evalúa mediante una función Matlab `function ddq=din_inv(tau,q,dq)` el segundo miembro de

$$\ddot{\mathbf{q}} = M(\mathbf{q})^{-1} (\boldsymbol{\tau} - C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})),$$

para seguidamente invocar el bloque de Simulink `integrador de segundo orden`. Las puertas de salida del bloque proporcionan los valores de $\mathbf{q}(t)$ y $\dot{\mathbf{q}}(t)$ que se obtienen al integrar las ecuaciones diferenciales del modelo. La Figura 4.2 muestra el diagrama de bloques con la función Matlab implementada

El bloque `Dinamica_inversa` toma en las puertas de entrada los valores que corresponden $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$ y $\ddot{\mathbf{q}}(t)$, y proporciona en la puerta de salida el valor del par en las articulaciones $\boldsymbol{\tau}(t)$, dado por

$$\boldsymbol{\tau} = M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})$$

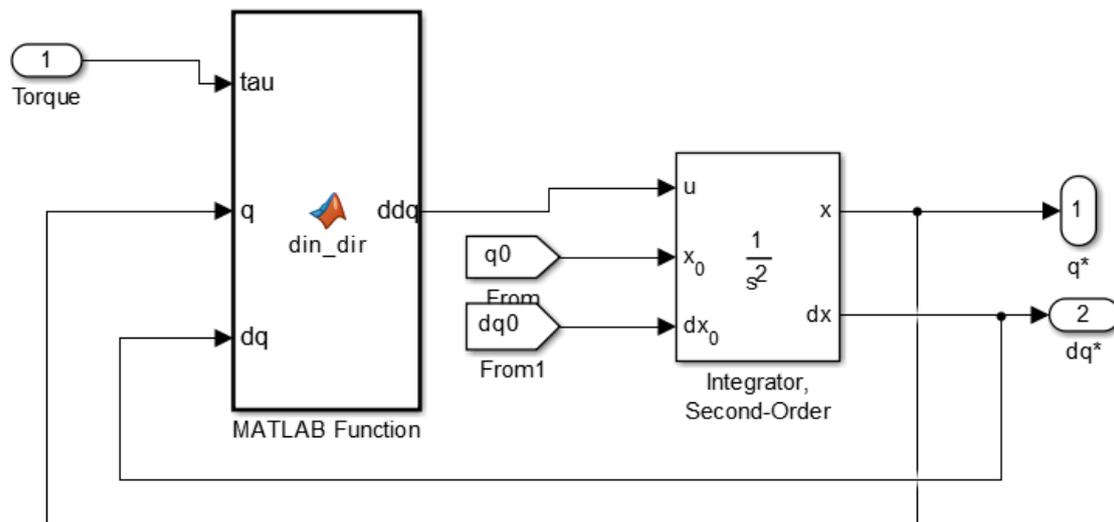


Figura 4.2: Diagrama de bloques de la dinámica directa.

Internamente el bloque contiene una función Matlab function $CG=cori_grav(q,dq)$ que evalúa la contribución de los vectores de Coriolis $C(q,\dot{q})\dot{q}$ y gravitatorio $g(q)$, y una función Matlab function $Mq=Inercia_matrix(q,ddq)$ que evalúa la contribución de los términos inerciales $M(q)\ddot{q}(t)$. El diagrama de bloques asociado se presenta en la figura 4.3

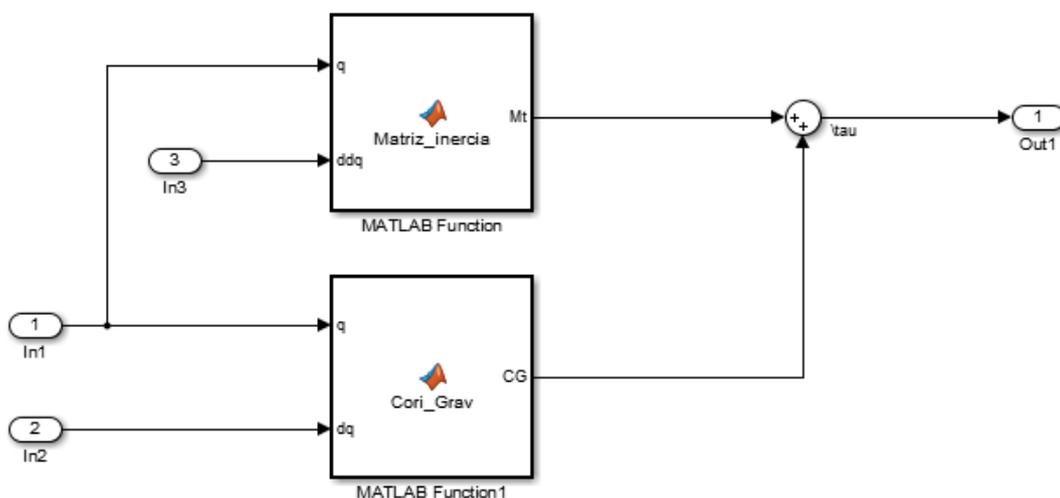


Figura 4.3: Diagrama de bloques de la dinámica inversa.

Todos los parámetros del entorno de Simulink están predefinidos para tomar su valor del entorno de trabajo de Matlab. Por ello, para agilizar el proceso, los parámetros principales de cada simulación se establecen ejecutando previamente un programa `Parametros_workspace_IRB120_model_PID.m` desde la ventana de comandos de Matlab.

En el Cuadro 4.1 se recogen los parámetros comunes del algoritmo de integración de las ecuaciones diferenciales de todas las simulaciones:

Tipo de "solver"	ode45 (Dormand-Prince)
Tolerancia relativa	0.001
Tolerancia absoluta	Automática
Tamaño máximo de paso	Automático
Tamaño mínimo de paso	Automático
Tamaño de paso inicial	Automático

Cuadro 4.1: Parámetros del algoritmo "solver" utilizado en las simulaciones

4.2. Simulación en lazo abierto

Simulación 1:

En esta primera simulación, se ha definido el modelo de la Figura 4.4. Se alimenta con dos vectores $\in \mathbb{R}^6$ correspondientes a la posición inicial q_0 y a la velocidad inicial \dot{q}_0 :

$$\begin{aligned} \mathbf{q}_d &= [0 \ 0 \ 0 \ 0 \ 0 \ 0] \\ \mathbf{q}_0 &= [\pi/6 \ \pi/12 \ -\pi/12 \ \pi/4 \ -\pi/6 \ \pi/2] \\ \dot{\mathbf{q}}_0 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0] \end{aligned}$$

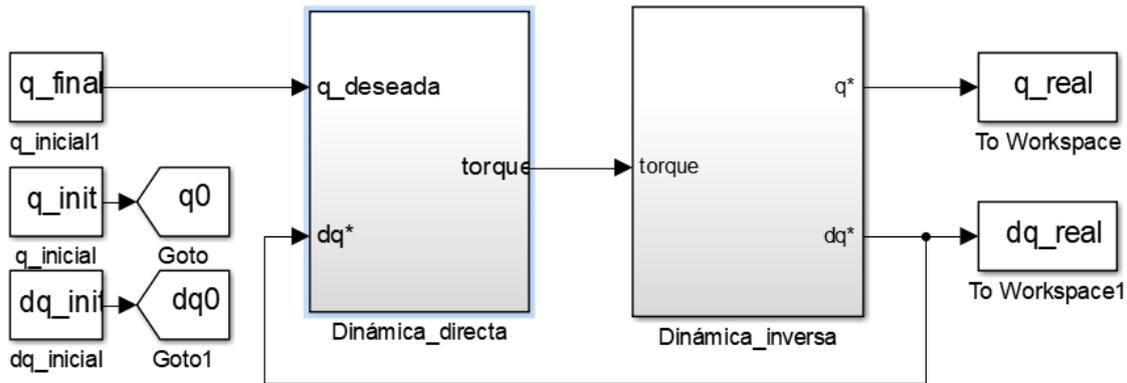


Figura 4.4: Esquema del modelo en Simulink con control en lazo abierto.

En primer lugar, se alimenta el bloque de dinámica inversa con el valor de referencia para la posición \mathbf{q}_d . Este bloque resuelve las matrices de inercia, de Coriolis y de gravedad para hallar el par correspondiente en cada iteración $\boldsymbol{\tau}(t)$. El bloque de dinámica directa se alimenta con el par calculado $\boldsymbol{\tau}$ y se realimentan la posición $\mathbf{q}(t)$ y la velocidad $\dot{\mathbf{q}}(t)$. Este bloque, conocidas estas variables, resuelve la ecuación de la dinámica del robot manipulador IRB 120 despejando $\ddot{\mathbf{q}}(t)$ y calculando las matrices de inercia, Coriolis y gravedad:

$$\ddot{\mathbf{q}}(t) = M(\mathbf{q})^{-1}[\boldsymbol{\tau}(t) - C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})]$$

Como se observa en la Figura 4.5, la salida del modelo devuelve valores totalmente descontrolados. Partiendo de un vector inicial $\mathbf{q}_0 = [\pi/6 \ \pi/12 \ -\pi/12 \ \pi/4 \ -\pi/6 \ \pi/2]$,

el modelo debería hacer tender la salida hacia los valores de referencia, que es el vector nulo $[0 \ 0 \ 0 \ 0 \ 0]$; sin embargo, se observa que la evolución tanto de la velocidad como de la posición son arbitrarias.

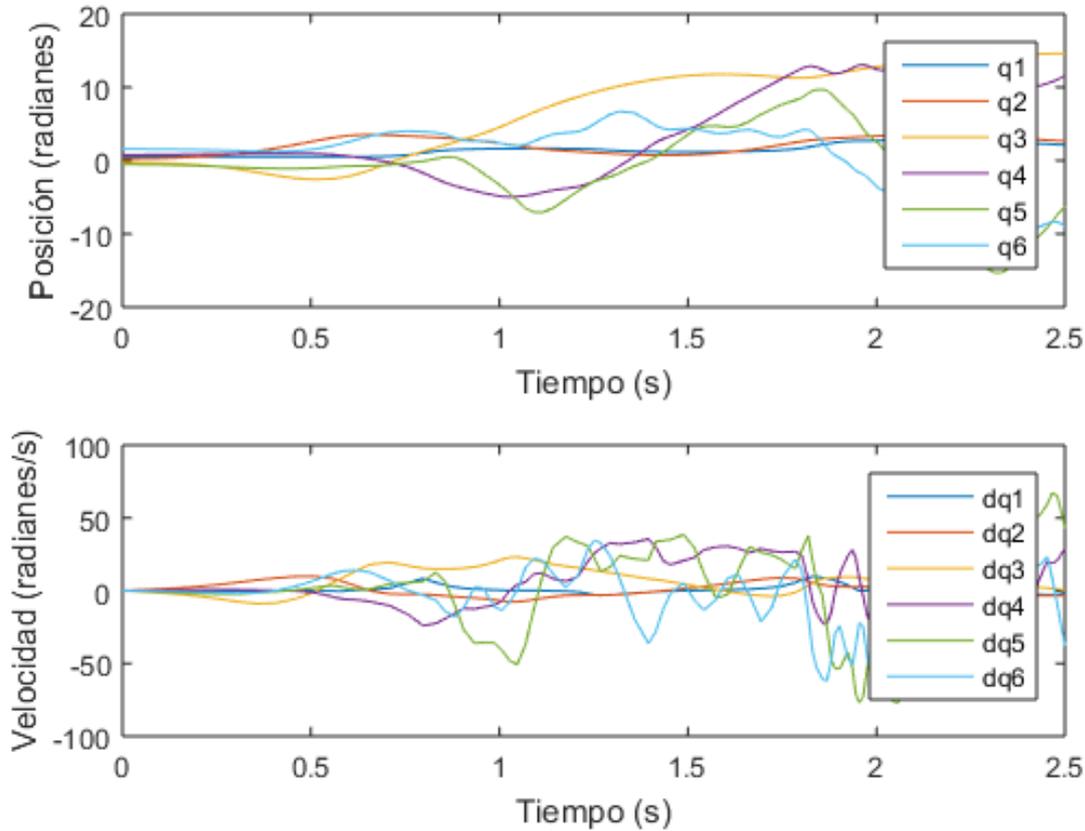


Figura 4.5: Resultados de la simulación en lazo abierto: posición y velocidad angulares.

Dada la arbitrariedad de la salida, la simulación en lazo abierto no nos permite, por sí sola, verificar la validez del modelo dinámico del robot IRB 120 que se ha diseñado. Sin embargo, en conjunto con el resto de simulaciones que se presentan en este capítulo, nos permite descartar el lazo abierto como una estrategia de control contemplable en el diseño de control de un robot manipulador cualquiera.

4.3. Simulación en lazo cerrado con control proporcional y realimentación de la velocidad

La siguiente estrategia de control que se presenta en este apartado consiste en un modelo en lazo cerrado con control proporcional y realimentación de la velocidad angular. Se trata del controlador de lazo cerrado más simple que puede utilizarse para el control de robots manipuladores y también es conocido como "*Controlador proporcional con realimentación tacométrica*". Su aplicación es común en el control de la posición angular de motores de corriente continua. La ecuación de este controlador resulta:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = K_p\tilde{\mathbf{q}} - K_d\dot{\mathbf{q}}$$

donde K_p y $K_d \in \mathbb{R}^{6 \times 6}$ son matrices simétricas definidas positivas que corresponden a la ganancia de posición y a la ganancia de velocidad respectivamente [15]. Los valores de estas

ganancias deben ser preestablecidos experimentalmente según el tipo de respuesta que se desee —amortiguada, sub-amortiguada, sobre-amortiguada— así como a la duración posible del transitorio. El vector $\tilde{\mathbf{q}} = \mathbf{q}_{final} - \mathbf{q} \in \mathbb{R}^6$ representa el error de posición. La llamada \mathbf{q}_{final} es la posición deseada o de referencia.

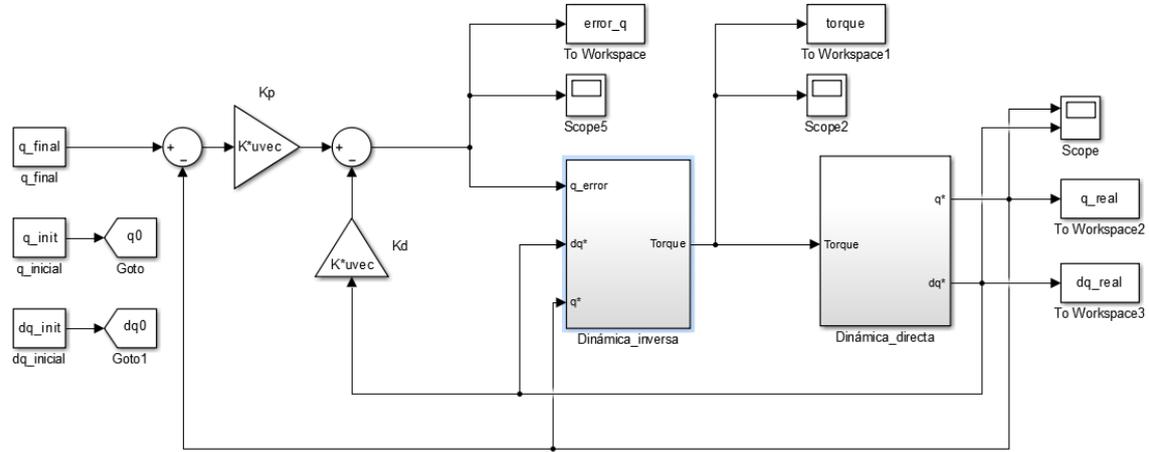


Figura 4.6: Esquema del modelo en Simulink con control proporcional y realimentación de la velocidad.

Las matrices de ganancias que se han utilizado en las simulaciones con lazo cerrado se muestran a continuación. Los valores de estas matrices diagonales definidas positivas se han escogido así buscando una respuesta amortiguada y tratando de conservar, al mismo tiempo, una curva de pares lo más estable posible.

$$K_p = \begin{bmatrix} 30 & 0 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 & 0 & 0 \\ 0 & 0 & 30 & 0 & 0 & 0 \\ 0 & 0 & 0 & 30 & 0 & 0 \\ 0 & 0 & 0 & 0 & 30 & 0 \\ 0 & 0 & 0 & 0 & 0 & 30 \end{bmatrix} \quad (4.2)$$

$$K_d = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 \end{bmatrix} \quad (4.3)$$

Simulación 2:

En la primera simulación se han considerado los valores siguientes de $\mathbf{q}_{final} = \mathbf{q}_{deseada}$ y \mathbf{q}_0 :

$$\mathbf{q}_0 = [\pi/6 \quad \pi/12 \quad -\pi/12 \quad \pi/4 \quad -\pi/6 \quad \pi/2]$$

$$\mathbf{q}_{final} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

Los resultados de evolución del error, de la posición y de la velocidad se muestran en las Figuras 4.7 y 4.8. La curva de evolución de los pares aparece representada en

la Figura 4.9. Con esta estrategia de control se logra una salida bien controlada que responde correctamente a los valores de referencia: el error $\mathbf{e}(t)$ es asintóticamente estable en el origen en un tiempo inferior a los dos segundos.

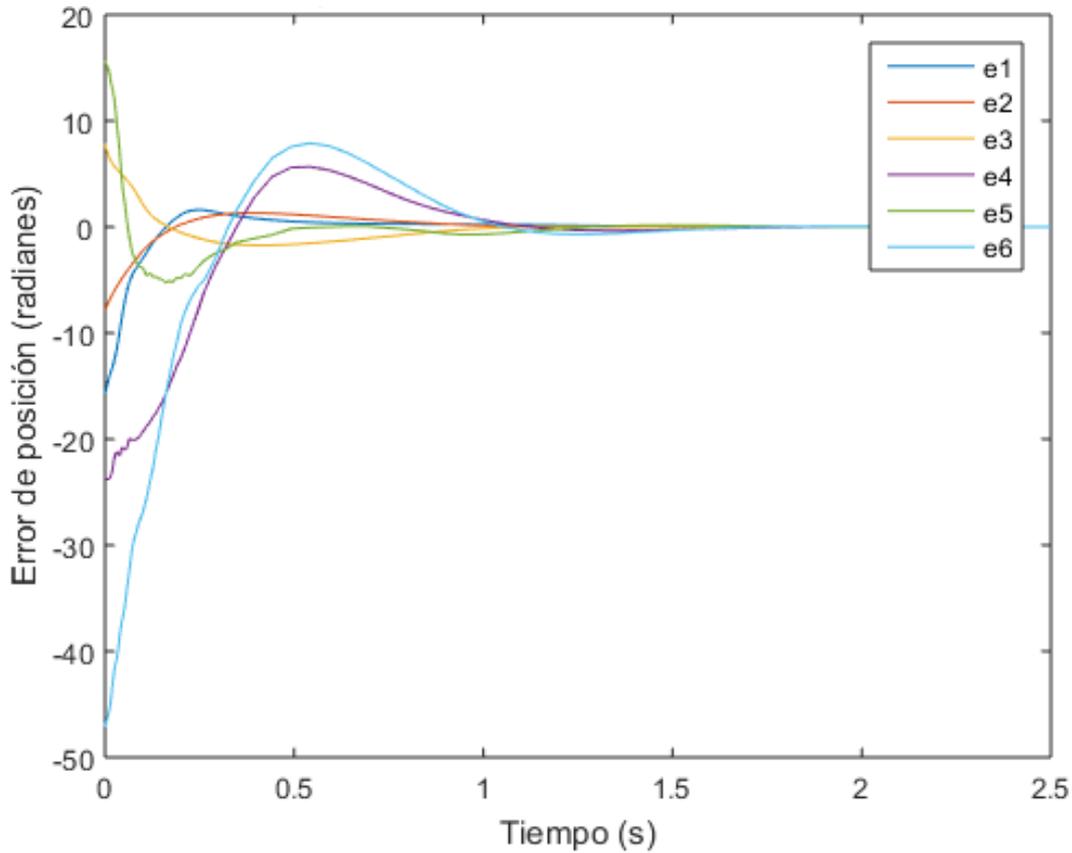


Figura 4.7: Simulación 2: evolución de $\mathbf{e}(t)$ en LC con control proporcional y realimentación de la velocidad.

4.4. Simulación en lazo cerrado con control PD

El control Proporcional-Derivativo es una extensión del control en lazo cerrado presentado en la sección 4.3. En este caso, la señal de error se compone del término proporcional (el error en la posición) y del término derivativo (el error en la velocidad). La ecuación que rige el comportamiento de este método de control es:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = K_p\tilde{\mathbf{q}} + K_d\dot{\tilde{\mathbf{q}}}$$

donde

$$\tilde{\mathbf{q}} = (\mathbf{q}_{final} - \mathbf{q}) \in \mathbb{R}^6$$

$$\dot{\tilde{\mathbf{q}}} = (\dot{\mathbf{q}}_{final} - \dot{\mathbf{q}}) \in \mathbb{R}^6$$

En la simulación ejecutada con este método de control se han utilizado las matrices de ganancias (4.2) y (4.3).

El esquema de este modelo se presenta en la figura 4.10.

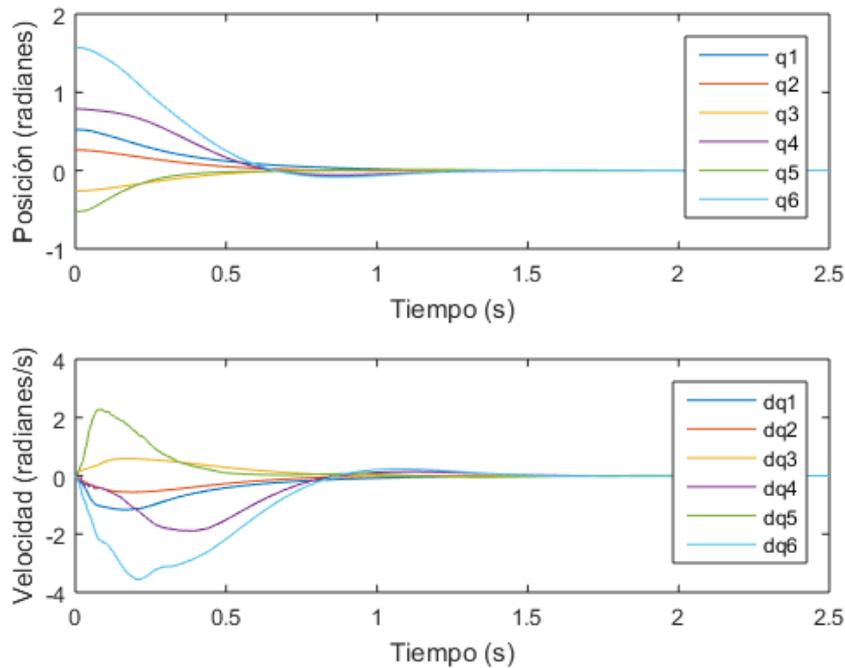


Figura 4.8: Simulación 2: evolución de \mathbf{q} y $\dot{\mathbf{q}}$ en LC con control proporcional y realimentación de la velocidad.

Simulación 3:

Los parámetros iniciales que se han establecido en esta simulación son idénticos a los de la Simulación 4.3 para poder comparar la evolución del error entre ambos métodos de control.

$$\mathbf{q}_0 = [\pi/6 \quad \pi/12 \quad -\pi/12 \quad \pi/4 \quad -\pi/6 \quad \pi/2]$$

$$\mathbf{q}_{final} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

Con el controlador PD se obtienen errores iniciales más reducidos y unas curvas más amortiguadas. En la Figura 4.11 se muestra la evolución del error en cada articulación (en radianes) y en la Figura 4.12 se han representado las trayectorias y velocidades de las articulaciones en radianes y en radianes/s.

En la gráfica 4.12 se observa que las curvas que representan la evolución de los pares en las articulaciones 1, 2 y 3 son mucho más prominentes que las de las articulaciones 4, 5 y 6, cuyos valores son inferiores a 1 nm inicialmente. Esto es debido a múltiples factores de carácter mecánico:

1. Las masas y las inercias de los eslabones 1, 2 y 3 son mucho mayores que las de los eslabones 4, 5 y 6.
2. En una cadena cinemática como la de un robot manipulador, los eslabones inferiores soportan el peso de todos los eslabones conectados encima y se ven afectados por los efectos inerciales y de Coriolis de éstos.

Por otra parte, los pares finales de τ_2 y τ_3 son distintos de 0 porque en el estado de reposo final —una vez alcanzados los valores de referencia o \mathbf{q}_{final} — los efectos de la

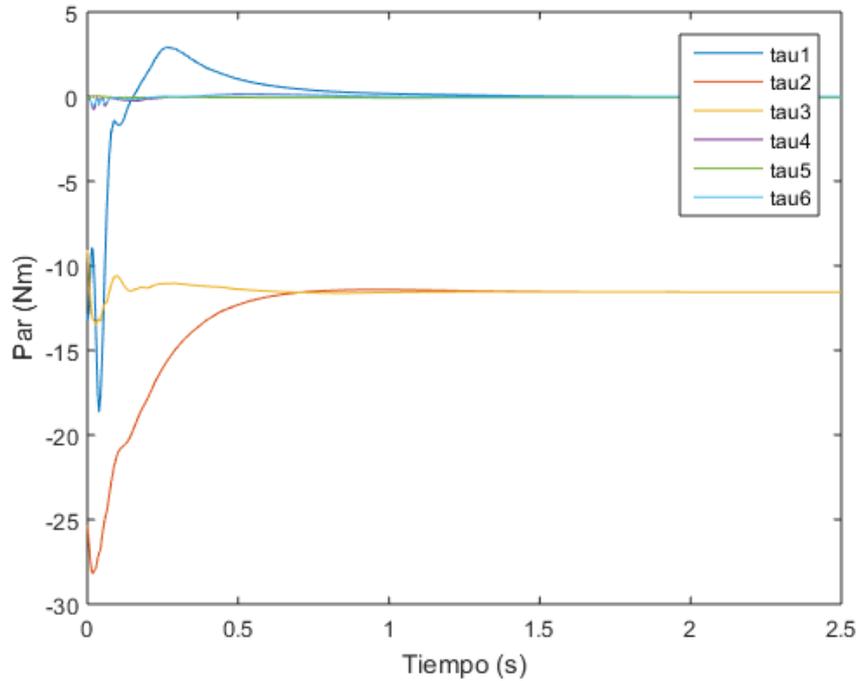


Figura 4.9: Simulación 2: evolución de τ en LC con control proporcional y realimentación de la velocidad.

gravedad sobre todos los eslabones superiores se cargan sobre las articulaciones 2 y 3. La articulación 5 también soportará el peso de los eslabones 5 y 6, pero como las masas de éstos son muy pequeñas, el par es despreciable y no se distingue en la gráfica.

4.5. Simulación en lazo cerrado con control PID

En las estrategias de control anteriores, basadas en controladores PD, el ajuste de las matrices de ganancias K_p y K_d es trivial, pues la única condición necesaria para conseguir el objetivo de control es que sean simétricas y definidas positivas. El controlador PID es el más común en los robots manipuladores comerciales. Mediante la introducción del componente integral, se logra llevar el error de posición a cero en modelos dinámicos en los que se incluye el término de gravedad, algo que con un controlador PD es teóricamente imposible sin un método de compensación de la gravedad [15].

La ecuación que rige el control PID es:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = K_p\tilde{\mathbf{q}} + K_d\dot{\tilde{\mathbf{q}}} + \mathbf{K}_i\tilde{\mathbf{e}}$$

donde $\dot{\tilde{\mathbf{e}}} = \dot{\tilde{\mathbf{q}}}$, con $\tilde{\mathbf{e}}(0) = \mathbf{0}$.

En el capítulo 9 de [15] se explica con detalle un procedimiento de ajuste de las matrices de ganancias K_p , K_d y K_i , derivado del análisis de la estabilidad asintótica del sistema mediante el método directo de Lyapunov. En este trabajo de fin de máster no se ha querido entrar al detalle teórico de este método, pero a continuación se presenta el procedimiento que permite ajustar el controlador PID con el fin de lograr localmente el control de la posición.

Este procedimiento se expresa en función de los autovalores de las matrices de ganancias K_p , K_d y K_i . Las tres condiciones que se deben satisfacer son:

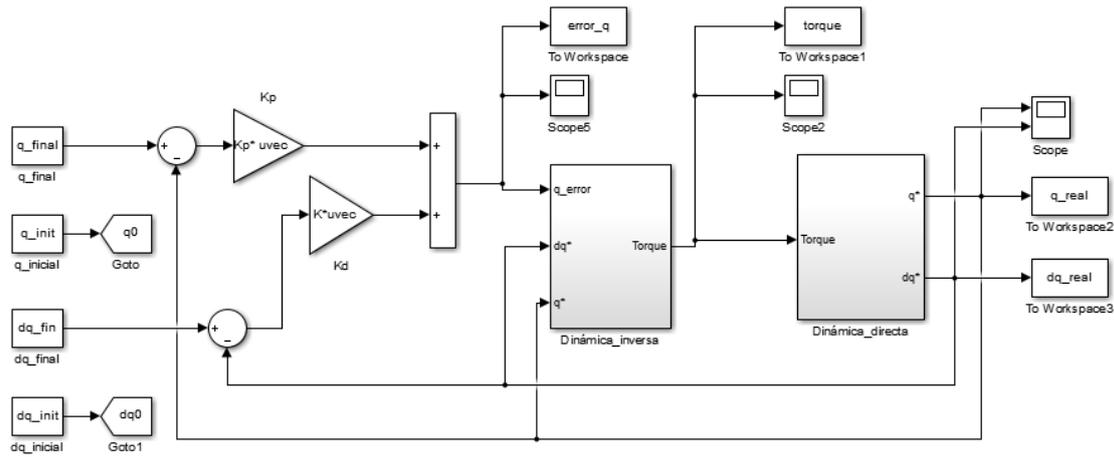


Figura 4.10: Esquema del modelo en Simulink con control PD en lazo cerrado.

$$\lambda_{Max}\{K_i\} \geq \lambda_{min}\{K_i\} > 0$$

$$\lambda_{Max}\{K_p\} \geq \lambda_{min}\{K_p\} > k_g$$

$$\lambda_{Max}\{K_d\} \geq \lambda_{min}\{K_d\} > \frac{\lambda_{Max}\{K_i\}}{\lambda_{min}\{K_p\} - k_g} \cdot \frac{\lambda_{Max}^2\{M\}}{\lambda_{min}\{M\}}$$

donde M es la matriz de inercias del modelo dinámico del IRB 120 y k_g debe satisfacer:

$$k_g \geq \left\| \frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} \right\| \quad \forall \mathbf{q} \in \mathbb{R}^6$$

Dado que en la función $\mathbf{g}(\mathbf{q})$, las componentes de \mathbf{q} aparecen siempre como argumento de funciones trigonométricas de senos y cosenos, podemos considerar que el valor máximo de las derivadas de estas funciones será 1, y de esta forma calcular el valor de k_g . Mediante la función `eig()` de Matlab (para el cálculo de autovalores y autovectores de una matriz) se ha obtenido $k_g = 11.4154$ —que corresponde con el autovalor máximo de la matriz de derivadas de $\mathbf{g}(\mathbf{q})$ respecto de las componentes de \mathbf{q} .

Los autovalores de la matriz de inercia M son:

$$eig(M) = [2.3080 \quad 0.8829 \quad 0.2347 \quad 0.0191 \quad 0.0031 \quad 0.0037]$$

y por tanto, su autovalor máximo es $\lambda_{Max}\{M\} = 2.3080$ y su autovalor mínimo $\lambda_{min}\{M\} = 0.0031$.

Con el valor de k_g e imponiendo las condiciones del procedimiento de ajuste explicado

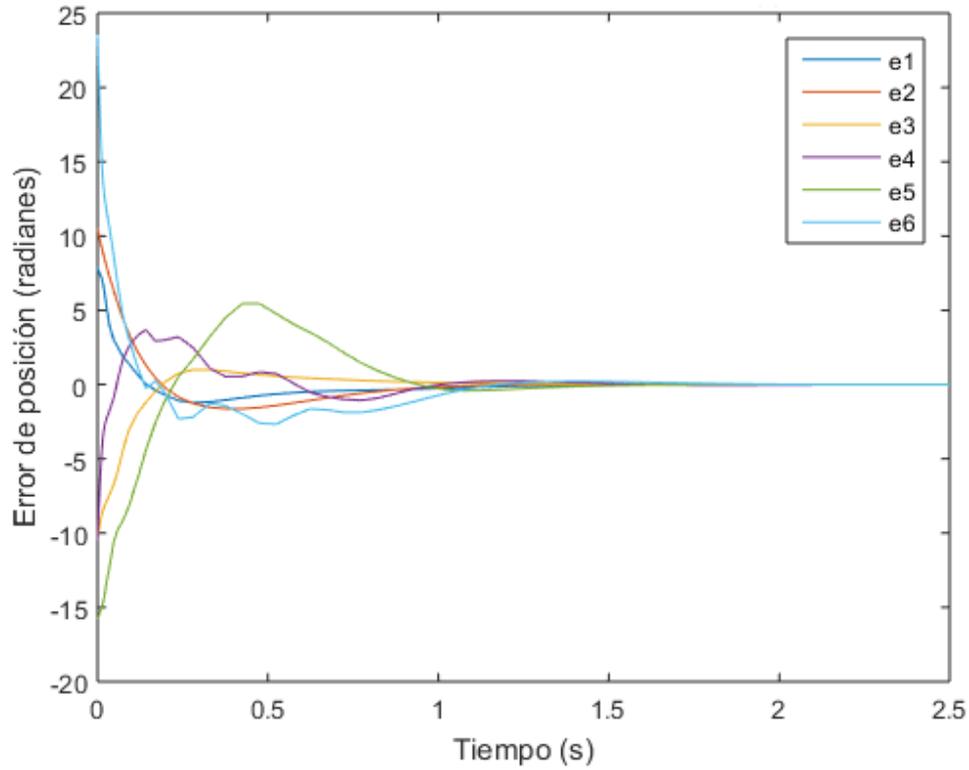


Figura 4.11: Simulación 3: evolución de la señal de error de posición con un controlador PD.

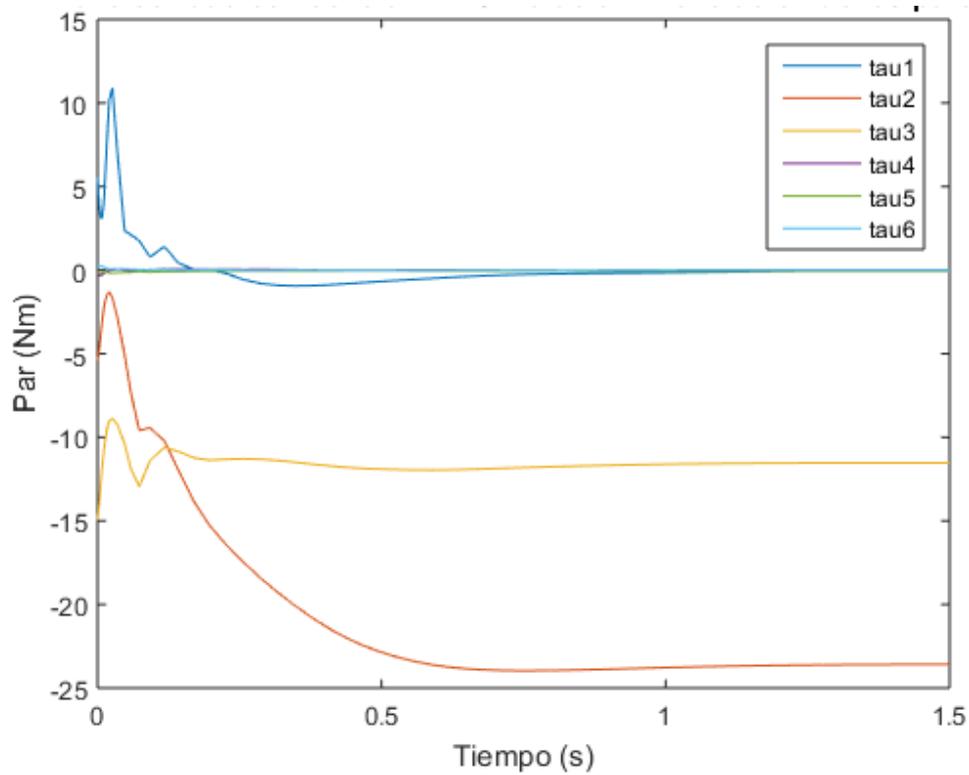


Figura 4.12: Simulación 3: evolución de los pares en las articulaciones con un controlador PD.

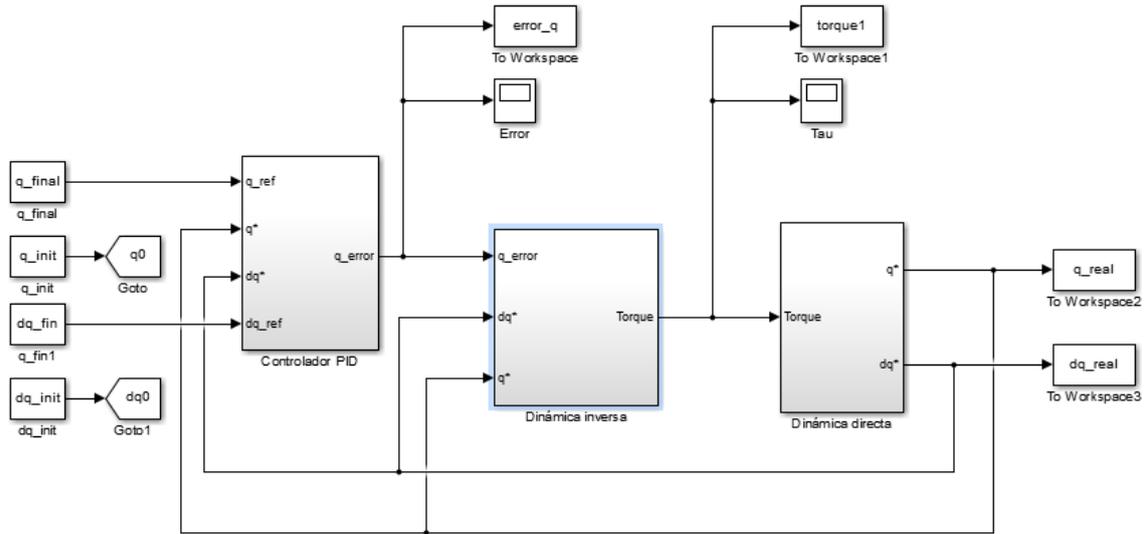


Figura 4.13: Esquema del modelo de Simulink con control PID en lazo cerrado.

previamente, se han establecido los siguientes valores para las ganancias K_p , K_d y K_i :

$$K_p = \begin{bmatrix} 40 & 0 & 0 & 0 & 0 & 0 \\ 0 & 40 & 0 & 0 & 0 & 0 \\ 0 & 0 & 40 & 0 & 0 & 0 \\ 0 & 0 & 0 & 40 & 0 & 0 \\ 0 & 0 & 0 & 0 & 40 & 0 \\ 0 & 0 & 0 & 0 & 0 & 40 \end{bmatrix} \quad (4.4)$$

$$K_d = \begin{bmatrix} 31 & 0 & 0 & 0 & 0 & 0 \\ 0 & 31 & 0 & 0 & 0 & 0 \\ 0 & 0 & 31 & 0 & 0 & 0 \\ 0 & 0 & 0 & 31 & 0 & 0 \\ 0 & 0 & 0 & 0 & 31 & 0 \\ 0 & 0 & 0 & 0 & 0 & 31 \end{bmatrix} \quad (4.5)$$

$$K_i = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix} \quad (4.6)$$

Simulación 4:

La simulación 4 se ha ejecutado con los siguientes valores iniciales:

$$\mathbf{q}_0 = [\pi/6 \quad \pi/12 \quad -\pi/12 \quad \pi/4 \quad -\pi/6 \quad \pi/2]$$

$$\mathbf{q}_{final} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

La evolución de los pares en esta simulación se han representado en la Figura 4.14. La posición angular y la velocidad angular se muestran en la Figura 4.15 y las curvas de errores del controlador se presentan en la Figura 4.14 .

Simulación 5:

La simulación 5 se ha ejecutado con los siguientes valores iniciales:

$$\mathbf{q}_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

$$\mathbf{q}_{final} = [\pi/6 \ \pi/12 \ -\pi/12 \ \pi/4 \ -\pi/6 \ \pi/2]$$

Los resultados de la simulación 5 se muestran a continuación. En primer lugar, en la Figura 4.15 se muestra la evolución de los pares en las articulaciones. En segundo lugar, la Figura 4.16 representa la evolución de la posición y velocidad angular de las articulaciones. Por último, la Figura 4.17 representa la evolución del error del controlador para cada una de las variables articulares.

Resultados:

Con un controlador PID se consiguen curvas de error sobre-amortiguadas y asintóticamente estables en el origen, como se aprecia en las Figuras 4.16 y 4.19. Estas curvas de errores tienen tiempos transitorios muy reducidos de aproximadamente una décima de segundo. A diferencia del controlador Proporcional-Derivativo, el controlador PID logra esta respuesta rápida y estable sin desestabilizar las curvas de pares (Figura 4.14 y Figura 4.17 para las simulaciones 4 y 5 respectivamente). Del mismo modo, la respuesta de \mathbf{q} y $\dot{\mathbf{q}}$ también están sobre-amortiguadas y sus transitorios para las simulaciones realizadas son de aproximadamente 2.5 s.

En el caso del controlador Proporcional-Integral-Derivativo, la ganancia integral permite hacer totalmente nulo el error de posición de las articulaciones. El valor de esta ganancia se ha escogido muy reducido para evitar desestabilizar las curvas de pares del modelo.

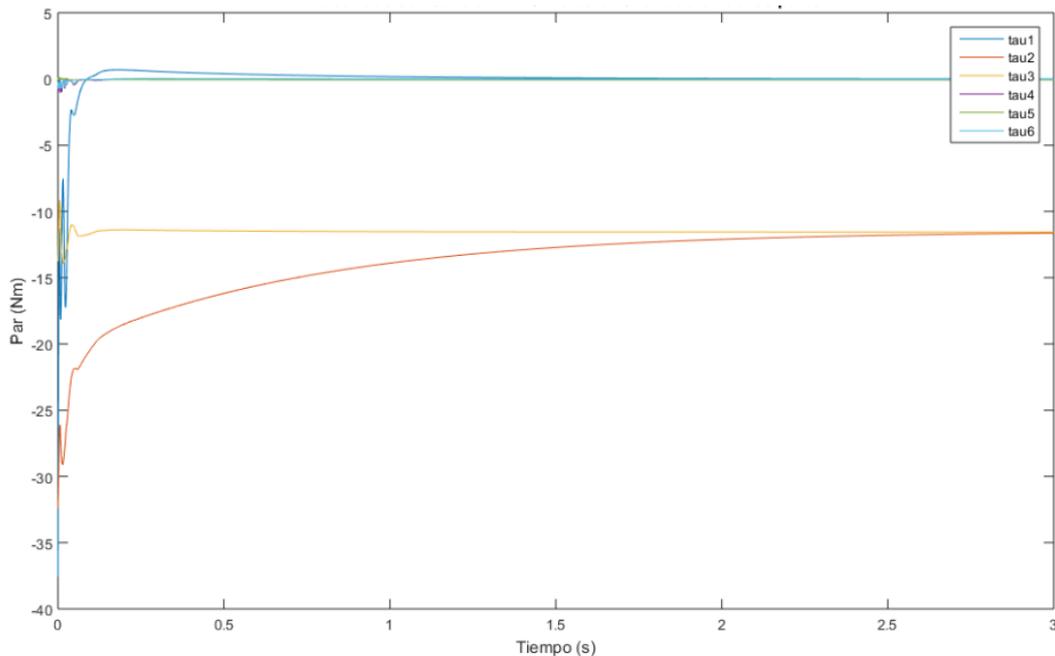


Figura 4.14: Simulación 4: evolución de los pares con el controlador PID.

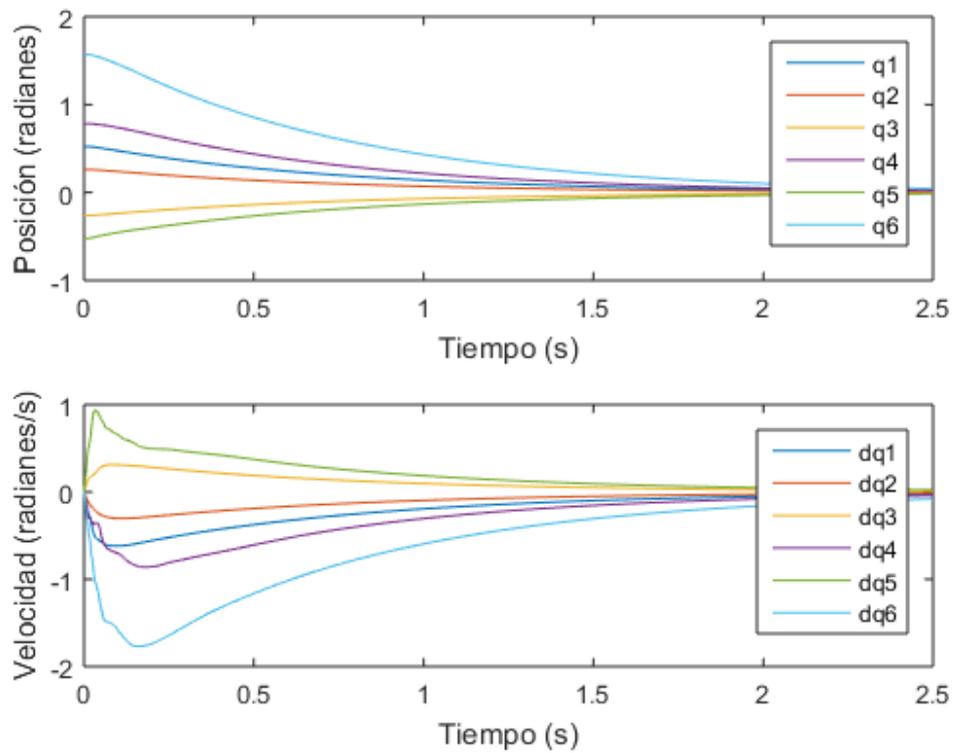


Figura 4.15: Simulación 4: evolución de la posición y la velocidad con el controlador PID.

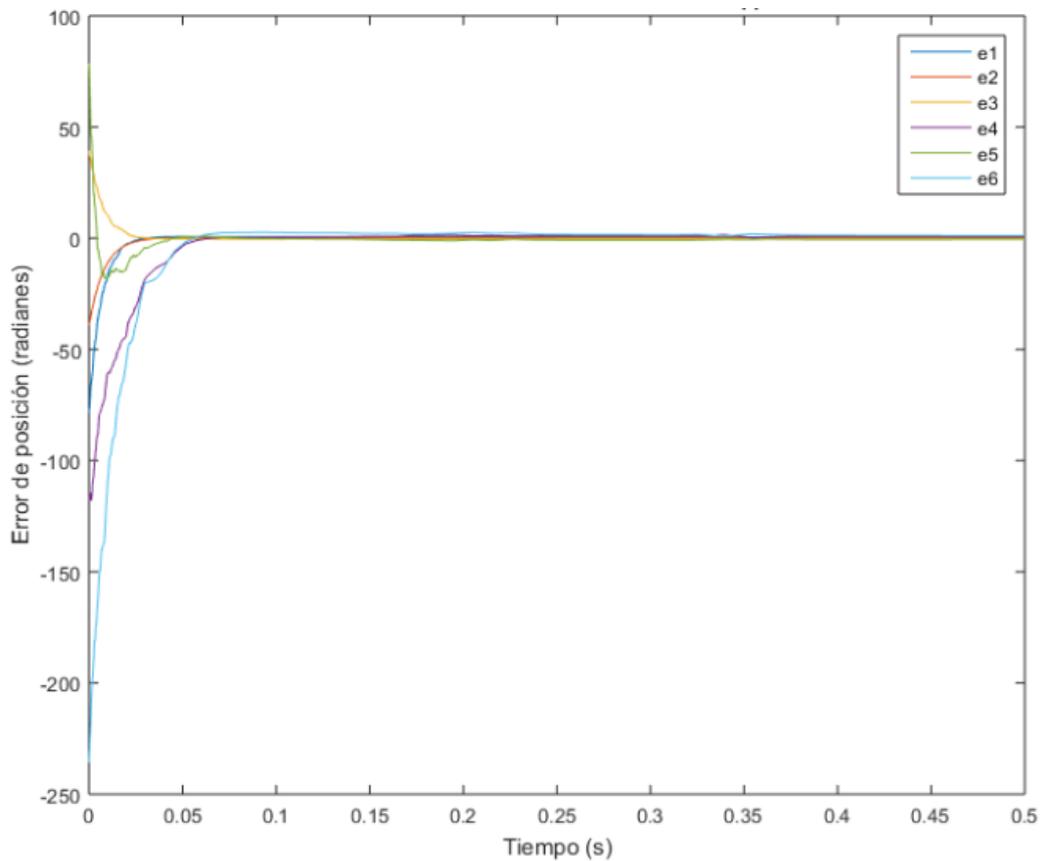


Figura 4.16: Simulación 4: evolución del error $e(t)$ con el controlador PID.

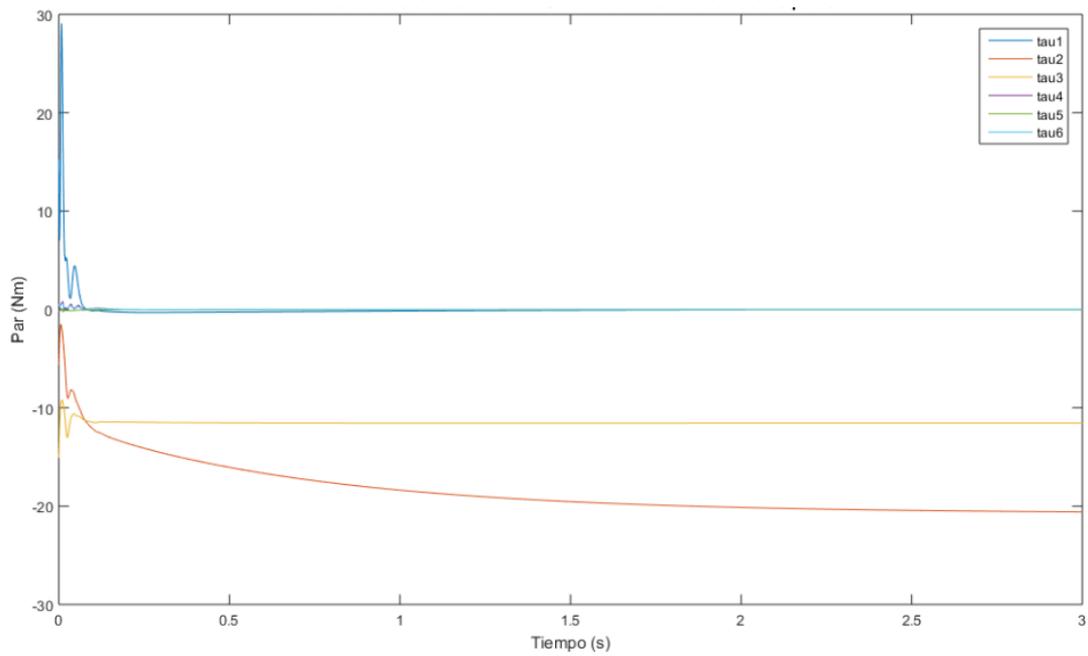


Figura 4.17: Simulación 5: evolución de los pares con el controlador PID.

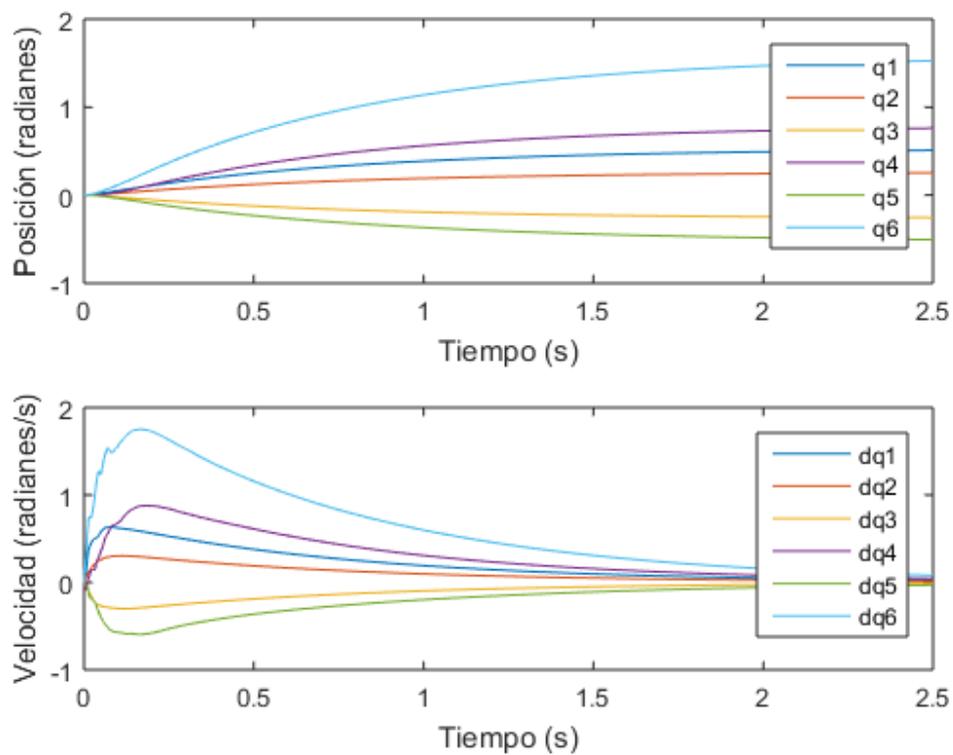


Figura 4.18: Simulación 5: evolución de la posición y la velocidad con el controlador PID.

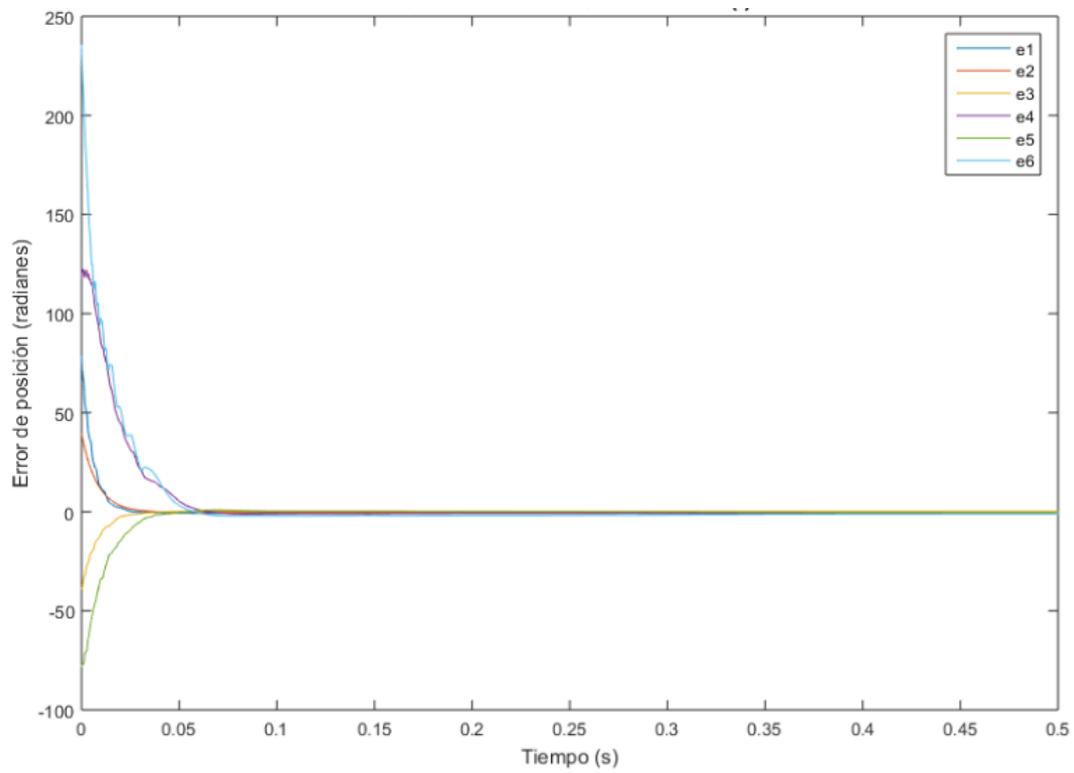


Figura 4.19: Simulación 5: evolución del error $e(t)$ con el controlador PID.

Capítulo 5

Conclusión del trabajo

En este capítulo se hace un resumen de las contribuciones de este trabajo de fin de máster y se sugieren posibles líneas de continuación del estudio realizado.

5.1. Contribuciones

El objetivo de este trabajo fue, desde un primer momento, obtener un modelo dinámico del robot IRB 120 y su validación mediante el diseño de controladores de lazo abierto y lazo cerrado, mediante la ejecución de una serie de simulaciones en el entorno de Matlab-Simulink.

Se ha realizado el modelado cinemático del robot ABB IRB1 120 mediante la utilización de los parámetros de Denavit-Hartenberg. Esto nos ha permitido el estudio de la cinemática del robot, como paso previo al modelado y control dinámico.

La obtención del modelo dinámico del robot IRB 120 es un problema complejo debido a la dificultad de poder obtener estimaciones correctas de los parámetros dinámicos. Por ello, se han utilizado diversas fuentes bibliográficas: datos de la empresa ABB, artículos y bibliografía científica. A partir de todas estas fuentes, se han obtenido, utilizando simplificaciones, buenas estimaciones de los parámetros dinámicos del robot: masas, centros de masas y las matrices de inercia respecto de los centros de masas de cada uno de los eslabones del robot IRB 120. Estos parámetros nos han permitido obtener el modelo mecánico del robot.

Este modelo ha sido calculado asumiendo una serie de simplificaciones que lo alejan del modelo dinámico exacto, pero que permiten caracterizar el robot apropiadamente como se ha demostrado en los resultados de las simulaciones del capítulo 4. Todo el modelo se ha llevado a cabo en el dominio articular, es decir, que no se ha requerido el uso de cinemática directa e inversa. Esto es así porque los controladores que se han implementado actúan sobre las señales de posición de las articulaciones y no en el espacio cartesiano.

Las ecuaciones que caracterizan al modelo mecánico del robot se han desarrollado en el apartado 3.5 de este TFM, y han sido obtenidas mediante programación con lenguaje MAPLE. Estas ecuaciones han sido integradas con el software Matlab-Simulink para obtener los controladores dinámicos (lazo abierto y lazo cerrado) del robot IRB 120.

Varias estrategias de control dinámico del robot IRB 120 han sido programadas y simuladas utilizando Matlab-Simulink. Todas las simulaciones se realizaron en el ámbito de las coordenadas articulares del robot. La simulación realizada con control dinámico

en lazo abierto demostró que no es válida para estabilizar al robot. Con control en lazo abierto las variables articulares del robot no alcanzaban los valores deseados, introducidos como consigna.

Por ello, se diseñaron varias estrategias de control en lazo cerrado: "Control proporcional con realimentación de velocidad", "control PD" y "control PID". Estos controladores sencillos han permitido verificar que proporcionan estabilidad al robot, y por lo tanto, como primeras estrategias de control dinámico en lazo cerrado para el robot IRB 120, proporcionan resultados adecuados.

5.2. Posibles líneas de trabajo a futuro

Se proponen las siguientes líneas de trabajo a partir del estudio ya realizado:

1. Estimación de los parámetros dinámicos del robot IRB 120 mediante la realización de ensayos con el ejemplar del laboratorio de la Escuela de Ingenierías Industriales de la Universidad de Valladolid y el ajuste multi-variable por mínimos cuadrados de los resultados para aproximar los valores de los centros de masas, de las matrices de inercias y de las masas. Esto permitiría mejorar el modelo dinámico propuesto en este trabajo simplemente actualizando dichos valores.
2. Implementación de la cinemática directa - cuya obtención es trivial a partir de este trabajo - y de la cinemática inversa para pasar del dominio articular al dominio de coordenadas cartesianas que permita conocer la posición y orientación del extremo del robot.
3. Estudio de los métodos de control de trayectorias. Este trabajo se ha centrado en el uso de estrategias de control de posición, pero el modelo dinámico obtenido representa una buena base para el estudio de estrategias de control más sofisticadas como el control robusto, el control adaptativo o el de compensación adaptativa.

Apéndice A

Ficha técnica del robot IRB 120

IRB 120

ABB's 6 axis robot – for flexible and compact production



The IRB 120 robot is the latest addition to ABB's new fourth-generation of robotic technology. It is ideal for material handling and assembly applications and provides an agile, compact and lightweight solution with superior control and path accuracy.

Compact and lightweight

IRB 120's compact design enables it to be mounted virtually anywhere at any angle without any restriction - for example inside a cell, on top of a machine or close to other robots.

IRB 120 is also the most portable and easy to integrate on the market with its 25 kg weight. The smooth surfaces are easy to clean and the cables for air and customer signals are internally routed, all the way from the foot to the wrist, ensuring that integration is effortless.

Multipurpose

IRB 120 is ideal for a wide range of industries including the electronic, food and beverage, machinery, solar, pharmaceutical, medical and research sectors.

The Food Grade Lubrication (NSF H1) option includes Clean Room ISO Class 5, which ensures uncompromising safety and hygiene for food and beverage applications.

Optimized working range

IRB 120 has a horizontal reach of 580 mm, the best in class stroke, the ability to reach 112 mm below its base and a very compact turning radius.

Fast, accurate and agile

Designed with a light, aluminum structure, the motors ensure the robot is enabled with a fast acceleration, and can deliver accuracy and agility in any application.

IRC5 Compact controller – optimized for small robots

ABB's new IRC5 Compact controller presents the capabilities of the IRC5 controller in a compact format. It brings accuracy and motion control to applications which have been exclusive to large installations and enables easy commissioning through one phase power input, external connectors for all signals and a built-in expandable 16 in, 16 out, I/O system.

RobotStudio for offline programming enables manufacturers to simulate a production cell to find the optimal position for the robot, and provide offline programming to prevent costly downtime and delays to production.

Reduced footprint

The combination of the new lightweight architecture of the IRB 120 with the new IRC5 Compact controller introduces a significantly reduced footprint.

Specification

Robot version	Reach (m)	Handling capacity (kg)	Armload (kg)
IRB 120-3/0.6	0.58	3*	0.30
Number of axes	6		
Protection	IP30		
Mounting	Any angle		
Controller	IRC5 Compact/IRC5 Single Cabinet		
Integrated signal supply	10 signals on wrist		
Integrated air supply	4 air on wrist (5 bar)		

* 4 with vertical wrist

Performance (according to ISO 9283)

IRB 120	
1 kg picking cycle	
25 x 300 x 25 mm	0.58 s
25 x 300 x 25 with 180° axis 6 reorientation	0.92 s
Acceleration time 0-1 m/s	0.07 s
Position repeatability	0.01 mm

Technical information

Electrical Connections

Supply voltage	200-600 V, 50/60 Hz
Rated power transformer rating	3.0 kVA
Power consumption	0.24 kW

Physical

Robot base	180 x 180 mm
Robot height	700 mm
Robot weight	25 kg

Environment

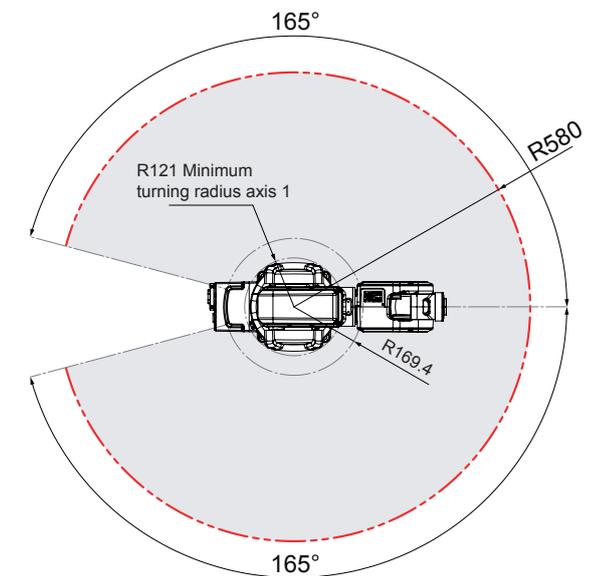
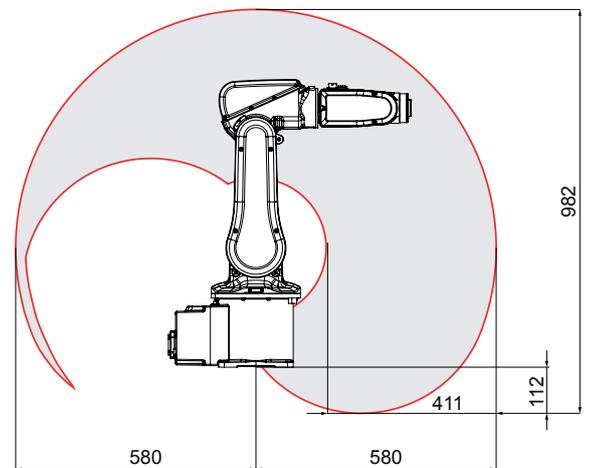
Ambient temperature for robot manipulator:	
During operation	+5°C (41°F) to +45°C (113°F)
During transportation and storage	-25°C (-13°F) to +55°C (131°F)
During short periods (max. 24 h)	up to +70°C (158°F)
Relative humidity	Max. 95%
Noise level	Max. 70 dB (A)
Safety	Safety and emergency stops 2-channel safety circuits supervision, 3-position enabling device
Emission	EMC/EMI-shielded
Options	Clean Room ISO class 5 (certified by IPA)**

** ISO class 4 can be reached under certain conditions. Data and dimensions may be changed without notice.

Movement

Axis movement	Working range	Velocity IRB 120
Axis 1 rotation	+165° to -165°	250°/s
Axis 2 arm	+110° to -110°	250°/s
Axis 3 arm	+70° to -110°	250°/s
Axis 4 wrist	+160° to -160°	320°/s
Axis 5 bend	+120° to -120°	320°/s
Axis 6 turn	Default: +400° to -400° Max. rev: +242 to -242	420°/s

Working range



Apéndice B

**Modelo dinámico del robot IRB 120
obtenido utilizando el software
MAPLE**

```

> restart :
> with(LinearAlgebra) :
>
> #Variables articulares (grados de libertad)
>
> q := Vector([q1(t), q2(t), q3(t), q4(t), q5(t), q6(t)]) :
> Dq := map(diff, q, t) :
> DDq := map(diff, Dq, t) :
>
> #Vectores entre articulaciones y vectores al centro de masas
>
> r01 := Vector([0.000, -0.290, 0.000]) :
> r12 := Vector([0.270, 0.000, 0.000]) :
> r23 := Vector([0.070, 0.000, 0.000]) :
> r34 := Vector([0.000, 0.302, 0.000]) :
> r45 := Vector([0.000, 0.000, 0.000]) :
> r56 := Vector([0.000, 0.000, 0.072]) :
>
> r0c1 := Vector([0.000, -0.240, 0.000]) :
> r1c2 := Vector([0.146, 0.000, 0.000]) :
> r2c3 := Vector([0.024, 0.000, 0.058]) :
> r3c4 := Vector([0.000, 0.192, 0.000]) :
> r4c5 := Vector([0.000, 0.000, 0.000]) :
> r5c6 := Vector([0.000, 0.000, 0.063]) :
>
> r2c2 := r1c2 - r12 :
> r3c3 := r2c3 - r23 :
> r4c4 := r3c4 - r34 :
> r5c5 := r4c5 - r45 :
> r1c1 := r0c1 - r01 :
>
> #Vector gravedad en sistema inercial, fuerzas y torques externos
>
> g0 := Vector([0.000, 0.000, -g]) :
> f7 := Vector([0, 0, 0]) :
> tau7 := Vector([0, 0, 0]) :
>
> R01 :=  $\begin{bmatrix} \cos(q[1]) & 0 & -\sin(q[1]) \\ \sin(q[1]) & 0 & \cos(q[1]) \\ 0 & -1 & 0 \end{bmatrix}$  :

```

```

> R12 := 
$$\begin{bmatrix} \sin(q[2]) & \cos(q[2]) & 0 \\ -\cos(q[2]) & \sin(q[2]) & 0 \\ 0 & 0 & 1 \end{bmatrix} :$$

>
> R23 := 
$$\begin{bmatrix} \cos(q[3]) & 0 & -\sin(q[3]) \\ \sin(q[3]) & 0 & \cos(q[3]) \\ 0 & -1 & 0 \end{bmatrix} :$$

>
> R34 := 
$$\begin{bmatrix} \cos(q[4]) & 0 & \sin(q[4]) \\ \sin(q[4]) & 0 & -\cos(q[4]) \\ 0 & 1 & 0 \end{bmatrix} :$$

>
> R45 := 
$$\begin{bmatrix} -\cos(q[5]) & 0 & -\sin(q[5]) \\ -\sin(q[5]) & 0 & \cos(q[5]) \\ 0 & 1 & 0 \end{bmatrix} :$$

>
> R56 := 
$$\begin{bmatrix} \cos(q[6]) & -\sin(q[6]) & 0 \\ \sin(q[6]) & \cos(q[6]) & 0 \\ 0 & 0 & 1 \end{bmatrix} :$$

>
> R67 := 
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} : \text{ # Orientation of the tool with respect orientation of the end-effector}$$

>
> R02 := MatrixMatrixMultiply(R01, R12) :
> R03 := MatrixMatrixMultiply(R02, R23) :
> R04 := MatrixMatrixMultiply(R03, R34) :
> R05 := MatrixMatrixMultiply(R04, R45) :
> R06 := MatrixMatrixMultiply(R05, R56) :
>
> #Ejes de rotación de articulaciones en sistema inercial
>
> z0 := Vector([[0], [0], [1]]) :
> b1 := MatrixVectorMultiply(Transpose(R01), z0) :
> b2 := combine(MatrixMatrixMultiply(Transpose(R02), MatrixVectorMultiply(R01, z0)),
> trig) :
> b3 := combine(MatrixMatrixMultiply(Transpose(R03), MatrixVectorMultiply(R02, z0)),
> trig) :

```

```

> b4 := combine(MatrixMatrixMultiply(Transpose(R04), MatrixVectorMultiply(R03, z0)),
  trig) :
=
> b5 := combine(MatrixMatrixMultiply(Transpose(R05), MatrixVectorMultiply(R04, z0)),
  trig) :
=
> b6 := combine(MatrixMatrixMultiply(Transpose(R06), MatrixVectorMultiply(R05, z0)),
  trig) :
=
>
> # Masas de los eslabones (extraídas de Barhaghtalab et al. [4])
>
> m1 := 3.36 :
> m2 := 6.8 :
> m3 := 6.22 :
> m4 := 2.0 :
> m5 := 1.3 :
> m6 := 0.11 :
>
> #Tensores de inercia en el centro de masas de cada eslabón (extraídas de Barhaghtalab et al. )
>
> I1 :=  $\begin{bmatrix} 0.012 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.011 \end{bmatrix}$  :
=
>
> I2 :=  $\begin{bmatrix} 0.064 & 0 & 0 \\ 0 & 0.012 & 0 \\ 0 & 0 & 0.064 \end{bmatrix}$  :
=
>
> I3 :=  $\begin{bmatrix} 0.35 & 0 & 0.23 \\ 0 & 0.37 & 0 \\ 0.23 & 0 & 0.18 \end{bmatrix}$  :
=
>
> I4 :=  $\begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.006 & 0 \\ 0 & 0 & 0.008 \end{bmatrix}$  :
=
>
> I5 :=  $\begin{bmatrix} 0.0023 & 0 & 0 \\ 0 & 0.0023 & 0 \\ 0 & 0 & 0.0015 \end{bmatrix}$  :
=
>

```

```
> I6 := 
$$\begin{bmatrix} 0.0004 & 0 & 0 \\ 0 & 0.004 & 0 \\ 0 & 0 & 0.008 \end{bmatrix} :$$

```

```
> #Recursión progresiva: eslabón 1
```

```
>  $\omega_1 := b_1 \cdot Dq[1] :$ 
```

```
>  $\alpha_1 := b_1 \cdot DDq[1] + \text{CrossProduct}(\omega_1, b_1 \cdot Dq[1]) :$ 
```

```
>  $D\omega_1 := \text{map}(\text{diff}, \omega_1, t) :$ 
```

```
>  $ae_1 := \text{CrossProduct}(D\omega_1, r_{01}) + \text{CrossProduct}(\omega_1, \text{CrossProduct}(\omega_1, r_{01})) :$ 
```

```
>  $ac_1 := \text{CrossProduct}(D\omega_1, r_{0c1}) + \text{CrossProduct}(\omega_1, \text{CrossProduct}(\omega_1, r_{0c1})) :$ 
```

```
>  $g_1 := \text{MatrixVectorMultiply}(\text{Transpose}(R_{01}), g_0) :$ 
```

```
> #Recursión progresiva: eslabón 2
```

```
>  $\omega_2 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R_{12}), \omega_1) + b_2 \cdot Dq[2], \text{trig}) :$ 
```

```
>  $\alpha_2 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R_{12}), \alpha_1) + b_2 \cdot DDq[2] + \text{CrossProduct}(\omega_2, b_2 \cdot Dq[2]), \text{trig}) :$ 
```

```
>  $D\omega_2 := \text{map}(\text{diff}, \omega_2, t) :$ 
```

```
>  $ae_2 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R_{12}), ae_1) + \text{CrossProduct}(D\omega_2, r_{12}) + \text{CrossProduct}(\omega_2, \text{CrossProduct}(\omega_2, r_{12})), \text{trig}) :$ 
```

```
>  $ac_2 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R_{12}), ac_1) + \text{CrossProduct}(D\omega_2, r_{1c2}) + \text{CrossProduct}(\omega_2, r_{1c2}), \text{trig}) :$ 
```

```
>  $g_2 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R_{02}), g_0), \text{trig}) :$ 
```

```
> #Recursión progresiva: eslabón 3
```

```
>  $\omega_3 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R_{23}), \omega_2) + b_3 \cdot Dq[3], \text{trig}) :$ 
```

```
>  $\alpha_3 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R_{23}), \alpha_2) + b_3 \cdot DDq[3] + \text{CrossProduct}(\omega_3, b_3 \cdot Dq[3]), \text{trig}) :$ 
```

```
>  $D\omega_3 := \text{map}(\text{diff}, \omega_3, t) :$ 
```

```
>  $ae_3 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R_{23}), ae_2) + \text{CrossProduct}(D\omega_3, r_{23}) + \text{CrossProduct}(\omega_3, \text{CrossProduct}(\omega_3, r_{23})), \text{trig}) :$ 
```

```
>  $ac_3 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R_{23}), ac_2) + \text{CrossProduct}(D\omega_3, r_{2c3}) + \text{CrossProduct}(\omega_3, r_{2c3}), \text{trig}) :$ 
```

```
>  $g_3 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R_{03}), g_0), \text{trig}) :$ 
```

```
> #Recursión progresiva: eslabón 4
```

```
>  $\omega_4 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R_{34}), \omega_3) + b_4 \cdot Dq[4], \text{trig}) :$ 
```

```

>  $\alpha_4 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R34), \alpha_3) + b_4 \cdot \text{DDq}[4]$ 
  +  $\text{CrossProduct}(\omega_4, b_4 \cdot \text{Dq}[4]), \text{trig}) :$ 
=
>  $D\omega_4 := \text{map}(\text{diff}, \omega_4, t) :$ 
=
>  $ae_4 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R34), ae_3) + \text{CrossProduct}(D\omega_4, r34)$ 
  +  $\text{CrossProduct}(\omega_4, \text{CrossProduct}(\omega_4, r34)), \text{trig}) :$ 
=
>  $ac_4 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R34), ae_3) + \text{CrossProduct}(D\omega_4, r3c4)$ 
  +  $\text{CrossProduct}(\omega_4, r3c4), \text{trig}) :$ 
=
>  $g_4 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R04), g_0), \text{trig}) :$ 
=
>
=
> #Recursión progresiva: eslabón 5
=
>
=
>  $\omega_5 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R45), \omega_4) + b_5 \cdot \text{Dq}[5], \text{trig}) :$ 
>  $\alpha_5 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R45), \alpha_4) + b_5 \cdot \text{DDq}[5]$ 
  +  $\text{CrossProduct}(\omega_5, b_5 \cdot \text{Dq}[5]), \text{trig}) :$ 
=
>  $D\omega_5 := \text{map}(\text{diff}, \omega_5, t) :$ 
>  $ae_5 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R45), ae_4) + \text{CrossProduct}(D\omega_5, r45)$ 
  +  $\text{CrossProduct}(\omega_5, \text{CrossProduct}(\omega_5, r45)), \text{trig}) :$ 
=
>  $ac_5 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R45), ae_4) + \text{CrossProduct}(D\omega_5, r4c5)$ 
  +  $\text{CrossProduct}(\omega_5, r4c5), \text{trig}) :$ 
=
>  $g_5 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R05), g_0), \text{trig}) :$ 
=
>
=
> #Recursión progresiva: eslabón 6
=
>
=
>  $\omega_6 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R56), \omega_5) + b_6 \cdot \text{Dq}[6], \text{trig}) :$ 
>  $\alpha_6 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R56), \alpha_5) + b_6 \cdot \text{DDq}[6]$ 
  +  $\text{CrossProduct}(\omega_6, b_6 \cdot \text{Dq}[6]), \text{trig}) :$ 
=
>  $D\omega_6 := \text{map}(\text{diff}, \omega_6, t) :$ 
>  $ac_6 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R56), ae_5) + \text{CrossProduct}(D\omega_6, r5c6)$ 
  +  $\text{CrossProduct}(\omega_6, r5c6), \text{trig}) :$ 
=
>  $g_6 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R06), g_0), \text{trig}) :$ 
>  $ae_6 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R56), ae_5) + \text{CrossProduct}(D\omega_6, r56)$ 
  +  $\text{CrossProduct}(\omega_6, \text{CrossProduct}(\omega_6, r56)), \text{trig}) :$ 
=
>
=
> #Recursión regresiva: eslabón 6
=
>
=
>  $f_6 := \text{Add}(m_6 \cdot ac_6, -m_6 \cdot g_6) + \text{MatrixVectorMultiply}(R67, f_7) :$ 
>  $\tau_6 := -\text{CrossProduct}(f_6, r5c6) + \text{MatrixVectorMultiply}(I_6, \alpha_6) + \text{CrossProduct}(\omega_6,$ 
   $\text{MatrixVectorMultiply}(I_6, \omega_6)) + \text{MatrixVectorMultiply}(R67, \tau_7)$ 
  +  $\text{CrossProduct}(\text{MatrixVectorMultiply}(R67, f_7), r6c6) :$ 
=
>  $\tau_6 \text{Dyn} := \text{collect}(\text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(b_6), \tau_6), \text{trig}), \{\text{DDq}[1],$ 
   $\text{DDq}[2], \text{DDq}[3], \text{DDq}[4], \text{DDq}[5], \text{DDq}[6], \text{Dq}[1], \text{Dq}[2], \text{Dq}[3], \text{Dq}[4], \text{Dq}[5],$ 

```

$Dq[6]) :$

> *#Recursión regresiva: eslabón 5*

> $f5 := Add(m5 \cdot ac5, -m5 \cdot g5) + MatrixVectorMultiply(R56, f6) :$

> $\tau5 := -CrossProduct(f5, r4c5) + MatrixVectorMultiply(I5, \alpha5) + CrossProduct(\omega5, MatrixVectorMultiply(I5, \omega5)) + MatrixVectorMultiply(R56, \tau6) + CrossProduct(MatrixVectorMultiply(R56, f6), r5c5) :$

> $\tau5Dyn := collect(combine(MatrixVectorMultiply(Transpose(b5), \tau5), trig), \{DDq[1], DDq[2], DDq[3], DDq[4], DDq[5], DDq[6], Dq[1], Dq[2], Dq[3], Dq[4], Dq[5], Dq[6]\}) :$

> *#Recursión regresiva: eslabón 4*

> $f4 := Add(m4 \cdot ac4, -m4 \cdot g4) + MatrixVectorMultiply(R45, f5) :$

> $\tau4 := -CrossProduct(f4, r3c4) + MatrixVectorMultiply(I4, \alpha4) + CrossProduct(\omega4, MatrixVectorMultiply(I4, \omega4)) + MatrixVectorMultiply(R45, \tau5) + CrossProduct(MatrixVectorMultiply(R45, f5), r4c4) :$

> $\tau4Dyn := collect(combine(MatrixVectorMultiply(Transpose(b4), \tau4), trig), \{DDq[1], DDq[2], DDq[3], DDq[4], DDq[5], DDq[6], Dq[1], Dq[2], Dq[3], Dq[4], Dq[5], Dq[6]\}) :$

> *#Recursión regresiva: eslabón 3*

> $f3 := Add(m3 \cdot ac3, -m3 \cdot g3) + MatrixVectorMultiply(R34, f4) :$

> $\tau3 := -CrossProduct(f3, r2c3) + MatrixVectorMultiply(I3, \alpha3) + CrossProduct(\omega3, MatrixVectorMultiply(I3, \omega3)) + MatrixVectorMultiply(R34, \tau4) + CrossProduct(MatrixVectorMultiply(R34, f4), r3c3) :$

> $\tau3Dyn := collect(combine(MatrixVectorMultiply(Transpose(b3), \tau3), trig), \{DDq[1], DDq[2], DDq[3], DDq[4], DDq[5], DDq[6], Dq[1], Dq[2], Dq[3], Dq[4], Dq[5], Dq[6]\}) :$

> *#Recursión regresiva: eslabón 2*

> $f2 := Add(m2 \cdot ac2, -m2 \cdot g2) + MatrixVectorMultiply(R23, f3) :$

> $\tau2 := -CrossProduct(f2, r1c2) + MatrixVectorMultiply(I2, \alpha2) + CrossProduct(\omega2, MatrixVectorMultiply(I2, \omega2)) + MatrixVectorMultiply(R23, \tau3) + CrossProduct(MatrixVectorMultiply(R23, f3), r2c2) :$

> $\tau2Dyn := collect(combine(MatrixVectorMultiply(Transpose(b2), \tau2), trig), \{DDq[1], DDq[2], DDq[3], DDq[4], DDq[5], DDq[6], Dq[1], Dq[2], Dq[3], Dq[4], Dq[5], Dq[6]\}) :$

> *#Recursión regresiva: eslabón 1*

> $f1 := Add(m1 \cdot ac1, -m1 \cdot g1) + MatrixVectorMultiply(R12, f2) :$

```

>  $\tau 1 := -\text{CrossProduct}(f1, r0c1) + \text{MatrixVectorMultiply}(I1, \alpha 1) + \text{CrossProduct}(\omega 1,$ 
   $\text{MatrixVectorMultiply}(I1, \omega 1)) + \text{MatrixVectorMultiply}(R12, \tau 2)$ 
   $+ \text{CrossProduct}(\text{MatrixVectorMultiply}(R12, f2), r1c1) :$ 
>  $\tau 1 \text{Dyn} := \text{collect}(\text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(b1), \tau 1), \text{trig}), \{DDq[1],$ 
   $DDq[2], DDq[3], DDq[4], DDq[5], DDq[6], Dq[1], Dq[2], Dq[3], Dq[4], Dq[5],$ 
   $Dq[6]\}) :$ 
>
> # Dinámica: vector de pares en la dirección de los ejes articulares
>
>  $\tau 0 := \text{Vector}([\tau 1 \text{Dyn}], [\tau 2 \text{Dyn}], [\tau 3 \text{Dyn}], [\tau 4 \text{Dyn}], [\tau 5 \text{Dyn}], [\tau 6 \text{Dyn}]) :$ 
>
> #Evaluando la matriz de inercias y el vector de torques gravitatorios
>
>  $m11 := \text{eval}(\tau 1 \text{Dyn}, \{DDq[1]=1, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=0, DDq[6]$ 
   $=0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0\}) :$ 
>  $m12 := \text{eval}(\tau 1 \text{Dyn}, \{DDq[1]=0, DDq[2]=1, DDq[3]=0, DDq[4]=0, DDq[5]=0, DDq[6]$ 
   $=0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0\}) :$ 
>  $m13 := \text{eval}(\tau 1 \text{Dyn}, \{DDq[1]=0, DDq[2]=0, DDq[3]=1, DDq[4]=0, DDq[5]=0, DDq[6]$ 
   $=0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0\}) :$ 
>  $m14 := \text{eval}(\tau 1 \text{Dyn}, \{DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=1, DDq[5]=0, DDq[6]$ 
   $=0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0\}) :$ 
>  $m15 := \text{eval}(\tau 1 \text{Dyn}, \{DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=1, DDq[6]$ 
   $=0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0\}) :$ 
>  $m16 := \text{eval}(\tau 1 \text{Dyn}, \{DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=0, DDq[6]$ 
   $=1, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0\}) :$ 
>
>  $m21 := \text{eval}(\tau 2 \text{Dyn}, \{DDq[1]=1, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=0, DDq[6]$ 
   $=0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0\}) :$ 
>  $m22 := \text{eval}(\tau 2 \text{Dyn}, \{DDq[1]=0, DDq[2]=1, DDq[3]=0, DDq[4]=0, DDq[5]=0, DDq[6]$ 
   $=0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0\}) :$ 
>  $m23 := \text{eval}(\tau 2 \text{Dyn}, \{DDq[1]=0, DDq[2]=0, DDq[3]=1, DDq[4]=0, DDq[5]=0, DDq[6]$ 
   $=0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0\}) :$ 
>  $m24 := \text{eval}(\tau 2 \text{Dyn}, \{DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=1, DDq[5]=0, DDq[6]$ 
   $=0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0\}) :$ 
>  $m25 := \text{eval}(\tau 2 \text{Dyn}, \{DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=1, DDq[6]$ 
   $=0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0\}) :$ 
>  $m26 := \text{eval}(\tau 2 \text{Dyn}, \{DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=0, DDq[6]$ 
   $=1, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0\}) :$ 
>
>  $m31 := \text{eval}(\tau 3 \text{Dyn}, \{DDq[1]=1, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=0, DDq[6]$ 
   $=0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0\}) :$ 

```



```

= 0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0}) :
> m63 := eval( τ6Dyn, {DDq[1]=0, DDq[2]=0, DDq[3]=1, DDq[4]=0, DDq[5]=0, DDq[6]
= 0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0}) :
> m64 := eval( τ6Dyn, {DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=1, DDq[5]=0, DDq[6]
= 0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0}) :
> m65 := eval( τ6Dyn, {DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=1, DDq[6]
= 0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0}) :
> m66 := eval( τ6Dyn, {DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=0, DDq[6]
= 1, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0, g=0}) :
>
> gr6 := eval( τ6Dyn, {DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=0, DDq[6]
= 0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0}) :
> gr5 := eval( τ5Dyn, {DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=0, DDq[6]
= 0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0}) :
> gr4 := eval( τ4Dyn, {DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=0, DDq[6]
= 0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0}) :
> gr3 := eval( τ3Dyn, {DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=0, DDq[6]
= 0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0}) :
> gr2 := eval( τ2Dyn, {DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=0, DDq[6]
= 0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0}) :
> gr1 := eval( τ1Dyn, {DDq[1]=0, DDq[2]=0, DDq[3]=0, DDq[4]=0, DDq[5]=0, DDq[6]
= 0, Dq[1]=0, Dq[2]=0, Dq[3]=0, Dq[4]=0, Dq[5]=0, Dq[6]=0}) :
>
> #Matrices y vectores del sistema dinámico
>
> M0 := Matrix([ [m11, m12, m13, m14, m15, m16], [m21, m22, m23, m24, m25, m26], [m31,
m32, m33, m34, m35, m36], [m41, m42, m43, m44, m45, m46], [m51, m52, m53, m54, m55,
m56], [m61, m62, m63, m64, m65, m66]]) :
>
> G0 := Vector([ [gr1], [gr2], [gr3], [gr4], [gr5], [gr6]]) :
>
> C0Dq := combine( τ0 - MatrixVectorMultiply(M0, DDq) - G0, trig ) :
>
> C0DqS := simplify(C0Dq) :
>
> # Código Matlab generado por Maple: asignación de variables
>
> with(CodeGeneration) :
>
> Matlab(m11, resultname="m11") :
> Matlab(m12, resultname="m12") :
> Matlab(m13, resultname="m13") :

```

> Matlab(m14, resultname = "m14") :
=> Matlab(m15, resultname = "m15") :
=> Matlab(m16, resultname = "m16") :
=>
=> Matlab(m21, resultname = "m21") :
=> Matlab(m22, resultname = "m22") :
=> Matlab(m23, resultname = "m23") :
=> Matlab(m24, resultname = "m24") :
=> Matlab(m25, resultname = "m25") :
=> Matlab(m26, resultname = "m26") :
=>
=> Matlab(m31, resultname = "m31") :
=> Matlab(m32, resultname = "m32") :
=> Matlab(m33, resultname = "m33") :
=> Matlab(m34, resultname = "m34") :
=> Matlab(m35, resultname = "m35") :
=> Matlab(m36, resultname = "m36") :
=>
=> Matlab(m41, resultname = "m41") :
=> Matlab(m42, resultname = "m42") :
=> Matlab(m43, resultname = "m43") :
=> Matlab(m44, resultname = "m44") :
=> Matlab(m45, resultname = "m45") :
=> Matlab(m46, resultname = "m46") :
=>
=> Matlab(m51, resultname = "m51") :
=> Matlab(m52, resultname = "m52") :
=> Matlab(m53, resultname = "m53") :
=> Matlab(m54, resultname = "m54") :
=> Matlab(m55, resultname = "m55") :
=> Matlab(m56, resultname = "m56") :
=>
=> Matlab(m61, resultname = "m61") :
=> Matlab(m62, resultname = "m62") :
=> Matlab(m63, resultname = "m63") :
=> Matlab(m64, resultname = "m64") :
=> Matlab(m65, resultname = "m65") :
=> Matlab(m66, resultname = "m66") :
=>
=> Matlab(gr1, resultname = "gr1") :
=> Matlab(gr2, resultname = "gr2") :
=> Matlab(gr3, resultname = "gr3") :
=> Matlab(gr4, resultname = "gr4") :
=> Matlab(gr5, resultname = "gr5") :
=> Matlab(gr6, resultname = "gr6") :

```

>
> Matlab(C0Dq(1), resultname="cdq1") :
> Matlab(C0Dq(2), resultname="cdq2") :
> Matlab(C0Dq(3), resultname="cdq3") :
> Matlab(C0Dq(4), resultname="cdq4") :
> Matlab(C0Dq(5), resultname="cdq5") :
> Matlab(C0Dq(6), resultname="cdq6") :

```

```

> # Test simetría de matriz de inercia M0 (debe ser cero en la precisión de la máquina)

```

```

> sm12 := m12 - m21

```

```

sm12 := 5.4210108624 10-20 cos(q5(t) - q4(t) + q2(t)) + 3.4694469520 10-18 sin(q2(t)
+ q3(t)) - 2.7105054312 10-20 sin(2 q6(t) - q5(t) - 2 q4(t) + q2(t) + q3(t))
- 2.7105054312 10-20 sin(2 q6(t) - 2 q4(t) + q5(t) + q2(t) + q3(t))
- 2.7105054312 10-20 cos(-q5(t) - q4(t) + q2(t) + q3(t)) + 2.7105054312 10-20 cos(q5(t)
- q4(t) + q2(t) + q3(t)) + 5.2041704279 10-18 cos(q2(t)) + 2.7105054312 10-20 cos(
-q5(t) + q4(t) + q2(t) + q3(t)) - 2.7105054312 10-20 cos(q5(t) + q4(t) + q2(t) + q3(t))
+ 1.7347234760 10-18 cos(q2(t) + q3(t)) + 2.7105054312 10-20 sin(-2 q6(t) + q5(t)
+ 2 q4(t) + q2(t) + q3(t)) + 5.4210108624 10-20 cos(-q5(t) + q4(t) + q2(t))
- 5.4210108624 10-20 cos(q5(t) + q4(t) + q2(t)) - 1.5178830415 10-18 cos(q2(t)
- 2 q3(t)) - 3.6862873865 10-18 cos(2 q3(t) + q2(t)) - 1.7347234760 10-18 cos(2 q4(t)
+ q2(t) + q3(t)) + 2.7105054312 10-20 sin(-2 q6(t) - q5(t) + q2(t) + q3(t))
- 2.7105054312 10-20 sin(2 q6(t) - q5(t) + q2(t) + q3(t)) + 2.7105054312 10-20 sin(
-2 q6(t) + q5(t) + q2(t) + q3(t)) - 2.7105054312 10-20 sin(2 q6(t) + q5(t) + q2(t)
+ q3(t)) - 6.9388939039 10-18 sin(-2 q4(t) + q2(t) + q3(t))
+ 6.9388939039 10-18 sin(2 q4(t) + q2(t) + q3(t)) + 2.7105054312 10-20 sin(-2 q6(t)
+ 2 q4(t) - q5(t) + q2(t) + q3(t)) - 3.4694469520 10-18 cos(2 q3(t) - 2 q4(t) + q2(t))
- 3.4694469520 10-18 cos(2 q4(t) + q2(t)) - 6.9388939039 10-18 cos(-2 q4(t) + q2(t))
- 5.4210108624 10-20 cos(-q5(t) - q4(t) + q2(t))

```

```

> sm13 := m13 - m31

```

```

sm13 := -2.7105054312 10-20 cos(-q5(t) - q4(t) + q2(t) + q3(t))
+ 2.7105054312 10-20 cos(q5(t) - q4(t) + q2(t) + q3(t)) + 2.7105054312 10-20 cos(-q5(t)
+ q4(t) + q2(t) + q3(t)) - 2.7105054312 10-20 cos(q5(t) + q4(t) + q2(t) + q3(t))
- 1.7347234760 10-18 cos(2 q4(t) + q2(t) + q3(t)) + 1.7347234760 10-18 cos(-2 q4(t)
+ q2(t) + q3(t)) + 5.4210108624 10-20 sin(-2 q6(t) - q5(t) + q2(t) + q3(t))
- 5.4210108624 10-20 sin(2 q6(t) - q5(t) + q2(t) + q3(t)) + 5.4210108624 10-20 sin(
-2 q6(t) + q5(t) + q2(t) + q3(t)) - 5.4210108624 10-20 sin(2 q6(t) + q5(t) + q2(t)
+ q3(t)) - 6.9388939039 10-18 sin(-2 q4(t) + q2(t) + q3(t))

```

$$+ 6.9388939039 \cdot 10^{-18} \sin(2 q_4(t) + q_2(t) + q_3(t))$$

> $sm_{23} := m_{23} - m_{32}$

$$sm_{23} := -1.0842021725 \cdot 10^{-19} \sin(q_5(t) + q_3(t)) - 5.4210108624 \cdot 10^{-20} \sin(q_5(t) + q_4(t) + q_3(t)) + 5.4210108624 \cdot 10^{-20} \sin(-q_5(t) + q_4(t) + q_3(t)) - 5.4210108624 \cdot 10^{-20} \cos(-2 q_6(t) + 2 q_4(t) - q_5(t)) - 5.4210108624 \cdot 10^{-20} \cos(-2 q_6(t) + 2 q_4(t) + q_5(t)) - 1.3877787808 \cdot 10^{-17} \cos(q_3(t)) + 5.4210108624 \cdot 10^{-20} \sin(-q_5(t) - q_4(t) + q_3(t)) - 5.4210108624 \cdot 10^{-20} \sin(q_5(t) - q_4(t) + q_3(t)) - 1.0842021725 \cdot 10^{-19} \sin(-q_5(t) + q_3(t)) + 5.4210108624 \cdot 10^{-20} \cos(2 q_6(t) + q_5(t)) + 5.4210108624 \cdot 10^{-20} \cos(-2 q_6(t) + q_5(t))$$

> # Ilustración: Valor del elemento $M_0(1,1)$

> $M_0(1, 1)$

$$\begin{aligned} & -0.0000562500 \cos(2 q_6(t) - 2 q_5(t) + 2 q_4(t)) - 0.0000562500 \cos(2 q_6(t) + 2 q_5(t) + 2 q_4(t)) \\ & - 0.0002250000 \cos(2 q_6(t) + 2 q_4(t) + q_5(t)) - 0.0002250000 \cos(2 q_6(t) + 2 q_4(t) - q_5(t)) \\ & - 0.0003375000 \cos(2 q_6(t) + 2 q_4(t)) - 0.0002425500 \sin(q_5(t) + q_4(t)) + 0.0002425500 \sin(-q_5(t) + q_4(t)) \\ & - 0.0009355500 \sin(q_5(t) + q_3(t)) + 0.0010464300 \cos(q_5(t) + 2 q_2(t) + 2 q_3(t)) + 0.0002425500 \sin(q_5(t) + 2 q_2(t) + 2 q_3(t)) \\ & + 0.0001212750 \sin(q_5(t) + q_4(t) + 2 q_2(t) + 2 q_3(t)) - 0.0001212750 \sin(-q_5(t) + q_4(t) + 2 q_2(t) + 2 q_3(t)) \\ & - 0.0004677750 \sin(q_5(t) + q_4(t) + q_3(t)) + 0.0004677750 \sin(-q_5(t) + q_4(t) + q_3(t)) - 0.0005232150 \cos(-q_5(t) + q_4(t) + 2 q_2(t) + 2 q_3(t)) \\ & + 0.0005232150 \cos(q_5(t) + q_4(t) + 2 q_2(t) + 2 q_3(t)) - 0.0001687500 \cos(2 q_6(t) + 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) \\ & - 0.0001125000 \cos(-2 q_6(t) - 2 q_4(t) + 2 q_2(t) + 2 q_3(t) - q_5(t)) + 0.0001125000 \cos(2 q_6(t) - 2 q_4(t) + 2 q_2(t) + 2 q_3(t) - q_5(t)) \\ & - 0.0001125000 \cos(-2 q_6(t) - 2 q_4(t) + 2 q_2(t) + 2 q_3(t) + q_5(t)) - 0.0001687500 \cos(2 q_6(t) - 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) \\ & - 0.0000281250 \cos(-2 q_6(t) - 2 q_5(t) + 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) + 0.0003375000 \cos(2 q_6(t) + 2 q_2(t) + 2 q_3(t)) \\ & - 0.0000562500 \cos(-2 q_6(t) - 2 q_5(t) + 2 q_4(t)) + 8.6736173799 \cdot 10^{-19} \sin(-2 q_4(t) + 2 q_2(t) + 2 q_3(t)) \\ & + 8.6736173799 \cdot 10^{-19} \sin(2 q_4(t) + 2 q_2(t) + 2 q_3(t)) + 0.0002250000 \cos(-2 q_6(t) + 2 q_4(t) - q_5(t)) + 0.0002250000 \cos(-2 q_6(t) + 2 q_4(t) + q_5(t)) \\ & - 0.0000562500 \cos(-2 q_6(t) + 2 q_5(t) + 2 q_4(t)) - 0.0001125000 \cos(-2 q_6(t) + 2 q_5(t) + 2 q_2(t) + 2 q_3(t) + q_4(t)) \\ & + 0.0001125000 \cos(-2 q_6(t) - 2 q_5(t) - q_4(t) + 2 q_2(t) + 2 q_3(t)) - 0.0001687500 \cos(-2 q_6(t) - 2 q_5(t) + 2 q_2(t) + 2 q_3(t)) \\ & - 0.0002250000 \cos(2 q_6(t) - q_4(t) + 2 q_2(t) + 2 q_3(t) - q_5(t)) - 0.0002250000 \cos(-2 q_6(t) + q_5(t) + 2 q_2(t) + 2 q_3(t) - q_4(t)) \\ & - 0.0000281250 \cos(2 q_6(t) + 2 q_5(t) + 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) - 0.0000281250 \cos(-2 q_6(t) - 2 q_5(t) - 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) \\ & - 0.0001687500 \cos(-2 q_6(t) + 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) + 0.0001125000 \cos(-2 q_6(t) + 2 q_4(t) + 2 q_2(t) + 2 q_3(t) - q_5(t)) \\ & - 0.0001125000 \cos(2 q_6(t) + 2 q_4(t) + 2 q_2(t) + 2 q_3(t) - q_5(t)) + 0.0001125000 \cos(2 q_6(t) - 2 q_5(t) + 2 q_2(t) + 2 q_3(t)) \end{aligned}$$

$$\begin{aligned}
&+ q_4(t) + 1.3552527156 \cdot 10^{-20} \cos(-2 q_6(t) - q_5(t) + 2 q_2(t) + 2 q_3(t)) \\
&- 0.0003375000 \cos(-2 q_6(t) + 2 q_4(t)) + 0.1047546000 \cos(q_3(t)) \\
&- 0.3160566000 \sin(q_3(t)) - 0.0001212750 \sin(-q_5(t) - q_4(t) + 2 q_2(t) + 2 q_3(t)) \\
&+ 0.0001212750 \sin(q_5(t) - q_4(t) + 2 q_2(t) + 2 q_3(t)) + 0.0004677750 \sin(-q_5(t) - q_4(t) \\
&+ q_3(t)) + 0.0002425500 \sin(-q_5(t) + 2 q_2(t) + 2 q_3(t)) - 0.0004677750 \sin(q_5(t) - q_4(t) \\
&+ q_3(t)) - 0.0009355500 \sin(-q_5(t) + q_3(t)) + 0.3160566000 \sin(q_3(t) + 2 q_2(t)) \\
&- 0.1646543600 \sin(2 q_2(t) + 2 q_3(t)) + 2.7105054312 \cdot 10^{-20} \cos(2 q_6(t) + q_5(t)) \\
&- 0.0002250000 \cos(2 q_6(t)) + 0.0001125000 \cos(-2 q_6(t) + 2 q_5(t)) + 0.0010464300 \cos(- \\
&-q_5(t) + 2 q_2(t) + 2 q_3(t)) - 0.0005232150 \cos(-q_5(t) - q_4(t) + 2 q_2(t) + 2 q_3(t)) \\
&+ 0.0005232150 \cos(q_5(t) - q_4(t) + 2 q_2(t) + 2 q_3(t)) + 2.7105054312 \cdot 10^{-20} \cos(-2 q_6(t) \\
&+ q_5(t)) - 0.0005704262 \cos(2 q_5(t) + 2 q_2(t) + 2 q_3(t) + q_4(t)) \\
&- 0.0005704262 \cos(2 q_5(t) + 2 q_2(t) + 2 q_3(t) - q_4(t)) + 0.0005352131 \cos(-2 q_4(t) \\
&+ 2 q_2(t) + 2 q_3(t)) - 0.0002852131 \cos(2 q_5(t) + 2 q_4(t)) - 0.0008556394 \cos(2 q_5(t) \\
&+ 2 q_2(t) + 2 q_3(t)) - 0.0001426066 \cos(-2 q_5(t) - 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) \\
&- 0.0001426066 \cos(2 q_5(t) - 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) + 0.0005704262 \cos(2 q_5(t)) \\
&+ 0.0010704262 \cos(2 q_4(t)) + 0.0005352131 \cos(2 q_4(t) + 2 q_2(t) + 2 q_3(t)) \\
&- 0.0001426066 \cos(2 q_5(t) + 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) + 0.0005704262 \cos(-2 q_5(t) \\
&+ 2 q_2(t) + 2 q_3(t) - q_4(t)) - 0.0008556394 \cos(-2 q_5(t) + 2 q_2(t) + 2 q_3(t)) \\
&- 0.0001426066 \cos(-2 q_5(t) + 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) + 0.0005704262 \cos(-2 q_5(t) \\
&+ q_4(t) + 2 q_2(t) + 2 q_3(t)) - 0.0002852131 \cos(-2 q_5(t) + 2 q_4(t)) \\
&+ 1.3552527156 \cdot 10^{-20} \cos(-2 q_6(t) + q_5(t) + 2 q_2(t) + 2 q_3(t)) \\
&- 0.0000281250 \cos(2 q_6(t) + 2 q_5(t) - 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) + 0.0001125000 \cos(- \\
&-2 q_6(t) - 2 q_5(t) + 2 q_2(t) + 2 q_3(t) + q_4(t)) - 0.0002250000 \cos(2 q_6(t) + q_5(t) \\
&+ 2 q_2(t) + 2 q_3(t) + q_4(t)) - 0.0001125000 \cos(2 q_6(t) + 2 q_5(t) + 2 q_2(t) + 2 q_3(t) \\
&- q_4(t)) + 0.0001125000 \cos(2 q_6(t) - 2 q_5(t) - q_4(t) + 2 q_2(t) + 2 q_3(t)) \\
&+ 0.0001125000 \cos(-2 q_6(t) + q_5(t) + 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) \\
&- 0.0001125000 \cos(2 q_6(t) + q_5(t) + 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) + 0.0002250000 \cos(- \\
&-2 q_6(t) + q_5(t) + 2 q_2(t) + 2 q_3(t) + q_4(t)) - 0.0000281250 \cos(-2 q_6(t) + 2 q_5(t) \\
&+ 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) + 1.3552527156 \cdot 10^{-20} \cos(2 q_6(t) - q_5(t) + 2 q_2(t) \\
&+ 2 q_3(t)) - 0.3974879000 \cos(2 q_2(t)) + 0.0001125000 \cos(2 q_6(t) - 2 q_4(t) + 2 q_2(t) \\
&+ 2 q_3(t) + q_5(t)) - 0.0001687500 \cos(-2 q_6(t) - 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) \\
&+ 0.0002250000 \cos(2 q_6(t) + q_5(t) + 2 q_2(t) + 2 q_3(t) - q_4(t)) \\
&- 0.0001687500 \cos(2 q_6(t) - 2 q_5(t) + 2 q_2(t) + 2 q_3(t)) + 0.0002250000 \cos(-2 q_6(t) \\
&- q_5(t) + 2 q_2(t) + 2 q_3(t) - q_4(t)) - 0.0002250000 \cos(-2 q_6(t) - q_5(t) + 2 q_2(t) \\
&+ 2 q_3(t) + q_4(t)) - 0.0001125000 \cos(2 q_6(t) + 2 q_5(t) + 2 q_2(t) + 2 q_3(t) + q_4(t)) \\
&- 0.0001687500 \cos(2 q_6(t) + 2 q_5(t) + 2 q_2(t) + 2 q_3(t)) + 0.0003375000 \cos(-2 q_6(t) \\
&+ 2 q_2(t) + 2 q_3(t)) + 0.0002250000 \cos(2 q_6(t) - q_5(t) + 2 q_2(t) + 2 q_3(t) + q_4(t)) \\
&- 0.0001125000 \cos(-2 q_6(t) + 2 q_5(t) + 2 q_2(t) + 2 q_3(t) - q_4(t)) - 0.0001687500 \cos(- \\
&-2 q_6(t) + 2 q_5(t) + 2 q_2(t) + 2 q_3(t)) - 0.0000281250 \cos(2 q_6(t) - 2 q_5(t) - 2 q_4(t) \\
&+ 2 q_2(t) + 2 q_3(t)) - 0.0000281250 \cos(-2 q_6(t) + 2 q_5(t) - 2 q_4(t) + 2 q_2(t) + 2 q_3(t)) \\
&- 0.0000281250 \cos(2 q_6(t) - 2 q_5(t) + 2 q_4(t) + 2 q_2(t) + 2 q_3(t))
\end{aligned}$$

$$\begin{aligned}
&+ 1.3552527156 \cdot 10^{-20} \cos(2 q_6(t) + q_5(t) + 2 q_2(t) + 2 q_3(t)) - 0.1047546000 \cos(q_3(t) \\
&+ 2 q_2(t)) + 0.1874085738 \cos(2 q_2(t) + 2 q_3(t)) + 0.0001125000 \cos(2 q_6(t) + 2 q_5(t)) \\
&+ 0.0020928600 \cos(q_5(t)) - 0.0004677750 \sin(-q_5(t) - q_4(t) + q_3(t) + 2 q_2(t)) \\
&+ 0.0009355500 \sin(-q_5(t) + q_3(t) + 2 q_2(t)) + 0.0004677750 \sin(q_5(t) - q_4(t) + q_3(t) \\
&+ 2 q_2(t)) + 0.0004677750 \sin(q_5(t) + q_4(t) + q_3(t) + 2 q_2(t)) - 0.0004677750 \sin(-q_5(t) \\
&+ q_4(t) + q_3(t) + 2 q_2(t)) + 0.0009355500 \sin(q_5(t) + q_3(t) + 2 q_2(t)) + 0.8724064888
\end{aligned}$$



Bibliografía

- [1] ABB. *Especificaciones del producto IRB 120*. ID de documento: 3HAC035960-005. Revisión: T, 2019.
- [2] ABB. *Ficha técnica del producto IRB 120*. ROBO149EN_D Rev: J. 2019.
- [3] Arakelian, V. (ed.), *Dynamic Decoupling of Robot Manipulators*, in Mechanisms and Machine Science, Vol. 56, Springer International Publishing AG, 2018, ISBN 978-3-319-74362-2.
- [4] Barhaghtalab H., Meigoli V., Golbahar Haghghi M. R., Nayeri S. A., Ebrahimi A., *Dynamic analysis, simulation, and control of a 6-DOF IRB-120 robot manipulator using sliding mode control and boundary layer method*, Journal of Central South University, 2018, 25(9), pp. 2219-2244. DOI: <https://doi.org/10.1007/s11771-018-3909-2>.
- [5] Barrientos A., Peñín L. F., Balaguer C., Aracil R., *Fundamentos de robótica*, 2ª edición, McGraw-Hill Interamericana de España S.A.U., Madrid, 2007, ISBN 978-84-481-5636-7.
- [6] Bonev I. *RoKiSim, (Robot Kinematic Simulation)*. École Supérieure de Technologie de Québec. Québec, Cadana. 2013. <http://www.parallemic.org/RoKiSim.html>.
- [7] Can Bignol, Mustafa; Hakan Akpolat, Zuhtu; Ozmen Koca, Gonca. *Robust Control of a Robot Arm Using an Optimized PID Controller*. Firat University, Elazig, Turquía, 2018.
- [8] Craig John J., *Robótica*, 3ª edición, Pearson Educación, México, 2006, ISBN 970-26-0772-8.
- [9] Corke P., *Robotics, Vision and Control. Fundamental Algorithms in Matlab*, in Springer Tracts in Advanced Robotics, Vol. 73, Springer-Verlag, Berlín, 2011, ISBN 978-3-642-20143-1.
- [10] De Gea, José; Kirchner, Frank. *Modelling and Simulation of Robot Arm Interaction Forces Using Impedance Control*. IFAC Proceedings Volumes. Volumen 41, Issue 2, 2008. Páginas 15589 - 15594.3
- [11] Garrido S., *Identificación, estimación y control de sistemas no lineales mediante RGO*, Tesis Doctoral, Universidad Carlos III de Madrid, Leganés, 1999, ISBN 0-4850.
- [12] Høifødt H., *Dynamic Modeling and Simulation of Robot Manipulators: The Newton-Euler Formulation*, NTNU MSc Thesis, Norwegian University of Science and Technology, 2011.

- [13] Hollerbach J. M., *A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-10(11), pp. 730–736, 1980.
- [14] Jun-Di Sun, Guang-Zhong Cao; Wen-Bo Li, Yu-Xin Liang; Su-Dan Huang. *Analytical Inverse Kinematic Solution Using the D-H Method for a 6-DOF Robot*. Shenzhen Key Laboratory of Electromagnetic Control, Shenzhen University, China. 2017.
- [15] Kelly R., Santibáñez V., Loría A., *Control of Robot Manipulators in Joint Space*, en Advanced Textbooks in Control and Signal Processing, Springer–Verlag London Ltd., 2005, ISBN 978-1-85233-994-4.
- [16] Mato San José M. A. *Simulación, control cinemático y dinámico de robots comerciales usando la herramienta de MATLAB: Robotic Toolbox*, Trabajo Fin de Grado, Universidad de Valladolid, Valladolid, 2014.
- [17] Mato M., Herreros A., Fraile J. C., González J. L., Baeyens E., Pérez-Turiel J., Gayubo F., *Aplicaciones en MATLAB y SIMULINK para el modelado y control del movimiento de una estación ABB IRB 120*, ITAP, Universidad de Valladolid, en Actas de las XXXV Jornadas de Automática, Valencia, 2014, ISBN 978-84-697-0589-6.
- [18] Ogata K., *Ingeniería de control moderna*, 5^a edición, Pearson Educación S. A., Madrid, 2010, ISBN 978-84-8322-660-5.
- [19] Pérez Menéndez, F., *Desarrollo de sistema de control para un manipulador de seis grados de libertad*, Trabajo Fin de Máster, Universidad de Oviedo, 2014.
- [20] Pérez Cisneros, M.A.; Cuevas Jimenez, E.V.; Zaldivar Navarro, D. *Fundamentos de Robótica y mecatrónica con MATLAB y SIMULINK*. Editorial RA-MA. 2014. Editorial RA-MA.
- [21] Siciliano B., Sciavicco L., Villani L., Oriolo G., *Robotics. Modelling, Planning and Control*, en Advanced Textbooks in Control and Signal Processing, Springer–Verlag London Ltd., 2010, ISBN 978-1-84628-641-4.
- [22] Spong M. W., Vidyasagar M., *Robot Dynamics and Control*, John Wiley & Sons, Inc., New York, 1989, ISBN 0-471-61243-X.
- [23] Vasilyev, I. A.; Lyashin, A. M. *Analytical Solution to Inverse Kinematic Problem for 6-DOF Robot-Manipulator*. Central Research Institute of Robot Engineering and Engineering Cybernetics, St. Petersburg, Russia. 2008.